Do-It-Yourself Data Analysis

Working with basic telescope data

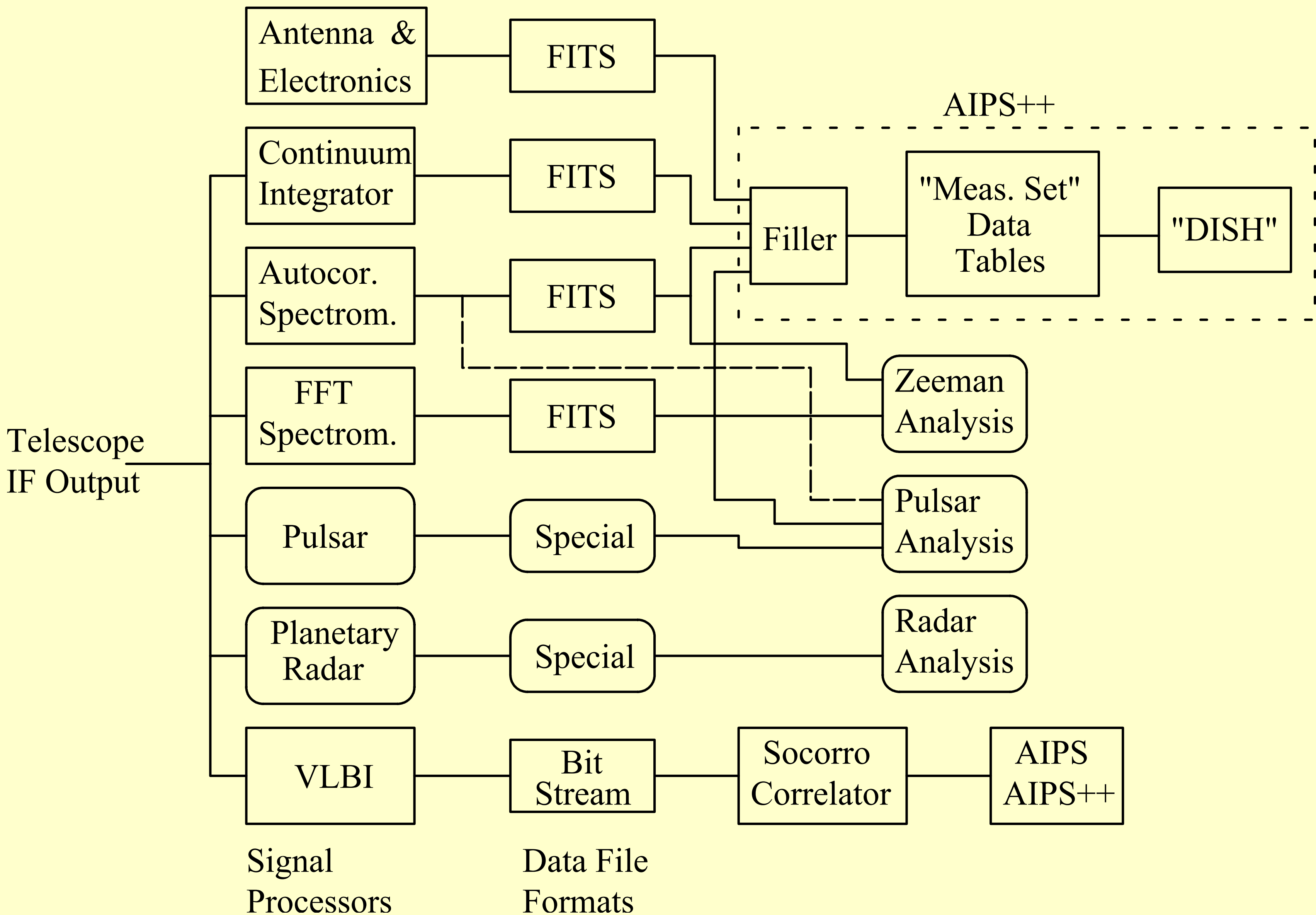
Data Analysis Environment Choices

- (Radio) Astronomy-specific packages
  - Class, Analyz, AIPS, AIPS++, Miriad, etc.
- General-purpose scientific and toolboxes
  - IDL, Matlab, Mathcad, Octave, etc.
- Scripting languages and toolboxes
  - Python, glish, Perl, Tcl, Java, etc.
  - Array arithmetic, linear algebra, scientific math, graphics, user interface, etc.
- Low-level language (C/C++) interfaces

Applications

- System tests and debugging
  - Algorithm comparisons
  - Isolation of software and hardware bugs
  - Special electronics tests
- New observing modes and algorithms
- New signal processors
- New software signal processing
- Post-analysis tasks, e.g. web publication

Data Flow on the GBT



Data Formats

- Flexible Image Transport System (FITS)
  - Self-documenting, text and binary data
- Binary data structure
  - Defined by data structures, like in ‘C’
- Sampled byte stream
  - Unformatted or simply formatted arrays

FITS File Structure

Main Header (Text)	HDU 0
Header (text) Table (Binary)	HDU 1
Header (text) Table (Binary)	HDU 2
Header (text) Table (Binary)	HDU 3

Etc.

Each HDU is generally different from the others.

For example,

HDU 0 contains major observing annotations

HDU 1 contains receiver setup parameters

HDU 2 contains switch state information

HDU 3 has spectrometer output data

Etc.

HDU = Header Data Unit

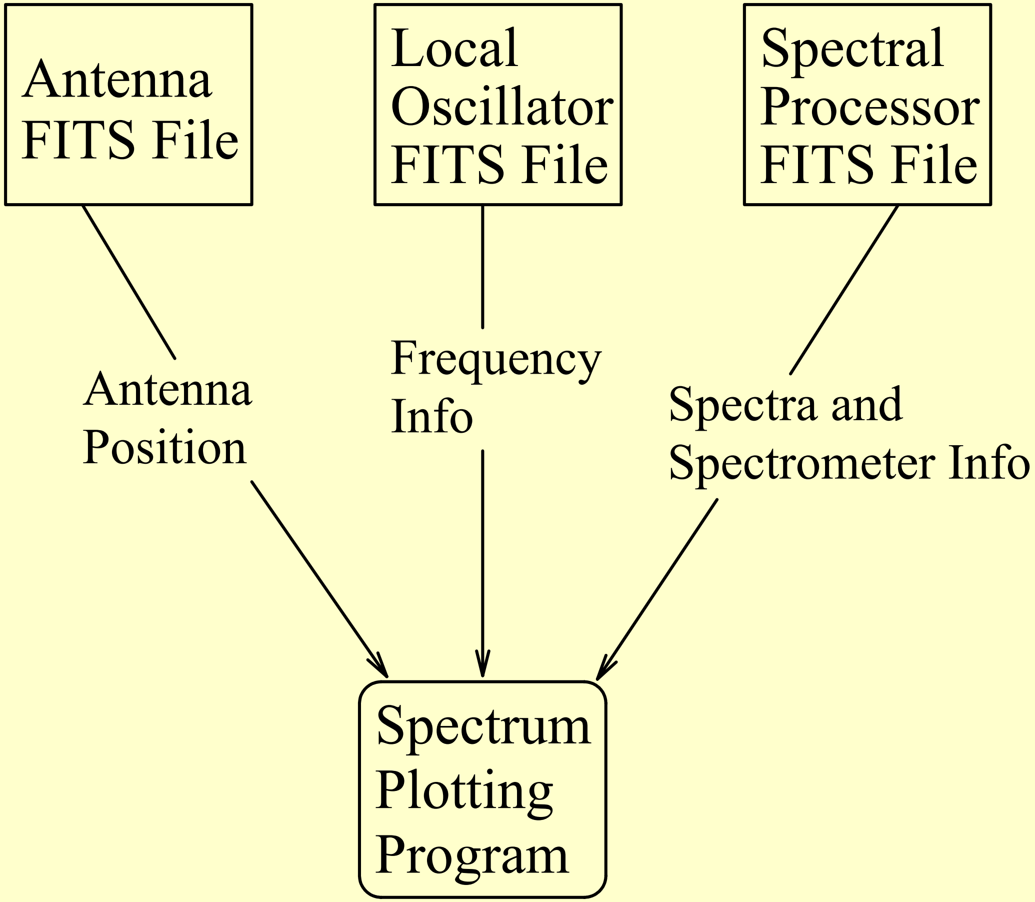
HDU Header Text Example

```
SIMPLE  =                    T / File conforms to FITS standards.
BITPIX  =                    8 /
NAXIS   =                    0 / No image data array present.
EXTEND  =                    T / Standard table extensions follow.
FORMATID= 'GBSDD007'           / SDD_FORMAT_ID
PROJECT = 'AGBT02A_069_01'     / Project Id
SCAN    =                  201 / Scan number
SCANID  = 'test    '           / Scan Id
BACKEND = 'SPAB    '           / Spectral processor
INSTRUME= 'SPAB    '           / Spectral processor
FITSVER = '0.1     '           / FITS definition version for this device
DATE-OBS= '2003-06-17T09:22:34.000' / Date of Observation
OBJECT  = 'CIG659  '           / Source name
SPMODE  = 'SSL StdSpectLine'   / SPECTRAL PROCESSOR MODE
UTDATE  =                52807 / MJD of start time
UTCSTART=   3.375399999994E+04 / start time seconds
UTCSTOP =   3.406100000000E+04 / stop time seconds
COMMENT    Green Bank Telescope Project
END
```

Binary Table Format Keywords in Data HDU Header

```
CTYPE1  = 'FREQUENCY'           / First data axis is Frequency
CTYPE2  = 'STATE   '            / Second data axis is State
CTYPE3  = 'RECEIVER'            / Third data axis is Receiver
TTYPE1  = 'SUBSCAN '            /
TFORM1  = '1J      '            /
TTYPE2  = 'UTDATE  '            /
TUNIT2  = 'MJD     '            /
TFORM2  = '1J      '            /
TTYPE3  = 'UTCSTART'            /
TUNIT3  = 'SECONDS '            /
TFORM3  = '1D      '            /
TTYPE4  = 'PSRPER  '            /
TUNIT4  = 'PULSAR PERIOD'       /
TFORM4  = '1D      '            /
TTYPE5  = 'DATA    '            /
TUNIT5  = 'COUNTS  '            /
TFORM5  = '4096E   '            /
TDIM5   = '(1024,2,2)   '       / data dimension of the field
EXTNAME = 'DATA                ' / extension name
END
```

Data Collation



GBT Telescope Data Directory Structure

/

home

gbtdata

<project ID>

Antenna LO1A Weather Spectrometer SpectralProcessor etc.

Antenna	SpectralProcessor
2003_06_07_09:22:34.fits	2003_06_07_09:22:34.fits
2003_06_07_09:27:54.fits	2003_06_07_09:27:54.fits
2003_06_07_09:33:13.fits	2003_06_07_09:33:13.fits

/home/gbtdata/AGBT02A_069_01/Antenna/2003_06_07_09:22:34.fits

FITS I/O Tools

- Python : CFITSIO library interface
  - http://ecf.hq.eso.org/~npirzkal/
  - http://www.gb.nrao.edu/~esessoms/igkya
- Glish : local glish client
  - ~rfisher/Applications/FitsCode/fits_client.g
- IDL : several fitsio packages
  - http://idlastro.gcfc.nasa.gov/fitsio.html

Example Code

- **`startup`** : Instructions for starting language interpreter in Green Bank
- **`list_scans`** : Lists scan numbers and source names from all FITS files under the specified project
- **`scan_info`** : Prints selected keywords and their values from several GBT device FITS files for a specified scan number
- **`get_fits`** : Retrieves the contents of a FITS file into a data structure of the scripting language or prints the FITS file contents to the screen
- **`plot_tsys`** : Plots system temperature spectra from a spectral processor scan that contains cal-on and cal-off spectra
- **`plot_actsys`** : Plots system temperature spectra from a autocorrelation spectrometer scan that contains cal-on and cal-off spectra

`~rfisher/Documents/SummerSchool/DIYanalysis/[Glish, IDL, or Python]`

GBT Standard Back-end Outputs

- Digital Continuum Receiver (DCR) : square-law detector integrated samples as a function of time
- Spectral Processor (FFT spectrometer) : integrated power spectra from different receiver states, e.g. cal-on, cal-off
- GBT Spectrometer (autocorrelation spectrometer) : integrated autocorrelation functions from different receiver states, e.g. cal-on, cal-off
  - Requires quantization corrections and Fourier transform to produce power spectra (tools available)

Binary Data Formats

Character : 8 bits (1 byte); ASCII coded text

char

Integer : 8, 16, 32 bits; (1, 2, 4 bytes) signed or unsigned

char, short, [int], long

Floating point : 32, 64 bits; (4, 8 bytes) IEEE-754 Standard

float, double

Byte order

Intel : least significant byte first (little endian)

Sun, Motorola : most significant byte first (big endian)

Directly Sampled IF Output Voltages

- Sample rate at twice the analog bandwidth
- Requires a lot of data storage capacity
  - 1 TB ~ 14 hours of 10 MHz bandwidth, 8 bits
- Nearly every signal (re)processing option then available in general purpose computers
  - 2-GHz P4 ~ 1 MHz bandwidth FFT real time