ADASS XIII
ASP Conference Series, Vol. XXX, 2004
F. Ochsenbein, M. Allen and D. Egret eds.

Increasing the Accessibility of Green Bank Telescope Data

Karen O'Neil, Nicole Radziwill, Ronald J. Maddalena

National Radio Astronomy Observatory, P.O. Box 2, Green Bank, WV 24944, U.S.A.

Abstract. 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 pre-processed 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. Here we describe prototyping exercises that were initiated during the summer of 2003 for the purpose of identifying how to make GBT data more readily accessible to a wider variety of data reduction tools. Although further refinement is needed to support the standard observing modes of the GBT in a production capacity, early results from the investigation demonstrate the feasibility and applicability of the approach.

1. Background

At present, a typical data set resulting from the Robert C. Byrd Green Bank Telescope (GBT) is composed of individual FITS files for each device required for an observation (e.g. the antenna, LO, backend) as well as a log (also a FITS file) which indexes all of the device files according to scans. GBT data can be assimilated into the AIPS++ DISH utility by using the AIPS++ d.import command, or by using the gbtmsfiller command, called from the UNIX command line. Either step transforms the raw data into a representation that is sensible from the astronomical perspective.

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. To fill data into an AIPS++ Measurement Set, the development team spent up to two years resolving issues associated with the data itself, and was eventually able to produce the gbtmsfiller routine which is in use today. Prior to the launch of the GBT data accessibility exploration, IDL users (for example) had to follow a similar process independently, writing their own modules to extract and pre-process relevant information from the collection of GBT FITS files. Users of other packages are still faced with this barrier.

The demand for greater accessibility has been expressed within NRAO as well as by visiting observers. Several astronomers at Green Bank have expressed a desire to do experiments in IDL, making use of IDL modules relevant to astronomers that have been developed by third parties. Engineers working on the Precision Telescope Control System (PTCS) project (a major initiative currently underway which will provide the pointing, collimation and surface accuracy required to allow the GBT to operate effectively at 3mm – see papers in this volume by Constantikes and Marganian) do much of their analysis in Matlab and need to access data from astronomical observations within the Matlab application. Requests have also been made to allow ready data reduction within the CLASS, Classic AIPS, and Mathematica packages.

2. Goals and Objectives

The primary goal for this effort is to make GBT more 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.

This is not exclusively a data format issue, although knitting together the disparate FITS files currently produced into one cohesive structure is one important step to enable many of the data paths. 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.)

Meeting several objectives will facilitate the accomplishment of these goals:

- Find an easier way to get data out of FITS files. This step has been accomplished through the development of FITS Query Language (FQL[1]);
- Extract the preprocessing steps from gbtmsfiller and rewrite them in Python, so they can be used by multiple programs;
- Validate the preprocessing components against previously verified parts of gbtmsfiller;
- Use the Python preprocessing components to generate a unified representation for GBT data;
- Reduce basic continuum and spectral line observations; plot them using various analysis packages and examine for correctness.

Once this process is complete, we will be able to verify the consistency of scientific results between data analysis packages (e.g. IDL vs. AIPS, AIPS++ vs CLASS, CLASS vs. IDL); until now we have not had two or more packages with which cross-comparisons can be performed. Being able to perform cross-comparisons will aid the process of commissioning data reduction for new

[1] `http://wiki.gb.nrao.edu/bin/view/Data/FitsQueryLanguage`

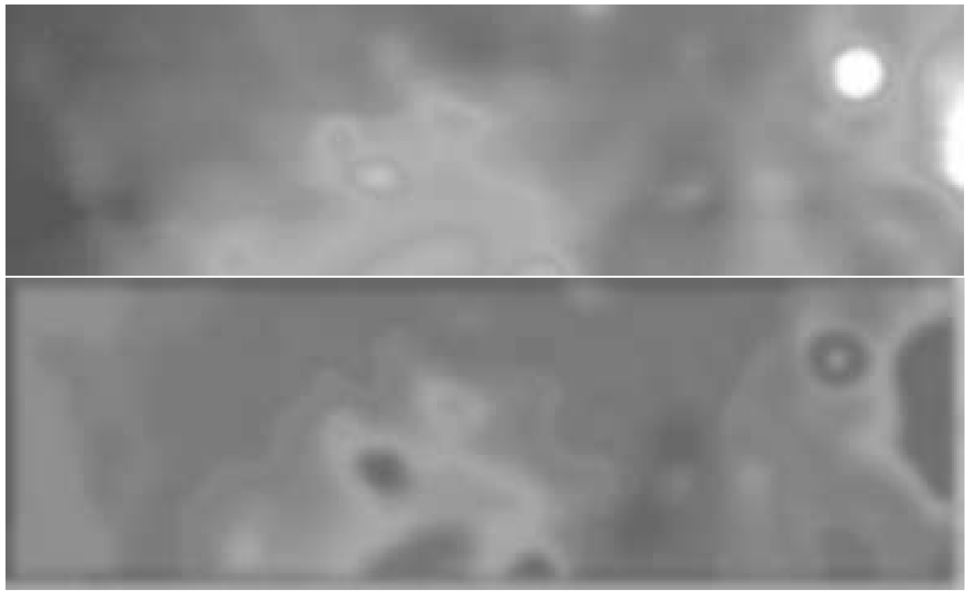

Figure 1. A continuum 21-cm map, completed as an assignment in the 2003 Single Dish Summer School held in Green Bank, was produced in both IDL (top) and AIPS++ (bottom) with similar results. Note that the color scale for the two images is different.

capabilities on the GBT, ensuring that errors are captured well in advance of live observations using a new device.

3. 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 and Matlab experiments are in progress (Figure 1). These experiments take advantage of the FITS Query Language 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.

4. Accessibility Strategy

Making GBT data accessible to additional data analysis packages is being done in a staged approach, aligned with demand from visiting observers and other development priorities of the GBT project. IDL is being targeted immediately, because of the strong demand that has been expressed by visiting observers and local astronomers alike. Accessibility of GBT data to Matlab is also being addressed at the present time to support critical PTCS experiments. In the next stage, access to CLASS will be investigated to support a wider audience of radio astronomers, and accessibility to AIPS will be explored, in part to support research for GBT development projects now in their earliest stages. Mathematica, which has the fewest identified users to date, will be explored once solutions are in place for other packages which are used more widely.

5. Current Status and Future Plans

On November 24th, 2003, the beta version of the SDFITS generator was released for wide internal review. Continuum data from the DCR, as well as spectral line data from both the spectrometer and the spectral processor, are fully supported. File sizes are somewhat smaller than the total size of the raw data files, and much smaller than equivalent MeasurementSets. The output in the SDFITS files has been validated against the AIPS++ filler and is at least as accurate, although performs much more slowly. Future plans include making the preprocessing components used to generate the SDFITS files fast enough to replace the AIPS++ filler, so that data to be reduced in most data reduction packages will be preprocessed by the same, uniformly validated components.

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 be able to easily use the data analysis package of their choice.

Up-to-date information on this project can be found online at
`http://wiki.gb.nrao.edu/bin/view/Data/WebHome`.

Acknowledgments. 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, also 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.