sdd·ipt

Increasing the Accessibility of GBT Data

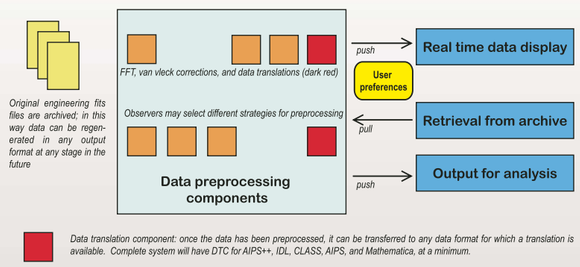
Part of the Single Dish Development Integrated Product Team (SDD IPT)

Karen O'Neil, Nicole Radziwill, Ronald J. Maddalena
National Radio Astronomy Observatory - Green Bank, West Virginia

The Robert C. Byrd Green Bank Telescope at dusk, located in Green Bank, West Virginia.

Introduction

The Green Bank Telescope (GBT) currently outputs its raw data as a suite of binary **FITS** files, approximately one per component device on the telescope, which are then consolidated and preprocessed before being written into an **AIPS++ Measurement Set** for more extensive analysis. This design decision by the **GBT** project had essentially restricted astronomers to a single data analysis package and reduced the productivity of those who prefer other analysis packages. To maximize the scientific returns from the unique features of the **GBT**, and to support a broader cross-section of observers' backgrounds and interests, work is being done to combine raw **GBT** data from the disparate **FITS** files into a variety of standardized **FITS** file formats such as **SDFITS** and **CLASS FITS**. These files can then be analyzed using tools such as **AIPS++, IDL, CLASS, Mathematica, and Matlab.**



Background

At present, a typical data set resulting from a **GBT** observation is composed of individual **FITS** files for each device required for an observation (e.g. the antenna, LO, backend) as well as a log (as a **FITS** file) which indexes all of the device files according to scans. Tools exist within the **AIPS++ DISH** utility which can transform the raw data into a representation that is sensible from the astronomical perspective (**AIPS++ Measurement Sets** or individual tables) .

Because the **GBT** was designed to produce its raw data as a collection of **FITS** files, it is a challenge to combine the information for analysis by any data reduction package. Filling the data into an **AIPS++ Measurement Set** took the a considerable effort by the development team to resolve issues associated with the data itself. Prior to the launch of the **GBT** data accessibility exploration project, anyone wishing to use any other data reduction program had to follow a similar process independently, writing their own modules to extract and preprocess relevant information from the collection of **GBT FITS** files.

One goal of the **GBT** data accessibility exploration project is to create a suite of common preprocessing routines for general use. For users of individual packages, all that would remain is to combine the components into a script that also contains a data translation component.

Goals & Objectives

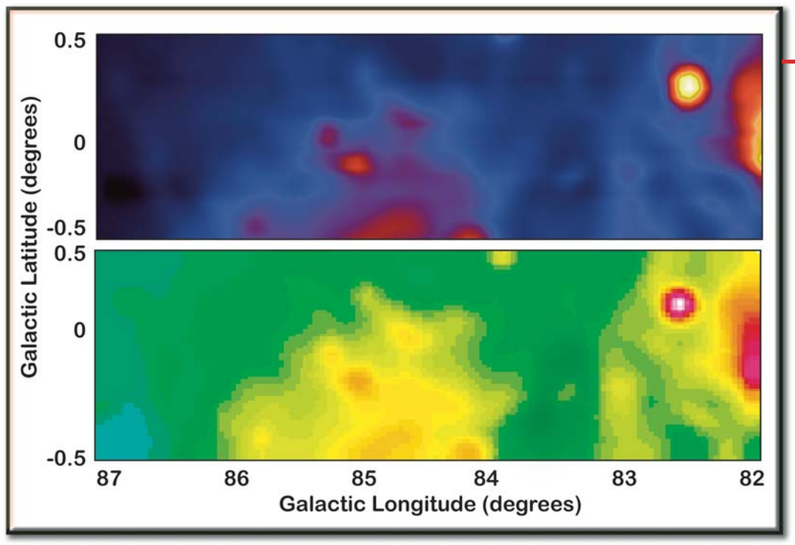
The primary goal for this effort is to make **GBT** data readily accessible to various data analysis packages. It is understood that each package has its own unique strengths and limitations, and not all packages may be able to reduce all types of **GBT** observations. However, with a clear understanding of what is possible with each package, an astronomer will have greater leverage in choosing the tool that best suits his or her needs for a particular investigation.

The intention is not to create a new, all-encompassing data format for the **GBT**, but to arrive at a reasonable representation that will make it straightforward to transition to future, standardized single dish data formats. (One possibility is the **MBFITS** specification that is under discussion by ALMA.)

Our objectives to facilitate the accomplishment of these goals are:

- Develop a ready means to access the data in the individual **fits** files
- Rewrite the data preprocessing steps in **Python**, for use by multiple programs
- Generate a unified representation for **GBT** data with the **Python** components

Each step in the process will be check against the well tested **AIPS++** routines. Additionally, once this process is complete, consistency of scientific results will by cross-checked between data analysis packages.



A continuum 21-cm map, completed as an assignment in the 2003 Single Dish Summer School held in Green Bank, was produced in both IDL (top) and AIPS++ (bottom) with similar results. Note that the color scheme for the two images is different.

Accessibility Strategy

Making **GBT** data accessible to additional data analysis packages in being done in a staged approach, aligned with demand from visiting observers and other development priorities of the **GBT**. **IDL** is being targeted immediately, because of the strong demand that has been expressed by visiting observers and local astronomers. Accessibility of **GBT** data into **Matlab** is also being addressed at the present time to support critical engineering experiments. In the next stage, access to **CLASS** and **Classic AIPS** will be investigated to support a wider audience of radio astronomers.

Standard data preprocessing components can be reused for many of these cases, making the entire system more maintainable while granting easier access to a larger variety of data analysis programs.

Prototyping Exercises

Three types of data were evaluated during the initial exercises: continuum data taken with the **Digital Continuum Receiver** and spectral line data from both the **GBT** spectrometer and spectral processor.

As it is a powerful language with the array handling needed for working with **GBT** data, **Python** was chosen as the programming language for all accessibility prototypes. It has a reasonably quick learning curve – skilled software engineers in Green Bank with no prior knowledge of **Python** were able to produce useful results within 2-3 days of beginning to work with the language. Additionally, several ALMA prototypes are being written in **Python**, indicating that **Python** could become a core competency among software engineers throughout NRAO.

Proof of concept exercises have been performed using **IDL**; **Matlab** experiments are in progress. These experiments take advantage of the **FITS Query Language** (developed on site to allow for ready access to the data in the individual fits files) to create an intermediary data format based on **SDFITS**. The next phase of prototype work to be completed by the end of the year will explore data accessibility by other analysis packages.

Current Status & Future Plans

Despite the demonstrated ability to import and plot data in **IDL**, and access raw data in **Matlab**, there is still much work remaining. Errors in content and form in the unified output data format are currently being resolved, and the prototype programs are being presented to a larger user community for feedback. After this time the feasibility of making the data sets a production offering can be accurately evaluated.

The **GBT** project does not intend to provide dedicated support to users of all the packages described herein; however, limited hands-on support for select packages such as **AIPS++** and **IDL** will be available. The intent is to provide sufficient documentation that all of the options, while making it possible for any observer to easily be able to use the data analysis package of their choice.

Data Analysis Package	Reuse Data	Data Accessibility Strategy
AIPS	✓	TBD; Potentially reusing Python preprocessing components and a unified fits file.
AIPS++	TBD	Currently using internal filler; Upgrade might include using python preprocessing components.
CLASS	✓	Under investigation; Plan is to create a unified FITS file and adjust it to suit CLASS.
IDL	✓	Create a unified GBT FITS file,; Use publicly availble programs from within IDL to import data.
MatLab	TBD	Under investigation; Native ODBC connectivity allows that the Fits Query Language may be used to extract data.
Mathematica	TBD	Under investigation; Approach will depend on results from Matlab investigation

Acknowledgements

Scientific validity of this activity has relied, and will continue to rely upon, the contributions of NRAO astronomers Bob Garwood and Jim Braatz, in consultation with Bill Cotton. Bob Garwood is also leading the work to qualitatively and quantitatively assess the accuracy and viability of reusable preprocessing components, and contributes extensive knowledge about the processing of GBT data and internals of gbtmsfiller, which he wrote. Technical development has been made possible thanks to the work of Green Bank Software Engineer Eric Sessoms, who conceived the idea and developed the FQL utility, and built all initial versions of data preprocessing components in Python. The technical efforts for producing a suitable evolutionary data format are now being led by David Fleming, a Software Engineer in Green Bank. Work to access GBT data in Matlab is being done by Software Engineers Ramon Creager and Paul Marganian. We also thank Kim Constantikes who is the lead user of Matlab as PTCS Project Engineer, as well as Carl Heiles and Tim Robishaw who have supplied us with tremendous insight about how they currently use IDL to analyze GBT data. Finally, we would like to thank the AIPS++ development team for their considerable work with GBT data formats and processing.