LOFAR LBA: Library for low frequencies

Data calibration and imaging pipeline for the lowest frequencies

Jort Boxelaar

Università di Bologna
Istituto di Radioastronomia - INAF

LOFAR LBA: Library for low frequencies

Data calibration and imaging pipeline for the lowest frequencies

LBA Commissioning Team

Francesco De Gasperin - Henrik Edler - Jort Boxelaar - Thomas Pasini - Christian Groeneveld - Cristina Cordun - Reinout van Weeren - Alexander Drabent - Tim Shimwell

LiLF tutorial / demo



- Clone git repository: `git clone https://github.com/revoltek/LiLF.git`

LOFAR

LOw Frequency ARray



HBA
110 - 240 MHz
6″ Dutch resolution



LBA
10 - 90 MHz
15″ Dutch resolution

LOFAR

LOw Frequency ARray

HBA
110 - 240 MHz
6″ Dutch resolution

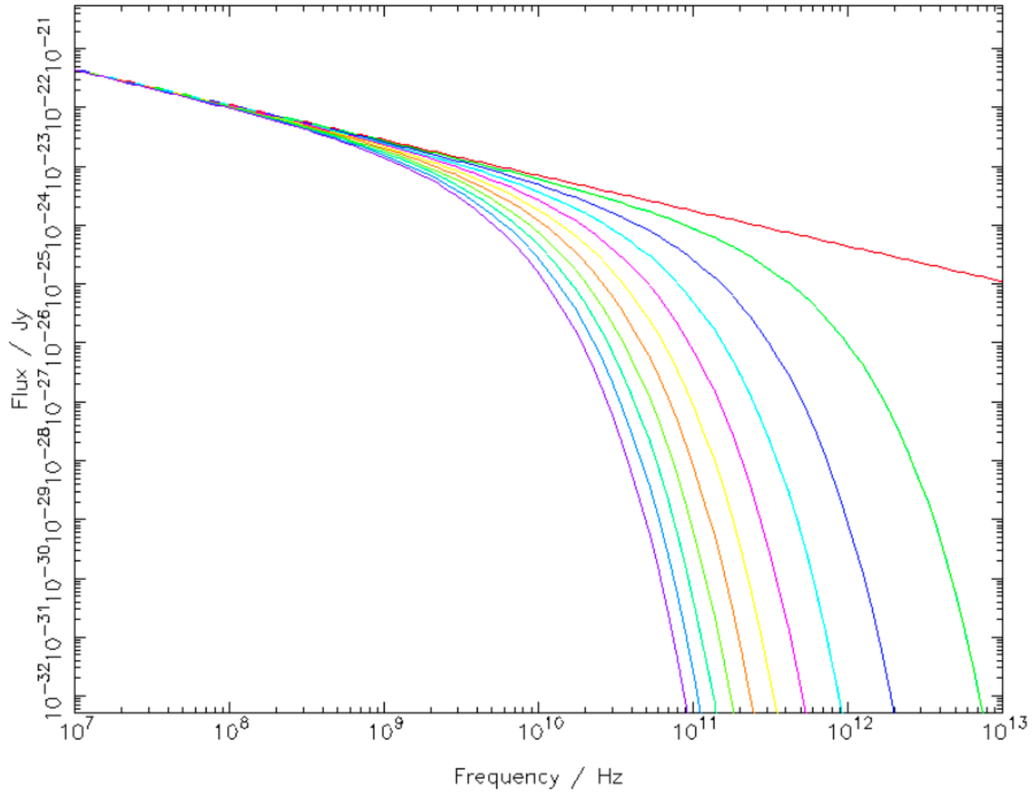


LBA
10 - 90 MHz
15″ Dutch resolution

Even lower frequencies

What can we learn from observations below 100 MHz?

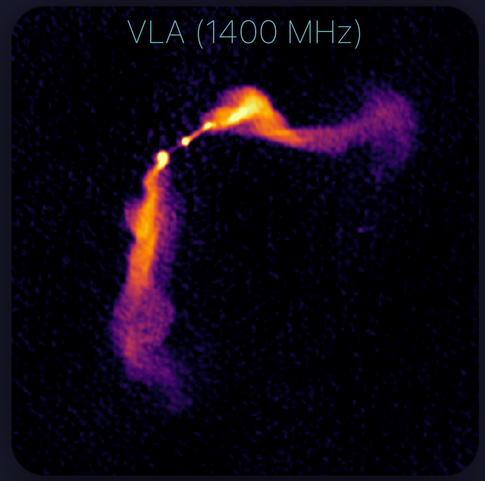
- Studying emission processes in galactic and extragalactic objects.
- Absorption processes at low frequencies (free-free absorption and synchrotron self absorption)

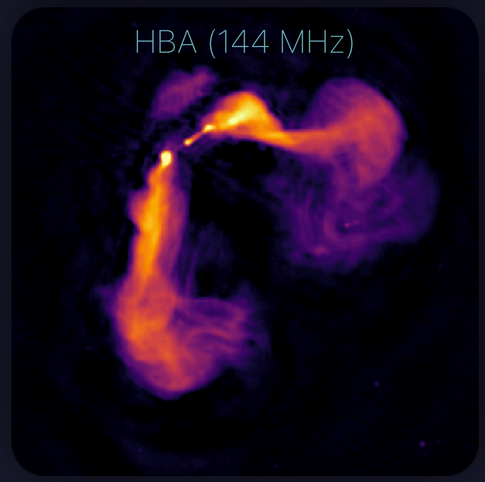


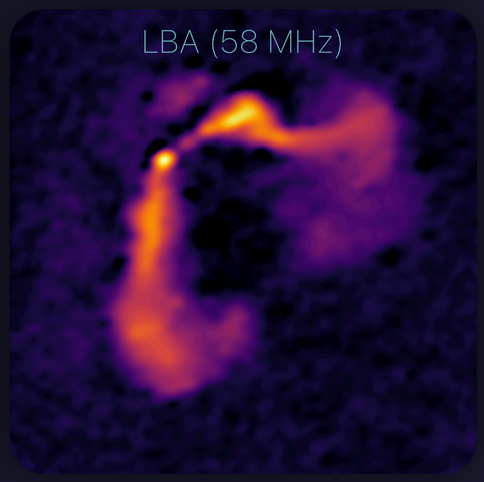
Even lower frequencies

What can we learn from observations below 100 MHz?

- Studying emission processes in galactic and extragalactic objects.
- Absorption processes at low frequencies (free-free absorption and synchrotron self absorption)
- Spectral properties of old diffuse emission in galaxies and clusters



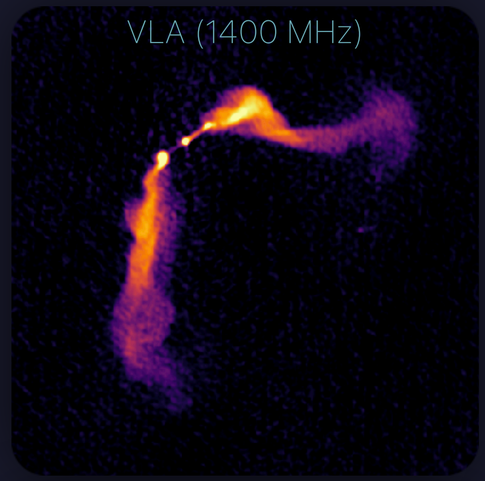


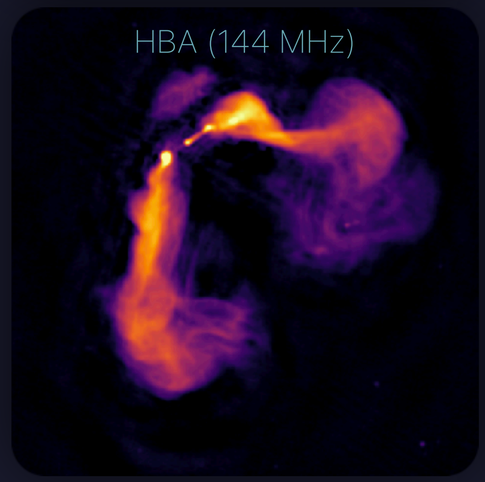


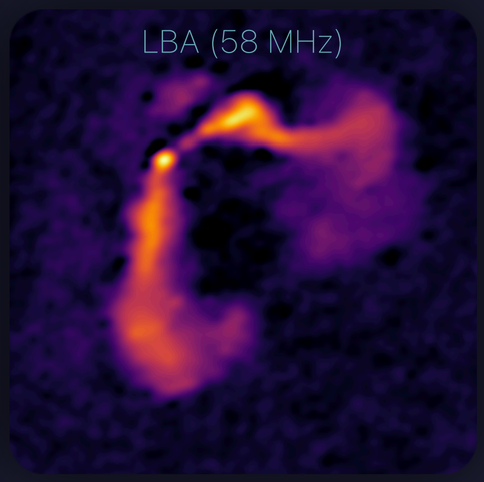
Even lower frequencies

What can we learn from observations below 100 MHz?

- Studying emission processes in galactic and extragalactic objects.
- Absorption processes at low frequencies (free-free absorption and synchrotron self absorption)
- Spectral properties of old diffuse emission in galaxies and clusters
- Large scale structure outside clusters
- Detect exoplanets (decametre band ~15 MHz)







LOFAR

LBA fact sheet

- Bandwidth: 10 - 90 MHz
- Maximum sensitivity: ~ 57 MHz
- Total dipole pairs: 4800
- Antennas per station:
  96 (48 usable for LOFAR 1)
- Area: ~ 75200 m^2
- Standard Dutch resolution: 15″



Observation setup

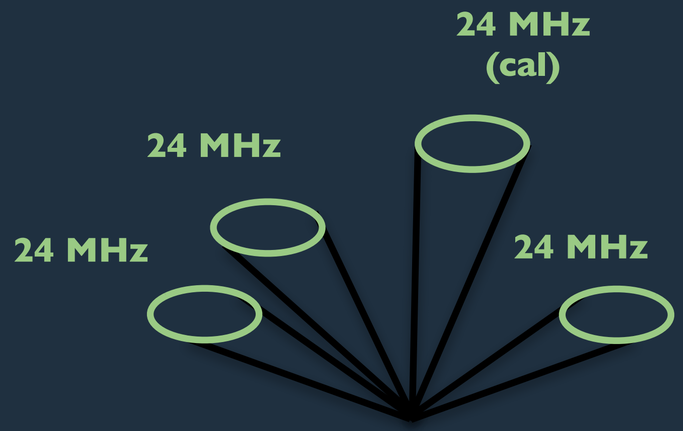
LOFAR 1

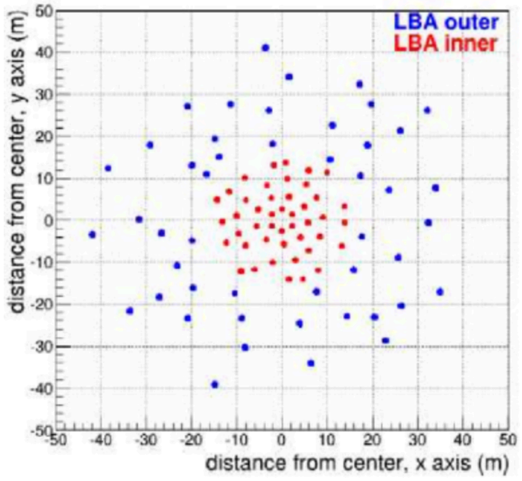
- **Beams:** 1 beam on a calibrator, other beams on one (or more) targets
- **Band:** 42-66 MHz (centre: 54 MHz). Other obs go down to 30 MHz or even ~15 MHz
- **Expected noise/angular resolution:**
  Dutch: 1-2 mJy/b at 15" resolution (at dec > 20 deg)
  IS: 1 mJy/b at ~1" resolution
- **Time/freq resolution (upper band):**
  Dutch: 4s 4ch/SB (limited by ionosphere)
  IS: 2s 16ch/SB (limited by smearing)

- **Caveats:**
  - Be far from the Sun and Jupiter
  - Highest possible elevation

LOFAR 2

- **Band/Beams:** same considerations, but we have twice the usable band (and HBA)
- **Expected noise/angular resolution:**
  Dutch: 250 uJy/b at 15" resolution
  (at dec > 20 deg)
  IS: 350 uJy/b at 1" resolution





LOFAR 2.0 improvement

Double the number of usable antennas

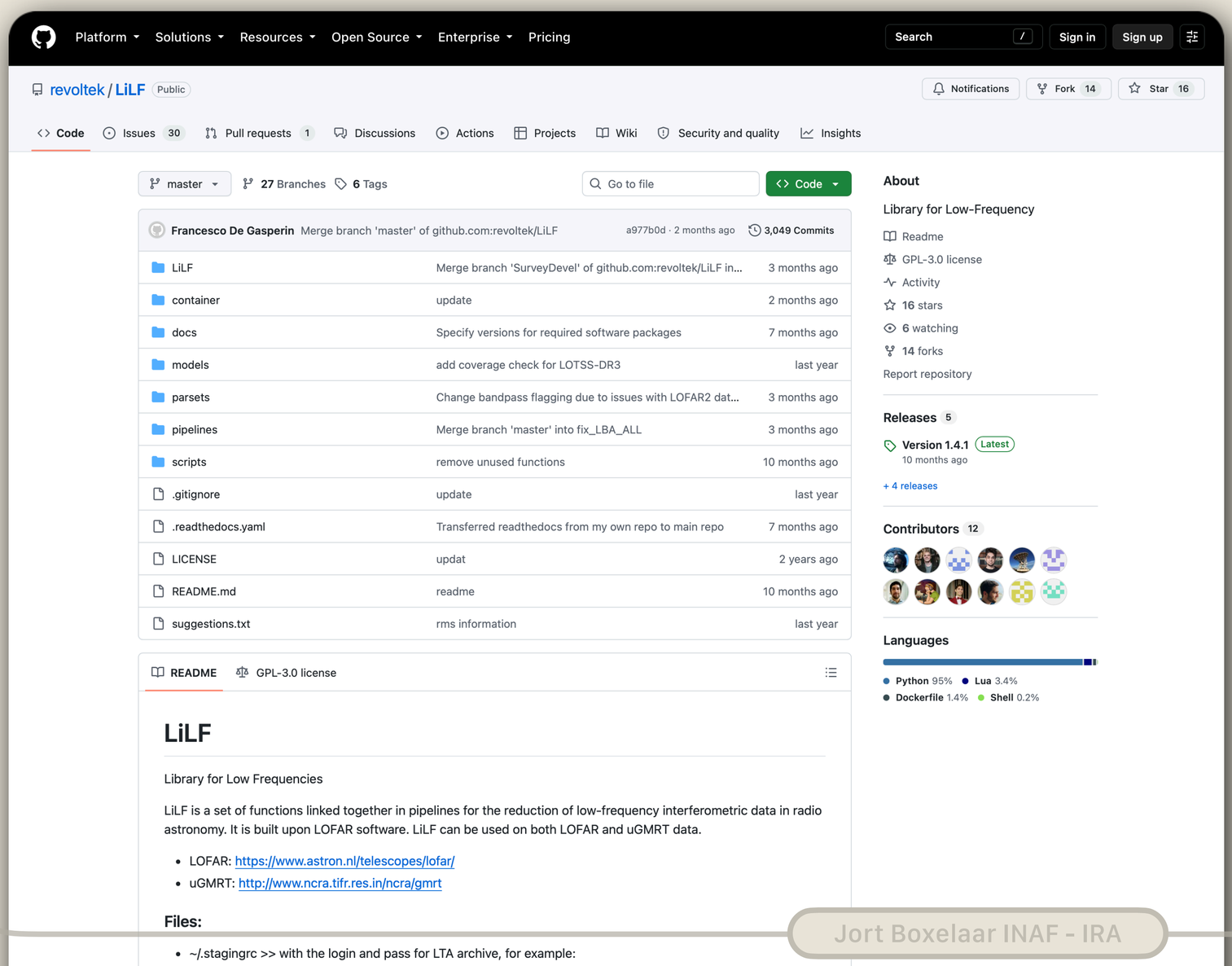
- To obtain an image with the same RMS noise (σ_{rms}) using the same total bandwidth ($\Delta\nu$), the required observing time is $t/4$
- For an observation with the same observing time and bandwidth as a LOFAR 1 observation, the RMS noise goes down by a factor $\sigma_{\mathrm{rms}}/2$
- To reach confusion noise with LOFAR 2, would require ~250 hrs of observations (instead of 1000 hrs)

$$N_a = 48 \rightarrow 96$$

$$N_{\mathrm{baseline}} \approx N_a^2/2$$

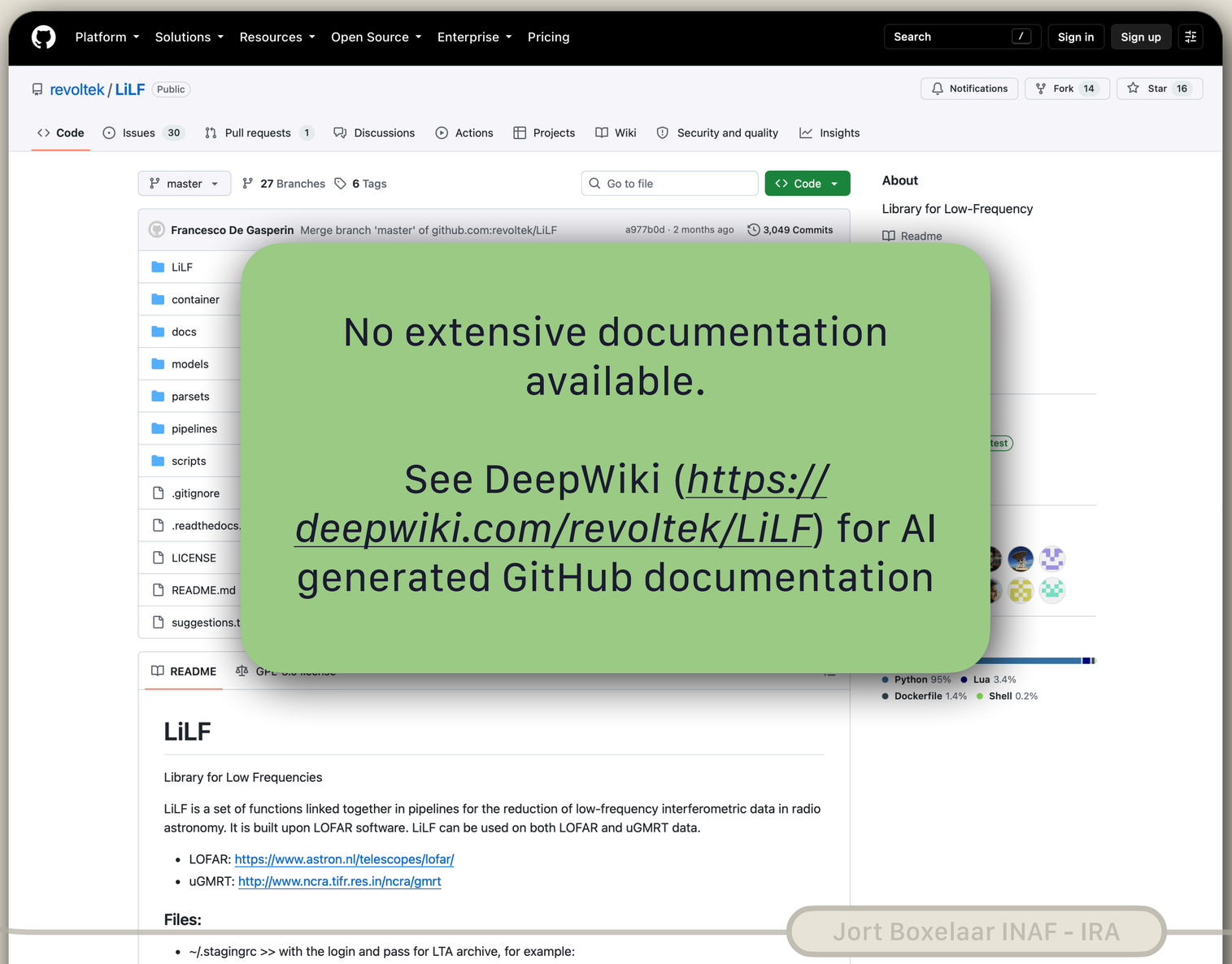
$$\sigma_{\mathrm{rms}} \propto \frac{1}{\sqrt{N_{\mathrm{baseline}} t \Delta\nu}}$$

LiLF: Library for Low Frequencies



- Standard pipeline for LBA data
- Special calibration procedures tailored for ultra-low frequencies
- First version released in 2021 (multiple major revisions since then)
- Currently rewriting code structure to prepare for LOFAR 2

LiLF: Library for Low Frequencies



No extensive documentation available.

See DeepWiki (*https://deepwiki.com/revoltek/LiLF*) for AI generated GitHub documentation

For publications related to LiLF, see:

- De Gasperin+ 2019 (arXiv:1811.07954)
- De Gasperin+ 2020 (arXiv:2007.16043)

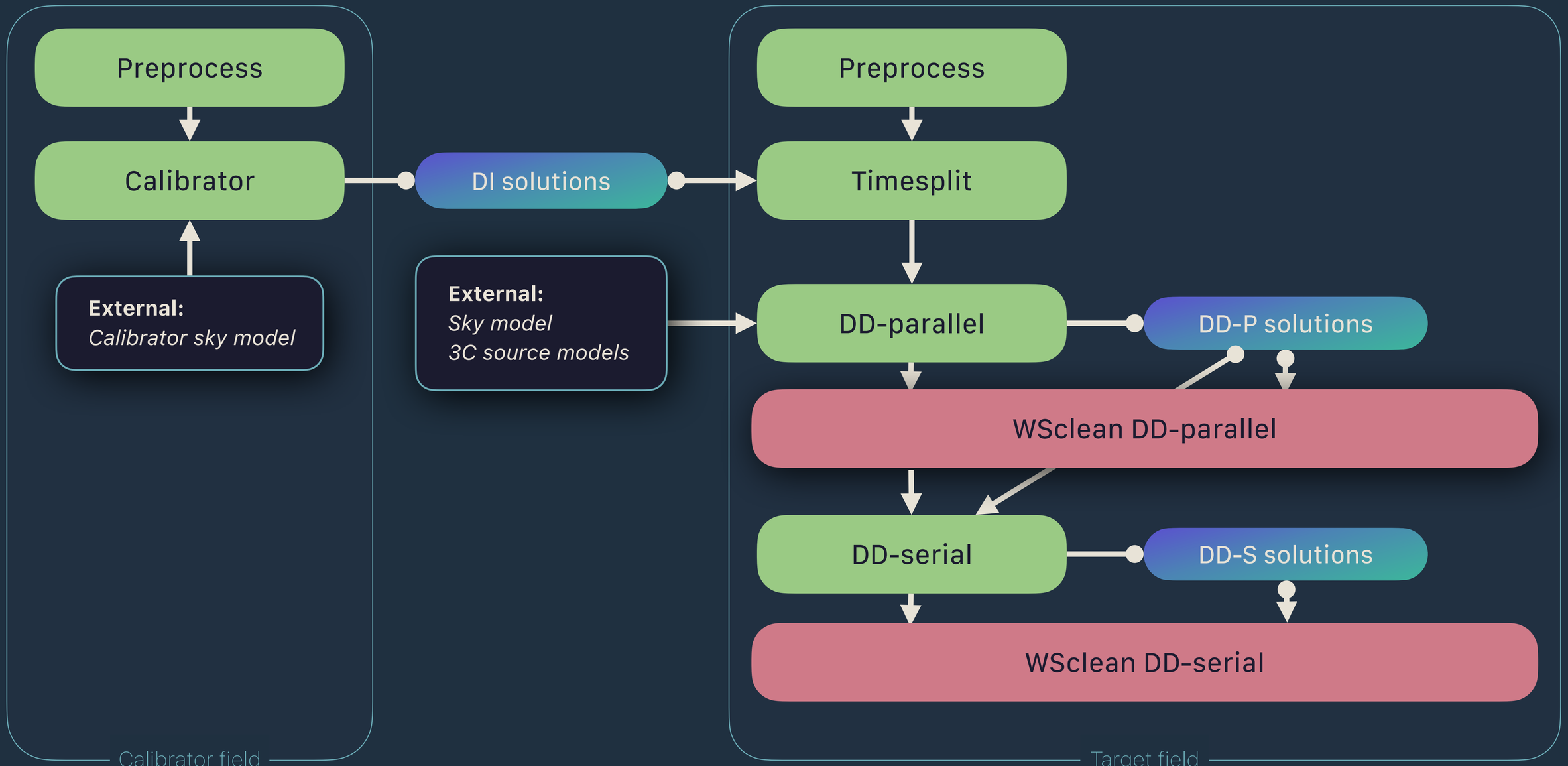
Info on ionospheric effects

- De Gasperin+ 2018 (arXiv:1804.07947)

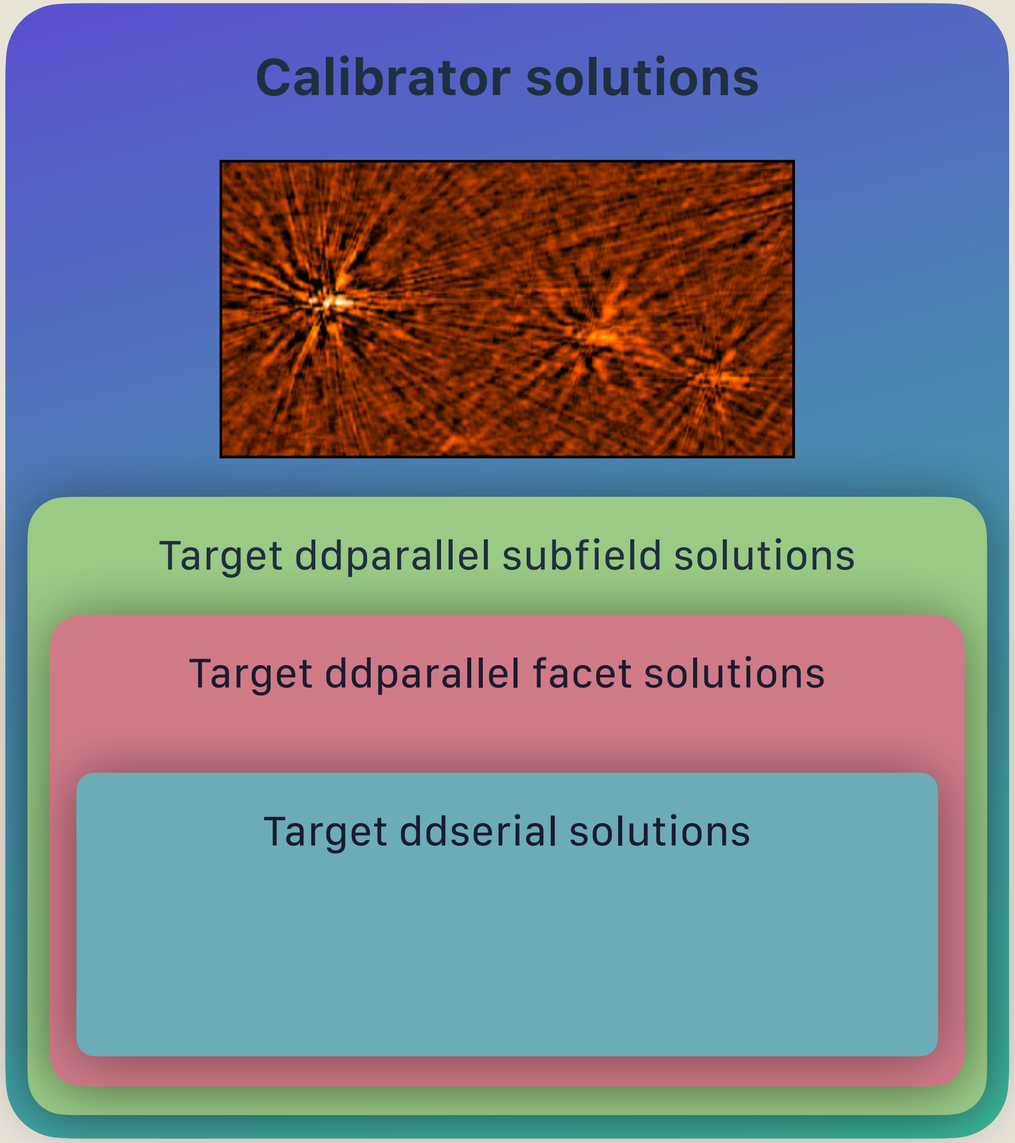
Demixing Description

- De Gasperin+2020 (arXiv:2002.10431)

LiLF: Pipeline overview



LiLF: Multi-layer calibration



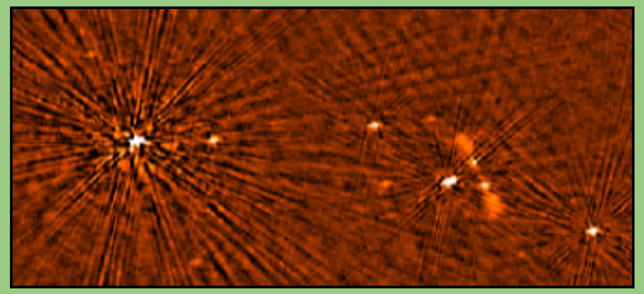
- **Conventional approach:** Iteratively improving the MODEL_DATA with imaging and solving from scratch
- **LiLF:** Apply previous solutions to DATA and solve for new solutions ‘on top of’ previous ones.

Slide by: F. De Gasperin

LiLF: Multi-layer calibration

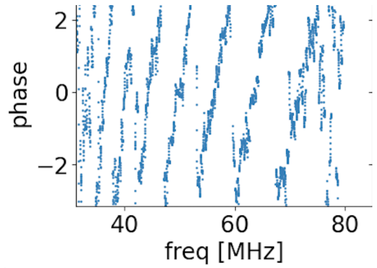
Calibrator solutions

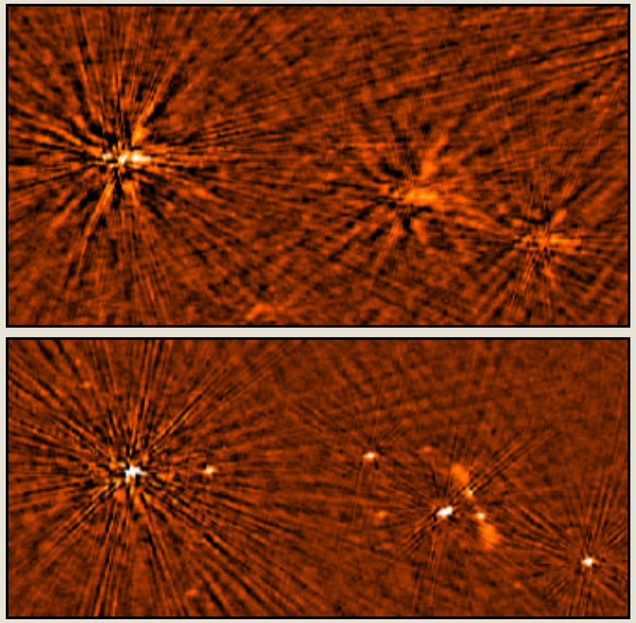
Target ddparallel subfield solutions



Target ddparallel facet solutions

Target ddserial solutions





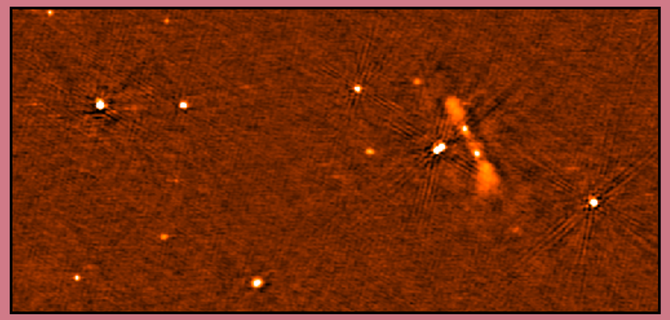
Slide by: F. De Gasperin

LiLF: Multi-layer calibration

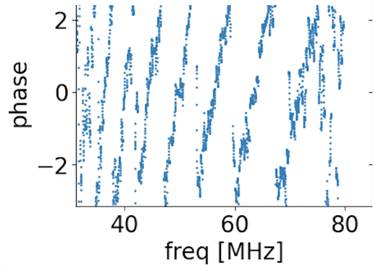
Calibrator solutions

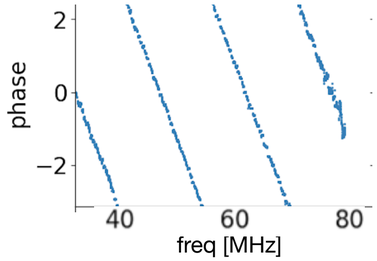
Target ddparallel subfield solutions

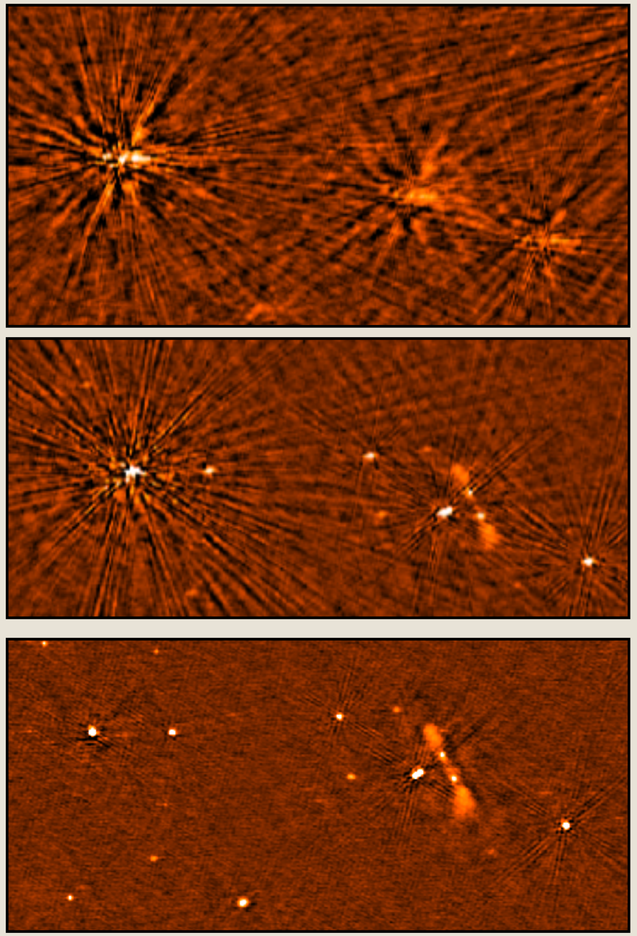
Target ddparallel facet solutions



Target ddserial solutions







Slide by: F. De Gasperin

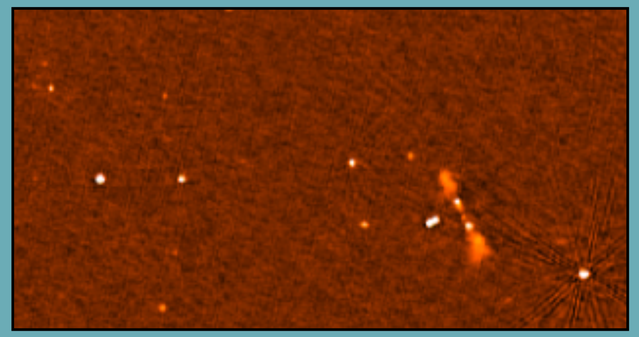
LiLF: Multi-layer calibration

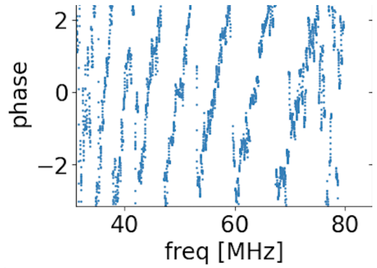
Calibrator solutions

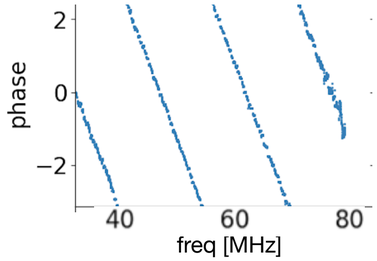
Target ddparallel subfield solutions

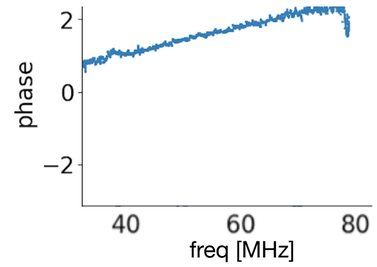
Target ddparallel facet solutions

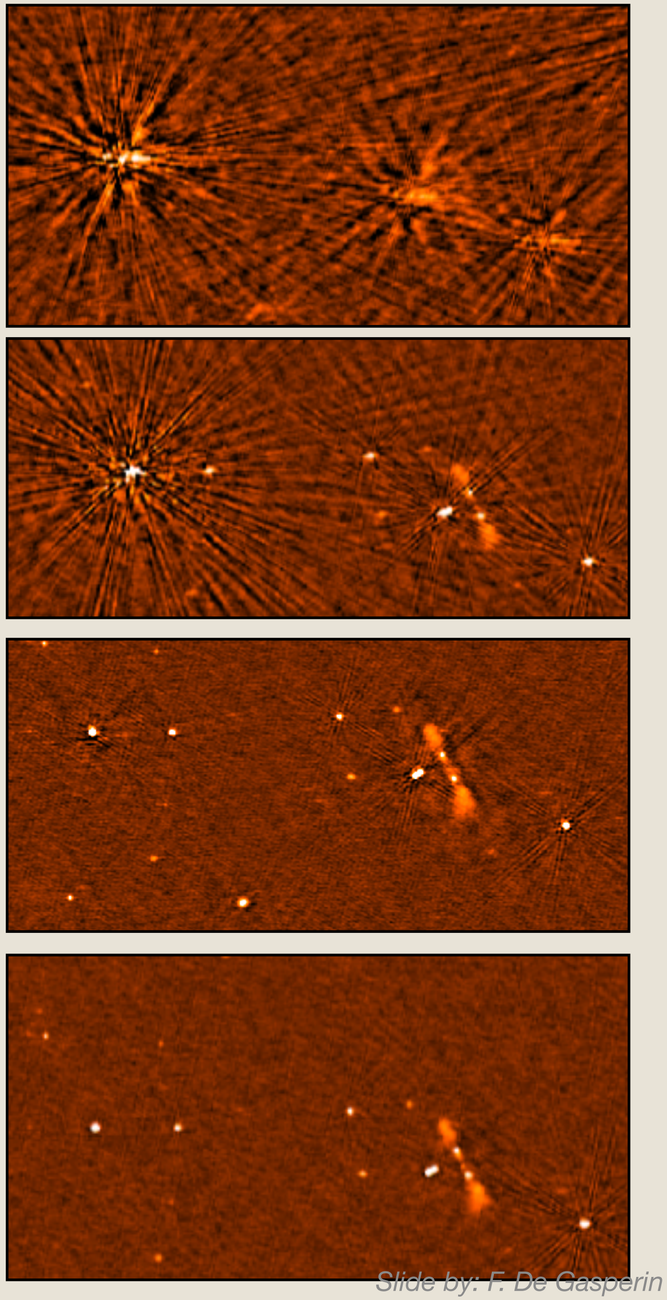
Target ddserial solutions









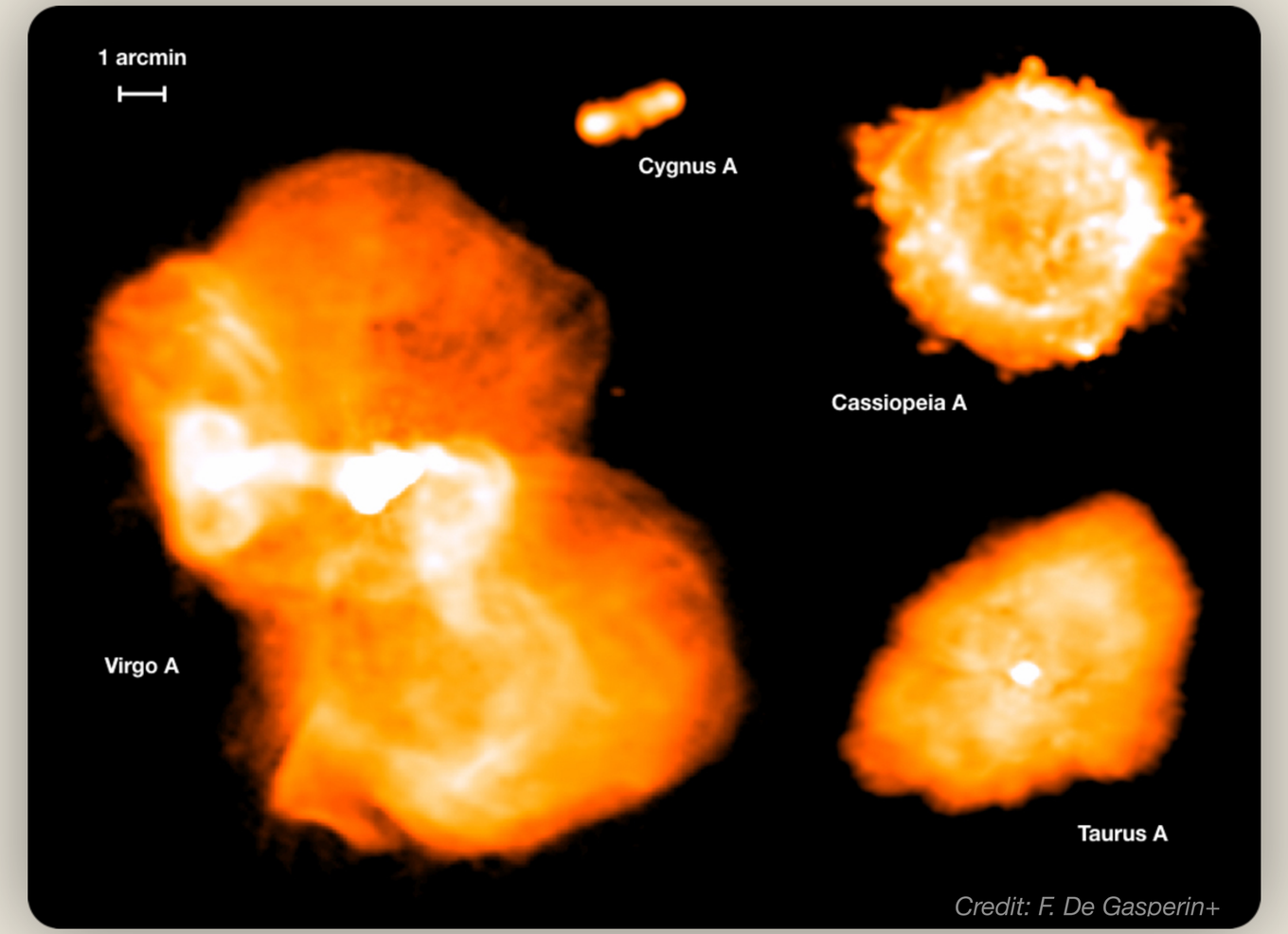


Slide by: F. De Gasperin

Calibration

Preprocess

- AO-Flagger
  (Should be done by observatory)
- Rescale flux
- Rename and average
- De-mixing of ‘A-Team’ sources
  (Should be done by observatory)
  - Cassiopeia A, Cygnus A,
    Taurus A, and Virgo A
  - Flux densities > 1000 Jy



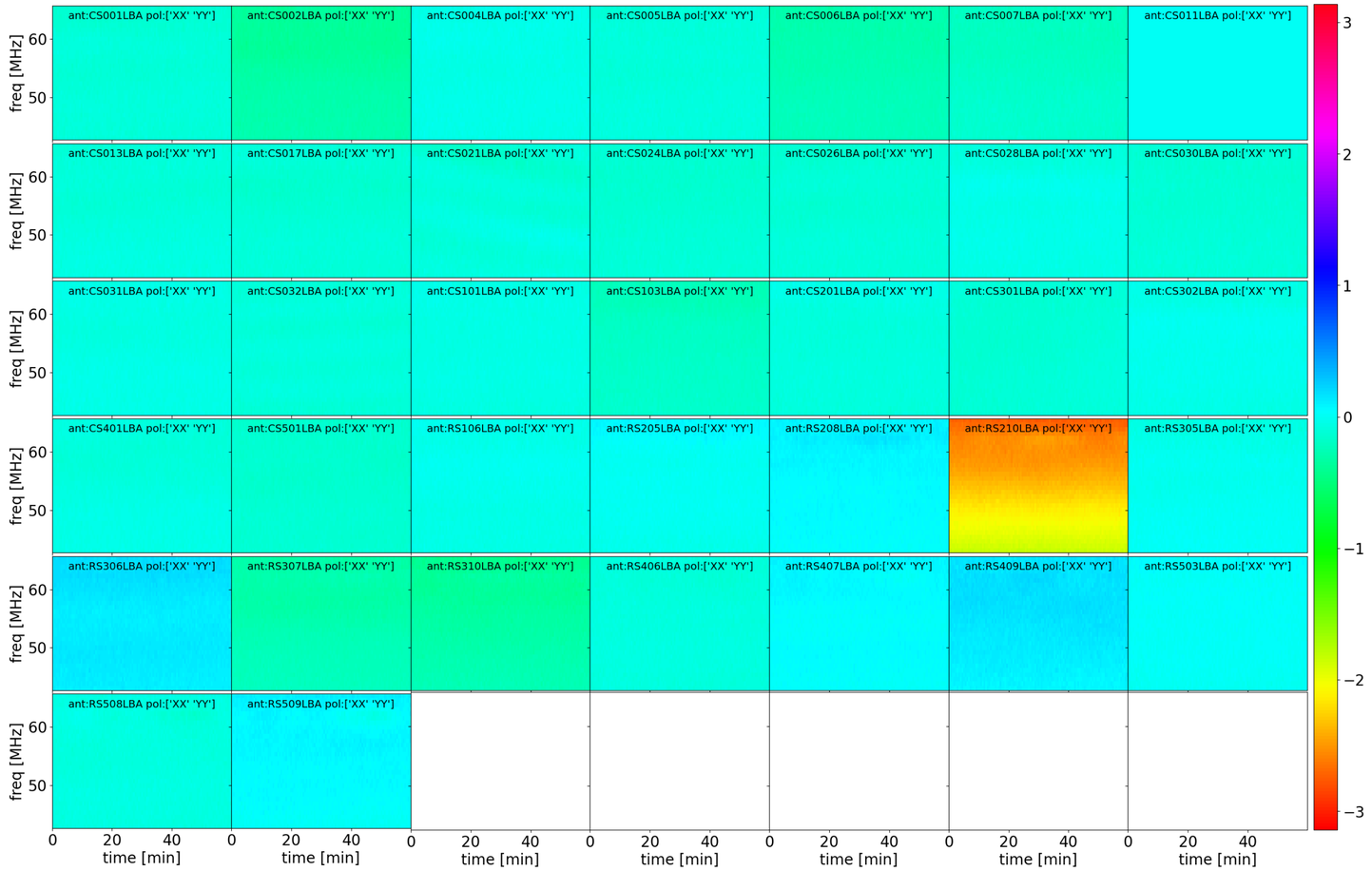
Calibration

Calibrator

Polarisation mis-alignment

$$C_{\mathrm{pa}} = \begin{pmatrix} e^{i\theta} & 0 \\ 0 & 0 \end{pmatrix}$$

Element beam
Bandpass
Faraday rotation
Ionospheric dispersion
Clock delays



Calibration

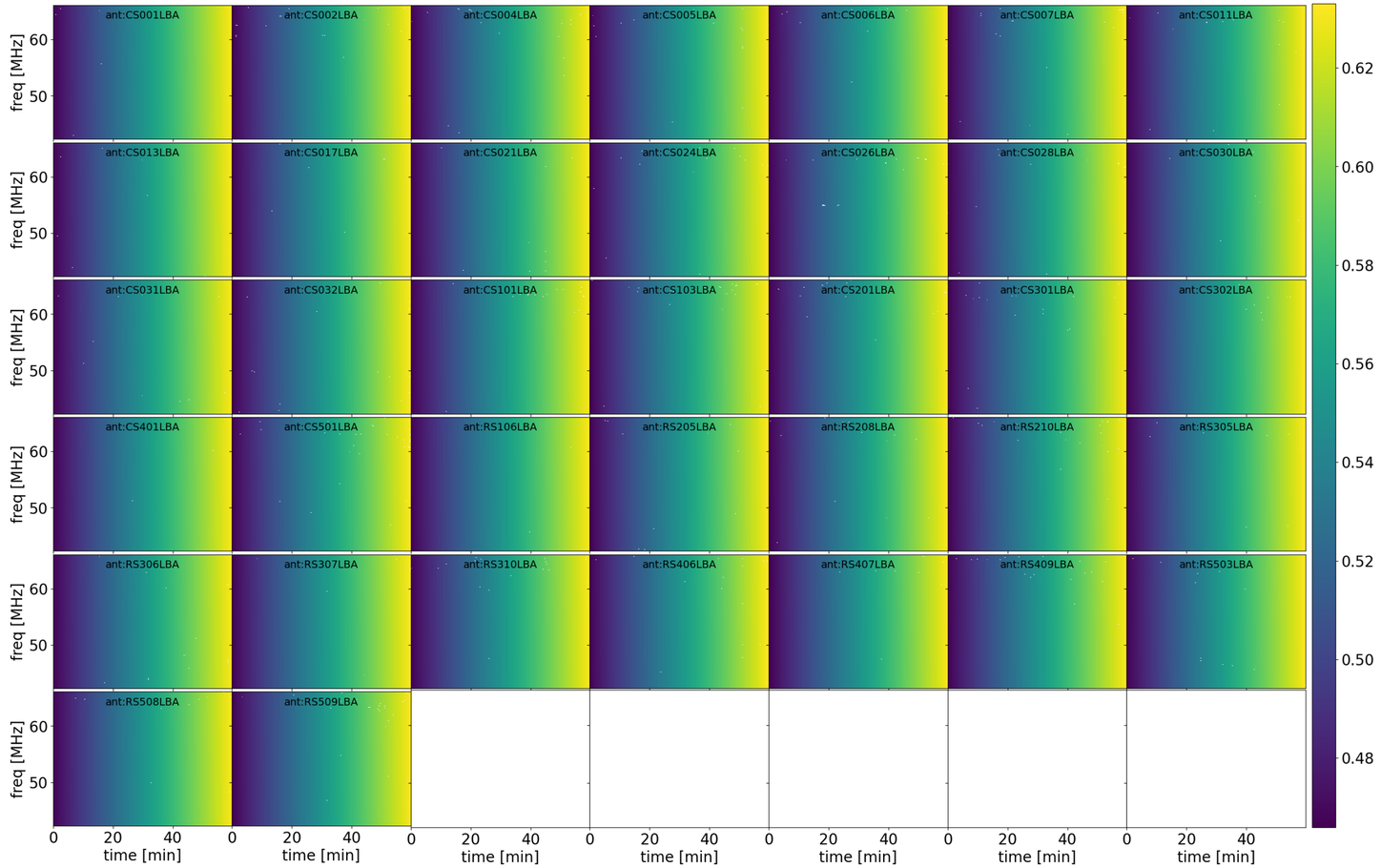
Calibrator

Polarisation mis-alignment

Element beam

$$G = \begin{pmatrix} g_{xx} & g_{xy} \\ g_{yx} & g_{yy} \end{pmatrix}$$

Bandpass
Faraday rotation
Ionospheric dispersion
Clock delays



Calibration

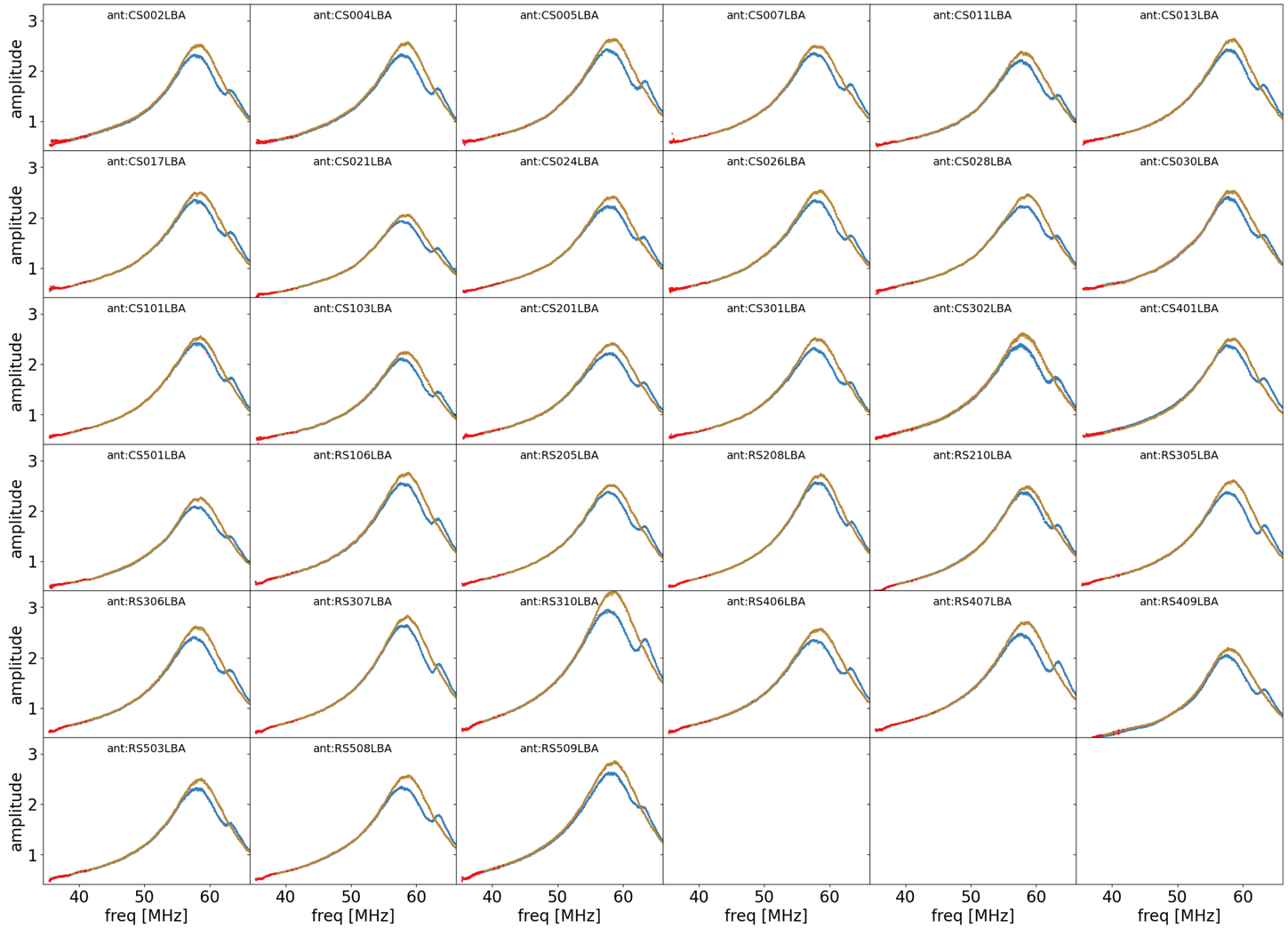
Calibrator

Polarisation mis-alignment
Element beam

Bandpass

$$C_{\mathrm{bp}} = \begin{pmatrix} g_x & 0 \\ 0 & g_y \end{pmatrix}$$

Faraday rotation
Ionospheric dispersion
Clock delays



Calibration

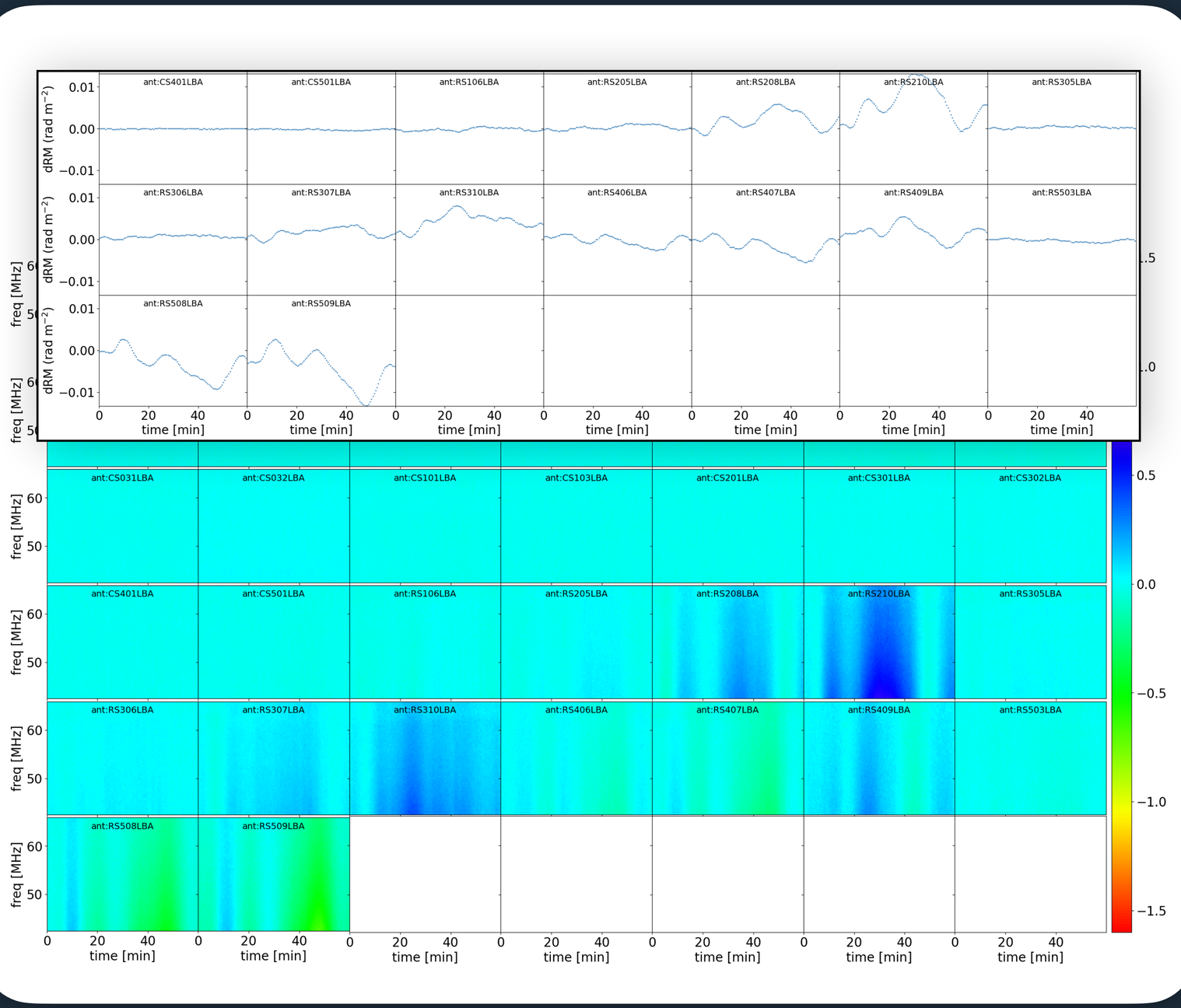
Calibrator

Polarisation mis-alignment
Element beam
Bandpass

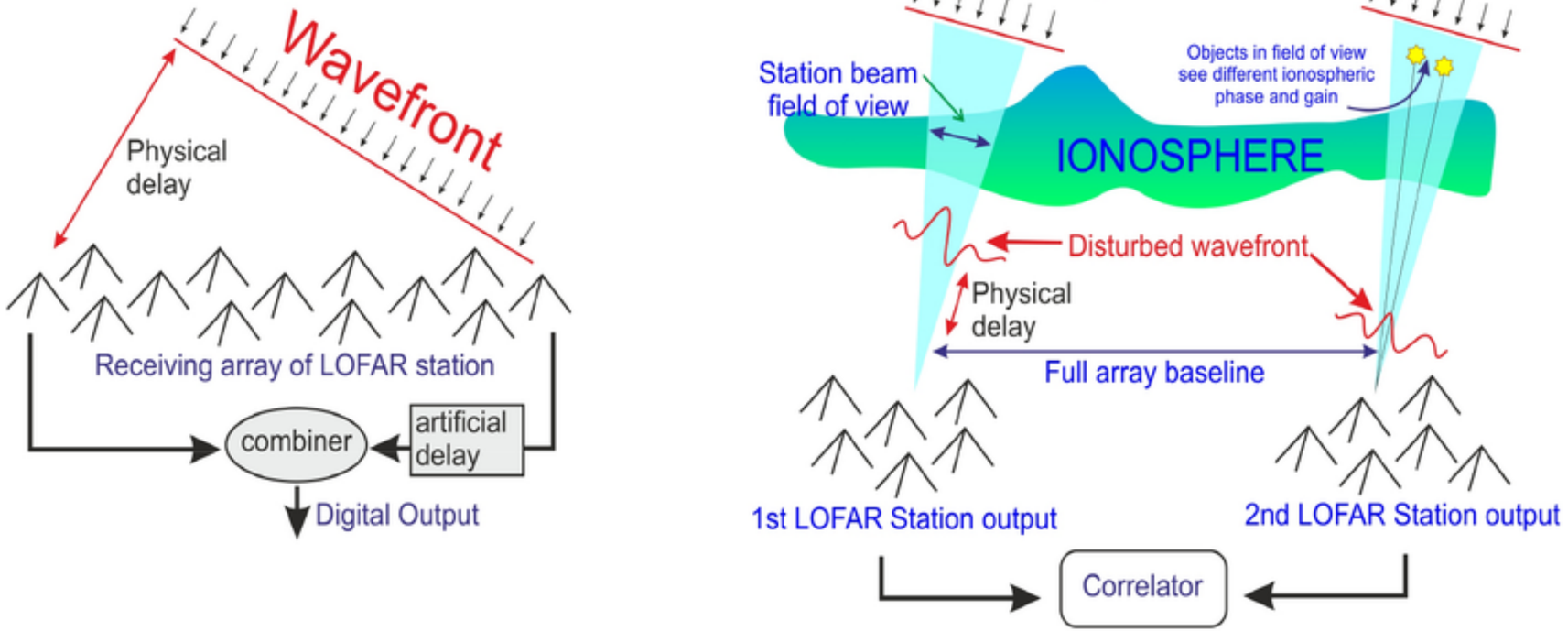
Faraday rotation

$$R_{\mathrm{fr}}(\phi) = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix}$$

Ionospheric dispersion
Clock delays

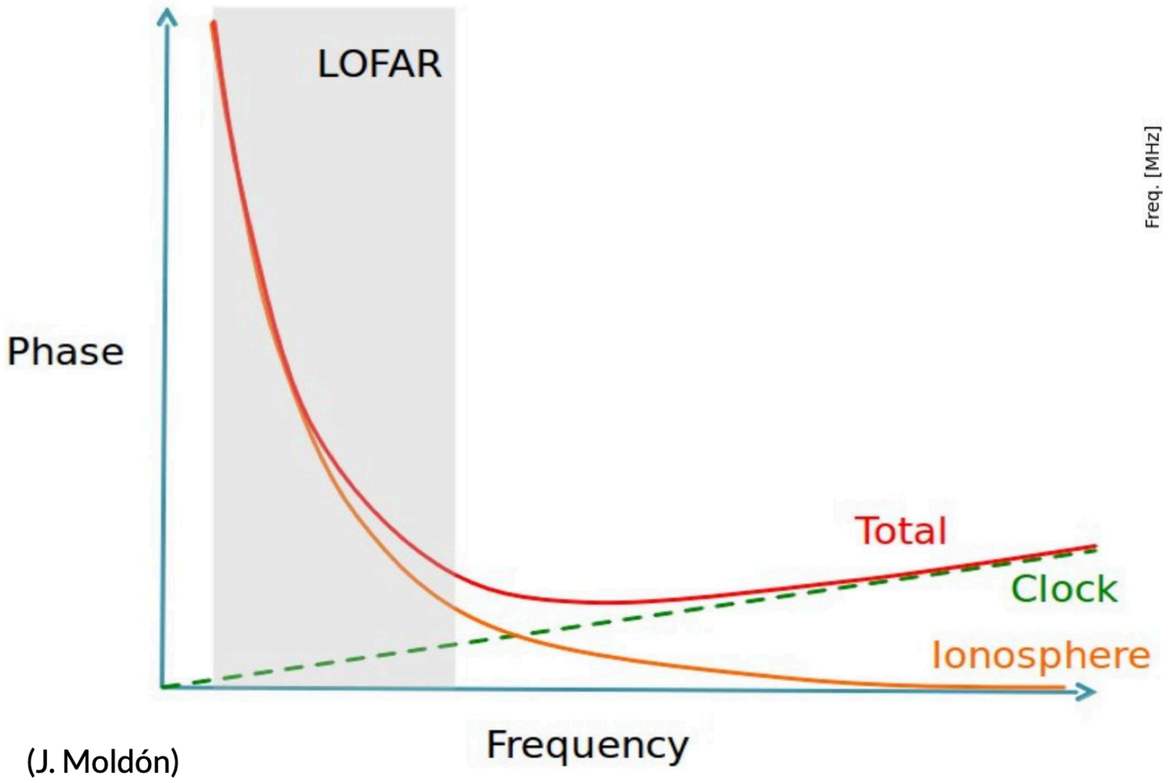


Intermezzo: The ionosphere



Credit: Van der Tol & Van der Veen (2007)

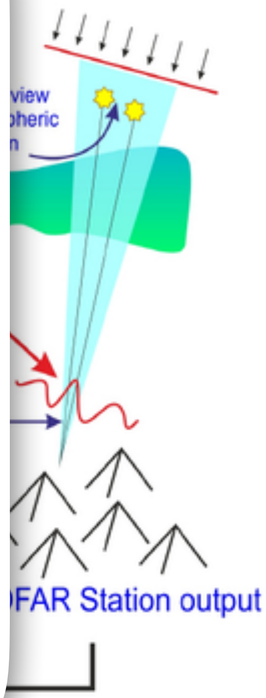
Intermezzo: The ionosphere



Freq. [MHz]

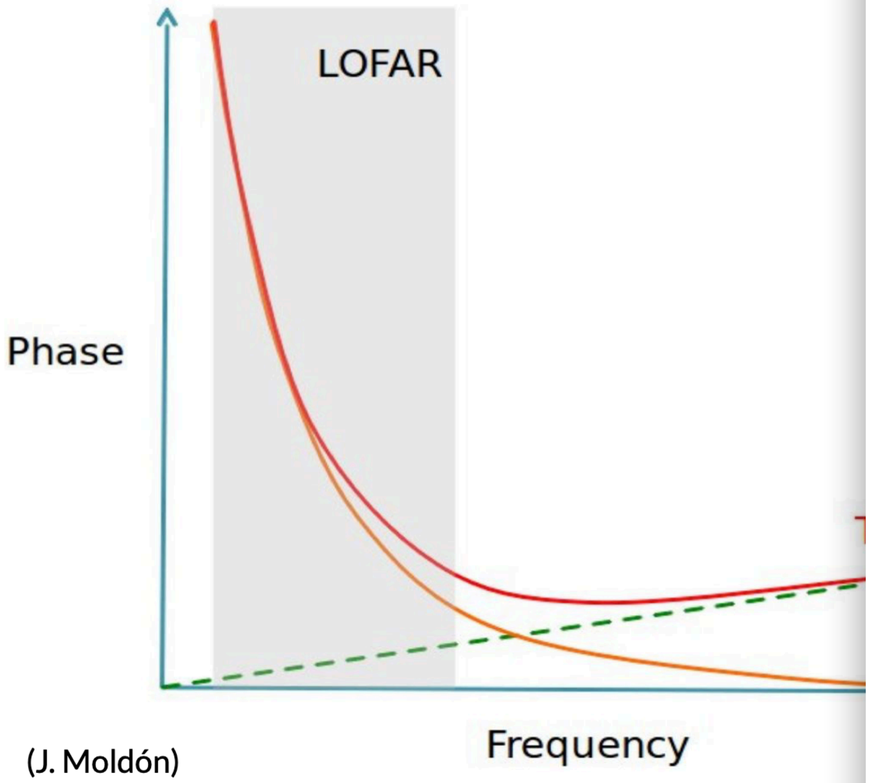
$$\delta\Phi_{TEC} \propto \nu^{-1}$$
$$\delta\Phi_{Faraday} \propto \nu^{-2}$$

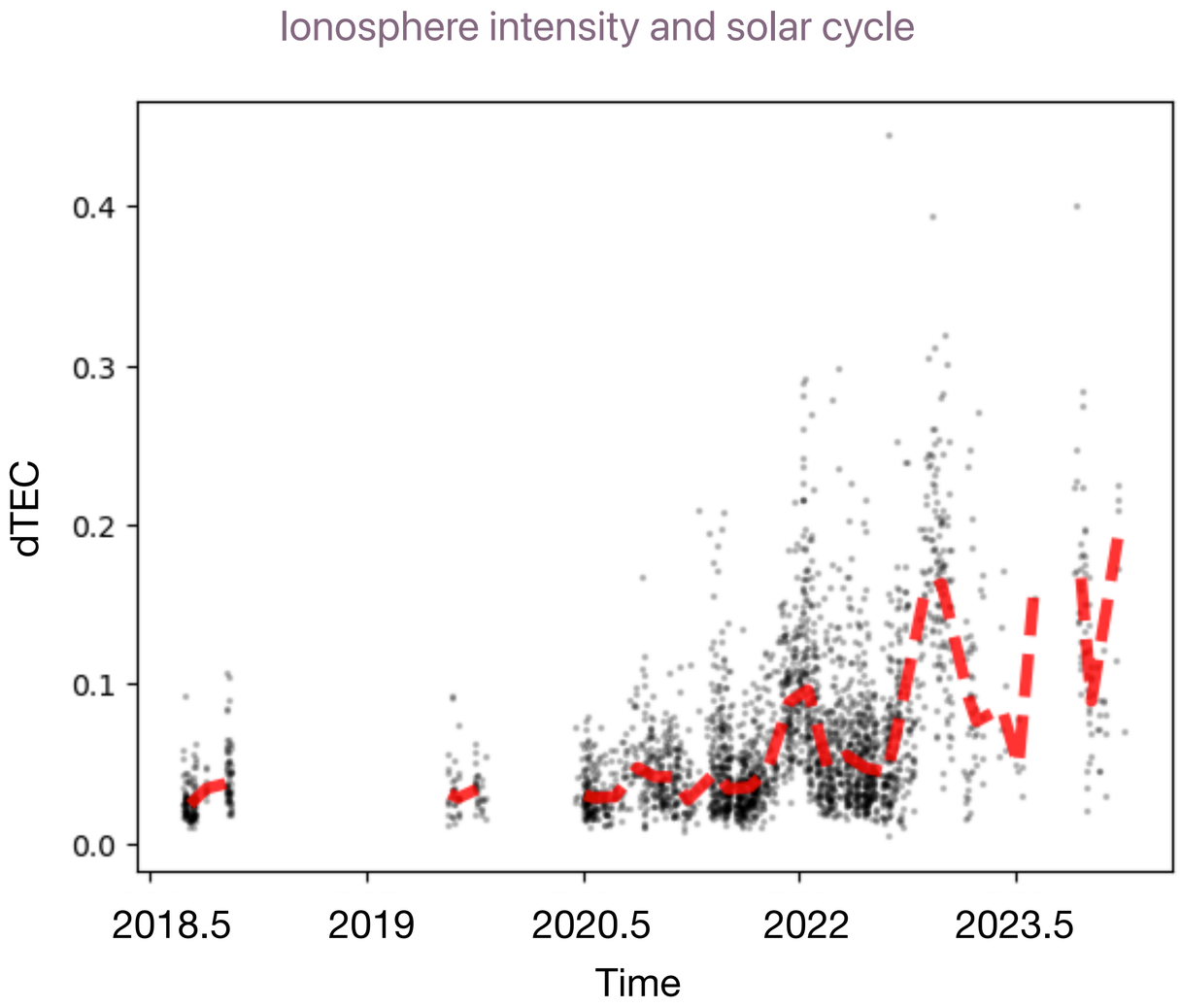
- Time, frequency, baseline and direction dependent



Credit: Van der Tol & Van der Veen (2007)

Intermezzo: The ionosphere





Calibration

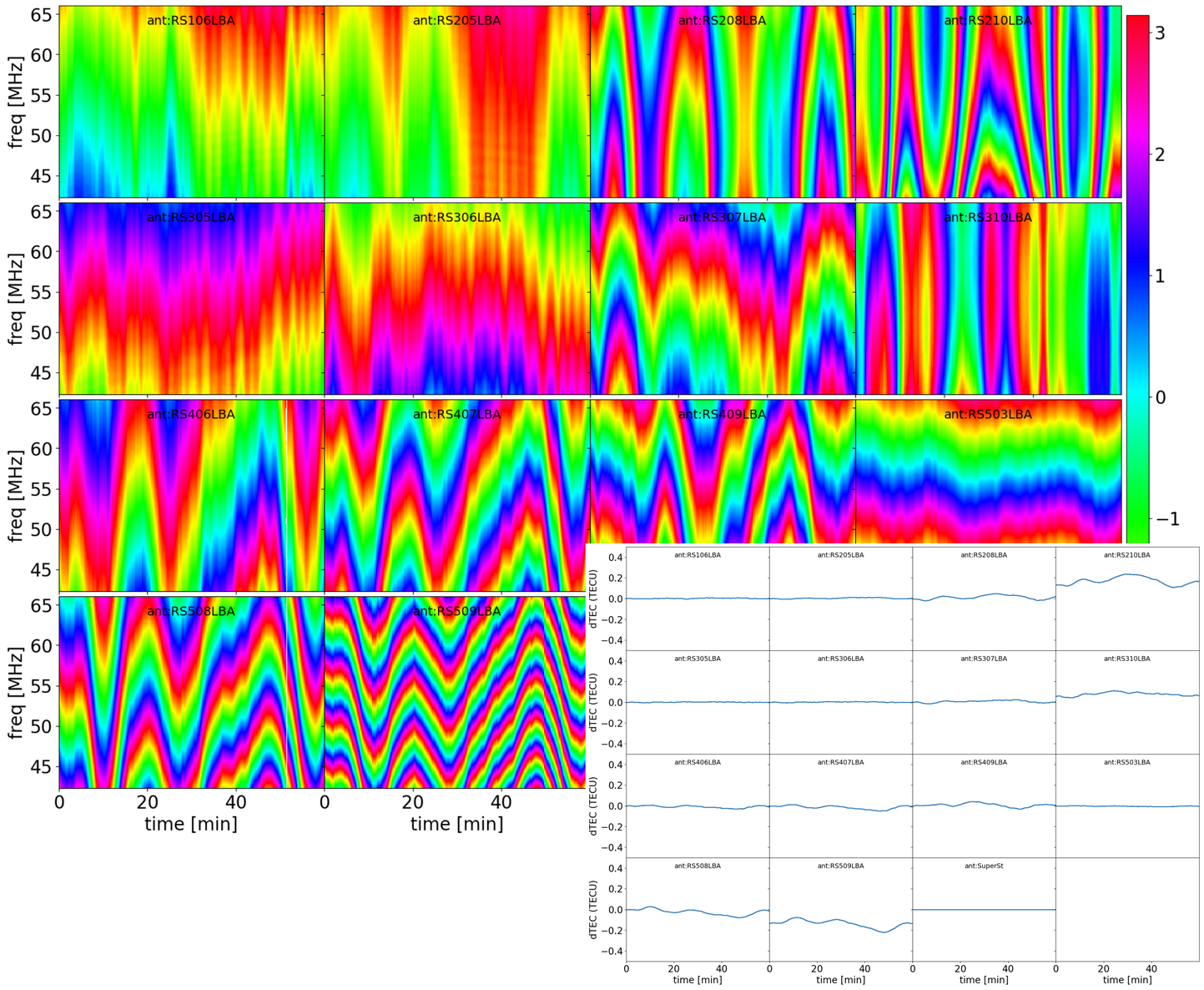
Calibrator

Polarisation mis-alignment
Element beam
Bandpass
Faraday rotation

Ionospheric dispersion

$$K_{\text{iono}} = \begin{pmatrix} e^{i\phi} & 0 \\ 0 & e^{i\phi} \end{pmatrix}$$

Clock delays



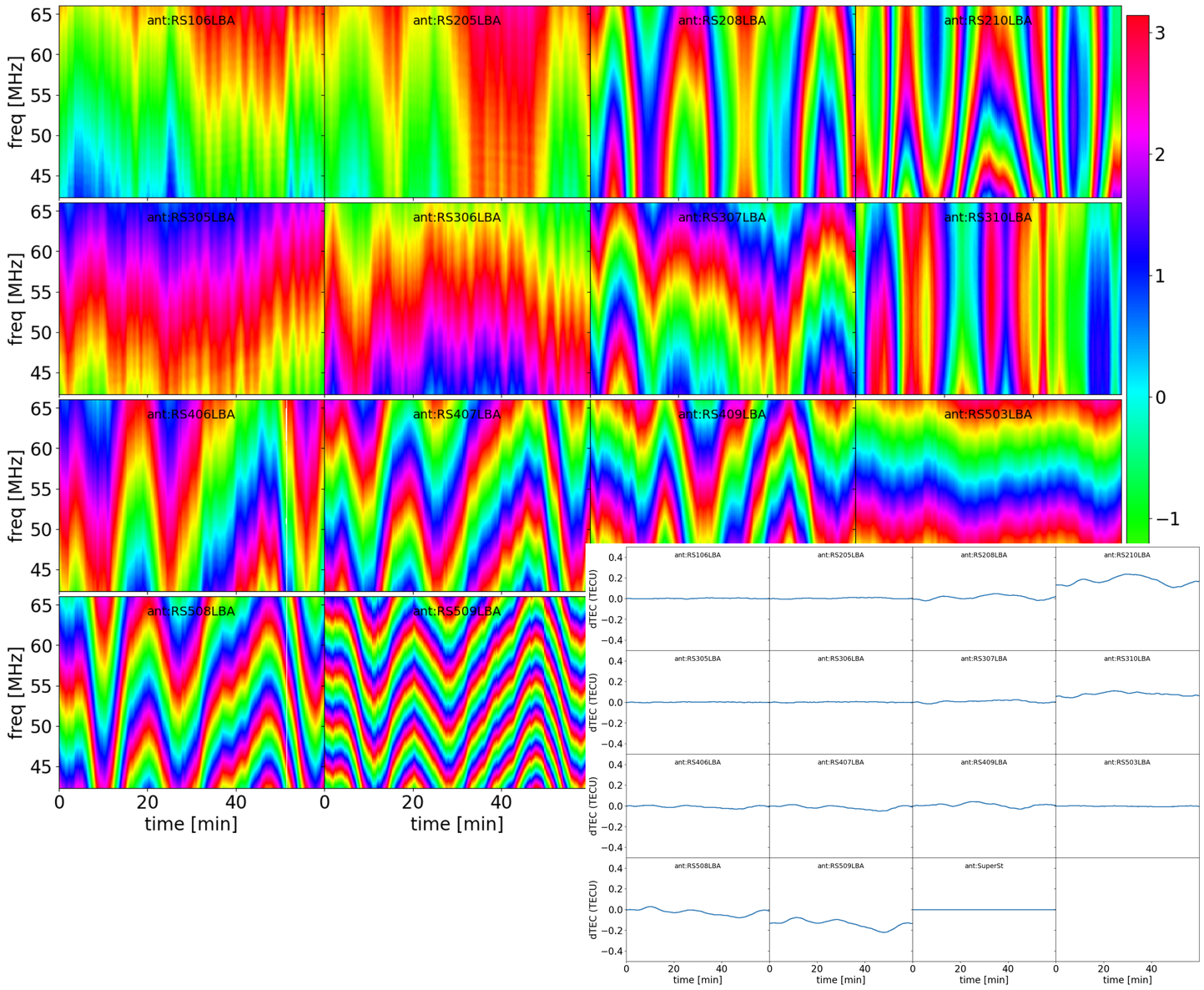
Calibration

Calibrator

Polarisation mis-alignment
Element beam
Bandpass
Faraday rotation
Ionospheric dispersion

Clock delays

$$K_{\text{clock}} = \begin{pmatrix} e^{i\phi} & 0 \\ 0 & e^{i\phi} \end{pmatrix}$$



Calibration

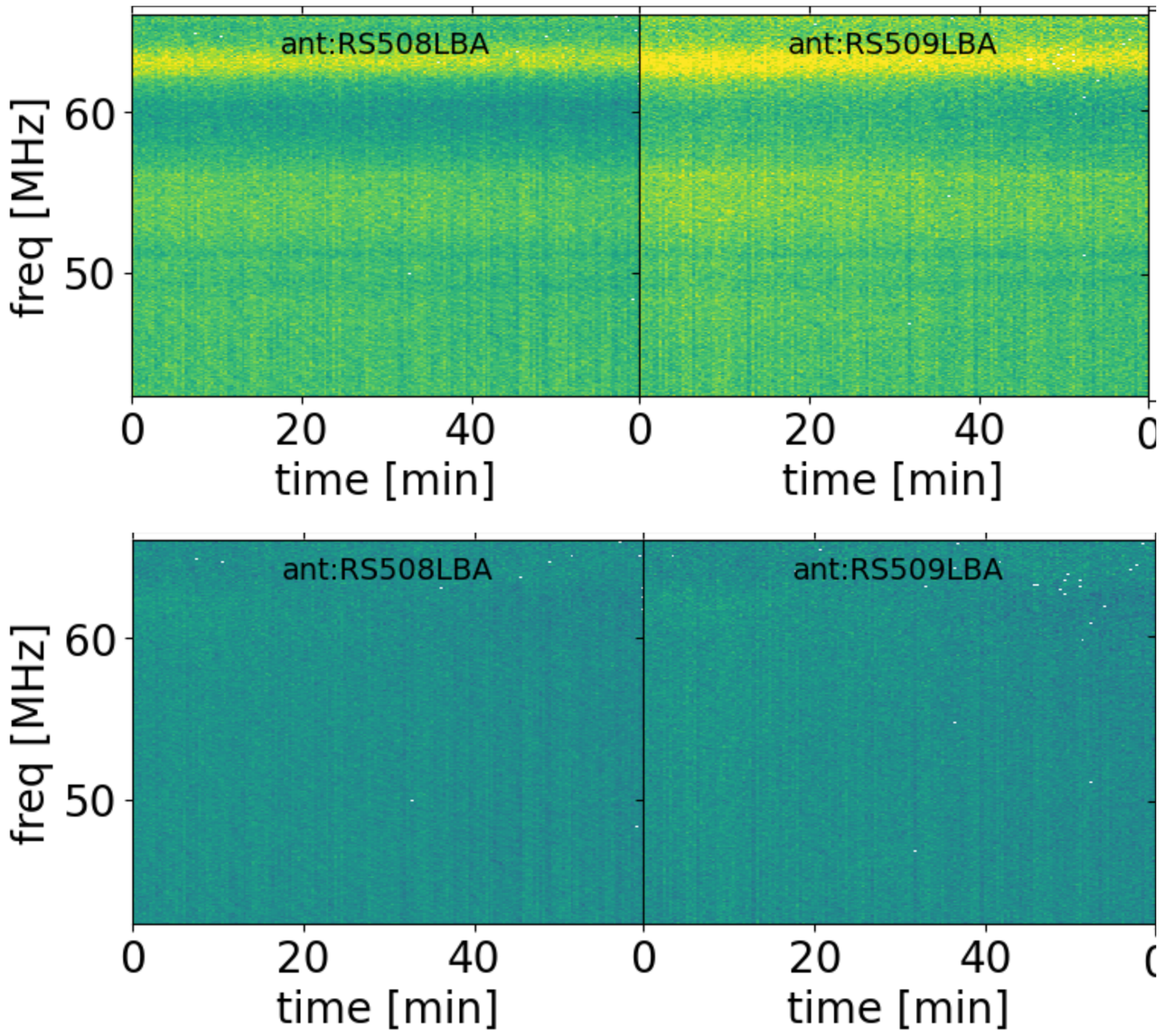
Calibrator

What we cannot model

Beam errors

Ionospheric scintillations

Station calibration errors



Calibration

Timesplit

Apply calibrator solutions to the target field.
These are direction independent (DI) solutions

- Clock + ionospheric dispersion delays
- Bandpass
- Polarisation alignment

Calibration

DD-parallel

Target field FR correction

DD correction (phase only)

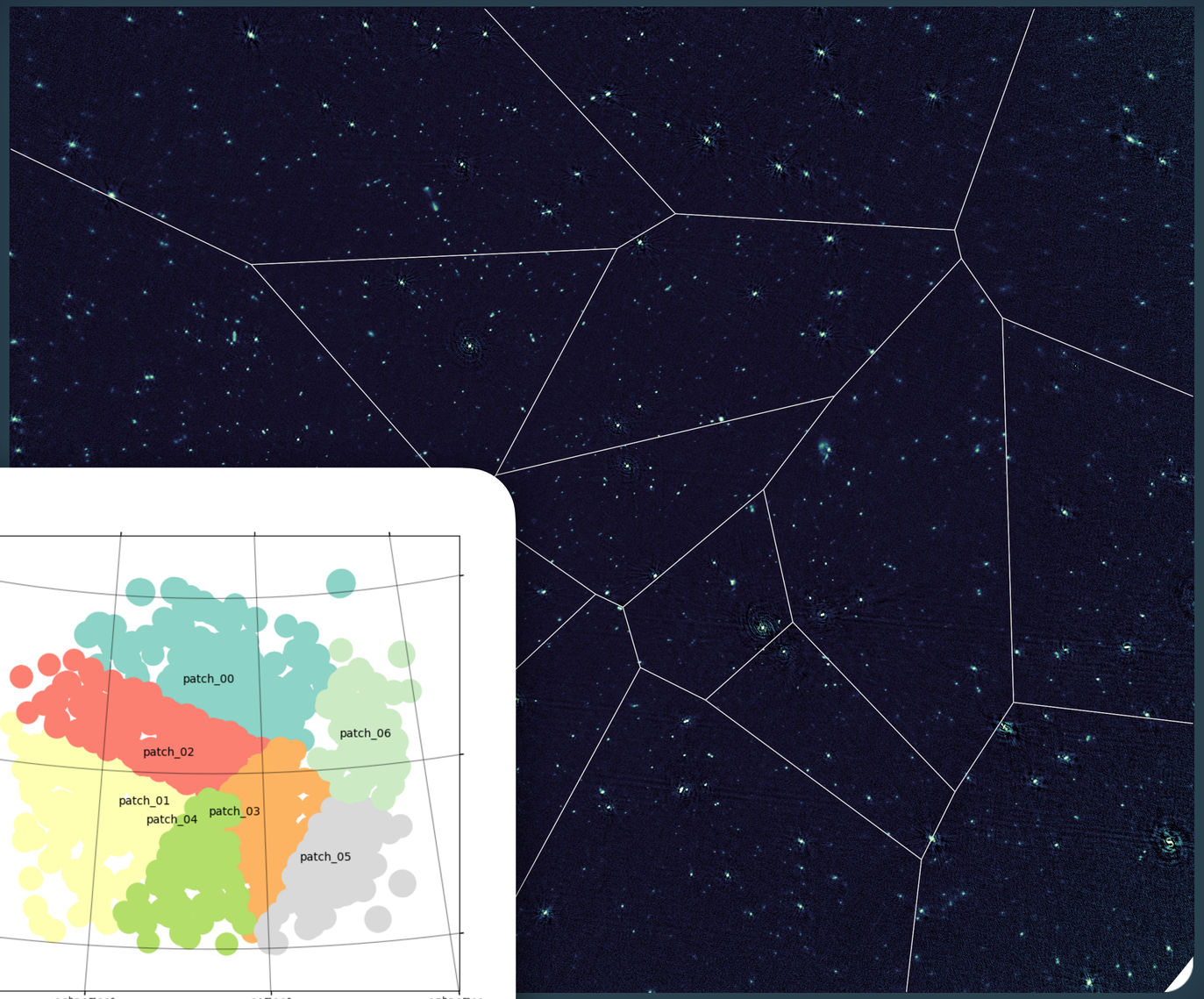
3C subtraction

Phase align on sub-field

Sidelobe subtraction

Second order beam DI

Image



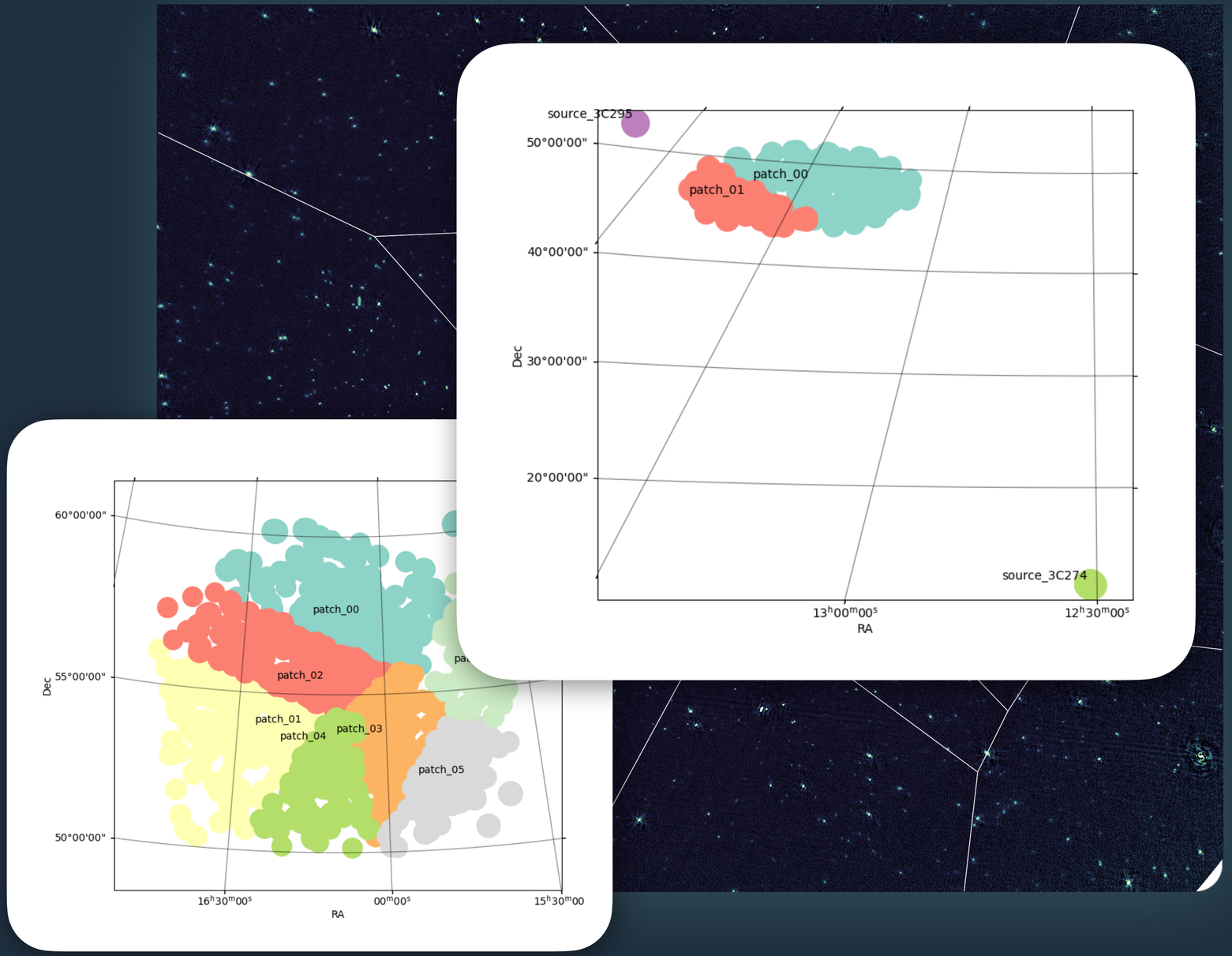
Calibration

DD-parallel

Target field FR correction
DD correction (phase only)

3C subtraction

Phase align on sub-field
Sidelobe subtraction
Second order beam DI
Image



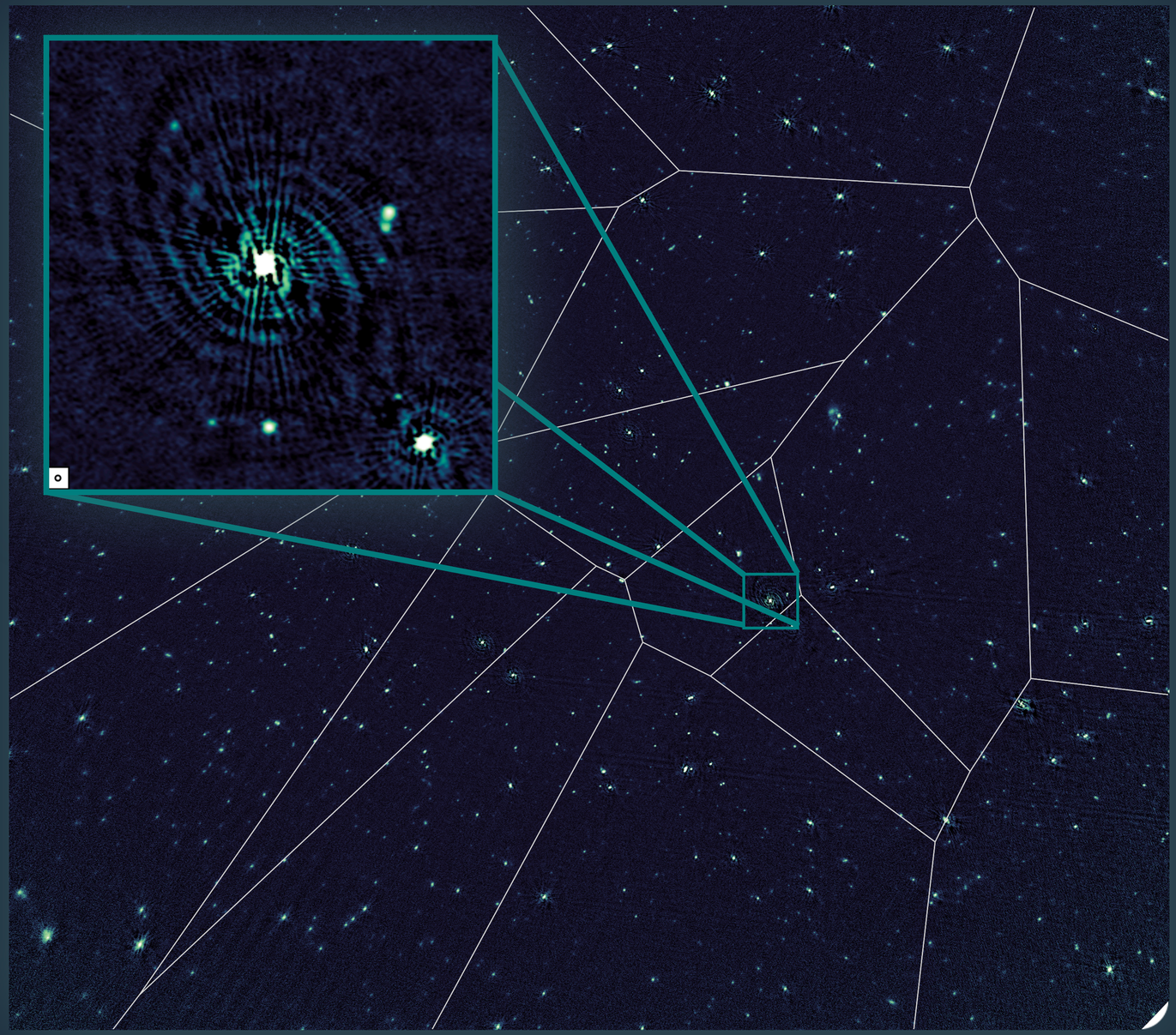
Calibration

DD-parallel

Target field FR correction
DD correction (phase only)
3C subtraction

Phase align on sub-field

Sidelobe subtraction
Second order beam DI
Image

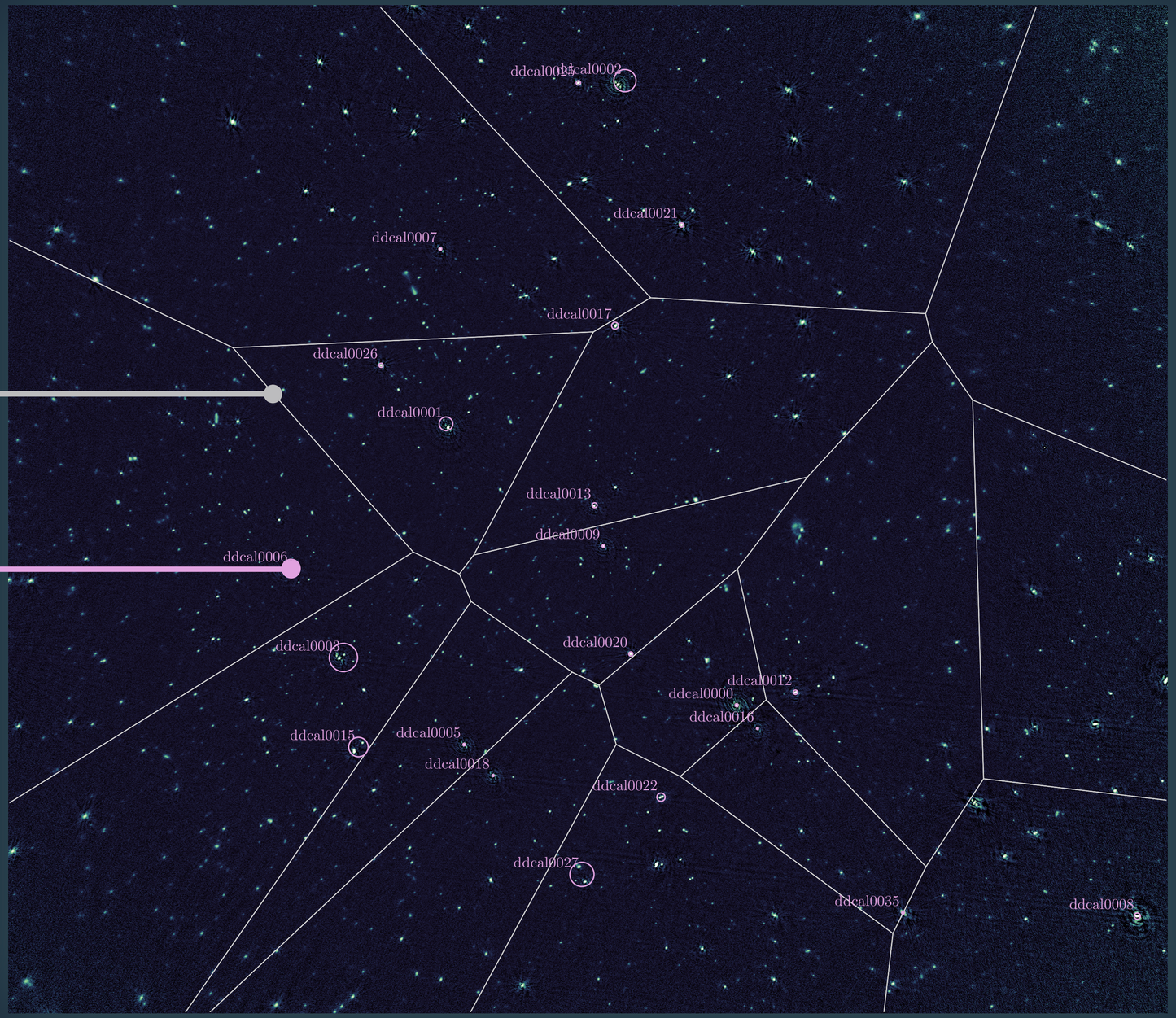


Calibration

DD-serial

DDparallel facets

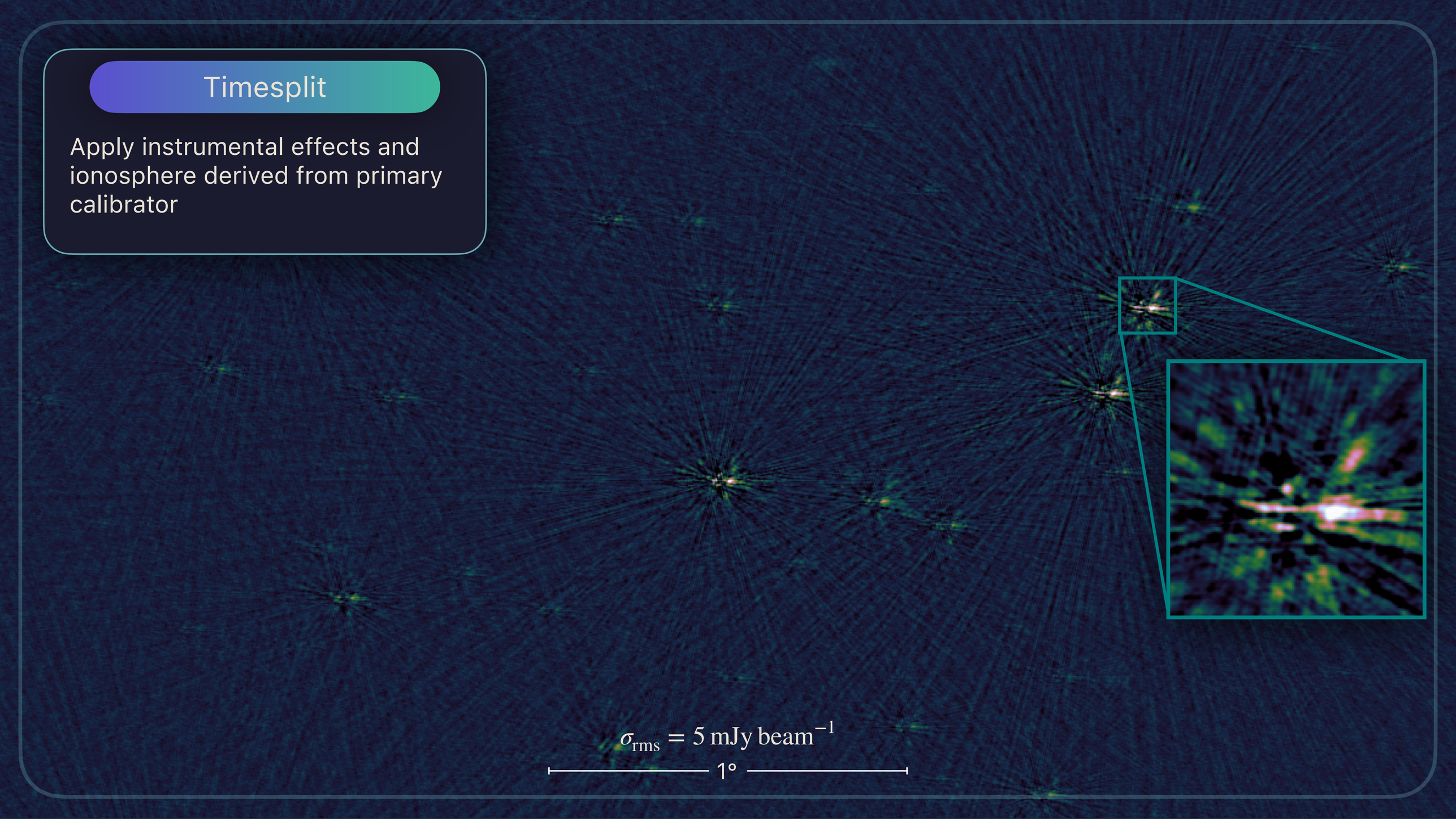
DDserial direction dependent calibrators



Solutions on top of DDparallel
Up to 30 directions
Multiple cycles of self-calibration
Phase + amplitude

Timesplit

Apply instrumental effects and ionosphere derived from primary calibrator

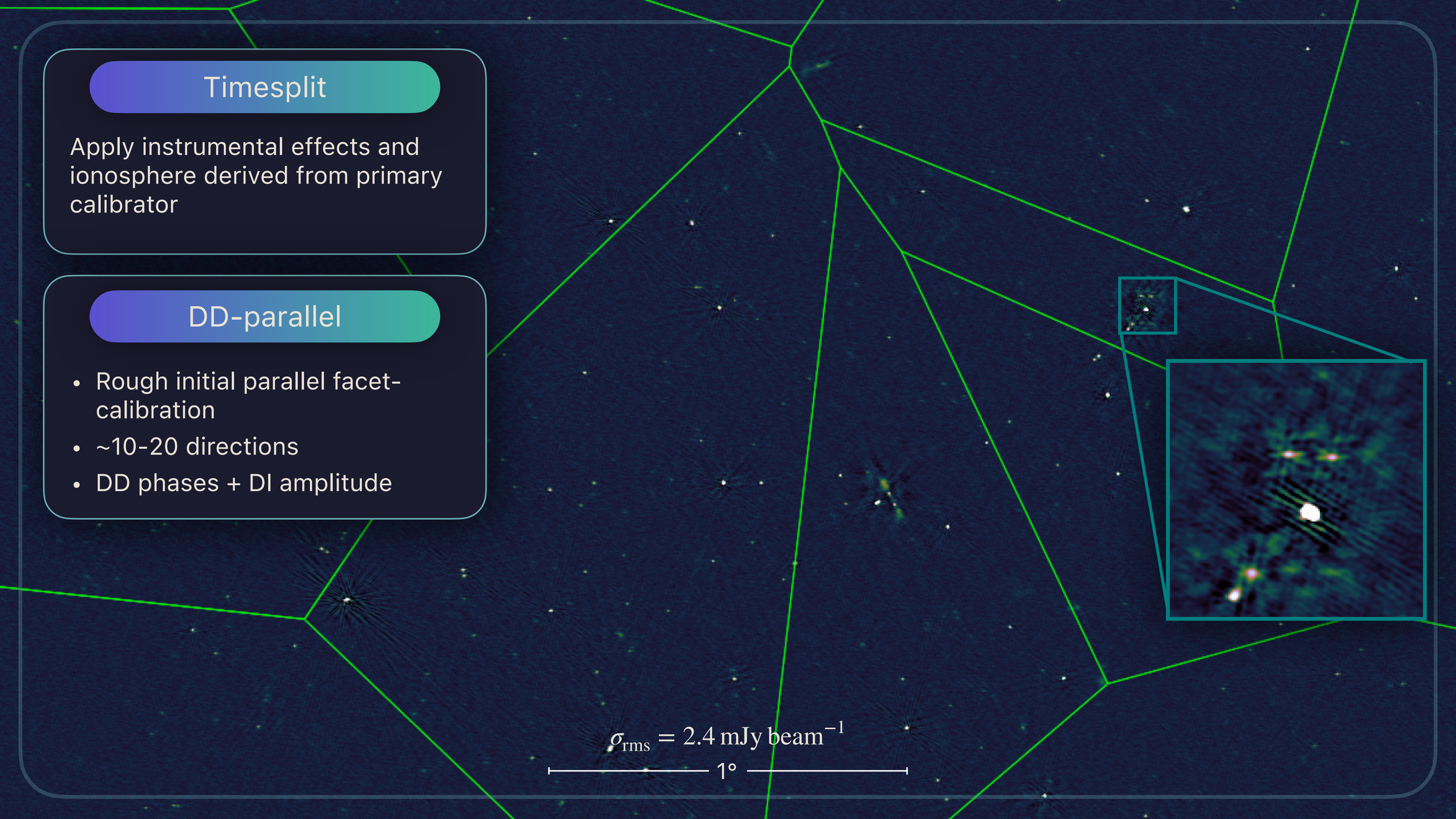


Timesplit

Apply instrumental effects and ionosphere derived from primary calibrator

DD-parallel

- Rough initial parallel facet-calibration
- ~10-20 directions
- DD phases + DI amplitude



Timesplit

Apply instrumental effects and ionosphere derived from primary calibrator

DD-parallel

- Rough initial parallel facet-calibration
- ~10-20 directions
- DD phases + DI amplitude

DD-serial

- Refine solutions with serial “peeling”
- ~40 directions
- DD phases **and** amplitudes

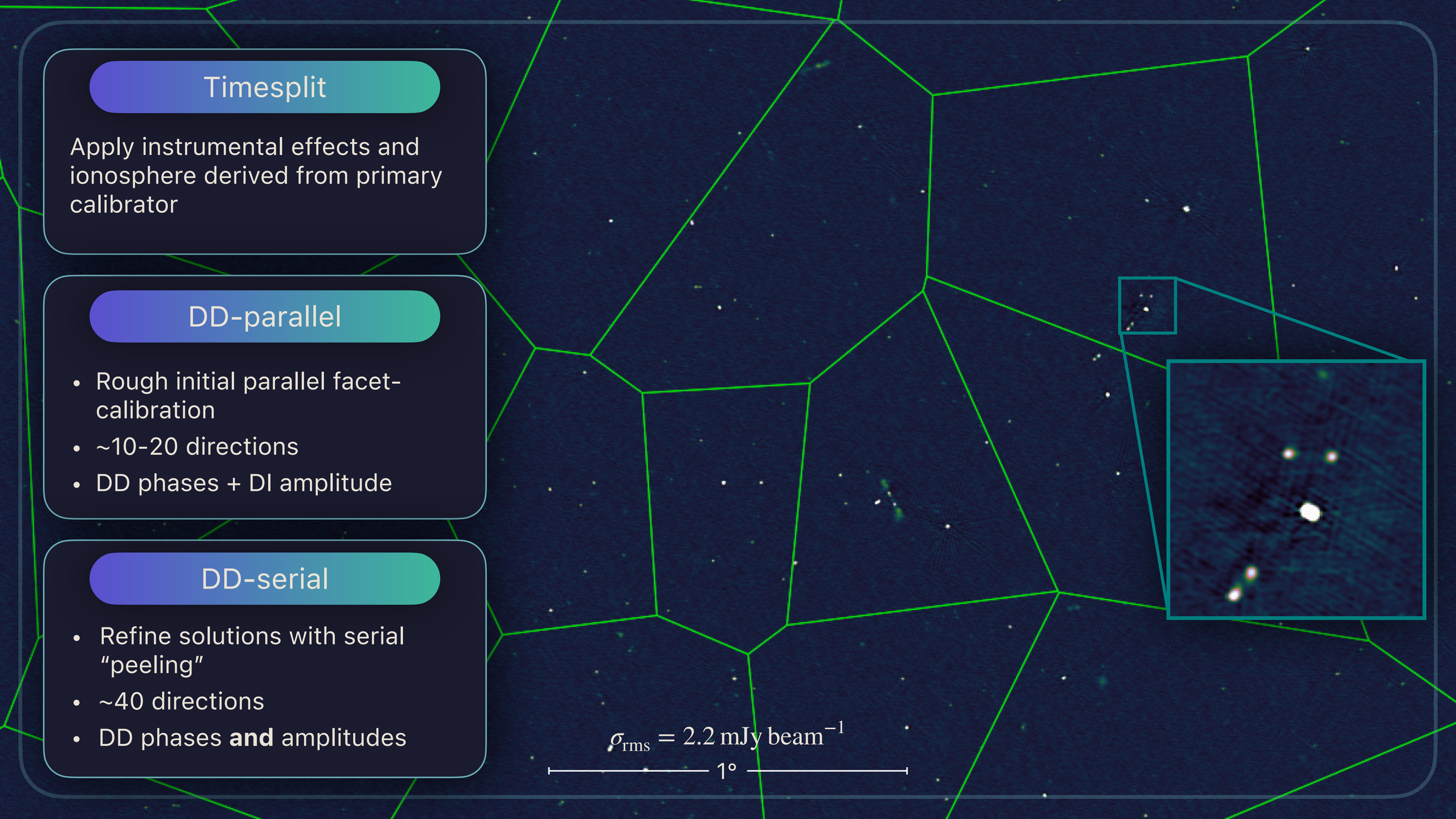


Image quality

Using the current pipeline set-up

Current pipeline

2022 Pipeline

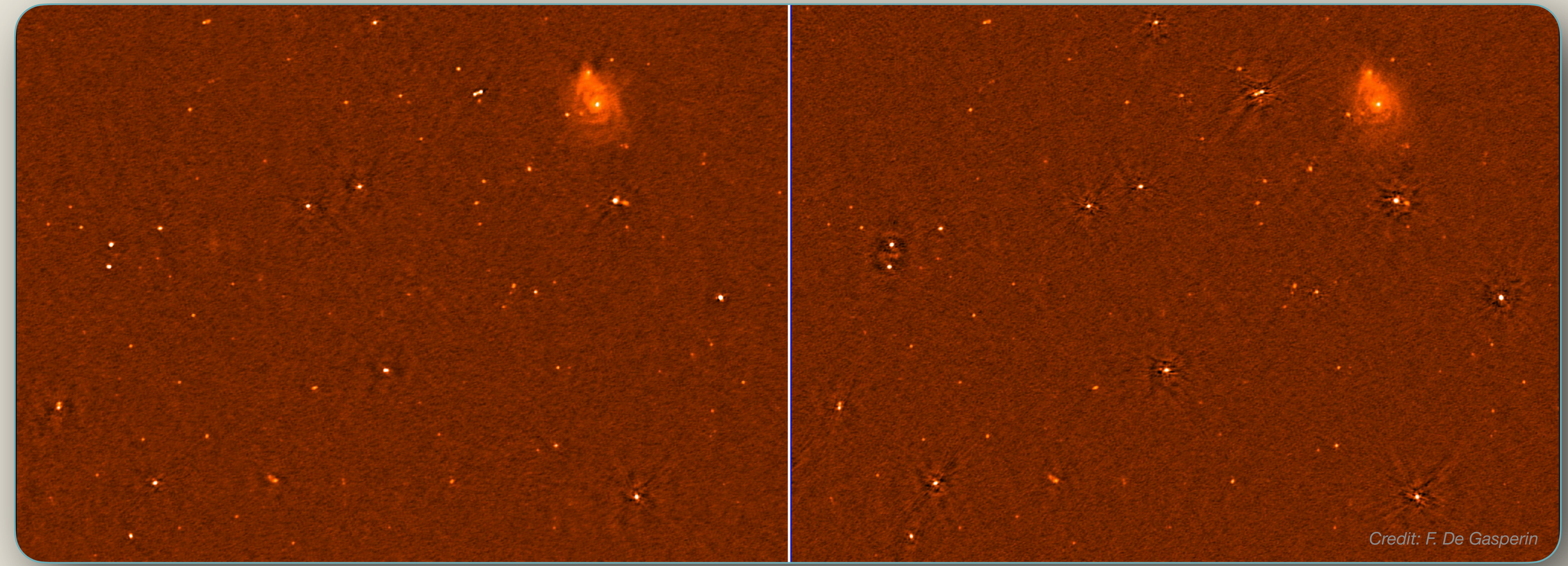
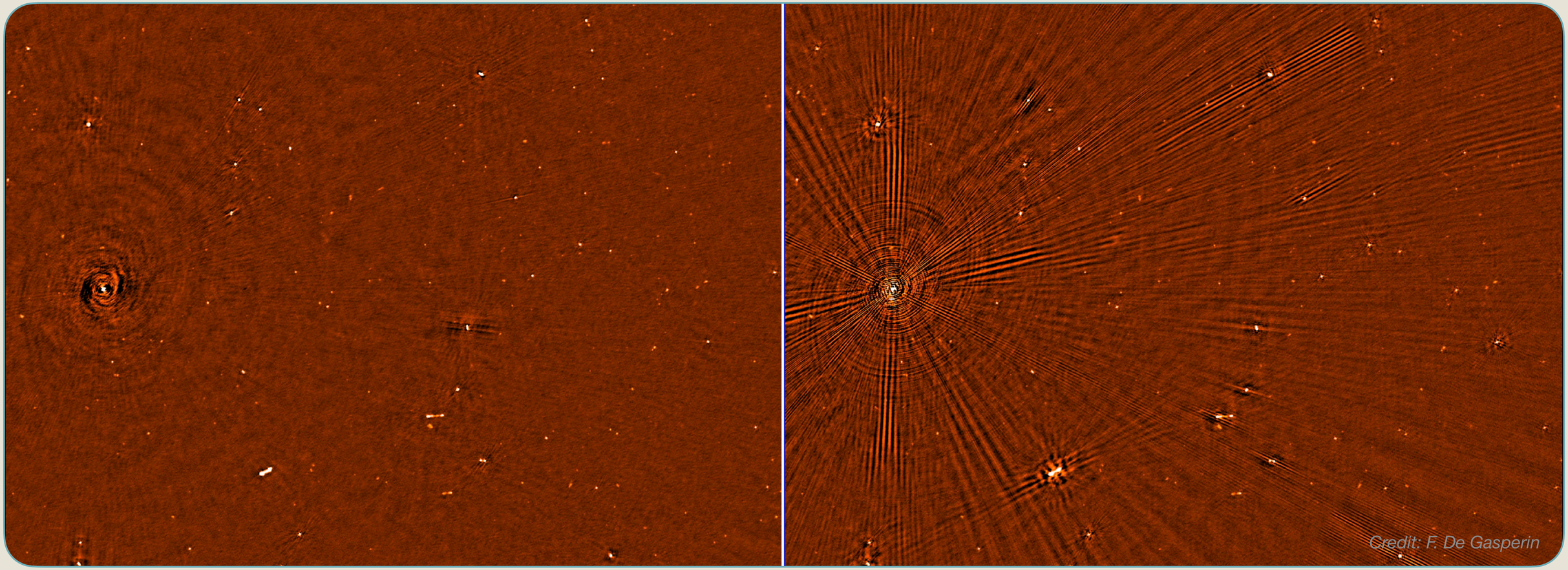


Image quality

Using the current pipeline set-up

Current pipeline

2022 Pipeline



Caveats

LBA & physical limitations

- The flux density scale is difficult to verify. Confirmed to be accurate down to 1 Jy flux densities
- Active ionosphere limits Calibratability of data. VLBI impossible with recent data
- For VLBI: possibly not enough compact bright sources for in-field calibration due to spectral turnover

LiLF

- Pipeline still under some development. Especially decametre (< 30 MHz) na VLBI parts of the pipeline
- Full data reduction is lengthy and resource heavy for long observations. Currently no implementation for deep observations
- Partial documentation (due to active development)

LOFAR 2.0

- With the LOFAR2.0 upgrade, the same depth can be reached with less observing time
- We will observe more SFG and high redshift galaxies at 50 MHz
- VLBI probably possible with calmer ionosphere

Using LiLF

Guide to the basics of library for low frequencies

Git structure

- LiLF/ *LiLF libraries*
  - lib_ms.py
  - lib_img.py
  - ⋮
- pipelines/ *Pipelines discussed above*
  - calibrator.py
  - ddparallel.py
  - ddserial.py
  - ⋮
- models/ *Initial models and cal. models*
  - 3CRR/
    - 3CXXX.skymodel
  - A-Team.skymodel
  - LoTSS-dr3.skymodel
- parsets/ *Default settings for DP3 and losoto calls*
  - ddparallel/
    - DP3-cor.parset
    - losoto-fr.parset
    - ⋮
  - ddserial/
    - DP3-cor.parset
    - losoto-fr.parset
    - ⋮
- scripts/ *Support scripts*

Standard pipeline outputs

Walker

Writes a `.walker` file recording each completed processing block. The `if_todo` context manager checks it before re-running a step, so a crashed multi-day pipeline restarts where it left off instead of from scratch.

```
cleaning # 0h 00m
concat_all # 3h 07m
scale_bp # 1h 28m
flag # 1h 39m
predict_all # 1h 32m
pre_iono # 7h 06m
cal_pa # 3h 27m
cal_fr # 5h 53m
cal_iono # 11h 56m
```

calibrator pipeline

```
MSs_concat_all = lib_ms.AllMSs(['concat_all.MS'], s, check_flags=False)

# flag bad stations, flags will propagate
with w.if_todo('flag'):
    logger.info("Flagging...")
    MSs_concat_all.run(f'DP3 {parset_dir}/DP3-flag.parset msin=$pathMS ant.baseline=\"{bl2flag}\" \
            aoflagger.strategy={parset_dir}/LBAdefaultwideband.lua',
            log='$nameMS_flag.log', commandType='DP3')

    # extend flags on bad time/freq slots
    logger.info('Remove bad time/freq stamps...')
    MSs_concat_all.run('flagonmindata.py -f 0.5 $pathMS', log='$nameMS_flagonmindata.log', commandType='python')
### DONE

# Predict cal concat_all.MS:MODEL_DATA
with w.if_todo('predict_all'):
    logger.info(f'Add model of {calname} from {os.path.basename(skymodel)} to MODEL_DATA...')
    MSs_concat_all.run(
        f"DP3 {parset_dir}/DP3-predict.parset msin=$pathMS pre.sourcedb={skymodel} pre.sources={calname}",
        log="$nameMS_pre.log", commandType="DP3")
```

Standard pipeline outputs

Scheduler

Queues shell commands for DP3 and WSClean with the `run()` command. Tries to detect the local cluster (e.g. Leiden, Hamburg) to size thread allocation.

We are currently implementing functionality to parallelise runs over multiple nodes with SLURM.

calibrator pipeline

```
MSs_concat_all = lib_ms.AllMSs(['concat_all.MS'], s, check_flags=False)

# flag bad stations, flags will propagate
with w.if_todo('flag'):
    logger.info("Flagging...")
    MSs_concat_all.run(f'DP3 {parset_dir}/DP3-flag.parset msin=$pathMS ant.baseline=\"{bl2flag}\" \
            aoflagger.strategy={parset_dir}/LBAdefaultwideband.lua',
            log='$nameMS_flag.log', commandType='DP3')

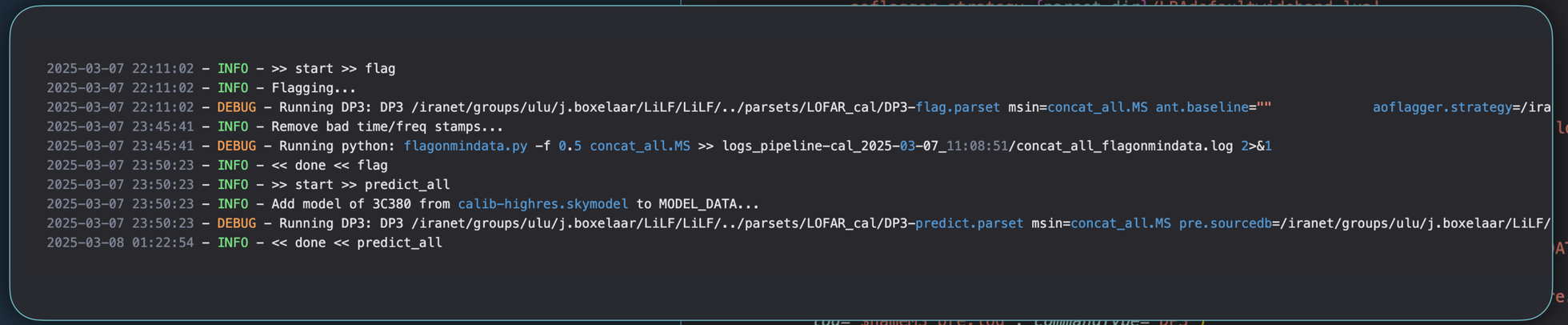
    # extend flags on bad time/freq slots
    logger.info('Remove bad time/freq stamps...')
    MSs_concat_all.run('flagonmindata.py -f 0.5 $pathMS', log='$nameMS_flagonmindata.log', commandType='python')
### DONE

# Predict cal concat_all.MS:MODEL_DATA
with w.if_todo('predict_all'):
    logger.info(f'Add model of {calname} from {os.path.basename(skymodel)} to MODEL_DATA...')
    MSs_concat_all.run(
        f"DP3 {parset_dir}/DP3-predict.parset msin=$pathMS pre.sourcedb={skymodel} pre.sources={calname}",
        log="$nameMS_pre.log", commandType="DP3")
```

Standard pipeline outputs

calibrator pipeline

pipeline-cal.logger



```
2025-03-07 22:11:02 - INFO - >> start >> flag
2025-03-07 22:11:02 - INFO - Flagging...
2025-03-07 22:11:02 - DEBUG - Running DP3: DP3 /iranet/groups/ulu/j.boxelaar/LiLF/LiLF/../parsets/LOFAR_cal/DP3-flag.parset msin=concat_all.MS ant.baseline="" aoflagger.strategy=/ira
2025-03-07 23:45:41 - INFO - Remove bad time/freq stamps...
2025-03-07 23:45:41 - DEBUG - Running python: flagonmindata.py -f 0.5 concat_all.MS >> logs_pipeline-cal_2025-03-07_11:08:51/concat_all_flagonmindata.log 2>&1
2025-03-07 23:50:23 - INFO - << done << flag
2025-03-07 23:50:23 - INFO - >> start >> predict_all
2025-03-07 23:50:23 - INFO - Add model of 3C380 from calib-highres.skymodel to MODEL_DATA...
2025-03-07 23:50:23 - DEBUG - Running DP3: DP3 /iranet/groups/ulu/j.boxelaar/LiLF/LiLF/../parsets/LOFAR_cal/DP3-predict.parset msin=concat_all.MS pre.sourcedb=/iranet/groups/ulu/j.boxelaar/LiLF/
2025-03-08 01:22:54 - INFO - << done << predict_all
```

```
MSs_concat_all = lib_ms.AllMSs(['concat_all.MS'], s, check_flags=False)

# flag bad stations, flags will propagate
with w.if_todo('flag'):
    logger.info("Flagging...")
    MSs_concat_all.run(f'DP3 {parset_dir}/DP3-flag.parset msin=$pathMS ant.baseline=\"{bl2flag}\" \
                                                                                    log', commandType='python')
                                                                                    ATA...')
                                                                                    e.sources={calname}",
```

Standard pipeline outputs

calibrator pipeline

```
concat.log                              concat_all_corFR.log          concat_pa.log
concat_all-phaseup-IONO_pre.log         concat_all_corIONO.log        concat_pa_pre.log
concat_all-phaseup-IONO_smooth.log      concat_all_corIONO_CS.log     concat_pa_solPA.log
concat_all-phaseup-IONO_sol.log         concat_all_corPA.log          h5_merger.log
concat_all-phaseup-preIONO_pre.log      concat_all_flag.log           losoto-bp-sub.log
concat_all-phaseup-preIONO_smooth.log   concat_all_flagonmindata.log  losoto-cal-bp.h5.log
concat_all-phaseup-preIONO_sol.log      concat_all_phaseup.log        losoto-fr.log
concat_all-preIONO_cor.log              concat_all_pre.log            losoto-iono-cs.log
concat_all-preIONO_sol.log              concat_all_smooth.log         losoto-iono.log
concat_all_beam.log                     concat_all_solBP.log          losoto-pa.log
concat_all_bpscale.log                  concat_all_solFR.log          losoto-preiono-cs.log
concat_all_bptemplate.log               concat_all_solIONO.log        losoto-preiono.log
concat_all_cor-preIONO.log              concat_all_weights.log        tar.log
```

logs/

```
MSs_concat_all = lib_ms.AllMSs(['concat_all.MS'], s, check_flags=False)

s will propagate

..")
P3 {parset_dir}/DP3-flag.parset msin=$pathMS ant.baseline=\"{bl2flag}\" \
ategy={parset_dir}/LBAdefaultwideband.lua',
flag.log', commandType='DP3')

time/freq slots
d time/freq stamps...')
agonmindata.py -f 0.5 $pathMS', log='$nameMS_flagonmindata.log', commandType='python')

S:MODEL_DATA
l'):
el of {calname} from {os.path.basename(skymodel)} to MODEL_DATA...')

MSs_concat_all.run(
    f"DP3 {parset_dir}/DP3-predict.parset msin=$pathMS pre.sourcedb={skymodel} pre.sources={calname}",
    log="$nameMS_pre.log", commandType="DP3")
```

Getting started

- Clone git repository: `git clone https://github.com/revoltek/LiLF.git`
- Set up singularity:
  1. Use Flocs apptainer or Docker from the LDS (not optimised for LiLF)

     ALTERNATIVE (PiLL singularity)

     ```
     singularity build pill-latest.simg docker://revoltek/pill:latest
     singularity shell -B /path/to/mount:/mount/to/path pill.simg
     ```

  2. Set Paths

     ```
     export PYTHONPATH=/opt/LiLF:$PYTHONPATH
     export PATH=/opt/LiLF/scripts:$PATH
     ulimit -n 8192
     ```

     Use the path pointing to your LiLF directory

LiLF Config

Branch: master

- During the summer, The code has been restructured to prepare for LOFAR 2 data and to accommodate a clearer workflow
- The `lilf.config` file including the steps is now required to run the pipeline
- Note: `preprocess` is renamed to `setup`

```
steps = [setup, calibrator, timesplit, [ddparallel], ddserial]
```

lilf.config

```
[LOFAR_preprocess]
fix_table = True
renameavg = True
keep_IS = True
backup_full_res = False
demix_sources =
demix_skymodel =
demix_field_skymodel = LOTSS-DR3
run_aoflagger = False
tar = True
data_dir =

[LOFAR_cal]
data_dir = data-bkp/
skymodel =
imaging = False
fillmissingedges = True
less_aggressive_flag = False
develop = False
use_GNSS = False
use_shm = False

[LOFAR_timesplit]
data_dir = data-bkp/
cal_dir =
ngroups = 1
initc = 0
fillmissingedges = True
apply_fr = False
no_aoflagger = False
ateam_clip =
use_GNSS = False

[LOFAR_ddparallel]
maxIter = 2
subfield =
subfield_min_flux = 20
```

LiLF tutorial / demo