COMAP reduce docs]**COMAP Memo 0: COMAP Manchester Reduction Pipeline Documentation**

[

Stuart Harper

March 5, 2019

Contents

1 Installation

Requirements:

- PYTHON 3.0 or higher.
- Compiled shared libraries of the `FORTRAN` version of the Starlink astronomical libraries (SLALIB) available from http://starlink.eao.hawaii.edu/starlink/2018ADownload.
- A copy of parallel ready H5Py. Installation instructions can be found here http://docs.h5py.org/en/stable/mpi.html. N.B.: If you are using an Anaconda packaged version of PYTHON installation you may need to remove the existing CONDA install of HDF5.
- The latest version of HEALPY and MPI4PY (either openMPI or MPICH work as backends).

To install the Manchester COMAP reduction pipeline:

- Clone/download the `github` repository found here: https://github.com/SharperJBCA/COMAPreduce.
- Enter the directory: *cd COMAPreduce* and run *python setup.py install*. If your SLALIB libaries are not in standard location you must define the environment variable: `SLALIB_LIBS`
- To run the COMAP pipeline make a new directory above COMAPreduce (e.g. *cd ../ && mkdir runcomapreduce*) and copy the RUN.PY, and .INI files there.
- The pipeline can then be run using the command: *mpirun -n X python run.py -F FILELIST.list -P PARAMETERS.ini*. *FILELIST.list* should contain a list of files with either just the filenames to be processed or the full path to files to be processed. *PARAMETERS.ini* will control the processing to be performed, details of which are described in Sections 2 and 3.

2 Usage

2.1 Parameter Files

There are several example parameter files already included:

- AmbLoad.ini - This will calculate the $T_{\rm sys}$ and gain (e.g. volts per Kelvin) from ambient load stare observations.
- Downsample.ini - This will downsample a data file in frequency by `factor` times and also check to see if any pointing needs to be added.
- FitJupiter.ini - This will fix the pointing, downsample, and calibrate a Jupiter observation to the ambient load. Then it will fit a Gaussian to the time ordered data to derive amplitude, pointing and beam width measurements. It will also produce a calibration scale in units of Janskys/Kelvin for every horn and frequency channel.

3 Classes

3.1 BaseClass.H5Data

Useful functions for defining new classes:

- getdset - Retrieve a dataset, if it is not in memory load it.
- setdset - Load a dataset into memory.
- resizedset - Resize a dataset by passing it a new array.
- updatedset - Update a dataset values.
- getAttr - Get an attribute (stored in output file)
- setAttr - Set an attribute
- getextra - Get a dataset from the extra outputs
- setextra - Set an array to be assigned to extra outputs (must describe the shape of the array, e.g. which axis refers to horns, frequencies, etc...)
- resizeextra - Change dimensions of an extra dataset.

N.B. Never write directly to the dset or extras attributes of the H5Data class.

Useful attributes/functions for MPI routines:

- splitType - Axis type being split for MPI purposes (i.e., either Types._HORNS_, Types._SIDEBANDS_, Types._FREQUENCY_, Types._TIME_).

- selectType - Axis type being explicity selected (i.e., as above)
- selectIndex - Index being selected along selectType axis.
- splitFields - Names of fields in COMAP data structure that will contain a split axis.
- selectFields - Names of fields in COMAP data structure that will have a selected axis.
- hi/lo - Dictionary, for each splitField, defining where in the larger structure this process is accessing data.
- ndims - Dictionary containing dimensions of each dataset in memory for this process.
- fullFieldLengths - Dictionary containing dimensions of each dataset in memory in total.
- getDataRange - Function that returns how to split N values between M processes.