# EVLA Pipeline Heuristics Overview

Claire Chandler
4/20/12

## The EVLA pipeline concept

- Sheer size of datasets in full operations requires automated flagging and calibration procedures in order to deliver EVLA science
- EVLA pipeline concept
  - NRAO will provide flagged and calibrated visibilities (no self-cal)
  - User will self-calibrate and image visibility data to meet science goals, using NRAO computing resources or resources at home institution
- Automated pipeline will run on all "standard" science SBs; "standard" means:
  - Uses an NRAO default correlator set-up (currently this means OSRO, but we are interested in edge cases so test on everything!)
  - Contains correctly labeled and complete scan intents
  - Has calibrators bright enough that calibration on a per-spw/ integration basis is possible (limitations may be relaxed in the future)

## Philosophy

- Python script, uses CASA tasks wherever possible
  - Provides compatibility and readability for users
- Minimal human intervention
  - Currently some interactive data inspection is required unless MS is already flagged – AUTOMATING THESE PARTS ARE THE HIGHEST PRIORITY
- Use existing heuristics/code: primarily based on Miriam's pipeline, AC982 pipeline, and POLCAL pipeline
  - Focus on high frequencies first (less RFI), but nothing about it is frequency dependent
  - Uses release version of CASA, 3.3.0, serial, no polarization calibration yet

## Script overview (1)

- Load the data
  - NOT currently loading switched power data, until this has been fully commissioned
  - Entire SDM is filled into a single MS, but need to test whether separating by band improves performance
- Retrieve information from the data that will be needed later
  - Integration time
  - Number of spectral window and channels/spw
  - Spw IDs for edges of baseband filters
  - Identify scans and fields associated with different calibrator intents
    - If no flux calibrator defined, abort
    - If no bandpass calibrator defined, use flux calibrator
    - If no delay calibrator defined, use bandpass calibrator

## Script overview (2)

- Do some data integrity checks
  - Check for zeroes (not yet implemented)
  - Check for missing scans
- Make some plots and list ms contents
  - Plot raw data (not yet implemented)
  - Plot online flags
  - Plot antenna positions
  - Make elevation vs. time plot using refant
  - Run listobs

## Script overview (3)

- Flag stuff we know about (time-based)
  - Online flags
  - Shadowed data
  - Zeroes
  - Pointing scans
  - Quack
- Hanning smooth the data
  - Optional, could be important if there is narrowband RFI
  - Currently commented out in the script, we are interested to hear if people find this is important!

## Script overview (4)

- Flag some more stuff (frequency-based)
  - Frequencies with known RFI (not yet implemented; may not be needed if rflag sufficient)
  - End 3 channels of each spw
  - End 10 channels at edges of baseband filters
- Prepare for calibrations
  - Add scratch columns
  - Choose a refant: uses ALMA refant selection algorithm
  - Identify sources for which we have models, and fill model column (currently missing some models, most notably at Ka-band and S-band; uses model from next highest frequency band instead)

## Script overview (5)

- Prior calibrations
  - Antenna position corrections
    - Important for SBs observed immediately after a move
    - For real-time pipeline these will not be available; will need to reprocess SBs that do not have good initial antenna positions
  - Apply switched power calibration (not yet implemented)
- Do initial test calibrations ("test" because more flagging may be needed prior to final versions)
  - Initial phase-only gaincal on delay calibrator with short solint to line up phases
  - Per-spw delay calibration
  - [Lower priority: test multiband delay calibration (not yet implemented)]

## Script overview (6)

- Initial test calibrations (cont.)
  - Initial amp&phase gaincal on bandpass and delay calibrators with short solint
    - NB: amp gains are only used for flagging purposes
    - Try different solints in case of failure (not yet implemented)
  - Plot amp gain solutions
  - Flag based on amp gain solutions (not yet implemented)
  - Bandpass calibration
  - Plot bandpass solutions (NB: these test ones are effectively normalized because of the gain calibration)
  - Check for missing spws, antennas (not yet implemented) – can be used as indicator of system health

## Script overview (7)

- Initial test calibrations (cont.)
  - Apply all test calibrations to BP and delay calibrators
  - Plot calibrated BP and delay calibrators
  - Use calibrated BP and delay calibrators for further automated flagging (not yet implemented)

## Script overview (8)

- (Semi-)Final delay and bandpass calibration
  - Delay calibration
    - Check delays, can be used as an indicator of system health
  - Phase-only gaincal on bandpass calibrator
  - Bandpass calibration
  - Plot semi-final BP solutions
- Test gain calibrations
  - Amp&phase with short solint
  - Use statistics of amps and phases in gain table for further flagging (needs to be implemented)
  - Determine solint for 1 solution per gain calibrator observation

## Script overview (9)

- Gain tables for flux density bootstrapping
  - Phase-only, short solint
  - "Scan" averages for final amp solutions
  - Flag fluxgaincal.g (needs to be automated)
  - Do flux density bootstrapping
  - Fit for spectral index of calibrators with a power-law, use fit in model column
- Final calibration tables
  - Delay, bandpass
  - Gain:
    - phase-only, short solint
    - "Scan" averages for final amp and phase solutions
  - Apply all calibrations

## Script overview (10)

- Flag some more, if needed
  - Check calibrated data (needs to be automated)
- Image
  - Determine imaging parameters from the data for QA images, 1per field per spw
  - Make QA images
  - Plot images (not yet implemented)
  - Evaluate quality (not yet implemented)

## Outstanding questions/issues

- Some steps dominate the run time:
  - Hanning smoothing (a good reason not to do it?)
  - Adding scratch columns
  - Setjy
  - Imaging
- For the narrow field case at high frequencies it is probably OK to do spectral average per spw after BP calibration, for speed
  - Q: is this OK for QA images in general, for all frequencies?
- Pipeline operation:
  - Calibration pipeline will run on ~8 nodes of cluster
  - Rest of cluster for making science images

# Next steps

- Refine heuristics (feedback from ECSV staff)
  - What input parameters would you like the option to change if you could run this through, say, a web interface?
- Convert to 3.4, and implement:
  - Switched power calibration
  - Automated flagging
  - Calibration optimization, add polarization calibration
  - Parallelization
- Pipeline testing:
  - Testing, evaluation, and feedback using pre-flagged MSs
  - When automated flagging is implemented, compare pipeline output with datasets that have been reduced by hand