Analysis for H1C IDR 2.2

Josh Dillon, 1/14/19

From Commissioning Team

4-pol Raw Data
zen.{JD}.HH.uvh5

List of Bad Antennas
bad_ants/{JD}.txt

RTP Antenna Metrics
zen.{JD}.HH.uv.ant_metrics.json

Extract Autocorrelations:
extract_autos.py
Extract autos and write to disk.

Raw 2-pol Autocorrelations
zen.{JD}.HH.autos.uvh5

Redcal:
redcal_run.py
— Cuts times based on solar altitude and edge channels
— Finds delays (firstcal)
— Performs redundant calibration per-time and per-channel (omnical)
— Removes antennas with high chi^2 and recalibrates if necessary.

FirstCal Calibration Solutions
zen.{JD}.HH.first.calfits

Run FirstCal Metrics:
firstcal_metrics_run.py
Assess FirstCal solutions.

FirstCal Metrics
zen.{JD}.HH.first.calfits.firstcal_metrics.json

Omnical Visibility Solutions
zen.{JD}.HH.omni_vis.uvh5

Omnical Calibration Solutions
zen.{JD}.HH.omni.calfits

Abscal:
omni_abscal_run.py
Use externally calibrated visibilities to solve for Omnical degeneracies.

Abscal Visibility Model
zen.{JD}.HH.uvRXLS.uvh5

Abscal Calibration Solutions
zen.{JD}.HH.abs.calfits

Cal XRFI
xrfi_run.py
Find and flag RFI based on Omnical gains and chi^2, Omnical visibility solutions, and Abscal gains and chi^2. Condenses flags to a single waterfall.

Initial Flags
zen.{JD}.HH.cal_flags.h5
Single waterfall of flags from calibration data products.

Delay XRFI
delay_xrfi_run.py
Find additional flags using delay-filtered visibilities. All baselines and polarizations flagged identically.

Final Flags
zen.{JD}.HH.final_flags.h5
Final set of flags from both Cal XRFI and Delay XRFI.

Flagged Absolute Calibration Solutions
zen.{JD}.HH.flagged_abs.calfits

Smoothcal:
smooth_cal_run.py
Smooth calibration solutions on a desired calibration and frequency scale. Also selections a reference antenna.

All other absolute calibration solutions for the same day.

Smoothed Absolute Calibration Solutions
zen.{JD}.HH.smooth_abs.calfits

Reflection Fitter:
reflections_fit.py
Use calibrated autocorrelations to model per-antenna cable reflections that can be multiplied by the final.calfits

Noise Estimation:
noise_from_autos.py
Use calibrated autocorrelations to model per-antenna noise standard deviations on visibilities.

Delay Filter:
delay_filter_run.py
Remove power inside the foreground wedge with a wide-band delay CLEAN.

Calibration Solutions of Just Cable Reflections
zen.{JD}.HH.reflections.calfits

Per-Antenna Noise Standard Deviation from Autocorrelations
zen.{JD}.HH.noise_std.uvh5

Calibrated, Flagged, and Delay-Filtered Residual Data
zen.{JD}.HH.OCRSD.uvh5

Legend

- **Data with External Origin**
- **Visibility Data Product**
- **hera_cal process**
- **Calibration Data Product**
- **hera_qm process**
- **Metrics Data Product**
- **Analogous Data or Calibration from Other Times**

LST-Binning Pipeline

LST-Binning with Foregrounds
lstbin_run.py
Combine together data from different days at the same LSTs using MAD clipping.

All other data (and calibrations) from a given group of days to LST-bin

LST-Binning Delay-Filtered Data
lstbin_run.py
Combine together data from different days at the same LSTs using MAD clipping.

LST-Binned Data with Foregrounds
zen.grp{N}.of{M}.LST.{LST in radians}.HH.OCRSL.uvh5

Standard Deviation of LST-Binned Data with Foregrounds
zen.grp{N}.of{M}.STD.{LST in radians}.HH.OCRSL.uvh5

LST-Binned, Delay-Filtered Data
zen.grp{N}.of{M}.LST.{LST in radians}.HH.OCRSDL.uvh5

Standard Deviation of LST-Binned, Delay-Filtered Data
zen.grp{N}.of{M}.STD.{LST in radians}.HH.OCRSDL.uvh5