RFI Flagging: IQRM

The saga continues...

Sofia Sheikh - 05/19/2022

AOFLAGGER is not the right tool for our application :(

- AOFLAGGER is currently the most popular software tool for removing RFI from interferometric radio astronomy observations
- Works exceedingly well on visibility data, where it can compare different baselines etc.
- It technically has strategies for single-dish telescopes, but none of them were working well for our data
- Need a different tool to get from filterbank files to ->

Output we want:
Table of Flagged RFI from Survey

	times	freqs	intensities
0	59635.346646	1718.875	4.367957e-07
1	59635.346646	1719.125	4.503195e-07
2	59635.346646	1719.375	4.462190e-07
3	59635.346646	1719.625	4.518197e-07
4	59635.346646	1719.875	4.445418e-07
...	...	...	...
28890	59635.346690	1817.125	1.418262e-06
28891	59635.346690	1817.375	1.461919e-06
28892	59635.346690	1817.625	1.435838e-06
28893	59635.346690	1817.875	1.407905e-06
28894	59635.346690	1818.125	1.461541e-06

Expand to include day of week, time of day, direction

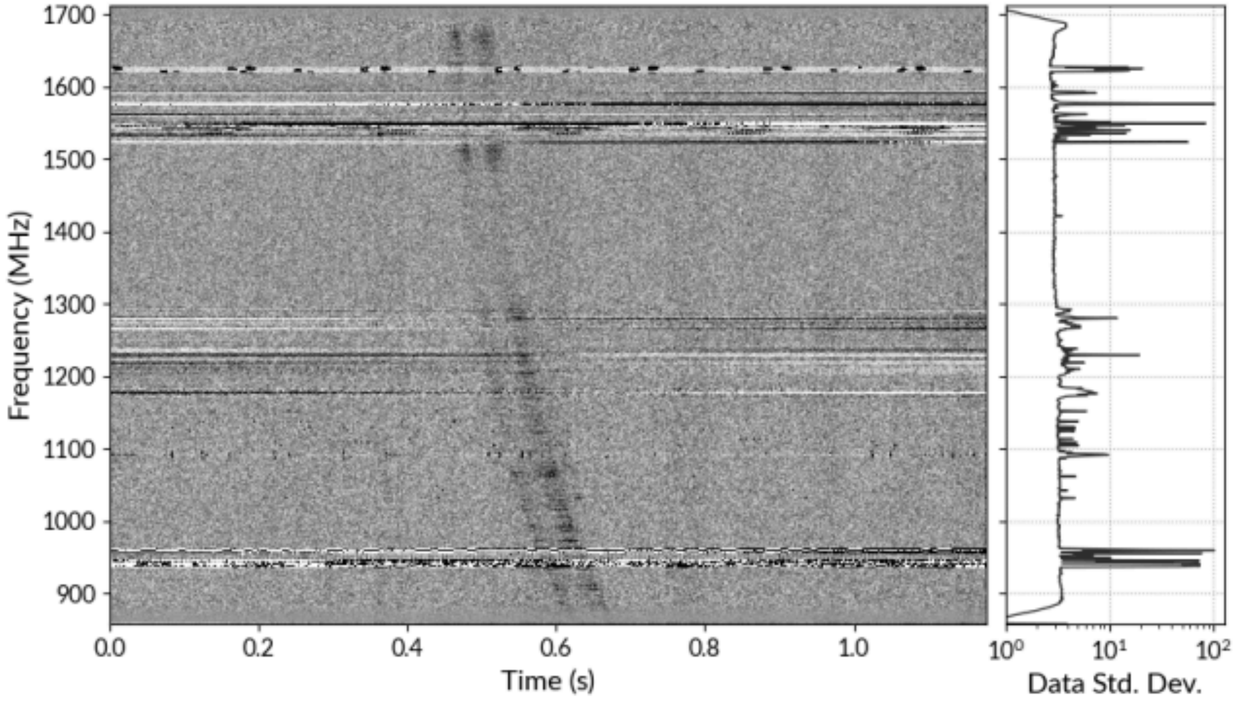
Need a RFI-Flagging Algorithm for single antennas

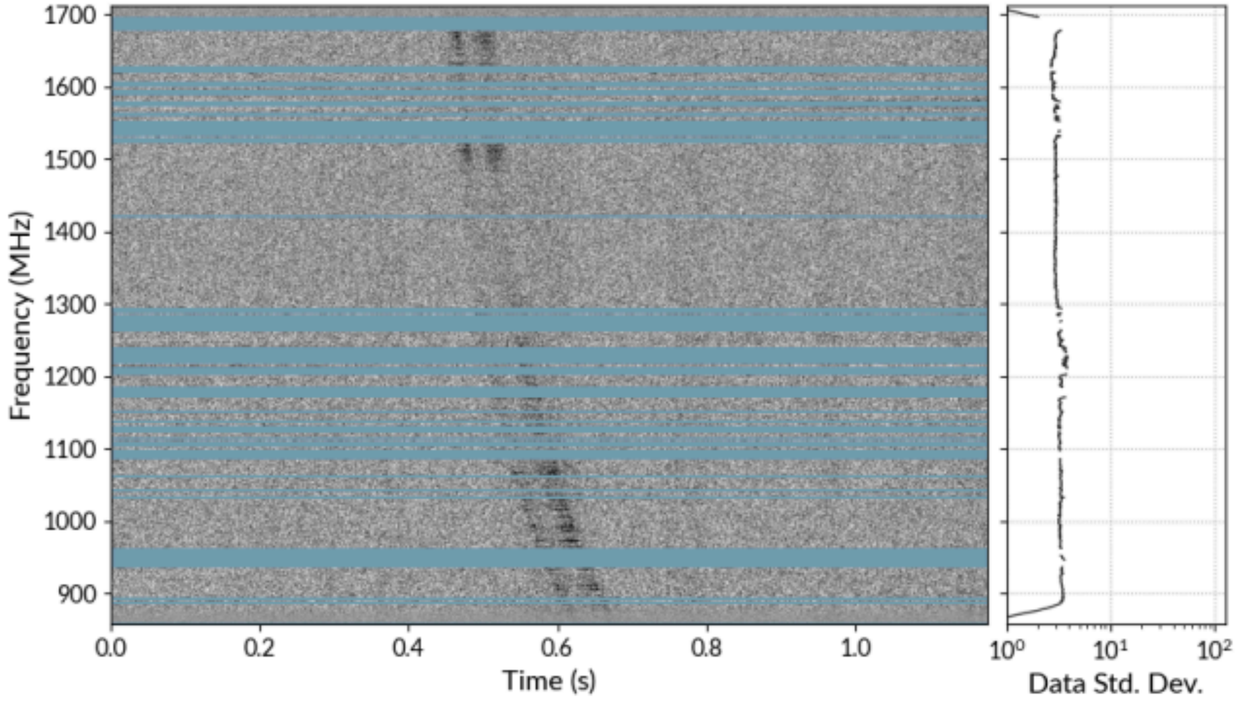
Enter: IQRM

- Inter-Quartile Range Mitigation (IQRM)
- Morello, Rajwade, and Stappers (2021)
- Advantages:
- Non-parametric, robust to trends in data, no training needed, used on MeerKAT, fast enough to be real-time
- How it works:
  - 1) Compute a statistic that is correlated with the presence of RFI (e.g., standard deviation)
  - 2) Find high outliers among those values with IQRM

How does IQRM work?

- Post-detection method (implemented on time-frequency-intensity data cubes) for narrowband RFI
- Reduce entire block to a summary statistic / frequency channel (stdev, spectral kurtosis, etc.)
- Looks at nearest *r* neighbors* of frequency channel f_i to see if any of the differences f_i - $f_{j<r}$ are above a threshold *t* x the interquartile range (of the inliers); make sure it's a max not a min



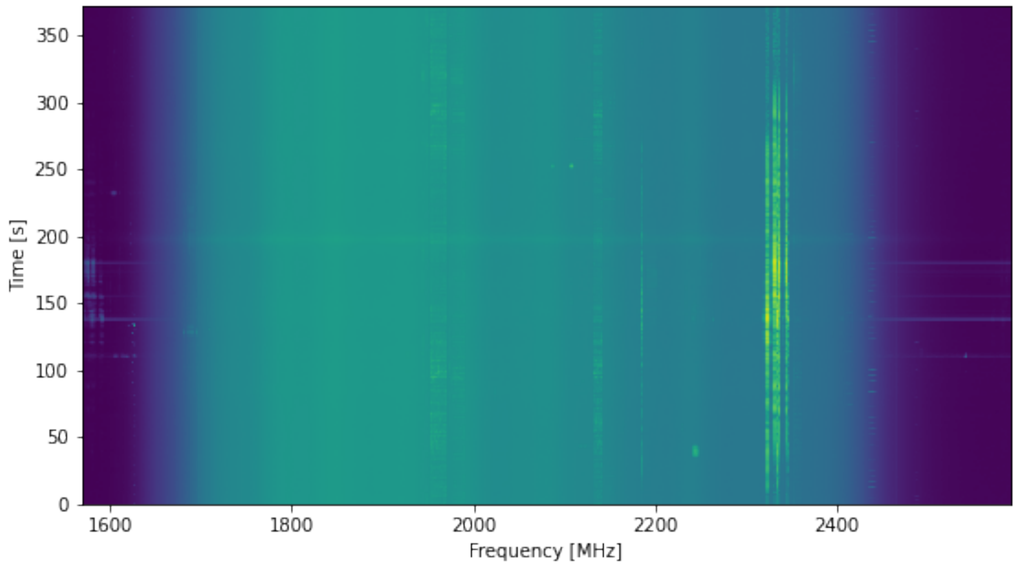


IQRM: Trial on ATA data

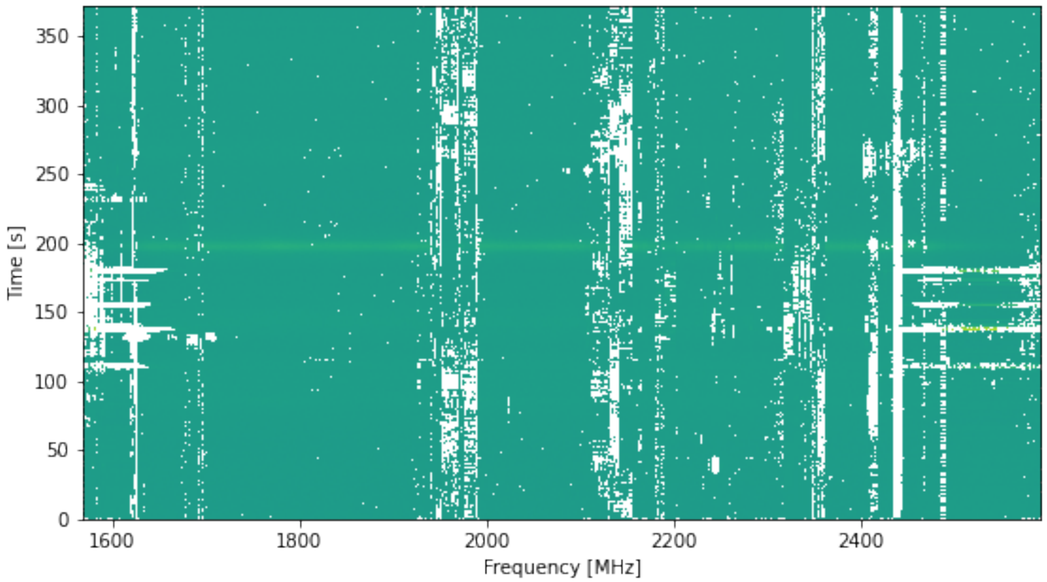
- Disclaimer: to get this to also work along the time axis, need to split waterfall into chunks in time
  - Valid and tested technique: They use this in the MeerTRAP implementation of IQRM!
- Disclaimer 2: IQRM has quite a few “sliders”
  - Remove bandpass (Y/N)
  - How many chunks in time?
  - Which outlier statistic to use?
  - Radius of the frequency channels to include?
  - Threshold for tagging as “outlier”?

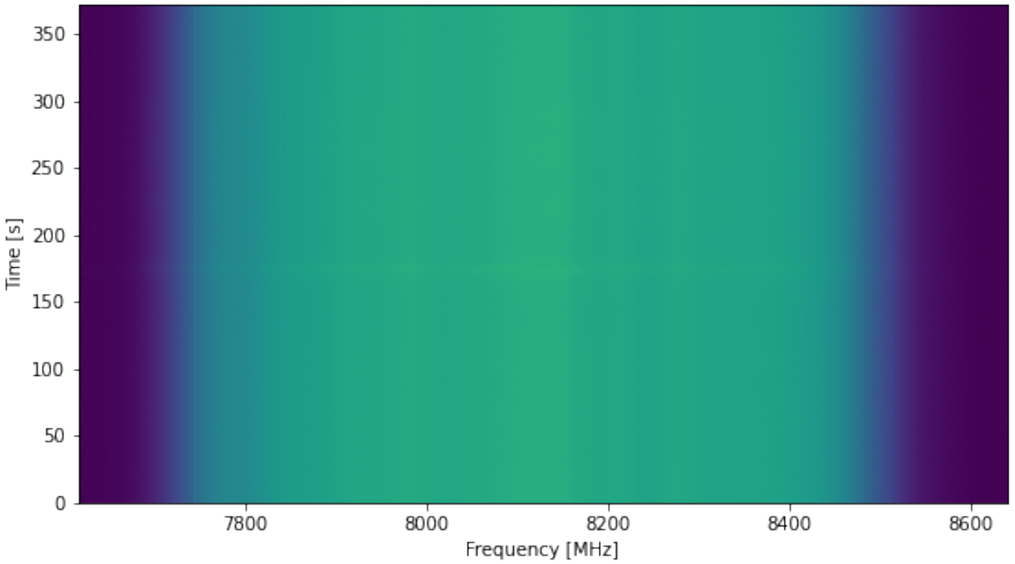
```
flagged = apply_iqrm(datafile,
                     remove_bandpass=True,
                     degree_scrunch_factor=1,
                     method='stdev',
                     saving=False,
                     radius=r,
                     threshold=3.0,
                     short=False)
```

IQRM: Works on ATA data!

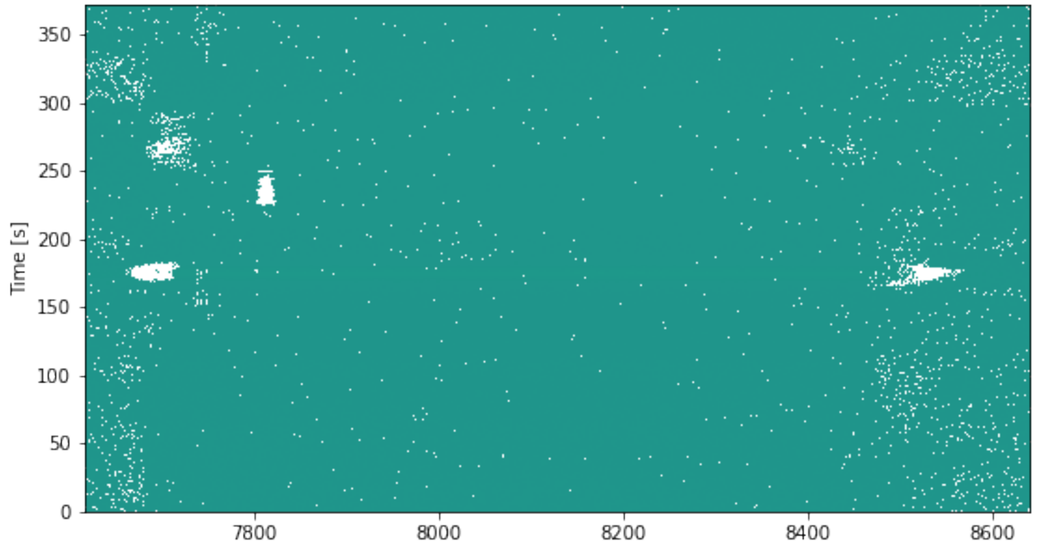


Standard Deviation



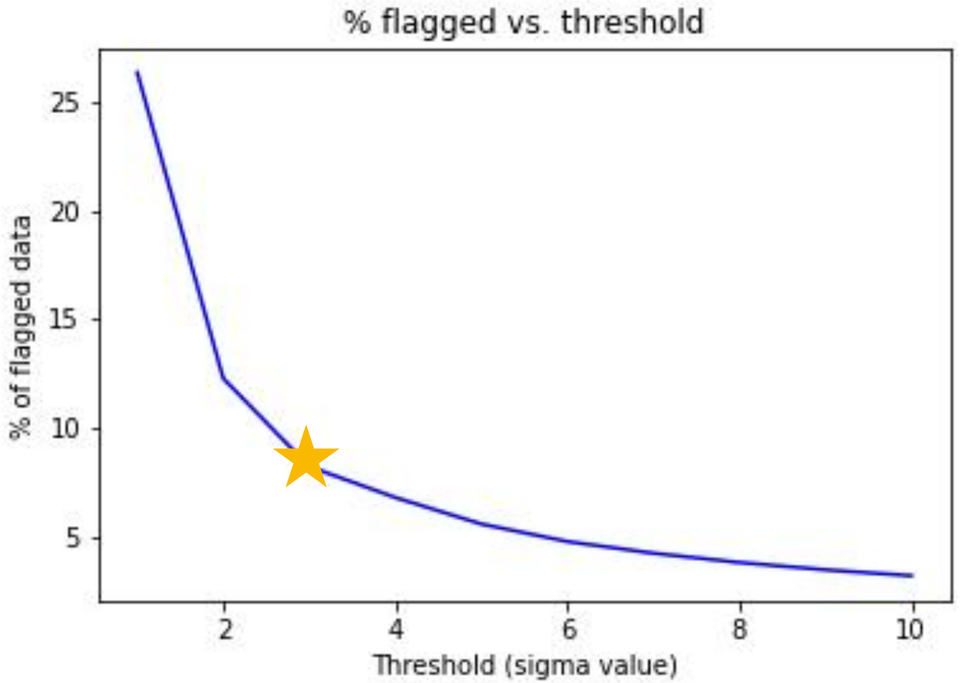


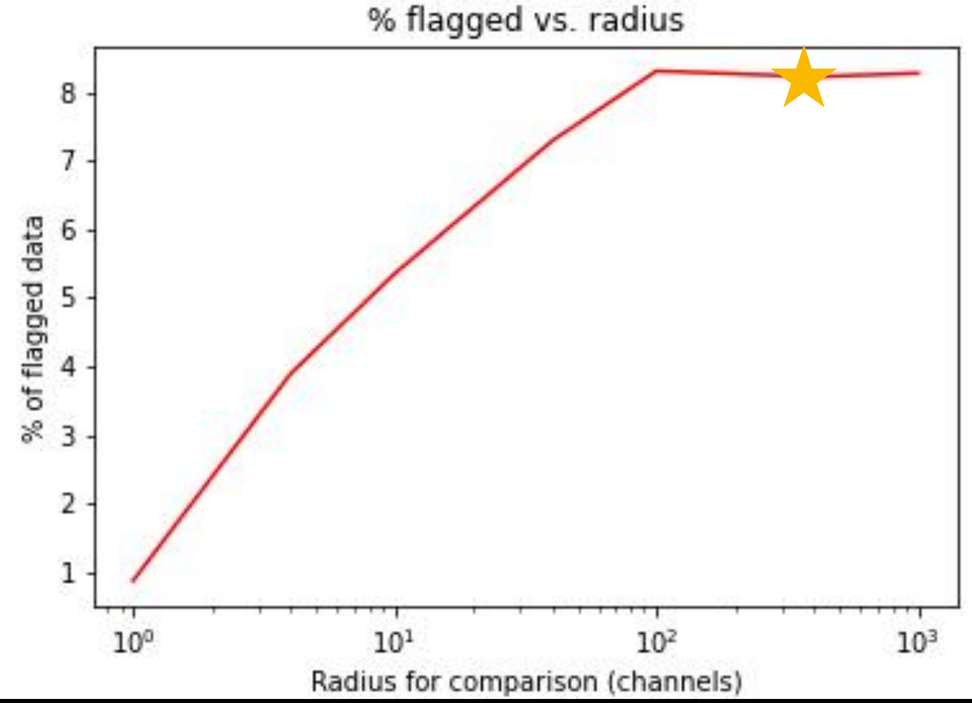
Median Absolute Deviation



IQRM: Implementation

- Only tangible way to understand “how well it did” is to measure the improvement of some scientific outcome (e.g., # of pulses detected from a pulsar)
- Often fails on broadband interference (like the tree) - combine with the “zero-DM” Eatough et al. (2009) approach?
- Not clear whether MAD or standard deviation is better for this application
  - Might just go with stdev, because it’s what the authors used
- Tried a few different radii and thresholds to determine the best values





Best IQRM parameters from testing:
- stdev
- r=400
- t=3.0
- bandpass removal
- 0.1 degree bins