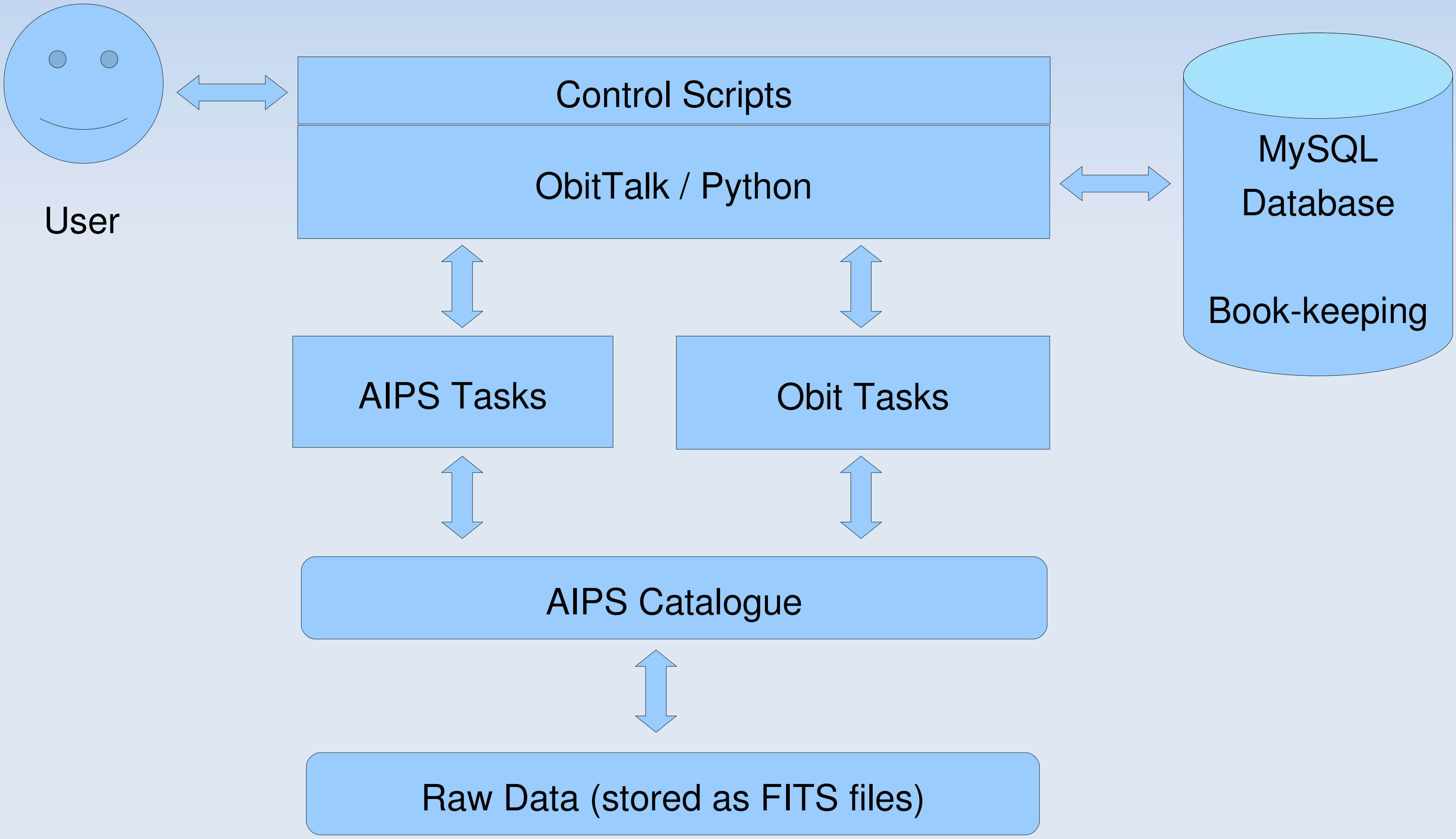
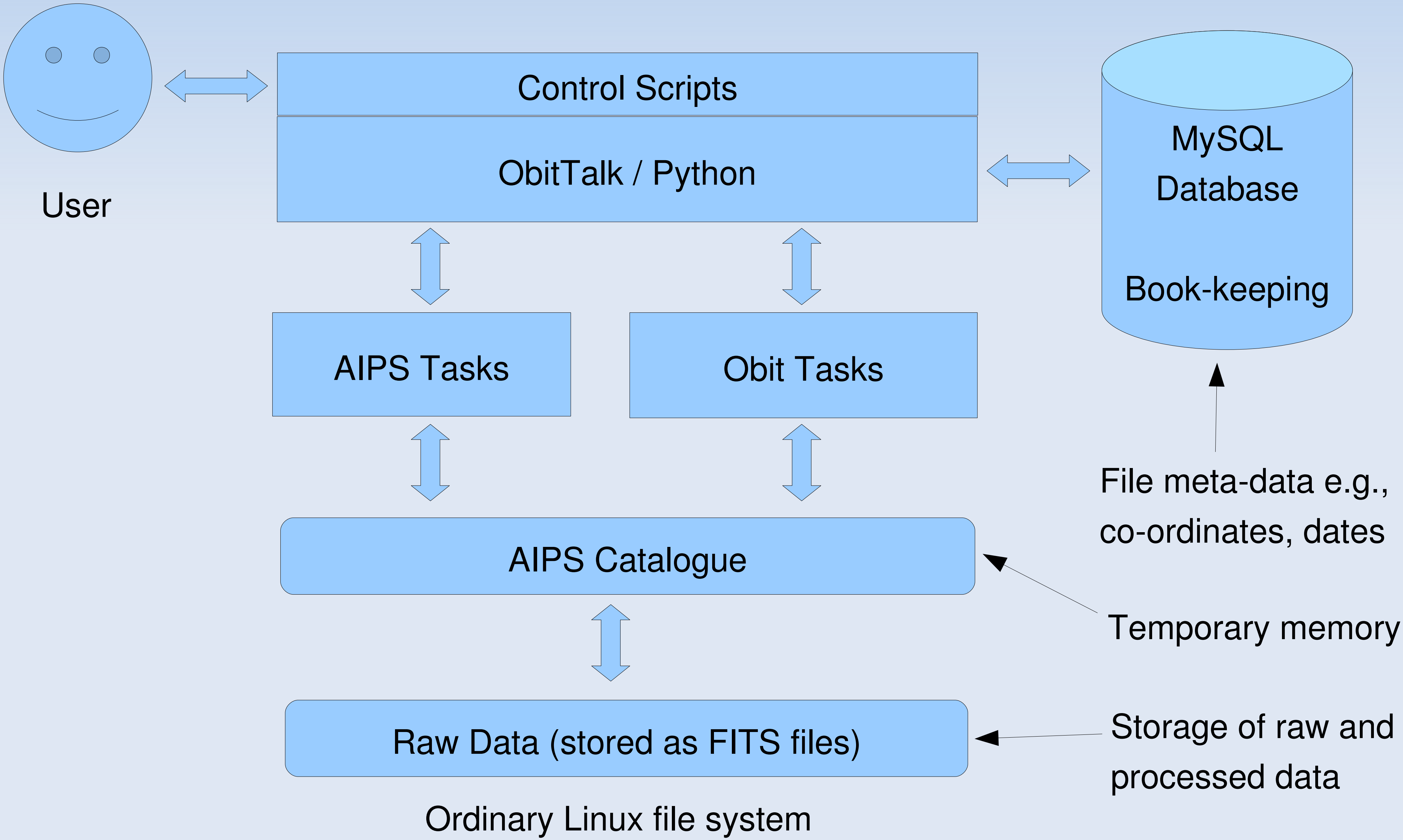
The CORNISH data reduction pipeline

Architecture - Overview



Architecture - Overview



Architecture - Control Scripts

CalibrateCORNISH.py

ImageCORNISH.py

MosaicCORNISH.py

Architecture - Control Scripts

block-name.config

Flagging &
Calibration parameters

in

Image.config

Generic Imaging,
Cleaning &
Self-cal settings

in

Configuration
Files

CalibrateCORNISH.py

ImageCORNISH.py

MosaicCORNISH.py

Architecture - Control Scripts

block-name.config

Flagging &
Calibration parameters

Image.config

Generic Imaging,
Cleaning &
Self-cal settings

Configuration Files

in

CalibrateCORNISH.py

out

Individual calibrated
integrations
50 seconds
(UV-FITS files)

in

ImageCORNISH.py

out

Individual imaged
fields / pointings

(Image-FITS files)

MosaicCORNISH.py

out

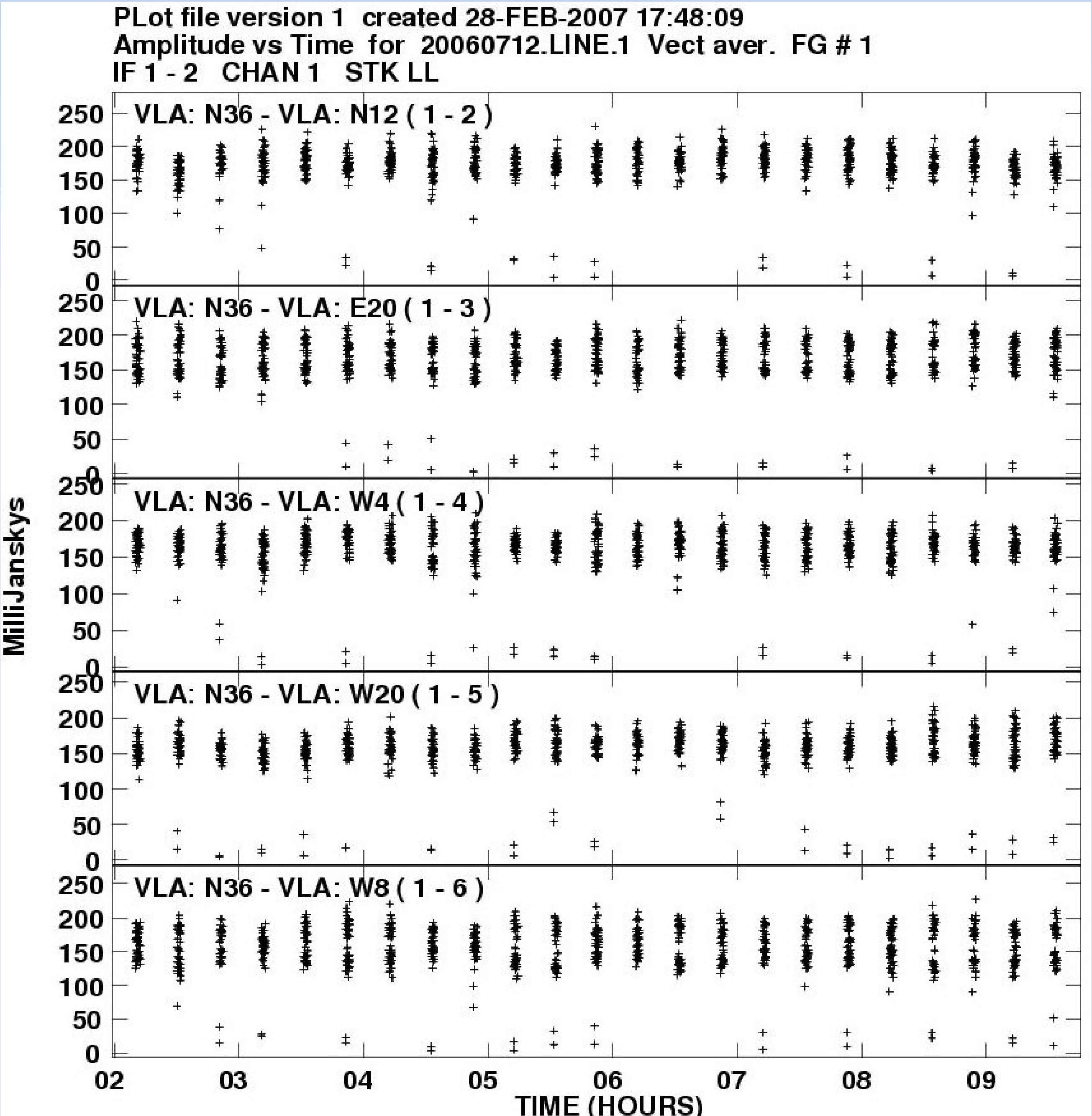
Individual mosaiced
tiles
3920px = ~19.6'
(Image-FITS files)

CalibrateCORNISH.py

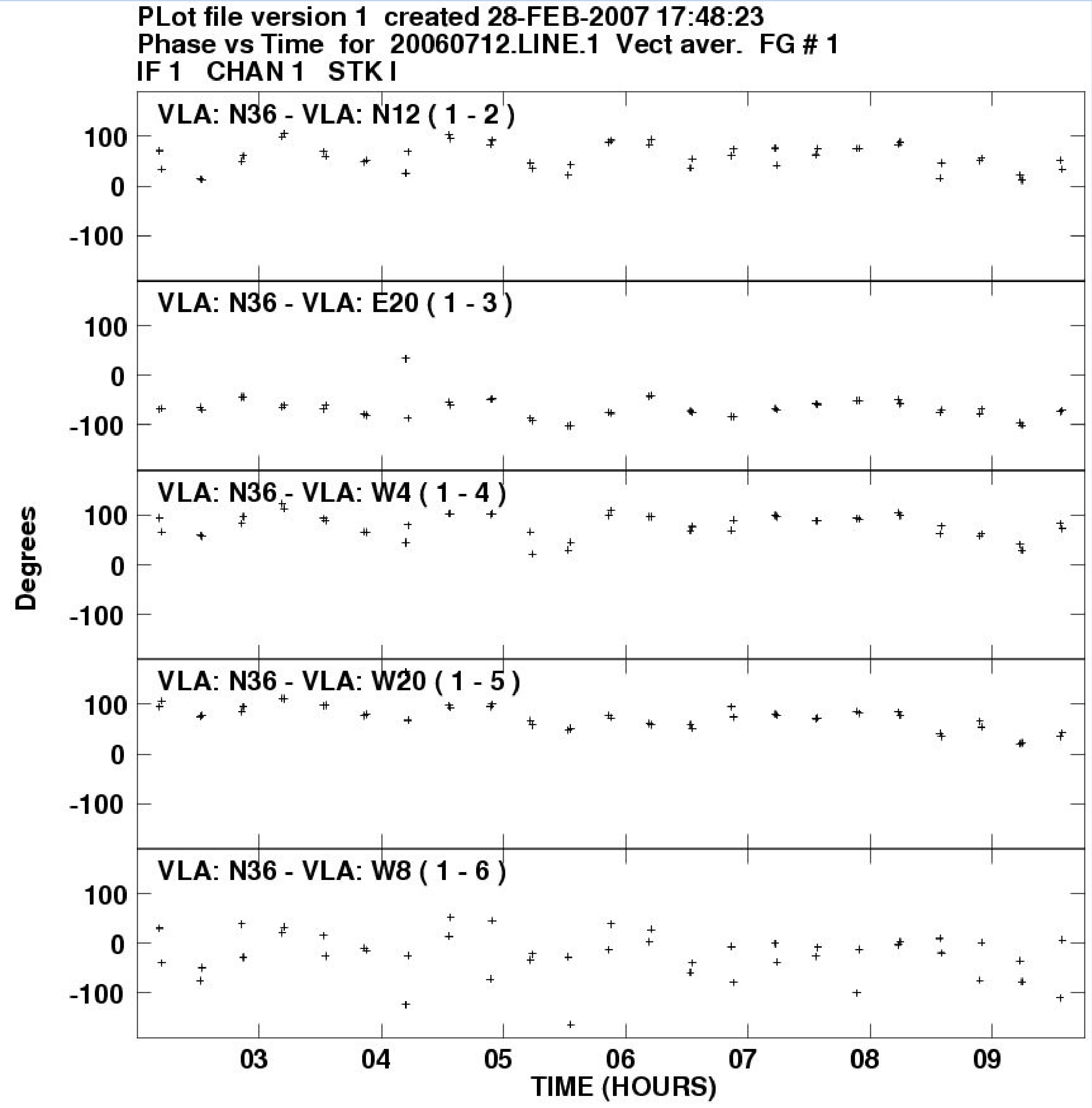
1. Run pre-set UVFLAG commands
2. Flag for gross amplitude spikes (phase-cal)
3. Calibrate absolute flux & gains vs time
4. Flag the science targets based on Stokes V, I and RMS noise (Obit AutoFlag)
5. Split into 50 second pointings and save each as an individual UV-FITS file on disk
6. Populate the database with meta-data
   - (field co-ordinates, times, durations etc.)

```
#=========================== Initial flagging ============================#

# Reset flagging at start of reduction?
reset_flags=y

# Dead antennas to be flagged out
bad_ants=13 14 16 18 24 26

# UVFLG commands to be run before calibration, e.g.
#uvflg=sources=1832-105 antennas=2 stokes=LL BIF=1 EIF=2 reason=Initial_flagging
 freqid=1 timerang=0,2,1,0,0,2,2,0

# Spike on antenna 4 (check)
uvflg=sources='' antennas=4 stokes=I BIF=1 EIF=2 reason=Initial_flagging freqid=
1 timerang=0,3,48,40,0,3,50,2

# Bad times on the flux-cal
uvflg=sources='' antennas=0 stokes=I BIF=1 EIF=2 reason=Initial_flagging freqid=
1 timerang=0,1,10,0,0,1,12,0

# Bad times on the backup-cal
uvflg=sources='' antennas=0 stokes=I BIF=1 EIF=2 reason=Initial_flagging freqid=
1 timerang=0,8,55,0,0,8,57,20

# Autoflag the uncalibrated phase-calibrator for gross amplitude & noise spikes
pcal_flag_iclip=100                         # Clip on stokes I level
pcal_flag_vclip=100                         # Clip on diff between LCP and RCP
pcal_flag_rms=0.5                           # RMS clip level @ 2x sample time
#pcal_flag_rms_mod=                         # clip= flag_rms + amp*rms_mod

#========================= Source types and names ========================#

flux_cal=1331+305                           # Flux (primary) calibrator
phase_cal=1832-105                          # Phase (secondary) calibrator
backup_cal=0137+331                         # Backup flux calibrator

#========================= Calibration parameters ========================#

calibrate_uvdata=y                          # Calibrate the UV data
fluxcal_ants=6 7 8 9 10 11 15 17 19 26      # Antennas to use with flux-cal
fluxcal_uvrange=0 25                        # UV-range where flux-cal is a point
refant=8                                    # Refrence antenna
solint=30                                   # Solution interval (sec)
save_calib_uvdata=n                         # Save the calibrated UV data?

#======================= Flag the science sources ========================#

# Autoflag the science sources for amplitude and noise spikes
sci_flag_iclip=100
sci_flag_vclip=20
sci_flag_rms=0.9
-  20060723.config      (Shell-script[bash])--L38--33%--------------------------
```

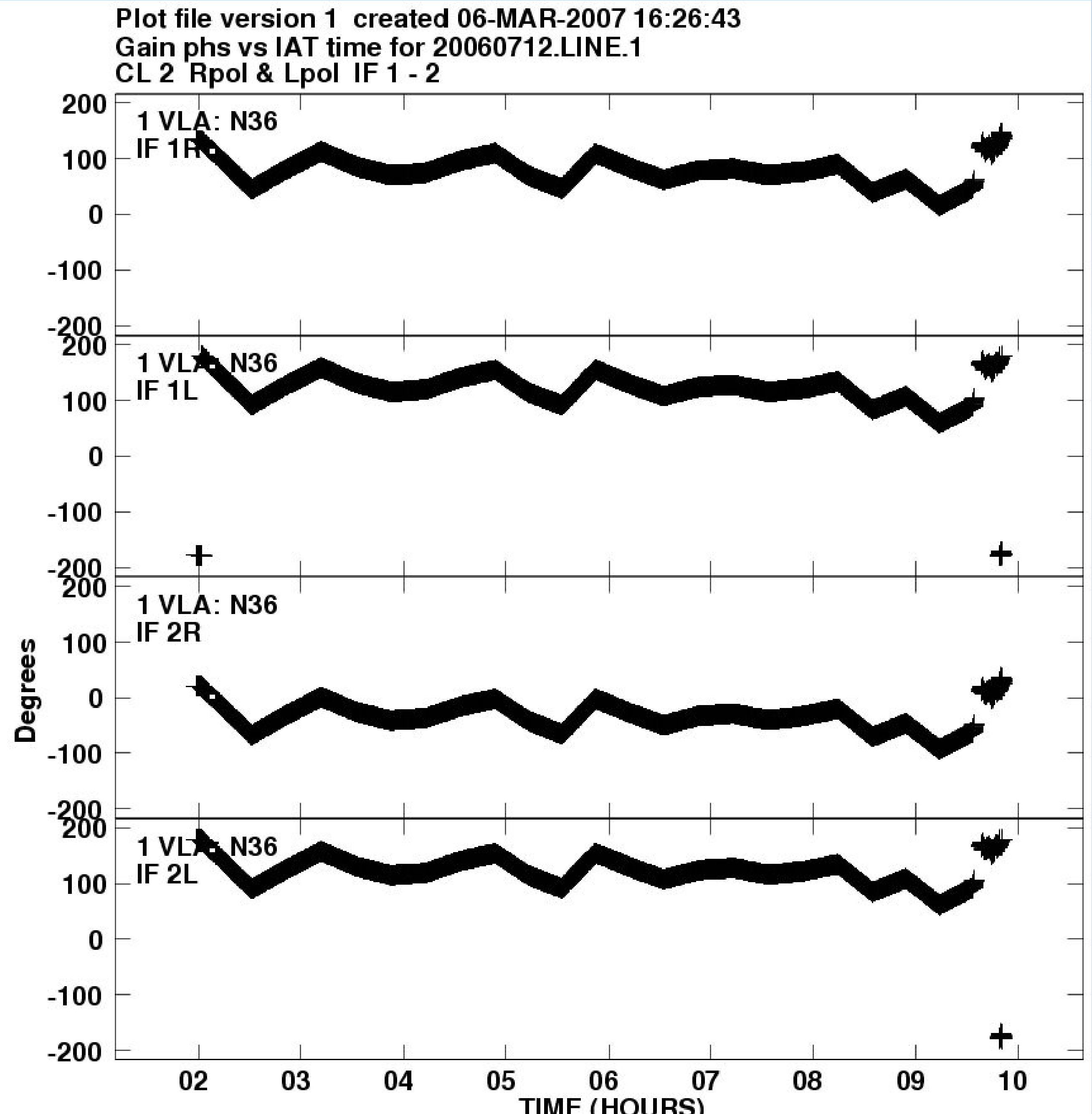
CalibrateCORNISH.py



CalibrateCORNISH.py



CalibrateCORNISH.py



CalibrateCORNISH.py

cpurcell@sprocket: ~/CORNISH/pipeline/data/uvfits_split

File Edit View Terminal Tabs Help

```
2006-07-12_0.06.42.15.fits  2006-07-21_0.03.37.05.fits  2006-07-21_0.08.33.55.fits
2006-07-12_0.06.43.05.fits  2006-07-21_0.03.37.55.fits  2006-07-21_0.08.34.45.fits
2006-07-12_0.06.43.55.fits  2006-07-21_0.03.38.45.fits  2006-07-21_0.08.35.35.fits
2006-07-12_0.06.44.45.fits  2006-07-21_0.03.39.35.fits  2006-07-21_0.08.36.25.fits
2006-07-12_0.06.45.35.fits  2006-07-21_0.03.40.25.fits  2006-07-21_0.08.37.15.fits
2006-07-12_0.06.46.25.fits  2006-07-21_0.03.41.15.fits  2006-07-21_0.08.38.05.fits
2006-07-12_0.06.47.15.fits  2006-07-21_0.03.42.05.fits  2006-07-21_0.08.38.55.fits
2006-07-12_0.06.48.05.fits  2006-07-21_0.03.42.55.fits  2006-07-21_0.08.39.45.fits
2006-07-12_0.06.48.55.fits  2006-07-21_0.03.43.45.fits  2006-07-21_0.08.40.35.fits
2006-07-12_0.06.49.45.fits  2006-07-21_0.03.44.35.fits  2006-07-21_0.08.41.25.fits
2006-07-12_0.06.50.35.fits  2006-07-21_0.03.45.25.fits  2006-07-21_0.08.42.15.fits
2006-07-12_0.06.51.50.fits  2006-07-21_0.03.46.15.fits  2006-07-21_0.08.43.05.fits
2006-07-12_0.06.54.10.fits  2006-07-21_0.03.47.05.fits  2006-07-21_0.08.43.55.fits
2006-07-12_0.06.55.15.fits  2006-07-21_0.03.47.55.fits  2006-07-21_0.08.44.45.fits
2006-07-12_0.06.56.05.fits  2006-07-21_0.03.48.45.fits  2006-07-21_0.08.45.35.fits
2006-07-12_0.06.56.55.fits  2006-07-21_0.03.49.35.fits  2006-07-21_0.08.46.25.fits
2006-07-12_0.06.57.45.fits  2006-07-21_0.03.50.25.fits  2006-07-21_0.08.47.15.fits
2006-07-12_0.06.58.35.fits  2006-07-21_0.03.51.15.fits  2006-07-21_0.08.48.05.fits
2006-07-12_0.06.59.25.fits  2006-07-21_0.03.52.05.fits  2006-07-21_0.08.48.55.fits
2006-07-12_0.07.00.15.fits  2006-07-21_0.03.53.10.fits  2006-07-21_0.08.50.10.fits
2006-07-12_0.07.01.05.fits  2006-07-21_0.03.55.50.fits  2006-07-21_0.08.56.20.fits
2006-07-12_0.07.01.55.fits  2006-07-21_0.03.56.55.fits
cpurcell@sprocket:~/CORNISH/pipeline/data/uvfits_split$
```

ImageCORNISH.py

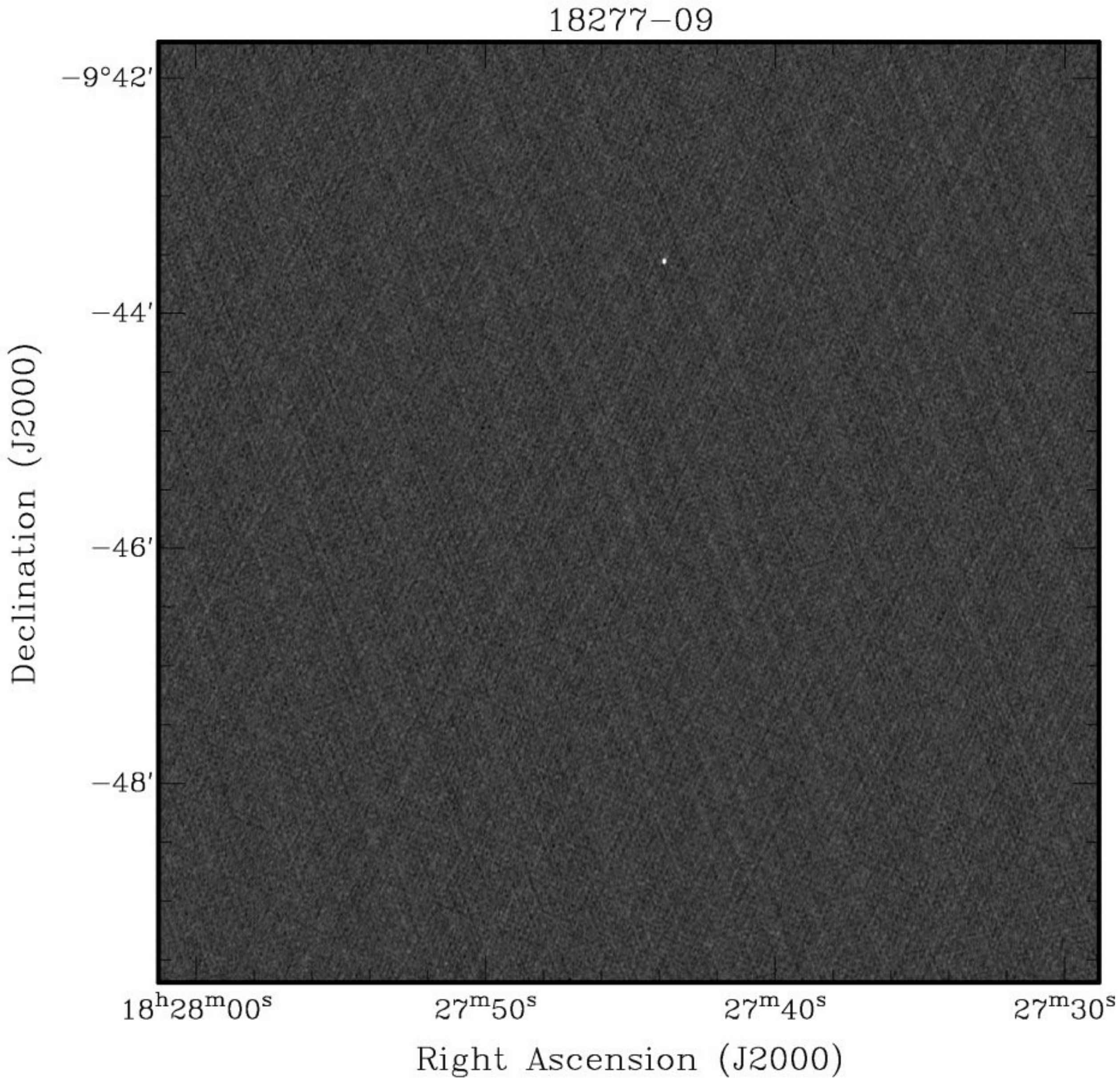
- Given a field-name and/or a date:

1. Find the fields and associated 50-s scans
2. Accumulate (DBCON) scans towards the same pointing (or not, if asked by user)
3. Run the Obit Imager in auto-window mode
4. Write out FITS images of each field
   - either a file for each pointings position
   - or a file for individual 50-s scans (QC)
5. Populate database with image parameters

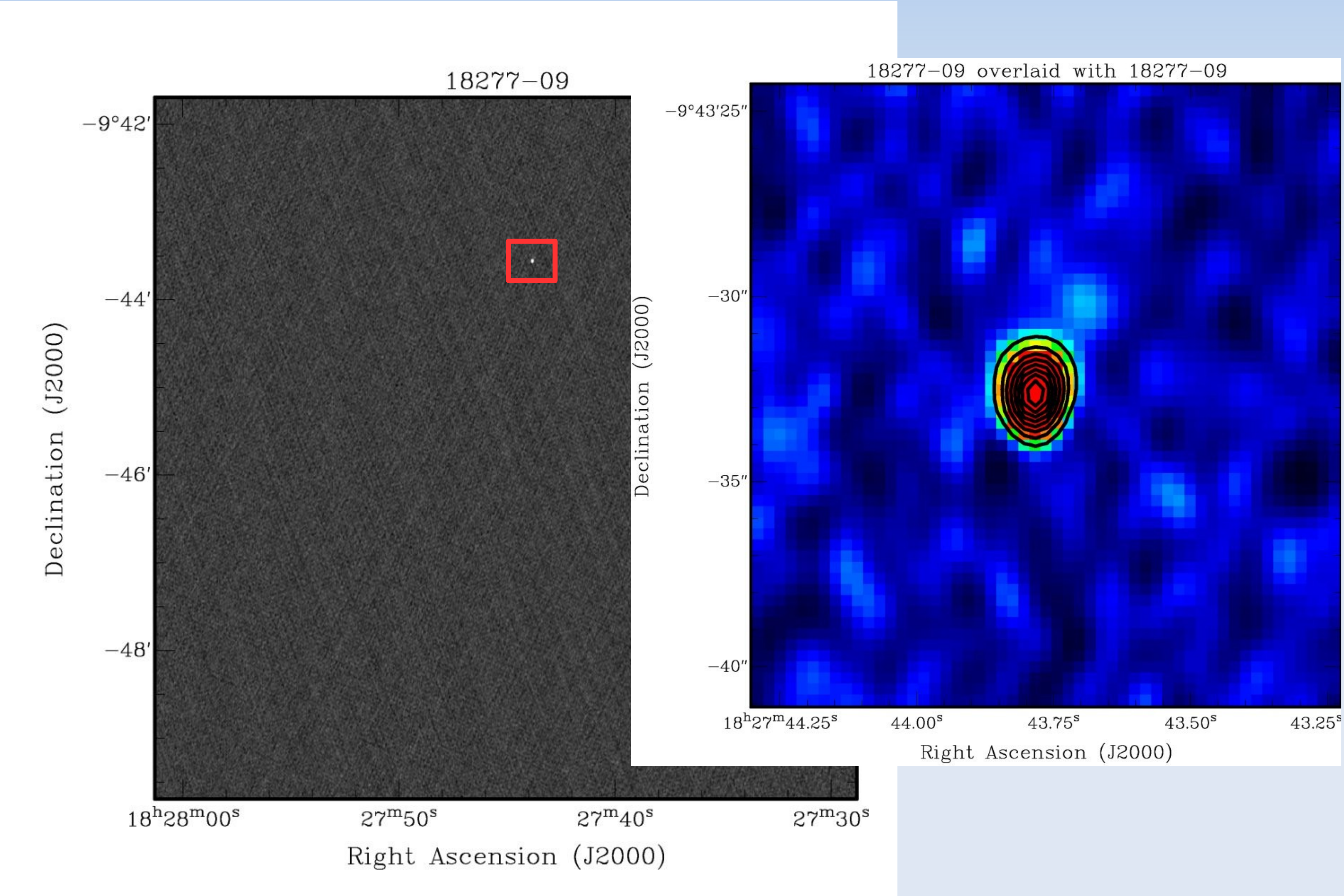
ImageCORNISH.py

- Imaging parameters:
  - Force a symmetric beam (1.4”)
  - Field of view – 8'
  - clean cutoff = 0.4 mJy
  - 2 self-cal passes, trigger on 5 mJy peaks
  - Place clean boxes at locations of NVSS sources with > 5 mJy peaks

ImageCORNISH.py



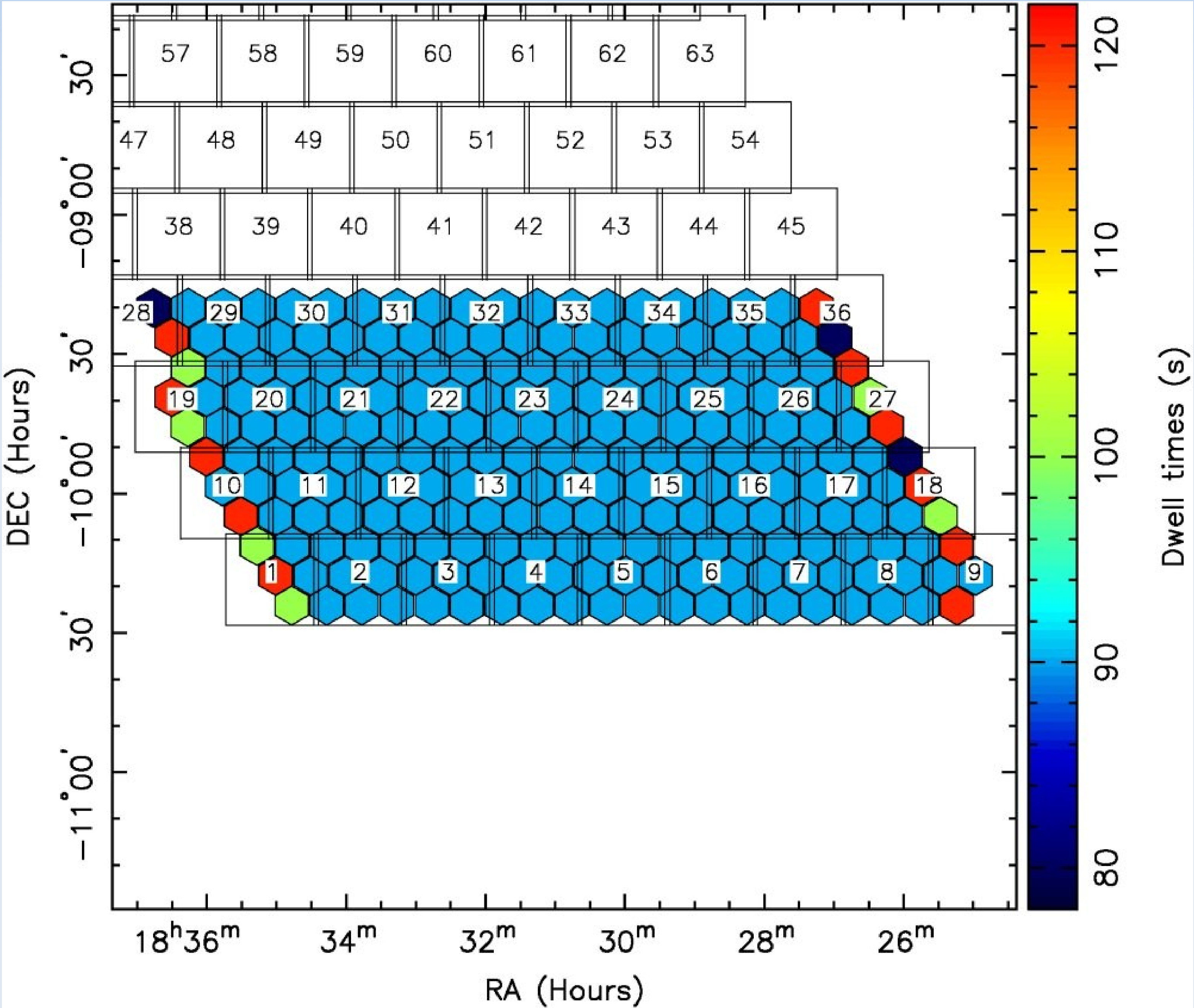
ImageCORNISH.py



MosaicCORNISH.py

- Tiles 19.6' on a side
  - 0.3” per pixel X 3920 pixels
- 9 tiles across a row
- Layout chosen to cover all observed sky
- Tiles overlap by 1'
- On average 22 fields contribute to each tile

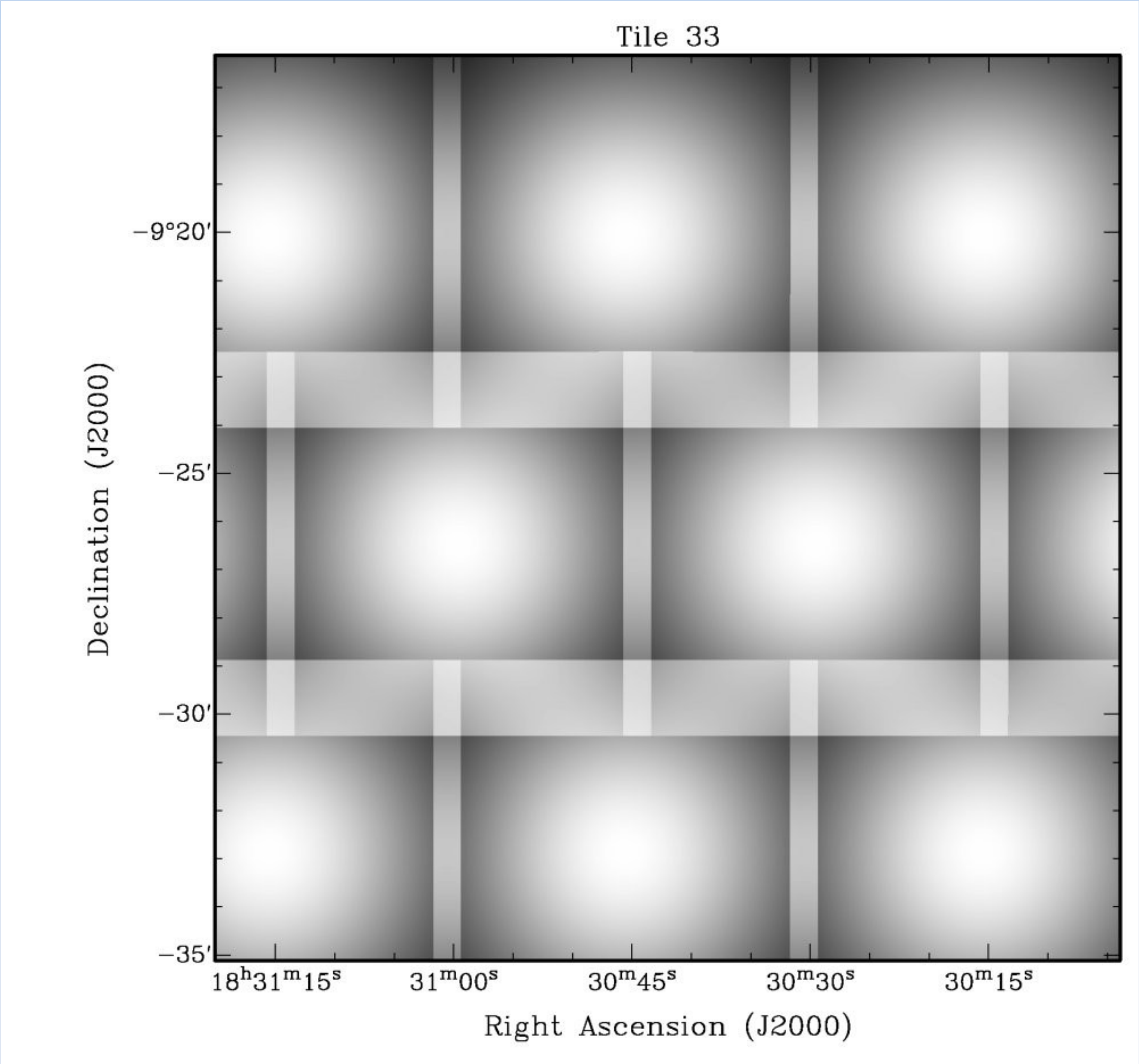
MosaicCORNISH.py



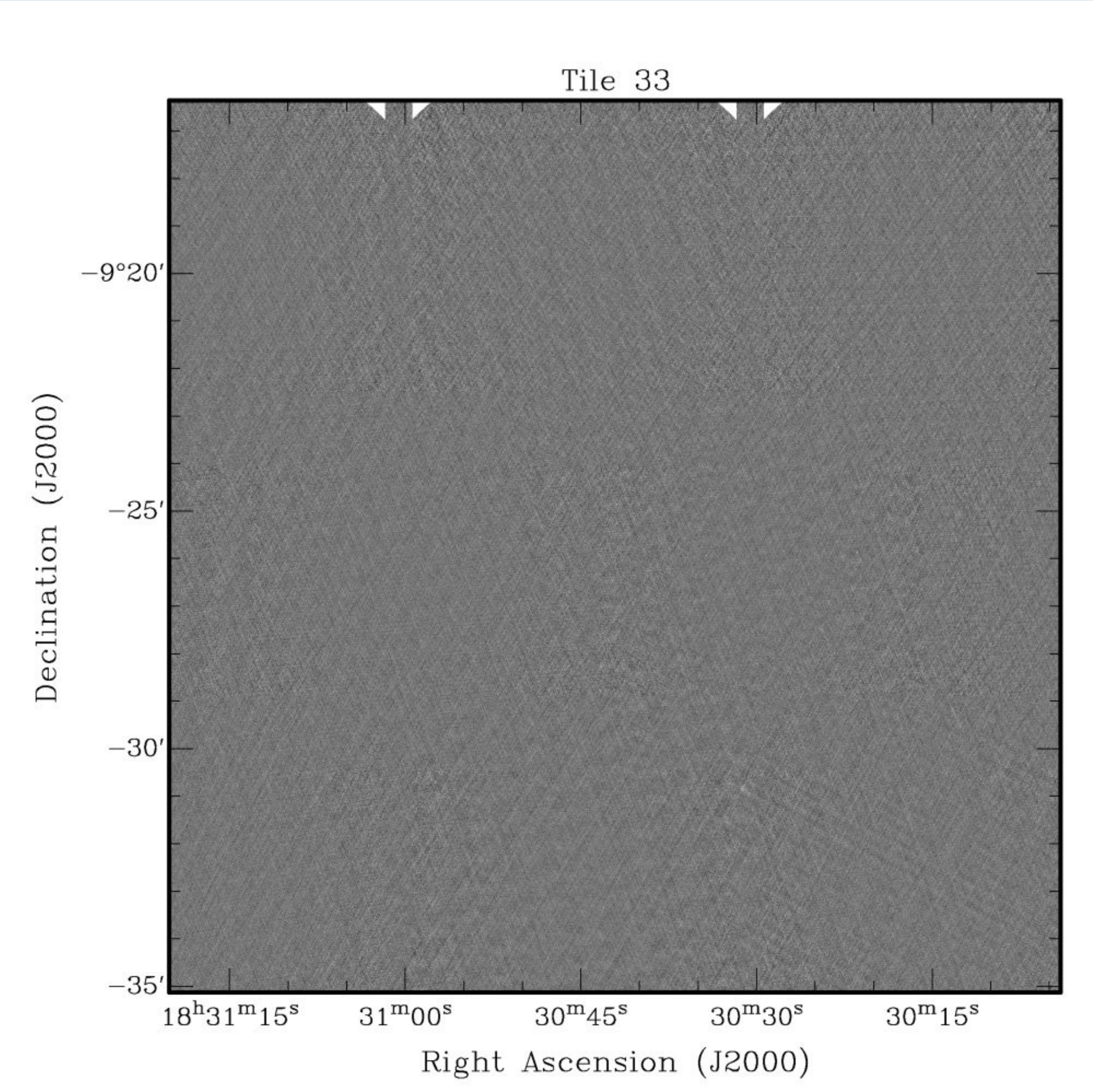
MosaicCORNISH.py

- Given a range in Dec or a list of tile IDs
- Query the DB for fields which overlap with each tile and return their parameters (RA, Dec, pixel-scale etc.)
- Run the Obit Mosaic procedure to accumulate individual fields into a Tile.

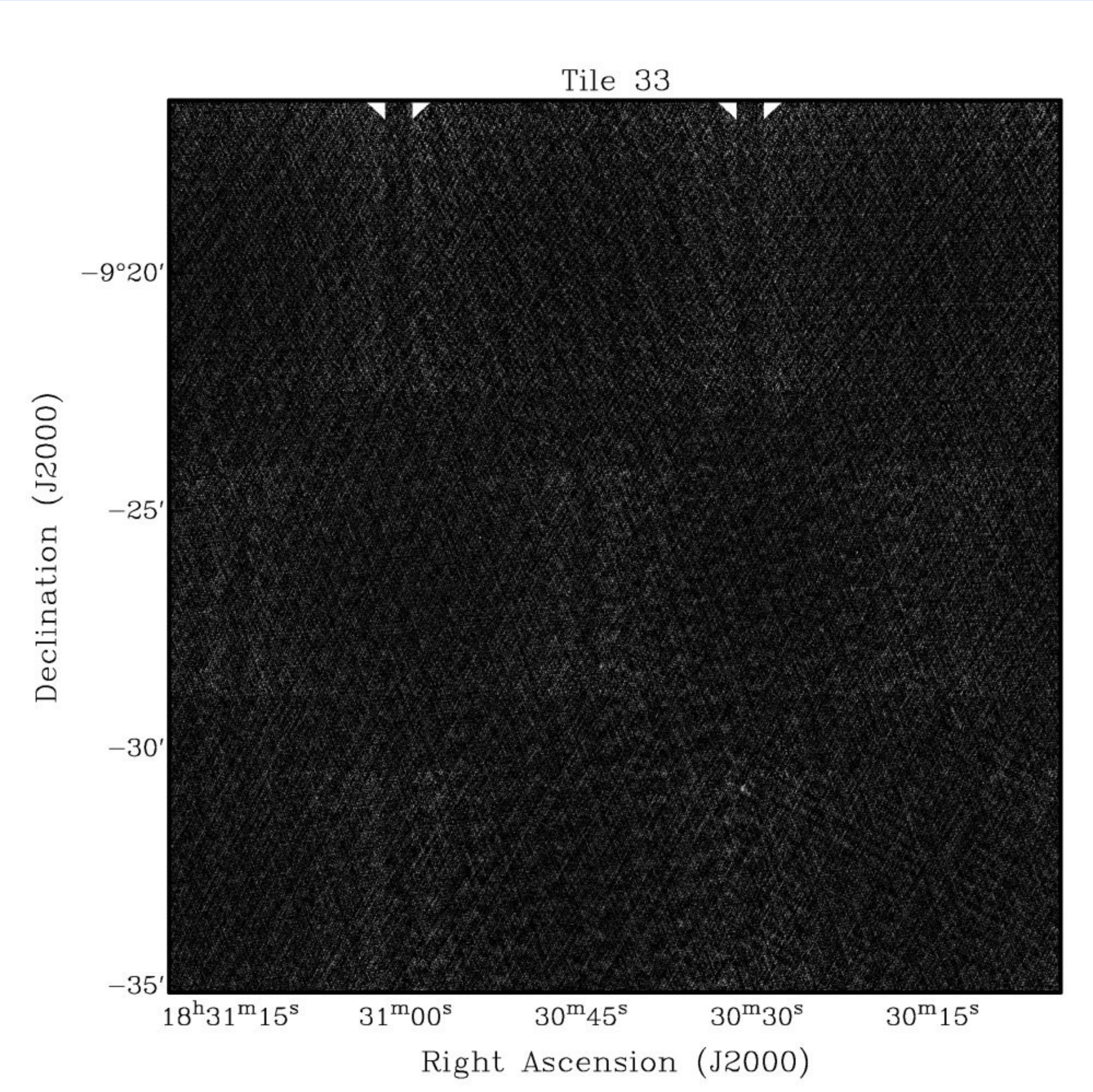
MosaicCORNISH.py



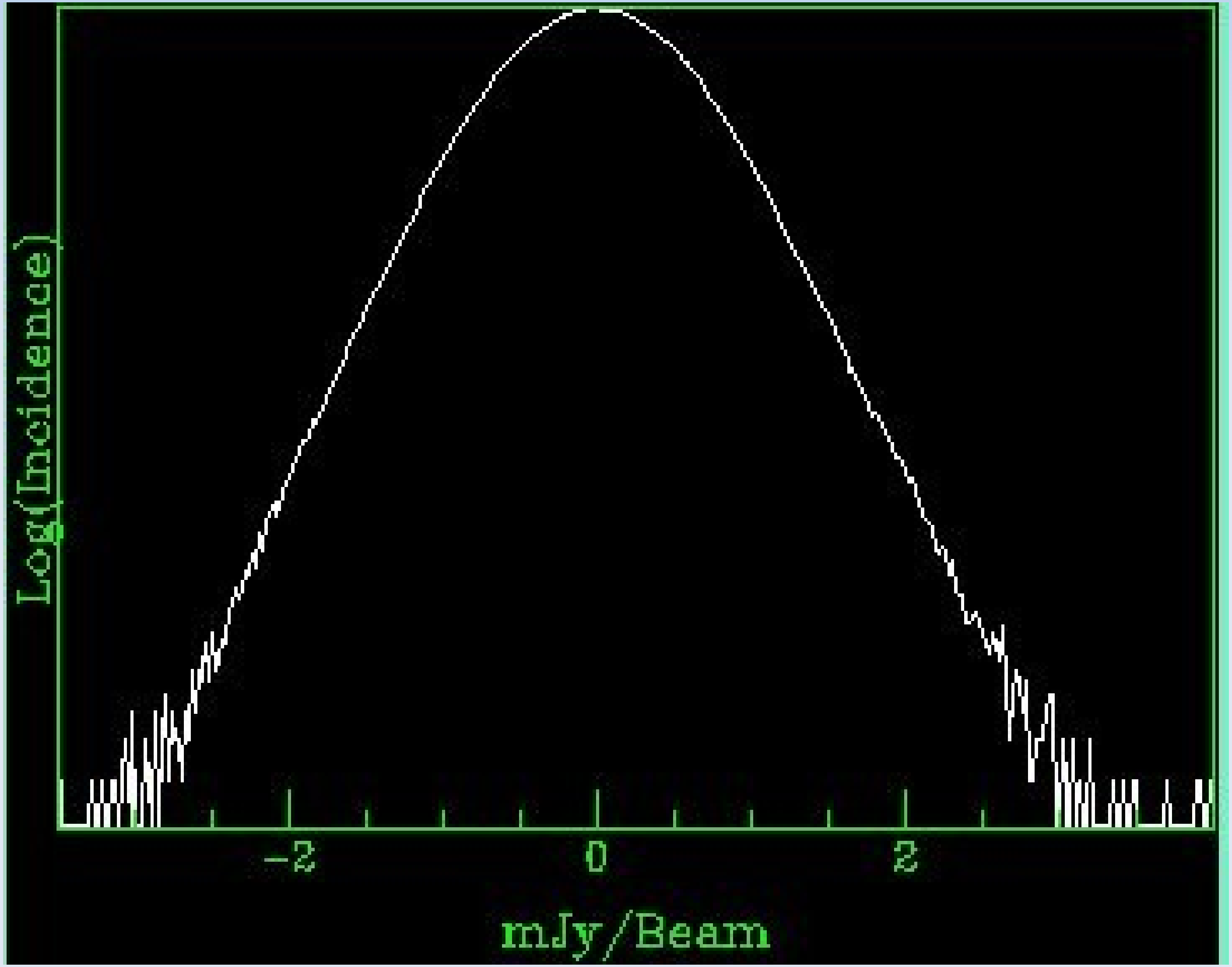
MosaicCORNISH.py



MosaicCORNISH.py



MosaicCORNISH.py



Timescales

- Calibration (per block) = 1 hour
- Imaging (per block) = 4-6 hours
- Mosaicing (per tile) = 20 minutes
- Mosaicing (per block) = 12 hours

Timescales

- Calibration (per block) = 1 hours
- Imaging (per block) = 4-6 hours
- Mosaicing (per tile) = 20 minutes
- Mosaicing (per block) = 12 hours
- Full reduction may be accomplished via 3 commands
  - Easy to reprocess: fire and forget.