

Data Processing School :: Exercise 63

Source directory	/data/lofarschool/data/Exercise-63
Contact person	Ger van Diepen

Context

This exercise helps in understanding the structure of a MeasurementSet and the basic (Python-based) tools that can be used to look at the data in a MeasurementSet.

Prerequisite

Not necessary, but makes code more understandable.

- python
- pyrap.tables (see www.astron.nl/casacore/trunk/pyrap/docs)
- TaQL (see www.astron.nl/casacore/trunk/casacore/doc/notes/199.html)
- matplotlib
- numpy

Description

This exercise consists of several parts

- look at a subtable of an MS to obtain info about the spectral band(s) used.
- use TaQL to remove obsolete antennae and to determine how many data points are flagged.
- use pyrap.tables and matplotlib to make some plots.
- use casabrowser and casaviewer to inspect the visibility data.

Files & Directories

/data/lofarschool/data/Exercise-63/ex1.ms is a WSRT MeasurementSet containing about 400 MBytes of visibility data. It contains the fake antennae 14 and 15. Exercise 1 deals with this.

/data/lofarschool/data/Exercise-63/ex2.ms is a LOFAR MeasurementSet containing about 500 MBytes of visibility data. It contains autocorrelations. In the same way as exercise 1 you could create a subset without autocorrelations.

By default the scripts shown in the following sections use ex1.ms. They can also be used for other measurementsets by giving the ms name as the first argument. For example:

```
python ex63-1.py ex2.ms
```

Before working with these tables, you need to copy them to your working directory in /data/lofarschool/users/<yourdir> like

```
cp -r /data/lofarschool/data/Exercise-63/ex1.ms .
```

You should also copy the .py files to be able to change them as you like.

Step-by-step instructions

Execute

```
source /app/scripts/doCasa
source /app/scripts/doCasacore
source /app/scripts/doPyrap
```

to setup the environment.

The following scripts perform various operations on the data set ex1.ms. This is a Westerbork data set of 3C343. Some scripts can take some time as they need to read and process over 400 MByte of data.

```
python ex63-1.py
```

finds the spectral info (#bands, #channels, freqs) (uses subtable SPECTRAL_WINDOW).

```
python ex63-2.py
```

creates a subset excluding antennae 14 and 15.

```
python ex63-3.py
```

determines in the subset how many data points exceed 5 times the median of the spectrum per timestamp per baseline. This could be used to flag data affected by RFI. Note that 5 is just a number in this example. It also determines how many of these data points are already flagged.

```
python ex63-4.py
```

makes a plot of the XX amplitude of channel 0 vs time for all baselines.

```
python ex63-5.py
```

makes a plot of the XX amplitude (averaged spectrum) vs time for all baselines.

```
python ex63-6.py
```

makes a plot of each channel for a given baseline.

```
casabrowser cex1.ms
```

starts the browser on the MeasurementSet. Play around with the browser. Find the subtables by clicking on the 'table keywords' button on the left. A subtable can be opened by double clicking on it.

The tool buttons at the top make it possible to do selection, plotting, etc. Clicking on a column header performs a sort. Note that only a limited number of rows is loaded in memory as can be seen at the bottom; the next chunk can be loaded by clicking the appropriate buttons.

Please give me your remarks as they can be valuable for the developers in Socorro.

```
casaviewer ex1.ms
```

starts the viewer on the MeasurementSet. Unfortunately it can only operate on a proper MS, not on a subset. It takes some time to start because it has to deal with all the data. You can see blueish bars which correspond to flagged baselines of antenna 14 and 15.

You can play a movie using the buttons under the image. At the top you can click the wrench tool button. It opens a widget that makes it possible to choose how the data are displayed. In the 'display axes' section you can choose to sort on Baseline Length. Other options are to display phase, etc. Another option is to choose Time as the animation axis.

Color fiddling and so can be done with the mouse after choosing which mouse button to use at the second row of buttons at the top.

Interactive flagging is possible by drawing a region around the bad pixels. By default the left mouse button can be used for zooming (draw rectangle and double click inside it). The right mouse button can be used for flagging (draw region and double click to flag all pixels inside it). The flag options panel can be used to extend it along other axes.

Bug Reports

If you are experiencing trouble getting the exercise to work, this is the place to leave a note about it.

From:
<https://www.astron.nl/lofarwiki/> - **LOFAR Wiki**

Permanent link:
https://www.astron.nl/lofarwiki/doku.php?id=public:meetings:2009-02_processing_school:exercise_63&rev=1319550485

Last update: **2011-10-25 13:48**

