GBT Spectral-Line Data Reduction and Mapping Reduction

Joy Skipper and Amber Bonsall

GREEN BANK OBSERVATORY

GBO FASTx Remote Access

http://greenbankobservatory.org/gbt-observers/observing/remote-unix-access/

The GBT has a lot of different observation types and many different methods for reduction

- Standard gbtidl routines and the GBT pipeline supports VEGAS spectral-line data from position-switched or frequency-switched observations for receivers with noise diodes
- Pulsar data ➔ Presto reduction package
- Mustang ➔ separate Mustang package (contact Brian Mason)
- VLBI ➔ VLBA folks
- Radar ➔ Radar folks
- W-band, Argus, SubBeamNod ➔ Dave Frayer
- Data reduction not supported for continuum observations or stokes polarization observations; however, non-official software exists

Reduction Software Documentation

GBTIDL:

-- User's Guide: http://www.gb.nrao.edu/GBT/DA/gbtidl/users_guide/

- Quick Reference: http://www.gb.nrao.edu/GBT/DA/gbtidl/QRG_release.pdf

- Calibration documentation:

http://www.gb.nrao.edu/GBT/DA/gbtidl/gbtidl_calibration.pdf

- Reference Manuals:

User: http://www.gb.nrao.edu/GBT/DA/gbtidl/release/user/

Contrib: http://www.gb.nrao.edu/GBT/DA/gbtidl/release/contrib

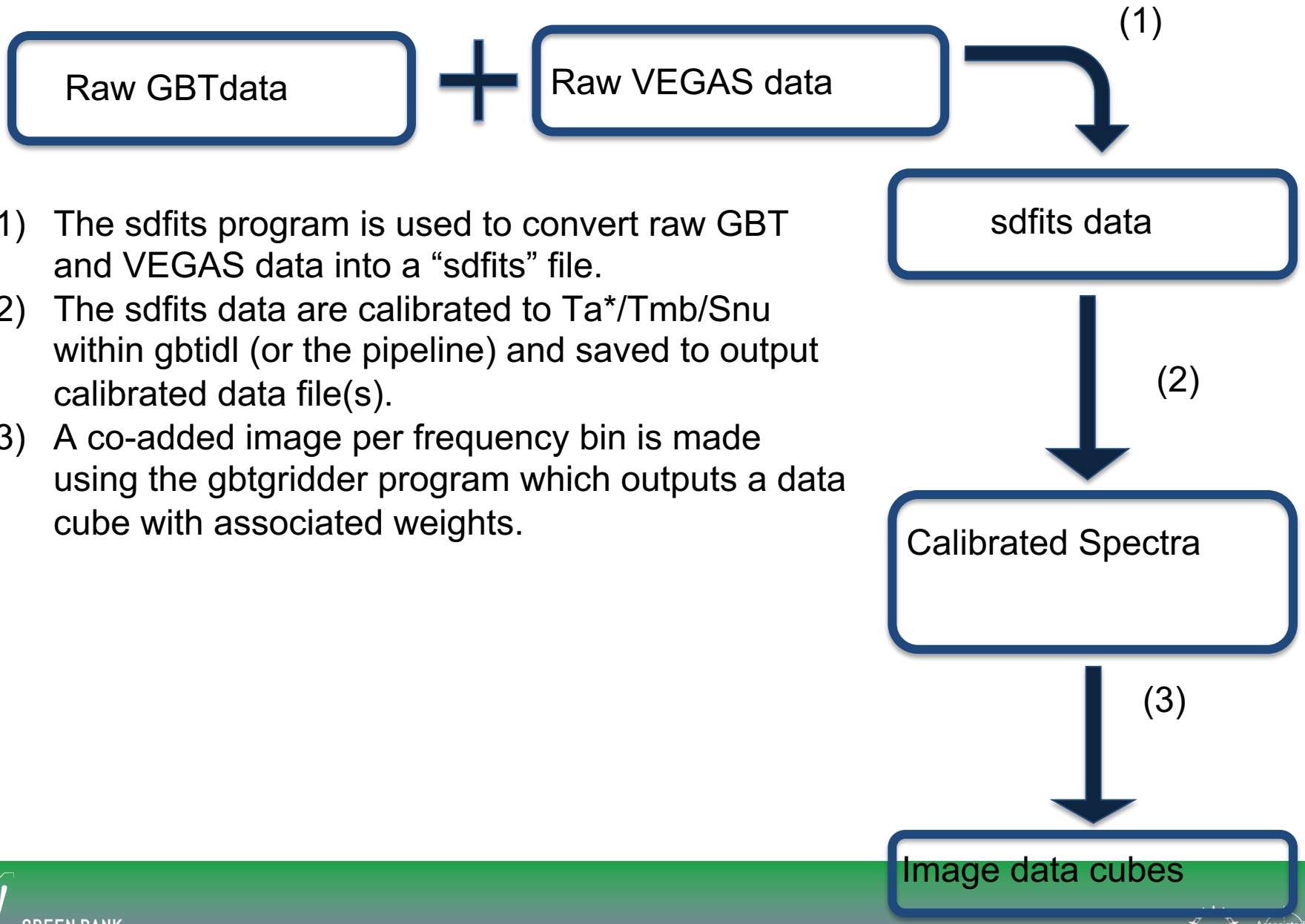
sdfits:

https://safe.nrao.edu/wiki/bin/view/Main/SdfitsDetails

gbtpipeline: https://safe.nrao.edu/wiki/bin/view/GB/Gbtpipeline/PipelineRelease

gbtgridder: (gbtgridder –h)

GBT Data Processing "Flow" Chart



(1) The sdfits program is used to convert raw GBT and VEGAS data into a "sdfits" file.

(2) The sdfits data are calibrated to Ta*/Tmb/Snu within gbtidl (or the pipeline) and saved to output calibrated data file(s).

(3) A co-added image per frequency bin is made using the gbtgridder program which outputs a data cube with associated weights.

The telescope measures:
Ta = “antenna temperature”

- Ta(total) = Tsource + {Trx + Tbg + Tatm + Tspill}
- Where {….} = other contributions
- Want Tsource, so carry out ON – OFF
- Ta(ON) =Tsource + {….}
- Ta(OFF) = {….}
- So Ta(ON)-Ta(OFF) = Tsource

➔ Need to carry out ON-OFF observations and there are different observing techniques for measuring ON-OFF

Different Observing Modes to derive the reference data (OFF)

Types of reference observations

- ➢ Frequency Switching
  - In or Out-of-band
- ➢ Position Switching
  - Reference-Off
  - Mapping-Off
- ➢ Dual-Beam Position Switching
  - **Nod** -- Move telescope
  - **SubBeamNod** -- Move Subreflector

GBT Definition of T_a

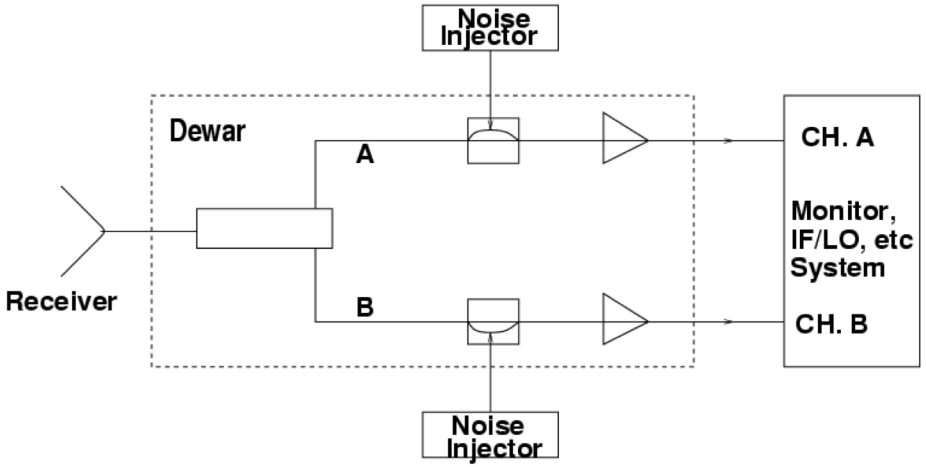
$$T_a = \frac{(ON - OFF)}{OFF} T_{system}$$

Blank Sky or other

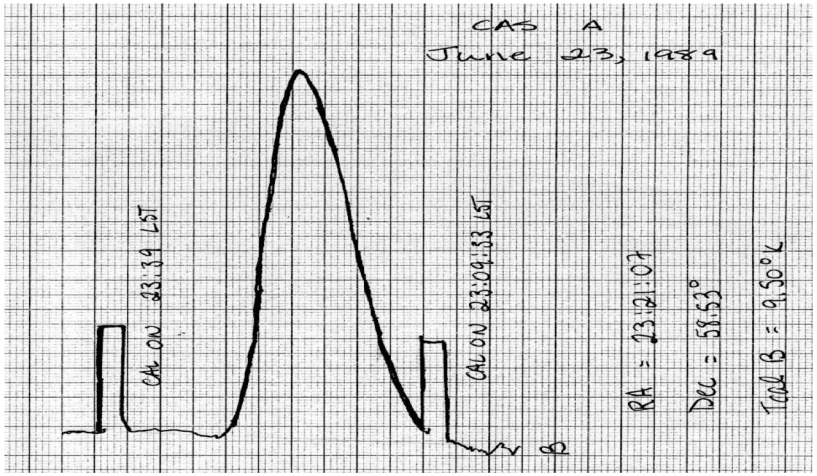
From diodes, Hot/Cold loads, etc.

Determining T_{sys}

Noise Diodes



All GBT receivers besides 4mm, Argus, and Mustang use noise diodes.



Determining T_{sys}

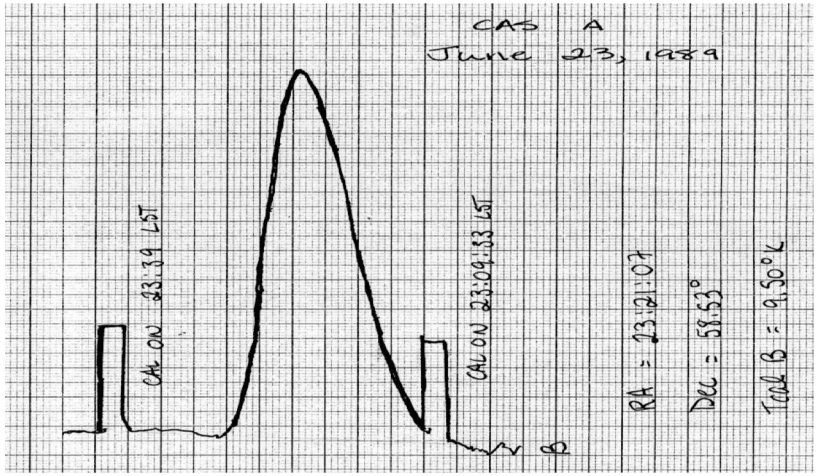
Noise Diodes

$$T_{sys} = T_{cal} * OFF/(ON - OFF)$$

GBT: Flicker diode on/off

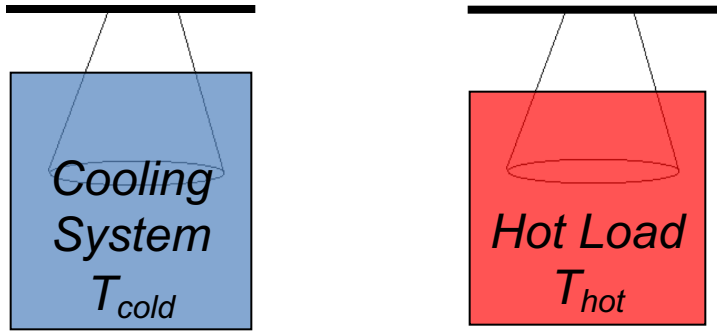
$$T_{sys} = T_{cal} * OFF/(ON - OFF) + T_{cal}/2$$

Typically choose low Tcal value to minimize Tsys and high Tcal value for very bright sources (for Rx that have two options)



Determining T_{sys}

Hot & Cold Loads



Gain: g =(Thot – Tcold)/(Vhot –Vcold) [K/Volts]

Tsys = g Voff

Example GBT 4mm Rx

Temperature Scales

- $T_a = T_{sys}\ (ON-OFF)/OFF$ (GBT typically uses uncorrected antenna temperature)
- $T_a' = T_a \exp(\tau_o A)$ (corrected for atmosphere)
- $T_{mb} = T_a'/\eta_{mb}$ ($\eta_{mb} \sim 1.3\ \eta_a$)
- $T_a^* = T_a'/\eta_l$ (Argus uses T_a^*, $\eta_l = \sim 0.99$ for the GBT)
- $T_a'/Sv = 2.84\ \eta_a$ (for the GBT)

Calibration:
Flux Density vs Antenna Temp vs Main-Beam Temp

$P_{rec} = ½ A_e S_v \Delta v = k T_a' \Delta v$

$A_e = \eta_a (\pi/4) D^2$

$S_v = 3520 T_a'/(\eta_a [D/m]^2)$

$\rightarrow T_a'/S_v = 2.84 \eta_a$ for the GBT (η_a=0.71 at low v)

- Know S_v (use ALMA calibration database for 3mm and VLA calibration papers for <50GHz) and derive η_a from measured Ta'
- Measure FWHM from good pointing scans or within your image to derived η_{mb} and Tmb; **$Tmb = Ta'/ \eta_{mb}$**
- **$\eta_{mb} = 0.8899 \eta_a (\theta_{FWHM} 100m/ \lambda)^2$** (assumes Gaussian beam, where beam FWHM is in radians)

GBO Data Directories

- Home area: /users/user_name
- Scratch data area: /home/scratch/user_name
- Raw gbtdata by project (e.g., AGBT16B_037_04): /home/gbtdata/AGBT16B_037_04
- Raw Vegas data by project: /lustre/gbtdata/AGBT16B_037_04/VEGAS
- sdfits data by project: /home/sdfits/AGBT16B_037/04

Public Data Processing Machines with lustre access:

- newton, planck, fourier (192GB ram)
- euclid, thales (16GB ram)
- arcturus (132GB ram) {pipeline machine}

- Extra disk space for data processing: /lustre/pipeline/scratch/user_name

gbtidl (=unipops [12m and 140ft reduction package] converted to IDL)

- Data access (connecting to sdfits file)
  - gbtidl> online
  - gbtidl> offline,'AGBT16B_037_04'
  - gbtidl> filein,'mysdfitsfile.fits'
  - gbtidl> summary

- User "pro" directory used by gbtidl: /users/user_name/gbtidlpro

Standard Reduction scripts

- Ta=Tsys(ON-OFF/OFF):
  - getps, scan (position switch)
  - getfs, scan (frequency switch)
  - getnod, scan (nod data)
  - getsigref, scan_on,scan_off
- Raw passband
  - gettp, scan (total power for scan)
  - **gettp,scan,ifnum=ifnum,plnum=plnum,fdnum=fdnum, sig_state=sig_state,subref=subref,wcalpos=wcalpos**
  - getrec, rec (get an individual record, see list)

Some Basic GBTIDL Commands:
DEMO_basics

- gbtidl> offline,'TGBT17A_506_11'
- gbtidl> summary (give summary of session)
- gbtidl> getsigref,6,7 (on,off position switch data reduction)
- gbtidl> header (provide some of the meta data in container 0)
- gbtidl> gettp,6 & copy,0,2 get ON and copy to container 2
- gbtidl> gettp,7 & copy,0,3 get OFF and copy to container 3
- gbtidl> subtract,2,3 (ON-OFF) (container math)
- gbtidl> divide,0,3 (ON-OFF)/OFF
- gbitidl> scale,17.34 (multiply by Tsys to give Ta=Tsys*(ON-OFF)/OFF)
- gbtidl> !g.s[0].units='Ta' (changing meta data units from counts to Ta [K])
- gbtidl> setregion (select regions for baseline removal)
- gbtidl> nfit,3 (use 3rd order polynominal for baseline fitting)
- gbtidl> bshape (fit baseline)
- gbtidl> baseline (remove baseline)
- gbtidl> gsmooth,3,/decimate (Gaussian smooth with width of 3 channels)
- gbtidl> fitgauss (fit Gaussian to data)
- gbtidl> keep (save current data in container 0 to output file !g.s[0].units='Ta')

Frequency Switching Example: **DEMO_frequency_switching**

- GBTIDL -> offline,'TGBT17A_506_11'
- GBTIDL -> summary (give summary of session)
- GBTIDL -> getfs,10 (reduce frequency switched data)
- GBTIDL -> gettp,10,sig=1 (look at one freq)
- GBTIDL -> freeze
- GBTIDL -> gettp,10,sig=0 (look at other freq)
- GBTIDL -> oshow
- Change between MHz and Channels at top of IDL plotting window to show how this works

GBTIDL position-switch Demo: DEMO_sdfits_HIsigref

- filein,'sdfits_example' (from DATAdemo directory)
- summary
- gettp,56 (look at passband)
- getsigref,56,57 (look at one ON-OFF pair)
- dotri,56,70 (reduces ON-OFF-ON sequence and co-adds data for scans 56-70)
- Smooth data and baseline removal

Example script: “dotri.pro”

```
pro dotri,sc1,sc2
;;16A054 HI project
;;does ON-OFF-ON reduction for scans sc1-->sc2
;;Session 1: sc1=16, sc2=39
sclear
for ii=sc1,sc2,3 do begin
   print,'combining ON-OFF',ii,ii+1
   getsigref,ii,ii+1,plnum=0
   accum
   getsigref,ii,ii+1,plnum=1
   accum
   print,'combining ON-OFF',ii+2,ii+1
   getsigref,ii+2,ii+1,plnum=0
   accum
   getsigref,ii+2,ii+1,plnum=1
   accum
endfor
ave
copy,0,10
return
end
```

Argus Mapping Demo: DEMO_argus_mapping

- Raw data (40min observations, 167k spectra) first calibrated and reduced at about 0.2 sec per spectrum which takes ~9hrs) using argus_mapcal.pro ➔ save15a901_27.fits
- filein,'save15a901_27.fits' (from DATAdemo directory)
- getrec,50000
- for i=70000,70200 do begin & getrec,i & accum & end
- ave (13CO near channel 6000)
- Run gbtgridder
- $gbtgridder –c 5900:6100 –a 7 –noline –nocont –o test3 save15a901_27.fits
  - Channels 5900:6100
  - Averaging over 7 channels
  - Avoid having software do continuum subtraction (already done)
- $casaviewer (or ds9) to view output cube (test3_cube.fits)

L-band Pipeline Demo: DEMO_NGC6946_HI_pipeline

- Gbtidl> offline,'TGBT17A_506_11'
- Gbtidl> summary (map scans 14-26 and reference scan is 27 want on HI which is IFNUM=0)
- $gbtpipeline –i /home/sdfits/TGBT17A_506_11/TGBT17A_506_11.raw.vegas –m 14:26 –refscan 27 –w 0 –c 1700:2300 -a 50
  - Firsts calibrates the data then grids the data
  - Map-scans are 14:26
  - Reference scan is 27
  - IFNUM= 0 (spectral window = 0)
  - Channels 1700:2300
  - Average 50 channels
- $casaviewer (or ds9) to view output cube data

Argus deep frequency switching coadd:
DEMO_argus_frequency_switching_deep_coadd

- GBTIDL -> offline,'TGBT17A_506_06'
- GBTIDL -> vanecal,53 (derive atmospheric parameters and Tsys* for all beams)
- GBTIDL -> argus_fsw,55,53,ifnum=0,fdnum=9 (calibrate one scan for center beam)
- GBTIDL -> for i=55,86 do begin & argus_fsw,i,53,ifnum=0,fdnum=9 & accum &end
- GBTIDL -> ave (Coadd all scans for one beam)
- GBTIDL -> argus_fsw_coadd,55,86,53,ifnum=0 (coadd all scans for all beams [takes 20min])

Saving data and Mapping

- fileout,'mysave.fits'
- Reduce data like you want then type: keep
- After calibration within gbtidl can run gbtgridder (eg.):

gbtgridder –c 11000:11251 –a 7 --noline –nocont –o myout mysave.fits

➔myout_cube.fits myout_weight.fits

Running the Pipeline

- Works for receivers with noise diodes (designed orignally for KFPA):
- %gbtpipeline –i my.sdfits.raw.vegas –m 14:24 –refscan 13,26

(where 14-24 are the map scans and 13 and 26 are the reference scans)

➔will calibrate and do the gridding

Example of reducing W-band DCR Data (daisy scan of point source)

➢ sdfits -backends=dcr TRCO_20160927 -scans=1

- IDL>ftab_ext,INdcr,[7,13,15,19,59,60],data,ra,dec,scan,plnum,fdnum,exten_no=1
- {calibrate data based on plnum and fdnum: data1=gain*data}
- IDL> hpfilt,data1,100,10,ndata
- IDL>mymap=griddata(ra,dec,ndata,dimension=[60,60])
- IDL>smmap=filter_image(mymap,FWHM_gaussian=3)
- IDL> atv,smmap

GREEN BANK OBSERVATORY

greenbankobservatory.org

The Green Bank Observatory is a facility of the National Science Foundation operated under cooperative agreement by Associated Universities, Inc.