# Reduction of LABOCA data with BoA

- Observing with LABOCA at APEX
- Basic data reduction
- More evolved steps

# Focus, pointing, calibration

- Checking (and correcting) Focus settings:

| | |
|---|---|
| fx, fy, fz | focus in X, Y and Z |

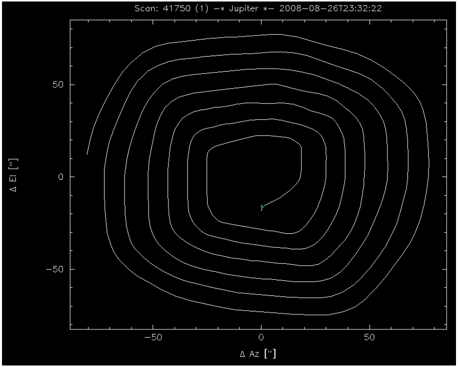
- Correcting pointing offsets + calibration scans:

| | |
|---|---|
| pspiral | one 20 s spiral, pointing on planets |
| pspirall | one 35 s spiral, pointing on $> 10$ Jy sources |
| spiralras | 4×20 s spirals, pointing on few Jy sources<br>calibration on secondaries |
| spirallras | 4×35 s spirals, pointing on $\sim$1 Jy sources |

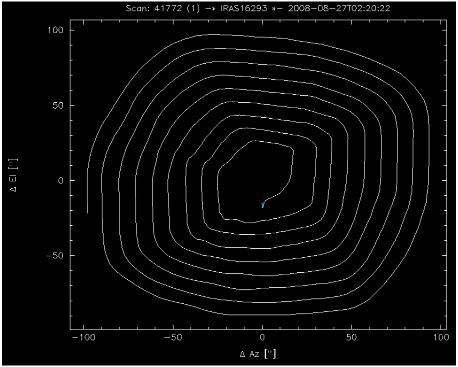
- Measuring the sky opacity:

| | |
|---|---|
| bolotip(az) | hot–sky + skydip |

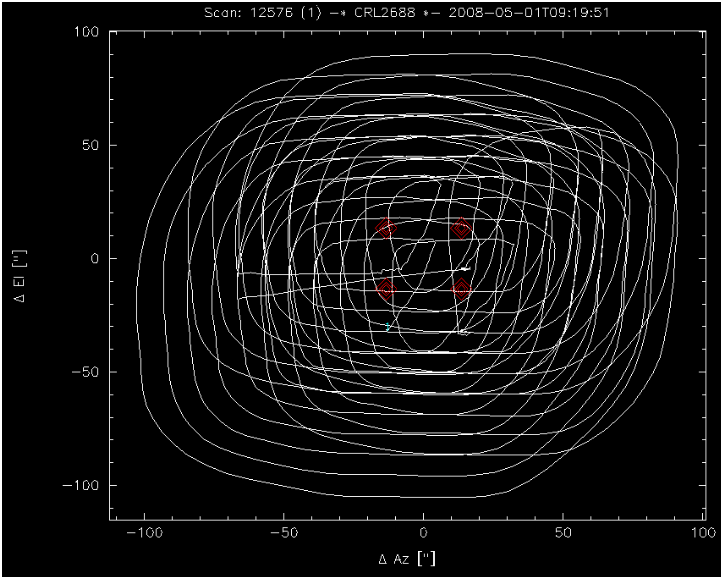
## pspiral()



## pspirall()



# spiralras()


Scan: 12576 (1) —* CRL2688 *— 2008-05-01T09:19:51
100
50
0
−50
−100
Δ El ["]
−100
−50
0
50
100
Δ Az ["]

# Typical observing session

- Instrument setup, pointing and focus corrections
- Calibration: observing primary calibrators (= planets) if any, secondary calibrators; skydip
- Observing *your* project:
  - Observing calibrator(s)
  - skydip
  - Pointing correction near science target
  - Observing *your* target(s)
  - Pointing correction near science target
  - Observing *your* target(s)
  - Observing calibrator(s), skydip
  - Pointing correction near science target
  - Observing *your* target(s)
  - (...)

# BoA Environment variables

A shortcut to start BoA, as well as required environment variables, are defined in a start-up script: **.boarc.(c)sh**

```
/homes/schuller> source /soft/astro/boa/.boarc.csh
/homes/schuller> which boa
boa:     aliased to python -i /soft/astro/boa/boa/boa/BoaStart.py
/homes/schuller> boa
BoA - the Bolometer Array data Analysis Software
Revision: 2447
Date: 2007-12-04 13:20:44 +0100 (Di, 04 Dez 2007)
boa< I: BoA - Offline mode
boa>
```

## Using a more recent version:

```
/homes/schuller> source /homes/schuller/.boarc.csh
/homes/schuller> which boa
boa:     aliased to python -i /homes/schuller/boa/SVN/boa/BoaStart.py
/homes/schuller> boa
BoA - the Bolometer Array data Analysis Software
Revision: 2610
Date: 2008-09-18 17:37:12 +0200 (Thu, 18 Sep 2008)
boa< I: BoA - Offline mode
boa>
```

## Defining input directory and project ID:

```
boa> indir('/data/schuller/Atlas/Chile/')
boa< I: input directory : /data/schuller/Atlas/Chile/
boa> proj('C-082.F-0010B-2008')
```

## What files do we have there?...

```
boa> ils()
boa< W: list empty, populating...
 53366-2008-10-12        GAL-p48.0       MAP   1      45   ['LABOCA-ABBA'] [318] 54751.959 (217.7 MB)
 53374-2008-10-13        GAL-p49.0       MAP   1      45   ['LABOCA-ABBA'] [318] 54752.014 (217.5 MB)
 53375-2008-10-13            G45.1     POINT   1      35   ['LABOCA-ABBA'] [318] 54752.038 (  2.6 MB)
 53376-2008-10-13        GAL-p49.5       MAP   1      45   ['LABOCA-ABBA'] [318] 54752.039 (216.9 MB)
```

# Calibration scheme

**Calibrating = correcting for opacity**

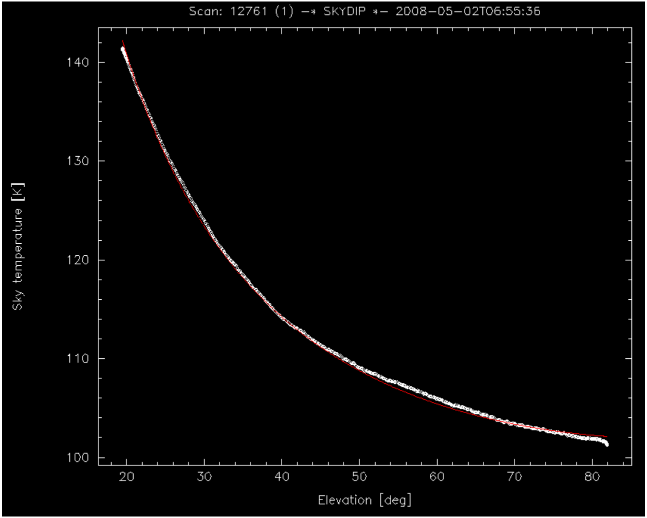
- Compute opacity from skydip measurements
- Check on calibrators (= sources with known fluxes) that the opacity is correct
- if not, apply an extra correction factor

**Primary calibrators = the planets**

- Primary calibrators = Mars, Uranus and Neptune
  → use astro to get expected flux at given time and date

```
Astro> let beam 19.2
Astro> let frequency 345.
Astro> time 08:27:13 02-MAY-2008
Astro> planet neptune
 NEPTUNE      RA 21:46:02.8989 Dec -13:47:18.598 Az     29.37775 El     26.69751 Sun.D.   78.2 Vel.
 DE=30.23 DS=30.04 Maj=  2.22 Min=  2.16 PA=337.68 TB= 81.04 S( 345.0)=   23.58
 Frequency  345.0: Beam   19.2  Tmb     0.65  Flux     23.47  Size   19.3
```

# Skydip reduction



# Skydip reduction

- Single scan:

```
boa> lab = os.getenv('BOA_HOME_LABOCA')
boa> scannr = 12760  # scan number of hot-sky
boa> execfile(lab+'/reduce-skydip.boa')
(...)
boa< I: ---- calibration hot-sky ----
boa< I: Tsky = 101.26 K
boa< I: ----   skydip  results   ----
boa< I: Signal corrected for blind bolos
boa< I: Coupling = 74.83 %
boa< I: tau_z    = 0.139
boa< I: Tatm     = 262.28 K
```

  The result: zenithal tau = 0.139

  The problem: usually underestimated... →**1.3** × **0.139**

- To process many scans, update list of scans in:
  `$BOA_HOME_LABOCA/reduce-skydip-loop.boa`
  (and output file name), and exec that file. Then use:
  `getTau(refmjd,method,tauFile)`

# Secondary calibrators

- Single scan:

```
boa> read(12763)
boa> opac(1.3 * 0.139)
boa> execfile(lab+’/reduce-calib-map.boa’)
(...)
boa< I: Peak flux    = 31.3443131588
boa< I: Delta Az ["] = -1.73020835982
boa< I: Delta El ["] = -4.62551226609
boa< I: FWHM_1   ["] = 23.2210223465
boa< I: FWHM_2   ["] = 26.2340775143
boa< I: Tilt   [deg] = -395.287459169
boa< I: map r.m.s. = 0.513640
----------------------------------------------------------------
FLUX G10.62:  31.344313 [expected: 32.200001, 97.342585 percent]
----------------------------------------------------------------
```

- To process many scans, update list of scans in: `$BOA_HOME_LABOCA/reduce-calib-loop.boa` (and output file name), and exec that file. Then use: `getCalCorr(refmjd,method,calFile)`

# reduce-map-weaksource.boa

```
$BOA_HOME_LABOCA/reduce-map-weaksource.boa
```

```
# Read LABOCA specific definitions
import os
if not os.getenv(’BOA_HOME_LABOCA’):
   raise ’Environment variable BOA_HOME_LABOCA undefined’
execfile(os.path.join(os.getenv(’BOA_HOME_LABOCA’), ’cabling.py’))

CntstoV(data)
updateRCP(’master-laboca-may07.rcp’)
data.zeroStart()
flat()

# Flag bad channels
flagC(resistor)
flagC(cross)
flagC(sealed_may07)
```

# reduce-map-weaksource.boa

```
# APPLY HE3 DRIFT CORRECTION IF SELECTED

try:
        tmp = apply_he3corr
        if tmp:
                print "***** Warning: He3 correction should not be used for SV
                print "***** Instead, the correctBlind(data) function can be us
except NameError:
        apply_he3corr = 0

if apply_he3corr:
        correctHe3(data)

# Convert to Jy/beam
data.Data *= array((VtoJy),’f’)
```

# reduce-map-weaksource.boa

```
# Flag stationary points and high acceleration
data.flagSpeed (below=30.)
data.flagSpeed(above=500.)
data.flagAccel(above=800.)

# Flag dead and very noisy channels
data.flagFractionRms(ratio=5)

# First correlated noise removal on all channels and despiking
medianNoiseRemoval(chanRef=-1,factor=0.8,nbloop=5)
despike(below=-3,above=3)

# correlated noise removal by groups and boxes of channels
# (Laboca wireing)
correlbox(data,factor=0.8,nbloop=2)
correlgroup(data,factor=0.8,nbloop=2)
```

# reduce-map-weaksource.boa

```
# Flag noisy channels
data.flagFractionRms(ratio=5)

# Despiking
despike(below=-3,above=3)

# Filter on low frequencies
data.flattenFreq(below=0.3,hiref=0.35)
base(order=1,subscan=0)
despike(below=-3,above=3)

# Compute weights based on rms of each channel
weight()
```

- Now you can compute a map:

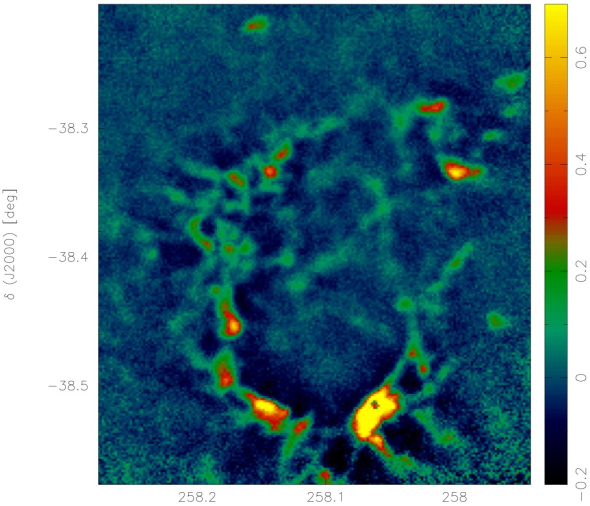
```
mapping(system='EQ',sizeX=[305.5,304],sizeY=[-12.5,-11.5],limitsZ=[-1,2.5])
```

# Loop over the scans

```
proj('M-079.F-5000-2007')
indir('...')  # path to the data

scans = [13358,13359,13564,13565,13593,13594]
taus  = [0.22,0.22,0.25,0.23,0.2,0.2]

ra1,ra2 = 262.58,258.84
dec1,dec2 = -38.5,-34.1
mapList = []
for snum in range(len(scans)):
    read(scans[snum])
    opac(taus[snum])
    execfile(lab+'/reduce-map-strongsource.boa')
    mapping(system='EQ',sizeX=[ra1,ra2],sizeY=[dec1,dec2],
               oversamp=2,limitsZ=[-2,10])
    mapList.append(data.Map)

ms = mapsum(mapList)  # or mapsum2, or mapsumfast
ms.display(limitsZ=[-1,5])
ms.dumpMap('atlas_map.data')
ms.writeFits('atlas-SV.fits')
```

# Combined map, 1st step

**$1^{st}$ iteration**

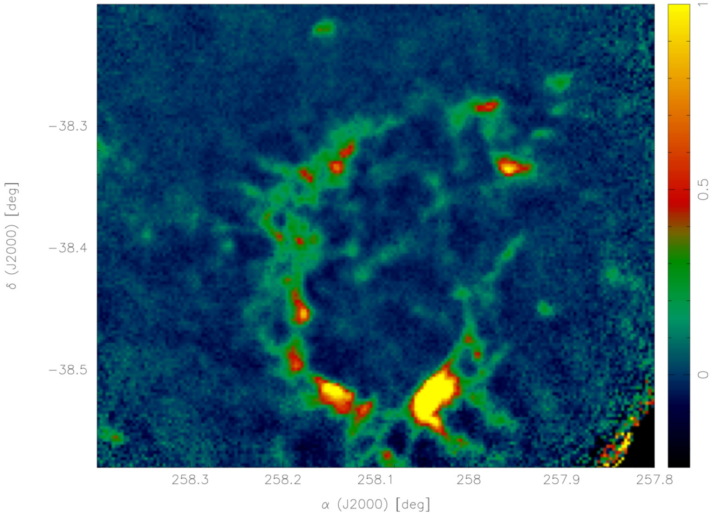


# Iteration with source model

Use map (or better, S/N map) from previous step to flag data.

```
snr = copy.deepcopy(ms)
snr.Data = ms.Data * sqrt(ms.Weight)
mapList = []

for snum in range(len(scans)):
    read(str(scans[snum]))
    opac(taus[snum])
    data.flagSource(threshold=5,model=snr)
    execfile(’reduce-map-strongsource.boa’)
    data.unflag(flag=8)
    mapping(system=’EQ’,sizeX=[ra1,ra2],sizeY=[dec1,dec2],
              oversamp=2,limitsZ=[-2,10])
    mapList.append(data.Map)
ms2 = mapsum(mapList)
ms2.display(limitsZ=[-1,5])
ms2.dumpMap(’atlas_map-2.data’)
ms2.writeFits(’atlas-SV-2.fits’)
```

# $2^{nd}$ iteration



# Summary

- The difficult step = **calibration**, i.e. getting good estimates for opacity, and optionally extra correction factors:

```
data.Data /= array(calcor,'f')
```

- Co-adding scans: use mapsum(fast), all maps must be computed with **same coordinate limits and same pixel size**
- All functions include a brief documentation:

```
boa> print data.flagFractionRms.__doc__
```

```
    DES: flag according to rms, with limits depending on median rms
    INP: (i list) chanList : list of channel to flag (default: current list)
         (f)         ratio : channels with rms below median/ratio and above
                             median*ratio will be flagged
         (i)          flag : value of flag to set (default: 2 'BAD SENSITIVITY')
         (b)          plot : plot the results
```

- Some jumps in your data? (= spikes seen by all bolo)

```
boa> execfile(lab+'/boaLaboca.py')
boa> jumps()
```

```
found 2 jumps
boa< I: 72 timestamps flagged ( 1.86 %) with flag 7
boa< I: 21 timestamps flagged ( 0.54 %) with flag 7
```

**More questions?... Contact Fred or Axel!**