1.2.3 Masking Bad Pixels

smalldata_tools will pass on an unmasked data array throughout the different functions, along with two masks that are also saved in the userDataCfg datasets in the hdf5 file.

- mask will be the status mask from the pedestal production , including masked edgepixels
- cmask will add a potential user mask as saved in pixel_mask

detFuncObjects will use the mask along with the data to derived the 'reduced/transformed' dataset (e.g. projections of an ROI, droplet or photon finding etc).

smalldata_tools provides an interface to create a user mask as described here. Among other options, you can tighten the cut used to mask bad pixels based on their pedestal or noise from the dark run. As we use the status mask as a base, you cannot loosen cuts this way but would need to rerun the pedestal creating step.

Creating mask file

```
SDAna In: anaps.MakeMask()
```

This will by draw the image and let you define a mask using rectangles, circles and polygons defined by mouse clicks. You can add several of these shapes. This mask can be saved locally or in the calib directory where it will be available for the littleData creation. Masks are e.g. used for the azimuthal integration. A session where a mask consisting of the inner circle and a polygon was developed is shown below. For each part of the mask, you can either use a circle, rectangle or polygon. The mask will then be shown to you. Once you have defined all the sub-masks, the total mask is calculated and applied to the image. If so desired, the mask can then be stored either locally or in the calibdirectory (where the littleData creating will pick it up automatically).

Other options are masking a number of edge-pixels (sometimes more than a single row of pixels should be masked). You can also mask pixels based on the calibration status (either the pedestal or the noise value).

Mask - circle/donut shape & polygon

```
SDana In [3]: anaps.MakeMask()
(32, 185, 388)
plot AvImg_cspad using the 5/99.5 percentiles as plot min/max: (0.199616, 263.383)
rectangle(r-click, R-enter), circle(c), polygon(p), dark(d), noise(n) or edgepixels(e)?:
c
Select center by mouse?
y
/reg/g/psdm/sw/releases/ana-0.17.30/arch/x86_64-rhel7-gcc48-opt/python/matplotlib/backend_bases.py:2399:
MatplotlibDeprecationWarning: Using default event loop until function specific to this GUI is implemented
  warnings.warn(str, mplDeprecation)
center:  98067.8500159   87780.7509265
Select outer radius by mouse?
y
Select inner radius by mouse?
n
radii:  6044.641774   0
mask from circle (shape): (32, 185, 388)
Done?
n
created a mask....
masked now:  9506.0
masked tot:  9506.0
rectangle(r), circle(c) or polygon(p)?:
p
Number of Points (-1 until right click)?
5
[[  81561.33638984   90357.32502125]
 [  75822.06188735   95379.19021093]
 [  80126.51776422  101477.16936983]
 [  86941.90623593  100401.05540061]
 [  86583.20157952   92509.55295969]]
mask from polygon (shape): (32, 185, 388)
not implemented yet....
Done?
y
created a mask....
masked now:  5649.0
masked tot:  15155.0
Save to calibdir?
n
Save to local?
y
Invert?
n
SDana Out[3]:
array([[1, 1, 1, ..., 1, 1, 1],
       [1, 1, 1, ..., 1, 1, 1],
       [1, 1, 1, ..., 1, 1, 1],
       ...,
       [1, 1, 1, ..., 1, 1, 1],
       [1, 1, 1, ..., 1, 1, 1],
       [1, 1, 1, ..., 1, 1, 1]])
```

Mask - noise/std from dark run

This can also be used to make masks with tighter cuts on e.g. the pedestal value and/or the noise/std from the used dark run:

```
SDAna In [2]: anaps.MakeMask()
plot AvImg_median_ePix100_1 using the 5/99.5 percentiles as plot min/max: (-0.0800912, 0.131976)
rectangle(r-click, R-enter), circle(c), polygon(p), dark(d), noise(n) or edgepixels(e)?:
n
/sdf/data/lcls/ds/mfx/mfxl1010021/results/smalldata_tools/smalldata_tools/BaseSmallDataAna_psana.py:975:
RuntimeWarning: divide by zero encountered in log
  plt.subplot(gsPed[1]).plot(hstPed[1][:-1],np.log(hstPed[0]),'o')
Enter allowed pedestal range (min max)1.5 7
created a mask.... 0
masked in this step:  488.0
masked up to this step:  488.0
masked tot:  488.0
Add this mask?
y
Done?
y
Invert [y/n]? (n/no inversion: masked pixels will get rejected)?
n
Save to calibdir?
y
save mask in /sdf/data/lcls/ds//mfx/mfxl1010021/calib/Epix100a::CalibV1/MfxEndstation.0:Epix100a.0/pixel_mask/
as 18-end.data
```