ALMA Manual Calibration and Imaging

Dominic Ludovici | 20th Synthesis Imaging Summer School| May 20th, 2024

National Radio Astronomy Observatory

Learning Objectives

At the end of this tutorial, participants should be able to…

1. Summarize the basic steps for calibrating ALMA data.
2. Use CASA to calibrate an ALMA observation.
3. Interpret graphs of ALMA data and calibration solutions
4. Demonstrate using the CASA task Tclean to create an image from ALMA visibilities.

Tutorial Guide

- I will use Active Learning techniques to keep you engaged.
  - We will often stop for discussion and questions.
  - Please sit near someone else. Help each other and check your work against your neighbors.

- Your understanding is important to us!
  - I will be walking around the room to help while you work.
  - We have many others here to help as well.

- Ask us questions! Raise your hand and someone will come to help!

Your CASA tutorial team

Dominic Ludovici
NRAO NAASC

Devaky Kunneriath
NRAO NAASC

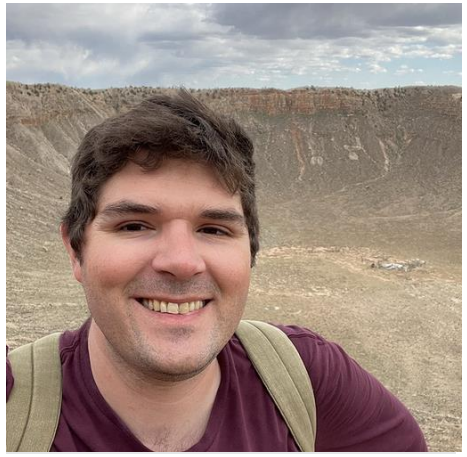
Erica Keller
NRAO NAASC

Cole Wampler
NRAO NAASC

Sarah Wood
NRAO NAASC

Your CASA tutorial team

Jess Speedie
ALMA Ambassador
University of Victoria



Patrick Kamieneski
ALMA Ambassador
Arizona State University

Natalie Butterfield
NRAO NAASC

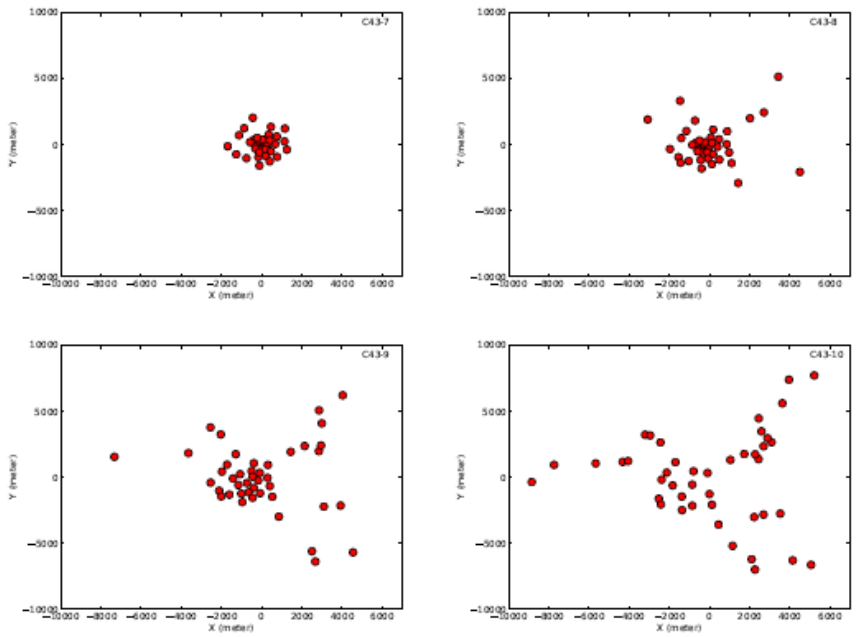
Overview

1. Data and Project Description
2. Getting Started in CASA
3. Initial Calibration and Data Preparation
4. Data Inspection
5. Calibration
6. Imaging

Data Description

Today's data is from the ALMA Long Baseline Campaign (September – December 2013)

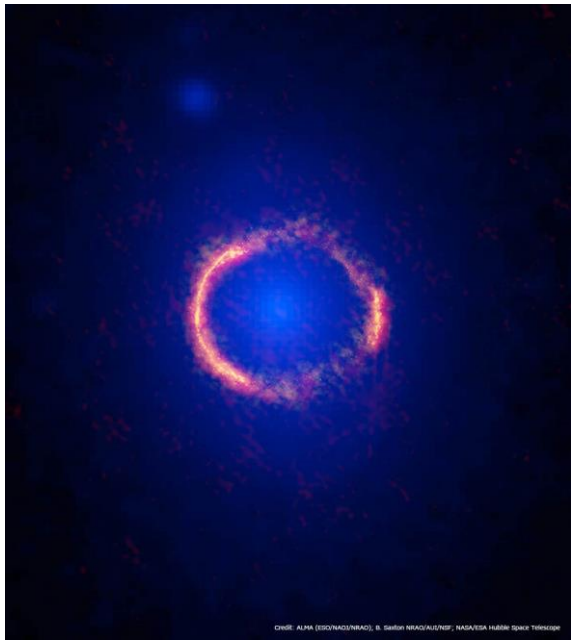
- Successful test of ALMA's longest baselines (i.e. highest resolutions) run from September through December 2014
- Baselines out to 15km (resolution up to 0.023")



Data Description

We will examine the Gravitational Lensed Galaxy SDP.81

- At z = 3.04, the star-forming galaxy SDP.81 sits behind a massive foreground elliptical galaxy (z = 0.299) which acts as a gravitational lens.
- During the Long Baseline Campaign, the dust continuum at 151, 236, and 290 GHz was mapped as well as emission lines from CO and water.
- These images allow for the determination of the physical and chemical properties of the lensed galaxy down to 180 pc size scales (similar to giant molecular clouds in the Milky Way … but at a redshift of 3!)

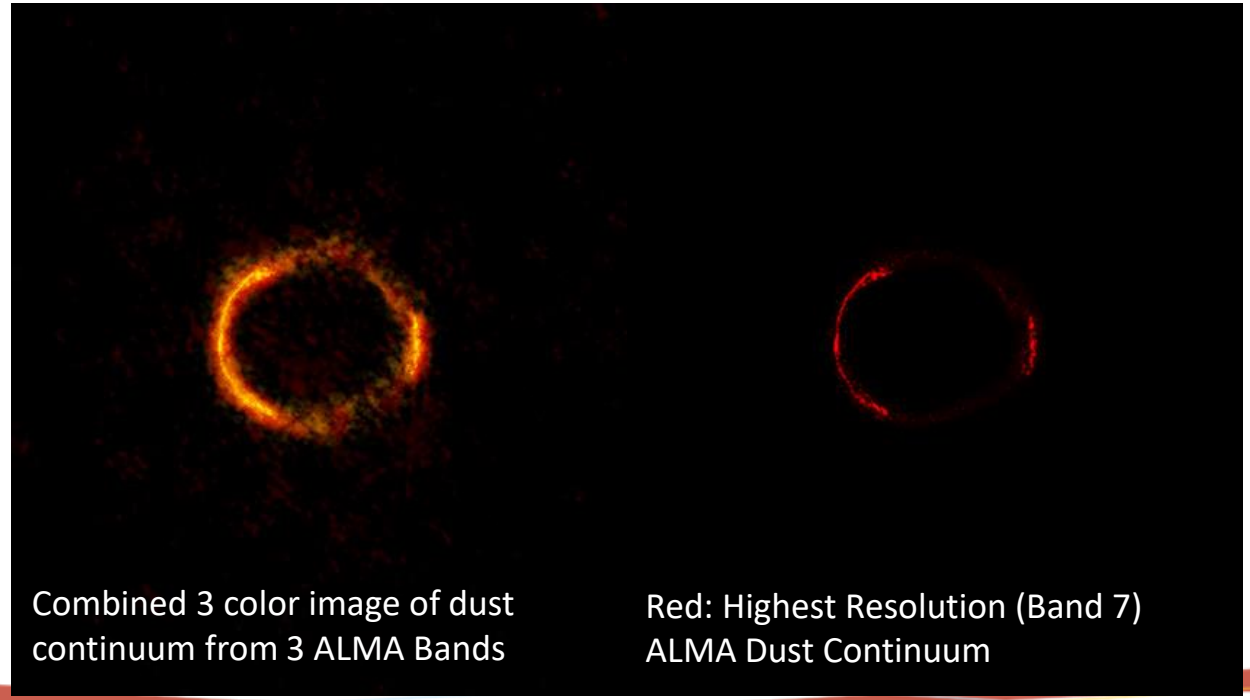


SDP.81 as seen by ALMA (in orange) and the Hubble Space Telescope (in blue)

Data Description

We will image the dust continuum emission and CO line emission observed at Band 4.

- Link to paper: http://arxiv.org/abs/1503.02652



Combined 3 color image of dust continuum from 3 ALMA Bands

Red: Highest Resolution (Band 7) ALMA Dust Continuum

Image Credits: ALMA (NRAO/ESO/NAOJ);
B. Saxton NRAO/AUI/NSF; NASA/ESA Hubble,
T. Hunter (NRAO)

Getting Started

Navigating to Your Data

Starting CASA

Task Help

Running Tasks

In your directory there should be two sub-directories labeled /Calibration and /Imaging.

In **/Calibration** you should have:

- **SDP81_B4_uncalibrated.ms.split** (the data file containing uncalibrated data with minor initial processing applied)
- **data_prep.py** (script detailing the initial processing that has already been applied)
- **calibration.py** (the script we will work through together to calibrate the data)

In **/Imaging** you have:

- **SDP.81_Band4_continuum.ms** (fully calibrated continuum measurement set ready or imaging)
- **SDP.81_Band4.ms** (fully calibrated measurement set containing both continuum and line emission ready for imaging)
- **SDP.81_Band4_COline.ms.contsub** (fully calibrated line-only measurement set)
- **imaging.py** (the script we will work through together to image the data)
- **combination.py** (a script detailing the steps taken to create the measurement sets for imaging: this is just for reference **we won't be using it!)**

Getting Started

Navigating to Your Data

Starting CASA

Task Help

Running Tasks

CASA (Common Astronomy Software Applications)

- CASA is the offline data reduction package for ALMA and the VLA (data from other telescopes usually work too, but not primary goal of CASA)
- Code is C++ (fast) bound to Python which provides ease of use and scripting
- Provides data import/export, inspection, editing, calibration, imageing, viewing, and analysis
- Also supports single dish data reduction
- We have a lot of documentation (CASA docs), reduction tutorials (CASA guides), and a helpdesk.

Getting Started

Start CASA.

Lets make sure everyone can start CASA!

If you put CASA in your path, you should be able to start it by typing "`casa`" on the command line.

If not, give the full path:
</path/to/casa/install>/bin/casa

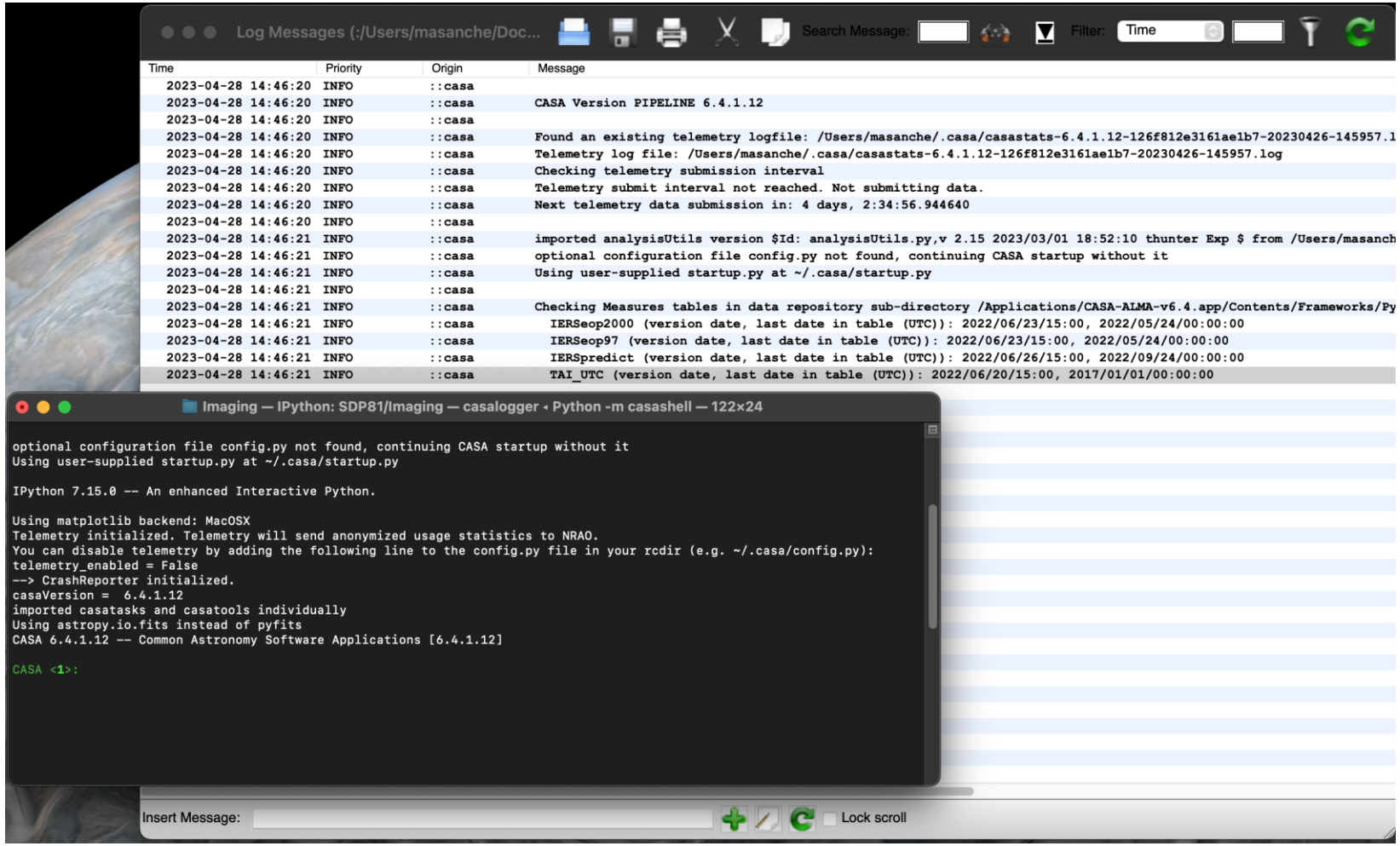
Getting Started

Navigating to Your Data

Starting CASA

Task Help

Running Tasks



National Radio Astronomy Observatory

Getting Started

Navigating to Your Data

Starting CASA

Task Help

Running Tasks

CASA runs within pythons scripts or through the interactive *IPython* (ipython.org) interface

- IPython Features:
  - shell access
  - auto-parenthesis (autocall)
  - Tab auto-completion
  - command history (arrow up and “hist [-n]”)
  - session logging
  - casaTIME.log – casa logger messages
  - numbered input/output
  - history/searching

Getting Started

Navigating to Your Data

Starting CASA

Task Help

Running Tasks

Tasks - high-level functionality

- function call or parameter handling interface
- these are what you should use in tutorials

Tools - complete functionality

- **tool.method()** calls
- they are internally used by tasks or can be used on their own sometimes shown in tutorial scripts and CASAGuides

Applications – some tasks/tools invoke standalone apps

- e.g. **casaviewer, mpicasa**

Shell commands can be run with a leading exclamation mark

- **!du –ls** or inside **os.system("shell command")**
- (some key shell commands like "ls" work without the exclamation mark
- We will use **os.system()** exclusively within this tutorial.)

Getting Started

Navigating to Your Data

Starting CASA

Task Help

Running Tasks

taskhelp

Calibration — IPython: SDP81/Calibration — casalogger ◂ Python -m casashell — 110×50

```
CASA <1>: taskhelp

--------------------------------------------------------------------
CASA tasks
--------------------------------------------------------------------
> analysis
--------------------------------------------------------------------
    imcollapse : Collapse image along one axis, aggregating pixel values along that axis.
     imcontsub : Estimates and subtracts continuum emission from an image cube
         imdev : Create an image that can represent the statistical deviations of the input image.
         imfit : Fit one or more elliptical Gaussian components on an image region(s)
        imhead : List, get and put image header parameters
     imhistory : Retrieve and modify image history
        immath : Perform math operations on images
    immoments : Compute moments from an image
      impbcor : Construct a primary beam corrected image from an image and a primary beam pattern.
         impv : Construct a position-velocity image by choosing two points in the direction plane.
       imrebin : Rebin an image by the specified integer factors
     imreframe : Change the frame in which the image reports its spectral values
      imregrid : regrid an image onto a template image
      imsmooth : Smooth an image or portion of an image
       imstat : Displays statistical information from an image or image region
   imsubimage : Create a (sub)image from a region of the image
      imtrans : Reorder image axes
        imval : Get the data value(s) and/or mask value in an image.
       listvis : List measurement set visibilities.
         rmfit : Calculate rotation measure.
       specfit : Fit 1-dimensional gaussians and/or polynomial models to an image or image region
     specflux : Report spectral profile and calculate spectral flux over a user specified region
    specsmooth : Smooth an image region in one dimension
        spxfit : Fit a 1-dimensional model(s) to an image(s) or region for determination of spectral index.
--------------------------------------------------------------------
> calibration
--------------------------------------------------------------------
         accor : Normalize visibilities based on auto-correlations
      applycal : Apply calibrations solutions(s) to data
      bandpass : Calculates a bandpass calibration solution
         blcal : Calculate a baseline-based calibration solution (gain or bandpass)
       calstat : Displays statistical information on a calibration table
      clearcal : Re-initializes the calibration for a visibility data set
        delmod : Deletes model representations in the MS
    fixplanets : Changes FIELD and SOURCE table entries based on user-provided direction or POINTING table, op
tionally fixes the UVW coordinates
     fluxscale : Bootstrap the flux density scale from standard calibrators
     fringefit : Fringe fit delay and rates
            ft : Insert a source model as a visibility set
       gaincal : Determine temporal gains from calibrator observations
        gencal : Specify Calibration Values of Various Types
   initweights : Initializes weight information in the MS
```

Getting Started

Navigating to Your Data

Starting CASA

Task Help

Running Tasks

Task Execution:

Write the full parameter set in a line:

- taskname(arg1=val1, arg2=val2, ...)
- e.g. tclean(vis='input.ms',imagename='galaxy', robust=0.5, imsize=[200,200])

unspecified parameters will be set to their default values (globals not used; i.e. not to previously set variables)

Useful in scripts, but also in 'pseudo-scripts':

- To keep a record, it is frequently a good idea to write down the full line as above in an editor, then cut and paste into CASA.
- When changes are needed, change in editor and cut and paste again. That is good practice to keep a record of the exact input.
- But note that the logger is also repeating the full task command

Getting Started

Navigating to Your Data

Starting CASA

Task Help

Running Tasks

Scripts can be run with the following:

execfile('<yourscript.py>')

Python 3 removed the execfile built-in function. CASA provides a convenience function that attempts to reproduce the behavior of the Python 2.7 built-in execfile function.

Getting Started

Navigating to Your Data

Starting CASA

Task Help

Running Tasks

CASA also has a asking interface, similar to AIPS, MIRIAD, etc.

- parameter manipulation commands
  - inp, default, saveinputs, tget, tput
- use parameters set as global Python variables
  - <param> = <value>
  - (e.g. vis = 'ngc5921.demo.ms')
- execute
  - <taskname> or go (e.g. tclean())

- return values (except when using "go")
  - some tasks return Python dictionaries, assign a variable name to get them, e.g. myval=imval()
  - Very useful for scripting based on task outputs

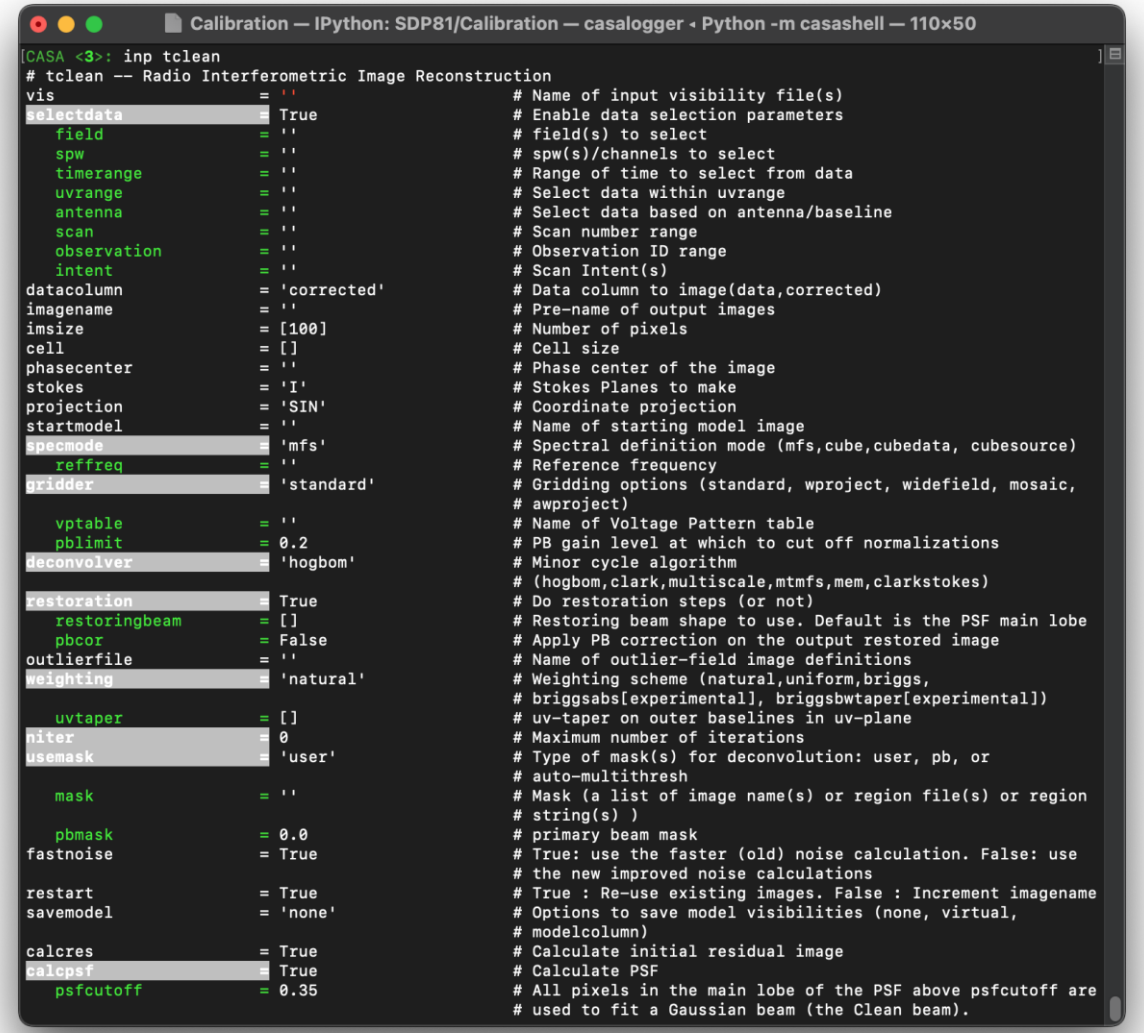
Getting Started

Navigating to Your Data

Starting CASA

Task Help

Running Tasks



Getting Started

Navigating to Your Data

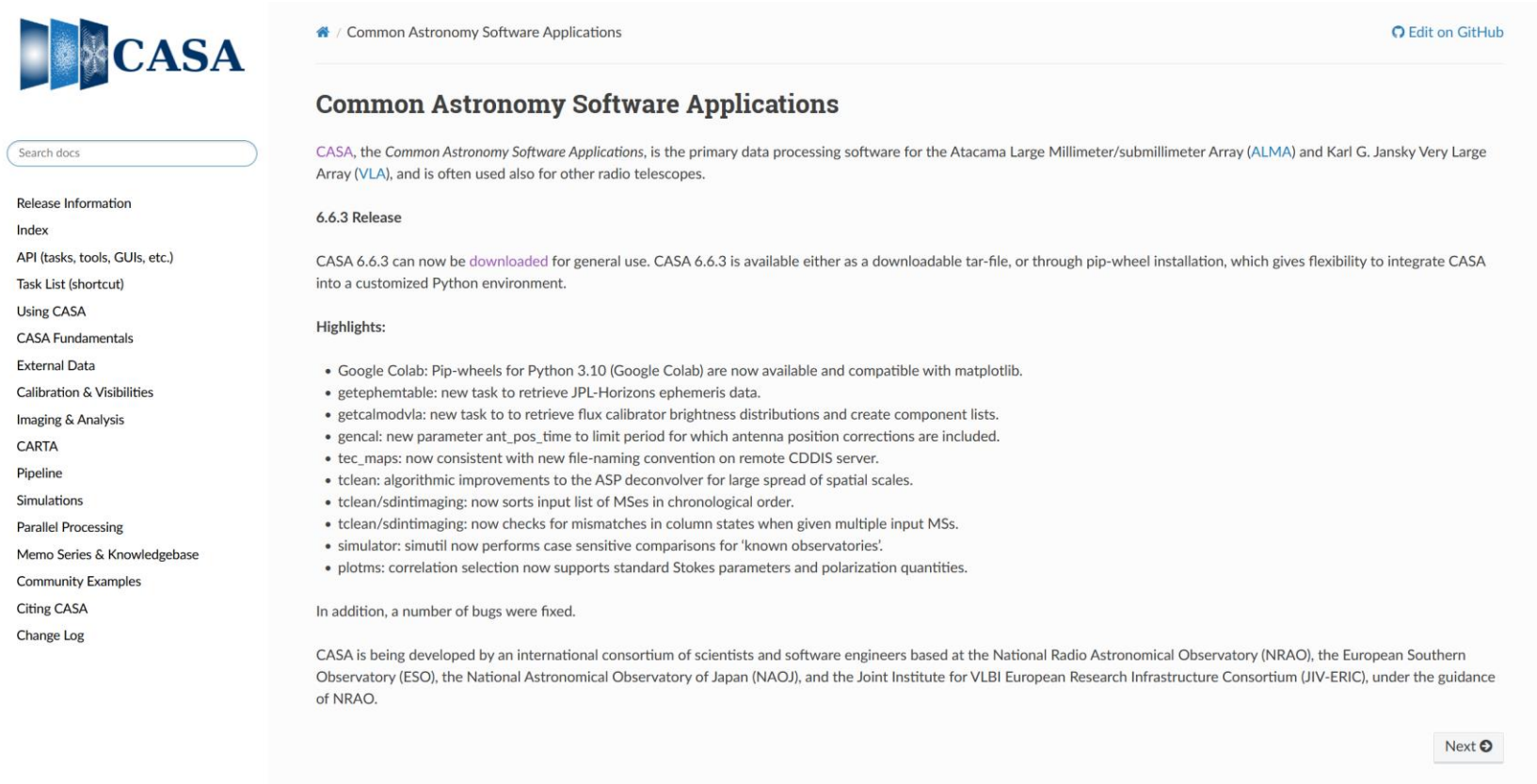
Starting CASA

Task Help

Running Tasks

Getting Help:

https://casadocs.readthedocs.io/en/stable/



National Radio Astronomy Observatory

Initial Calibration

Data Preparation

Importing your ASDM

Table Corrections

Initial Flaging

Online Correction: WVR

Online Correction: Tsys

Online Correction: Antenna Position

Applying Corrections and split

Downloading data from the ALMA archive will return raw data along with the scripts necessary for calibrating the data. In the interest of time, we have already applied some initial corrections to the raw data for you. All of these steps are detailed in

Data_prep.py

In this section, we will briefly explain the steps taken in data_prep.py

- Import the raw data into a casa measurement set.
- Occasionally a dataset will require a fix to some of the metadata (i.e. the header). In this case, some coordinates in the metadata are adjusted.
- Data that is known to be irrelevant to calibration or to be problematic (even without inspection of the data) is flagged. Examples: data taken when the telescope was not on source yet, when the system temperature load was too close to the beam, when the receivers were not yet tuned)
- Create 3 correction tables (WVR, Tsys, antenna positions) and apply them.
- The output of data_prep.py is SDP81_B4_uncalibrated.ms.split

(we will start calibration with this data file)

Data Preparation

Importing your ASDM

Table Corrections

Initial Flaging

Online Correction: WVR

Online Correction: Tsys

Online Correction: Antenna Position

Applying Corrections and split

When downloading data from the ALMA archive, the data arrive in .asdm format. These data must be imported before processing.

From data_prep.py:

```
importasdm('uid___A002_X924c91_X34f',
 asis='Antenna Station Receiver Source
  CalAtmosphere CalWVR CorrelatorMode SBSummary',
  bdfflags=True,
 lazy=False,
 process_caldevice=False)
```

Data Preparation

Importing your ASDM

Table Corrections

Initial Flaging

Online Correction: WVR

Online Correction: Tsys

Online Correction: Antenna Position

Applying Corrections and split

A dataset may require a fix to some of the metadata (i.e. the header). In this case, some coordinates in the metadata are adjusted.

From data_prep.py:

```
es.fixForCSV2555('SDP81_B4_uncalibrated.ms')
fixsyscaltimes(vis = 'SDP81_B4_uncalibrated.ms')
```

Data Preparation

Importing your ASDM

Table Corrections

Initial Flaging

Online Correction: WVR

Online Correction: Tsys

Online Correction: Antenna Position

Applying Corrections and split

We proceed by flagging the data which will not use for the calibration: pointing measurements and tsys measurements, as well as autocorrelations (correlations within a given antenna). Note that the information on tsys is still contained in the measurement step, as a table (which we extract later).

From data_prep.py:

```
flagdata(vis = 'SDP81_B4_uncalibrated.ms',
   mode = 'manual',
   spw = '1~21',
   autocorr = True,
   flagbackup = False)

flagdata(vis = 'SDP81_B4_uncalibrated.ms',
   mode = 'manual',
   intent = '*POINTING*,*ATMOSPHERE*',
   flagbackup = False)

flagcmd(vis = 'SDP81_B4_uncalibrated.ms',
   inpmode = 'table’,
   useapplied = True,
   action = 'apply')
```

Data Preparation

Importing your ASDM

Table Corrections

Initial Flaging

Online Correction: WVR

Online Correction: Tsys

Online Correction: Antenna Position

Applying Corrections and split

The Water Vapor Radiometers (WVR) provide corrections to the phase delay due to the atmosphere.

- Key to correcting short-timescale phase variations
- Time variable phase calibration

From data_prep.py:

```
wvrgcal(vis = 'SDP81_B4_uncalibrated.ms',
  caltable = 'SDP81_B4_uncalibrated.ms.wvr',
  spw = [11, 13, 15, 17],
  smooth = '2.016s',
  toffset = 0,
  tie = ['SDP.81,J0909+0121'],
  statsource = 'SDP.81')
```

Data Preparation

Importing your ASDM

Table Corrections

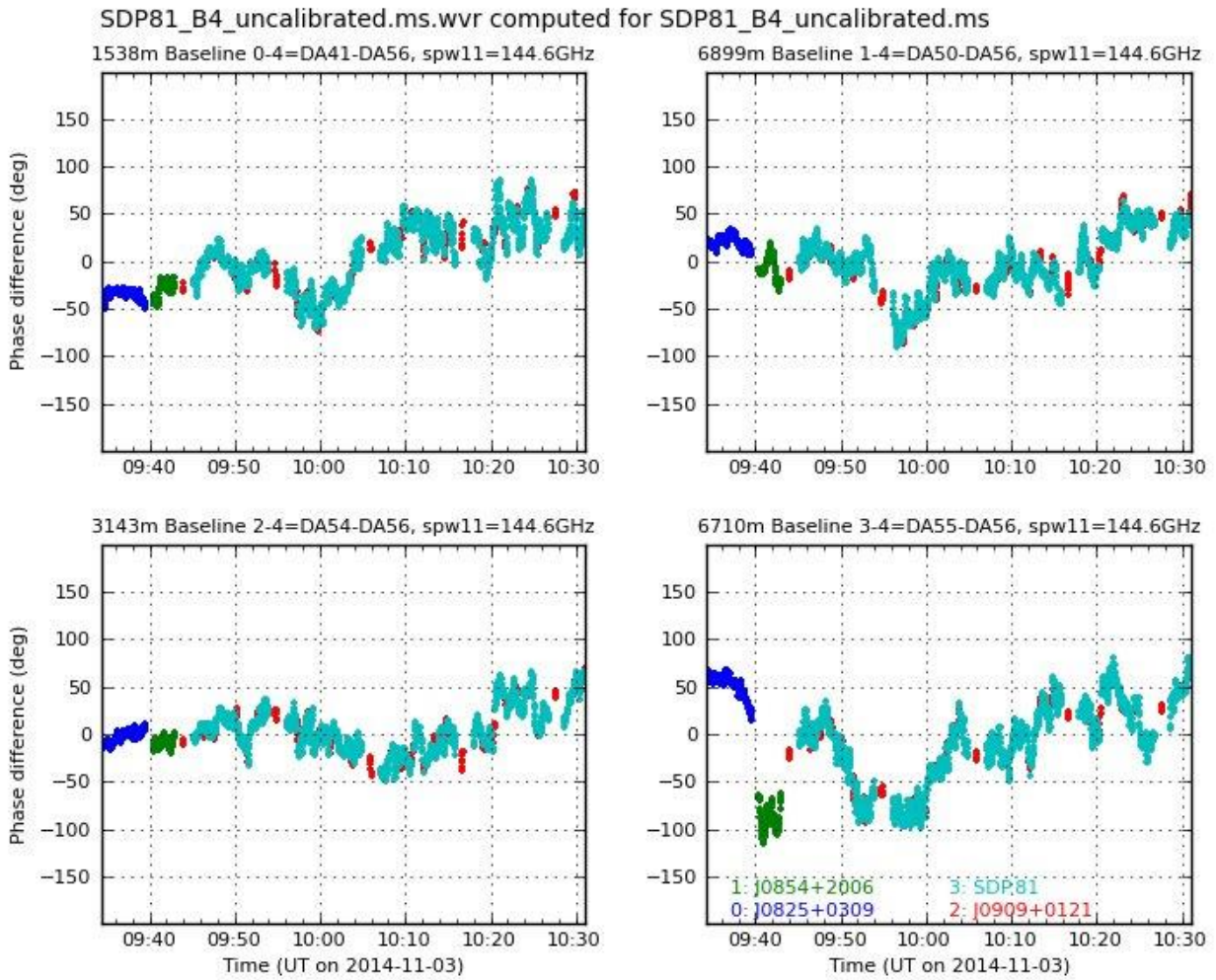
Initial Flaging

Online Correction: WVR

Online Correction: Tsys

Online Correction: Antenna Position

Applying Corrections and split



Data Preparation

Importing your ASDM

Table Corrections

Initial Flaging

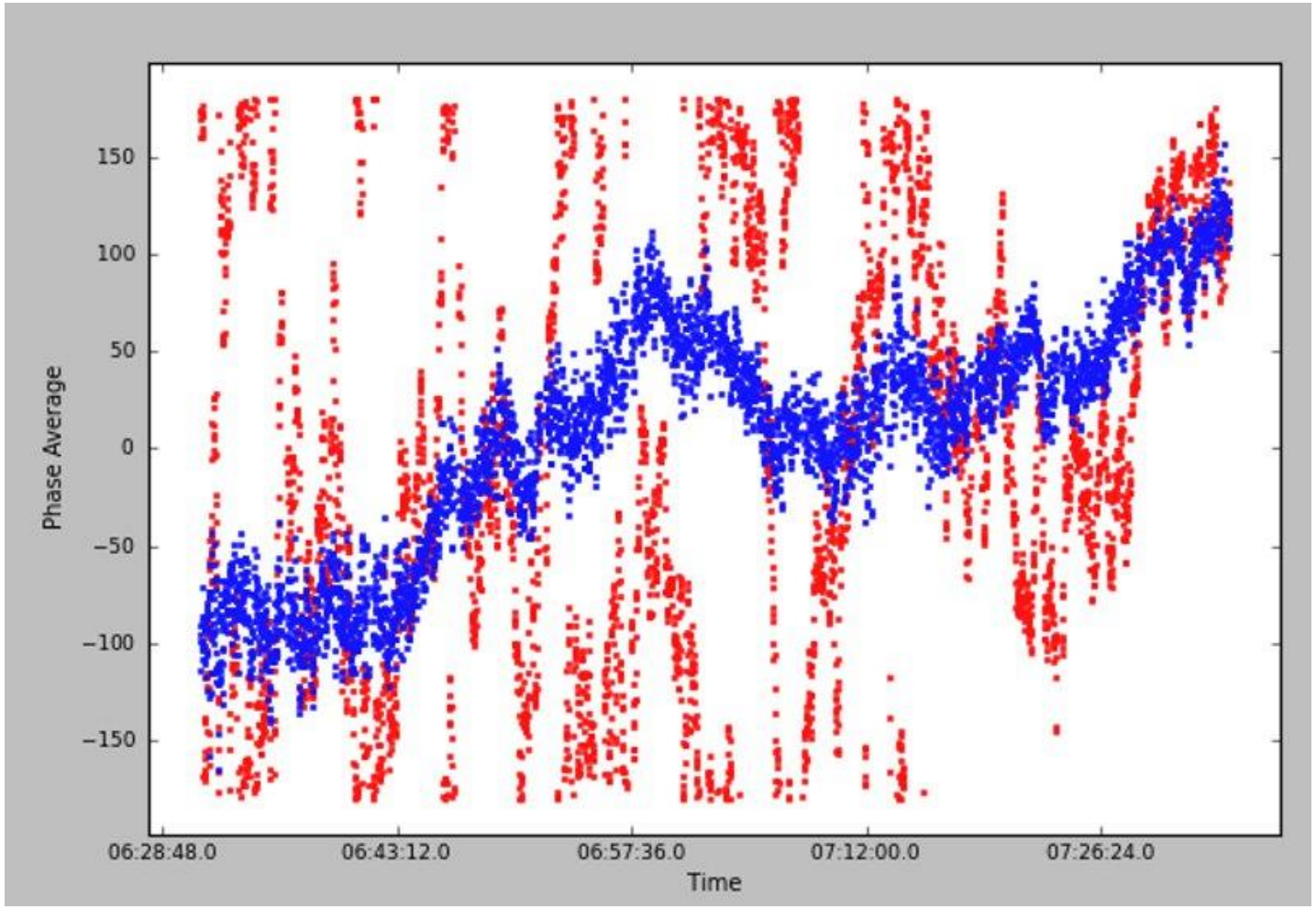
Online Correction: WVR

Online Correction: Tsys

Online Correction: Antenna Position

Applying Corrections and split

High Frequency data from TW Hydra.



Data Preparation

Importing your ASDM

Table Corrections

Initial Flaging

Online Correction: WVR

Online Correction: Tsys

Online Correction: Antenna Position

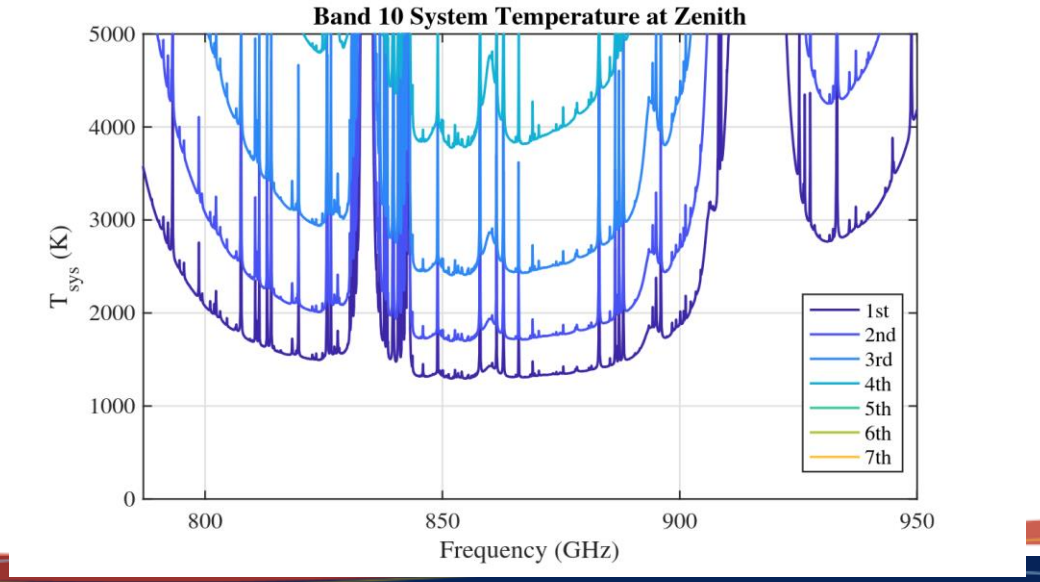
Applying Corrections and split

System Temperature (Tsys) corrections allow for the correction of atmospheric emission and opacity.

System Temperatures are important for

- Gain transfer across elevation
- Frequency dependent amplitude calibration

Tsys can vary from from ~50 K in Band 1 to ~1500 K in Band 10.



Data Preparation

Importing your ASDM

Table Corrections

Initial Flaging

Online Correction: WVR

Online Correction: Tsys

Online Correction: Antenna Position

Applying Corrections and split

Tsys calibration table application:

From data_prep.py:

```
gencal(vis = 'SDP81_B4_uncalibrated.ms',
  caltable = 'SDP81_B4_uncalibrated.ms.tsys',
  caltype = 'tsys')
```

Data Preparation

Importing your ASDM

Table Corrections

Initial Flaging

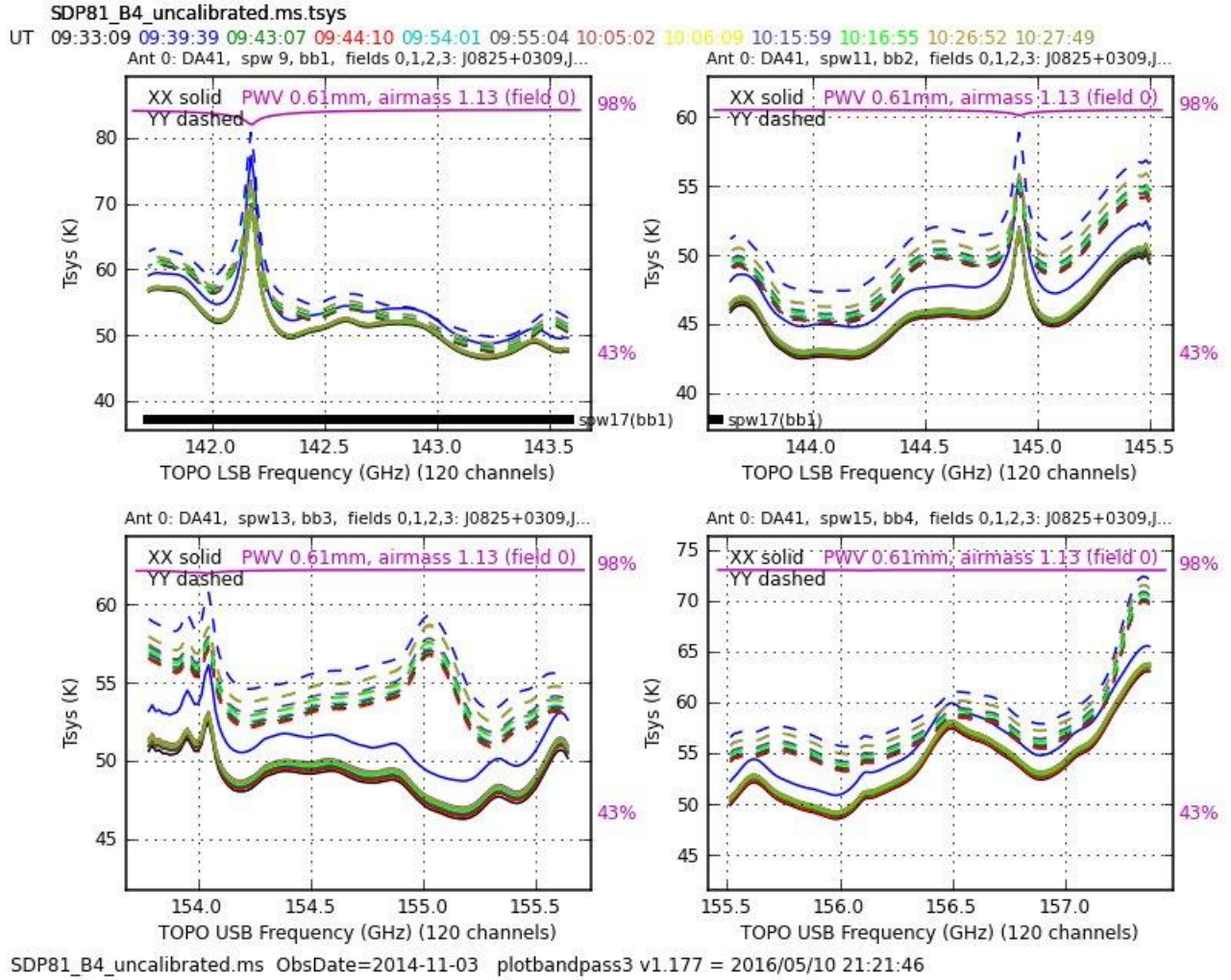
Online Correction: WVR

Online Correction: Tsys

Online Correction: Antenna Position

Applying Corrections and split

When looking at Tsys plots, look for very high spikes and discontinuities. These look good.



National Radio Astronomy Observatory

Data Preparation

Importing your ASDM

Table Corrections

Initial Flaging

Online Correction: WVR

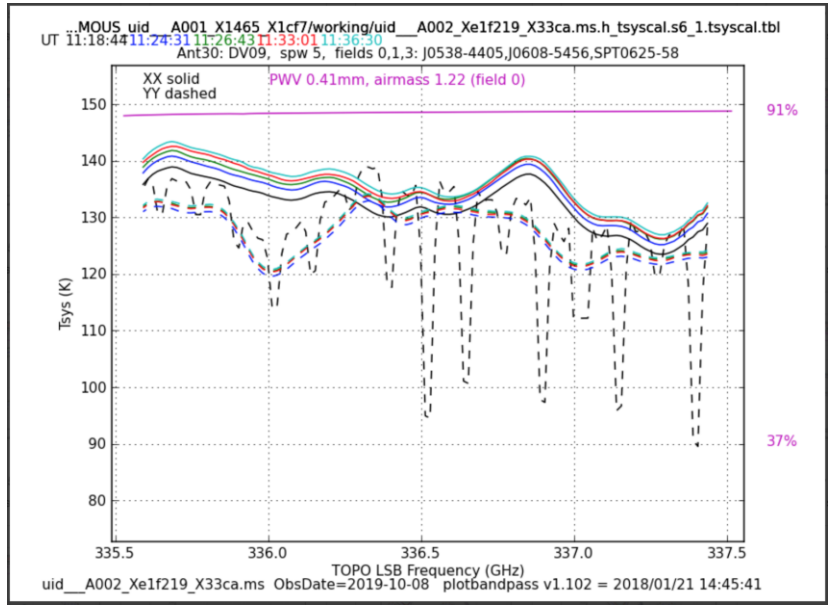
Online Correction: Tsys

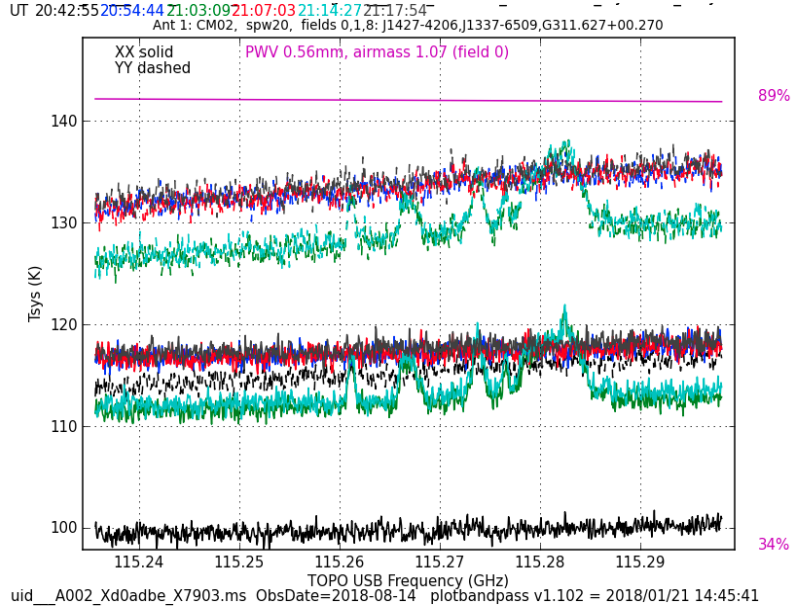
Online Correction: Antenna Position

Applying Corrections and split

When looking at Tsys plots, look for very high spikes and discontinuities. These do not.

On the right is an example of Tsys contamination from an astronomical source. The channels involved would need flagged so we can interpolate across the contamination.





Data Preparation

Importing your ASDM

Table Corrections

Initial Flaging

Online Correction: WVR

Online Correction: Tsys

Online Correction: Antenna Position

Applying Corrections and split

Antenna position corrections updates the accuracy of the antenna positions.

The offsets are in meters and normally look like this:

```
# antenna x_offset y_offset z_offset total_offset baseline_date
# DV14    -4.61575e-04    7.57190e-04    1.74002e-03 1.95296e-03      2014-10-31 11:27:40
# DA50     4.24031e-05   -4.98282e-04    1.51997e-03 1.60012e-03      2014-10-31 11:27:40
# DV22    -9.64679e-04    1.07473e-03    3.88599e-04 1.49554e-03      2014-10-31 11:27:40
# DV08     5.53798e-04   -1.32566e-03    2.52869e-04 1.45877e-03      2014-10-31 11:27:40
# DA64    -2.80747e-04    2.60536e-04    1.39146e-03 1.44321e-03      2014-10-31 11:27:40
# DA54     7.92693e-04   -1.16213e-03   -4.01242e-05 1.40731e-03      2014-10-31 11:27:40
# DA62     1.95323e-04   -4.82360e-06    1.32798e-03 1.34227e-03      2014-10-31 11:27:40
# DV17     1.09515e-04   -3.07546e-04    1.20603e-03 1.24944e-03      2014-10-31 11:27:40
# DV04     3.70800e-04   -4.36427e-04    4.07359e-04 7.02782e-04      2014-10-31 11:27:40
# DA41     5.09151e-04   -3.88547e-04    1.20386e-04 6.51687e-04      2014-10-31 11:27:40
```

Data Preparation

Importing your ASDM

Table Corrections

Initial Flaging

Online Correction: WVR

Online Correction: Tsys

Online Correction: Antenna Position

Applying Corrections and split

Our dataset did require antenna position corrections. However, we still apply a table with 0.0 offsets for consistency and to provide an example.

From data_prep.py:

```
gencal(vis='SDP81_B4_uncalibrated.ms',
caltable='SDP81_B4_uncalibrated.ms.antpos',
caltype='antpos',
antenna='DA41,DA50,DA54,DA62,DA64,DV04,DV08,DV14,DV17,DV22',
parameter=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0])
```

Data Preparation

Importing your ASDM

Table Corrections

Initial Flaging

Online Correction: WVR

Online Correction: Tsys

Online Correction: Antenna Position

Applying Corrections and Split

Finally, we can apply the solutions and split the data set.

```
applycal(vis = 'SDP81_B4_uncalibrated.ms',
  field = '0',
  spw = '11,13,15,17',
  gaintable= ['SDP81_B4_uncalibrated.ms.tsys',
   'SDP81_B4_uncalibrated.ms.wvr',
   'SDP81_B4_uncalibrated.ms.antpos'],
  gainfield = ['0', '', ''],
  interp = 'linear,linear',
  spwmap = [tsysmap,[],[]],
  calwt = True,
  flagbackup = False)
```

(Repeat for all sources, changing the gainfield and the field number.)

```
split(vis = 'SDP81_B4_uncalibrated.ms',
  outputvis = 'SDP81_B4_uncalibrated.ms.split',
  datacolumn = 'corrected',
  spw = '11,13,15,17',
  width=[2,2,2,2,],
  keepflags = True)
```

Data Inspection

Data Inspection

Viewing Observation Information

Plotting Antenna Distribution

Viewing Uncalibrated Data

We are going to inspect our data to learn about the observation and check for problems.

Grab your computers and get ready to follow along!

The calibration.py script contains the commands for you if you want to copy them.

Jim Torson analyzes data from the Very Large Array in its computer room in 1983. (Image Credit: NRAO)

Data Inspection

Viewing Observation Information

Plotting Antenna Distribution

Viewing Uncalibrated Data

Lets start by looking at a summary of the measurement set.

Use the task `listobs()`

From calibration.py:

```
listobs("SDP81_B4_uncalibrated.ms.split")
```

Question: Listobs

After taking a look at the listobs output in the logger window, what information do you see that you think may be useful during processing?

You can write your listobs to a file using:

```
listobs(vis = 'SDP81_B4_uncalibrated.ms.split’,
   listfile = 'SDP81_B4_uncalibrated.ms.split.listobs')
```

Data Inspection

Viewing Observation Information

Plotting Antenna Distribution

Viewing Uncalibrated Data

Lets start by looking at a summary of the measurement set.

Use the task `listobs()`

```
listobs("SDP81_B4_uncalibrated.ms.split")
```

```
MeasurementSet Name:  SDP81_B4_uncalibrated.ms.split
MS Version Fields: 4
IDCode         Name           RA                      Decl              Epoch       SrcId      nRows
0    none      J0825+0309     08:25:50.338355         +03.09.24.52006   J2000       0          218400
1    none      J0854+2006     08:54:48.874929         +20.06.30.64088   J2000       1          120900
2    none      J0909+0121     09:09:10.091592         +01.21.35.61768   J2000       2          318500
3    none      SDP.81         09:03:11.610000         +00.39.06.70000   J2000       3         1287000
Spectral Windows:  (4 unique spectral windows and 1 unique polarization setups)
SpwID          Name              #Chans      Frame          Ch0(MHz)       ChanWid(kHz)      TotBW(kHz)  CtrFreq(MHz)      Num
0              ALMA_RB_04#BB_2      64       TOPO           145550.922    -31250.000         2000000.0  144566.5468         2
1              ALMA_RB_04#BB_3      64       TOPO           153727.218     31250.000         2000000.0  154711.5928         3
2              ALMA_RB_04#BB_4      64       TOPO           155459.988     31250.000         2000000.0  156444.3626         4
3              ALMA_RB_04#BB_1   1920       TOPO           143586.559      -976.562         1875000.0  142649.5468         1
```

Data Inspection

Viewing Observation Information

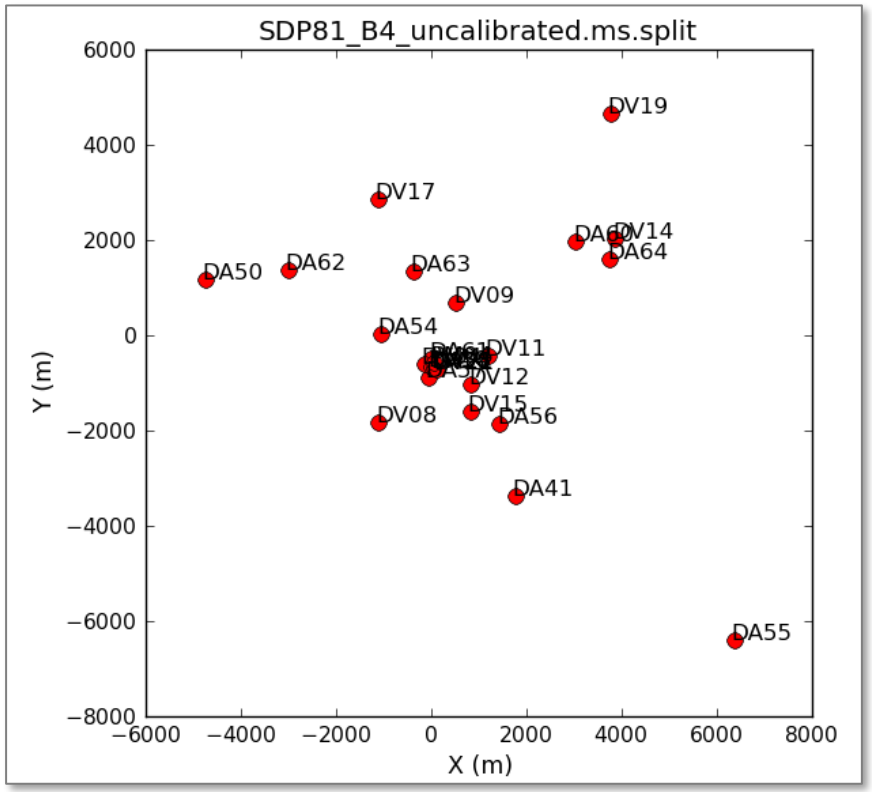
Plotting Antenna Distribution

Viewing Uncalibrated Data

Now, we will plot the antenna distribution using listobs()

From calibration.py:

```
plotants("SDP81_B4_uncalibrated.ms.split",
    figfile="plotants.png")
```



Question: Reference Antenna

- When calibrating, we typically want a reference antenna that is well behaved and towards the center of the array.

- What antennas do you think would work out well?

Data Inspection

Viewing Observation Information

Plotting Antenna Distribution

Viewing Uncalibrated Data

We are going to choose DA56 as our reference antenna. This will be used in all commands going forward.

WAIT?!?

I just said we wanted to use an antenna in the center of the array?

Well, I didn't notice before I sent out the script… And it works for this dataset. However, choosing an antenna outside the array center is not best practice.

Data Inspection

Viewing Observation Information

Plotting Antenna Distribution

Viewing Uncalibrated Data

Plotms is a general purpose graphical interface for plotting and flagging UV data and calibration tables.

You can start plotms from the CASA interface:

```
Plotms()
```

You can also fully specify the plotms command from the command line. For example:

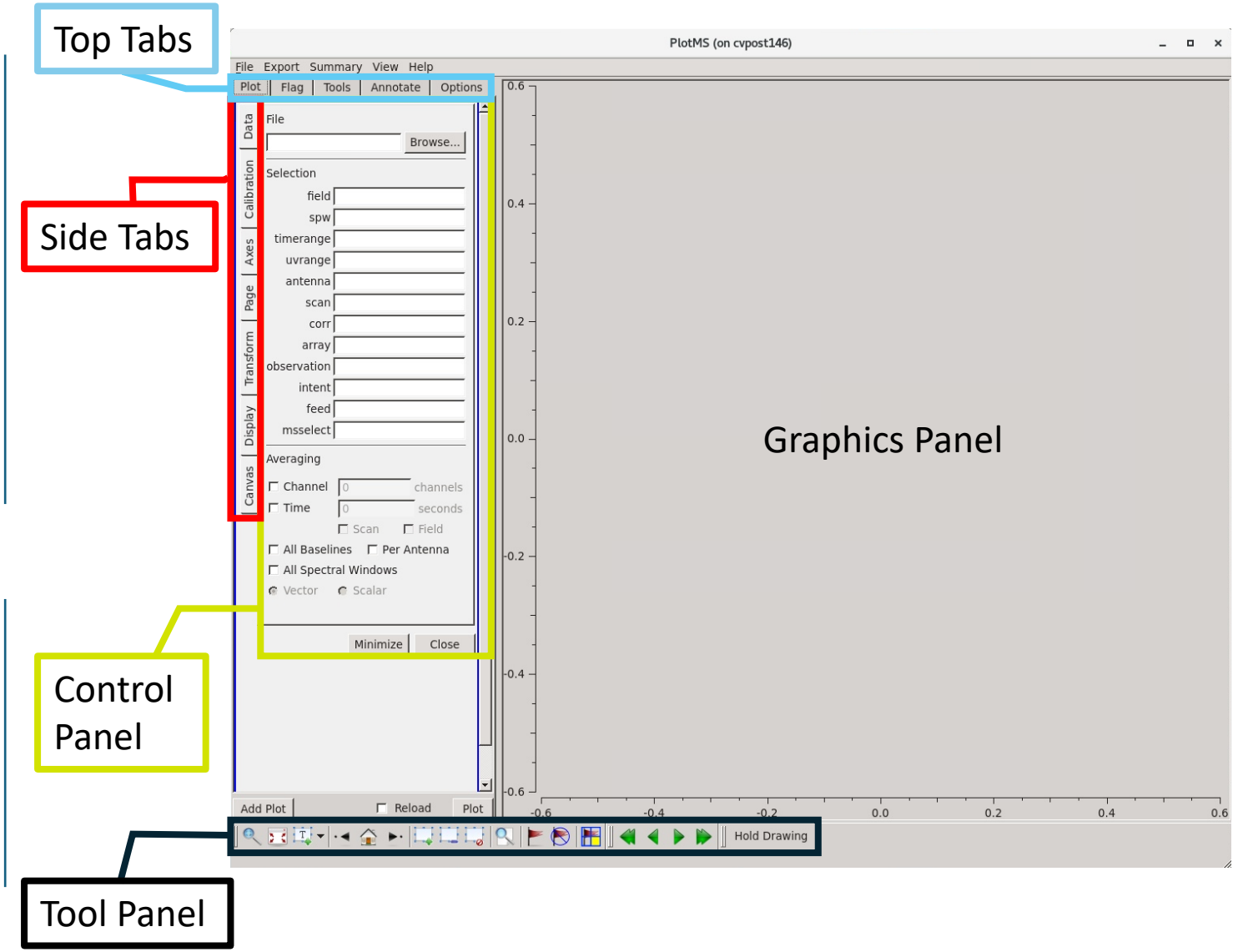
```
plotms(vis="SDP81_B4_uncalibrated.ms.split",
  xaxis="time",
  yaxis="amp",
  ydatacolumn="data",
  field="0,1,2",
  averagedata=True,
  avgchannel="1e3",
  avgtime="1e3",
  coloraxis="field")
```

Data Inspection

Viewing Observation Information

Plotting Antenna Distribution

Viewing Uncalibrated Data



Data Inspection

Viewing Observation Information

Plotting Antenna Distribution

Viewing Uncalibrated Data

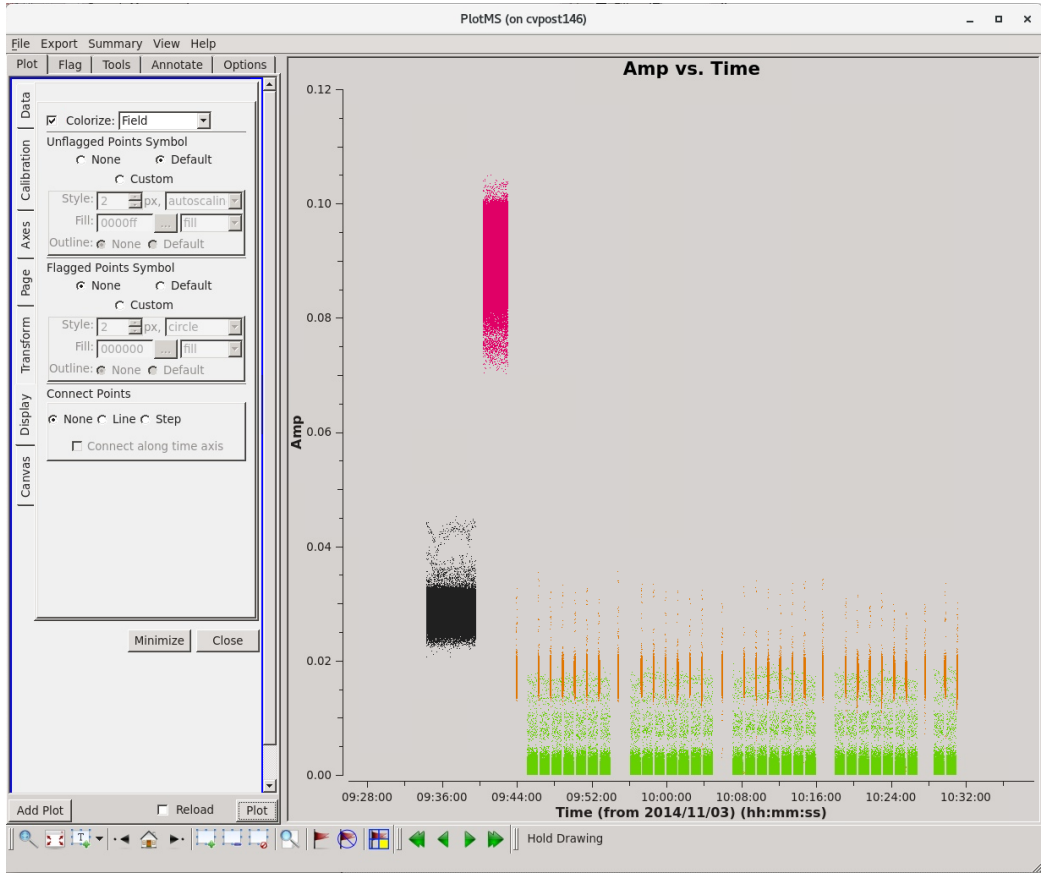
Start plotms(). We are going to make a plot!

We want to look at SDP81_B4_uncalibrated.ms.split, plotting amplitude vs time. WE are going to average the data by channel, and color the points by field.

1. In the side tab labeled “Data”, click on browse and select the file measurement set.
2. In the side tab labeled “Data”, check the box for Channel averaging. Then enter a number that averages all the channels (in an spw) together. (Look at your listobs output!)
3. In the side tab labeled “Axis”, select the axis to be Amplitude vs Time. Make sure the data column is “data”.
4. In the side tab labeled “Display” select “Colorize” and choose to colorize by field.

Question: Plotms

- Did you face difficulty when choosing a channel number for the averaging?
- From the graph, can you tell what source is the science target?
- Did you look at some of the options? What were some?
  - Axis options?
  - Colorize options?

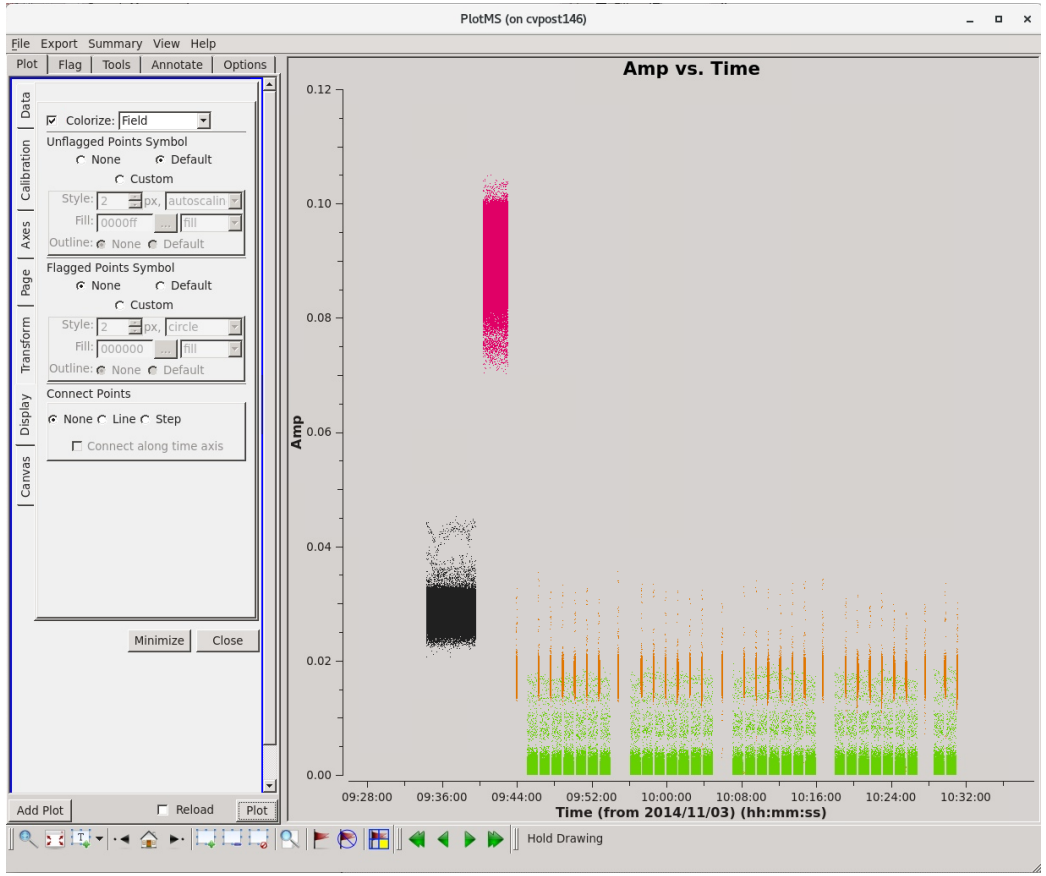


Expected plotms output

Your Turn:

- Make a plot using plotms of the uncalibrated data using a different combination of axis. Ask an assistant if you need assistance!

- What combo did you use? Did you run into any difficulties.



Calibration

Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

Gain Calibration

Setting the Flux Scale

Applying the Calibration

Checking the Calibration

Renormalization Correction

We are finally ready to discuss calibration!

Derive Calibration Tables

- setjy: set “model” (correct) visibilities using known model for a calibrator
- bandpass: calculate bandpass calibration table (amp/phase vs frequency)
- gaincal: calculate temporal gain calibration table (amp/phase vs time)
- fluxscale: apply absolute flux scaling to calibration table from known source

Manipulate Your Measurement Set

- flagdata/flagcmd/flagmanager: flag (remove) bad data
- applycal: apply calibration table(s) from previous steps
- split: split off calibrated data from your ms

Inspect Your Data and Results

- plotms: inspect your data and calibration tables interactively

Calibration

Before proceeding, we want to save the flagging of our data. This is good practice in case a mistake is made later.

Only run this once. The if statement before this in **calibration.py** will ensure it is not run twice.

```
flagmanager(vis = 'SDP81_B4_uncalibrated.ms.split',
  mode = 'save',
  versionname = 'Original')
```

Later, if you want to restore to the original flagging state, we can restore the flags to this original state using:

From **calibration.py**

```
flagmanager(vis = 'SDP81_B4_uncalibrated.ms.split'
    mode = 'restore',
    versionname = 'Original')
```

Calibration

Initial Flagging includes data we know to be problematic even without visual inspection:

Shadowing

- Issue at low elevations
- Issue for compact arrays

In CASA:

```
flagdata(vis = 'SDP81_B4_uncalibrated.ms.split’,
  mode = 'shadow’,
  flagbackup = False)
```

Observing Log

- Many observatories will note weather or hardware problems that affect the data.

Other obvious errors

- Edge Channels

In CASA

```
flagdata(vis='SDP81_B4_uncalibrated.ms.split',
  mode=‘manual’,
  spw = '0:0~3;60~63,
  1:0~3;60~63,2:0~3;60~63,3:0~7;1912~1919',
  Flagbackup = False)
```

Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

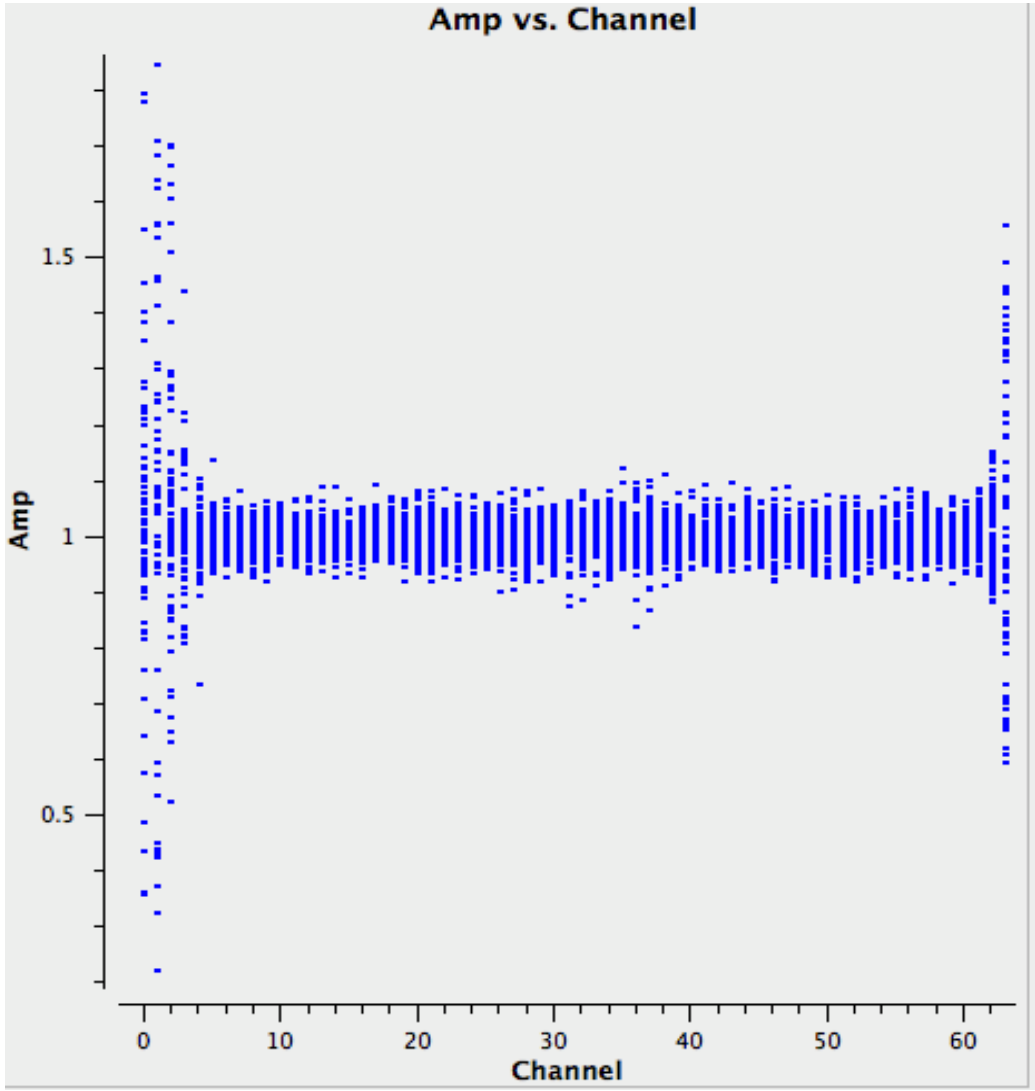
Gain Calibration

Setting the Flux Scale

Applying the Calibration

Checking the Calibration

Renormalization Correction



Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

Gain Calibration

Setting the Flux Scale

Applying the Calibration

Checking the Calibration

Renormalization Correction

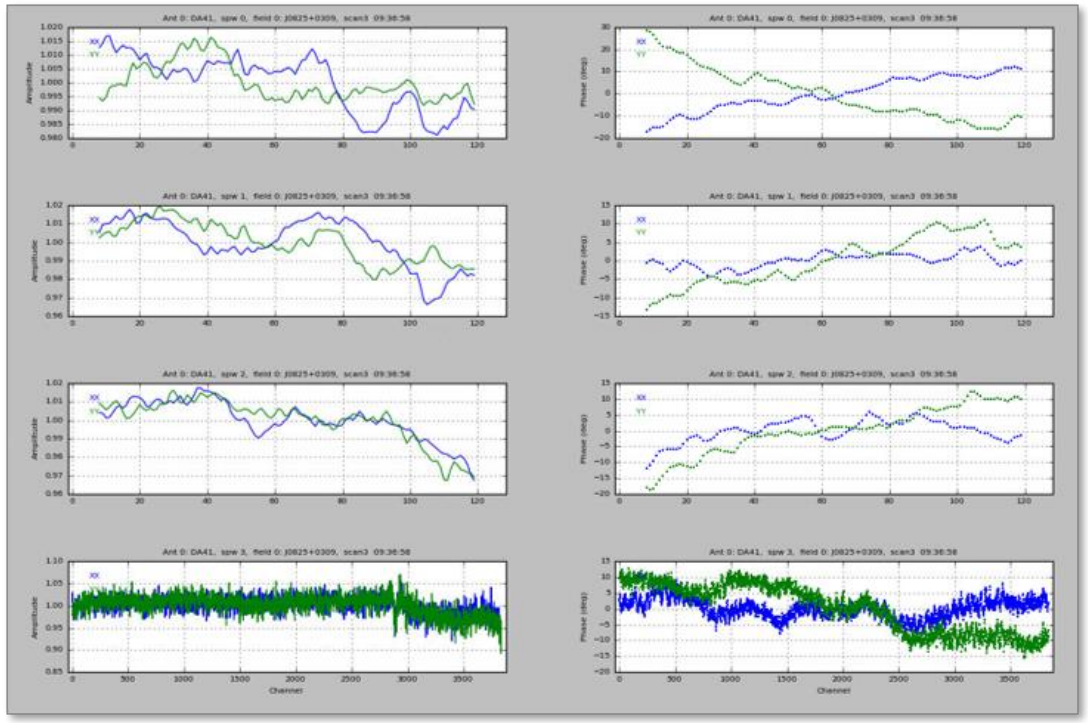
Your first time though the data, no other flags are applied.

After the rest of the calibration, if you identify any bad data, flags are placed in this part of the script.

So we will skip this for now.

Question: Bandpass

Why is bandpass calibration important?



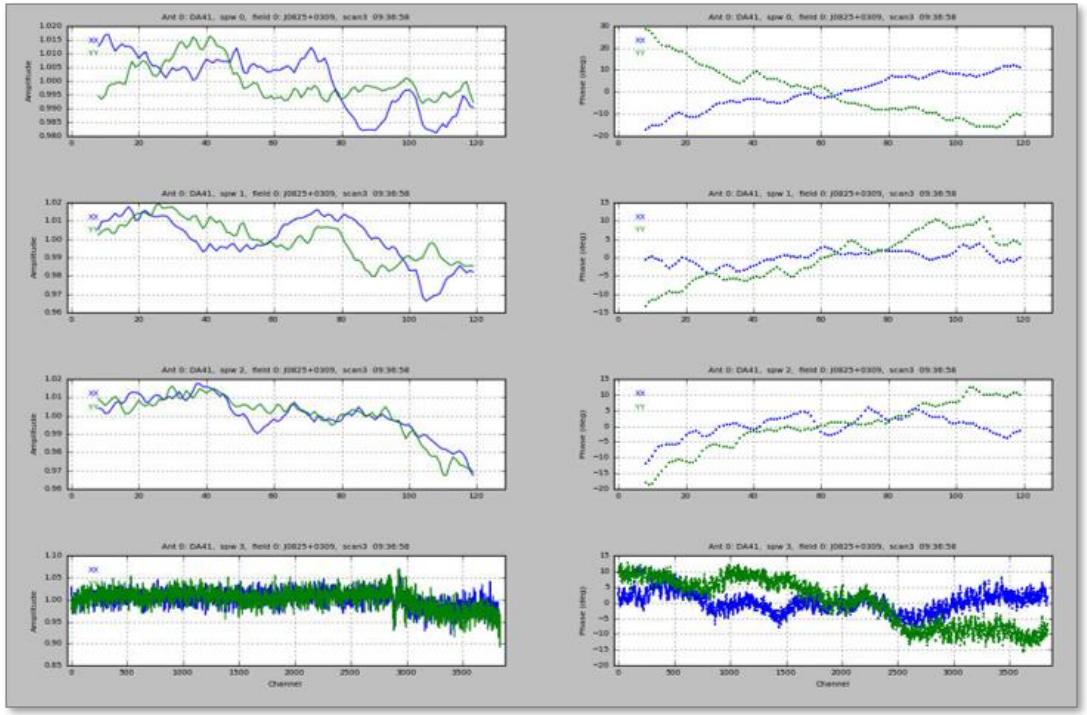
A sample of bandpass solutions.

Question: Bandpass

Why is bandpass calibration important?

Bandpass Calibration allows us to correct for the frequency dependant part of the gains $\left(B_{ij}(t,\nu)\right)$.

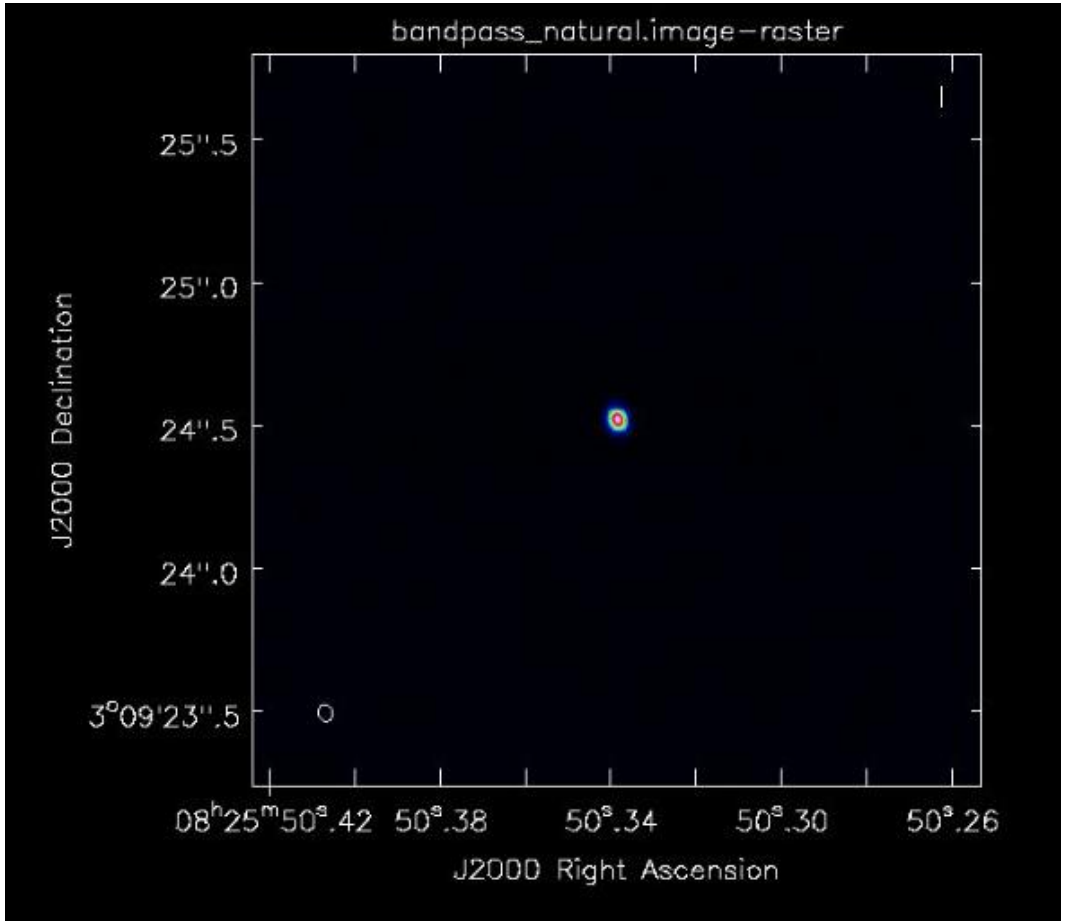
B_{ij} is usually treated as constant over the length of an observation but may have a slow time dependence.



A sample of bandpass solutions.

Question: Bandpass

What makes a good bandpass calibrator?



Bandpass Calibrator J0825+0309

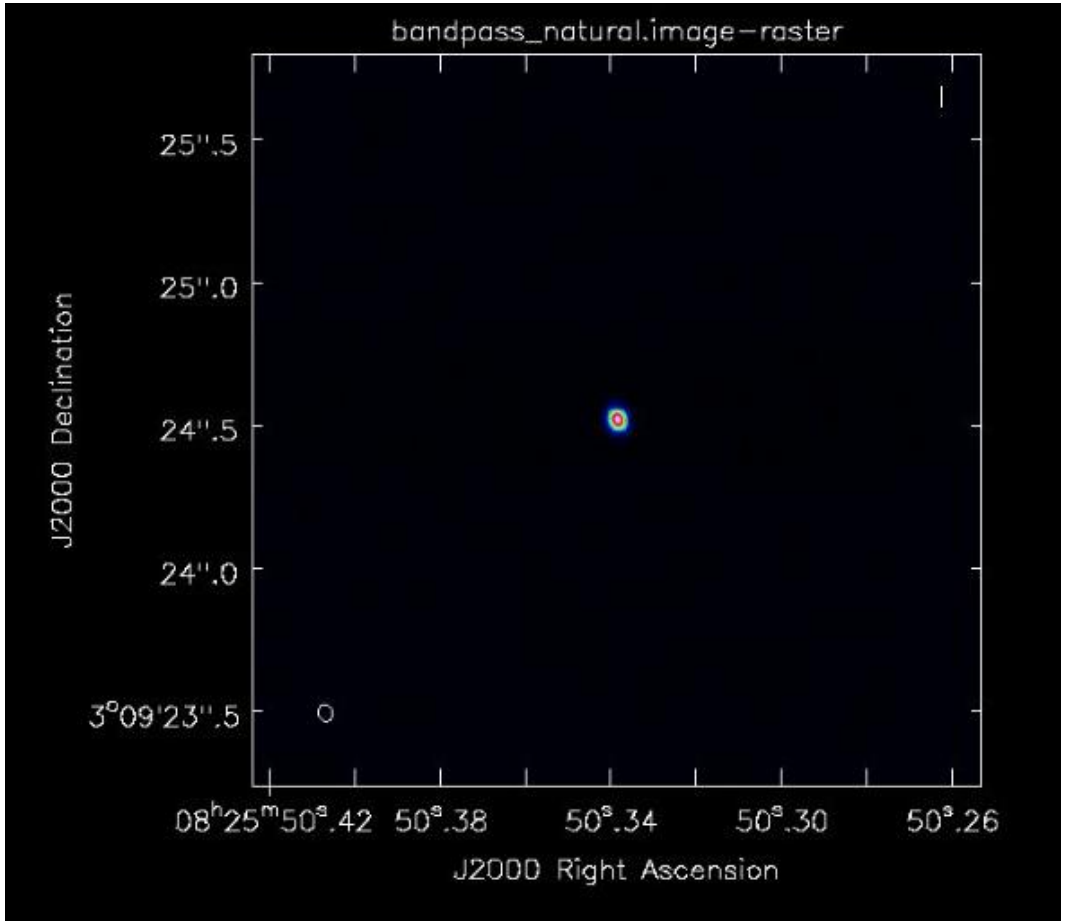
Question: Bandpass

What makes a good bandpass calibrator?

Best targets are bright, flat-spectrum sources with featureless spectra

- Although point-source not absolutely required, beware frequency dependence of resolved sources
- If necessary, can specify a spectral index using *setjy*

Don’t necessarily need to be near science target on the sky



Bandpass Calibrator J0825+0309

Question: Bandpass

Bandpass corruption is typically antenna-based.

Does this give us any advantages compared to a baseline-based calibration?

Question: Bandpass

Bandpass corruption is typically antenna-based.

Does this give us any advantages compared to a baseline-based calibration?

Yes, this means we have N-1 baselines to consider. The problem is over-determined.

Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

Gain Calibration

Setting the Flux Scale

Applying the Calibration

Checking the Calibration

Renormalization Correction

- We will use *gaincal* to measure time variation of phase
- Then use *bandpass* task
  - We will calibrate channel-to-channel variation (preferred method)
  - Alternatively, could fit a smooth function
  - Pay close attention to solutions; e.g. bright calibrators are rare, esp. at Band 9
- Use *applycal* to apply the bandpass solution to other sources

Calibration

Using the Listobs output, determine which source is the bandpass.

- This is J0825+0309 (identified as field 0).

Run the following command:

```
gaincal(vis="SDP81_B4_uncalibrated.ms.split",
    caltable="phase_int_bpass.cal",
    field="0",
    scan="3",
    solint="int",
    refant="DA56",
    calmode="p")
```

Calibration

While gaincal runs:

Gaincal is the general purpose task to solve for time-dependent amplitude and phase variations for each antenna.

Here we carry out a short-timescale phase (“p”) solution ("int") on the bandpass calibrator. This is saved as a calibration table "phase_int_bpass.cal".

We only fit the phase to the central channels to avoid edge channel effects.

The Field and scan designations are redundant, but good to make sure the correct data for the solve.

Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

Gain Calibration

Setting the Flux Scale

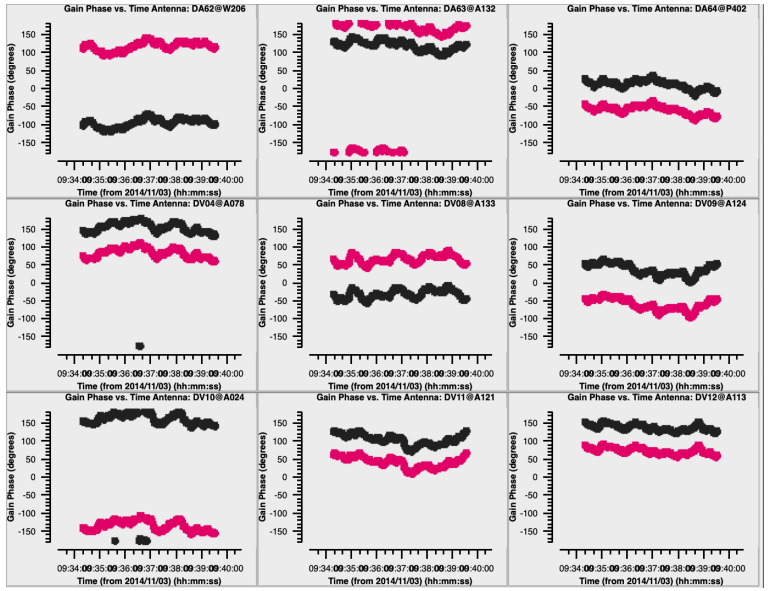
Applying the Calibration

Checking the Calibration

Renormalization Correction

Plot the solutions using plotms!

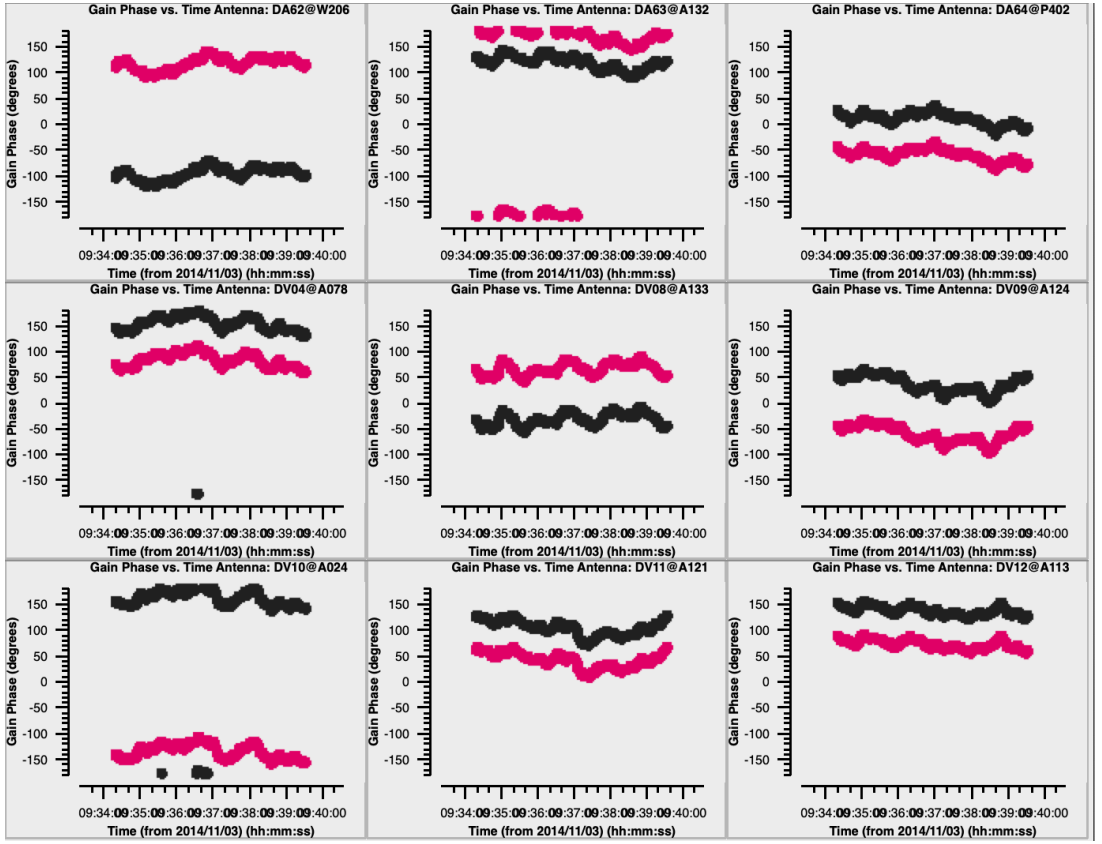
- Plot the phase_int_bpass.cal table in phase vs time. Spw 0 only. Color by correlation.
- Under the side tab labeled Axis, set the range on phase to be -180 to +180 degrees.
- New this time: iteration axis! Under the page tab, select axis => antenna
- Under the top tab labeled “Options”, set rows and columns both to three.
- Use the tool panel buttons  to navigate the plots.



Question: Phases

A few plots stand out in our phase solutions.

- Why does the phase plot for DA56 have the shape it has?
- What is the cause of the discontinuities for DA61?



Good phase plots for the bandpass.

Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

Gain Calibration

Setting the Flux Scale

Applying the Calibration

Checking the Calibration

Renormalization Correction

Now we will carry out a bandpass solution. This will solve for the amplitude and phase corrections needed for each channel on a per-antenna basis.

Lets run the bandpass task:

```
bandpass(vis = 'SDP81_B4_uncalibrated.ms.split’,
    caltable = 'bandpass.cal',
    field = '0', # J0825+0309,
    spw = '0:22~42, 1:22~42,2:22~42,3:800~1200',
    solint = 'inf',
    scan = '3',
    combine = 'scan',
    refant = 'DA56',
    solnorm = True,
    bandtype = 'B',
    gaintable = 'phase_int_bpass.cal')
```

Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

Gain Calibration

Setting the Flux Scale

Applying the Calibration

Checking the Calibration

Renormalization Correction

While that is running:

- We use gaintable to feed the short-timescale phase solution to the task.
  - This table will be applied before the bandpass solution is carried out.
- We have not flux calibrated yet, so we are determining a normalized solution with solnorm=True.
- Solint=inf and combine=scan combines all data.
- Bandtype = B is the standard

Question: Bandpass

Why is it okay to combine all the bandpass data together?

Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

Gain Calibration

Setting the Flux Scale

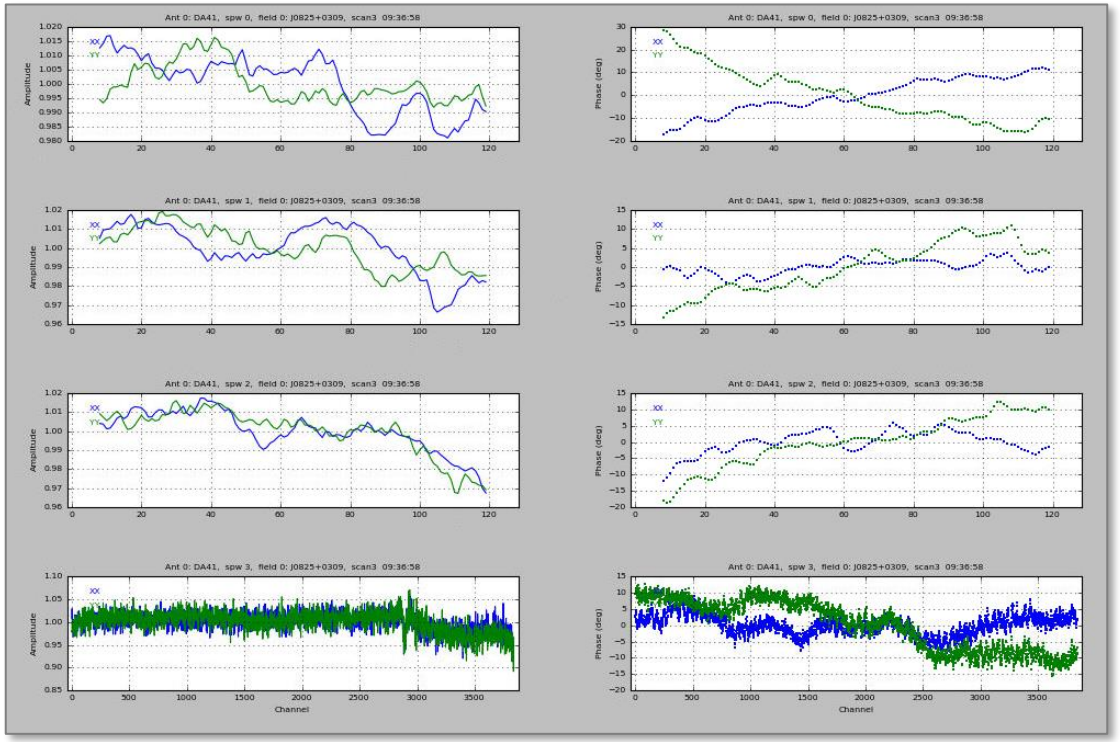
Applying the Calibration

Checking the Calibration

Renormalization Correction

We are going to use the plotbandpass() task to plot the bandpass solutions.

```
plotbandpass(caltable="bandpass.cal",
    xaxis="chan",
    yaxis="both",
    subplot=42)
```



Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

Gain Calibration

Setting the Flux Scale

Applying the Calibration

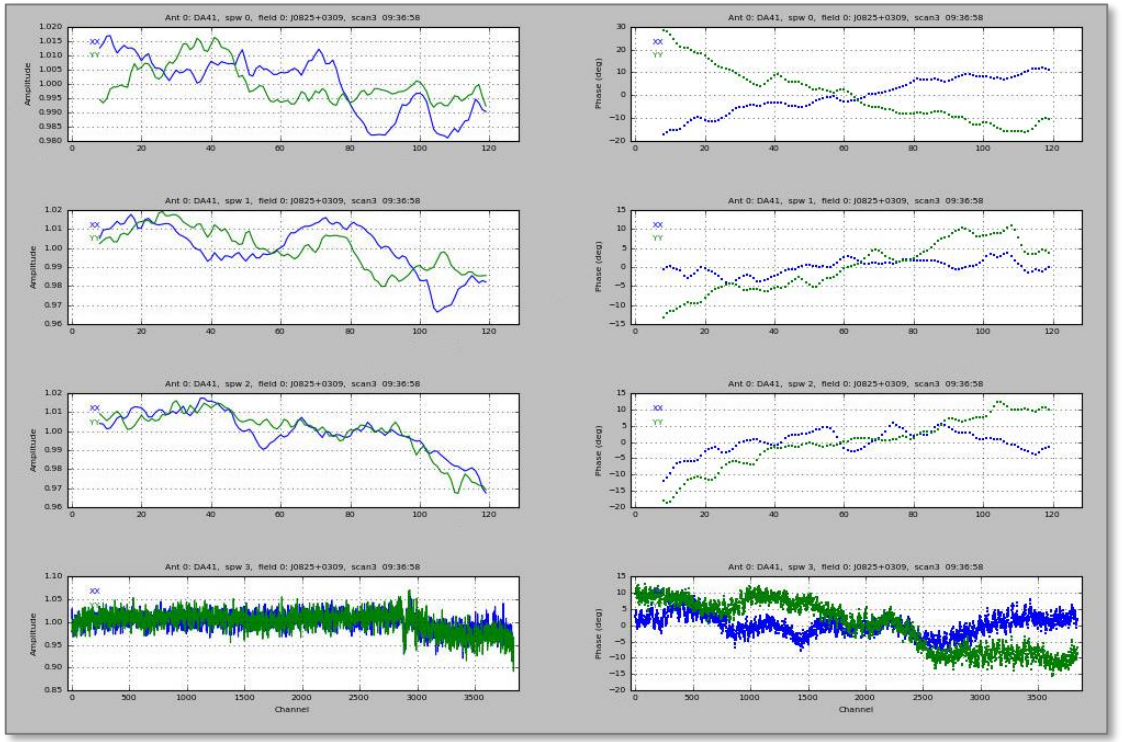
Checking the Calibration

Renormalization Correction

What are we looking for in a bandpass solution?

The bandpass solution should be smooth. Some features at the location of atmospheric lines are expected.

Keep an eye out for discontinuities.. Standard practice is to flag whole antenna if you find something wrong.



Calibration

SPW 3 has a lot of noise. This is because of the increased spectral resolution.

Lets redo the bandpass calibration, with smoothing for spw 3.

Delete your existing bandpass table.

Redoing Bandpass Calibration

SPW 0, 1, and 2

```
bandpass(vis =
'SDP81_B4_uncalibrated.ms.split',
   caltable = 'bandpass.cal',
   field = '0', # J0825+0309
   spw = '0,1,2',
   solint = 'inf',
   scan = '3',
   combine = 'scan',
   refant = 'DA56',
   solnorm = True,
   bandtype = 'B',
   gaintable =
   'phase_int_bpass.cal')
```

SPW 3

```
bandpass(vis =
'SDP81_B4_uncalibrated.ms.split',
   caltable =
   'bandpass_smooth.cal',
   field = '0', # J0825+0309
   spw = '3',
   scan = '3',
   solint = 'inf,5ch',
   combine = 'scan',
   refant = 'DA56',
   solnorm = True,
   bandtype = 'B',
   append = True,
   gaintable =
   'phase_int_bpass.cal')
```

Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

Gain Calibration

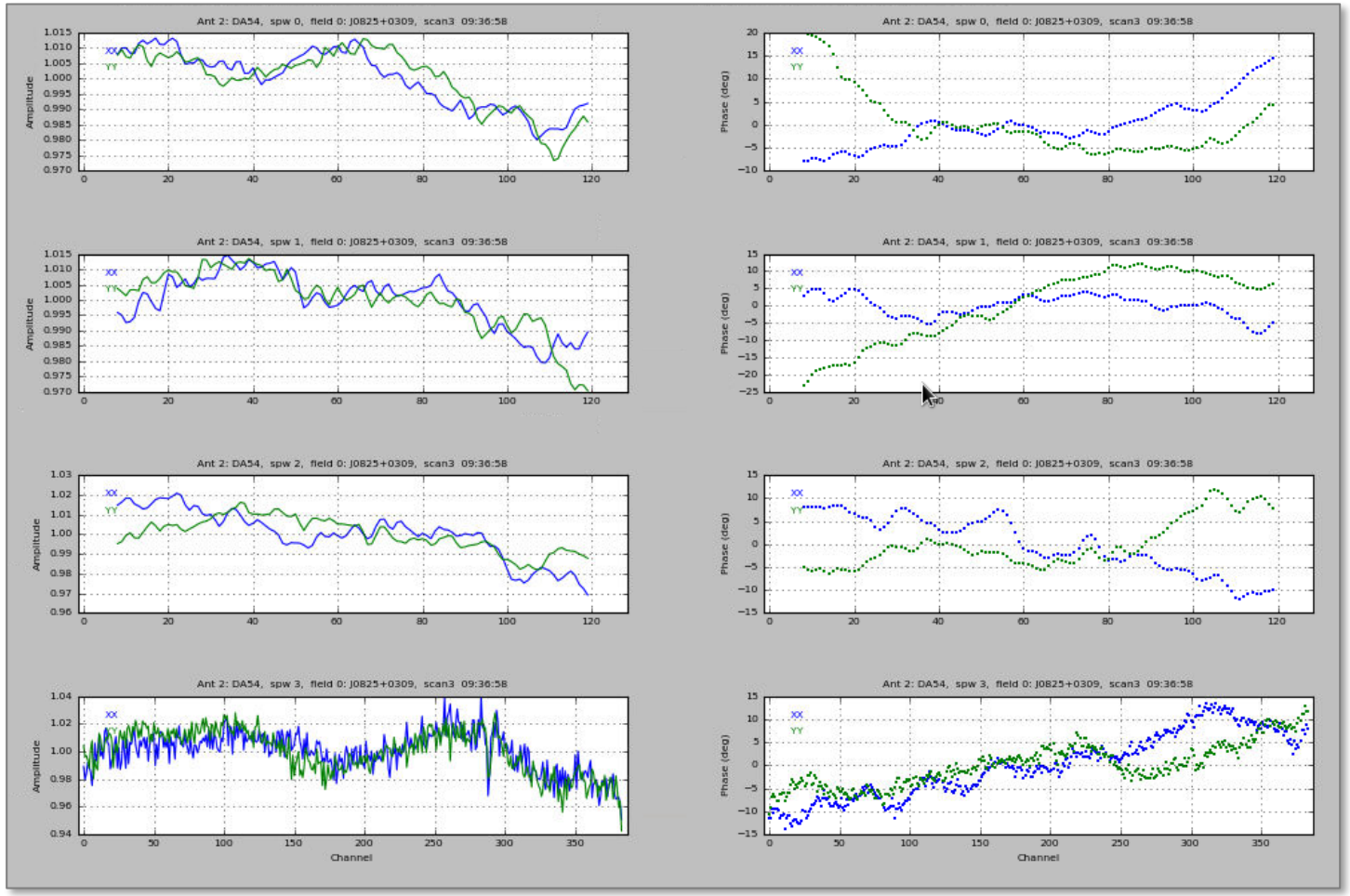
Setting the Flux Scale

Applying the Calibration

Checking the Calibration

Renormalization Correction

Use plotbandpass to plot the new table.



Calibration

To check the bandpass solution, we need to apply the corrections.

```
applycal(vis="SDP81_B4_uncalibrated.ms.split",
    field="0",
    gaintable=["bandpass_smooth.cal",
    "phase_int_bpass.cal"],
    interp=["linear","linear"],
    gainfield=["0","0"],
    applymode='calonly’)
```

We will talk about this command while it runs!

Calibration

To check the bandpass solution, we need to apply the corrections.

```
applycal(vis="SDP81_B4_uncalibrated.ms.split",
    field="0",
    gaintable=["bandpass_smooth.cal",
    "phase_int_bpass.cal"],
    interp=["linear","linear"],
    gainfield=["0","0"],
    applymode='calonly’)
```

We will talk about this command while it runs!

Field is telling applycal to only apply to the bandpass calibrator.
Gaintable is the list of calibration tables we want to use.
Interp is telling what type of interpolation we use for each table.
Gainfield tells what source solution we will apply.

Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

Gain Calibration

Setting the Flux Scale

Applying the Calibration

Checking the Calibration

Renormalization Correction

Open plotms.

- select SDP81_B4_uncalibrated.ms.split,
- Plot phase on the Y axis and use the data column for phase.
- Plot channel on the X axis.
- Select the bandpass calibrator (field = 0), spw=3, and Average over the whole scan.
- Color by correlation.

Make some notes about what you see. Then switch the phase column from data to corrected. Remake the plot, and check the results.

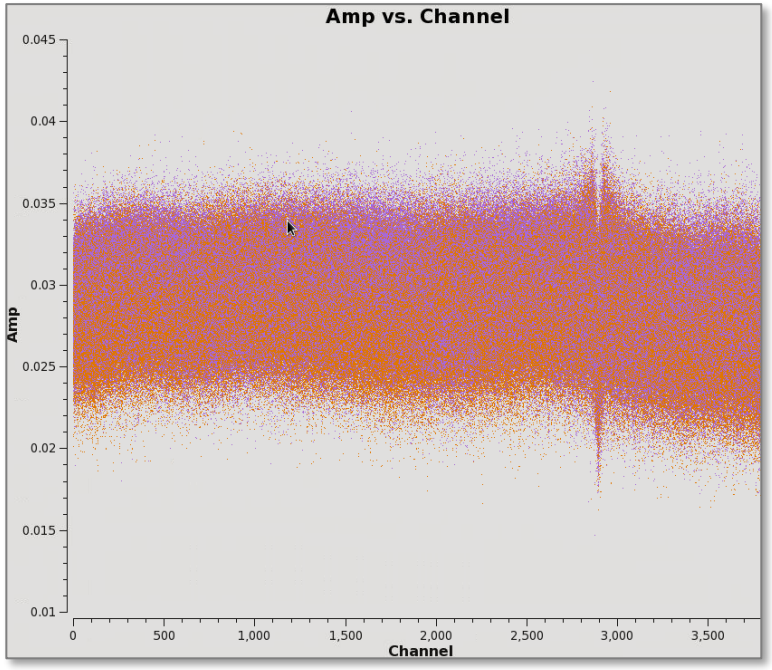
Remake the plot again using amplitude instead of phase. Compare the data column to the corrected column data.

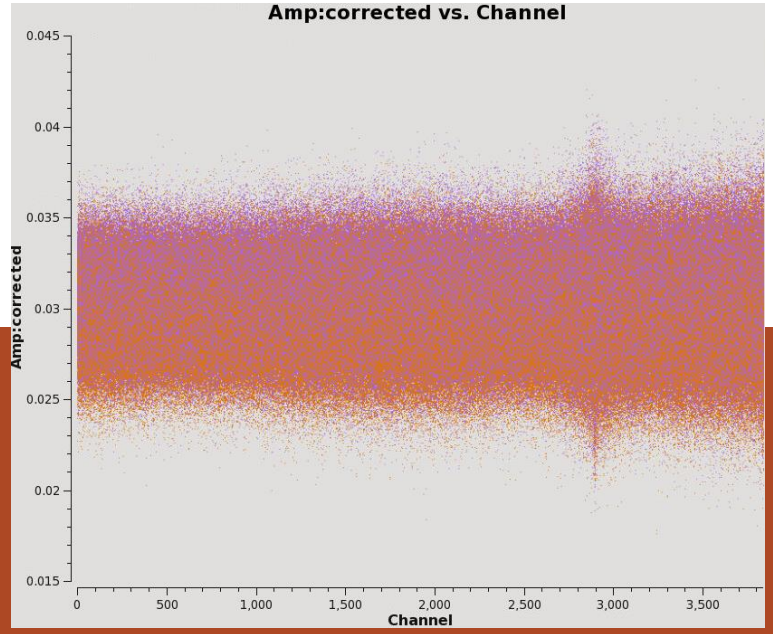
Question: Bandpass

- What did the bandpass calibration do to the data?
- Are there any features you are concerned about?

Question: Bandpass

- What did the bandpass calibration do to the data?
- Are there any features you are concerned about?





Calibration

After successfully determining the bandpass solutions, we want to set the model for our flux calibrator. Our flux reference source is a quasar J0854+2006 (field 1).

We use a routine to parse the ALMA calibrator database, interpolate the expected flux for the calibrator reference, and put in the 'model' column of the data using setjy.

```
aU.getALMAFluxForMS("SDP81_B4_uncalibrated.ms.split")
```

```
'J0854+2006': {'fluxDensity': 3.986837001941747,
 'fluxDensityUncertainty': 0.09682730818345603,
 'frequency': 149593012274.14886,
 'spectralIndex': -0.456158812091857 5,
 'spectralIndexUncertainty': 0.033431828095445 4,
 'meanAge': 22.0,
 'monteCarloFluxDensity': 3.614815251056603,
 'spectralIndexAgeSeparation': 0,
 'spectralIndexAgeOldest': -27,
 'spectralIndexAgeYoungest': -27,
 'spectralIndexNPairs': 1},
```

Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

Gain Calibration

Setting the Flux Scale

Applying the Calibration

Checking the Calibration

Renormalization Correction

Now run setjy to set the model.

```
setjy(vis = 'SDP81_B4_uncalibrated.ms.split',
    standard = 'manual',
    field = '1',#J0854+2006
    fluxdensity = [3.986837, 0, 0, 0],
    reffreq = '149.593012274GHz',
    spix = -0.456158812)
```

```
'J0854+2006': {'fluxDensity': 3.986837001941747,
 'fluxDensityUncertainty': 0.09682730818345603,
 'frequency': 149593012274.14886,
 'spectralIndex': -0.456158120918575,
 'spectralIndexUncertainty': 0.033431828095445 4,
 'meanAge': 22.0,
 'monteCarloFluxDensity': 3.61481525105 6603,
 'spectralIndexAgeSeparation': 0,
 'spectralIndexAgeOldest': -27,
 'spectralIndexAgeYoungest': -27,
 'spectralIndexNPairs': 1},
```

Calibration

Gain Calibration: Long-term phase solutions

First, we calibrate the phase for each antenna for each scan. This is the right cadence to transfer to the science target, which is visited only on a every-other-scan timescale.

```
gaincal(vis=”SDP81_B4_uncalibrated.ms.split",
      caltable="phase_inf.cal",
      field=”0~2",
      solint="inf",
      refant=”DA56”,
      gaintype=“G”,
      calmode=“p”,
      gaintable=“bandpass_smooth.cal”)
```

Question: Gain Calibrators

- What makes a good gain (phase) calibrator?

Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

Gain Calibration

Setting the Flux Scale

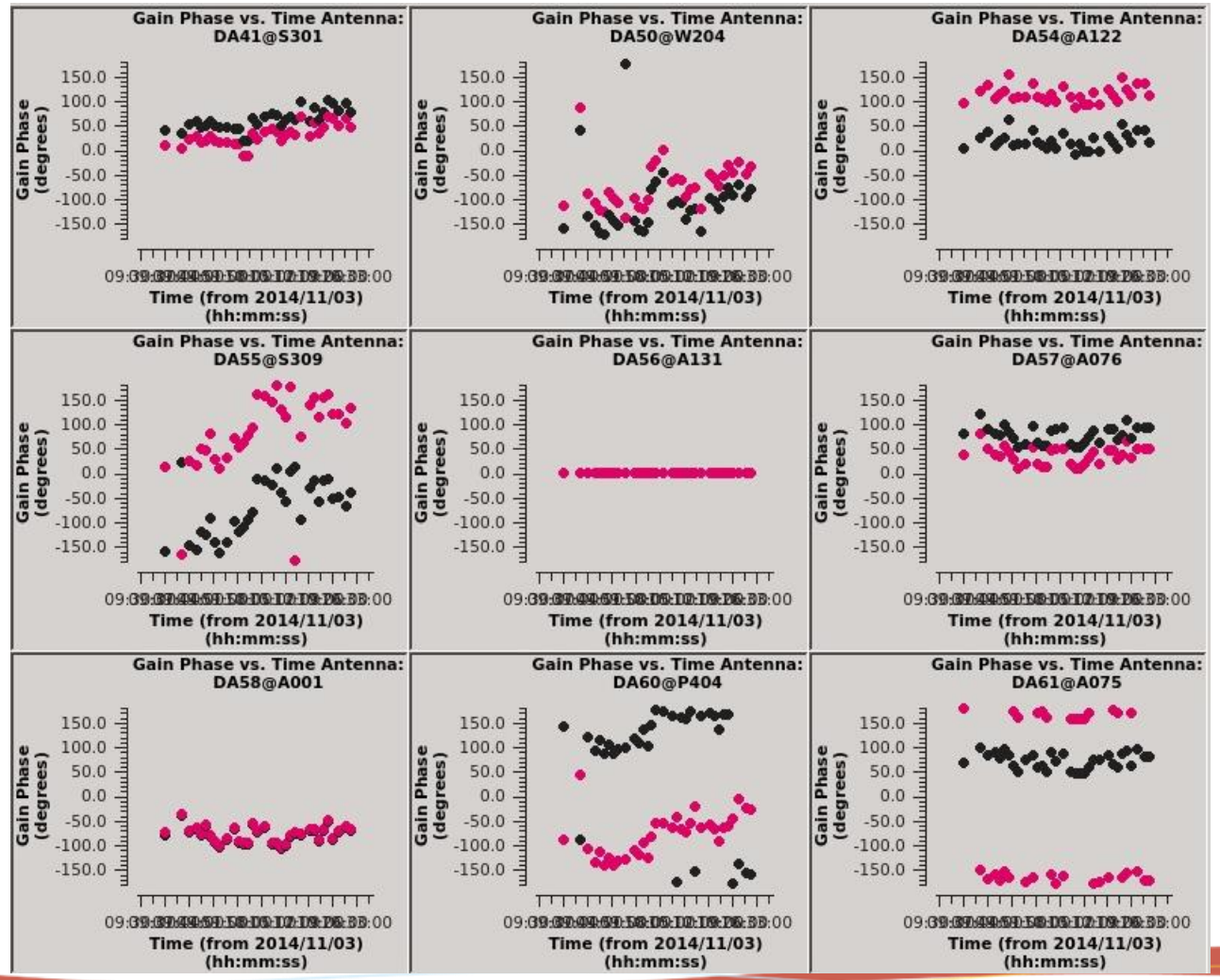
Applying the Calibration

Checking the Calibration

Renormalization Correction

Plot the resulting phase calibration

Use the plotms commands in calibration.py



Calibration

Gain Calibration: Short-term Phase Solutions

Now we want to remove any short timescale phase variation from the sources involved in the bandpass and flux calibration. We do so using gaincal.

```
gaincal(vis = 'SDP81_B4_uncalibrated.ms.split',
    caltable = 'phase_int.cal',
    field = '0~2', #
    J0825+0309,J0854+2006,J0909+0121
    solint = 'int',
    refant = 'DA56',
    gaintype = 'G',
    calmode = 'p',
    gaintable = 'bandpass_smooth.cal')
```

Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

Gain Calibration

Setting the Flux Scale

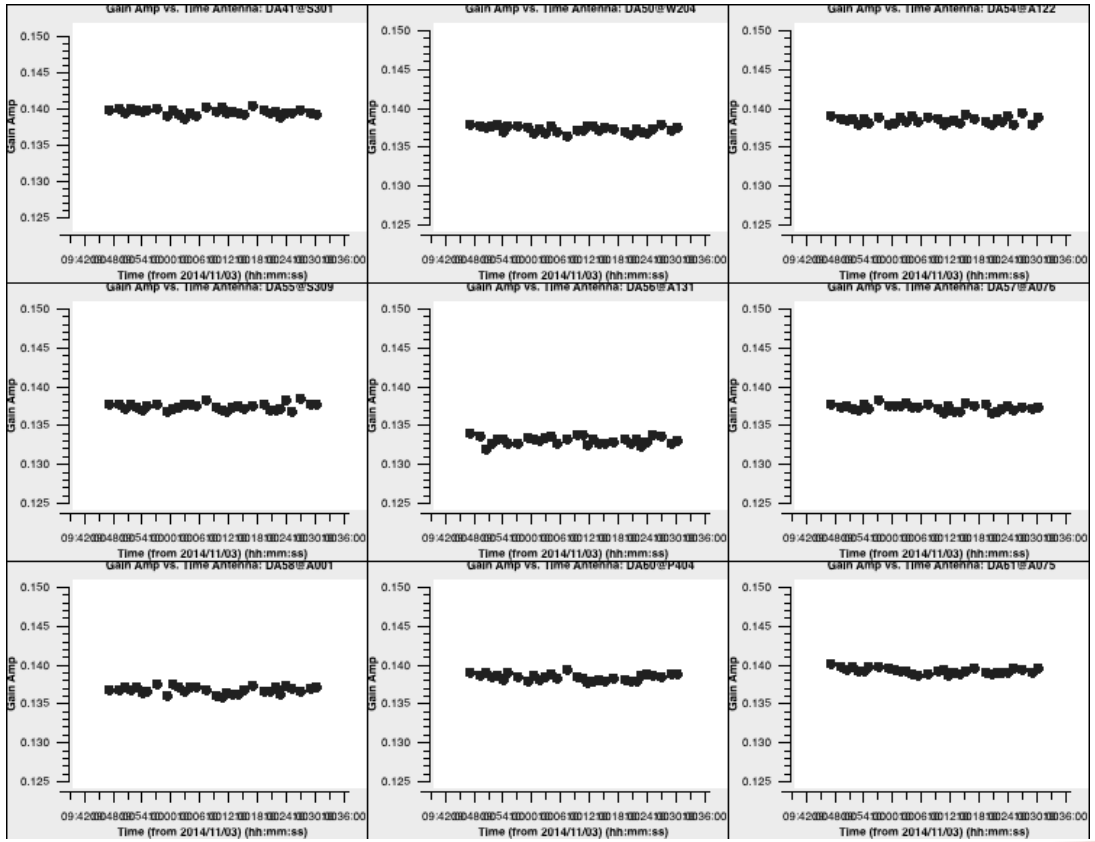
Applying the Calibration

Checking the Calibration

Renormalization Correction

Plot the resulting short timescale phase calibration.

Just like before, you need to run the four plotms commands from your calibration.py script.



Calibration

Gain Calibration: Amplitude Solutions

To create long term amplitude corrections, we will use the calmode = ‘a’ option in gaincal.

```
gaincal(vis = 'SDP81_B4_uncalibrated.ms.split',
    caltable = 'ampli_inf.cal',
    field = '0~2',
    solint = 'inf',
    refant = 'DA56',
    gaintype = 'T',
    calmode = 'a',
    gaintable = ['bandpass_smooth.cal',
    'phase_int.cal'])
```

Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

Gain Calibration

Setting the Flux Scale

Applying the Calibration

Checking the Calibration

Renormalization Correction

The gaincal solved for the amplitude scaling to make the data match the current model.

For the quasar J0854+2006, we have taken care to set the correct model using setjy.

For the other two calibrators, however, we don't a priori know the flux.

Those have been calibrated using the default model, which is a point source of amplitude 1 Jy at the middle of the field.

We now use fluxscale to bootstrap from the (correct) flux of the quasar through the amplitude calibration table to estimates of the true flux of the other two calibrators.

This will output both a new table and the flux estimates themselves.

Run the first four lines of the “Setting the flux scale” section in calibration.py

Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

Gain Calibration

Setting the Flux Scale

Applying the Calibration

Checking the Calibration

Renormalization Correction

```
fluxscaleDict = fluxscale(vis =
'SDP81_B4_uncalibrated.ms.split',
    caltable = 'ampli_inf.cal',
    fluxtable = 'flux_inf.cal',
    reference = '1') # J0854+2006

casalog.setlogfile(mylogfile) #Set output back to
logger
```

Afterwards, plot the solutions with plotms.

```
plotms(vis="flux_inf.cal",
    xaxis="time",
    yaxis="amp",
    gridcols=3,
    gridrows=3,
    iteraxis="antenna",
    plotrange=[0,0,0.13,0.18],
    symbolsize=10,
    plotfile="ss20_flux_scan.png")
```

Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

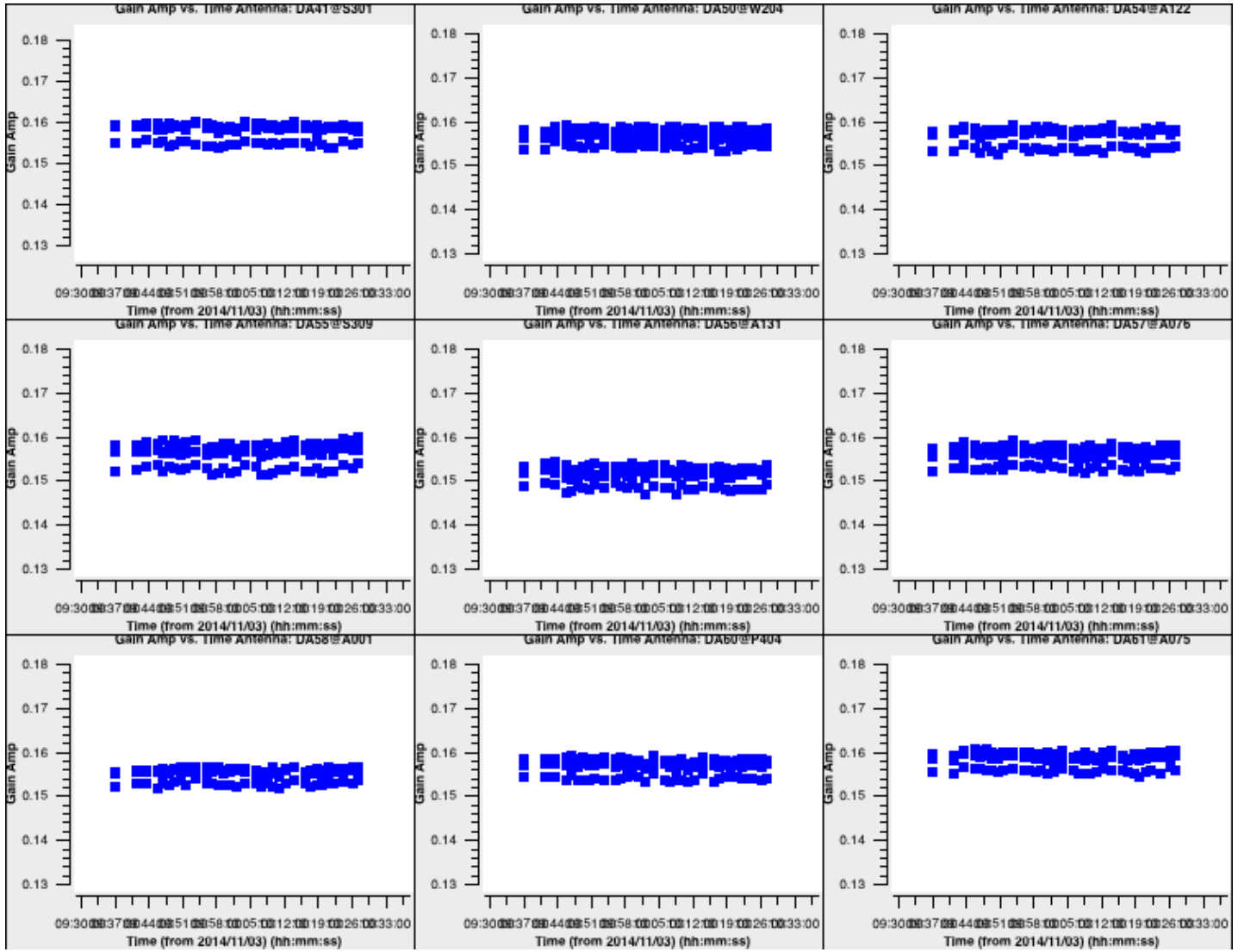
Gain Calibration

Setting the Flux Scale

Applying the Calibration

Checking the Calibration

Renormalization Correction



Calibration is finished!

- We have now created all the calibration tables.
- All we need to do now is apply the calibration.

Calibration

For our bandpass and flux calibrators (fields 0 & 1), we apply our bandpass calibration and our gain calibration (short term phase + flux).

For our science target and phase calibrator (fields 2 & 3), we apply our bandpass calibration and our gain calibration (long term phase + flux).

Before applying the calibration, we want to save the flags.

```
flagmanager(vis='SDP81_B4_uncalibrated.ms.split’,
mode='save’,
versionname='BeforeApplycal')
```

Calibration

Now we can apply the calibration to the bandpass and flux calibrators:

```
for i in ['0', '1']: # J0825+0309,J0854+2006
applycal(vis = 'SDP81_B4_uncalibrated.ms.split’,
    field = str(i),
    gaintable=['bandpass_smooth.cal','phase_int.cal',
       'flux_inf.cal’]
    gainfield = ['', i, i],
    interp = 'linear,linear’,
    calwt = True,
    flagbackup = False)
```

This loops through the bandpass and flux calibrators and applies the calibration to them. While this runs, lets talk about what the options change.

Calibration

Finally, we are going to apply the calibration to the phase calibrator (field='2') to the science field (field='3')

```
applycal(vis = 'SDP81_B4_uncalibrated.ms.split',
    field = '2,3', # SDP.81
    gaintable = ['bandpass_smooth.cal',
        'phase_inf.cal', 'flux_inf.cal'],
    gainfield = ['', '2', '2'], # J0909+0121
    interp = 'linear,linear',
    calwt = True,
    flagbackup = False)
```

We are interpolating linearly across the solutions from the phase calibrator and applying them to the science field.

Calibration

- Next, we will review the basics of data inspection and flagging.
- We will now use plotms to make a series of diagnostic plots. These plots have been picked because we have a good expectation of what the calibrators (fields 0, 1, and 2 here) should look like in each space.
- Typical plots for calibrated data inspection are the following:
  - Amplitude vs time
  - Phase vs time
  - Amplitude vs frequency
  - Phase vs frequency
  - Amplitude vs UV distance
  - Phase vs UV distance
- Iterate over Antenna, Spectral Window, or source
- Typically, start by plotting the calibrators.
  - Be careful flagging the target, as you could be flagging out real information on the target.

We will go through the plots in the order they are listed in calibration.py

Calibration

Flagging with plotms

It is possible to flag within the GUI with plotms. This is done by boxing the data with the  button, and then flagging with the  button.

Unflagging can be performed by plotting flagged points and using the  button.

<u>FLAGGING USING PLOTMS IS NOT ENCOURAGED!!!</u>

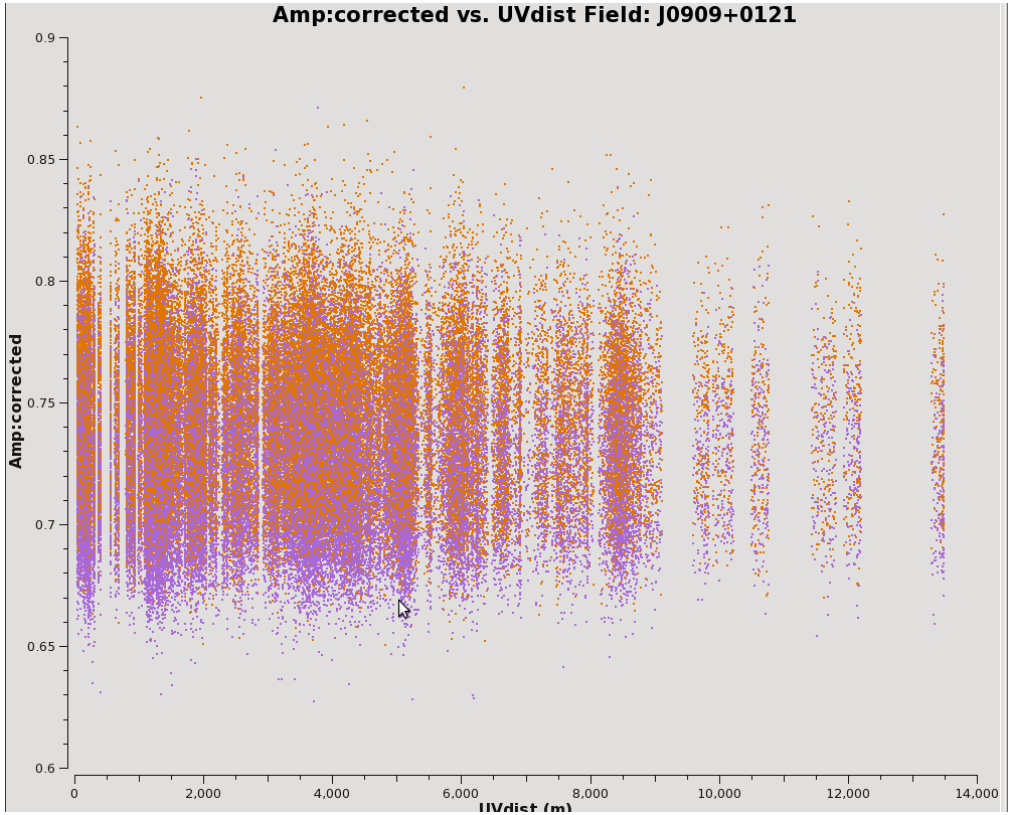
Flagging with plotms is not considered reproduceable. Due to the nature of drawing boxes in plotms, it is difficult to tell what points were flagged.

It is best to use plotms to identify bad data, and then flag using flagdata.

Question: Amp vs UVdist

Look for individual baselines or antennas not following the other baselines.

What do we expect a point source at the phase center to look like in terms of Amplitude vs UV distance?



UV distance vs amplitude for Calibrator J0909+0121

Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

Gain Calibration

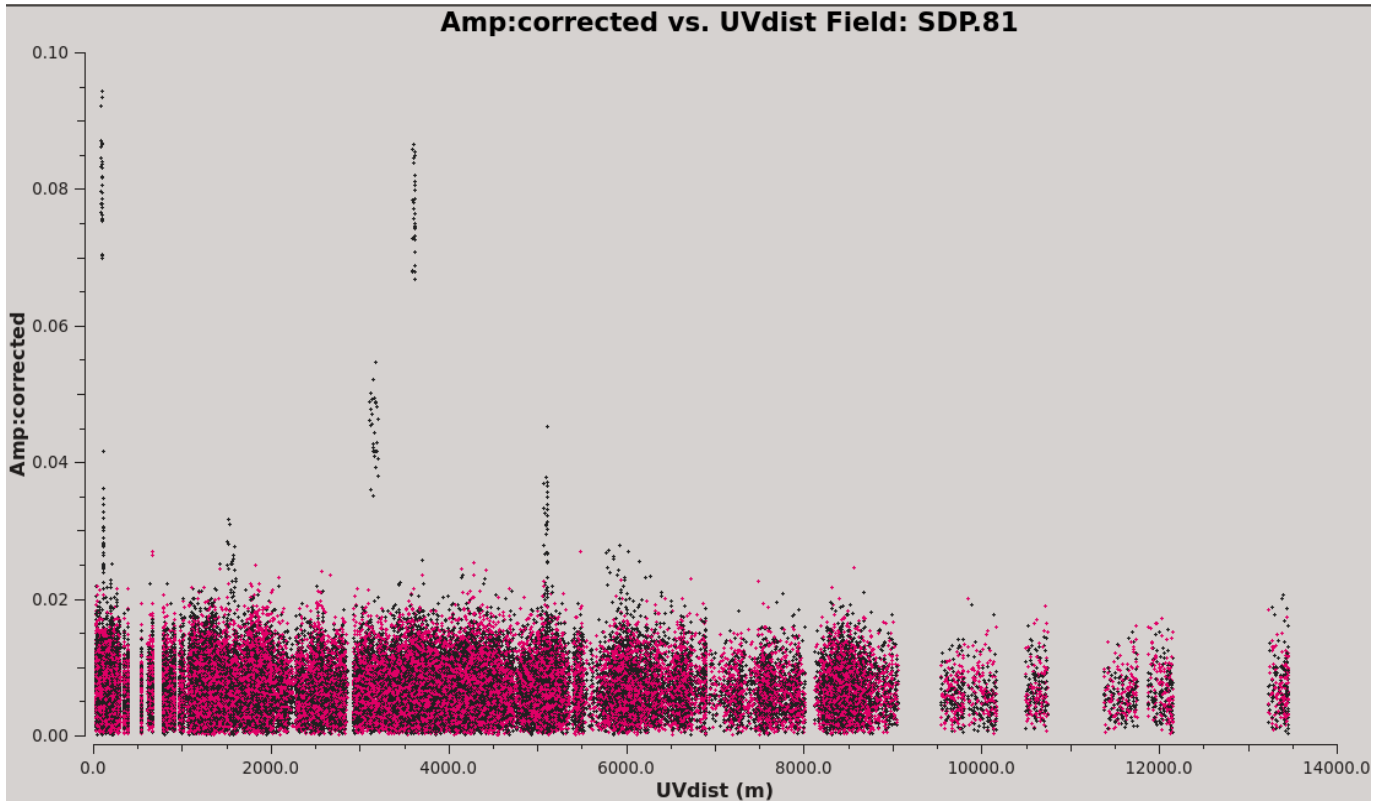
Setting the Flux Scale

Applying the Calibration

Checking the Calibration

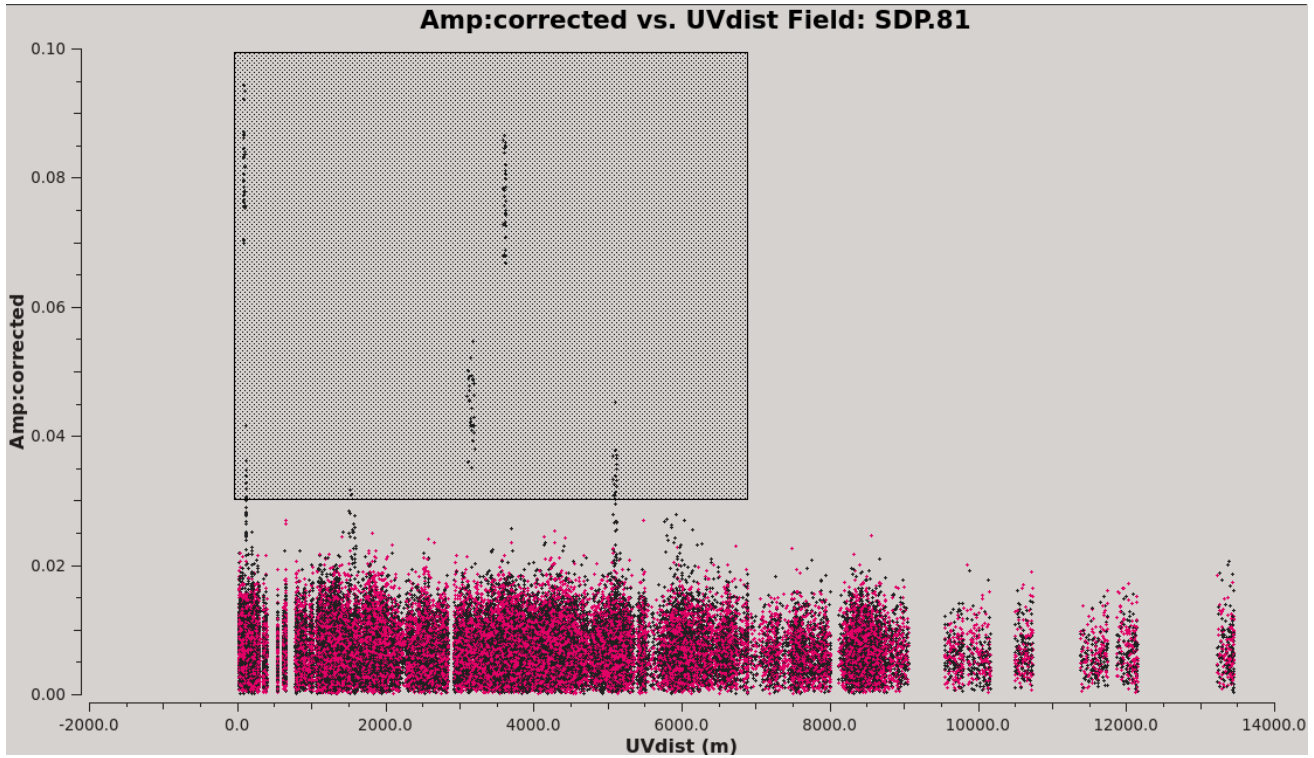
Renormalization Correction

Look at all four sources. Do you notice any points that stand out?



Calibration

- Highlight points with  and click locate ().
- What is in common?




```
Scan=59 Field=SDP.81 [3] Time=2014/11/03/10:19:48.1440 BL=DA50@W204 & PM04@T703 [1&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=5119.21 Y=0.0364852 (
Scan=59 Field=SDP.81 [3] Time=2014/11/03/10:19:48.1440 BL=DA62@W206 & PM04@T703 [9&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=3608.39 Y=0.0727789 (
Scan=59 Field=SDP.81 [3] Time=2014/11/03/10:19:48.1440 BL=DV10@A024 & PM04@T703 [15&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=99.0455 Y=0.0777865
Scan=61 Field=SDP.81 [3] Time=2014/11/03/10:21:05.8080 BL=DA41@S301 & PM04@T703 [0&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=3135.01 Y=0.0470909 (
Scan=61 Field=SDP.81 [3] Time=2014/11/03/10:21:05.8080 BL=DA50@W204 & PM04@T703 [1&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=5119.11 Y=0.0355896 (
Scan=61 Field=SDP.81 [3] Time=2014/11/03/10:21:05.8080 BL=DA62@W206 & PM04@T703 [9&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=3606.49 Y=0.0845674 (
Scan=61 Field=SDP.81 [3] Time=2014/11/03/10:21:05.8080 BL=DV10@A024 & PM04@T703 [15&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=99.1765 Y=0.0752248
Scan=61 Field=SDP.81 [3] Time=2014/11/03/10:21:05.8080 BL=DV21@A004 & PM04@T703 [22&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=119.004 Y=0.0361741
Scan=63 Field=SDP.81 [3] Time=2014/11/03/10:22:23.4720 BL=DA41@S301 & PM04@T703 [0&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=3131.89 Y=0.0453449 (
Scan=63 Field=SDP.81 [3] Time=2014/11/03/10:22:23.4720 BL=DA62@W206 & PM04@T703 [9&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=3604.52 Y=0.0779968 (
Scan=63 Field=SDP.81 [3] Time=2014/11/03/10:22:23.4720 BL=DV10@A024 & PM04@T703 [15&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=99.309 Y=0.0817332 (
Scan=65 Field=SDP.81 [3] Time=2014/11/03/10:23:41.1360 BL=DA41@S301 & PM04@T703 [0&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=3128.73 Y=0.0477873 (
Scan=65 Field=SDP.81 [3] Time=2014/11/03/10:23:41.1360 BL=DA50@W204 & PM04@T703 [1&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=5118.47 Y=0.0371042 (
Scan=65 Field=SDP.81 [3] Time=2014/11/03/10:23:41.1360 BL=DA62@W206 & PM04@T703 [9&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=3602.45 Y=0.0681095 (
Scan=65 Field=SDP.81 [3] Time=2014/11/03/10:23:41.1360 BL=DV10@A024 & PM04@T703 [15&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=99.4368 Y=0.0755574
Scan=67 Field=SDP.81 [3] Time=2014/11/03/10:24:58.8000 BL=DA41@S301 & PM04@T703 [0&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=3125.55 Y=0.0501428 (
Scan=67 Field=SDP.81 [3] Time=2014/11/03/10:24:58.8000 BL=DA62@W206 & PM04@T703 [9&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=3600.29 Y=0.0858272 (
Scan=67 Field=SDP.81 [3] Time=2014/11/03/10:24:58.8000 BL=DV10@A024 & PM04@T703 [15&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=99.5662 Y=0.0753881
Scan=67 Field=SDP.81 [3] Time=2014/11/03/10:24:58.8000 BL=DV21@A004 & PM04@T703 [22&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=118.901 Y=0.0346102
Scan=69 Field=SDP.81 [3] Time=2014/11/03/10:26:16.4640 BL=DA41@S301 & PM04@T703 [0&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=3122.34 Y=0.0359895 (
Scan=69 Field=SDP.81 [3] Time=2014/11/03/10:26:16.4640 BL=DA62@W206 & PM04@T703 [9&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=3598.06 Y=0.0727033 (
Scan=69 Field=SDP.81 [3] Time=2014/11/03/10:26:16.4640 BL=DV10@A024 & PM04@T703 [15&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=99.6965 Y=0.0839206
Scan=73 Field=SDP.81 [3] Time=2014/11/03/10:29:08.7360 BL=DA41@S301 & PM04@T703 [0&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=3115.13 Y=0.0487719 (
Scan=73 Field=SDP.81 [3] Time=2014/11/03/10:29:08.7360 BL=DA62@W206 & PM04@T703 [9&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=3592.79 Y=0.0783596 (
Scan=73 Field=SDP.81 [3] Time=2014/11/03/10:29:08.7360 BL=DV10@A024 & PM04@T703 [15&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=99.9755 Y=0.0934339
Scan=75 Field=SDP.81 [3] Time=2014/11/03/10:30:26.4000 BL=DA41@S301 & PM04@T703 [0&24] Spw=2 Chan=<0~63> Avg Freq=156.444 Corr=XX X=3111.84 Y=0.0460959 (
```

Calibration

While these points are from the target, only one correlation and one antenna seems to be affected.

Thus we will flag these data. (Standard practice is to flag both correlations).

```
## Outlier antenna in spw2
flagdata(vis = 'SDP81_B4_uncalibrated.ms.split',
  mode = 'manual',
  spw = '2',
  antenna= 'PM04',
  flagbackup = False)
```

Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

Gain Calibration

Setting the Flux Scale

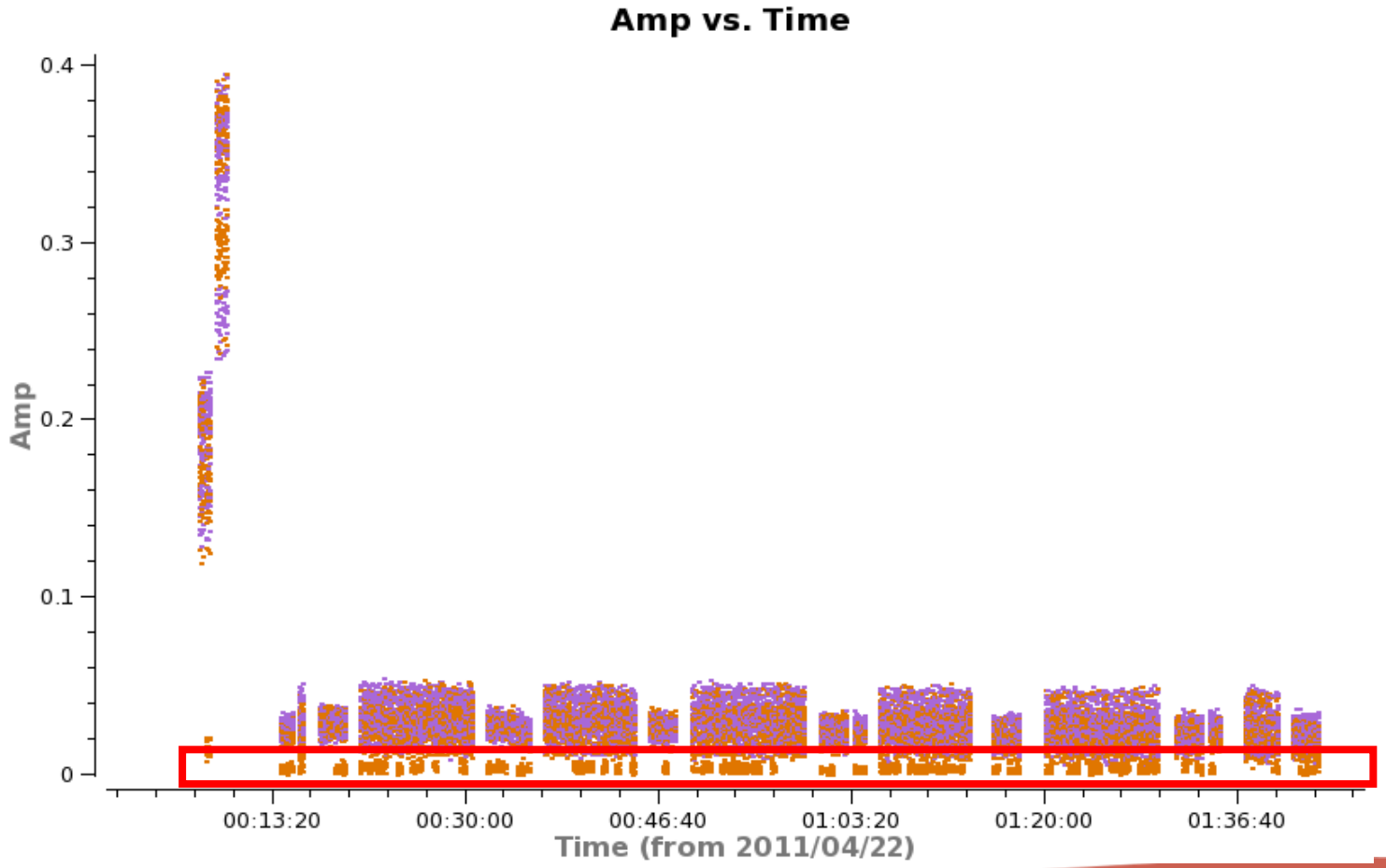
Applying the Calibration

Checking the Calibration

Renormalization Correction

Amplitude vs time: Look for scans that are different than other scans of the same target.

Look for outliers (very low or very high antennas or baselines).



Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

Gain Calibration

Setting the Flux Scale

Applying the Calibration

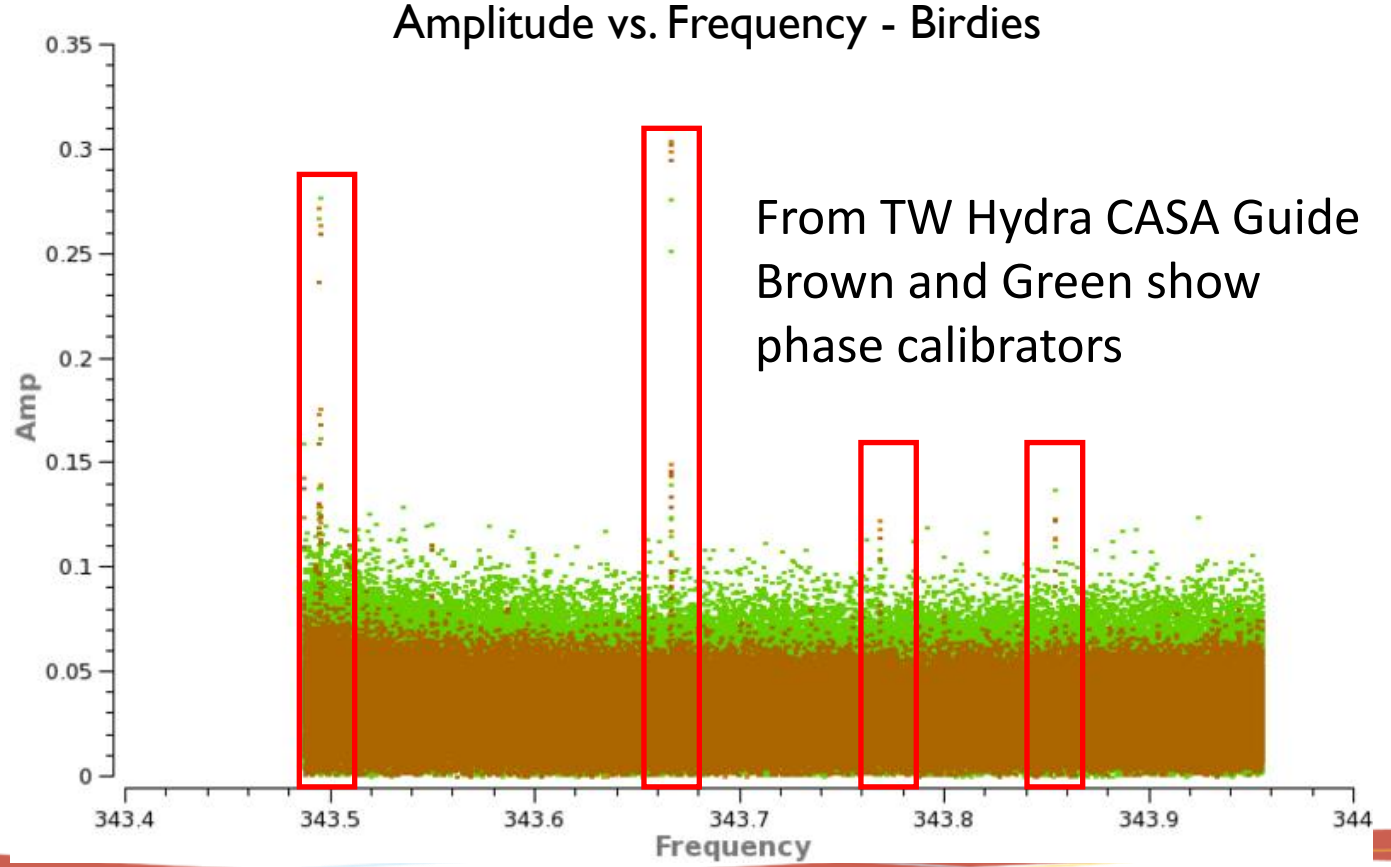
Checking the Calibration

Renormalization Correction

For Amplitude vs frequency (or channel):

Look for bad edge channels, birdies, and other effects **IN THE CALIBRATORS!**

The target may have spectral lines that can be real. Gain calibrators can also sometimes have native lines. Lines don't need to be flagged form the gain calibrator.



From TW Hydra CASA Guide
Brown and Green show phase calibrators

Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

Gain Calibration

Setting the Flux Scale

Applying the Calibration

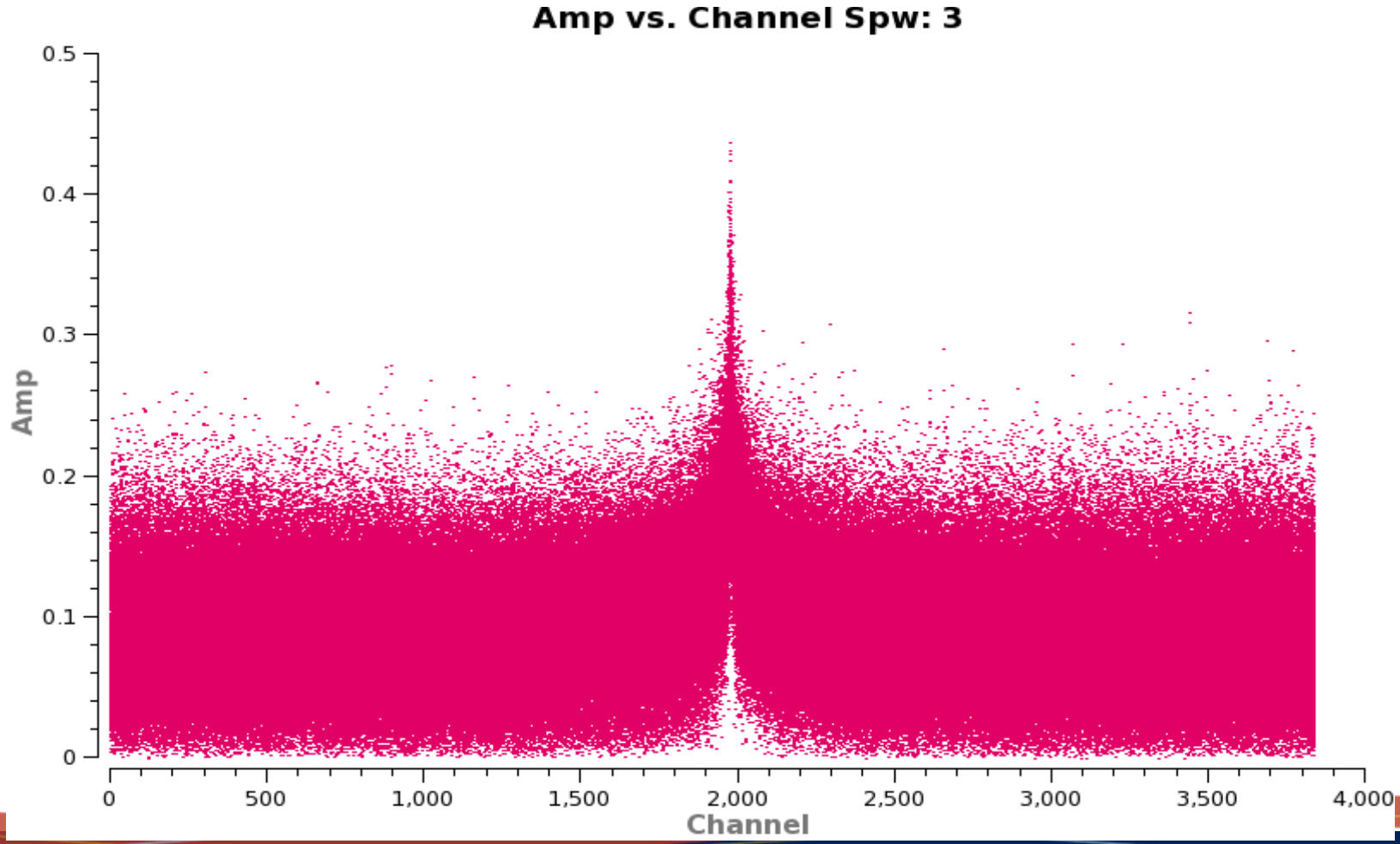
Checking the Calibration

Renormalization Correction

Spectral lines in flux calibrators should be flagged.

From TW Hydra Band 7 Guide

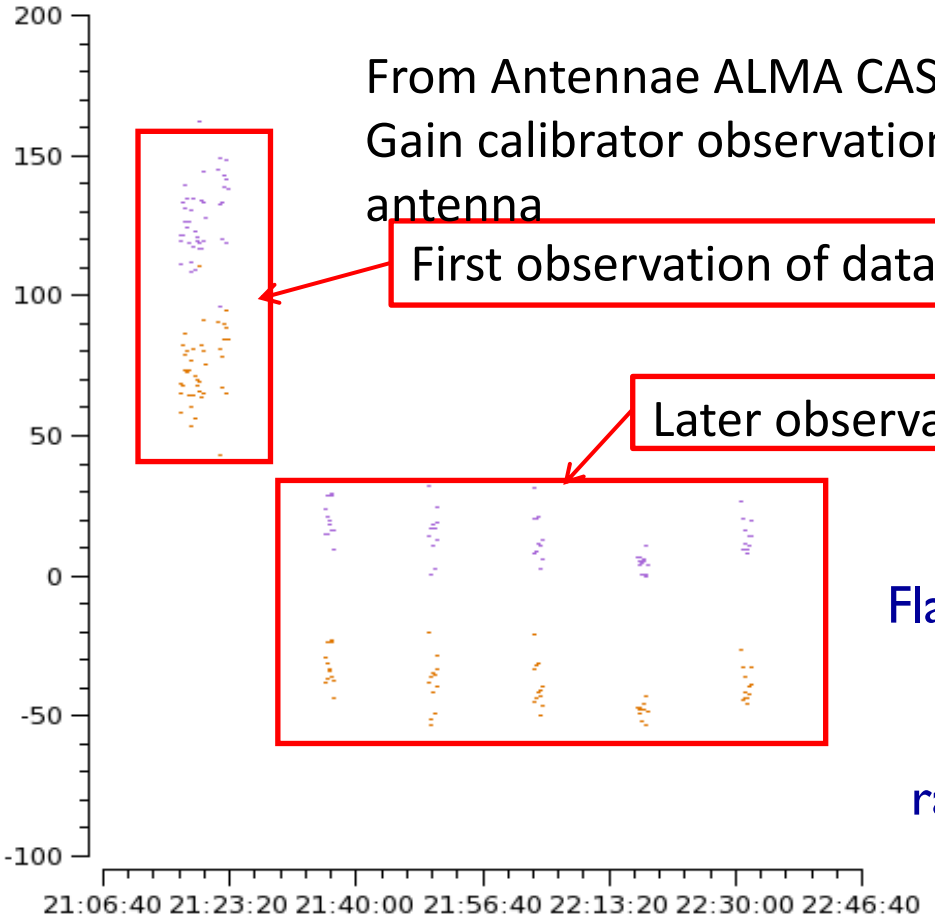
Spectral line in Titan (Flux Calibrator)



Calibration

Phase vs time and phase vs frequency:

Look for discontinuities and where the phase changes rapidly. Increased phase noise is normal around atmospheric lines.



From Antennae ALMA CASA Guide
Gain calibrator observations on one antenna

First observation of data

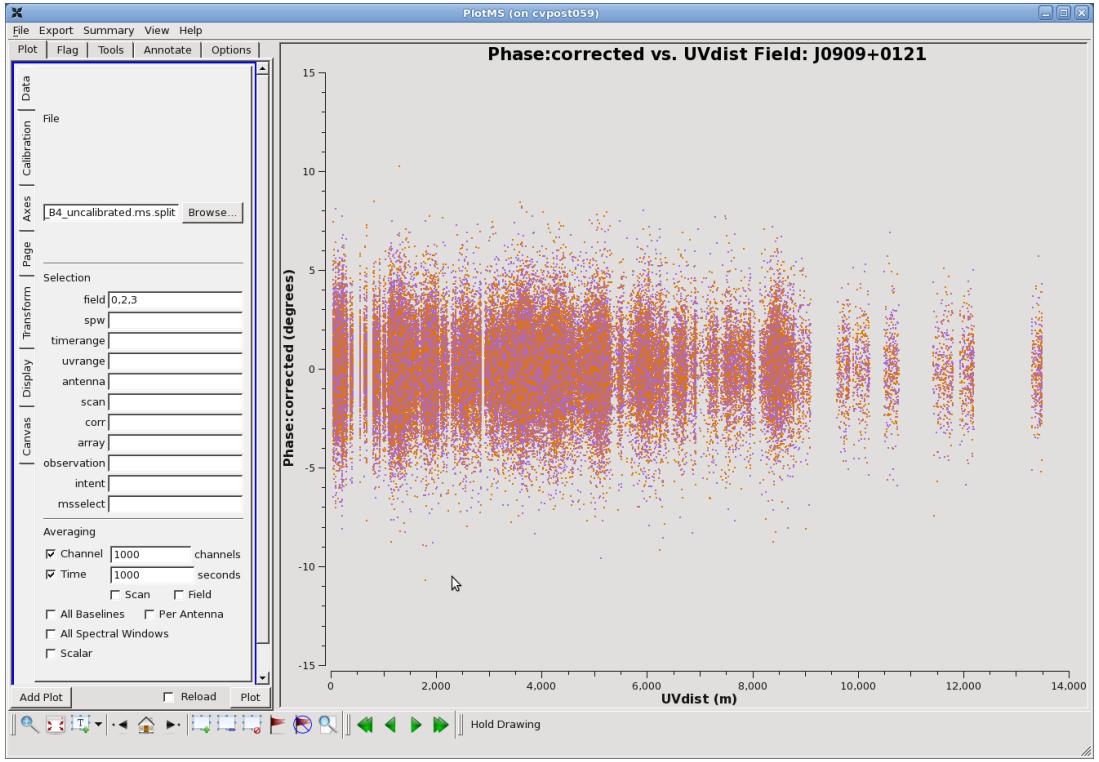
Later observations of data

Flag target data in the region where the solution is changing rapidly/discontinuously

Your turn:

Look at the plots you did not look at yet:

- Amplitude vs time
- Phase vs time
- Amplitude vs frequency
- Phase vs frequency



Phase vs UV distance for J0909+0121

Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

Gain Calibration

Setting the Flux Scale

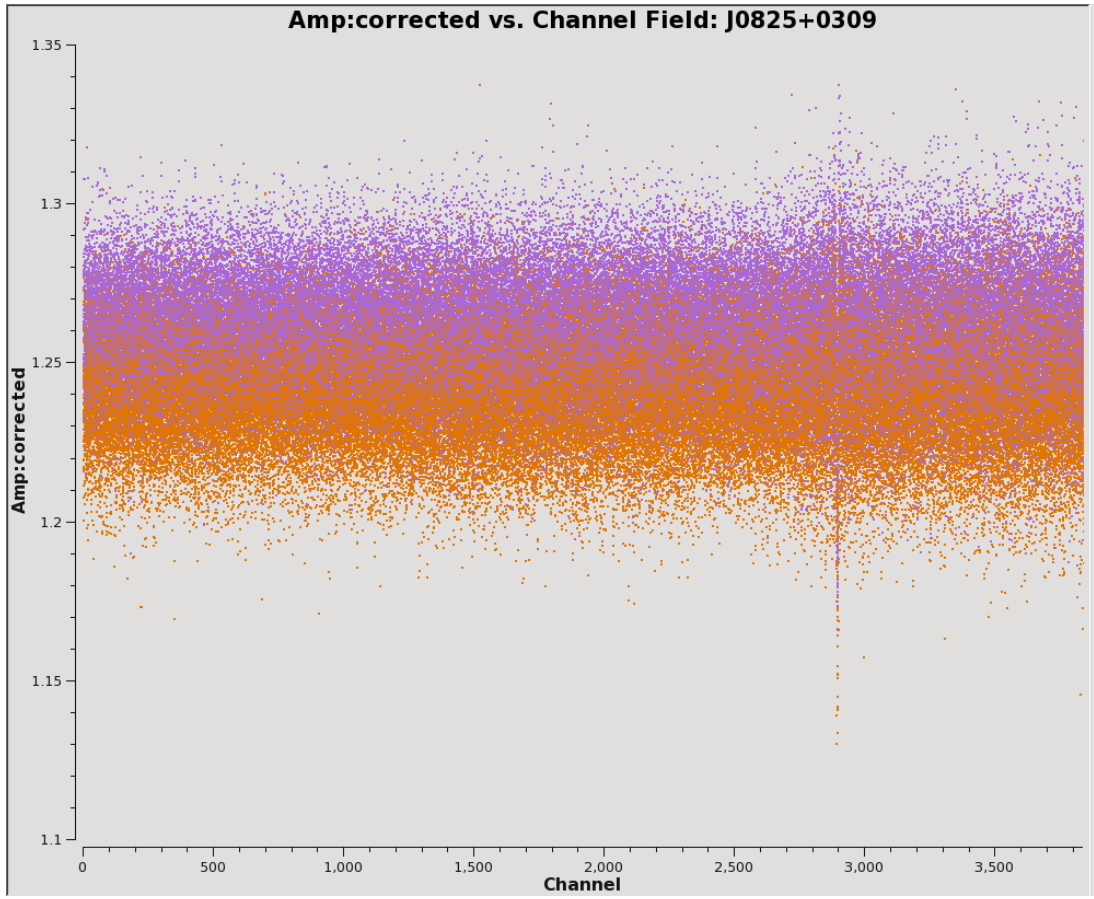
Applying the Calibration

Checking the Calibration

Renormalization Correction

Remember: We do not flag telluric lines in the data.

After writing out all the flag commands, re-run the entire calibration!



Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

Gain Calibration

Setting the Flux Scale

Applying the Calibration

Checking the Calibration

Renormalization Correction

Renormalization:

A visibility amplitude calibration error that affects fields containing strong line emission

Corrected in affected datasets that have >10% flux offset since Cycle 7.

Knowledgebase Article:

https://help.almascience.org/kb/articles/what-are-the-amplitude-calibration-issues-caused-by-alma-s-normalization-strategy

New pipeline stage for pipeline-calibrated datasets

Manually calibrated datasets are checked and corrected before delivery. This dataset does not require renorm correction. However, the script includes a demonstration using the pipeline renorm module

Calibration

Traditional scheme

$$c_{ij}(f)\ [\mathrm{V}^2] \xrightarrow{\text{norm}} \frac{c_{ij}(f)}{\sqrt{\langle c_{ii}\rangle_f \langle c_{jj}\rangle_f}}\ [\mathrm{V}^2] \xrightarrow{\text{cal}} c_{ij}(f)\ [\mathrm{V}^2]\ \frac{\sqrt{\langle T_i\rangle_f[K] \cdot \langle T_j\rangle_f[K]}}{\sqrt{\langle c_{ii}\rangle_f \cdot \langle c_{jj}\rangle_f}\ [V^2]} \longrightarrow c_{ij}(f)[\mathrm{K}]$$

Averaged

ALMA scheme

$$c_{ij}(f)\ [\mathrm{V}^2] \xrightarrow{\text{norm}} \frac{c_{ij}(f)}{\sqrt{c_{ii}(f)c_{jj}(f)}}\ [\mathrm{V}^2] \xrightarrow{\text{cal}} c_{ij}(f)\ [\mathrm{V}^2]\ \frac{\sqrt{T_i(f)[K] \cdot T_j(f)[K]}}{\sqrt{c_{ii}(f) \cdot c_{jj}(f)}\ [V^2]} \longrightarrow c_{ij}(f)[\mathrm{K}]$$

Not spectrally averaged

Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

Gain Calibration

Setting the Flux Scale

Applying the Calibration

Checking the Calibration

Renormalization Correction

$$c_{ij}(f)[V^2] \rightarrow c_{ij}(f)[V^2] \frac{\sqrt{T_i(f)[K] \cdot T_j(f)[K]}}{\sqrt{c_{ii}(f) \cdot c_{jj}(f)}[V^2]} \rightarrow c_{ij}(f)[K]$$

- Both the autocorrelations and the system temperature measurements are a total power-like measurement of the sky.
- If the target source is highly extended and bright, then the source can be picked up in these total power measurements which then impacts this normalization scheme in any channels where the emission was picked up!
- Dividing by the autocorrelations will under-scale the cross-correlations in any affected channels.
- Multiplying by T_{sys} measurement will over-scale the cross-correlations in any affected channels.
- These effects perfectly cancel each other ONLY if they are of the same field.

Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

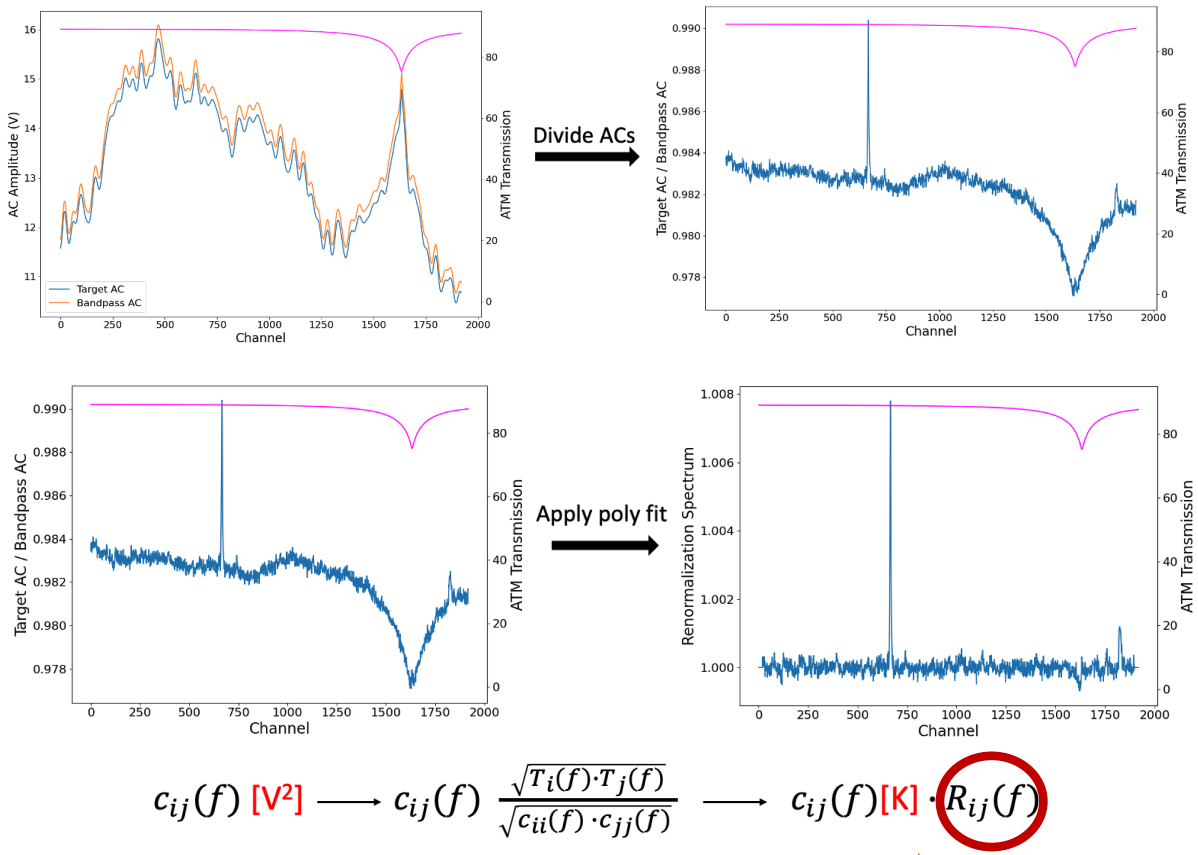
Gain Calibration

Setting the Flux Scale

Applying the Calibration

Checking the Calibration

Renormalization Correction



$$c_{ij}(f)\ [\mathrm{V}^2] \longrightarrow c_{ij}(f)\ \frac{\sqrt{T_i(f)\cdot T_j(f)}}{\sqrt{c_{ii}(f)\cdot c_{jj}(f)}} \longrightarrow c_{ij}(f)[\mathrm{K}]\cdot R_{ij}(f)$$

Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

Gain Calibration

Setting the Flux Scale

Applying the Calibration

Checking the Calibration

Renormalization Correction

Final Steps:

After you finish

1. Calibrating the data
2. Inspecting the data
3. Adding flags
4. Repeating steps 1-4 as needed.
5. Applying renorm

It is time to split the corrected columns to a new dataset.

This has already been done for you in the interest of time.

If you need to free up space on your computer, you may now remove SDP81_B4_uncalibrated.ms.split from your computer.

Calibration

Saving Flagging State

Flagging Known Bad Data

Other Flags

Pre-Bandpass Phase Calibration

Bandpass Calibration

Checking Bandpass Calibration

Apply Flux Calibrator Model to Data

Gain Calibration

Setting the Flux Scale

Applying the Calibration

Checking the Calibration

Renormalization Correction

Normally, we would now repeat this step for all other executions.

For SDP.81, there were a total of 12 observations.

To save time, we have done the other 11 executions for you. In your directory you will find:

SDP.81_Band4.ms

We will use this measurement set to practice imaging.

Imaging

Imaging

Basics of Imaging Review

Bandpass Calibrator Imaging

CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging

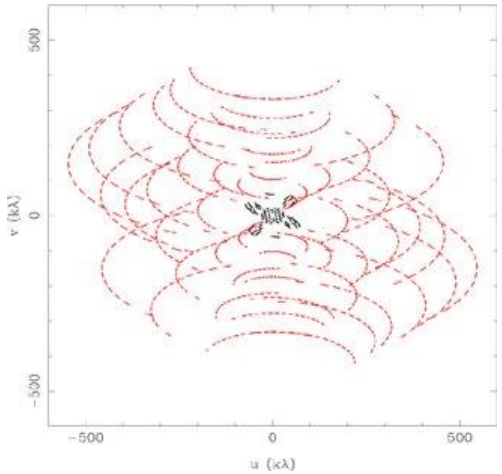
We want to now image our data! Where do we start?

Most of this section is going to be familiar. You should have seen it in the imaging talks earlier in the workshop!

- Thursday:
  - Introduction to imaging and deconvolution (*Joshua Marvil*)
- Friday:
  - Narrow band mosaicking and data combination (*Adele Plunkett*)
  - We will perform a quick review!

Dirty beams and images

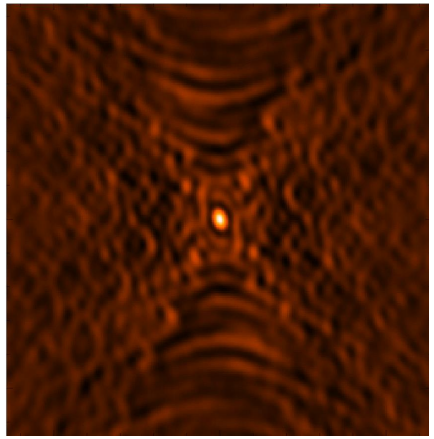
B(u,v)
(sampled visibilities)



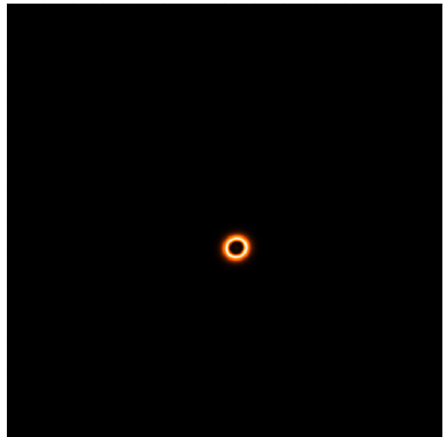
(F. Transform)

⇌

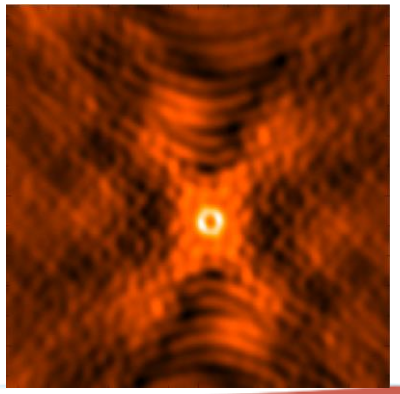
b(x,y)
(dirty beam or psf)



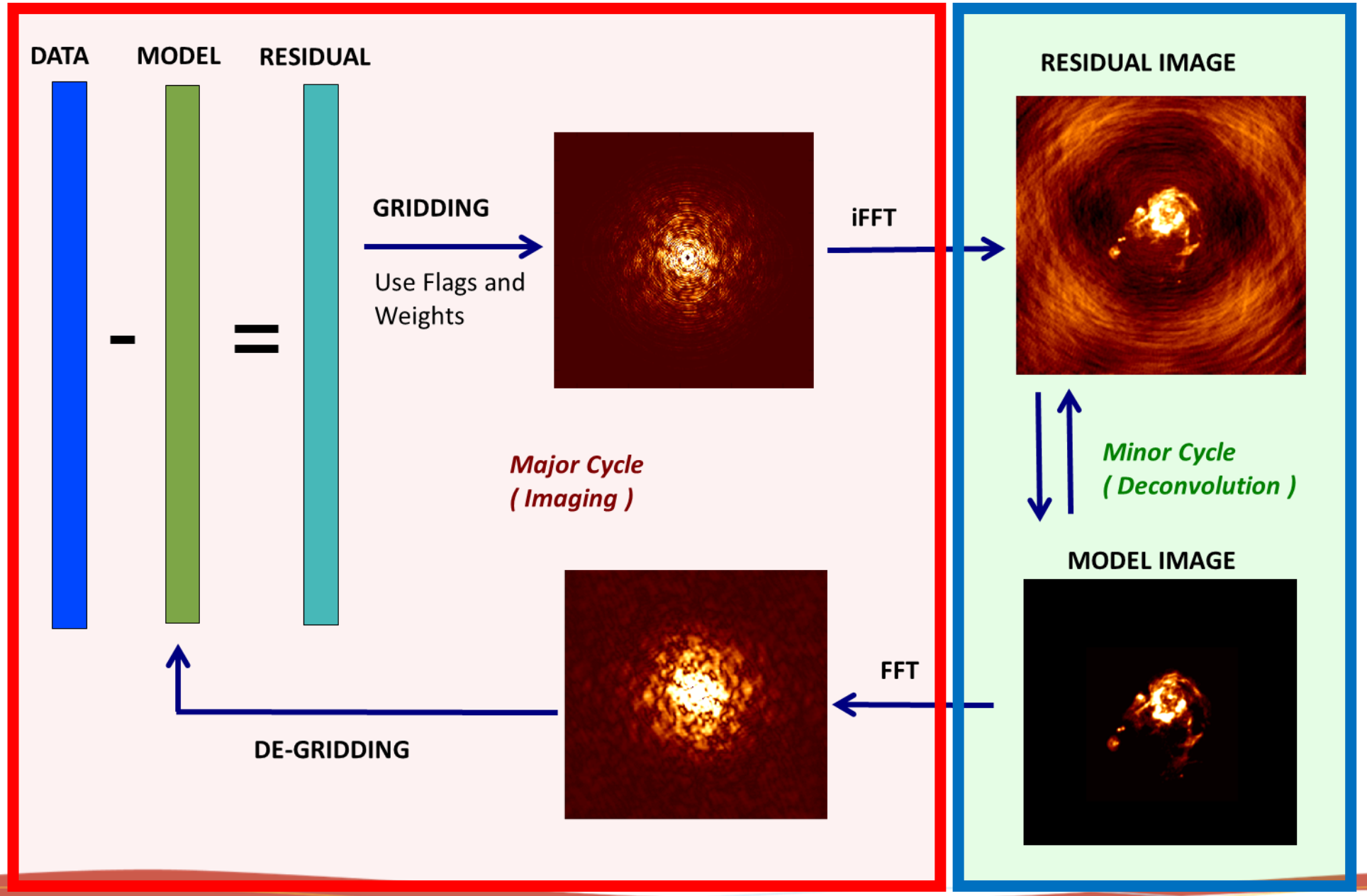
Convolve



T(x,y)
(True sky brightness)



TD(x,y)
(dirty image)


DATA
MODEL
RESIDUAL
GRIDDING
Use Flags and Weights
iFFT
RESIDUAL IMAGE
Major Cycle
(Imaging)
Minor Cycle
(Deconvolution)
MODEL IMAGE
FFT
DE-GRIDDING

Imaging

Stopping criteria:

- r*esidual* map max < multiple of rms (when noise limited)
- r*esidual* map max < fraction of dirty map max (dynamic range limited)
- max number of clean components reached (no justification)

loop gain

- good results for g ~ 0.1 to 0.3
- lower values can work better for smoother emission, g ~ 0.05

easy to include *a priori* information about where to search for clean components (“clean boxes”)

Weighting Options

Natural

- Better sensitivity, lower resolution
- $W(u,v) = 1/\sigma^2$
- With ALMA, more weight to shorter baselines
- (More short baselines, more weight)

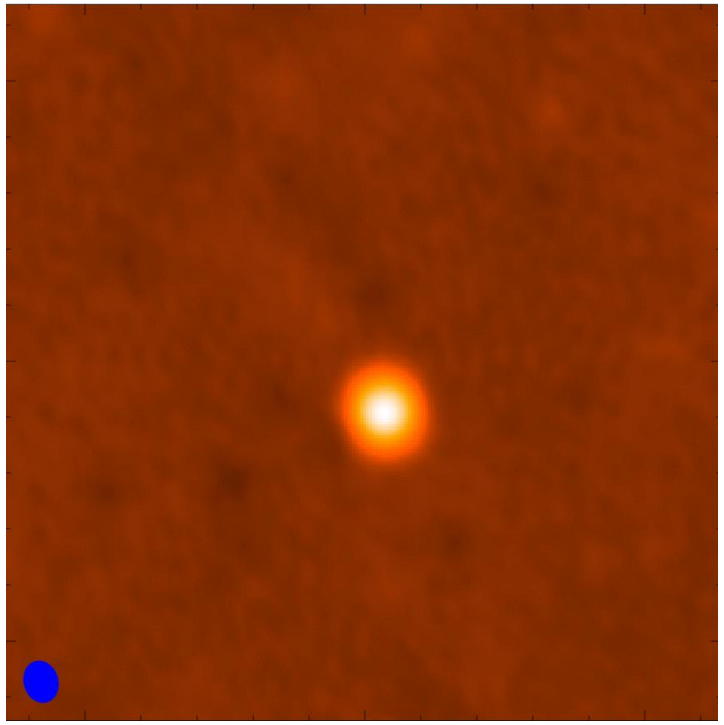
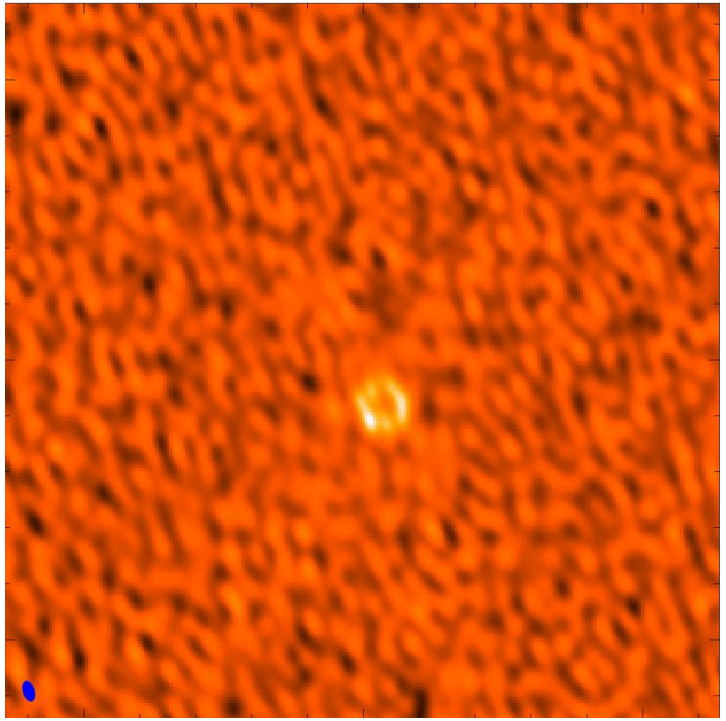
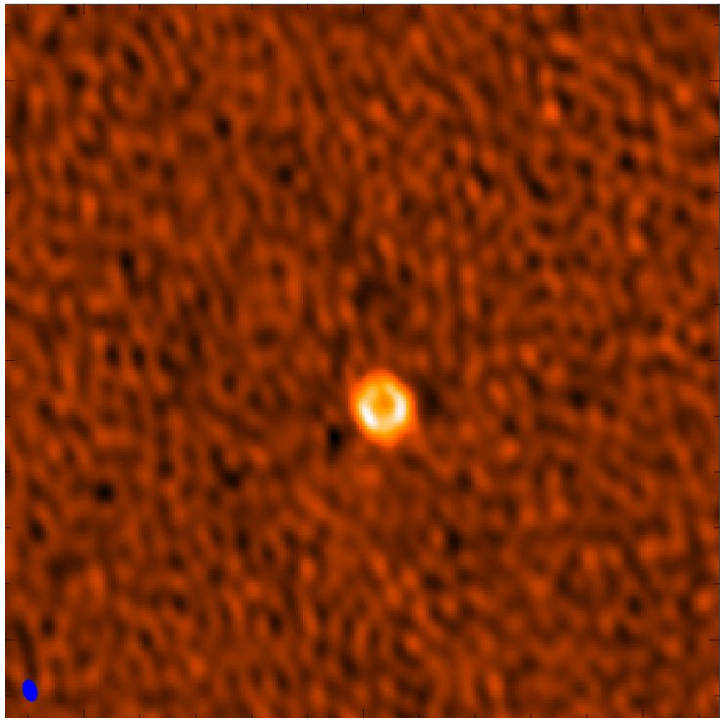
Uniform

- Worse sensitivity, better resolution.
- Note: can cause problems with sparse (u,v) coverage as cells with few samples have the same weight as cells with many.
  - Think 7m ACA observations.
- Weight all the same.

Robust ("Briggs")

- Compromise. Can recover most of the sensitivity and resolution.
- Variation of Uniform that avoids giving large weights to cells with small natural weigh.
- Robust = -2 => Uniform
- Robust = 2 => Natural
- Stick to robust ≥ 0.0
- Imaging uses robust =0.5 by default in ALMA pipeline.

Weighting Options

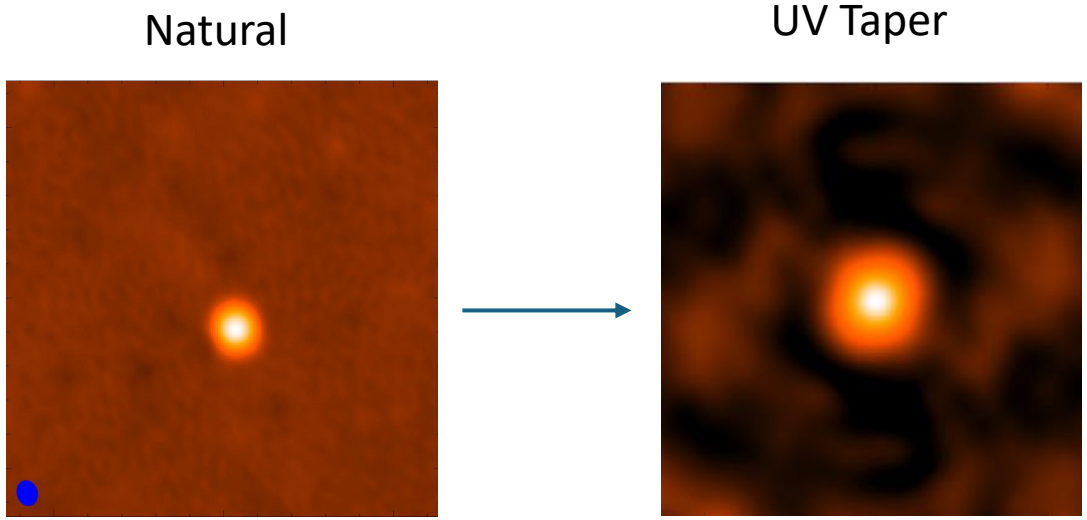
Natural	Uniform	Robust = 0
		

Imaging

UV Tapering is similar to convolving the image by a Gaussian

Added weight to short baselines degrades resolution

May improve sensitivity to extended structure.



Imaging

Basics of Imaging Review

Bandpass Calibrator Imaging

CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging

Clean Boxes:

- Because we do not fully sample the uv-plane in our imaging, there is generally no unique solution to the deconvolution process
- We use clean ‘boxes’, or masks, to identify regions of the image or cube with real emission
- Clean boxes are a way to create the best possible model for your source – particularly sources with complex emission
- As a first step, include bright features in your mask, drawing a close contour around the emission
- For cubes, you can mask channel-by-channel, or all channels
- As tclean progresses, strong residuals that do not appear to be due to sidelobes (i.e., do not disappear in subsequent cycles) can be added iteratively
- Be careful when masking – adding a mask around noise or beam sidelobes can create features in your final image that are not real

Imaging

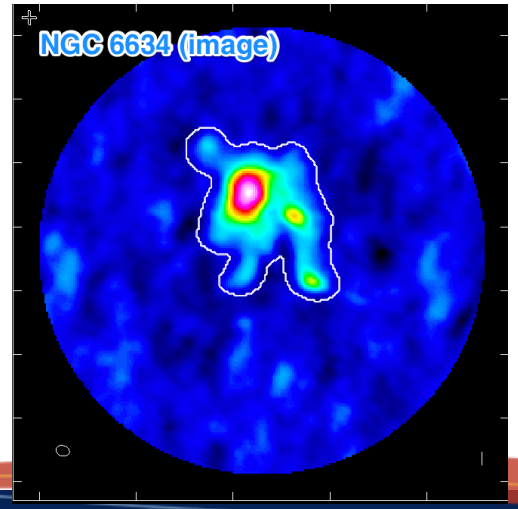
Basics of Imaging Review

Bandpass Calibrator Imaging

CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging



Automasking (auto-multithresh):

- Algorithm developed by A. Kepley, T. Tsutsumi (+Yoon, Indebetouw, Brogan)
  - parameterized in terms of fundamental image parameters (S/N, fraction of beam, sidelobe level) ⇒ instrument independent
  - Masks are re-calculated every major cycle within tclean ⇒ follows evolution of image
- **Available in tclean since CASA 5.1**
  - usemask='auto-multithresh'
- Deployed in ALMA Cycle 5 pipeline
- CASA guide: https://casaguides.nrao.edu/index.php?title=Automasking_Guide
- Be careful if you have poor UV coverage.

Imaging

Basics of Imaging Review

Bandpass Calibrator Imaging

CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging

Tclean!

Tclean has many options. You should check out the tclean page on the CASAdocs to see what it is capable of!

```
tclean(vis, selectdata=True, field='', spw='', timerange='', uvrange='', antenna='', scan='', observation='', intent='', datacolumn='corrected',
imagename='', imsize=[100], cell='"1arcsec"', phasecenter='', stokes='I', projection='SIN', startmodel='', specmode='mfs', reffreq='',
nchan=-1, start='', width='', outframe='LSRK', veltype='radio', restfreq='', interpolation='linear', perchanweightdensity=True,
gridder='standard', facets=1, psfphasecenter='', wprojplanes=1, vptable='', mosweight=True, aterm=True, psterm=False, wbawp=True,
conjbeams=False, cfcache='', usepointing=False, computepastep=360.0, rotatepastep=360.0, pointingoffsetsigdev='', pblimit=0.2,
normtype='flatnoise', deconvolver='hogbom', scales='', nterms=2, smallscalebias=0.0, fusedthreshold=0.0, largestscale=-1,
restoration=True, restoringbeam='', pbcor=False, outlierfile='', weighting='natural', robust=0.5, noise='1.0Jy', npixels=0, uvtaper=[''],
niter=0, gain=0.1, threshold=0.0, nsigma=0.0, cycleniter=-1, cyclefactor=1.0, minpsffraction=0.05, maxpsffraction=0.8, interactive=False,
nmajor=-1, fullsummary=False, usemask='user', mask='', pbmask=0.0, sidelobethreshold=3.0, noisethreshold=5.0, lownoisethreshold=1.5,
negativethreshold=0.0, smoothfactor=1.0, minbeamfrac=0.3, cutthreshold=0.01, growiterations=75, dogrowprune=True,
minpercentchange=-1.0, verbose=False, fastnoise=True, restart=True, savemodel='none', calcres=True, calcpsf=True, psfcutoff=0.35,
parallel=False)  [source]
```

Radio Interferometric Image Reconstruction

Pixel and Image Size

Pixel (cell) Size

- Satisfy sampling theorem for long baselines
  - In practice, 5 to 8 pixels across dirty beam main lobe.
- $\theta_{res} \approx \frac{k\lambda}{L_{max}} [radians]$
  - L_{max} is the longest baseline [meters]
  - λ is the wavelength in meters
  - $0.7 < k < 1.2$ (typically) depends on weighting choice

Image Size

- Avoid aliasing from targets outside your image.
- Usual choice is the primary beam.
- 12m : $FOV[arcsec] = \frac{6300}{\nu[GHz]}$
- 7m: $FOV[arcsec] = \frac{10608}{\nu[GHz]}$

Imaging

Basics of Imaging Review

Bandpass Calibrator Imaging

CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging

Largest Angular Size

- **Range** from synthesized beam to maximum angular scale (MAS)
- **Smooth** structures larger than LAS begin to be resolved out.
- All flux on scales larger than $\frac{\lambda}{B_{min}} \sim (2 \times MAS)$ completely resolved out.

Band	Frequency (GHz)	Primary beam (“)	Range of Scales (“) C32-1	C32-9
3	84-116	72 - 52	4.2 - 24.6	0.7 - 15.1
6	211-275	29 - 22	1.8 - 10.7	0.3 - 6.6
7	275-373	22 - 16	1.2 - 7.1	0.2 - 4.4
9	602-720	10 – 8.5	0.6 - 3.6	0.1 - 2.2

Imaging

Output files from tclean (additional files made for some modes)

***.pb**

Relative sky sensitivity - shows the primary beam response

***.image**

Cleaned and restored image

***.mask**

Clean “boxes” shows where you cleaned

***.model**

Clean components - the model used by clean (in Jy/pixel)

***.psf**

Dirty beam - shows the synthesized beam

***.residual**

Residual shows what was left after you cleaned (the "dirty" part of the final image)

Imaging

- Since the Bandpass calibrator is a point source, we are going to experiment a little bit with options in clean!

- Execute all the commands up to the first tclean command in your imaging.py script.

```
tclean(vis='bandpass.ms',
imagename='bandpass_natural’,
  field='0’,
  spw=‘’,
  specmode='mfs’,
  deconvolver='hogbom’,
  gridder='standard’,
  imsize=[512,512],
  cell=['0.005arcsec’],
  weighting='natural’,
  threshold='0mJy’,
  niter=10000,
  interactive=True)
```

Imaging

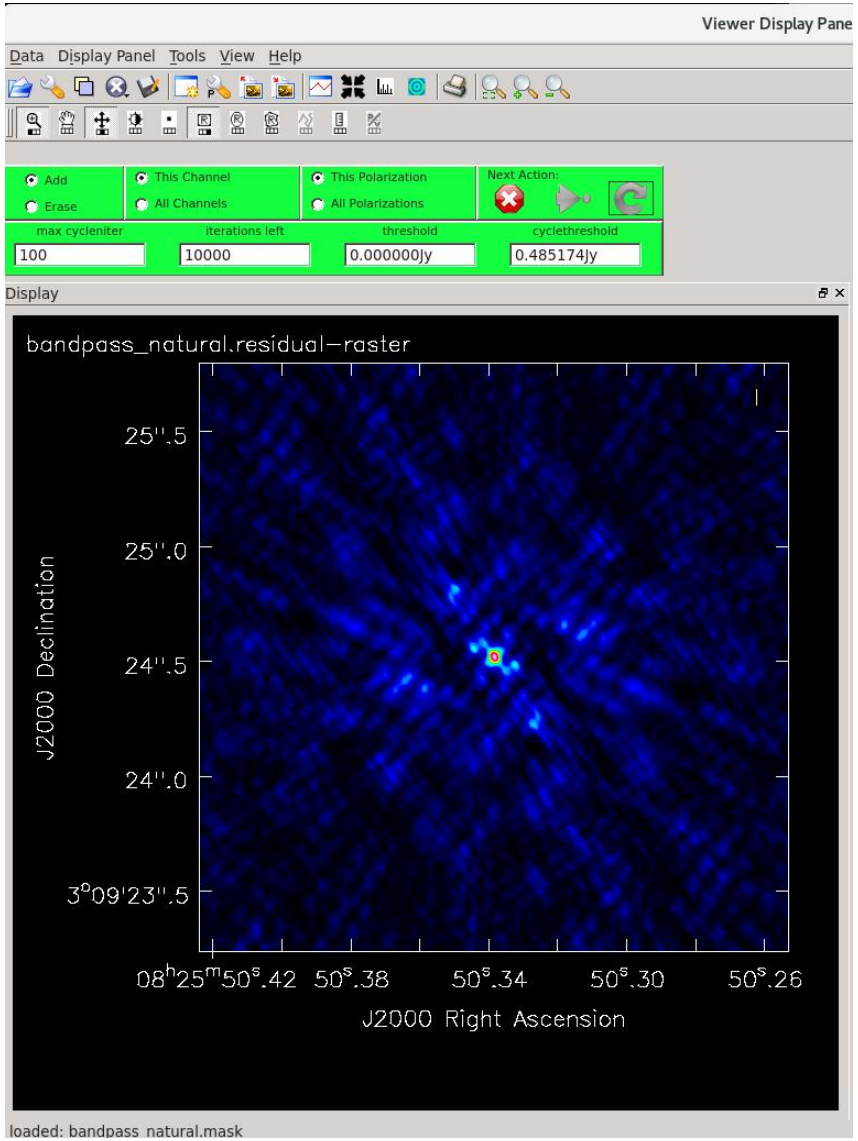
Basics of Imaging Review

Bandpass Calibrator Imaging

CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging



This is the dirty image of our calibrator.

Imaging

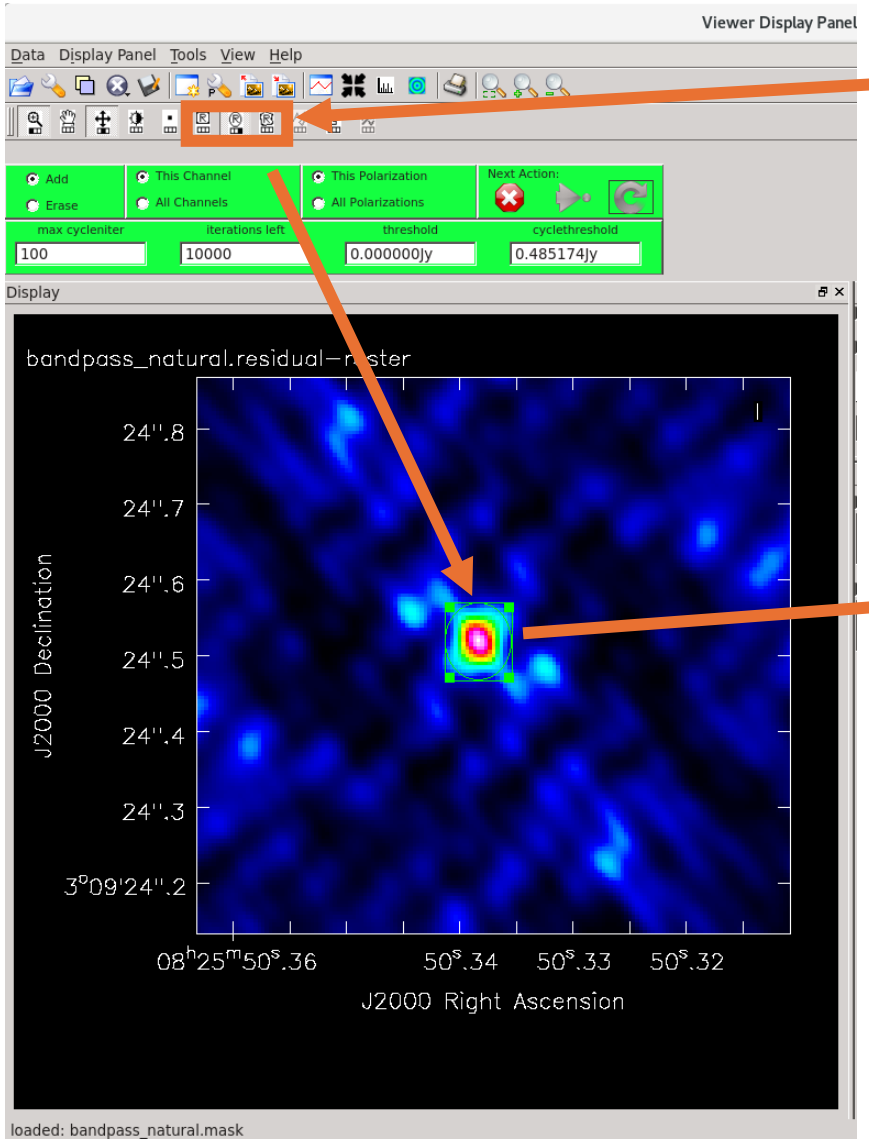
Basics of Imaging Review

Bandpass Calibrator Imaging

CARTA

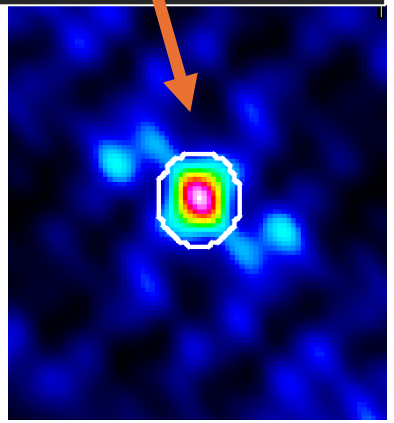
SDP.81 Continuum Imaging

SDP.81 CO Line Imaging



Click on a region tool and draw a mask around the emission you know is real.

Double click region to confirm mask. The mask outline will turn white.



Imaging

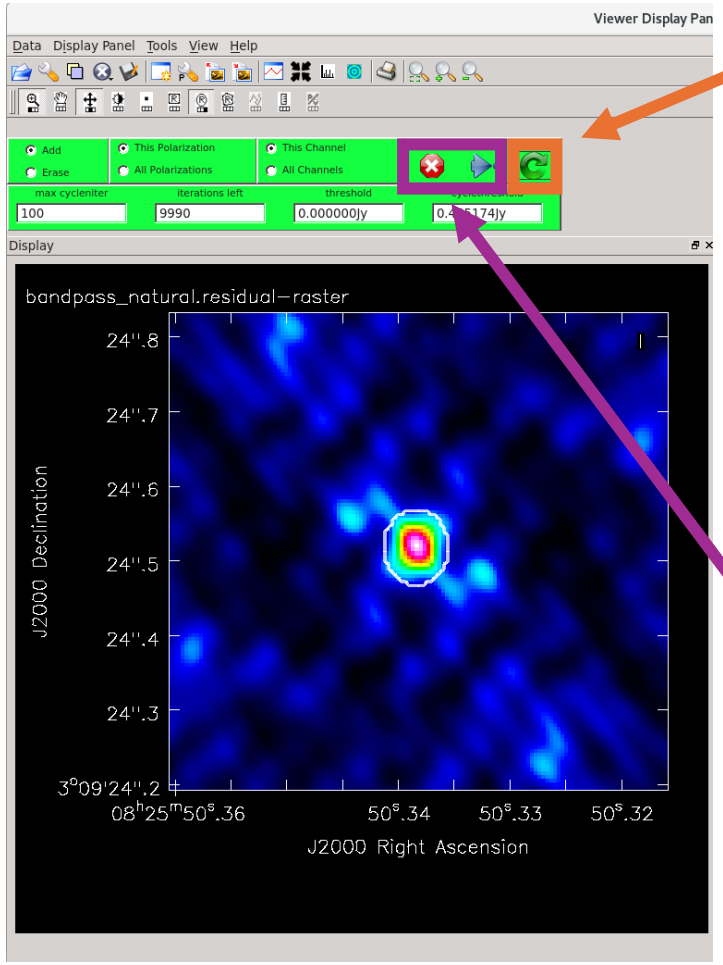
Basics of Imaging Review

Bandpass Calibrator Imaging

CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging

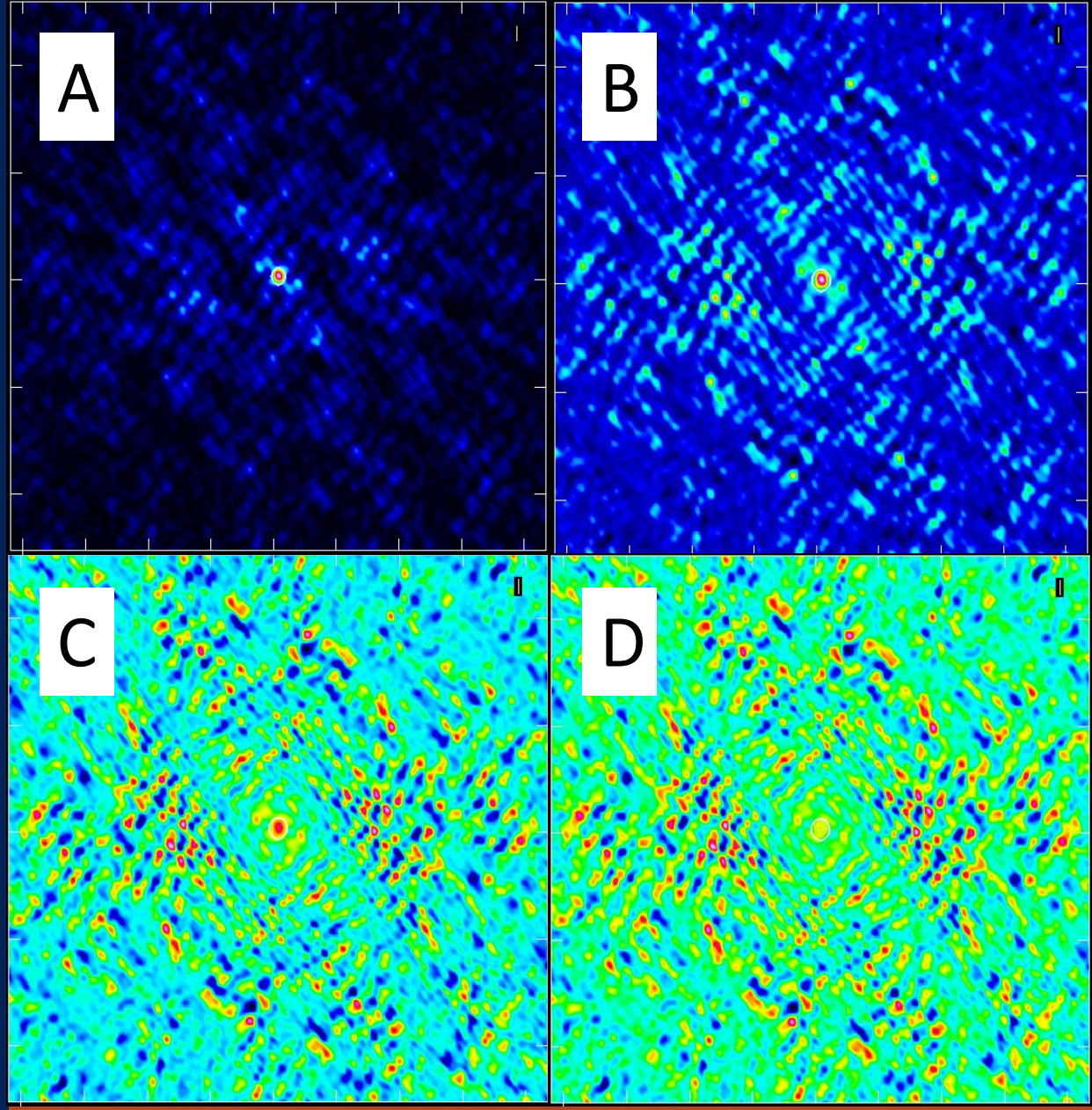


After masking all emission that you are confident is real, click the green circular arrow to run the next major cycle.

The blue arrow will run until stopping criteria are met. The red octagon with a white X stops cleaning. We won't use these yet.

Question:

- How clean is clean enough?
- "Clean until the residuals from the source are comparable to the residual noise outside the source."
- What does this mean to you? Select an image on the right and prepare to vote on where you would stop.



Imaging

How deep should you clean?

- Not a simple answer. Depends on UV coverage / dirty beam, SNR, structure of the emission, and what you are trying to accomplish.
- Science imaging:
  - If low SNR and good UV coverage, dirty beam residuals will be buried in the noise. No cleaning needed.
  - If data is well behaved, and source structure is well determined with the clean model, clean until the noise looks like thermal (non-patterned) noise.
  - Sources where you don't know the structure ahead of time, clean until residual source is comparable to the noise.
  - Data with calibration errors, clean until residual source is comparable to the noise.
- Self-Calibration
  - Phase self-cal: clean shallow to ensure only real structure is included.
  - Amplitude self-cal: Clean deep to ensure all flux included in model.

Imaging

Image viewing with CARTA!

Hopefully you were able to attend the section on CARTA earlier in the workshop.

CARTA should be launched, as we will need it going forward.

Imaging

Basics of Imaging Review

Bandpass Calibrator Imaging

CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging

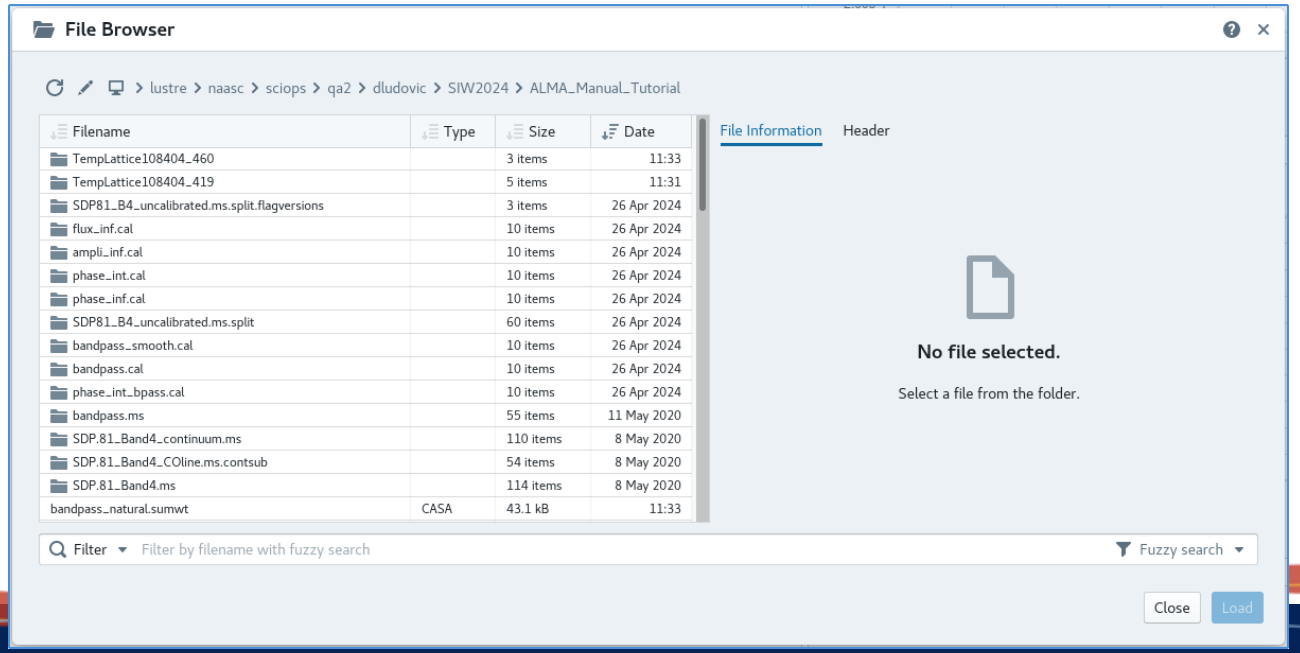
Opening your images in CARTA

If you installed with...

Package manager: type `carta`

Download AppImage: type `/<path to carta>/<carta_version>.appimage`

You should see the following prompt in your web browser:



Imaging

Basics of Imaging Review

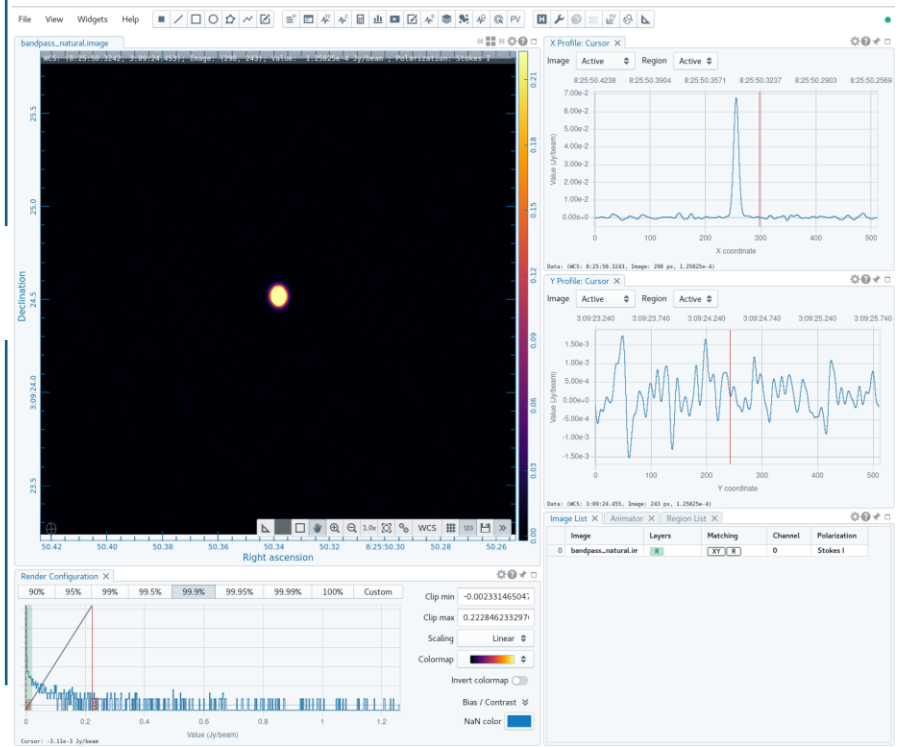
Bandpass Calibrator Imaging

CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging

- Select bandpass_natural.image. Information will be displayed on the right side. Then click "load".



File Information Header

```
Name = bandpass_natural.image
Data type = float
Shape = [512, 512, 1, 1] (RA, DEC, STOKES, FREQ)
Number of channels = 1
Number of polarizations = 1
Coordinate type = Right Ascension, Declination
Projection = SIN
Image reference pixels = [257, 257]
Image reference coords = [08:25:50.3384, +003.09.24.5201]
Image ref coords (deg) = [126.46 deg, 3.15681 deg]
Pixel increment = -0.005", 0.005"
Pixel unit = Jy/beam
Celestial frame = FK5, J2000
Spectral frame = LSRK
Velocity definition = RADIO
Restoring beam = 0.0632901" X 0.0524473", 4.49087 deg
RA range = [08:25:50.253, 08:25:50.424]
DEC range = [+03.09.23.240, +03.09.25.795]
Frequency = 150.4977 (GHz)
```

Imaging

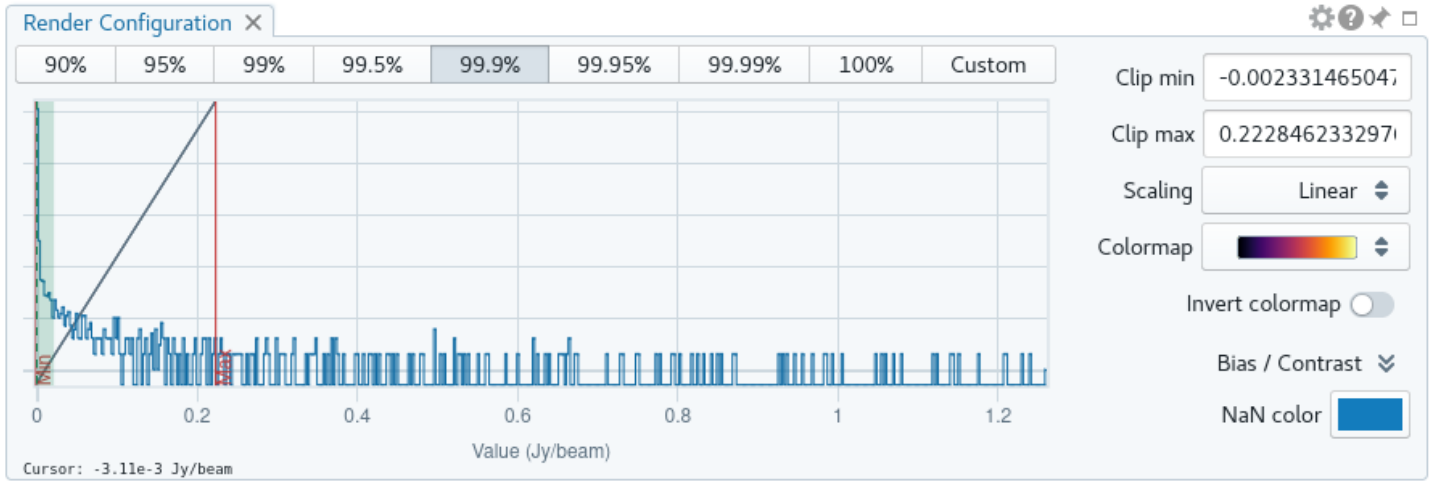
Basics of Imaging Review

Bandpass Calibrator Imaging

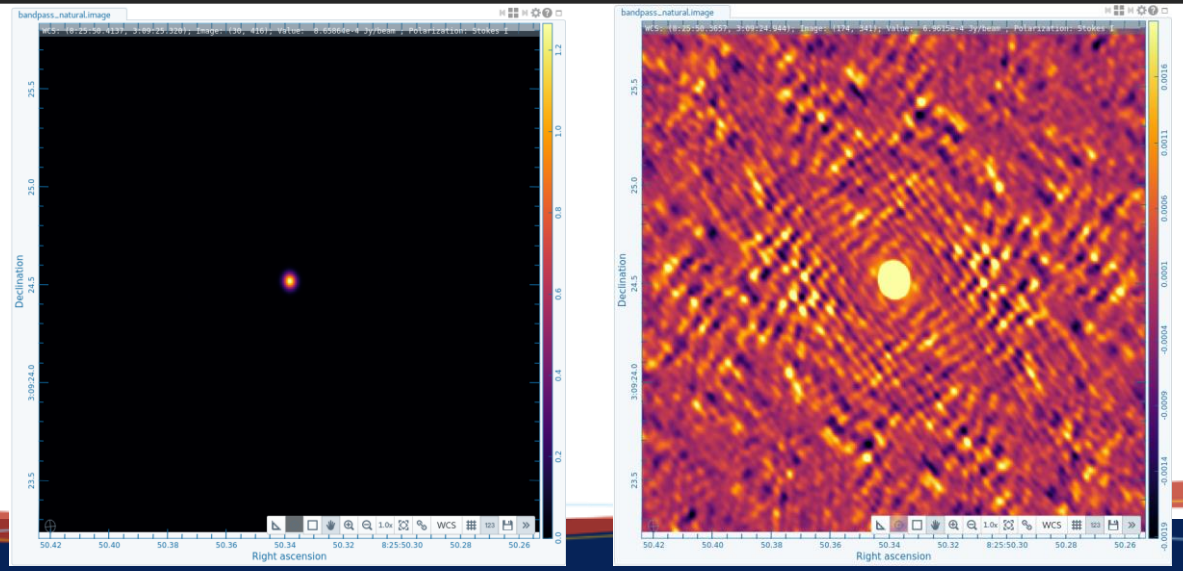
CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging



The render configuration allows you to adjust the histogram for the display. Try adjusting using the buttons, and using a custom range by clicking and dragging the min and max bars on the plot.



Imaging

Basics of Imaging Review

Bandpass Calibrator Imaging

CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging

- What happens if we change our weighting?
- You can use the next tclean command. Go ahead and clean it yourself!

```
os.system('rm -rf bandpass_robust.*')
tclean(vis='bandpass.ms',
imagename='bandpass_robust',
  field='0',
  spw='',
  specmode='mfs',
  deconvolver='hogbom',
  gridder='standard',
  imsize=[512,512],
  cell=['0.005arcsec'],
  weighting='briggs',
  robust=0.0,
  threshold='0mJy',
  niter=10000,
  interactive=True)
```

Imaging

Basics of Imaging Review

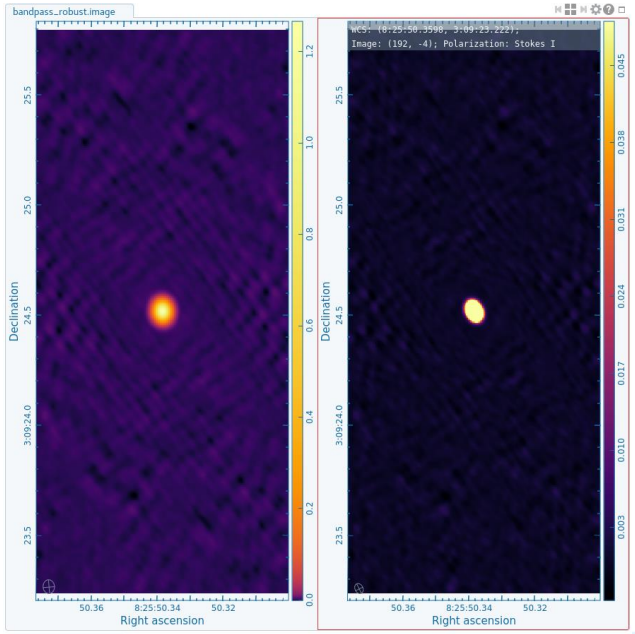
Bandpass Calibrator Imaging

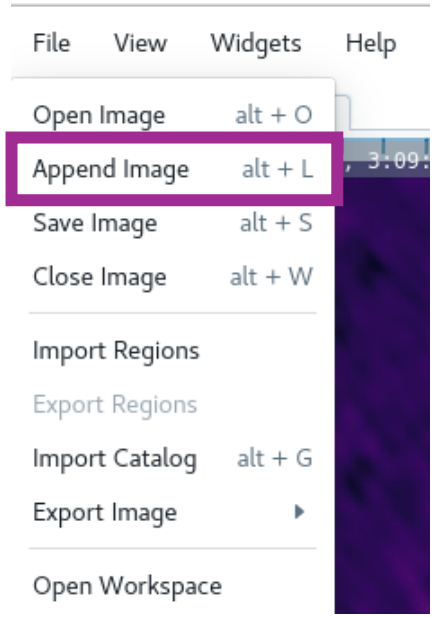
CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging

- After cleaning is finished. Go to CARTA.
- Click File > Append Image
- Choose your new robust weighted bandpass image





Imaging

Basics of Imaging Review

Bandpass Calibrator Imaging

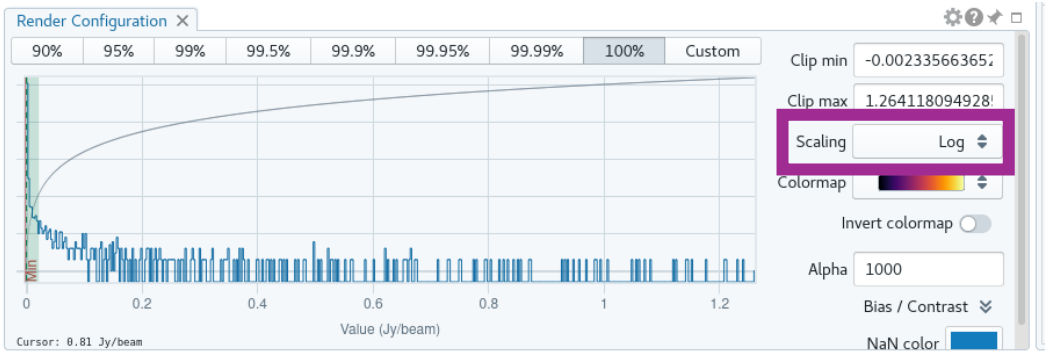
CARTA

SDP.81 Continuum Imaging

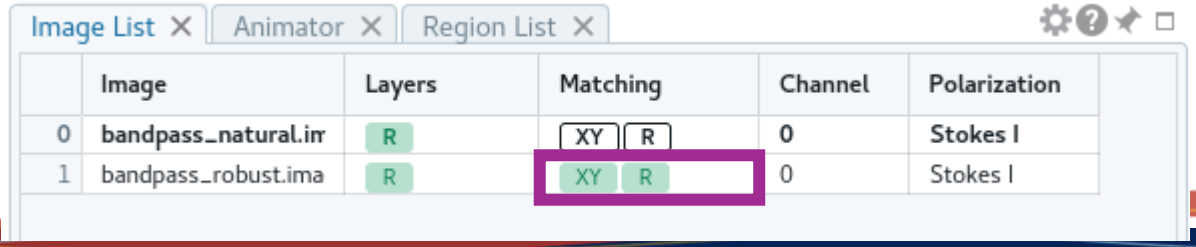
SDP.81 CO Line Imaging

Let's prepare to compare images.

1. Set scaling to log so we can see the full range of structures in the image.



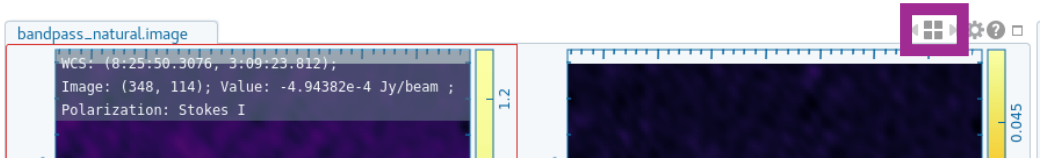
2. In the Image List, set matching on image 2 for XY (Coordinates) and R(Render Config)



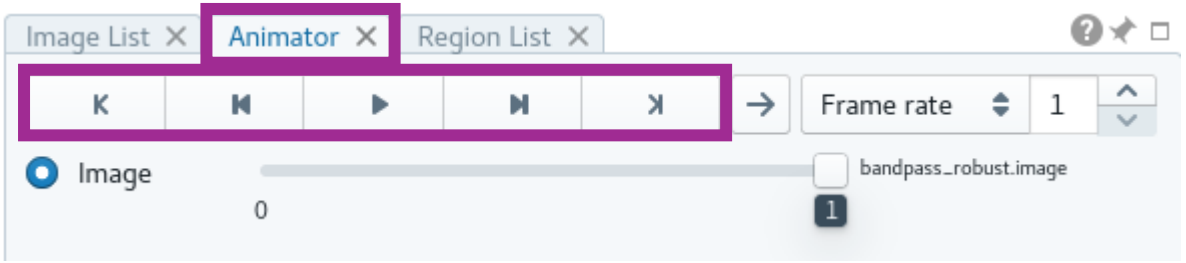
Imaging

Blink the images!

- On the top of the image window, click the button to switch to a single image.



- Click on the animator tab beside the image list tab. Use the tools to switch between images or blink.



Imaging

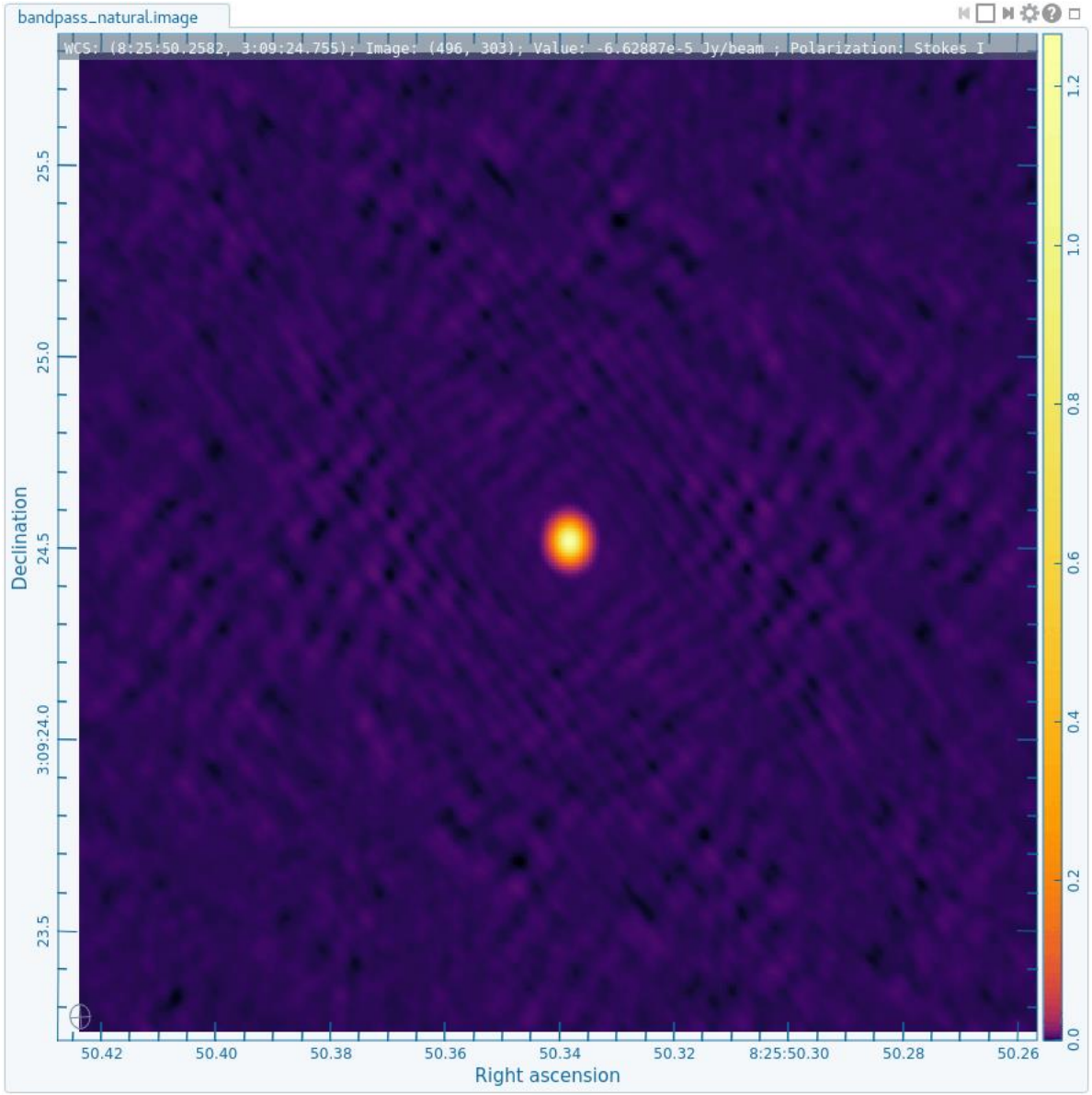
Basics of Imaging Review

Bandpass Calibrator Imaging

CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging



Imaging

Let's make an image incorrectly.

We will make an image with pixels that are too large.

```
tclean(vis='bandpass.ms',
  imagename='bandpass_bigpix',
  field='0',
  spw='',
  specmode='mfs',
  deconvolver='hogbom',
  gridder='standard',
  imsize=[128,128],
  cell=['0.05arcsec'],
  weighting='briggs',
  robust=-1,
  threshold='0mJy',
  niter=10000,
  interactive=True)
```

Imaging

Basics of Imaging Review

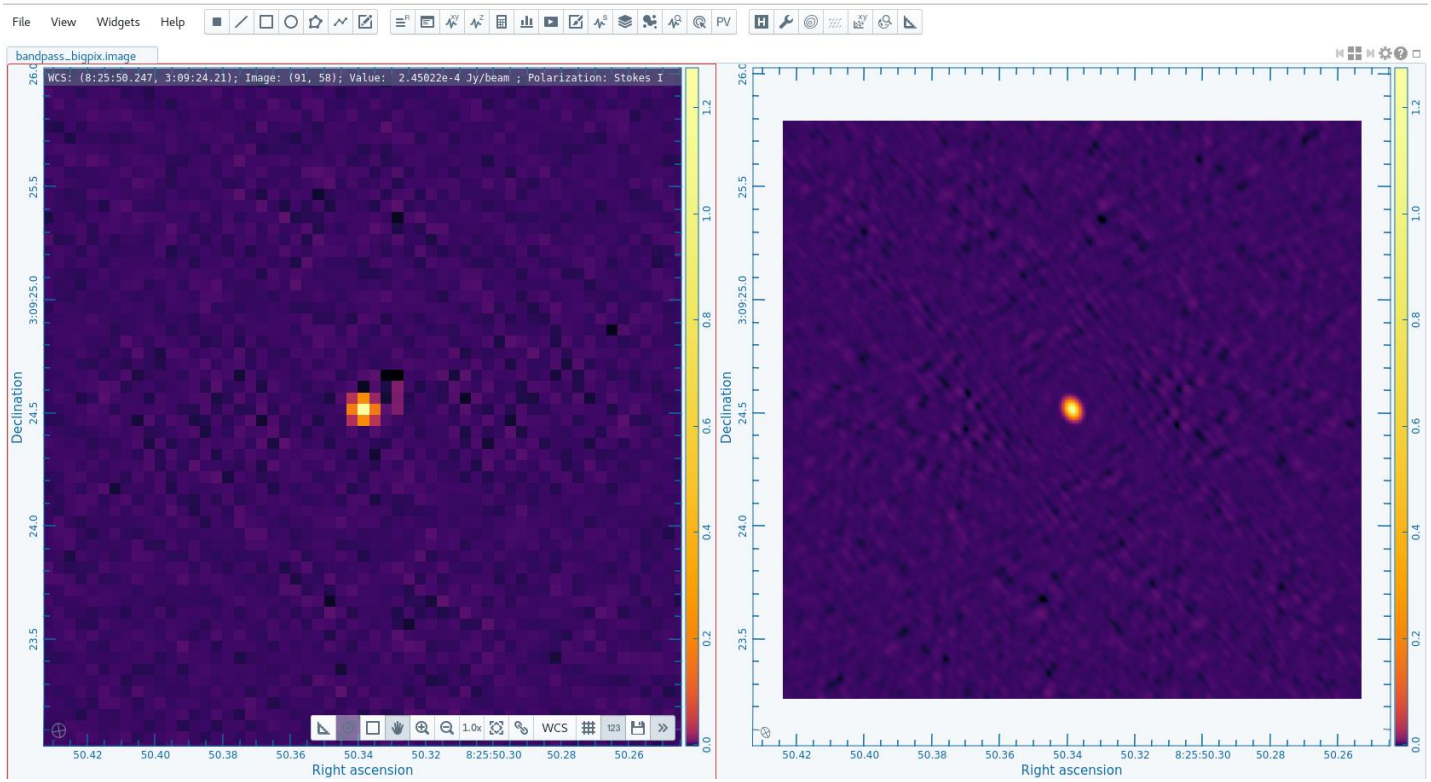
Bandpass Calibrator Imaging

CARTA

SDP.81 Continuum Imaging

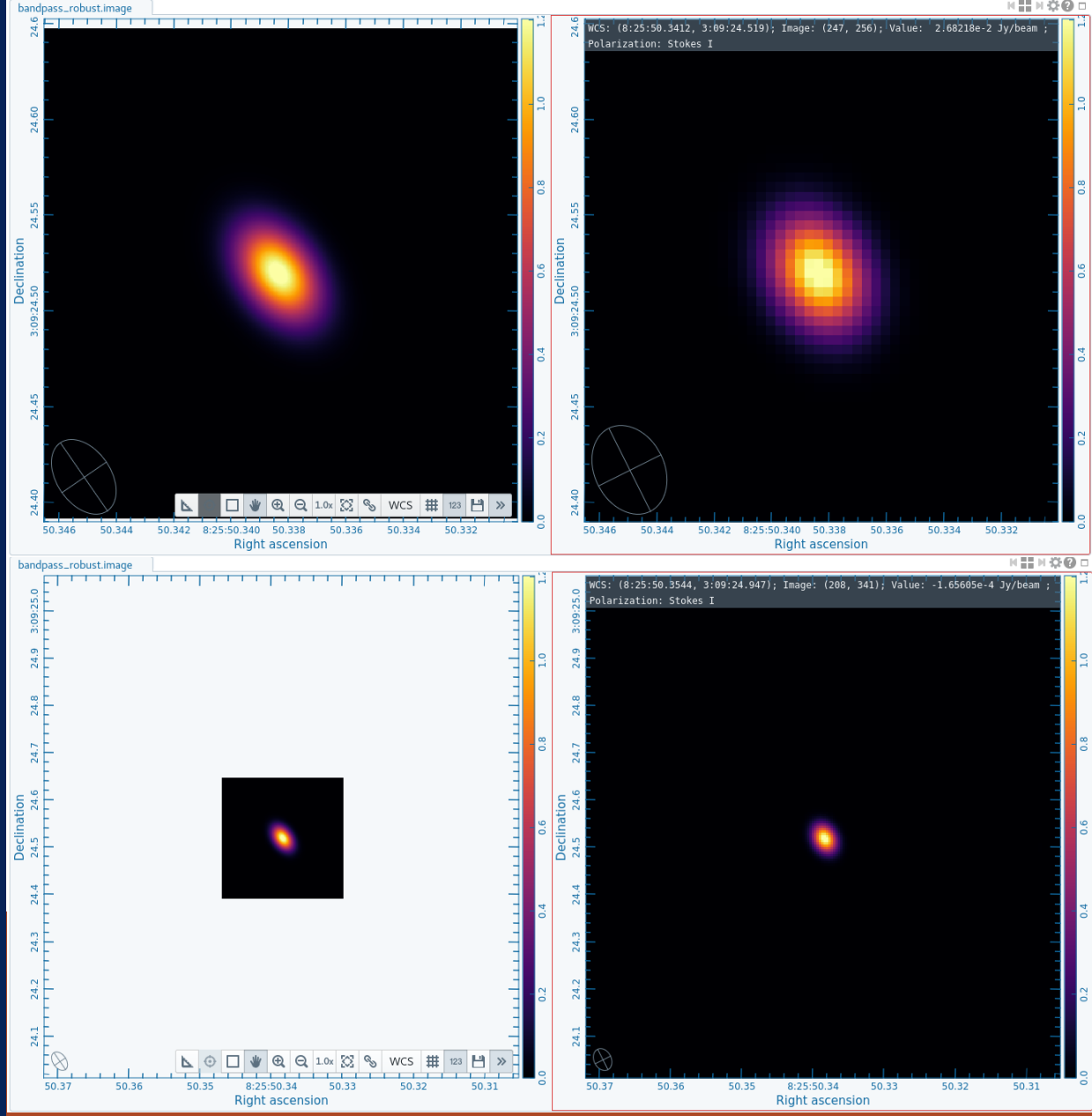
SDP.81 CO Line Imaging

Comparing big pixels to correct pixels



Question: Pixels

- What about really small pixels?
- Why would we want to avoid really small pixels?



Imaging

Basics of Imaging Review

Bandpass Calibrator Imaging

CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging

Imaging SDP.81 Continuum

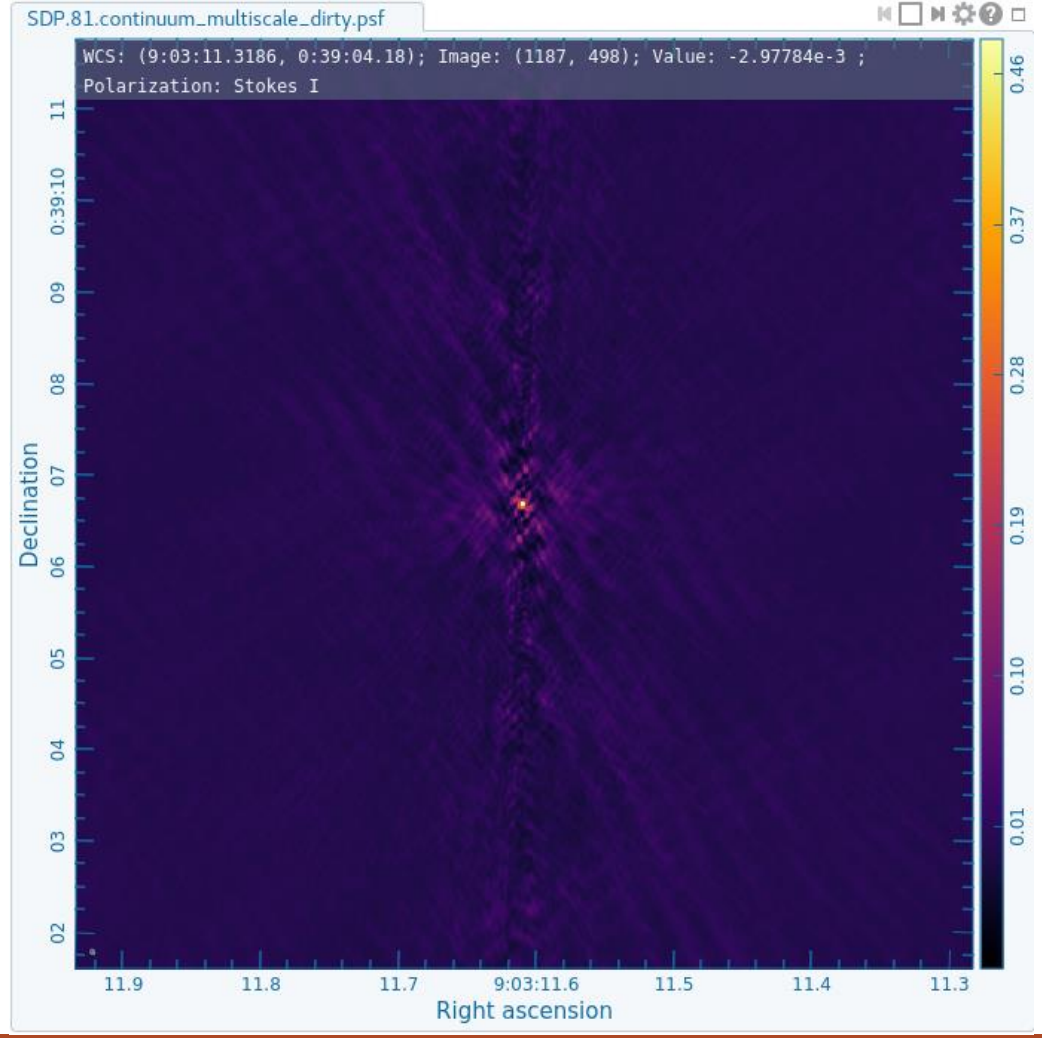
We will image the continuum emission in SDP.81 using a multiscale clean. For more information on multiscale cleaning, see the information/references in your imaging.py script.

NOTE: After clean starts, immediately press the red stop button. This will create a dirty image.

```
tclean(vis='SDP.81_Band4_continuum.ms',
  imagename='SDP.81.continuum_multiscale',
  spw='',
  field='SDP*',
  specmode='mfs',
  gridder='standard',
  deconvolver='multiscale',
  imsize=1500,
  cell='0.01arcsec',
  scales=[0,5,15,45],
  interactive=True,
  mask = '',
  weighting='briggs',
  robust=1.0,
  niter=10000,
  threshold='0.02mJy')
```

Question:

- How would you classify this PSF? Are the side lobes high or low?



PSF from the SDP.81 observation. Robust = 1.0

Imaging

Basics of Imaging Review

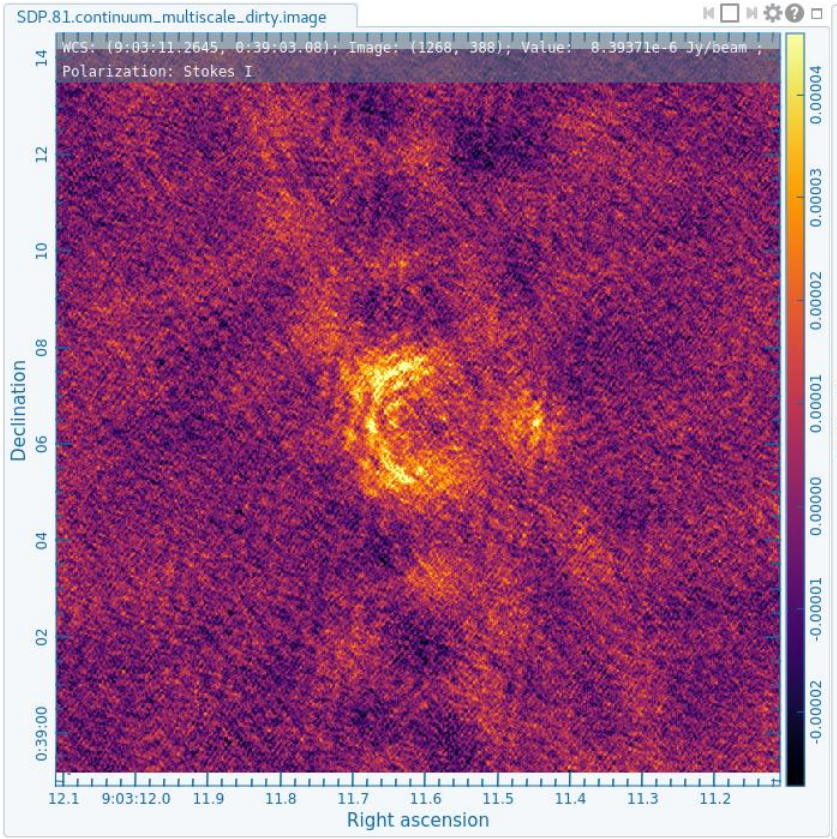
Bandpass Calibrator Imaging

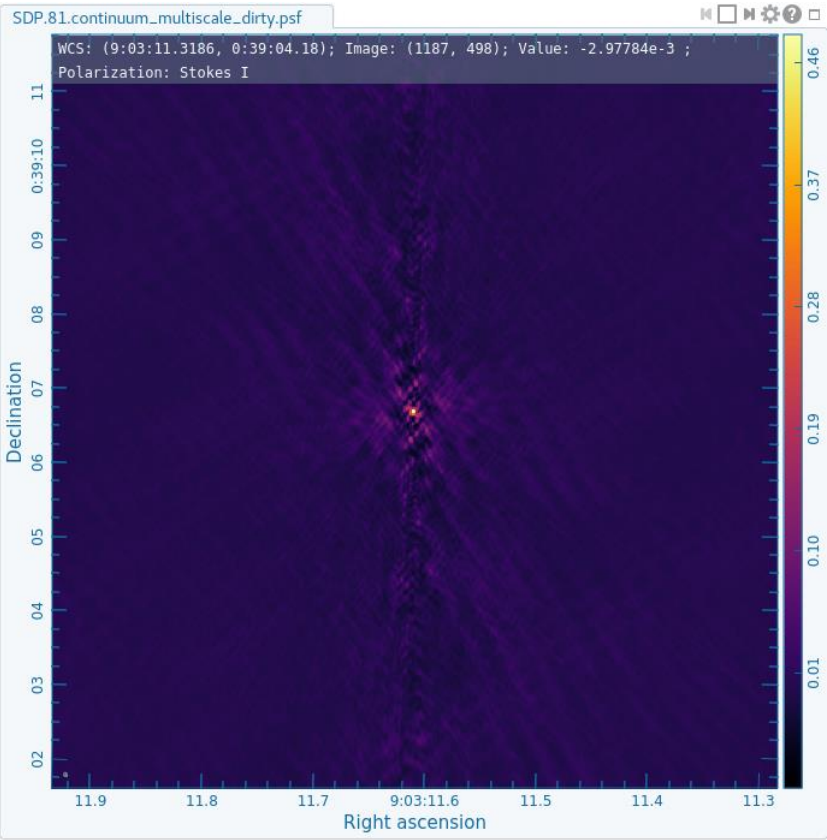
CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging

Rerun the previous command. Tclean will pick up where you left off. Finish cleaning the image.





Imaging

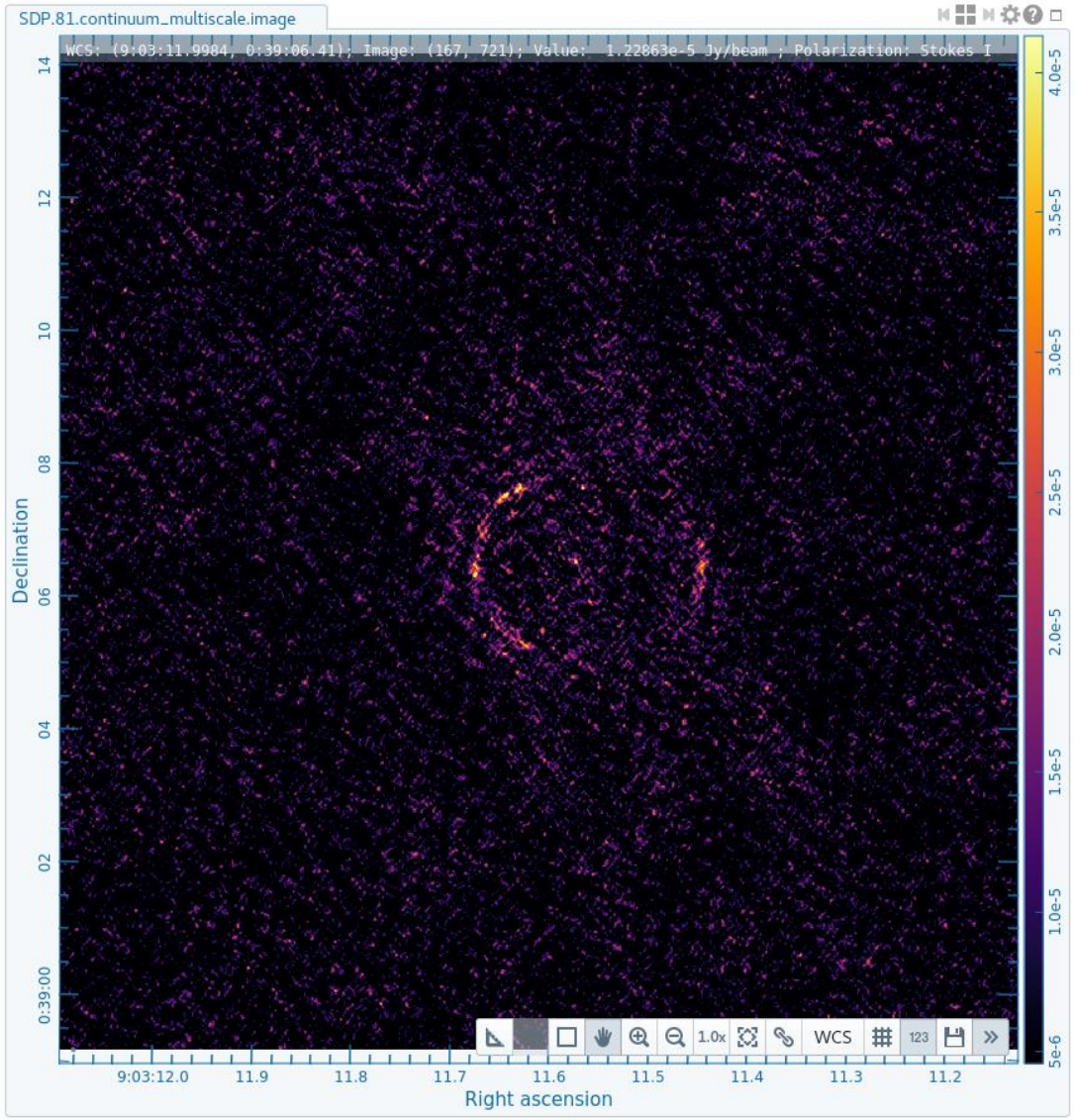
Basics of Imaging Review

Bandpass Calibrator Imaging

CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging



Imaging

Basics of Imaging Review

Bandpass Calibrator Imaging

CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging

Since the source is very weak, we can use UV tapering and a higher robust value to increase our SNR.

```
tclean(vis="SDP.81.Band4_continuum.ms",
imagename="SDP.81.continuum_smooth",
spw="",
field="SDP*",
specmode="mfs",
gridder="standard",
deconvolver="multiscale",
imsize=1500,
cell="0.01arcsec",
scales=[0,5,15,45],
interactive=True,
mask="",
weighting="briggs",
robust=1.0,
uvtaper=["1000klambda"],
niter=10000,
threshold="0.025mJy")
```

I recommend changing this to `robust = 2.0`

Imaging

Basics of Imaging Review

Bandpass Calibrator Imaging

CARTA

SDP.81 Continuum Imaging

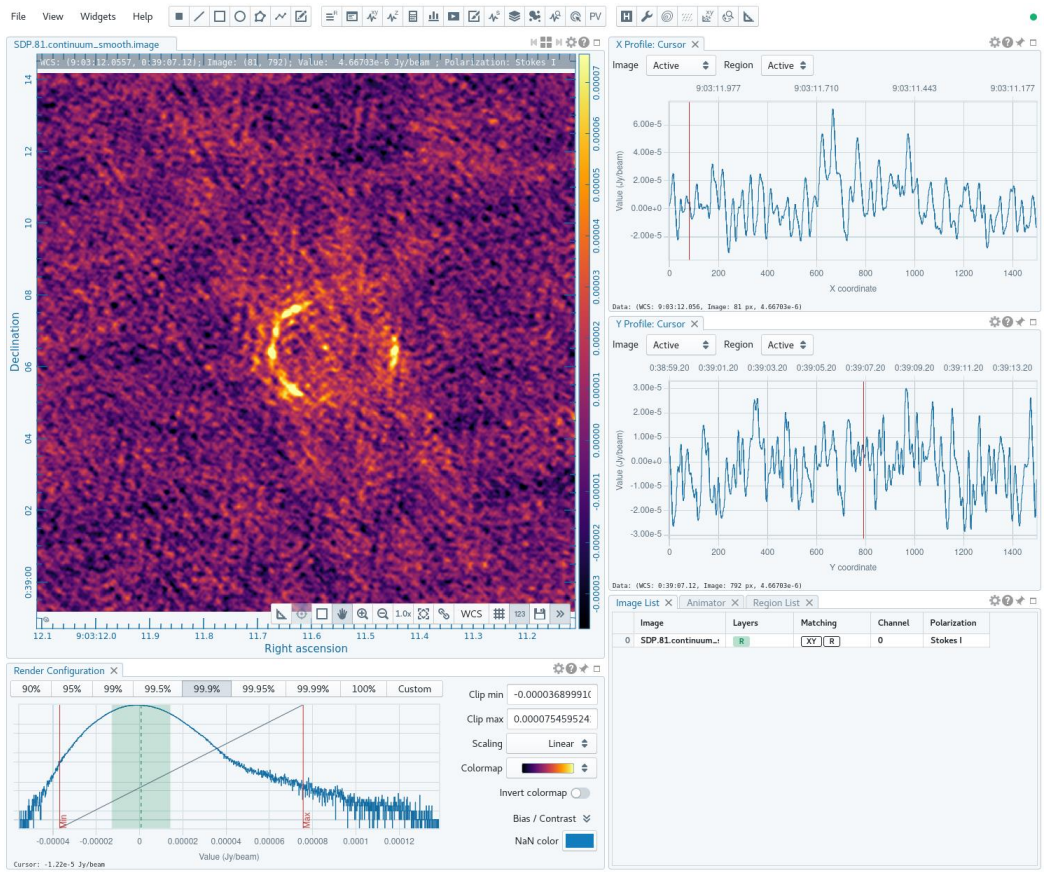
SDP.81 CO Line Imaging

See if you can clean this one yourself!

When finished cleaning, you should open both this smoothed image and the regular image in CARTA and compare them.

Question:

- What improved between when using a UVtaper?
- What worsened when using a UV taper?



SDP.81 with Robust = 1.0 and UV taper

Imaging

The spectral line we want to image today is CO(5-4) [rest frequency = 576.2679305 GHz] at a redshift of $z = 3.042$.

The first step is to continuum subtract the data. This can take some time, so we have done it for you. However, I want talk about the steps.

First, you can use plotms to try to plot the data and locate the location of the line.

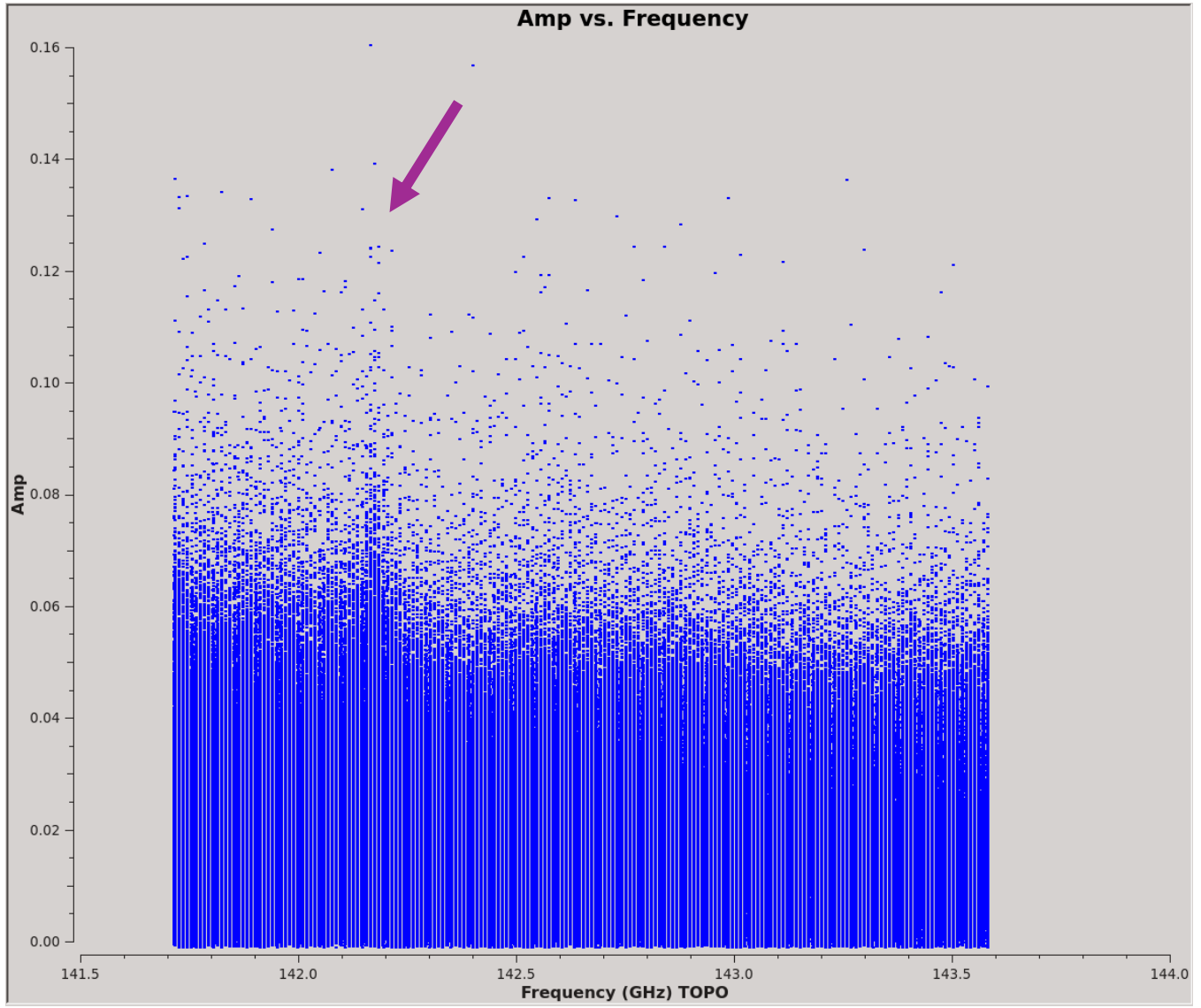
Imaging

Basics of Imaging Review

Bandpass Calibrator Imaging

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging



Imaging

After locating the line channels, split out a new MS, and run uvcontsub:

```
spw_line = '3,7,11,15,19,23,27,31,35,39,43,47'

split(vis='SDP.81_Band4.ms',
  outputvis='SDP.81_Band4_COline.ms',
  spw=spw_line,
  datacolumn='data')

uvcontsub(vis="SDP.81_Band4_COline.ms",
  fitorder=1,
  fitspw="0~11:5~45:170~187")
```

Imaging

Basics of Imaging Review

Bandpass Calibrator Imaging

CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging

T(cleaning) the cube!

```
tclean(vis='SDP.81_Band4_COline.ms.contsub',
   imagename='SDP.81.Band4.CO_smooth',
   specmode='cube',
   deconvolver='multiscale',
   imsize=672,
   cell='0.02arcsec',
   start='-520km/s',
   width='21km/s',
   nchan=45,
   outframe='LSRK',
   restfreq='142.5700GHz', #576.2679305 GHz
   scales=[0,5,15,45],
   interactive=True,
   restoringbeam='common',
   weighting='briggsbwtaper',
   robust=1.0,
   uvtaper=['1000klambda'],
   perchanweightdensity = True,
   niter=10000,
   threshold='0.52mJy')
```

Imaging

Basics of Imaging Review

Bandpass Calibrator Imaging

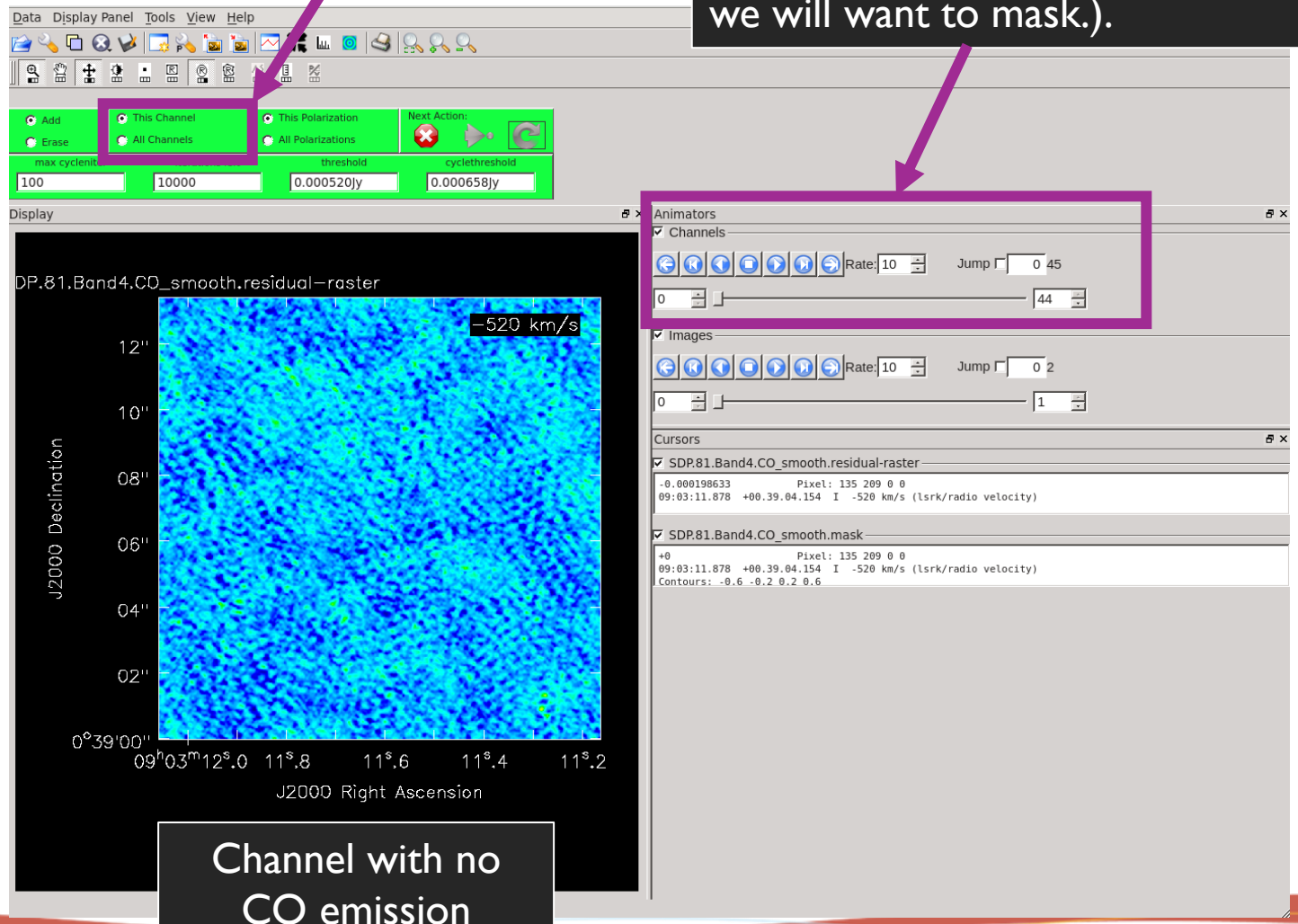
CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging

Can mask all channels or each individual channel

Moving through channels using the arrows shows which channels have CO line emission in them (which we will want to mask.).



Channel with no CO emission

Imaging

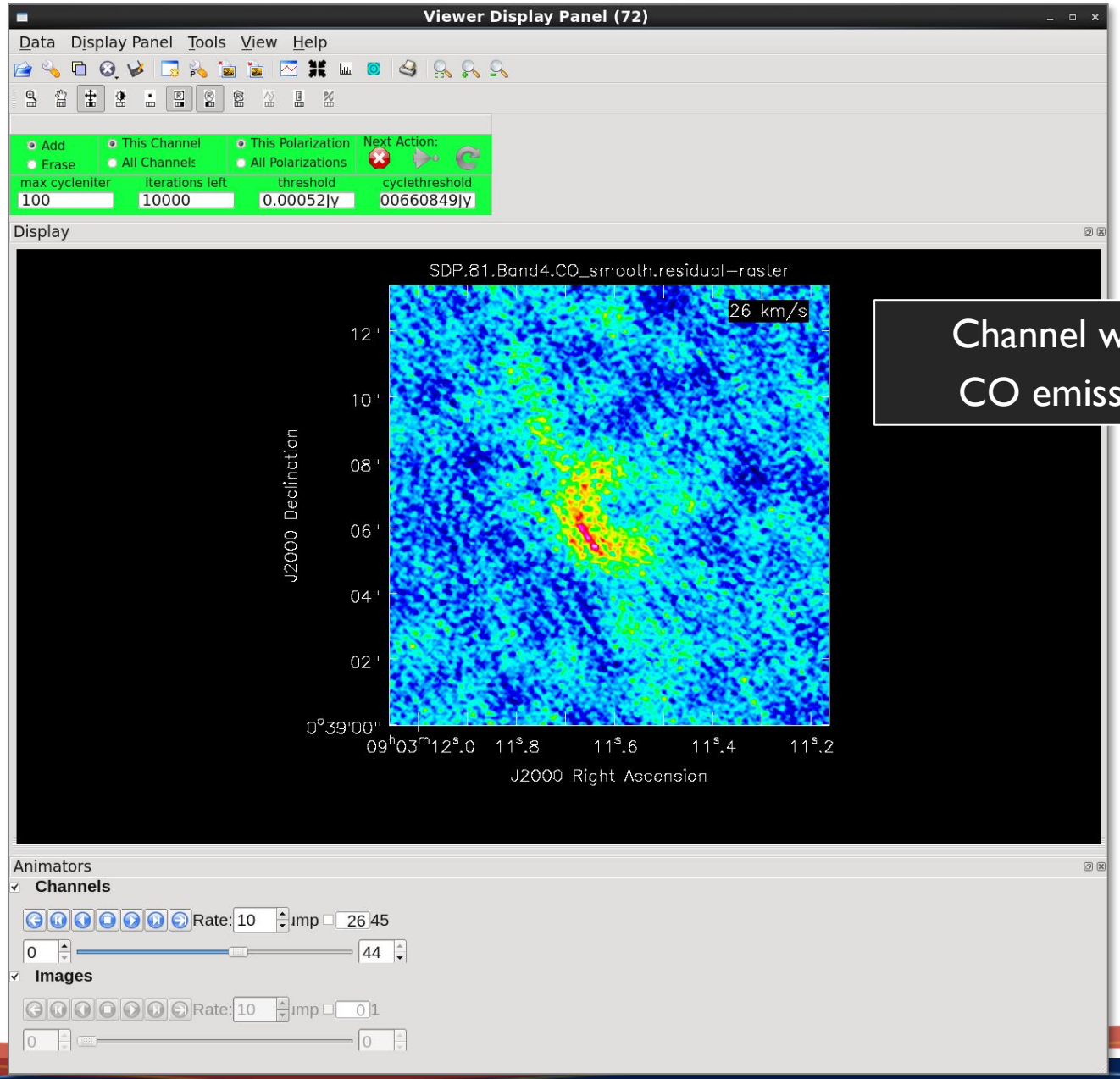
Basics of Imaging Review

Bandpass Calibrator Imaging

CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging



Channel with CO emission

Imaging

Basics of Imaging Review

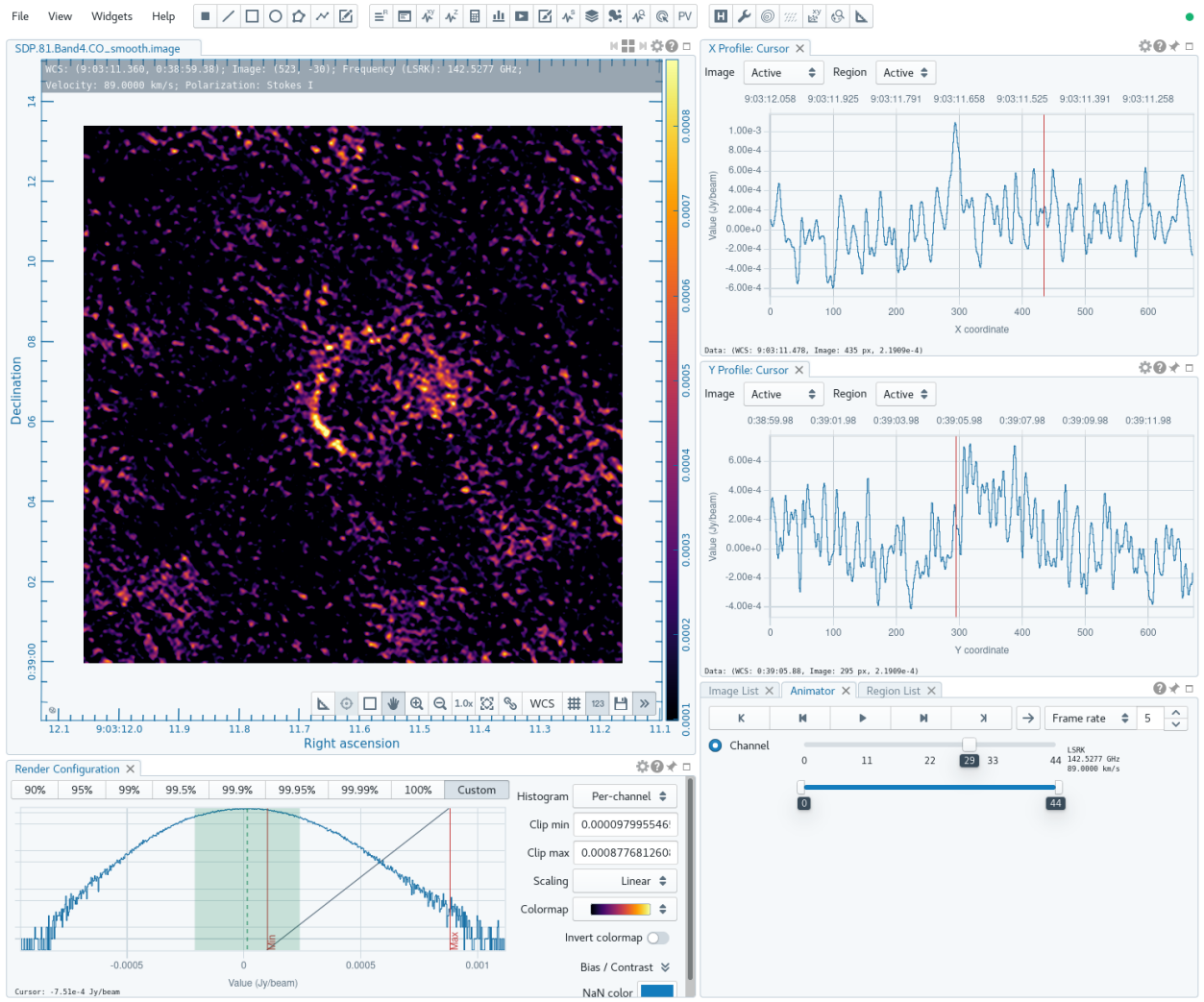
Bandpass Calibrator Imaging

CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging

Channel 29 of the CO cube



National Radio Astronomy Observatory

Imaging

Basics of Imaging Review

Bandpass Calibrator Imaging

CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging

Moment maps can be created from spectral line images.

- moments = -1 - mean value of the spectrum
- moments = 0 - integrated value of the spectrum
- moments = 1 - intensity weighted coordinate; traditionally used to get “velocity fields”
- moments = 2 - intensity weighted dispersion of the coordinate; traditionally used to get “velocity dispersion”
- moments = 3 - median value of the spectrum
- moments = 4 - median coordinate
- moments = 5 - standard deviation about the mean of the spectrum
- moments = 6 - root mean square of the spectrum
- moments = 7 - absolute mean deviation of the spectrum
- moments = 8 - maximum value of the spectrum
- moments = 9 - coordinate of the maximum value of the spectrum
- moments = 10 - minimum value of the spectrum
- moments = 11 - coordinate of the minimum value of the spectrum

Imaging

Basics of Imaging Review

Bandpass Calibrator Imaging

CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging

```
immoments(imagename='SDP.81.Band4.CO_smooth.image’,
  chans='10~36’,
  moments=[0],
  includepix=[2*0.20e-3,100], # -300 to +240 km/s
  outfile='SDP.81.Band4.CO_smooth.image.mom0_2sigma')

immoments(imagename='SDP.81.Band4.CO_smooth.image’,
  chans='10~36’,
  moments=[1],
  includepix=[4*0.20e-3,100], # -300 to +240 km/s
  outfile='SDP.81.Band4.CO_smooth.image.mom1_4sigma’)
```

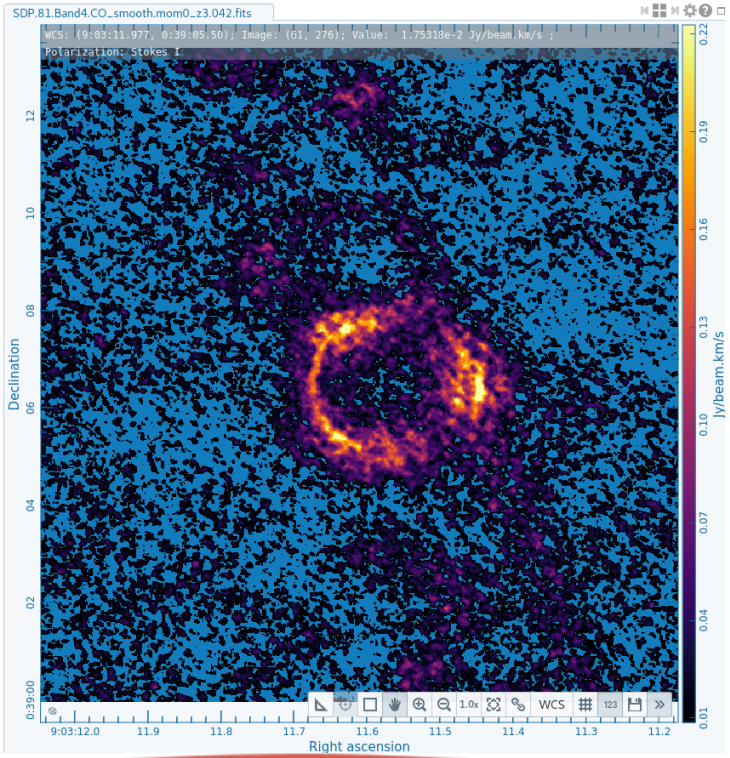
These commands will create the moment maps. We are making a integrated intensity map and an intensity weighted velocity map (velocity dispersion)

Question

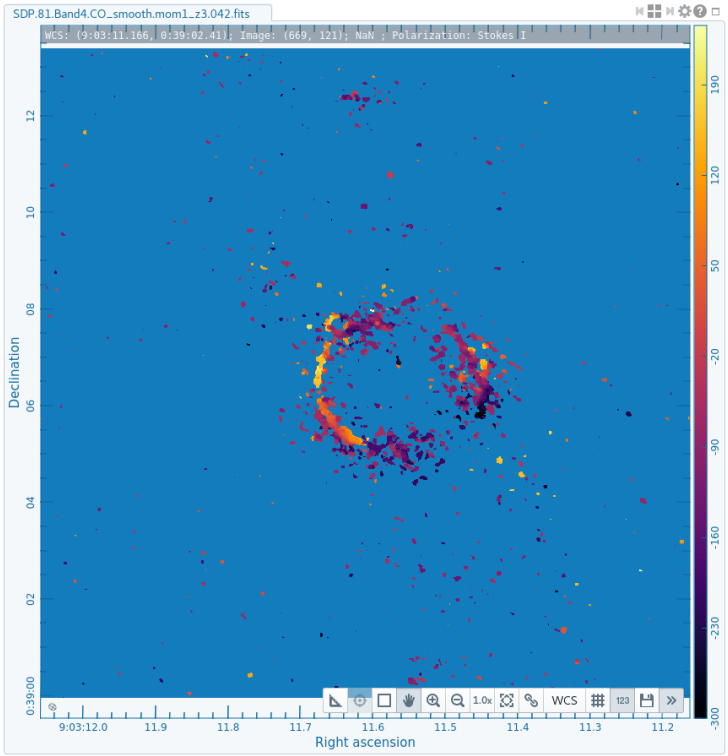
- Why did we set limits for includepix? Why not include all data?

Moment Maps for SDP.81

Moment zero map



Moment 1 map



Imaging

Basics of Imaging Review

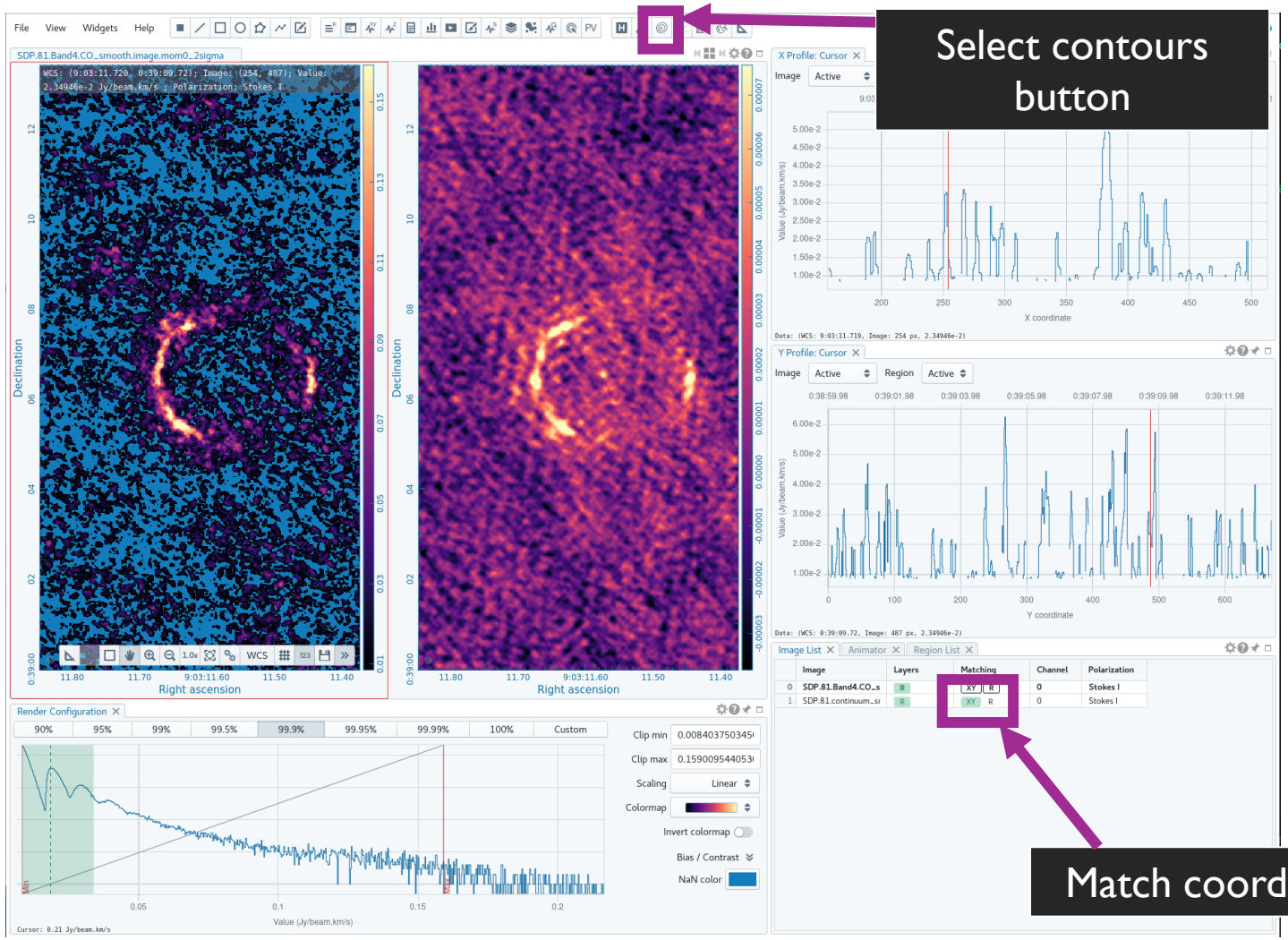
Bandpass Calibrator Imaging

CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging

Lets add continuum contours to the spectral lines.



Imaging

Basics of Imaging Review

Bandpass Calibrator Imaging

CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging

Select Continuum Image

Generate / set your contour levels.

Clear contours from image

Apply contours to image

Imaging

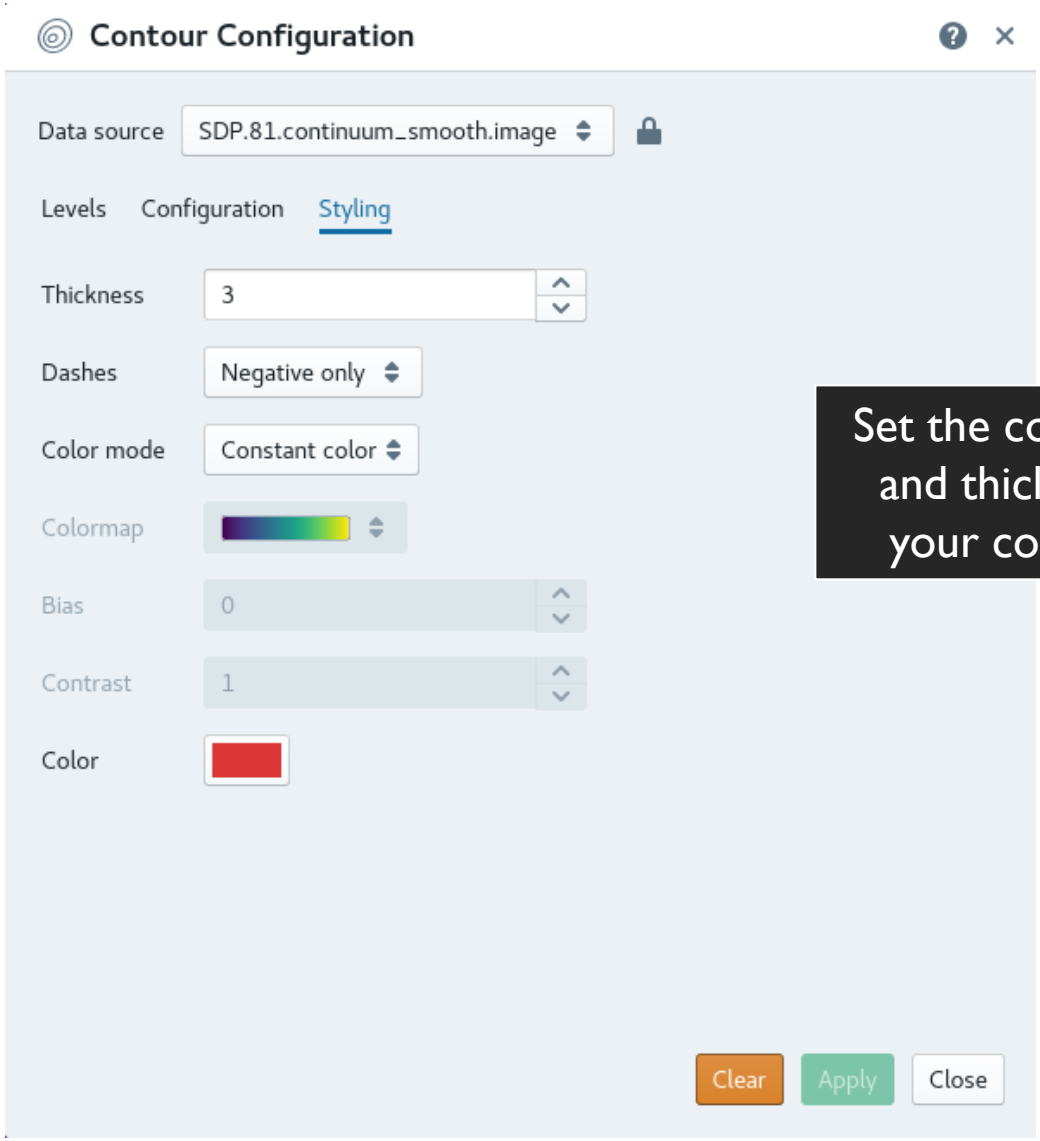
Basics of Imaging Review

Bandpass Calibrator Imaging

CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging



Set the color, style, and thickness of your contours!

Imaging

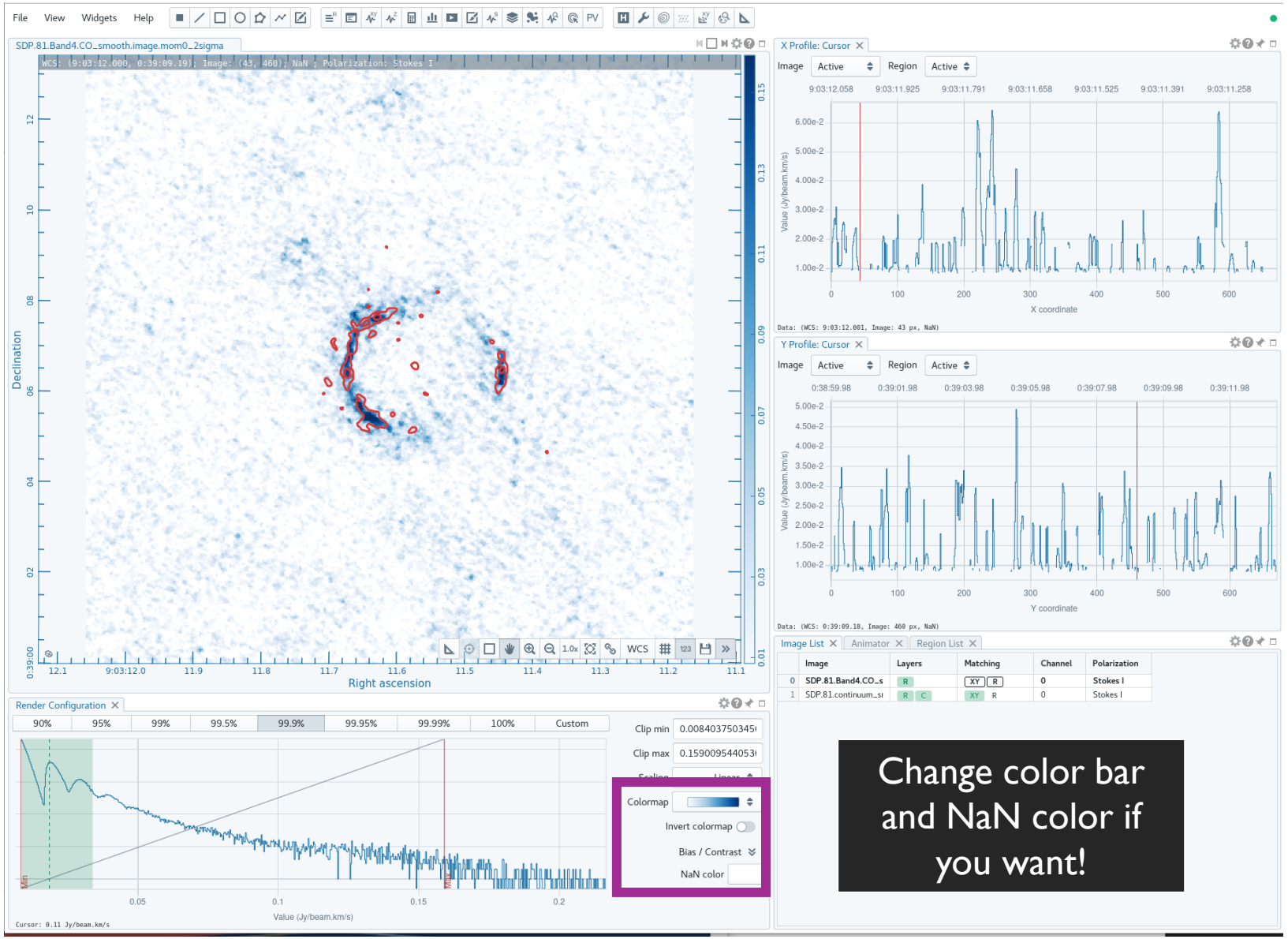
Basics of Imaging Review

Bandpass Calibrator Imaging

CARTA

SDP.81 Continuum Imaging

SDP.81 CO Line Imaging



National Radio Astronomy Observatory

And we are finished!

- Thanks for sticking with me though that long workshop!
- If you have any further questions, feel free to ask!

- If you have problems or need to ask questions later, contact us at the ALMA help desk! We are happy to help with your technical questions!

Additional Resources:

- ALMA CASA Guides:
  - https://casaguides.nrao.edu/index.php?title=ALMAguides
- ALMA Documentation
  - https://almascience.nrao.edu/documents-and-tools
- ALMA Helpdesk:
  - https://help.almascience.org/

Thank You!