GMRT Imaging pipline for Point source, Transients

Sanjay Kudale
Team:Jayaram Chengalur, Niruj Mohan

Workshop on Gamma-ray Bursts : Prompt to Afterglow

NCRA, Pune, INDIA.
ksanjay@gmrt.ncra.tifr.res.in

4-7th Jul 2017

Outline

- Introduction
- Basic Elements of Pipeline
- Flowchart : Flagging, Calibration
- Flowchart : Main Pipeline
- Flowchart : CASA – autoclean (modified)
- User Input paramers to pipeline
- Analysis results: Sample images
- Analysis results: Lightcurves (In-Progres)
- Timing
- Future plans

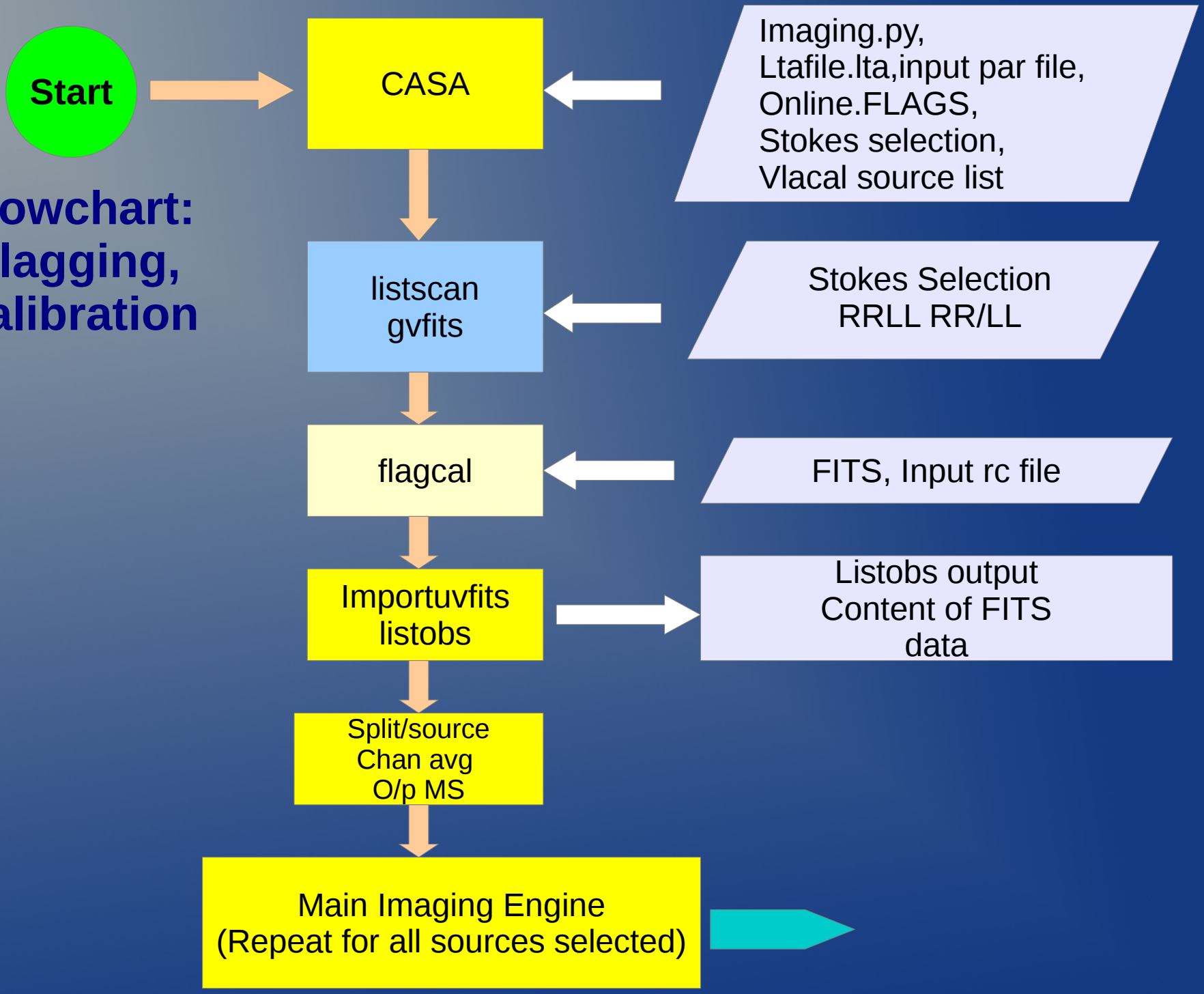
Introduction

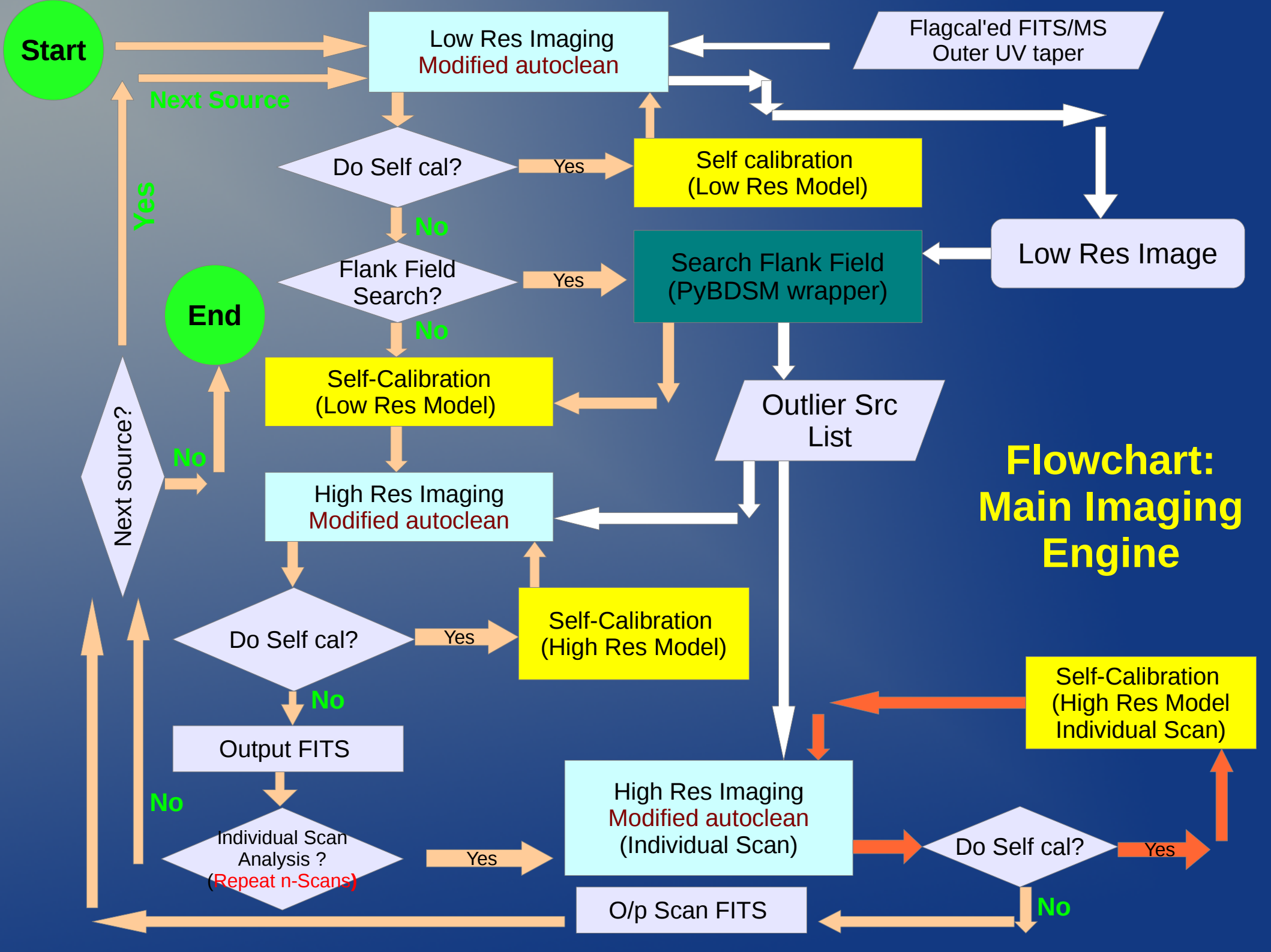
- Goal : To design general purpose Imaging pipeline for GMRT
- To start with by searching for variable/transient sources at low frequencies
  - Phase calibrator scans : Intra-day & long term variability of background sources
  - Target source scans intra-day variability
    - As the target fields are rarely repeated

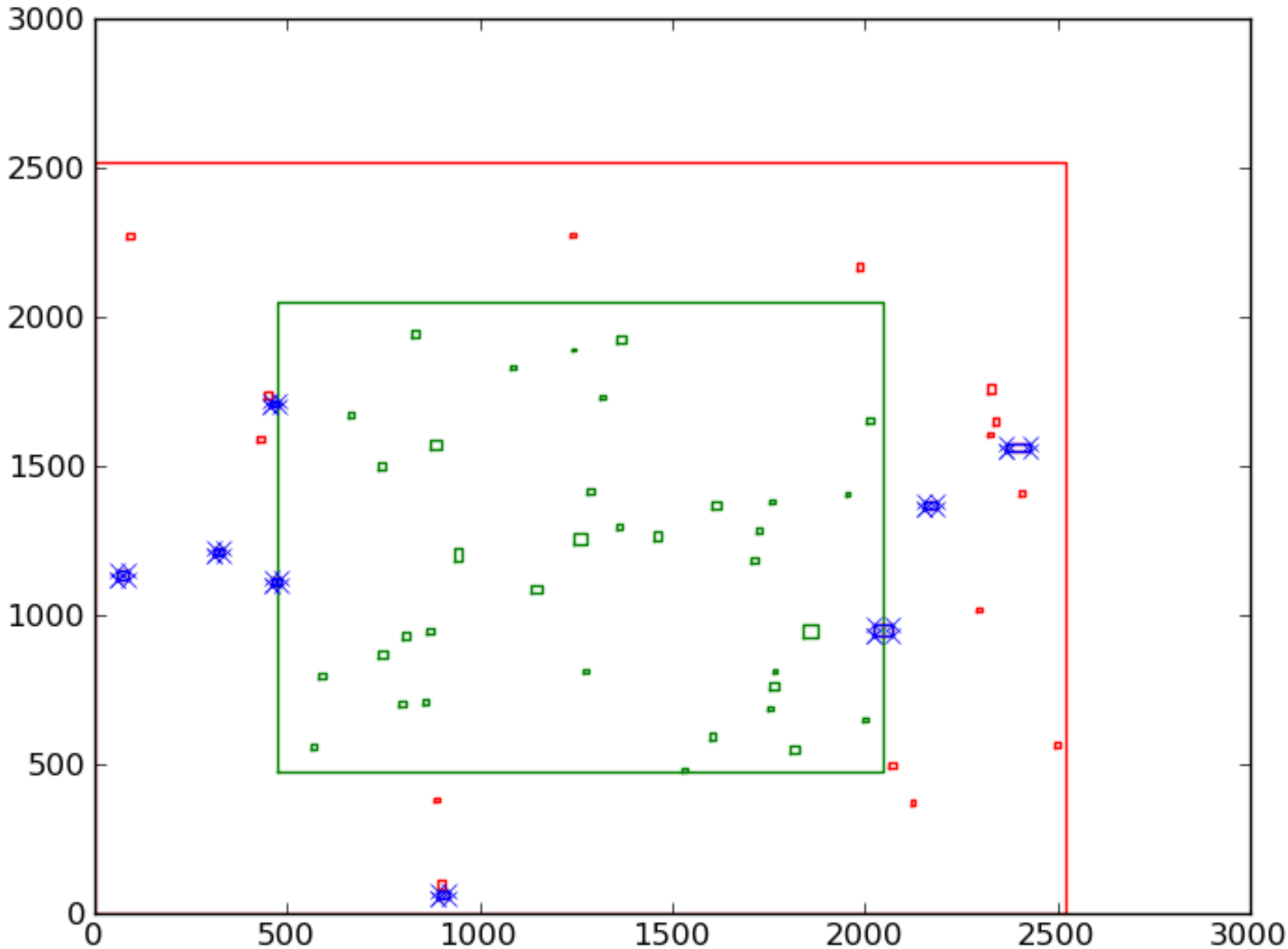
Elements of Pipeline

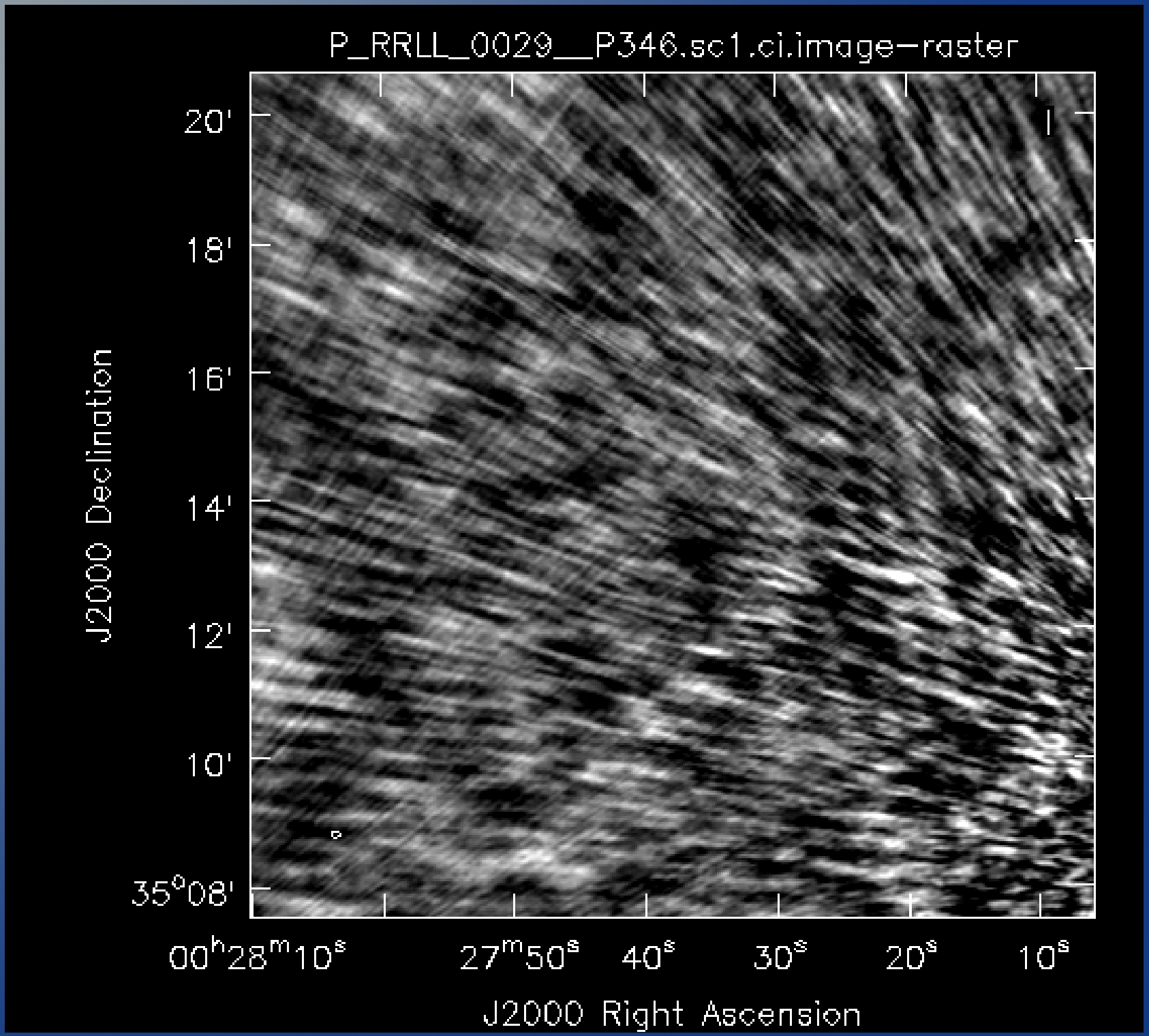
- flagcal : Package to flag and calibrate GMRT interferometric data. (Chengalur, 2013)
  - I/p GMRT lta data o/p calibrated FITS
  - Various thresholds for flagging & Calibration (User's input)
- PyBDSM : Python based software - 'Blob Detection and Source Management'. (Mohan 2009)
  - Wrapper which calls PyBDSM from this pipeline
  - Package to decompose an image into sources
  - Source detection over local RMS instead of whole image RMS
  - Output clean region written in CASA CRTF format
- CASA : Used for imaging and self-calibration
  - Execute custom 'imaging' python scripts in CASA shell

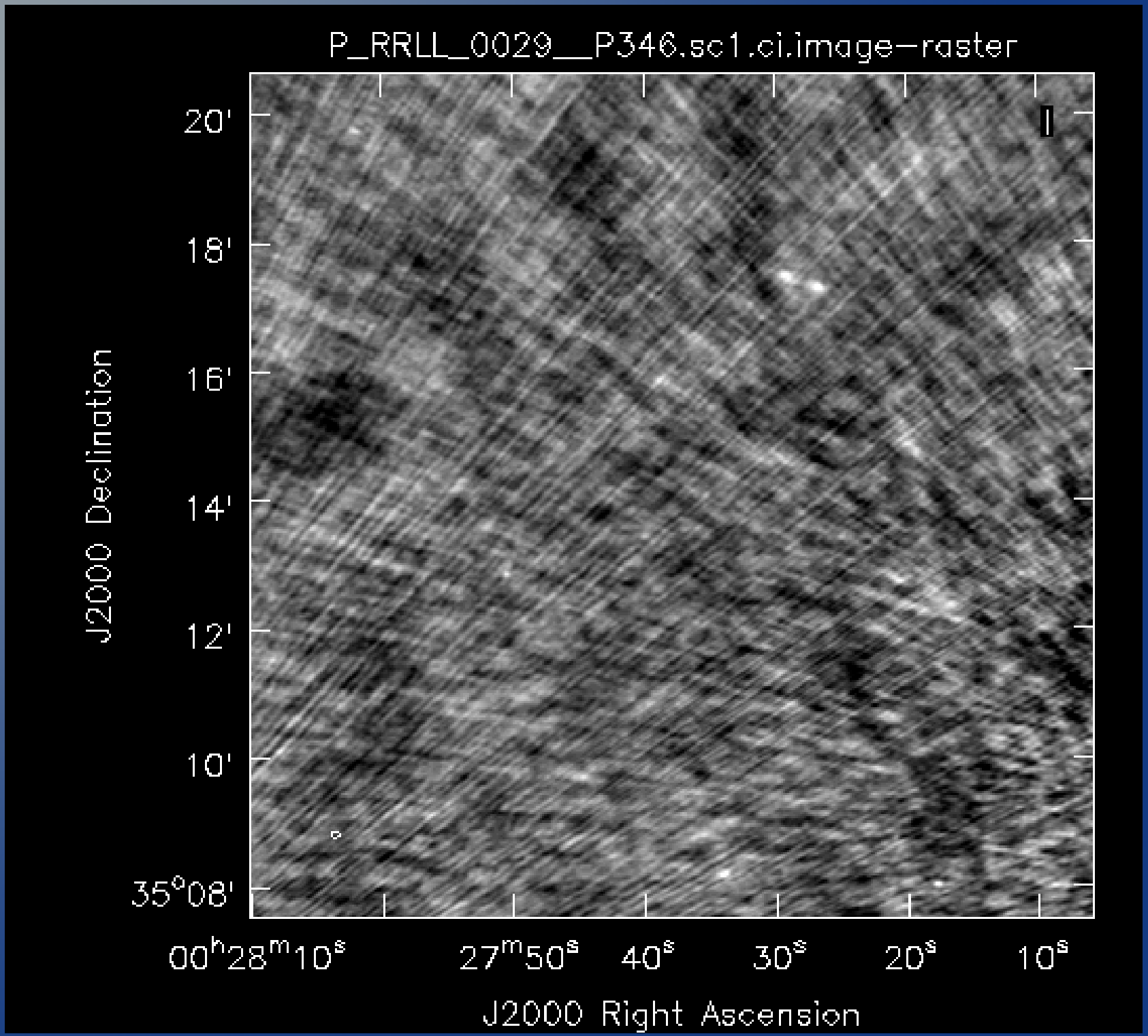
Flowchart: Flagging, Calibration

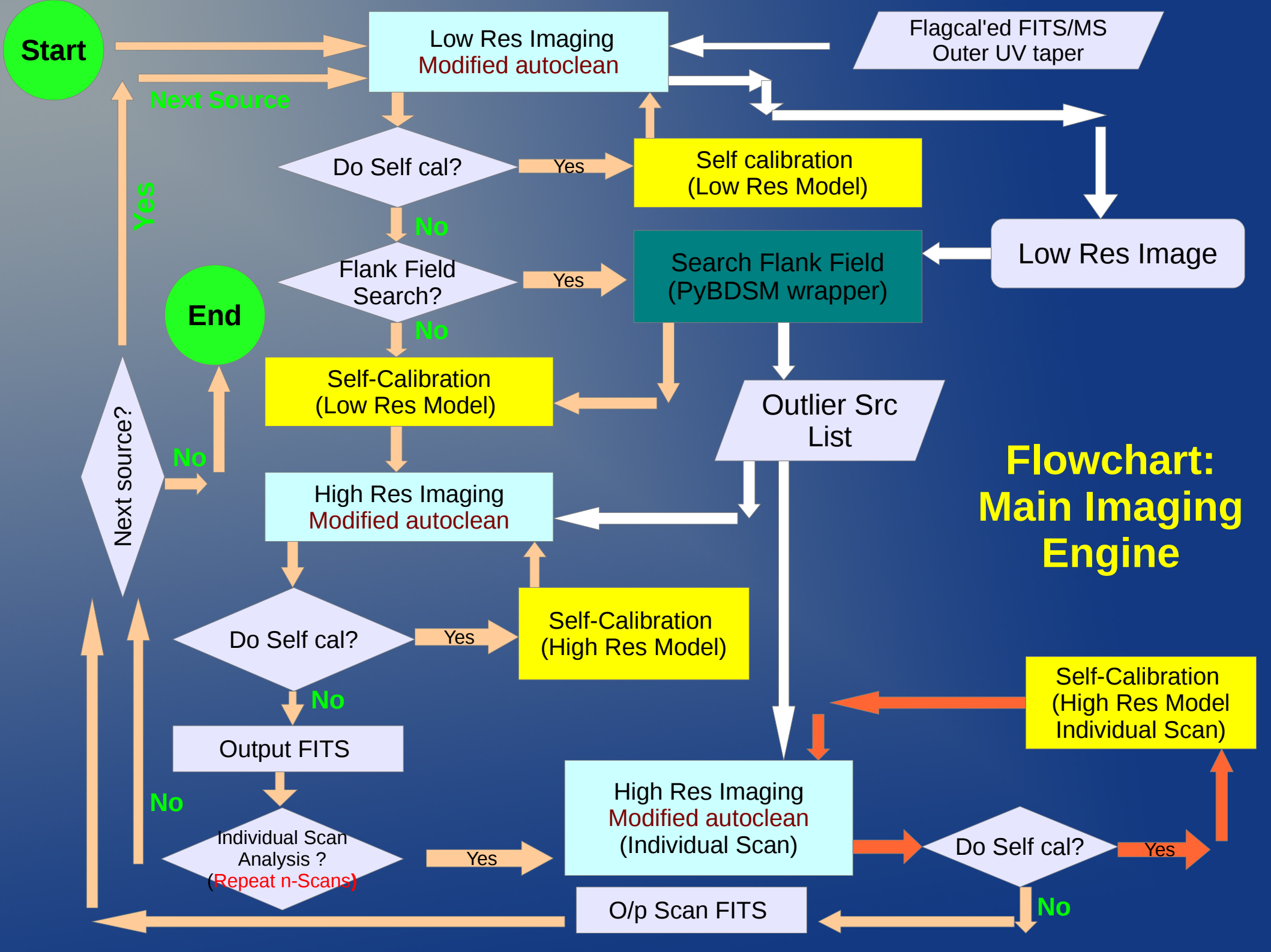



Start
Low Res Imaging
Modified autoclean
Flagcal'ed FITS/MS
Outer UV taper
Next Source
Yes
Do Self cal?
Yes
Self calibration
(Low Res Model)
No
Low Res Image
Flank Field
Search?
Yes
Search Flank Field
(PyBDSM wrapper)
End
No
Self-Calibration
(Low Res Model)
Outlier Src
List
Next source?
No
High Res Imaging
Modified autoclean
Flowchart:
Main Imaging
Engine
Do Self cal?
Yes
Self-Calibration
(High Res Model)
No
Output FITS
Self-Calibration
(High Res Model
Individual Scan)
No
Individual Scan
Analysis ?
(Repeat n-Scans)
Yes
High Res Imaging
Modified autoclean
(Individual Scan)
Do Self cal?
Yes
No
O/p Scan FITS

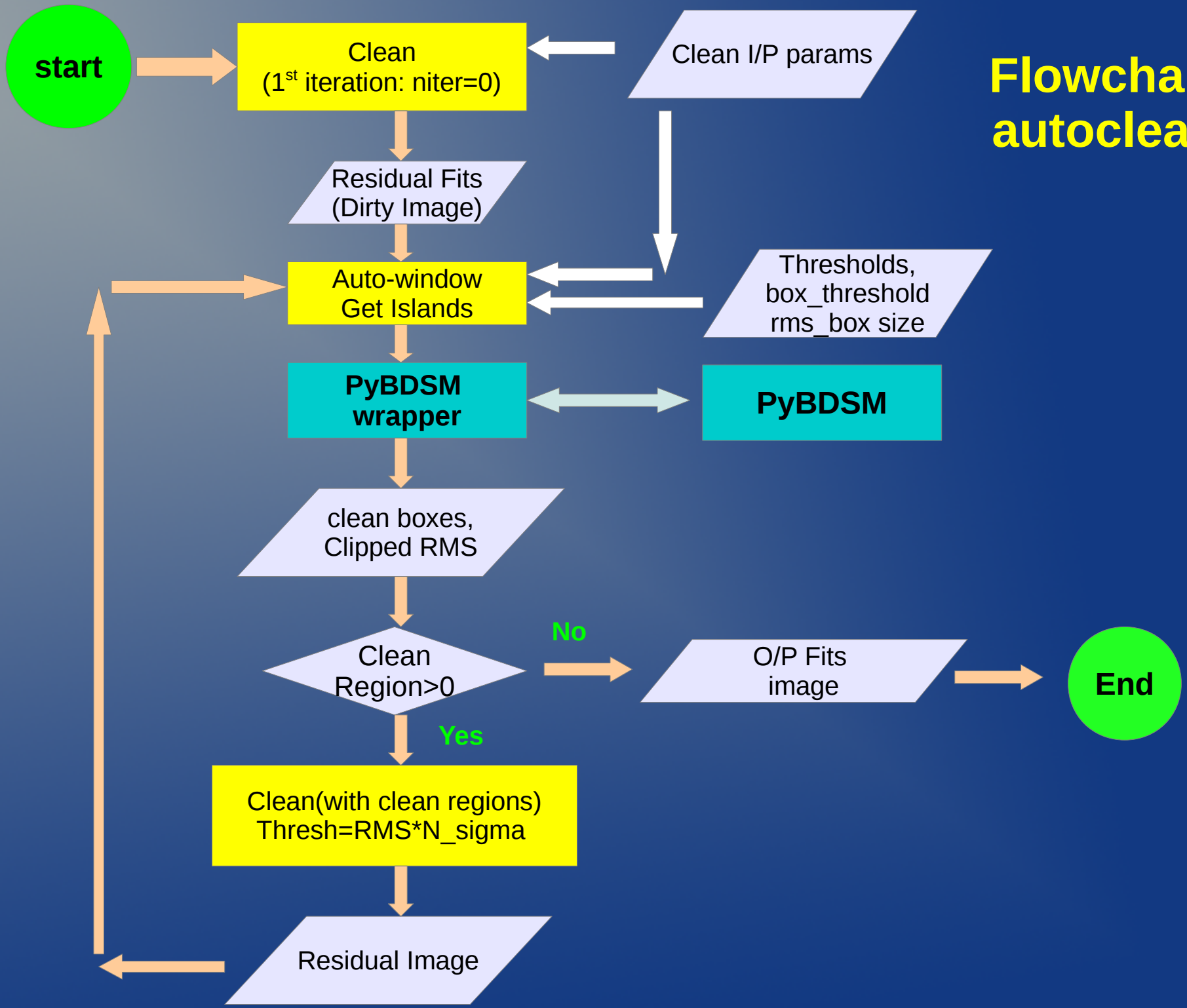

3000
2500
2000
1500
1000
500
0
0
500
1000
1500
2000
2500
3000

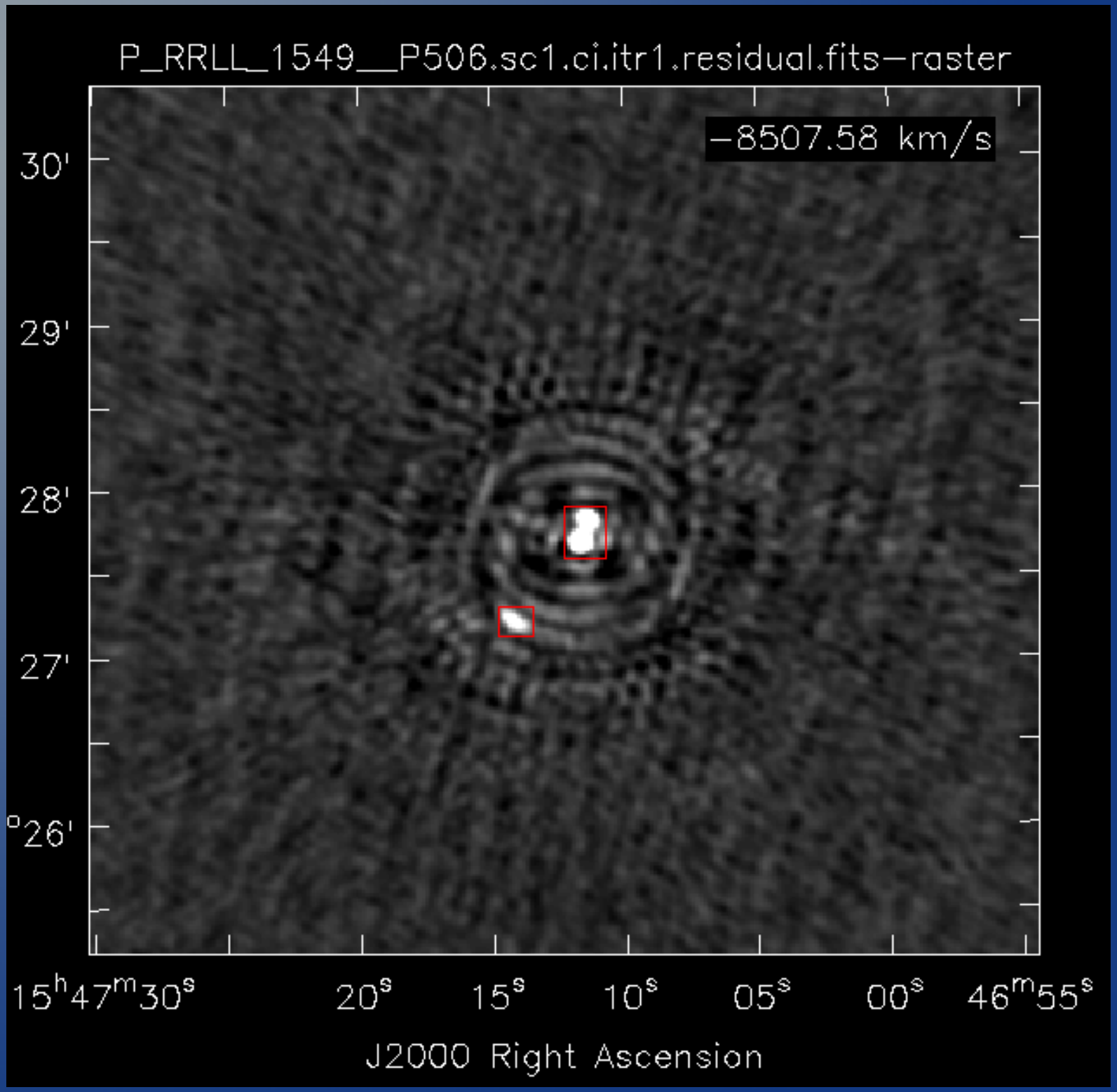

P_RRLL_0029__P346.sc1.ci.image-raster
20'
18'
16'
14'
12'
10'
35°08'
J2000 Declination
00h28m10s
27m50s
40s
30s
20s
10s
J2000 Right Ascension


P_RRLL_0029__P346.sc1.ci.image-raster
20'
18'
16'
14'
12'
10'
35°08'
J2000 Declination
00h28m10s
27m50s
40s
30s
20s
10s
J2000 Right Ascension

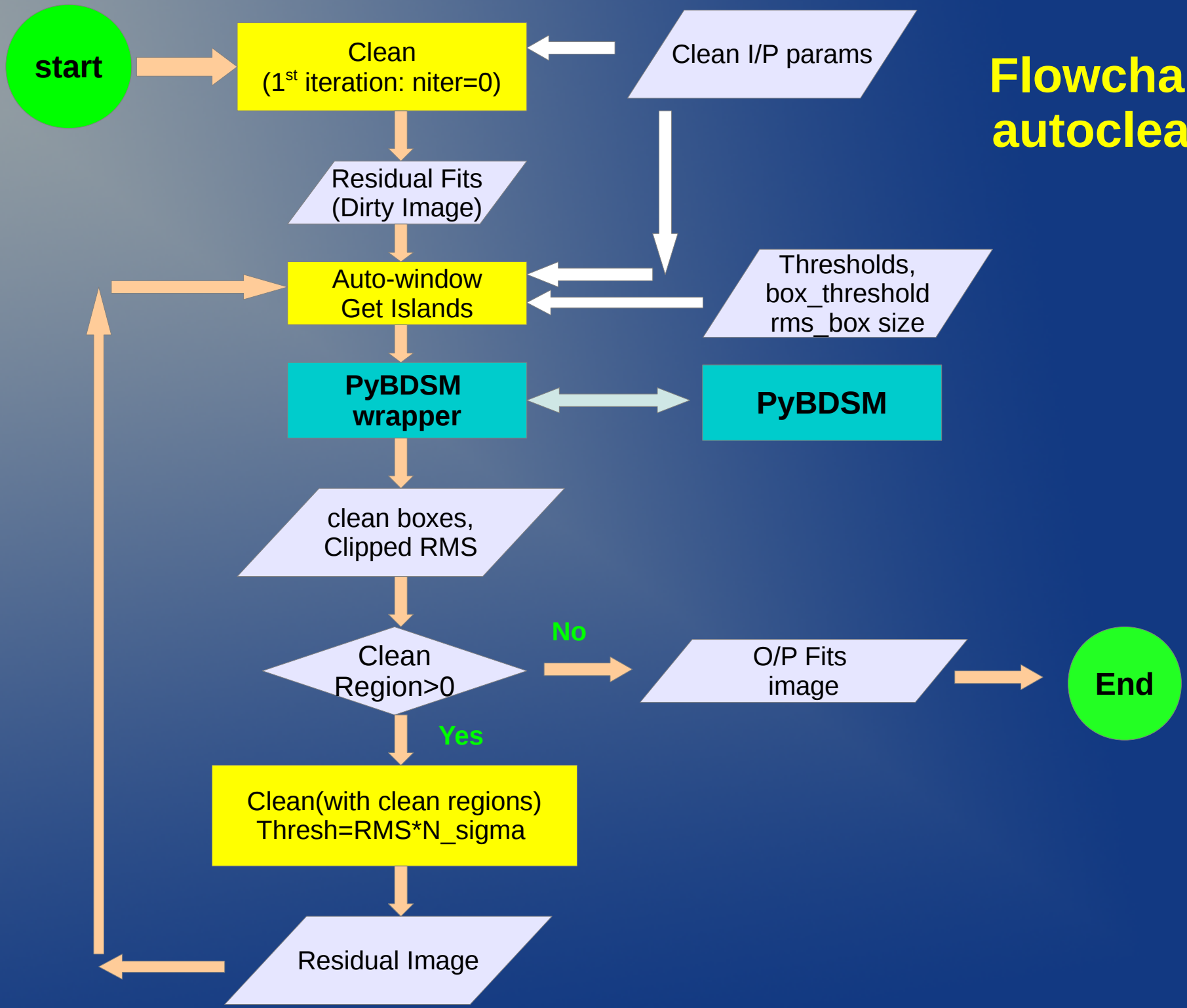

Start
Low Res Imaging
Modified autoclean
Flagcal'ed FITS/MS
Outer UV taper
Next Source
Yes
Do Self cal?
Yes
Self calibration
(Low Res Model)
No
Low Res Image
Flank Field
Search?
Yes
Search Flank Field
(PyBDSM wrapper)
End
No
Self-Calibration
(Low Res Model)
Outlier Src
List
Next source?
No
High Res Imaging
Modified autoclean
Flowchart:
Main Imaging
Engine
Do Self cal?
Yes
Self-Calibration
(High Res Model)
No
Output FITS
Self-Calibration
(High Res Model
Individual Scan)
No
Individual Scan
Analysis ?
(Repeat n-Scans)
Yes
High Res Imaging
Modified autoclean
(Individual Scan)
Do Self cal?
Yes
No
O/p Scan FITS

Flowchart: autoclean



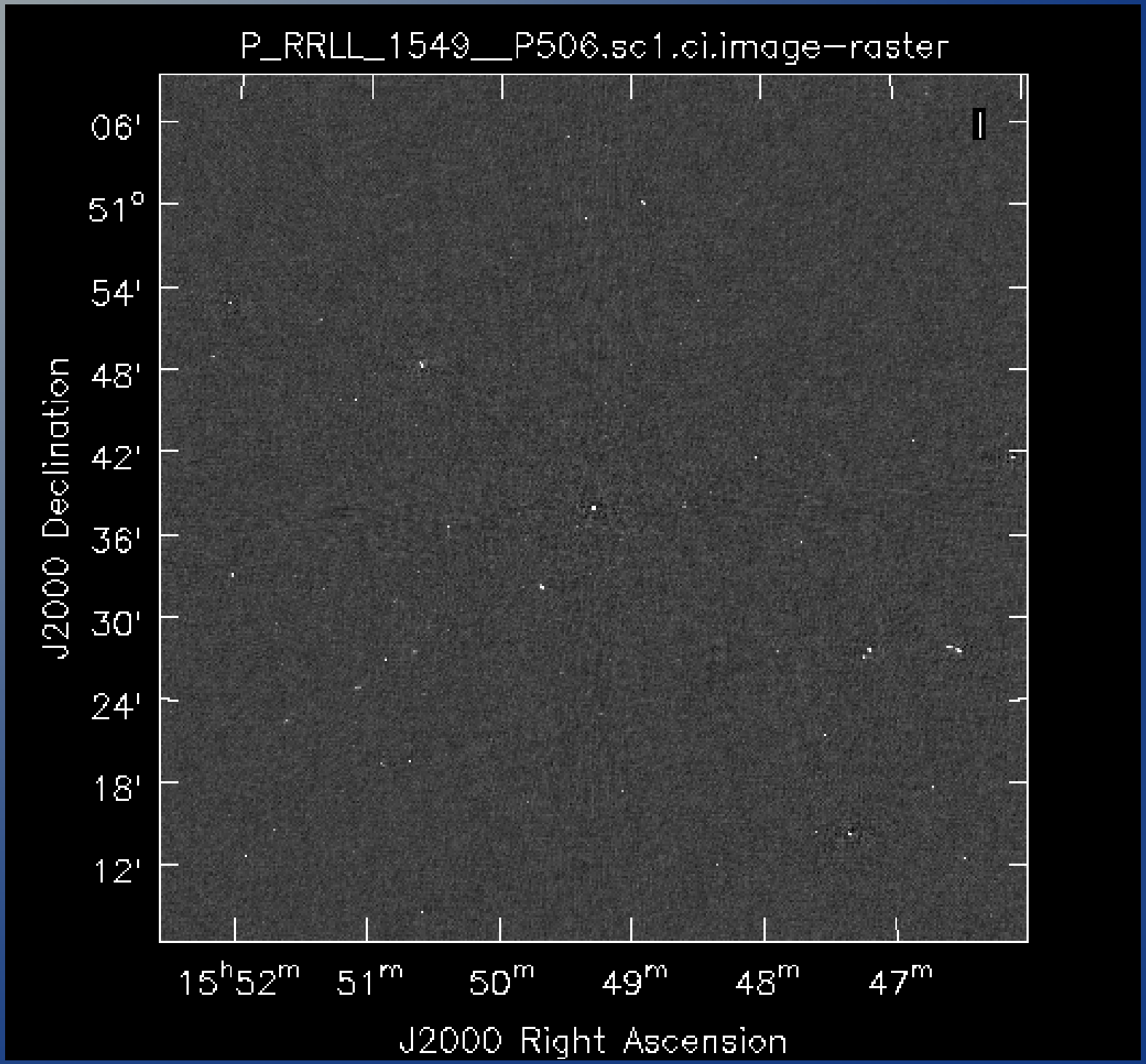

P_RRLL_1549__P506.sc1.ci.itr1.residual.fits-raster
-8507.58 km/s
30'
29'
28'
27'
°26'
15h47m30s
20s
15s
10s
05s
00s
46m55s
J2000 Right Ascension

Flowchart: autoclean

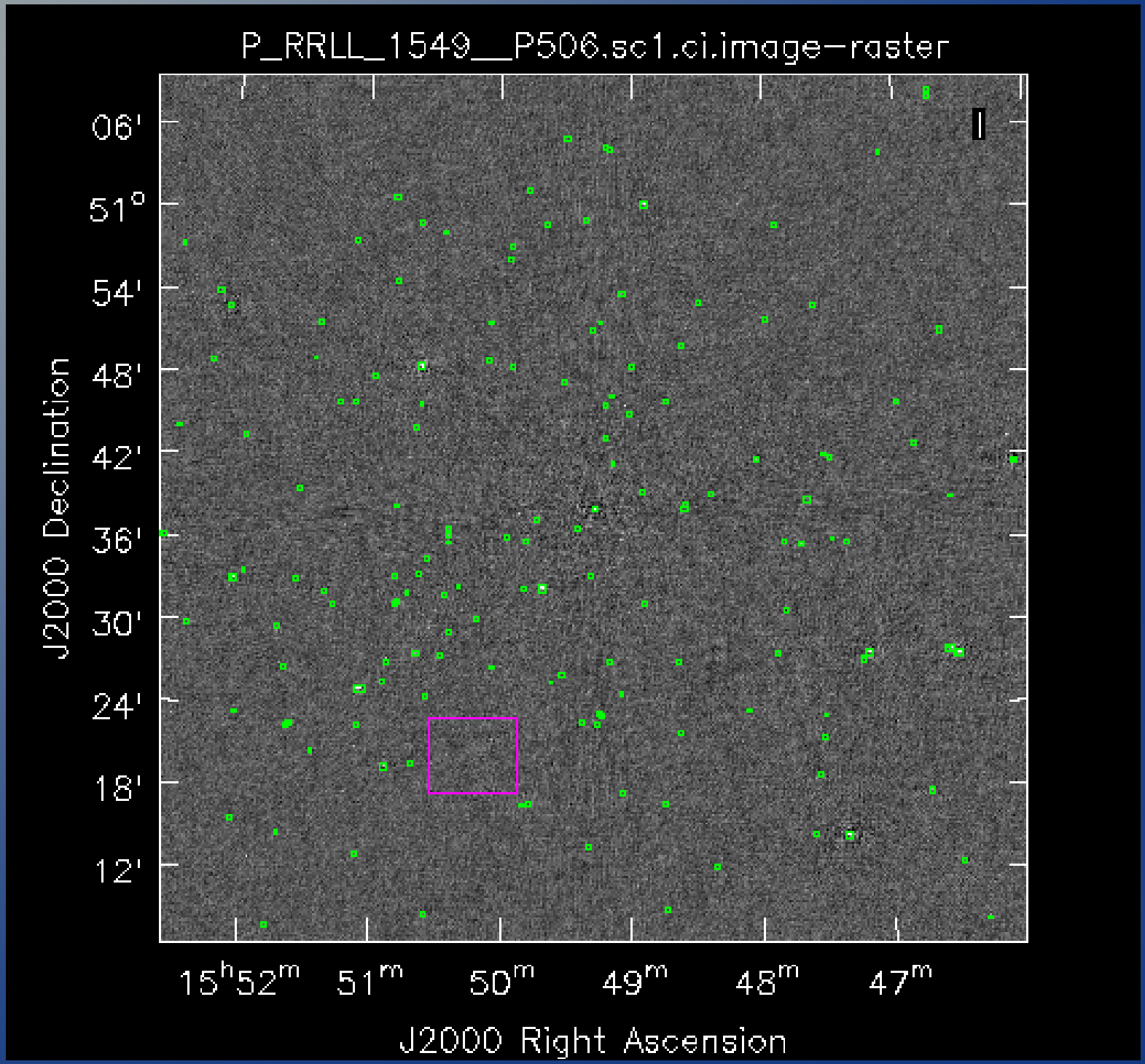


Input Parameters

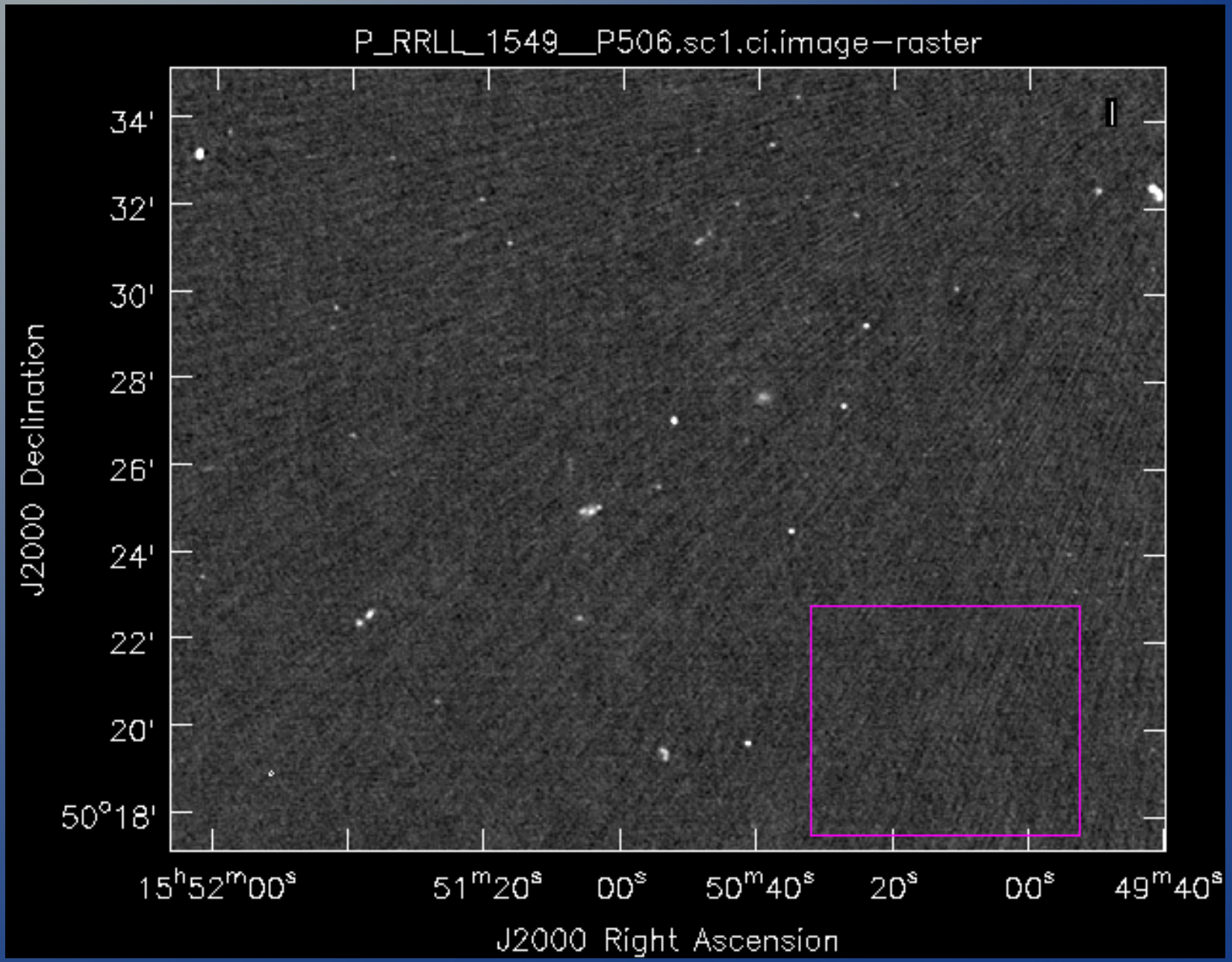
- MAP_FLUX_CAL = False
- MAP_PHASE_CAL = True
- MAP_TARGET_SRC = False
- PLOT_UV = True
- PLOT_UVCOVERAGE = True
- DO_FLANK = True
- CELL = 2.0
- IMSIZE = 110
- FLANK_IMSIZE = 120
- FLANK_NSELFCALCYCLES = 0
- NSELFCALCYCLES = 2
- SCAN_NSELFCALCYCLES = 0
- SCAN_LIGHTCURVE = False
- CONTAM_SIGMA_THRESH = 3.0
- CONTAMINATION_LEVEL = 0.20
- INDIVIDUAL_SCANS = False
- MULTISCALE = [ ]
- RESTORING_BEAM = False



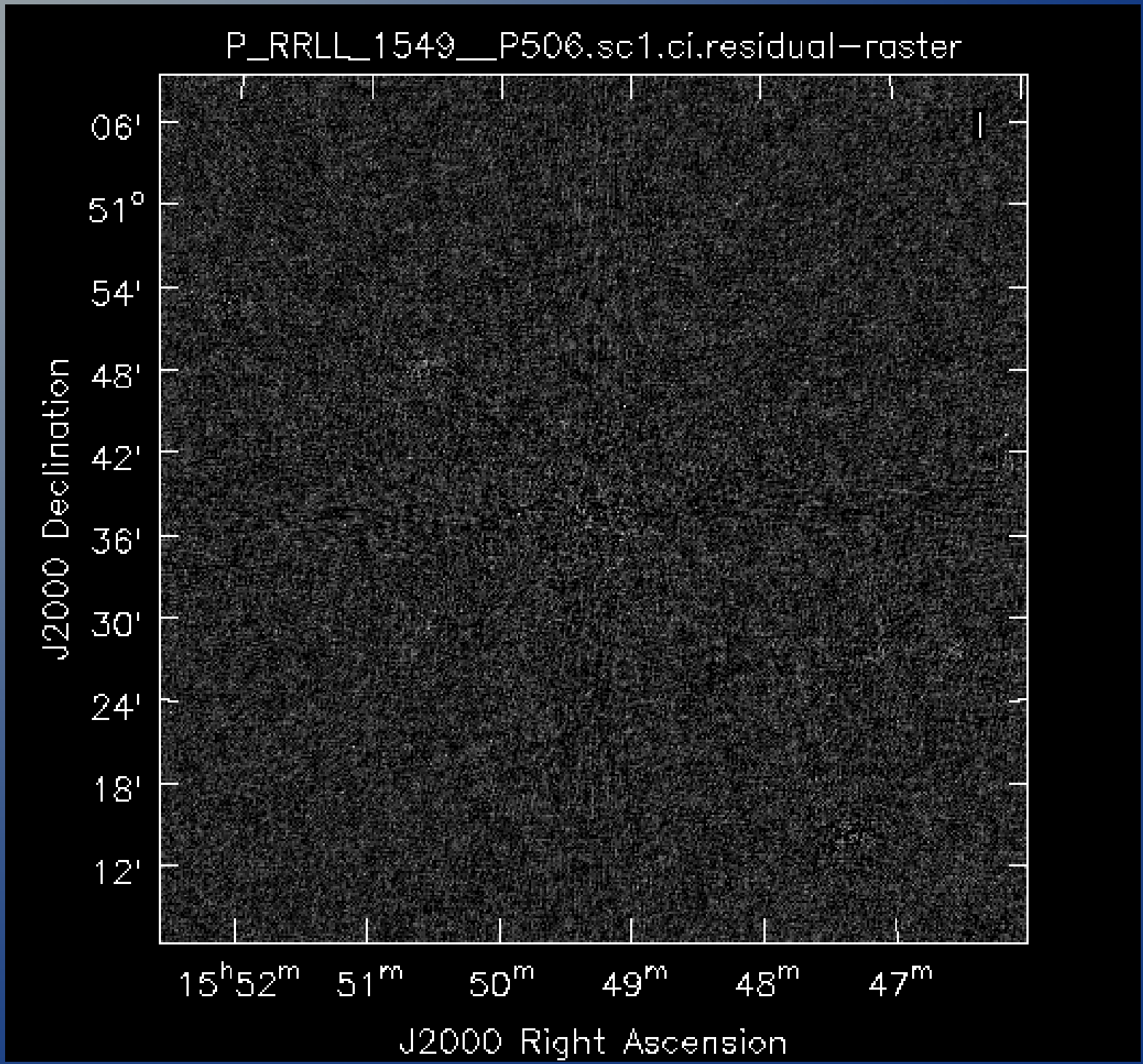
68.62uJy/beam
93.4 min
610MHz
33.33 MHz



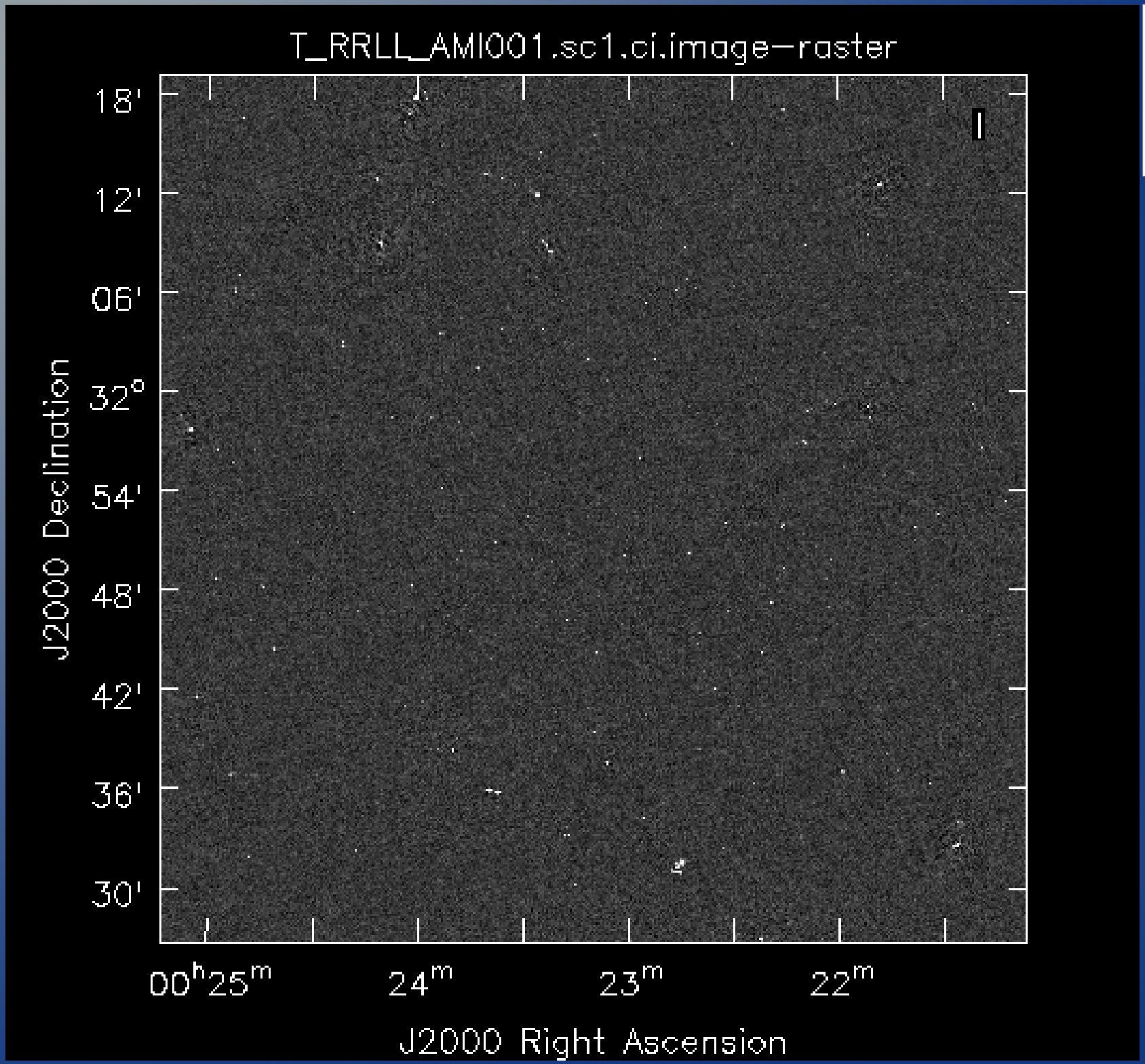
68.62uJy/beam
93.4 min
610MHz
33.33 MHz



93.4 min
61.72 uJy/beam,
610/33.3 MHz

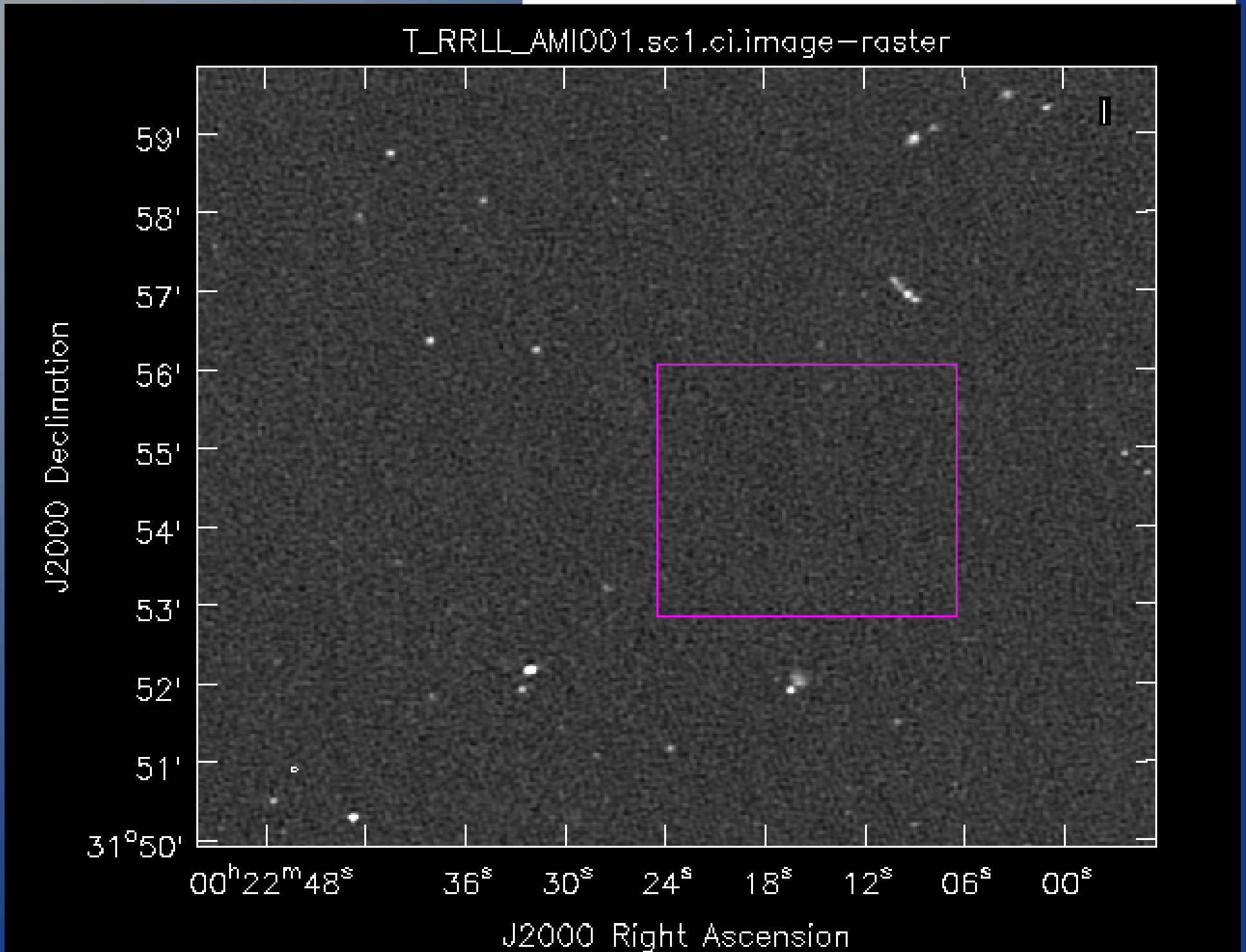


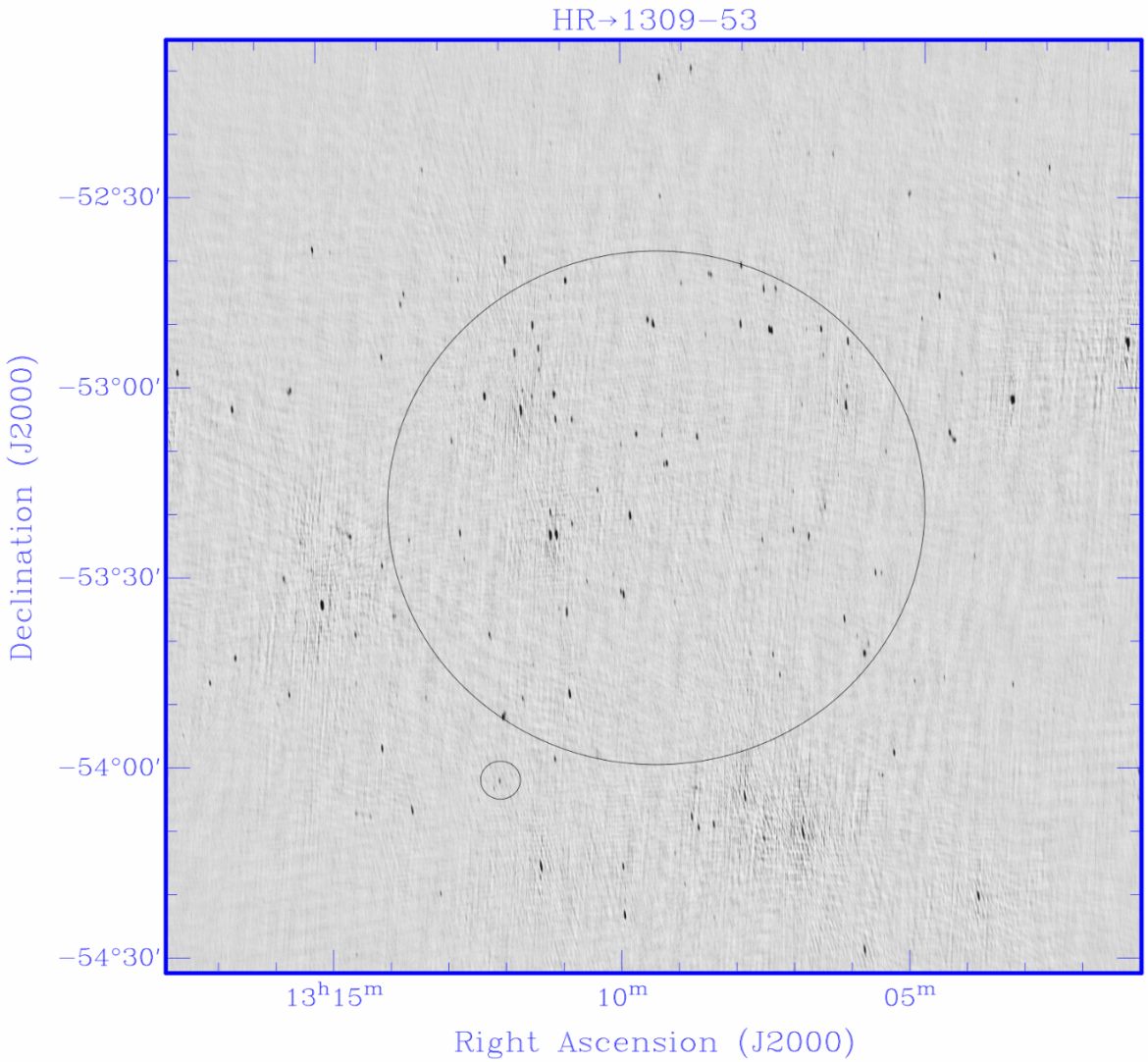
68.62uJy/beam
93.4 min
610MHz
33.33 MHz



610/33 MHz
6.92 Hrs,
~350 boxes
31.54 uJy/beam

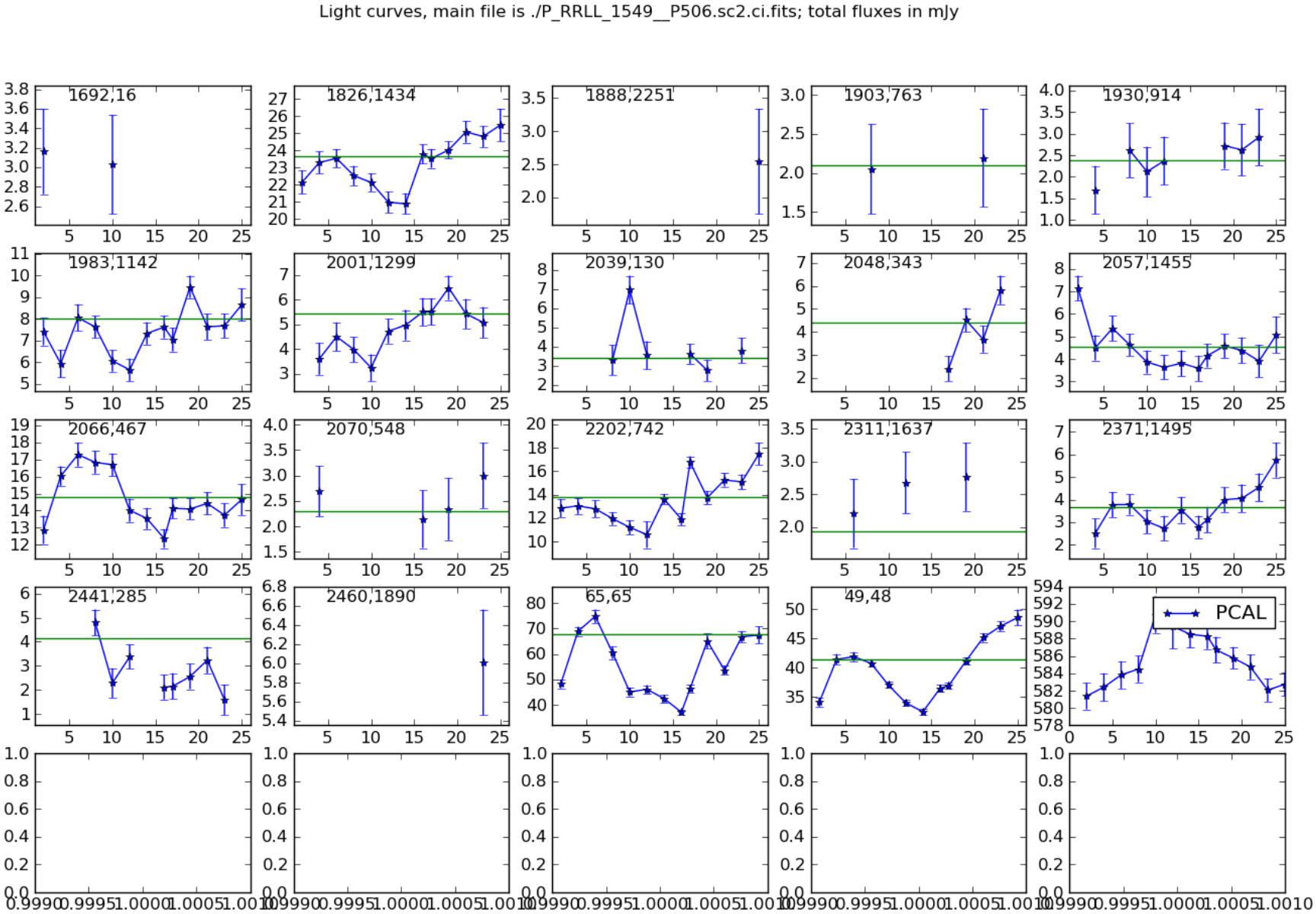
610/33.33 MHz 6.92 Hrs, 31.54 uJy/beam



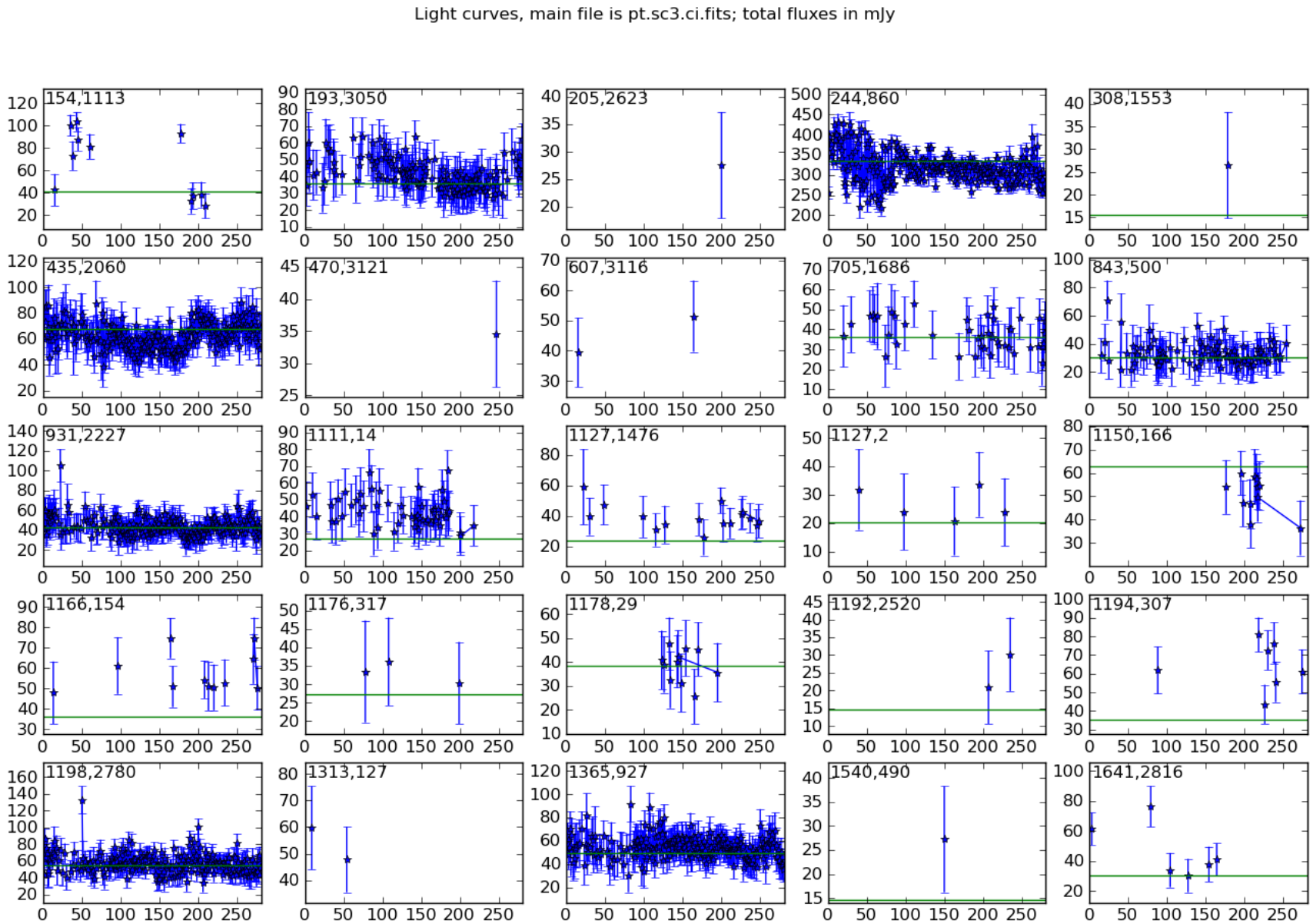


17.1 min,
325/33 MHz
Pulsar (12.419mJy),
rms=472.45 uJy/beam,
47' from PC
Dec=-53deg

Light-curve (scan based)

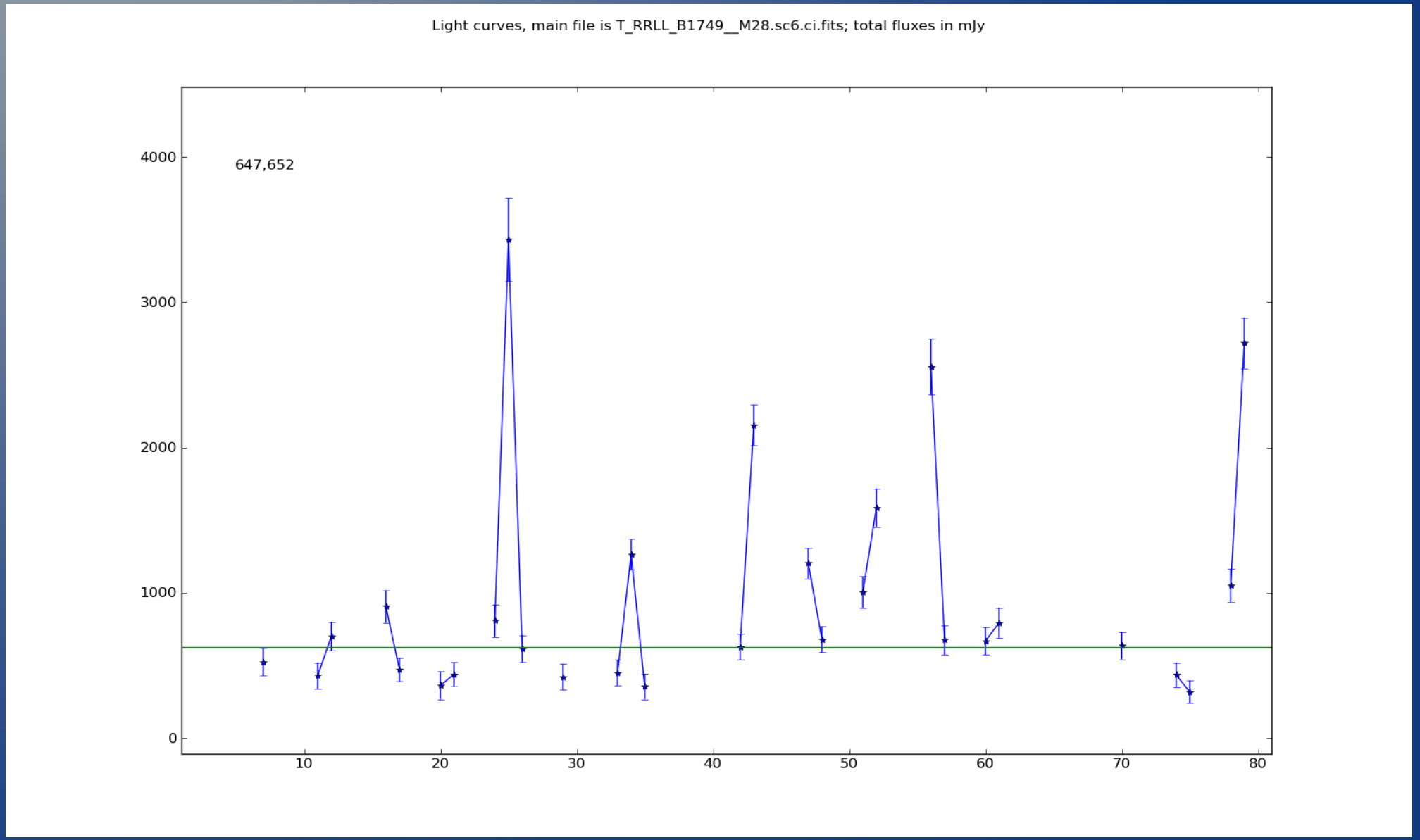


2-Sec Resolution light-curve
(manual followup)



Shapshot imaging of pulsar:B1749-28

Period ~ 562.580 mSec, time frame : 2.01 sec.



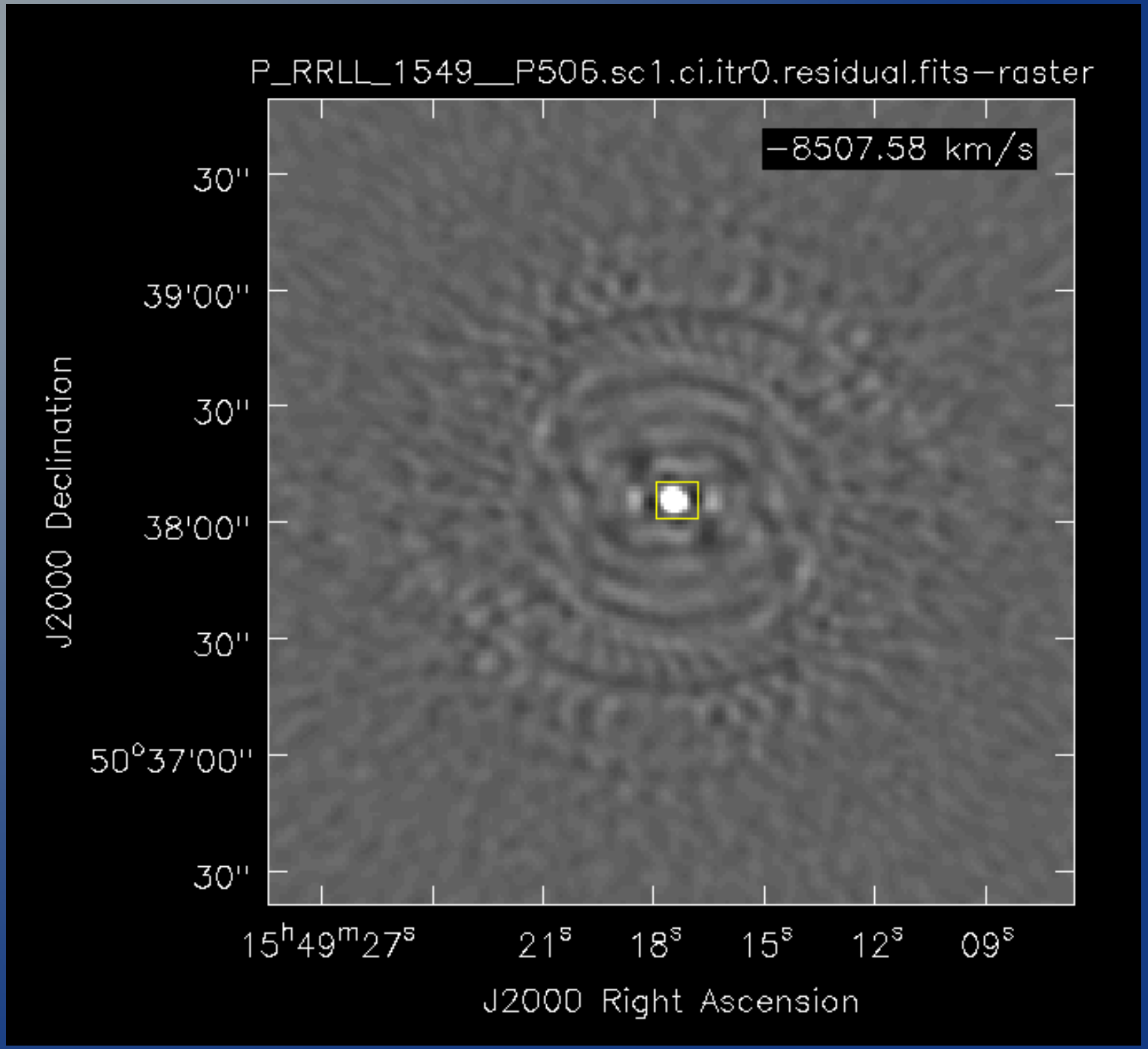
Timing

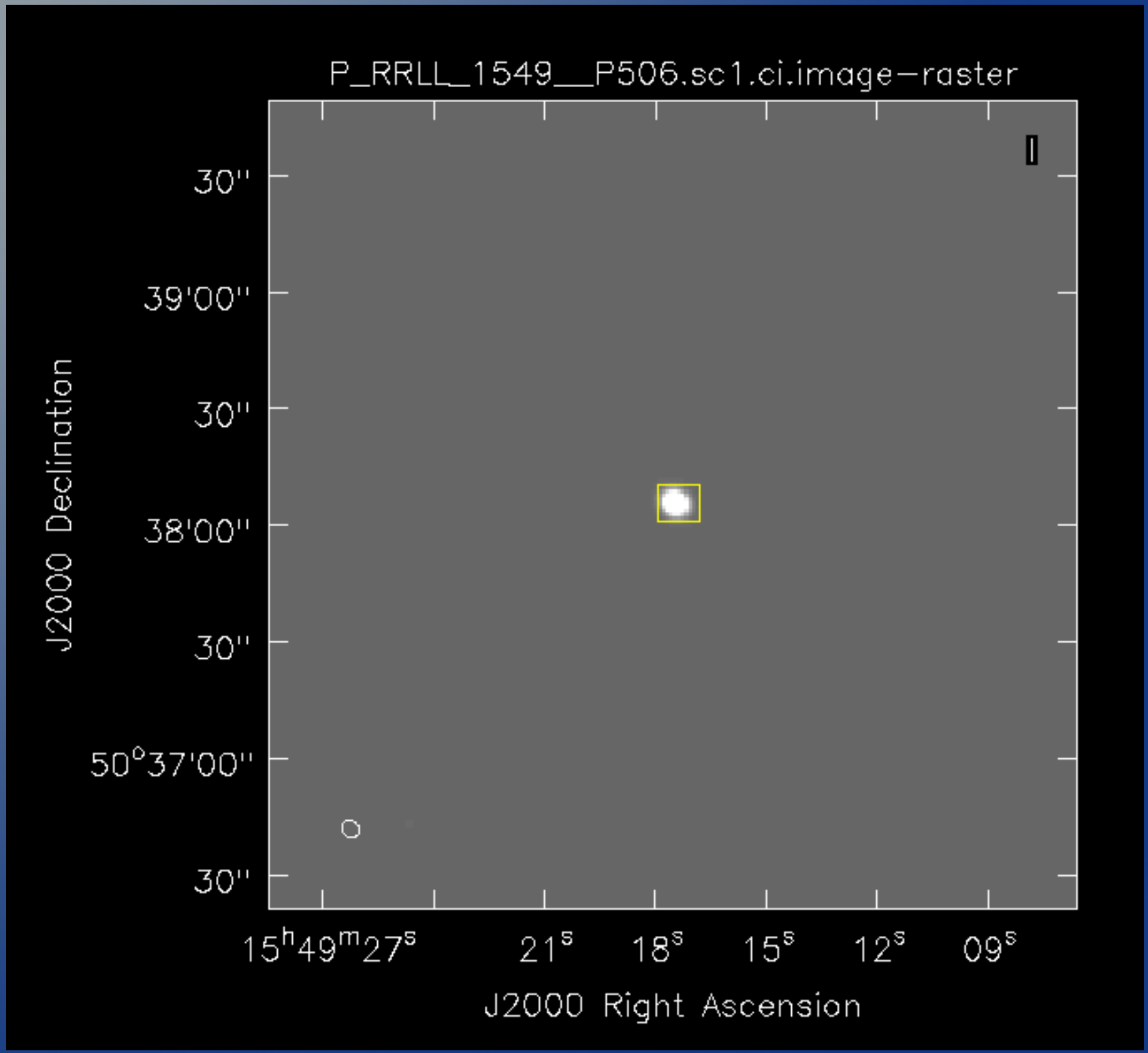
- Machine used is of high compute power
  - 32 cores (2599.969 MHz), 256GB RAM, RAID storage disk (faster disk i/o)
  - But CASA does not use all cores all the time, but heavy disk I/O
  - PyBDSM is not parallel
  - 'flagcal' makes use of all cores (Typical 9hrs data ~= 3.3 mins.)
- Faster than real time for HPFW image size without flank field (60% of real-time)
- Quasi-real time for HPFW image size with (few) flank field (~100% of real-time)
- For full size image (1.5 * HPFW) without flank field is ~1.5 times real-time
- Analysis time increases with image size
- Analysis time increses with no. of time frames (snapshot imaging)
- Analysis time increases with number of detection
- 'flagcal', 'PyBDSM' : OK. autoclean takes significant time

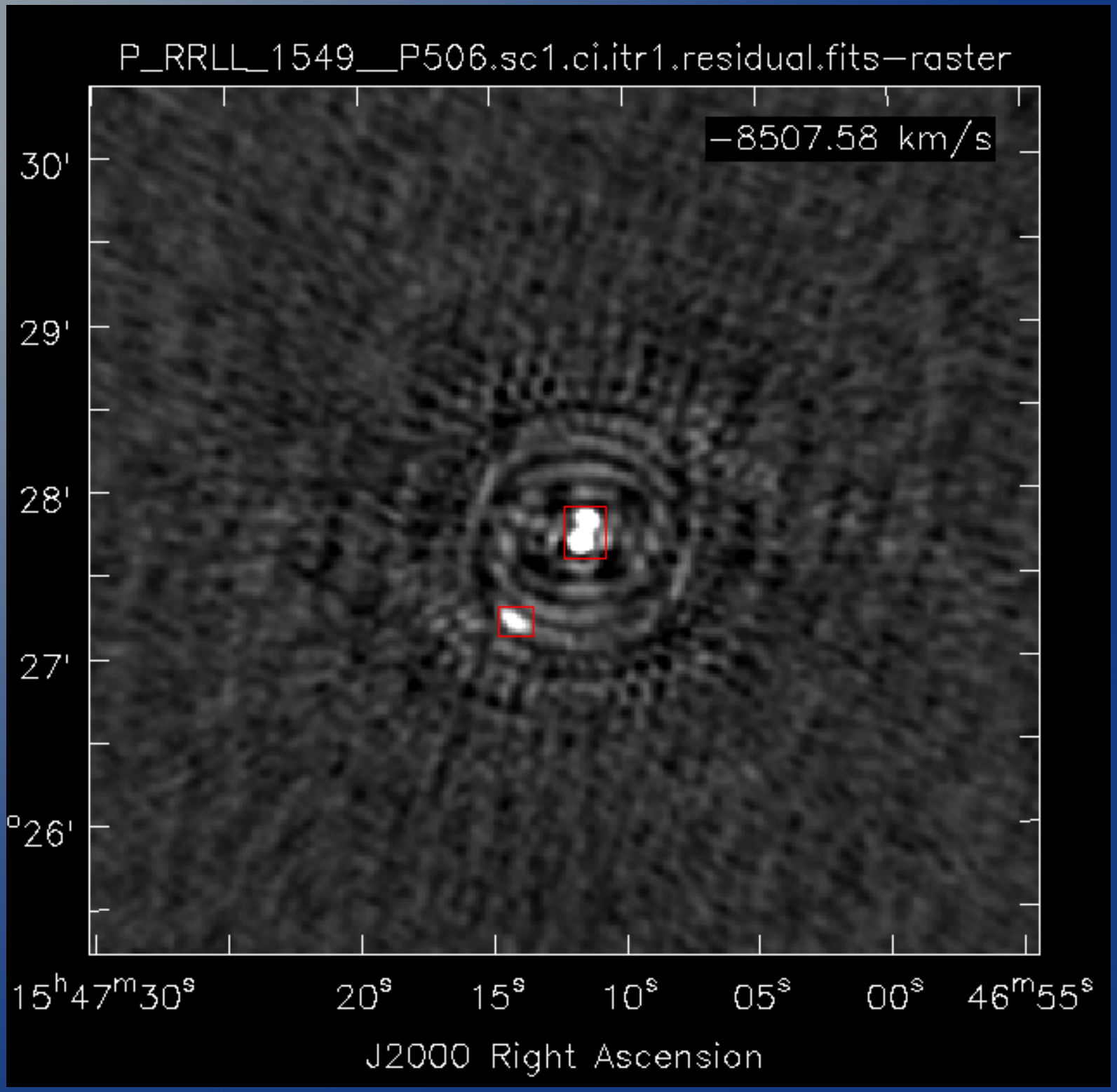
Future plans

- Upgrade pipeline for extended sources
- Upgrade to new version of CASA:
  - casa-4.5.2 (current pipeline version)
  - New features, algorithms and Parallel/distributed execution
- Parallactic angle dependent Primary Beam correction
- Light-curves (In Progress):
  - intra-day variability of points sources in field of phase calibrators.
- Develop pipeline for uGMRT data

Thank You!


P_RRLL_1549__P506.sc1.ci.itr0.residual.fits-raster
-8507.58 km/s
30''
39'00''
30''
38'00''
30''
50°37'00''
30''
J2000 Declination
15h49m27s
21s
18s
15s
12s
09s
J2000 Right Ascension

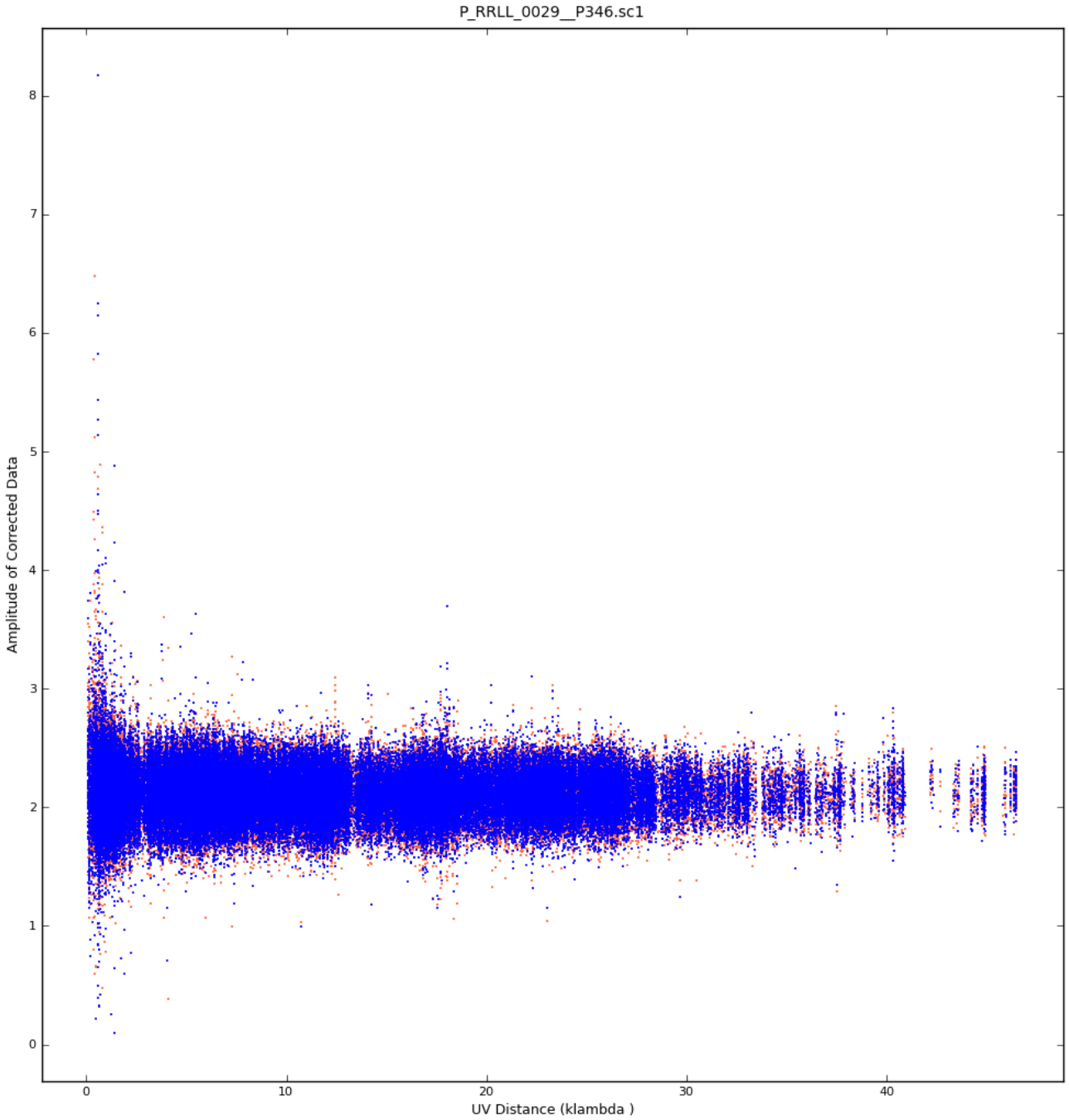

P_RRLL_1549__P506.sc1.ci.image-raster
30''
39'00''
30''
38'00''
30''
50°37'00''
30''
J2000 Declination
15h49m27s
21s
18s
15s
12s
09s
J2000 Right Ascension

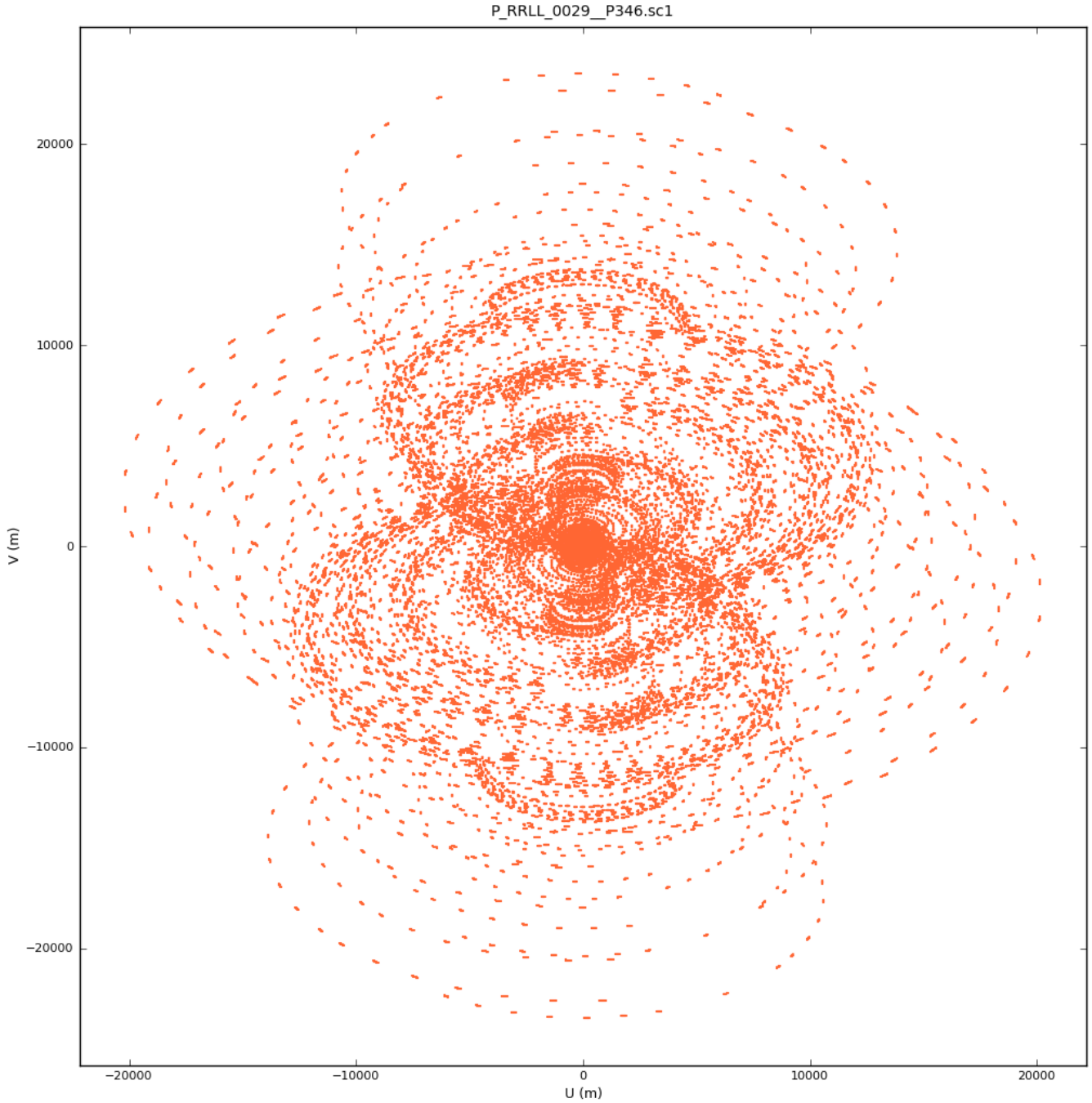

P_RRLL_1549__P506.sc1.ci.itr1.residual.fits-raster
-8507.58 km/s
30'
29'
28'
27'
°26'
15h47m30s
20s
15s
10s
05s
00s
46m55s
J2000 Right Ascension

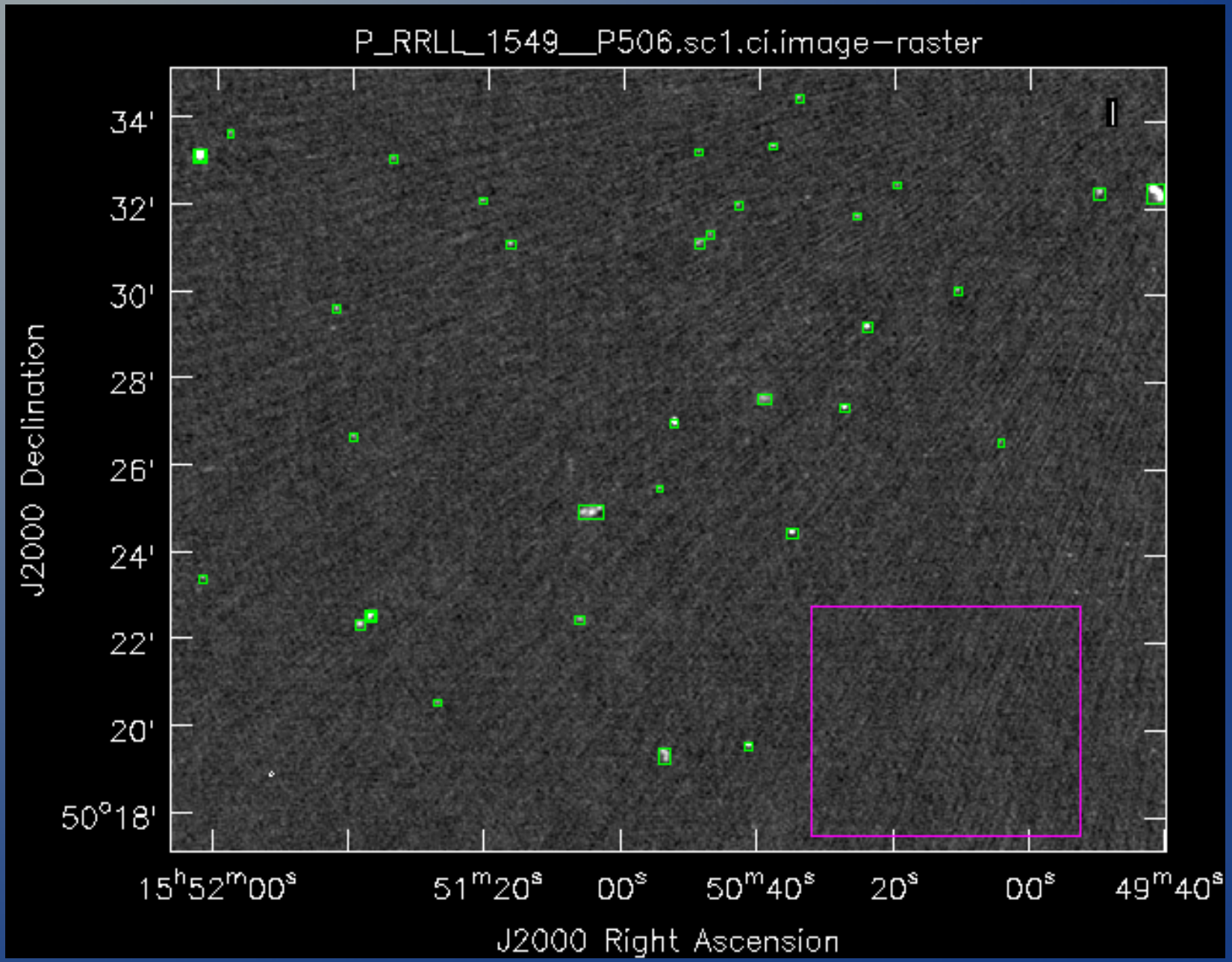
Output Data, Logs and Plots

- 'flagcal' keeps flagging information per source/antenna/stokes, reference antenna, timing information
- All information that CASA shell + imaging.py prints is kept in log, which contains the input parameters to every step in pipeline
- All tasks of CASA output is retained in casalog
- Required final images are converted to fits
- If required by user, UVPLOT, UVCOVERAGE are plotted
- PyBDSM keeps log of all fitting, residual, models and plenty of other plots and information

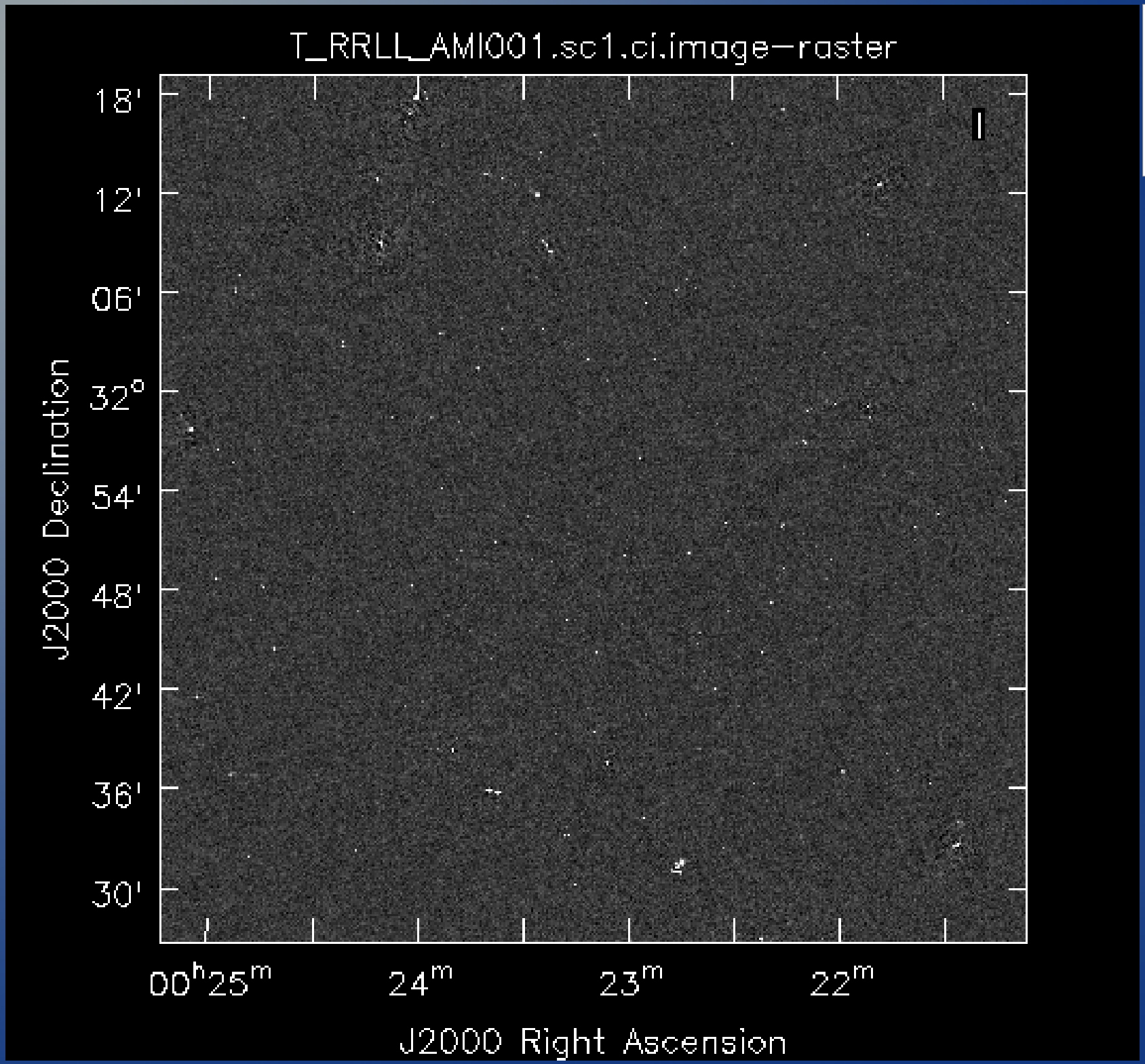
- Writes boxes in “crtf” (casa region text format) format for the region which passed the selection criteria
- Puts upper limit on no. of boxes to be written to 'crtf' file, via selection of thresholds which is 'fraction of the flux of peak flux' for which box to put
- 'rms_box' size to be defined by user is key parameter to make proper box, or avoid putting box on sidelobe
- Other parameters island_rms, and peak_rms defines islands of continuous pixel which are greater than island_rms* RMS and contains at least one pixel which is greater than peak_rms*RMS. RMS is local rms in RMS image
- Provision to generate elliptical clean region if the beam is highly elliptical (e.g. a/b=>2.5, user defined). Useful for the case when observation is snapshot of the source of southern dec
- Used PyBDSM to generate list of outlier detection which are greater than some threshold (say X% of flux if greater than Y time clipped sigma, X & Y users input.) Used in Flanking Field Search


P_RRLL_0029__P346.sc1
Amplitude of Corrected Data
UV Distance (klambda)
0
1
2
3
4
5
6
7
8
0
10
20
30
40

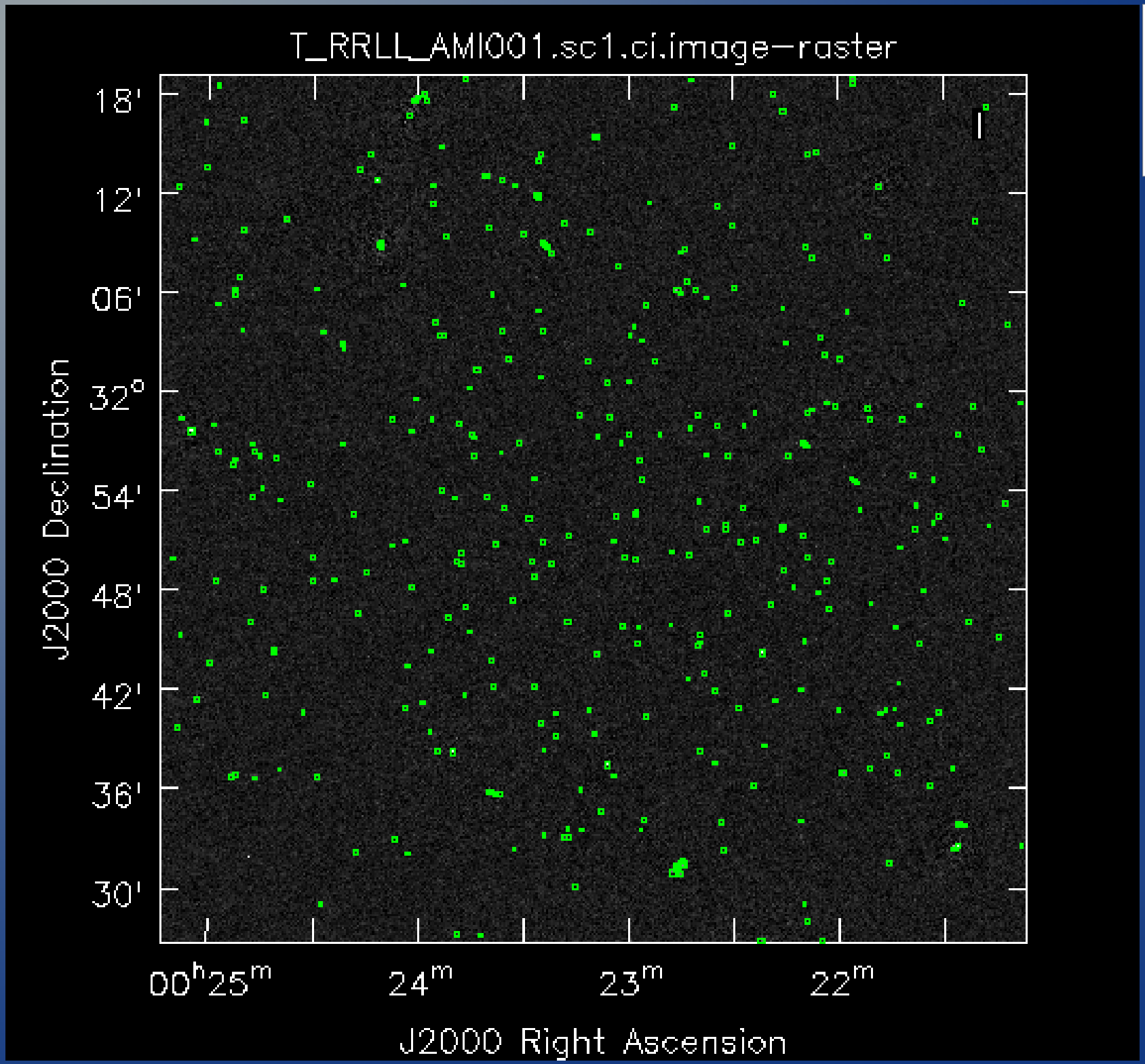

P_RRLL_0029__P346.sc1
20000
10000
0
−10000
−20000
V (m)
−20000
−10000
0
10000
20000
U (m)



93.4 min
61.72 uJy/beam,
610/33.3 MHz

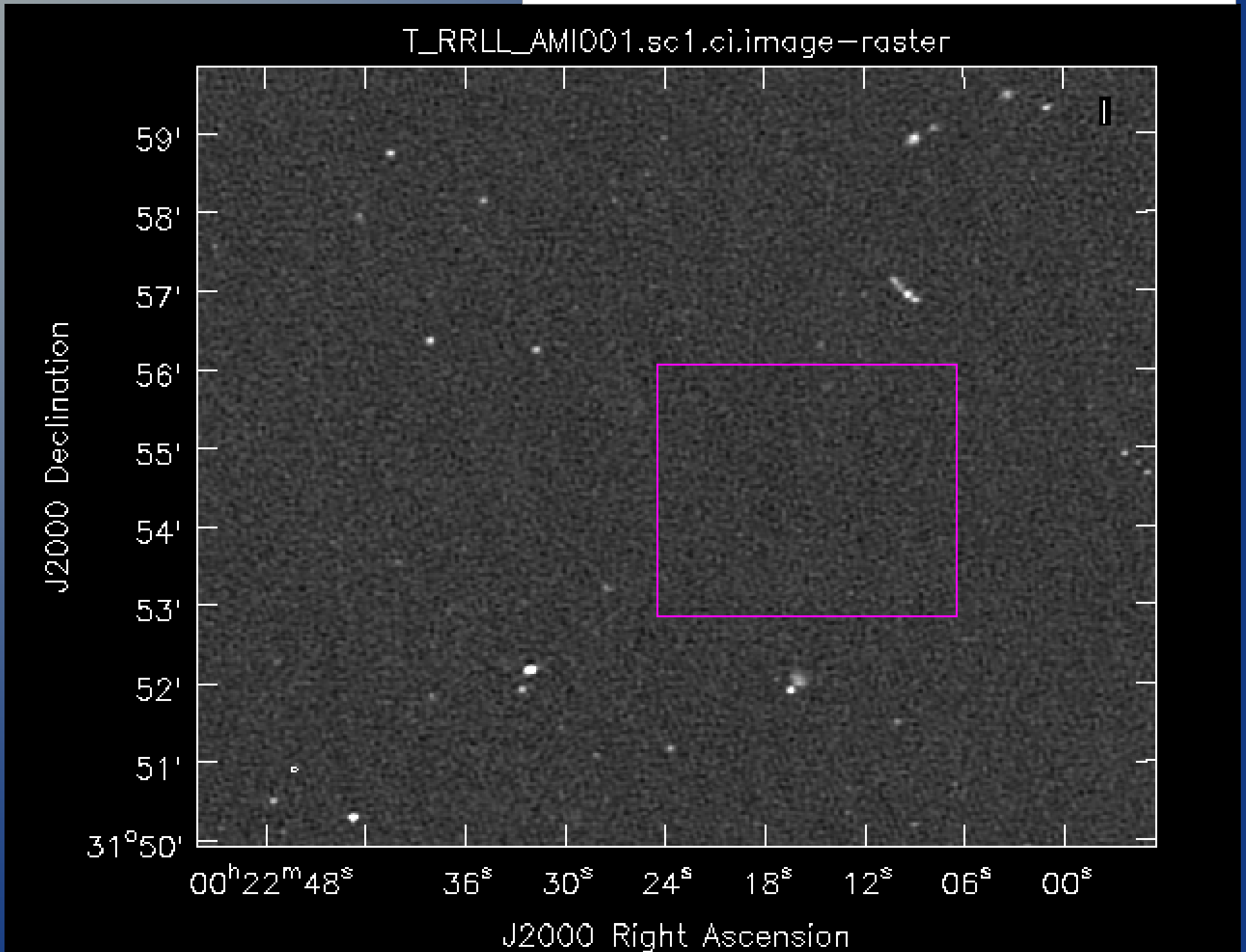


610/33 MHz
6.92 Hrs,
~350 boxes
31.54 uJy/beam

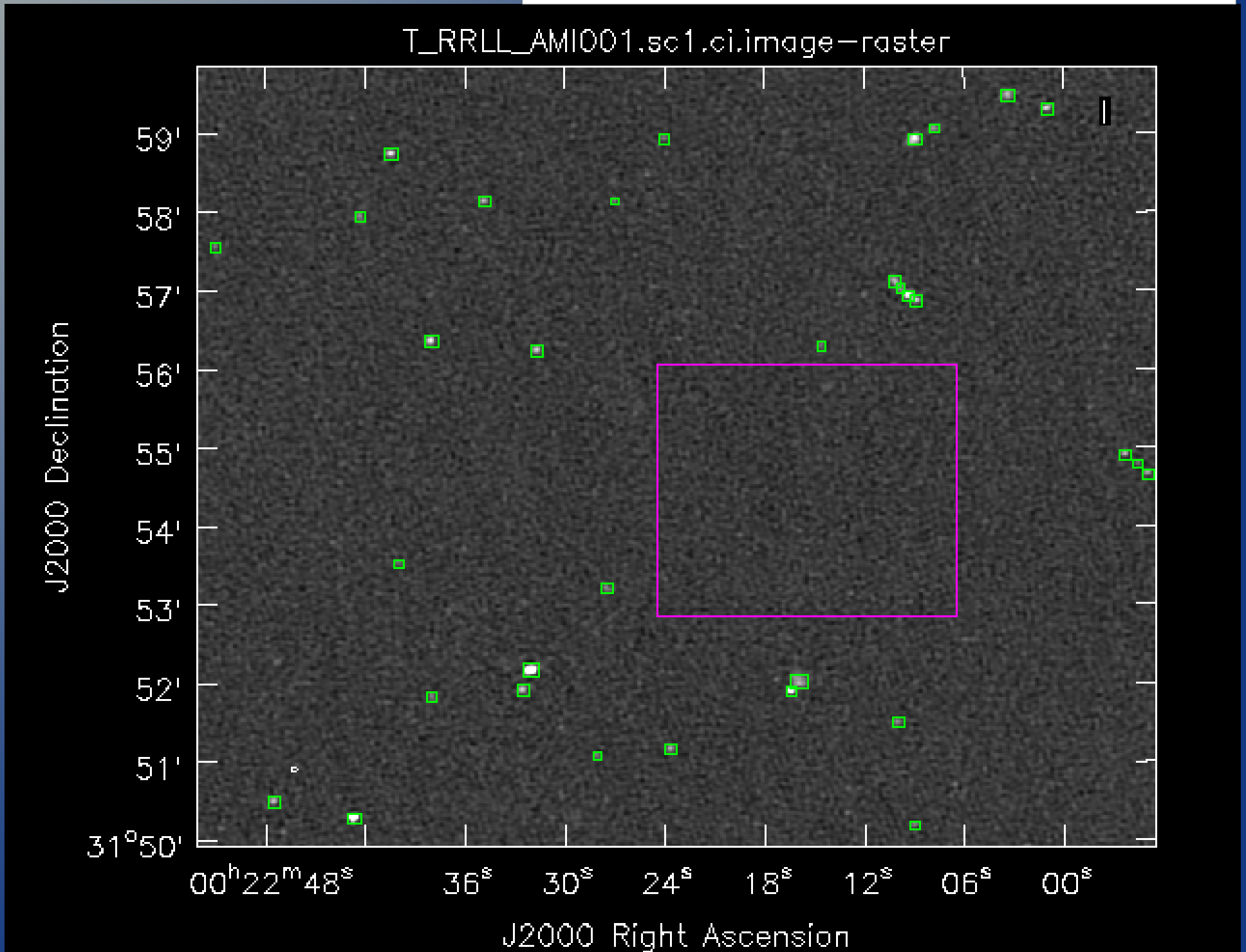


610/33 MHz
6.92 Hrs,
~350 boxes
31.54 uJy/beam

610/33.33 MHz 6.92 Hrs, 31.54 uJy/beam



610/33.33 MHz 6.92 Hrs, 31.54 uJy/beam



Modification to CASA autoclean

- 'autoclean' is an iterative process in which box-making and cleaning runs in loop, until cleaning threshold is reached
- Original CASA autoclean was closing the loop without cleaning all sources defined by cleaning threshold, some times not going to cleaning loop at all
- Generating clean region (mask) file from the boxes was taking huge time
- No provision for w-projection/widefield and multiscale deconvolution in autoclean to pass to clean
- So, we retained the shell structure of autoclean which passes the parameters to clean, and replaced box-making algorithm
- For box-making used Call_PyBDSM.get_islands()
  - Added new routine get_islbox() which will read residual image of last iteration, thresholds and run PyBDSM to get clean boxes
  - Make islands of continuous pixel which are greater than some threshold and contains at least one pixel with value greater than some other threshold