06/24/2024
Last updated: 08/19/2024

Testing the new EDGES Analysis Pipeline: Reproducing Bowman et al. 2018 results

Nivedita Mahesh, Steven Murray

Abstract

The memo summarizes the results obtained from processing EDGES low-band data, the same dataset used in Bowman et al. 2018 (referenced as B18), with the new edges-analysis pipeline.

Primary objective: Validate the new open-source Python pipeline by processing the same dataset with the exact settings used in B18 to see if we can obtain the same results.

Summary of Processing

Dataset: same as the one used in B18: Lowband ***2016_250 to 2017_095***

Processing steps: The steps and their order are shown in the figure below. These are the same as the ones used in the legacy C pipeline to produce results in 2018.

06/24/2024
Last updated: 08/19/2024

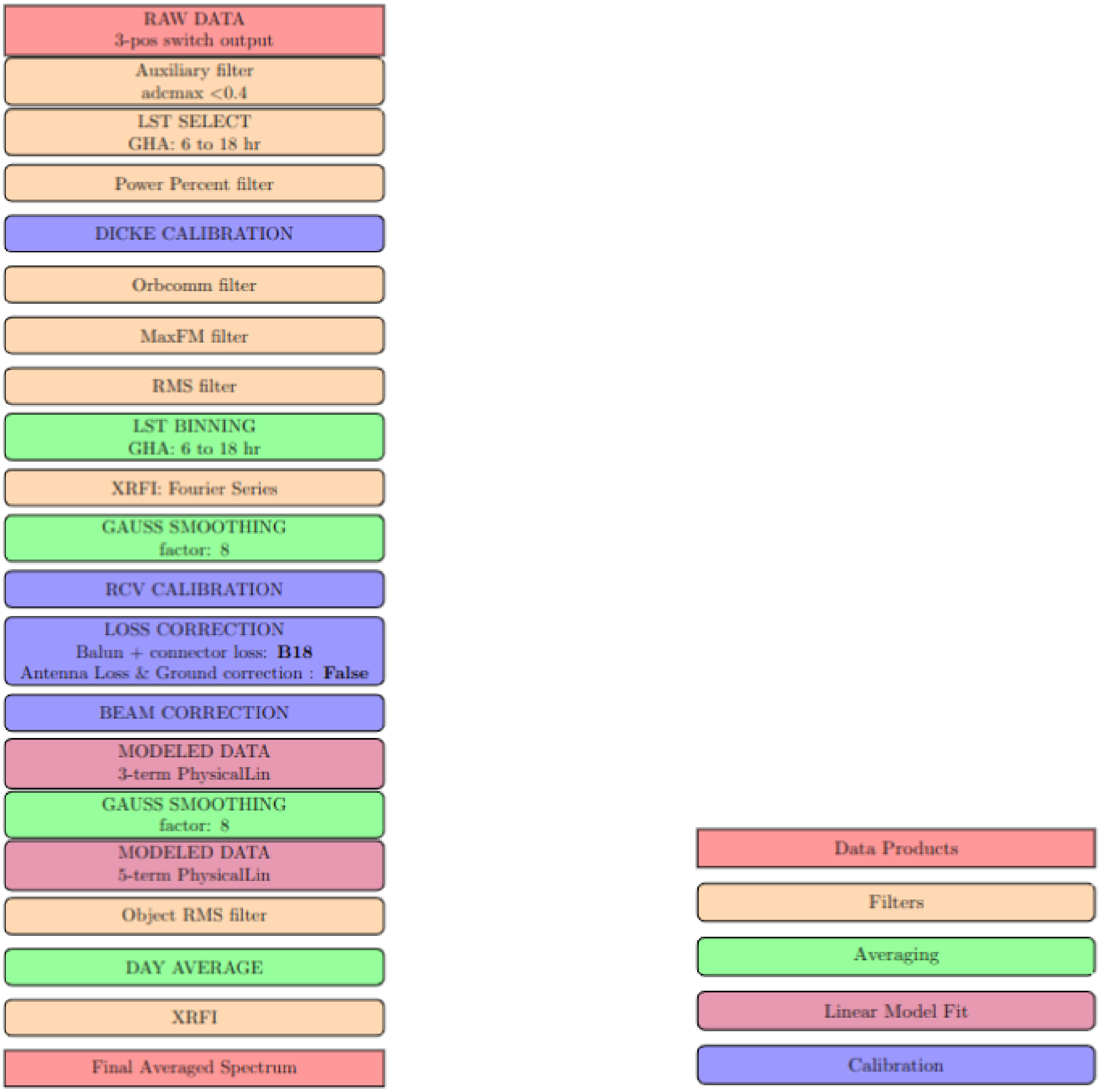


Figure 1: Flowchart of the steps in the pipeline that were applied to process the EDGES raw data from the field.

Processing the dataset through edges-analysis with the above steps, we obtain the final spectra and residuals as shown (blue) in Figure 2. We over-plot the final spectra and residuals from B18 (black dashed) for comparison. In the second plot of Fig 2. we see that the absolute difference of the final processed and averaged 12-hour spectra obtained from e-a versus B18 is less than 0.1K, which is still not within the assumed noise level of EDGES, but it is the closest agreement we have obtained between the legacy C pipeline and the new python edges-analysis pipeline.

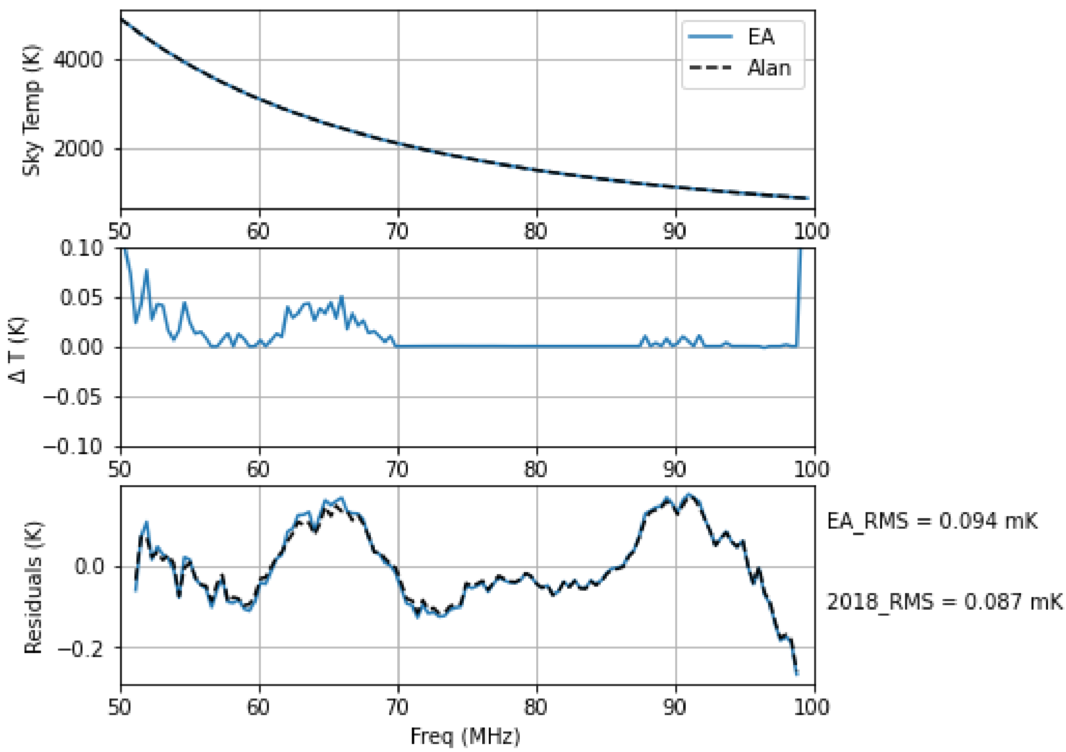
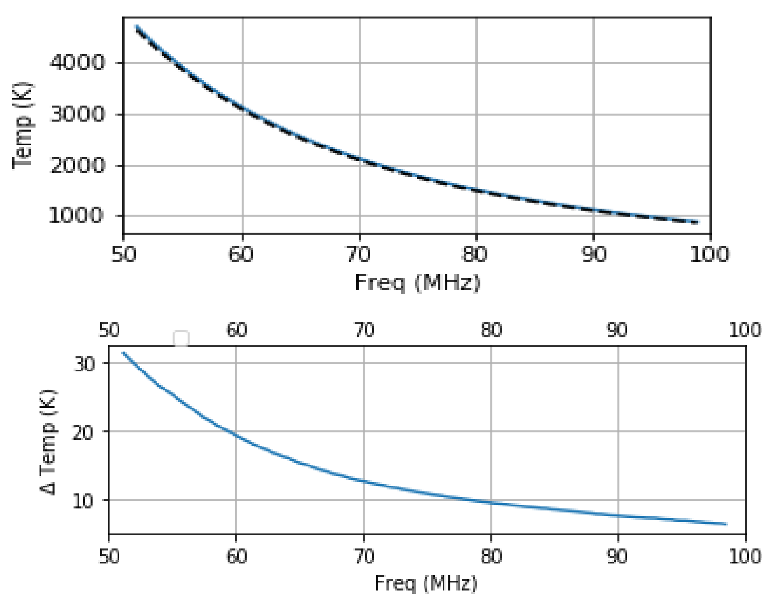


Figure 2: The final 12-hour averaged processed spectra EDGES low-band dataset obtained from the new pipeline (blue) and legacy C pipeline (black dotted). The absolute difference between the two spectra is shown in the second subplot. The final subplot is the residuals obtained after fitting a 5-term polynomial of the form log poly to the spectra in the first plot.

How did we get here?

Before we asked the question, "If we use the exact same dataset and the exact same processing steps with the exact same settings, can we get the B18 results from the new pipeline?", the processed spectra output and differences noted are shown in Figure 3.



06/24/2024
Last updated: 08/19/2024

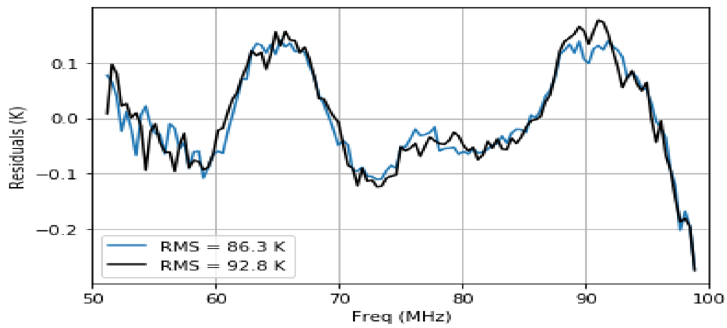


Figure 3: The final 12-hour averaged processed spectra EDGES low-band dataset obtained from the new pipeline (blue) and legacy C pipeline (black dotted). The absolute difference between the two spectra is shown in the second subplot. The final subplot is the residuals obtained after fitting a 5-term polynomial of the form log poly to the spectra in the first plot. These results from the edges-analysis pipeline correspond to those before we matched each step with the legacy C pipeline.

Summary of changes required to match B18

Starting from our "initial" results as processed through edges-analysis, several changes were required to finally obtain the "final" results that match B18 to high precision. These changes can be split into a few categories:

1. Re-ordering of existing analysis steps
2. Bug found in C code
3. Bug found in edges-analysis
4. Missing/Unused step in edges-analysis
5. Different choice of algorithm/parameter in edges-analysis for which there is no in-principle correct choice
6. Different choice of algorithm/parameter in edges-analysis that was, in principle, incorrect
7. Different choice of algorithm/parameter in edges-analysis that was in principle, **correct**

We will go through each of the steps in the processing pipeline and describe the changes that were made, also categorizing them into these categories and plotting the resulting difference.

Auxiliary filter

A. We turned off the receiver temperature filtering since the C code doesn't check this. There is evidence that some receiver_temp values are like -100000000. Indicating that the corresponding data should be filtered. An example effect is shown in Figure 4.
B. We also turned off the humidity check since the C code doesn't check for this. But in principle, it would be useful to filter data during rainy days since moisture in the soil alters the behavior of the beam

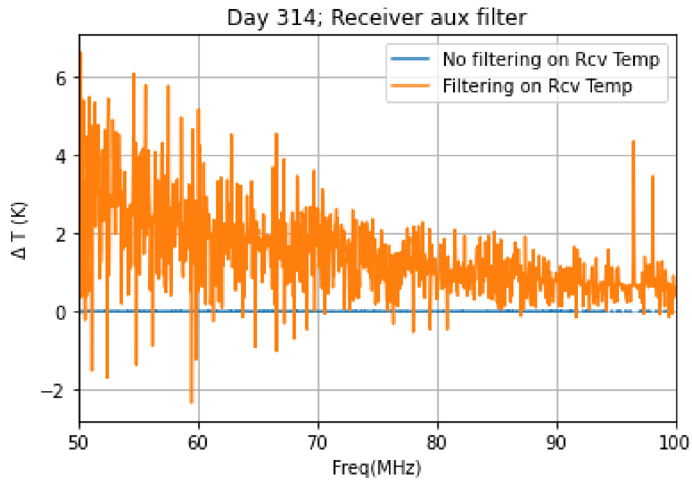


Figure 4: Difference spectra for one day of data (2016_314) between the outputs from B18 and edges-analysis after the auxiliary filter step. The difference in the orange curve is higher and non-zero because, in edges-analysis, we filtered the data based on the receiver temperature threshold of 100ºC.

LST Select

a. Reordering of steps: We changed the flow of the processing steps to first LST select and bin the data between 6 and 18hr GHA and then perform the dicke calibration. This was to match what was done in the C code. It is not clear if this has an advantage over calibrating the data in its raw time resolution and then LST binning. In Figure 1, the flowchart indicates the change made to edges-analysis.

b. Different choice of parameter in edges-analysis, which is in principle correct: We added the value of the longitude of EDGES location assumed in the C code which is `116.5. This is` slightly different from the default value of `116.605528 in edges-analysis.` This difference leads to slight difference in the time to LST conversion which in turn leads to different time steps of data passing the LST selection filter function. In principle the default value in edges-analysis is more accurate.

c. Different choice of parameter in edges-analysis, which is in principle correct: We added a function that is a transcribed version of the C code's 'tosecs()'. This matched the time to LST conversion exactly with that of the C code. Because the alternate default option in edges-analysis was filtering slightly different times compared to the C code. It is worth noting that the 'tosecs()' calculates GHA using the magical numbers in the C code. The default option in edges-analysis is to use astropy functions which we think should be more accurate

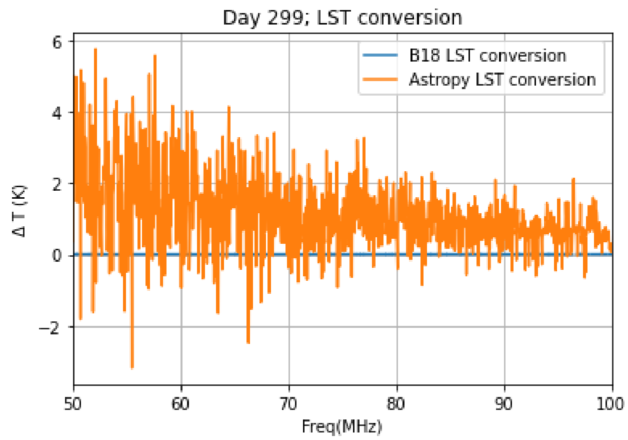


Figure 5: Difference spectra for one day of data (2016_299) between the outputs from B18 and edges-analysis after the LST select step. The difference in the orange curve is higher and non-zero because, in edges-analysis, we assumed a different longitude location, plus the time to LST conversion code was slightly different.

Power Percent filter

- Re-ordering of existing analysis steps: We changed the point at which the data is checked and filtered in the pipeline for high power above 100MHz. It is now done after the LST selection and before the Dicke calibration. See Figure 1. This used to be applied after Dicke calibration. It is unclear which works better.

Dicke Calibration

- Different choice of parameter in edges-analysis, which is in principle correct: The temperature of the noise load used in 3-position switch calibration is matched to the values used in the C pipeline. And they are, in principle, the correct values used in the system.

LST binning

- Re-ordering of existing analysis steps We reordered the processing steps to match what was done in B18 i.e., average each day's data between 6 and 19 hr GHA into one time bin before receiver calibration. In previous runs, with edges-analysis, we processed and calibrated the dataset in its raw time resolution. The change is reflected in Figure 1. It would be interesting to test the effect of this choice on the final result.

RFI filter

1. Different choice in edges-analysis which has no in principle correct: The main algorithm of RFI filtering is the same between the two pipelines. But we noted differences between the fitting/modeling algorithms -the 'lstsq' method used in edges-analysis versus 'qrd' method used in the legacy C pipeline. This results in different flag fractions, an example shown for one day in the figure below. The input settings were kept the same for the RFI step for both pipelines:

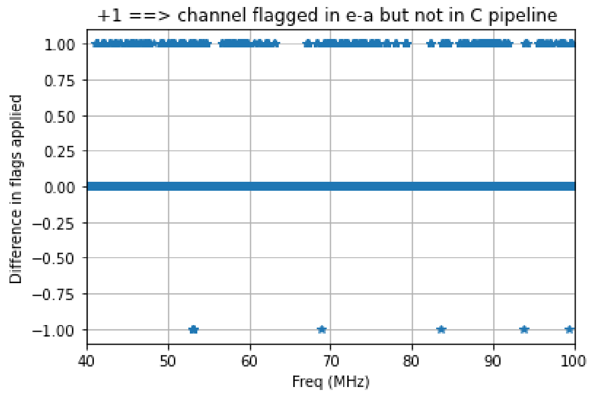


Fig. 6: Differences in the flagging of the preprocessed 2016_260 data by the two pipelines. edges-analysis flagged more channels even though the same RFI parameters were used in both pipelines.

2. We added the ‘qrd’ fitting method to edges-analysis. However, differences remain, attributed to Python modeling versus C modeling. So, we added an option to add_flags to inject Alan’s flags.

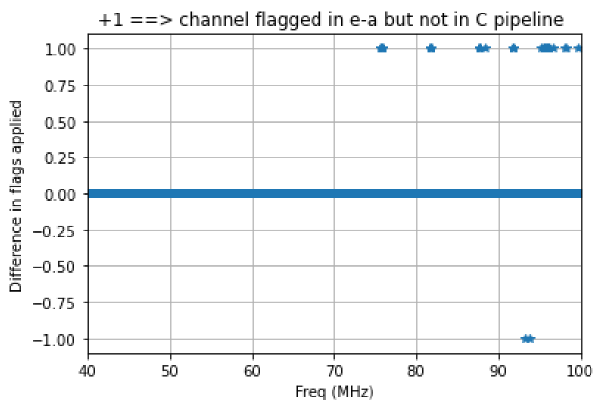


Fig. 7: Differences in the flagging of the preprocessed 2016_260 data by the two pipelines. Edges-analysis used the new ‘qrd’ algorithm, which is similar to the one in the legacy pipeline. Edges-analysis is still seen to flag more channels compared to the legacy pipeline, but the differences are lower than in Figure 6.

Gaussian Smoothing

1. Missing step in edges-analysis: In the previous runs, edges-analysis did not have a Gaussian smoothing function, where data is binned along the frequency axis by applying a Gaussian kernel. This has now been added and applied in edges-analysis.
2. Missing step in edges-analysis: Even after forcing the flags from the RFI step to be the same, the flags after Gauss Smooth were found to be different. We learned that the C code applied an additional flag thresholding while smoothing. This has been added to the code now. The effect of this thresholding is shown in the figure below. The effect of this on the final result should be tested.

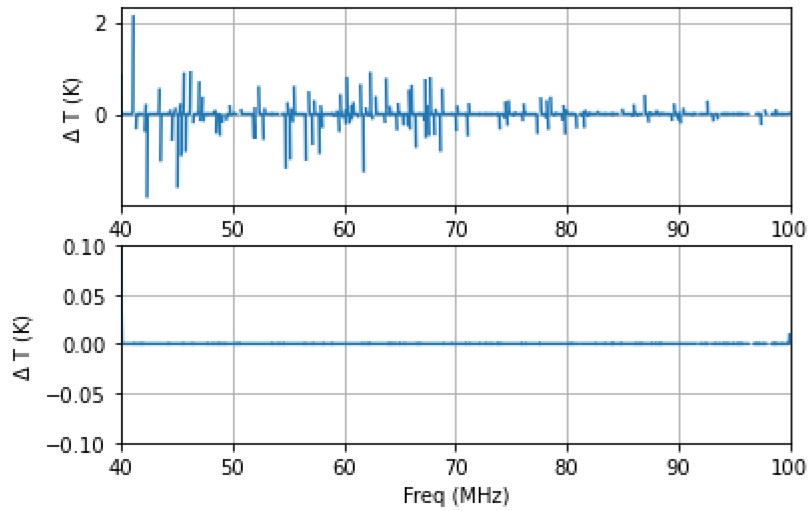


Figure 8: Difference spectra for one day of data (2016_260) between the outputs from B18 and edges-analysis after the Gaussian smoothing step. The differences in the top plot are non-zero before adding the flagging threshold in Gauss smooth step. The bottom plot is zero after adding the flagging threshold similar to the one in the C pipeline.

Noise wave calibration

1. Murray et al. (2022) note some differences between the calibration coefficients obtained from the two pipelines. So, Alan's Receiver coefficients are injected.
2. Modeled LNA S11 with a higher number of terms

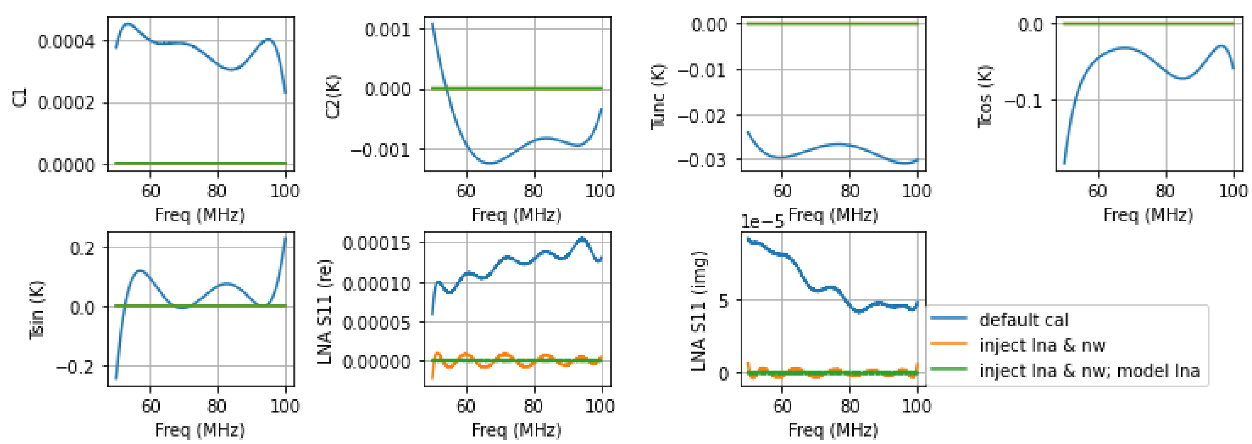


Fig. 9: Receiver calibration coefficients and LNA S11 differences calculated using the edges-cal & legacy C pipelines. The blue curves correspond to independently obtaining the parameters and S11 values using edges-cal but with the same input parameters as used in B18 analysis. The yellow curves correspond to injecting the Noise wave parameters and LNA s11 values from legacy C code into edges-cal. The green curves include injecting the LNA s11 after modeling it with the legacy C.

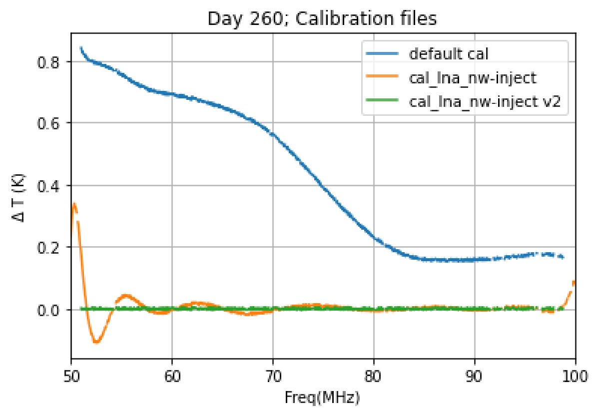


Figure 10: Difference spectra for one day of data (2016_260) between the outputs from B18 and edges-analysis after the receiver calibration step. The different curves correspond to calibrating the field data with different calibration files shown in figure 8.: (blue) default calibration, (yellow) injecting noise waves and LNA s11, (green) similar to yellow but also modeling the LNA S11 before injection.

Loss Correction

1. Connector length & brass conductivity differences were smoothed out. The connector length changed from 0.8” to 1.18” (See memos #). Brass conductivity changed from 0.24*sigma_copper to 0.29*sigma_copper

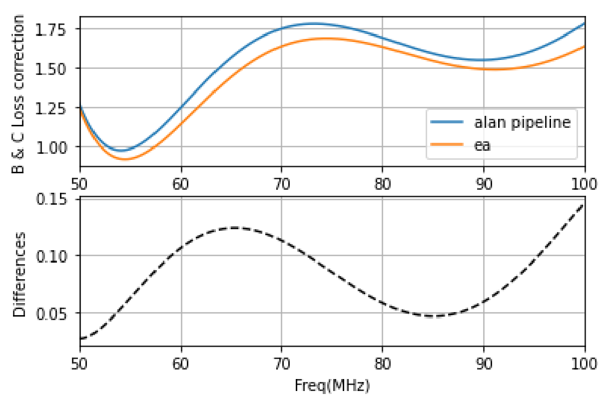


Fig. 11: Balun and Connector loss correction as a function of frequency obtained from the C pipeline (orange) and edges-analysis (blue). The differences between the two are shown in the second plot. The differences were due to differences in the values assumed for the balun length and brass conductivity.

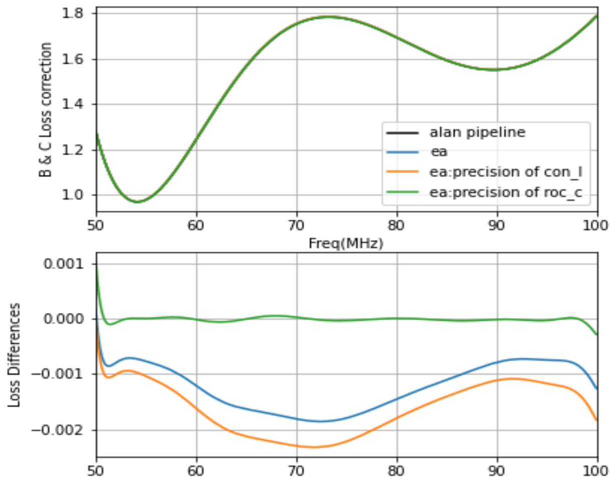


Fig. 12: Balun and Connector loss correction obtained for different assumptions made in the edges-analysis pipeline (colored curves). These are after matching the connector length and brass conductivity to the values in the C pipeline. The black curve corresponds to the B&C loss correction from the C pipeline (shown for reference).

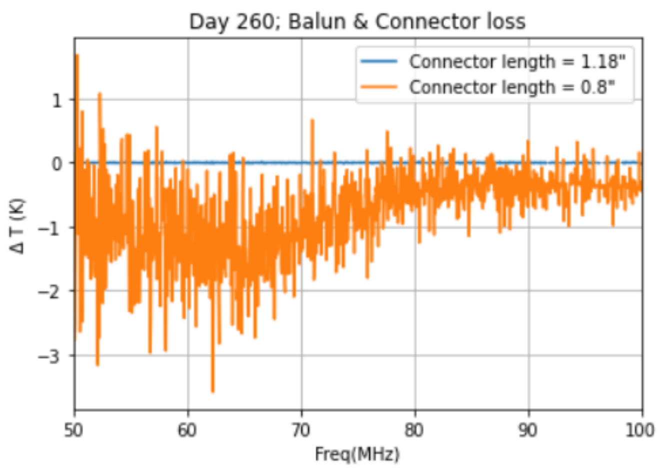


Fig. 13: *Difference spectra for one day of data (2016_260) between the outputs from B18 and edges-analysis after the balun and connector loss correction. Blue is the curve* after setting all the assumptions related to B&C loss correction to be the same as C pipeline.

Beam Calibration

Several changes/new options were added to edges-analysis to obtain a beam-factor correction that matches (almost) the C-code.

a. Different choice in edges-analysis which in principle correct: A new sky model, called "Haslam408Noh," was added, which matches the input sky model that the C-code uses. This is the Haslam map at 408 MHz (for which edges-analysis already had the same model, simply called "Haslam408") but in rectilinear RA/DEC coordinates in degrees rather than a healpix map. While the origin of the sky map file is unknown, after some searching, it seems likely that it came from https://www3.mpifr-bonn.mpg.de/cgi-bin/survey. **There are a few things about this particular sky-model that are sub-optimal** with respect to the default Haslam408 map: the resolution is lower, and suffers from the fact that each pixel has a different solid angle. This is accounted for in the C-code with a weighting, but this weighting is discrete and gives exactly zero weight to pixels on the horizon. We 'match' this in edges-analysis for this sky model, but this is not particularly ideal.

b. Different choice in edges-analysis which has no in principle correct**:** Added a new spectral index model `Constantindex`--i.e. A single index for the whole sky. This matches the C-code, but is probably less realistic than other models.
c. Added a missing option**:** Added the ability to subtract the CMB from the sky model before applying the spectral index model, then re-adding afterwards. This matches C-code and is theoretically well-motivated.
d. Different choice in edges-analysis which has no in principle correct: New beam interpolation option added: "nearest". We previously used splines defined on 2D spherical coordinates, which should be a little more accurate.
e. Different choice in edges-analysis which has no in principle correct: We updated the exact LSTs for which the beam factor was computed to match the output from the C-code (every 30min between 6-18hr instead of every 20min).
f. Different choice in edges-analysis which has no in principle correct: We updated the way the beam factor was calculated to be the ratio of the means instead of the mean of ratios, i.e. sum_over_lst(beam_at_nu) / sum_over_lst(beam_at_refnu) instead of sum_over_lst(beam_at_nu/beam_at_refnu). It is hard to say whether this is preferable or not.
g. Different choice in edges-analysis which has no in principle correct: We use a smooth model of the final beam factor – a Fourier model based on the one used in the C-code, instead of a simple interpolation. Unclear if this is better or not.
h. Different choice in edges-analysis which is in principle correct: We transcribed the functions that convert LST to RA/Dec from the C-code into edges-analysis. The original used astropy to do these conversions, which should be preferred.

Figure to add: Beam factor differences even after Stevens changes

Object RMS filter

1. Modeling differences lead to the RMS of the days calculated to be different by a few mK . which implies that some additional days get flagged by edges-analyis.
2. We forced the same days to get flagged before LST averaging by adding an additional filter called 'explicit_day_filter'. In this step, we realized that specifying the two-digit days with a zero in the front of the number would cause YAML to think it a hexadecimal and it was associating the wrong dates to the numbers.

LST average

Fixed a small bug where the models were not being averaged. Added:

```
1. tot_model = sum(obj.model for obj in objs)
2. tot_model = tot_model/len(objs)
```