Produce an interferometric image

Getting started

- Read carefully the slides **The Techniques of Radio Interferometry II: Calibration** and **The Techniques of Radio Interferometry II: Imaging**. This practicum will be based on them. References to these slides will be in red.

- We will make use of a software suite called CASA (casa.nrao.edu). Commands in CASA will be highlighted in green.

- The practicum is divided into sections that follow the lecture notes. Please follow the order.

Evaluation

- At the end of the practicum, you need to produce a short report containing the plots and answering the questions below.

- Read carefully what it is being asked and, in case of doubt, ask.

- Each section will explain what to do and what to put in the final report. Answers and plots to include in the report will be highlighted in blue.

Goals

- Basic usage of CASA, the most common program for interferometric image.

- Understanding and inspection of data products from interferometers.

- Flagging and cleaning imaging data.

- Calibrate and deconvolve data in order to produce a radio image.

Part 1 - How to use CASA

CASA (Common Astronomy Software Applications) is a collection of routines for radio imaging.
We will run it on a publicly available measurement set to obtain a radio image.
For simplicity, you should access the folder containing the measurement set before to running CASA.

You can use CASA on any of the computers in the Master rooms. You need to use always the same computer because the data will be stored locally. You can also ssh the pc with ssh username@uXXXXXX.uwp.science.uva.nl
The commands necessary to set up the system after you log in are (substitute your username):

```
mkdir /scratch/username
mkdir /scratch/username/RadioPrakticum
cd /scratch/username/RadioPrakticum
wget http://casa.nrao.edu/Data/EVLA/3C391/3c391_ctm_mosaic_10s_spw0.ms.tgz
tar -zxf 3c391_ctm_mosaic_10s_spw0.ms.tgz
rm 3c391_ctm_mosaic_10s_spw0.ms.tgz
```

Each time you login, you need to run

```
cd /scratch/username/RadioPrakticum
source /public/API/scripts/sw-astro.sh
setup_casa
casa
```

You can also install CASA on your own laptop.
For unix users (substitute /path/to with the actual path):

```
cd /path/to
wget https://casa.nrao.edu/download/distro/linux/release/el7/casa-release-4.7.2-el7.tar.gz
wget http://casa.nrao.edu/Data/EVLA/3C391/3c391_ctm_mosaic_10s_spw0.ms.tgz
tar -zxf 3c391_ctm_mosaic_10s_spw0.ms.tgz
tar -zxf casa-release-4.7.2-el7.tar.gz
rm casa-release-4.7.2-el7.tar.gz 3c391_ctm_mosaic_10s_spw0.ms.tgz
/path/to/casa-release-4.7.2-el7/bin/casa
```

For mac users:
Visit https://casa.nrao.edu/casa_obtaining.shtml, download and install the version for your distribution. Also download the data at
http://casa.nrao.edu/Data/EVLA/3C391/3c391_ctm_mosaic_10s_spw0.ms.tgz
Windows is not supported.

Commands in CASA can be run with command().
The list of available commands, their description and the accepted arguments can be found at casa.nrao.edu/docs/TaskRef/TaskRef.html

Each command has a set of arguments with default values. Default values can be changed for the whole session or temporarily inside the command. For example, command(arg1='1', arg2='2') will set arg1 and arg2 to the indicated values for this command and will not affect the following, while writing arg1='1' will set arg1 to 1 for the whole session, except where a different value is explicitly indicated. Note that all the values passed to the arguments must be surrounded by ''. A particularly important argument is vis, which indicates the ".ms" directory path.

The "Log Messages" window contains the text output of the commands. It is also possible to insert notes or search for specific text. The content of the log can be saved to a text file.

Part 2 - Inspecting visibility data

(Slide II-20)

General information on the observation can be obtained using listobs(). Use this command to check which telescope was used, which day and how long the observation was in total. Also report the total number of scans (i.e. the different sub-observations), the number of fields on the sky observed and the number of antennas used.

Among the four sources observed, three are point-source calibrators and one (3C391) is the target. It was observed in seven contiguous fields necessary to cover the whole extended source. Search in Simbad which kind of source it is.

The antenna configuration reported in the log can be plotted by using plotants(). It should look familiar. Insert it in the report.

Visibility properties can be plotted by using plotms(). This graphical interface has many plotting options to select. This command can be very slow, therefore we will plot only part of the data.

Arguments can either be given when running the command or interactively.

For example, a plot similar to the one on Slide II-24 can be obtained by inserting "0:22~41" in the "spw" field (it selects channels 22 to 41 in subband 0), "RR" in the "corr" field (it will plot only one polarisation) and by selecting "Colorize: Field" in the "Display" menu. You can average over e.g. 64 channels to make the plotting faster. Which are the two bright sources observed in the first and last scans?

Reproduce the plot on Slide II-29 (multiple antennas can be selected with the "&" symbol).

Plot amplitude vs UV distance for J1822-0938 and 3C391 (field 2) using all the frequency channels. How can you tell from the plots which one is the point source and which is the extended source?

Part 3 - Data editing

(Slide II-32)

Bad data can be manually flagged in CASA using the flagdata() module.

It is common that the first scan of an observation is only a quick test that must be removed. To do this, give the command with the argument scan='1' and indicating the filename.

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. In order to flag the first 10 seconds of each scan in the observation, we use the command

```
flagdata(vis='3c391_ctm_mosaic_10s_spw0.ms', mode='quack', quackinterval=10.0, quackmode='beg')
```

During the observation, two antennas had problems ('ea13' and 'ea15'). We can see this in the observation log (http://www.vla.nrao.edu/operators/logs/2010/4/2010-04-24_0801_TDEM0001.pdf).

We can flag those antennas using

```
flagdata(vis='3c391_ctm_mosaic_10s_spw0.ms', antenna='ea13,ea15')
```

Create again the plot visible on Slide II-24 and compare it with the one obtained before flagging. Insert it in the report. Which differences do you notice? (You can try to zoom to a specific field to have a better idea)

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/)

We have a pretty clean dataset, let's try to image it! Begin with the point-source calibrator

```
clean(vis='3c391_ctm_mosaic_10s_spw0.ms', imagename='cal_clean', field='J1331+3030', niter=5000, threshold='1.0mJy', imsize=[480,480], cell=['2.5arcsec','2.5arcsec'], stokes='I', interactive=True)
```

Report the image that is shown, then click the red X to avoid additional processing. How does the image look like with respect the expected one? Why?

Now image one field of the target source and again click the red X after you have done

```
clean(vis='3c391_ctm_mosaic_10s_spw0.ms', imagename='target_clean', field='2~8', niter=5000, threshold='1.0mJy', imagermode='mosaic', multiscale=[0], imsize=[480,480], cell=['2.5arcsec','2.5arcsec'], weighting='briggs', robust=0.5, stokes='I', interactive=True)
```

Insert it in the report. How does it compare to the previous one?

The images created can be viewed again using viewer('imagename.image'), where imagename is the value given to this argument in clean().

Part 4 - Calibration

(Slide II-42)

We are now going to calculate gains for the calibrator and, assuming that these gains do not vary during the observation, we will apply them to the source under study. Why is it necessary to calibrate each observation of the telescope?

In order to calculate the gains, we compare the calibrator to a pre-existing model, under the assumption that the calibrator does not vary significantly. Standard models can be listed using setjy(vis='3c391_ctm_mosaic_10s_spw0.ms', listmodels=T). We will use 3C286 (also known as J1331+3030) and, since the observation is in band "C" (which frequencies does it cover?), the model to use is "3C286_C.im". Gains are calculated with the command

```
setjy(vis='3c391_ctm_mosaic_10s_spw0.ms', field='J1331+3030', standard='Perley-Butler 2013', model='3C286_C.im', usescratch=False, scalebychan=True, spw='')
```

Signal delays from different antennas is carefully calculated when the telescope is built. Why the delays between antennas is so crucial?

Changes in the system occur e.g. because of variations in antenna positions when they are moved. Antenna positions are regularly monitored. It is possible to update their values using the command

```
gencal(vis='3c391_ctm_mosaic_10s_spw0.ms',
caltable='3c391_ctm_mosaic_10s_spw0.antpos', caltype='antpos')
```

Check the command output, offsets are in meters. What is the maximum correction?

Residual phase delays due e.g. to temperature variation or source elevation can be corrected with respect to one antenna (e.g. 'ea21') using calibrator observations

```
gaincal(vis='3c391_ctm_mosaic_10s_spw0.ms',
caltable='3c391_ctm_mosaic_10s_spw0.G0all', field='0,1,9', refant='ea21', spw='0:27~36',
gaintype='G', calmode='p', solint='int', minsnr=5,
gaintable=['3c391_ctm_mosaic_10s_spw0.antpos'])

gaincal(vis='3c391_ctm_mosaic_10s_spw0.ms',
caltable='3c391_ctm_mosaic_10s_spw0.G0', field='J1331+3030', refant='ea21',
spw='0:27~36', calmode='p', solint='int', minsnr=5,
gaintable=['3c391_ctm_mosaic_10s_spw0.antpos'])

gaincal(vis='3c391_ctm_mosaic_10s_spw0.ms',
caltable='3c391_ctm_mosaic_10s_spw0.K0', field='J1331+3030', refant='ea21',
spw='0:5~58', gaintype='K', solint='inf', combine='scan', minsnr=5,
gaintable=['3c391_ctm_mosaic_10s_spw0.antpos', '3c391_ctm_mosaic_10s_spw0.G0'])
```

Plot the phase variation over time for antenna 'ea02' (R polarisation) using

```
plotcal(caltable='3c391_ctm_mosaic_10s_spw0.G0all', xaxis='time', yaxis='phase',
antenna='ea02', poln='R', iteration='antenna', plotrange=[-1,-1,-180,180])
```

Notice that phase delay can be corrected if the variation is smooth over time, i.e. if there are not phase jumps. If you plot the phase variation of antenna 'ea05' you can notice phase jumps. Therefore, flag out this antenna before to continue the analysis. Check that the other antennas are fine (you can call plotcal with antenna='', then you can iterate through the antennas with the "Next" button).

We now need to correct for gain variation with frequency (bandpass calibration). To understand this, plot the amplitude versus channel for RR polarisation, using a color scale based on the first antenna in the baselines, for the field 'J1331+3030'. How should the plot look for a perfect instrument? The bandpass calibration is done with

```
bandpass(vis='3c391_ctm_mosaic_10s_spw0.ms',
caltable='3c391_ctm_mosaic_10s_spw0.B0', field='J1331+3030',spw='', refant='ea21',
combine='scan', solint='inf', bandtype='B',
gaintable=['3c391_ctm_mosaic_10s_spw0.antpos', '3c391_ctm_mosaic_10s_spw0.G0',
'3c391_ctm_mosaic_10s_spw0.K0'])
```

Now, we can calculate the gains for the target source using the calibrators as discussed at the beginning of this section. We will run the command three times, once for each calibrator.

```
gaincal(vis='3c391_ctm_mosaic_10s_spw0.ms',
caltable='3c391_ctm_mosaic_10s_spw0.G1', field='J1331+3030',spw='0:5~58', solint='inf',
refant='ea21', gaintype='G', calmode='ap', solnorm=F,
gaintable=['3c391_ctm_mosaic_10s_spw0.antpos', '3c391_ctm_mosaic_10s_spw0.K0',
'3c391_ctm_mosaic_10s_spw0.B0'], interp=['linear', 'linear', 'nearest'])

gaincal(vis='3c391_ctm_mosaic_10s_spw0.ms',
caltable='3c391_ctm_mosaic_10s_spw0.G1', field='J1822-0938', spw='0:5~58', solint='inf',
refant='ea21', gaintype='G', calmode='ap',
gaintable=['3c391_ctm_mosaic_10s_spw0.antpos', '3c391_ctm_mosaic_10s_spw0.K0',
'3c391_ctm_mosaic_10s_spw0.B0'], append=True)

gaincal(vis='3c391_ctm_mosaic_10s_spw0.ms',
caltable='3c391_ctm_mosaic_10s_spw0.G1', field='J0319+4130', spw='0:5~58', solint='inf',
refant='ea21', gaintype='G', calmode='ap',
gaintable=['3c391_ctm_mosaic_10s_spw0.antpos', '3c391_ctm_mosaic_10s_spw0.K0',
'3c391_ctm_mosaic_10s_spw0.B0'], append=True)
```

Now, we need to apply the calculated gains to the three calibrators and the target source

```
applycal(vis='3c391_ctm_mosaic_10s_spw0.ms', field='J1331+3030',
gaintable=['3c391_ctm_mosaic_10s_spw0.antpos', '3c391_ctm_mosaic_10s_spw0.K0',
'3c391_ctm_mosaic_10s_spw0.B0'], calwt=[False], parang=True)

applycal(vis='3c391_ctm_mosaic_10s_spw0.ms', field='J0319+4130',
gaintable=['3c391_ctm_mosaic_10s_spw0.antpos', '3c391_ctm_mosaic_10s_spw0.K0',
'3c391_ctm_mosaic_10s_spw0.B0'], calwt=[False], parang=True)

applycal(vis='3c391_ctm_mosaic_10s_spw0.ms', field='J1822-0938',
```

```
gaintable=['3c391_ctm_mosaic_10s_spw0.antpos', '3c391_ctm_mosaic_10s_spw0.K0',
'3c391_ctm_mosaic_10s_spw0.B0'], calwt=[False], parang=True)
```

```
applycal(vis='3c391_ctm_mosaic_10s_spw0.ms', field='2~8',
gaintable=['3c391_ctm_mosaic_10s_spw0.antpos', '3c391_ctm_mosaic_10s_spw0.K0',
'3c391_ctm_mosaic_10s_spw0.B0'], calwt=[False], parang=True)
```

Create a new measurement set containing the calibrated data using

```
split(vis='3c391_ctm_mosaic_10s_spw0.ms', outputvis='3c391_calibrated.ms',
datacolumn='corrected')
```

Plot the amplitude versus channel for RR polarisation, using a color scale based on the first antenna in the baselines, for the field J1331+3030. Insert the plot. How does it compare to the uncalibrated data?

Create two images for the new calibrated measurement set with the same command used at the end of part 3. Report the plots. Describe what you see in the images, how they changed and what must be still improved.

Part 5 - Deconvolution

(Slide III-3)

Earlier we used the clean() command only to plot the fields. Actually, clean can deconvolve the image before plotting (Slide III-19+). Explain the necessity to deconvolve the image.

Use the commands used at the end of part 3 to plot the point source but this time set the interactive argument to False. Report the plot obtained. What did change with respect the previous part?

The clean command produces various output files. The .image extension indicates the radio image. The .psf extension indicates the point source function, i.e. the synthesized beam. Use the code developed in the first practicum to simulate the beam pattern produced by the same array configuration and report the comparison between simulation and measurement.

Use again the same command used at the end of part 3 to plot the target source. This time we are going to deconvolve the image. It is possible to select a region of interest in the dirty image. You can draw a region of interest using the right mouse button. A green rectangle should appear. You can cancel the selection by double-clicking with the right mouse button outside the selected region. Alternatively, you can confirm the selection by double-clicking with the right mouse button inside the selected region. The border of a confirmed region will turn white.

Select the bright part of the image (around the red area). Click on the green triangle to proceed with the deconvolution (it will take a long time). Report the plot obtained.

Try to apply the three weightings indicated on Slide III-15 to obtain three plots for the calibrator J1331+3030. Which combination seems to perform best?