VLA Data Reduction Tutorial: Continuum calibration and imaging (incl. polarimetry)

Anna D. Kapińska (NRAO)

VLA Data Reduction

This tutorial will take you through the VLA radio data reduction process [*manual calibration*].

Parts of this process are automated for the VLA [*pipeline*], which you can use once you understand the process and if suitable for your science goals. The VLA calibration pipeline will be introduced later in this workshop.

This talk is based on extensive and detailed online VLA tutorials (http://casaguides.nrao.edu), specific ones will be indicated on the slides as we go.

This tutorial steps

Part I

- Data inspection
- Calibration (Stokes I)

Part II

- Calibration (Stokes QUV)
- Imaging (Stokes IQUV)
- Primary beam, polarisation images & image analysis

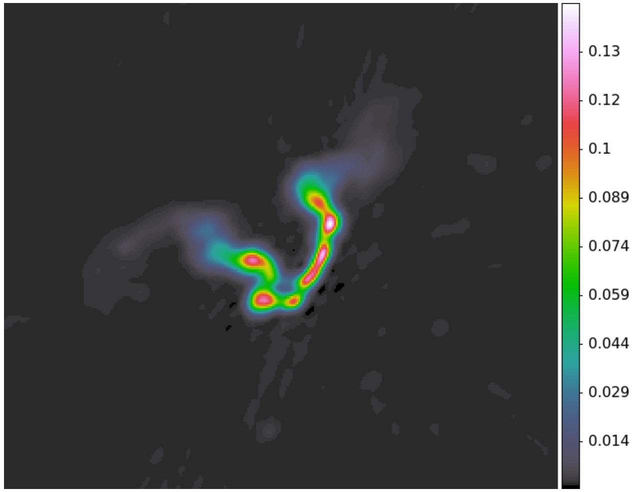
VLA Data Reduction Tutorial

Part I

Data set & files

Data: Single pointing, continuum observations of a binary black hole system 3C75 residing in the galaxy cluster Abell 400.

[Project code TDRW0001]



Observed in October 2018 with VLA in D conf.

S-band (3 GHz) with 1 GHz bandwidth and full polarisation (RR,LL,RL,LR)

Resolution 17 arcsec, FOV 14 arcmin, and LAS 8.2 arcmin

Source size ~9 arcmin → single pointing

Data set & files

Files: 3C75.tar

uncompress the file

```
~$ tar -xvf 3C75.tar
```

In the directory you will find files and folders we will use in this tutorial:

commands.txt	*(text file with commands used in this tutorial)*
3C75_raw.ms	*(uncalibrated data set)*
3C75_initcalib.ms	*(data set with initial calibration applied)*
3C75_calibrated.ms	*(data set fully calibrated for Stokes I)*
3C75_target_spw4.ms	*(fully calibrated excerpt of target data: 2 scans, 1 spw)*
3C75_target_full.ms	*(fully calibrated target data)*
3C75_final.image.tt0	*(multi-scale, multi-freq Stokes I image)*
3C75_final.POLI.image	*(multi-scale, multi-freq linear polarisation intensity image)*
3C75_final.POLA.masked.image	*(multi-scale, multi-freq polarisation angle image*
3C75_IQUVcube.image	*(multi-scale Stokes IQUV image cube)*

And some backup copies of tables/images which we will create in this tutorial:

copies/3C75_initcalib.B0	*(table with bandpass solutions)*
copies/3C75_calibrated.D1	*(table with leakage terms)*
copies/3C75_spw4.image.tt0	(*multi-scale, multi frequency Stokes I image of spw 4*)

Observer Logs & Online flags

Online Flags

→ Online flags are created during the observing session, such as subreflector issues and antennas not being on source

Make sure you apply these while downloading your data from the NRAO archive!

Observer Logs

→ Check the observer logs before starting
- information in weather during the observation
- record of problematic antennas that may need a priori flagging

→ Observer logs available on the NRAO science data archive

(note, if the VLA operator reports e.g. antenna 10 was down, it means antenna *ea10* just as referenced in CASA)

CASA startup (5.4.0 version)

> casa

```
=========================================
The start-up time of CASA may vary
depending on whether the shared libraries
are cached or not.
=========================================

IPython 5.1.0 -- An enhanced Interactive Python.

CASA 5.4.0-70   -- Common Astronomy Software Applications

--> CrashReporter initialized.
Enter doc('start') for help getting started with CASA...
Using matplotlib backend: TkAgg

CASA <1>: 
```

Interactive slides: Tasks that we will go through manually in this tutorial will be highlighted in **sky blue** and slides are marked with red/yellow numbers. You can run CASA tasks either via 'input' method, or via scripted method (*all tasks are in the commands.txt file*).

1

Non-interactive slides: Tasks will be highlighted in **navy blue**.

Radio Data Inspection

Important CASA tasks are:

– See summary of the data set **`listobs()`** DEMO: 1

– See a plot of antenna positions **`plotants()`** DEMO: 2

– Plot and inspect the visibilities **`plotms()`** DEMO: 3,4

– Flag RFI and bad data if necessary **`flagdata()`** DEMO: 5

Radio Data Inspection: listobs()

1

List the summary of the observation.

In CASA terminal type the following:

> `default listobs`	set the task parameters to default
> `inp`	see input parameters
> `vis = '3C75_raw.ms'`	choose data file (measurement set)
> `go`	execute the task

Now, check the **Casalogger**

Radio Data Inspection: listobs()

```
Time                 Priority  Origin          Message
2019-05-31 00:10:35 INFO      ...operties     Computing scan and subscan properties...
2019-05-31 00:10:35 INFO      ...::summary    Data records: 1036152       Total elapsed time = 1865 seconds
2019-05-31 00:10:35 INFO      ...:summary+      Observed from   04-Oct-2018/05:49:05.0   to   04-Oct-2018/06:20:10.0 (UTC)
2019-05-31 00:10:35 INFO      ...::summary
2019-05-31 00:10:35 INFO      ...:summary+      ObservationID = 0         ArrayID = 0
2019-05-31 00:10:35 INFO      ...:summary+     Date        Timerange (UTC)          Scan  FldId FieldName             nRows     SpwIds   Average Interval(s)    ScanIntent
2019-05-31 00:10:35 INFO      ...:summary+     04-Oct-2018/05:49:05.0 - 05:53:25.0     5      0 0137+331=3C48           146016  [0,1,2,3,4,5,6,7]  [5, 5, 5, 5, 5, 5, 5, 5] [CALIBRATE_BANDPASS#UNSPI
2019-05-31 00:10:35 INFO      ...:summary+                 05:53:30.0 - 05:57:55.0     6      1 J2355+4950              148824  [0,1,2,3,4,5,6,7]  [5, 5, 5, 5, 5, 5, 5, 5] [CALIBRATE_AMPLI#UNSPECII
2019-05-31 00:10:35 INFO      ...:summary+                 05:58:00.0 - 06:03:55.0     7      2 J0259+0747              199368  [0,1,2,3,4,5,6,7]  [5, 5, 5, 5, 5, 5, 5, 5] [CALIBRATE_AMPLI#UNSPECII
2019-05-31 00:10:35 INFO      ...:summary+                 06:04:00.0 - 06:18:55.0     8      3 3C75                    502632  [0,1,2,3,4,5,6,7]  [5, 5, 5, 5, 5, 5, 5, 5] [OBSERVE_TARGET#UNSPECIF
2019-05-31 00:10:35 INFO      ...:summary+                 06:19:00.0 - 06:20:10.0     9      2 J0259+0747               39312  [0,1,2,3,4,5,6,7]  [5, 5, 5, 5, 5, 5, 5, 5] [CALIBRATE_AMPLI#UNSPECII
2019-05-31 00:10:35 INFO      ...::summary             (nRows = Total number of rows per scan)
2019-05-31 00:10:35 INFO      ...::summary    Fields: 4
2019-05-31 00:10:35 INFO      ...:summary+     ID   Code Name                RA               Decl            Epoch   SrcId      nRows
2019-05-31 00:10:35 INFO      ...:summary+     0    NONE 0137+331=3C48       01:37:41.299431 +33.09.35.13299 J2000   0          146016
2019-05-31 00:10:35 INFO      ...:summary+     1    NONE J2355+4950          23:55:09.458169 +49.50.08.34001 J2000   1          148824
2019-05-31 00:10:35 INFO      ...:summary+     2    NONE J0259+0747          02:59:27.076633 +07.47.39.64322 J2000   2          238680
2019-05-31 00:10:35 INFO      ...:summary+     3    NONE 3C75                02:57:42.630000 +06.01.04.80000 J2000   3          502632
2019-05-31 00:10:35 INFO      ...::summary    Spectral Windows:  (8 unique spectral windows and 1 unique polarization setups)
2019-05-31 00:10:35 INFO      ...:summary+     SpwID  Name           #Chans   Frame   Ch0(MHz)  ChanWid(kHz)  TotBW(kHz) CtrFreq(MHz) BBC Num  Corrs
2019-05-31 00:10:35 INFO      ...:summary+     0      EVLA_S#A0C0#2      64   TOPO    2488.000      2000.000    128000.0   2551.0000       12  RR  RL  LR  LL
2019-05-31 00:10:35 INFO      ...:summary+     1      EVLA_S#A0C0#3      64   TOPO    2616.000      2000.000    128000.0   2679.0000       12  RR  RL  LR  LL
2019-05-31 00:10:35 INFO      ...:summary+     2      EVLA_S#A0C0#4      64   TOPO    2744.000      2000.000    128000.0   2807.0000       12  RR  RL  LR  LL
2019-05-31 00:10:35 INFO      ...:summary+     3      EVLA_S#A0C0#5      64   TOPO    2872.000      2000.000    128000.0   2935.0000       12  RR  RL  LR  LL
2019-05-31 00:10:35 INFO      ...:summary+     4      EVLA_S#A0C0#6      64   TOPO    3000.000      2000.000    128000.0   3063.0000       12  RR  RL  LR  LL
2019-05-31 00:10:35 INFO      ...:summary+     5      EVLA_S#A0C0#7      64   TOPO    3128.000      2000.000    128000.0   3191.0000       12  RR  RL  LR  LL
2019-05-31 00:10:35 INFO      ...:summary+     6      EVLA_S#A0C0#8      64   TOPO    3256.000      2000.000    128000.0   3319.0000       12  RR  RL  LR  LL
2019-05-31 00:10:35 INFO      ...:summary+     7      EVLA_S#A0C0#9      64   TOPO    3384.000      2000.000    128000.0   3447.0000       12  RR  RL  LR  LL
2019-05-31 00:10:35 INFO      ...::summary    Sources: 32
2019-05-31 00:10:35 INFO      ...:summary+     ID   Name                SpwId RestFreq(MHz)  SysVel(km/s)
2019-05-31 00:10:35 INFO      ...:summary+     0    0137+331=3C48       0     -              -
2019-05-31 00:10:35 INFO      ...:summary+     0    0137+331=3C48       1     -              -
2019-05-31 00:10:35 INFO      ...:summary+     0    0137+331=3C48       2     -              -
2019-05-31 00:10:35 INFO      ...:summary+     0    0137+331=3C48       3     -              -
2019-05-31 00:10:35 INFO      ...:summary+     0    0137+331=3C48       4     -              -
2019-05-31 00:10:35 INFO      ...:summary+     0    0137+331=3C48       5     -              -
2019-05-31 00:10:35 INFO      ...:summary+     0    0137+331=3C48       6     -              -
2019-05-31 00:10:35 INFO      ...:summary+     0    0137+331=3C48       7     -              -
2019-05-31 00:10:35 INFO      ...:summary+     1    J2355+4950          0     -              -
2019-05-31 00:10:35 INFO      ...:summary+     1    J2355+4950          1     -              -
2019-05-31 00:10:35 INFO      ...:summary+     1    J2355+4950          2     -              -
2019-05-31 00:10:35 INFO      ...:summary+     1    J2355+4950          3     -              -
2019-05-31 00:10:35 INFO      ...:summary+     1    J2355+4950          4     -              -
2019-05-31 00:10:35 INFO      ...:summary+     1    J2355+4950          5     -              -
2019-05-31 00:10:35 INFO      ...:summary+     1    J2355+4950          6     -              -
2019-05-31 00:10:35 INFO      ...:summary+     1    J2355+4950          7     -              -
2019-05-31 00:10:35 INFO      ...:summary+     2    J0259+0747          0     -              -
2019-05-31 00:10:35 INFO      ...:summary+     2    J0259+0747          1     -              -
2019-05-31 00:10:35 INFO      ...:summary+     2    J0259+0747          2     -              -
2019-05-31 00:10:35 INFO      ...:summary+     2    J0259+0747          3     -              -
nsert Message:                                   Lock scroll
```

Radio Data Inspection: listobs()

Summary of the observing strategy

Field ID	Field name	Intent
0	0137+331=3C48	Flux density scale calibrator Bandpass calibrator Polarisation angle calibrator
1	J2355+4950	Polarisation leakage calibrator
2	J0259+0747	Complex gain calibrator
3	3C75	Target

Frequency: 8 spws (ids: 0~7), 128 MHz each
total bandwidth 1024 MHz = 1 GHz, centered on 3GHz

Observing strategy:
* bracket target with complex gain calibrator
* observe flux/bandpass/pol angle calibrator at least once
* observe leakage calibrator once (if unpolarised)

[Only 5 scans included in this demo MS]

Radio Data Inspection: plotants()

2

Plot the antenna positions.

In CASA window type the following:

```
> default plotants

> inp

> vis = '3C75_raw.ms'

> go
```

Now, a new window with a plot will appear.

Radio Data Inspection: plotants()

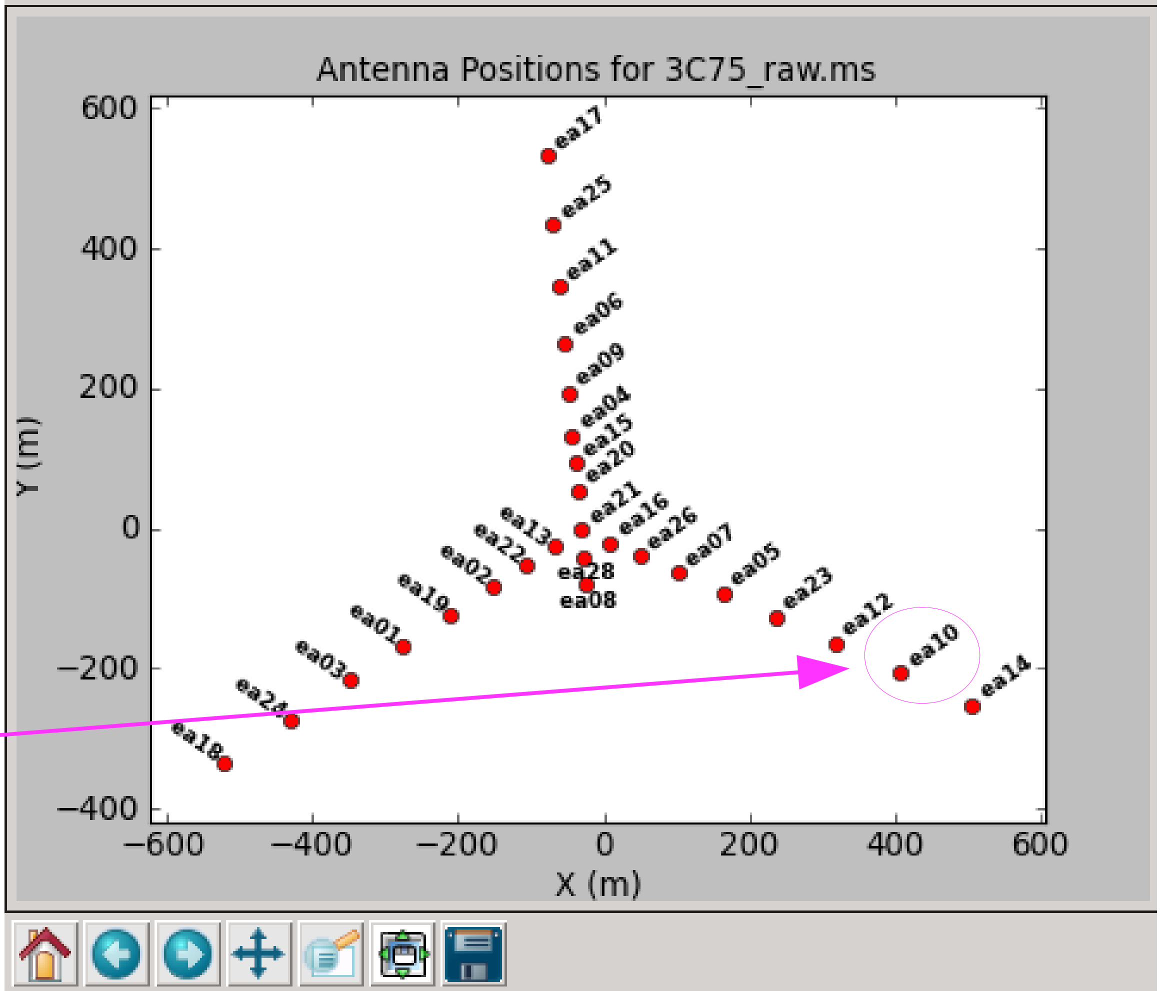
<u>For calibration:</u>

you need to select a *reference antenna*

Select one that has
(i) good data on all baselines, and
(ii) has good balance between number of short and long baselines

Here in configuration D we choose:
refant = 'ea10'

In larger configurations refant is often near the center of array



Radio Data Inspection: plotms()

3

Plot the data (visibilities): amplitude *vs* frequency.

```
plotms (vis='3C75_raw.ms', field='1', xaxis='frequency', yaxis='amp',
selectdata=True, averagedata=True, avgtime='1e100', coloraxis='spw')
```

In CASA window type the following:

[Scripting method for CASA]

```
> default plotms
> inp
```

Set parameters

```
> vis = '3C75_raw.ms'
> field = '1'
```

we pick complex gain calibrator to plot

```
> xaxis = 'frequency'
> yaxis = 'amp'
```

```
> selectdata = True
> averagedata = True
> avgtime = '1e100'
```

average data for faster loading

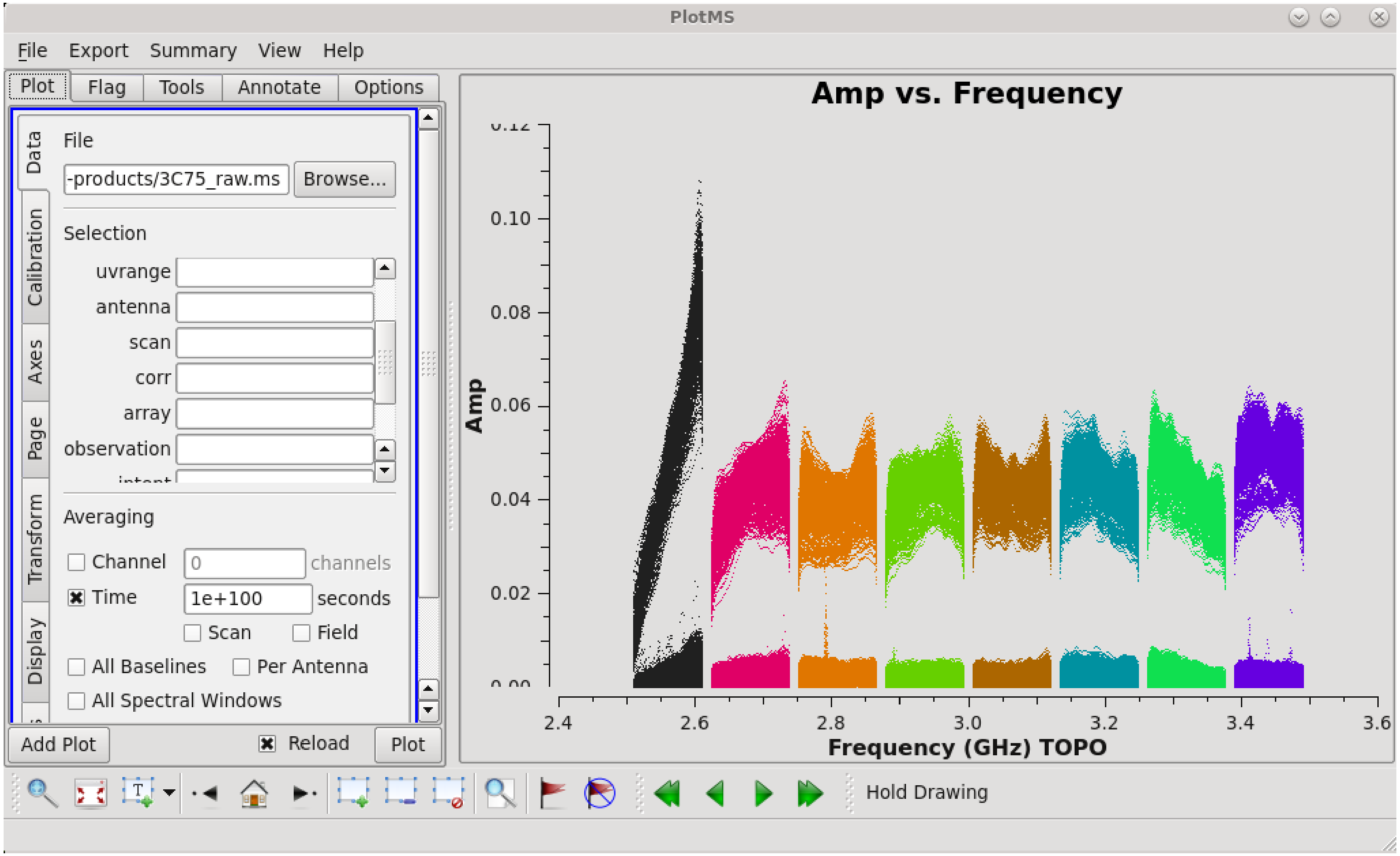
```
> coloraxis = 'spw'
```

display each spw in different colour

Double check input parameters and execute the task

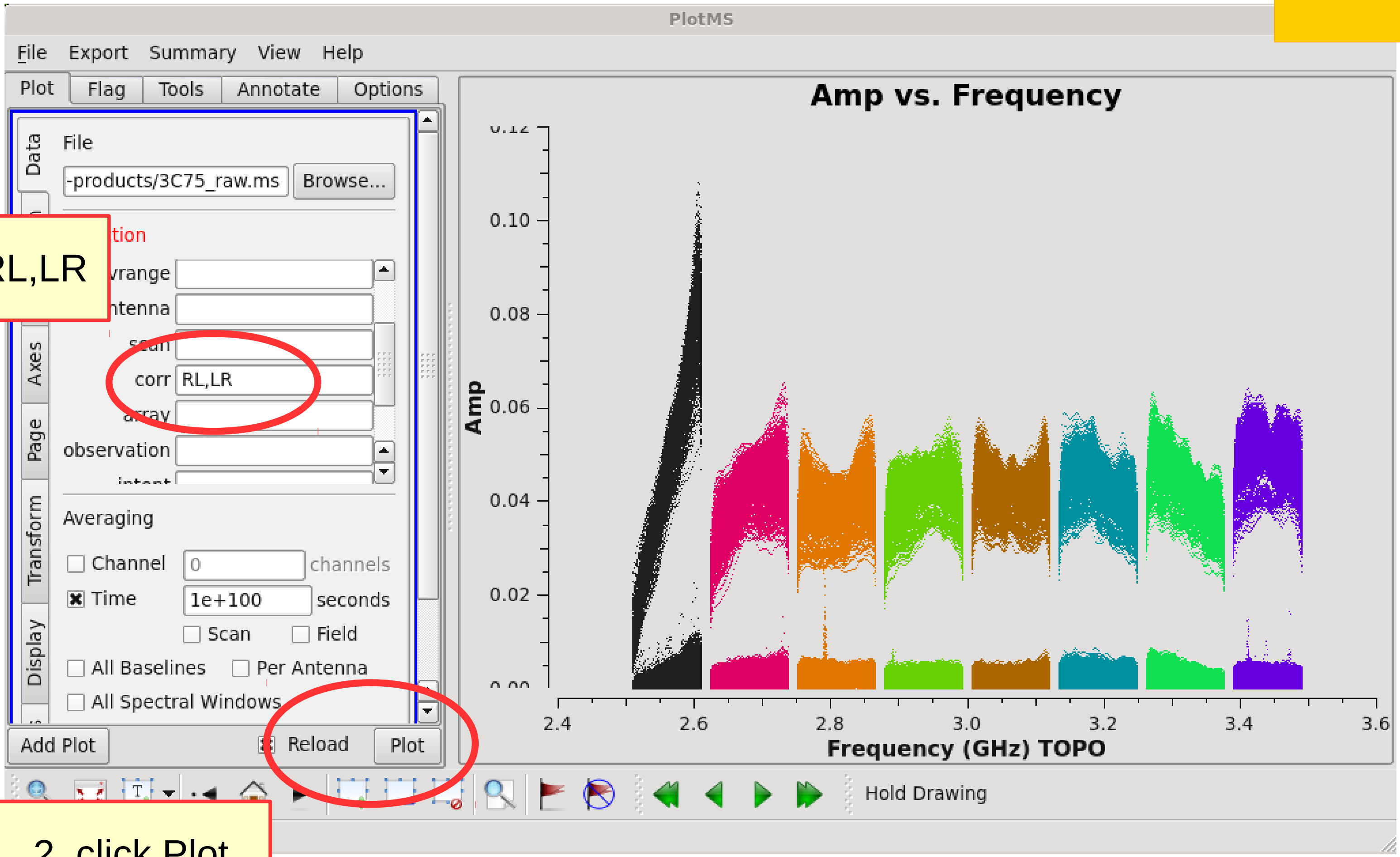
```
> inp
> go
```

Radio Data Inspection: plotms()



Radio Data Inspection: plotms()

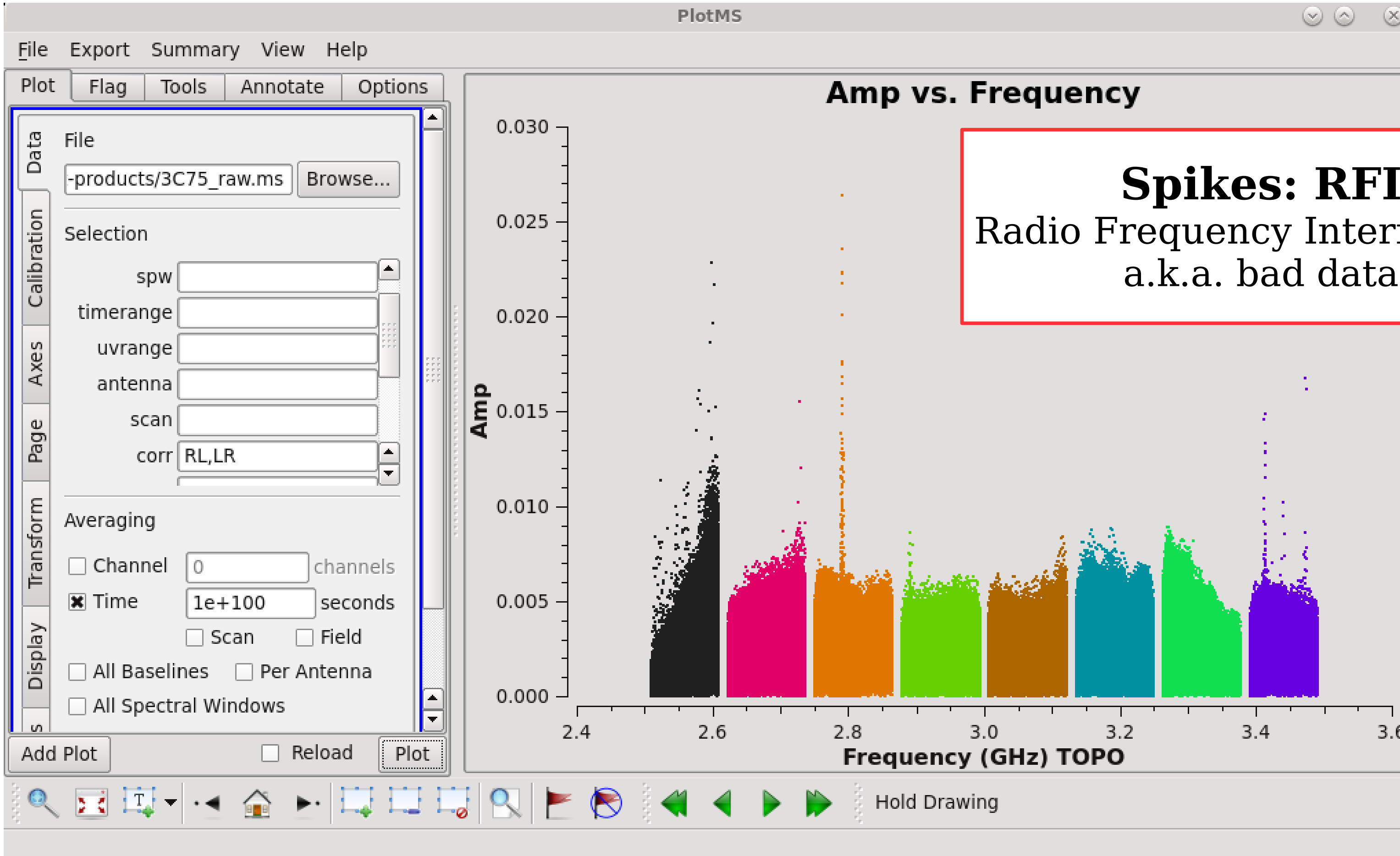
4



1. type RL,LR

2. click Plot

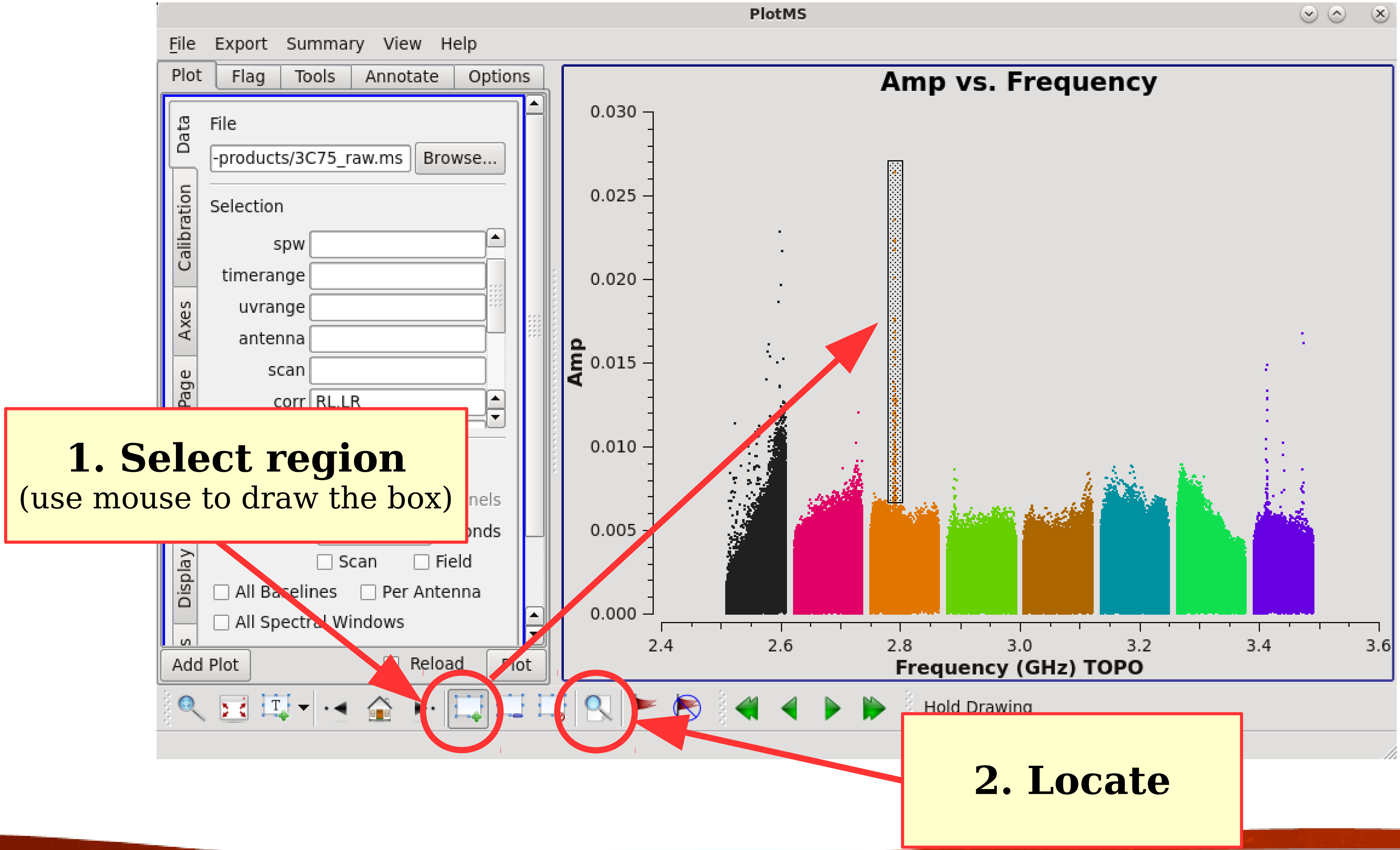
Radio Data Inspection: plotms()



Spikes: RFI

Radio Frequency Interference,
a.k.a. bad data

Radio Data Inspection: plotms()



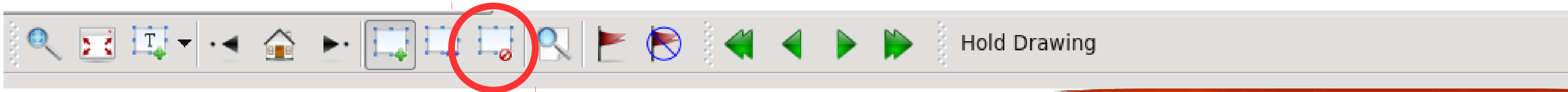
Radio Data Inspection: plotms()

Now, check the **Casalogger** (but <u>do not close</u> plotms)

Located RFI data points are: field id=1, spw=2, channel=22~24

```
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea11@N07 & ea25@N08 [10&24] Spw=2 Chan=22 Freq=2.788 Corr=LR X=2.788 Y=0.00784305 Observation=0
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea11@N07 & ea25@N08 [10&24] Spw=2 Chan=23 Freq=2.79 Corr=LR X=2.79 Y=0.01531 Observation=0
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea11@N07 & ea25@N08 [10&24] Spw=2 Chan=24 Freq=2.792 Corr=LR X=2.792 Y=0.00688168 Observation=0
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea15@N03 & ea20@N02 [14&19] Spw=2 Chan=22 Freq=2.788 Corr=LR X=2.788 Y=0.0116323 Observation=0
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea15@N03 & ea20@N02 [14&19] Spw=2 Chan=23 Freq=2.79 Corr=RL X=2.79 Y=0.00939895 Observation=0
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea15@N03 & ea20@N02 [14&19] Spw=2 Chan=23 Freq=2.79 Corr=LR X=2.79 Y=0.0223639 Observation=0
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea15@N03 & ea20@N02 [14&19] Spw=2 Chan=24 Freq=2.792 Corr=LR X=2.792 Y=0.0118626 Observation=0
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea15@N03 & ea21@N01 [14&20] Spw=2 Chan=22 Freq=2.788 Corr=LR X=2.788 Y=0.0068914 Observation=0
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea15@N03 & ea21@N01 [14&20] Spw=2 Chan=23 Freq=2.79 Corr=LR X=2.79 Y=0.0119754 Observation=0
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea15@N03 & ea21@N01 [14&20] Spw=2 Chan=24 Freq=2.792 Corr=LR X=2.792 Y=0.00692457 Observation=0
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea15@N03 & ea28@W01 [14&26] Spw=2 Chan=22 Freq=2.788 Corr=LR X=2.788 Y=0.011995 Observation=0
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea15@N03 & ea28@W01 [14&26] Spw=2 Chan=23 Freq=2.79 Corr=LR X=2.79 Y=0.0235919 Observation=0
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea15@N03 & ea28@W01 [14&26] Spw=2 Chan=24 Freq=2.792 Corr=LR X=2.792 Y=0.011328 Observation=0
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea16@E02 & ea26@E03 [15&25] Spw=2 Chan=23 Freq=2.79 Corr=LR X=2.79 Y=0.00738353 Observation=0
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea17@N09 & ea25@N08 [16&24] Spw=2 Chan=23 Freq=2.79 Corr=RL X=2.79 Y=0.0130654 Observation=0
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea17@N09 & ea25@N08 [16&24] Spw=2 Chan=23 Freq=2.79 Corr=LR X=2.79 Y=0.0114727 Observation=0
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea20@N02 & ea21@N01 [19&20] Spw=2 Chan=23 Freq=2.79 Corr=RL X=2.79 Y=0.00982026 Observation=0
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea20@N02 & ea21@N01 [19&20] Spw=2 Chan=23 Freq=2.79 Corr=LR X=2.79 Y=0.00675957 Observation=0
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea20@N02 & ea28@W01 [19&26] Spw=2 Chan=23 Freq=2.79 Corr=RL X=2.79 Y=0.00754542 Observation=0
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea20@N02 & ea28@W01 [19&26] Spw=2 Chan=23 Freq=2.79 Corr=LR X=2.79 Y=0.0121364 Observation=0
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea20@N02 & ea28@W01 [19&26] Spw=2 Chan=24 Freq=2.792 Corr=LR X=2.792 Y=0.00743906 Observation=0
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea21@N01 & ea28@W01 [20&26] Spw=2 Chan=22 Freq=2.788 Corr=LR X=2.788 Y=0.00908199 Observation=0
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea21@N01 & ea28@W01 [20&26] Spw=2 Chan=23 Freq=2.79 Corr=RL X=2.79 Y=0.00755853 Observation=0
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea21@N01 & ea28@W01 [20&26] Spw=2 Chan=23 Freq=2.79 Corr=LR X=2.79 Y=0.0217939 Observation=0
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea21@N01 & ea28@W01 [20&26] Spw=2 Chan=24 Freq=2.792 Corr=LR X=2.792 Y=0.0128751 Observation=0
…7 INFO   …::locate+ Scan=6 Field=J2355+4950 [1] Time=2018/10/04/05:55:40.0000 BL=ea21@N01 & ea28@W01 [20&26] Spw=2 Chan=25 Freq=2.794 Corr=LR X=2.794 Y=0.007476 Observation=0
…7 INFO   …::locate+ Found 90 points (90 unflagged) among 359424 in 0.02s.
```

If you want to check more RFI, clear regions in plotms() and repeat steps 1& 2:



Radio Data Editing: flagdata()

5

Remove (flag) bad data: outliers and RFI.

In CASA window type the following:

```
> default flagdata
> inp
```

Set parameters

```
> vis = '3C75_raw.ms'
> mode = 'manual'
> field = '1'
> spw = '2:22~24'
```

← set mode to manual

← select channels 22 to 24 in spw 2

Double check input parameters and execute the task

```
> inp
> go
```

Now, in `plotms()` tick 

and click 

CASA Flagging Tutorial:
https://casaguides.nrao.edu/index.php/VLA_CASA_Flagging-CASA5.4.0

Calibration strategy (Stokes I)

Standard calibration steps (total intensity = Stokes I)
– basic CASA task sequence:

Low/high frequency tutorials:
https://casaguides.nrao.edu

– Set the model of flux density scale calibrator **`setjy()`** DEMO: 6

– Produce auxillary tables not dependent on data **`gencal()`**

– Delay calibration **`gaincal()`**

– Initial complex gain calibration **`gaincal()`**

– Bandpass calibration **`bandpass()`** DEMO: 7

– Complex gain calibration: phases & amp **`gaincal()`**

– Bootstrapping the flux densities of remaining calibrators **`fluxscale()`**

– Apply the calibration to all fields **`applycal()`** DEMO: 8

Calibration: flux density model

6

First, set the flux density scale model

– VLA uses a few bright sources as standard flux density scale calibrators, and has well curated models of these sources. These calibrators will set the absolute flux density scale of your data.

– use task `setjy()` to set the model

```
> vis = '3C75_initcalib.ms'
> field = '0'
> model = '3C48_S.im'
> standard = 'Perley-Butler 2017'
```

our flux density scale calibrator is 3C48 (field = '0')

CASA has build-in models, choose one that matches your calibrator and band (model = '3C48_S.im')

specify which flux density scale standard you want to use (standard = 'Perley-Butler 2017')

Calibration: flux density model

Now check terminal:

```
CASA <136>: setjy(vis='3C75_raw.ms',field='0',model='3C48_S.im',standard='Perley-Butler 2017')
Out[136]:
{'0': {'0': {'fluxd': array([ 9.98645401,  0.        ,  0.        ,  0.        ])},
  '1': {'fluxd': array([ 9.55216122,  0.        ,  0.        ,  0.        ])},
  '2': {'fluxd': array([ 9.15378475,  0.        ,  0.        ,  0.        ])},
  '3': {'fluxd': array([ 8.78703308,  0.        ,  0.        ,  0.        ])},
  '4': {'fluxd': array([ 8.44827557,  0.        ,  0.        ,  0.        ])},
  '5': {'fluxd': array([ 8.13441944,  0.        ,  0.        ,  0.        ])},
  '6': {'fluxd': array([ 7.84281111,  0.        ,  0.        ,  0.        ])},
  '7': {'fluxd': array([ 7.57116461,  0.        ,  0.        ,  0.        ])},
  'fieldName': '0137+331=3C48'},
 'format': "{field Id: {spw Id: {fluxd: [I,Q,U,V] in Jy}, 'fieldName':field name }}"}
```

Or **Casalogger**:

```
…1 INFO    …::setjy()  Using channel dependent flux densities
…1 INFO    …selection  Selected 146016 out of 1036152 rows.
…1 INFO    …::setjy()  0137+331=3C48 (fld ind 0) spw 0  [I=9.9865, Q=0, U=0, V=0] Jy @ 2.488e+09Hz, (Perley-Butler 2017)
…1 INFO    …::setjy()  0137+331=3C48 (fld ind 0) spw 1  [I=9.5522, Q=0, U=0, V=0] Jy @ 2.616e+09Hz, (Perley-Butler 2017)
…1 INFO    …::setjy()  0137+331=3C48 (fld ind 0) spw 2  [I=9.1538, Q=0, U=0, V=0] Jy @ 2.744e+09Hz, (Perley-Butler 2017)
…1 INFO    …::setjy()  0137+331=3C48 (fld ind 0) spw 3  [I=8.787, Q=0, U=0, V=0] Jy @ 2.872e+09Hz, (Perley-Butler 2017)
…1 INFO    …::setjy()  0137+331=3C48 (fld ind 0) spw 4  [I=8.4483, Q=0, U=0, V=0] Jy @ 3e+09Hz, (Perley-Butler 2017)
…1 INFO    …::setjy()  0137+331=3C48 (fld ind 0) spw 5  [I=8.1344, Q=0, U=0, V=0] Jy @ 3.128e+09Hz, (Perley-Butler 2017)
…1 INFO    …::setjy()  0137+331=3C48 (fld ind 0) spw 6  [I=7.8428, Q=0, U=0, V=0] Jy @ 3.256e+09Hz, (Perley-Butler 2017)
…1 INFO    …::setjy()  0137+331=3C48 (fld ind 0) spw 7  [I=7.5712, Q=0, U=0, V=0] Jy @ 3.384e+09Hz, (Perley-Butler 2017)
…5 INFO    …::setjy()  Using model image /home/casa/data/distro/nrao/VLA/CalModels/3C48_S.im
…5 INFO    …::setjy()  Scaling spw(s) [0, 1, 2, 3, 4, 5, 6, 7]'s model image by channel to  I = 9.99032, 8.45105, 7.31959 Jy @(2.4
…5 INFO    …::setjy()  The model image's reference pixel is 1.32027e-07 arcsec from 0137+331=3C48's phase center.
```

Calibration: auxillary correction tables

We can also calculate other corrections that may be required (tropospheric opacity, antenna gaincurves, etc), some of such corrections may depend e.g. on the observing frequency.

These corrections are visibility independent.

Example: Sometimes antenna positions need to be corrected.

The CASA task will calculate these and store results in table 3C_75_raw.ants (note, if corrections not required, no table will be created)

```
gencal(vis='3C_75_raw.ms', caltable='3C_75_raw.ants', caltype='antpos')
```

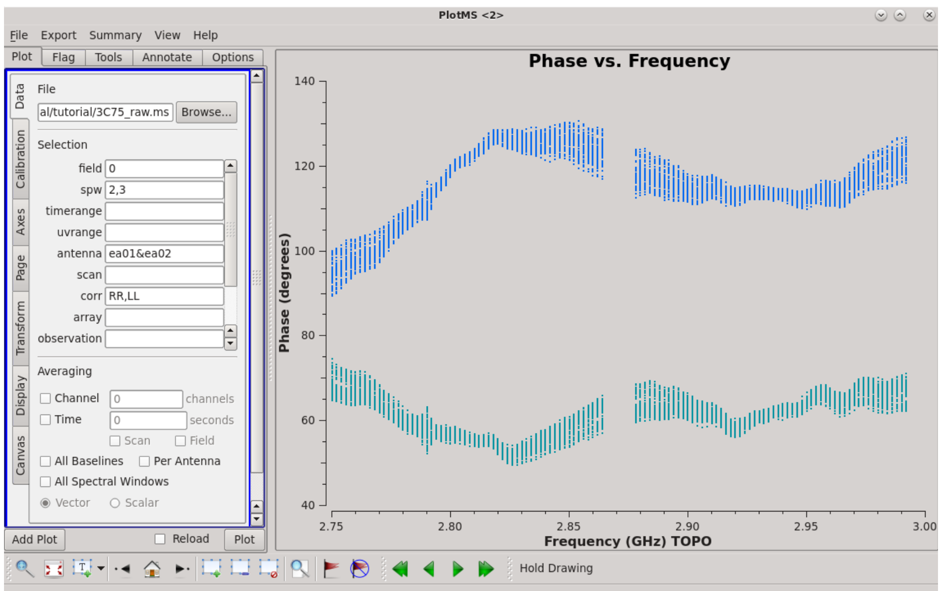
Note: In the .ms files used in this tutorial we already have these corrections applied to the data.

Stokes I: parallel hands (RR,LL) delays

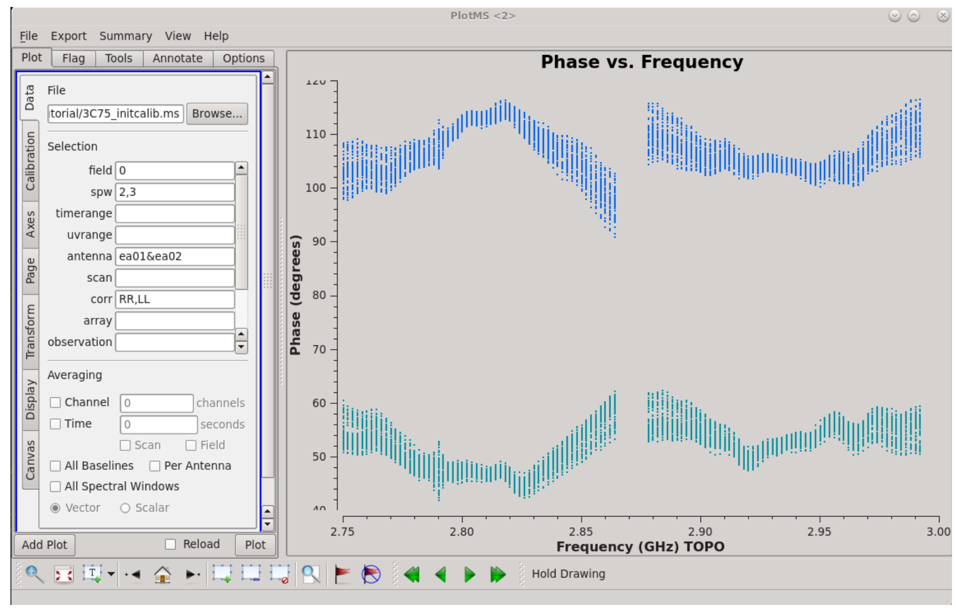
Solve for antenna-based delays (seen as slope in phase vs frequency) relative to reference antenna

```
gaincal(vis='3C75_raw.ms', caltable='3C75_raw.K0', field='0',
refant='ea10', gaintype='K', solint='inf', combine='scan',
gaintable=['….'])
```

before



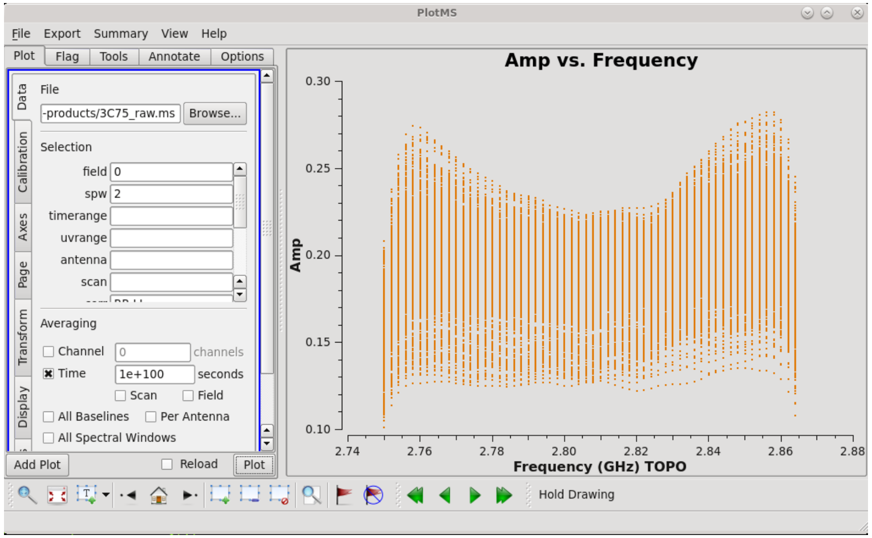
after



Stokes I: bandpass calibration

Solve for gain variations with frequency

Often one will first solve for short timescale phase variations (initial gain calibration) before bandpass calibration, and store results in .G0 table, with `gaincal()`, but this is *not necessary* for S-band frequencies.



Stokes I: bandpass calibration

7

1. Solve for bandpass and store in .B0 table with `bandpass()`

```
> vis='3C75_initcalib.ms'
```
← dataset has already initial calibration applied

```
> caltable='3C75_initcalib.B0'
```
← output table with solutions

```
> field='0'
```
← bandpass calibrator only

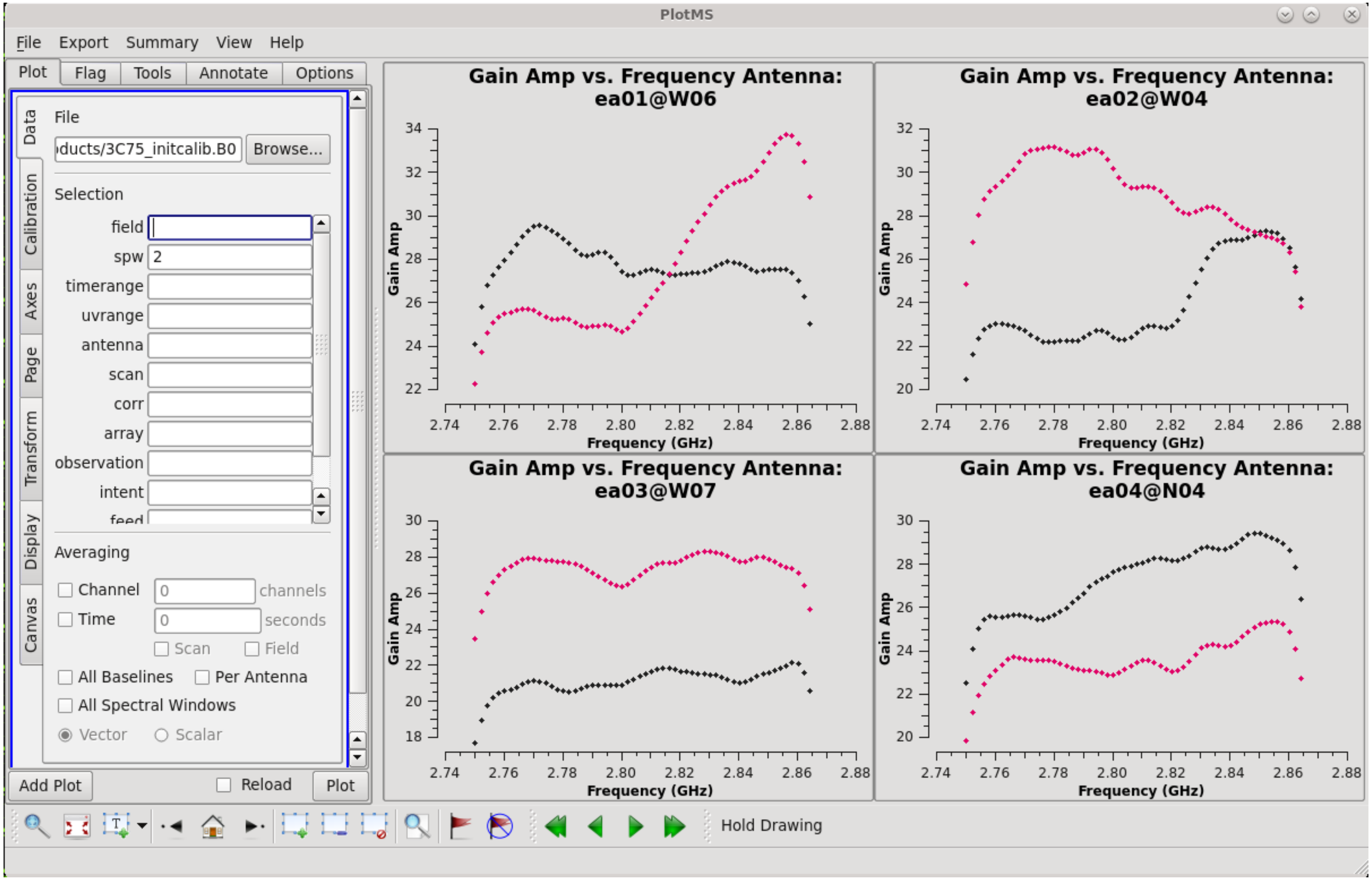
```
> bandtype='B'
```
← specify type of calibration

```
> refant='ea10'
> solint='inf'
> gaintable=['']
```
← This dataset has already auxillary tables and delay calibration applied, so no tables need to be applied here

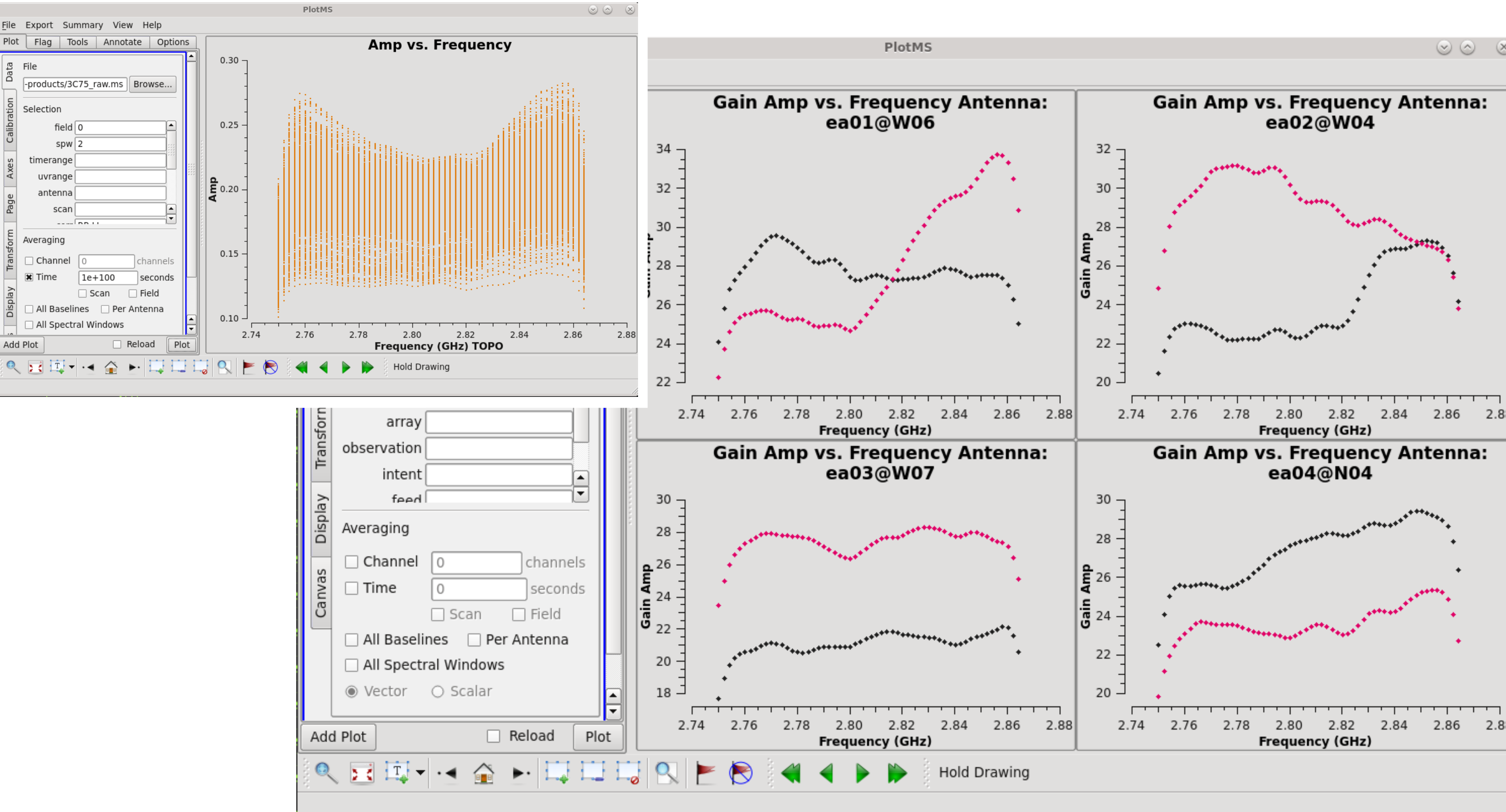
2. Inspect the solutions

```
> plotms(vis='3C75_initcalib.B0', xaxis='frequency',
    yaxis='amp', gridrows=2, gridcols=2, coloraxis='corr',
    iteraxis='antenna', spw='2')
```

Stokes I: bandpass calibration



Stokes I: bandpass calibration



Stokes I: bandpass calibration

8

3. Apply the solutions to bandpass calibrator data with `applycal()`

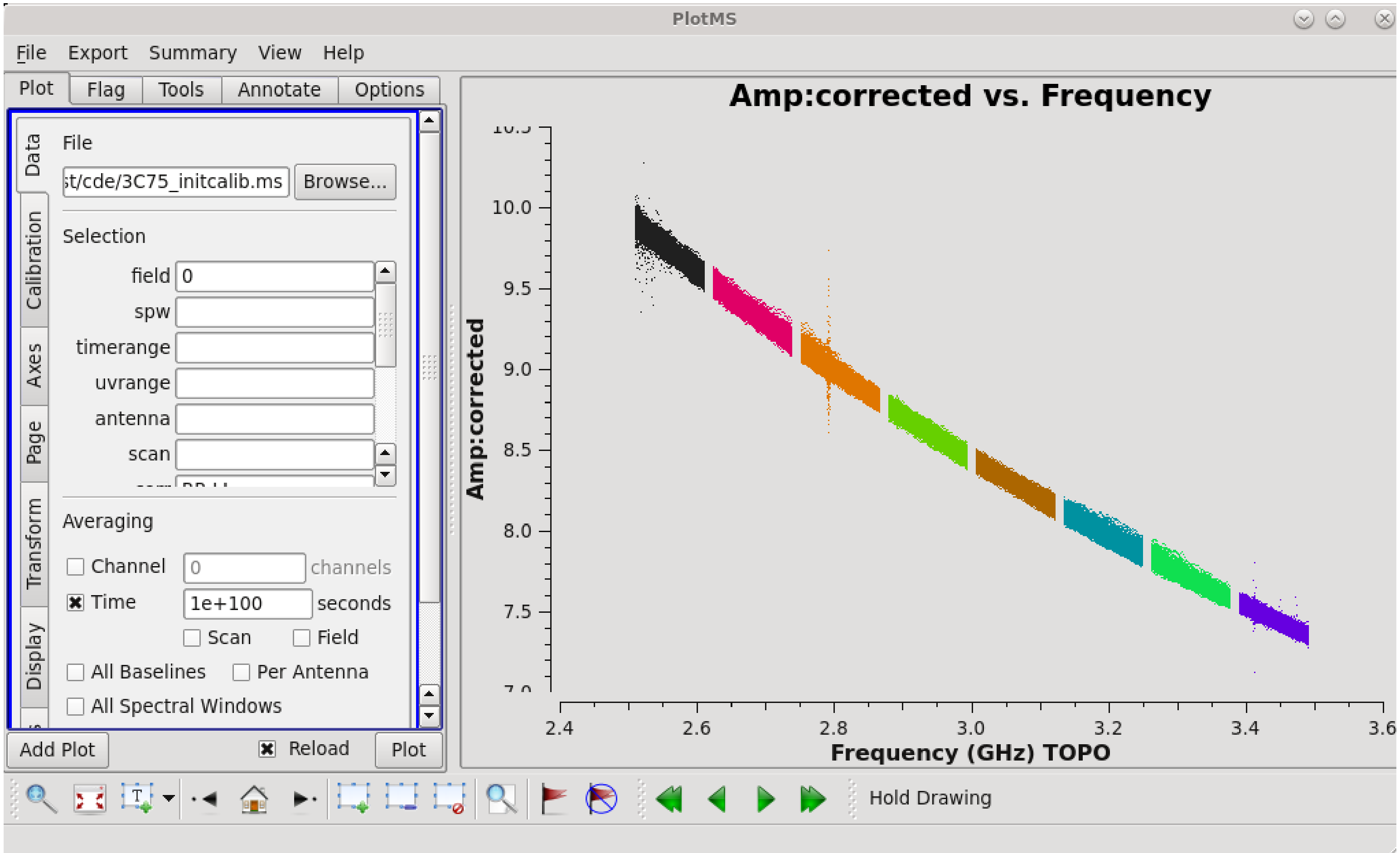
```
> vis='3C75_initcalib.ms'
> field='0137+331=3C48'
> gaintable=['3C75_initcalib.B0']
> calwt=[False]
> parang=False
```

- field: alternative way of specifying field
- gaintable: apply table with solutions that we just created
- calwt: do not calculate data weights at this stage
- parang: do not correct for polarisation angle at this stage

4. Let's see what this calibration step did

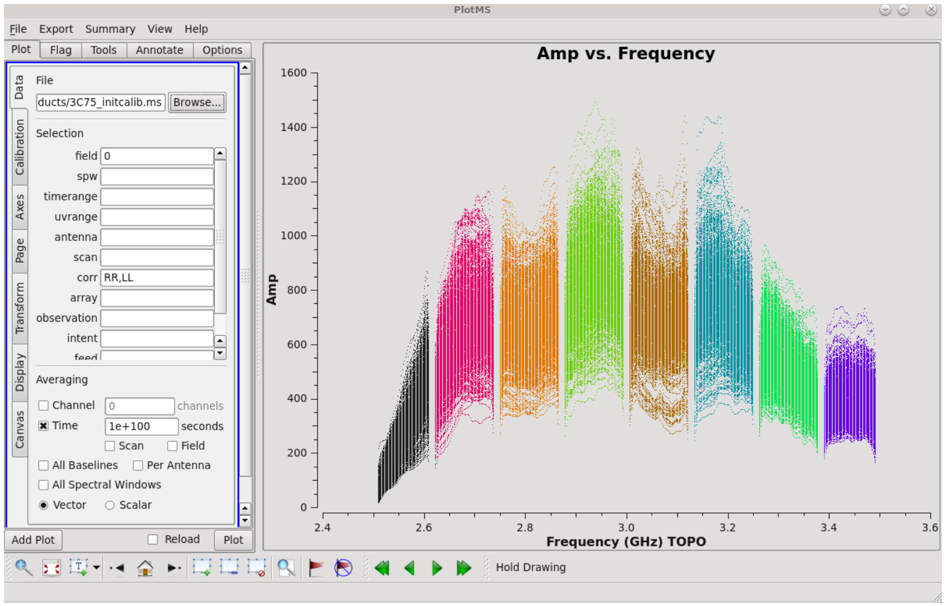
```
> plotms(vis='3C75_initcalib.ms', field='0', xaxis='frequency',
  yaxis='amp', selectdata=True, ydatacolumn='corrected',
  correlation='RR,LL', averagedata=True, avgtime='1e100',
  coloraxis='spw')
```

Stokes I: bandpass calibration

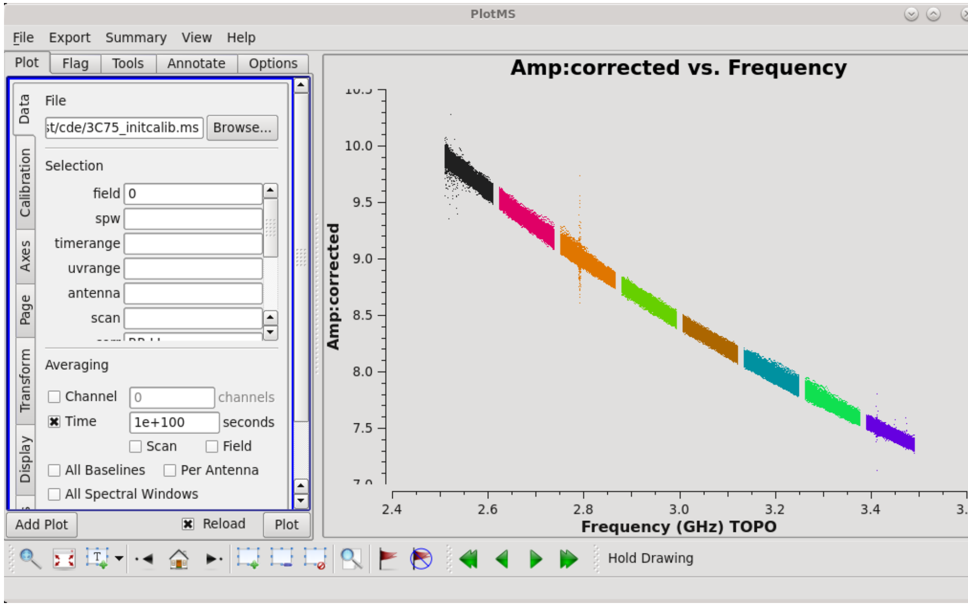


Stokes I: bandpass calibration

before



after



Calibration strategy (Stokes I)

Standard calibration steps (total intensity = Stokes I)
– basic CASA task sequence:

Low/high frequency tutorials:
https://casaguides.nrao.edu

– Set the model of flux density scale calibrator **`setjy()`** DEMO: 6
– Produce auxillary tables not dependent on data **`gencal()`**

– Delay calibration **`gaincal()`**
– Initial complex gain calibration **`gaincal()`**
– Bandpass calibration **`bandpass()`** DEMO: 7
– Complex gain calibration: phases & amp **`gaincal()`**

– Bootstrapping the flux densities of remaining calibrators **`fluxscale()`**

– Apply the calibration to all fields **`applycal()`** DEMO: 8

Complex gain: phases and amplitudes

Now one needs to perform complex gain calibration for variation in both phases and amplitudes in time:

`gaincal()` is used for that, and the results will be stored in .G1 table

For final amplitudes the table .G1 is now copied and the absolute flux density scale is applied via transfer of results from the setjy() initial execution

`fluxscale()` is used for this step, results will be stored in .flx table

To finish off the calibration use **`applycal()`** task.

→ only now the **`applycal()`** task should be used
→ apply the tables created up until now
(.antpos, .gc, .opac, .rq, .swpow, .K0, .B0, .G1, .flx)

Verifying & finalising Stokes I calibration

1. Now examine the calibrated data with **`plotms()`**

2. If there is RFI or any other bad data or antennas use **`flagdata()`** and redo the whole calibration

3. Redo **`applycal()`** and examine quality of the calibration again

4. Keep redoing the calibration until everything looks good.

If you would like to stop here then use **`split()`** task to separate the calibrated target data (`datacolumn='corrected'`) and you are done – your continuum total intensity (Stokes I) data set is calibrated and ready for imaging!

VLA Data Reduction Tutorial

Part II

Calibration strategy (Stokes QUV)

Once the data set is calibrated for Stokes I,
the Polarisation Calibration is done in 4 steps:

1. Set the polarisation model `setjy()`

2. Solve for the *Cross-Hand delays* (i.e. the instrumental delay between two polarisation outputs) `gaincal()`

3. Solve for *Leakage Terms* (D-terms; i.e. instrumental polarisation) `polcal()` **DEMO: 9**

4. Solve for *R-L polarisation position angle* `polcal()`

Only ONE reference antenna must be used for all steps in polarisation calibration

Polarisation model

First a polarisation model has to be applied to the polarisation angle calibrator (here 3C48), this has to be done manually at present.

In CASA you can set variables just like in python:

```
> reffreq = ‘3.0GHz’
> I = 8.45650174
> alpha = [ -0.90366565, -0.14262821]
> polfrac = [0.021429, 0.0391826, 0.00234878, -0.0230125]
> polangle = [1.4215, 1.36672, -2.12678, 3.48384, -2.71914]
```

known fractional polarisation

known polarisation angle (East of North)

setjy() will now calculate values for Q and U Stokes:

```
> vis = ’3C75_calibrated.ms’
> field = ’0137+331=3C48’
> standard = ’manual’
> fluxdensity = [I,0,0,0]
> spix = alpha
> reffreq = reffreq
> polindex = polfrac
> polangle = polangle
> rotmeas = 0
> interpolation = ‘nearest’
```

Polarisation: Cross-Hand (RL,LR) delays

Just like we did before for parallel (RR, LL) delays we now need to solve for the cross-hands due to the residual delay difference between R and L on reference antenna (instrumental delay).

This correction is done again with **`gaincal()`** and KCROSS type of calibration (`gaintype='KCROSS'`).

```
gaincal(vis='3C75_calibrated.ms', caltable=3C75_calibrated.Kcross,
field='0137+331=3C48', spw='0~7:5~58', refant='ea10',
gaintype="KCROSS", solint="inf", combine="scan,spw", calmode="ap",
append=False, gaintable=[''], gainfield=[''], interp=[''],
spwmap=[[]], parang=True)
```

Here, it is recommended to fit the cross-hand delay across multiple spectral windows (multiband delay, `combine="scan,spw"`) to take advantage of the wideband data we have.

Polarisation: Leakage terms

Next, we need to know how much the instrument itself (bits and bobs of VLA) contributes to the recorded polarised signal.
There are two methods:

1. observe once with an unpolarised calibrator (here J2355+4950), or
2. observe a calibrator with known polarisation (here J0259+0747, 4.7% frac.pol.), requires multiple observations across a range of parallactic angle

In this tutorial, we will be using method 1.

Polarisation: Leakage terms

9

We need to know how much the instrument itself (bits and bobs of VLA) contributes to the recorded polarised signal.

Instrumental calibration is done with `polcal()`

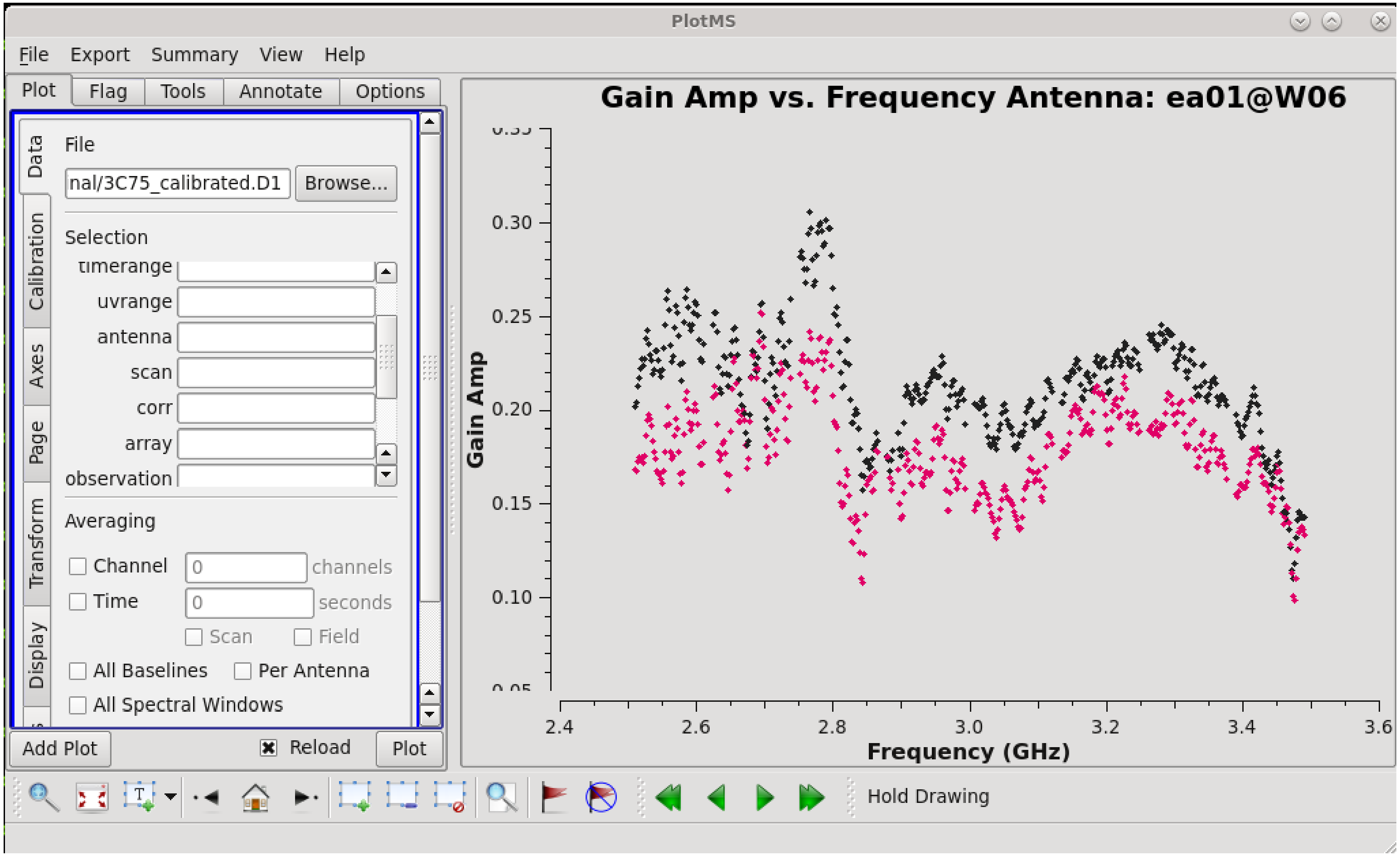
```
> vis='3C75_calibrated.ms'
> caltable='3C75_calibrated.D1'
> field='J2355+4950'
> spw='0~7'
> refant='ea10'
> poltype='Df'
> solint='inf,2MHz'
> combine='scan'
> gaintable=['']
```

- field: unpolarised calibrator
- poltype: specify type of calibration: frequency-dependent (f) leakage terms (D)
- solint: solution interval (one solution per 2MHz)
- gaintable: in current dataset we already applied KCROSS delays

Now, inspect the solutions

```
> plotms (vis='3C75_calibrated.D1', xaxis='freq', yaxis='amp',
    iteraxis='antenna', coloraxis='corr')
```

Polarisation: Leakage terms



Polarisation: R-L polarisation angle

Now we have total polarization of our data calibrated (so correct), but the R-L phase is still uncalibrated. In this step we need to solve for the polarisation angle using a calibrator with known polarisation angle.

Here again **`polcal()`** task is used with `poltype='Xf'` – a parameter that specifies we want to solve for frequency dependent (*f*) position angle (*X*).

Now, to finish off we need to apply the calibration tables to the data (especially the target field!). Again this is done with **`applycal()`** task.

And that's it! We have now a calibrated dataset!

Imaging

It's time to make an image!

In CASA we will use task **`tclean()`**

We will image 3C75 radio source in full Stokes and with multi-scale clean, but only for 2 scans in 1 spw for demonstration purposes.

More examples, including advanced techniques and self-calibration, see online CASA tutorials:

https://casaguides.nrao.edu/index.php/VLA_CASA_Imaging-CASA5.4.0

And new Polarization CASA guide is coming!

General imaging strategy

Important to consider what you are going to image:

– 3C75 is complex radio source with both diffuse emission and compact components

→ need *multi-scale* – algorithm that will simultaneously reconstructs emission at different angular scales

→ general rule: use scales equal 0, 2, 5, … x beam, up to half of minor axis of the largest scale (0 x beam is a point source)

– *weighting* scheme – i.e. how your image is weighted

→ *natural* weighting gives better sensitivity

→ *uniform* weighting gives better resolution

→ '*briggs*' weighting with its *robust* parameter allows to control between natural and uniform methods

General imaging strategy

Important to consider what you are going to image:

- VLA configuration was *D* in *S*-band, and resolution is ~17 arcsec
  - → *cell size* of image should be 3–5 pixels across the FWHM of the synthesised beam, so we should proceed with cell size of 3.4 arcsec

- 3C75 extends out to 9 arcmin, make image large enough to cover this
  - → let's image 3x as much (~27 arcmin) to account for any bright close-by sources
  - → *image size* is expressed in pixels, so here 480 on each side (a combination of 2, 3 and 5 numbers that `tclean()` prefers)

Imaging (Full Stokes, multi-scale)

10a

Let's have a look at our `tclean()` execution.

```
> vis = '3C75_target_spw4.ms'
> imagename = '3C75_spw4'

> imsize = 480
> cell = "3.4arcsec"

> stokes = "IQUV"

> specmode = "mfs"
> gridder = "standard"
> pblimit = -0.0001

> deconvolver = "mtmfs"
> scales = [0, 6, 18]

> nterms = 2
```

- stokes: deconvolving over all Stokes
- specmode: continuum imaging with only one output image channel
- deconvolver / scales: minor cycle algorithm (mtmfs=multi-term multi-freq synthesis)
- nterms: number of Taylor coefficients in the spectral model

Imaging (Full Stokes, multi-scale)

10b

Further `tclean()` parameters

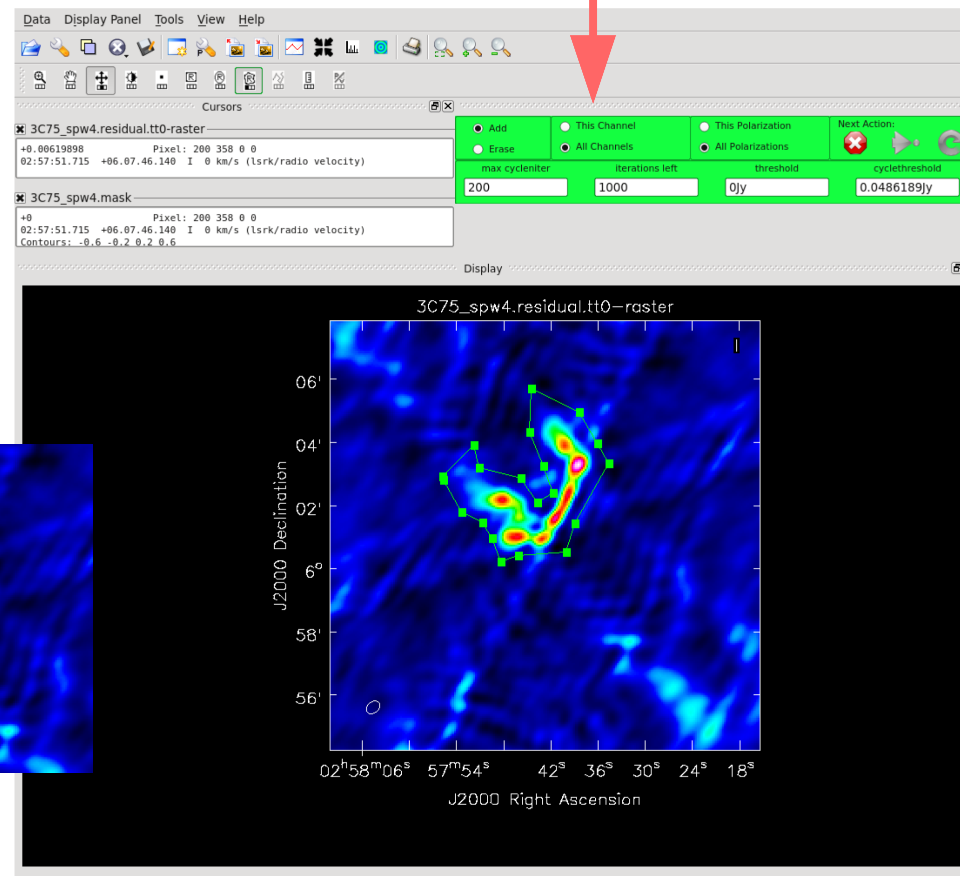
`> pbcor = False`	we will perform primary beam correction separately
`> weighting = "briggs"` `> robust = 0.5`	intermediate weighting between natural and uniform
`> niter = 1000` `> cycleniter = 200`	how many iterations to do in total (niter), and how many iterations per each cycle (cycleniter)
`> interactive = True`	we will do this imaging interactively
`> savemodel = 'modelcolumn'`	worth saving if in further steps you would like to do self-calibration

Imaging – interactive tclean()

11a

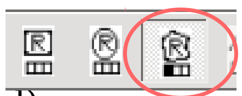
Once the window (Viewer Display) pops us:

→ Click *“All channels”* and *“All polarisations”*



→ Zoom in with

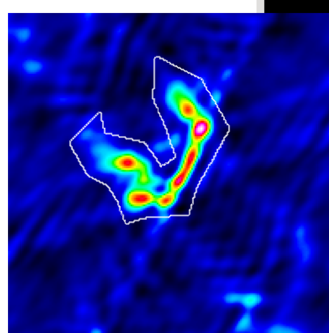
→ Create a region by choosing any of these



(here polygon used)

→ Double-click inside the region to activate

once activated, it will look like this:



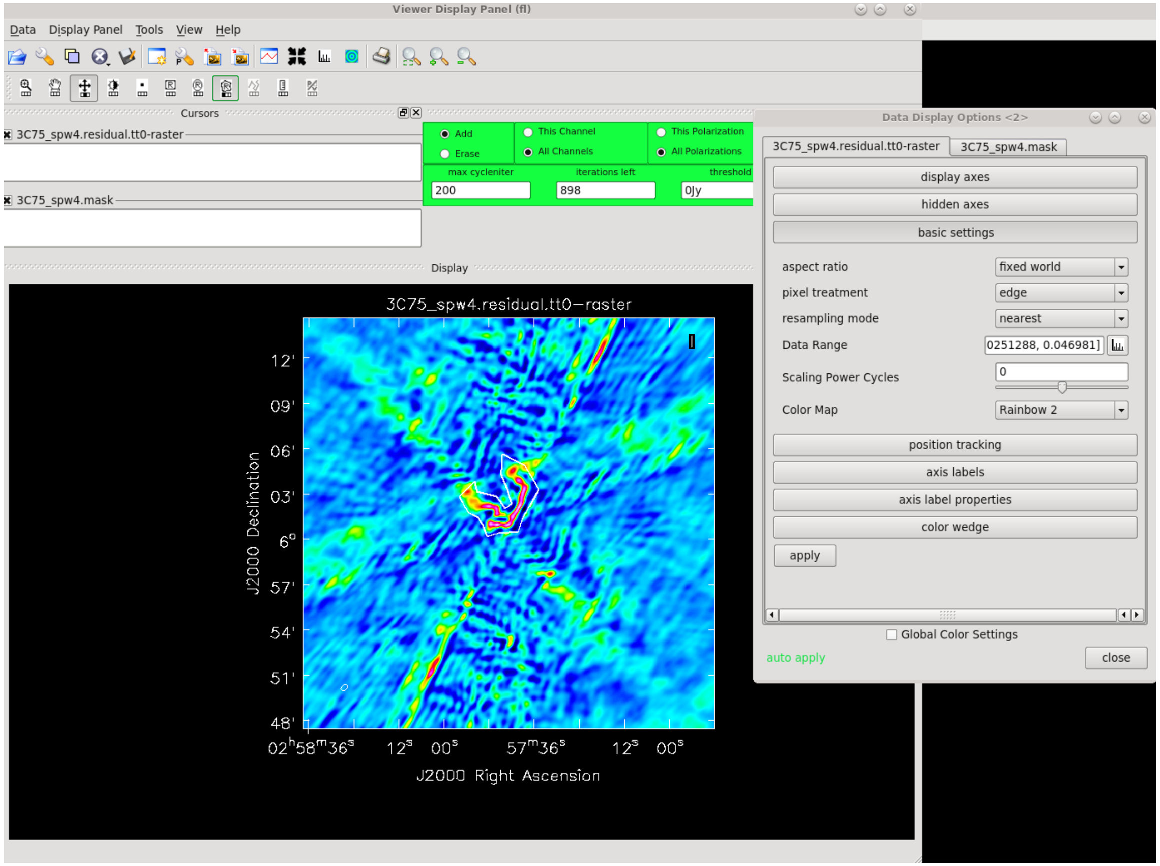
→ Start the clean

Imaging – interactive tclean()

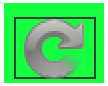
11b

After first cycle you should get something like this:

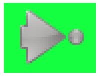
→ To adjust colour scale click the wrench icon and a new window will appear where you can play with colour scale



→ Adjust regions if needed and continue



→ If happy with the regions, let it clean until it's done



→ Or if you want to stop the cleaning earlier click

Imaging – inspect the result

12

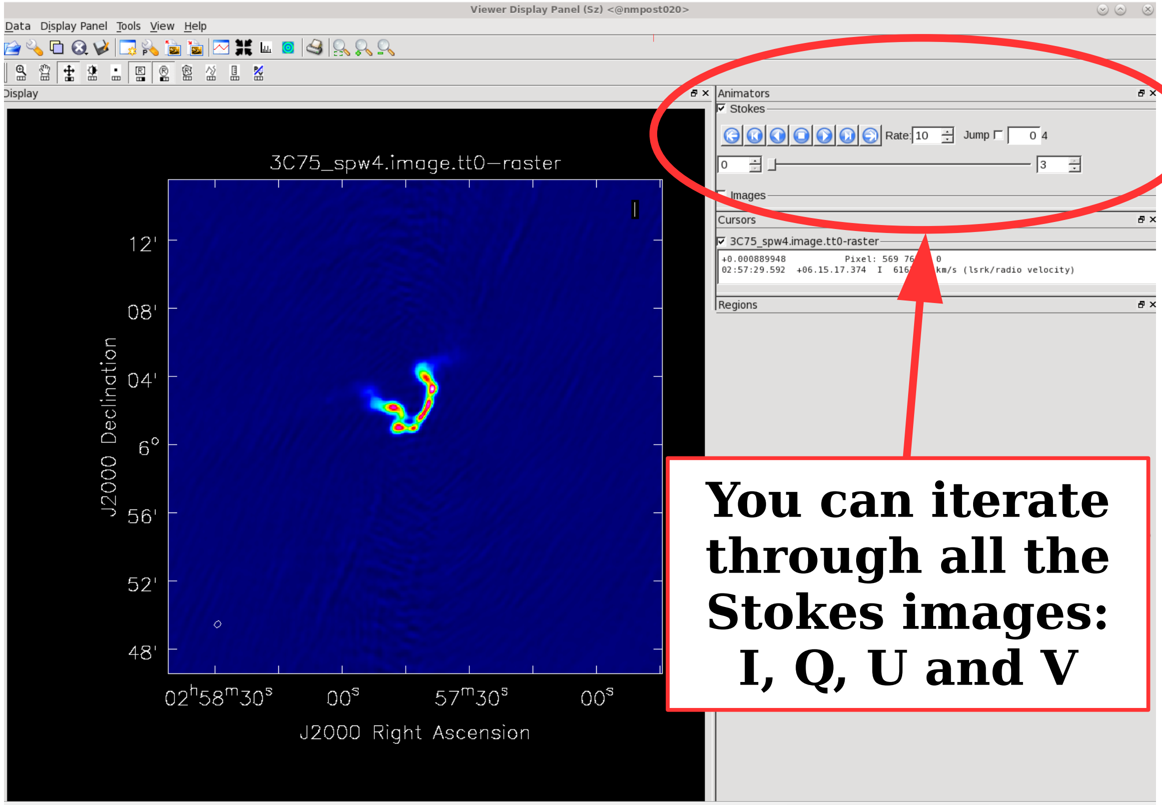
To view CASA images or fits files use task **viewer()**

– if image selection window doesn't open automatically click

– select your image

‘3C75_spw4.image.tt0’

and click



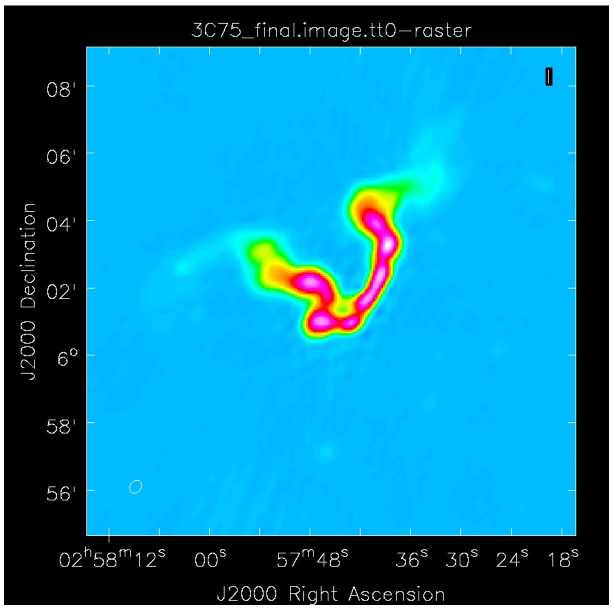
You can iterate through all the Stokes images: I, Q, U and V

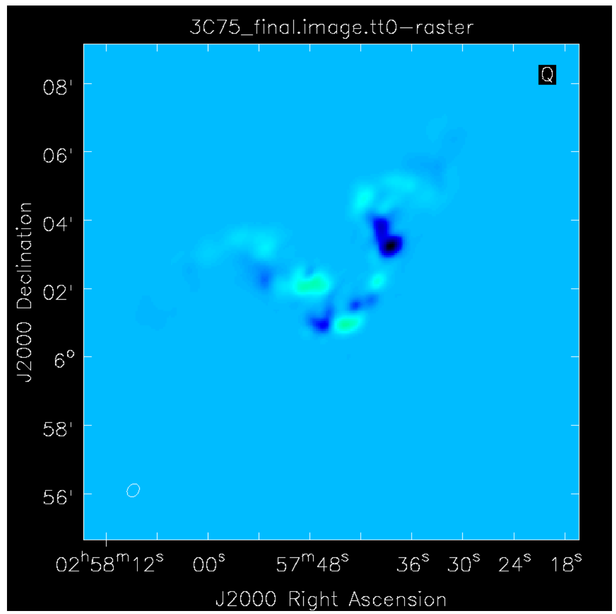
Viewer can also be used outside of CASA as a stand alone program; just type casaviewer at the terminal command line to access it.

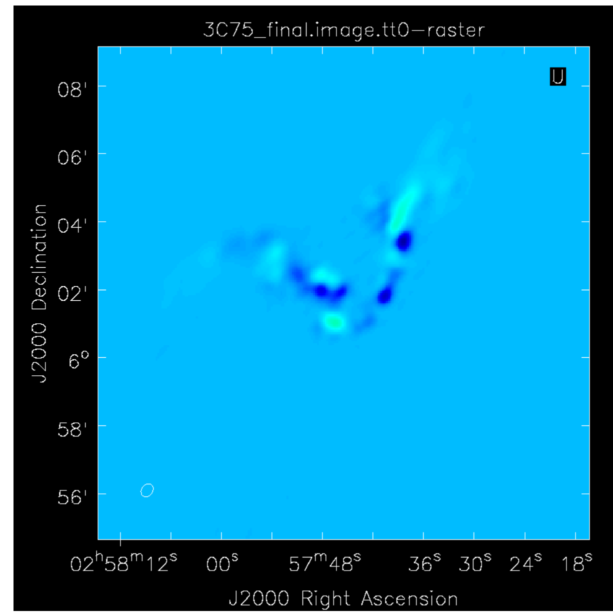
Imaging – inspect the result

In your folder you will also find another image: **`3C75_final.image.tt0`**

This image was made from full observation (8 spw and all the scans). You may want to inspect it in the viewer too.







Stokes: **I** **Q** **U**

Image analysis

The three most fundamental image analyses are:

(1) determine the peak brightness,

(2) the flux density, and

(3) the image noise level.

Option A: Use **`imstat()`** task

```
imstat('3C75_final.image.tt0')
```

Casalogger:

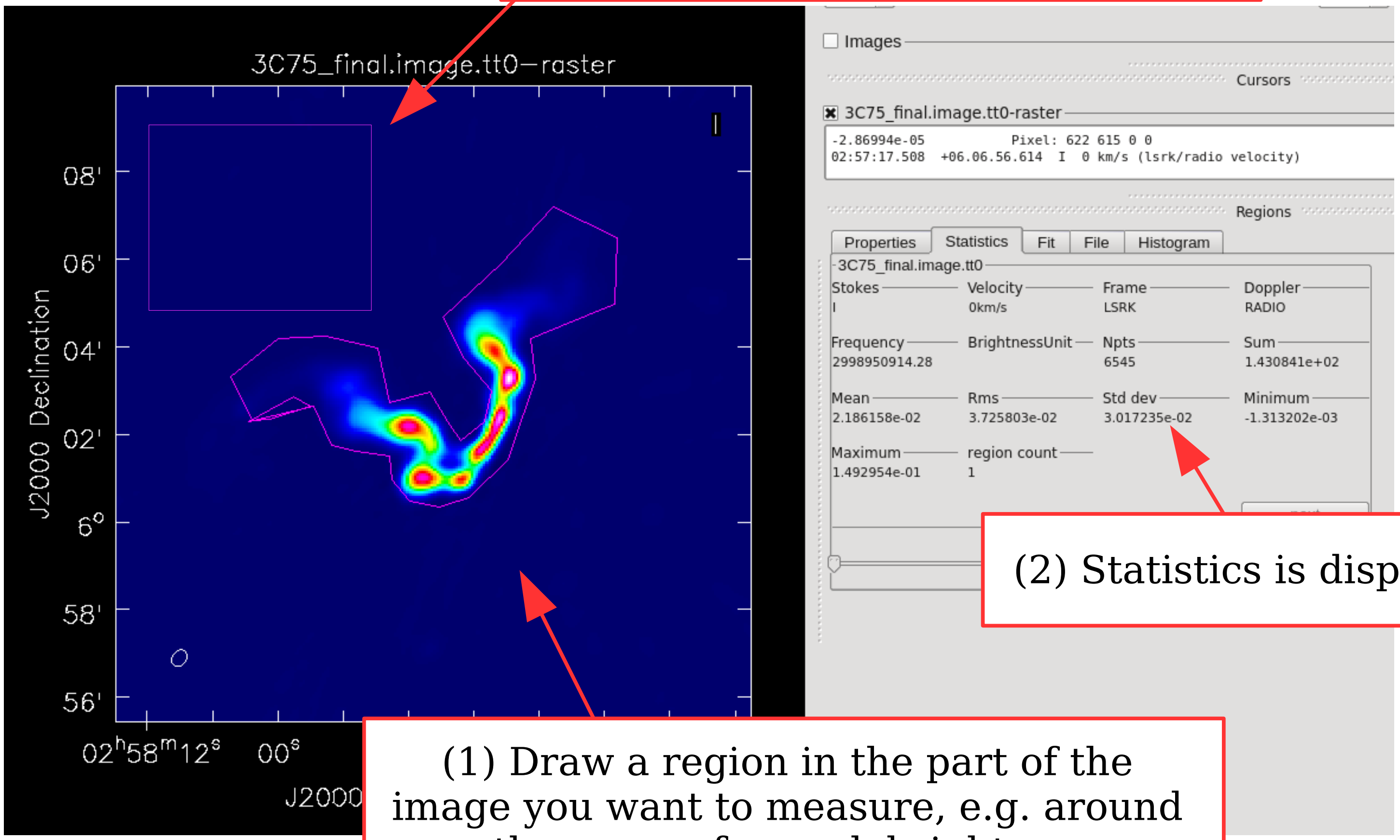
```
…ogStartup Regions ---
…ogStartup          -- bottom-left corner (pixel) [blc]:  [0, 0, 0, 0]
…ogStartup          -- top-right corner (pixel) [trc]:    [1023, 1023, 3, 0]
…ogStartup          -- bottom-left corner (world) [blcf]: 02:59:39.228, +05.32.03.206, I, 2998950914.28Hz
…ogStartup          -- top-right corner (world) [trcf]:   02:55:46.052, +06.30.01.448, V, 2998950914.28Hz
imstat:::: Values ---
imstat::::          -- number of points [npts]:                4.1943e+06
imstat::::          -- maximum value [max]:                    0.149295
imstat::::          -- minimum value [min]:                    -0.0178602
imstat::::          -- position of max value (pixel) [maxpos]: [529, 551, 0, 0]
imstat::::          -- position of min value (pixel) [minpos]: [530, 550, 1, 0]
imstat::::          -- position of max value (world) [maxposf]: 02:57:38.755, +06.03.17.399, I, 2998950914.28H
imstat::::          -- position of min value (world) [minposf]: 02:57:38.527, +06.03.13.999, Q, 2998950914.28H
imstat::::          -- Sum of pixel values [sum]:               150.212
imstat::::          -- Sum of squared pixel values [sumsq]:     9.18366
imstat:::: Statistics ---
imstat::::         -- Mean of the pixel values [mean]:         3.58133e-05
imstat::::         -- Variance of the pixel values :           2.18827e-06
imstat::::         -- Standard deviation of the Mean [sigma]:  0.00147928
imstat::::         -- Root mean square [rms]:                  0.00147971
imstat::::         -- Median of the pixel values [median]:     -6.44428e-08
imstat::::         -- Median of the deviations [medabsdevmed]: 9.80346e-06
imstat::::         -- IQR [quartile]:                          1.96096e-05
imstat::::         -- First quartile [q1]:                     -9.80892e-06
imstat::::         -- Third quartile [q3]:                     9.80065e-06
…tatistics Sum column unit =
…atistics+ Mean column unit =
…atistics+ Std_dev column unit =
…atistics+ Minimum column unit =
…atistics+ Maximum column unit =
…tatistics Npts          Sum           Mean          Rms           Std_dev       Minimum       Maximum
…tatistics  4.194304e+06  1.502117e+02  3.581327e-05  1.479715e-03  1.479281e-03 -1.786024e-02  1.492954e-01
imstat:::: ##### End Task: imstat            #####
…stat::::+ ########################################
```

Image analysis

13

Option B: Use **`viewer()`**

Draw region outside of source if you want to measure image noise (rms)



(2) Statistics is displayed here

(1) Draw a region in the part of the image you want to measure, e.g. around the source for peak brightness
(note: this should be done only on primary beam corrected image)

Polarisation images

To get the actually useful polarisation images (polarised intensity and polarisation angle) we need to do some math on the images.

First, one needs to separate Stokes Q and U from the image cube:

```
imsubimage(imagename='3C75_final.image.tt0',
outfile='3C75_final.Q.image', stokes='Q')

imsubimage(imagename='3C75_final.image.tt0',
outfile='3C75_final.U.image', stokes='U')
```

Then the polarisation images are from the following equations:

Linear polarisation intensity image (POLI)

$$P = (Q^2 + U^2)^{0.5}$$

```
immath(imagename=['3C75_final.Q.image', '3C75_final.U.image'],
mode='poli',
outfile='3C75_final.POLI.image')
```

Polarisation position angle image (POLA)

$$X = \frac{\mathrm{atan}(U/Q)}{2}$$

```
immath(imagename=['3C75_final.Q.image', '3C75_final.U.image'],
mode='pola',
outfile='3C75_final.POLA.image')
```

Final images

At this stage we can create a final image including all information: total intensity (Stokes I), linear polarised intensity (POLI) and polarisation position angle (POLA).

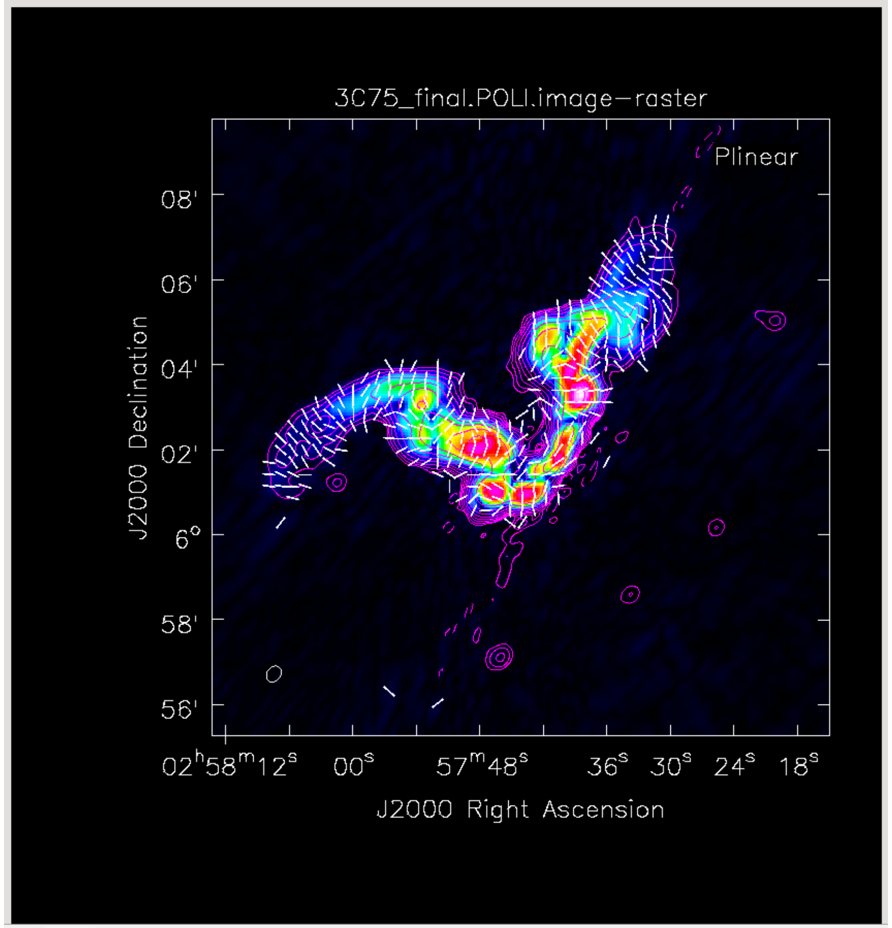


Image: POLI

Vectors: POLA–masked

[pixels masked if below 0.1 mJy/bm=6σ in POLI]

Contours: Stokes I

[contour levels 0.4 mJy/bm=10σ x 2^i]

To adjust appearance and contour levels use Data Display Options accessed with

Primary beam correction

For the images to be *science ready* they need to be corrected for the telescope's primary beam.

Primary beam (PB) is the antenna response. For VLA antennas it is roughly a 2D Gaussian with peak response at the phase centre, and smaller fractional response as we move away to the image edges.

Tasks to create PB response image:

- **`tclean()`** task will do it if parameter `pbcor=True` – use it only if `nterms=1`
- **`widebandpbcor()`** task will create PB response for wideband images (those that were created with `tclean()` parameter `nterms>1`)

To correct for the PB the final cleaned image is divided by the PB response

In this tutorial we need to use **`widebandpbcor()`** task as we used `nterms=2`. The task outputs PB corrected image

Extra: wideband depolarisation

So far in all our imaging we averaged data across 1 GHz bandwidth.

Sometimes, this is not the best choice, e.g. our 3C 75 radio galaxy undergoes quite a bit of depolarisation across such a wide band.

Alternative: create an image cube

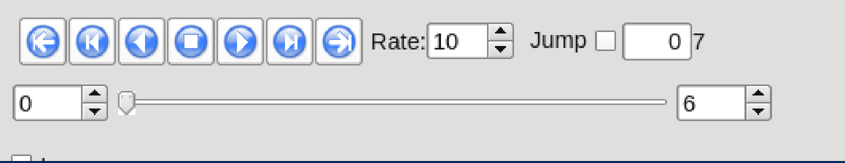
In your folder you will find another image: **`3C75_IQUVcube.image`**

If you load it into viewer() you may notice:

→ there are 8 spws (marked as km/s by default you can change it to GHz)

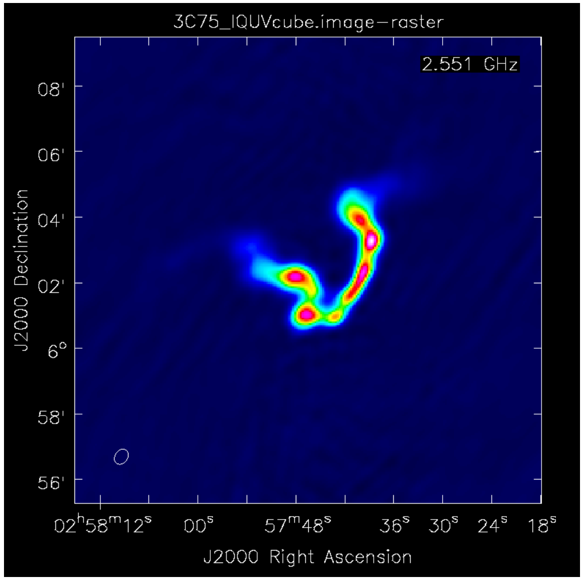
→ Stokes I loads first as default, Stokes Q, U and V are hidden axes (you can change that again and switch between the Stokes)

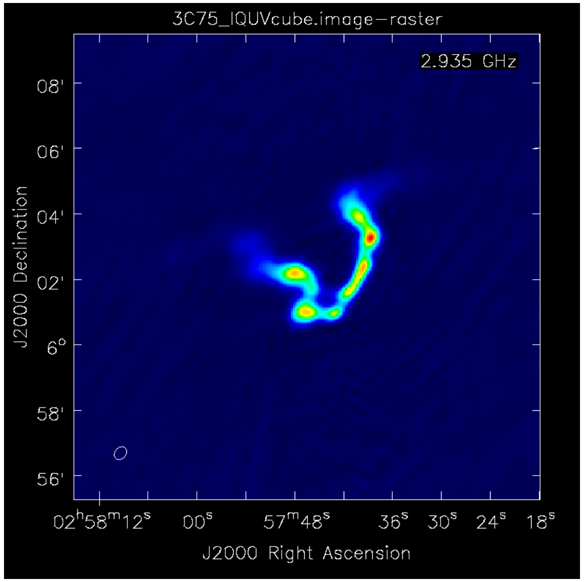
→ to see the difference between spws play a movie

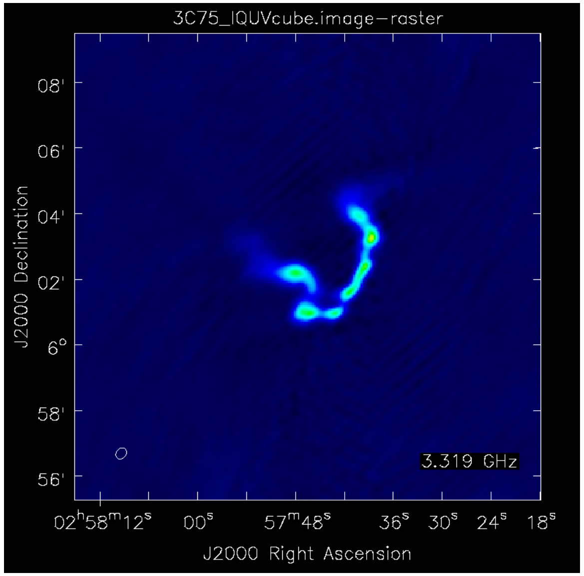


Extra: wideband depolarisation

Stokes I at 3 frequencies:







You can also check out Stokes Q, U and V

Or create POLI and POLA images for each spw and examine changes across the band width.

www.nrao.edu
science.nrao.edu
public.nrao.edu

The National Radio Astronomy Observatory is a facility of the National Science Foundation operated under cooperative agreement by Associated Universities, Inc.