NenuFAR Python Tools
Nenupy

Alan Loh & the NenuFAR team

Outline

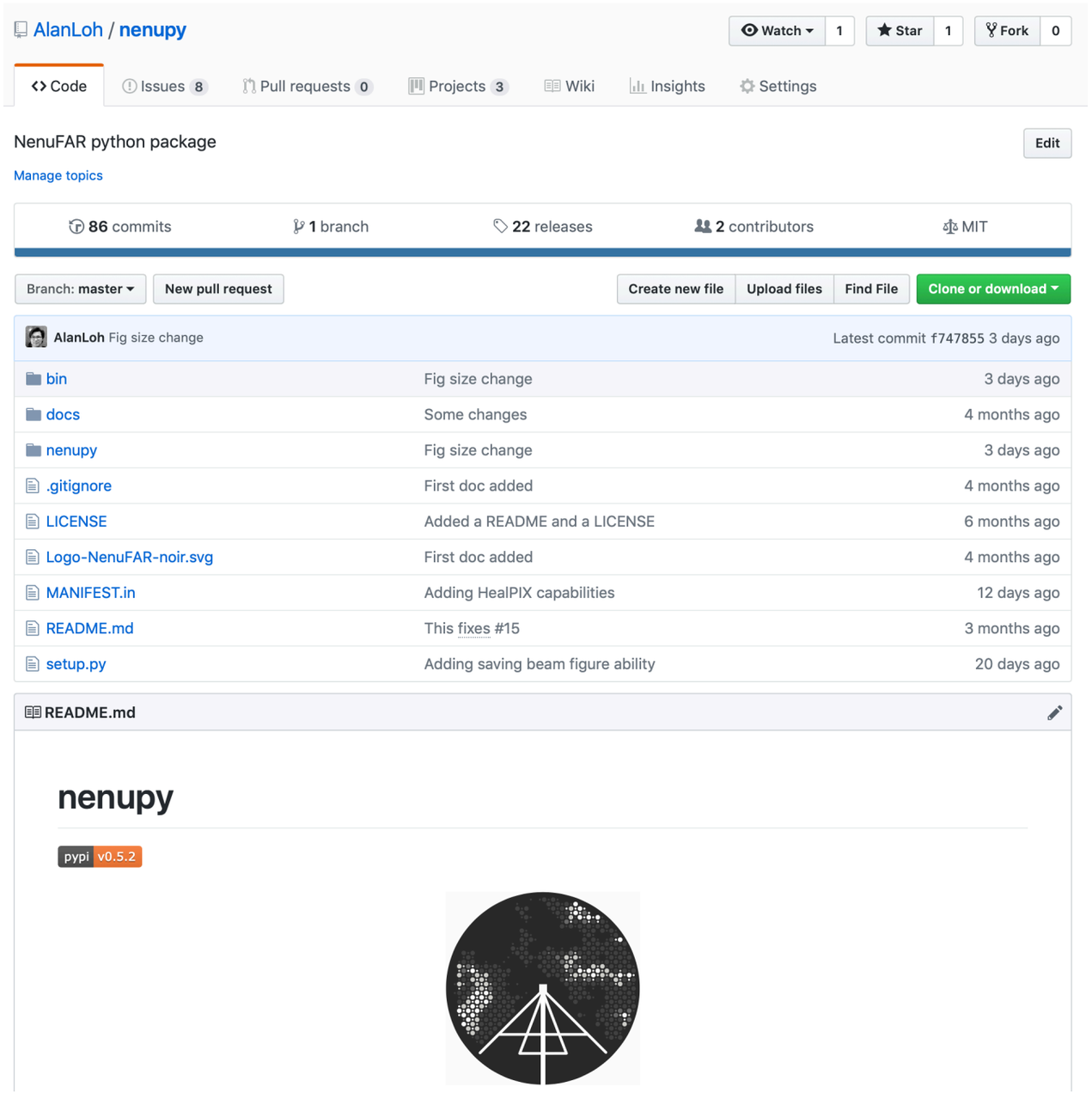
- **Nenupy**: Python3 package
  - Installation
  - Philosophy
  - Usage

- Read & select NenuFAR data
  - Via Python and within your own scripts
  - Guided User Interface

- Simulation
  - Sky model
  - NenuFAR beam
  - Predict / confront observations

Nenupy

Nenupy on Github



- Python3 package
- 'a' NenuFAR data reading software, experimental, mostly developed by a single user...
- **Contributions/feedback are welcome!**
- Github repo: *https://github.com/AlanLoh/nenupy*

Installation

- On your computer (Python3 needed):
  - Same as a 'classical' Python package:

```
pip install nenupy
```

  - Update the package after a new release:

```
pip install nenupy --upgrade
```

    - Dependencies: **astropy**, **healpy**, **pyGSM** (https://github.com/telegraphic/PyGSM), **dash** (*https://github.com/plotly/dash*)

- On *nancep*:
  - Initialize the paths towards the python library:

```
use Nenupy3
```

Usage

- nancep / hands-on sessions:
  - SSH connection:

```
ssh -XY <user_name>@nancep1.obs-nancay.fr
```

  - Initialization

```
use Nenupy3
```

  - Data directory

```
cd /databf/nenufar
```

  - You're set!

- Usage:
  - via Python

```
import nenupy
```

  - Command line

```
nenuplot -o *BST.fits
```

Data Reading

Opening FITS files (without nenupy)

```
[[aloh@nancep1] /databf/nenufar/20190316_133300_20190316_141400_3C47_TRANSIT > ipython3
Python 3.5.2 (default, Nov 12 2018, 13:43:14)
Type "copyright", "credits" or "license" for more information.

IPython 2.4.1 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

[In [1]: from astropy.io import fits

[In [2]: hdu = fits.open('20190316_133300_BST.fits')

[In [3]: hdu.info()
Filename: 20190316_133300_BST.fits
No.    Name         Type      Cards   Dimensions   Format
0    PRIMARY     PrimaryHDU      21   ()
1    SETUP_INSTR  BinTableHDU     46   1R x 17C     [1I, 512E, 1I, 1I, 60I, 1I, 56I, 3D, 20A, 180D, 180D, 60I, 120D, 1I, 4A, 1I, 512E]
2    SETUP_OBS   BinTableHDU     41   1R x 16C     [50A, 50A, 50A, 50A, 1L, 1L, 1I, 1I, 100A, 1L, 1I, 1I, 1I, 1I, 1I, 1I]
3    SETUP_ANABEAM  BinTableHDU     46   1R x 18C     [50A, 1I, 1D, 1D, 50A, 100A, 20A, 1J, 1I, 96B, 96I, 50A, 1I, 10I, 200A, 1I, 1I, 1L]
4    SETUP_BEAM  BinTableHDU     44   1R x 17C     [50A, 1I, 1I, 1D, 1D, 1D, 1D, 50A, 100A, 20A, 1J, 1I, 768I, 768E, 1I, 96B,  1I]
5    SETUP_POINTING_ABEAM  BinTableHDU     19   2R x 4C      [1I, 20A, 1D, 1D]
6    SETUP_POINTING_BEAM  BinTableHDU     19   1R x 4C      [1I, 20A, 1D, 1D]
7    BST         BinTableHDU     14   2400R x 2C   [1D, 1536E]

[In [4]: data = hdu[7].data

[In [5]: data.names
Out[5]: ['jd', 'DATA']

[In [6]: data['DATA'].shape
Out[6]: (2400, 2, 768)
```

(time, polarization, beamlet)

https://doc-nenufar.obs-nancay.fr/Soft-nenuFAR/detail_FITS.pdf

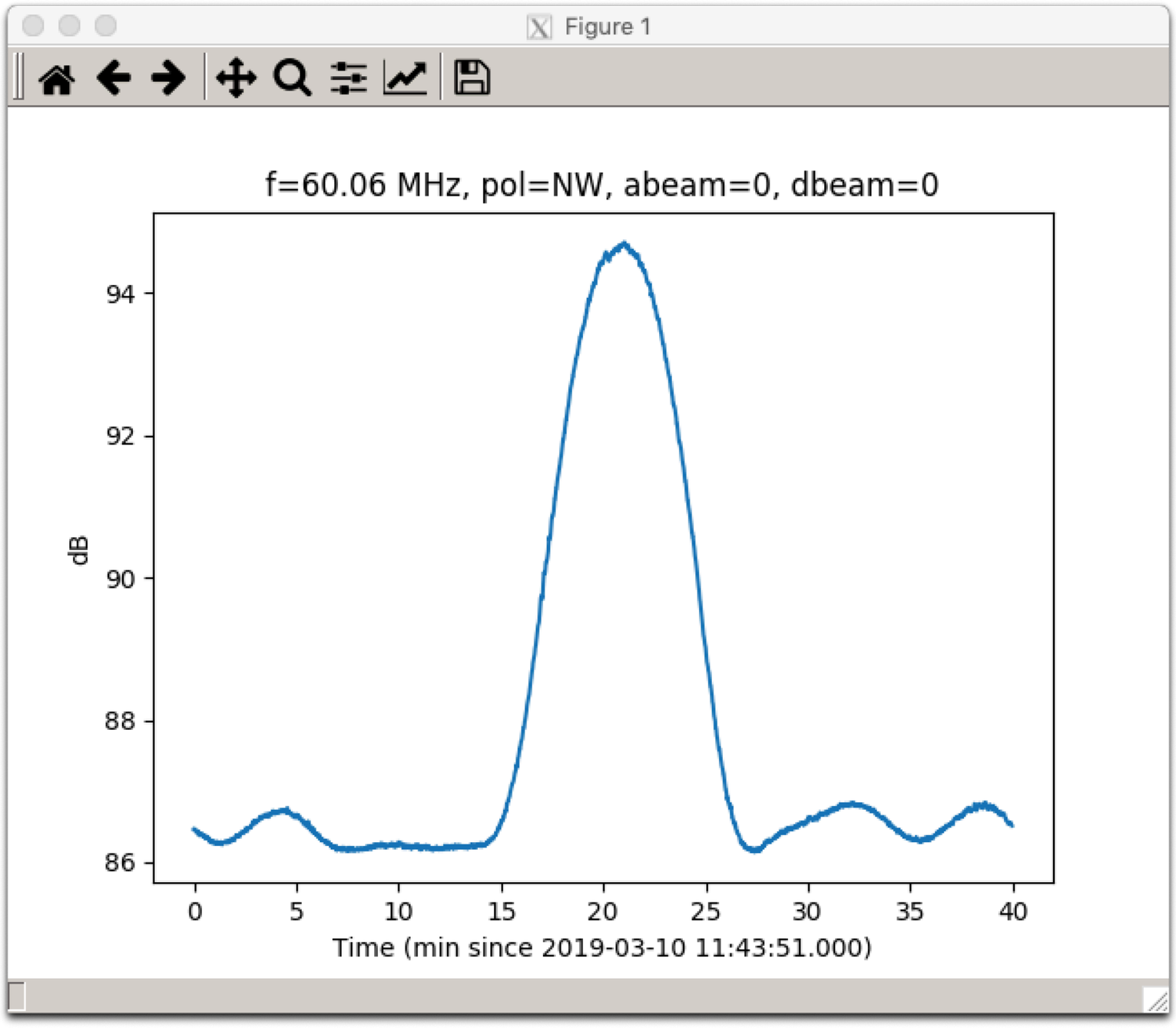
Command-line: info

```
nenuinfo -o ./20190316_133300_BST.fits
```

```
       === Class BST of nenupy ===
       List of all current attributes:
abeam: 0
abeams: [0]
angles: (180.256, 64.0344, 'ALTAZ')
dbeam: 0
dbeams: [0]
exposure: 0.02777777777777779
freq: 50
freqmax: 78.80859
freqmin: 30.175781
from_transit: <bound method BST.from_transit of <BST object: obsfile=/databf/nenufar/20190316_133300_20190316_141400_3C47_TRANSIT/20190316_133300_BST.fits>>
ma: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
 48 49 50 51 52 53 54 55]
mean: <bound method BST.mean of <BST object: obsfile=/databf/nenufar/20190316_133300_20190316_141400_3C47_TRANSIT/20190316_133300_BST.fits>>
name: ['"3C47_TRANSIT"']
obsfile: /databf/nenufar/20190316_133300_20190316_141400_3C47_TRANSIT/20190316_133300_BST.fits
obsname: 20190316_133300
obstart: 2019-03-16T13:33:10.000
obstop: 2019-03-16T14:13:10.000
polar: NW
time: [<Time object: scale='utc' format='isot' value=2019-03-16T13:33:10.000>, <Time object: scale='utc' format='isot' value=2019-03-16T14:13:10.000>]
type: transit


[aleh@nancep1] /databf/nenufar/20190316_133300_20190316_141400_3C47_TRANSIT > ls
```

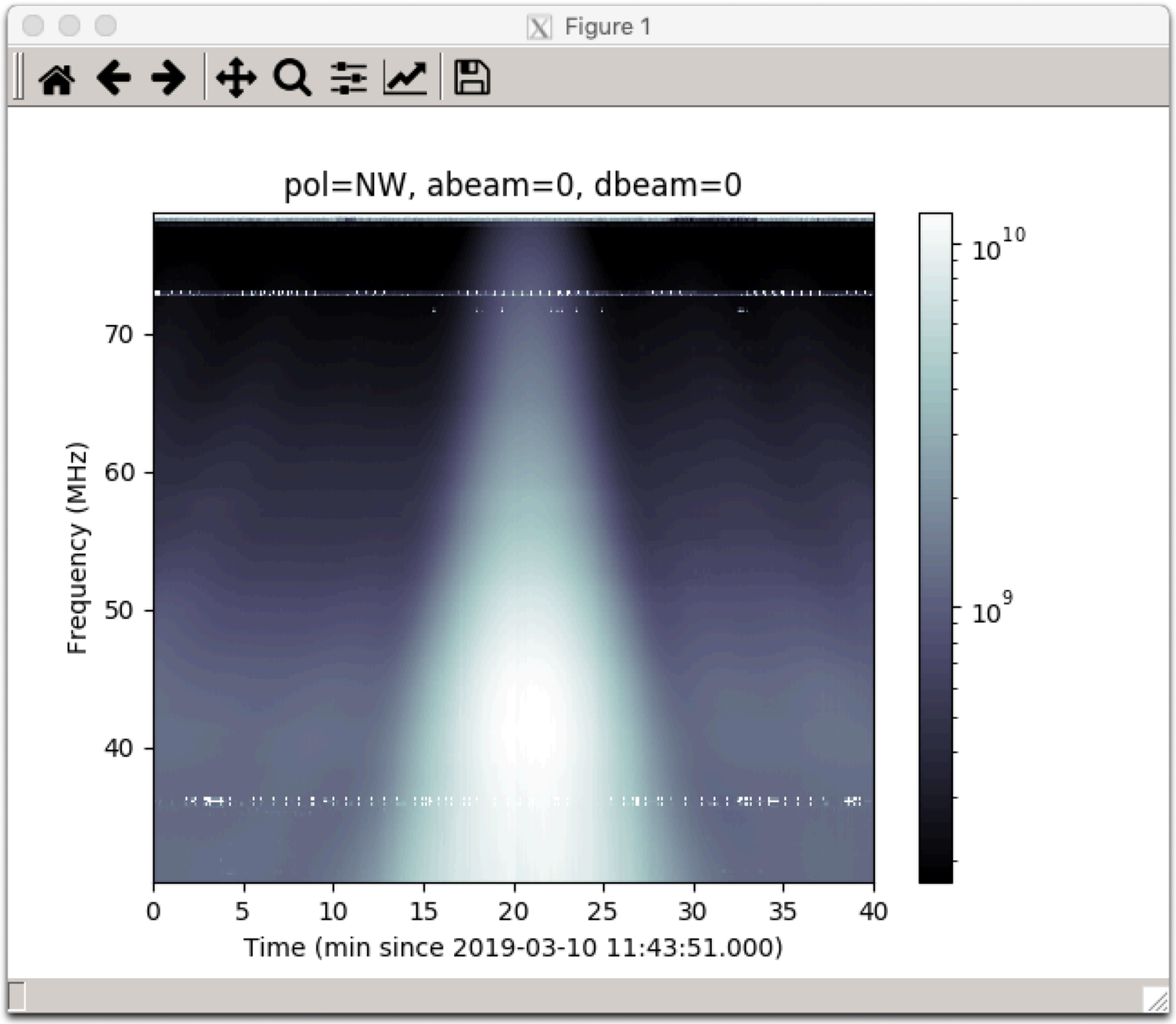
Command-line: plot

```
nenuplot -o 20190310_114300_BST.fits -f 60 -p NW
```



Command-line: plot

```
nenuplot -o 20190310_114300_BST.fits -f 10 90 -p NW
```



Command-line: GUI

```
nenuplot-gui
```

In another session:

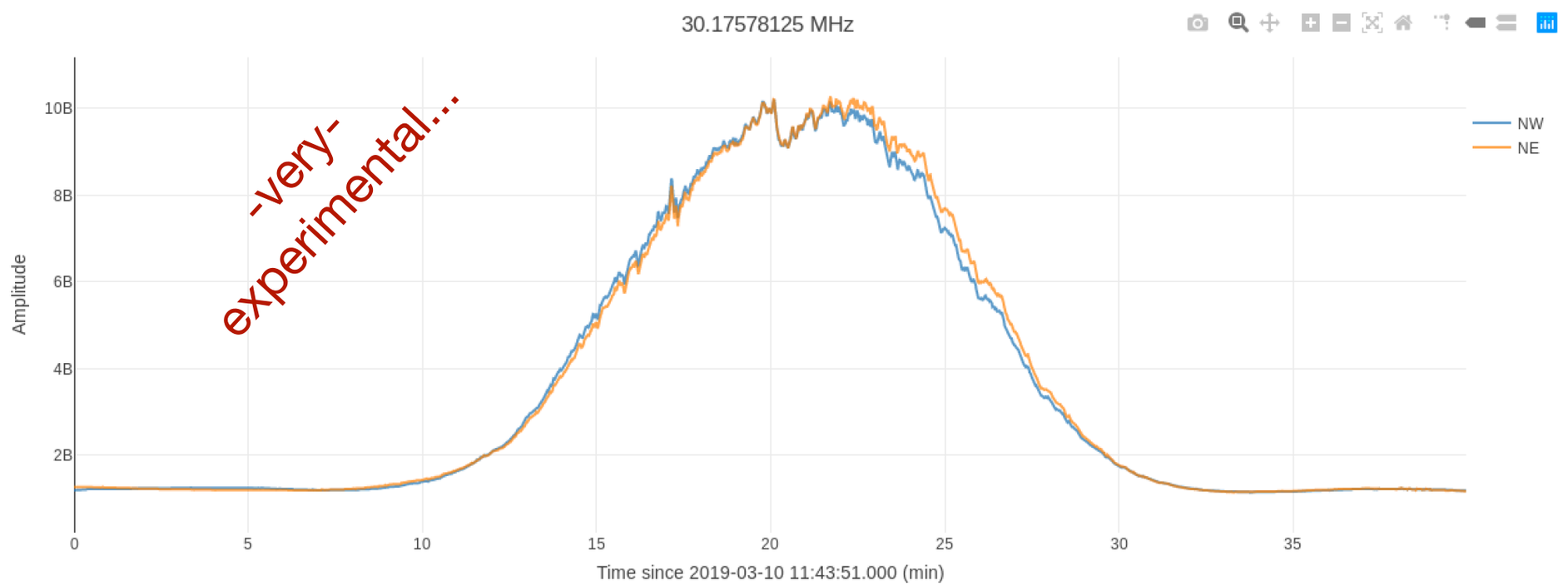
```
firefox http://127.0.0.1:8050/
```

BST plotting interface

/databf/nenufar/20190310_114300_20190310_122400_CAS_A_TRANSIT SUBMIT

Filename:

/databf/nenufar/20190310_114300_20190310_122400_CAS_A_TRANSIT/20190310_114300_BST.fits



Frequency (MHz):

30.18 34.28 38.38 42.48 46.58 50.68 54.79 58.89 62.99 67.09 71.19 75.29

Analogical beam: Numerical beam:

0 0

BST reading

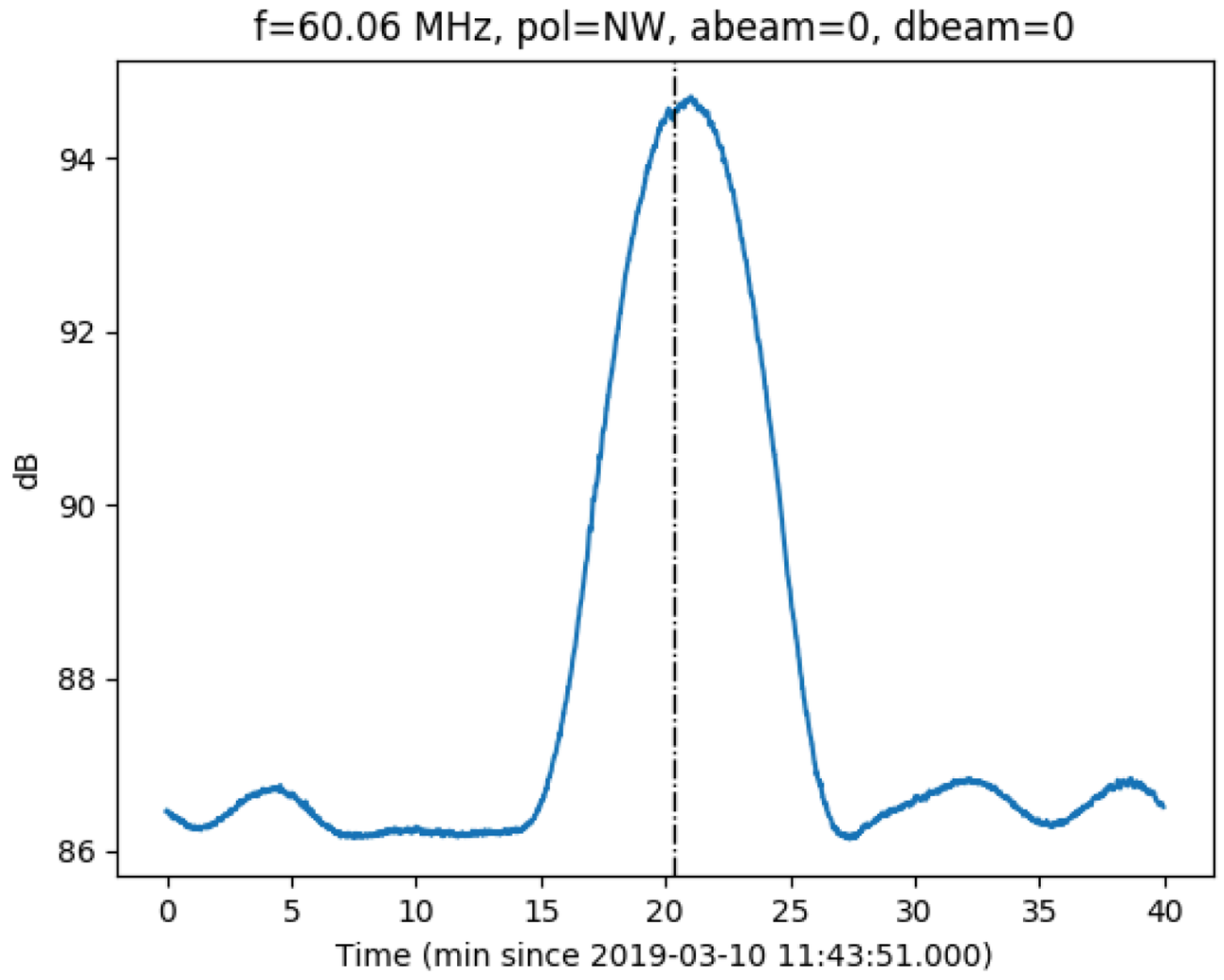
```
from nenupy import BST
bst = BST('20190310_114300_BST.fits')
bst.freqmin, bst.freqmax
    (30.175781, 78.80859)
bst.obstart
    <Time object: scale='utc' format='isot' value=2019-03-10T11:43:50.000>
bst.obstart.jd
    2458552.988773148
bst.type
    'transit'
bst.azdig, bst.eldig
    (359.8211, 78.1978)
bst.ma
   array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
       34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
       51, 52, 53, 54, 55], dtype=uint8)
```

BST data selection

```
from nenupy import BST


bst = BST('20190310_114300_BST.fits')


bst.select(freq=60)
bst.data.keys()
    dict_keys(['amp', 'freq', 'time'])
bst.data['amp'].shape
    (2399,)


bst.select(freq=[40, 60])
bst.data['amp'].shape
    (2399, 104)
bst.data['time'].isot
    array(['2019-03-10T11:43:51.000', '2019-03-10T11:43:52.000',
       '2019-03-10T11:43:53.000', ..., '2019-03-10T12:23:47.000',
       '2019-03-10T12:23:48.000', '2019-03-10T12:23:49.000'], dtype='<U23')
```

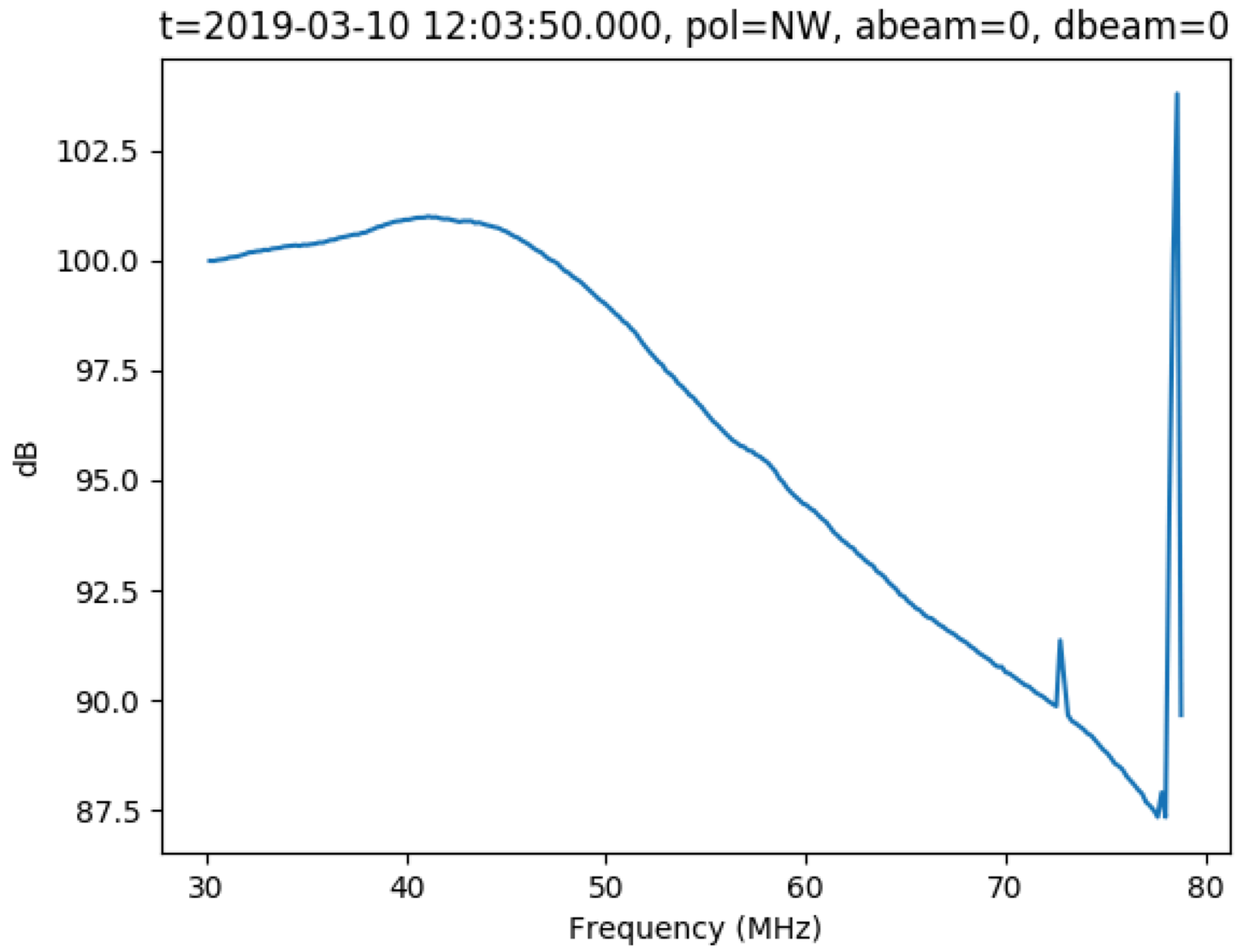
BST plotting

```
from nenupy import BST
bst = BST('20190310_114300_BST.fits')
bst.select(freq=60)
bst.src = 'Cas A'
bst.plot()
```



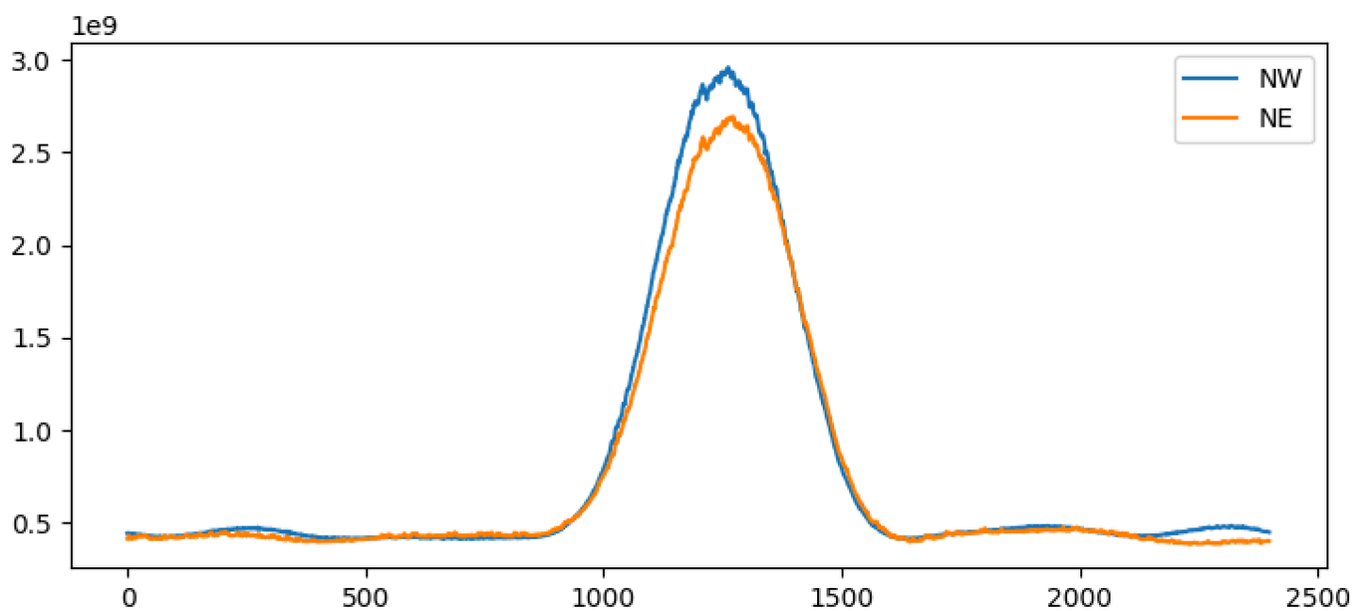
BST plotting

```
from nenupy import BST
bst = BST('20190310_114300_BST.fits')
bst.select(freq=[10, 90], time='2019-03-10 12:03:50')
bst.plot()
```



BST plotting

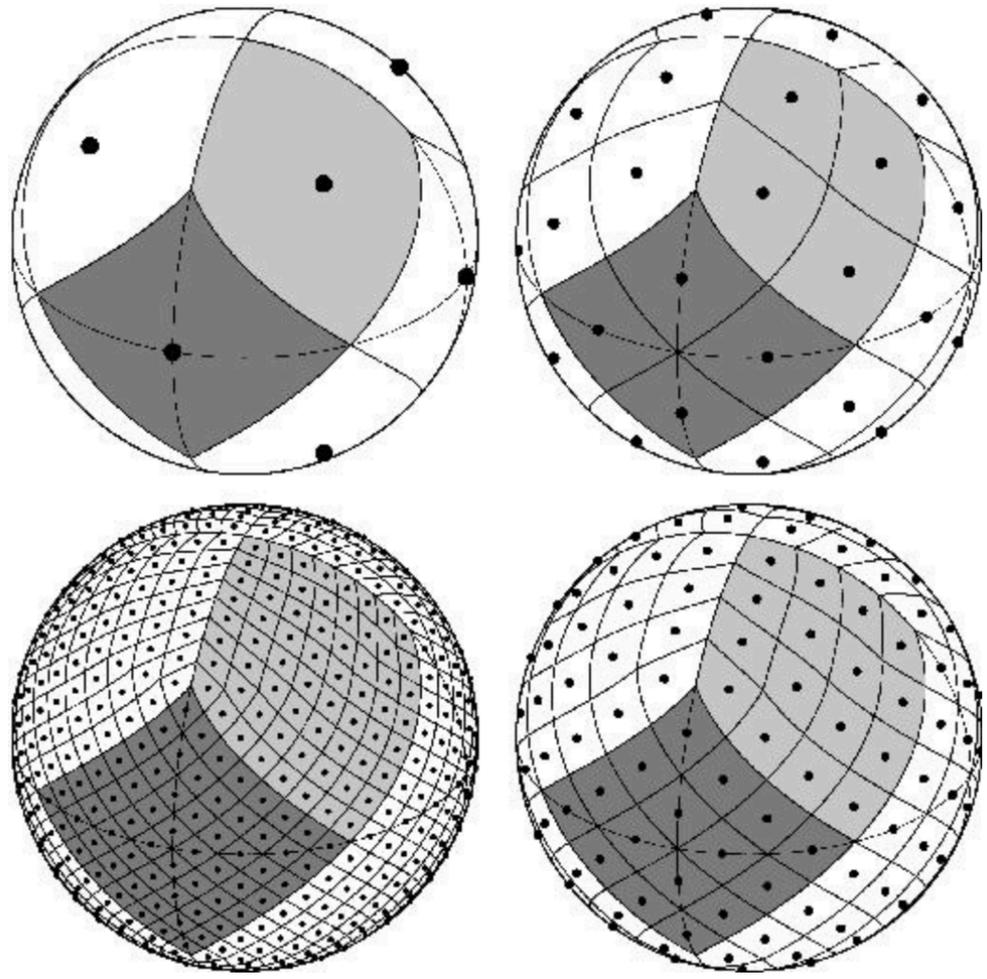
```
from nenupy import BST
import pylab as plt
bst = BST('20190310_114300_BST.fits')
bst.select(freq=60, polar='NW')
plt.plot((bst.data['time']-bst.data['time'][0]).sec, bst.data['amp'],
label='NW')
bst.select(freq=60, polar='NE')
plt.plot((bst.data['time']-bst.data['time'][0]).sec, bst.data['amp'],
label='NE')
plt.legend()
plt.show()
```



Simulation

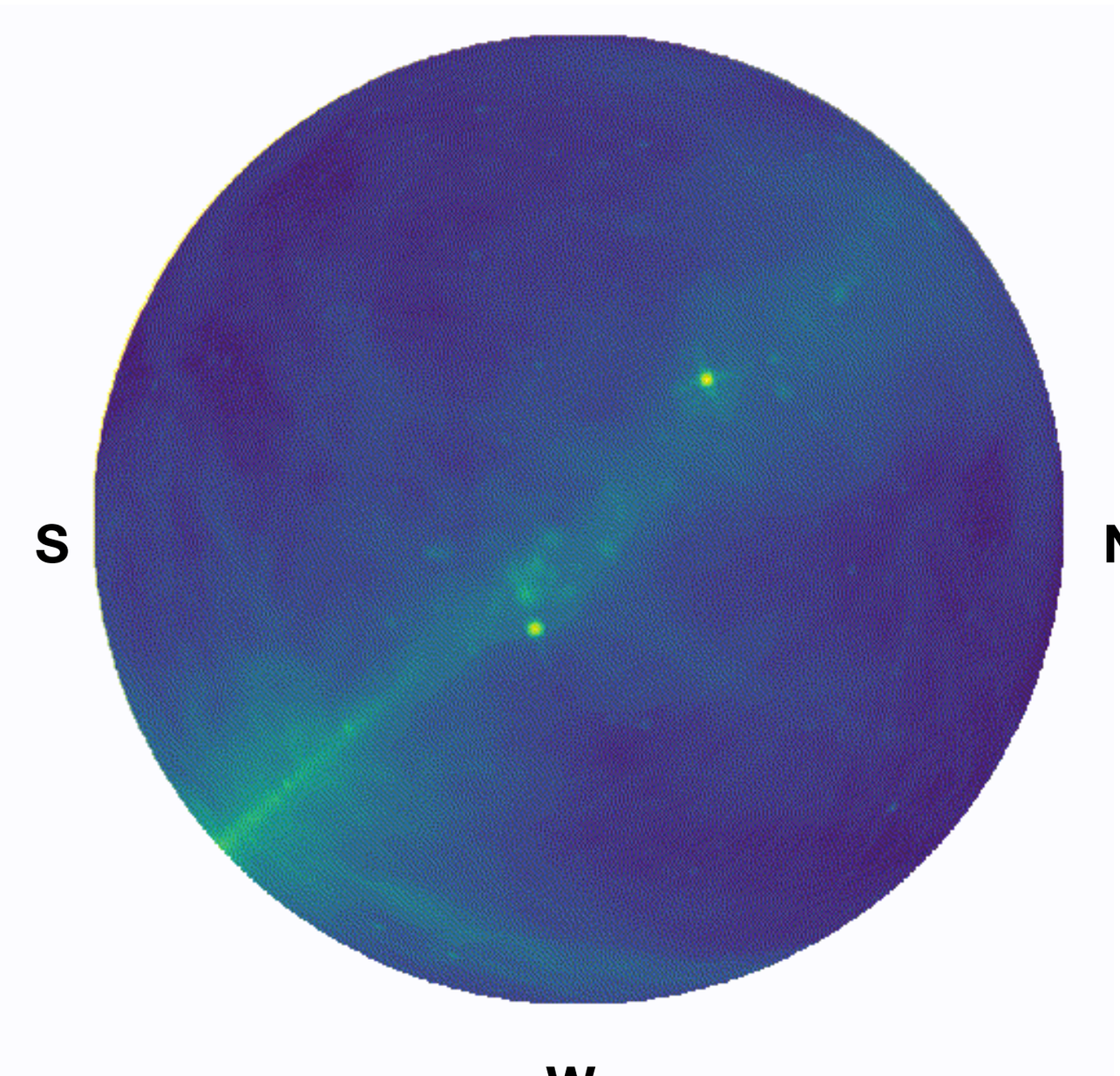
HEALPix

- All-sky representation
- **H**ierarchical **E**qual **A**rea isO**L**atitude **Pix**elization
- Subdivision of a spherical surface: **each pixel covers the same surface area as every other pixel**
- Faster, easier to manipulate
- Python package: **healpy**



https://healpix.jpl.nasa.gov/index.shtml

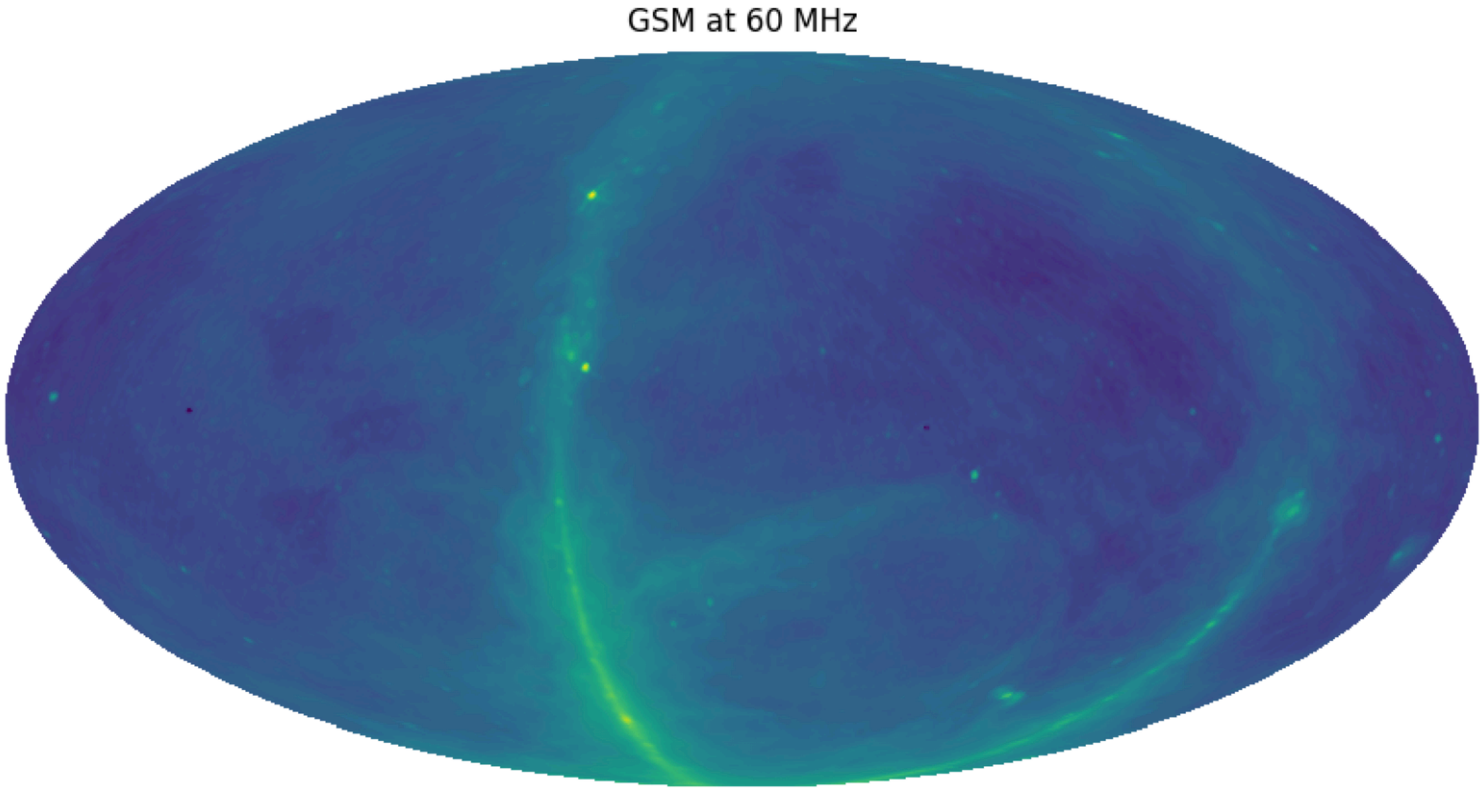
Global Sky Model



- **Diffuse Galactic Radio Emission** model (Oliveira-Costa et al. 2008)
- 10 MHz to 100 GHz
- Python package: **PyGSM** (*https://github.com/telegraphic/PyGSM*)

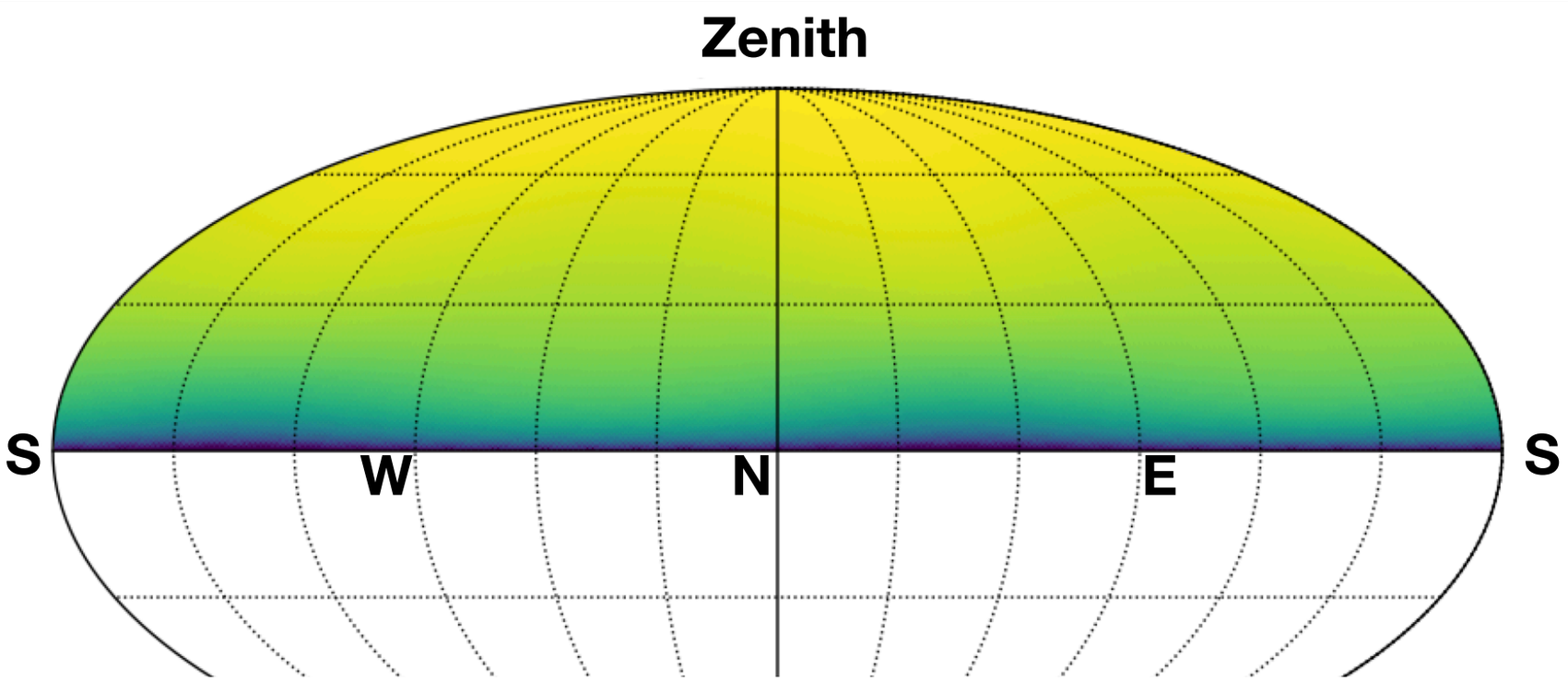
Global Sky Model

```
import healpy as hp
import pylab as plt
from nenupy.hpx import Skymodel

sky = Skymodel(freq=60, nside=256)
map = sky.get_skymodel(time='2019-03-18 16:30:00', model='gsm')
hp.mollview(map, title='GSM at 60 MHz', norm='log')
plt.show()
```



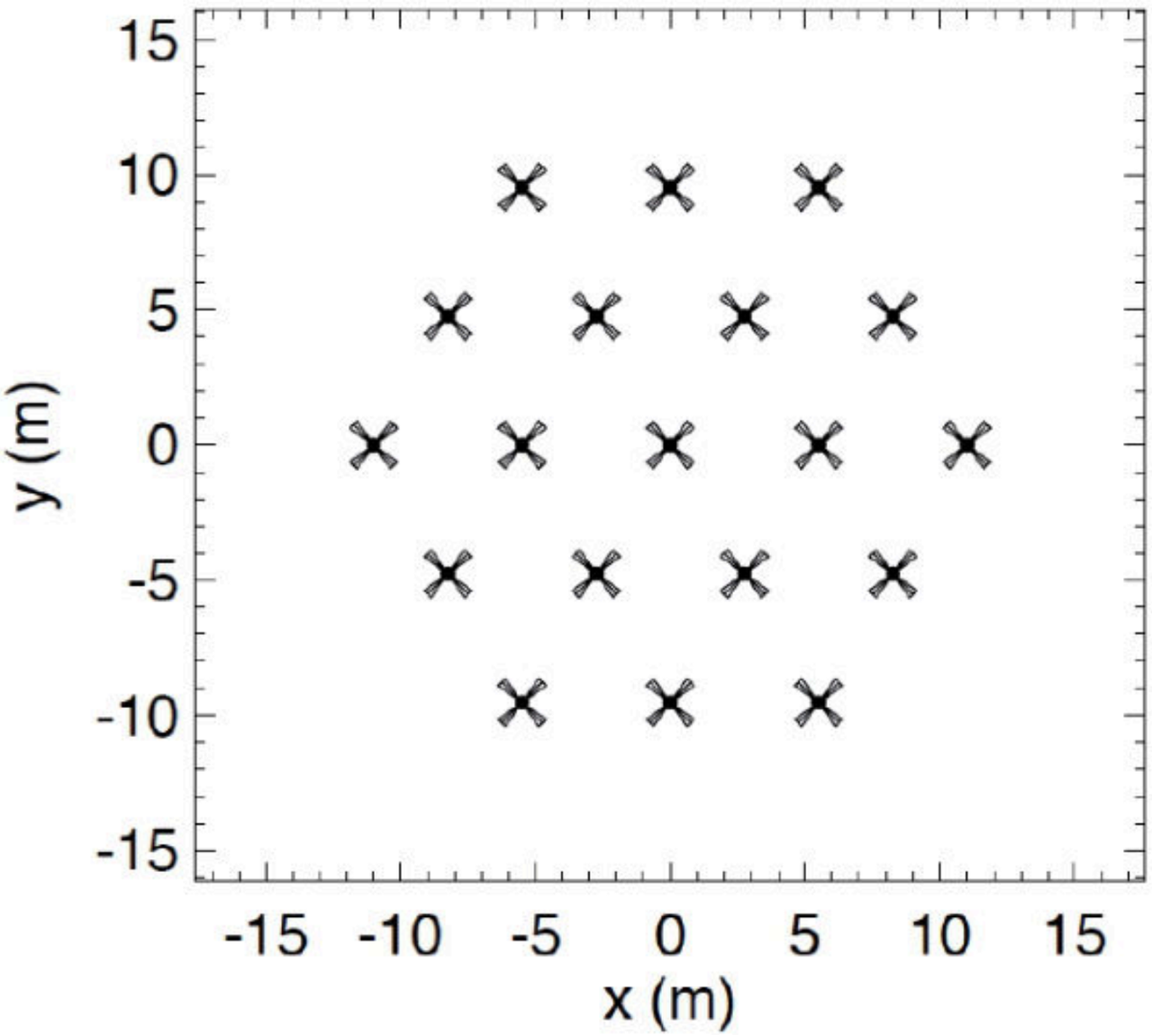
Beam - Antenna

```
import healpy as hp
import pylab as plt; import numpy as np
from nenupy.hpx import Anabeam
ma_beam = Anabeam(ma=0, freq=60, resol=0.2, ant=9)
ma_beam.get_anabeam()
hp.mollview(np.log10(ma_beam.anabeam))
hp.graticule()
plt.show()
```



NEC Simulation by D. Charrier

Beam - Mini-array

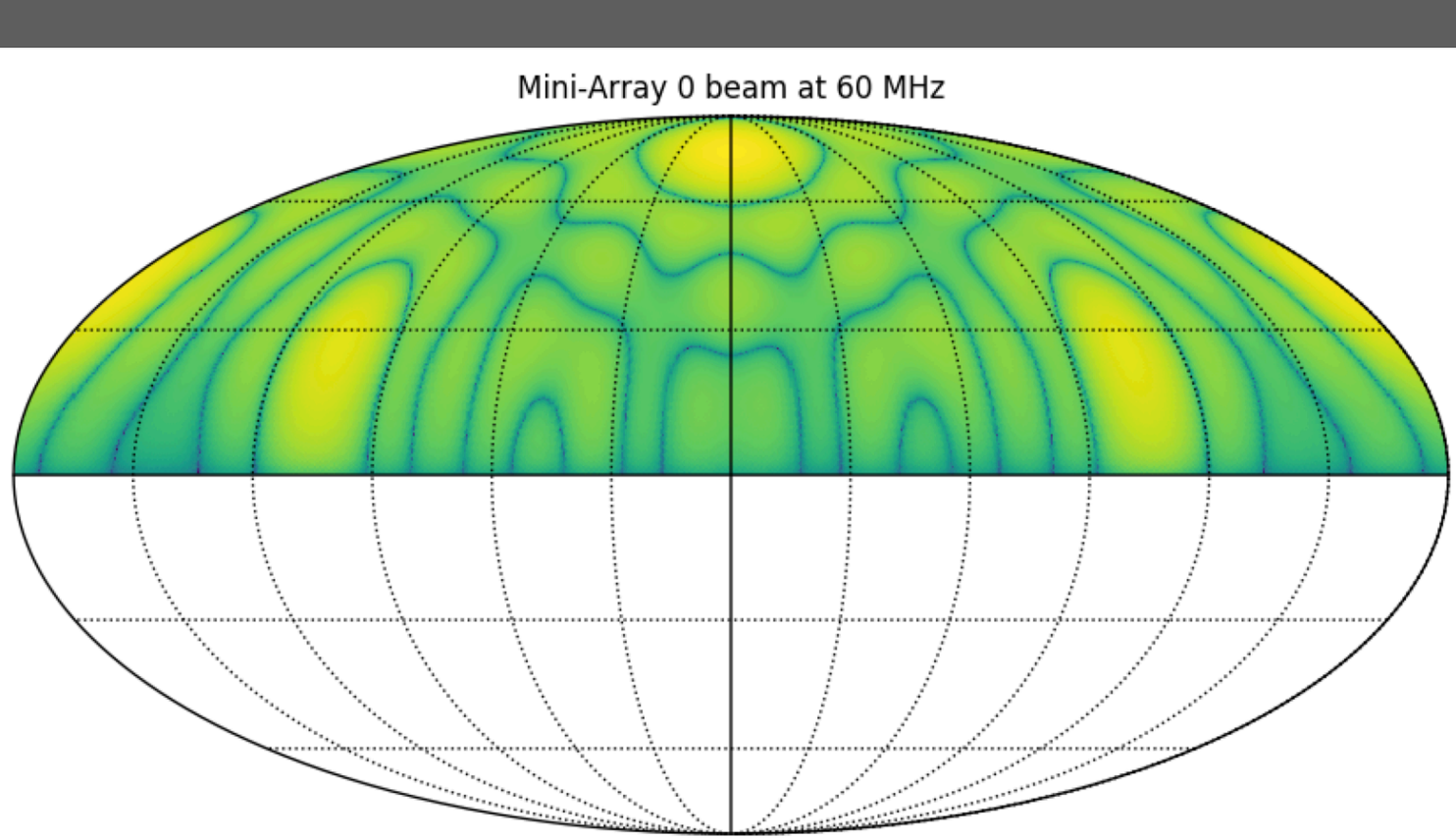


$$G_{MA} \sim G_{ant} \Sigma_{antennas} e^{i \phi(ant)}$$

- MA beam simulation:
- rotations
- beam squint
- discrete pointing grid

Beam - Mini-array

```
import healpy as hp
import pylab as plt; import numpy as np
from nenupy.hpx import Anabeam
ma_beam = Anabeam(ma=0, freq=60, azana=0, elana=75, resol=0.2)
ma_beam.get_anabeam()
hp.mollview(np.log10(ma_beam.anabeam), title='Mini-Array 0 beam at 60 MHz')
hp.graticule()
plt.show()
```

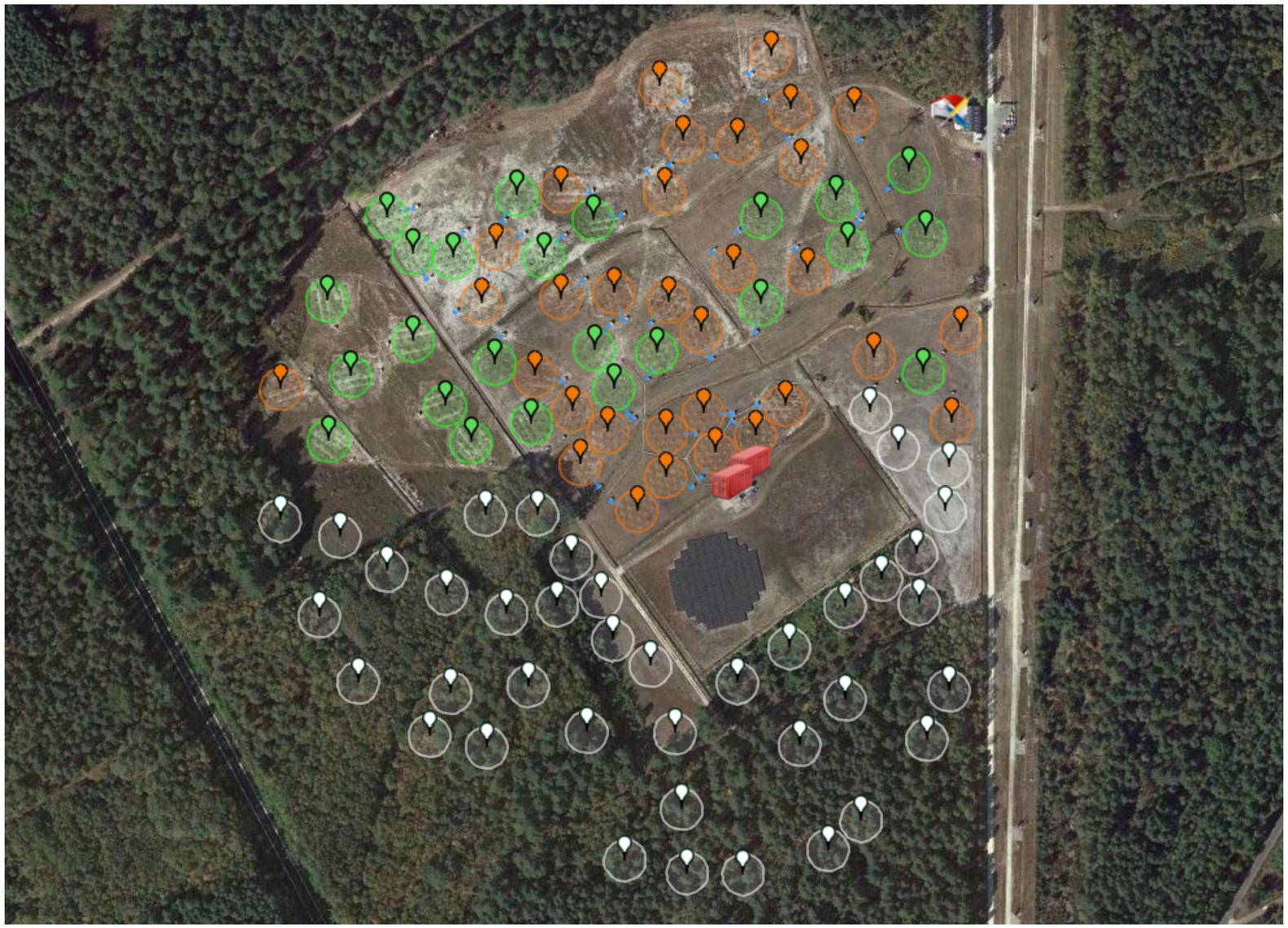


Beam - NenuFAR

$$G_{MA} \sim \Sigma_{\text{mini-arrays}} \left(G_{ant} \Sigma_{\text{antennas}} e^{i\phi(ant)} \right) \Sigma_{\text{mini-arrays}} e^{i\phi(ma)}$$

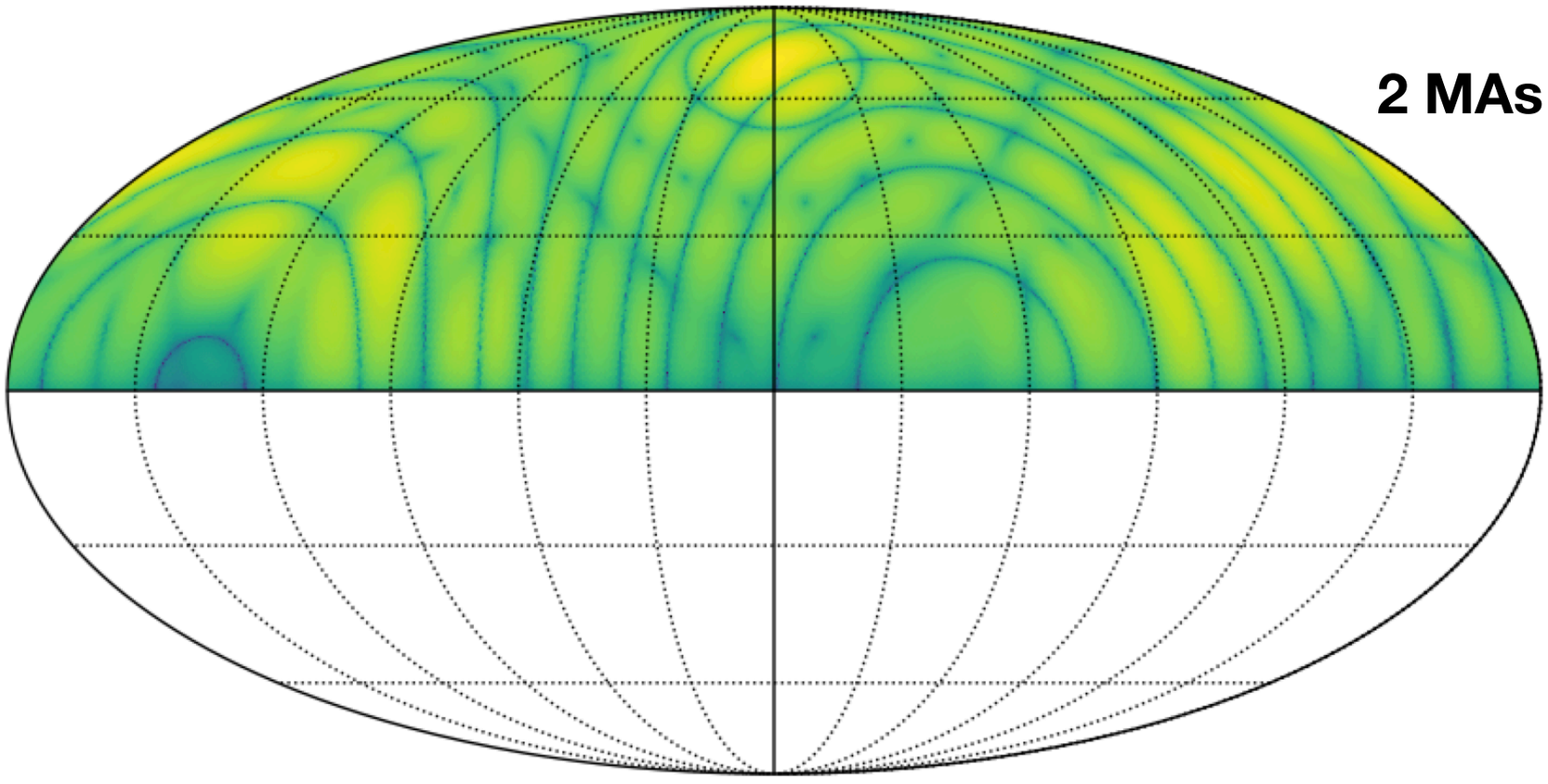
Summed 'antenna' (=MAs) response

Array factor



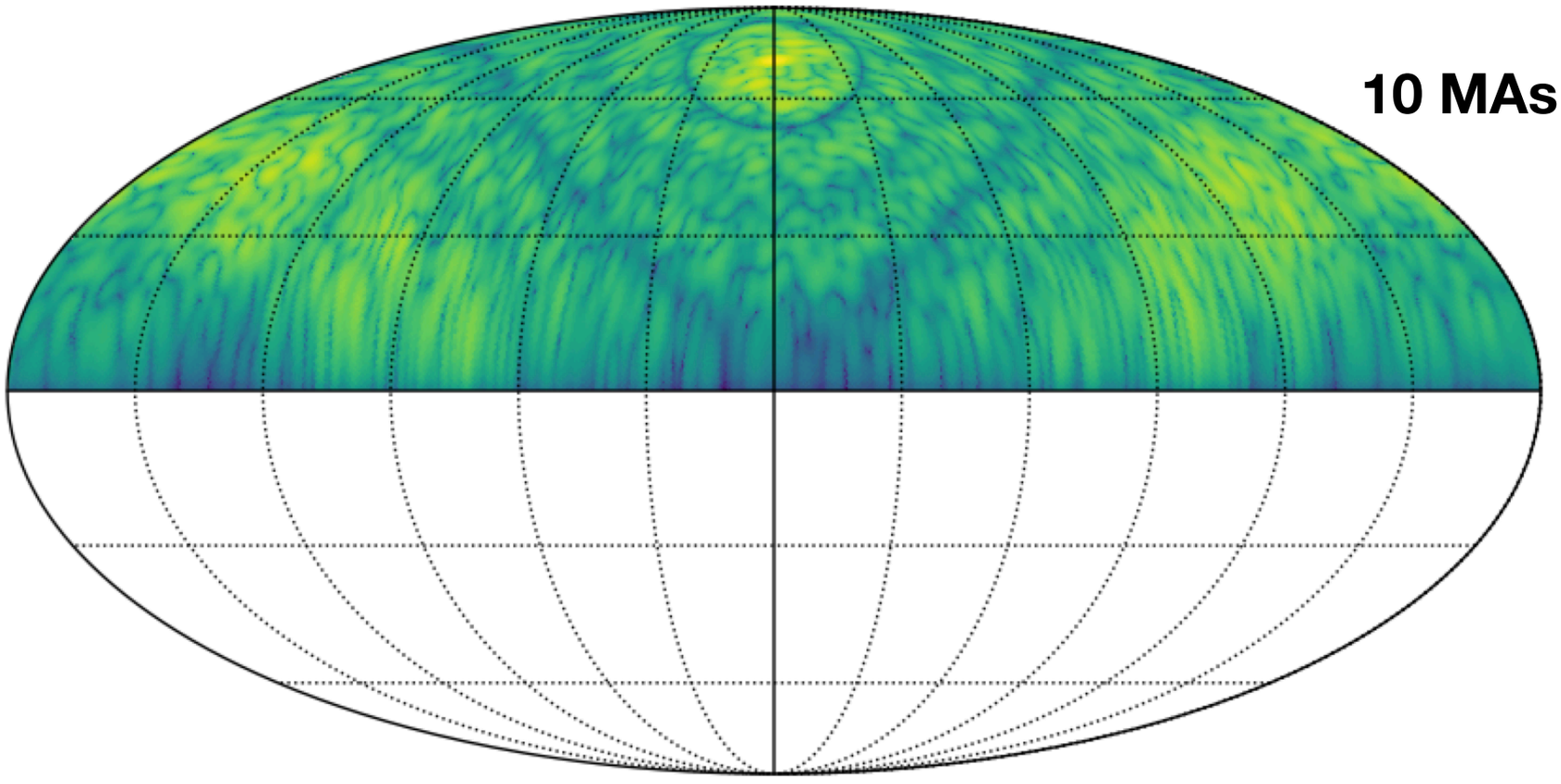
Beam - NenuFAR

```
import healpy as hp
import pylab as plt; import numpy as np
from nenupy.hpx import Digibeam
beam = Digibeam(azana=0, elana=70, azdig=0, eldig=70,
miniarrays=[0, 1], freq=60, polar='NW', resol=0.2)
nenufar_beam = beam.get_digibeam()
hp.mollview(np.log10(nenufar_beam), title='')
hp.graticule(); plt.show()
```



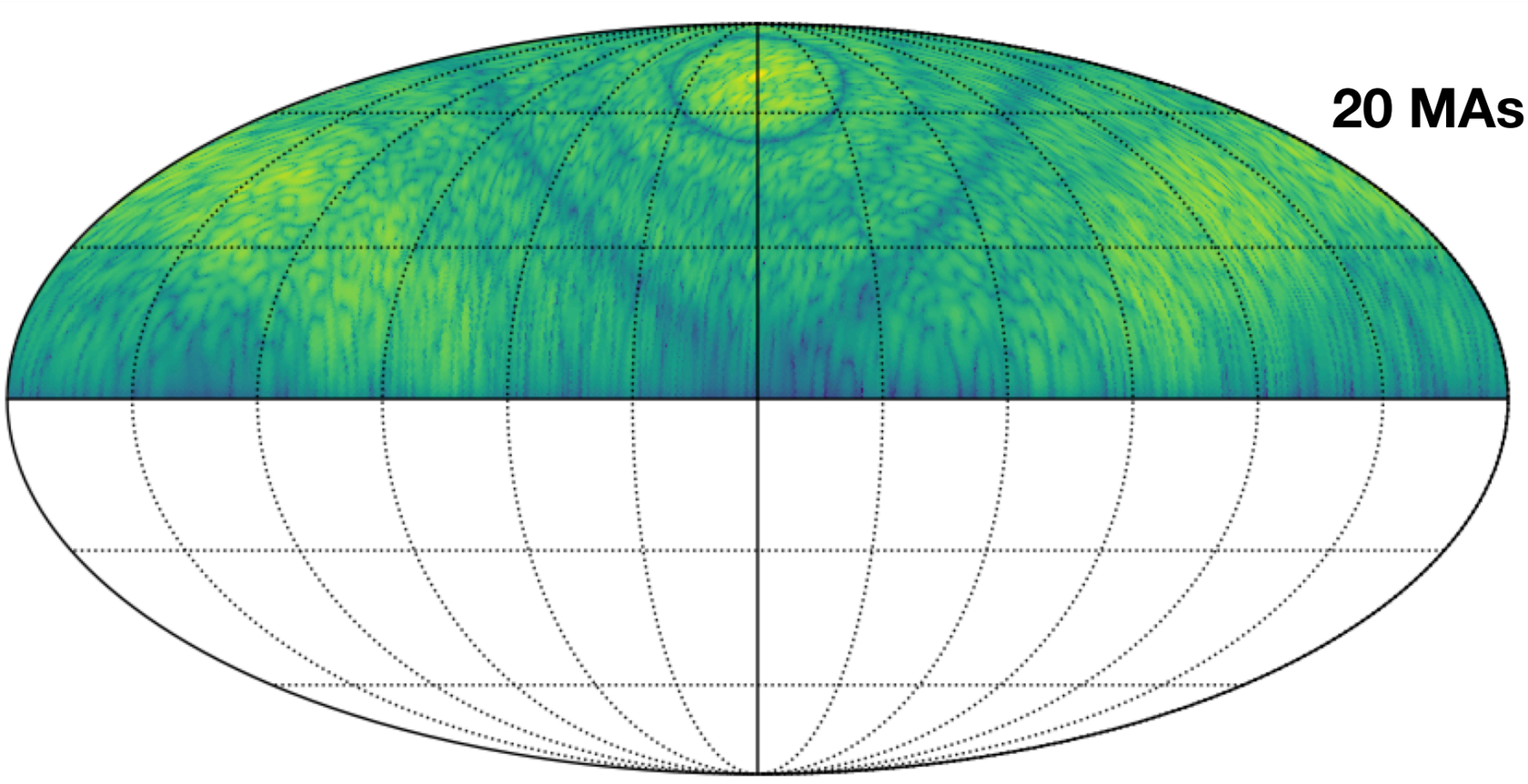
Beam - NenuFAR

```
import healpy as hp
import pylab as plt; import numpy as np
from nenupy.hpx import Digibeam
beam = Digibeam(azana=0, elana=70, azdig=0, eldig=70,
miniarrays=np.arange(10), freq=60, polar='NW', resol=0.2)
nenufar_beam = beam.get_digibeam()
hp.mollview(np.log10(nenufar_beam), title='')
hp.graticule(); plt.show()
```



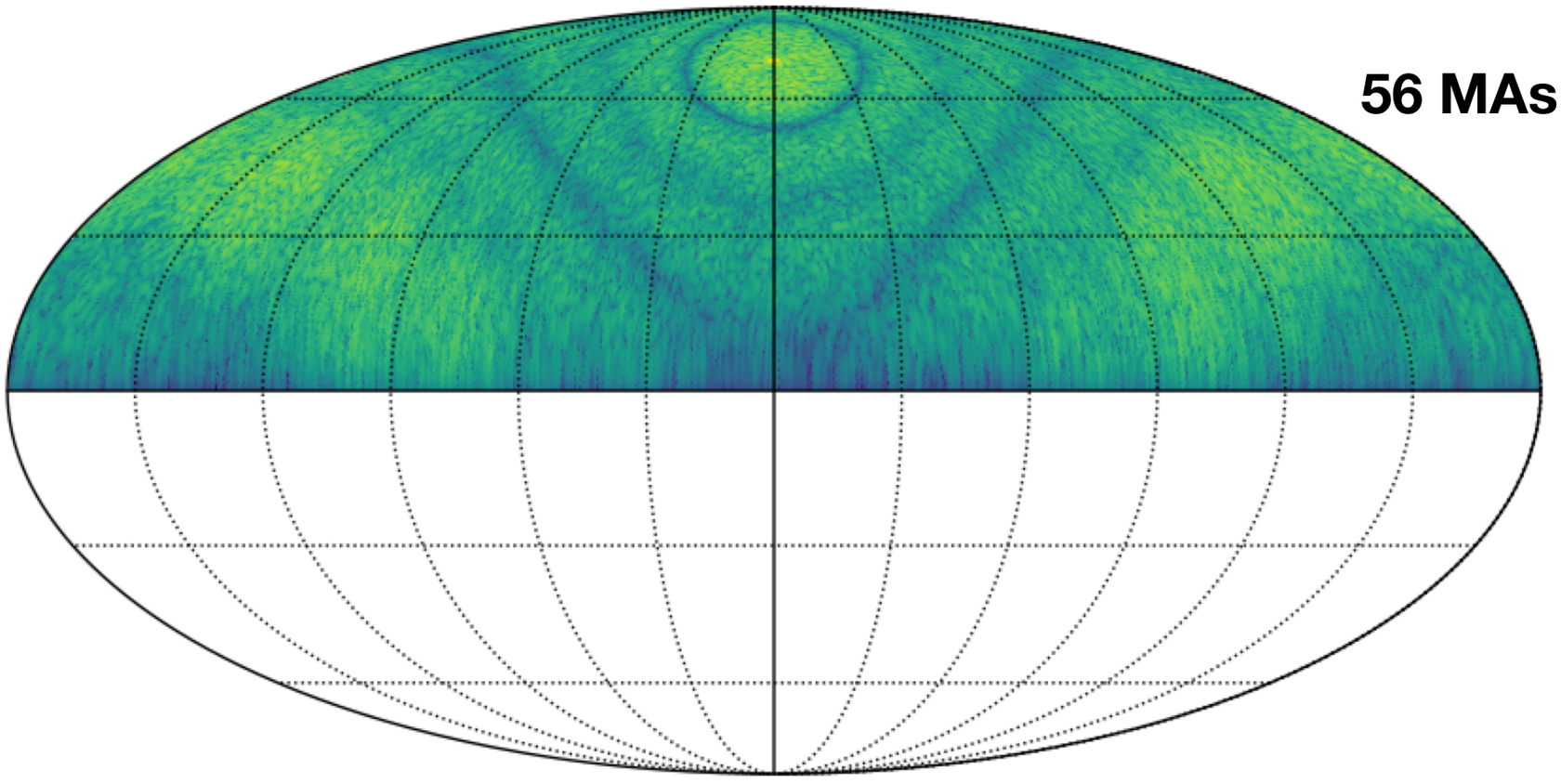
Beam - NenuFAR

```
import healpy as hp
import pylab as plt; import numpy as np
from nenupy.hpx import Digibeam
beam = Digibeam(azana=0, elana=70, azdig=0, eldig=70,
miniarrays=np.arange(20), freq=60, polar='NW', resol=0.2)
nenufar_beam = beam.get_digibeam()
hp.mollview(np.log10(nenufar_beam), title='')
hp.graticule(); plt.show()
```



Beam - NenuFAR

```
import healpy as hp
import pylab as plt; import numpy as np
from nenupy.hpx import Digibeam
beam = Digibeam(azana=0, elana=70, azdig=0, eldig=70,
miniarrays=None, freq=60, polar='NW', resol=0.2)
nenufar_beam = beam.get_digibeam()
hp.mollview(np.log10(nenufar_beam), title='')
hp.graticule(); plt.show()
```

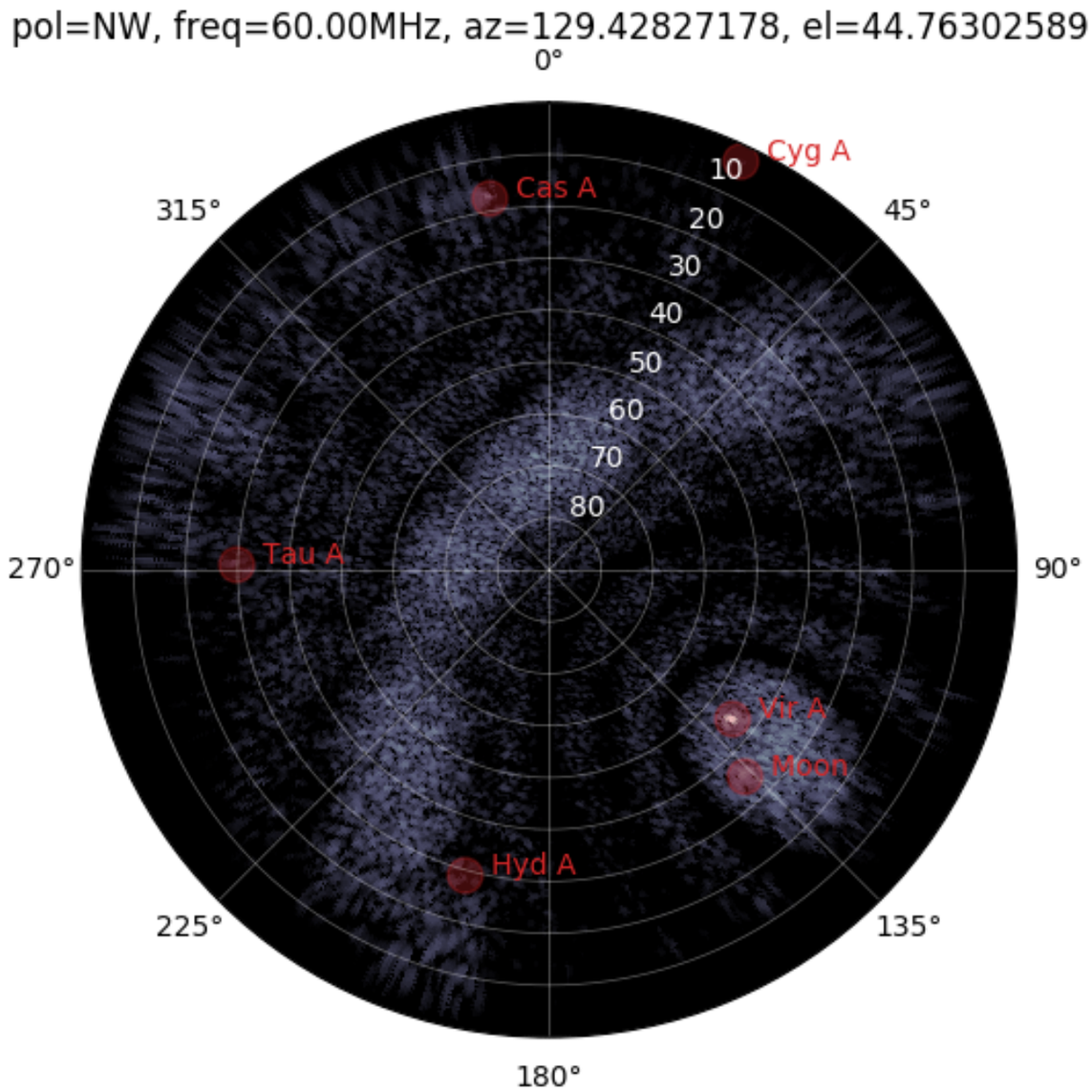


GSM x Beam

```
[[aloh@MacBook-Pro-de-Alan] ~ > nenuskybeam -h
usage: nenuskybeam [-h] -t TIME -s STORE [-f FREQ] [-a AZIMUTH] [-e ELEVATION]
                   [-p POLAR] [-m MA [MA ...]]

optional arguments:
  -h, --help            show this help message and exit
  -t TIME, --time TIME  UTC Time (e.g. 2016-09-30 14:00:00)
  -s STORE, --store STORE
                        Filename (needs to end by .fits/png)
  -f FREQ, --freq FREQ  Frequency in MHz
  -a AZIMUTH, --azimuth AZIMUTH
                        Azimuth in degrees
  -e ELEVATION, --elevation ELEVATION
                        Elevation in degrees
  -p POLAR, --polar POLAR
                        Polarization (NW or NE)
  -m MA [MA ...], --ma MA [MA ...]
                        Mini-Array name indices (e.g. -m -1: all mini-arrays;
                        -m 10 20)
```

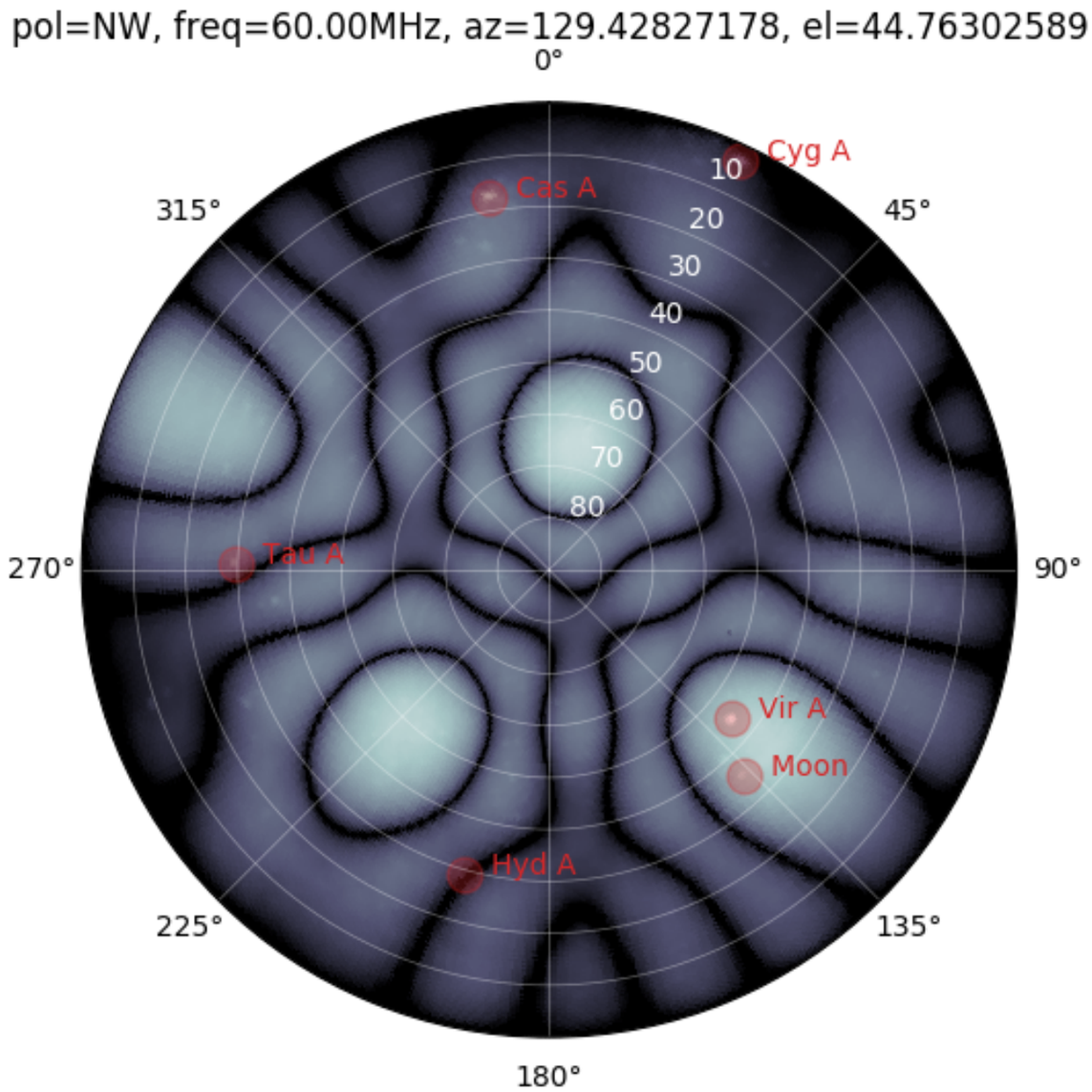
GSM x Beam (whole array)

```
nenuskybeam -t '2019-02-22 00:00:00' -a 129.42827178 -e 44.76302589 -s none -f 60 -m -1
```



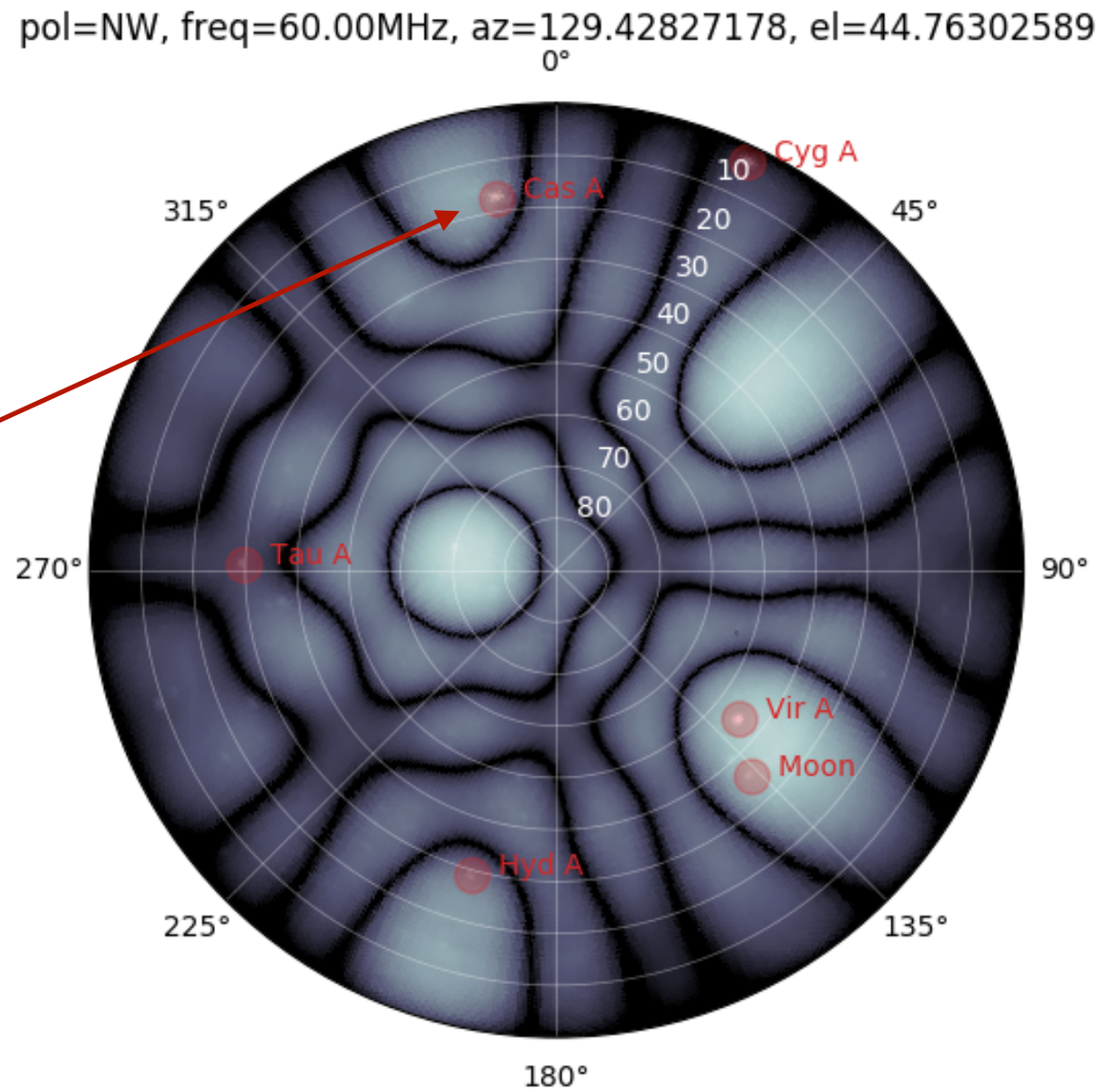
GSM x Beam (MA 1)

```
nenuskybeam -t '2019-02-22 00:00:00' -a 129.42827178 -e 44.76302589 -s none -f 60 -m 1
```



GSM x Beam (MA 0)

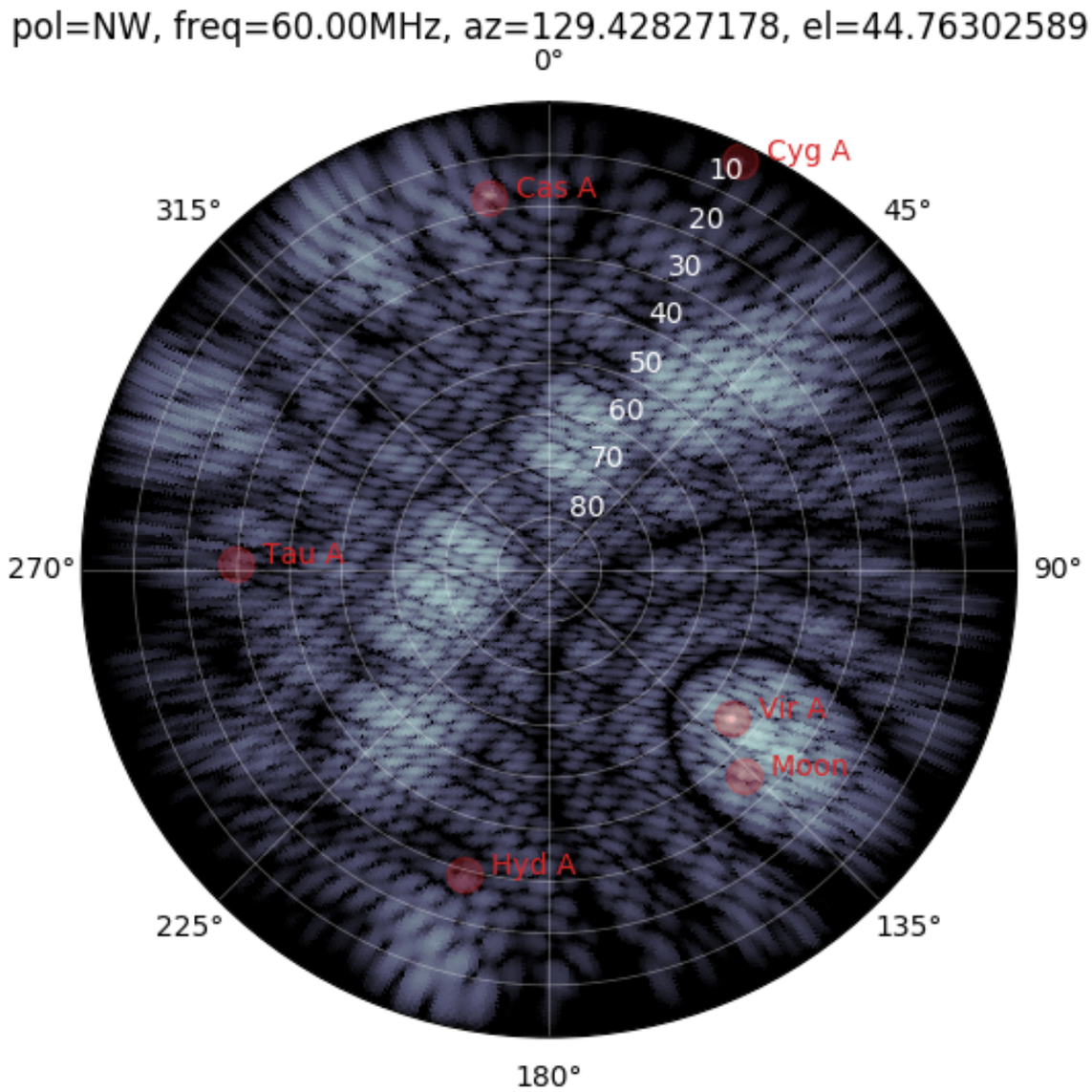
```
nenuskybeam -t '2019-02-22 00:00:00' -a 129.42827178 -e 44.76302589 -s none -f 60 -m 0
```



Cas A in grating lobe

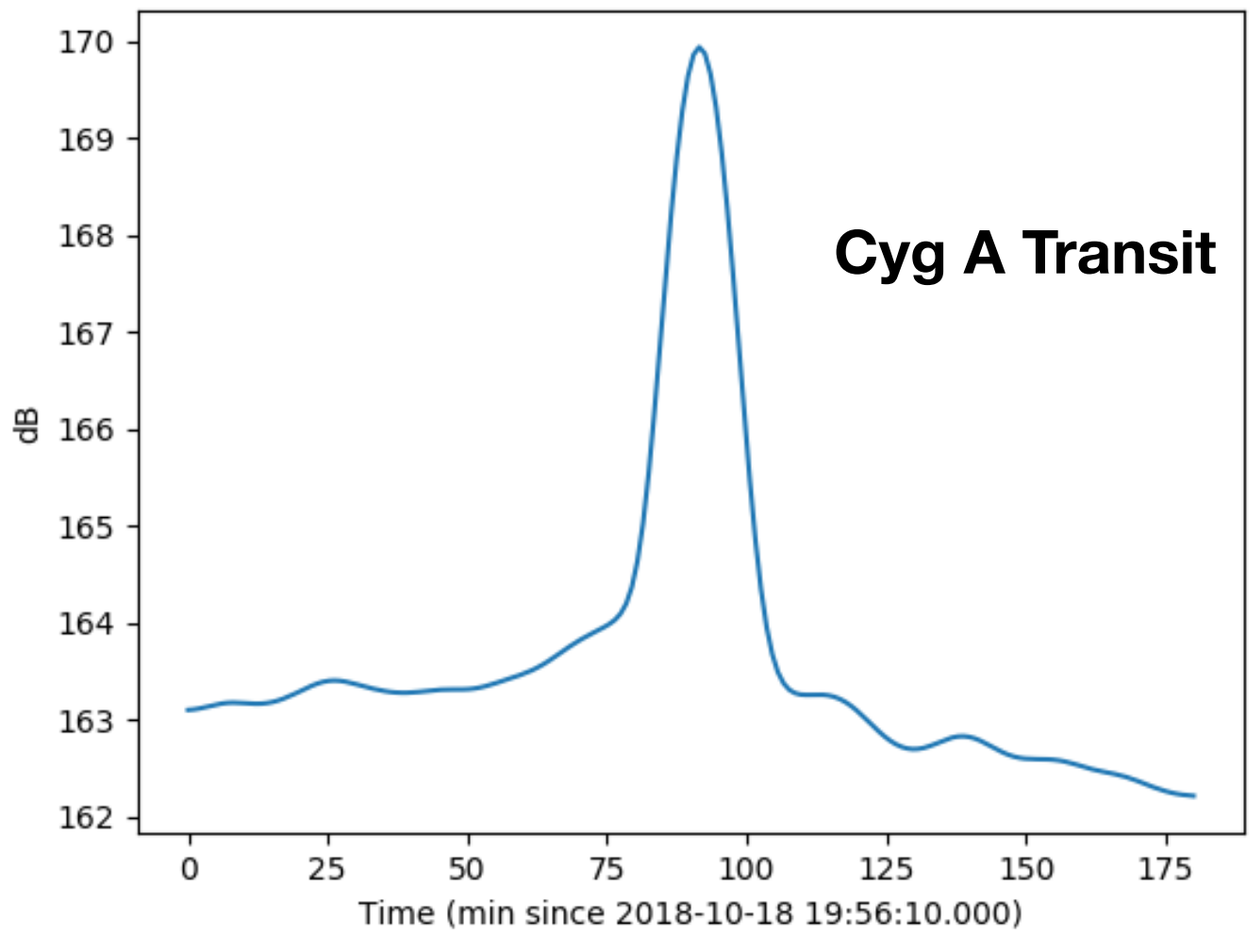
GSM x Beam (arbitrary MAs)

```
nenuskybeam -t '2019-02-22 00:00:00' -a 129.42827178 -e 44.76302589 -s none -f 60 -m 0 1 2 10 34
```



Transit simulation - from obs

```
from nenupy.hpx import Transit
t = Transit()
t.from_bst('20181018_195600_BST.fits', freq=60)
t.plot()
```



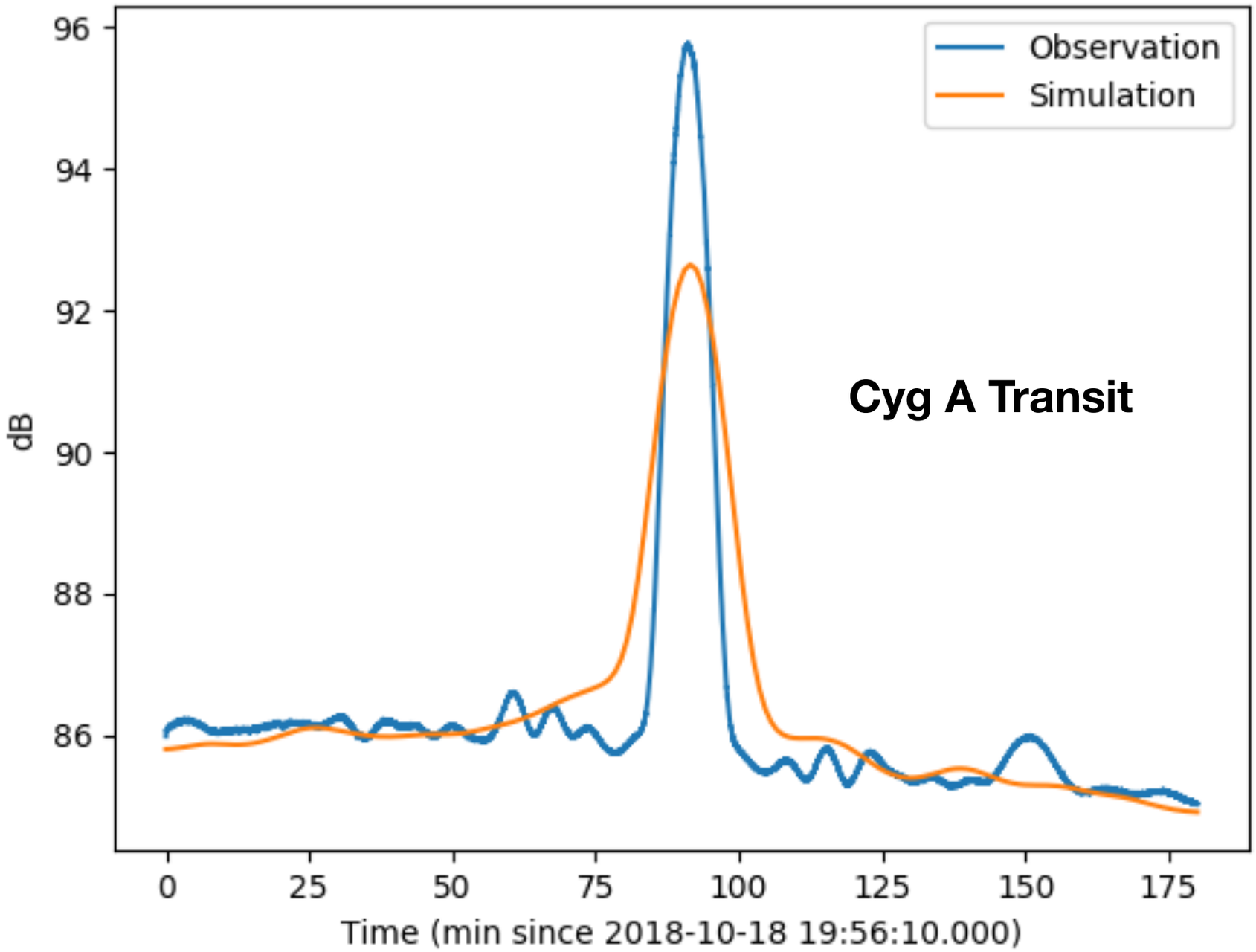
Transit simulation - from obs

```
from nenupy.hpx import Transit
t = Transit()
t.from_bst('20181018_195600_BST.fits', freq=60)
print(t.simulation.keys())
    dict_keys(['time', 'amp'])
print(t.simulation['amp'].size)
    180
print(t.simulation['time'][0:10])
    [58409.8306713  58409.83136956 58409.83206782 58409.83276607
     58409.83346433 58409.83416259 58409.83486085 58409.83555911
     58409.83625737 58409.83695563]
print(t.simulation['time'][0:10].iso)
    ['2018-10-18 19:56:10.000' '2018-10-18 19:57:10.330'
     '2018-10-18 19:58:10.659' '2018-10-18 19:59:10.989'
     '2018-10-18 20:00:11.318' '2018-10-18 20:01:11.648'
     '2018-10-18 20:02:11.978' '2018-10-18 20:03:12.307'
     '2018-10-18 20:04:12.637' '2018-10-18 20:05:12.966']
```

Transit simulation - from obs

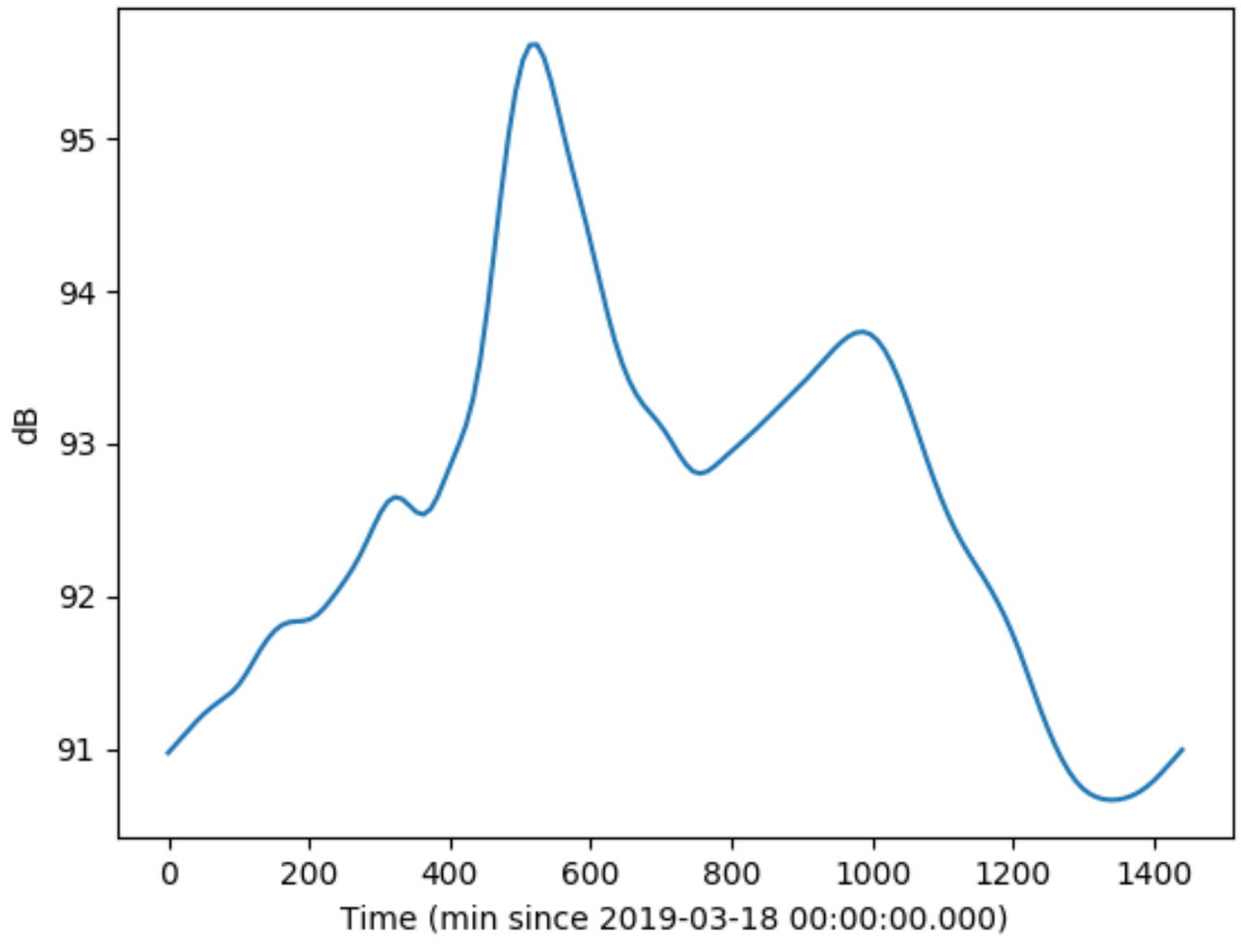
```
from nenupy.hpx import Transit
from nenupy import BST
import pylab as plt; import numpy as np

bst = BST('20181018_195600_BST.fits')
bst.select(freq=60)
t = Transit()
t.from_bst(bst, freq=60, dt=60)

scale = np.median(bst.data['amp']) / np.median(t.simulation['amp'])
plt.plot( (bst.data['time']-t.simulation['time'][0]).sec/60,
10*np.log10(bst.data['amp']), label='Observation' )
plt.plot( (t.simulation['time']-t.simulation['time'][0]).sec/60,
10*np.log10(t.simulation['amp']*scale), label='Simulation' )
plt.legend()
plt.xlabel('Time (min since {})'.format(t.simulation['time'][0].iso))
plt.ylabel('dB')
plt.show()
```

Transit simulation - from obs



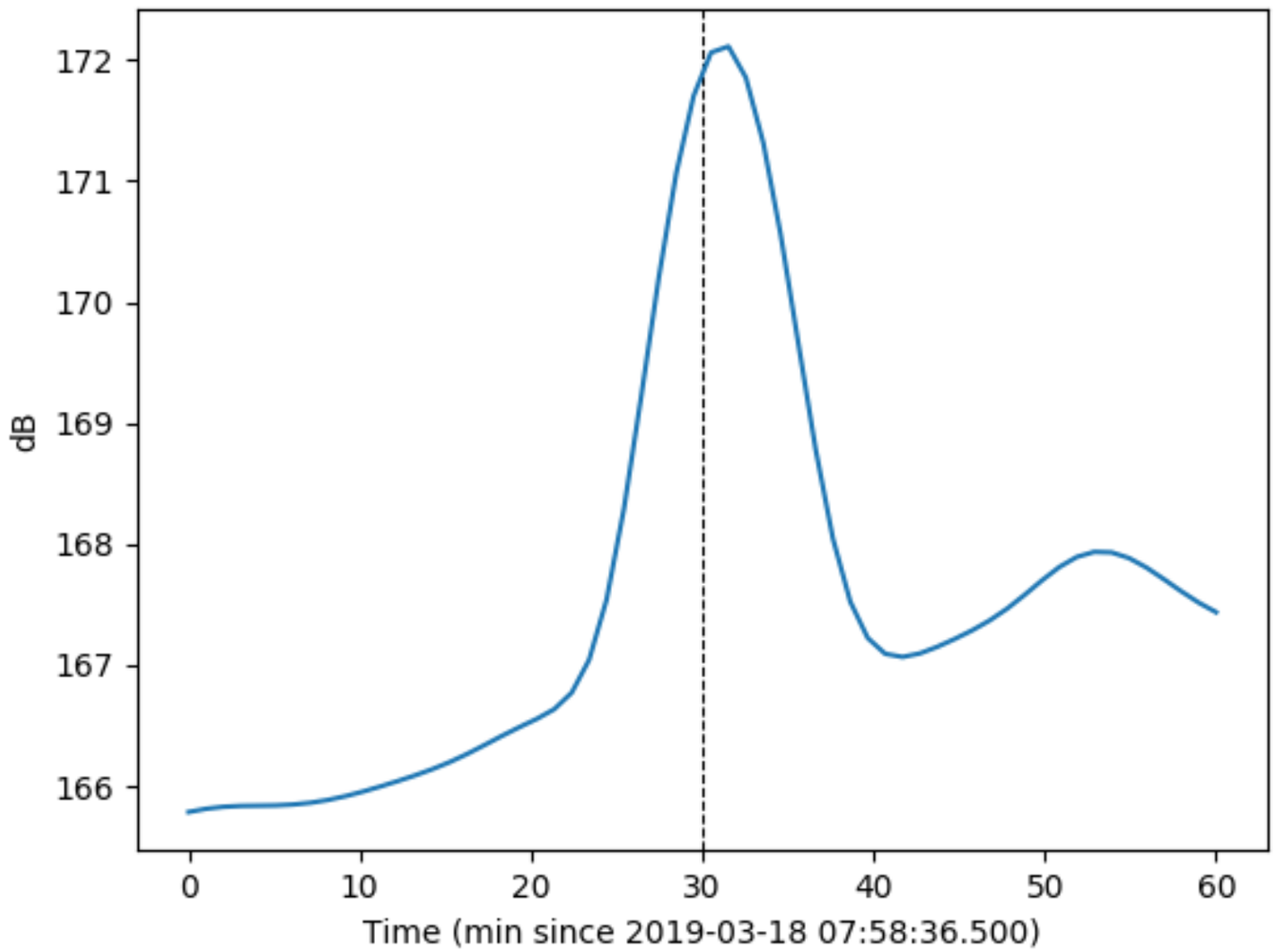
Transit simulation - predict

```
from nenupy.hpx import Transit
t=Transit()
t.predict(src=None, time='2019-03-18 00:00:00', resol=1, freq=55,
duration=86400, miniarrays=[10], azana=180, elana=90, azdig=180, eldig=90,
dt=600)
t.plot()
```



Transit simulation - predict

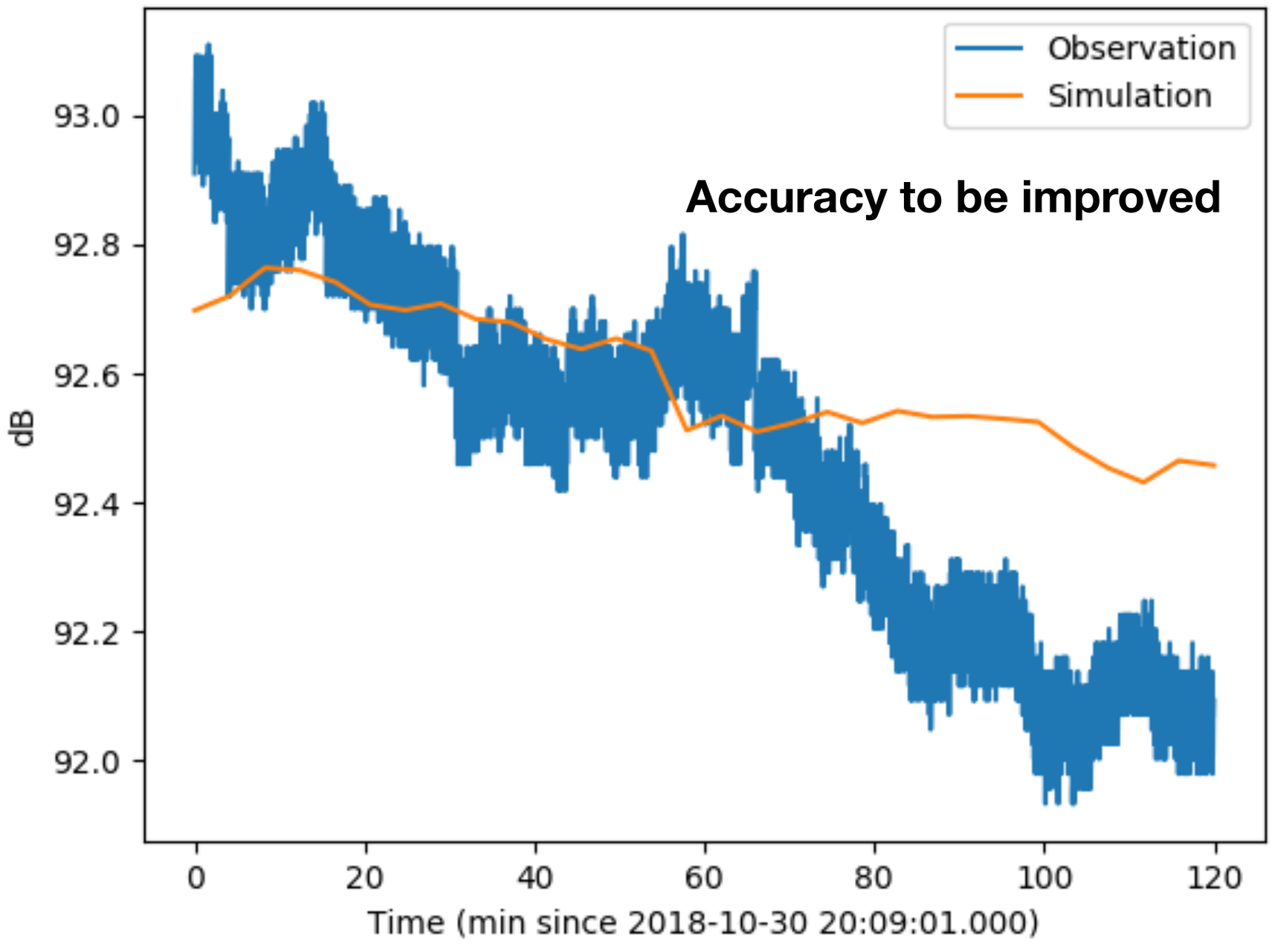
```
from nenupy.hpx import Transit
t=Transit()
t.predict(src='Cyg A', az=210, time='2019-03-18 00:00:00', resol=0.2,
freq=65, duration=3600, miniarrays=None, dt=60)
t.plot()
```



Tracking simulation - from obs

```
from nenupy.hpx import Tracking
from nenupy import BST
import pylab as plt; import numpy as np

bst = BST('20181030_200800_BST.fits')
bst.select(freq=60)
t = Tracking()
t.from_bst(bst, freq=60, dt=240, resol=0.2)

scale = np.median(bst.data['amp']) / np.median(t.simulation['amp'])
plt.plot( (bst.data['time']-t.simulation['time'][0]).sec/60,
10*np.log10(bst.data['amp']), label='Observation' )
plt.plot( (t.simulation['time']-t.simulation['time'][0]).sec/60,
10*np.log10(t.simulation['amp']*scale), label='Simulation' )
plt.legend()
plt.xlabel('Time (min since {})'.format(t.simulation['time'][0].iso))
plt.ylabel('dB')
plt.show()
```

Tracking simulation - from obs



Thanks!