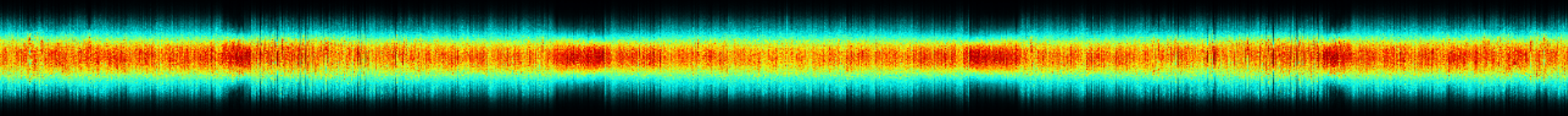
Data formats, inspection, editing & **R**adio-**F**requency **I**nterference flagging

ERIS 2017, Dwingeloo / The Netherlands

André Offringa
(ASTRON Netherlands Institute for Radio Astronomy)

2017-10-16



RadioNet has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 730562

Outline

- Interferometric data formats
- Why data editing
- Radio Frequency Interference (RFI)
- Plotting data (CASA, aoqplot)
- Manual data flagging
- Automated RFI flagging algorithms
- Data averaging

Data formats

- There are several popular formats for storing interferometric data:
  - **The CASA "measurement set" format**;
    Most popular, most feature complete (but therefore also most complex), most development on.

    Used by most major telescopes: LOFAR, ALMA, WSRT, JVLA, MWA

  - **UVFits files**; often used as intermediate format to make different tools talk to each other.

  - **Software-specific formats**: AIPS, Miriad.

Data formats

- All formats store “Tables” of data
- Column names specify the type of data:
  - DATA, CORRECTED_DATA, MODEL_DATA, TIME, ANTENNA1/2, …
- One table “row” holds the data for:
  - one timestep
  - one baseline
  - one “spectral window”
  - Multiple channels and polarizations
- Observations have several tables, e.g. for data, list of antennas, etc.

Introduction

Why data editing?

Introduction

Why data editing?

Some antennas might not have been functioning properly...

Introduction

Why data editing?

- Broken elements (antennas/stations)
- Correlator malfunctions
- Shadowing
- Initial pointing delay

- Bandpass issues
- Low elevation
- Correlated noise on some baselines
  (e.g. LOFAR split stations)
- Interference

Introduction

Why data editing?

- Broken elements → remove antennas
- Correlator malfunctions → remove timesteps
- Shadowing → remove antennas in time range
- Initial pointing delay → remove first timesteps

- Bandpass issues → remove channels
- Low elevation → remove antennas with low elevation
- Correlated noise on some baselines
  (e.g. LOFAR split stations) → Flag baselines
- Interference → remove antennas, timestep,
  frequencies or baselines...

Data can't be (self-)calibrated when any of these issues are still in the data.

Therefore, data inspection & editing is the first step :

INSPECTION + EDITING

↓

(DATA AVERAGING)

↓

CALIBRATION

↓

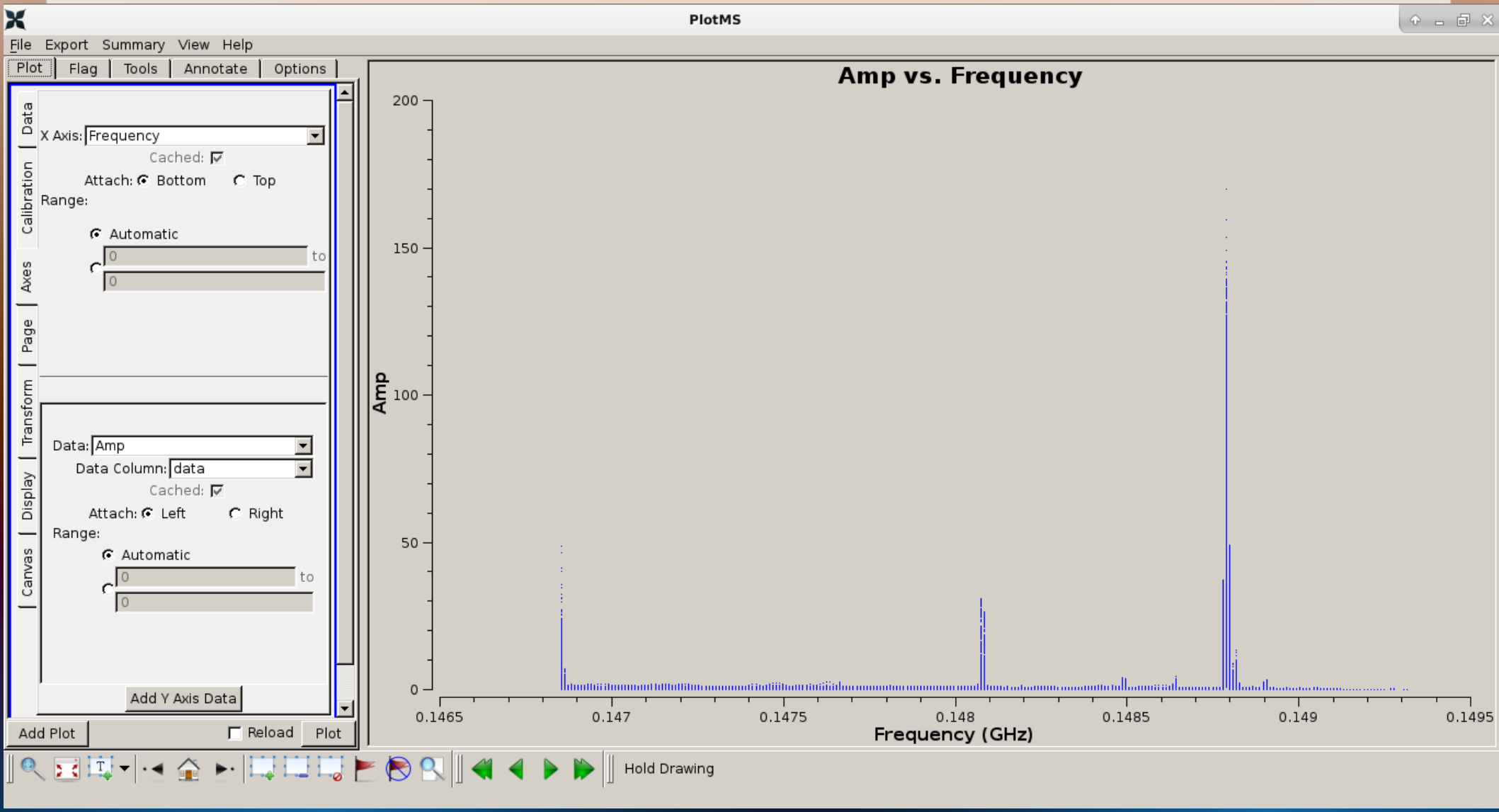
IMAGING

First step in data reduction: Data inspection (example of `casaplotms` on other screen)

- Start casaplotms
- Open MS ('3c196_spw5_sub1.ms')
- Press 'plot' (plots amplitude vs time)
- Goto 'axes', select “frequency” as x-axis
  RFI is visible
- Select 'antenna1'
  antenna5 has no data

- (Enter:
  antenna: “0;13”
  msselect: “ANTENNA1!=ANTENNA2”)

casaplotms


PlotMS
File Export Summary View Help
Plot Flag Tools Annotate Options
Data Calibration Axes Page Transform Display Canvas
X Axis: Frequency
Cached:
Attach: Bottom Top
Range:
Automatic
0 to
0
Data: Amp
Data Column: data
Cached:
Attach: Left Right
Range:
Automatic
0 to
0
Add Y Axis Data
Add Plot
Reload
Plot
Amp vs. Frequency
Amp
200
150
100
50
0
0.1465
0.147
0.1475
0.148
0.1485
0.149
0.1495
Frequency (GHz)
Hold Drawing

casaplotms

- What should we see in casaplotms (time vs amplitude) if we observe a single unresolved (=delta function) source with a certain flux?

(That's what we want calibrators to be –
strong / dominating / 'single' and unresolved)

casaplotms

- casaplotms is useful for many things:
  - Browsing for bad antennas, frequencies, etc.
  - Also useful for inspecting calibration results
  - Or getting an idea of model data
  - Further discussed in Andy Biggs' tutorial
- Many observatories have specialized plotting tools

Removing data

- If an issue is found (bad antenna, baseline, channels, ...) how do we remove it from our dataset?
  - We don't actually remove data, we **'flag'** data and ignore these in further processing.
  - Flagging is not the same as setting to zero(!)
- 'taql' is a useful tool for data editing.

TaQL (Table Query Language)

- TaQL is an 'SQL'-like language for quick data editing of CASA data.
- Command line tool 'taql' available, easy for scripting
- Be careful when editing! Always keep backups.
- Some examples: (from the cmdline)
  - `taql "select unique TIME from obs.ms"`
  - `taql "update obs.ms set FLAG=true where ANTENNA1==ANTENNA2"`
- → See taql doc ("casacore note 199")

http://www.astron.nl/casacore/trunk/casacore/doc/notes/199.html

Other ways to edit data...

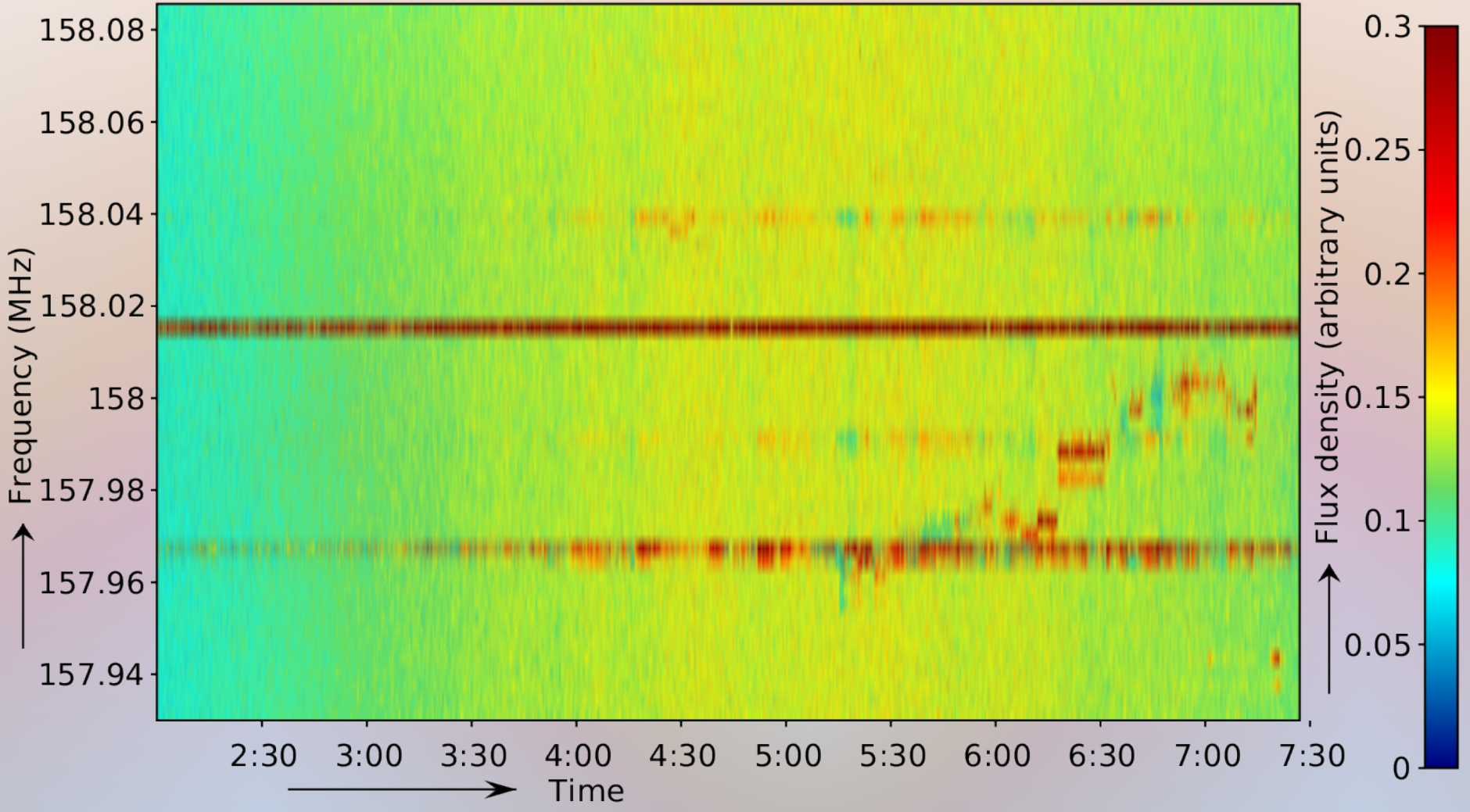
- CASA task '`flagdata`'

  → Andy Biggs' tutorial
- Or, for CASA data: Write Python scripts
- Other packages have their own scripting languages / tasks

Radio-Frequency Interference

- Our radio spectrum is almost entirely allocated to services other than radio astronomy
- FM, airplane communication, satellite downlink, remote controls, digital broadcasts, …
- Also “accidental” and natural occurrence of RFI:
  - Cars, electrical fences, high-voltage lines (anything that sparks), lightning, the sun, etc.
- RFI can cause (self-)calibration to fail and/or reduce imaging sensitivity

Example of LOFAR data with RFI



Cross-correlations of two stations showing strong RFI

RFI

- Lots of interference at low frequencies (<1.5 GHz, e.g. LOFAR, GMRT, WSRT, EVLA, MWA, …)
- Less of an issue for
  - higher frequencies (ALMA); or
  - VLBI

  but mitigation still required in most cases.

Excising RFI

- Detection methods are common in radio astronomy
- Common methods:
  - Manual selection by data reducing astronomer
  - Thresholding / specialized project pipelines
    (e.g. Baan et al. 2004, Winkel et al. 2007)
- **Manual selection is not practical for modern observatories:**
  - Enormous data volumes, computationally fast algorithms required.
  - Needs to be more accurate than thresholding

RFI stages / strategies

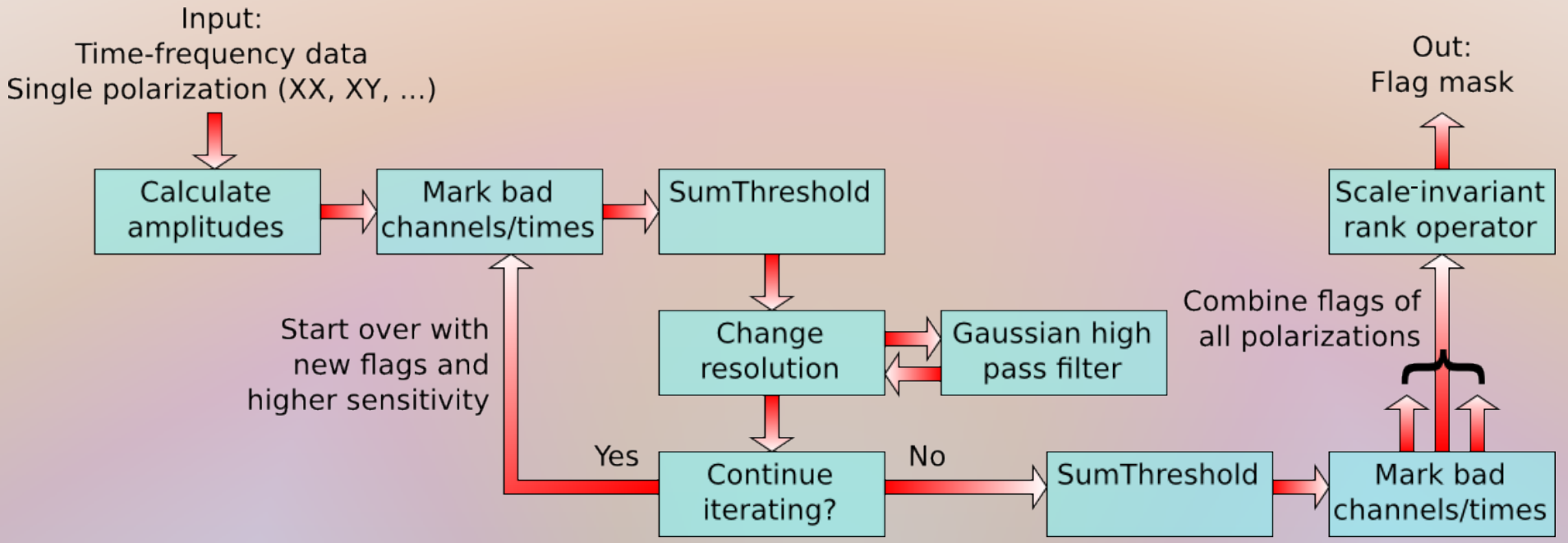
Many RFI excision options:

- Online pre-/post-correlation mitigation
  - Memory/computational constraints
  - Required for coherent (high time res) modes
- Offline mitigation
  - Post-optimizable, not real-time, data can be reordered
- LOFAR: Station level spatial filtering
  - Expensive, low SNR, only “one chance”
  - Allows data recovery

Automated excision of RFI

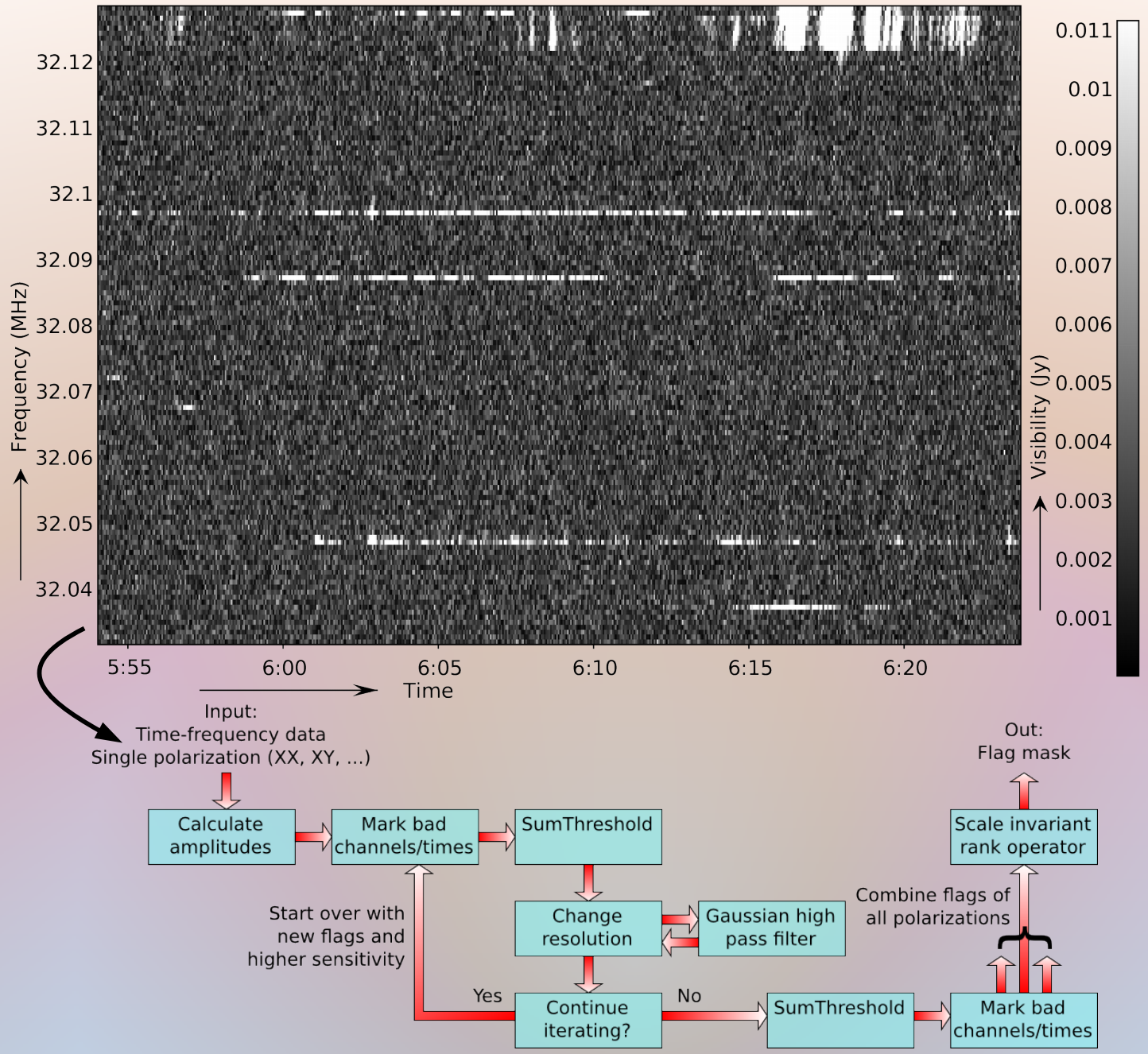
- Two classes of RFI excision methods:
  - Detection: find & throw away affected data
  - Filtering or subtracting: estimate RFI contribution and restore affected data
- Detection methods (“flagging”) commonly used
  - Some specialized pipelines for surveys or instruments
- Filtering RFI is harder
  - Resulting data quality is not well understood
  - Requires more resources
  - Lack of full (automated) filtering pipelines

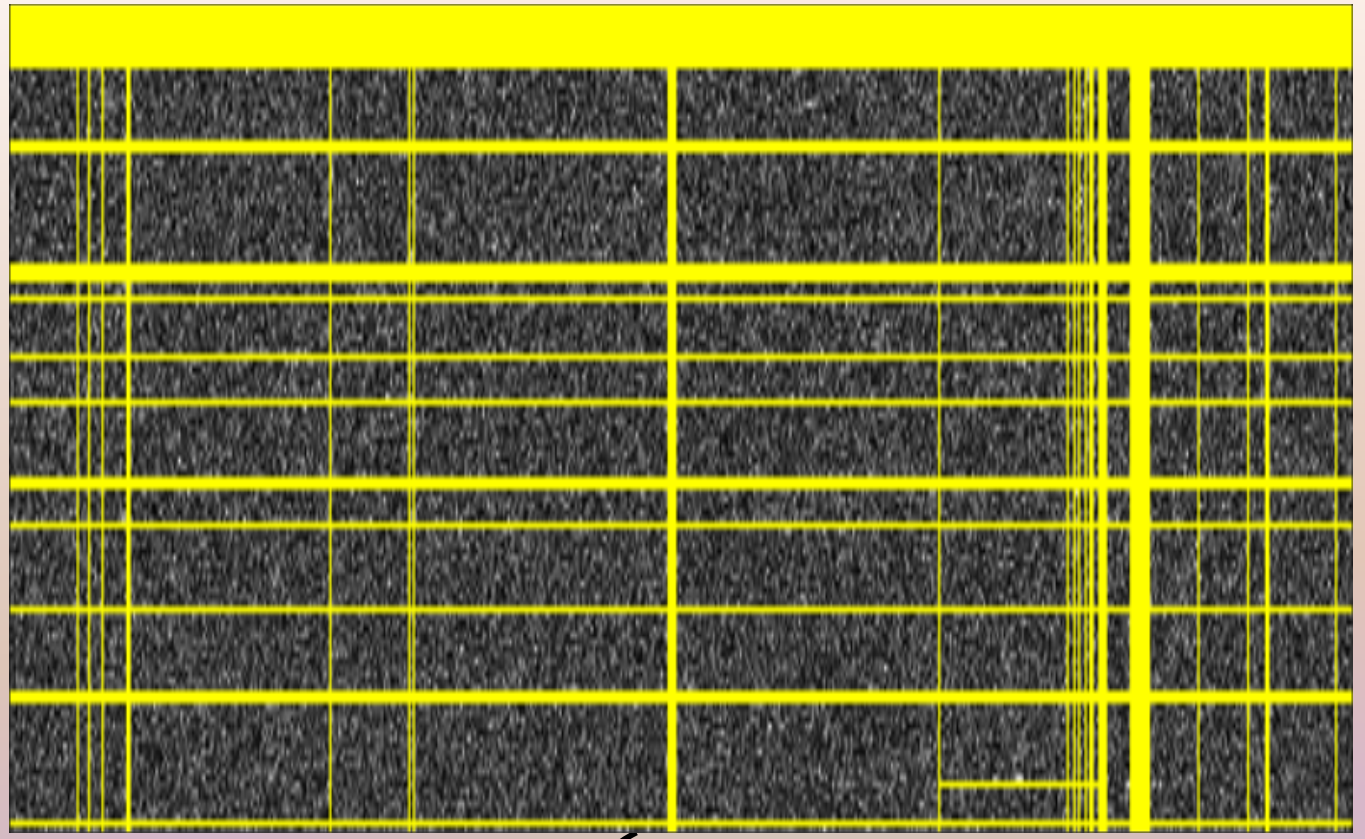
The AOFlagger
(example of automated RFI detection)
External package[1], works with CASA sets

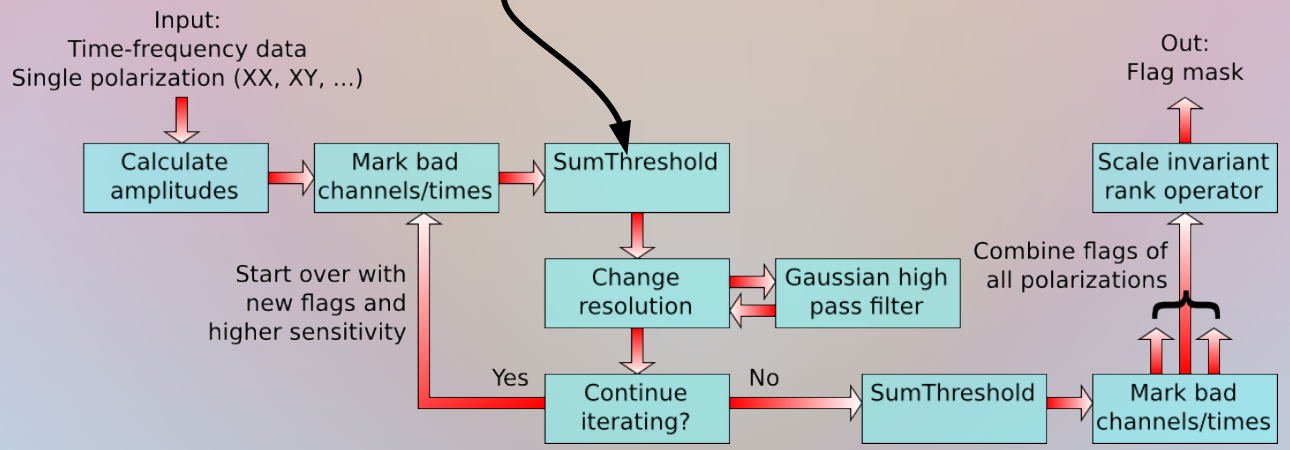


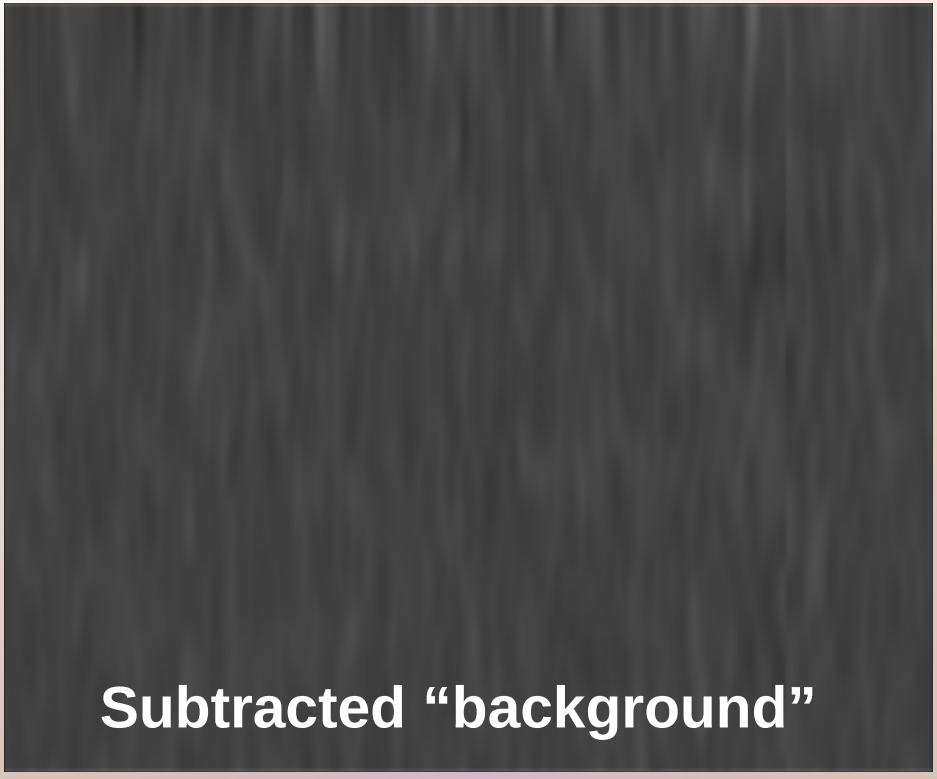
Offringa et al., MNRAS (2010), Offringa et al., A&A (2012)

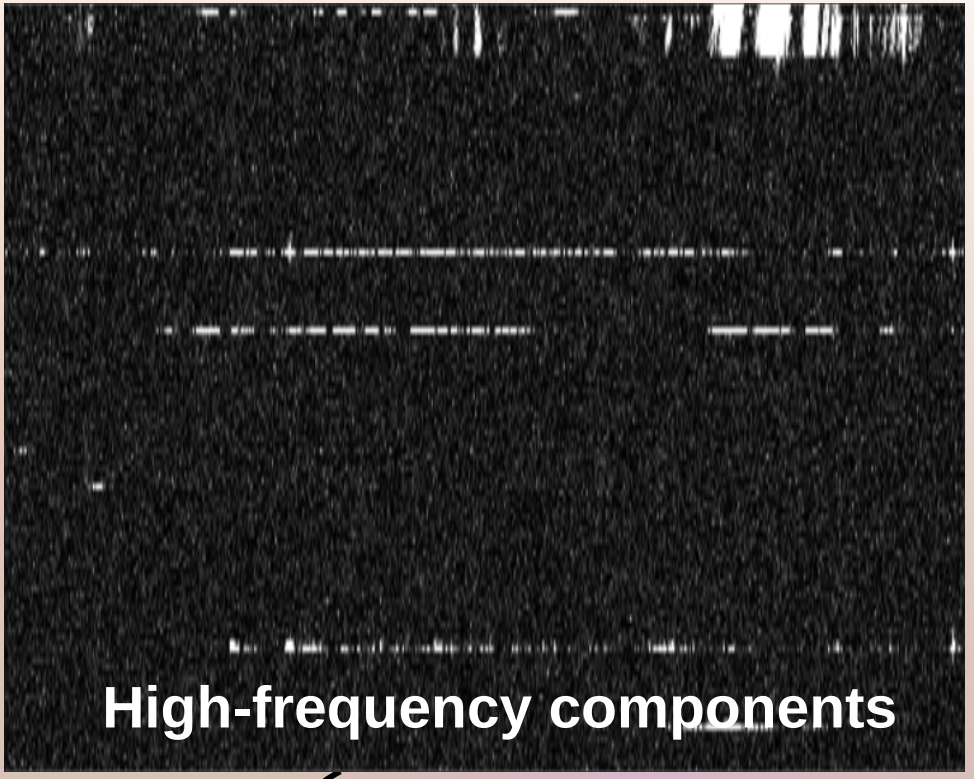
[1]AOFlagger webpage: http://aoflagger.sourceforge.net/

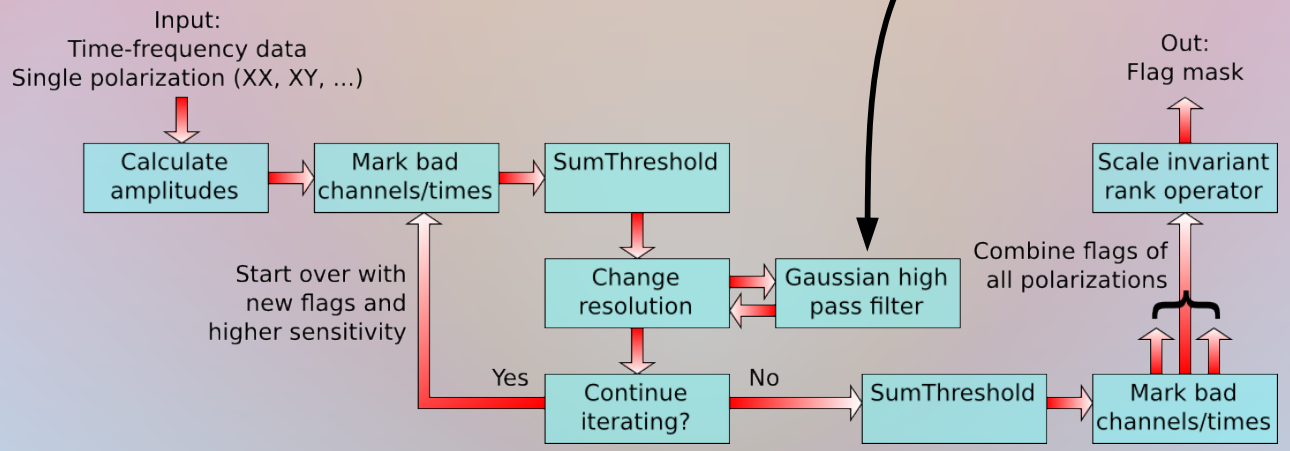

32.12
32.11
32.1
32.09
32.08
32.07
32.06
32.05
32.04
Frequency (MHz)
5:55
6:00
6:05
6:10
6:15
6:20
Time
0.011
0.01
0.009
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
Visibility (Jy)
Input:
Time-frequency data
Single polarization (XX, XY, ...)
Calculate amplitudes
Mark bad channels/times
SumThreshold
Change resolution
Gaussian high pass filter
Start over with new flags and higher sensitivity
Yes
Continue iterating?
No
SumThreshold
Mark bad channels/times
Combine flags of all polarizations
Scale invariant rank operator
Out:
Flag mask

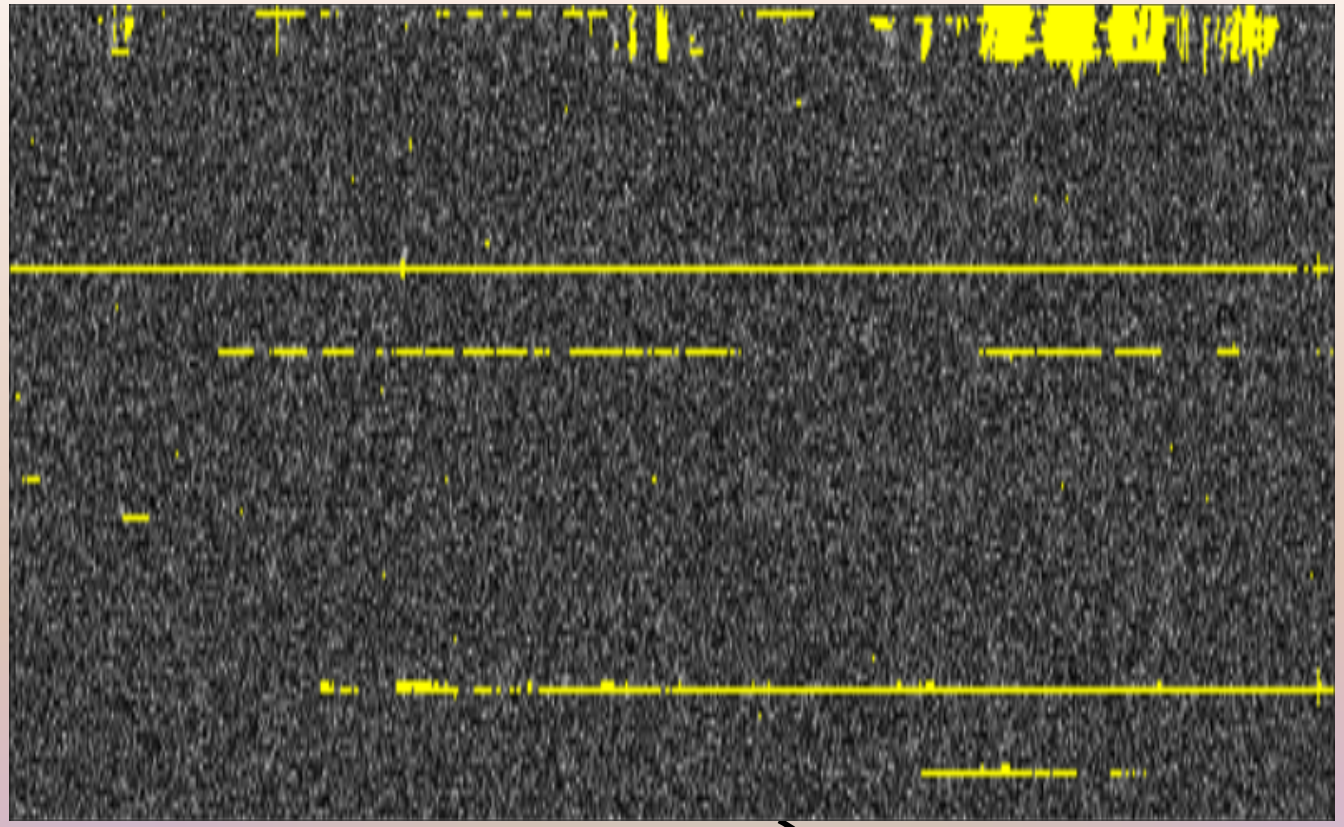


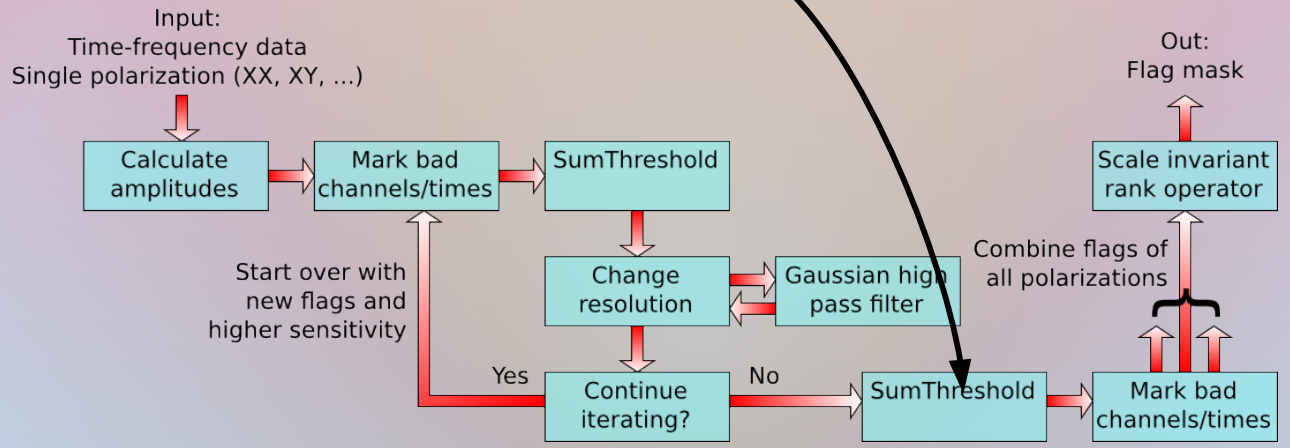

Input:
Time-frequency data
Single polarization (XX, XY, ...)
Calculate amplitudes
Mark bad channels/times
SumThreshold
Change resolution
Gaussian high pass filter
Start over with new flags and higher sensitivity
Yes
Continue iterating?
No
SumThreshold
Mark bad channels/times
Combine flags of all polarizations
Scale invariant rank operator
Out:
Flag mask

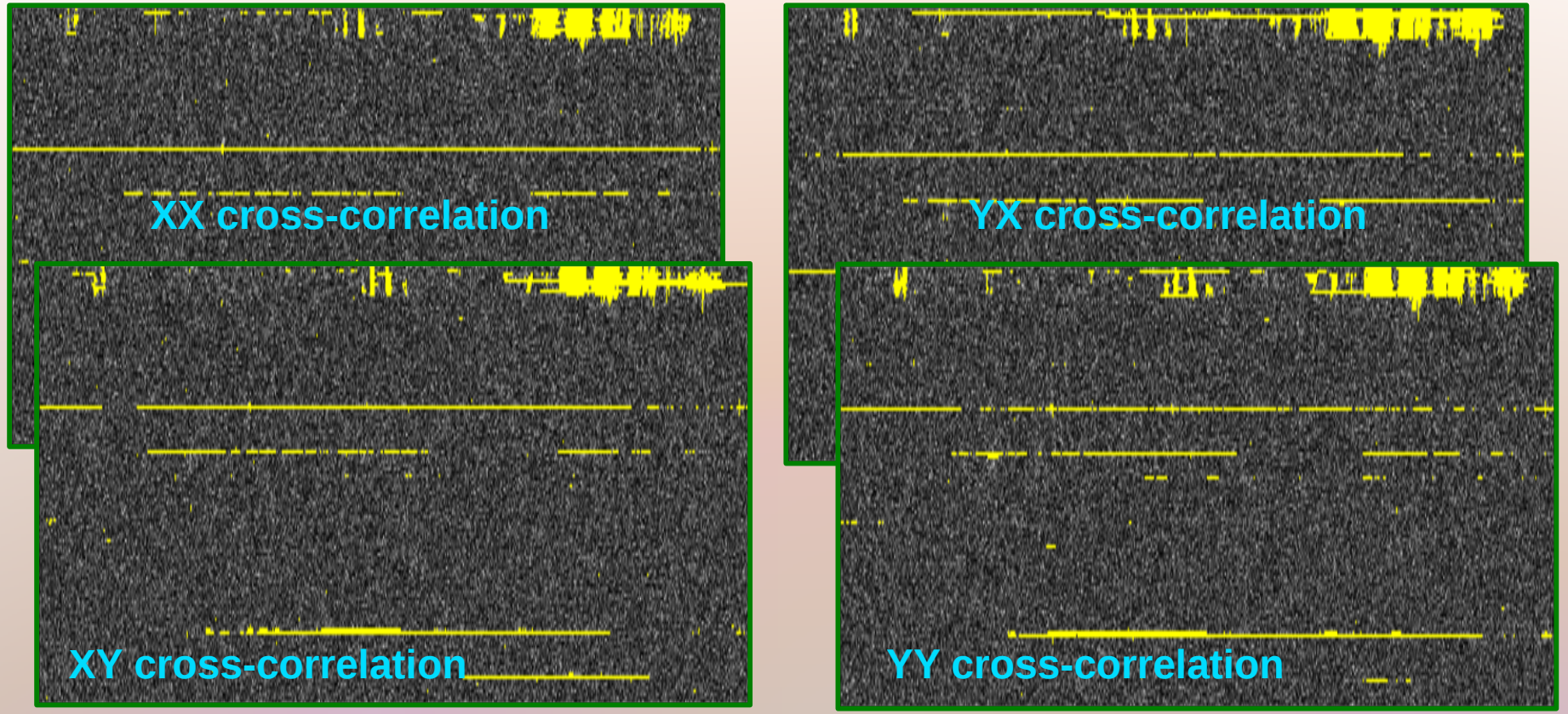

Subtracted “background”

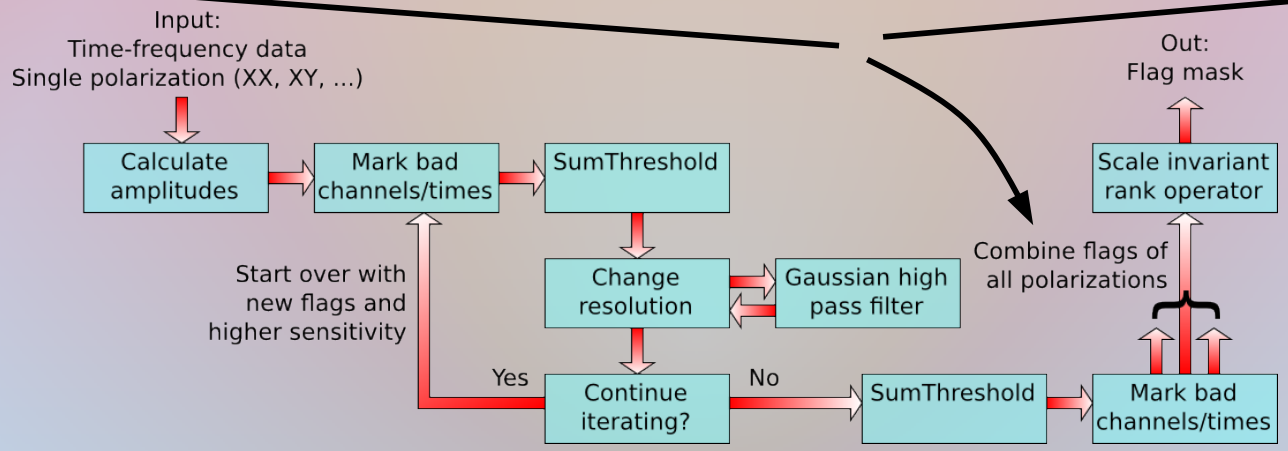

High-frequency components

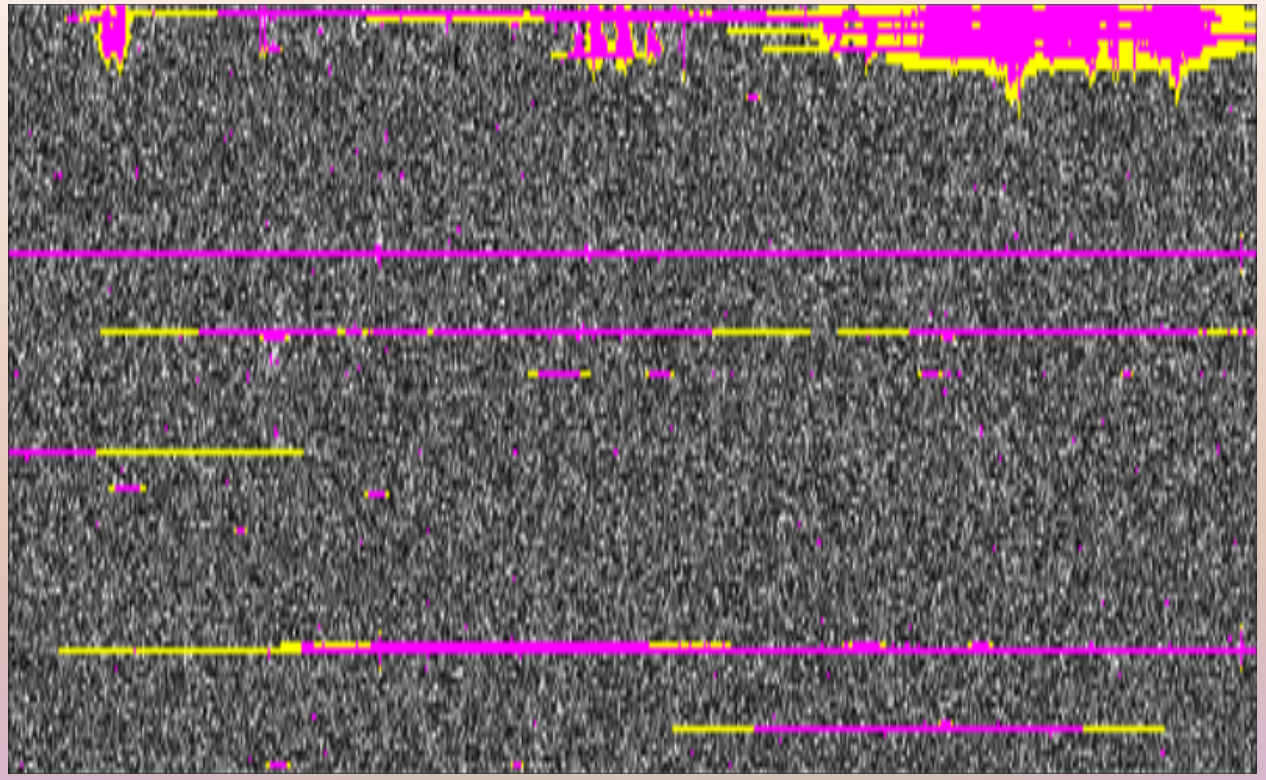

Input:
Time-frequency data
Single polarization (XX, XY, ...)
Calculate amplitudes
Mark bad channels/times
SumThreshold
Change resolution
Gaussian high pass filter
Start over with new flags and higher sensitivity
Yes
Continue iterating?
No
SumThreshold
Mark bad channels/times
Combine flags of all polarizations
Scale invariant rank operator
Out:
Flag mask

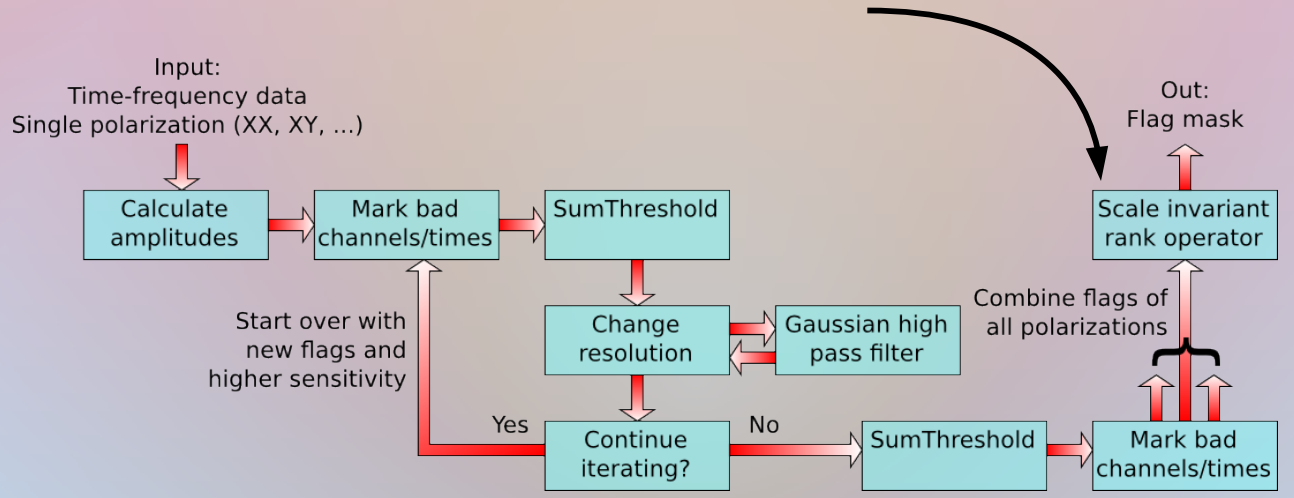



Input:
Time-frequency data
Single polarization (XX, XY, ...)
Calculate amplitudes
Mark bad channels/times
SumThreshold
Change resolution
Gaussian high pass filter
Start over with new flags and higher sensitivity
Yes
Continue iterating?
No
SumThreshold
Mark bad channels/times
Combine flags of all polarizations
Scale invariant rank operator
Out:
Flag mask

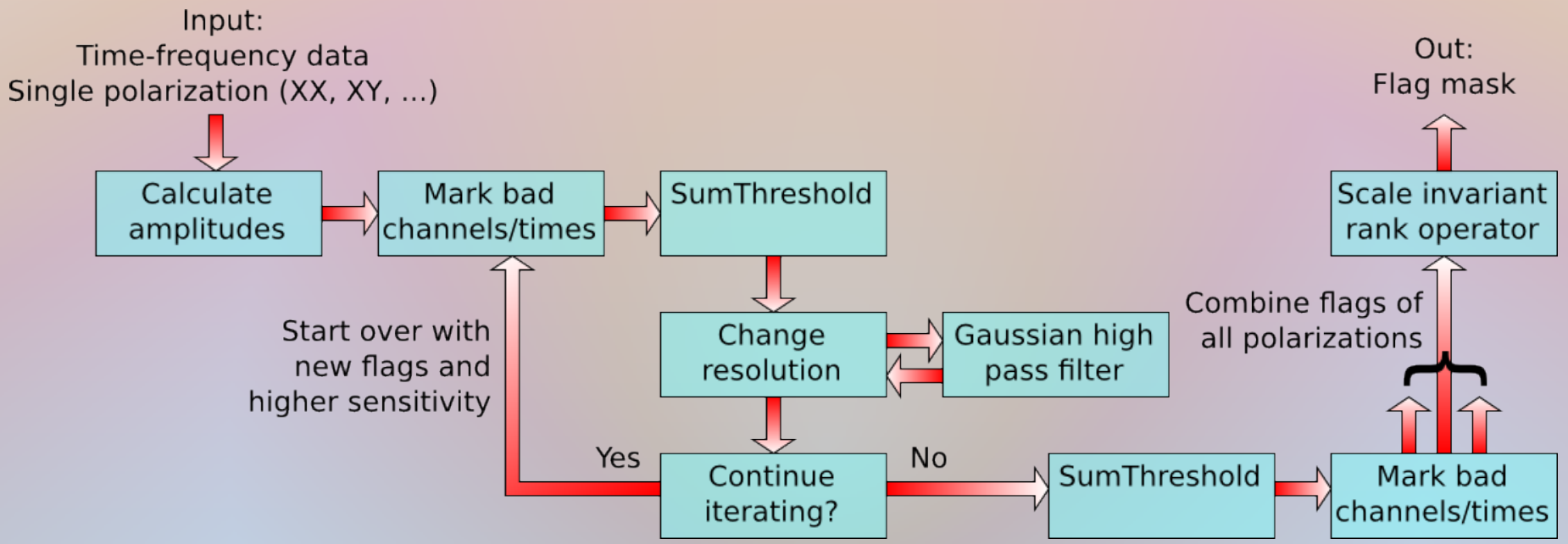

XX cross-correlation
YX cross-correlation
XY cross-correlation
YY cross-correlation


Input:
Time-frequency data
Single polarization (XX, XY, ...)
Calculate amplitudes
Mark bad channels/times
SumThreshold
Change resolution
Gaussian high pass filter
Start over with new flags and higher sensitivity
Yes
Continue iterating?
No
SumThreshold
Mark bad channels/times
Combine flags of all polarizations
Scale invariant rank operator
Out:
Flag mask




Input:
Time-frequency data
Single polarization (XX, XY, ...)
Calculate amplitudes
Mark bad channels/times
SumThreshold
Change resolution
Gaussian high pass filter
Start over with new flags and higher sensitivity
Yes
Continue iterating?
No
SumThreshold
Mark bad channels/times
Combine flags of all polarizations
Scale invariant rank operator
Out:
Flag mask

- What could go wrong??



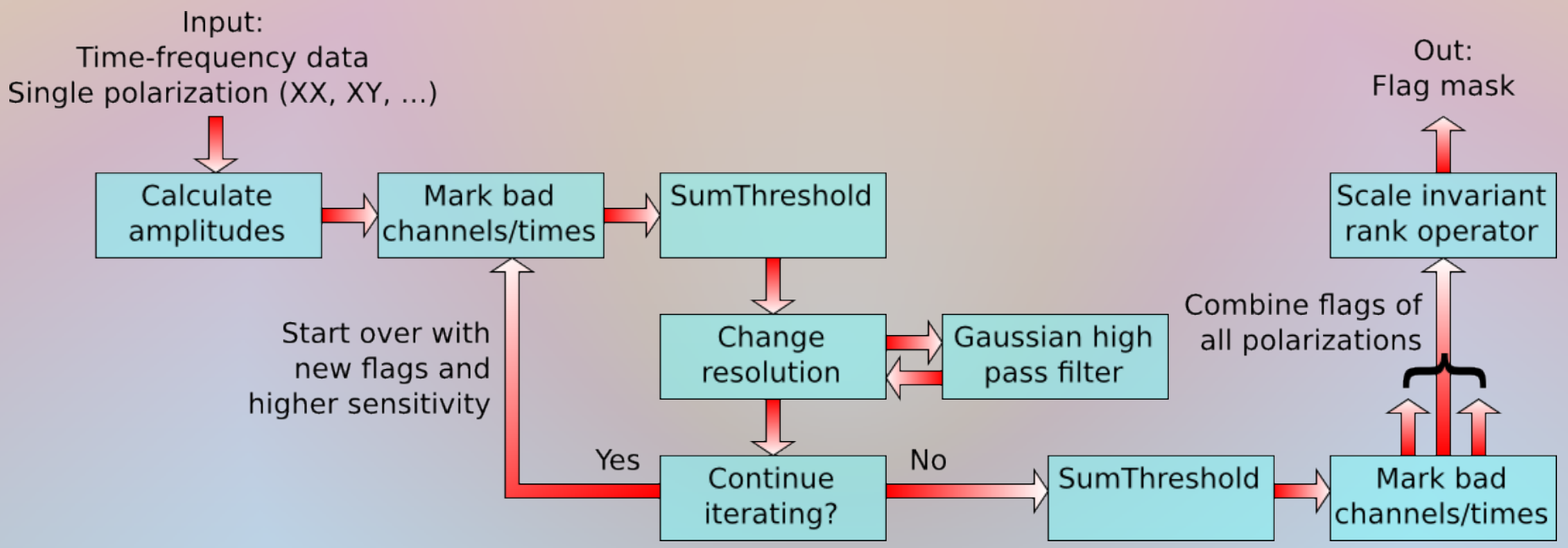
- What could go wrong?

  - Some astronomical sources vary quickly in time (Sun, pulsars, …)

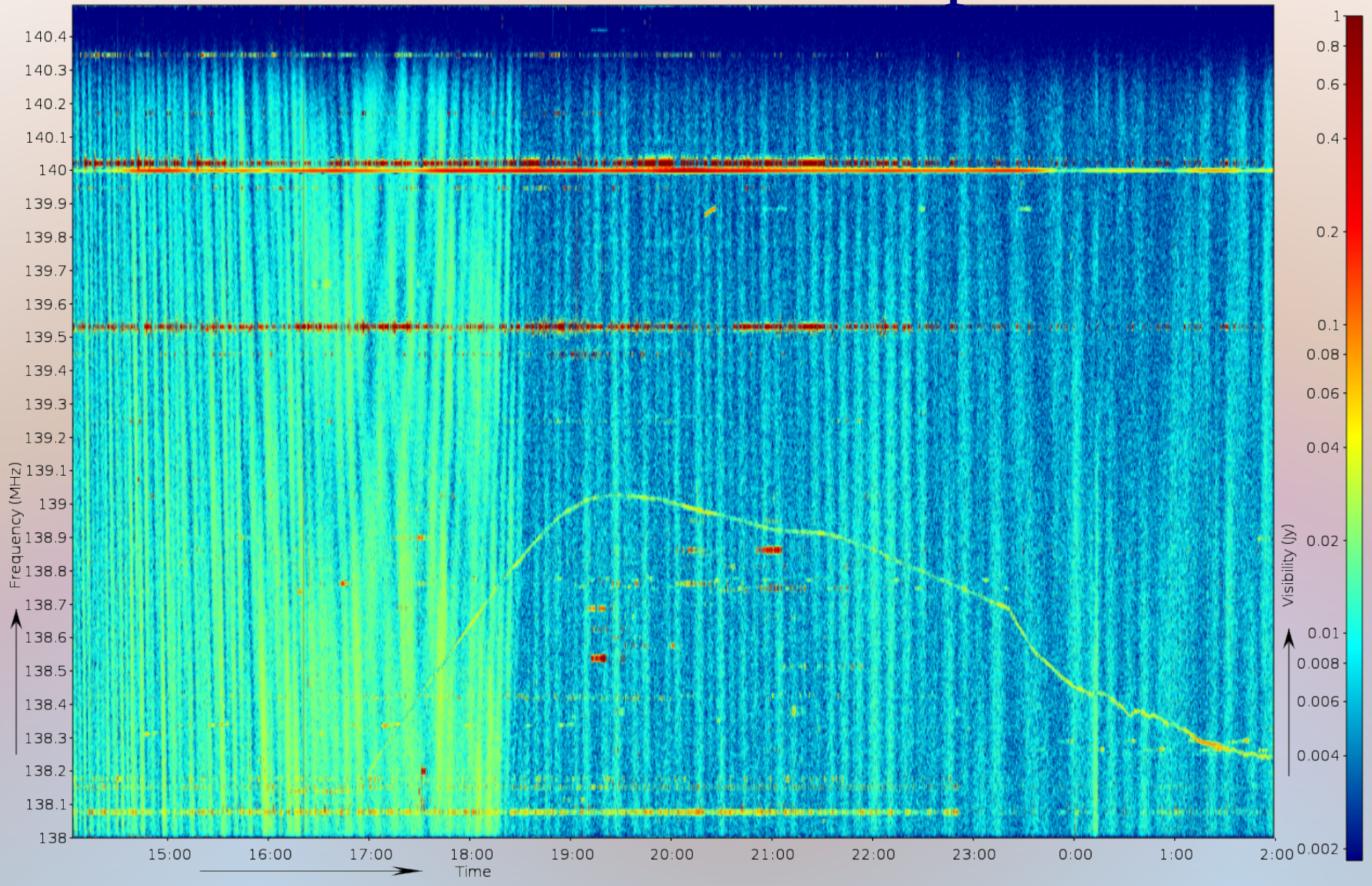
  - Quick fringes are line-like patterns

  - Spectral line observations

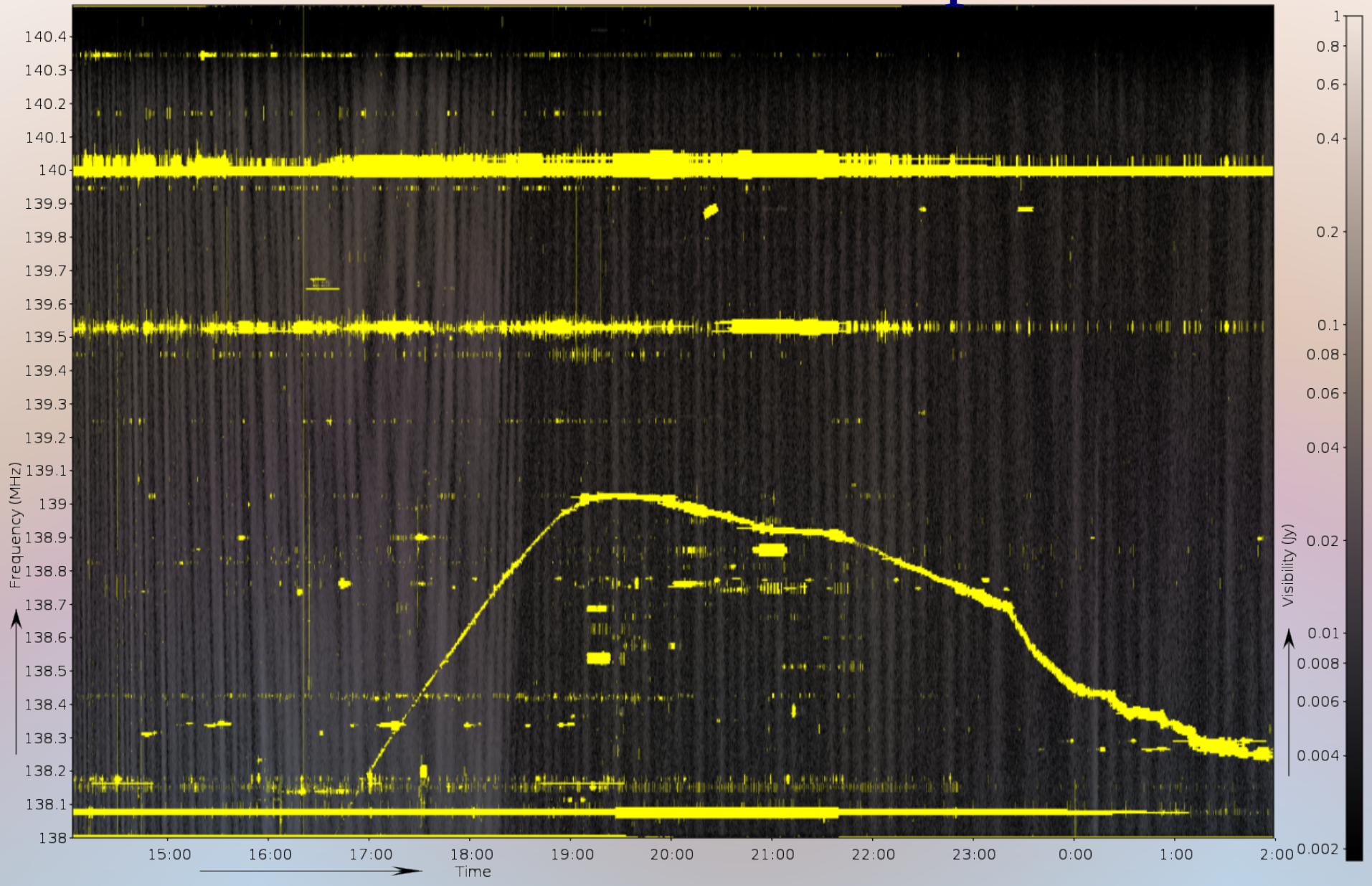
- Mostly not an issue – sources are ***mostly*** much weaker than RFI, and invisible in single correlations.



WSRT data example



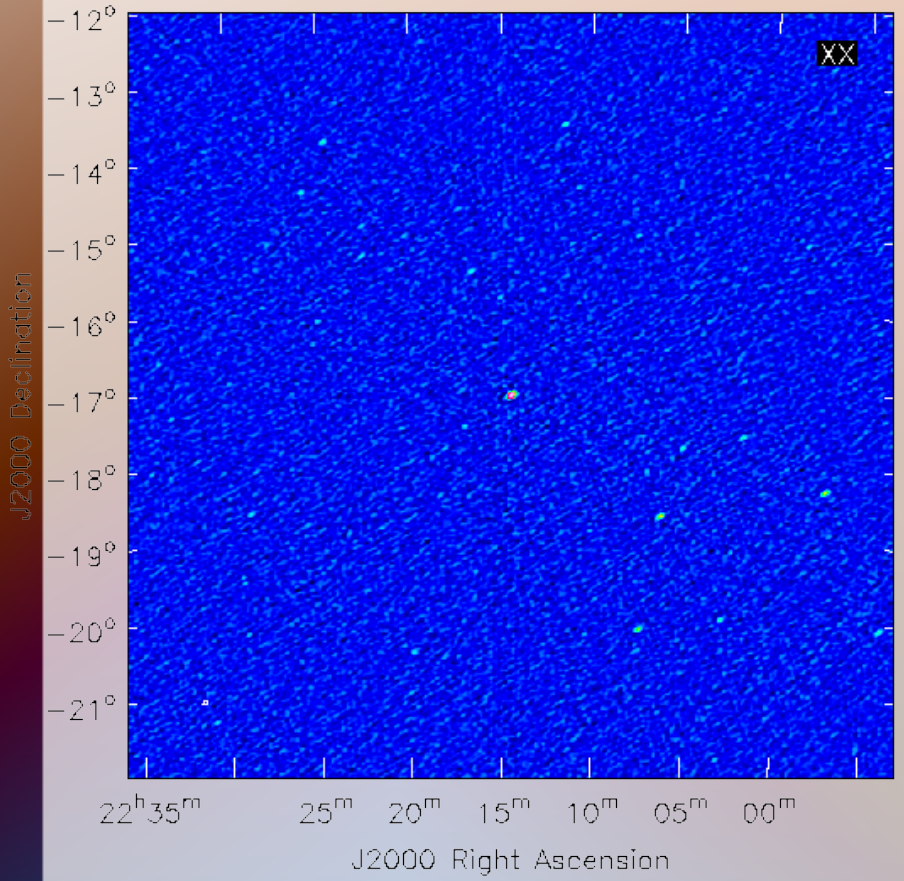
WSRT data example



Thresholding vs. AOFlagger

MWA 3 min observation with 32 tiles

Thresholding



AOFlagger

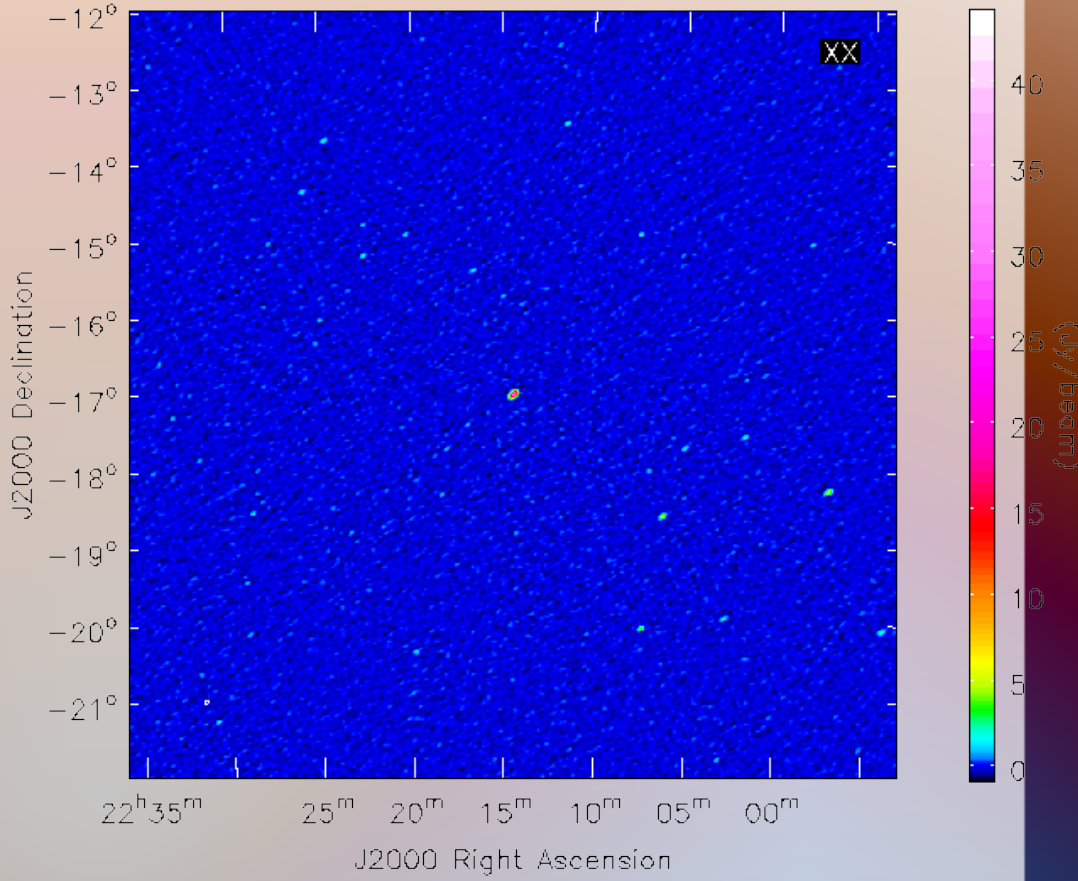


Image credit: Natasha Hurley-Walker (MWA data)

More about AOFlagger

- (Almost) same algorithm can be used for many telescopes
  - Software has been successfully used for: LOFAR (Offringa et al 2012), MWA (Offringa et al. 2015), WSRT, JVLA, GMRT, ATCA, Parkes, Arecibo, and BIGHORNS
- For Miriad users:
  Miriad has an implementation of AOFlagger
- SumThreshold algorithm available in E-Merlin "SERPent" pipeline (Peck & Fenech, 2013)

RFI excision for LOFAR

- LOFAR's case:
  - Fully automated detection, only a few % lost data
  - Only small residuals, do not affect image quality
- Why such good results?
  - LOFAR has very high time/freq resolutions
  - Design has *(mostly)* accounted for interference
  - High accuracy of algorithms
- Some transmitters do remain problematic (e.g., DAB, FM, wind turbines)
- Tweaking still required for special cases

Analysing RFI

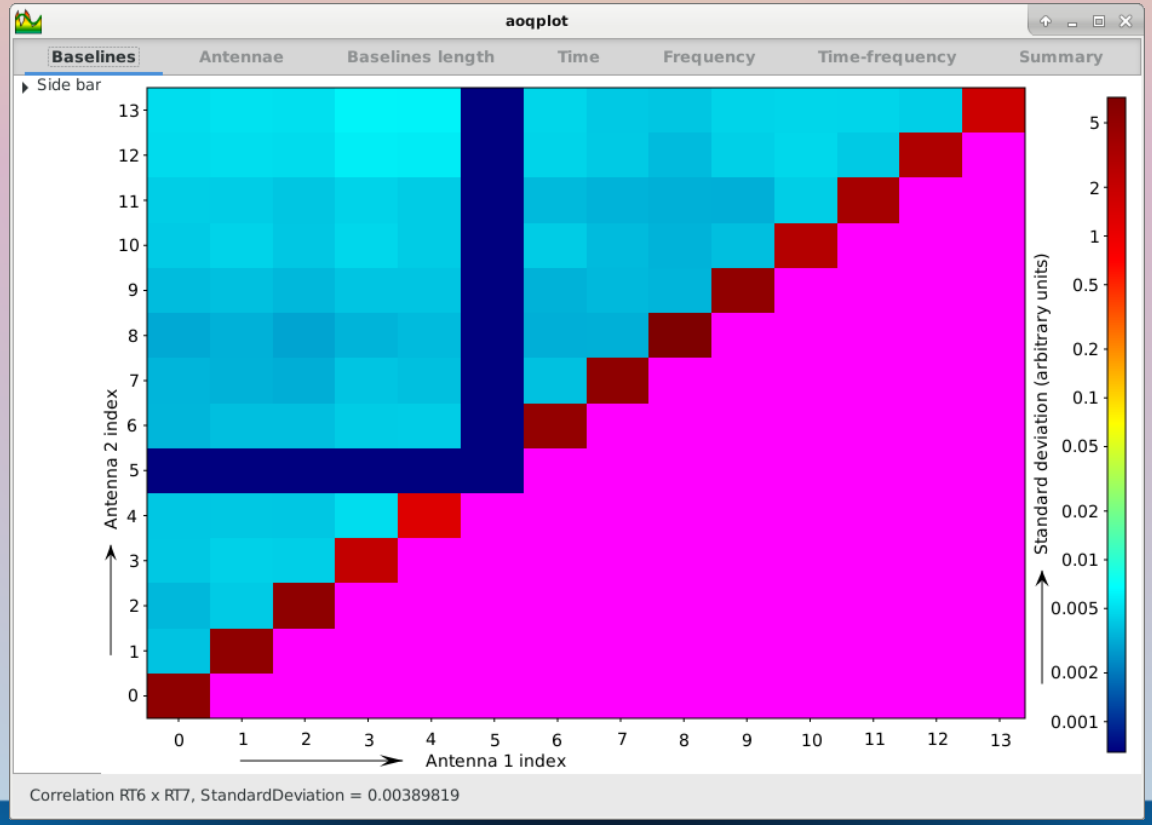
(Demo: open `rfigui` in other window)

- Open set, goto RT1 x RT2.
- Execute strategy
- Edit strategy: change flagged polarizations, change sumthreshold sensitivity
- Save strategy
- Execute '`aoflagger`' on cmdline.

Further analyses

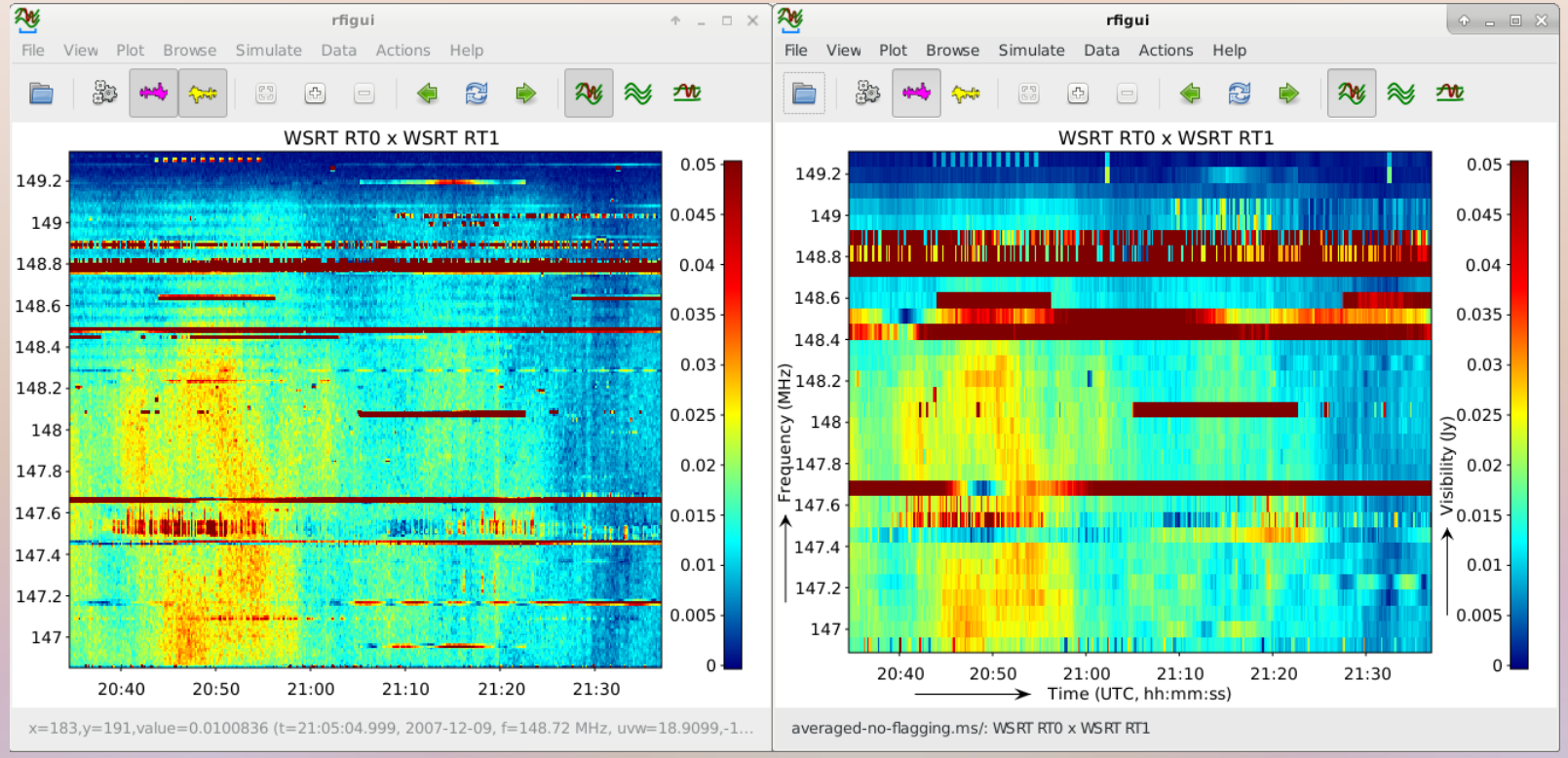
(Demo: open `aoqplot` in other window)

- casaplotms is slow for very big files
- aoqplot can give a quick overview



- Always flag (first) at highest possible resolution:

Highest resolution: Averaged without RFI detection:



- Always flag “incrementally”: don't reset flags! (don’t do `taql update obs.ms set FLAG=false`). Correlator might have set flags. These will be lost. To undo flagging, use backup.

Averaging & smearing

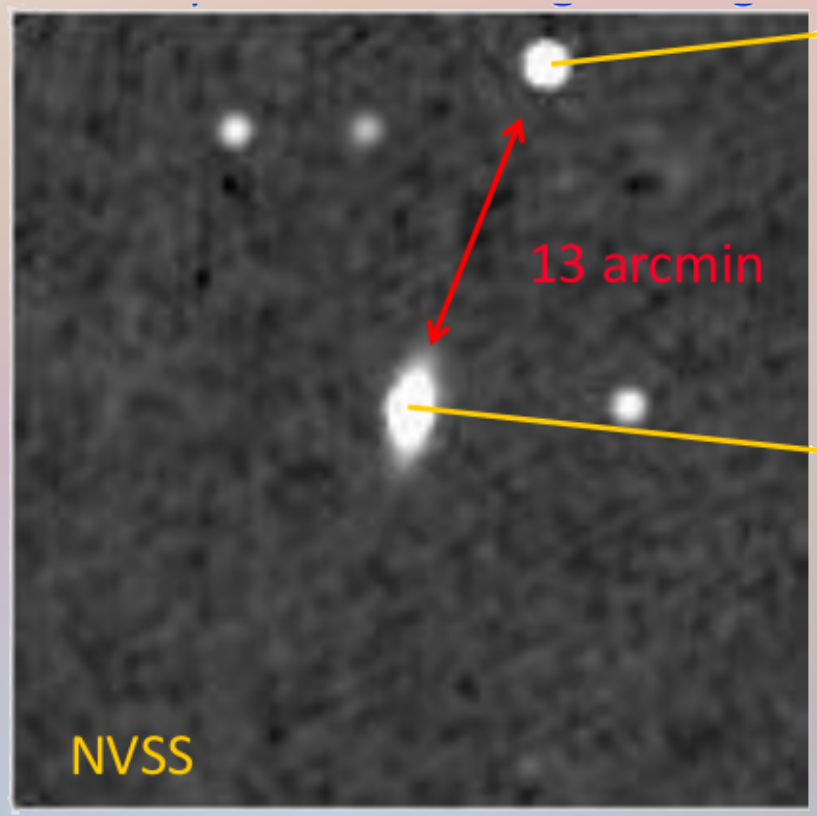
- Data size can be reduced by averaging data in time and/or frequency direction
- Only average *after* RFI detection
- Over-averaging causes *smearing*
  - *Time-smearing*: in tangential direction
  - *Frequency-smearing*: in radial direction
- Calibration might also constrain averaging factor
  → Next talk by George Heald

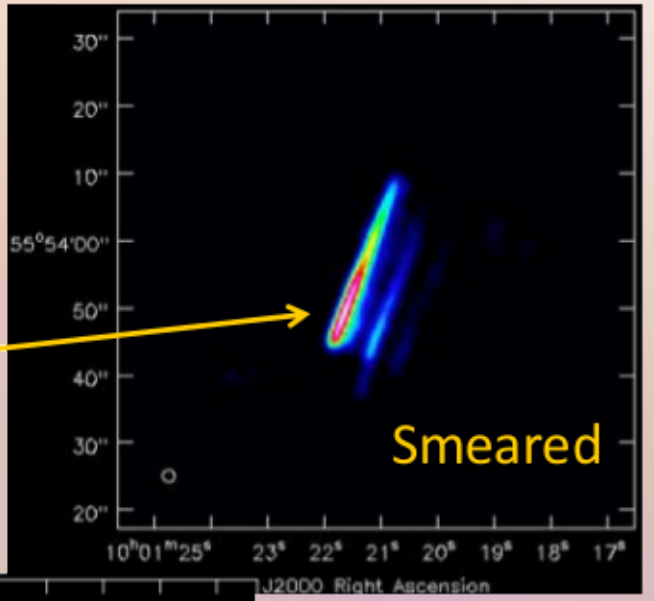
Bandwidth smearing

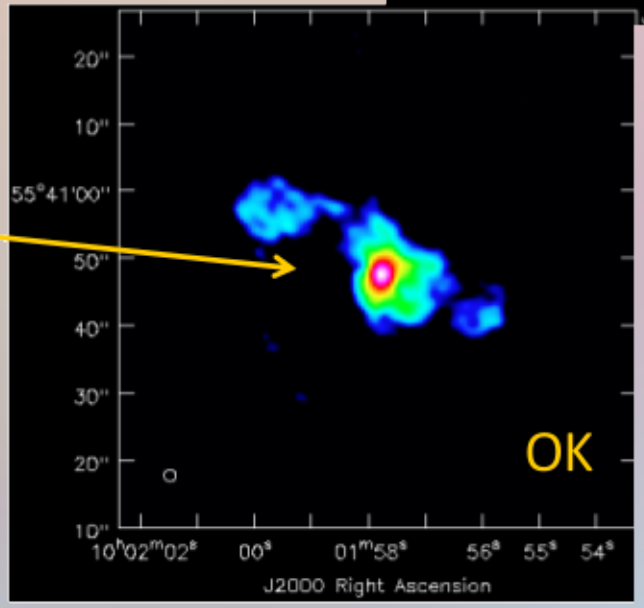
Off-axis sources fringe faster

(→ See previous lectures)

Smearing is proportional to distance from phase centre

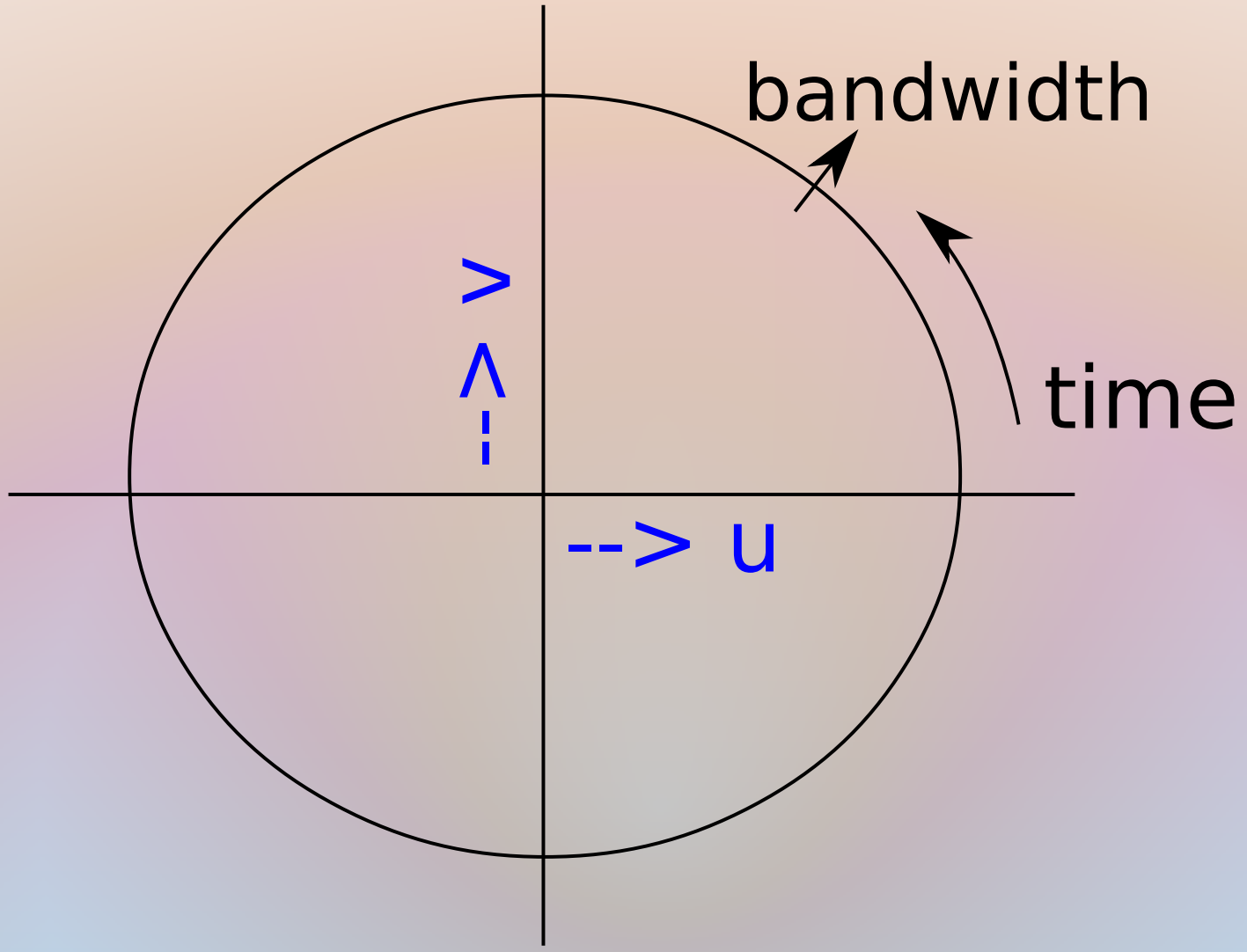






(slide by Tom Muxlow, who might have gotten it from someone else)

Averaging in uv-space



Smearing

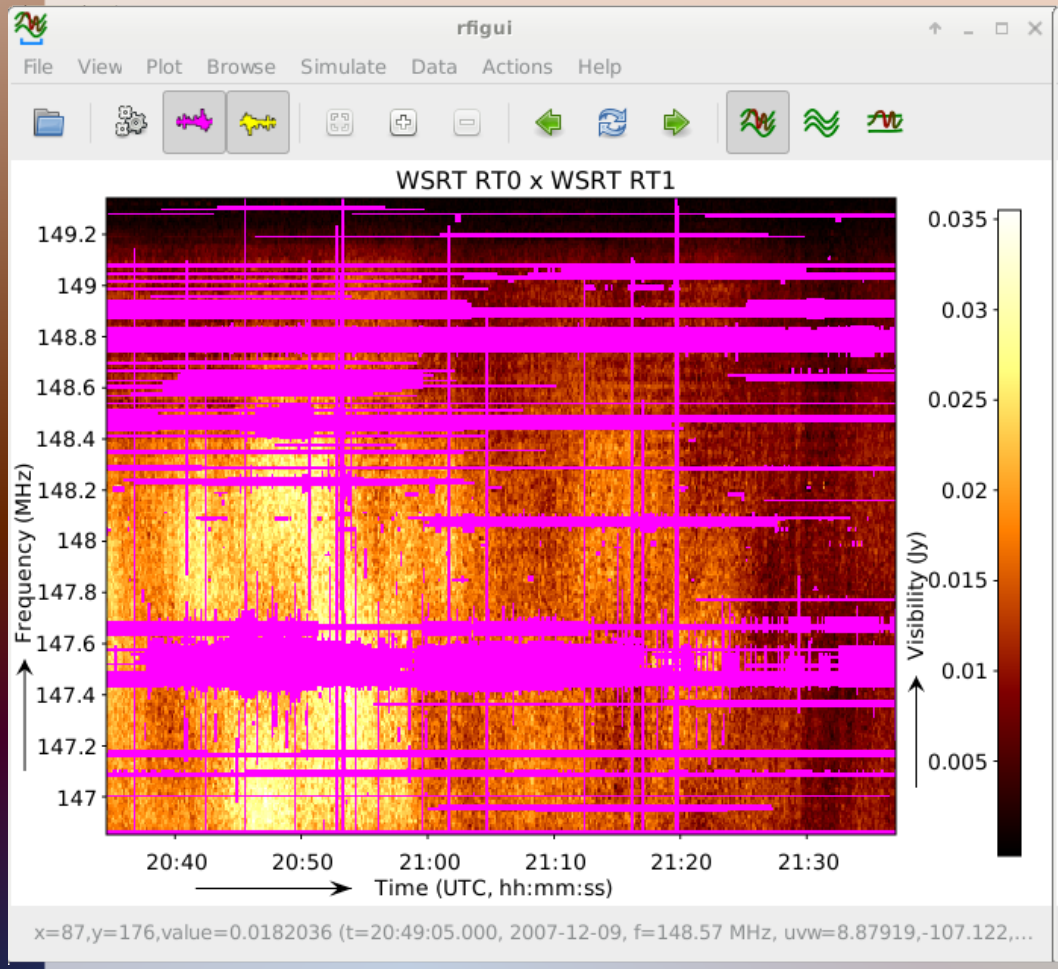
- General rule: phase turn along time / frequency should be sampled << 1/4th of a turn.
- Example with 1" resolution (e.g. LOFAR international baseline) and 1 deg off-axis source:
  - Source is 3600 resolution elements away
  - Phase turns ~3600 times in 6 hours (or over observing frequency)
  - Need ~14000 samples in 6 hours
  - Time res Δt < ~2 s ($\Delta \nu$ < ~10 kHz @ 150 MHz).

Data averaging with CASA

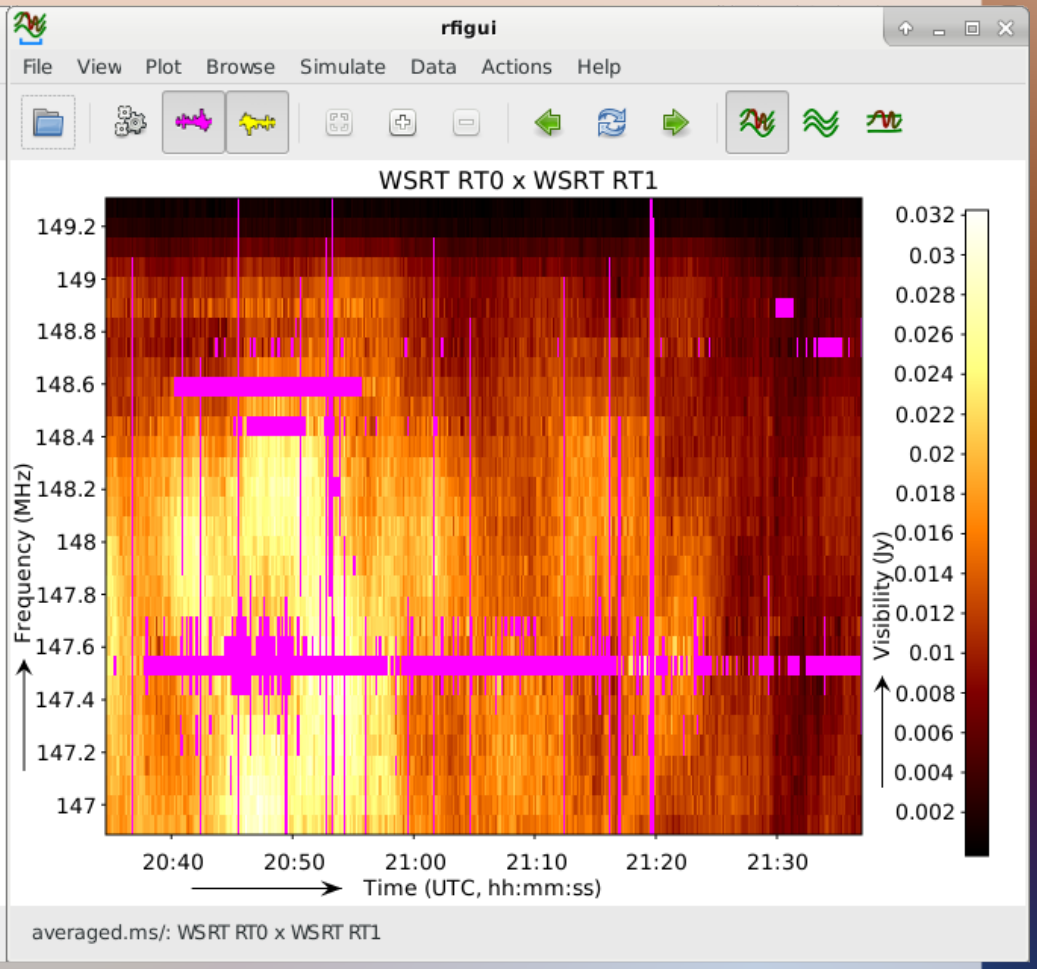
- (Demo: casa split)
- Example: (from casapy shell)

```
inp split
vis='3C196_spw5_sub1.MS' (input)
outputvis='averaged.MS'
width=8                  (Average over 8 channels)
timebin='60s'            (Average over 60 s)
go
```

Original resolution:



After averaging:



Averaging DATA

- Processing data can be very time expensive, but almost all steps scale linear with nr. of visibilities.

- Work on averaged data (and/or subset) while experimenting with settings

```
anoko@DOP348:~/ERIS2015$ du 3C196_spw5_sub1.MS/ -sh
998M    3C196_spw5_sub1.MS/
anoko@DOP348:~/ERIS2015$ du averaged.ms/ -sh
45M     averaged.ms/
anoko@DOP348:~/ERIS2015$
```

NDPPP: Averaging LOFAR data

- Almost all telescopes have existing sets of scripts to do preprocessing… Use them!
- '`split`' task does not work well on LOFAR data (see LOFAR cookbook for details)
- Instead, a specialized LOFAR pipeline was made to perform several steps at once:
  DPPP: the “Default Pre-Processing Pipeline”
- Can run aoflagger and perform averaging at once (as well as several other things)
- See LOFAR Cookbook for detailed info
- (MWA has a similar pipeline called '`cotter`').

Averaging DATA

- Processing data can be very expensive, but almost all steps scale linear with nr. of visibilities.
- While experimenting with settings, it can be very beneficial to work on averaged data.

Summary

- First step in data processing is
  data inspection
- Second step is data flagging
  ...or isn't it?

Summary

- First step in data processing is
  data inspection
- ~~Second step is data flagging~~
- Second step is BACKUP YOUR DATA
- Third step is data flagging and RFI detection
- Calibration, imaging, … to be discussed!

Summary

- I've shown:
  - Data inspection (with e.g. CASA casaplotms, rfigui and aoqplot)
  - Flagging data manually (with taql)
  - Automated RFI detection (with the AOFlagger)
  - Data averaging (with CASA split or NDPPP)
  - Issues with insufficient resolution (smearing, bad RFI detection)
- Good luck!