Practical: Make a VLA interferometric image

**Starting data: Raw imaging data from VLA.**

**Objective: Reconstruct the sky image**

In a radio array telescope, complex visibilities are recorded for each baseline (i.e. for each couple of antennas) with a certain time and frequency resolutions.

In order to produce an image of the sky, these raw data must go through a processing pipeline. This pipeline constitutes of ***three*** main processes:

1. Flagging
2. Calibration
3. Imaging

**The raw imaging data**

The data which we will be starting with is in .ms format - ‘3c391\_ctm\_mosaic\_10s\_spw0.ms’.

It contains a table for each baseline with the complex visibilities at any time and frequency, other than the metadata on the telescope on the observation, such as the date of the observation, the bandwidth, the backends used, etc. (~see. slide 17 from lecture 7)

Usually it also contains the observation of the calibrator.

# **Part 1 - Data inspection**

One of the mostly used programs to examine and process imaging file is CASA (casa.nrao.edu).

In CASA it is possible to set specific parameters and to recall different modules in order to produce a radio image. A list of the available modules, their functionality and parameters to set is given in [casa.nrao.edu/docs/TaskRef/TaskRef.html](http://casa.nrao.edu/docs/TaskRef/TaskRef.html)

Note that while we are using basic techniques and procedures, many advantage functionalities and strategies are possible with this program.

1. To load it in your playground type *use Casa.*
2. Now in your local directory make a folder called ***Imaging.*** Cd to this directory.

Make a folder ***plots*** where we will be storing all the images.

1. In this folder type the following sequence of commands to obtain the data which we will be working on :
   1. wget <http://casa.nrao.edu/Data/EVLA/3C391/3c391_ctm_mosaic_10s_spw0.ms.tgz> -- this will get the zipped data.
   2. Unzip this by using : tar -zxvf [3c391\_ctm\_mosaic\_10s\_spw0.ms.tgz](http://casa.nrao.edu/Data/EVLA/3C391/3c391_ctm_mosaic_10s_spw0.ms.tgz)
   3. Once the unzipping is done you will have the folder : [3c391\_ctm\_mosaic\_10s\_spw0.ms](http://casa.nrao.edu/Data/EVLA/3C391/3c391_ctm_mosaic_10s_spw0.ms.tgz) in your directory.
   4. remove the downloaded tarred version by doing:   
      rm [3c391\_ctm\_mosaic\_10s\_spw0.ms.tgz](http://casa.nrao.edu/Data/EVLA/3C391/3c391_ctm_mosaic_10s_spw0.ms.tgz)
2. We will use a software called *CASA* (Common Astronomy Software Applications)which is used to calibrate, analyze and image the data from radio interferometers such as EVLA
3. To start CASA type *casapy*. If everything is fine, a window named Log Messages   
   should appear. Most of the on-screen output will appear in this window.
4. You are now in CASA!
5. We will now examine the details of the observation with the listobs() module, <http://casa.nrao.edu/docs/TaskRef/listobs-task.html>
6. The only information necessary to call listobs() is the name of the ms file, represented by the vis parameter. Therefore, you can type  
   *vis = '*3c391\_ctm\_mosaic\_10s\_spw0.ms*'  
   listobs()*Alternatively it is possible to directly insert the parameter in the function:  
   *listobs(vis = '*3c391\_ctm\_mosaic\_10s\_spw0.ms*')*
7. Note that in CASA all the parameters to every module must be passed as strings (i.e. within the ' '), even the numeric values.
8. The log output will appear in the Log Message window. Here you can find detailed information on the antennas of the telescope as well as on the observation itself. Study this log and find out:
   1. What are the sources that have been observed? Find out which one of them are the calibrators?
   2. How many antennas have been used in the observation? What were their dimensions?
   3. What was the total observation duration? How many sky fields have been observed? How many different sources were present and why the two numbers don’t coincide in your opinion?
9. It is possible to save the content of the Log Messages to view it later in a text editor. In Log Messages go to **file → save as → save it as list\_obs\_log.**
10. The real configuration of the antennas, as described in the log file, can be plotted. To do this call the plotants() routine with the file name defined as in the previous listobs() command. In this configuration see the antenna towards the center. ***Note down the name of this antenna as it will be required in future.***
11. Save and report the obtained image.
12. The routine to actually open and inspect the data is called plotms(). The default view will plot the amplitude vs time for all the scans present in the file. But, it permits to plot all or part of the data with many different axes and options. It will open a graphical interface that shows the plot and that allows to change the options initially passed to the function.   
    Call the routine passing the filename in order to open the graphical interface. It can be useful to rapidly check if there is some problem in one or more antennas or with a certain source. So plot the following:   
    1) Amp vs. time, 2) Phase vs. time, 3) Frequency vs. time, 4) Amplitude vs. UVdist, 5) Amplitude vs. UVdist setting the item 'field' in the header 'data' to a value of 3, 6) Amplitude vs. UVwave.
13. In the Amp. vs. time plot plotted above try to get color coded visibilities plot as shown in lecture. (~see slide 25 lec. 7)
14. From the resolution formula (~see slide 6 lec. 6) try to calculate the resolution for EVLA. (Hint: refer listobs output to find mid channel wavelength then get the baseline)
15. Compare the plots number 4 and 5 plotted above. Why do they seem different?
16. What do you think is plot number 6 plotting?
17. Now, use plotms() to check if all the baselines are fine during the observation by plotting the signal measured by each baseline.

Report the relative plot, problems that are eventually highlighted and the causes of these problems. (Hint: try to plot other parameters, such as the signal measured by the first antenna of each baseline).

# 

# 

# **Part 2 - Flagging**

In order to produce quality images bad data need to be flagged. Data can be corrupted due to malfunctionings of some system of the telescopes (e.g. antennas, frontends, recorders, etc.), presence of strong terrestrial interferences, etc.

1. This corrupted data can be manually flagged in CASA using the flagdata() module.
2. It is common that the first scan of an observation is only a quick test that must be removed. To do this use the flagdata() module with the usual vis parameter and specifying that we want to only remove the first scan, therefore set *scan= '1'*.
3. Another common operation is to flag the first seconds of each scan, since the system will need some time to start to operate normally. The mode designed for this purpose is called *quack* and its behaviour is controlled by the parameters *quackinterval, quackmode and quackincrement.* Pass the parameter *mode='quack'* to flagdata() and read the relative usage file in the CASA web page to properly set the other three parameters in order to flag the first 10 seconds of each scan in the observation.
4. We know from the telescope log output that antenna ‘ea15’ contains corrupted data and therefore must be flagged as well.
5. Finally, flag the most relevant problem you may have found in the previous section.
6. At the end of the flagging operation, again plot the amplitude detected from each antenna.
7. Compare the plot from above question to the amplitude vs. time plot obtained before flagging. What difference do you see?
8. In this exercise we will not mask terrestrial signals, because it is usually done automatically by computer programs. For example, the the standard program to flag and mask RFI in imaging processes with LOFAR is called AOFlagger <http://sourceforge.net/p/aoflagger/wiki/Home/>.

# 

# 

# **Part 3 - Imaging**

We will make an image with the flagged data prepared after above two steps.

1. To do so, we will use the following CASA task called ***clean***.  
   The clean task interactively cleans the visibility data and also outputs multiple information files. See : refer <http://casa.nrao.edu/docs/TaskRef/clean-task.html> for available options.
2. First assign an output file name and use it to store files. The command to do so is:   
     
   clean(*vis = '*3c391\_ctm\_mosaic\_10s\_spw0.ms*'*,  
    imagename='O/P\_file\_name',  
    field='',spw='',  
    mode='mfs',  
    niter=5000,  
    gain=0.1, threshold='1.0mJy',  
    psfmode='clark',  
    imagermode='mosaic', ftmachine='mosaic',  
    multiscale=[0],   
    interactive=True,  
    imsize=[480,480], cell=['2.5arcsec','2.5arcsec'],  
    stokes='I',  
    weighting='briggs',robust=0.5,  
    usescratch=False)

1. To see this image do :   
   viewer('O/P\_file\_name.image')  
   What do you observe?
2. To get all the images from this practical onto your local machine run:

*scp -r -oProxyCommand="ssh username@portal.lofar.eu nc -v %h %p" lhd002:path\_to\_plots\_folder .*

In order to know your current folder you do *‘pwd’*