## **Analysis for H1C IDR 2.2 Josh Dillon, 3/13/19 From Commissioning Team** 4-pol Raw Data **List of Bad Antennas RTP Antenna Metrics** zen.{JD}.HH.uvh5 bad\_ants/{JD}.txt zen.{JD}.HH.uv.ant\_metrics.json Redcal: redcal\_run.py Cuts times based on solar altitude and edge channels Finds delays (firstcal) **FirstCal Calibration Solutions Extract Autocorrelations:** - Performs redundant calibration per-time and perzen.{JD}.HH.first.calfits extract\_autos.py channel (omnical) Extract autos and write to disk. - Removes antennas with high chi^2 and recalibrates if necessary. **Run FirstCal Metrics:** firstcal\_metrics\_run.py Assess FirstCal solutions. **Omnical Calibration Solutions Raw 2-pol Autocorrelations Omnical Visibility Solutions** zen.{JD}.HH.autos.uvh5 zen.{JD}.HH.omni\_vis.uvh5 zen.{JD}.HH.omni.calfits FirstCal Metrics zen. {JD}.HH.first.calfits.firstcal \_metrics.hdf5 Abscal: omni\_abscal\_run.py **Abscal Visibility Model** Use externally calibrated visibilities zen.{JD}.HH.uvRXLS.uvh5 to solve for Omnical degneracies. **Abscal Calibration Solutions** zen.{JD}.HH.abs.calfits Flagging Metadata zen.{JD}.HH.init\_xrfi\_metrics.h5 zen.{JD}.HH.init\_flags.h5 **XRFI** zen.{JD}.HH.final\_xrfi\_metrics.h5 xrfi\_run.py Intermediate data products from XRFI Find and flag RFI based on raw data, Omnical gains and chi^2, Omnical visibility solutions, and Abscal gains and chi^2. Condenses flags to a single waterfall. Legend **Final Flags** zen.{JD}.HH.final\_flags.h5 Final set of flags from XRFI **Data with Flagged Absolute External Origin Calibration Solutions** zen.{JD}.HH.flagged\_abs.calfits **Visibility Data Product Smoothcal:** smooth\_cal\_run.py hera\_cal process All other absolute calibration Smooth calibration solutions on a solutions for the same day. desired calibration and frequency scale. Also selects a reference antenna. **Calibration Data Product Smoothed Absolute Calibration Solutions** hera\_qm process zen.{JD}.HH.smooth\_abs.calfits **Metrics Data Product Reflection Fitter:** casa\_imaging reflections\_fit.py **Noise Estimation:** process **CASA Imaging: Delay Filter:** Use calibrated autocorrelations to noise\_from\_autos.py delay\_filter\_run.py sky\_image.py model per-antenna cable Use calibrated autocorrelations to Produce 4pol multi-frequency-Remove power inside the foreground reflections that can be multiplied by model per-antenna noise standard **CASA Imaging** synthesis images of each data file. wedge with a wide-band delay CLEAN. the final.calfits deviations on visibilities. **Data Product Analogous Data or** Calibrated, Flagged, and **Calibration from Calibration Solutions of Per-Antenna Noise Standard** 4-pol MFS Images **Deviation from Autocorrelations Just Cable Reflections Delay-Filtered Residual Data** Other Times zen.{JD}.HH.OCRSD.uvh5 zen.{JD}.HH.reflections.calfits zen.{JD}.HH.noise\_std.uvh5 **LST-Binning Pipeline LST-Binning with Foregrounds LST-Binning Delay-Filtered Data** Istbin\_run.py Istbin\_run.py All other data (and calibrations) from Combine together data from different days Combine together data from different days a given group of days to LST-bin at the same LSTs using MAD clipping. at the same LSTs using MAD clipping.

Standard Deviation of LST-Binned,

**Delay-Filtered Data** 

zen.grp{N}.of{M}.STD.{LST in

radians}.HH.OCRSDL.uvh5

**LST-Binned, Delay-Filtered Data** 

zen.grp{N}.of{M}.LST.{LST in

radians}.HH.OCRSDL.uvh5

**Standard Deviation of LST-Binned** 

**Data with Foregrounds** 

zen.grp{N}.of{M}.STD.{LST in

radians}.HH.OCRSL.uvh5

**LST-Binned Data with Foregrounds** 

zen.grp{N}.of{M}.LST.{LST in

radians}.HH.OCRSL.uvh5