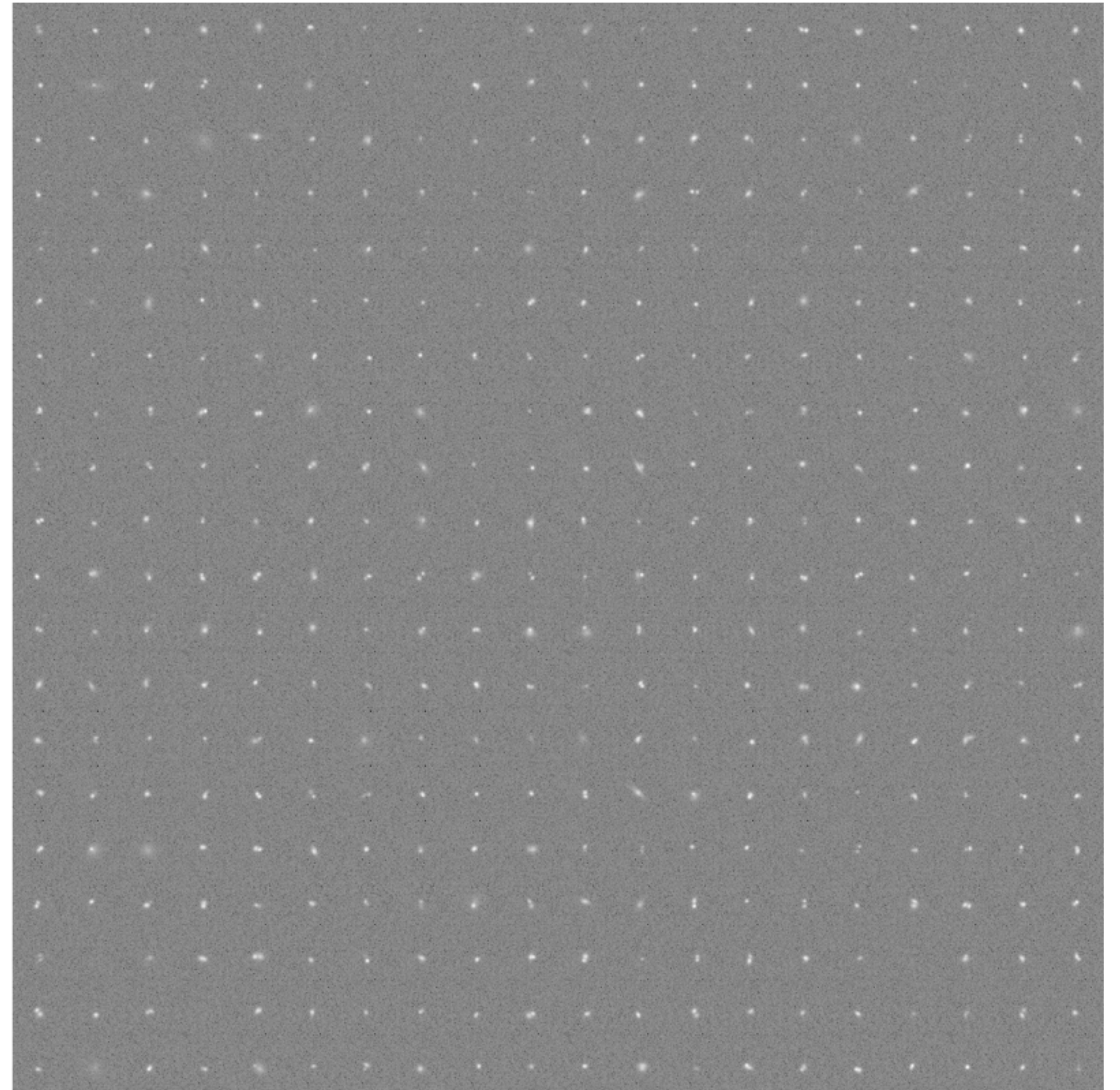


# Running DM on a GalSim Image

Will Dawson, Perry Gee, Michael Schneider

# Simulated blends (20x20 grid)

- No background
- No WCS



# Script to run DM on GalSim

```
41 def runProcess(fitsFile):
42
43     exposure = afwImage.ExposureF.readFits(fitsFile)
44     kernel = afwMath.FixedKernel(afwImage.ImageD.readFits("psf_fixed.fits"))
45     exposure.setPsf(measAlg.KernelPsf(kernel))
46     CD = numpy.array([[5.55E-5, 0.0], [0.0, 5.55E-5]])
47     crpix = afwGeom.Point2D(0.0,0.0)
48     crval = afwGeom.Point2D(0.0,0.0)
49     exposure.setWcs(afwImage.Wcs(crval, crpix, CD))
50     exposure.getMaskedImage().getVariance().set(100)
51     config = ProcessCcdTask.ConfigClass()
52     config.doCalibrate = False
53     config.doDetection = True
54     config.doDeblend = True
55     config.doMeasurement = True
56     config.doWriteCalibrate = False
57     config.persistBackgroundModel = False
58     config.doWriteCalibrateMatches = False
59     config.doWriteSources = False
60     config.doWriteSourceMatches = False
61     config.doWriteHeavyFootprintsInSources = False
62
63     config.measurement.slots.centroid = "base_GaussianCentroid"
64     config.measurement.slots.shape = None
65     config.measurement.slots.psfFlux = "base_PsfFlux"
66     config.measurement.slots.apFlux = None
67     config.measurement.slots.instFlux = None
68     config.measurement.slots.modelFlux = None
69     config.measurement.doReplaceWithNoise = False
70     config.measurement.doApplyApCorr = "no"
71     config.measurement.plugins.names = ["base_GaussianCentroid", "base_PsfFlux"]
72     processTask = ProcessCcdTask(name="xyzy", config=config)
73     result = processTask.process(None, exposure)
74     result.sources.writeFits("sources.fits")
75     result.exposure.writeFits("exposure.fits")
76
77 if __name__ == "__main__":
78
79     parser = argparse.ArgumentParser()
80     parser.add_argument("fitsFile", type=str, help="Name of file to process",
81                         *
82                         default=None)
81     args = parser.parse_args()
82     runProcess(args.fitsFile)
```

# Script to run DM on GalSim: DM packages

```
41 def runProcess(fitsFile):
42
43     exposure = afwImage.ExposureF.readFits(fitsFile)
44     kernel = afwMath.FixedKernel(afwImage.ImageD.readFits("psf_fixed.fits"))
45     exposure.setPsf(measAlg.KernelPsf(kernel))
46     CD = numpy.array([[5.55E-5, 0.0], [0.0, 5.55E-5]])
47     crpix = afwGeom.Point2D(0.0,0.0)
48     crval = afwGeom.Point2D(0.0,0.0)
49     exposure.setWcs(afwImage.Wcs(crval, crpix, CD))
50     exposure.getMaskedImage().getVariance().set(100)
51     config = ProcessCcdTask.ConfigClass()
52     config.doCalibrate = False
53     config.doDetection = True
54     config.doDeblend = True
55     config.doMeasurement = True
56     config.doWriteCalibrate = False
57     config.persistBackgroundModel = False
58     config.doWriteCalibrateMatches = False
59     config.doWriteSources = False
60     config.doWriteSourceMatches = False
61     config.doWriteHeavyFootprintsInSources = False
62
63     config.measurement.slots.centroid = "base_GaussianCentroid"
64     config.measurement.slots.shape = None
65     config.measurement.slots.psfFlux = "base_PsfFlux"
66     config.measurement.slots.apFlux = None
67     config.measurement.slots.instFlux = None
68     config.measurement.slots.modelFlux = None
69     config.measurement.doReplaceWithNoise = False
70     config.measurement.doApplyApCorr = "no"
71     config.measurement.plugins.names = ["base_GaussianCentroid", "base_PsfFlux"]
72     processTask = ProcessCcdTask(name="xyzyz", config=config)
73     result = processTask.process(None, exposure)
74     result.sources.writeFits("sources.fits")
75     result.exposure.writeFits("exposure.fits")
76
77 if __name__ == "__main__":
78
79     parser = argparse.ArgumentParser()
80     parser.add_argument("fitsFile", type=str, help="Name of file to process",
81                         *
82                         default=None)
81     args = parser.parse_args()
82     runProcess(args.fitsFile)
```

```
31 import lsst.meas.base as measBase
32 import lsst.afw.geom as afwGeom
33 import lsst.afw.math as afwMath
34 import lsst.afw.table as afwTable
35 import lsst.afw.image as afwImage
36 import lsst.meas.algorithms as measAlg
37 from lsst.pipe.tasks.processCcd import ProcessCcdTask
```

# Script to run DM on GalSim: “formatting”

```
41 def runProcess(fitsFile):
42     exposure = afwImage.ExposureF.readFits(fitsFile)
43     kernel = afwMath.FixedKernel(afwImage.ImageD.readFits("psf_fixed.fits"))
44     exposure.setPsf(measAlg.KernelPsf(kernel))
45     CD = numpy.array([[5.55E-5, 0.0], [0.0, 5.55E-5]])
46     crpix = afwGeom.Point2D(0.0,0.0)
47     crval = afwGeom.Point2D(0.0,0.0)
48     exposure.setWcs(afwImage.Wcs(crval, crpix, CD))
49     exposure.getMaskedImage().getVariance().set(100)
50
51     config = ProcessCcdTask.ConfigClass()
52     config.doCalibrate = False
53     config.doDetection = True
54     config.doDeblend = True
55     config.doMeasurement = True
56     config.doWriteCalibrate = False
57     config.persistBackgroundModel = False
58     config.doWriteCalibrateMatches = False
59     config.doWriteSources = False
60     config.doWriteSourceMatches = False
61     config.doWriteHeavyFootprintsInSources = False
62
63     config.measurement.slots.centroid = "base_GaussianCentroid"
64     config.measurement.slots.shape = None
65     config.measurement.slots.psfFlux = "base_PsfFlux"
66     config.measurement.slots.apFlux = None
67     config.measurement.slots.instFlux = None
68     config.measurement.slots.modelFlux = None
69     config.measurement.doReplaceWithNoise = False
70     config.measurement.doApplyApCorr = "no"
71     config.measurement.plugins.names = ["base_GaussianCentroid", "base_PsfFlux"]
72     processTask = ProcessCcdTask(name="xyzyz", config=config)
73     result = processTask.process(None, exposure)
74     result.sources.writeFits("sources.fits")
75     result.exposure.writeFits("exposure.fits")
76
77 if __name__ == "__main__":
78
79     parser = argparse.ArgumentParser()
80     parser.add_argument("fitsFile", type=str, help="Name of file to process",
81                         *
82                         default=None)
81     args = parser.parse_args()
82     runProcess(args.fitsFile)
```

```
41 def runProcess(fitsFile):
42
43     exposure = afwImage.ExposureF.readFits(fitsFile)
44     kernel = afwMath.FixedKernel(afwImage.ImageD.readFits("psf_fixed.fits"))
45     exposure.setPsf(measAlg.KernelPsf(kernel))
46     CD = numpy.array([[5.55E-5, 0.0], [0.0, 5.55E-5]])
47     crpix = afwGeom.Point2D(0.0,0.0)
48     crval = afwGeom.Point2D(0.0,0.0)
49     exposure.setWcs(afwImage.Wcs(crval, crpix, CD))
50     exposure.getMaskedImage().getVariance().set(100)
```

# Script to run DM on GalSim: configuring DM

```
41 def runProcess(fitsFile):
42     exposure = afwImage.ExposureF.readFits(fitsFile)
43     kernel = afwMath.FixedKernel(afwImage.ImageD.readFits("psf_fixed.fits"))
44     exposure.setPsf(measAlg.KernelPsf(kernel))
45     CD = numpy.array([[5.55E-5, 0.0], [0.0, 5.55E-5]])
46     crpix = afwGeom.Point2D(0.0,0.0)
47     crval = afwGeom.Point2D(0.0,0.0)
48     exposure.setWcs(afwImage.Wcs(crval, crpix, CD))
49     exposure.getMaskedImage().getVariance().set(100)
50
51     config = ProcessCcdTask.ConfigClass()
52     config.doCalibrate = False
53     config.doDetection = True
54     config.doDeblend = True
55     config.doMeasurement = True
56     config.doWriteCalibrate = False
57     config.persistBackgroundModel = False
58     config.doWriteCalibrateMatches = False
59     config.doWriteSources = False
60     config.doWriteSourceMatches = False
61     config.doWriteHeavyFootprintsInSources = False
62
63     config.measurement.slots.centroid = "base_GaussianCentroid"
64     config.measurement.slots.shape = None
65     config.measurement.slots.psfFlux = "base_PsfFlux"
66     config.measurement.slots.apFlux = None
67     config.measurement.slots.instFlux = None
68     config.measurement.slots.modelFlux = None
69     config.measurement.doReplaceWithNoise = False
70     config.measurement.doApplyApCorr = "no"
71     config.measurement.plugins.names = ["base_GaussianCentroid", "base_PsfFlux"]
72     processTask = ProcessCcdTask(name="xyzyzy", config=config)
73     result = processTask.process(None, exposure)
74     result.sources.writeFits("sources.fits")
75     result.exposure.writeFits("exposure.fits")
76
77 if __name__ == "__main__":
78
79     parser = argparse.ArgumentParser()
80     parser.add_argument("fitsFile", type=str, help="Name of file to process",
81                         *
82                         default=None)
81     args = parser.parse_args()
82     runProcess(args.fitsFile)
```

```
51     config = ProcessCcdTask.ConfigClass()
52     config.doCalibrate = False
53     config.doDetection = True
54     config.doDeblend = True
55     config.doMeasurement = True
56     config.doWriteCalibrate = False
57     config.persistBackgroundModel = False
58     config.doWriteCalibrateMatches = False
59     config.doWriteSources = False
60     config.doWriteSourceMatches = False
61     config.doWriteHeavyFootprintsInSources = False
```

# Script to run DM on GalSim: defining measurements to be done

```
41 def runProcess(fitsFile):
42
43     exposure = afwImage.ExposureF.readFits(fitsFile)
44     kernel = afwMath.FixedKernel(afwImage.ImageD.readFits("psf_fixed.fits"))
45     exposure.setPsf(measAlg.KernelPsf(kernel))
46     CD = numpy.array([[5.55E-5, 0.0], [0.0, 5.55E-5]])
47     crpix = afwGeom.Point2D(0.0,0.0)
48     crval = afwGeom.Point2D(0.0,0.0)
49     exposure.setWcs(afwImage.Wcs(crval, crpix, CD))
50     exposure.getMaskedImage().getVariance().set(100)
51     config = ProcessCcdTask.ConfigClass()
52     config.doCalibrate = False
53     config.doDetection = True
54     config.doDeblend = True
55     config.doMeasurement = True
56     config.doWriteCalibrate = False
57     config.persistBackgroundModel = False
58     config.doWriteCalibrateMatches = False
59     config.doWriteSources = False
60     config.doWriteSourceMatches = False
61     config.doWriteHeavyFootprintsInSources = False
62
63     config.measurement.slots.centroid = "base_GaussianCentroid"
64     config.measurement.slots.shape = None
65     config.measurement.slots.psfFlux = "base_PsfFlux"
66     config.measurement.slots.apFlux = None
67     config.measurement.slots.instFlux = None
68     config.measurement.slots.modelFlux = None
69     config.measurement.doReplaceWithNoise = False
70     config.measurement.doApplyApCorr = "no"
71     config.measurement.plugins.names = ["base_GaussianCentroid", "base_PsfFlux"]
72
73     processTask = ProcessCcdTask(name="xyzyz", config=config)
74     result = processTask.process(None, exposure)
75     result.sources.writeFits("sources.fits")
76     result.exposure.writeFits("exposure.fits")
77
78 if __name__ == "__main__":
79
80     parser = argparse.ArgumentParser()
81     parser.add_argument("fitsFile", type=str, help="Name of file to process",
82                         default=None)
83     args = parser.parse_args()
84     runProcess(args.fitsFile)
```

```
63     config.measurement.slots.centroid = "base_GaussianCentroid"
64     config.measurement.slots.shape = None
65     config.measurement.slots.psfFlux = "base_PsfFlux"
66     config.measurement.slots.apFlux = None
67     config.measurement.slots.instFlux = None
68     config.measurement.slots.modelFlux = None
69     config.measurement.doReplaceWithNoise = False
70     config.measurement.doApplyApCorr = "no"
71     config.measurement.plugins.names = ["base_GaussianCentroid", "base_PsfFlux"]
```

# Script to run DM on GalSim: run process, output the results

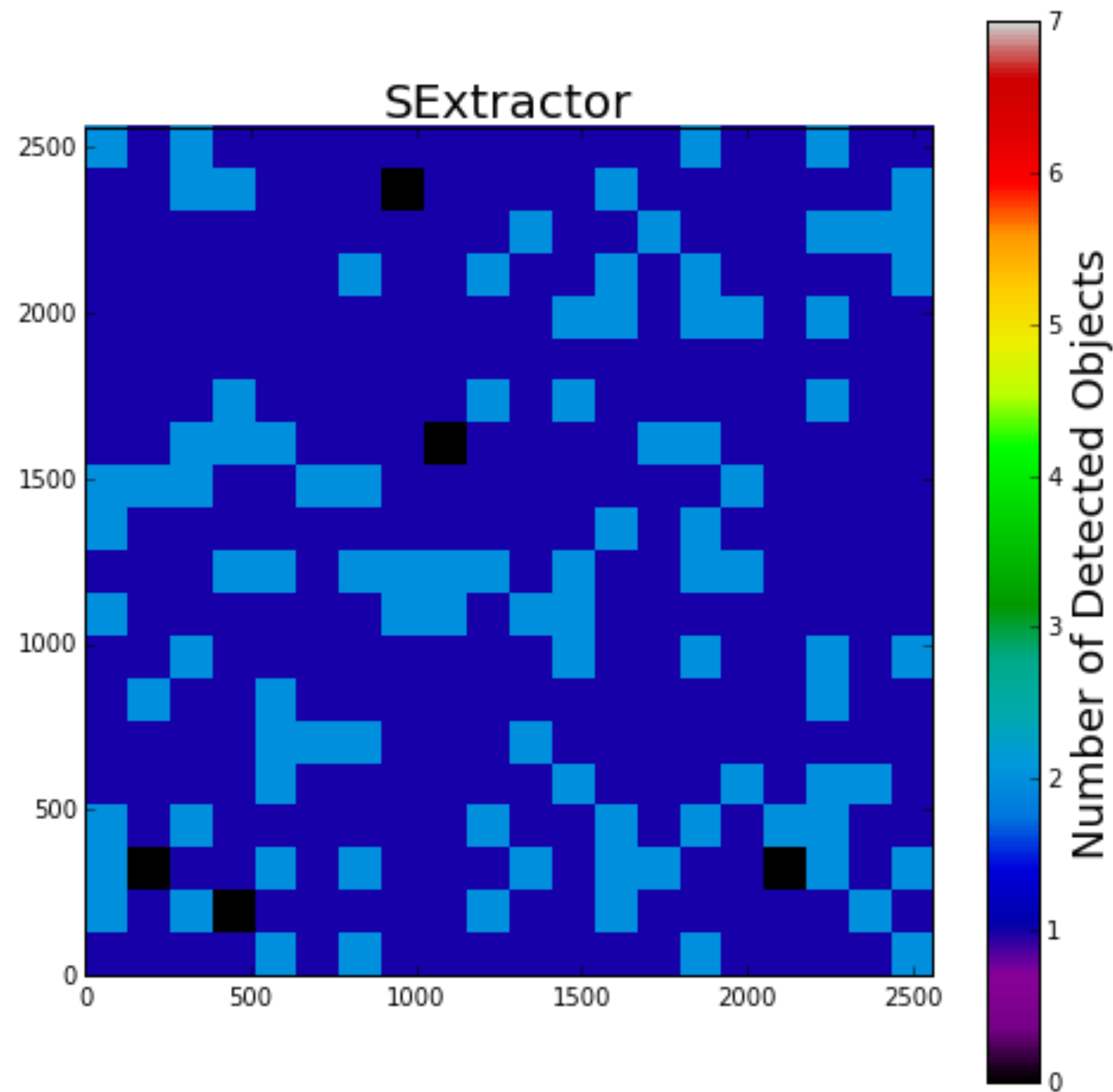
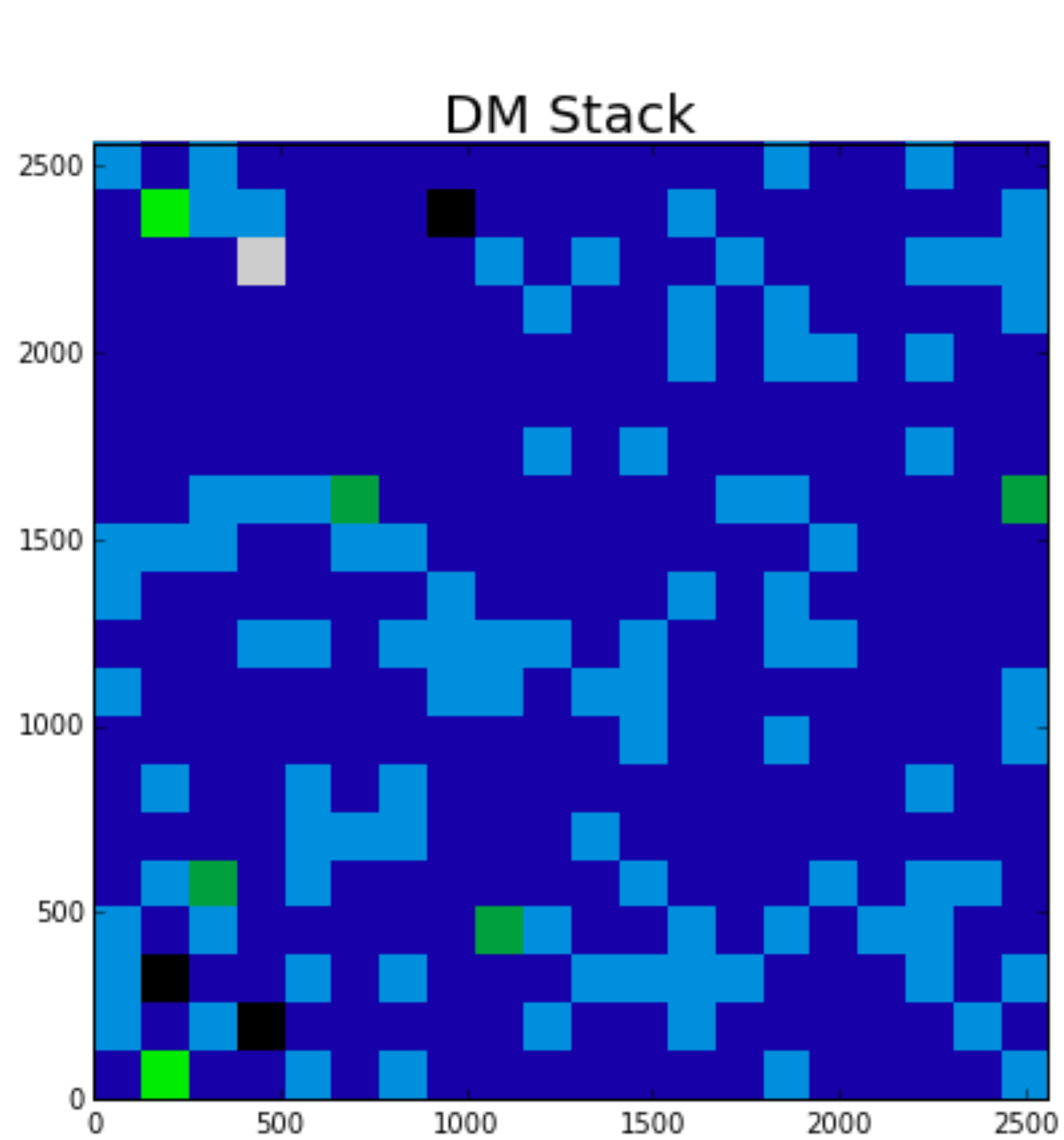
```
41 def runProcess(fitsFile):
42
43     exposure = afwImage.ExposureF.readFits(fitsFile)
44     kernel = afwMath.FixedKernel(afwImage.ImageD.readFits("psf_fixed.fits"))
45     exposure.setPsf(measAlg.KernelPsf(kernel))
46     CD = numpy.array([[5.55E-5, 0.0], [0.0, 5.55E-5]])
47     crpix = afwGeom.Point2D(0.0,0.0)
48     crval = afwGeom.Point2D(0.0,0.0)
49     exposure.setWcs(afwImage.Wcs(crval, crpix, CD))
50     exposure.getMaskedImage().getVariance().set(100)
51     config = ProcessCcdTask.ConfigClass()
52     config.doCalibrate = False
53     config.doDetection = True
54     config.doDeblend = True
55     config.doMeasurement = True
56     config.doWriteCalibrate = False
57     config.persistBackgroundModel = False
58     config.doWriteCalibrateMatches = False
59     config.doWriteSources = False
60     config.doWriteSourceMatches = False
61     config.doWriteHeavyFootprintsInSources = False
62
63     config.measurement.slots.centroid = "base_GaussianCentroid"
64     config.measurement.slots.shape = None
65     config.measurement.slots.psfFlux = "base_PsfFlux"
66     config.measurement.slots.apFlux = None
67     config.measurement.slots.instFlux = None
68     config.measurement.slots.modelFlux = None
69     config.measurement.doReplaceWithNoise = False
70     config.measurement.doApplyApCorr = "no"
71     config.measurement.plugins.names = ["base_GaussianCentroid", "base_PsfFlux"]
72     processTask = ProcessCcdTask(name="xyzyzy", config=config)
73     result = processTask.process(None, exposure)
74     result.sources.writeFits("sources.fits")
75     result.exposure.writeFits("exposure.fits")
76
77 if __name__ == "__main__":
78
79     parser = argparse.ArgumentParser()
80     parser.add_argument("fitsFile", type=str, help="Name of file to process",
81                         *
82                         default=None)
81     args = parser.parse_args()
82     runProcess(args.fitsFile)
```

```
72     processTask = ProcessCcdTask(name="xyzyzy", config=config)
73     result = processTask.process(None, exposure)
74     result.sources.writeFits("sources.fits")
75     result.exposure.writeFits("exposure.fits")
```

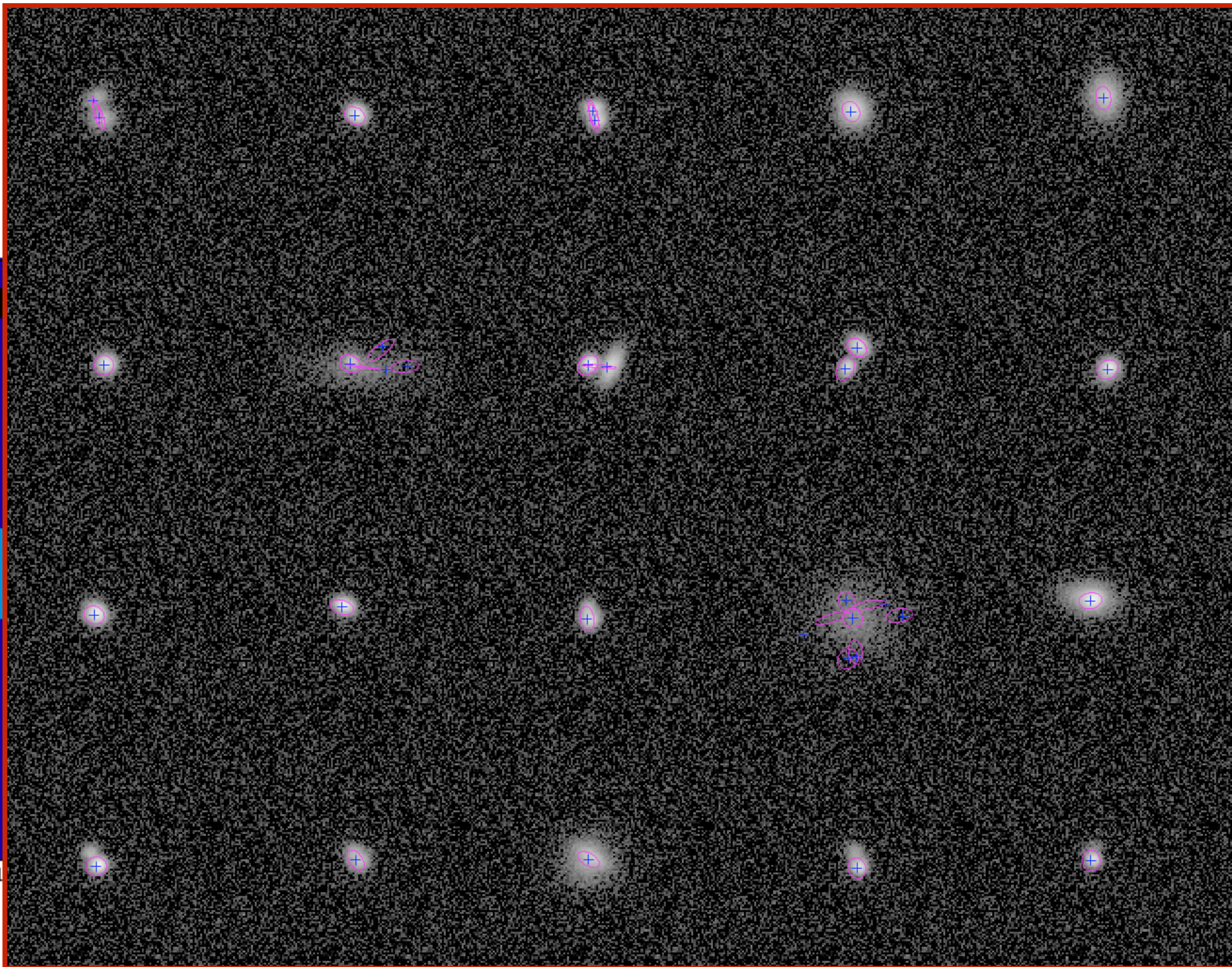
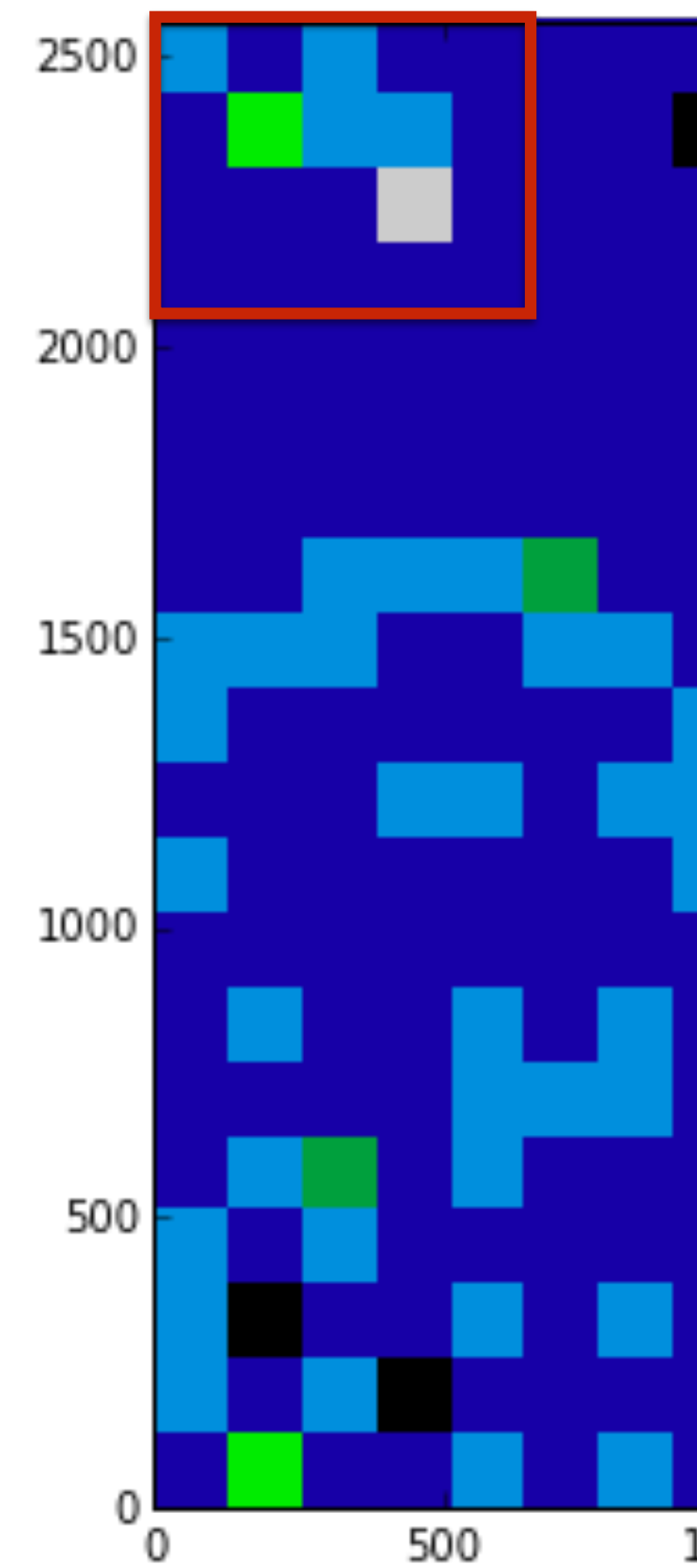




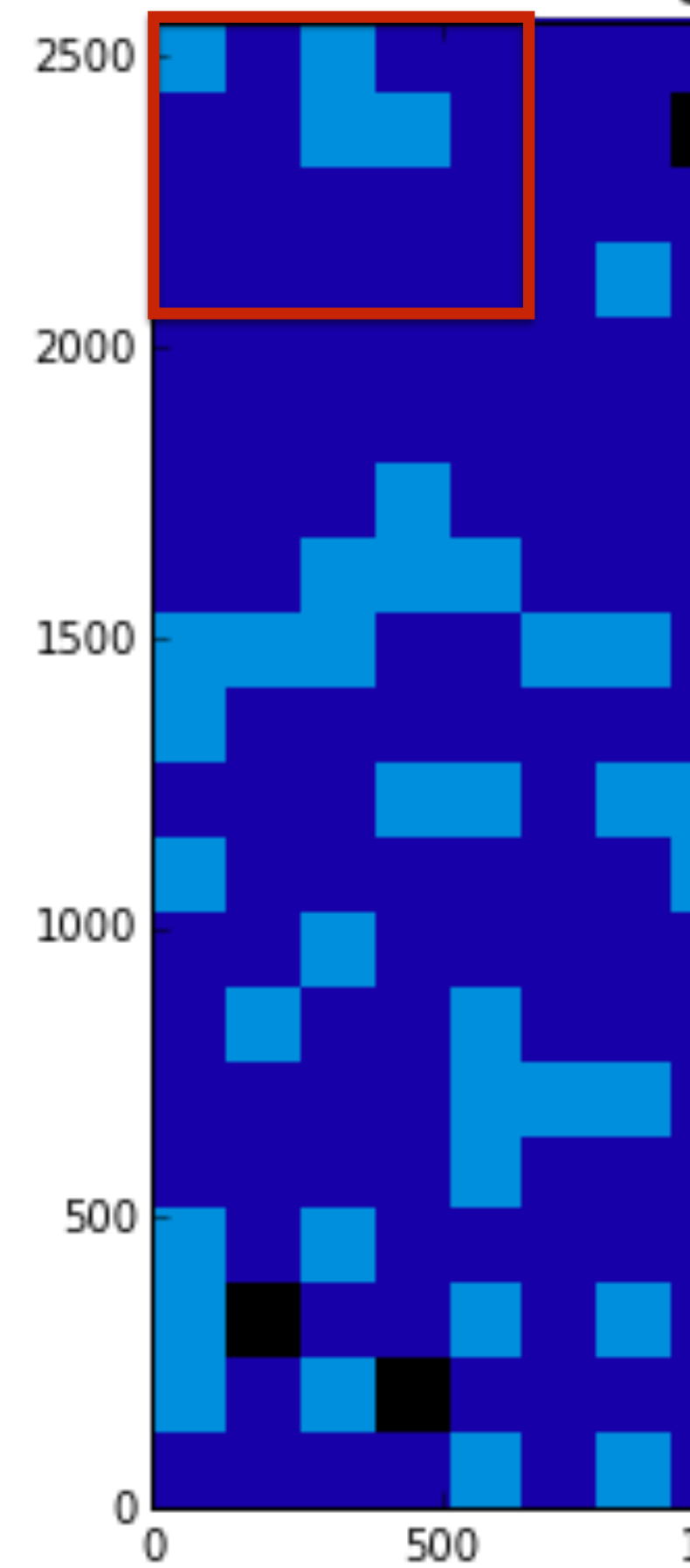
# 2D histogram of detected objects



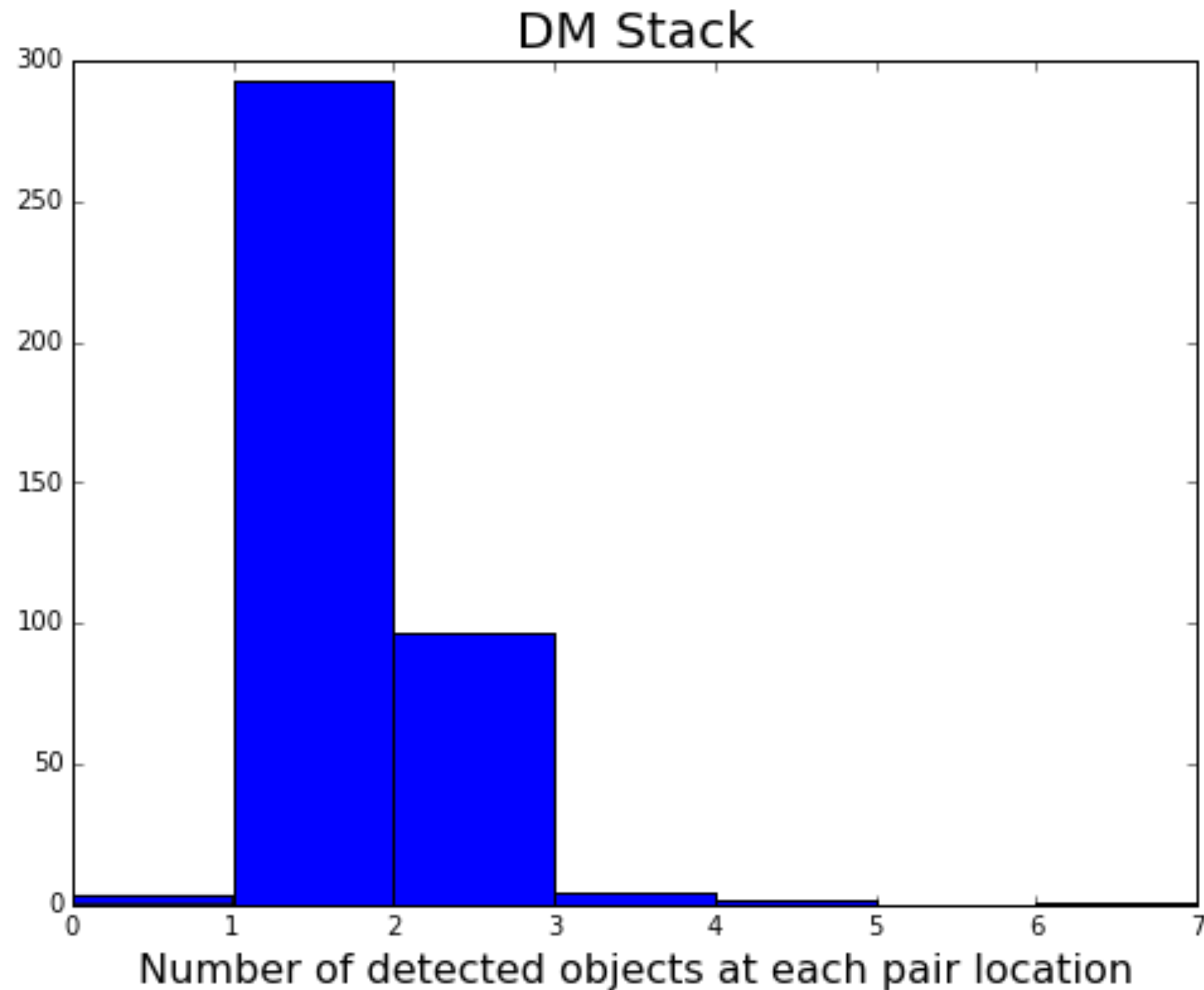
DM Stack



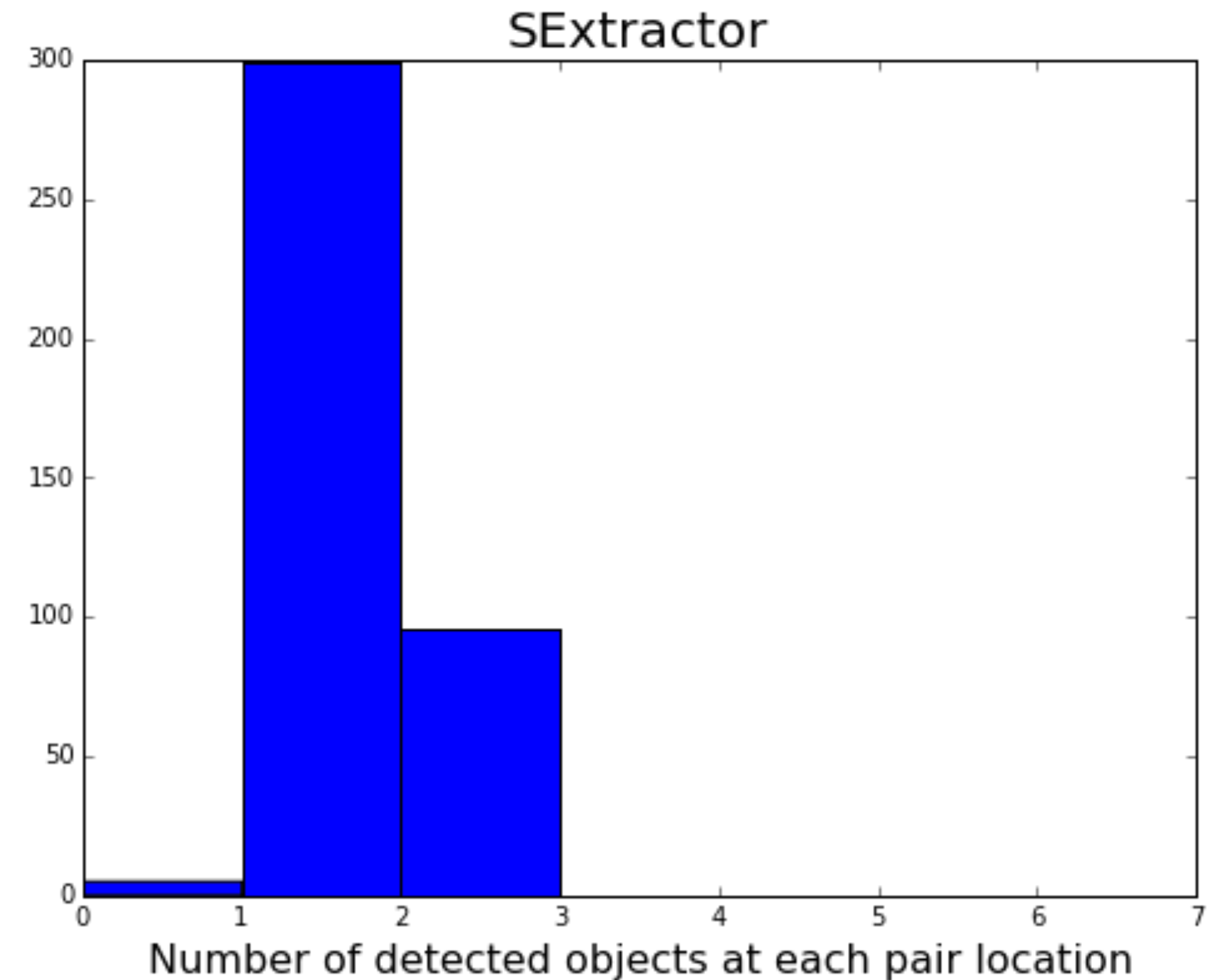
SExtractor



# 1D histogram of detected objects



3 (1%): failed detections  
293 (73%): failed deblend identification  
97 (24%): correctly deblend  
7 (2%): deblended too many sources



5 (1%): failed detections  
299 (75%): failed deblend identification  
96 (24%): correctly deblend  
0 (0%): deblended too many sources