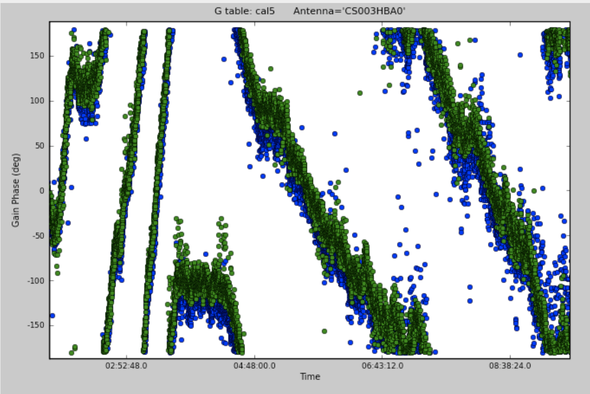
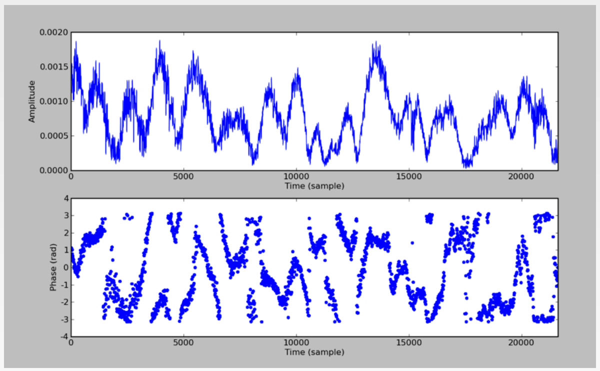
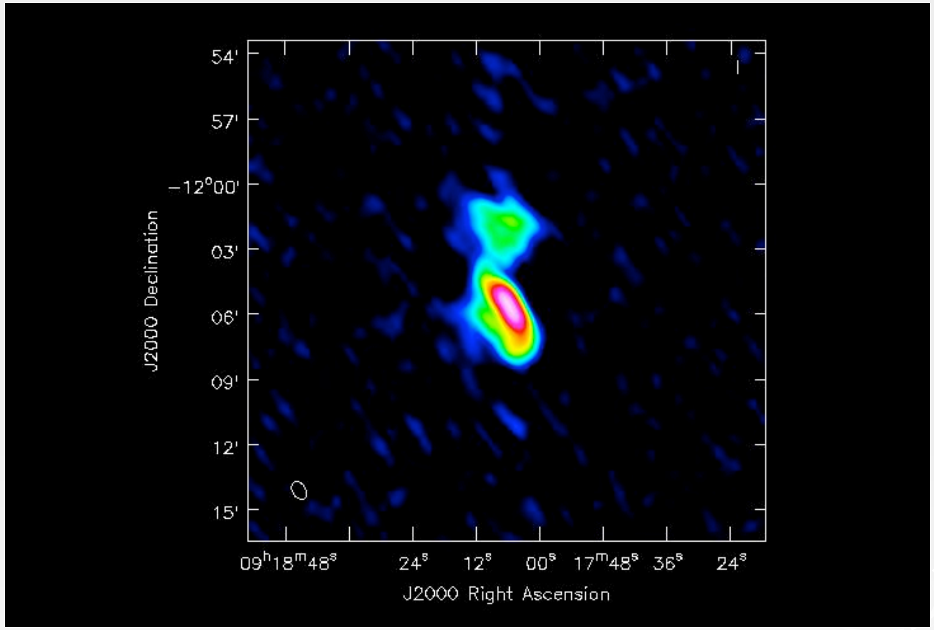
Introduction to LOFAR data reduction





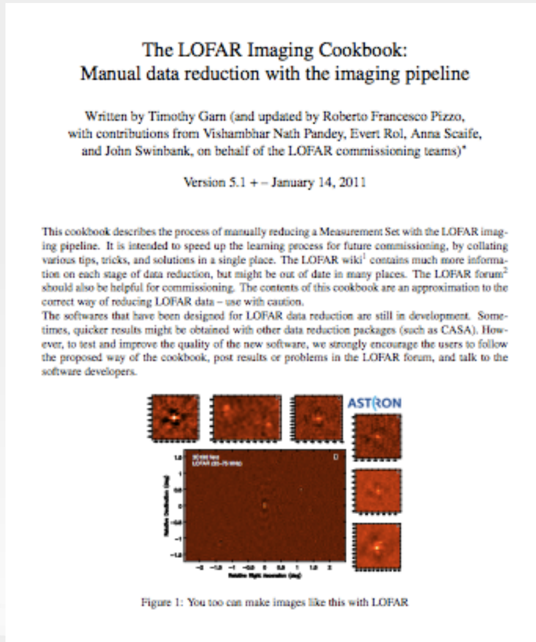


R. J. van Weeren
Leiden Observatory

Overview of processing

1. flagging (RFI console, NDPPP), talk David
2. compression (NDPPP)
3. LBA: vdtol demixing, talk Bas
4. BBS (calibration), talk Joris
5. imaging

See the cookbook v7.0



The LOFAR Imaging Cookbook:
Manual data reduction with the imaging pipeline

Written by Timothy Garn (and updated by Roberto Francesco Pizzo, with contributions from Vishambhar Nath Pandey, Evert Rol, Anna Scaife, and John Swinbank, on behalf of the LOFAR commissioning teams)*

Version 5.1 + – January 14, 2011

This cookbook describes the process of manually reducing a Measurement Set with the LOFAR imaging pipeline. It is intended to speed up the learning process for future commissioning, by collating various tips, tricks, and solutions in a single place. The LOFAR wiki[1] contains much more information on each stage of data reduction, but might be out of date in many places. The LOFAR forum[2] should also be helpful for commissioning. The contents of this cookbook are an approximation to the correct way of reducing LOFAR data – use with caution.

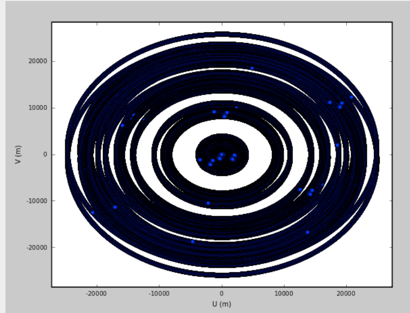
The softwares that have been designed for LOFAR data reduction are still in development. Sometimes, quicker results might be obtained with other data reduction packages (such as CASA). However, to test and improve the quality of the new software, we strongly encourage the users to follow the proposed way of the cookbook, post results or problems in the LOFAR forum, and talk to the software developers.

ASTRON

Figure 1: You too can make images like this with LOFAR

Before starting....

- check a few long/short baselines (plotxy, plotms), one baseline at the time
  - get an idea of the A-team effects
  - use BBS to "predict"
  - All LBA data needs demixing for good results



- What sources in the field center (FWHM of station beam)?
- How far away are the A-team sources?

```
~weeren/scripts/plot_Ateam_elevation.py your.ms
```

- check uv-coverage, what are the longest baselines?
- `makebeamtables ms=your.ms antennaset=LBA_OUTER` (data taken before April 1, 2011)
- check every step !! (especially look at the calibration solutions)

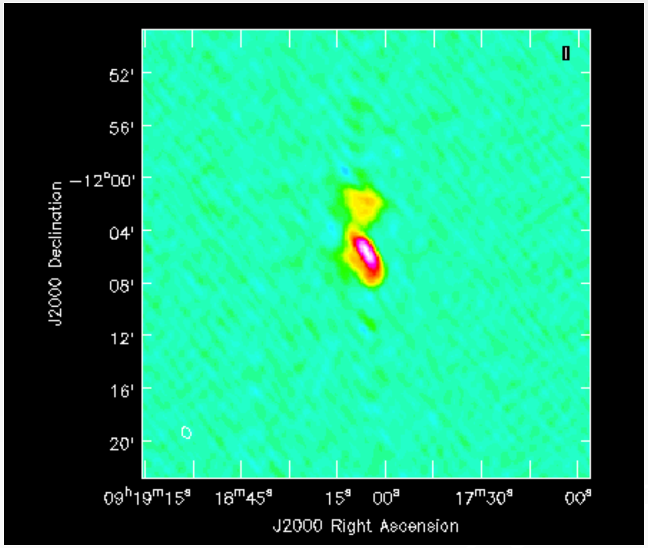
RFIconsole & NDPPP

- dataset passes through RFIconsole & NDPPP
- removes most of the RFI (broadband RFI is a problem, e.g., A2255 HBA data)
- averaging strategy often used:
  - from 256 (or 64) chan to 1 chan of 0.18 MHz
  - time averaging also (i.e., from 1 sec to 5 sec)
- bandwidth/time smearing issues
- demixing for LBA needed for good results

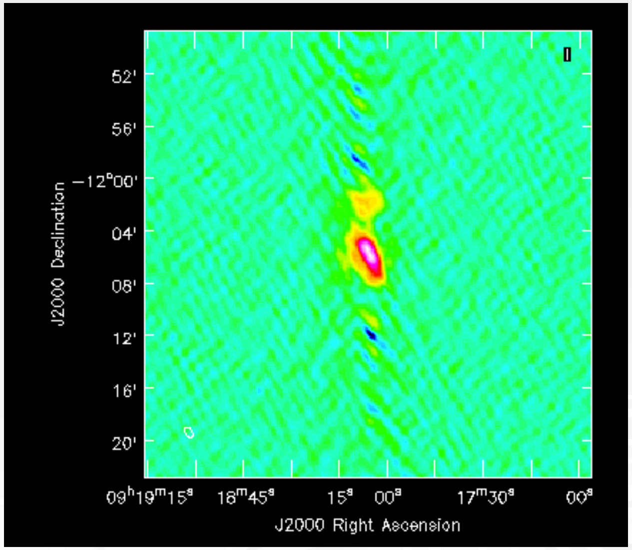
Calibration models

- Get a good skymodel: CC components, WENSS, VLSS, pyBDSM
- casapy2bbs.py (on casa .model images)

- Be careful: resolution of model vs. resolution longest LOFAR baseline
- CC models show best performance, but slow....
- Use patches in model or not?
- Double check model!!



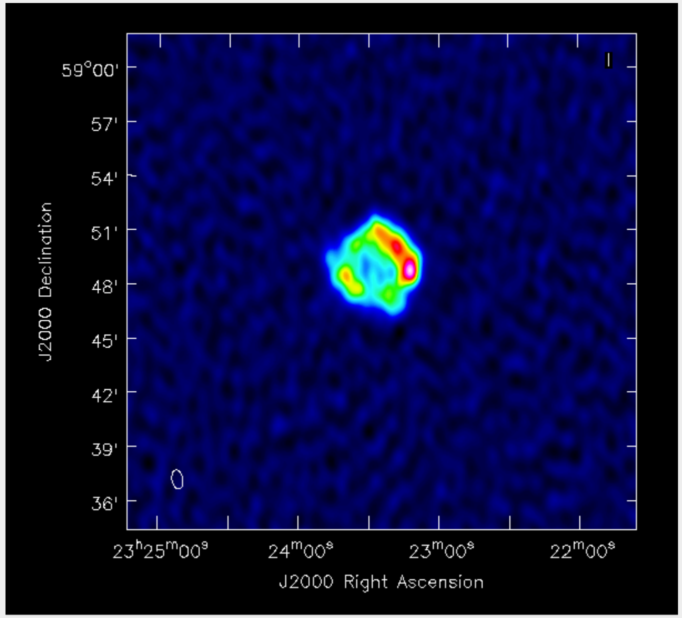
CC model



pyBDSM model from Niruj

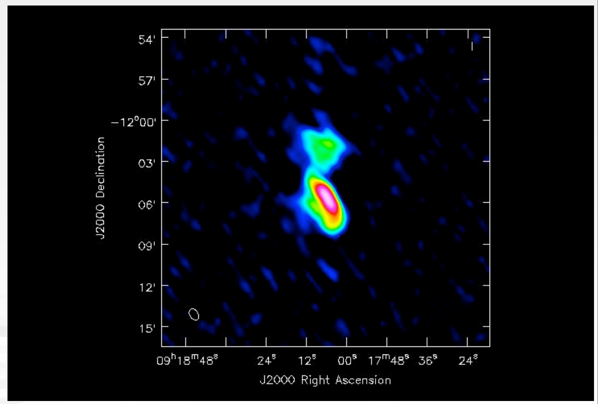
LBA data: demixing

- average down to 10 sec and 1 channel
- Cas A, Cyg A, Vir A, Tau A
- need raw (but flagged) data, takes time, needs lot of disk space (> 100 GB typically)
- see talk Bas van der Tol
- solve for Gain:0:0, 0:1, 1:0, 1:1 for best result (slower)



50 MHz 1 SB Cas A, 127 degr (!!) from pointing center (Hydra A) with demixing

- no demixing: with some effort you may get something...(playing with uvranges helps)
- HBA: if A-team source is nearby your target source



Hydra A, after demixing

-12 declination (!)

BBS

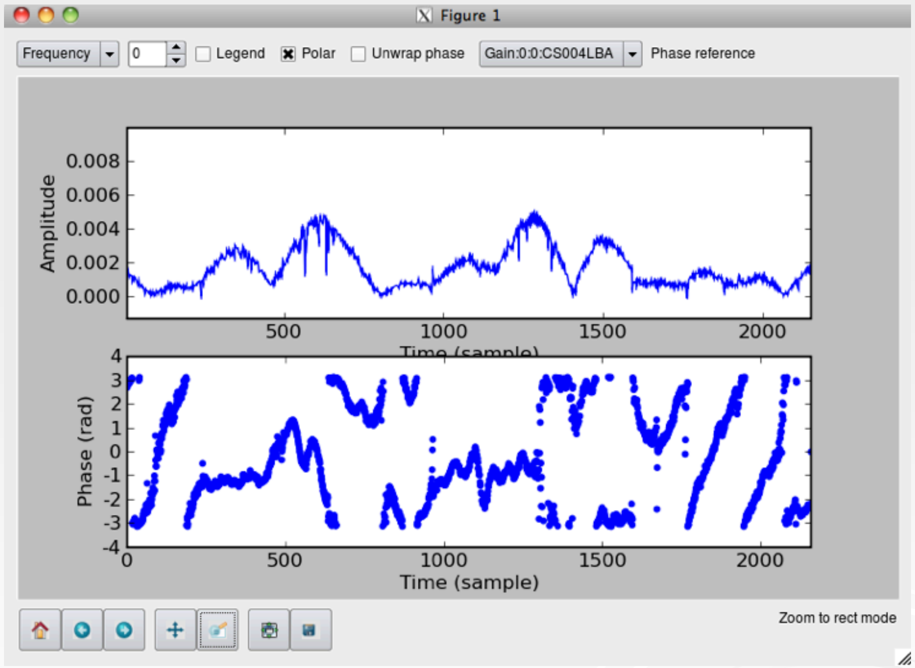
- monitor progress with (`Strategy.ChunkSize = 100`)
  `grep -i 'Reading chunk' ~/kernel_control.log| wc -l`
- ~~casa/aips/difmap selfcal~~ use BBS
- enable the station beam for `solving' `Beam.Enable = T`
- always(!!) check the solutions with parmdbplot.py (do they make sense?)
- some options for the "correct" step:

  - `Beam.Enable=T, for casa clean(), need to pick a direction`
  - `Beam.Enable=F, for the "vdtol" imager`
  - `Step.correct.Output.WriteCovariance = T for the "vdtol" imager`

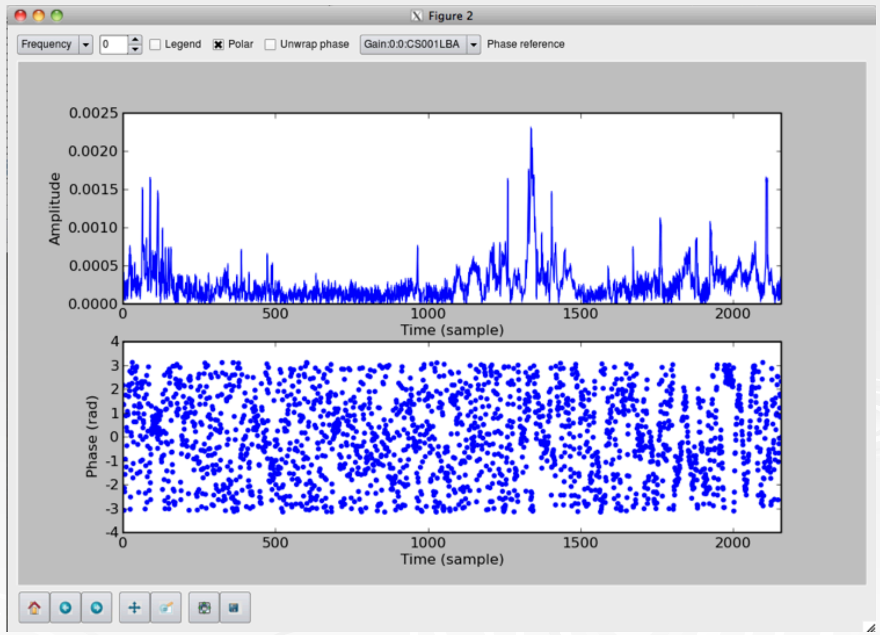
parmdbplot.py

- plot phases against a reference station before deciding if they make sense
- “randomness” in solutions is bad (especially for core stations)
- when starting with a low-resolution model (e.g., from WENSS) one should expect noisy solutions for remote stations (update model with selfcal, or better model)

OK



hmmm...



note: `beam.enable = F` & off-axis source

BBS advanced

L1, L2, L1R, L2R solve

- `Step.solve.Solve.Algorithm = L1  # or L2 (default L2)`
- `Step.solve.Solve.EpsilonL1 = [1e-4, 1e-5, 1e-6]`
- `Step.solve.Solve.OutlierRejection.Enable = T #(T then L1R or L2R)`
- `Step.solve.Solve.OutlierRejection.Threshold = [7.0, 5.0, 4.0, 3.5, 3.0, 2.8, 2.6, 2.4, 2.2, 2.5]`

global solve

- example 8 bands global solve
- global solve only with Algorithm *L2*

```
Strategy.UseSolver = T
Step.solve.Solve.CalibrationGroups = [4,4] or
Step.solve.Solve.CalibrationGroups = [8] or
Step.solve.Solve.CalibrationGroups = [1,2,3,2]
```

(note: 4+4=8, 1+2+3+2=8)

phase/amplitude only (phase only example, non-dir gain)

- `Step.solve.Model.Phasors.Enable = T`
- `Solve.Parms = ["Gain:0:0:Phase:*", "Gain:1:1:Phase:*"]`
- `Solve.Mode = PHASE`
- WARNING! : `Solve.Mode = COMPLEX` needed for DirectionalGain or when solving for off-diagonal terms in Jones matrix, i.e., `Solve.Parms = ["Gain:0:0:Phase:*", "Gain:1:1:Phase:*", "Gain:0:1:Phase:*", "Gain:1:0:Phase:*"]`

after BBS: flagging

Solutions looks reasonable (for core stations) :

(if not, go back to BBS)

- manually remove high data points, casapy flagdata(),
  - be careful (i.e., extended source)

- solution outliers: “solflag” (see cookbook)

3C196 LBA example (no demixing, averaged data)

- 1. Cas A, at 60 deg (Cyg A > 90 deg)
- 2. simulate Cas A, Cyg A with BBS: conclude Cas A is the “problem”
- 3. skymodel contains Cas A & 3C196 (point)
  - Cas A dominates short baselines
  - 3C196 dominates long baselines > 2km
- use directional gain and solve 3C196 + Cas A, subtract Cas A

3C196 LBA example (no demixing, averaged data)

- 1. Cas A, at 60 deg (Cyg A > 90 deg)
- 2. simulate Cas A, Cyg A with BBS: conclude Cas A is the “problem”
- 3. skymodel contains Cas A & 3C196 (point)
  - Cas A dominates short baselines
  - 3C196 dominates long baselines > 2km
- use directional gain and solve 3C196 + Cas A, subtract Cas A

  - alternative approach:
    - skymodel only contains 3C196, but now limit uvrange > 2 km when solving
      - non-directional gain
    - first approach gives better results (can use most of the short baselines also)

3C196 (with demixing)

- demix, see talk Bas
  (data can now be safely averaged because Ateam sources are removed)
- solve for 3C196 only on highly averaged (10 sec, 1 channel)
- that's it...

BBS: warnings

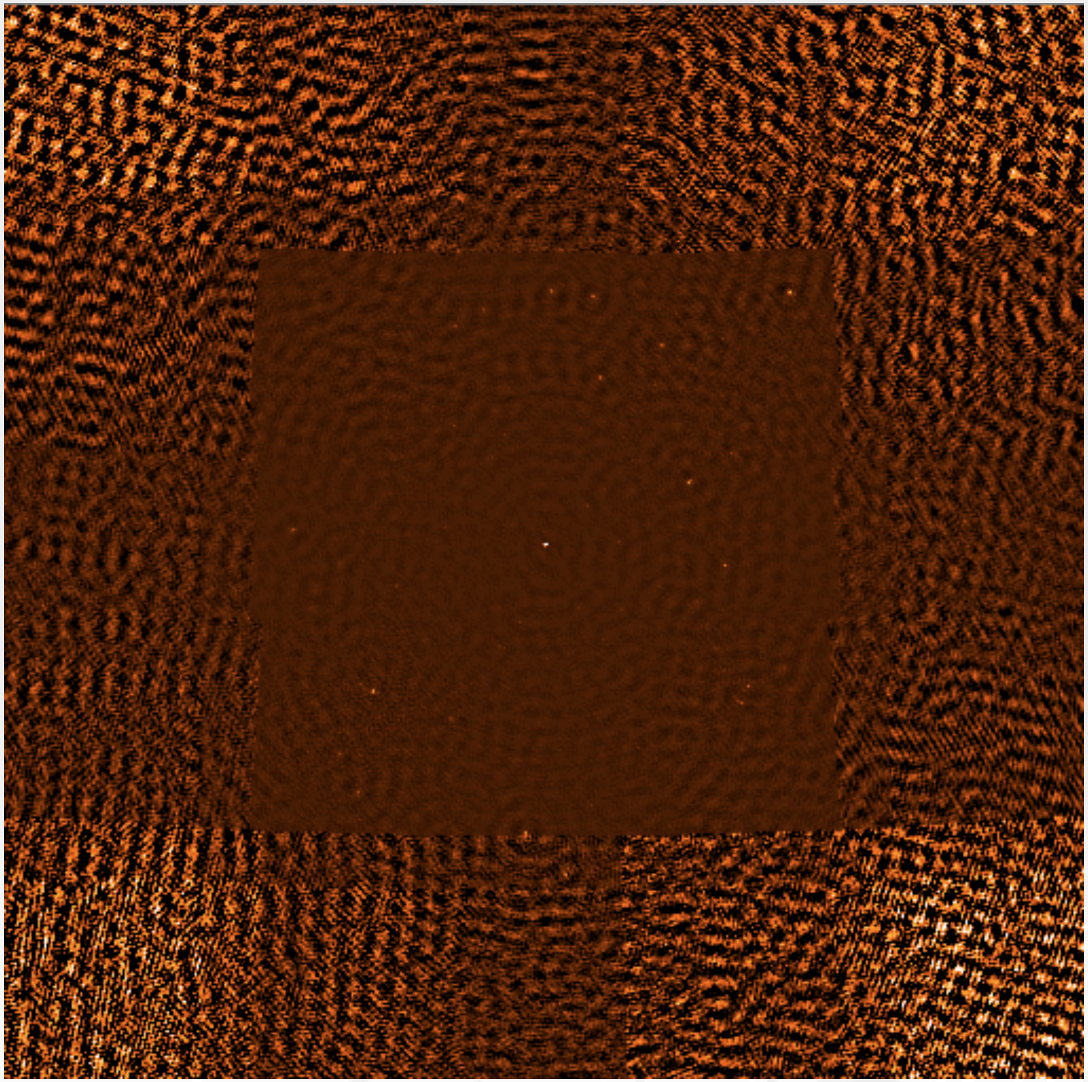
- incomplete skymodels:
  - you cannot solve for Cyg A only if the Cas A effects are of the same order as Cas A)
- be careful about the “patch” specification in the skymodel
  - patch has to be “tiny” as beam is computed only once for the center of the patch
  - will be fixed (Joris is working on it)
- note: A-team solutions will be rubbish for remote stations if there is bandwidth smearing (for 1 channel avg. data)
  - affects quality of selfcal solutions for other stations
  - demixing solves this problem

Imaging: casapy clean()

- imagermode='csclean' (stabilizes clean)
  - limiting uvranges: can be helpful
  - use wprojection with enough planes
- weighting, briggs, uniform, natural
- pixelsize (< beamsize/4)

- Remember: the casapy imager cannot make flux corrected images over widefields
  - widefield selfcal will probably not work
  - expect artifacts caused by "wrong" deconvolution
  - BBS can only correct the data in one single direction
- vdtol imager can make flux corrected images but is "difficult" to use, untested for HBA data, and slow

“vdtol” imager



55 MHz, 1 SB

Conclusions

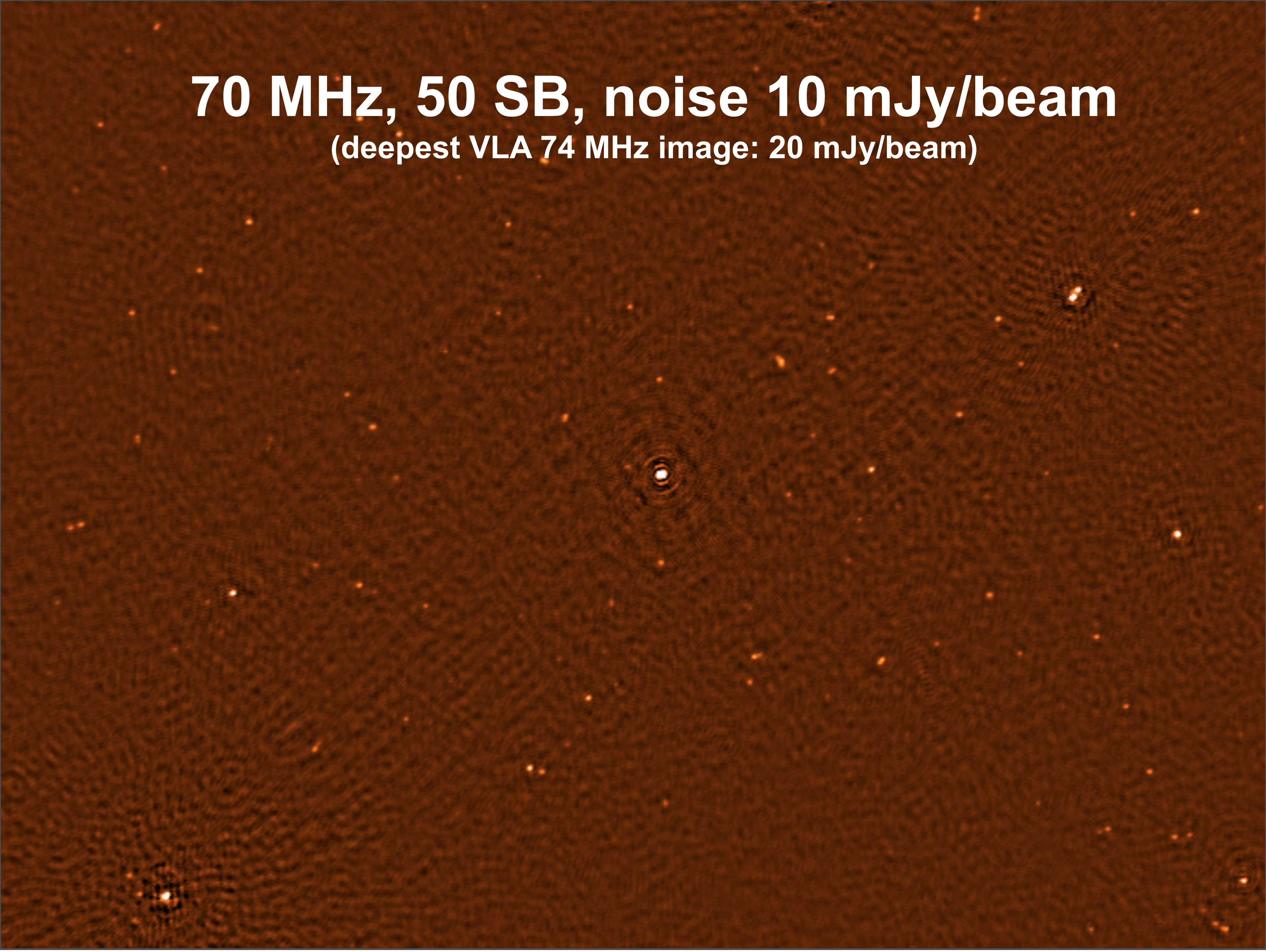
- LBA data: demixing for best results
- good models do half of the work
- check solutions in parmdbplotxy
- *No "LOFAR data reduction for dummies" (yet)*

Bright dominating central source

- take "high-res" CC model if available
  - ~weeren/scripts/Ateam_LBA_CC.skymodel2
- calibrate with bbs
- parmdbplot.py + flagdata()
- enough SNR for calibration on single SB
- selfcal with casapy, casapy2bbs.py, bbs
- casapy clean: use clean boxes

deep fields, faint central source

- take image from WENSS, VLSS
  - FOV must be large enough
  - for example skyview.gsfc.nasa.gov
- create model with pyBDSM
- calibrate with bbs
- parmdbplot.py + flagdata()
- image with casapy
- selfcal will likely not improve you image
- probably not enough SNR for good calibration on single SB, may need global solve
- expect problem with calibration remote stations


70 MHz, 50 SB, noise 10 mJy/beam
(deepest VLA 74 MHz image: 20 mJy/beam)