

LOFAR Imaging tutorial using WSClean

André Offringa

Tutorial goals

- Get an understanding of basic imaging with WSClean
- For “production” science goals, the data is often run through automated imaging pipelines like Pilot, PiLL, Rapthor, TRAP, etc.
  - They do the imaging for you
  - Useful to understand what these do
  - A specific (favourite) target source is often extracted and reimaged with WSClean with tweaked settings

- We start from calibrated data

  Everyone should have 4 sub-bands (or use your own data) from: `LDS2026 / tutorials / imaging`
  (untar using e.g. tar -xjvf observation-band1.ms.tar.bz2)
- Topics:
  - Making a quick image
  - Cleaning
  - Multi-scale
  - LOFAR primary beam
  - Weighting, tapering
  - Wide bandwidth imaging
- Challenge:
  - Measure integrated flux of fuzzy source around 17h27m25s +58d36m06s as accurately as possible.

We are going to use WSClean. A lot of help on WSClean is available at the WSClean wiki:

https://wsclean.readthedocs.io/

WSClean is installed in the docker / singularity image.

Check which version you are running:

```
$ wsclean --version
```

Get the WSClean command line help by running wsclean without parameters: `$ wsclean | less`

Whenever you run WSClean in this tutorial, be sure to inspect the output.

A quick look at the data

A quick look is useful...

- ...to check if calibration went well
- ...to determine a reasonable size and scale for the image

A quick look at the data

Pick a random band and run wsclean as follows:

```
$ wsclean -size <width> <height> -scale <val>asec  \
          -name quick observation-band1.ms
```

(Change the sb number to your random sb number)

Replace **width** and **height** by a number of pixels.

val is the image resolution, here specified in asec.

Determine good values for these for imaging this LOFAR set:
- Try e.g. to image ~up to the first null to capture the entire field.
- Note that angular width ≈ **width** x **scale**.
- Try it out and run it again until you're happy with the image.

More notes:

- Note that **<val>** and "asec" have no space between them, e.g.: "-scale 2.5asec"
- Other units can be specified, e.g.: "6amin", "50masec", "0.1deg".
- In order to keep processing fast for the tutorial, don't make images > 4k or wider than 20 deg. This quick imaging should not take more than ~3 min.

A quick look at the data

Example command:

```
$ wsclean -size 3200 3200 -scale 10asec  \
      -name quick observation-band1.ms
```

This will output “quick-dirty.fits” and “quick-image.fits”.

Inspect these with your favourite fitsviewer
(e.g., kvis, ds9, casaviewer, carta).

For kvis:

```
$ kvis quick-*.fits
```

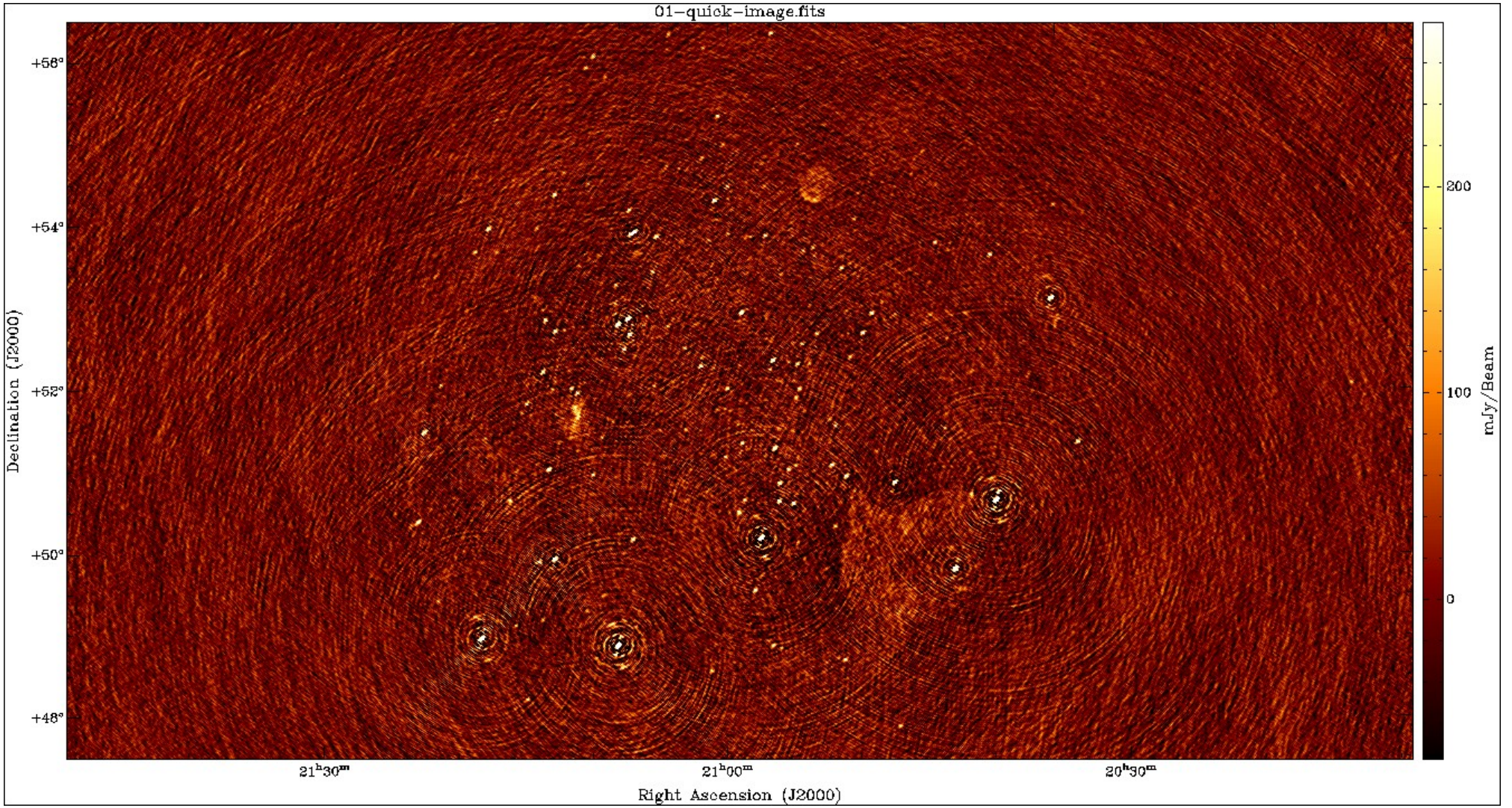
For ds9:

```
$ ds9 quick-*.fits
```

If you’re early, try imaging again with larger FOV or different scale

Can you see the first “null” of the LOFAR primary beam?

Quick imaging result



Cleaning

The main parameters for cleaning are:

-niter <count> Turns cleaning on and sets max iterations.
Normally, cleaning should end at the
the threshold, not at the max iterations.

-mgain <gain> How much flux of the peak is subtracted
before a major iteration is restarted.
Depends on how good the PSF is.
0.8 is safe, 0.9 almost always works and
is faster.

-auto-threshold <sigma>
Set the apparent flux at which to stop.
This is relative to the computed noise.
Should typically be 3-5.

Run the following command:

```
$ wsclean -size <width> <height> -scale <val>asec      \
    -niter <niter> -mgain 0.8 -auto-threshold <level> \
            -name clean observation-band<1>.ms
```

Cleaning

Example command:

```
$ wsclean -size 3200 3200 -scale 10asec          \
    -niter 50000 -mgain 0.8 -auto-threshold 5    \
          -name clean observation-band1.ms
```

- It is convenient to store the above command in a shell script.
- Inspect all output .fits images – can you explain what is what?
- Notice in the output the cleaning process:

Peak flux

Reached Threshold in ~9000 iters

Image noise

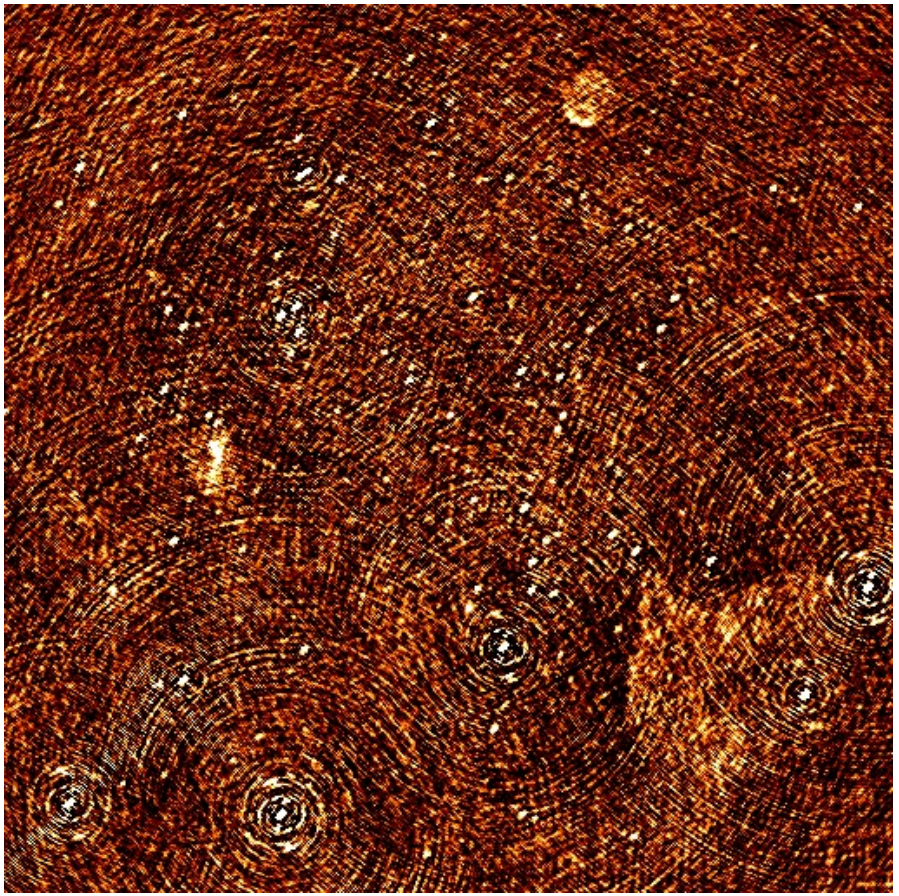
```
 == Cleaning (1) ==
Estimated standard deviation of background noise: 15.64 mJy
Initial peak: 2.39 Jy at 777,717
Next major iteration at: 477.06 mJy
Performed 398 iterations in total, 398 in this major iteration with Clark opt...
Stopped on peak 476.43 mJy, because the minor-loop threshold was reached...
[..]
 == Cleaning (2) ==
Estimated standard deviation of background noise: 12 mJy
Initial peak: -549.48 mJy at 392,632
Next major iteration at: 109.9 mJy
[..]
 == Cleaning (3) ==
Estimated standard deviation of background noise: 9.06 mJy
Initial peak: -164.94 mJy at 584,613
Major iteration threshold reached global threshold of 45.31 mJy: final major it...
Performed 9036 iterations in total, 5868 in this major iteration with Clark op...
Stopped on peak 45.31 mJy, because the threshold was reached.
```

Cleaning

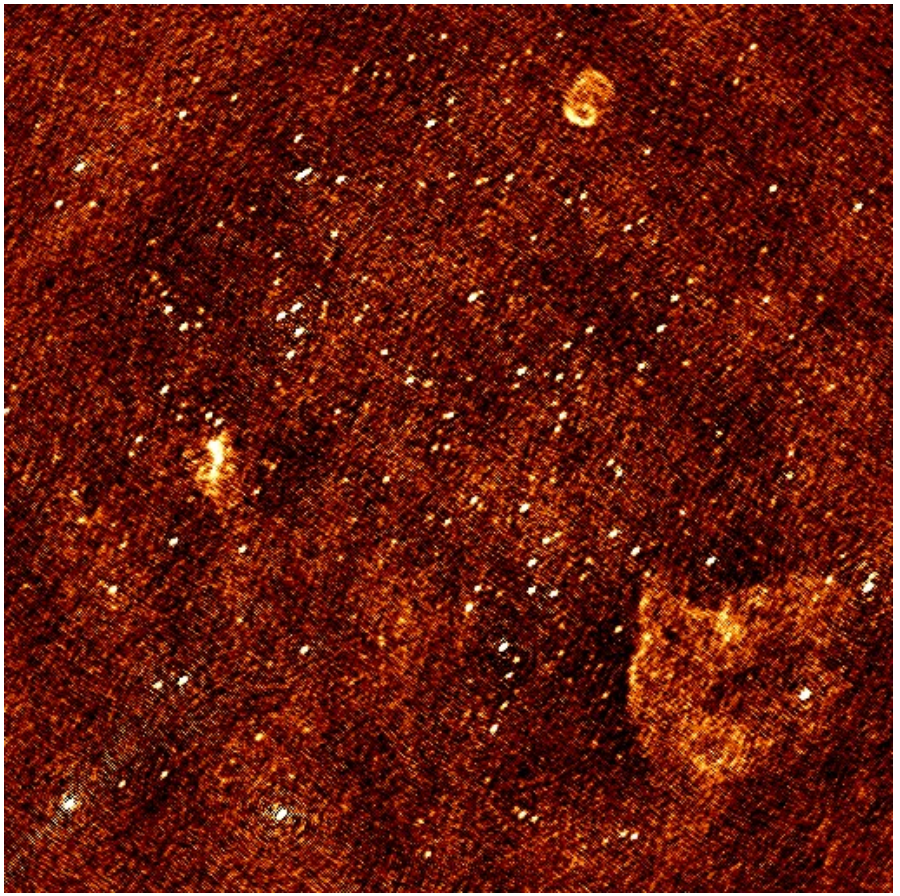
Example command:

```
$ wsclean -size 1200 1200 -scale 20asec          \
    -niter 50000 -mgain 0.8 -auto-threshold 5    \
              -name clean observation.ms
```

clean-dirty.fits



clean-image.fits



Apply LOFAR beam

The LOFAR beam is applied by adding
`-apply-primary-beam`

The Measurement Set contains metainformation about the state of the stations during the observation, which are used to calculate the primary beam.

Repeat the previous imaging with the beam, similar to:

```
$ wsclean -size <width> <height> -scale <val>asec      \
    -apply-primary-beam                                \
    -niter <niter> -mgain 0.8 -auto-threshold <level> \
              -name lofarbeam observation.ms
```

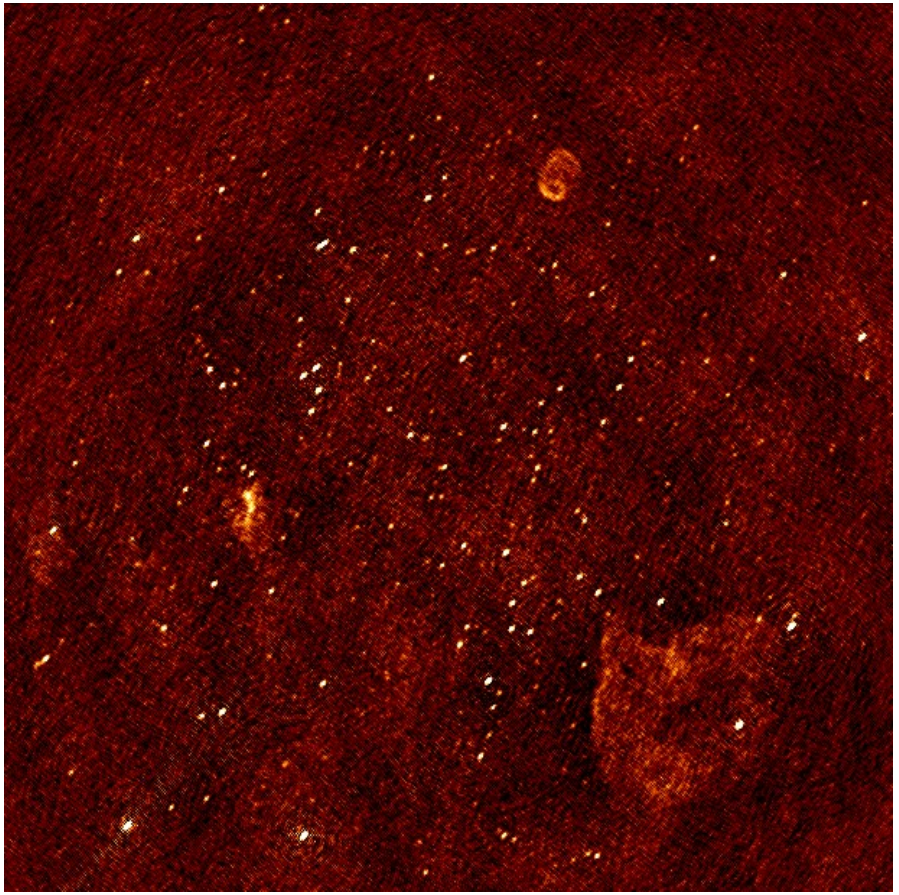
Inspect all the output images.

LOFAR primary beam correction

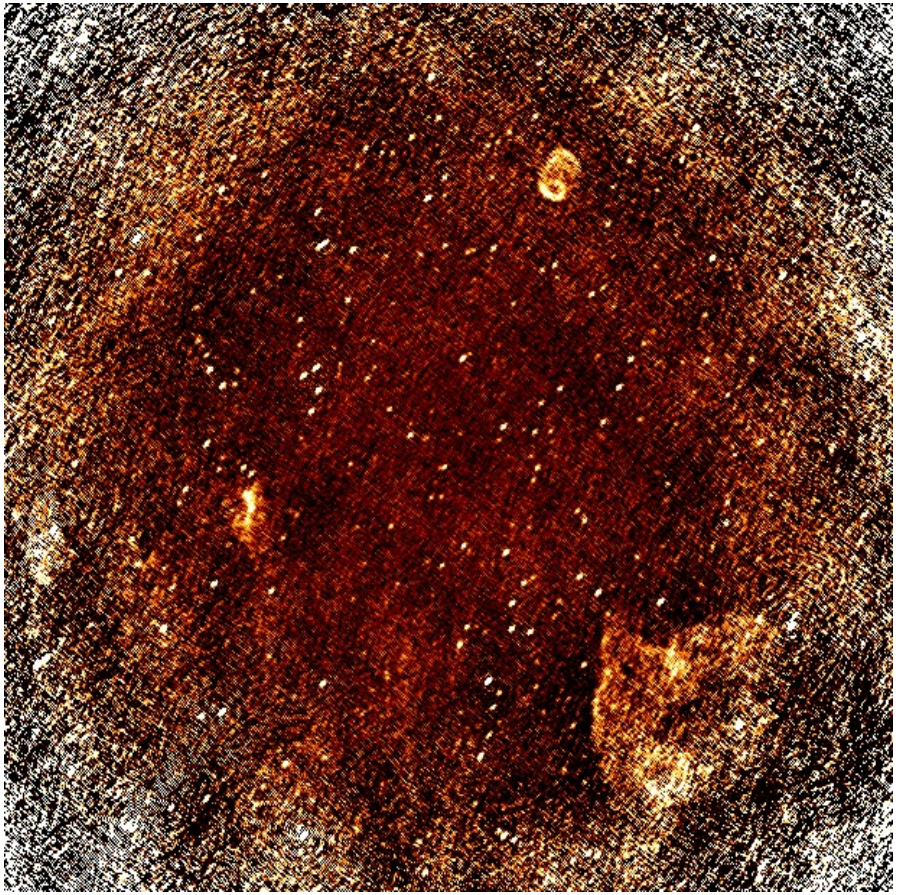
Example command:

```
$ wsclean -size 3200 3200 -scale 10asec                    \
    -apply-primary-beam -use-differential-lofar-beam \
    -niter 50000 -mgain 0.8 -auto-threshold 5          \
          -name lofarbeam observation-band3.ms
```

No beam applied:



Beam applied:



Weighting and tapers

Read the documentation for `-weight`, `-taper-gaussian` and optionally other weighting/tapering methods.

Repeat the previous imaging, but with settings for these parameters that are useful to:
- accentuate the diffuse emission; and
- to make the beam Gaussian like, to measure the flux of the emission more easily.
- If you have more time, try using `-auto-mask`

Correct for the primary beam as before

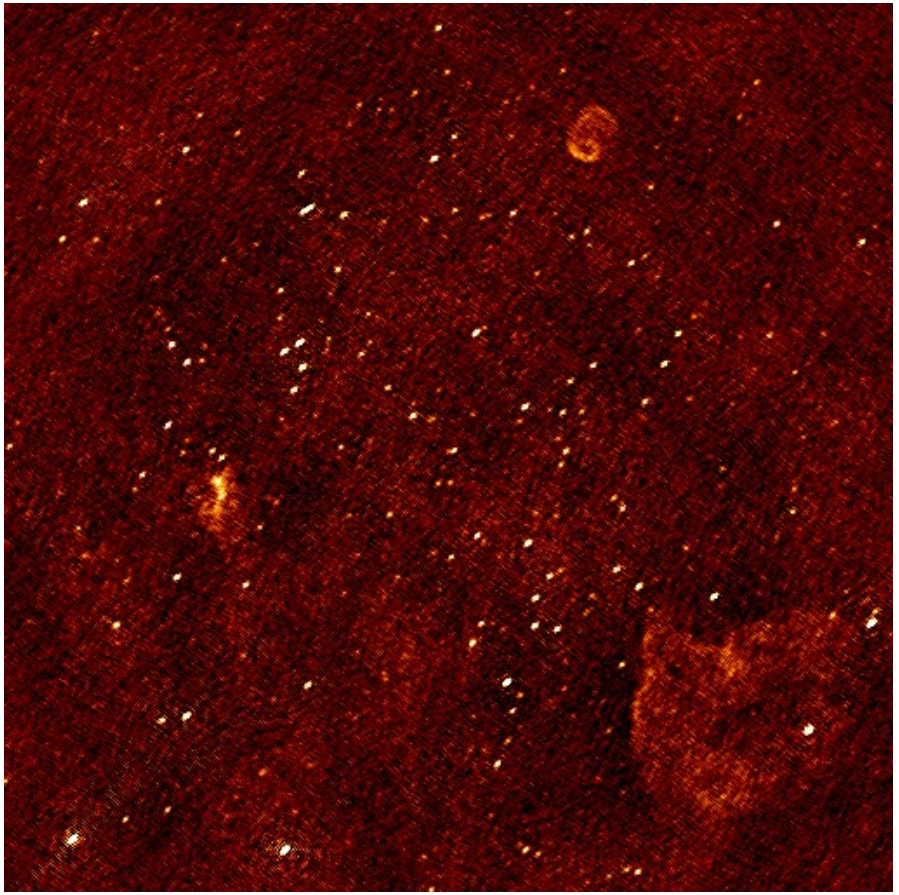
```
$ wsclean -size <width> <height> -scale <val>asec      \
    -apply-primary-beam                                \
    -niter <niter> -mgain 0.8 -auto-threshold <level>  \
    -weight [briggs <robustness> or natural]           \
    -taper-gaussian <val>asec                          \
          -name weighting observation-band1.ms
```

Weighting & tapers

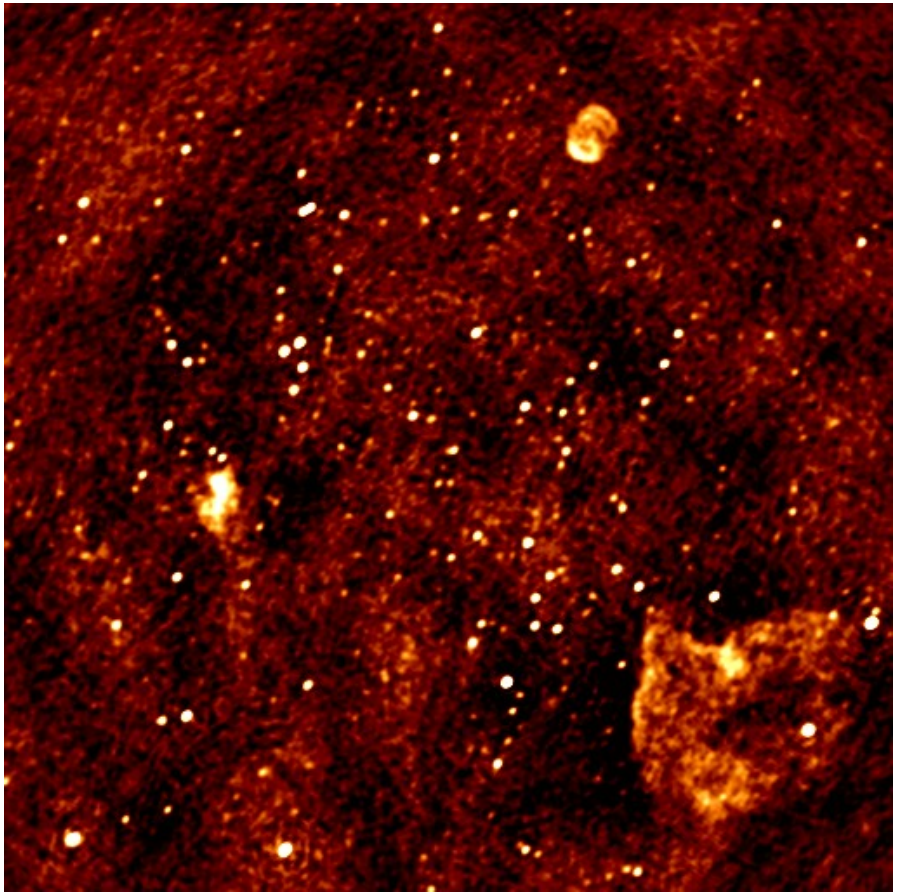
Example command:

```
$ wsclean -size 2400 2400 -scale 12asec                    \
      -weight briggs 0.5                                   \
      -niter 50000 -mgain 0.8 -auto-threshold 5            \
          -name weighting observation-band4.ms
```

With -weight briggs 0

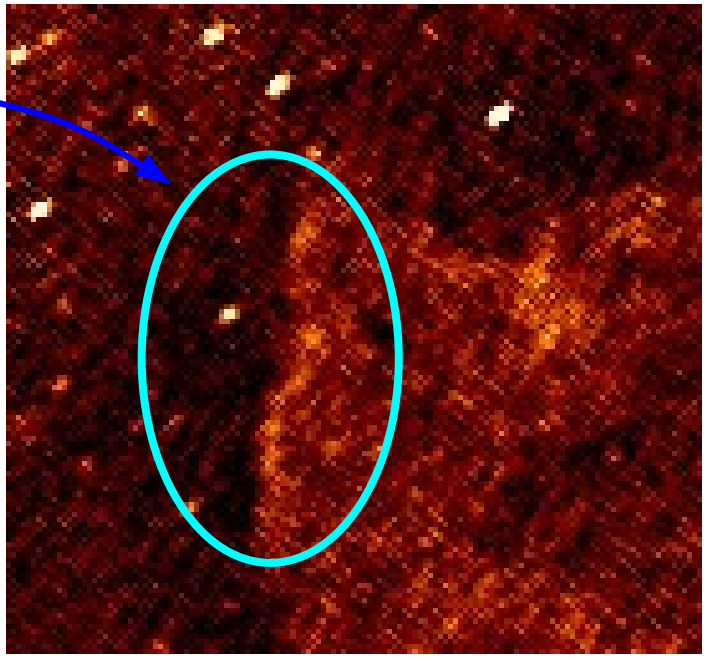


With -weight briggs 0 and -taper-gaussian 2amin



Multi-scale clean

Note the negative areas around the Diffuse sources.
Inspect the “model” image – how did WSClean model the diffuse emission?



Repeat the previous imaging, but use Multiscale cleaning.

Compare these images with the previous run.

```
$ wsclean -size <width> <height> -scale <val>asec      \
    -apply-primary-beam                                \
    -niter <niter> -mgain 0.8                          \
    -weight [your weighting choice]                    \
    -taper-gaussian <val>asec                          \
    -multiscale -auto-mask 5 -auto-threshold 1         \
          -name multiscale observation-band3.ms
```

Multiple output channels & joining

Several approaches for combining multiple subbands (MSes) :

- 1. Run WSClean on each band and combine images afterwards
  → Only limited cleaning possible.

- 2. Image all channels in one run with WSClean
  → Cleans deep, but assumes flux is constant over frequency.

- ...

Multiple output channels & joining

Several approaches for combining multiple subbands (MSes) :

- 1. Run WSClean on each band and combine images afterwards
  → Only limited cleaning possible.

- 2. Image all channels in one run with WSClean
  → Cleans deep, but assumes flux is constant over frequency.

- **3. Image all channels and use multi-frequency deconvolution**
  → Cleans deep & incorporates frequency dependency.
  Relevant parameters: -channels-out <count> -join-channels -fit-spectral-pol <terms> -deconvolution-channels <count>.

```
$ wsclean -size <width> <height> -scale <val>asec          \
    [..]                                                   \
    -channels-out <count> -join-channels                   \
    -fit-spectral-pol <terms>                              \
    -name mfclean *.ms
```

Multiple output channels & joining

Example command using multi-frequency deconvolution:

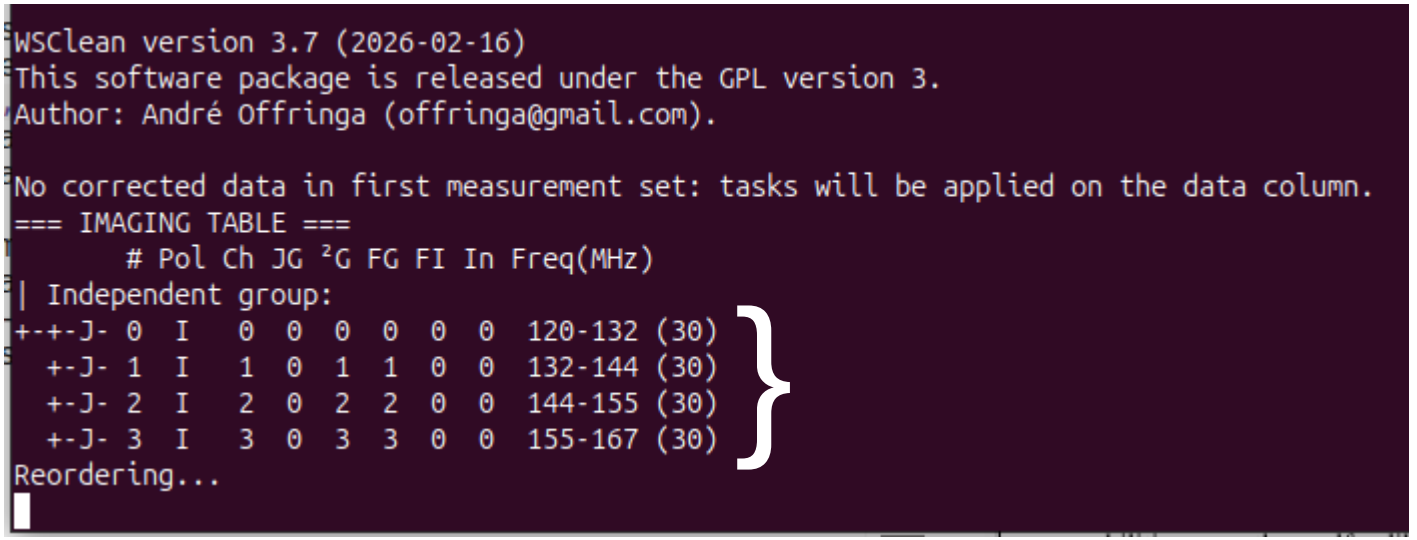
```
$ wsclean -size 2400 2400 -scale 12asec                          \
    -apply-primary-beam                                          \
    -weight briggs 0                                             \
    -multiscale -auto-mask 5                                     \
    -niter 10000000 -mgain 0.8 -auto-threshold 1                 \
    -channels-out 4 -join-channels -fit-spectral-pol 2           \
    -parallel-gridding 4 -save-source-list                       \
    -name mfclean *.ms
```

You can do this with one band, or even with all bands combined.
Also add ‘-save-source-list` and look at the resulting source file.

After starting a MF run, get some coffee :)
(it takes significantly longer...)
Analyse the individual output images and the MFS images.

Extra: Try using `-parallel-deconvolution 500` to speed up deconvolution.

- Note what WSClean says:



Polarized imaging

Example command using multi-polarization deconvolution:

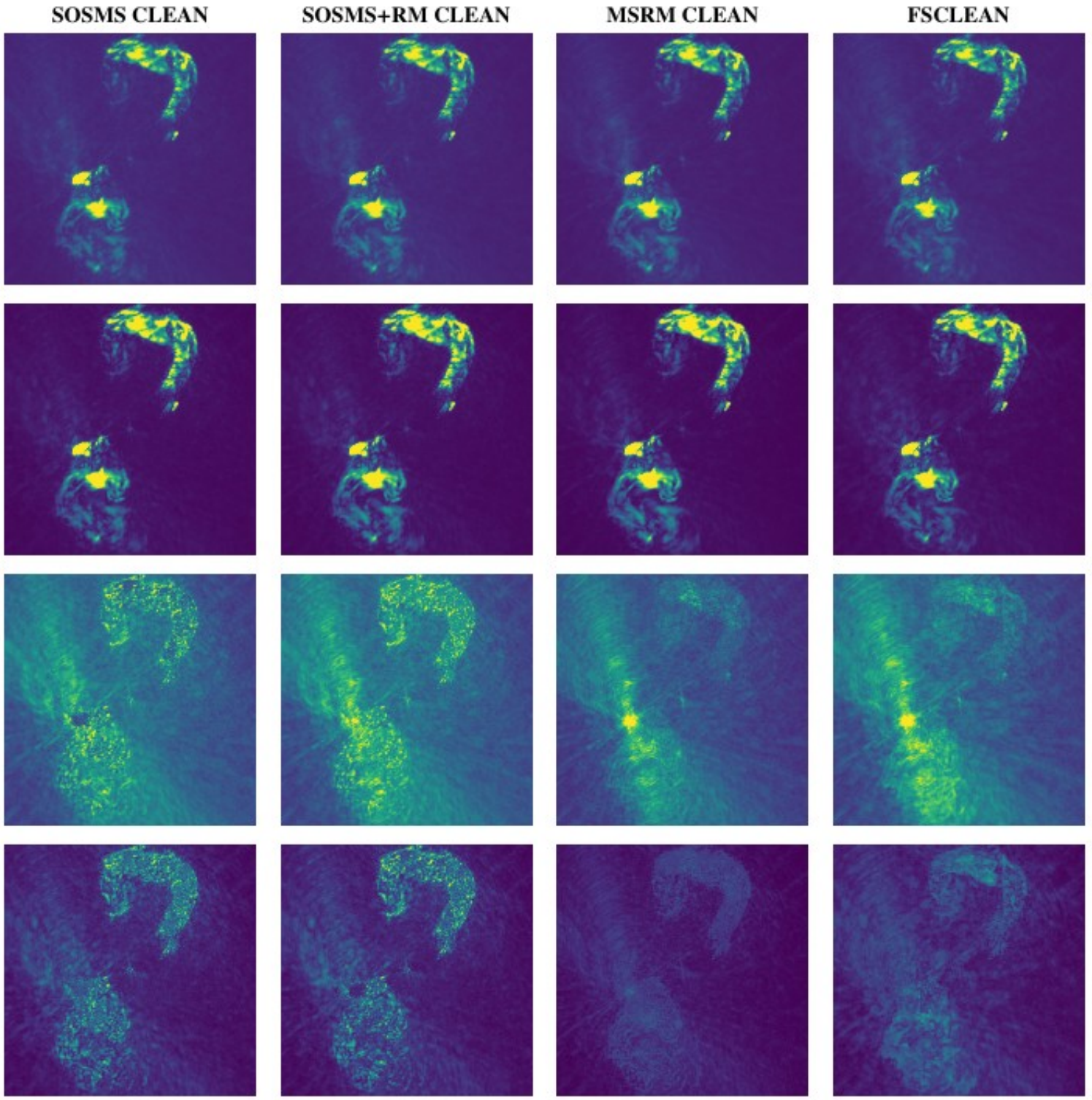
```
$ wsclean -size 2400 2400 -scale 12asec                    \
    -apply-primary-beam                                    \
    -pol qu -join-polarizations                            \
    -weight briggs 0                                       \
    -multiscale -auto-mask 5                               \
    -niter 1000000 -mgain 0.8 -auto-threshold 1            \
    -name polclean observation-band1.ms
```

More options for polarized cleaning:

`-squared-channel-joining`

`-join-polarizations`

`-rm-deconvolution` (in next version of WSClean)



(data credit: Vijay Mahatma)

Do lots of tweaking of WSClean parameters and see what they do! You can also try imaging more bands and channels. Using ‘-deconvolution-channels’ becomes then useful.

WSClean can make use of “*Image Domain Gridding*” (IDG). This makes imaging faster using GPUs and can make use of screens to correct direction dependent effects. Idg can be turned on using ‘`-gridder idg`’.

Another option to correct for direction dependent effects is using *facetting*, which wsclean also supports… but requires more “book keeping tasks” to get right. The Rapthor pipeline uses WSClean in this mode for HBA NL imaging.