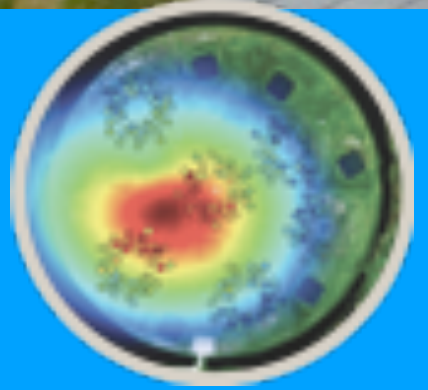
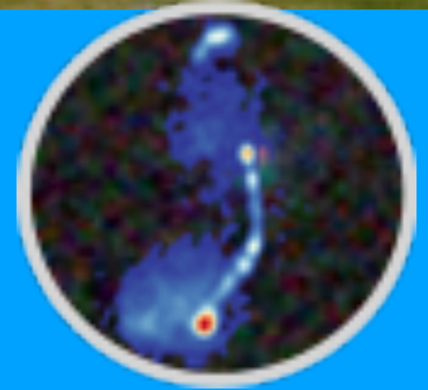
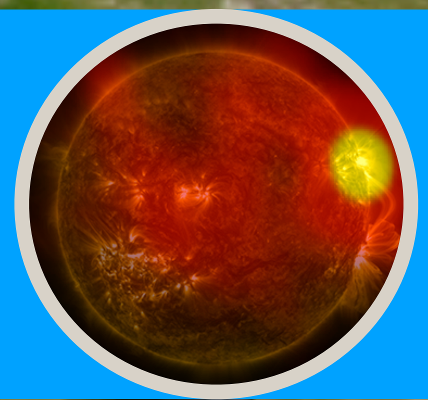
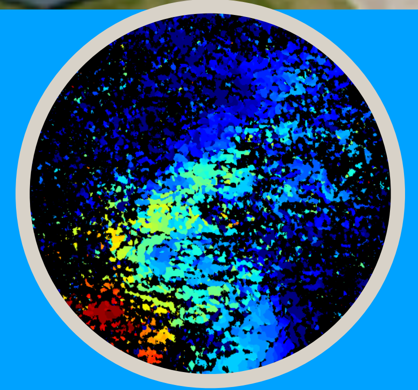
Rapthor

a Direction-Dependent Pipeline for LOFAR

David Rafferty

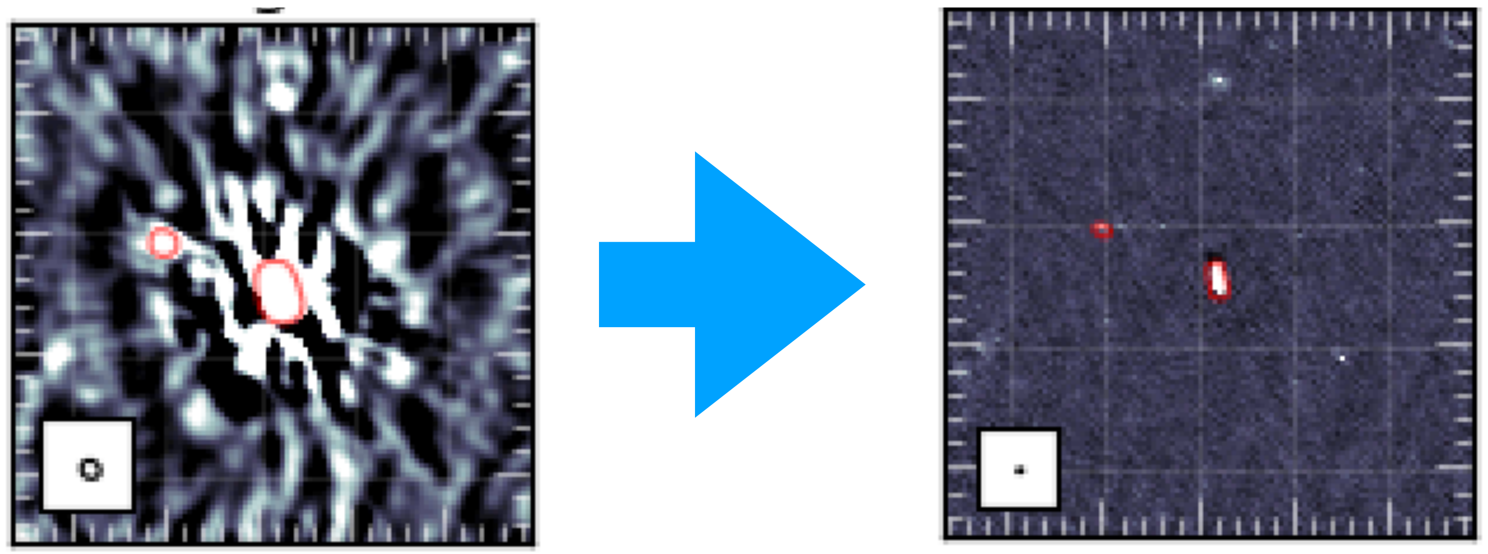



6th LOFAR Data School

Introduction

- Direction-dependent calibration and imaging is critical to obtaining high-quality wide-field images



- But this is quite involved and complicated!

➡ Direction-dependent pipelines are needed to automate the processing

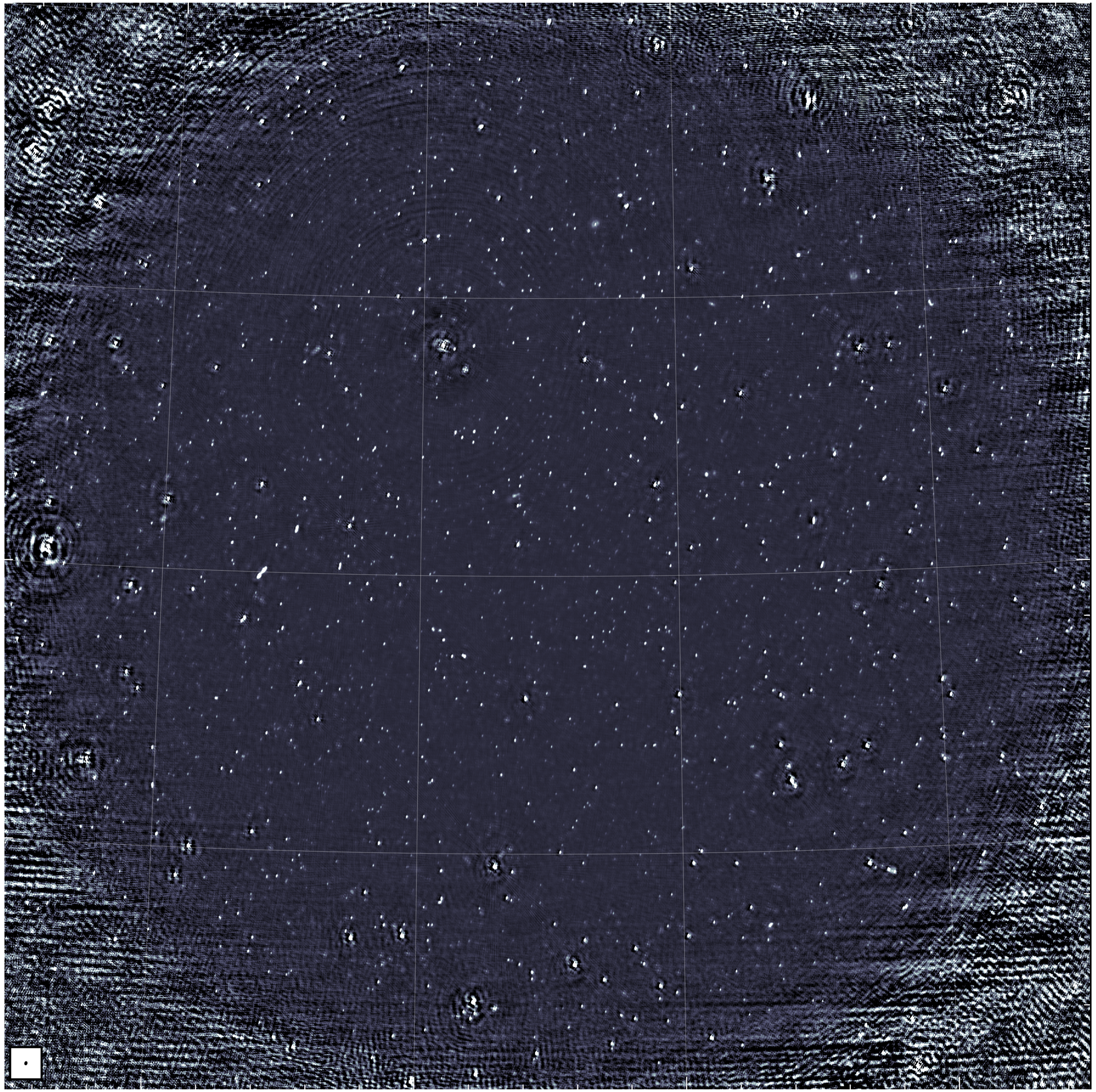


Image credits: A. Corstanje, F. Sweijen, C. Van Eck, P. Zucca

Rapthor

- Rapthor does direction-dependent calibration and imaging (HBA only for now)
- Mostly Python, but uses the Common Workflow Language (CWL) for pipelines
- Allows distribution over cluster nodes, resuming of interrupted jobs (due to node failure, etc.), supports GPUs, MPI, and more
- Uses DPPP for calibration and WSClean+IDG for imaging
- Available from GitHub at https://github.com/darafferty/rapthor
  - Warning: beta software — not yet feature complete or fully tested!

Direction-dependent Calibration

- Calibration is performed in multiple directions ("patches") simultaneously
- Each patch contains at least one bright source
- Each patch gets a single calibration solution

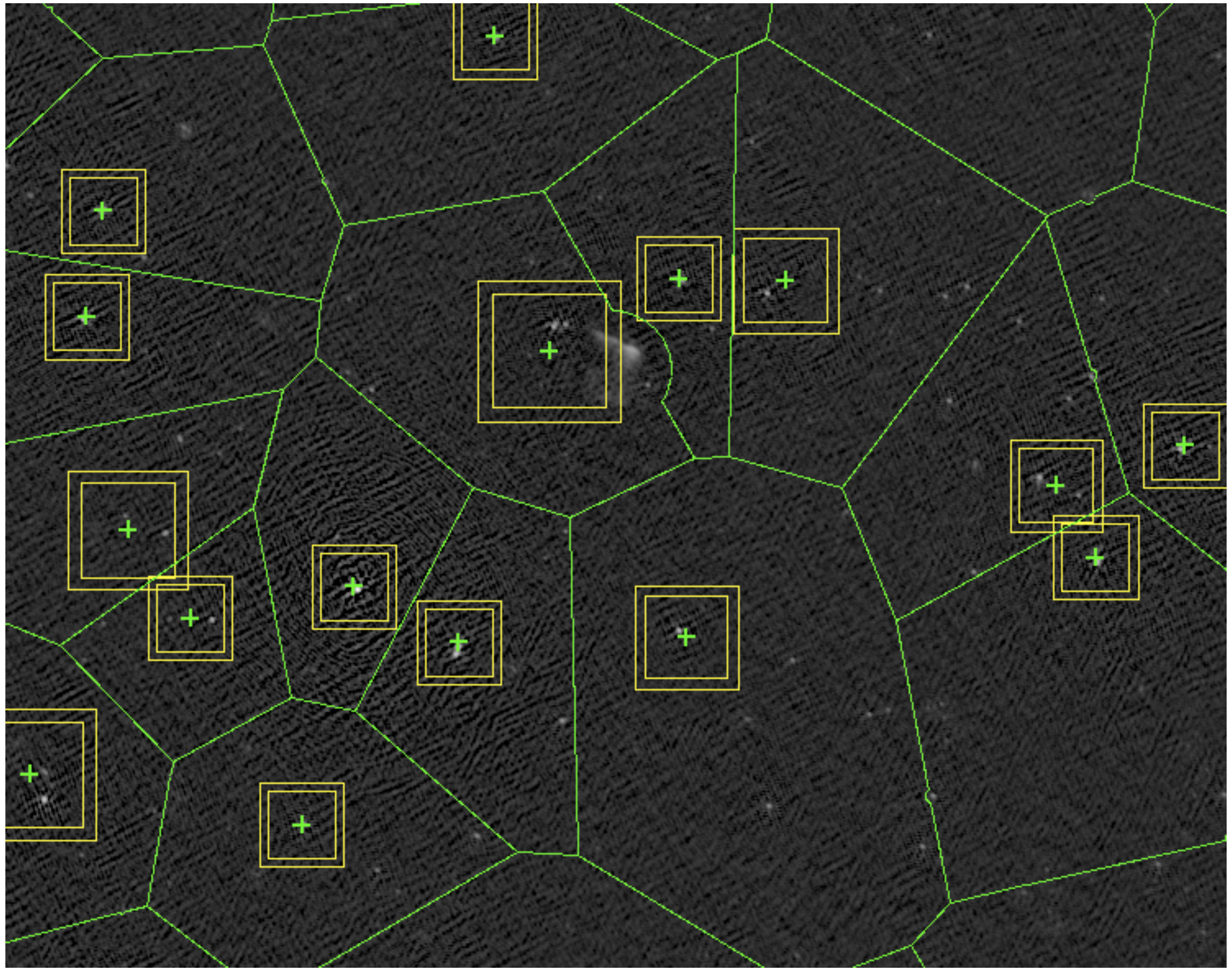


Image credits: A. Corstanje, F. Sweijen, C. Van Eck, P. Zucca

Direction-dependent Calibration

Ionospheric effects

- Ionospheric effects are (primarily) phase effects on timescales of < a minute that are smooth in frequency ($\propto \nu^{-1}$)
  - So solve for phases on short timescales and enforce smooth frequency behavior
- Unfortunately, residual clock errors and other dispersive delays are usually present as well — hard to fit for TEC alone

Phase corrections

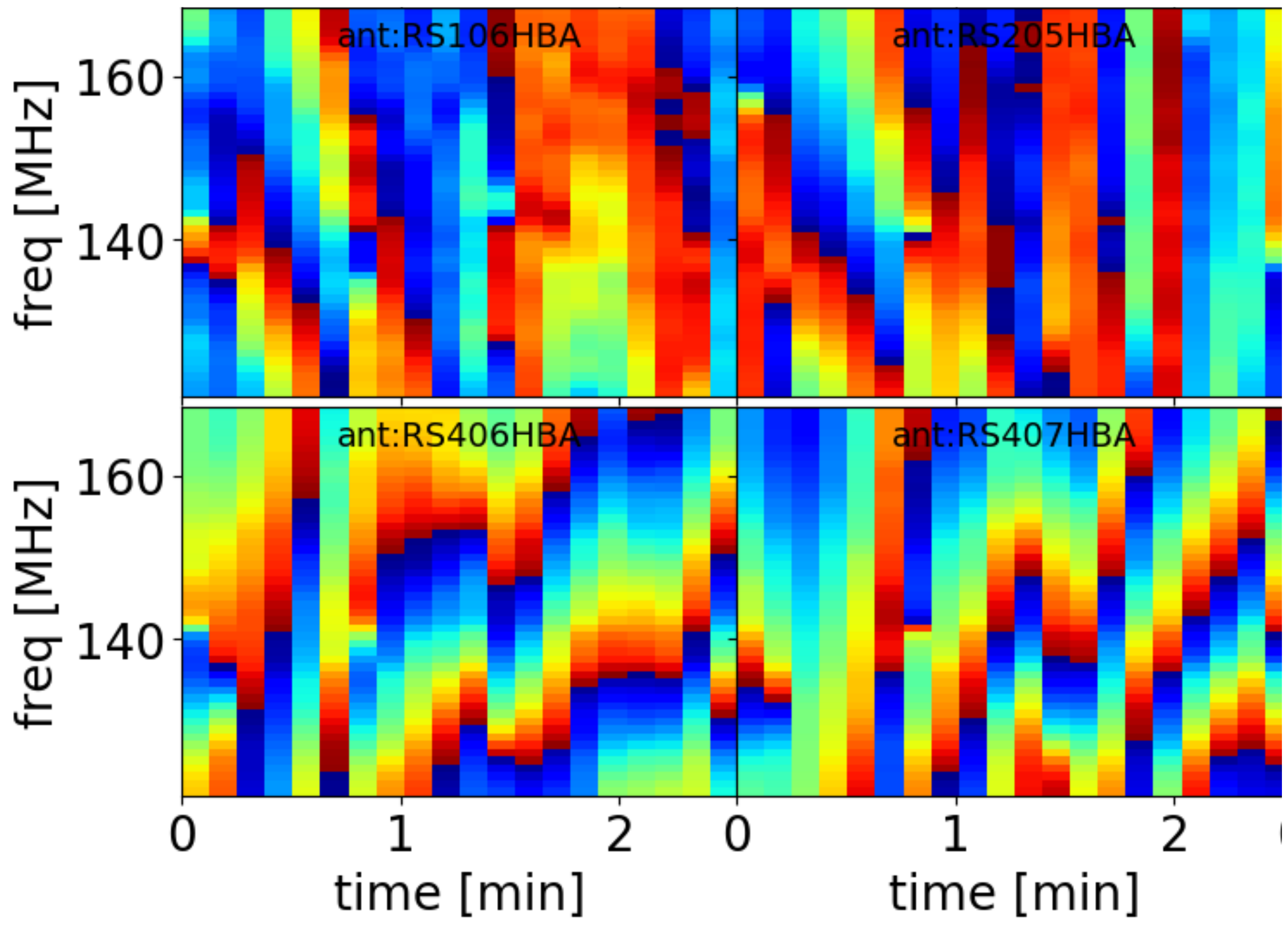


Image credits: A. Corstanje, F. Sweijen, C. Van Eck, P. Zucca

Direction-dependent Calibration

Beam effects

- Residual beam effects are (primarily) amplitude effects on timescales of tens of minutes that are smooth in frequency
  - So solve for amplitudes on long timescales and enforce smooth frequency behavior

Amplitude corrections

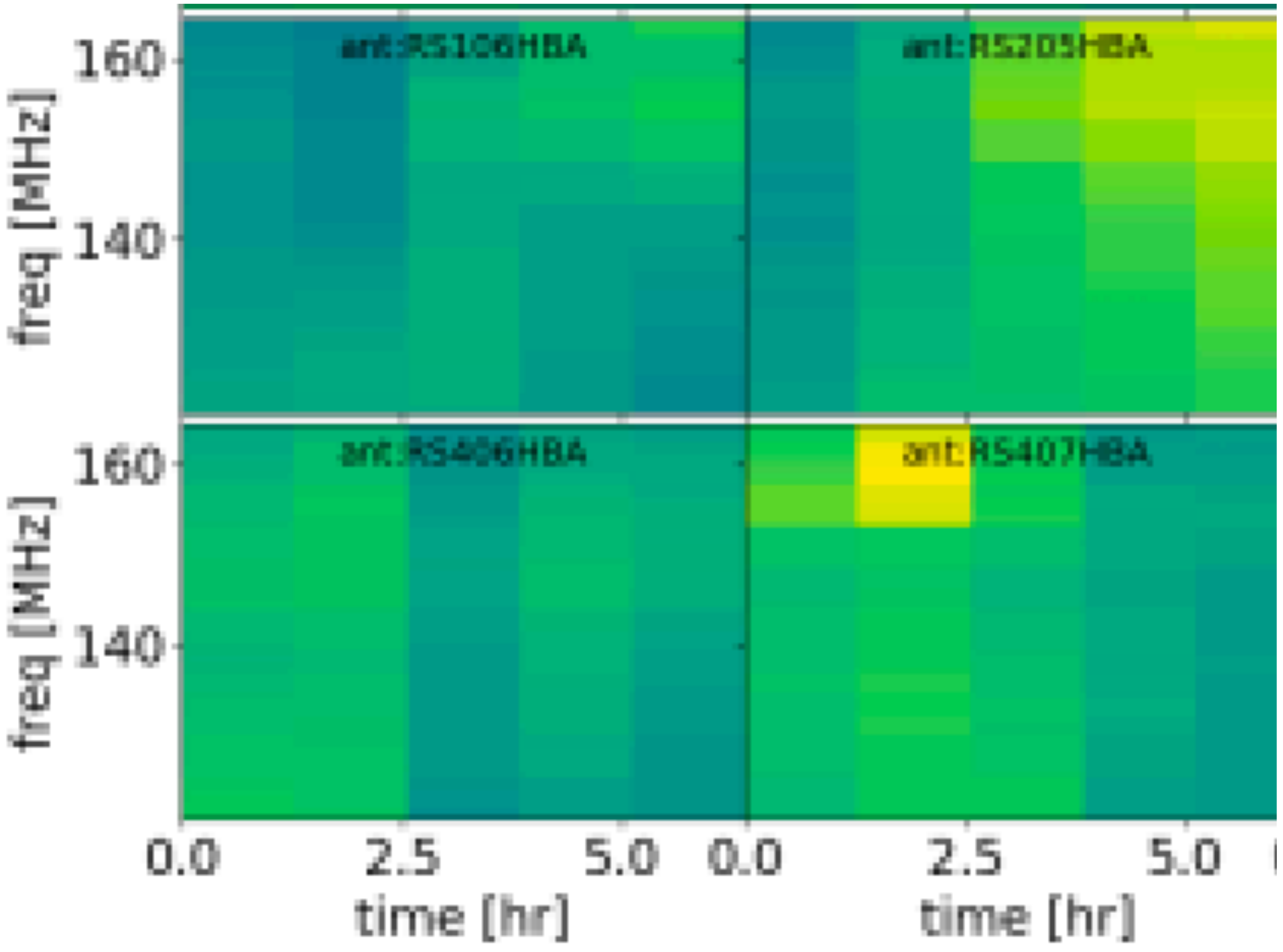


Image credits: A. Corstanje, F. Sweijen, C. Van Eck, P. Zucca

Direction-dependent Imaging

- The solutions must be applied during imaging to account for their direction-dependent nature:

1. Divide the field into facets and image each facet with a single, constant correction (Factor approach)

2. Image the full field using smooth 2-D screens of corrections (Rapthor approach)

    - More physical (spatially smooth), adds additional spatial constraints that can improve SNR, should be better for extended emission that would span multiple facets

2-D Screens

- Screens are derived separately for each station
- A Karhunen-Loeve (KL) decomposition is done to fit the solutions with smooth screens
  - The KL screens encapsulate the turbulent structure of the ionosphere (i.e., more structure on larger scales)

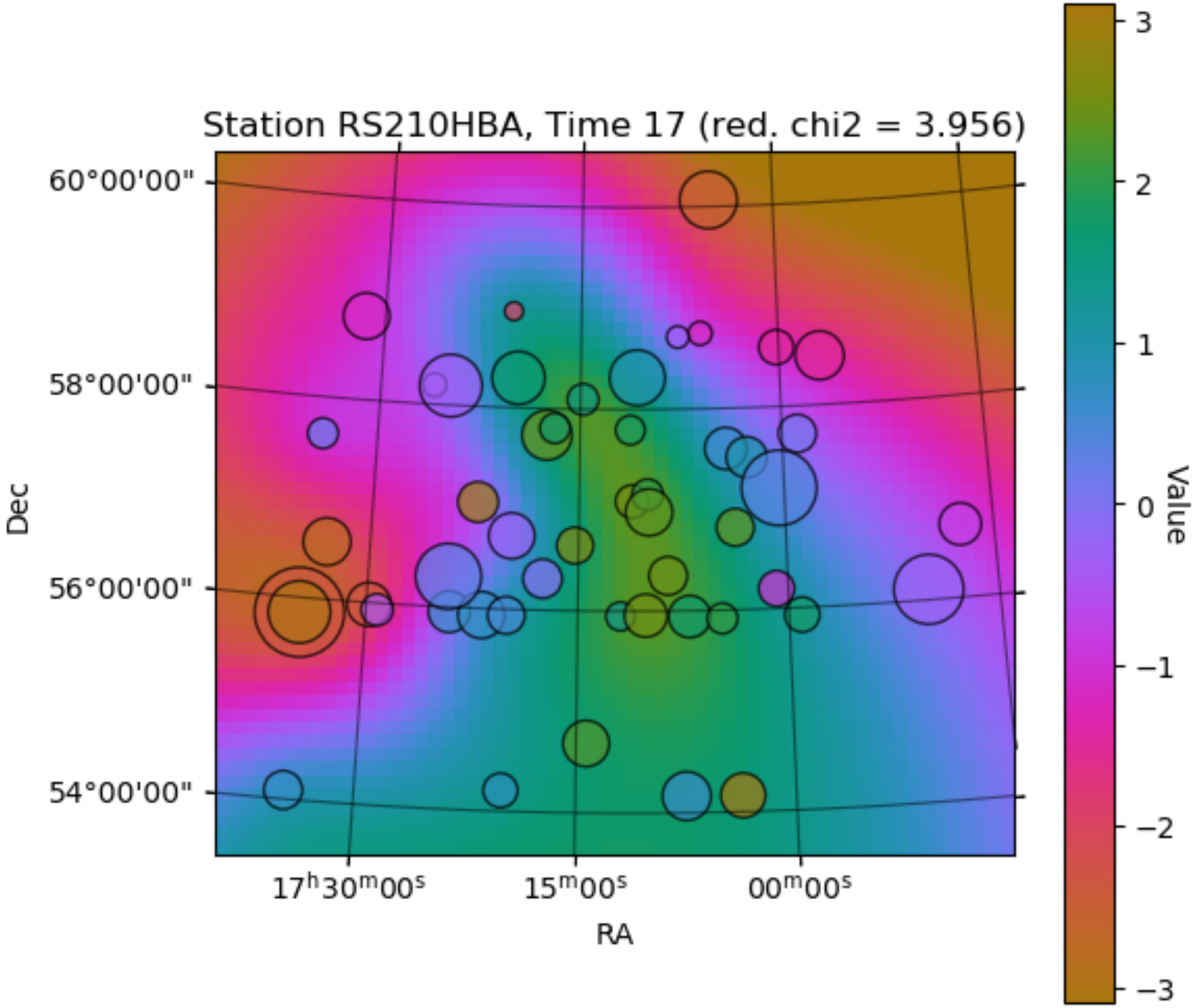


Image credits: A. Corstanje, F. Sweijen, C. Van Eck, P. Zucca

Rapthor Processing Overview

- Divides sky model into patches with a bright source or group of sources in each
- Calibrates over all patches simultaneously
- Fits smooth, 2-D screens to the calibration solutions
- Images with the screens to correct for direction-dependent effects
- Repeats as needed (for self calibration)

Data Preparation

- Data should first be processed through Prefactor (https://github.com/lofar-astron/prefactor)
- The initial-subtract Prefactor pipeline can be run to:
  - Image the field at medium and low resolution to make initial models of the sources
  - Subtract sources at large radii (beyond the first null of the primary beam) from the uv data

Tutorial

Tutorial: General Info

- Hands on session scheduled for March 25th, starting at 13:40 CET
- Slack channel: #t8-rapthor
- Use the **`rapthor_data_v2.tar`** data file
- Dockerfile was posted to #t8-rapthor in case you need to build it yourself
- CEP3 users only: please copy **`rapthor.sif`** again

Tutorial: Input Data

- For this tutorial, the data are in the **`rapthor_data_v2.tar`** file:

```
$ tar xvf rapthor_data_v2.tar
$ ls rapthor_data
simulated_data_1.ms  simulated_data_4.ms  rapthor_strategy.py
simulated_data_2.ms  simulated_data_5.ms  skymodel.txt
simulated_data_3.ms  rapthor.parset
```

measurement sets (each ~ 2.5 min, 48 MHz)

strategy file

sky model

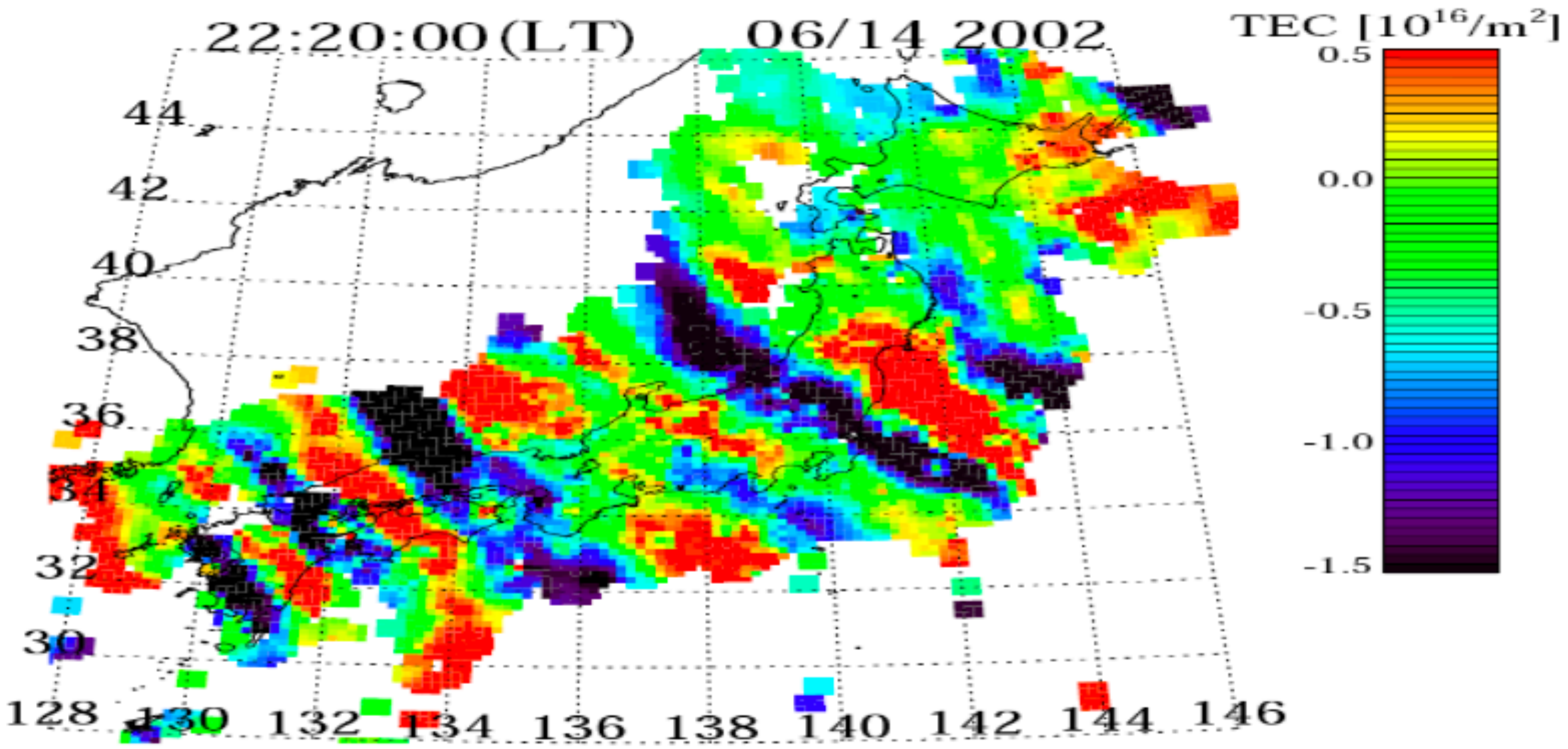
parset

- Data are from a simulation (made with LoSiTo) of five 5-Jy point sources, with 250 subbands (= 48 MHz) concatenated together, of 2.5 minutes each, spaced out over 5 hours total (with ~ 1 hour gap between each observation)

- Averaged to ≈ 0.1 MHz per channel (500 channels total) and 8 seconds per time slot

Tutorial: Input Data Simulation

- Ionosphere effects simulated as a TID
- Amplitude of 0.05 TECU
- Wavelength of 200 km
- Speed of 500 km/s



see Maaijke Mevius' talk

Tutorial: The Rapthor Parset

- The parset is divided into sections (`[global]`, `[calibration]`, etc.)
- See https://github.com/darafferty/rapthor/blob/master/examples/rapthor.parset for a full description of all the parameters
- An example parset that you can use for this tutorial is the **`rapthor.parset`** file in the **`rapthor_data`** directory
- For most parameters, the default values will be fine

Tutorial: The Rapthor Parset

- Change to the directory with the data:
  - ```
    $ cd /path/to/rapthor_data
    ```
- Edit **rapthor.parset** and change the following paths:
  - ```
    [global]
    dir_working = /path/to/rapthor_data/run1
    input_ms = /path/to/rapthor_data/*3.ms
    input_skymodel = /path/to/rapthor_data/skymodel.txt
    apparent_skymodel = /path/to/rapthor_data/skymodel.txt
    strategy = /path/to/rapthor_data/rapthor_strategy.py
    ```

Note the “3” — this tells it to just use the third MS file (at middle of observation)

Tutorial: Input Sky Model

- The input sky model (**`skymodel.txt`**) is the same one that was used for the simulations:

Note — no need to edit this file

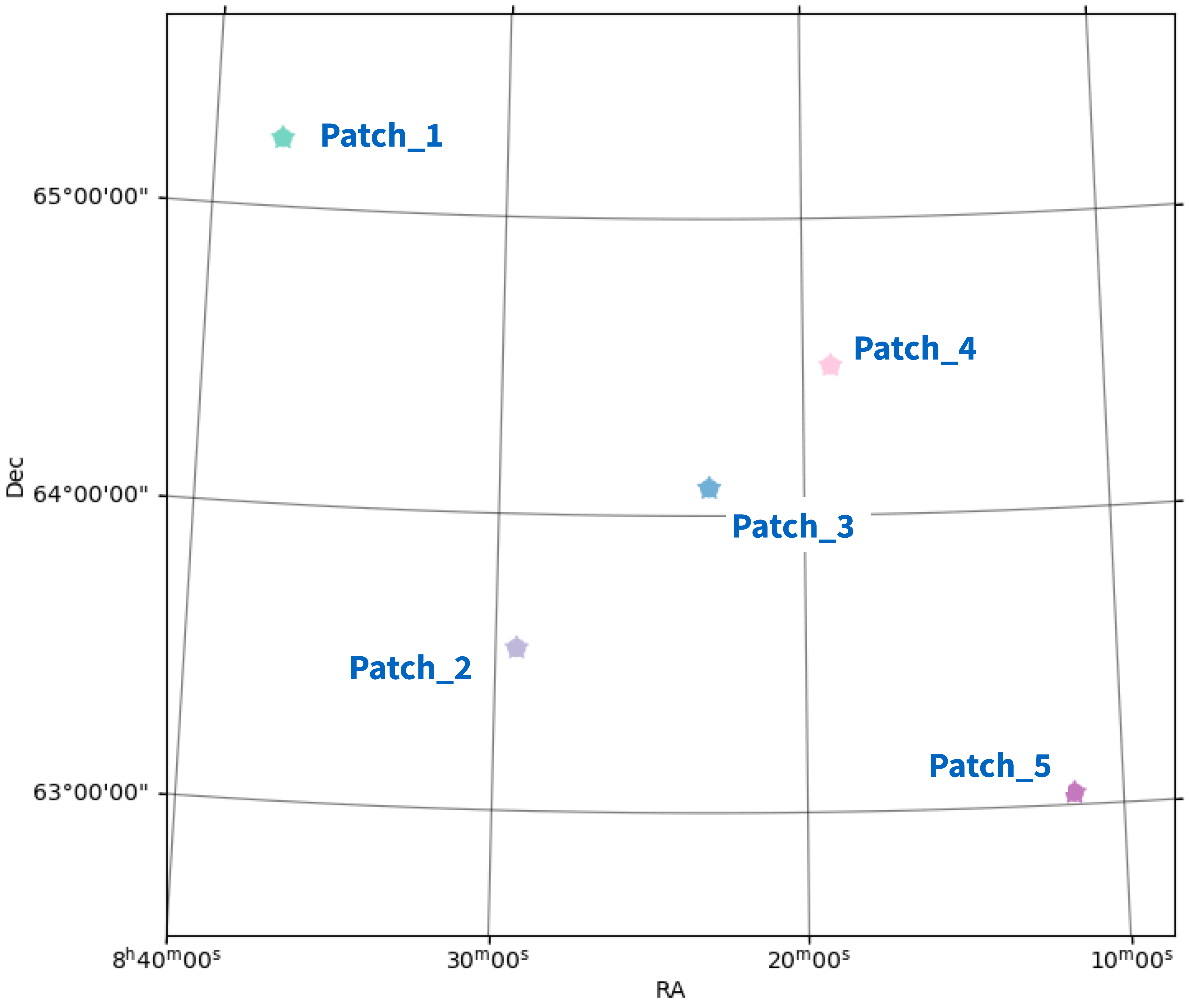
```
• FORMAT = Name, Type, Patch, Ra, Dec, I, …
   , , Patch_1, 8:37:42.9518, 65.13.47.4993
   , , Patch_2, 8:29:19.4111, 63.32.51.7385
   , , Patch_3, 8:23:07.1265, 64.05.34.771
   , , Patch_4, 8:19:04.7255, 64.30.13.5853
   , , Patch_5, 8:11:33.4617, 63.02.23.3212
s1, POINT, Patch_1, 8:37:42.9518, 65.13.47.4993, 5.0, …
s2, POINT, Patch_2, 8:29:19.4111, 63.32.51.7385, 5.0, …
s3, POINT, Patch_3, 8:23:07.1265, 64.05.34.771, 5.0, …
s4, POINT, Patch_4, 8:19:04.7255, 64.30.13.5853, 5.0, …
s5, POINT, Patch_5, 8:11:33.4617, 63.02.23.3212, 5.0, …
```

Patch centers

All are 5.0 Jy sources

Calibration patch names

Tutorial: Input Sky Model



Tutorial: Processing Strategy

- The strategy is set by the strategy file (**`rapthor_data/rapthor_strategy.py`**):

- ```
  """
  Script that defines a user strategy
  """
  strategy_steps = []
  max_selfcal_loops = 1
  for i in range(max_selfcal_loops):
      strategy_steps.append({})

      strategy_steps[i]['do_calibrate'] = True
      strategy_steps[i]['do_slowgain_solve'] = True
      strategy_steps[i]['do_image'] = False
      strategy_steps[i]['target_flux'] = 0.75
      …
  ```

Note — no need to edit this file
  ```

Tutorial: Starting Rapthor

- Enter the Singularity or Docker image, change to **`rapthor_data/`**
- Run Rapthor with the parset:
  - `$ rapthor -v rapthor.parset`

    = use **verbose mode**
- Rapthor will start by making the working directory (**`rapthor_data/run1`**), checking the input files and sky model, and starting the first pipeline (calibration)

```
Workflow Progress  38%|██████████          | 3/8 (0 failures) [00:29<00:49, 0.10 jobs/s]
```

Tutorial: Calibration

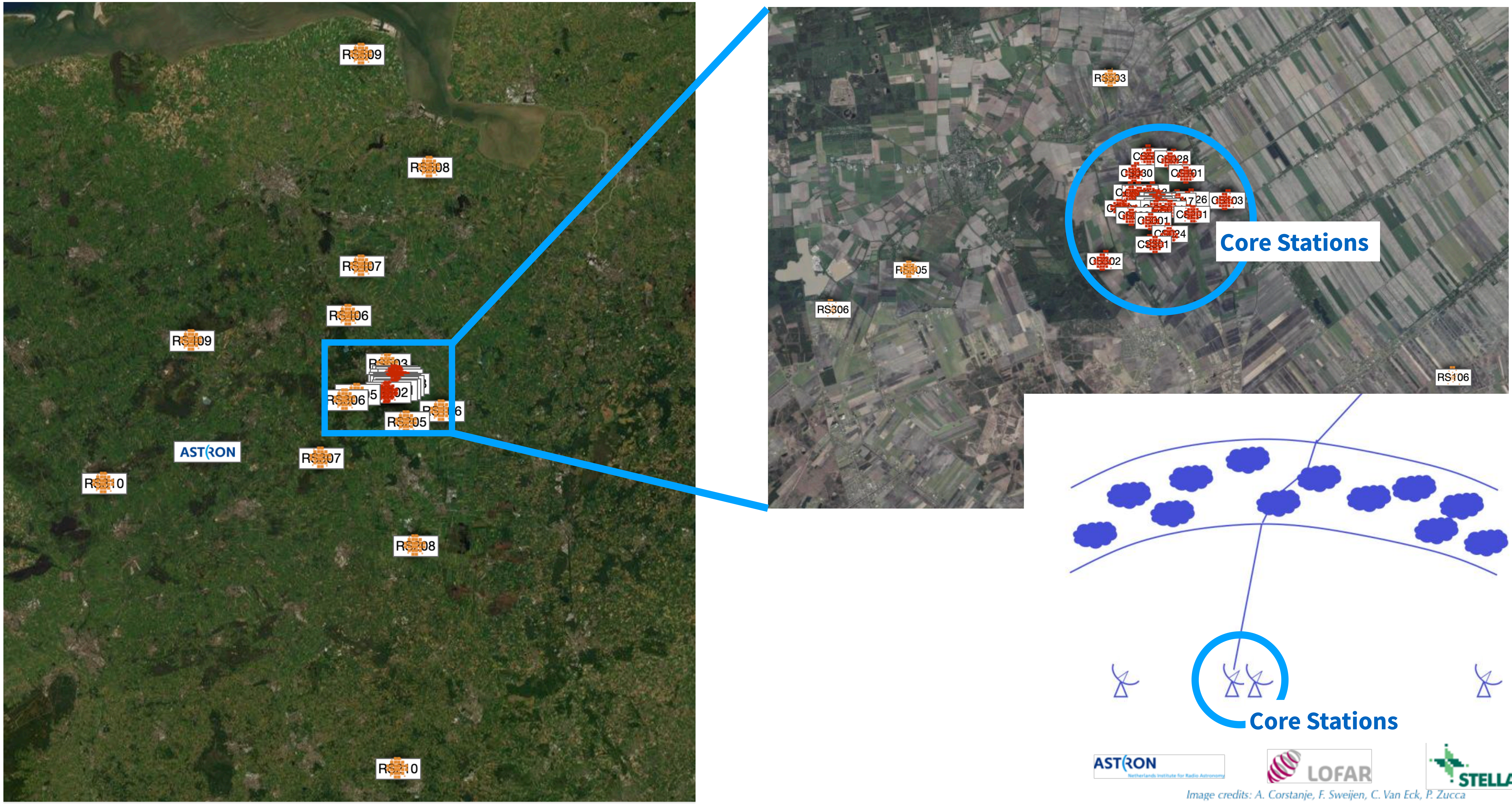
- The sky model used in the calibration is stored in **`run1/skymodels/calibrate_1/calibration_skymodel.txt`** (it should be just a copy of the input sky model in this case)
- The output of calibration is in **`run1/pipelines/calibrate_1`**
- Many output files are created during calibration, but the most important are:
  - `fast_phases.h5parm` ← **Ionospheric corrections**
    `combined_solutions.h5` ← **Combined ionospheric and beam corrections**
    `spit_solutions_*.h5` ← **Screen fits**
    `diagonal_aterms_*.fits` ← **Actual screens ("a-terms") passed to WSClean**

Tutorial: Calibration Solutions

Ionospheric effects

- Ionospheric effects are solved for with the “fast-phase” solve:
  - Typical solution interval of 8-32 seconds in time and 1 MHz in frequency
  - Solve is a scalar one (same solutions for XX and YY)
  - All core stations constrained to have the same solutions

Tutorial: Calibration Solutions



ASTRON LOFAR STELLAR

Image credits: A. Corstanje, F. Sweijen, C. Van Eck, P. Zucca

Tutorial: Plot the Phase Solutions

- Change to the calibration working directory
  - `$ cd run1/pipelines/calibrate_1`
- Get the updated **`plotrapthor`** tool. From inside the Docker/Singularity container, after changing to the `calibrate_1` directory above, do:
  - `$ wget -O plotrapthor https://github.com/darafferty/rapthor/raw/master/bin/plotrapthor`
  - `$ chmod u+x plotrapthor`
- Plot the solutions:
  - `$ ./plotrapthor fast_phases.h5parm scalarphase`

H5parm file **Type of solutions to plot**

Tutorial: Plot the Phase Solutions

- You should now have PNG files with the plots, one per calibration patch:

  - ```
    $ ls
    scalarphase_dir[Patch_1].png
    scalarphase_dir[Patch_2].png
    scalarphase_dir[Patch_3].png
    scalarphase_dir[Patch_4].png
    scalarphase_dir[Patch_5].png
    ```
    ```

Tutorial: Calibration Solutions

Ionospheric corrections

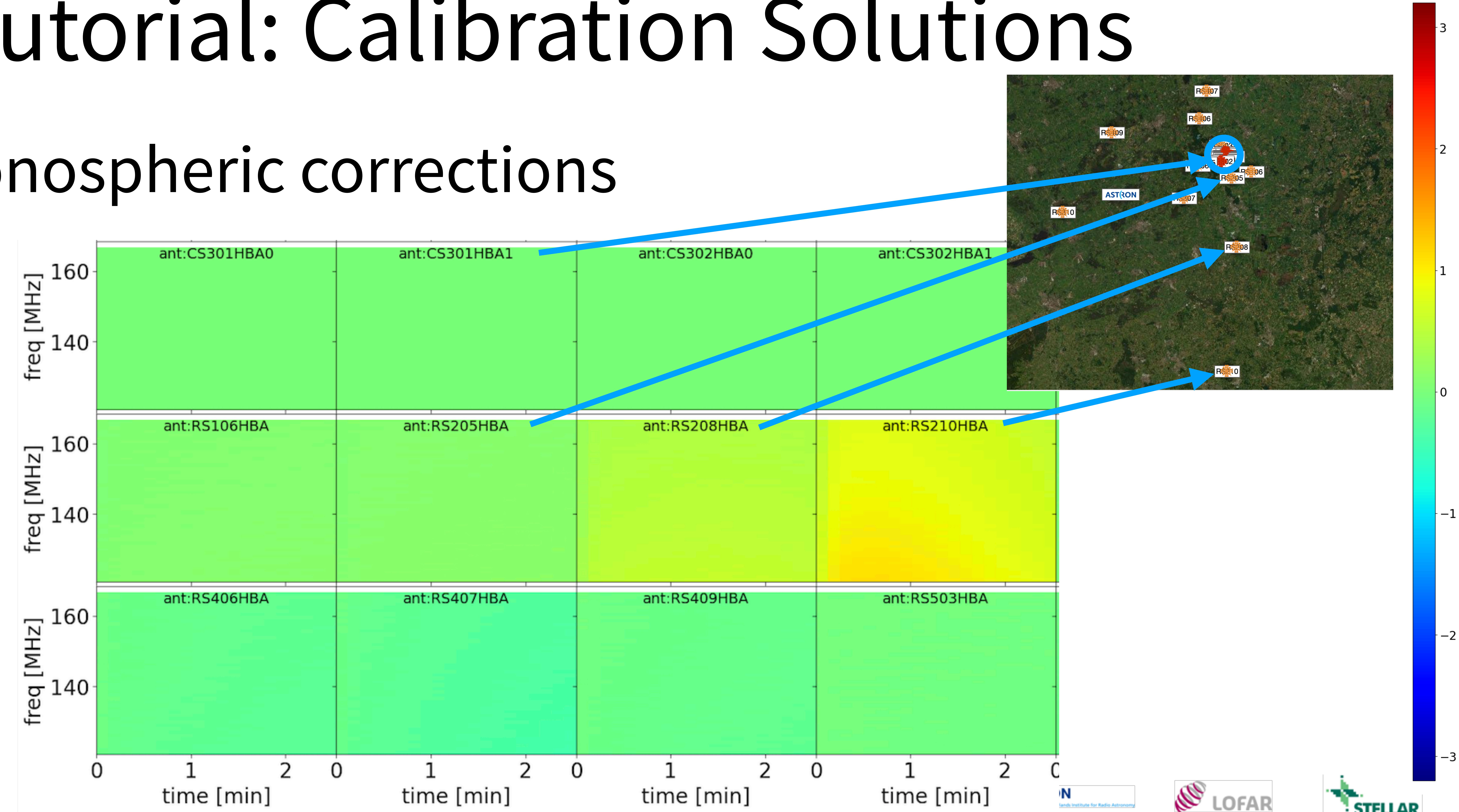


Image credits: A. Corstanje, F. Sweijen, C. Van Eck, P. Zucca

6th LOFAR Data School

Tutorial: Calibration Solutions

Ionospheric corrections

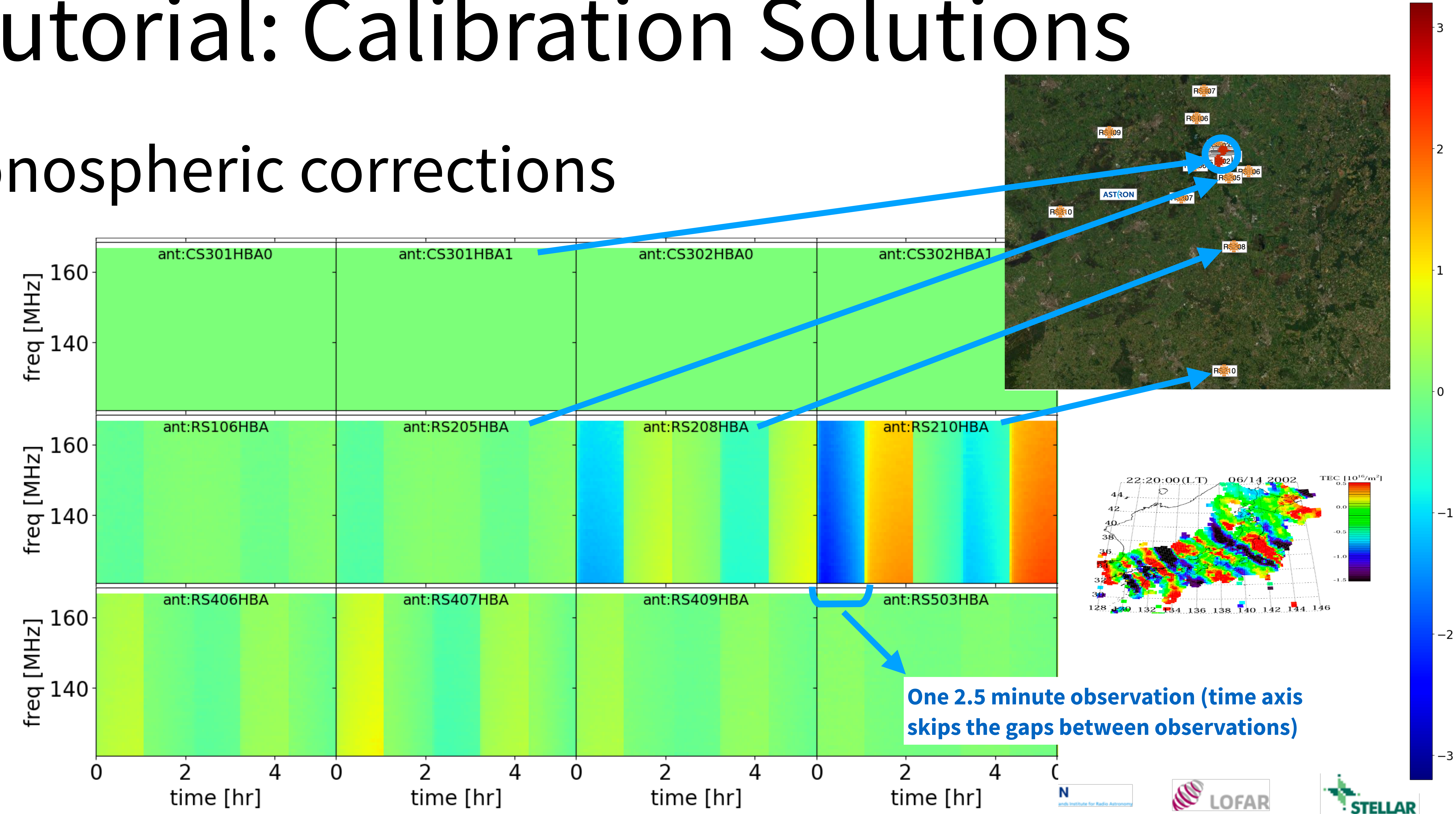


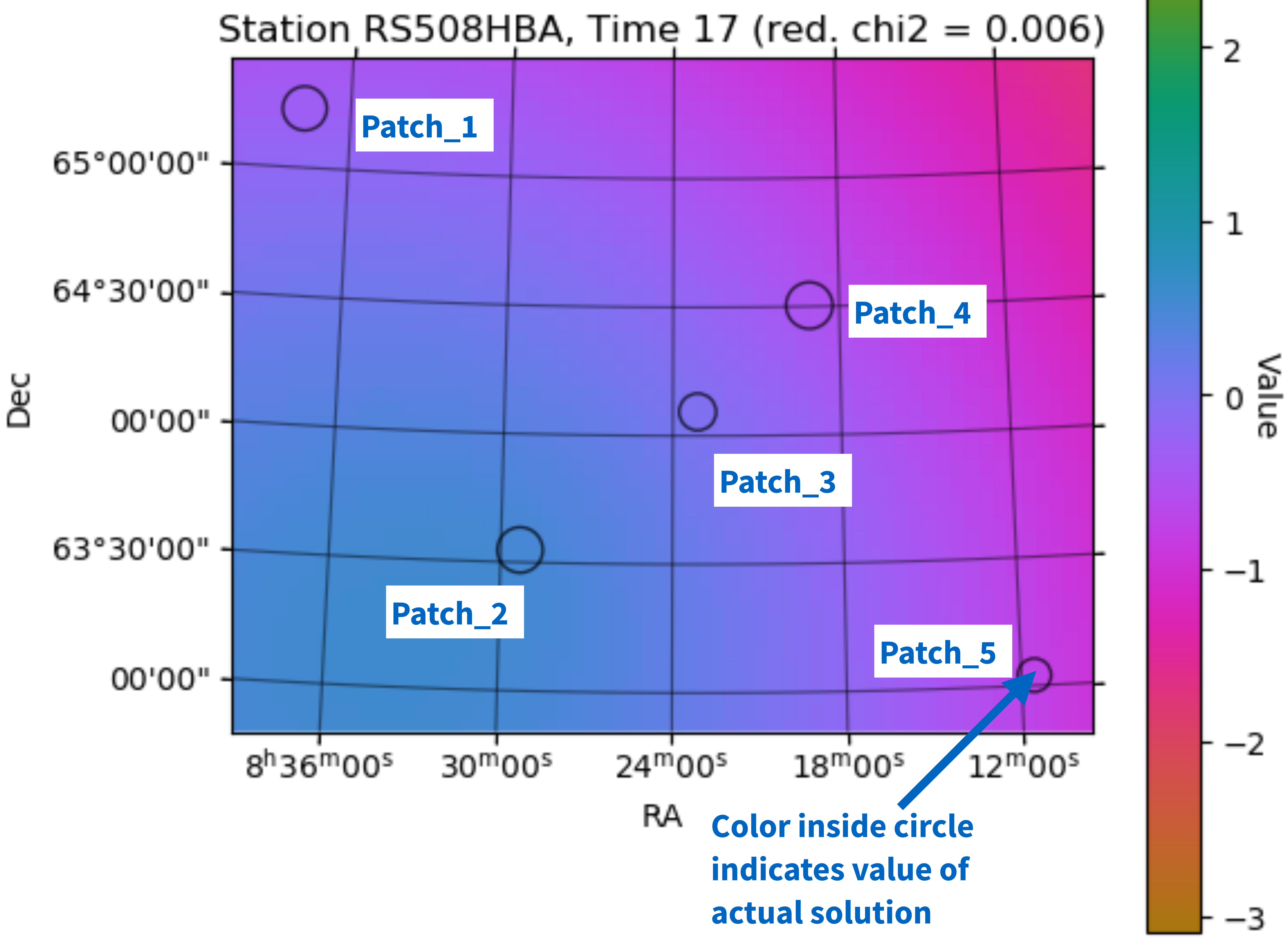
Image credits: A. Corstanje, F. Sweijen, C. Van Eck, P. Zucca

Tutorial: Plot the Phase Screens

- Use the **`plotrapthor`** tool again
- Change to the calibration working directory
  - `$ cd run1/pipelines/calibrate_1`
- Screens fits are stored in the **`split_solutions_*.h5`** files. Plot the fast-phase screens with, e.g. (for station RS508):
  - `$ ./plotrapthor split_solutions_0.h5 phasescreen --ant=RS210HBA`

Tutorial: Phase Screens

- Each station is treated independently
  - All effects due to 3-D structure of the ionospher are collapsed into one correction “image”
- One screen per time slot, per frequency

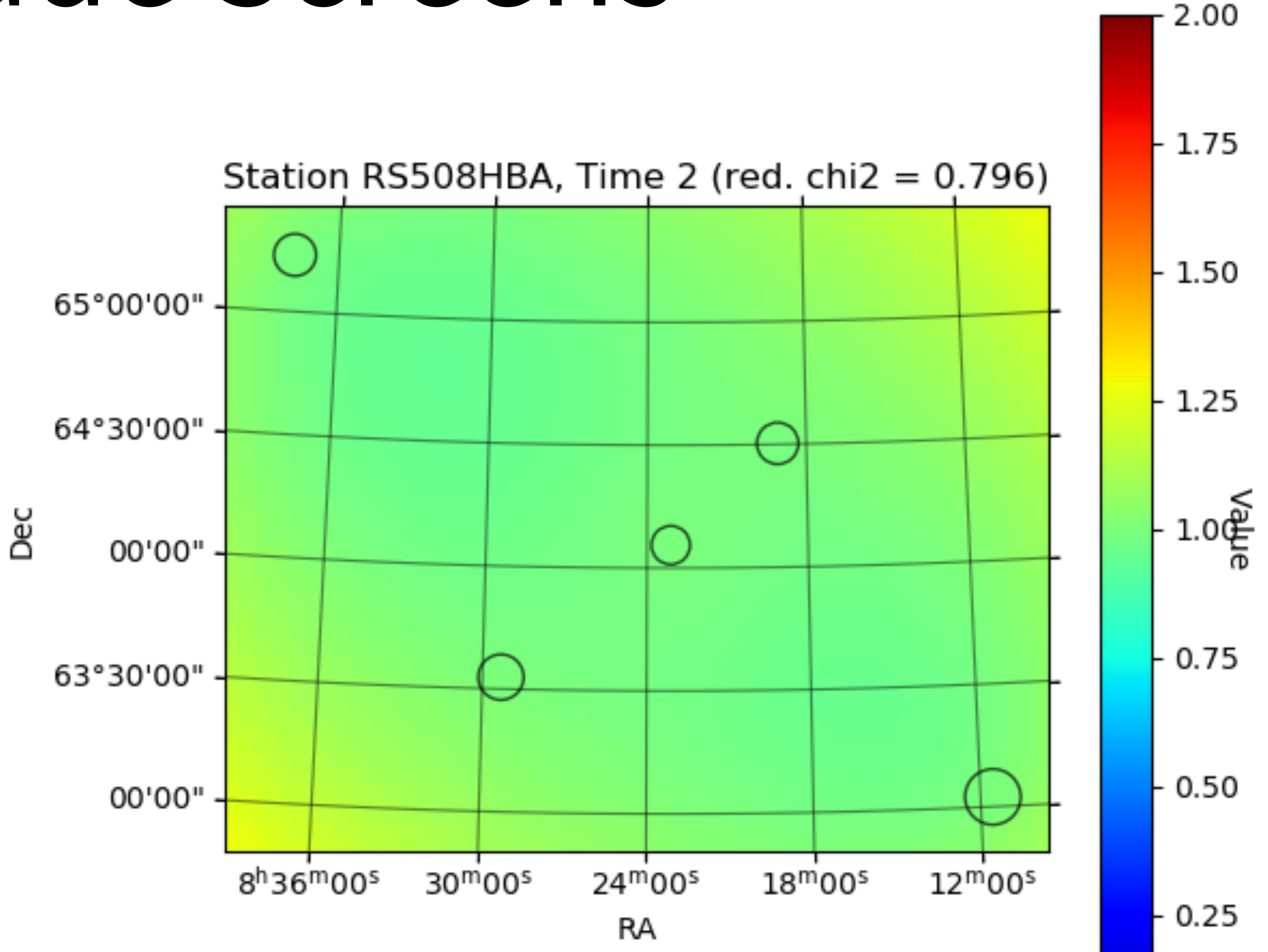


Tutorial: Plots

- Some things to look at:
  - How do the phase screens evolve with time? View the time evolution of the screens by paging through the screen images (there should be 20 frames, each of 8 seconds)
  - Do the values in the phase screens match those in the scalar phase plots you made earlier?
- Optional:
  - Use LoSoTo to derive the dTEC values from some of the phase solutions in **`fast_phases.h5parm`** (hint: use the TEC operation). What are typical values (in TECU) for the core stations (e.g., CS501HBA0) vs. the remote stations (e.g., RS210HBA)?

Tutorial: Amplitude Screens
(optional)

- Amplitude effects solved for with a "slow-gain" solve:
  - Typical solution interval of 5-10 minutes
- No unmodeled effects were simulated, so we just get ~ 1 everywhere



- `$ ./plotrapthor split_solutions_0.h5 ampscreen --ant=RS210HBA`

Image credits: A. Corstanje, F. Sweijen, C. Van Eck, P. Zucca

Tutorial: Self Calibration (optional)

- To run full self calibration, edit the parset as follows:

```
[global]
dir_working = /path/to/rapthor_data/run2
input_ms = /path/to/rapthor_data/*.ms
strategy = selfcal
```

Use all MS files (to improve uv coverage)

Selfcal strategy included in Rapthor

- Warning: may take a few hours per self calibration cycle
- To save time, only the source in the center will be imaged. Output images will be located in **`run2/images`**