The VLA Calibration Pipeline

Data Reduction Workshop, March 2016

Claire Chandler

Atacama Large Millimeter/submillimeter Array
Expanded Very Large Array
Robert C. Byrd Green Bank Telescope
Very Long Baseline Array

NRAO

Pipeline Overview

- With the start of Jansky VLA Full Operations (January 2013), we started a new operational model:
  - Deliver flagged and calibrated visibility data
  - You will self-calibrate and image visibility data to meet science goals, using resources at home institution or NRAO computing resources
- Automated pipeline should run correctly on all "standard" Stokes I science SBs; "standard" means:
  - 128 MHz spws, but may work on other set-ups as well
    - Some constraints on strength of calibrators needed
  - Contains correctly labeled and complete scan intents
- Current versions available:
  - "scripted" pipeline is a collection of python scripts that use CASA tasks wherever possible, but also uses toolkit calls; readable and easy to modify
  - CASA integrated pipeline is compatible with ALMA pipeline infrastructure, improved diagnostics in weblog, used as real-time pipeline since Sep 2015

Pipeline Operation

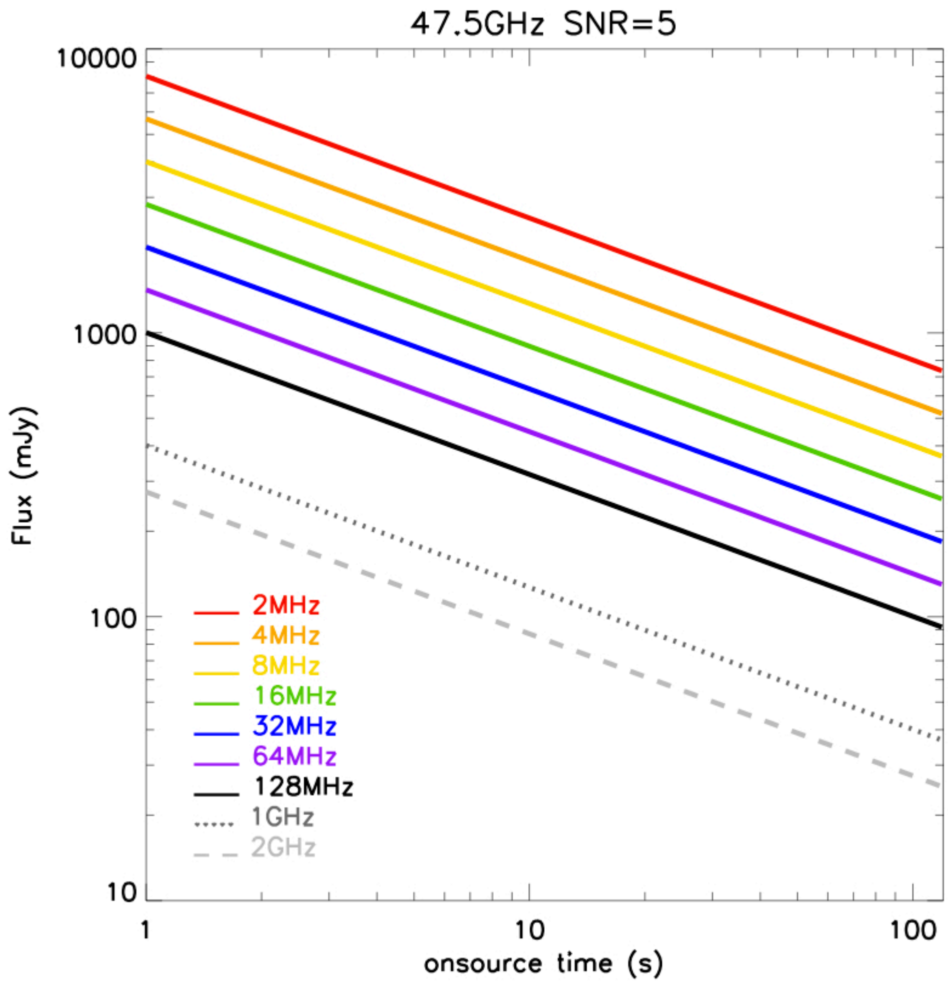
- Real-time pipeline:
  - Minimal human intervention
    - Pipeline is run automatically on *every* science SB as it completes (not just "continuum")
  - Pipeline output undergoes quality assurance checks by NRAO staff upon request; reports generated are archived as pipeline products
- At your home institution:
  - Instructions for installation and operation of the VLA CASA Calibration Pipeline are available at https://science.nrao.edu/facilities/vla/data-processing/pipeline
    - Uses CASA 4.3.1, similar to current real-time pipeline
    - CASA 4.5.2 currently being validated (you are helping with this!)
- Scripted pipelines for CASA versions through 4.5.0 also available
  - Provides more flexibility in how to use the pipeline, options suitable for spectral line datasets, mixed correlator set-ups, multi-band observations, etc.
  - Working to incorporate these into the CASA integrated pipeline

Pipeline Requirements

- "Standard" Stokes I science SB means:
  - 128 MHz spws, but may work on other set-ups as well
    - Can work for narrower BWs, depends on the strength of the calibrators
    - Heuristics currently make some assumptions about the strength of the calibrators, in particular, the delay calibrator
  - Contains correctly labeled and complete scan intents
    - And also that the observation has been set up correctly!
- Will the pipeline work for you?
  - The pipeline successfully completes on ~95% of all science SBs observed on the VLA; whether the output can be used for science depends on the science goal, and whether the observation was correctly set up
    - Pipeline includes Hanning smoothing, RFI flagging, and weight calculations that may not be appropriate for spectral line projects (but can modify scripted pipeline)
    - No polarization calibration (yet) but can use pipeline output as starting data for pol. cal.
    - Will probably work well for data taken since May 2012, may work for earlier EVLA data, likely that extra flagging may be needed in these cases

Pipeline Requirements

- Calibrator strength:
  - Conservative limit on strength of BP and complex gain calibrators can be derived from requirement for initial gain calibration to work at high end of Q-band
  - Heuristic for delay calibration currently requires the SNR=3 limit on initial gain calibration *per integration*



Pipeline Requirements

- Correct observation set-up
  - Independent of whether you want to run the pipeline!
  - Remember: simple observing set-ups are always easier to calibrate
  - Do not skimp on calibration to spend more time on your target – you may end up not being able to calibrate the target data at all
    - Spending 3 minutes pointing could buy you more sensitivity than doubling the time on your target
- Scan intents
  - The pipeline relies entirely on correct *scan intents* to be defined in each SB
  - In order for the pipeline to run successfully on an SB it must contain, *at minimum*, scans with the following intents:
    - A flux density calibrator scan that observes one of the primary calibrators (3C48, 3C138, 3C147, or 3C286) – this will also be used as the delay and bandpass calibrator if no bandpass or delay calibrator is defined
    - Complex gain calibrator scans

(Real-Time) Heuristics (I)

- Assuming requirements are met, the pipeline:
  - Loads the data
  - Hanning smooths**
  - Retrieves information about the observing set-up from the data
  - Applies deterministic flags (online flags, shadowed data, end channels of sub-bands, etc.)
  - Identifies primary calibrators and loads models
  - Derives all prior calibrations (antenna position corrections, gain curves, atmospheric opacity, requantizer gains)
  - Iteratively determines initial delay and bandpass solutions, including running RFLAG (RFI flagging algorithm), and identifying other system (deformatter) problems
  - Derives initial gain solutions, does flux density bootstrapping and derives spectral index of all calibrators

**May want to modify inputs and/or omit entirely for spectral line reductions

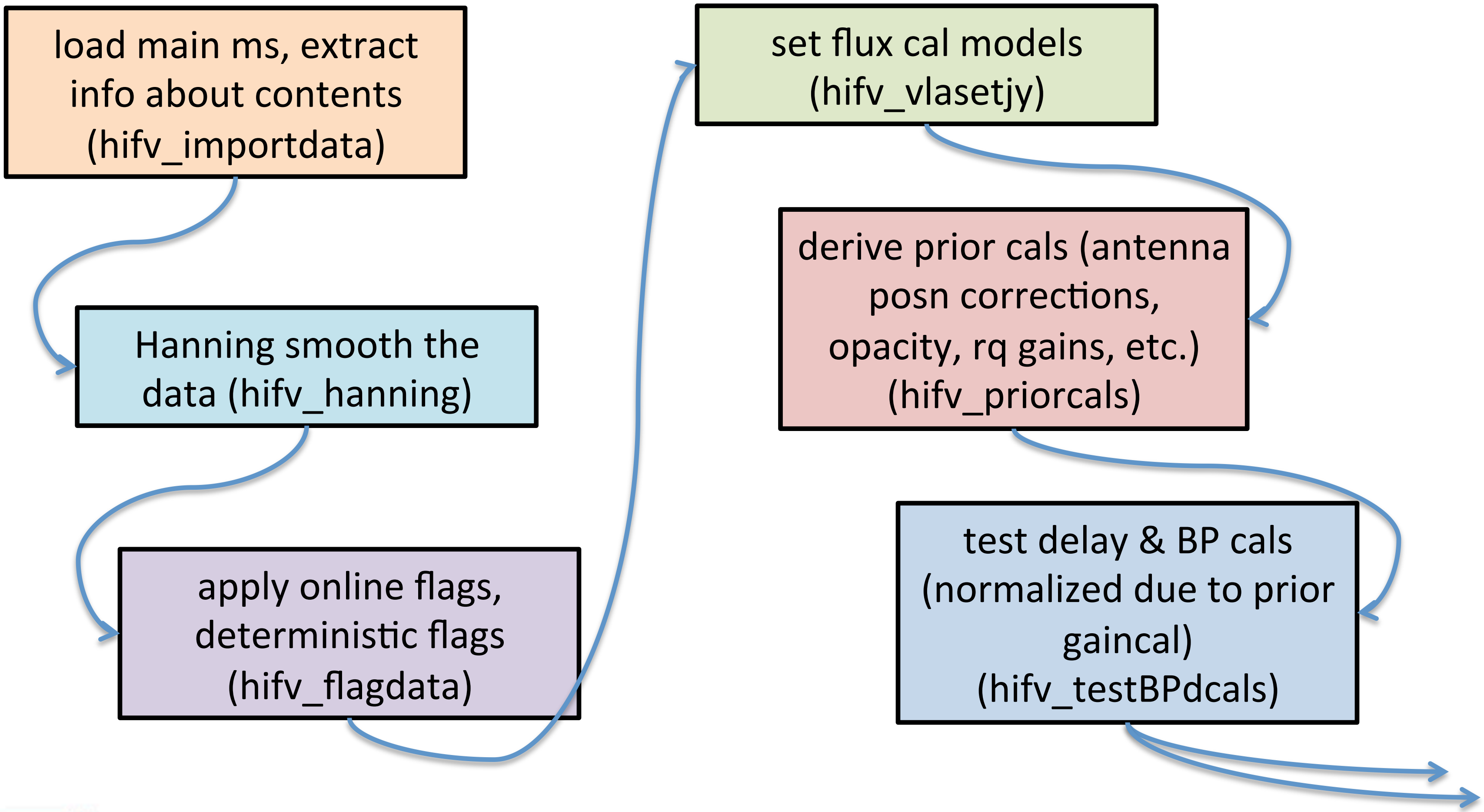
(Real-Time) Heuristics (II)

- Heuristics (cont.): the pipeline:
  - Derives final delay, bandpass, and gain calibrations
  - Applies all calibrations to the MS
  - Runs RFLAG algorithm on all fields, including target**
  - Runs statwt to derive proper relative weights per antenna/spw**

**May want to modify inputs and/or omit entirely for spectral line reductions

- Pipeline products and output
  - Flag and calibration tables
  - Calibrated MS (available for 15 days, not archived)
  - Logs, including weblog used by quality assurance (QA) staff and QA report if requested

Flow chart (CASA pipeline)



Flow chart

flag bad deformatters/spws (hifv_flagbaddef)

run rflag on calibrated delay & BP cals (hifv_checkflag)

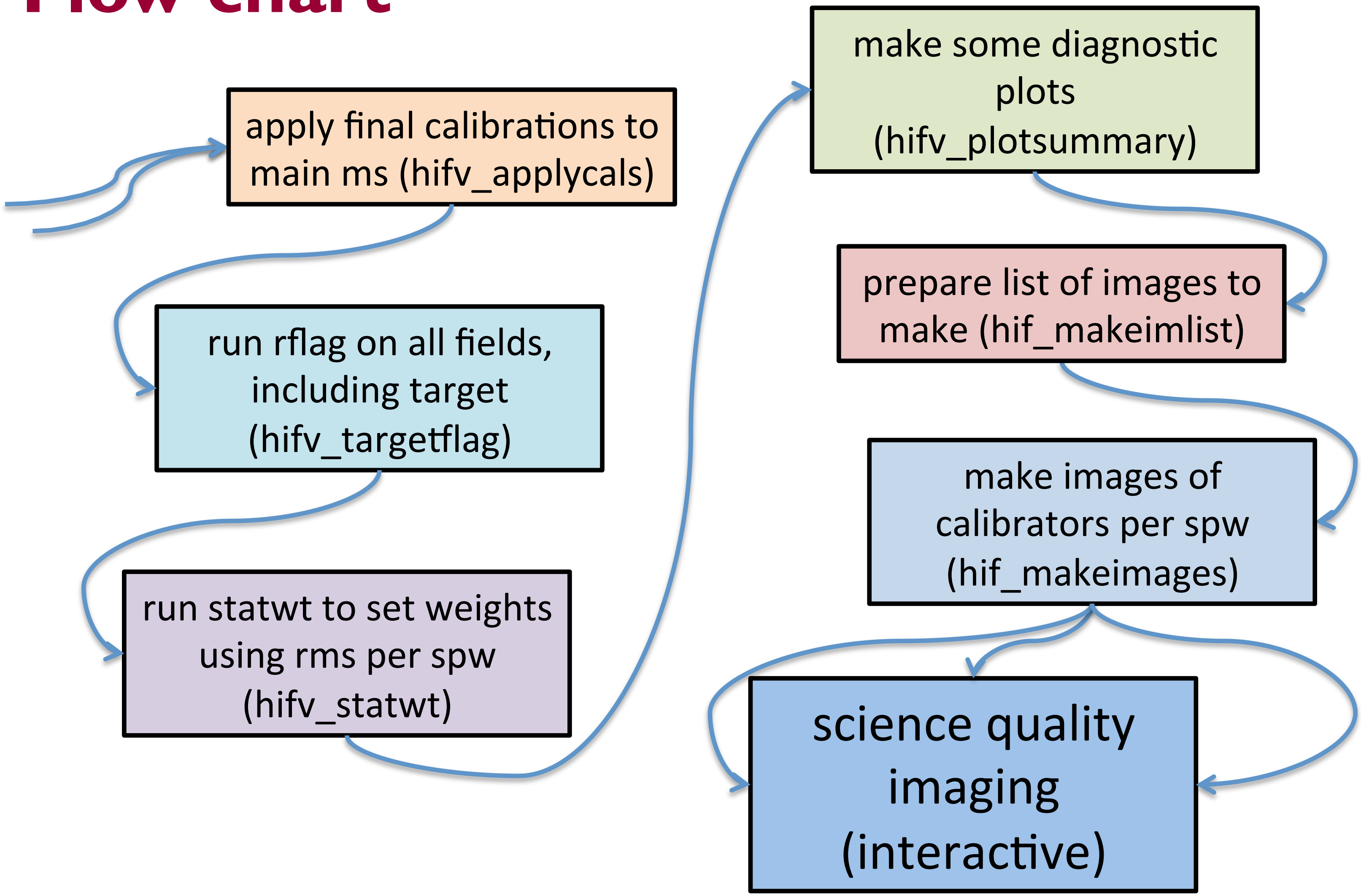
"semiFinal" delay & BP cals (not normalized, sp. index of BPcal not yet determined) (hifv_semiFinalBPdcals, hifv_checkflag, hifv_semiFinalBPdcals)

split out calibrators with spectral averaging into calibrators.ms, do a test gaincal to determine short and long solints (hifv_solint)

make amp gain table for flux density bootstrapping, and do flux density bootstrapping; includes deriving sp. index of calibrators and re-inserting into main ms (hifv_fluxboot)

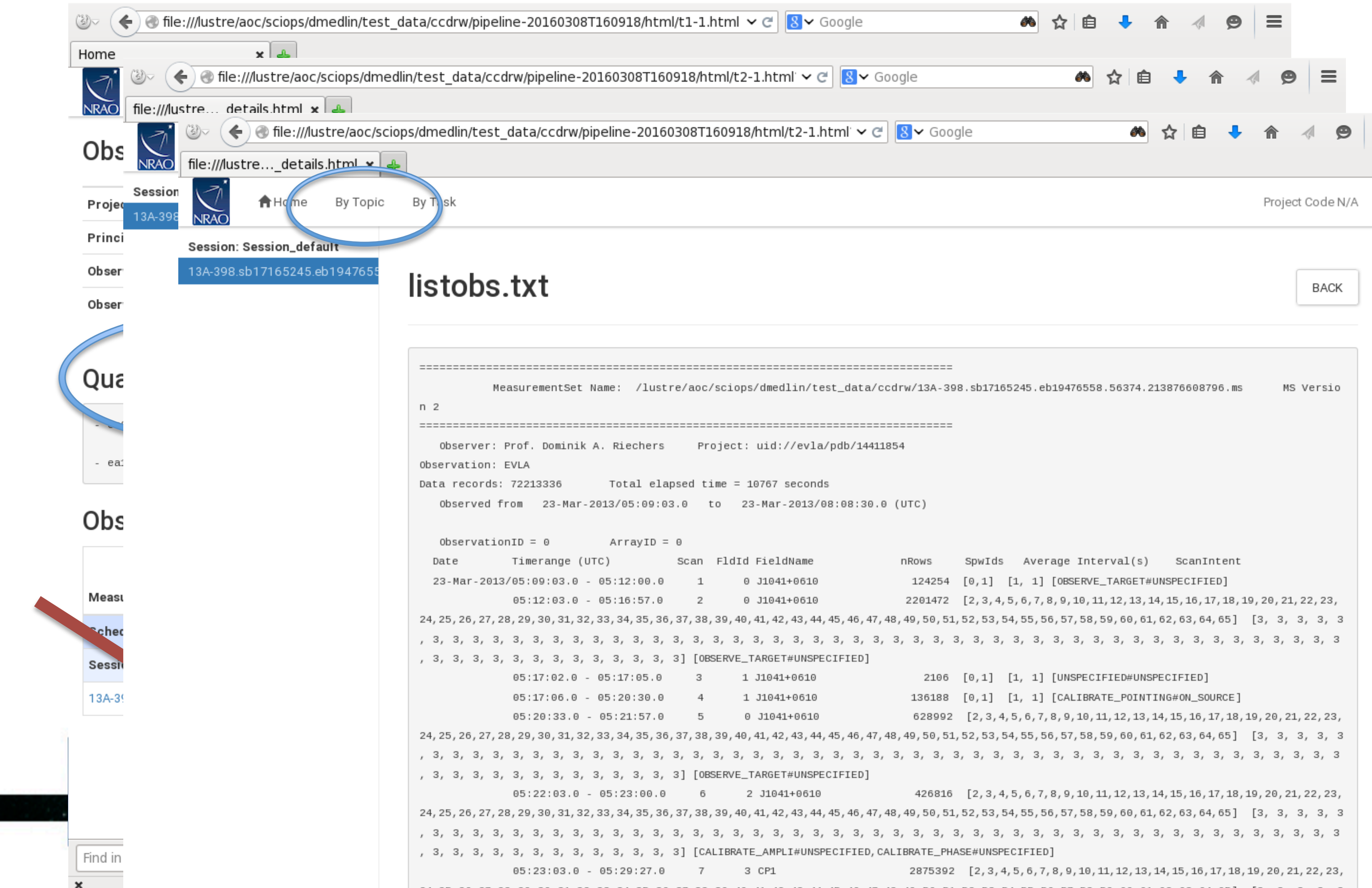
make final cal tables (delay, BP, amp, phase) – includes redoing split, fluxboot, etc. (hifv_finalcals)

Flow chart

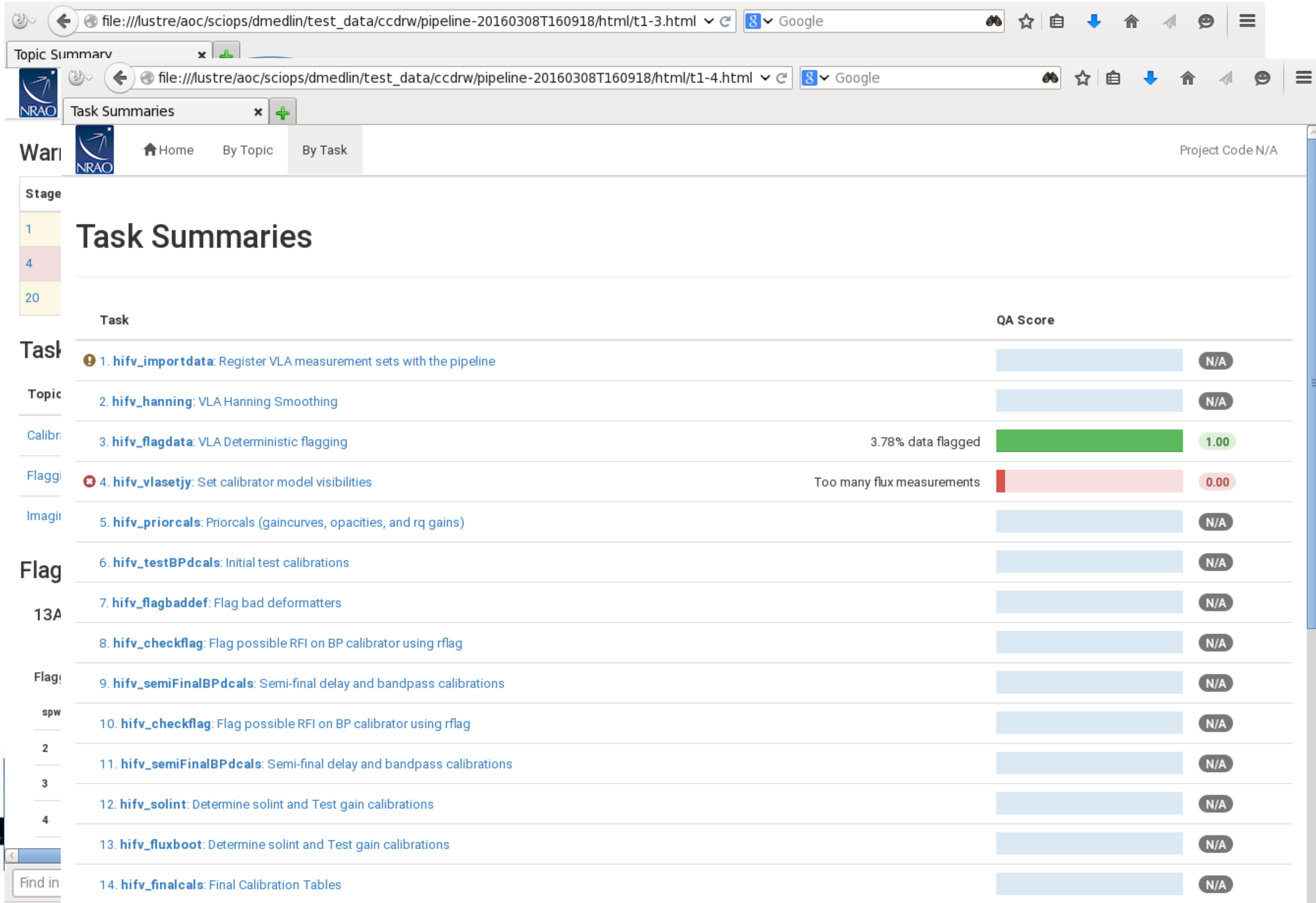


apply final calibrations to main ms (hifv_applycals)
run rflag on all fields, including target (hifv_targetflag)
run statwt to set weights using rms per spw (hifv_statwt)
make some diagnostic plots (hifv_plotsummary)
prepare list of images to make (hif_makeimlist)
make images of calibrators per spw (hif_makeimages)
science quality imaging (interactive)

Pipeline Weblog



Pipeline Weblog



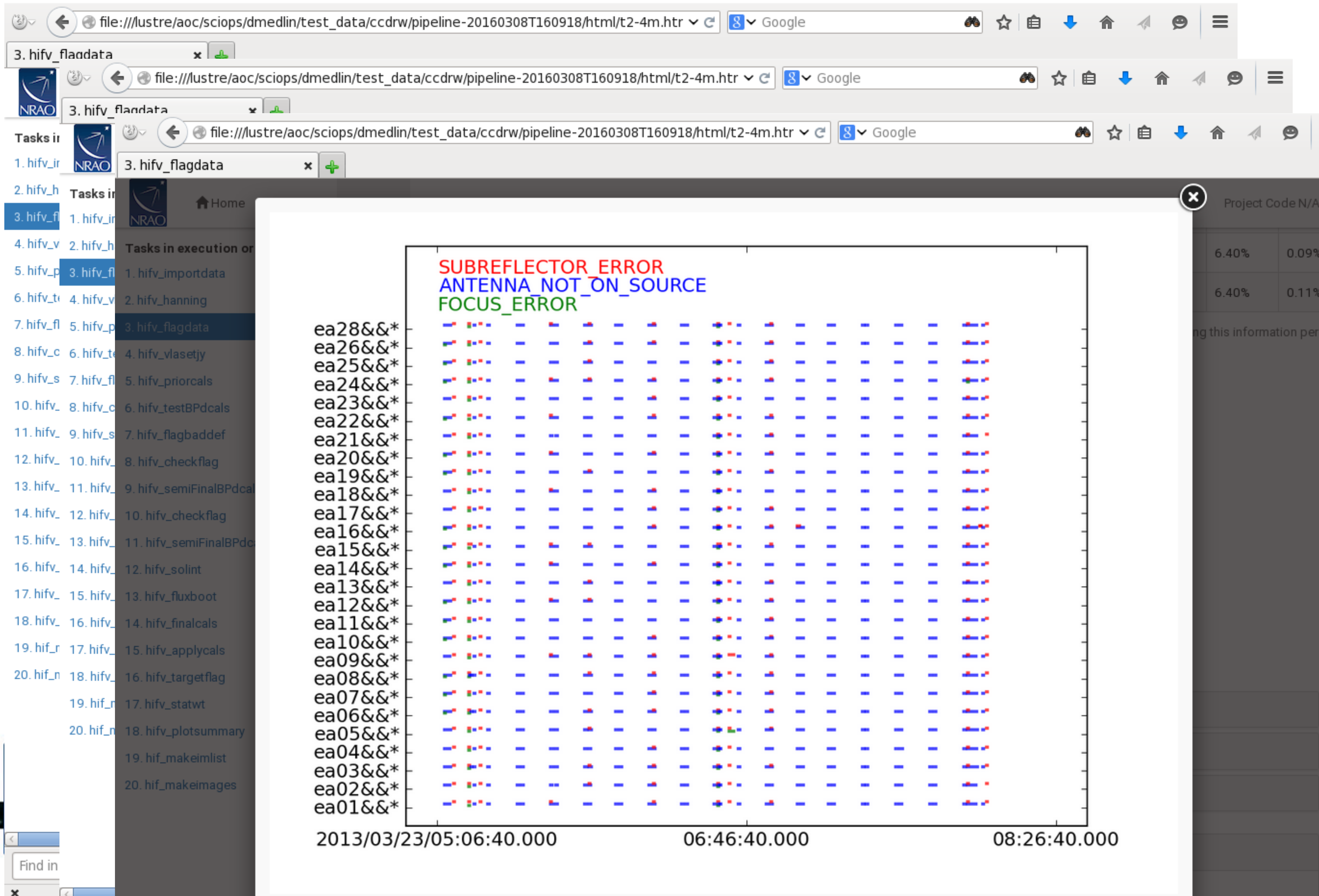
Pipeline Weblog

- The following pipeline steps provide key checks for calibration quality:

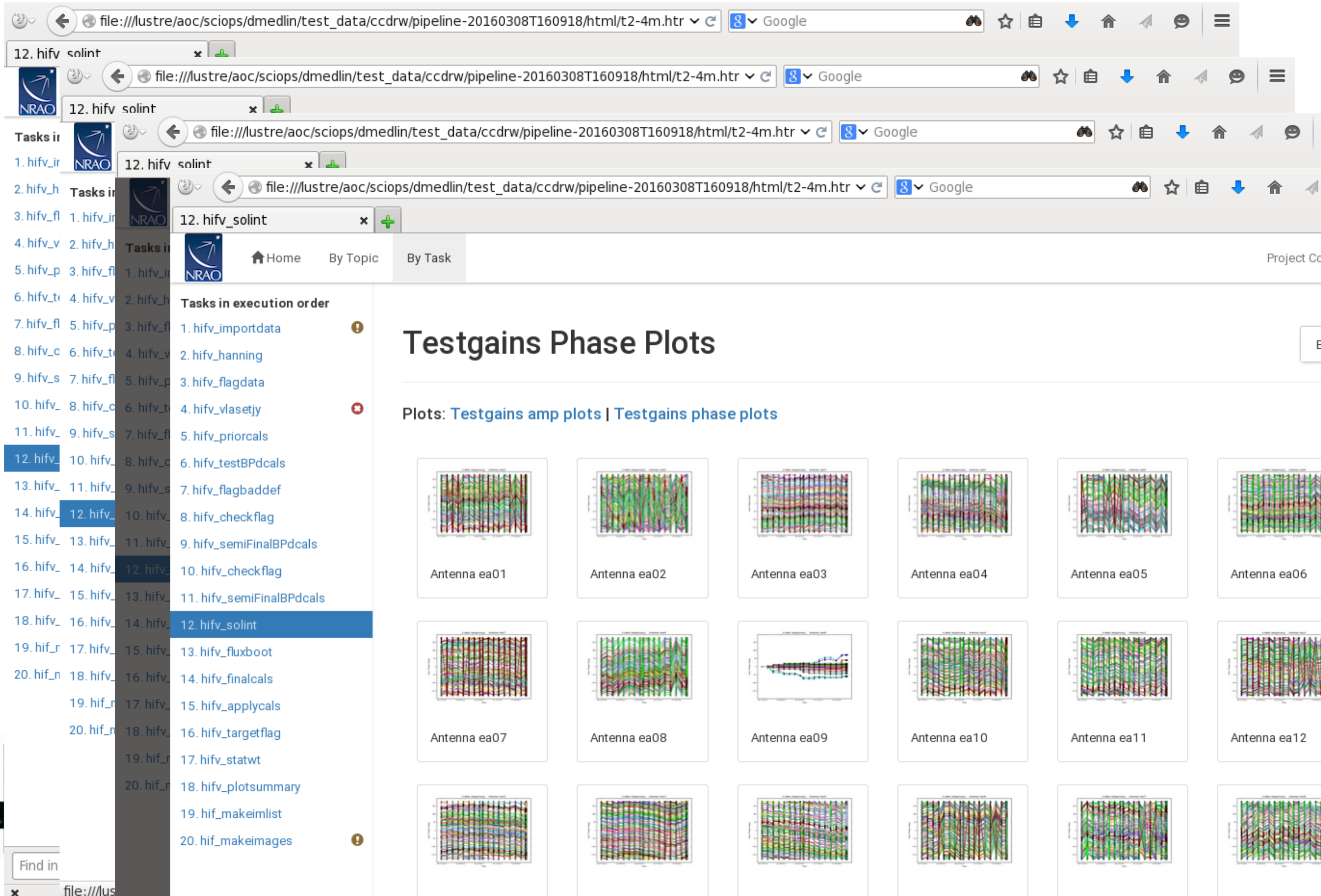
  - hifv_flagdata — *deterministic flagged data fraction*
  - hifv_solint — *solution intervals used for short/long phase cals*
  - hifv_fluxboot — *fitted calibrator flux densities and spectral indices*
  - hifv_finalcals — *final calibration tables to be applied to the data*
  - hifv_plotsummary — *some useful diagnostic plots of calibrated data*

- If something funny shows up in these steps you can look at the intermediate steps to see what might have gone wrong

Deterministic Flags (hifv_flagdata)



Gain Solution Intervals (hifv_solint)



Flux Density Bootstrapping (hifv_fluxboot)

file:///lustre/aoc/sciops/dmedlin/test_data/ccdrw/pipeline-20160308T160918/html/t2-4m.htr

13. hifv_fluxboot

Home | By Topic | By Task | Project Code N/A

Tasks in execution order

1. hifv_importdata
2. hifv_hanning
3. hifv_flagdata
4. hifv_vlasetjy
5. hifv_priorcals
6. hifv_testBPdcals
7. hifv_flagbaddef
8. hifv_checkflag
9. hifv_semiFinalBPdcals
10. hifv_checkflag
11. hifv_semiFinalBPdcals
12. hifv_solint
13. hifv_fluxboot
14. hifv_finalcals
15. hifv_applycals
16. hifv_targetflag
17. hifv_statwt
18. hifv_plotsummary
19. hif_makeimlist
20. hif_makeimages

13. Flux density bootstrapping and spectral index fitting

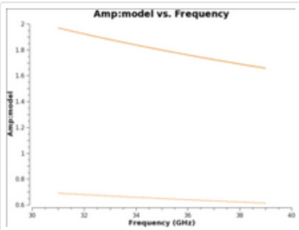
BACK

Make a gain table that includes gain and opacity corrections for final amp cal and for flux density bootstrapping.

Fit the spectral index of calibrators with a power-law and put the fit in the model column.

Fluxboot summary plot

13A-398.sb17165245.eb19476558.56374.213876608796.ms



Model calibrator

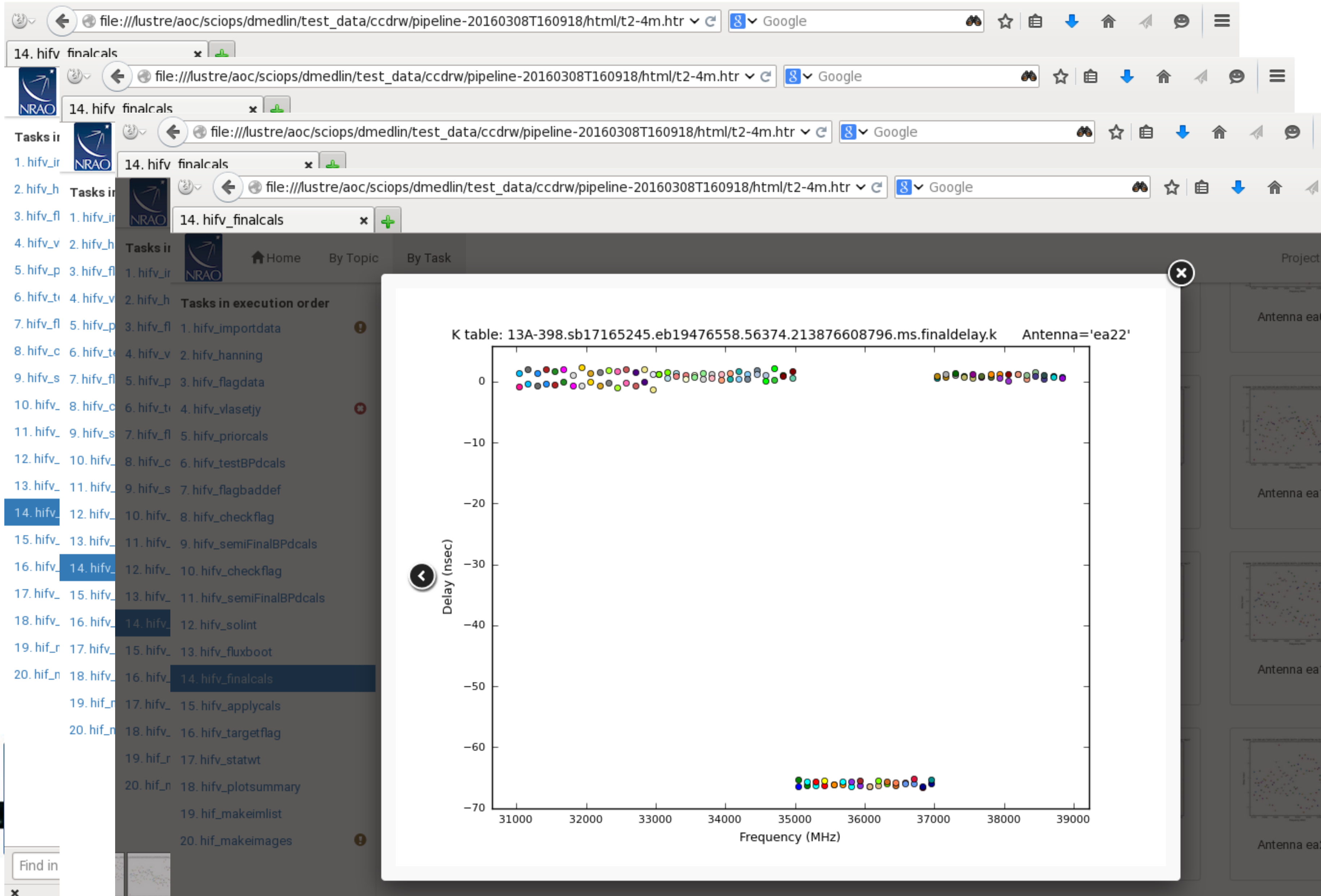
Source	Band	Fitted Spectral Index	SNR
J1041+0610	A	-0.509757951591	1168.45475415

Spectral Indices

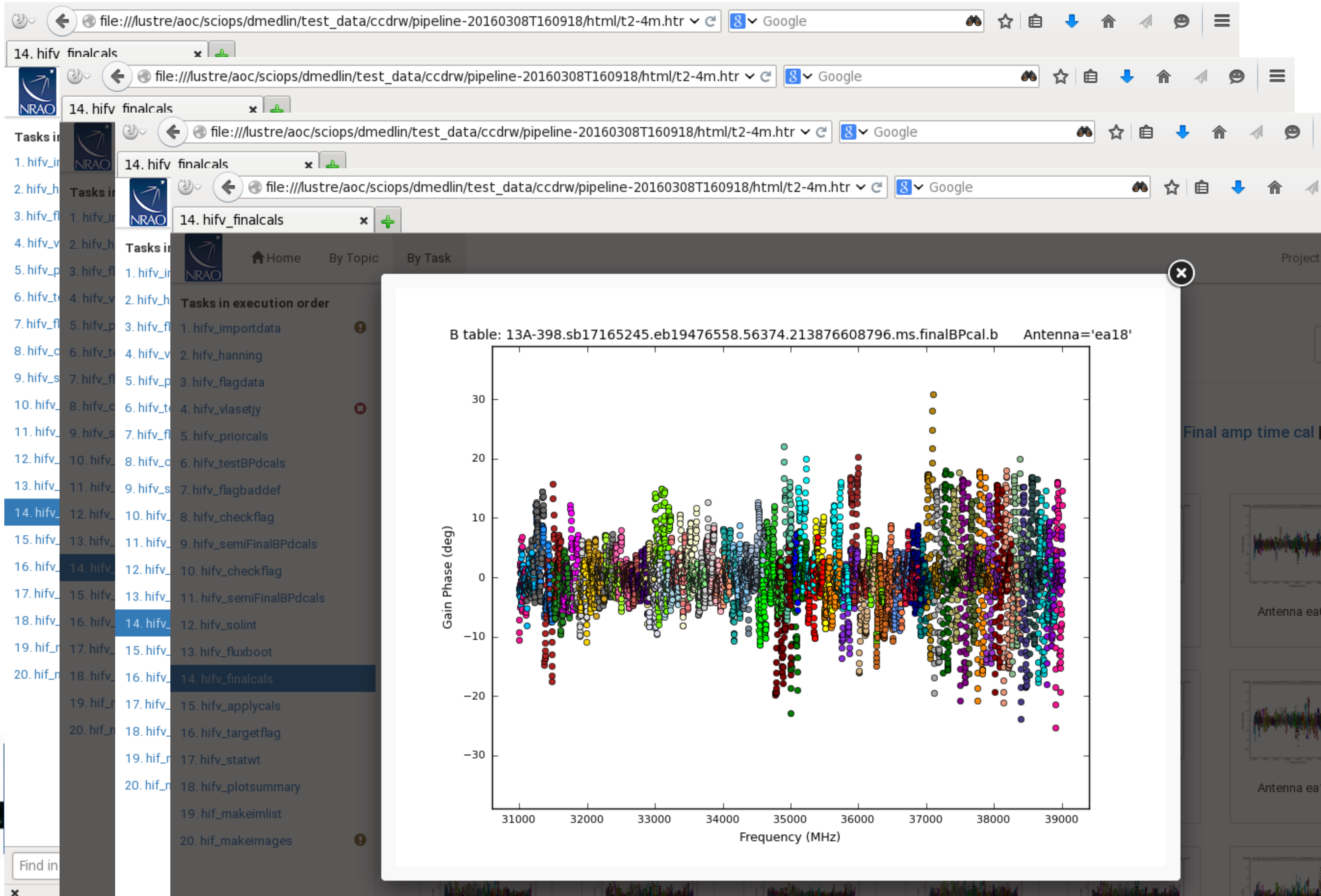
Source	Frequency [GHz]	Data	Error	Fitted Data
J1041+0610	31.04	0.681946578263	0.0314551736485	0.691613092942
	31.168	[illegible]	[illegible]	[illegible]

Find in page | Highlight All | Match Case

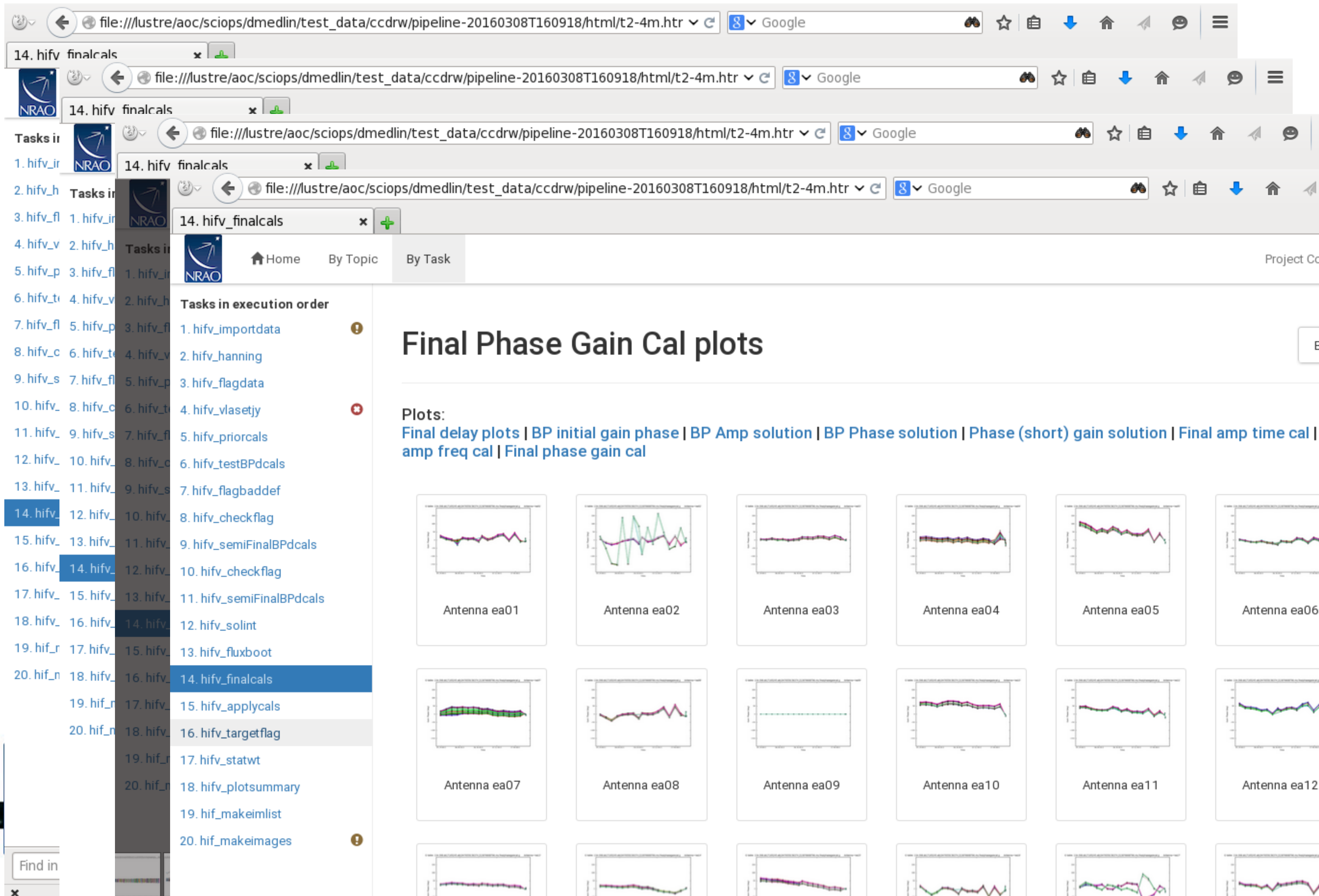
Final Calibration Tables (hifv_finalcals)



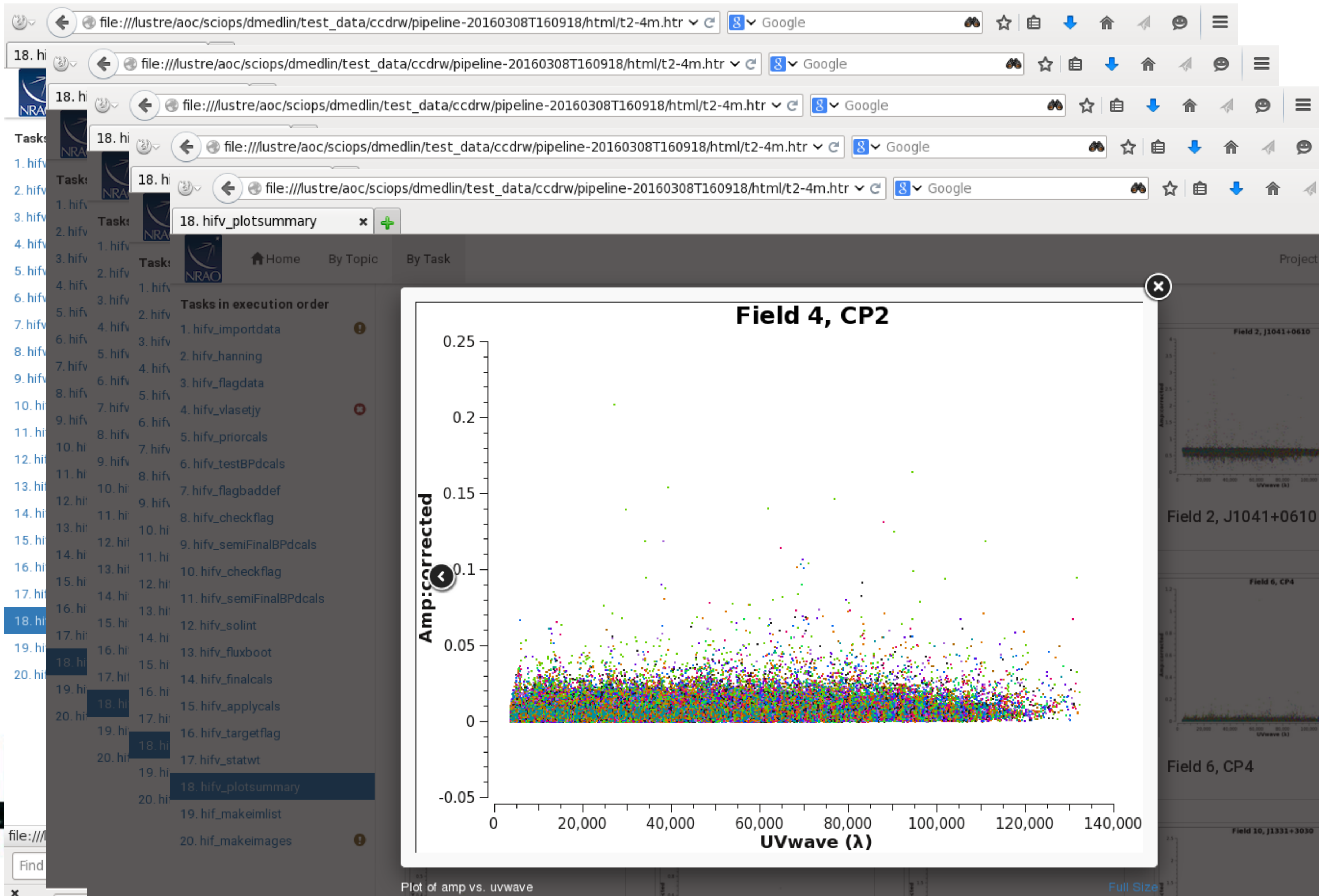
Final Cal Tables: bandpass



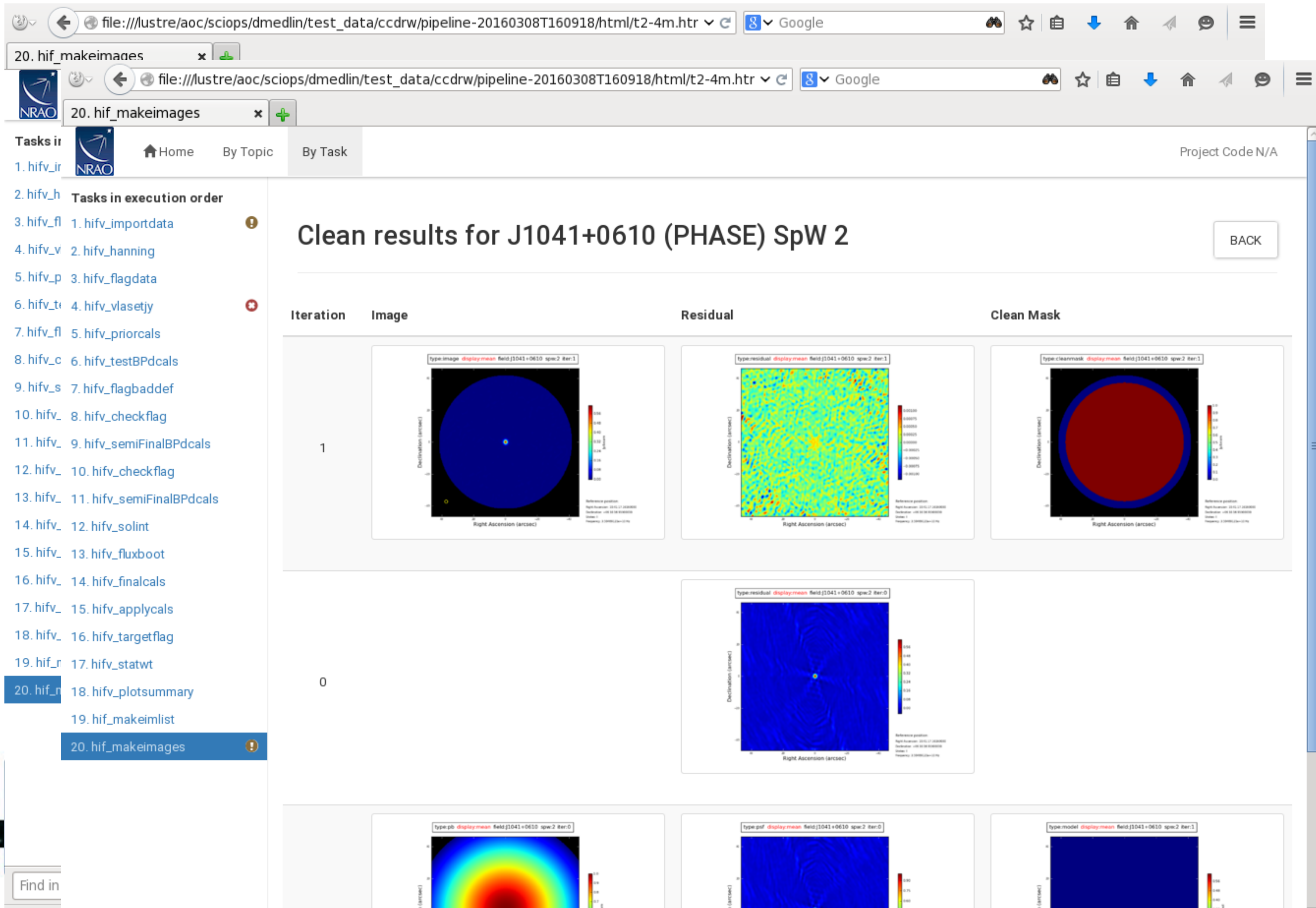
Final Cal Tables: amplitude and phase



Summary Plots (hifv_plotsummary)



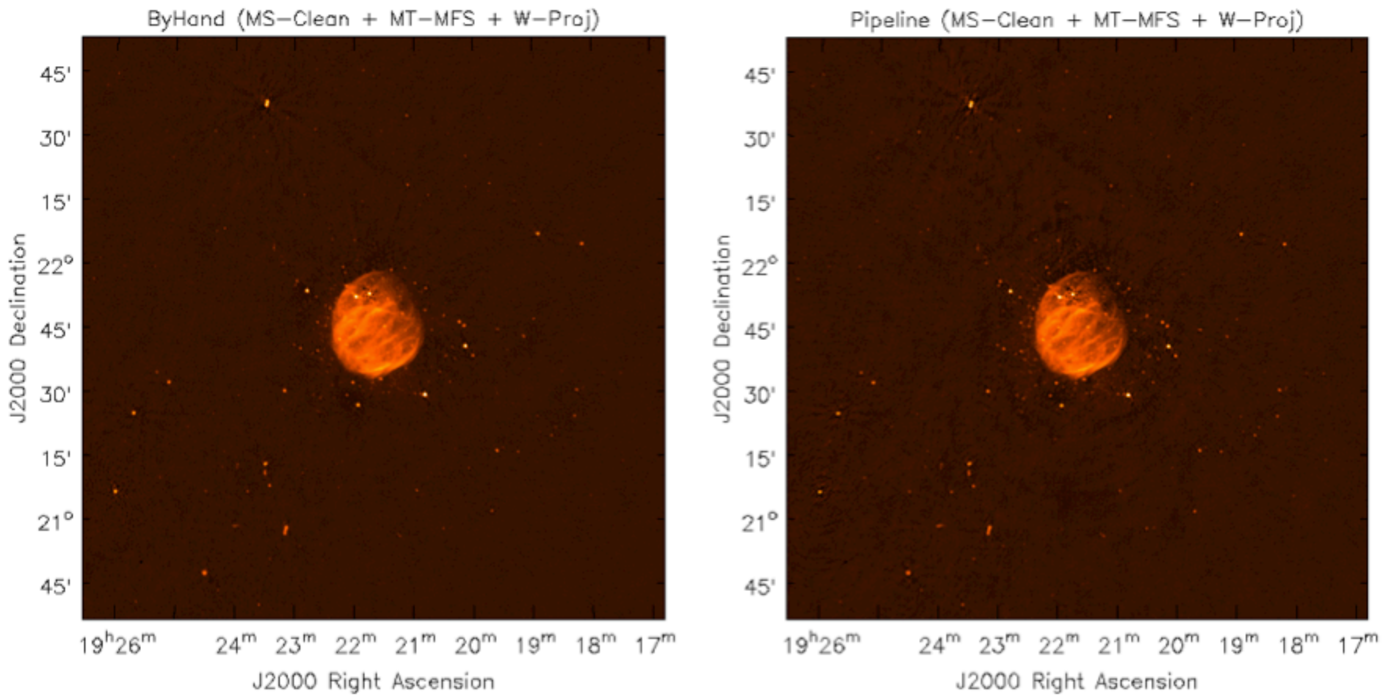
Calibrator Images (hif_makeimages)



Pipeline outputs

- The real-time pipeline produces a calibrated and flagged MS for download (follow the directions in the email from the data analysts)
  - You may request a QA2 report from the data analysts
  - If you are happy with the pipeline calibration, then:
    - Do further flagging if necessary
    - Split out your target and image
  - If you have the SDM or uncalibrated MS and the calibration and flag tables, instructions for applying flags and calibration tables may be found at https://science.nrao.edu/facilities/vla/data-processing/pipeline
- In some cases the pipeline and/or the MS may need to be modified
  - Download the SDM from the archive plus pipeline scripts
  - Follow the directions at above link
- In some cases the pipeline heuristics may not be appropriate for your data (e.g., some L-band set-ups do not work well with the pipeline yet)
  - Reduce data by hand

Imaging comparison



Left: L-band image of G55.7+3.4 produced from data flagged and calibrated by hand; the rms noise is 11.5 μJy/beam. Right: an image made from data flagged and calibrated by the VLA calibration pipeline; the rms noise is 12.2 μJy/beam. Differences in the source structure and/or source flux density are dominated by the uncertainty in the deconvolution process, not the calibration and flagging (images provided by Urvashi Rao).

Known failure modes and issues

- In general the pipeline does very well, but there are possible failure modes:
  - No flux density or gain calibrator intents defined, or flux density calibrator not one for which we have models
    - *work around in scripted pipeline*
  - Wrong scan intents
    - *work around in scripted pipeline*
  - Does not always identify deformatter problems (but does NOT usually have false positives – L-band may be an exception)
    - *flag remaining bad spws*
  - Calibrators are too weak for given spw bandwidth
    - *heuristics have been developed and are currently being implemented*

Spectral line data

- Several steps in the real-time pipeline may not be appropriate for spectral line data:
  - Hanning smoothing (increases effective channel width)
  - Last run of RFLAG on target (may eliminate your line as interference!)
  - Statwt calculates rms based on scatter of channels per spw, per visibility; may want to run manually with channel selection turned on to eliminate use of channels containing line emission in calculating the rms
- With the above modifications, the pipeline will work with spectral line data as long as the calibrators are strong enough

Mixed correlator set-ups

- With the new WIDAR capabilities it is common to observe both wide and narrow spws to obtain both continuum and spectral line data simultaneously, or multiple receiver bands
  - A single heuristic (e.g., gain calibration solution interval) for entire dataset may not be appropriate
- Solution:
  - Run pipeline through application of deterministic flags, including Hanning smoothing if you are going to use it
  - Split the MS by spw and/or scans
  - Run pipeline on split MSs WITHOUT Hanning smoothing (you have already applied it, if you are going to use it)
  - Warning: output flagging statistics may not be correct

Special cases

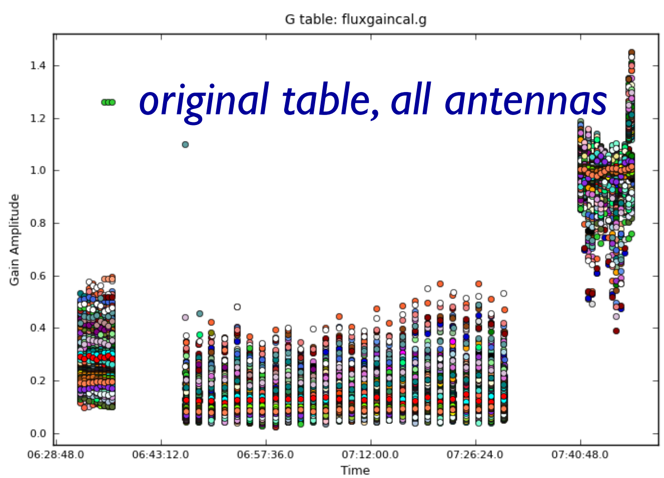
- Incorrect scan intents
  - Best to use the scripted pipeline (otherwise have to edit SDM)
  - Can run through msinfo.py, then re-set the following string variables to refer to the correct scan and field IDs:

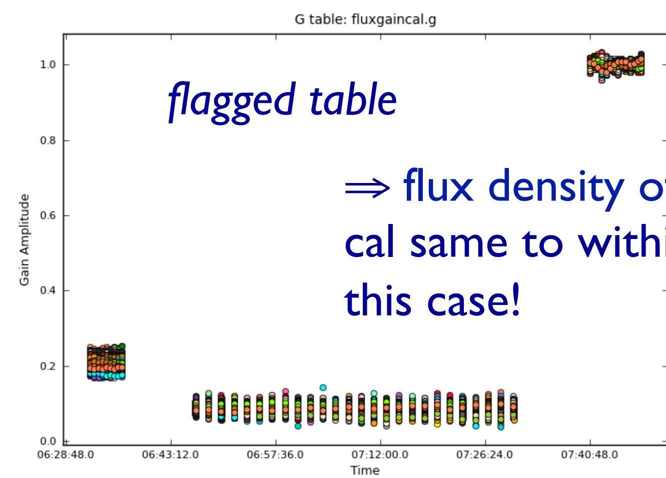
```
flux_field_select_string='2'
bandpass_scan_select_string='8'
bandpass_field_select_string='4'
delay_scan_select_string='8'
delay_field_select_string='4'
calibrator_scan_select_string='4,5,7,8,10,11,12'
calibrator_field_select_string='1,2,3,4,5,6,7'
phase_scan_list=[1,3,5,7,9,11,13,15]
```

- If a standard flux density calibrator was not observed, you may still be able to use the pipeline IF you know the flux density and spectral index of one of your other calibrators, with a bit more work – contact the NRAO helpdesk

Special cases

- Accurate flux density bootstrapping
  - hifv_fluxboot uses medians to bootstrap flux densities: fairly robust, but in some cases (e.g., high frequencies with pointing, elevation dependent gains) you can do better by flagging the gain table used for the bootstrapping
  - In scripted pipeline, run pipeline through fluxgains.py
  - Flag gain table using "plotcal"
  - Similar heuristic for the CASA pipeline currently being tested
  - With care (match elevation of flux cal, flag bad data), can reproduce flux density scale to a few % at Q-band





⇒ flux density of phase cal same to within 5% in this case!

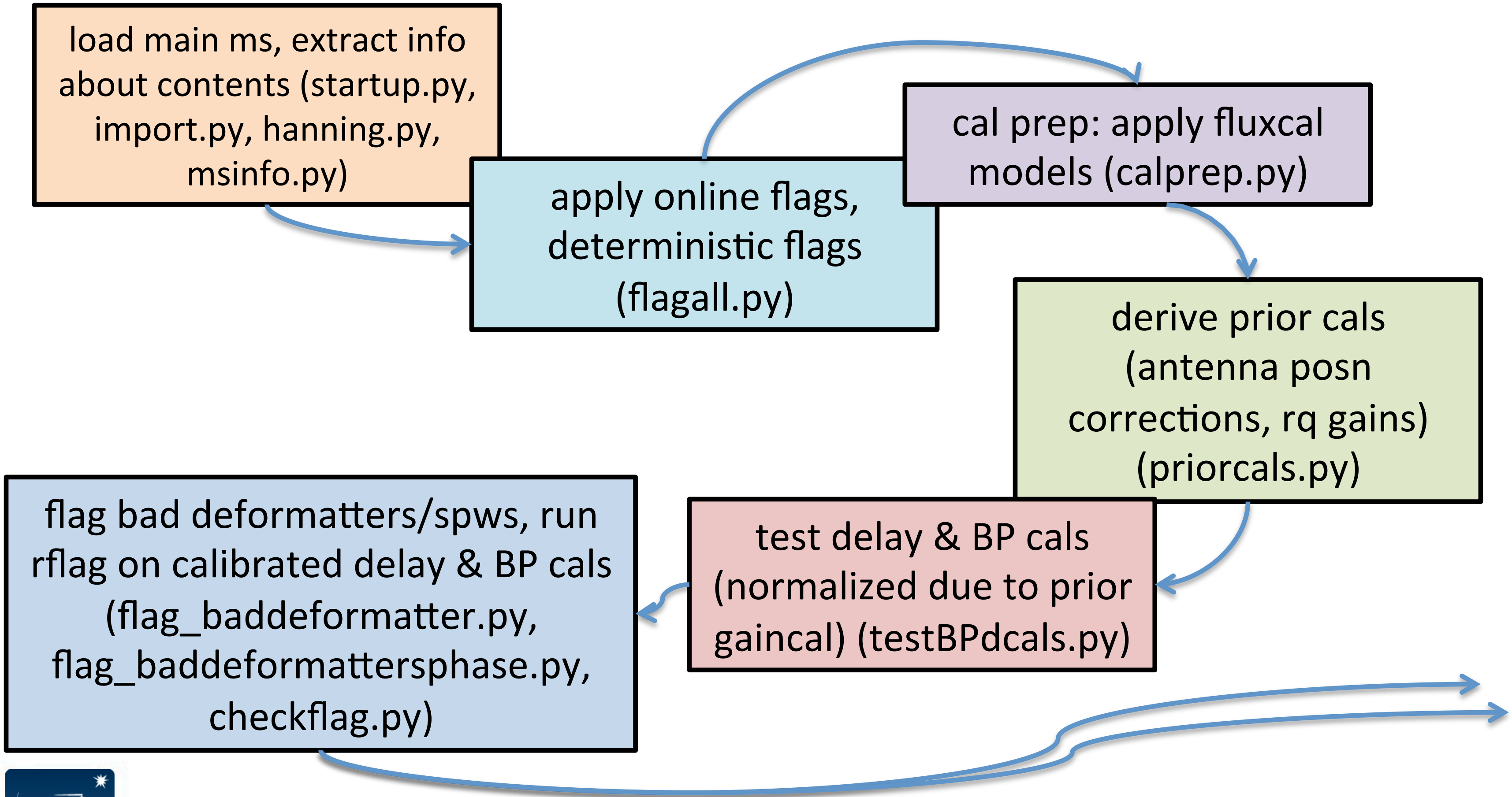
Future Developments

- Heuristics for Stokes I continuum now well-tested, stable, minor modifications allow the pipeline to be used for certain spectroscopy projects as well
- Scripted pipeline used to validate the CASA pipeline, and heuristics development
- Heuristics being implemented:
  - Weak bandpass calibrators and/or weak phase calibrators
- Heuristics being developed for:
  - Polarimetry
  - Handling gain compression due to RFI
  - Low-frequencies (<1GHz)
  - Imaging
- Heuristics developed in consultation with expert users and staff; feedback, suggestions welcome!

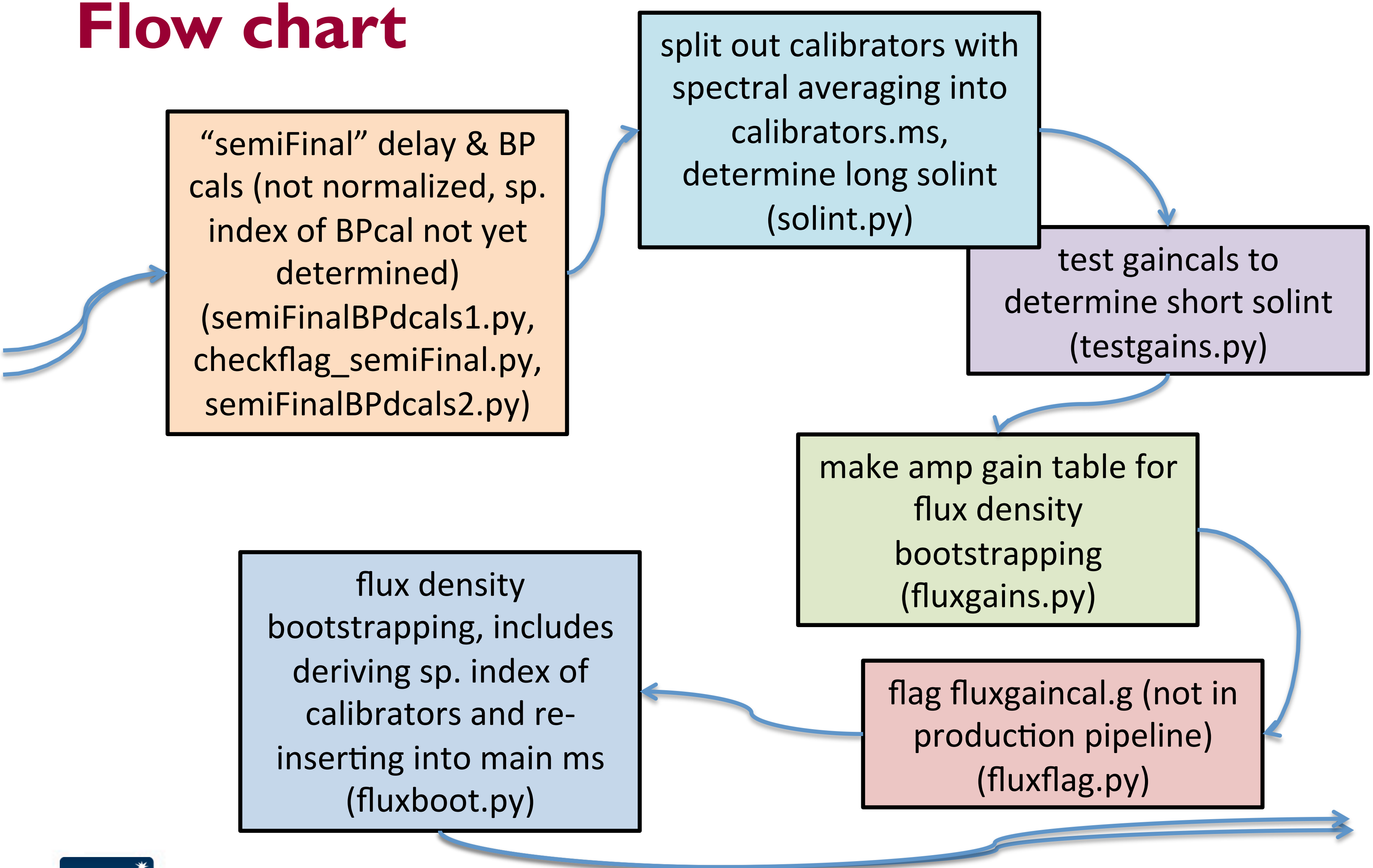
Additional slides on scripted pipeline

Flow chart (scripted pipeline)

View scripts at /lustre/pipeline/script/prod; master script is EVLA_pipeline.py

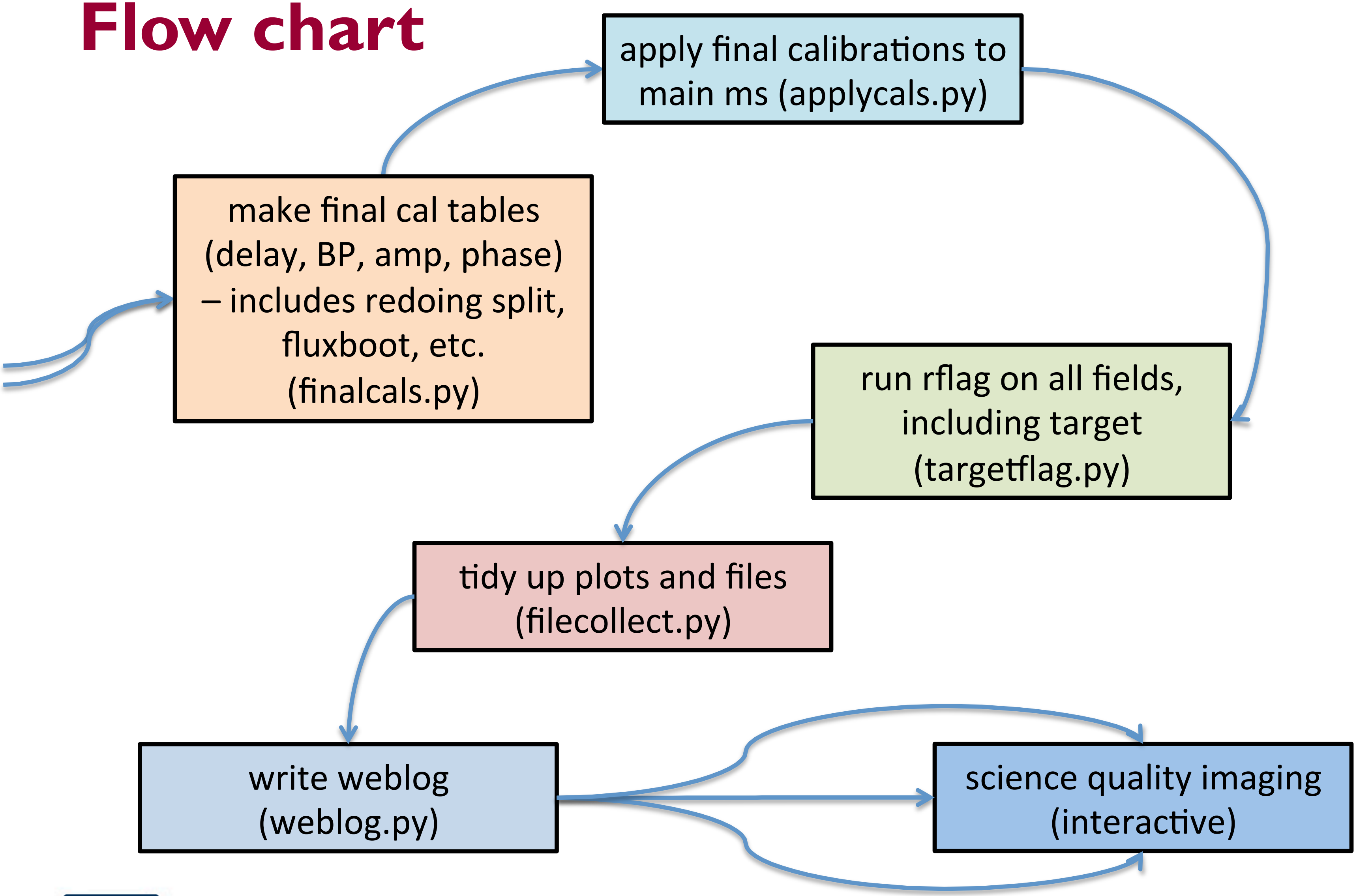


Flow chart



“semiFinal” delay & BP cals (not normalized, sp. index of BPcal not yet determined) (semiFinalBPdcals1.py, checkflag_semiFinal.py, semiFinalBPdcals2.py)
split out calibrators with spectral averaging into calibrators.ms, determine long solint (solint.py)
test gaincals to determine short solint (testgains.py)
make amp gain table for flux density bootstrapping (fluxgains.py)
flag fluxgaincal.g (not in production pipeline) (fluxflag.py)
flux density bootstrapping, includes deriving sp. index of calibrators and re-inserting into main ms (fluxboot.py)

Flow chart


apply final calibrations to main ms (applycals.py)
make final cal tables (delay, BP, amp, phase) – includes redoing split, fluxboot, etc. (finalcals.py)
run rflag on all fields, including target (targetflag.py)
tidy up plots and files (filecollect.py)
write weblog (weblog.py)
science quality imaging (interactive)