HERA Memo #97
Received: 16 April 2021

# HERA Team Memo: H1C Internal Data Release 3.2

Joshua S. Dillon for the HERA Analysis Team

April 15, 2021

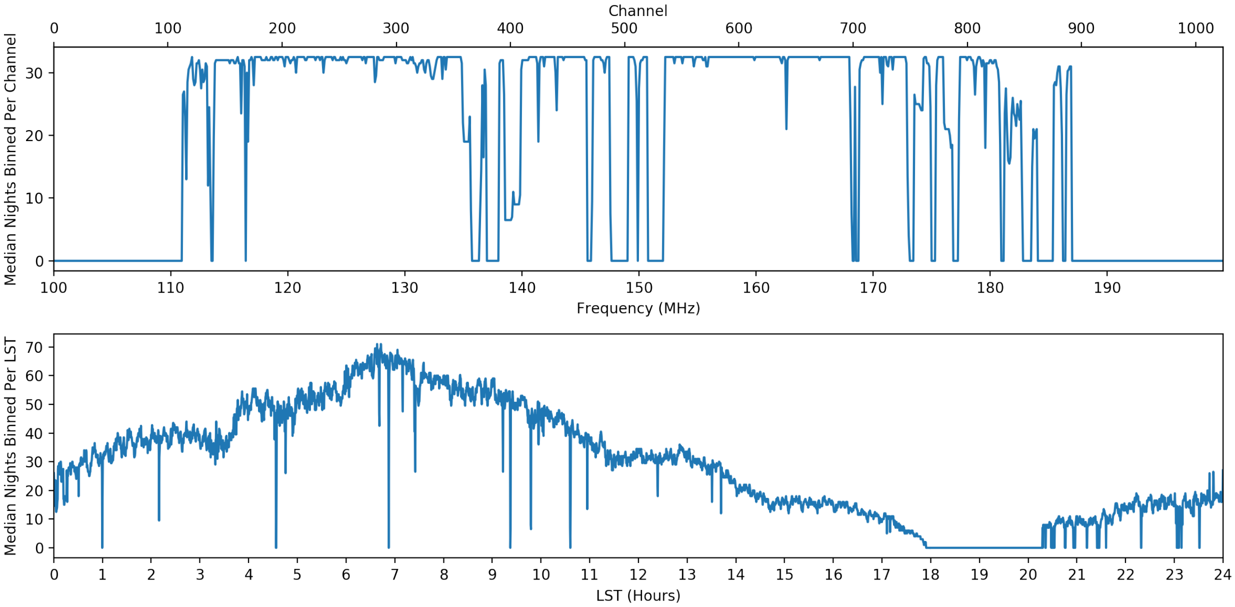

Figure 1: The frequency and LST coverage of H1C IDR3.2 for the baseline between antennas 40 and 85, neither of which was ever flagged. Here we show the median number of nights that went into the final LST-binned stack, excluding LSTs/channels with no samples. For details, see this Jupyter notebook.

## 1 Executive Summary

This memo details the process for data selection, analysis, and reduction of 94 nights of data spanning roughly 22 hours in LST (see Figure 1) to produce the most sensitive LST-binned HERA data set to date. This work follows from our first analysis and reduction of 18 days of HERA data, H1C IDR2.1, and from H1C IDR2.2, the re-analysis of that same data set that underlies the recently completed limits (HERA Collaboration, 2021).

H1C IDR3 was a release by the HERA Commissioning team of 107 nights of data between Julian dates 2458041 (October 14, 2017) and 2458208 (March 30, 2018). This includes the data in H1C IDR2. This release, H1C IDR3.2,[1] is of data products derived from that data, though further down-selection of nights, antennas, times, and frequencies was performed, as described in Section 3.

[1] What happened to H1C IDR3.1 you may ask? Well, after running through the entirety of the data through a pipeline much more similar to H1C IDR2.2, we found a number of issues, including a discontinuity in the Abscal model and numerous days with excess wideband RFI or other problems. Also, this project was on the backburner for a while as we focused on H1C IDR2.2 analysis, power spectra, and validation and on H3C/H4C analysis. Fixing those issues, plus bringing over pipeline improvements from Validation and later seasons, meant creating a new pipeline and an entirely new analysis run through the data. So H1C IDR3.1 was abandoned and we moved on to H1C IDR3.2.

Most data products can be found at NRAO at /lustre/aoc/projects/hera/H1C_IDR3/IDR3_2/. Full-size, raw data for individual nights are not staged on lustre and but are accessible via the Librarian. For a quick look at any particular night as it proceeds along the analysis pipeline, we recommend starting with the H1C_IDR3 Notebooks repository.

## 2 Observing Epochs and Data Selection

H1C IDR3 is subdivided into four "epochs" to keep the data volume manageable for any given task. Each epoch was LST-binned separately before all four were LST-binned together. The nights are listed in Table 1. Note that H1C IDR2 is a subset of Epoch 1.

| **Epoch** | **JDs Included** |
|---|---|
| Epoch 0 | 2458041, 2458042, 2458043, 2458044, 2458045, 2458046, 2458047, 2458048, 2458049, 2458050, 2458051, 2458052, 2458058, 2458059, 2458062, 2458063, 2458064, 2458067, 2458068, 2458069, 2458070, 2458071, 2458072 |
| Epoch 1 | 2458081, 2458083, 2458084, 2458085, 2458086, 2458087, 2458088, 2458090, 2458091, 2458092, 2458094, 2458095, 2458096, 2458097, 2458098, 2458099, 2458101, 2458102, 2458103, 2458104, 2458105, 2458106, 2458107, 2458108, 2458109, 2458110, 2458111, 2458112, 2458113, 2458114, 2458115, 2458116 |
| Epoch 2 | 2458134, 2458135, 2458136, 2458139, 2458141, 2458142, 2458143, 2458144, 2458145, 2458146, 2458147, 2458148, 2458149, 2458150, 2458151, 2458153, 2458154, 2458155, 2458157, 2458158 |
| Epoch 3 | 2458185, 2458187, 2458188, 2458189, 2458190, 2458195, 2458196, 2458197, 2458198, 2458199, 2458200, 2458201, 2458202, 2458203, 2458204, 2458205, 2458206, 2458207, 2458208 |

Table 1: Summary of days included in H1C IDR3.2 and how they were divided into epochs.

Epochs were chosen to be roughly equal in duration while reflecting natural breaks in good observing. They were also chosen to be small enough that the entirety of the data could be staged to lustre and LST-binned together while not impairing other HERA work.

In total, 94 days of data were kept post-analysis for LST-binning. The rationale for the 13 days thrown out is detailed in Table 2. Sometimes issues like broadband RFI are sufficiently localized in time that the night can be only partially flagged, as discussed in Section 3.3. Many of these cuts were simply judgement calls based on inspecting notebooks, all of which can be viewed in the H1C_IDR3 Notebooks repository.

| **JD** | **Reason for Exclusion** |
|---|---|
| 2458054 | Substantial X-engine dropouts. |
| 2458055 | Substantial X-engine dropouts. |
| 2458056 | Substantial X-engine dropouts followed by complete X-engine failure. |
| 2458061 | Broadband RFI for much of the night. |
| 2458065 | Broadband RFI for much of the night. |
| 2458066 | Broadband RFI for much of the night. |
| 2458089 | Extra narrowband RFI for much of the night and difficulties with smooth_cal. |
| 2458140 | Broadband RFI for much of the night. |
| 2458159 | Broadband RFI for much of the night. |
| 2458161 | Broadband RFI for much of the night. |
| 2458172 | Broadband RFI for much of the night. |
| 2458173 | Broadband RFI for much of the night with some extra narrowband RFI at the end of the night. |
| 2458192 | X-engine failure. |

Table 2: Days in H1C IDR3 excluded from H1C IDR3.2.

# 3 “A Priori” Flagging

The starting point for this IDR was the HERA Commissioning team’s H1C IDR3, which included a list of flagged antennas identified by the RTP. Ultimately, after flagging antennas, channels, and parts of days (as opposed to whole day, see Table 2), we produced one a priori flag YAML file per night, which were used in the final analysis of H1C IDR3.2. In this section, we describe how these flags were derived.

## 3.1 Antenna Flagging

First off, we flagged all antennas off the redundant grid of (what would ultimately become) the Southwest sector of the array (see Figure 2). These antennas could not be redundantly calibrated without a proper treatment of additional tip-tilt degeneracies.

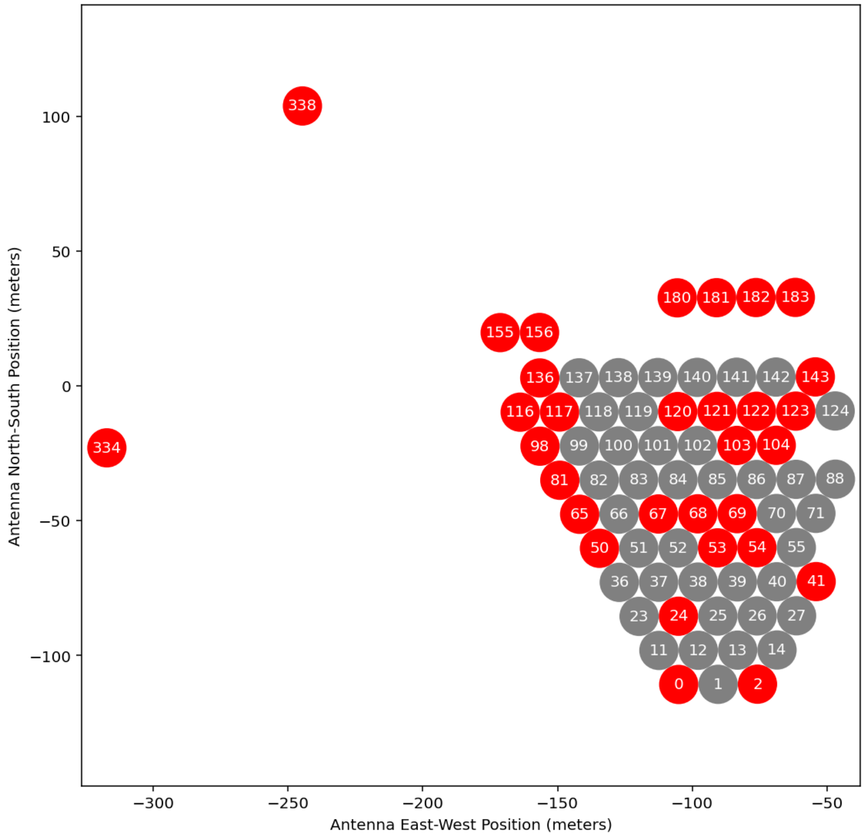

Figure 2: HERA at the end of H1C (2458208), with antennas flagged in this analysis shown in red.

Next, we took the Commissioning antenna flags and harmonized them. If, in any given epoch, an antenna was flagged more than 10% of the days, we flagged it for the whole epoch under the presumption that it was unreliable. If after that an antenna was completely flagged for more than 60% epochs it appeared in (so 3 of 4 if it was in the data starting with Epoch 0), then it was completely removed as this notebook explains.

These harmonized flags were fed into `redcal` where antenna outliers of more than $4\sigma$ in their modified $z$-scores of $\chi^2$ in any given file were noted. For more on this process, of identifying and removing outliers in non-redundancy, see Dillon et al. (2020). Antennas frequently flagged in this way were flagged for the entire night, the entire epoch, or the entire season, as this notebook describes. The result of this process is shown in Figure 3.

## 3.2 Channel Flagging

We imposed three sets of a priori frequency flags:

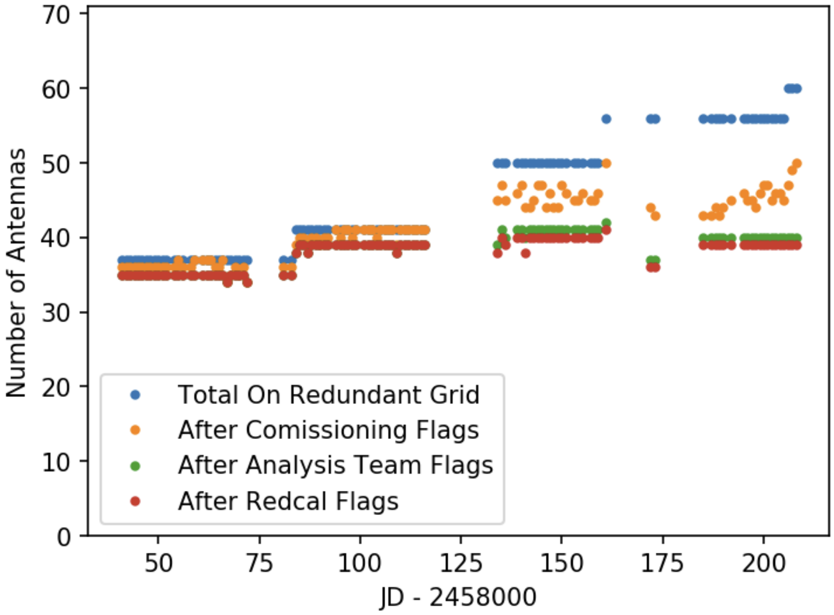

Figure 3: The number of antennas (on the Southwest redundant grid) after each successive round of flagging. The flags in orange are from H1C IDR3. The flags in green show a harmonization of those flags, where antennas frequently flagged in an epoch are completely flagged as unreliable. Finally, in red we show antennas after flagging on $\chi^2$ per antenna in redundant calibration. This plot also justifies the division of the data into four epochs; the only two nights not clearly assignable to an epoch (2458172 and 2458173) were subsequently flagged for excess RFI (see Table 2).

- Below 111 MHz was always flagged for RFI from FM radio.
- 137–138 MHz was always flagged for RFI from ORBCOMM.
- Above 187 MHz was always flagged for RFI from TV.

These flags were applied after the full `XRFI` process, so they did not affect `XRFI`'s ability to detect RFI (see Section 4).

### 3.3 Time Flagging

In addition to the days completely thrown out (see Table 2), we also hand-flagged a number of partial nights due to a variety of issues. A summary of these flags is in Table 3. Just as with the a priori channel flags, these flags were applied after the full `XRFI` pipeline, so they did not affect any automatically detected flagged times or antennas.

These flags were created and expanded on inspecting both the results of the original H1C IDR3.1 run[2] and the notebooks produced by the H1C IDR3.2 pipeline (see Section 5). Most of the flags were due to broadband RFI, which tends to occupy only a portion of the night. The impact of this RFI is quite clear in the `redcal` $\chi^2$, `abscal` $\chi^2$ and in the `XRFI` metrics. While these are relatively spectrally smooth (and thus less likely to affect the power spectrum than unflagged, narrowband RFI), they are also harder to detect because they are often so closely clumped together that `XRFI` misses them because they cease to appear as outliers in the narrow region it computes statistics over (17 × 17 pixels). A few nights showed evidence of narrowband RFI at frequencies not normally contaminated. While `XRFI` can likely flag this as well, we felt that it was safest to flag those partial nights. A few other flags were introduced due to algorithmic failures, either of `redcal` or `smooth_cal`.

[2] This work was done by Berkeley undergraduate Vighnesh Nagpal.

| **Flagged JD Range** | **Reason for Flagging** |
|---|---|
| 2458041.50 – 2458041.70 | Excess narrowband RFI. |
| 2458049.00 – 2458049.41 | Broadband RFI. |
| 2458052.50 – 2458052.99 | Excess narrowband RFI. |
| 2458058.00 – 2458058.48 | Broadband RFI. |
| 2458059.00 – 2458059.48 | Broadband RFI. |
| 2458085.56 – 2458085.99 | Broadband RFI. |
| 2458088.52 – 2458088.99 | Excess narrowband RFI. |
| 2458090.50 – 2458090.99 | Excess narrowband RFI. |
| 2458095.00 – 2458095.30<br>2458095.49 – 2458095.58 | Broadband RFI. |
| 2458096.00 – 2458096.52 | Broadband RFI. |
| 2458104.00 – 2458104.47 | Broadband RFI. |
| 2458105.00 – 2458105.43 | Broadband RFI. |
| 2458109.20 – 2458109.46 | Broadband RFI. |
| 2458110.47 – 2458110.99 | Excess narrowband RFI. |
| 2458114.00 – 2458114.32 | X-engine restart. |
| 2458135.00 – 2458135.43 | Broadband RFI. |
| 2458136.20 – 2458136.43 | Broadband RFI. |
| 2458139.00 – 2458139.34 | Broadband RFI. |
| 2458141.00 – 2458141.52 | Broadband RFI. |
| 2458144.00 – 2458144.31 | Broadband RFI. |
| 2458145.00 – 2458145.38 | Broadband RFI. |
| 2458148.00 – 2458148.37 | Broadband RFI. |
| 2458157.46 – 2458157.99 | Redundant calibration did not converge in a large region of the waterfall. |
| 2458185.00 – 2458185.52 | Broadband RFI. |
| 2458187.64 – 2458187.99 | Flag some galaxy data to prevent `smooth_cal` issues. |
| 2458189.52 – 2458189.99 | Broadband RFI. |
| 2458190.63 – 2458190.99 | Flag some galaxy data to prevent `smooth_cal` issues. |
| 2458196.64 – 2458196.99 | Flag some galaxy data to prevent `smooth_cal` issues. |
| 2458199.00 – 2458199.29 | Broadband RFI. |
| 2458200.00 – 2458200.26 | Broadband RFI. |
| 2458201.64 – 2458201.99 | Flag some galaxy data to prevent `smooth_cal` issues. |
| 2458205.00 – 2458205.28 | Broadband RFI. |
| 2458206.00 – 2458206.34 | Broadband RFI. |

Table 3: By-hand flagging of partial days after inspection of Jupyter Notebooks produced by the pipeline (see Section 5). Most of these were due to broadband RFI, which can often result in a handful of untrustworthy integrations remaining unflagged after `XRFI`, which is based on finding local outliers.

# 4 The H1C IDR3.2 Pipeline

The H1C IDR3.2 analysis pipeline is quite similar to the H1C IDR2.2 analysis pipeline. The core of the pipeline is still: redundant-baseline calibration (`redcal`), absolute calibration of the degenerate degrees of freedom (`abscal`), RFI excision (`XRFI`), and calibration smoothing (`smooth_cal`). This pipeline is depicted in Figure 4 and can be found in the `hera_pipelines` repository.

Full-size data (i.e. no redundant averaging) are LST-binned together in each of the four epochs, using the same MAD-clipping and factor-of-2 temporal averaging procedure as in H1C IDR2.2. These four epochs are then LST-binned together into a single final data product without MAD-clipping or further temporal averaging. Additionally, an LST-binned stack of the days that went into H1C IDR2 (but using H1C IDR3 processsing) was computed for testing and cross-checking purposes.

While the core analysis pipeline is very similar to H1C IDR2.2, it is worth explicitly enumerating here

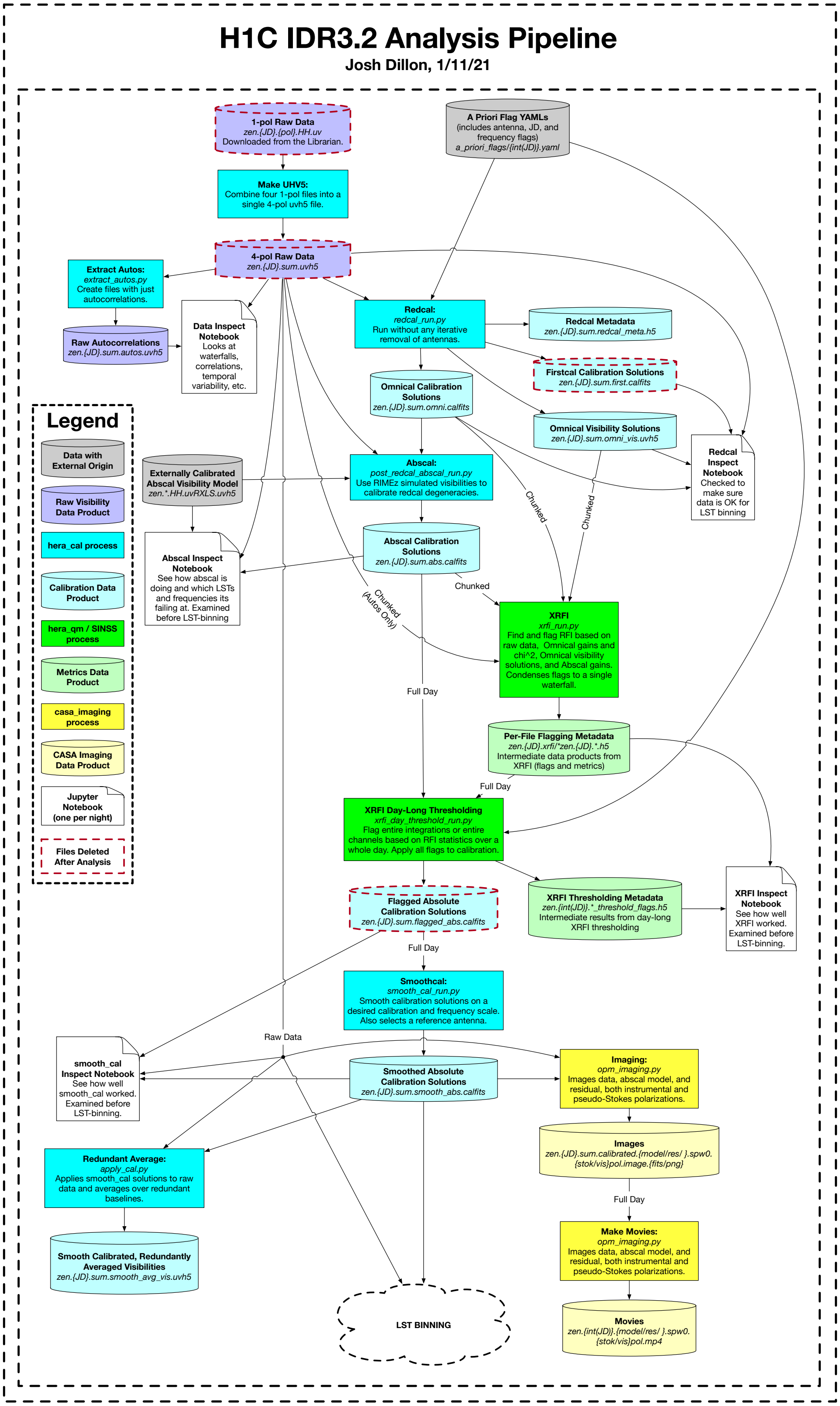
H1C IDR3.2 Analysis Pipeline
Josh Dillon, 1/11/21
1-pol Raw Data zen.{JD}.{pol}.HH.uv Downloaded from the Librarian.
A Priori Flag YAMLs (includes antenna, JD, and frequency flags) a_priori_flags/{int(JD)}.yaml
Make UHV5: Combine four 1-pol files into a single 4-pol uvh5 file.
4-pol Raw Data zen.{JD}.sum.uvh5
Extract Autos: extract_autos.py Create files with just autocorrelations.
Raw Autocorrelations zen.{JD}.sum.autos.uvh5
Data Inspect Notebook Looks at waterfalls, correlations, temporal variability, etc.
Redcal: redcal_run.py Run without any iterative removal of antennas.
Redcal Metadata zen.{JD}.sum.redcal_meta.h5
Firstcal Calibration Solutions zen.{JD}.sum.first.calfits
Omnical Calibration Solutions zen.{JD}.sum.omni.calfits
Omnical Visibility Solutions zen.{JD}.sum.omni_vis.uvh5
Redcal Inspect Notebook Checked to make sure data is OK for LST binning
Legend
Data with External Origin
Raw Visibility Data Product
hera_cal process
Calibration Data Product
hera_qm / SINSS process
Metrics Data Product
casa_imaging process
CASA Imaging Data Product
Jupyter Notebook (one per night)
Files Deleted After Analysis
Externally Calibrated Abscal Visibility Model zen.*.HH.uvRXLS.uvh5
Abscal: post_redcal_abscal_run.py Use RIMEz simulated visibilities to calibrate redcal degeneracies.
Abscal Inspect Notebook See how abscal is doing and which LSTs and frequencies its failing at. Examined before LST-binning
Abscal Calibration Solutions zen.{JD}.sum.abs.calfits
Chunked
Chunked
Chunked
Chunked (Autos Only)
XRFI xrfi_run.py Find and flag RFI based on raw data, Omnical gains and chi^2, Omnical visibility solutions, and Abscal gains. Condenses flags to a single waterfall.
Full Day
Per-File Flagging Metadata zen.{JD}.xrfi/*zen.{JD}.*.h5 Intermediate data products from XRFI (flags and metrics)
Full Day
XRFI Day-Long Thresholding xrfi_day_threshold_run.py Flag entire integrations or entire channels based on RFI statistics over a whole day. Apply all flags to calibration.
XRFI Thresholding Metadata zen.{int(JD)}.*_threshold_flags.h5 Intermediate results from day-long XRFI thresholding
XRFI Inspect Notebook See how well XRFI worked. Examined before LST-binning.
Flagged Absolute Calibration Solutions zen.{JD}.sum.flagged_abs.calfits
Full Day
Smoothcal: smooth_cal_run.py Smooth calibration solutions on a desired calibration and frequency scale. Also selects a reference antenna.
Raw Data
smooth_cal Inspect Notebook See how well smooth_cal worked. Examined before LST-binning.
Smoothed Absolute Calibration Solutions zen.{JD}.sum.smooth_abs.calfits
Imaging: opm_imaging.py Images data, abscal model, and residual, both instrumental and pseudo-Stokes polarizations.
Images zen.{JD}.sum.calibrated.{model/res/ }.spw0.{stok/vis}pol.image.{fits/png}
Full Day
Make Movies: opm_imaging.py Images data, abscal model, and residual, both instrumental and pseudo-Stokes polarizations.
Movies zen.{int(JD)}.{model/res/ }.spw0.{stok/vis}pol.mp4
Redundant Average: apply_cal.py Applies smooth_cal solutions to raw data and averages over redundant baselines.
Smooth Calibrated, Redundantly Averaged Visibilities zen.{JD}.sum.smooth_avg_vis.uvh5
LST BINNING

Figure 4: The H1C IDR3.2 pipeline up to LST-binning. See Section 4 for details.

what has changed. Later, in Section 5 we will highlight new ancillary data products, such as notebooks and movies.

### 4.1 Updates to redcal Since H1C IDR2.2

Algorithmically, the only change to redundant baseline calibration is the addition of an step in firstcal to try to identify antennas that are 180° rotated. This was developed for the H3C RTP, where it was more necessary, and here it simply helps firstcal converge faster and more reliably. The result of this algorithm, along with firstcal delays, omnical iterations, and other useful metadata are stored in the new redcal_meta.hdf5 files (see Section 5.2 for details).

Two important changes were made to the runtime parameters, which are detailed in the pipeline configuration TOML. The first was that the maximum number of iterations of omnical was increased from 500 to 10,000. In practice, this probably makes little difference because omnical usually only fails to converge when there is significant RFI, but this was found to be more useful for H3C/H4C RTP and so was backpropagated to H1C IDR3.

The second was the removal of per-file flagging of antennas for high $\chi^2$ per antenna. After a full run through redcal of the entire season and a selection of flagged antennas described above (see Section 3.1), no further antenna flagging was performed. This change was made to try to avoid temporal discontinuities that might make it harder to subtract off fringe-rate 0 systematics.

### 4.2 Updates to abscal Since H1C IDR2.2

Two important changes were made to absolute calibration as compared to H1C IDR2.2. The first is a change to how flags are propagated from the absolute-calibrated reference visibilities (colloqually known as the "abscal model") to the data. In H1C IDR2.2, antennas that were flagged in the model were flagged in the data. With more recent updates to abscal, antennas flagged in the model get zero weight when determining the degeneracies of redundant-baseline calibration. However, once those parameters are solved for they are applied to all antennas in the data. The antenna flags in the a priori flag YAML files are the final per-antenna flags.

Second, we added a new step in absolute calibration to fix the bias discovered in the course of validating the H1C IDR2.2 analysis with end-to-end simulations. We found that absolute calibration produced gains that were biased low and that that bias depends on the SNR of the visibilities (Aguirre et al., 2021). This bias is due to the fact that adding complex thermal noise (with equal real and imaginary variances) is always more likely to increase the amplitude of a complex number than it is to decrease it. More directions move away from the origin than toward it, and the fraction of directions that take one away from the origin depends on the SNR.[3] A fuller, mathematical account of the bias is in Appendix B of Byrne et al. (2021). However, by calibrating the overall multiplicative amplitude as a complex number and then only taking the absolute value at the very one, one can avoid this bias. This change was implemented in a recent hera_cal PR which demonstrated the technique. That said, of all the changes to the processing between H1C IDR2.2 and H1C IDR3.2, this is the most important to formally validate.

### 4.3 Updates to XRFI Since H1C IDR2.2

While the core algorithm remains unchanged, there are two significant changes in how XRFI flags data. The first is a change to how file boundaries are handled. hera_opm now enables "chunking", meaning that one can pass multiple files to a single job, potentially overlapping others, all with the correct sent of dependencies and prerequisites. That means that for each file we flag on, we can also load its neighboring files and use them properly detect outliers that happen to fall near file boundaries. This was necessary for H4C, when the number of integrations per file went down from 60 to 2. While the chunking procedure is somewhat inefficient and leads to some analysis getting repeated unnecessarily, this will be less of a problem in the future when file sizes are smaller.

[3] Geometrically, a small circle whose center lies on a larger circle which is itself centered on the origin always has more than 50% of its area outside the larger circle—and therefore a larger distance from the origin that then center of the smaller circle.

The second change in to the data products used for flagging. In H1C IDR2.2, we use redcal gains, overall $\chi^2$ (both locally and globally), and visibility solutions; abscal gains and overall $\chi^2$ (both locally and globally); and the raw visibilities (but only in the second round of flagging which uses meanfilt). Based on some experiments with which data products provided uniquely useful information, we removed the global cuts on $\chi^2$ from both redcal and abscal as well as the time- and memory-intensive analysis of raw visibilities. However, based on its high SNR and utility thus in H4C RTP, we've added raw autocorrelations as another source of possible flags.

# 5 Summary of Available Data Products and Notebooks

The main product of this IDR is a set of LST-binned visibilities, one per epoch and another for all epochs combined. We are making available a large number intermediate data products and Jupyter notebooks that maybe be useful for a variety of analyses using this data. The full list of data products is outlined below.

## 5.1 Overview of Available Notebooks

Perhaps the most useful new data products are the Jupyter notebooks which are run automatically as part of each night's pipeline (see Figure 4). Most of these notebooks were adapted from the Stage 1 and Stage 2 RTP pipelines developed for H4C. They have proven tremendously helpful in identifying and eliminating all sorts of malfunctions.

Most of the notebooks are produced one per day, except the LST-binning notebooks which were produced one per epoch. The full list is as follows:

- data_inspect Notebooks: These notebooks summarize basic information about the data before any calibration. This includes the sky coverage of the night, example autocorrelation spectra, autocorrelation waterfalls, correlation matrices using temporal interleaving, and mean-subtracted waterfalls. They were adapted by Dara Storer from her H3C/H4C RTP notebooks.

- redcal_inspect Notebooks: These notebooks check the performance of redundant-baseline calibration. This starts with the redundant calibration (amplitude and phase) of an example baseline and file for both polarizations. It also shows whole-night information about includes firstcal delays, omnical $\chi^2$ (both overall and per antenna), and the gain amplitudes and phases of a single antenna for both polarizations.

- abscal_inspect Notebooks: These notebooks check the performance of post-redcal absolute calibration. Again this includes an example redundant baseline group, but this time compared to the absolute calibration "model." They also include full-night results for abscal $\chi^2$ and the gain amplitudes and phases of a single antenna for both polarizations.

- rfi_inspect Notebooks: This notebook starts with a full-night flagging waterfall and then walks through the process that made it. It shows an example full-night XRFI metrics waterfall, the quantity used to detect outliers and flag them. Then it walks through the three rounds of flagging (median, mean, threshold) and shows which flags are uniquely attributable to which sources. Finally, it shows the flagging fraction at each of those steps, both as a function of time and as a function of frequency.

- smooth_cal_inspect Notebooks: These notebooks check the performance of smooth_cal. This starts with a comparison of a redundantly-averaged, smooth-calibrated baselines to the unsmoothed equivalent and to the abscal "model." It also includes example whole-night gains and phases, both before and after smoothing, and the average relative difference between the smoothed and unsmoothed gains.

- lst_bin_inspect Notebooks: These notebooks help test the quality of LST-binning for each epoch. They show the array layout and which antennas were completely flagged during the epoch. They also include a comparison of a particular file after LST-binning to the per-night data. They include an assessment of noise levels—both from night-to-night variation and from frequency-interleaving—compared to the theoretical expectation, which was adapted from Aguirre et al. (2021). Lastly, they show full-night waterfalls of amplitude, phase, and $N_{\text{samples}}$ for a single baseline for both polarizations.

Unlike, the above notebooks, these notebooks were run manually one per epoch. Additionally, one notebook was made for just the nights in H1C IDR2 with the H1C IDR3.2 analysis and another was made with the H1C IDR2.2 analysis. Finally, an adapted version was made to examine the binning of all four epochs into one final LST stack.

When starting with a night of data, taking a quick look at the notebooks is often the most useful place to start. All of the figures produced by the notebooks are captioned with an explanation of what they mean and how to interpret them. The notebooks should be straightforwardly re-runable at NRAO, though often the raw data will need to be re-staged from the Librarian and recombined into `.uvh5` format (see Section 5.2). Finally, the notebooks are meant to serve as a worked example for how to load data, manipulate it, and visualize it.

### 5.2 Data Products Before LST Binning

In Table 4 we list the full set of available data products prior to LST-binning. These, along with the notebooks, are the data products without red dashed outlines in Figure 4.

All the `.calfits`, `.uvh5` and `.h5` files are readable with `pyuvdata` as `UVCal`, `UVData`, or `UVFlag` objects respectively. Likewise, specialized subclasses of `UVCal` and `UVData` are available in `hera_cal.io`, namely `HERACal` and `HERAData`.

We have added the word "sum" to make the data files more easily forwards-compatible with some of the notebooks. The files are not actually the sum of even and odd sub-integrations and there are no difference files available.

Finally, it should be noted that raw data is not available on `lustre` due to lack of space. Raw data in the miriad format is available from the Librarian, which is split by instrumental polarization into four files.[4] We use this script to re-combine the four files into one `.uvh5` file, passing the $xx$-polarized file as the argument.

### 5.3 Data Products After LST Binning

All LST-binned data products are available in `/lustre/aoc/projects/hera/H1C_IDR3/IDR3_2/LSTBIN/`. As in H1C IDR2.2, two kinds of data products are available after LST-binning. The first, which look like `zen.grp1.of1.LST.1.43455.sum.uvh5` are the LST-binned visibilities. In this case, 1.43455 is the starting time in radians. The second, which looks like `zen.grp1.of1.STD.1.43455.sum.uvh5` contain the standard deviations of the real and imaginary parts (separately as real and imaginary parts of this file) of the files which went into the `LST` file.

There are six LST binned data sets. One for each epoch 0 through 3. One just the nights that went into H1C IDR2.2, which is available in `/lustre/aoc/projects/hera/H1C_IDR3/IDR3_2/LSTBIN/idr2`. Finally, the LST-binned stacks of LST-binned files—the ultimate, most sensitive data products—are available in `/lustre/aoc/projects/hera/H1C_IDR3/IDR3_2/LSTBIN/all_epochs`. In Figure 5 we show one baseline's waterfalls of phase, amplitude, and $N_{\mathrm{samples}}$ in the final stack of all epochs.

## 6 Known Issues and Possible Future Improvements

As always, this IDR has some unresolved issues which should be highlighted. While we do not believe that these are showstoppers, they should be kept in mind as we form power spectra and further average down. These include:

- **Overflagging:** Some of our flagging metrics produce spurious extra flags. For example, it now appears that flagging based on `omnical` visibility solutions appears to create symmetric bands of extra flags around strong, consistent RFI lines (e.g. the clock line at 150 MHz). As an example see the third panel of Figure 3 here. There are likely other examples of unwarranted overflagging. A systematic study of what kinds of flags are attributable to a single source or a small subset of sources whose inputs are

[4] Due a disk failure at NRAO, some files are not currently available in the Librarian. These were re-transferred from site and are on `herastore01` at `/export/hera/herastore01-12/site_transfer/` and can be accessed via `scp`. These should be added back into the Librarian in the near future.

| **Filename** | **Description** |
|---|---|
| `zen.{JD}.?????.sum.autos.uvh5` | Autocorrelations extracted from raw data. |
| `zen.{JD}.?????.sum.omni.calfits` | Gains and flags from redundant-baseline calibration. Also includes $\chi^2$, both overall and per-antenna. |
| `zen.{JD}.?????.sum.omni_vis.uvh5` | Visibility solutions for each unique baseline group from redundant-baseline calibration. |
| `zen.{JD}.?????.sum.redcal_meta.hdf5` | Metadata related to `firstcal` and `omnical` including delays, phase offsets, numbers of iterations, etc. Readable with `hera_cal.io.read_redcal_meta()`. |
| `zen.{JD}.?????.xrfi/*.h5` | Per-source and per-round (median/mean) results from `XRFI`, including both metrics ("outlierness" above the noise in terms of numbers of $\sigma$) and resulting flags. |
| `zen.{JD}.*threshold*_flags.h5` | Results from day-long thresholding of `XRFI` results. Includes both per-source intermediate results and the final set of flags as a single day-long waterfall, which are in `total_threshold_and_a_priori_flags.h5`. |
| `zen.{JD}.?????.sum.flagged_abs.calfits` | Gains and flags after `abscal` and then `XRFI` (including day-long thresholding). Flagging does not modify gains. The overall $\chi^2$ reflects the difference between the absolute calibrated data and the externally calibrated "model." |
| `zen.{JD}.?????.sum.smooth_abs.calfits` | Gains and flags after temporal and spectral smoothing. The flags match the `flagged_abs` files. The `quals` and `total_qual` arrays reflect relative differences between `abscal` and `smooth_cal` per antenna and on average. |
| `zen.{JD}.?????.sum.smooth_avg_vis.uvh5` | Visibilities calibrated and flagged with with `smooth_cal` gains and flags and then redundantly averaged. |
| `zen.{JD}.?????.sum.uvh5_image/*` | Results of per-file multi-frequency synthesis imaging using CASA. Results are saved both as static `.png` files and more versatile `.fits` files. Images are made of the data, the rephased `abscal` model (labeled `.model.`) and the residual of the two (labeled `.res.`). Images are made in both instrumental pol (EE/NN, labeled `.vispol.`) and pseudo-Stokes (IQUV, labeled `.stokpol.`). |
| `zen.{JD}.*.mp4` | Per-day movies of the `.png` files produced by the imaging step. Available for data, model, and residual and for both instrumental and pseudo-Stokes polarizations. |

Table 4: List of available data products prior to LST-binning. Here `{JD}` stands in for the integer Julidan date of the observation, `?????` for the decimal part of the date, and `*` is a wildcard. All these files can be found in `/lustre/aoc/projects/hera/H1C_IDR3/IDR3_2/{JD}/`.

correlated is likely warranted. Its possible we can do just as good a job with a subset of the sources of flagging we currently use. We may also want to reconsider flagging based on delay-filtered visibilities, as has been tried in the past—perhaps after redundant averaging.

- **X-Engine Failures:** In analyzing epoch 0, we found clear effects of frequent packet loss on four X-engines (meaning that 4 groups of contiguous channels had low amplitude in certain integrations). These occurred on 2458054 and 2458055 and then culminated with complete X-engine failure on 2458056. While complete X-engine failure is certainly a just cause for flagging that part of the night, one might decide not to completely flag JDs with this issue since `XRFI` should be able to easily catch and flag these sorts of temporal and spectral discontinuities. Regardless, out of an abundance of caution all three days were completely flagged. However, in analyzing epoch 1, we found that a similar issue occurred on every day from 2458101 to 2458116 (though without anything like the complete failure

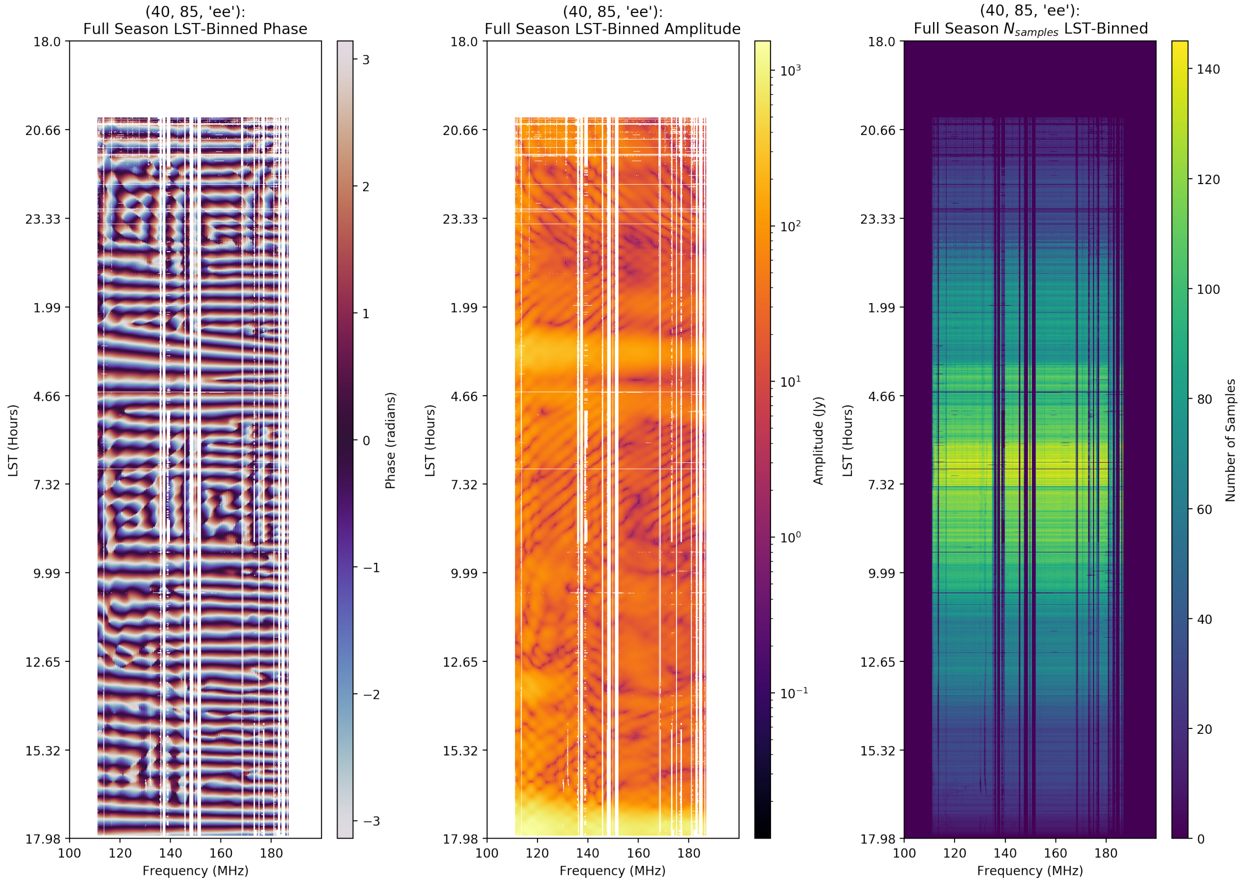

Figure 5: Example phase, amplitude, and $N_{\text{samples}}$ waterfalls for the final LST-binned data product from all four epochs. Baseline `(40, 85, 'ee')`, the same baseline shown in Figure 1. It was chosen as an example to be long enough (43.8 m Northwest, see Figure 2) to show significant fringing while neither antenna was ever flagged "a priori" (see Section 3). It is therefore a sort of best-case scenario. For details, see this Jupyter notebook.

we saw on 2458056). Since this includes the majority of H1C IDR2, we decided not to completely flag these days and instead rely on `XRFI`. This is not exactly an internally consistent decision and it is worth looking out for adverse effects from these issues, but since it did not appear to be a showstopper for H1C IDR2.2, we felt that it was better to include these days in H1C IDR3.2 to enable direct comparison to H1C IDR2.2.

- **Residual RFI in `abscal` model:** As in H1C IDR2.2, some small amount of unflagged RFI persisted in the externally calibrated data set that then got spread out by the low-pass delay filter performed on the visibilities to bridge RFI gaps and lower noise. For that reason, `abscal` gains and $\chi^2$ were used to flag RFI. Unfortunately, this creates RFI that is stationary in LST, meaning that LST-binning does not alieviate the problem. These sorts of gaps are clearly visible in Figure 5.

- **`abscal` Amplitude Failures:** One downside of the fix to the `abscal` amplitude bias is that when the model and the data are highly discrepant (in phase), the solver can drive the `abscal` gain amplitude to 0 in order to minimize $\chi^2$. This does not happen when the amplitude is solved for as a real number. This effect is rare and generally results in the integration getting flagged. Even if these failures aren't flagged, they should be largely fixed by `smooth_cal`. That said, it should be monitored and perhaps merits an investigation into how to prevent it.

- **Reflections and cross-talk:** It should be noted that no reflection fitting or cross-talk modeling and subtraction has been performed yet. That's traditionally been the responsibility of the "postprocessing" pipeline, which has fallen jointly on the Analysis and Power Spectrum teams. That work is already in progress.

Additionally, it is worthwhile here to note possible future improvements:

- **Detecting failures in `abscal`:** The `abscal` amplitude failures described above are likely detectable (and then hopefully correctable) via the difference in the amplitude before and after the final correction step, which should be predictable from the SNR. Alternatively, it may be possible to predict the bias and correct it directly via the SNR. This remains an open question.
- **Improved flagging of broadband RFI:** Ideally, our RFI algorithms should be better at catching broadband RFI, including TV allocations and especially when large swaths of time are full of broadband RFI. By-hand flagging is not a great solution for future, real-time analysis. Perhaps this is an application for SSINS (Wilensky et al., 2019).
- **A better basis for `smooth_cal`:** As mentioned in Table 3, a small amount of time was cut from some days in epoch 3 because the Galactic center was driving `smooth_cal` to large temporal fluctuations. This was fixed with a little flagging of LSTs that we won't use for an EoR limit, but a different approach would be to use something other than a Fourier basis for calibration smoothing. The Fourier basis is not good, for example, at handling linear temporal variability in gains. Perhaps the DPSS basis (Ewall-Wice et al., 2021) would be better. Alternatively, we could try zero-padding to get finer resolution in delay and fringe-rate space.
- **Systematics subtraction before LST-binning:** If systematics like reflections and cross-talk vary substantially from night-to-night, they may be more difficult to correct post-LST binning than before LST-binning. Whether this has any impact merits further study.
- **Delay filtering before LST-binning:** The differences in flagging from night-to-night, along with the differences in the calibration and smoothing, could create spectral structure. This effect needs to be studied. The effect would be substantially alleviated if the visibilities were delay filtered before LST-binning, though this would be computationally expensive.

# References

Aguirre, J. E., S. G. Murray, R. Pascua, et al. (2021). "Validation of the HERA Phase I Epoch of Reionization 21 cm Power Spectrum Software Pipeline". In: *Ap.J. in review*.

Byrne, Ruby, Miguel F. Morales, Bryna J. Hazelton, et al. (May 2021). "A unified calibration framework for 21 cm cosmology". In: MNRAS 503.2, pp. 2457–2477. DOI: 10.1093/mnras/stab647. arXiv: 2004.08463 [astro-ph.IM].

Dillon, Joshua S., Max Lee, Zaki S. Ali, et al. (Oct. 2020). "Redundant-Baseline Calibration of the Hydrogen Epoch of Reionization Array". In: MNRAS. DOI: 10.1093/mnras/staa3001. arXiv: 2003.08399 [astro-ph.IM].

Ewall-Wice, Aaron, Nicholas Kern, Joshua S. Dillon, et al. (Jan. 2021). "DAYENU: a simple filter of smooth foregrounds for intensity mapping power spectra". In: MNRAS 500.4, pp. 5195–5213. DOI: 10.1093/mnras/staa3293. arXiv: 2004.11397 [astro-ph.CO].

HERA Collaboration (2021). "HERA Phase I Limits on the Epoch of Reionization 21 cm Power Spectrum". In: *Ap.J. in review*.

Wilensky, Michael J., Miguel F. Morales, Bryna J. Hazelton, et al. (Nov. 2019). "Absolving the SSINS of Precision Interferometric Radio Data: A New Technique for Mitigating Faint Radio Frequency Interference". In: PASP 131.1005, p. 114507. DOI: 10.1088/1538-3873/ab3cad. arXiv: 1906.01093 [astro-ph.IM].