Guidelines for DETECT(s) in L3

- Provide initial seed source lists for later Source Properties steps (2-d fitting)
- Sources may be deleted or added (resolved) in later steps
- Better to allow false sources than reject real ones
- Must run on entire data set
- Must handle variable exposure/edges
- Must be fast (4 bands x 5000 OBSIDS)

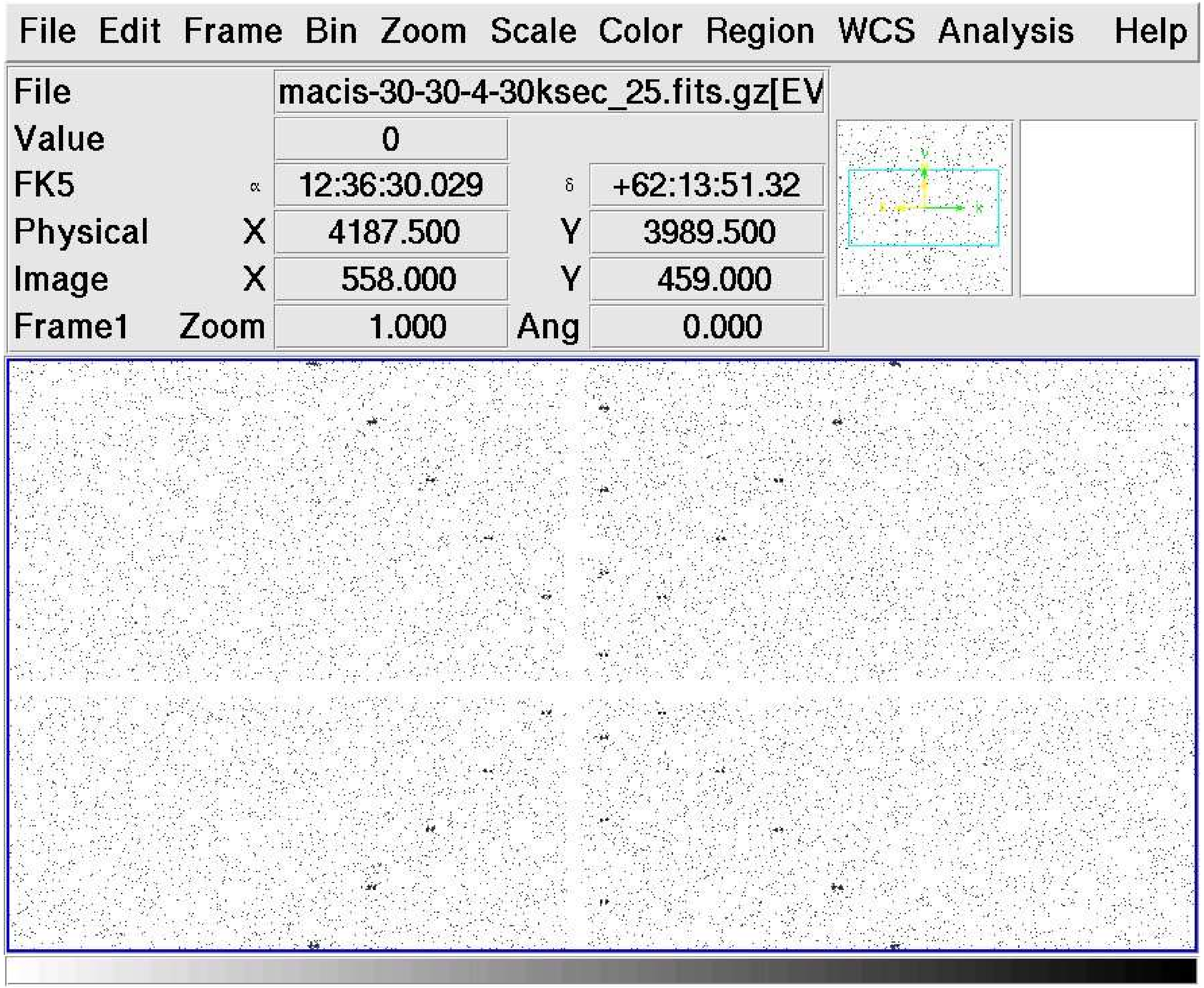
Current DETECT Status in L3 Pipeline

- Current pipeline can run any of 3 detect tools
  - CELLDETECT with “Exposure Masks”
  - SExtractor with exposure masks
  - WAVDETECT with real exposure maps
- All use recursive blocking to handle full dataset,either built-in (CELLDETECT) or via driver scripts (SE, WAVDETECT)
- Multiple source lists merged into one list per band prior to 2-d fitting

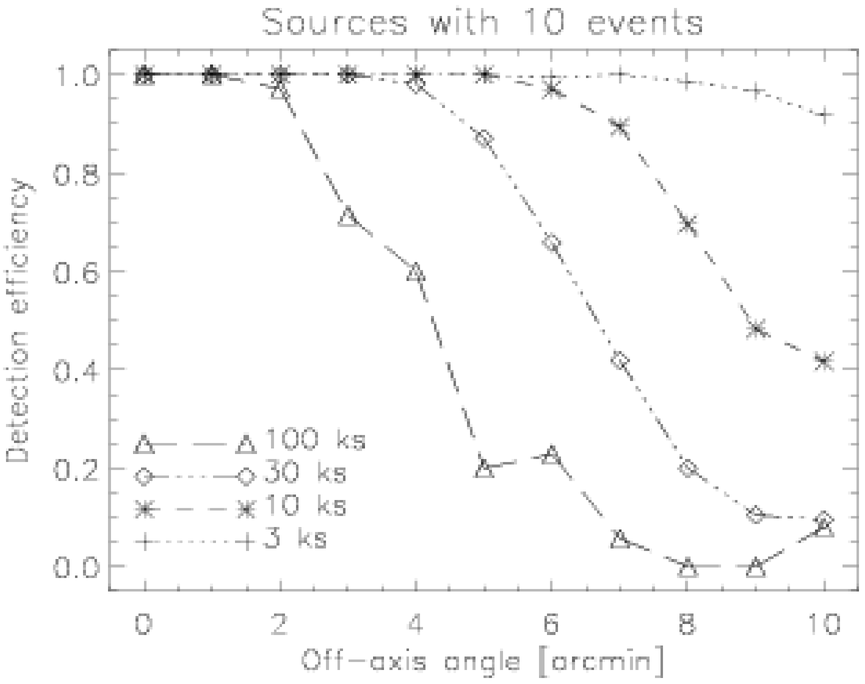
General Sextractor Operation

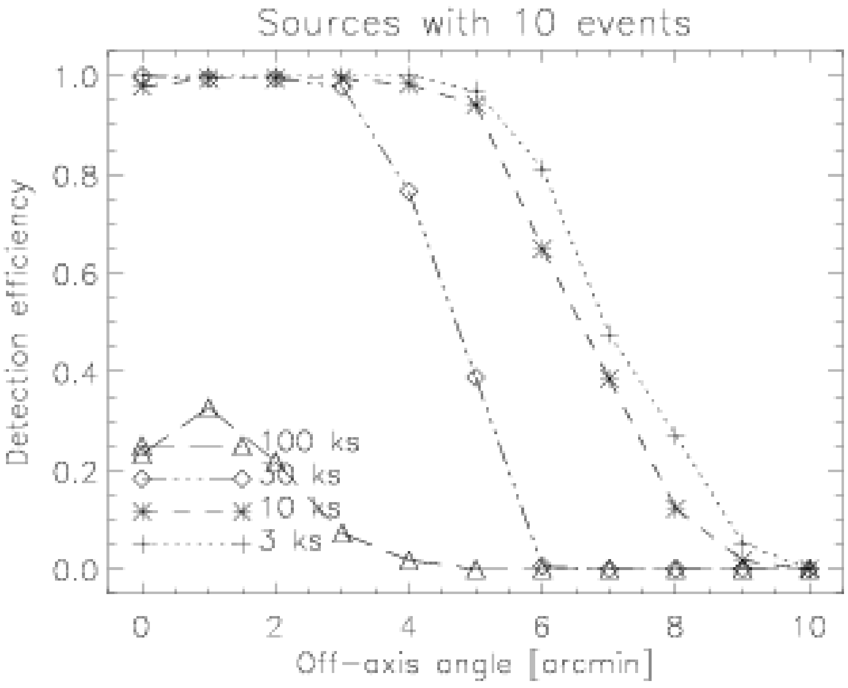
1) Generate Background Map
    a) Divide image into many small (64x64 pixel) “background meshes”
    b) Determine mean and sigma in each mesh using iterative “clipping” to remove source pixels. Median of sigmas used in thresholding.
    c) Background map generated from median-filtered mesh means via bicubic spline interpolation
2) Compute background-subtracted image
    a) Smooth result with convolution kernel that approximates PSF
3) Search background-subtracted, smoothed image for sources
    a) Flag pixels > NxMedian(sigma) above local value of background map
    b) Merge adjacent flagged pixels into source “islands”
    c) Reject islands with fewer than TBD pixels
    d) Examine remaining islands for “saddle points”, indicating multiple sources

Sample ACIS Simulation for Testing WAVDETECT and SE

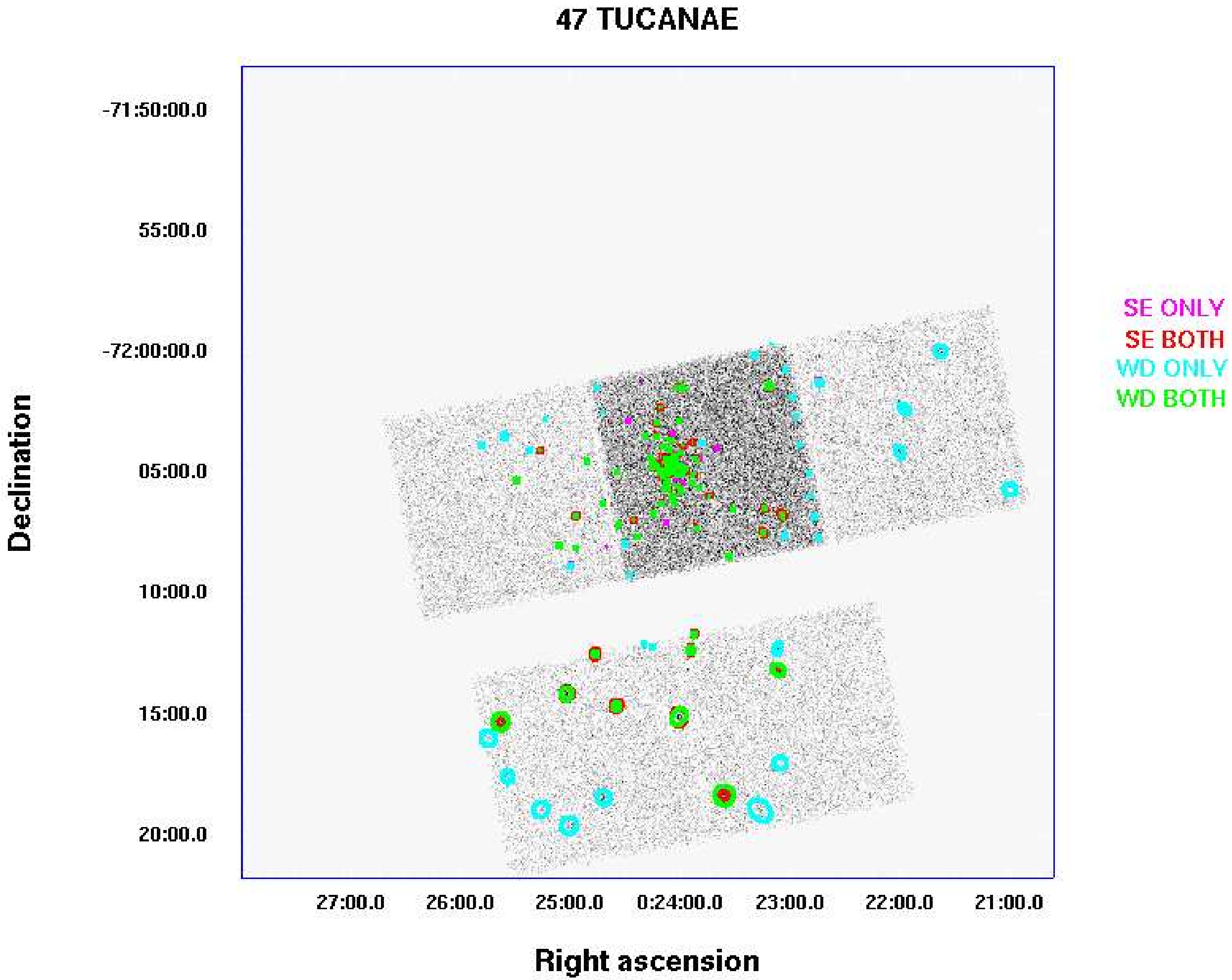


Comparison of WAVDETECT and Sextractor Detection Efficiency





Comparison of WAVDETECT and SE on Real Data



Sextractor Summary

- SExtractor can perform well, but not as well as WAVDETECT. Moreover, it has at times surprised us with inconsistent results (large numbers of spurious sources, or missing relatively obvious sources in off-axis areas) which required ad-hoc solutions.
- We don't yet know how much more work will be required to make us confident of consistently good SExtractor performance in a L3 pipeline.
- Recommend suspending SExtractor development using L3 resources.
- Can perhaps use it as is in L3 pipe as a backup to WAVDETECT.
- With relatively little work, can be packaged as stand-alone tool for, e.g. quick interactive analysis.

New WAVDETECT Results

- Benchmarking on fast machine indicates WAVDETECT can be run in L3 pipeline in less than Hubble Time.
- Simple background map algorithm improves performance and can deal with readout streak sources.

WAVDETECT Benchmarking

- Done a linux machine with 2 GHz processor and 1 GB RAM.
- Used Adam's "recursive blocking" scheme.
- Exposure maps created only the field being searched and at the same resolution as the image.
- Full-resolution (0.5 arcsec) aspect histograms used.
- Average time to an obsid with 6 chips active was ~80 minutes. Of this ~30 minutes was in running wavdetect and ~50 minutes making exposure maps.
- Using reduced resolution aspect histograms (4 arcsec) reduces total time to ~45-50 minutes. Number of detections seems to be the same but this needs to be checked.

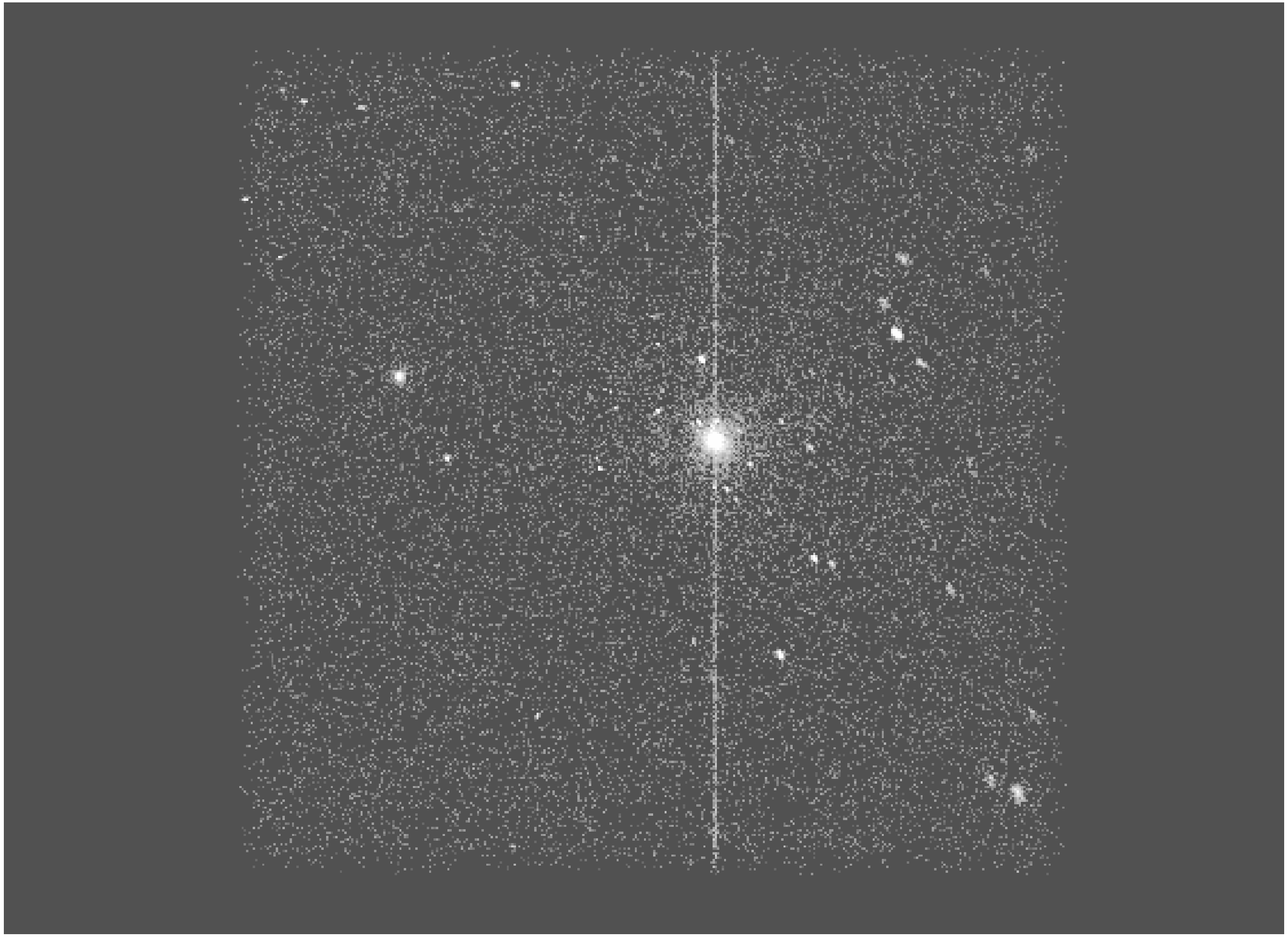
WAVDETECT Benchmarking Details

OBSID	13	114	635	650	735	953	1845	2000	2556
Total Time	01:20:08.27	00:57:32.36	01:24:52.79	01:27:56.96	01:22:09.06	01:22:56.35	51:23.35	02:55.71	01:45:38.57
Emap 1	40:12.99	16:13.65	41:11.67	39:23.58	36:59.83	42:35.41	15:04.56	01:30.91	51:45.61
Emap 2	08:43.18	02:13.54	10:55.79	12:08.52	09:56.75	08:10.22	04:26.48		13:57.42
Emap 4	03:41.43		05:59.04	06:45.17	04:38.72	03:28.18	02:14.44		07:11.22
Emap Total	00:52:37.60	18:27.18	00:58:06.49	00:58:19.27	00:51:35.30	00:54:13.81	21:45.48	01:30.91	01:12:54.25
Wave 1	11:14.68	31:50.75	11:29.68	11:15.65	12:06.58	11:22.16	10:32.53	51.36	13:05.81
Wave 2	10:50.76	05:22.16	11:11.93	13:00.97	13:01.77	10:59.21	10:20.78		13:26.79
Wave 4	05:15.92		03:55.14	05:09.58	05:15.14	05:13.68	08:34.76		06:02.48
Wave Total	27:21.36	37:12.92	26:36.74	29:26.20	30:23.49	27:35.05	29:28.07	51.36	32:35.08
Chips	6	1	5	6	6	6	6	1	6
Exp.Time (ks)	34.9	50.6	101.9	96.6	50.6	32.1	5.2	23.4	97.9
Sources	85	1307	90	175	269	149	36	160	451

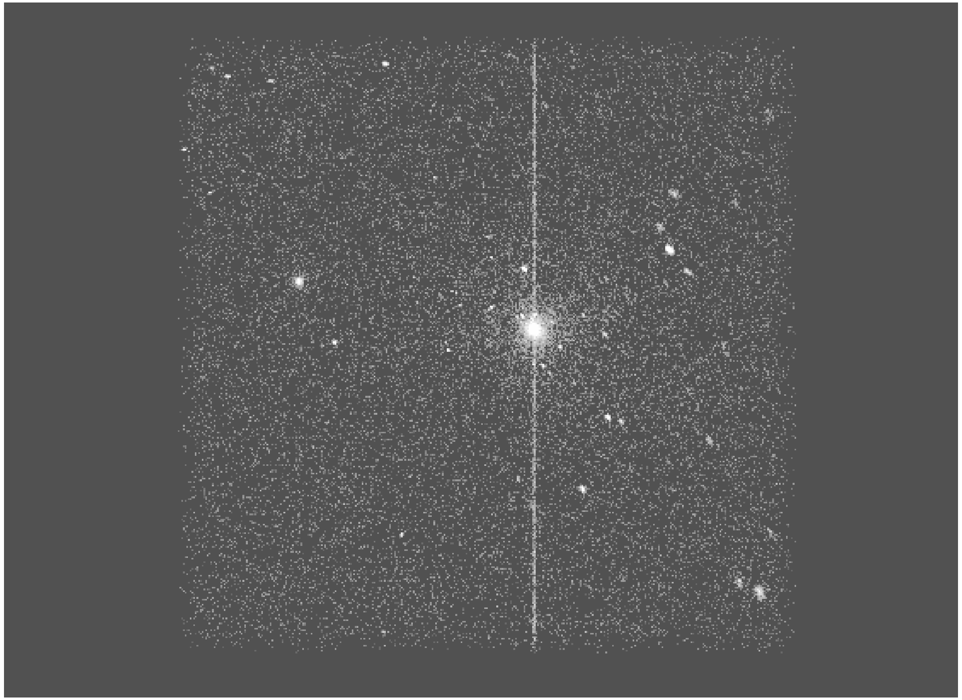
Method for Handling Readout Streaks and Background Map Creation

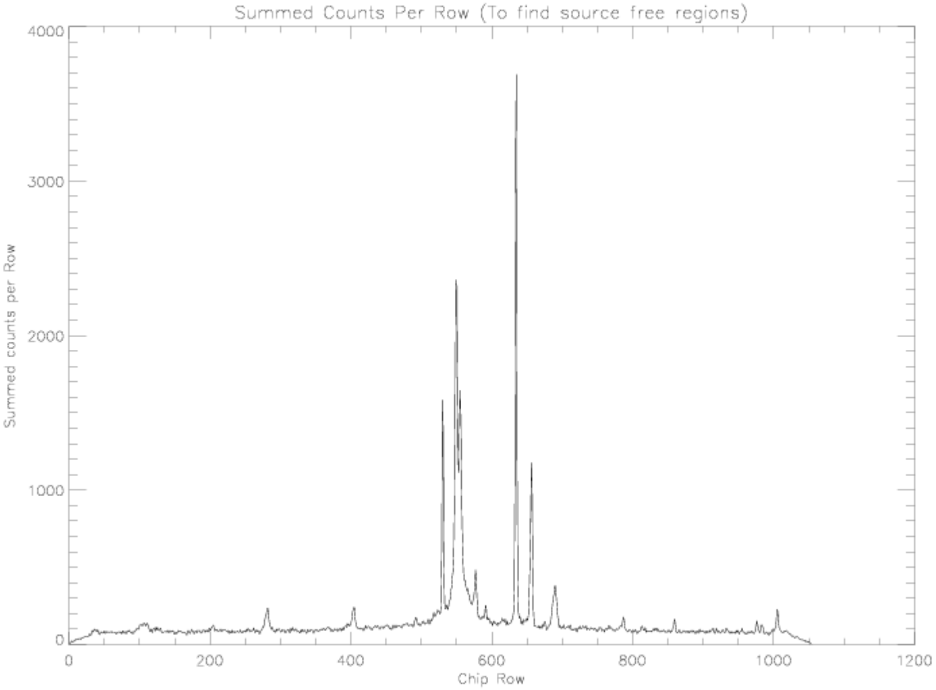
- Create separate map for each chip and regrid to actual image.
- Project across readout channels to identify source regions.
- Project source-free regions along readout channels to identify readout streaks.
- Replicate latter projection (including readout streaks) to generate background map.

Example: M81 in ACIS-S3



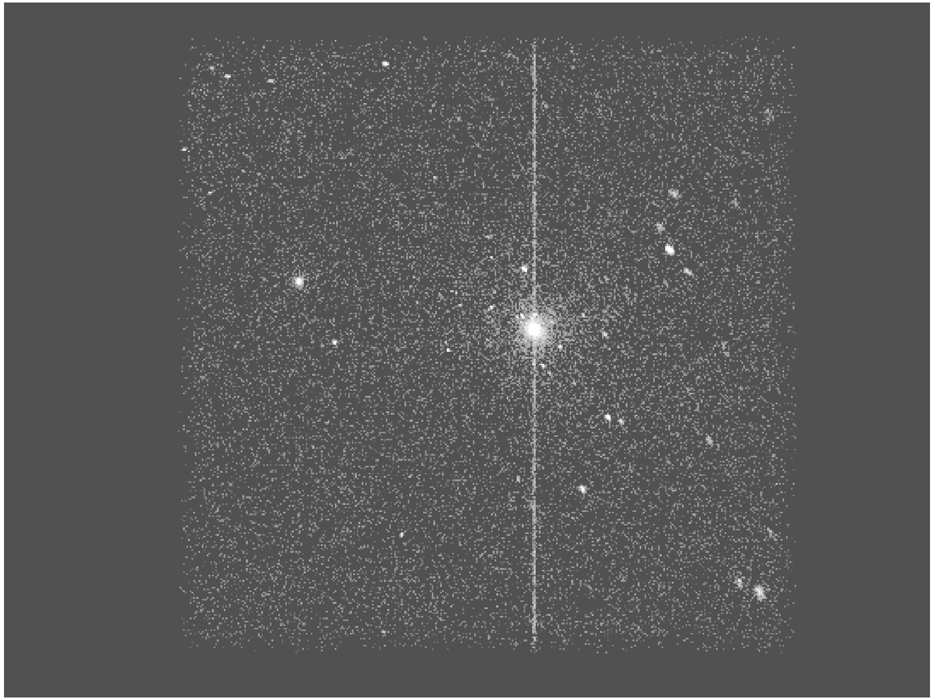
M81 (Cont.)

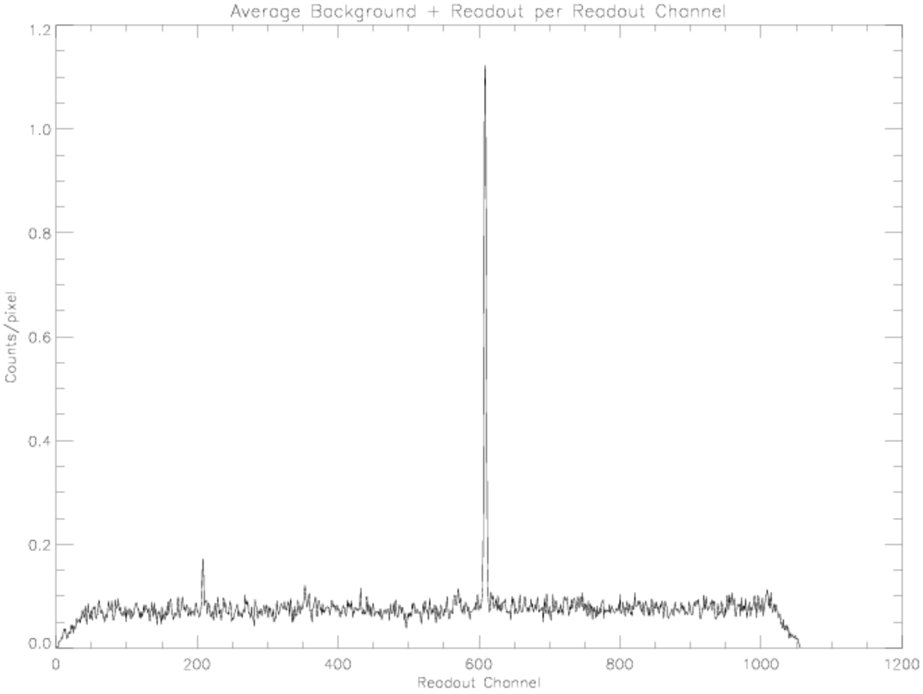




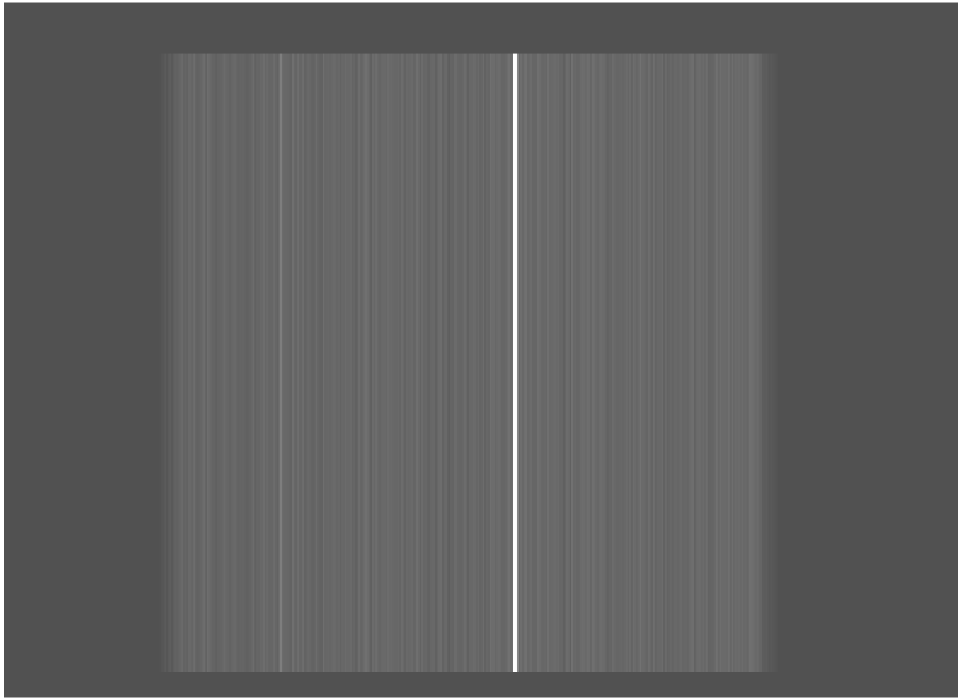
Summed Counts Per Row (To find source free regions)
Summed counts per Row
Chip Row
0
1000
2000
3000
4000
0
200
400
600
800
1000
1200

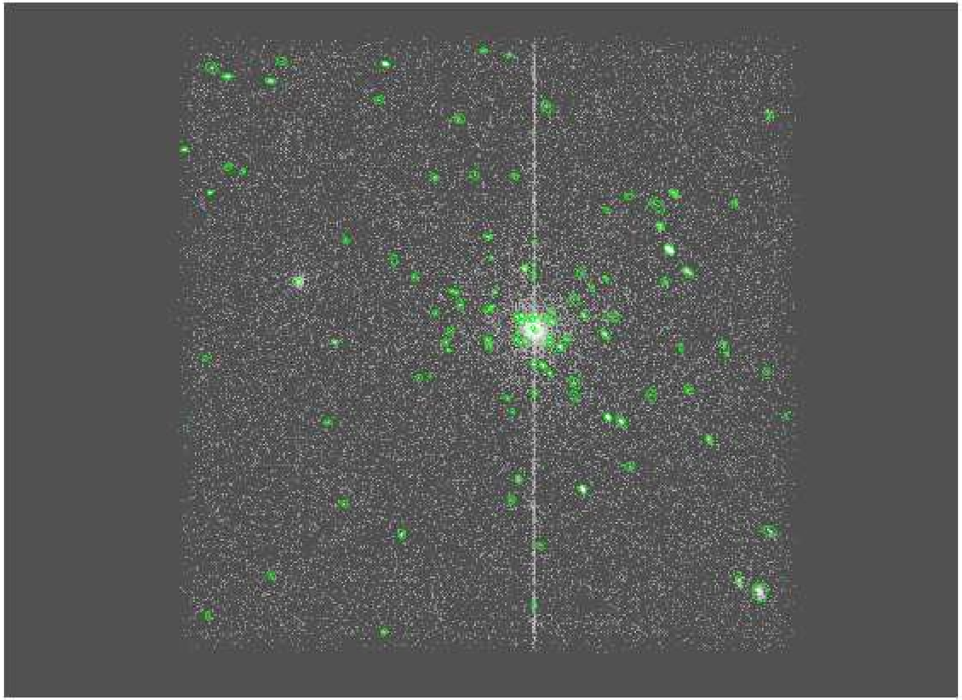
M81 (Cont.)




Average Background + Readout per Readout Channel
Counts/pixel
Readout Channel
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
200
400
600
800
1000
1200

M81 (Cont.)





DETECT Summary

- WAVDETECT with exposure maps can be run in L3
- Simple background map algorithm looks promising
- No further work on Sextractor anticipated (at least in L3 context)

L3 DETECT Issues

- Short-Term Issues
  - Sort out L3 WAVDETECT details (scales, asphist resolution...)
    - Compare with PSU and Palermo s/w
  - Automate background map generation
  - Do we include Sextractor as is?
  - Post-DETECT (but pre-2-d fitting) source filtering
    - Shape
    - Location wrt edges, readout streaks

L3 DETECT Issues

- Longer-Term Issues
  - Point Source Sensitivity and False Source Rates
  - Use of Blank Sky Background Maps
  - Detection and characterization of extended sources
    - VTPDETECT?