# Guide to ALMA Reduction Data with CASA

Patrizia Bussatori, Fabrizio Muratore

May 2022

# Premise

This book contains the notes taken from the laboratory lessons of the astronomical interferometry course held by prof. D'Onofrio for the master degrees in Cosmology and Astrophysics, in the academic year 2021/2022. As these are not official professor notes, we do not guarantee the correctness of the contents. Indeed, any error reporting is welcome! Enjoy the reading!

*Patrizia Bussatori & Fabrizio Muratore*

# Contents

# Chapter 1

# Calibration steps

## 1.1 The software: CASA

This small guide has been designed for data calibration and data reduction based on CASA software, the Common Astronomy Software Applications package (1.1). This is a primary data processing software for the Atacama Large Millimeter/submillimeter Array (ALMA) and NSF's Karl G. Jansky Very Large Array (VLA), and is frequently used also for other radio telescopes. It can be used for both single-dish and aperture-synthesis telescopes, and one of its core functionalities is to support the data reduction and imaging pipelines for ALMA, VLA and the VLA Sky Survey (VLASS).

### 1.1.1 In laboratory

First of all, it is necessary to access to the right directory because CASA Software is not on local computer, so it is necessary to give the following command in the terminal in order to enter in *superlab*:

```
ssh -X superlab
```

Then it is necessary to enter the following line of command:

```
export PATH=$PATH:/usr/local/casa-6.2.1/bin
```

After this, you can enter in the software simply entering in the terminal:

```
casa
```

PAY ATTENTION! You must be in the right directory where data are located. Once you have

Figure 1.1: Symbol of CASA Software.

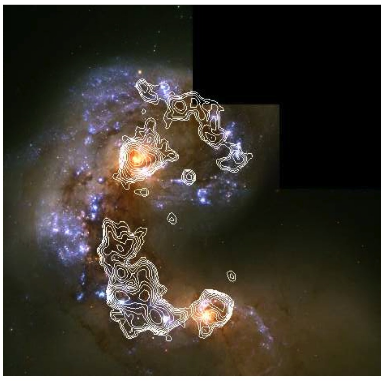

Figure 1.2: HST image of the Antennae.

done this, it is possible to begin the data calibration following the guide find in website `https://casaguides.nrao.edu/index.php?title=AntennaeBand7_Calibration_6.2`

## 1.2 Overview: the Target

The data used for this exercise are the ones of the Antennae, an extended source source which is a pair of merging galaxies, also known as NGC4038 in the North and as NGC4039 in the South. It is 22 *pc* distant from us and it is a complex structure given by two galaxies that began to interact a few hundred million years ago. In visible images, the two bright spots correspond to the nuclei of the two galaxies while extended long filaments appear brown and pervade the region between the nuclei. All over the system, star-forming regions surrounded by *HII* regions are observable.

## 1.3 Unpack the Data

This guide requires CASA 6.2 and assumes that you have downloaded Antennae_Band7_UnCalibratedMSAndTablesForReduction.tgz from `https://casaguides.nrao.edu/index.php?title=AntennaeBand7#Obtaining_the_Data`. Once you have downloaded the data, unpack the file in a terminal outside CASA then change directory to the new directory. In this directory there are some files with extension *.ms*: these are measurements set files (MS). In addiction there are calibration tables containing system temperature (Tsys), water vapor radiometer (WVR), and antenna position information.

## 1.4 Confirm the CASA version

This guide has been written for CASA release 6.2.1.7. Please confirm your version before proceeding.

```
import casalith
version = casalith.version_string()
print("You are using {}".format(version))
if (version < '6.2'):
    print ("YOUR VERSION OF CASA IS TOO OLD FOR THIS GUIDE.")
    print ("PLEASE UPDATE IT BEFORE PROCEEDING.")
else:
    print ("Your version of CASA is appropriate for this guide.")
```

## 1.5 Initial Inspection

First of all, let's see what kind of data there are. The 10 data sets each target either the Northern or the Southern Mosaic, as follows.

For **Northern mosaic**:

- uid___A002_X1ff7b0_Xb.ms
- uid___A002_X207fe4_X3a.ms
- uid___A002_X207fe4_X3b9.ms
- uid___A002_X2181fb_X49.ms

For **Southern mosaic**:

- uid___A002_X1ff7b0_X1c8.ms
- uid___A002_X207fe4_X1f7.ms
- uid___A002_X207fe4_X4d7.ms
- uid___A002_X215db8_X18.ms
- uid___A002_X215db8_X1d5.ms
- uid___A002_X215db8_X392.ms

The first step is to get basic information about the data: targets observed, time range, spectral setup, and so on. We do this using the task *listobs*, which will output a detailed summary of each dataset. Listobs reports various metadata related to an MS. The listing is sent to the logger or can be saved to a file. Standard MS selection parameters can be used to limit the listing.

Note that you may also want to take care to paste a line at a time if you are having trouble copy and pasting. Moreover, sometimes you have to press return twice to execute.

In this case the command defines a python list called basename_all, which contains the name of all 10 MS files. The "for" loop executes for each element in basename_all, calling listobs and directing the output to a file called, e.g., uid___A002_X1ff7b0_Xb.listobs.txt for the first measurement set.

The output shows that three sources were observed in each data set: 3c279, Titan, and the Antennae.

1. The **Antennae** are our science target. Note that the source name changes between the Northern Mosaic, where it is "NGC4038 - Antennae", and the Southern Mosaic, where it is just "Antennae". Also note that the source corresponds to a number of individual fields (see the Field ID column). These are the individual mosaic pointings. There are 23 for the Northern Mosaic and 29 for the Southern Mosaic.
2. **Titan** is observed once and will be used to set the absolute flux scale of the data.
3. **3c279** plays two roles: it will serve as our bandpass calibrator, to characterize the frequency response of the antennas, and because it is fairly close on the sky to the Antennae (18.6 degrees away) it will serve as our secondary calibrator (also referred to as the "phase calibrator" or "gain calibrator"), to track changes in the phase and amplitude response of the telescopes over time. Observations of 3c279 are interleaved with observations of the Antennae.

The output also shows that the data contain many spectral windows. Using the labeling scheme in the listobs above these are:

- **spw 0** targets $\sim 185\ GHz$ and holds water vapor radiometer data;
- **spw 1** and **spw 3** hold our science data. These are "Frequency Domain Mode" (FDM) data with small (0.49 $MHz$) channel width and wide (1.875 $GHz$) total bandwidth. As a result these have a lot of channels (3840). Spw 1 holds the lower sideband (LSB) data and includes the CO(3-2) line. We will focus on these data. For the CO(3-2) line the channel width corresponds to 0.426 $km/s$ and the bandwidth of spw 1 to 1634 $km/s$;
- **spw 2** and **spw 4** hold frequency-averaged versions of spw 1 and 3. These are used for some quick automated inspection. We will not use them here but we will carry out an equivalent inspection using spw 1;

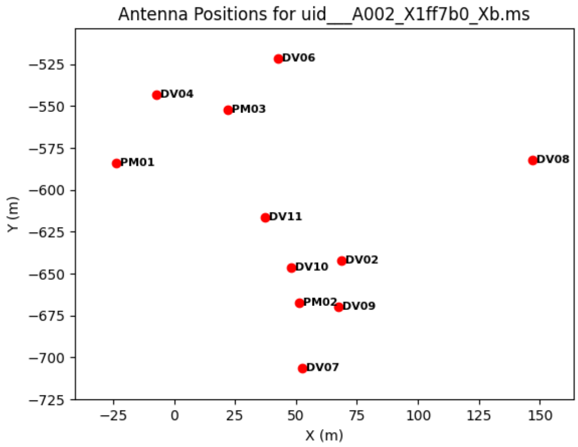


Figure 1.3: Position of antennas in dataset uid_A002_X1ff7b0_Xb obtained using task plotants.

- **spw 5** and **spw 7** hold lower a resolution processing ("Time Domain Mode", TDM) of the data from the same part of the spectrum (baseband) as spws 1 and 3. These data have only 128 channels across 2 $GHz$ bandwidth and so have a much coarser channel spacing than the FDM data. These were used to generate the calibration tables that we include in the tarball but will not otherwise appear in this guide.

The final column of the listobs output in the logger gives the scan intent. Later we will use this information to flag the pointing scans and the hot and ambient load calibration scans.

We'll now have a look at the configuration of the antennas used to take the data using the task *plotants*. This task is a simple plotting interface to produce plots of the antenna positions at the time that the data were taken. The antenna positions are taken from the ANTENNA sub-table of a MeasurementSet and are given in the local reference frame. The antennas are plotted Y vs. X in meters, where Y is toward local north and X is toward local east. The name of each antenna, and its ID when requested, is shown next to its respective location.

This command will loop through all 10 data sets, show you the antenna position for each, and save that as a file named, e.g., uid___A002_X1ff7b0_Xb.plotants.png for the first data set. Notice that the antenna setup changes, but only slightly, over the course of the 10 data sets. An example is visible in figure 1.3.

## 1.6 How to Deal with 10 Measurements Sets

It should already be clear from the initial inspection that dealing with 10 data sets at the same time can be a bit problematic. This is especially tricky in our case because the Antennae data contain two distinct sets of observations: the Northern and Southern Mosaics. The source name changes between these two scripts and there are different numbers of fields in the mosaic.

As a general rule one would reduce each individual observation separately or at the very least only group data observed in a uniform way and very close in time.

Unfortunately, a CASA Guide stepping through the reduction for each of 10 data sets would quickly become unwieldy. Therefore we will use a few tricks to reduce the Antennae data in a kind of batch mode. You have already seen the first trick: we can define a python list holding the names of each data set and then loop over this list to execute the same command on each data set.

You only need to define your list of MS files once per CASA session. Then basename_all will be a variable in the casapy shell. You can check if it exists by typing print (basename_all).

One potential problem is that the source name changes between the two data sets. Therefore at several points we will break apart our loop to iterate separately over the Northern and Southern Mosaics, where the source can be referred unambiguously. Another subtle point is that 3c279 appears with two distinct field IDs in the Southern Mosaic, but only one in the Northern Mosaic. We will largely avoid this by referring to the source by its name but if you tried to use field ID numbers and mingled the two data sets this could cause confusion.

## 1.7 A Priori Flagging

Even before we look in detail, we know that there are some data that we wish to exclude. We will start by flagging "shadowed" data where one antenna blocks the line of sight of another. We will also flag scans that were used to carry out pointing and atmospheric calibration, identified by their scan intent. Finally, we'll flag the autocorrelation data (the correlation of the signal from an antenna with itself) as we are only interested in cross-correlation data to make an interferometric image.

Start by defining our list of MS files:

```
basename_all=["uid___A002_X1ff7b0_Xb","uid___A002_X207fe4_X3a","uid___A002_X207fe4%_X3b9",
     "uid___A002_X2181fb_X49","uid___A002_X1ff7b0_X1c8","uid___A002_X207fe4_X1f7",
     "uid___A002_X207fe4_X4d7","uid___A002_X215db8_X18","uid___A002_X215db8_X1d5",
     "uid___A002_X215db8_X392"]
```

You may want to reset the flagging if you have tried this step before and are starting over though this is not necessary on your first time through. Do so using *flagdata*. This task can flag a MeasurementSet or a calibration table. It has two main types of operation. One type will read the parameters from the interface and flag using any of the various available modes. The other type will read the commands from a text file, a list of files or a Python list of strings, containing a list of flag commands (each line containing data selection parameters and any parameter specific for the mode being requested).

In the flagdata task we choose:

- vis = asdm+'.ms': each measurement set;
- mode = 'shadow': flag shadowed data;
- flagbackup = False: Do not automatically back up the flag files. We will save all of the a priori flags together using *flagmanager* at the end of this subsection and save some space and time.

The relevant calibration information has already been extracted from the pointing and atmospheric scans and we will not need them below.

First flag the pointing scans using flagdata in "manualflag" mode and selecting on "intent". Note that because the atmospheric calibration scans contain only TDM spectral windows, they will be removed automatically when we separate out the FDM data below. Then flag the autocorrelation data. Finally store the current flags information using flagmanager. Here **versionname** is just the name we assign to the current flagging state in the data. We can now roll back the flags to match the current version, called "Apriori", whenever we want.

## 1.8 Create and Apply Tsys, WVR, and Antenna Position Calibration Tables

### 1.8.1 Tsys

The Tsys calibration gives a first-order correction for the atmospheric opacity as a function of time and frequency and associates weights with each visibility that persists through imaging.

Use *gencal* to create the Tsys calibration tables from the spectral windows with CALIBRATE_ATMOSPHERE intents in listobs , these are spectral windows 5 and 7. Later in the applycal stage this TDM Tsys table will be interpolated to the FDM (3840 channels per spw) science spectral windows 1 and 3. Next we inspect the Tsys tables for the spectral window **spw** = 5 with the

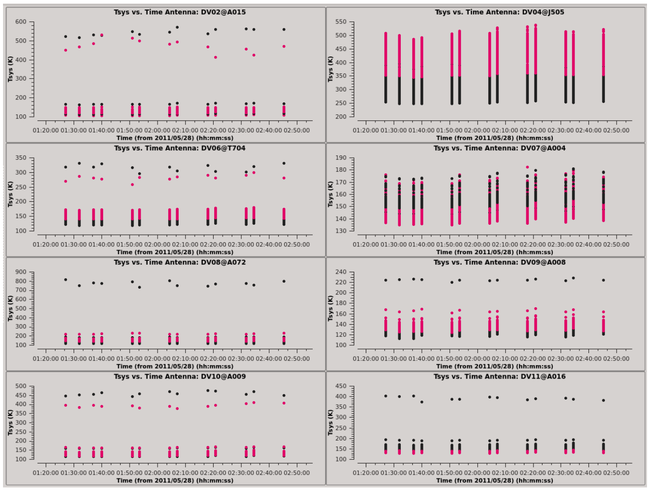


Figure 1.4: Tsys vs. time plot for uid_A002_X1ff7b0_Xb (northern mosaic). First 8 antennas. Note the high y-axis values for DV04. The two different colors indicate the two polarizations (XX and YY).

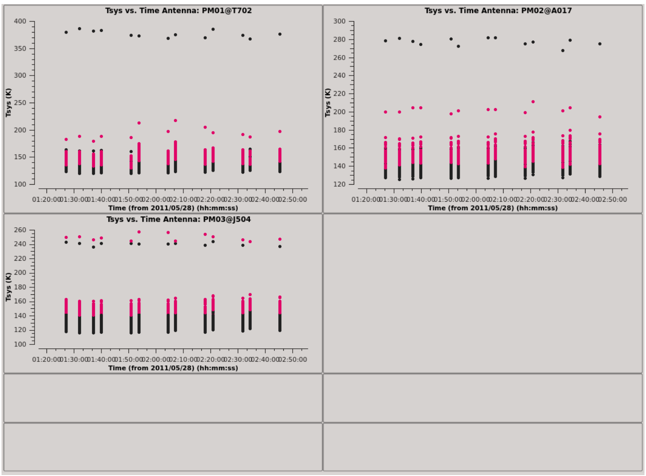


Figure 1.5: Tsys vs. time plot for uid_A002_X1ff7b0_Xb (northern mosaic). Remaining antennas.

task *plotms*. It is a task for plotting and interacting with visibility data. A variety of axes choices (including data column) along with MS selection and averaging options are provided. All of the provided parameters can also be set using the GUI once the application has been launched. We want to check that Tsys data have reasonable values and identify any unexpected features as a function of either time or frequency.

We start by plotting the Tsys for all the antennas and polarizations (XX and YY) as a function of time for each, plots are shown in the figure 1.4 and 1.5.

Because 8 panels (2 panels for each antenna - LSB and USB) are not enough to show all antennas on one page, there are two plotms calls: one for the first 8 antennas (antenna='0 $\sim$ 7'), and then for the remaining antennas (antenna='8 $\sim$ 15').

The Tsys values in Figure 1.4 and Figure 1.5 look reliable, with typical values $\sim 150K$ except for some large values of Tsys at $\sim 400$ and $500K$ for DV04. We will flag the data for that antenna later.

We will also want to look at Tsys as a function of frequency, as in figure 1.6 and 1.7.

Now have a look at the Tsys vs. frequency plots. You can see the effect of a close pair of atmospheric ozone absorption lines at about 343.2 $GHz$ that makes Tsys larger near that frequency in all antennas. Applying the Tsys calibration tables will minimize the contribution of these atmospheric lines. Again DV04 stands out with its very high Tsys.

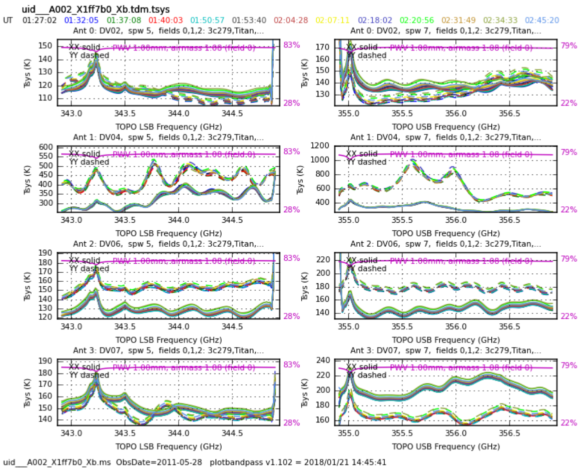

Figure 1.6: Tsys vs. frequency plot for uid_A002_X1ff7b0_Xb (northern mosaic). First 4 antennas. Note the high y-axis values for DV04 and the mesospheric line near 343.2 $GHz$.

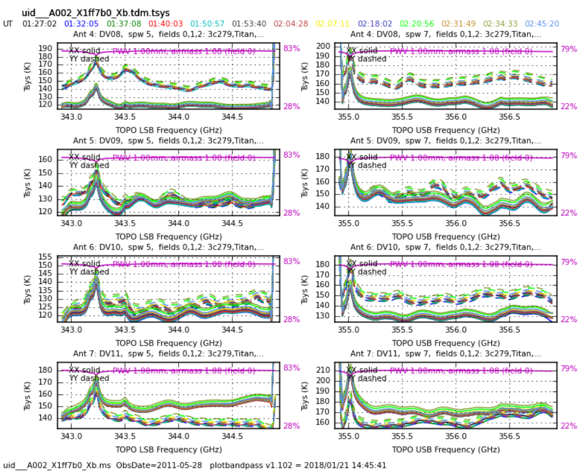

Figure 1.7: Tsys vs. frequency plot for uid_A002_X1ff7b0_Xb (northern mosaic). Next 4 antennas.

### 1.8.2 WVR

The WVR calibration uses observations of the wings of the 183 $GHz$ atmospheric water line to correct for phase variations as a function of time. The approach is very similar to one used for Tsys.

### 1.8.3 Antenna Positions

In general, the most appropriate antenna positions for the dataset will be determined and updated if needed during QA2 [1].

The antenna position table reflects refinements in the measured positions of the antennas from those stored in the data. The *gencal* command will now be used to put antenna position data into each observation. Again, gencal will merely append to existing antenna position data, ruining any subsequent results. We start by removing any existing antenna position refinements, followed by defining the antenna names, then their refinements (both as arrays), finally running gencal to create the information that CASA can refer to for antenna positions.

### 1.8.4 Applycal

We are now ready to apply the Tsys and the WVR calibration tables to the data with *applycal*, which reads the specified gain calibration tables, applies them to the (raw) data column, and writes the calibrated results into the corrected column. Again, we loop through all the datasets. It is important to only apply Tsys and WVR corrections obtained close in time to the data being corrected, so in addition to looping over data sets we define the list of unique source names and loop over these. Then by setting gainfield and field to the same value we ensure that Tsys and WVR calibrations are only applied to the source for which they are measured. Because the source has a different name in the Northern Mosaic and the Southern Mosaic, we will carry out two loops. We will only correct spw 1 and 3, our science windows, because we will drop the other data in a moment.

The applycal task has much more flexibility for interpolating and applying calibrations derived in one spectral window to another, even if they do not share the same spectral shape (number of channels and channel width). This functionality is used to interpolate the TDM (128 channel) Tsys measurements to the FDM (3840 channel) spectral windows.

## 1.9 Inspect Data

We are not quite done with the original ".ms" data sets yet. Before going further it will be useful to use plotms to show the effects of applying the calibration. In the process we'll take a quick look at each antenna and search for pathologies in the data.

For this basic inspection, we want to compare the phase and amplitude as a function of frequency and time in the DATA and CORRECTED columns of each measurement set. The CORRECTED column has had the Tsys and WVR calibrations applied and so we expect lower phase scatter and flatter amplitude response as a function of time and frequency. We are looking for antenna-based issues, so cycling through a set of baselines that includes each antenna once will be a good start. We'll focus these plots on the phase+bandpass calibrator, 3c279, and on baselines that include antenna DV11, which we will make our reference antenna in just a bit.

First, we plot amplitude as a function of frequency for 3c279. We start by plotting the DATA column, set color to indicate the two correlations (i.e., the XX and YY polarizations), and ask plotms to iterate over baseline. Some important setting to get the right plot:

- By setting antenna to 'DV11&*' we select only baselines that include DV11.

[1] ALMA data sets undergo a series of quality assurance (QA) processes. The QA2 stage is an assurance that the observations have achieved the characteristics requested in the proposal. This document describes the QA2 process and the data products delivered to the users.

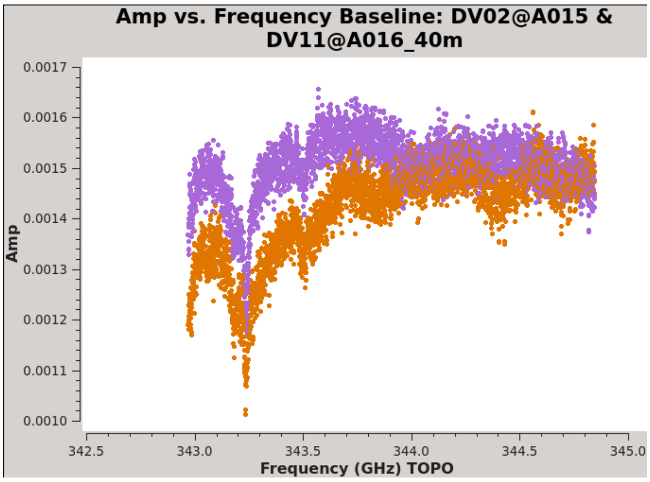


Figure 1.8: Example of Amplitude vs. Frequency before correction for the 3c 279.

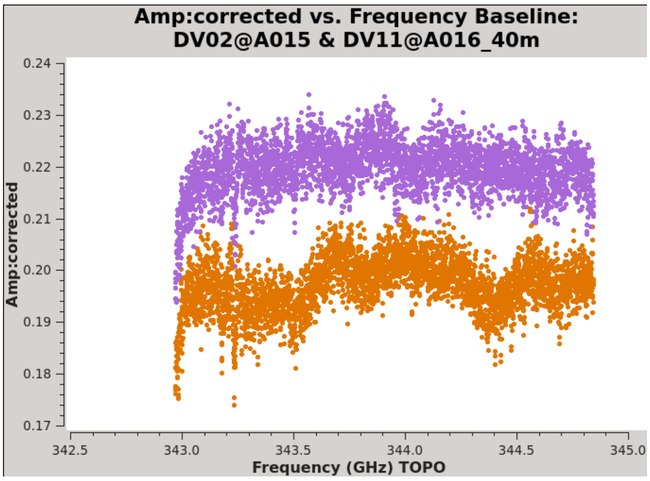


Figure 1.9: Same baseline as figure 1.8 but now after correction using WVR and Tsys calibrations.

- We ask plotms to average all data over a very long timescale, avgtime $= 1^8$ seconds $\sim$ 3 years or much longer than the time spanned by the whole data set.
- By setting avgscan = True we allow plotms to average across scan boundaries. The result is a plot of average amplitude per channel vs. frequency.

You can now make analogous calls to examine the phase vs. frequency, amplitude vs. time, and phase vs. time.

With the Tsys and WVR calibrations applied successfully and the a priori flagging taken care of we will now split out the corrected data using the command *split*. We will keep only the corrected data, specified via **datacolumn**, and only spectral window 1, which contains the FDM (high spectral resolution) observations of the CO(3-2) line. Setting **keepflags**=False tells split not to carry over any fully flagged rows from the original data set to the new MS. We give the new MS files the extension ".wvrtsys.ms" to indicate that they have been corrected for WVR and Tsys effects. Because split will not overwrite existing files, we remove any previous versions of the new MS before beginning.

The split process may take awhile. The WVR and Tsys-corrected data now sit in the DATA column of the new measurement sets, which have only one spectral window (now labeled spectral window 0 though it was spectral window 1 in the original data).

First we plot amplitude versus time (see image 1.10 and 1.11), averaging over all channels (by setting avgchannel to the very large value 10,000). We colorize by field so that scans on Titan are red, the bandpass and phase calibrator 3c279 is black (and orange in the Southern Mosaic where it has two

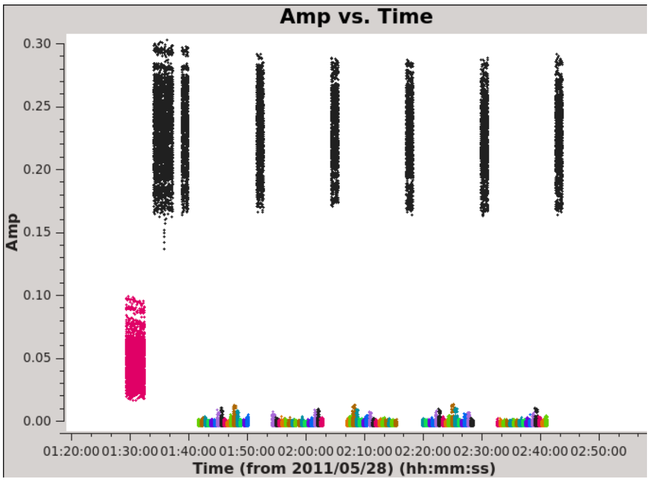


Figure 1.10: Example of amplitude vs. time for a northern mosaic data set.

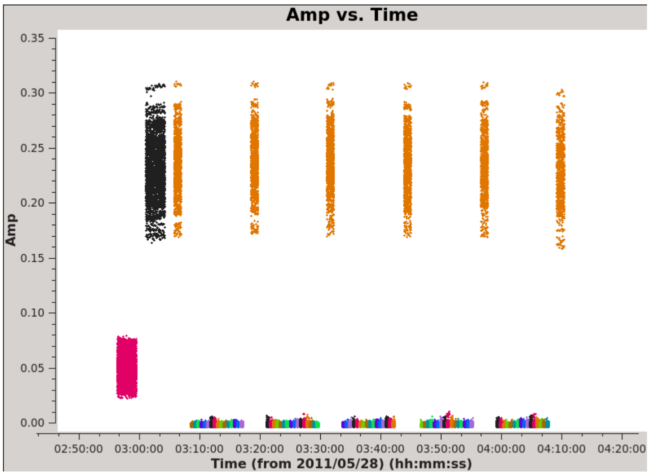


Figure 1.11: Example of amplitude vs. time for a southern mosaic data set.

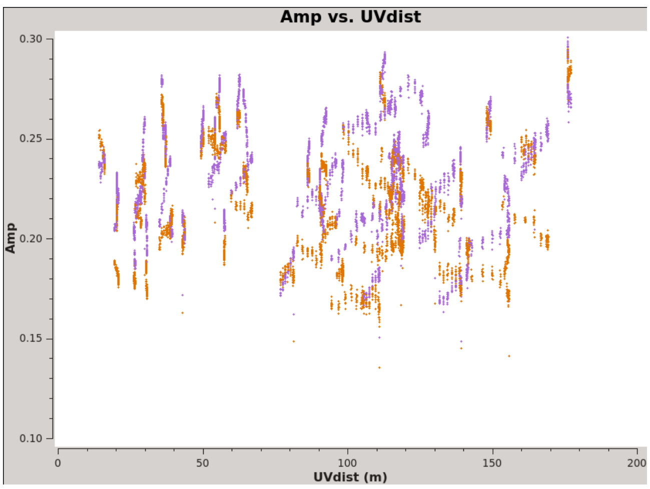


Figure 1.12: Example of amplitude vs. uv-distance for 3c279 in the first northern mosaic data set.

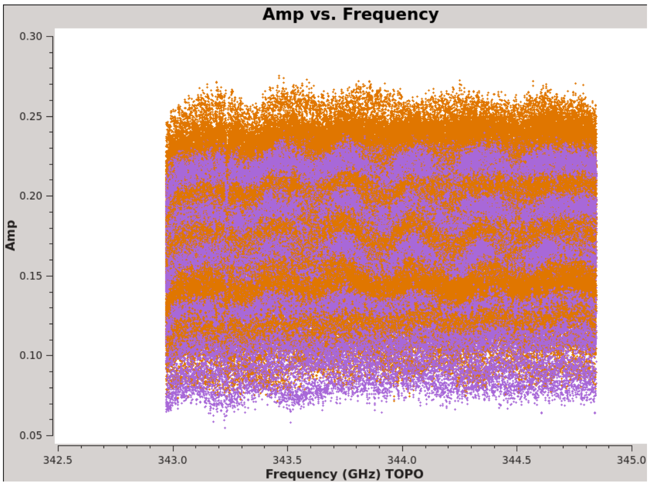


Figure 1.13: Example of amplitude vs. frequency for 3c279 in the second northern mosaic data set.

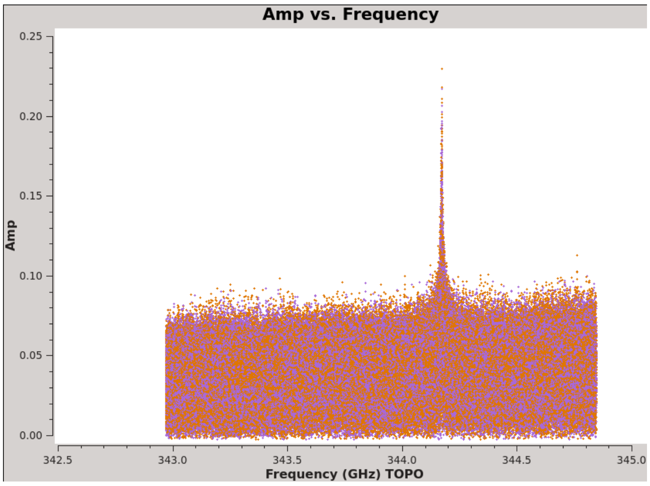


Figure 1.14: Example of amplitude vs. frequency for Titan in the second northern mosaic data set. Note the strong line (this is CO 3-2)!

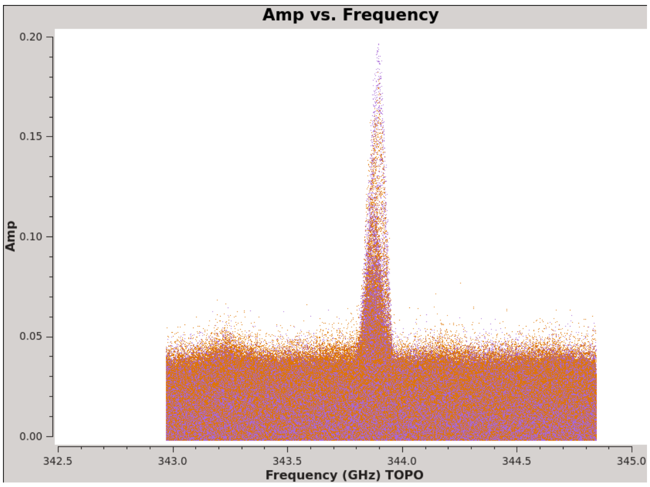


Figure 1.15: Example of amplitude vs. frequency for the Antennae in the second northern mosaic data set. The CO(3-2) line is visible.

field IDs), and the Antennae mosaic appears as a range of colors (one per pointing).

Here look for:

- **Missing data** - The source needs to be flanked by phase calibrator scans, if those are missing for any reason we need to flag the appropriate time range.
- **Dramatic outliers** - Does the source suddenly get very bright or the otherwise bright calibrator appear anomalously faint for a brief time? This likely indicates problematic data that should be identified and flagged. You can use the "select" (box with green plus along the bottom row in plotms) and "locate" (magnifying glass) buttons in plotms to isolate and identify problem data (it will print to the log).
- **Smooth variation with time** - A sudden jump may indicate a problem and often the safest approach is to flag data near a discontinuity.

There are two other very useful "averaging" plots worth making. First, we plot amplitude as a function of u-v distance (projected antenna separation). Discontinuities and spikes in this plot are often from non-astrophysical sources. In the phase analog to the plot, the effects of atmospheric decorrelation can be assessed from increased scatter at longer u-v distances. While using the moon Titan as our flux calibrator, we may want to watch for flaring amplitudes at short u-v distances. These may indicate that Saturn is contaminating our beam. For a perfect, bright point source, we expect flat amplitudes as a function of u-v distance at the source amplitudes. Figure 1.12 shows an example of this plot.

It can also be useful to examine the average amplitude as a function of frequency for each target. This allows one to check for lingering atmospheric effects, unexpected line emission or absorption in the calibrators, or decreased sensitivity due to "roll-off" of the telescope sensitivity at the band edges.

For these plots notice that we can see the CO(3-2) line in the Antennae even before calibration (see figure 1.15) and that Titan also shows evidence of a strong line (see 1.14) This will need to be flagged before we can use Titan to calibrate the flux scale of our data.

This suite of plots (along with the earlier inspection of the Tsys tables) gives us the tools we need to identify problematic data through the data sets. We use this to generate a set of inspection-driven flagdata commands for each data set. We apply these before the bandpass and gain calibration.

## 1.10 Apply Flagging

Based on this inspection and the other plots we have made, we now flag problematic portions of the data. As you reduce your own data it may be more efficient to group flags by data set and make use of the flagcmd command. As before, we may wish to reset our flags before beginning using the command *flagdata*. Remember that we dropped the flagged data when splitting out after the WVR and Tsys calibration, so this should not undo your "A Priori" flagging of shadowed data, autocorrelations, etc.

### 1.10.1 Edge Channels

ALMA's sensitivity decreases near the edge of the baseband and it is often useful to check for a 'roll-off' in sensitivity near the edge of the band. This will appear as a downturn in amplitude as a function of channel near the edge of the window in the uncalibrated data.

Because our FDM spw does not cover the full baseband, we do not see a strong roll off in our data (see Figure 1.16), where there is only a mild hint of a roll-off at the high end) but we do flag a (very) few channels at the high and low-frequency edge of the data set to be safe.

### 1.10.2 Problematic Tsys measurements

Above we noted issues with the Tsys measurements for both DV04 and DV12. We flag the affected data. Each of these issues should be visible in the Tsys plots you made above (e.g., see Figures 1.6

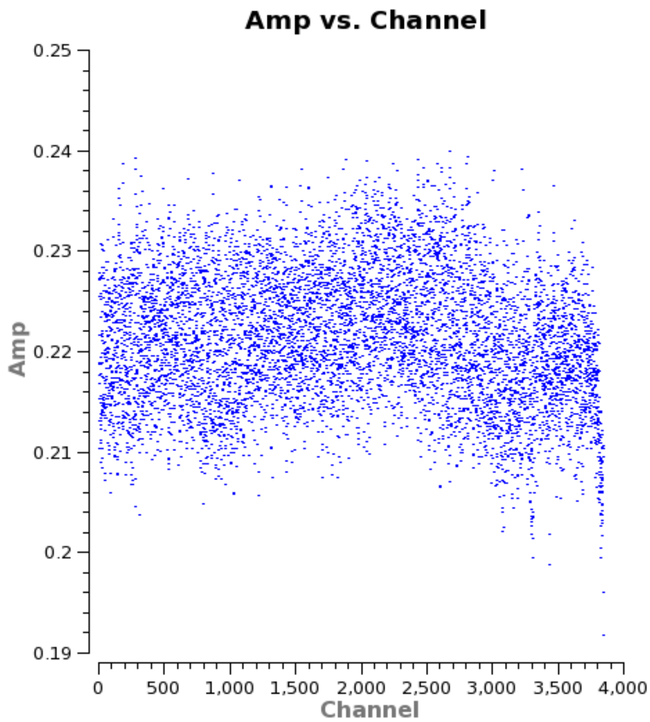


Figure 1.16: Amplitude vs. channel for one uncalibrated antenna pair. This kind of plot can be inspected to get an idea of the presence or magnitude of any roll-off in sensitivity near the edges of the spectral window.

and 1.7).

### 1.10.3 Unreliable Short-Spacing Measurements on Titan

Saturn may contaminate the short u-v spacings from Titan. In any case these often show significant scatter (Figure 1.17), so we flag them. There are still enough baselines to determine a good amplitude calibration for each antenna, so we only use the more extended baselines for flux calibration.

### 1.10.4 Delay Issues

DV13 and a few other antennas show signatures of an imperfect delay calibration. This is most easily identified via strong "wrapping" of phase as a function of frequency (see Figure 1.18). Such effects can be calibrated out with mild delay issues largely accounted for by the bandpass solution. The phase wrapping in DV13 seems weak enough that we will trust the calibrations to remove it.

### 1.10.5 Missing Phase Calibrator Observations

As a general rule, we want to be sure that observations of the phase calibrator (3c279) bracket each source observation. Two of the data sets do not include a final phase calibrator observation (see Figure 1.19) and for those two cases we flag the affected source observations.

### 1.10.6 Unexpected Scatter or Discontinuity in the Data

For several antennas we find sudden jumps in the phase of the phase calibrator as a function of time. These are visible in the plots of uncalibrated phase vs. time for single baselines above, and we show an example in Figure 1.20. It will not be possible to effectively interpolate the phase between measurements when we see these discontinuities. The safest approach is to flag the source data across these jumps.

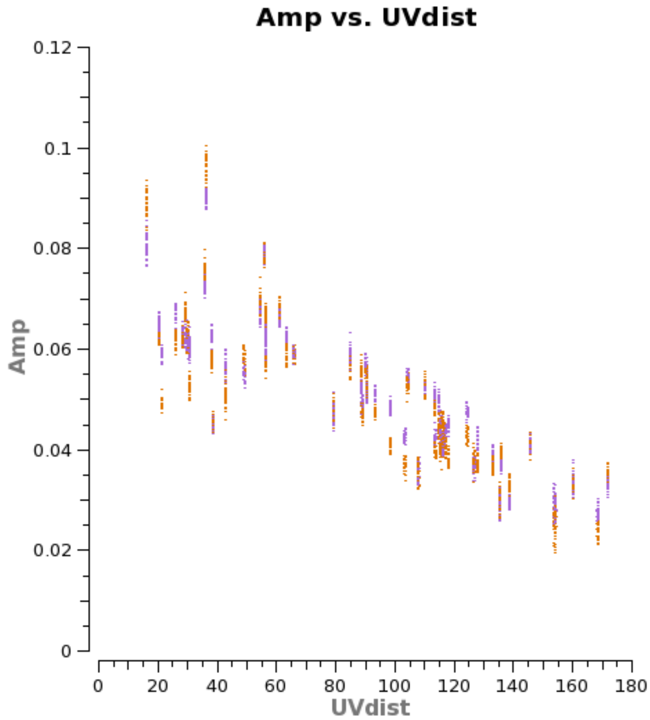


Figure 1.17: Observed amplitude vs. uv-distance for observations of Titan in the first data set. Note the scatter for low projected antenna separations. We will flag these short-spacing data, which may reflect contamination by Saturn, and use only the more extended baselines for flux calibration.

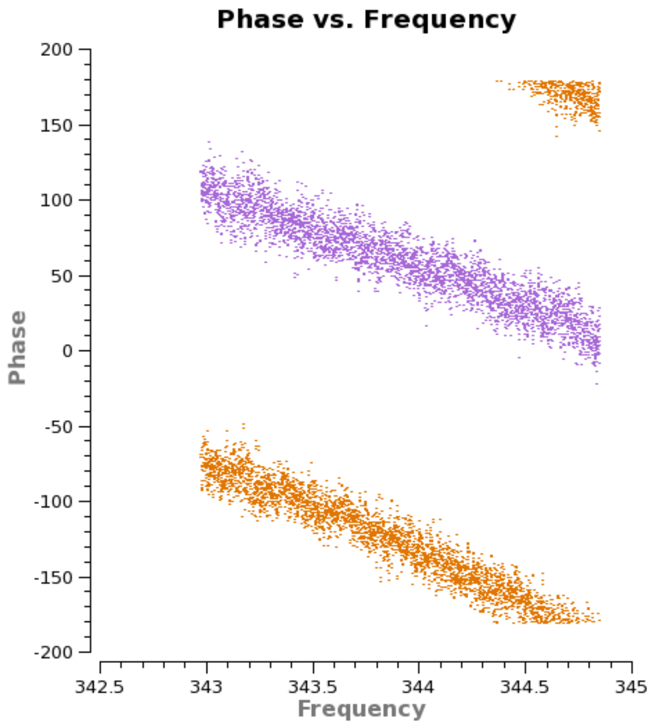


Figure 1.18: Phase vs. frequency for DV13 in a data set where this antenna shows evidence of imperfect delay calibration.

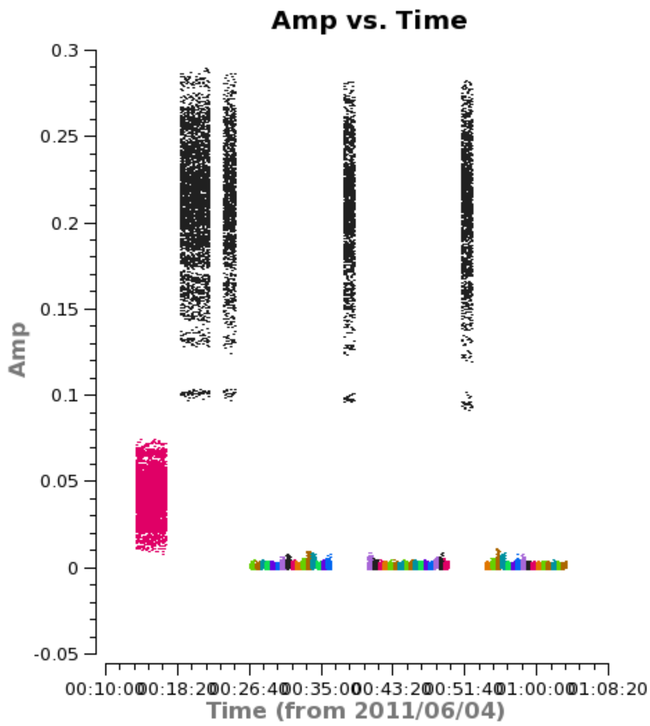


Figure 1.19: Amplitude vs. time colored by field for a data set where the final visit to the phase calibrator is missing. We will flag the last set of source data to ensure that each visit to the source is flanked in time by visits to the phase calibrator.

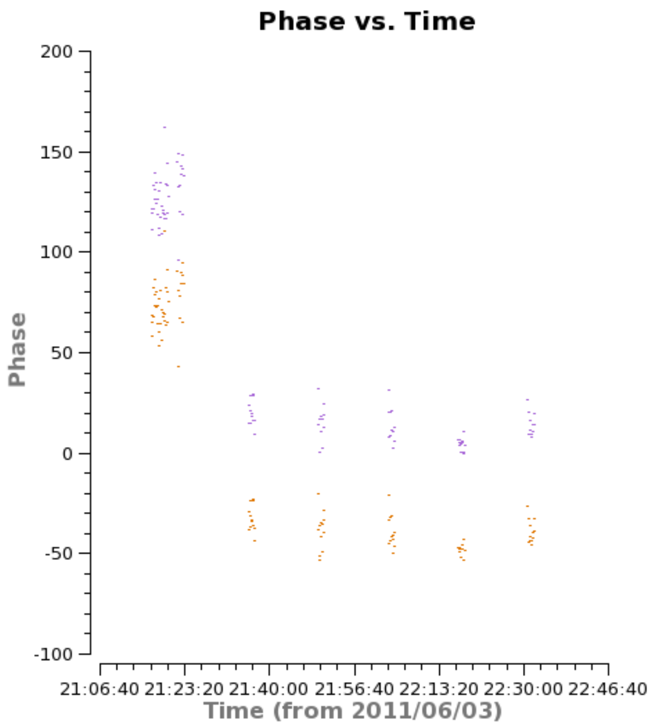


Figure 1.20: Phase vs. time for DV09 on a problematic day. It may prove problematic to calibrate the data near this discontinuity so we flag data near this time.

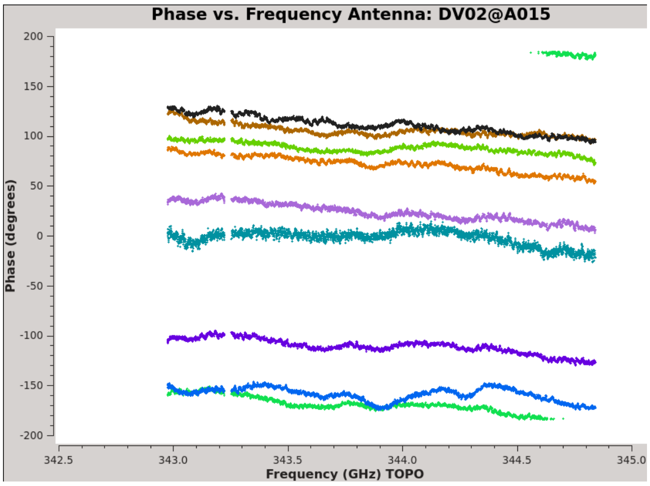


Figure 1.21: Phase vs. frequency for the bandpass calibrator, 3c279 in the first data set. We plot all baselines with DV02, averaged over time, and show only the 'XX' correlation.

### 1.10.7 Outliers Visible After Calibration

Often issues with the data may become evident after calibration. These data can appear as outliers in diagnostic plots for the calibrated data or even show up in the imaging stages. Once these are identified, best practice is to apply this new flagging then redo the calibration (if the issue is very minor, then re-calibrating may not be necessary).

## 1.11 Bandpass Calibration

We begin by calibrating the phase and amplitude response of each antenna as a function of frequency, called "bandpass calibration." We have already seen that the data contain smooth but systematic variations in both phase and amplitude as a function of frequency. We can see this again in a more compact form by plotting phase as a function of frequency for all baselines associated with each antenna (figure 1.21).

Each plot shows phase as a function of frequency for all baselines with one antenna for 3c279. We plot only the 'XX' correlation, colorizing by baseline. With **iteraxis** set to antenna the green arrows at the bottom of plotms will cycle through antennas. By using **avgscan** and a large **avgtime** we average all scans and integrations.

The phase (and amplitude) also varies as a function of time, as we saw before. Here are the similar plots for phase vs. time (see figure 1.22).

Figure 1.22 shows that the phase varies with time. We need to take this temporal variation into account when we solve for the frequency variations. Therefore we carry out the bandpass calibration in two steps. First, we use *gaincal* command to solve for the variation of phase as a function of time for 3c279 on very short timescales.

We solve, averaging together only a small fraction of the total bandpass (channels 1100 – 1300) to avoid the effects of the phase vs. frequency behavior. We will then apply this solution to remove time-dependent behavior when we solve for the frequency response of the antennas with *bandpass*. This task determines the amplitude and phase as a function of frequency for each spectral window containing more than one channel.

Now we use *bandpass* to solve for the frequency response of each antenna. To do this, we average all data in time by setting **solint** to 'inf' (that is, 'infinite'). We allow combination across scans and the different field IDs found for 3c279 by setting **combine** to "scan,field". We apply the phase vs. time calibration that we just derived on-the-fly using the parameter "gaintable".

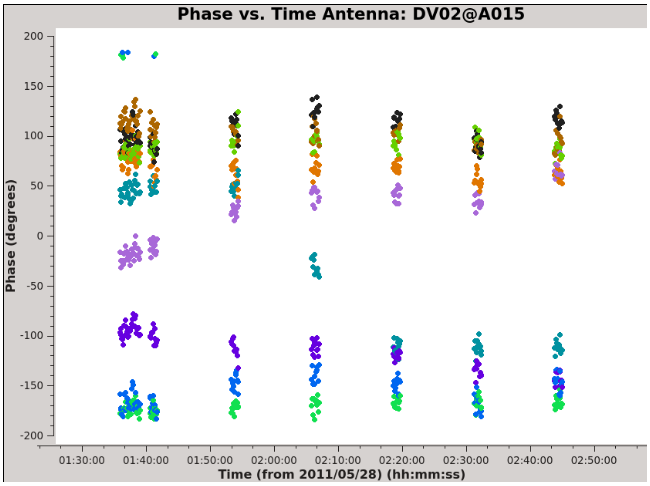


Figure 1.22: Phase vs. time for the bandpass calibrator, 3c279. Averaged over channel. Only baselines with antenna DV02, and corr='XX'

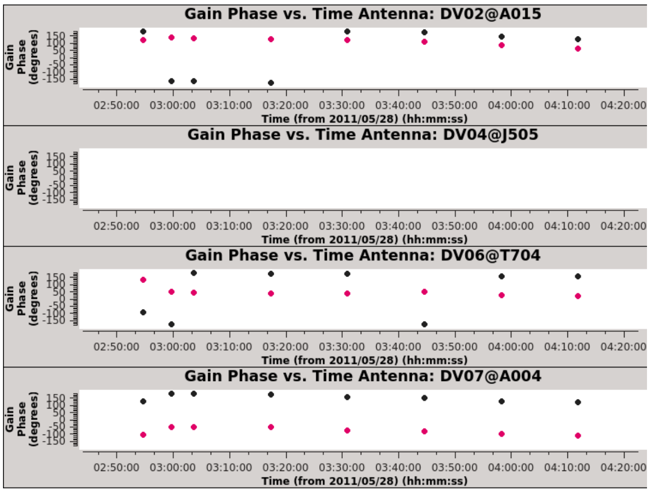


Figure 1.23: Short interval phase vs. time solution carried out and applied during bandpass calibration, here for the first set of antennas for the first data set. This solution removes any systematic variations in phase vs. time.

Do not worry about the message "Insufficient unflagged antennas" when running the bandpass task. This indicates that *bandpass* is failing on the flagged edge channels, which is expected.

It is now a good idea to plot both sets of solutions to look for irregularities, especially:

- discontinuities in the phase vs. time solution;
- rapid wrapping of phase in either phase vs. time or bandpass solution;
- large roll-off in the amplitude response near the band edge in the bandpass solution;
- large scatter in any solution.

## 1.12 Absolute Flux Calibration

The bandpass calibration will account for the phase and amplitude response of our antennas as a function of frequency. We now solve for the absolute flux scale of the data by referencing to Titan and in the next section we will calibrate the phase and amplitude behavior of the antennas as a function of time.

Before using Titan to set the flux, there is an important systematic to account for. When we looked at

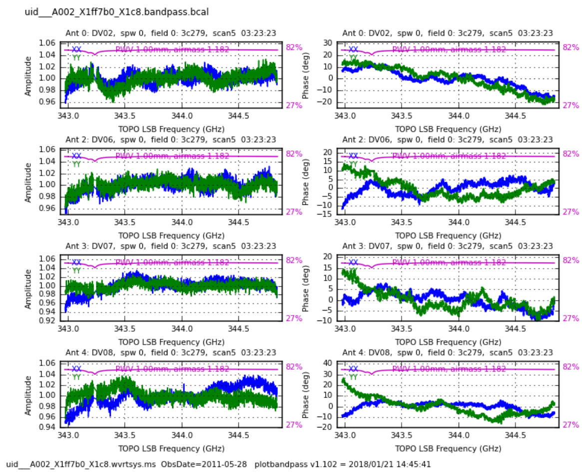


Figure 1.24: Bandpass amplitude solution for the first set of antennas and the first data set.

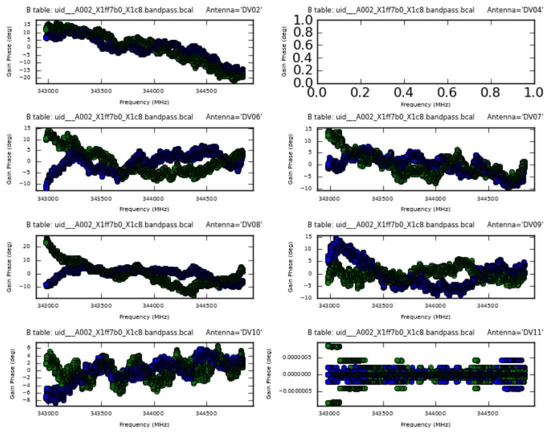


Figure 1.25: Phase vs. frequency calibration from the bandpass calibration for the first set of antennas and the first data set.

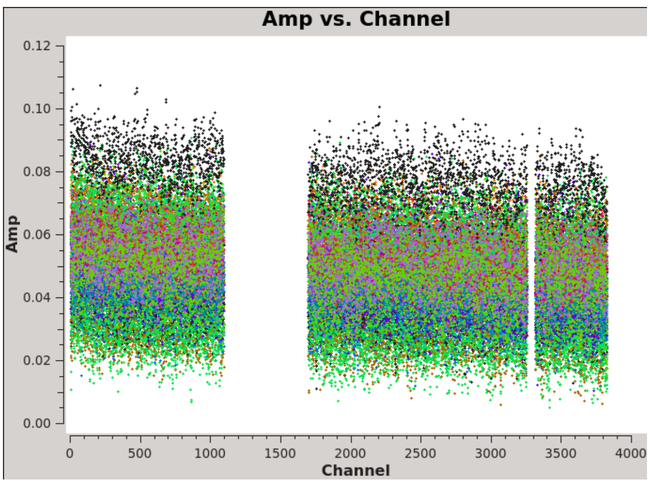


Figure 1.26: Uncalibrated amplitude vs. channel plot for the flux calibrator, Titan. Averaged over time, corr='XX', and colorized by baseline.

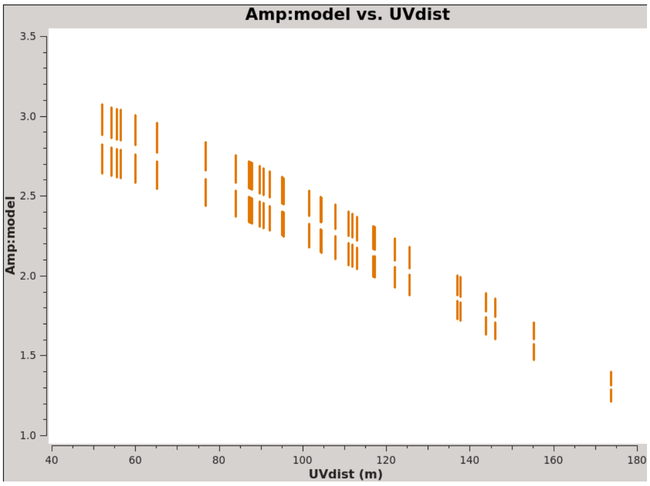


Figure 1.27: Tital model.

the integrated spectra of our targets above, remember that Titan showed a whopping spectral line, in fact the same CO(3-2) line that we wish to observe in the Antennae. We will set the flux of Titan (and thus all of our data) by referencing to a model in casa that does not account for this line. Therefore we need to flag the part of the Titan observations contaminated by the line before we calibrate.

Next, we will use the *setjy* task to read the predicted complex visibilities for Titan into the MODEL column of each data set. Then we take a look at the Titan model for one dataset in figure 1.27.

## 1.13 Gain, Phase and Amplitude Calibration

With the flux now properly scaled, we will calibrate the phase and amplitude behavior of the antennas as a function of time.

We begin by running a short-solution interval *gaincal* to solve for phase variation on short timescales during observations of our two calibrators, with **solint** set to "int". By applying this on-the-fly, we can remove any decorrelation in the data due to phase scatter when we solve for the amplitude calibration.

Now we derive the longer timescale phase calibration table using **solint** set to "inf", but not allowing scan combination. This calibration has higher signal to noise due to combining more data, and for the purposes of correcting the source, it is just as precise as the short timescale solution. Then we apply the short-timescale phase solution and carry out a scan length (**solint** set to "inf") calibration of the data using **calmode** of 'ap'.

This "amp.cal" solution gives us the amplitude variations as a function of time, but they are not yet pinned to a realistic scale except in the case of Titan, where we have solved using the model input by setjy. We will set the flux of our secondary calibrator 3c279 with reference to Titan using *fluxscale*.

This new correctly-scaled flux table ".flux.cal" replaces the previous ".amp.cal" table as the correct amplitude calibration table to apply to the data, i.e., the ".flux.cal" contains both the time variability of the amplitude solved for in ".amp.cal" and the correct flux scaling set with fluxscale.

Now we plot the final phase and amplitude calibration tables for each data set. A well-behaved calibration table will show smooth variations as a function of time. Sudden jumps or wild variations among the antenna amplitude gains should prompt further investigation and possibly additional flagging.

## 1.14 Apply the Calibrations and Inspect

Now we will use *applycal* to apply the bandpass, phase, and amplitude calibration tables that we generated in the previous sections to the data. We apply the solutions separately to the bandpass and secondary ("phase") calibrator 3c279, the flux calibrator Titan, and the target source. In most

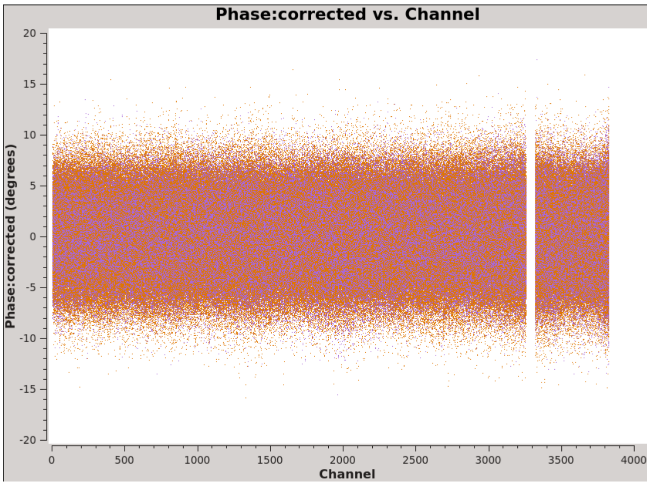


Figure 1.28: Calibrated phase vs. channel plot for 3c279.

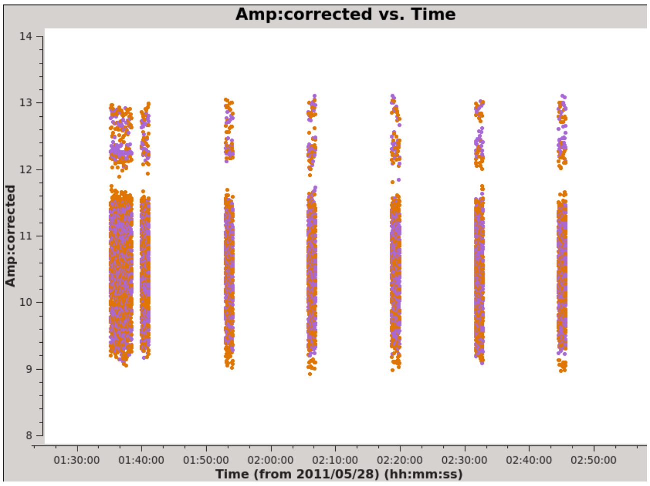


Figure 1.29: Calibrated amplitude vs. time plot for 3c279.

data sets, the bandpass and secondary calibrator will not be the same and this step would include one additional applycal.

Once calibrations are applied, it is important to go back and inspect the calibrated data. New problematic antennas or baselines may be visible after calibration. Repeat the steps above, focusing on the CORRECTED data column. Bear in mind that for any point source calibrators we now expect to find phase scattering around zero and to find flat amplitudes as a function of u-v distance. Look for outliers and other signatures of problematic data. As a general rule, you will want to incorporate these data into your overall flagging script then rerun the whole calibration process, so that reduction is iterative. If the data only represent a minor problem, however, it may not be terribly harmful to flag them after the fact so that they do not interfere with imaging but trust that the calibrations are mostly unaffected.

As an example of this inspection, we cycle through the corrected amplitudes and phases of 3c279 and Titan as a function u-v distance, to check that the phases are close to zero and the amplitudes are constant.

In figure 1.28 and 1.29 we plot phase vs. channel and amp vs. time for 3c279 for the uid___A002_X1ff7b0_Xb dataset.

Finally we can use plotms to examine the corrected amplitude and phase of Antennae galaxies as a function of time and uv-distance.

## 1.15 Split and Concatenate Data for Northern and Southern Mosaics

The individual data sets are now calibrated. We can safely split out the calibrated data for our science target and drop the calibrators. As we do so, we will smooth the data in frequency, averaging together groups of 23 channels by setting **width=23** in split. The new data will have a channel width corresponding to about $\sim 10\ km/s$, very similar to the SMA data being verified. The factor of $> 20$ drop in data volume will also make the imaging steps much more tractable.

For convenience we concatenate all data for the Northern Mosaic into a single big MS and place all data for the Southern Mosaic into another file. To do this, we construct a list that holds the names of all the Southern Mosaic MS files and another that holds the name of all the Northern Mosaic MS files then feed these into the *concat* task.

# Chapter 2

# Imaging steps

## 2.1 Imaging Mosaics

Mosaics like other kinds of images are created in the CASA task *tclean*. This task handles continuum images and spectral line cubes, full Stokes polarization imaging, supports outlier fields, contains point-source CLEAN based algorithms as well as options for multi-scale and wideband image reconstruction, widefield imaging correcting for the w-term, full primary-beam imaging and joint mosaic imaging (with heterogeneous array support for ALMA).

To invoke mosaic mode, you simply set the parameter **gridder='mosaic'**. This is a joint deconvolution algorithm that works in the uv-plane. A convolution of the primary beam patterns for each pointing in the mosaic is created: the primary beam response function. The corresponding image of the mosaic response function will be called *¡imagename¿.pb*.

Additionally, for mosaics it is essential to pick the center of the region to be imaged explicitly using the **phasecenter** parameter. Otherwise it will default to the first pointing included in the **field parameter**, since this is often at one corner of the mosaic, the image will not be centered. For the Northern mosaic, the center pointing corresponds to field id 12. Note that during the final split in the calibration section that selected only the Antennae fields, the field ids were renumbered, so that the original centers have changed: field id 14 becomes 12 for the Northern mosaic and field 18 becomes 15 for the Southern mosaic. You can also set an explicit coordinate.

## 2.2 Continuum Imaging

We will make 345 $GHz$ continuum images for the two regions covered by the mosaics. We use the task *clean* over the channels that are free of the line emission; we avoid the edge channels which tend to be noisier due to bandpass rolloff effects. The line-free channels are found by plotting the average spectrum (all fields). We find the CO(3-2) line from channels 50 to 100 in the southern mosaic (figure 2.1), and from 70 to 100 in the northern mosaic (figure 2.2).

Then the *avgtime* is set to a large value so that it averages over all the integrations, and *avgscan* is set to allow averaging of the different scans.

Next we create continuum images from the line-free channels.

### 2.2.1 Northern Continuum Mosaic

For illustrative purposes we first make a dirty image to see if there is emission and what the exact beam size is. It should be on the order of 1" but this will vary a bit according to the uv-coverage in the actual data. We will start with a **cell** size of 0.2" to oversample the beam by a factor of 5. The **imsize** needs to be large enough given the cell size to comfortably encompass the mosaic. From

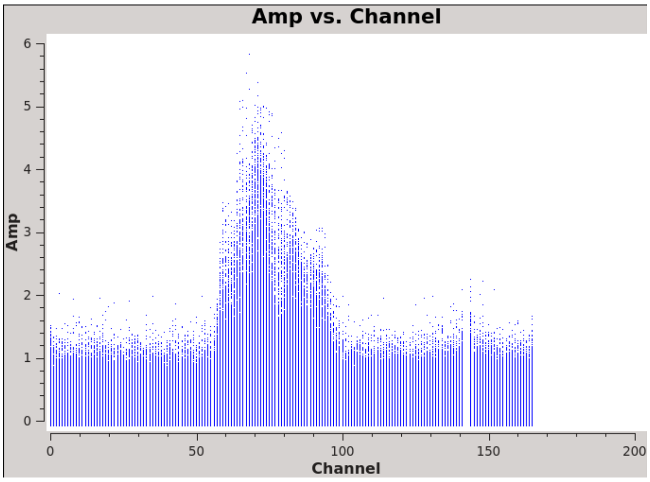


Figure 2.1: Southern mosaic: Amplitude vs. channel. The CO(3-2) line is seen from 50 to 100.

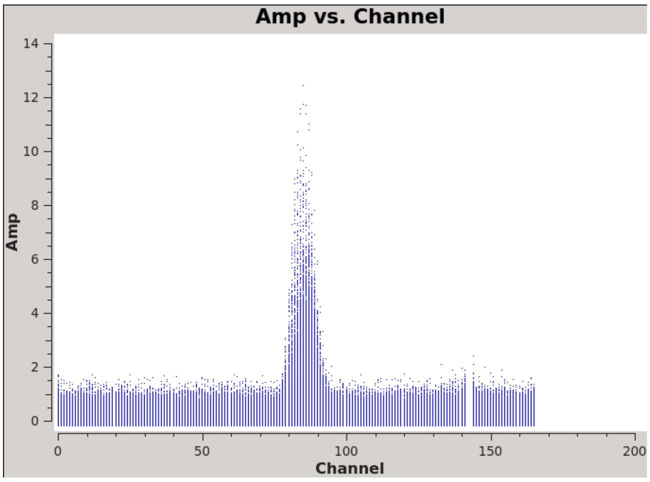


Figure 2.2: Northern mosaic: Amplitude vs. channel. The CO(3-2) line is seen from 70 to 100.

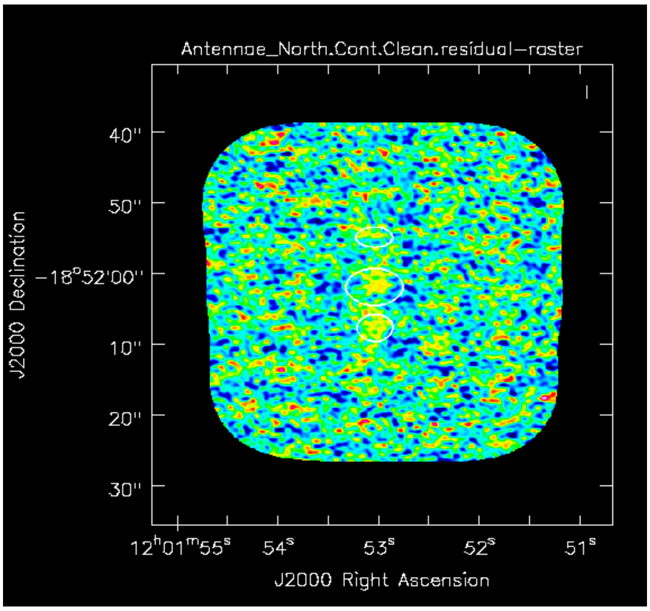


Figure 2.3: Residual for Northern continuum mosaic after 100 iterations; the clean mask is shown by the white contour.

the mosaic footprints shown in the overview, we can see that the Northern mosaic imsize needs to be about 1 arcmin. With 0.2" pixels, this requires **imsize=300**.

The reported beam size is about 1.23" × 0.72", with a position angle (P.A.) of 85.4 degrees.

There is definitely a detection in the vicinity of the Northern nucleus (see figure 5 and figure 6). Using the square region icon, and drawing a box near but not including the emission, we find the rms noise is about 0.5 mJy/beam in the dirty image.

Next we switch to refined values of **cell=**$'0.13\ arcsec'$ and **imsize=500** based on the observed beam size. We also switch to interactive mode so that you can create a clean mask using the polygon tool.

The residuals are "noise-like" after only $\sim$ 30 iterations (see figure 2.3), so hit the red X symbol in the interactive window to stop cleaning here.

### 2.2.2 Southern Continuum Mosaic

For the southern mosaic we modify the **phasecenter**, mosaic **imsize**, and the line-free channels (**spw**) to be consistent for this mosaic. We also bypass the dirty image step we did above for the Northern mosaic. We expect the beam to be about the same and use the same **cell** size.

The beam size reported in the logger for the Southern mosaic: 1.13" ×0.67", and P.A.= 61 *deg*; which is a little better than the beam for the North owing to better uv-coverage.

### 2.2.3 Image Statistics

You can determine statistics for the images using the task *imstat*. This task displays statistical information from an image or image region.

From this we find, that for the Northern continuum image the peak is 3.86 $mJy/beam$ and the rms is 0.49 $mJy/beam$. For the Southern continuum we find a peak of 4.70 $mJy/beam$ and an rms of 0.47 $mJy/beam$.

However, the calculation of the rms comes with a couple of caveats. First, the mosaic primary beam response rolls off toward the edges of the mosaic, as do correspondingly the flux density and rms. Thus if you don't restrict the measurement to areas of full sensitivity, the apparent rms is skewed downward. Second, since there is real emission in the image, the rms will be skewed upward (with the

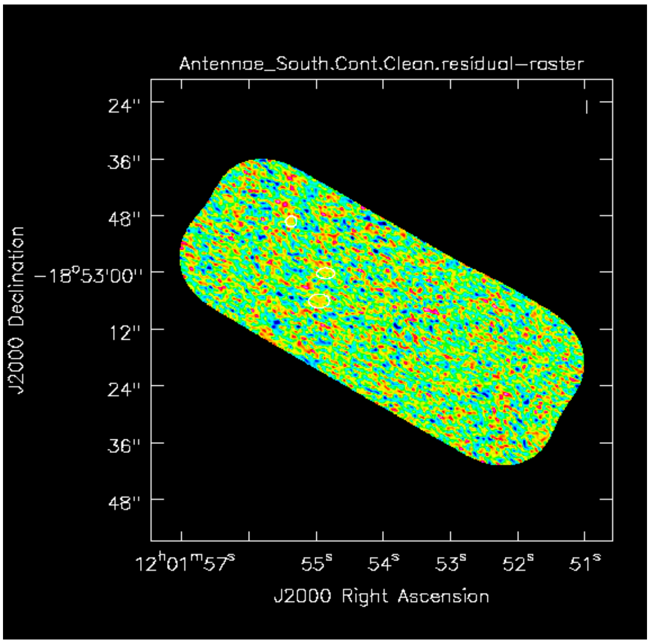


Figure 2.4: Residual for Southern continuum mosaic after 100 iterations; the clean mask is shown by the white contour.

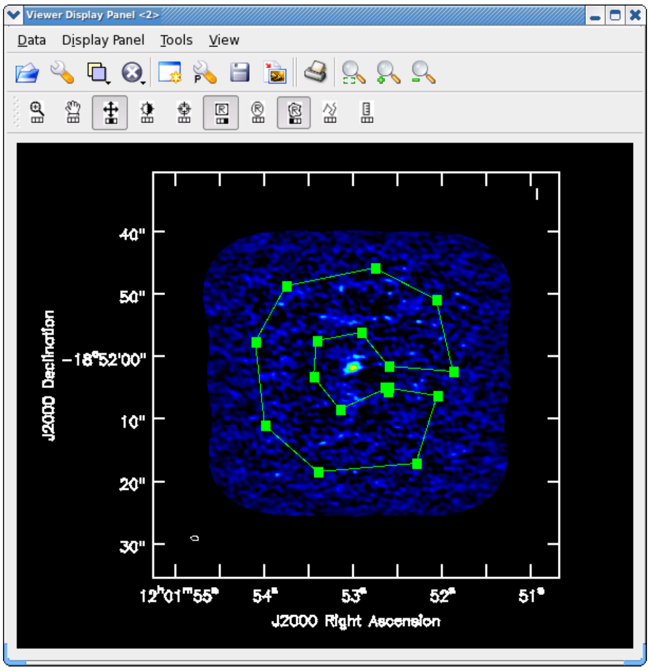


Figure 2.5: Example polygon for rms determination using the viewer.

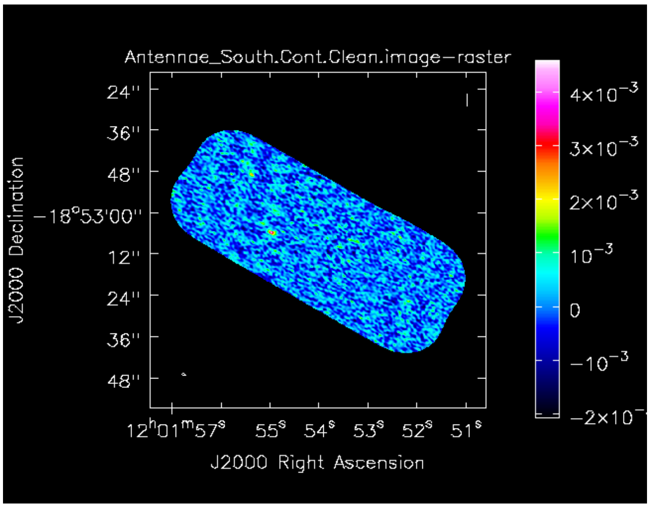


Figure 2.6: 345 $GHz$ continuum image of the northern mosaic.

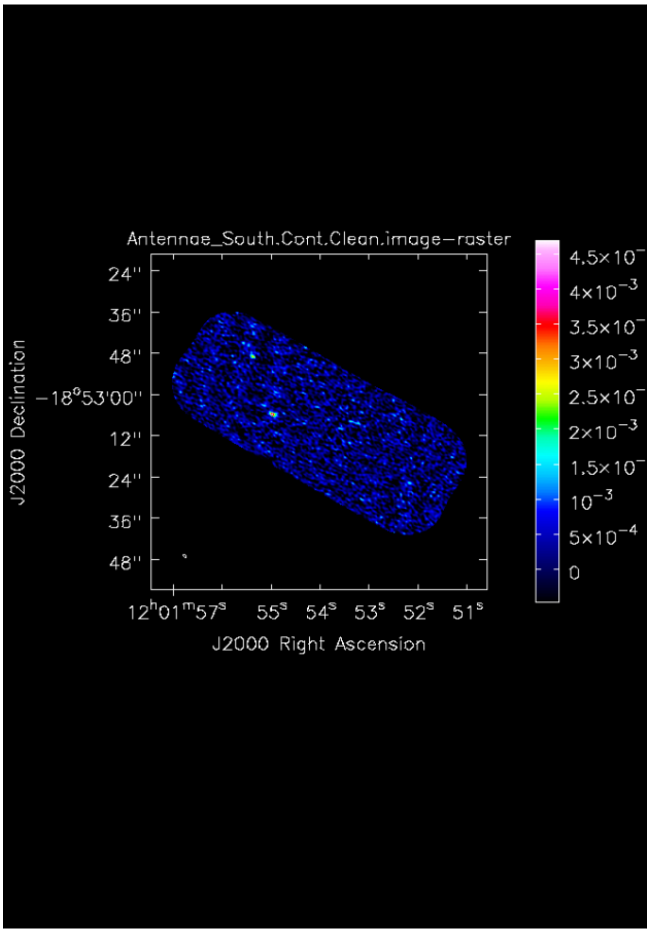


Figure 2.7: 345 $GHz$ continuum image of the southern mosaic.

error increasing with brighter emission). Both can be solved by picking boxes that exclude the edges of the mosaic and the real emission.

How does this compare to theory? You can find out using the ALMA sensitivity calculator. The continuum bandwidth of the Northern mosaic is about 1 $GHz$ and about 0.85 $GHz$ for the Southern mosaic. The number of antennas is about 12 and the time on a single pointing is about 300 $s$. This yields an expected rms of about 1 $mJy/beam$. However, this needs to be decreased by about a factor of 2.5 near the center of the mosaic due to the hexagonal Nyquist sampling of the mosaic for a theoretical rms of about 0.4 $mJy/beam$, in good agreement with observation.

## 2.3 Continuum subtraction

In these data, the continuum emission is too weak to contaminate the line emission (i.e. the peak continuum emission is less than the rms noise in the spectral line channels). Nevertheless, for illustrative purposes we demonstrate how to subtract the continuum emission in the uv-domain using the task *uvcontsub*.

This task estimates the continuum emission by fitting polynomials to the real and imaginary parts of the spectral windows and channels selected by fitspw. This fit represents a model of the continuum in all channels. The fitted continuum spectrum is subtracted from all channels selected in spw, and the result (presumably only line emission) is stored in a new MS that is always called vis + ".contsub".

Here, **fitspw** gives the line-free channels for each mosaic and **fitorder=1**. Higher order fits are not recommended. If you do not have line-free channels on both sides of the line **fitorder=0** is recommended.

## 2.4 CO(3-2) Imaging

Now we are ready to make cubes of the line emission. The imaging parameters are similar to the continuum except for those dealing with the spectral setup: **especmode**, **start**, **width**, **nchan**, **restfreq**, and **outframe** parameters. When making spectral images you have three choices for the **specmode** parameter: **mfs**, **cube**, and **cubedata**. As noted above, setting **specmode='mfs'** gives an output image with only one channel, and is used for continuum imaging. The other **specmode** options create output data cubes with one or more channels.

Data are taken using constant frequency channels. For spectral line analysis it's often more useful to have constant velocity channels, and this is also the best way to make images of multiple lines with the exact same channelization for later comparison.

It is important to note that ALMA does not do on-line Doppler Tracking and the native frame of the data is TOPO, a temporale frame. If you do not specify **outframe** the output cube will also be in TOPO, which is not very useful for scientific analysis. The Doppler Shift is taken out during the regridding to the desired outframe in *tclean* or alternatively it can be done separately by the *cvel* task which would need to be run before *tclean*.

### 2.4.1 Northern Mosaic

As before, it is very important that you make a clean mask. There are many ways to do this ranging from the complicated to simple. For this example we provide a single clean mask that encompasses the line emission in every channel and apply it to all channels. This is much better than no clean mask, though not quite as good as making an individual mask for each channel. You can choose to use and edit the provided mask or create your own mask.

Figure 10 shows what the final clean box should look like. Cycle through the channels until you reach $\sim 1650$ $km/s$, where you can more easily see where to draw the clean box around the CO emission. Be sure the check the circle next to 'All Channels' when making your clean mask. Now clean using

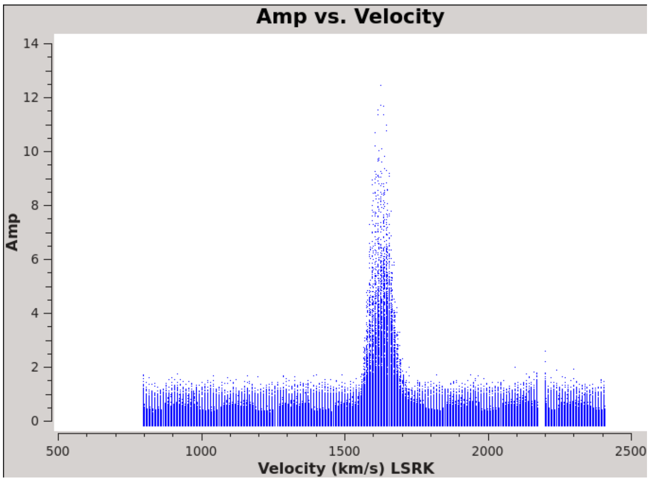


Figure 2.8: Northern CO(3-2) uv-spectrum in LSRK velocity space.

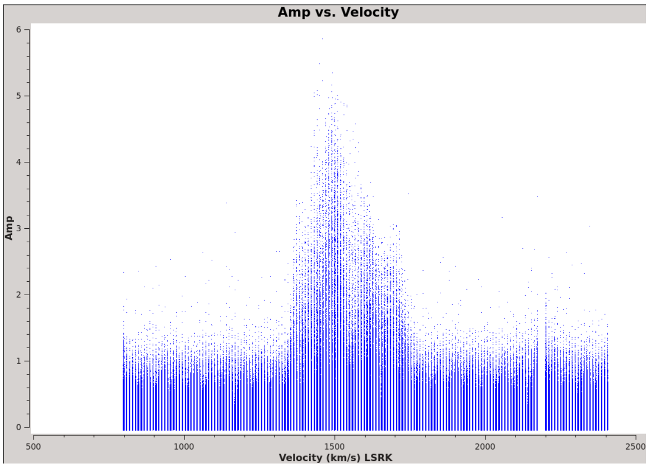


Figure 2.9: Southern CO(3-2) uv-spectrum in LSRK velocity space.

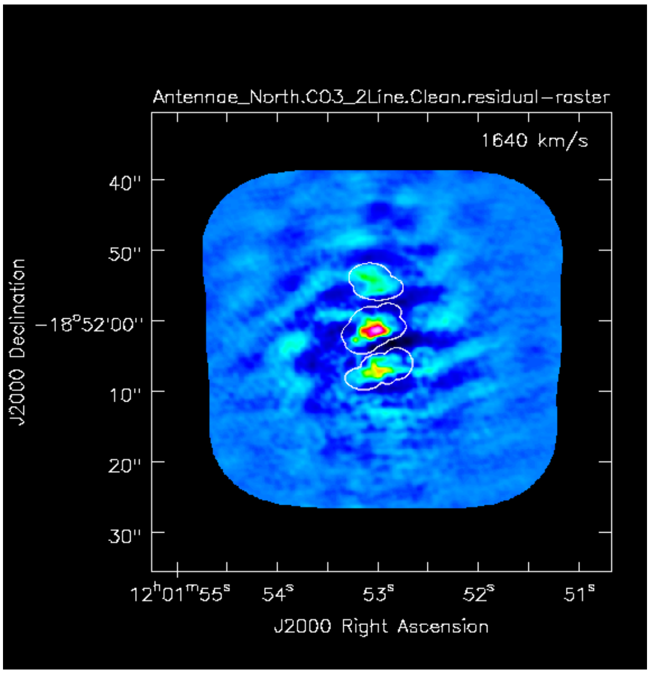


Figure 2.10: Interactive cleaning of northern mosaic. The clean mask is shown in white. In this example we used the same mask for all channels by selecting "all channels" before drawing mask.

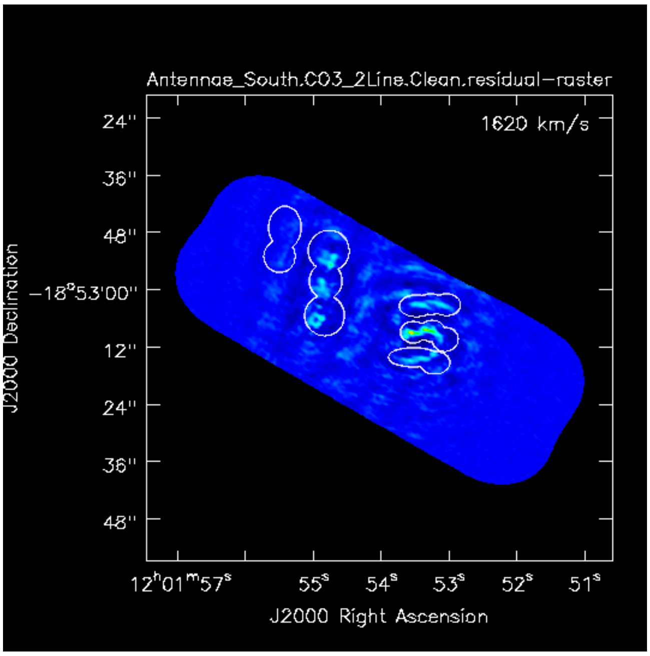


Figure 2.11: Interactive cleaning of southern mosaic. The clean mask is shown in white. In this example we used the same mask for all channels by selecting "all channels" before drawing mask.

the green circle arrow. You can watch the progress in the logger. When the first ∼ 650 iterations are done, the viewer will show the residual map for each channel.

These data are pretty severely dynamic range limited, in part due to the sparse uv coverage. In other words, the noise in bright channels is set by a maximum signal-to-noise. This effectively prevents us from stopping clean based on an rms based threshold because the effective rms changes as a function of the brightest signal in each channel. Thus, in this case we need to stop clean interactively.

Stop cleaning after about ∼ 7000 iterations (Red X), when the artifacts start to look as bright as the residual.

### 2.4.2 Southern Mosaic

Repeat process for Southern mosaic.

For each mosaic use the viewer polygon tool as described above for the continuum, to find the image statistics in both a line-free channel and the channel with the strongest emission.

The line-free channel rms for the northern and southern mosaics are about 4.0 $mJy/beam$ and 3.5 $mJy/beam$, respectively. However in a bright channel this degrades to about 18 $mJy/beam$ and 9 $mJy/beam$, respectively. This is the aforementioned dynamic range limit.

## 2.5 Self Calibration

Next we attempt to self-calibrate the uv-data. The process of self-calibration tries to adjust the data to better match the model image that you give it. When *tclean* is run, it saves a uv-model of the resulting image in the MODEL column of the measurement set if savemodel = modelcolumn. This model can be used to self-calibrate the data using *gaincal*. That is why it is important to make the model as good as you can by making clean masks.

Mosaics can be tricky to self-calibrate. The reason is that typically only a few of the pointings may have strong emission. The stronger the signal, the stronger the signal-to-noise will be for a given self-calibration solution interval (**solint**). This means that typically only a few fields are strong enough for self-calibration, though it can still bring about overall improvement to apply these solutions to all fields.

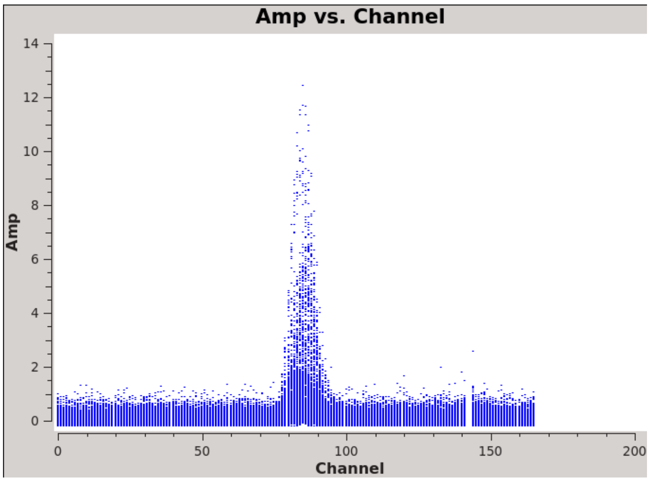


Figure 2.12: Plot of the CO(3-2) emission from the data column of field 12 of the Northern mosaic.

The fine tuning of each field in the mosaic over small solutions intervals is often impossible. Indeed fine-tuning only a few fields in the mosaic on small time scales can result in a mosaic with dramatically different noise levels as a function of position. Below, we will simply attempt to self-calibrate on the strongest field in the mosaic, obtaining one average solution per original input dataset.

Here, the continuum is far too weak so we will use the CO(3-2) line emission. To increase signal-to-noise it is often helpful to limit solutions to the range of channels containing the strongest line emission.

We will also use **gaintype='T'** to tell CASA to average the polarizations before deriving solutions which gains us a sqrt(2) in sensitivity.

### 2.5.1 Northern Mosaic

The strongest emission in the Northern mosaic comes from the center pointing, already identified above as **field='12'**. We make a uv-plot of the spectral line emission from this field to chose only the strongest channels to include in the self-calibration. It is essential that the self-calibration channel range be chosen based on the UV data and not on the image cube. This is because the channels in the image cube were modified by the velocity/channel specifications in the first invocation of clean. This plot shows the channel range of the uv data.

We also want to check the channel amplitudes in the model column of the measurement set. The depth of the initial clean will determine what model channels have emission. Self calibration will attempt to make the data look like the model. Therefore, it is important that we select a channel range where the model is realistic.

Zooming in on each dataset we see that they are about 1 hour long, and that each dataset has 2 or 3 observations of field 12. Each time, it's observed for about 25 seconds. We set **solint='3600s'** to get one solution per dataset.

Apply the solutions to all fields and channels with *applycal*, which will overwrite the corrected data column. Then re-image the data with the selfcal applied.

After 3000 iterations or so, inspect each channel to see if there is more emission that needs to be included in the mask. Remember to select the "all channels" toggle. Stop after 9000 or so iterations.

What you should notice is that the peak flux density has increased substantially while the rms noise is about the same. This is reasonable for this self-calibration case because we have mostly adjusted relative position offsets between the datasets and not taken out any short-term phase variations which would reduce the rms.

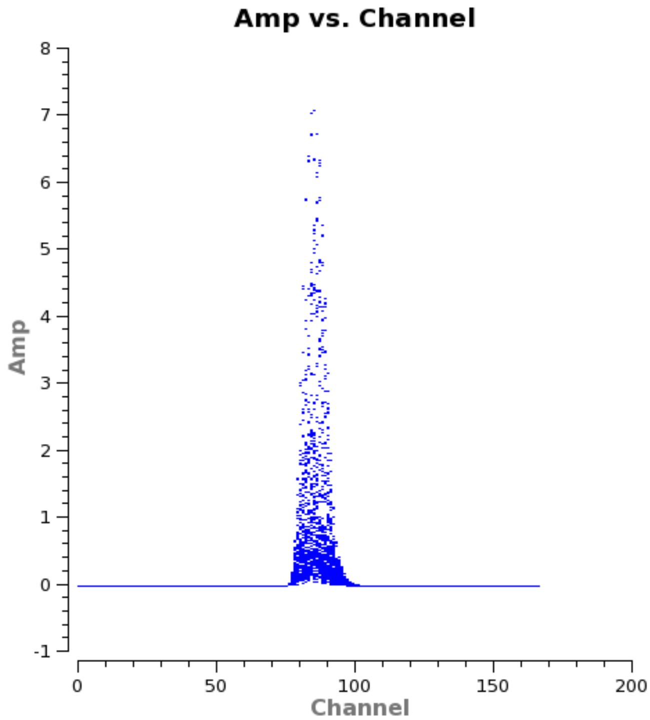


Figure 2.13: Plot of the CO(3-2) emission from the data column of field 12 of the Northern mosaic.

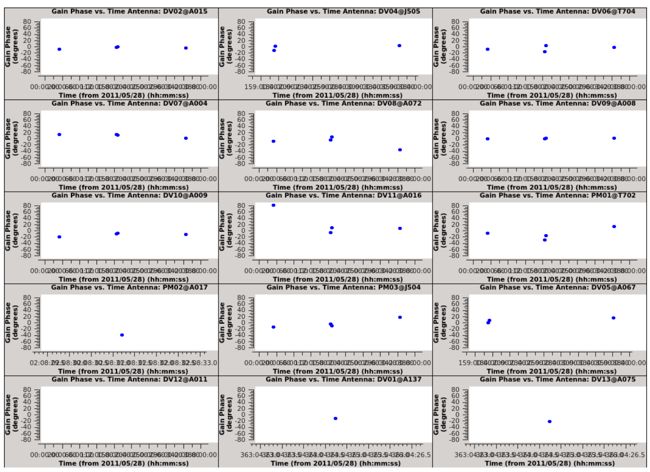


Figure 2.14: Phase-only self-calibration solutions for the Northern mosaic.

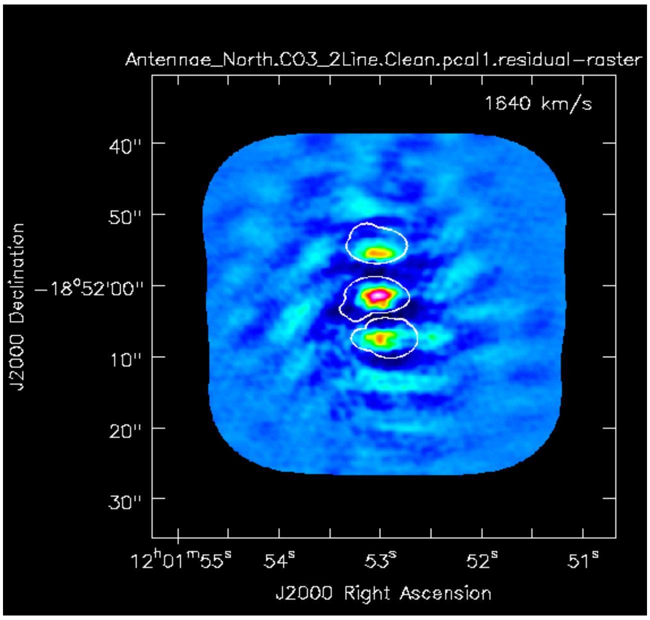


Figure 2.15: Interactive cleaning of Northern mosaic after self-cal. The clean mask is shown in white. In this example we used the same mask for all channels by selecting "all channels" before drawing mask.

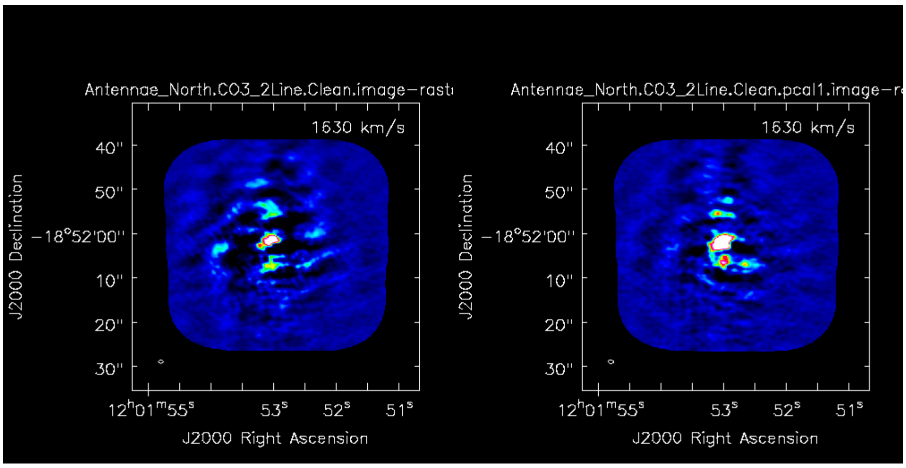


Figure 2.16: Comparison of channel 43 before selfcal (left) and after selfcal (right).

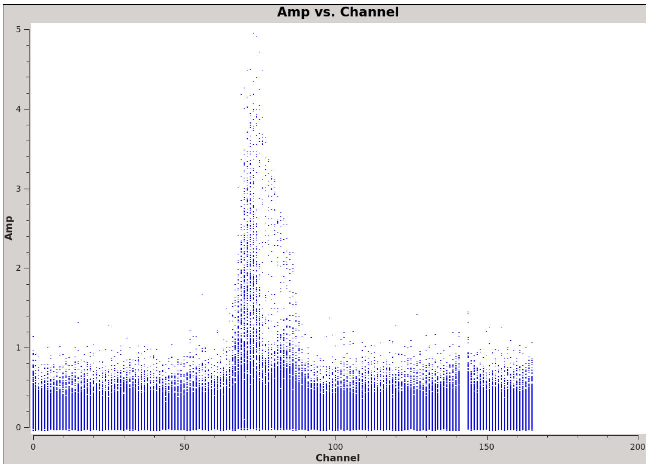


Figure 2.17: Plot of the CO(3-2) emission from the data column for Field 7 of the Southern mosaic.

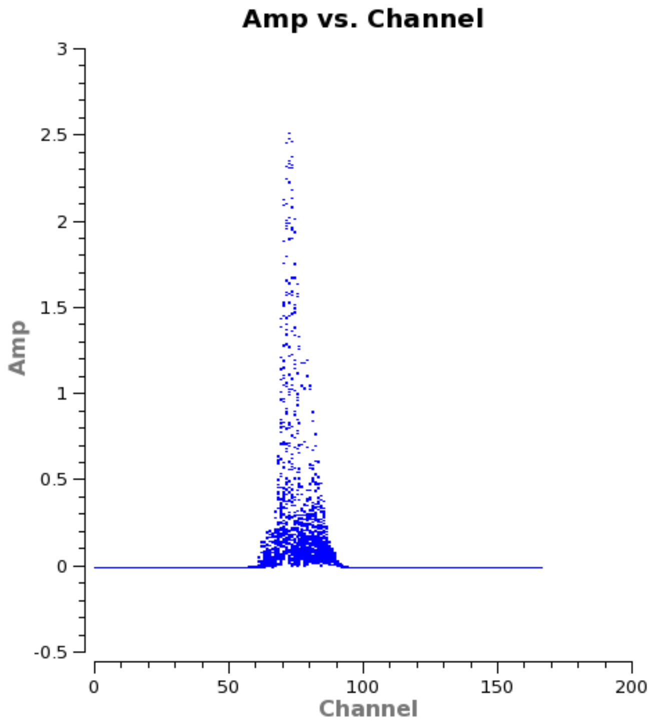


Figure 2.18: Plot of the CO(3-2) emission from the data column for Field 7 of the Southern mosaic.

### 2.5.2 Southern Mosaic

Now repeat the self-calibration process for the Southern mosaic. It is a bit tougher in this case to pick the best field. The most compact bright emission is toward field id 11 in the Overview. Adjusting for the renumbering of field ids after the final calibration split, this becomes **field='7'**.

From the following plots we can assess the brightest channels to pick and see that the timing in the 6 Southern datasets is similar to the North, with each lasting about 1 hour.

As for the Southern mosaic, you can compare the before and after self-cal images

## 2.6 Image Analysis

### 2.6.1 Moment Maps

Next we will make moment maps for the CO(3-2) emission: Moment 0 is the integrated intensity; Moment 1 is the intensity weighted velocity field; and Moment 2 is the intensity weighted velocity

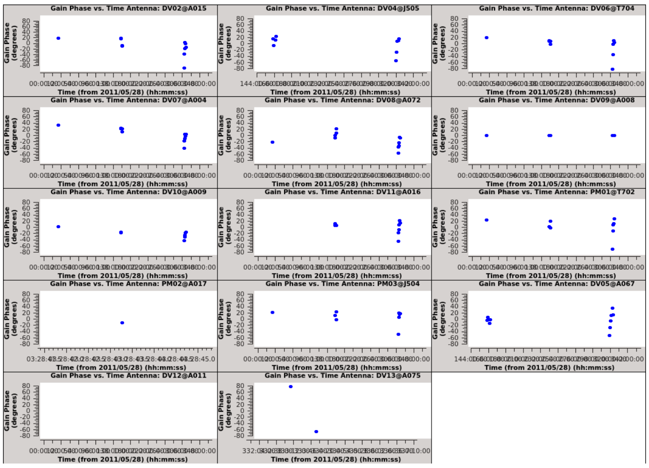

Figure 2.19: Phase-only self-calibration solutions for the Southern mosaic.

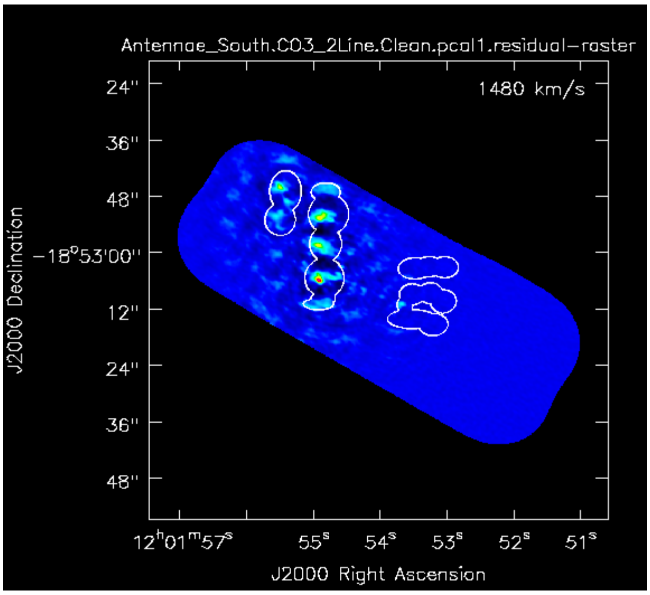


Figure 2.20: Interactive cleaning of Southern mosaic after self-cal. The clean mask is shown in white. In this example we used the same mask for all channels by selecting "All Channels" before drawing mask.

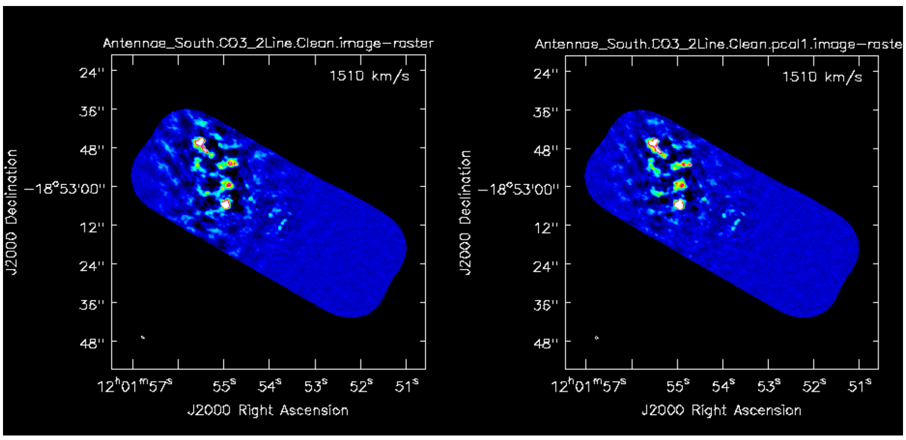


Figure 2.21: Comparison of channel 31 before selfcal (left) and after selfcal (right).

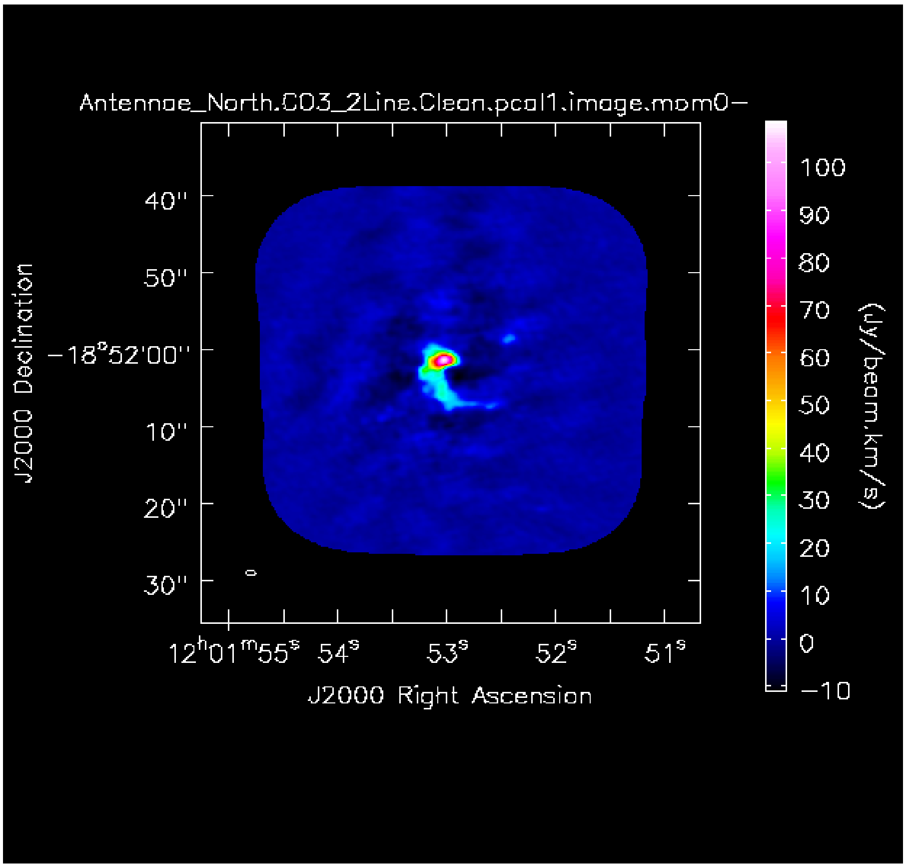


Figure 2.22: The CO(3-2) integrated intensity map (moment 0) of the Northern mosaic.

dispersion.

Above we determined the rms noise levels for both the North and South mosaics in both a line-free and a line-bright channel. We want to limit the channel range of the moment calculations to those channels with significant emission. One good way to do this is to open the cube in the viewer overlaid with 3-sigma contours, with sigma corresponding to the line-free rms.

For moment 0 (integrated intensity) maps you do not typically want to set a flux threshold because this will tend to noise bias your integrated intensity.

For higher order moments it is very important to set a conservative flux threshold. Typically something like 3sigma, using sigma from a bright line channel works well. We do this with the **mask** parameter in the commands. When making multiple moments, *immoments* appends the appropriate file name suffix to the value of **outfile**.

Next we can create the six moment maps (figures 2.23, 2.24, 2.25, 2.26, and 2.27) from these images using **imview**.

### 2.6.2 Exporting Images to Fits

If you want to analyze the data using another software package it is easy to convert from CASA format to FITS, using the *exportfits* task.

Although "FITS format" is supposed to be a standard, in fact most packages expect slightly different things from a FITS image. If you are having difficulty, try setting **velocity=True** and/or **dropstokes=True**.

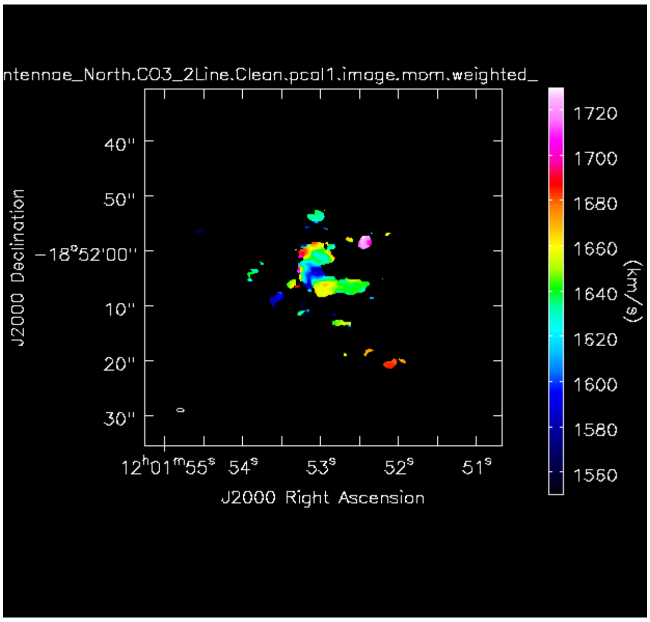


Figure 2.23: The CO(3-2) velocity field (moment 1) of the Northern mosaic.

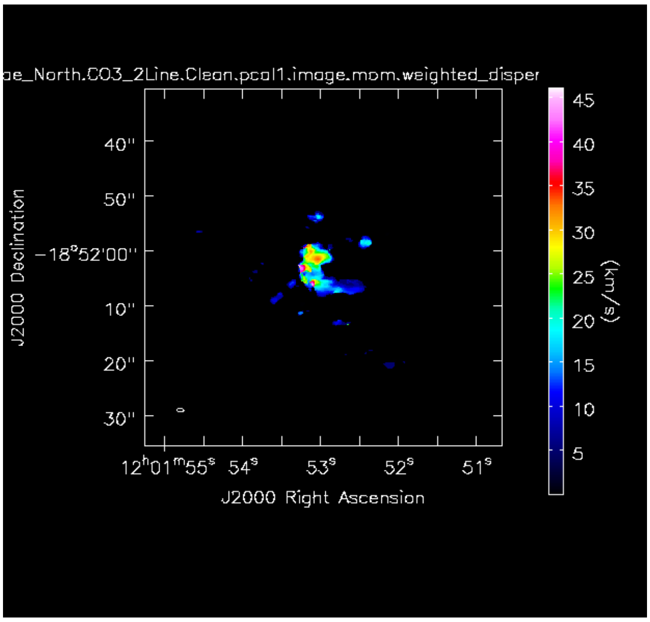


Figure 2.24: The CO(3-2) velocity dispersion (moment 2) of the Northern mosaic.

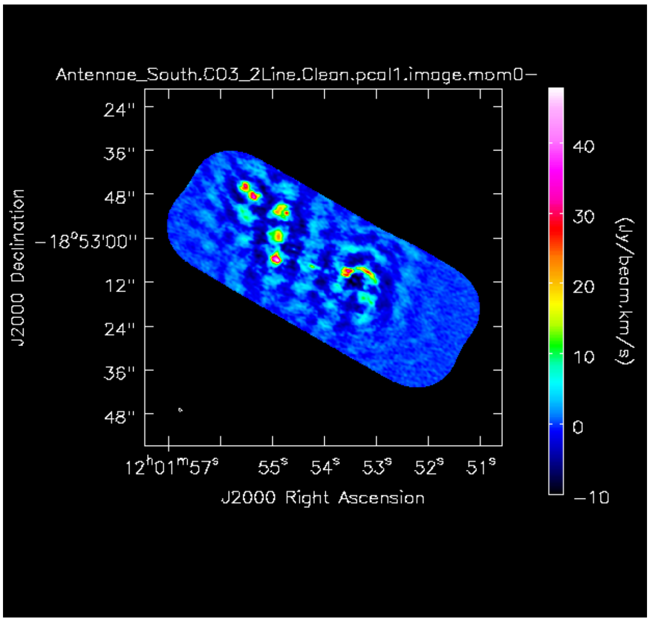


Figure 2.25: The CO(3-2) integrated intensity map (moment 0) of the Southern mosaic.

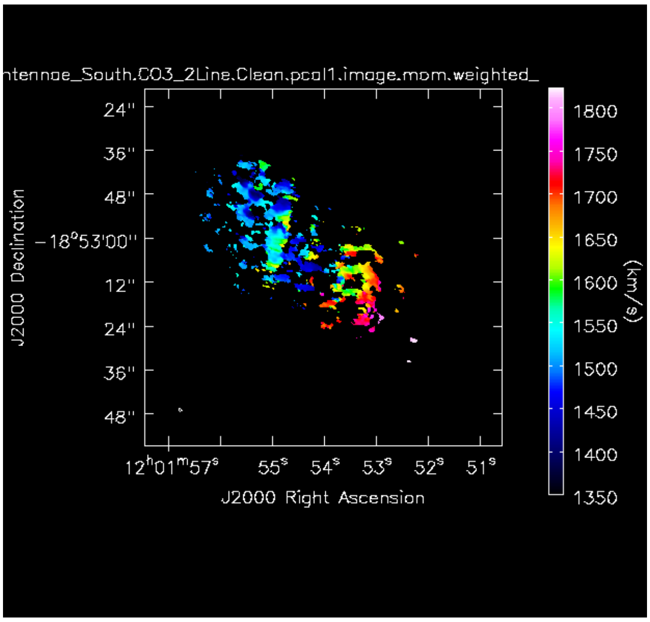


Figure 2.26: The CO(3-2) velocity field (moment 1) of the Southern mosaic.

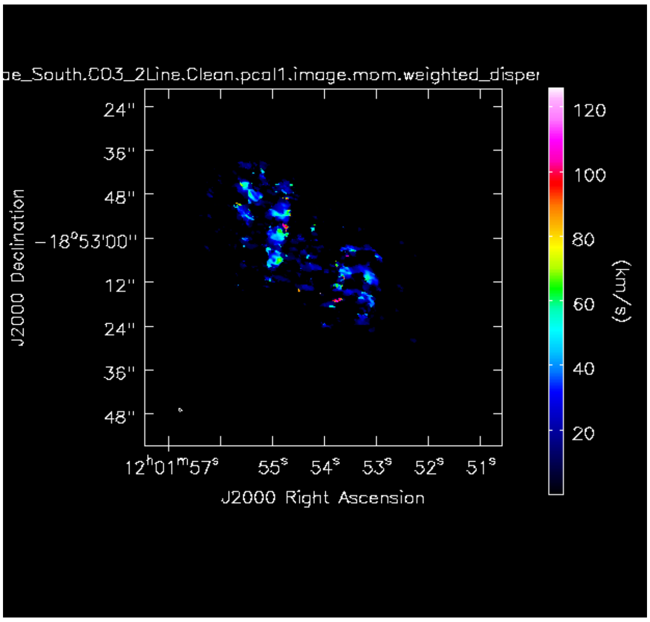


Figure 2.27: The CO(3-2) velocity dispersion (moment 2) of the Southern mosaic.

## 2.7 Comparison with previous SMA CO(3-2) data

Now we compare with SMA CO(3-2) data. Figure 2.28 shows a comparison plot between the moment 0 maps of ALMA and SMA data using the viewer. The fluxes, peak locations, and large scale structure are consistent. Both southern and northern components have been combined.

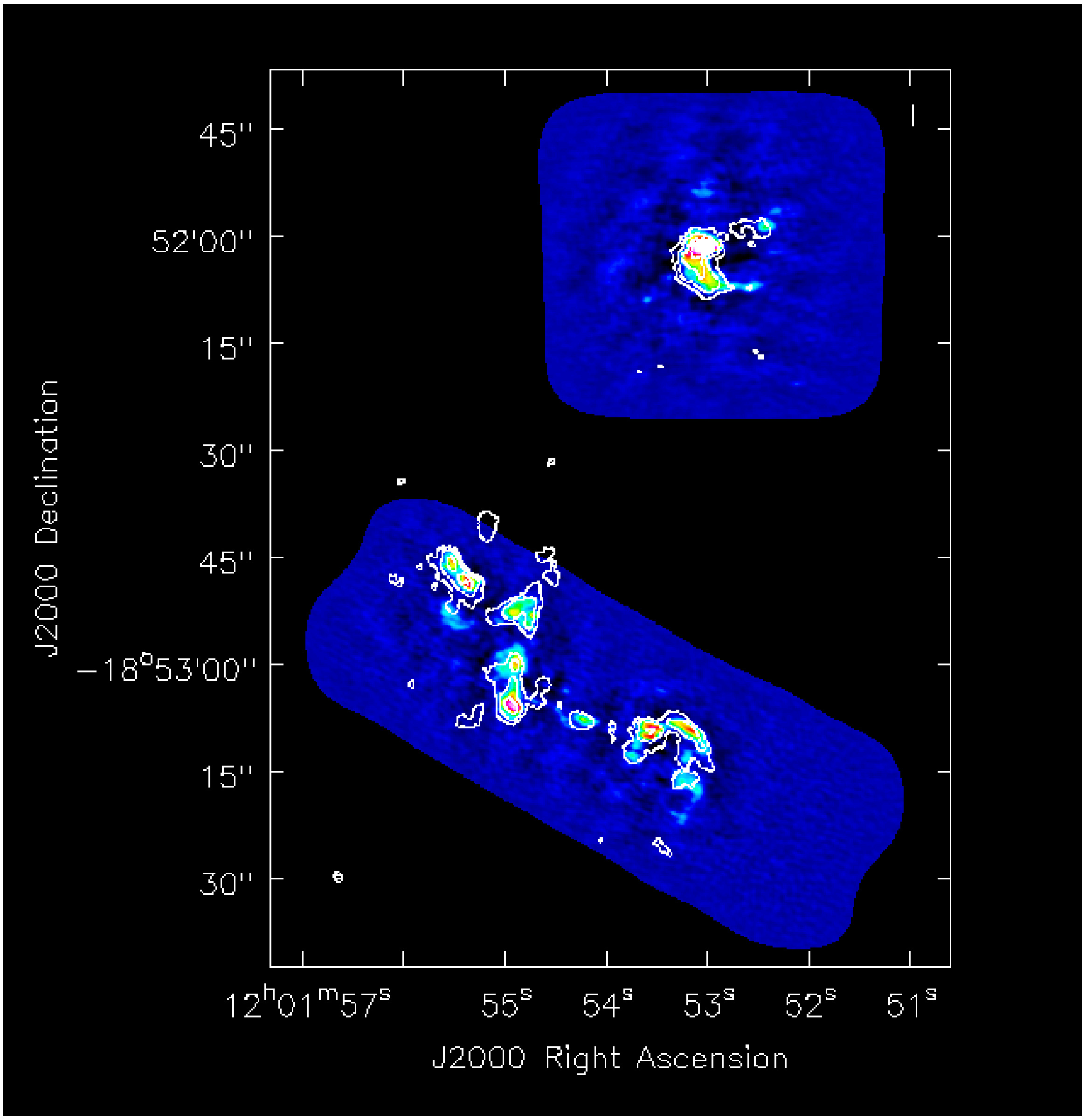


Figure 2.28: The CO(3-2) total intensity map (moment 0) comparison with SMA data. Colour image is ALMA data, combining southern and northern mosaics. Contours show SMA data (Ueda, Iono, Petitpas et al. 2012, ApJ, 745, 65).