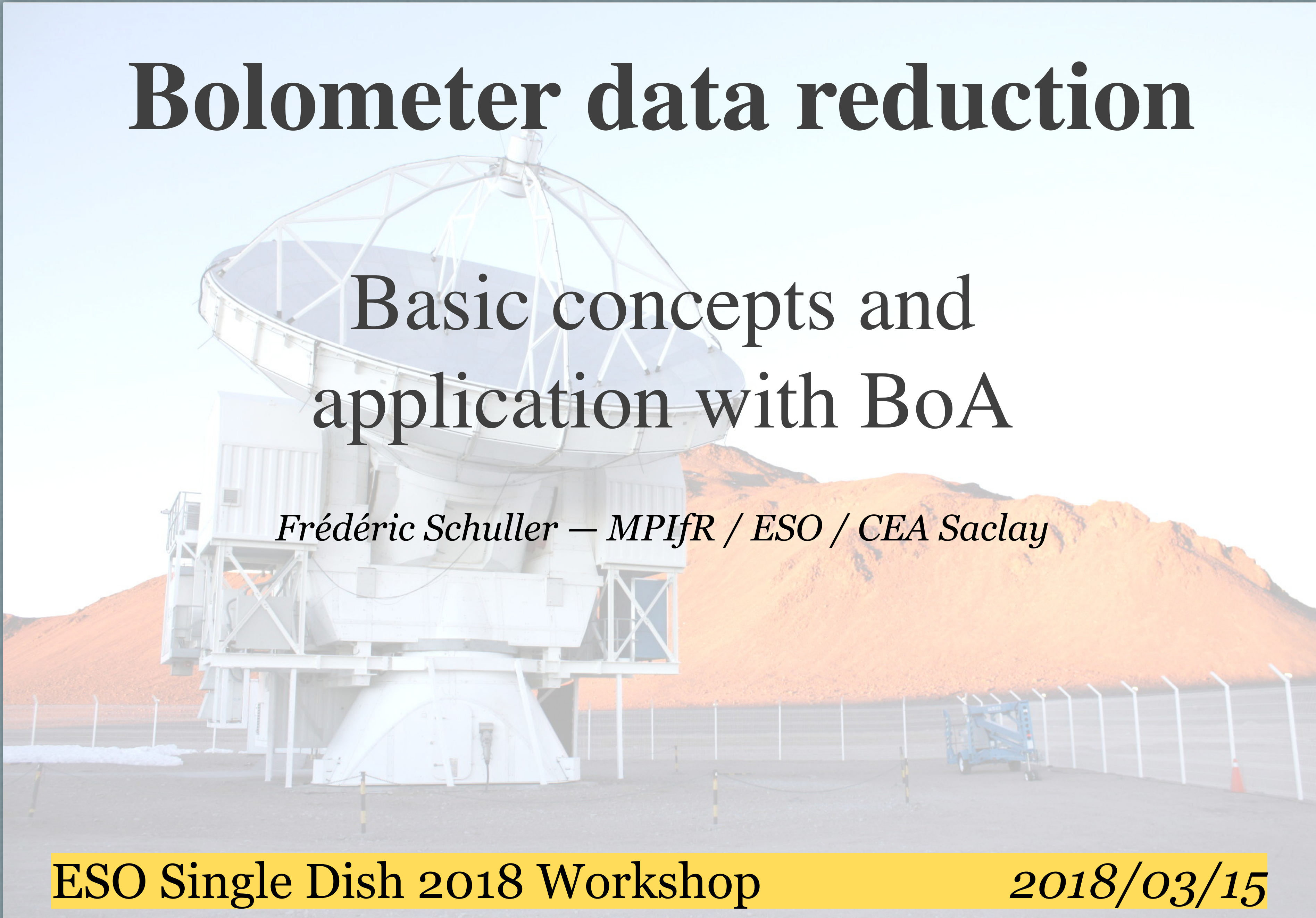


Bolometer data reduction

Basic concepts and application with BoA

Frédéric Schuller — MPIfR / ESO / CEA Saclay

ESO Single Dish 2018 Workshop *2018/03/15*

Introduction

- Bolometers = thermometers, sensitive to **all** incoming radiation

⇒ bolometer signal = sky + astronomical source + noise

Introduction

- Bolometers = thermometers, sensitive to **all** incoming radiation

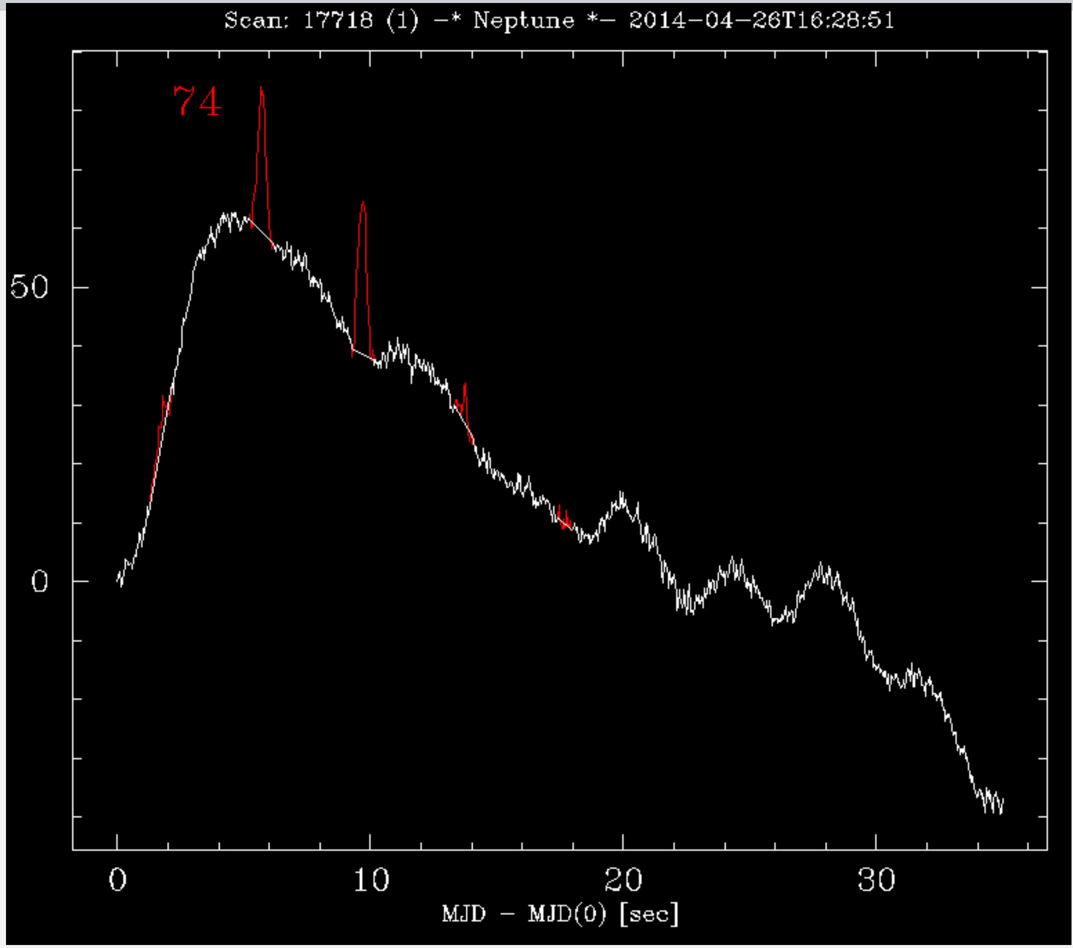
⇒ bolometer signal = sky + astronomical source + noise

- Critical steps in bolometer data reduction :

➢ extract astronomical signal from [NOISE + signal]

➢ flux calibration (= good estimate of sky opacity)

Skynoise and signal



Variations of sky emission = **skynoise**

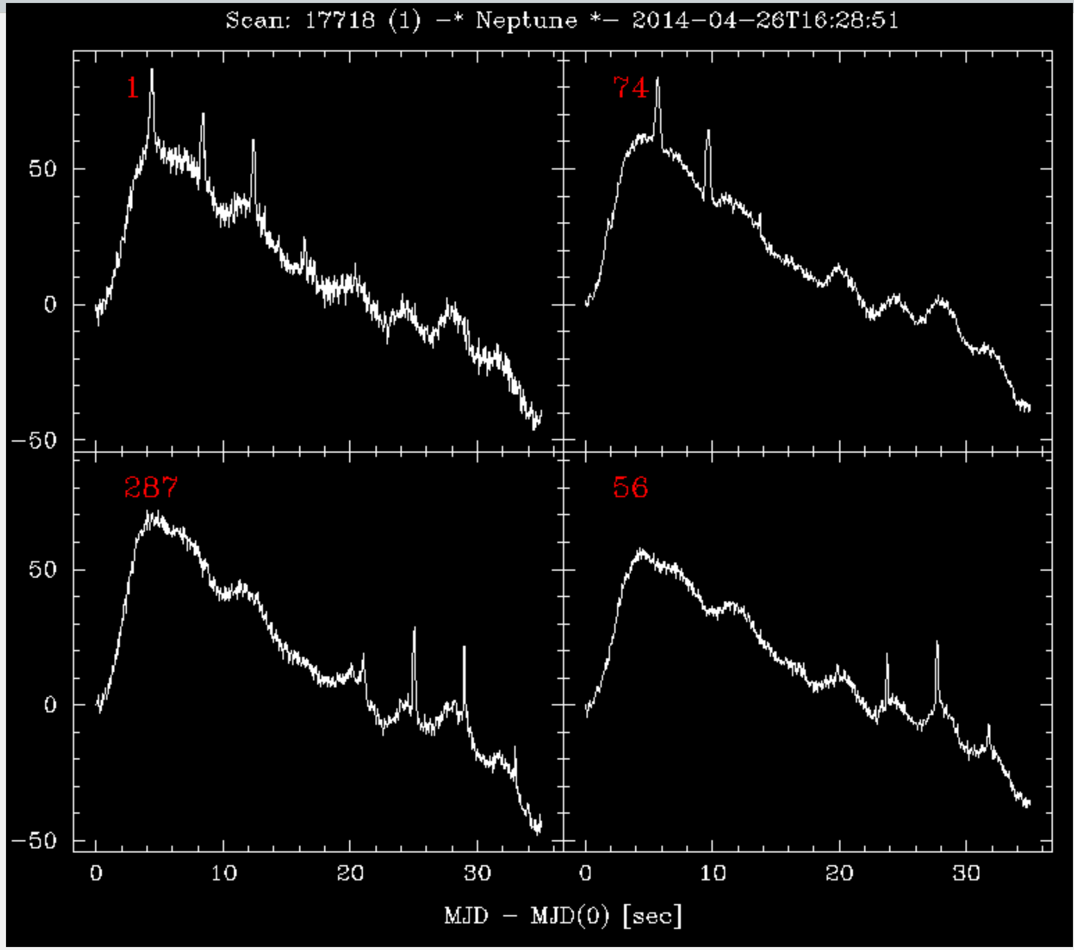
Amplitude: typically 10s or 100s of Jy

Neptune: 23 Jy

Pointing source: ~1 Jy

Science target << 1 Jy

Skynoise and signal



Skynoise is **correlated** between bolometers

The source is seen by different bolometers at different times

Flux calibration

- Data recorded in some instrumental unit (counts)

➢ Convert to astronomical unit: Jy, or Jy/beam

➢ Need to correct for absorption in atmosphere :

→ Estimate zenithal opacity, tau_z

→ Correct for line-of-sight opacity

$$tau_{LOS} = tau_z / \sin(El)$$

$$F(real) = F(obs) * \exp(tau_{LOS})$$

Flux calibration

Data recorded in some instrumental unit (counts)

Convert to astronomical unit: Jy, or Jy/beam

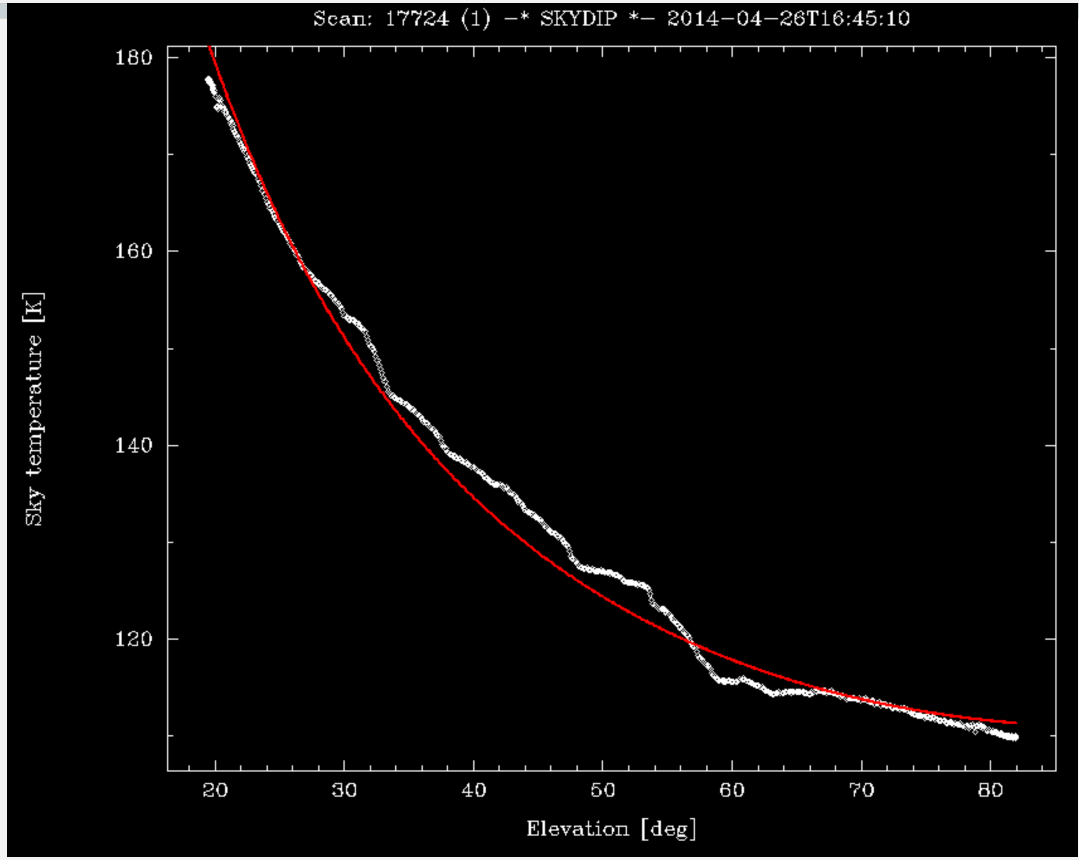
- Need to correct for absorption in atmosphere :
  - → Estimate zenithal opacity, tau_z

Opacity estimate

- Skydips : measure sky emission versus elevation
- Radiometer : PWV + instrument bandpass ➔ tau
- Primary and secondary calibrators:
  - primary = planets (Mars, Uranus, Neptune)
  - secondary = bright, stable sources
- Use skydips + calibrators observed with your project

(LABOCA: see also online interface)

Opacity estimate : bolotip



But… usually underestimated => tau = tau_{SD} x 1.3

(*for LABOCA*)

Calibrators

- Primary calibrators, example: Neptune

```
> redcal(17718)
  (...)

---------------------------------------------------------------
--
   FLUX Neptune:     28.30 [expected:    23.09, 122.53
percent]

---------------------------------------------------------------
--
```

Calibrators

- Secondary calibrators, example: N2071IR

> redcal(17752)

```
(...)

----------------------------------------------------------------
--
   FLUX N2071IR:      9.82 [expected:     9.10, 107.86
percent]

----------------------------------------------------------------
--
```

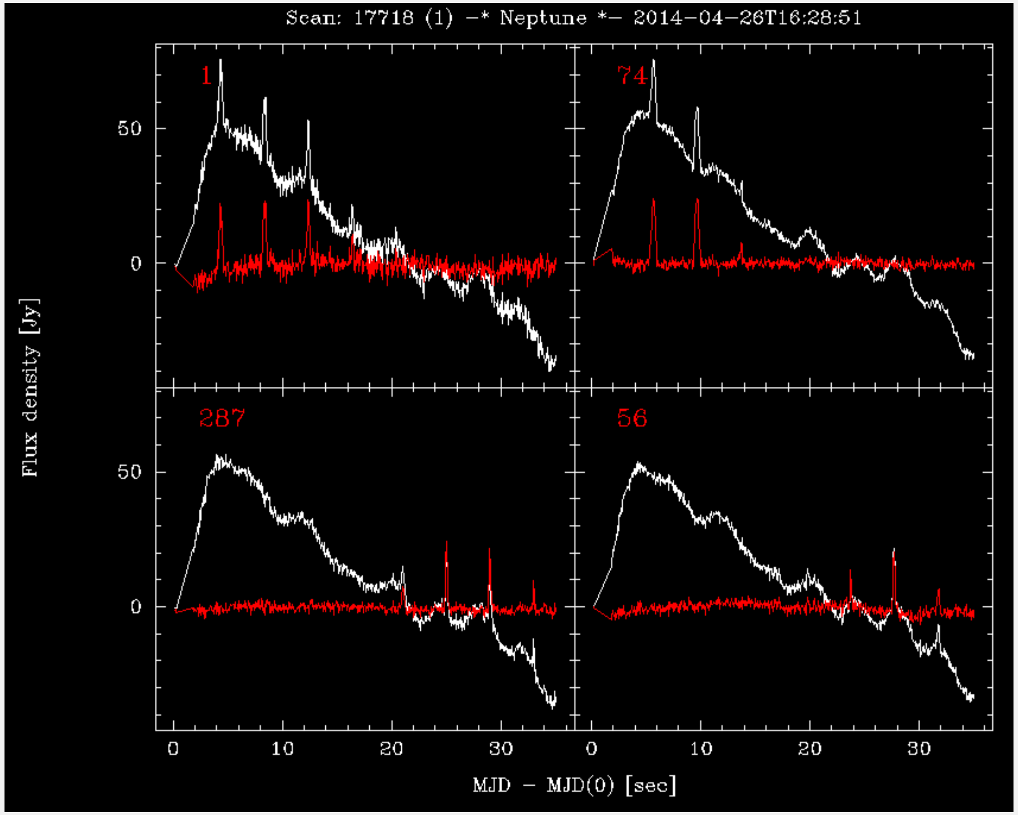
Data reduction

Step by step…

Correlated noise removal

In BoA:
sky emission
=
median of
bolometer
signal

```
> medianNoiseRemoval(…)
```



Correlated noise removal

LABOCA-specific

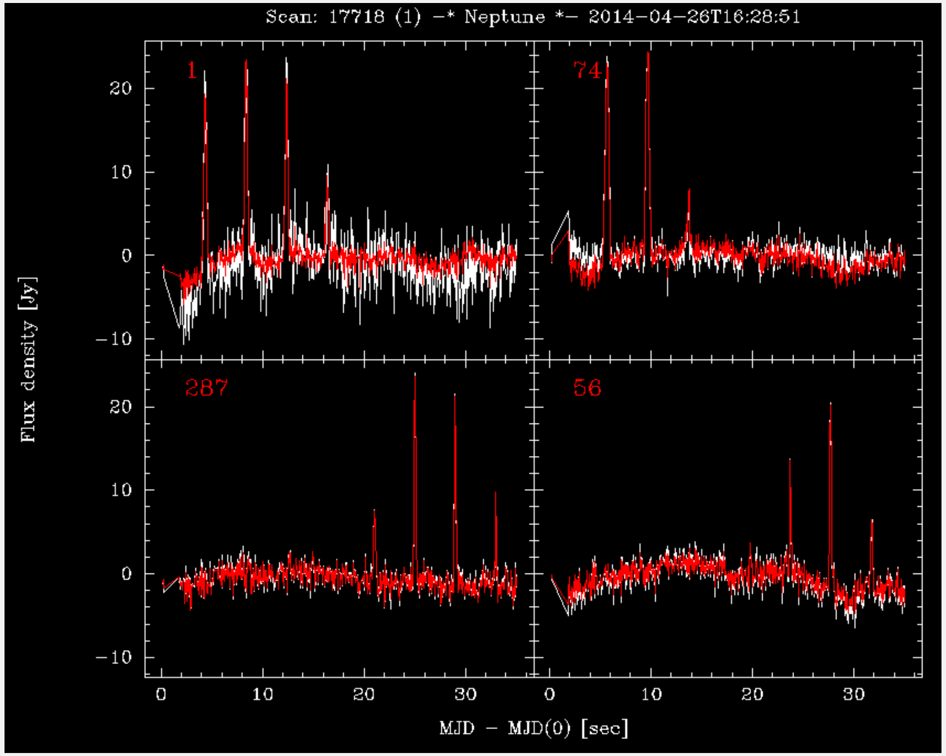
```
> # Amplifier box: 80 bolometers connected
> correlbox(data,factor=0.9,nbloop=2)

> # Groups = 26 bolometers connected to same cable
> correlgroup(data,factor=0.9,nbloop=2)
```

ArTéMiS

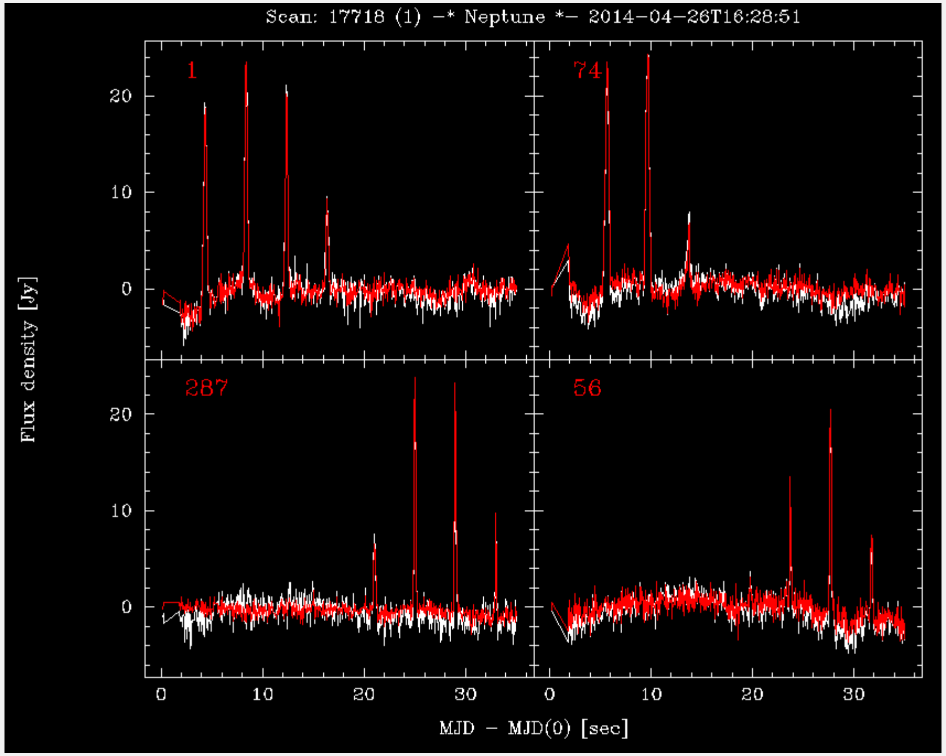
- Correlated signal per sub-array
- Rows and columns in each sub-array (caution with extended sources)

Correlated noise removal

correlbox



correlgroup

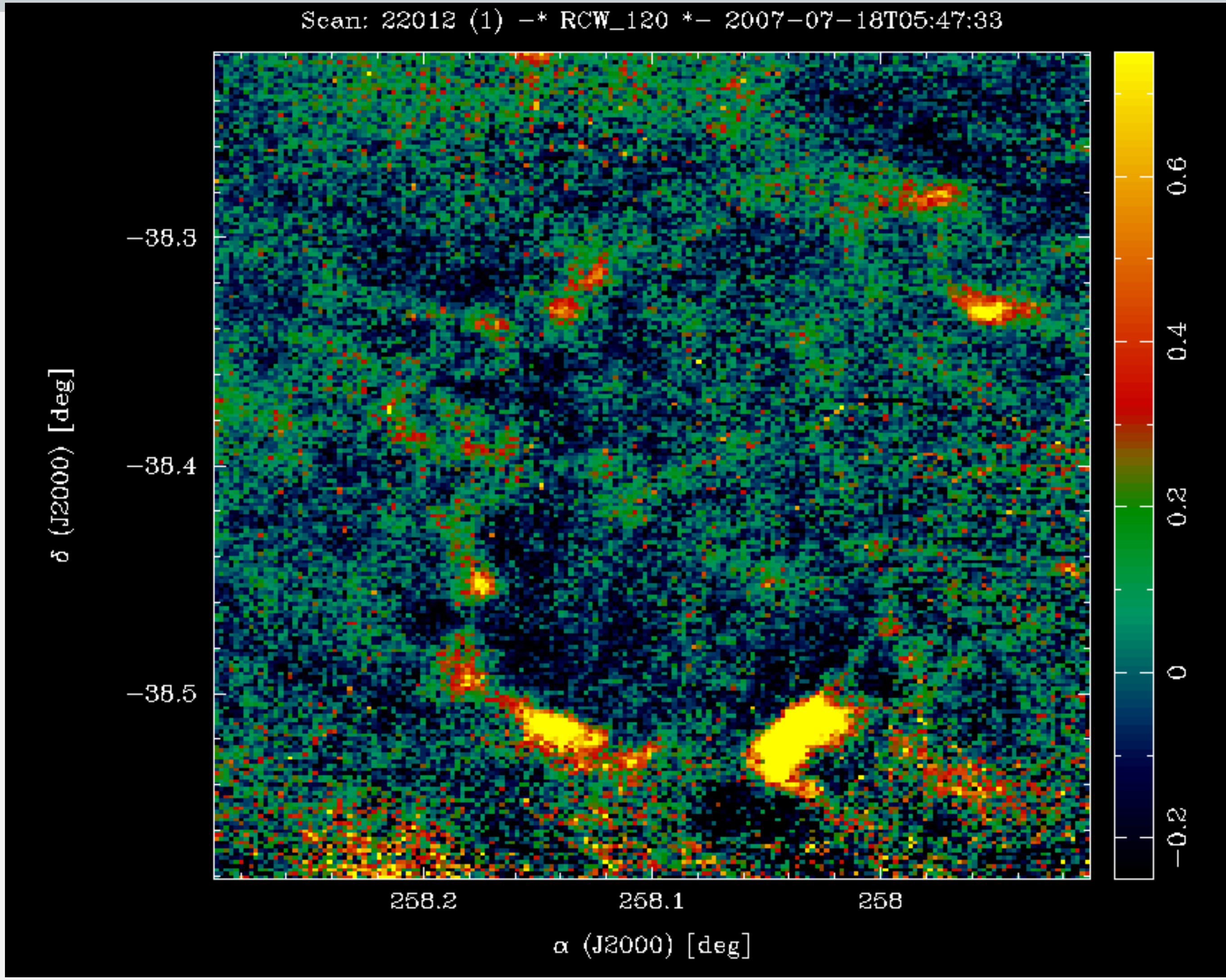


Final steps

- flag noisy channels
- despiking (**WARNING** with bright sources)
- high-pass filter (**WARNING** with extended emission)
- computing weights ($1/\sigma^2$)
- compute map in EQ (or GAL, or HO, or…)

BoA-specific : the map limits have to be input by the user if several scans are going to be combined

Build a map



Scan: 22012 (1) -* RCW_120 *- 2007-07-18T05:47:33
−38.3
−38.4
−38.5
δ (J2000) [deg]
258.2
258.1
258
α (J2000) [deg]
0.6
0.4
0.2
0
−0.2

Combine several scans

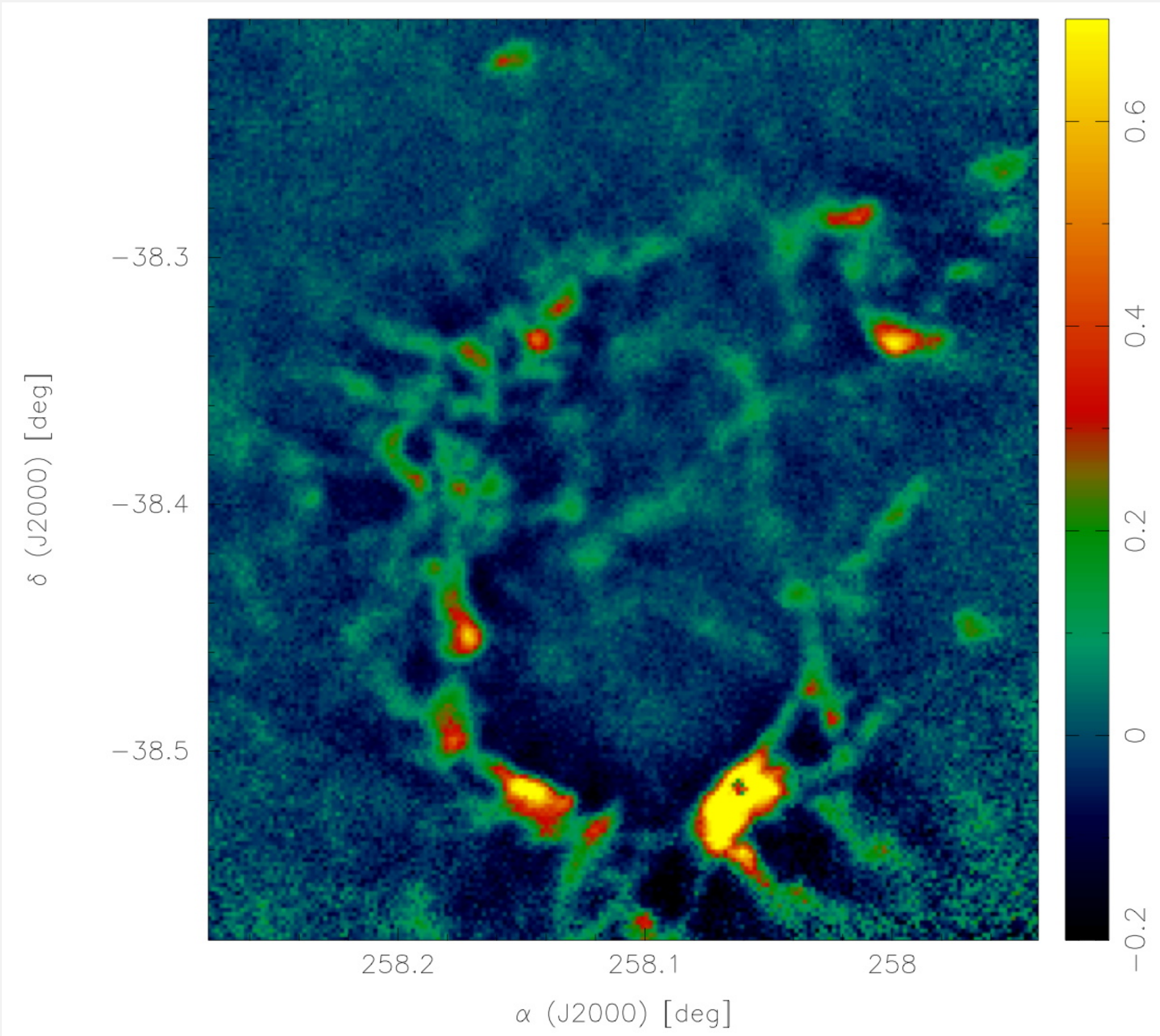
- Processing of individual scans (calibration, skynoise removal, despiking, weighting, …)
- Compute a map for each scan
- Co-add maps with proper weighting

- BoA: co-add individual maps with ‘mapsumfast’

```
> mtot = mapsumfast[mapList]
```

- Finally, display / save / export the results

Combined map



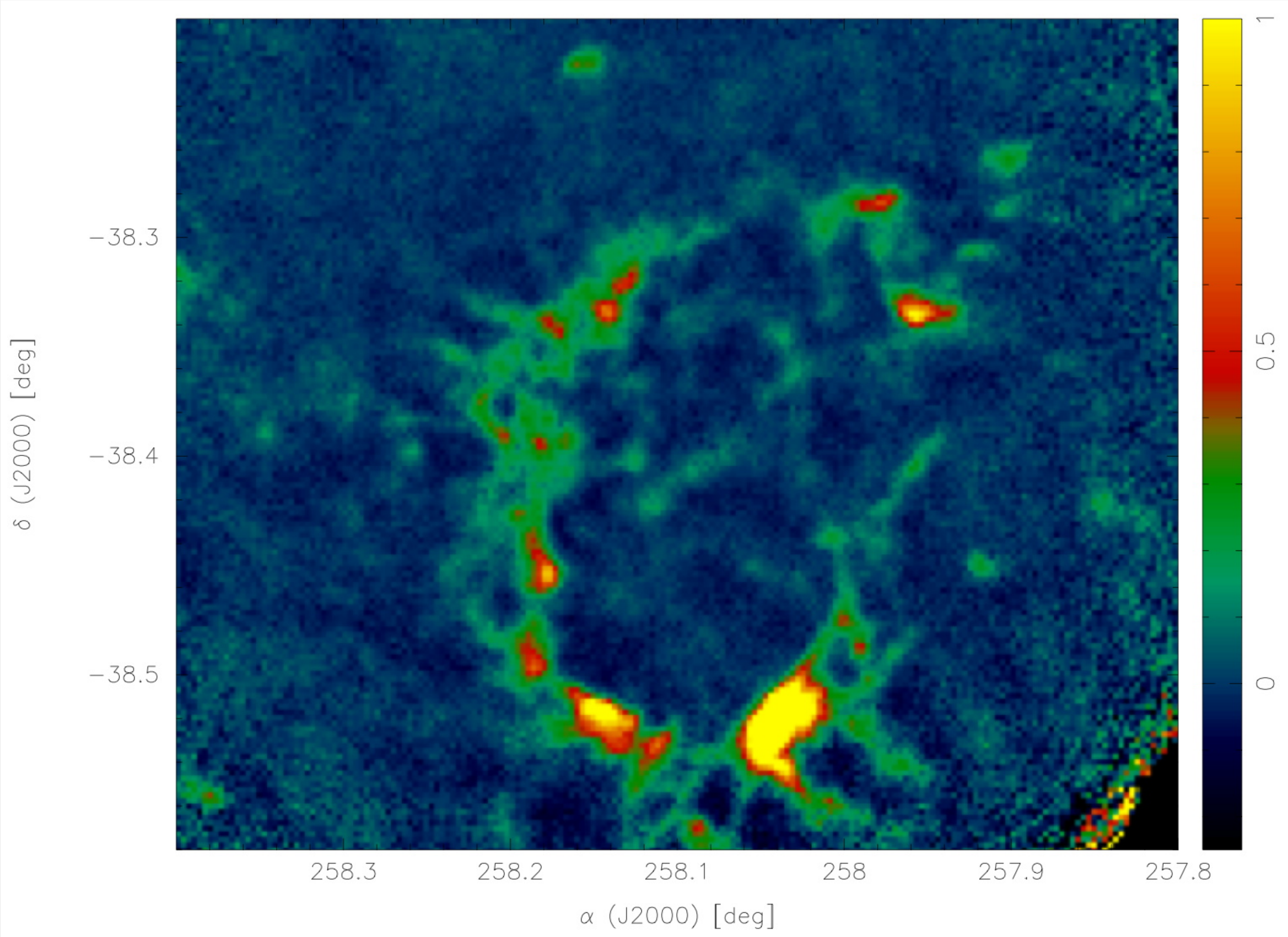
Iterative reduction

- Use combined map, or S/N map, to mask the data
- Process each scan individually
- Then combine all scans as before
- New result (combined map) = input for next iteration
- Can also **subtract** a map in the data

  ⇒ all data considered in all steps

  (masked data are **not** considered, but results of all operations also applied)

Combined map: 2nd iteration



Some more useful functions

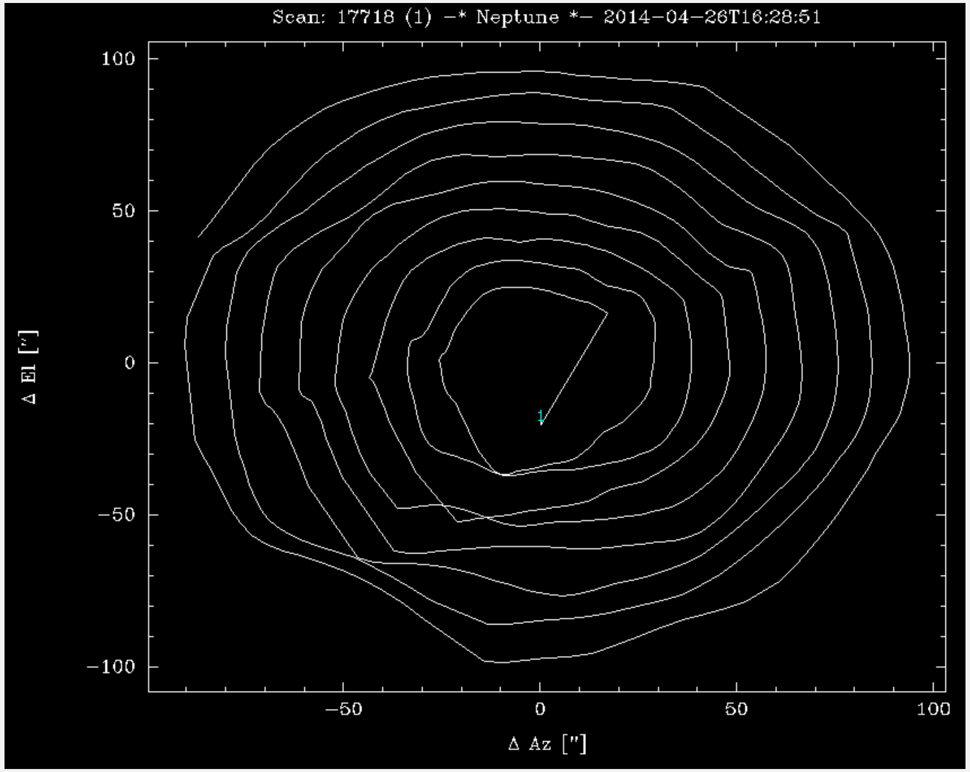
- Plot bolometer signals

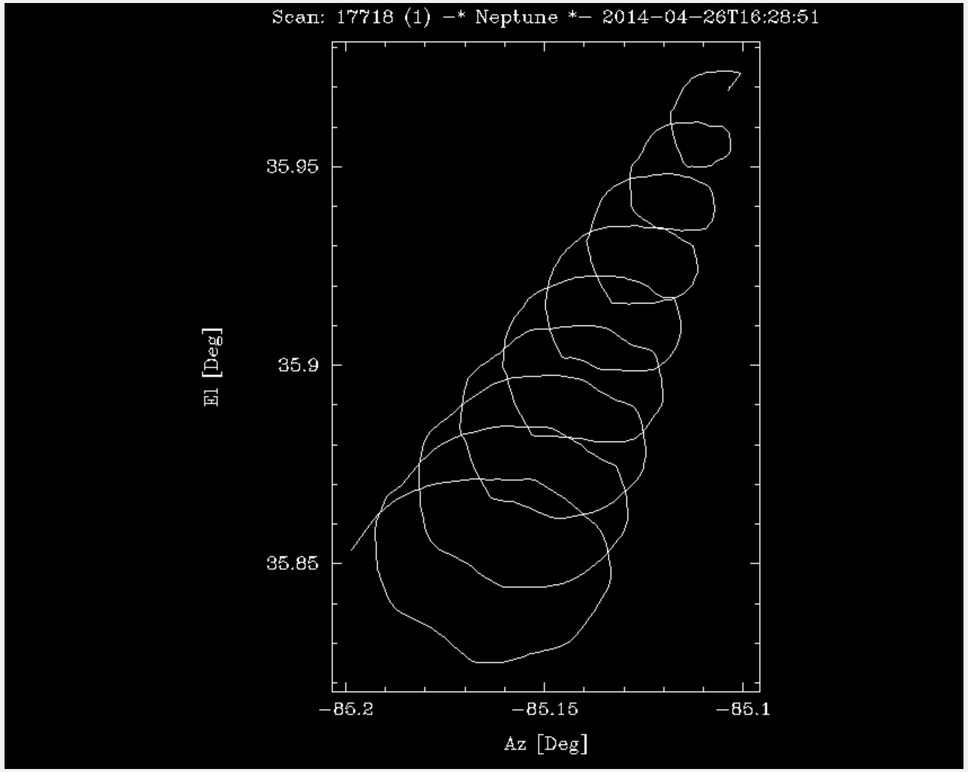
```
> sig(74)
> sig(range(50,100))
```

- Plot telescope pattern

```
> azelo()
> azel()
```

Some more useful functions





Some more useful functions

- Plot bolometer signals

```
> sig(74)
> sig(range(50,100))
```

- Plot telescope pattern

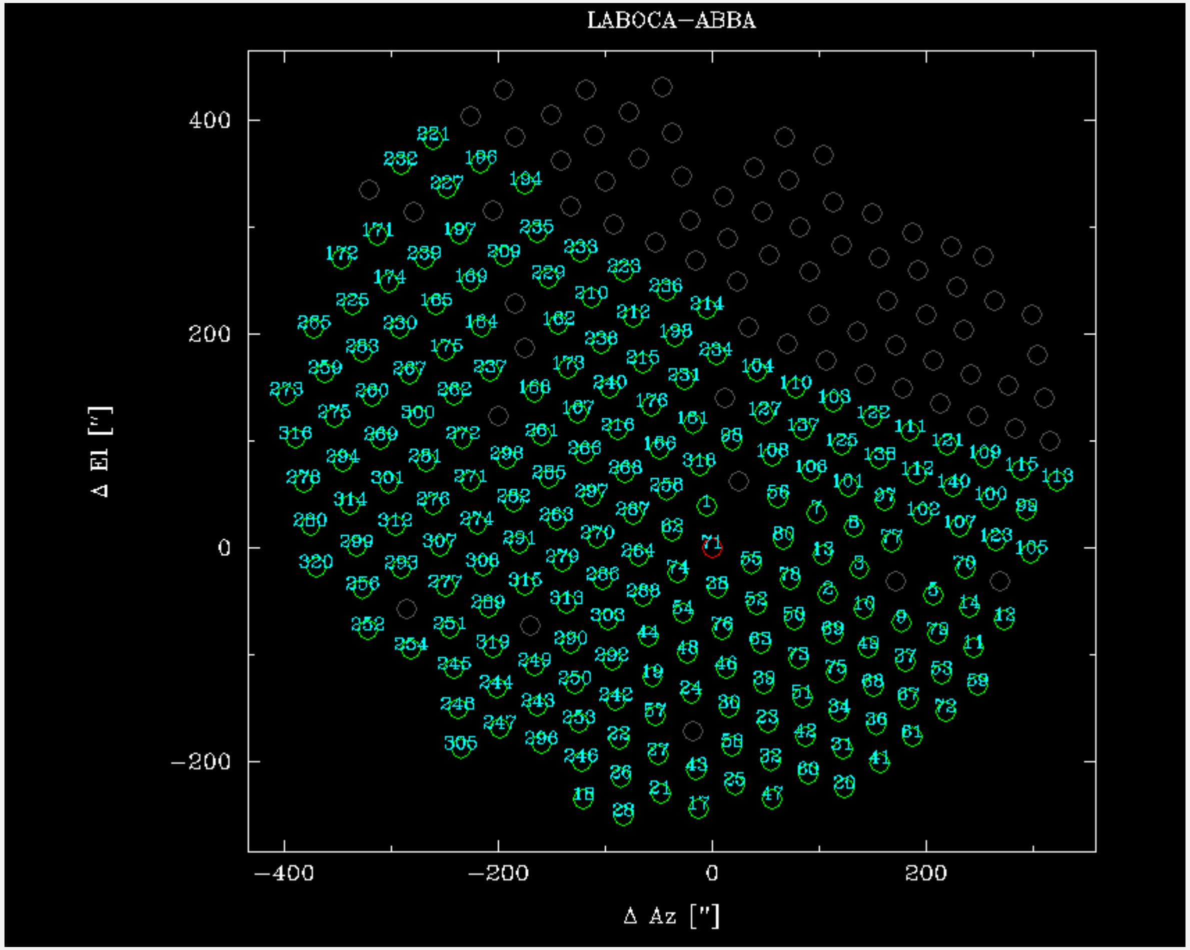
```
> azelo()
> azel()
```

- Plot bolometer array

```
> plotArray(num=1)
```

Some more useful functions

- Plot bolometer array: `> plotArray(num=1)`



Most useful feature

Each function is self-documented !

- What does flagSpeed do?

```
> print flagSpeed.__doc__

>     DES: Flag data according to telescope speed
>     INP: (float)    below   = flag data below this value
>          (float)    above   = flag data above this value
>          (int)      flag    = flag to be set (default 3
                     'ELEVATION VELOCITY THRESHOLD')
```

Most useful feature

Each function is self-documented !

- What does writeFits do?

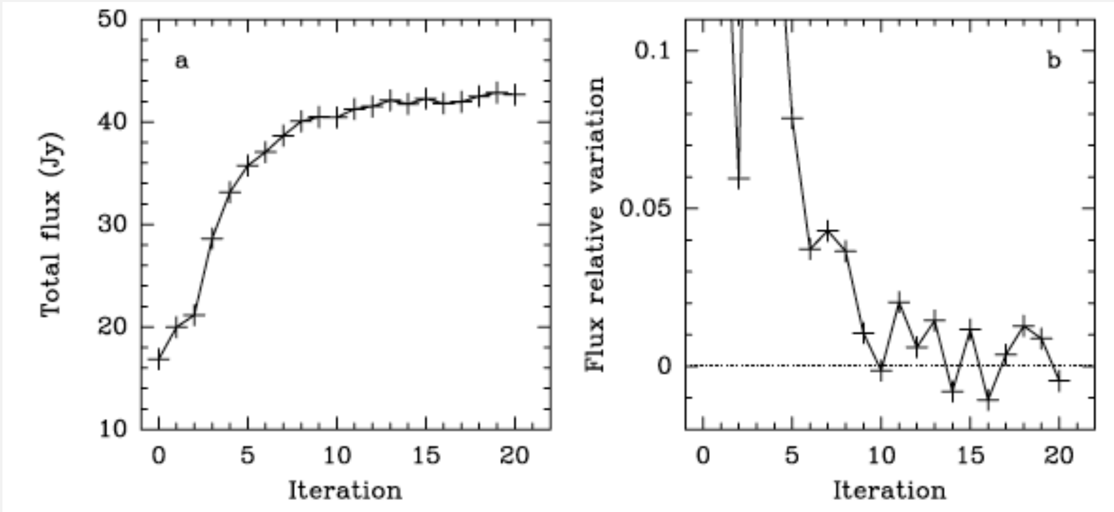
```
>   print data.Map.writeFits.__doc__

>          DES: store the current map (2D array with WCS info) to a FITS file
>          INP: (str)   outfile: output file name (default boaMap.fits)
>               (bool) overwrite: overwrite existing file -
>                                 default = 0: do not overwrite existing file
>               (f list) limitsX/Y: optional map limits (in world coordinates)
>               (string) intensityUnit: optional unit of the intensity(default:"Jy/beam")
>               (bool) writeFlux, writeWeight, writeCoverage: should these planes be
>                                                included in the output file? (def. yes)
>               (bool) writeRms: should another file with rms plane be written? (def. no)
```

Conclusions

- Tricky step = flux calibration.
  What to trust? skydips? radiometer → tau?
  primary calibrators? secondary calibrators?
  Typical uncertainty 10% (LABOCA) to 20% (SABOCA)
- Parameters specific to kind of source:
  - nb loops for various correlated noise removal
  - despike : caution with bright sources!
  - FFT filtering / baseline : caution with extended emission!

Conclusions

- Importance of iterative processing, especially with bright or extended sources – e.g. Belloche et al. 2011, A&A 535



- BoA can reduce data from LABOCA, SABOCA, ArTéMiS, but also SHFI, CHAMP+…

Thank you!