VLA Data Reduction Techniques II
Calibration

Emmanuel Momjian
NRAO

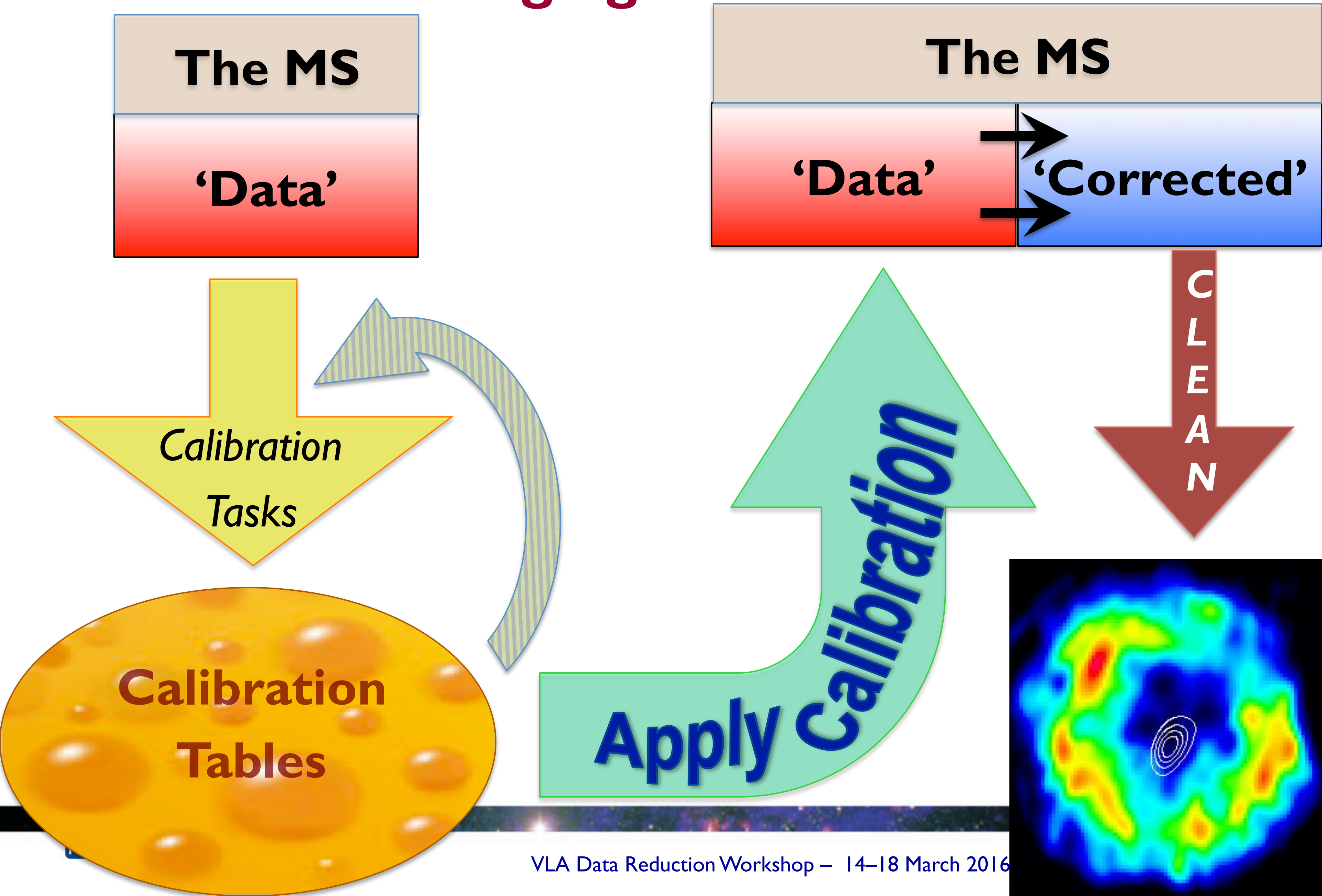
Atacama Large Millimeter/submillimeter Array
Karl G. Jansky Very Large Array
Robert C. Byrd Green Bank Telescope
Very Long Baseline Array

The MS structure

'Data' column	'Corrected' Column	'Model' Column (optional)
Raw Data	Calibrated Data	FT of source model

- When you load your data from the archive, your MS will only have the 'Data' column.
- The other two columns can be created by various means.
- The creation of the other two columns → MS tripling in size.
- The 'Model' Column is optional.
  - If not created → MS doubling in size.
  - Models can be "attached" to the MS, FT-ed and used when needed (replacing the need for the 'Model' column).

Calibration & Imaging Flow



Calibration

- Correcting antenna positions
- Gain Curves
- Opacity (HF) and Ionospheric (LF) corrections
- Requantizer gain calibration
- Setting the flux density scale
- Delay calibration
- Initial Phase only calibration (HF)
- Bandpass calibration
- Complex gain calibration
- Polarization Calibration
- Setting the flux scales of the secondary calibrators

} Prior Calibration

gencal

- *gencal* is a task for various types of corrections:

```
'amp'= amplitude correction
'ph' = phase correction
'sbd'= single-band delay
'mbd'= multi-band delay
'antpos' = ITRF antenna position corrections
'antposvla' = VLA-centric antenna pos. corrections
'swpow' = EVLA switched-power gains
'rq' = EVLA requantizer gains
'swp/rq' = EVLA switched power gains/req. gains
'opac' = Tropospheric opacity
'gc' = Gain curve (zenith-angle-dependent gain)
'eff' = Antenna efficiency (sqrt(K/Jy))
'gceff' = Gain curve and efficiency
'tecim' = Total electron content for ionospheric
          corrections
```

Antenna Positions: *gencal*

- Check the operator's log to see if any antennas were recently moved.
- Use the task *gencal* to produce a calibration table that will include the antenna position corrections

```
caltype  = 'antpos'
caltable = 'antpos.cal'
```

- Baseline correction related information is at:

  http://www.vla.nrao.edu/astro/archive/baselines/

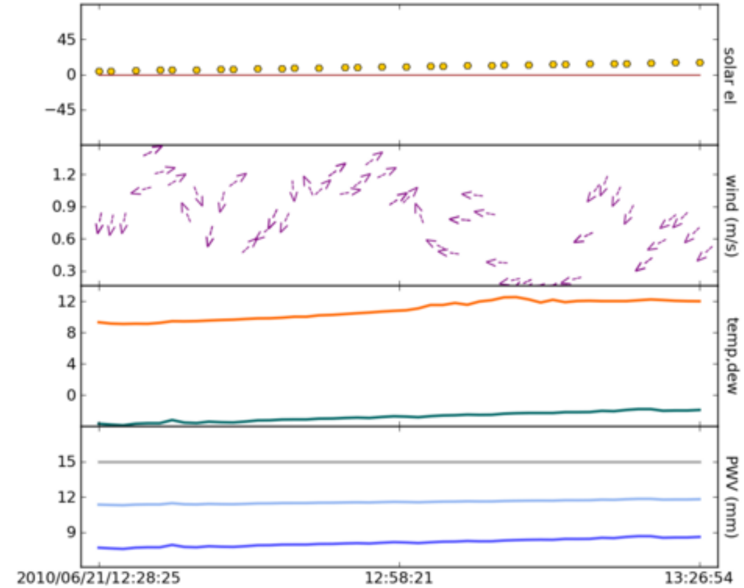
Gain Curves: *gencal*

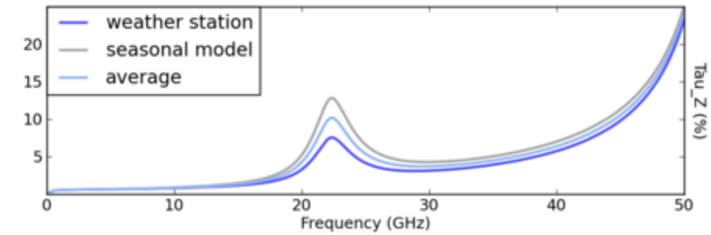
- Large antennas have a forward gain that changes with elevation.
- Gain curves describe how each antenna behaves as a function of elevation, for each receiver band.
- The polynomial coefficients for the VLA are available directly from the CASA data repository.
- Especially important for higher frequencies.
- In *gencal*, set

```
caltype  = 'gc'
caltable = 'gaincurve.cal'
```

Opacity Corrections (HF): *plotweather*

- Atmospheric optical depth corrections.
- Important for high frequency observations (>15 GHz).
- *Plotweather* is the task to estimate opacities and to make weather plots.
  - ➢ Uses weather statistics and/or seasonal models.
  - ➢ `tau_val=plotweather(vis='my.ms',doPlot=T)`
  - ➢ Gives one value per spw





Opacity Corrections (HF): *gencal*

- After *plotweather* use *gencal* to make a calibration table using the derived opacities:

```
caltype   = 'opac'
caltable  = 'opacity.cal'
parameter = tau_val
spw       = 'match it to tau_val’s'
```

Ionosphere Correction: *gencal*

Needed for lower frequency observations (P, L, and S-bands)

- Important for polarimetry
- Under testing

1) Import the TEC image

```
from recipes import tec_maps
tec_image, tec_rms_image =
  tec_maps.create(vis='my.ms',doplot=True)
```

2) Run gencal

```
caltype    = 'tecim'
caltable   = 'tecim.cal’
infile     =  tec_image
```

Requantizer gains: *gencal*

- Optimizes the digital power within each spectral window.
- Required for 3-bit data.
- Strongly recommended for P-band 8-bit data.
- In *gencal*, set

```
caltype  = 'rq'
caltable = 'requant_gains.cal'
```

Setting the flux density scale: *setjy*

- Calculates the absolute flux density as a function of frequency (and time):
  - for standard flux density calibrators (e.g., Perley-Butler 2013)
  - and for Solar System objects (e.g., Butler-JPL-Horizons 2012)
- If provided, attaches a model record to the MS

```
vis                 =    'my.ms'
field               =      '?'
spw                 =      ''
scalebychan         =      True
standard            =     'Perley-Butler 2013'
     model          =      '?'
     listmodels     =      False
usescratch          =      False
```

Setting the flux density scale: *setjy*

- Calculates the absolute flux density as a function of frequency (and time):
  - for standard flux density calibrators (e.g., Perley-Butler 2013)
  - and for Solar System objects (e.g., Butler-JPL-Horizons 2012)
- If provided, attaches a model record to the MS

```
vis              =   'my.ms'
field            =     '?'
spw              =     ''
scalebychan      =     True
standard         =    'Perley-Butler 2013'
      model      =     '?'
      listmodels =     False
usescratch       =     False
```

Calibration: *setjy*

- `listmodels`
  - If `True`, the task will only list the available primary calibrator models (3C138, 3C147, 3C286, 3C48; at L, S, C, X, U, K, A, Q bands).
  - If `False`, the task will calculate the flux density.

- `usescratch`
  - If `True`, the 'Model' column will be created. This will increase the size of the MS.
  - If `False`, the model is simply attached to the MS. When needed, it will be FT-ed and used.

Setting the flux density scale: *setjy*

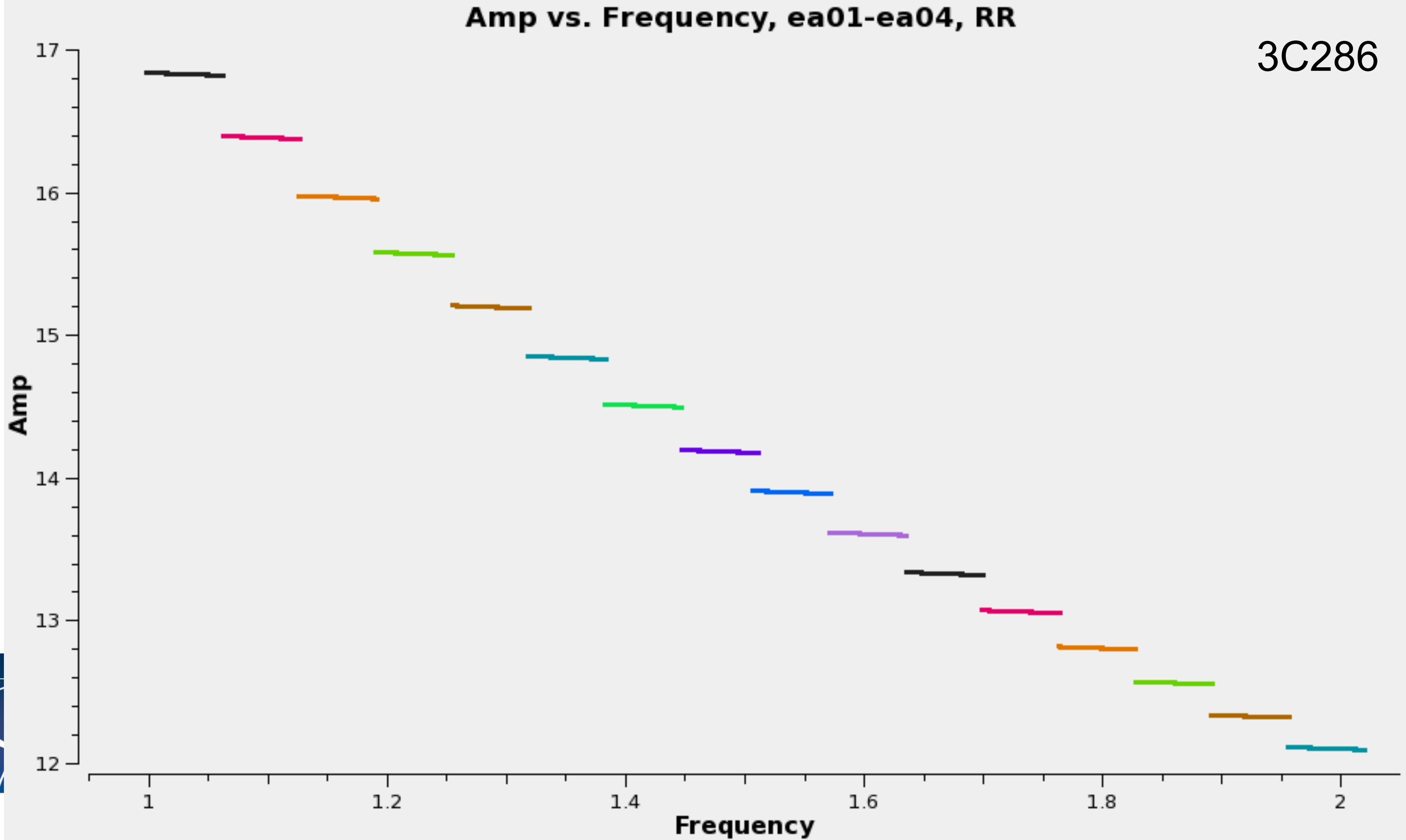
- Calculates the absolute flux density as a function of frequency (and time):
  - for standard flux density calibrators (e.g., Perley-Butler 2013)
  - and for Solar System objects (e.g., Butler-JPL-Horizons 2012)
- If provided, attaches a model record to the MS

```
vis              =   'my.ms'
field            =     '?'
spw              =     ''
scalebychan      =     True
standard         =    'Perley-Butler 2013'
      model      =     '?'
      listmodels =     False
usescratch       =     False
```

Calibration: *setjy*

The `scalebychan` **parameter**

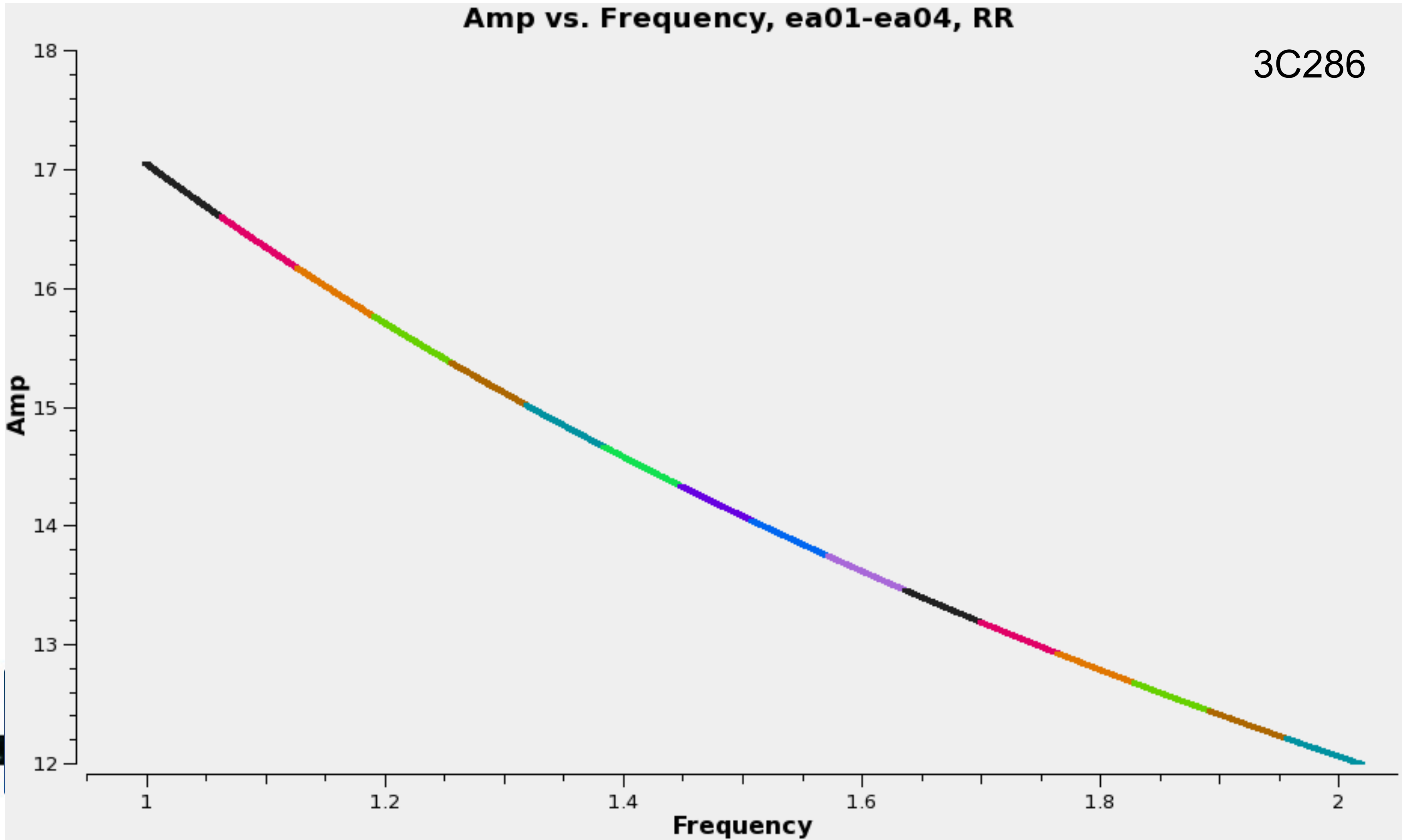
- When **`False`**: The values will be per spectral window.
- **Do not set it to False!**



Calibration: *setjy*

The `scalebychan` parameter

- When **`True`**: The values will be per spectral channel



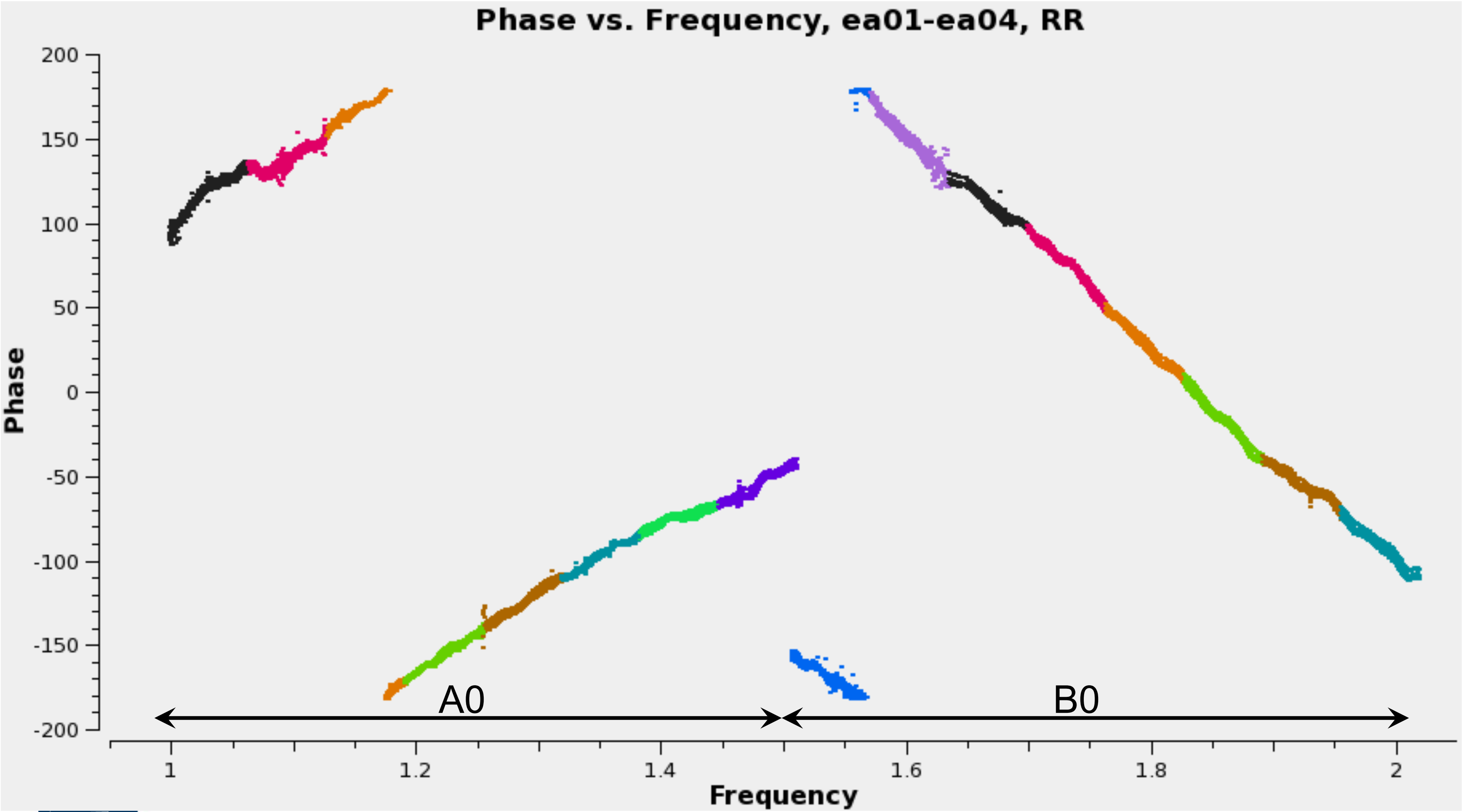
Setting the flux density scale: *setjy*

- User can also provide flux density values instead of letting the task calculate them (manual mode)

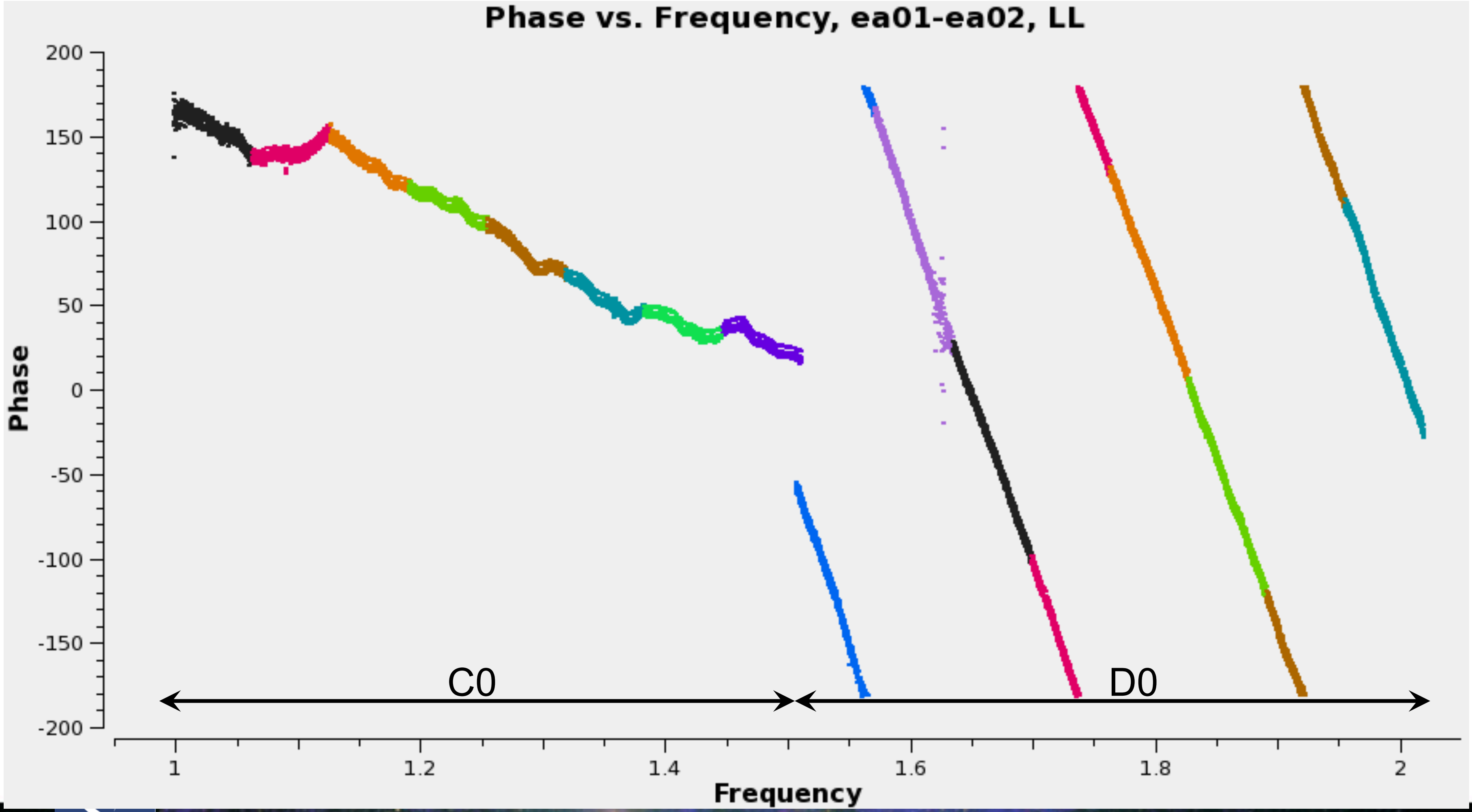
```
standard          =   'manual'
    fluxdensity   = [1, 0, 0, 0]
    spix          = [alpha,beta]
    reffreq       = '1GHz'
```

```
Can also provide
  polindex: coefficients for linear pol. fraction expression
  polangle: coefficients for pol. angle expression
  rotmeas : rotation measure
```

Delays


Phase vs. Frequency, ea01-ea04, RR
Phase
200
150
100
50
0
-50
-100
-150
-200
Frequency
1
1.2
1.4
1.6
1.8
2
A0
B0

Delays


Phase vs. Frequency, ea01-ea02, LL
Phase
Frequency
200
150
100
50
0
-50
-100
-150
-200
1
1.2
1.4
1.6
1.8
2
C0
D0

Delay Calibration: *gaincal*

```
vis         =     'my.ms'
caltable    =     'delays.cal'
field       =     '?'
solint      =     '60s'
refant      =     'ea??'
gaintype    =     'K'
gaintable   =     'previous cal tables'
```

- Choose 1 min of data on a strong source (through `selectdata` → `timerange`).
- `refant` should have baselines to all the antennas in the selected time range.
- This is not a Global Fringe Fitting; it solves for antenna based single-band delays or multi-band delays.

A note on the new parameter *docallib*

In CASA version 4.5.2, make sure

```
docallib = False
```

- `docallib` refers to a "calibration library", a new portable interface for describing ensembles of calibration replacing `gaintable, gainfield,` etc… parameters.
- Will enable on-the-fly calibration in various tasks.
- Will provide increased capability and flexibility.
- Can be used now but remains limited in some aspects.
- New features, additional flexibility, and broader deployment in more tasks will be offered in CASA 4.6 and later releases.

Before Bandpass Calibration

- Bandpass calibration is not only needed for spectral-line observations, but also for continuum.
- Before calibrating the bandpass, do phase-only calibration on the bandpass calibrator (to be applied when calibrating the bandpass).
  - Prevents decorrelation when vector averaging.
  - Critical for high frequency observations.
  - Can also be used in low frequency observations.

Initial Phase only calibration: *gaincal*

- Run *gaincal* on the bandpass calibrator using:
  - a short solution interval, and
  - a few channels per spw (free of RFI).
- This table should only be used while calibrating the bandpass.
- In *gaincal*, set

```
– caltable  ='bpphase.gcal’
– field     ='bandpass_cal_field_name_or_#'
– calmode   ='p'
– gaintype  ='G'
– gaintable ='various calibration tables'
– solint    ='a short time interval'
– spw       ='x~y:n~m'
```

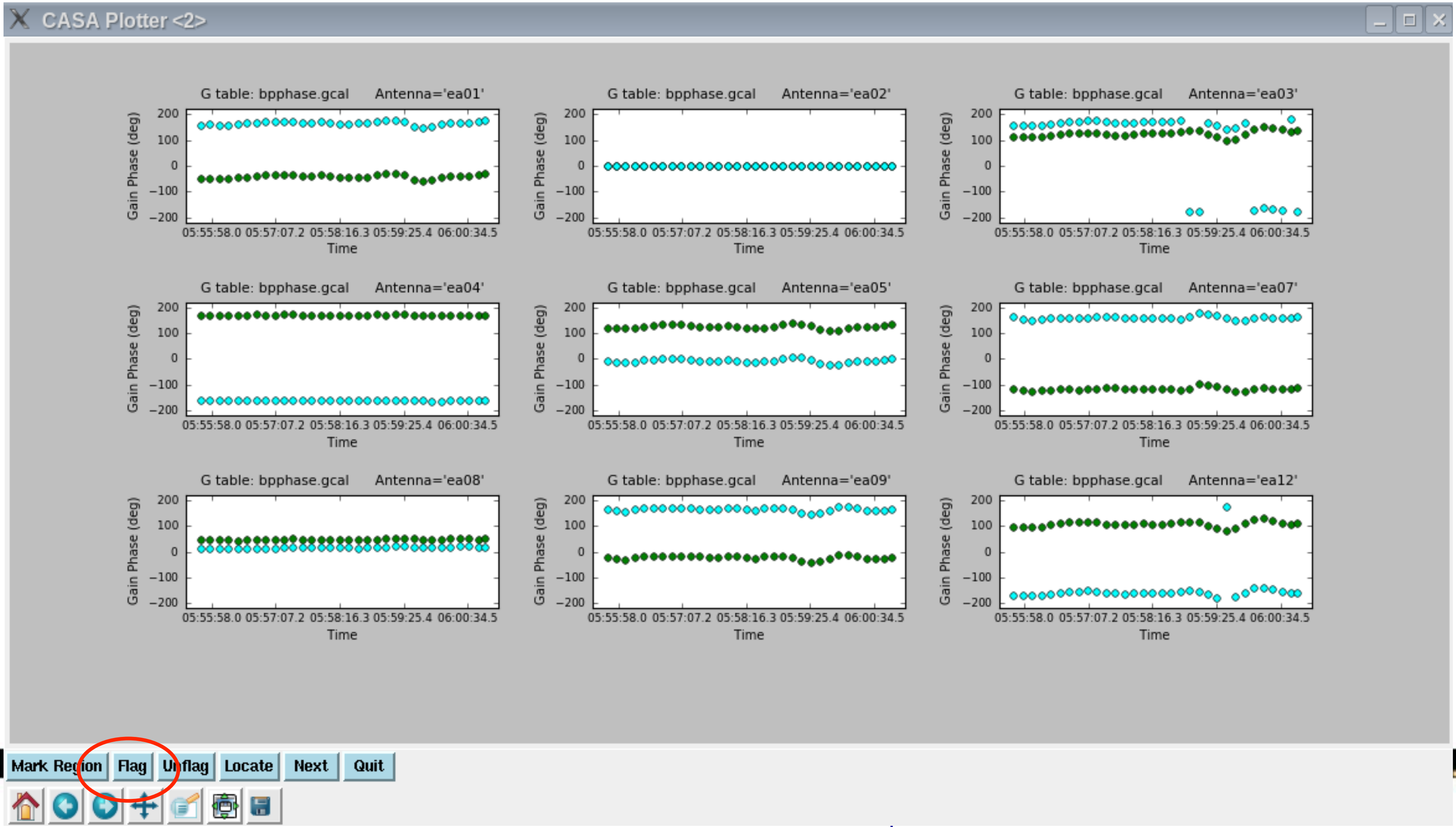
Initial Phase only calibration
Plotting the solutions: *plotcal*

- *plotcal* is a multi-purpose plotter (editor) for calibration results
- To plot the phase calibration results:

```
caltable    = 'bpphase.gcal'
xaxis       =     'time'
yaxis       =    'phase'
spw         =        '1'
subplot     =        331
iteration   =  'antenna'
plotrange   = [0, 0, -200, 200]
```

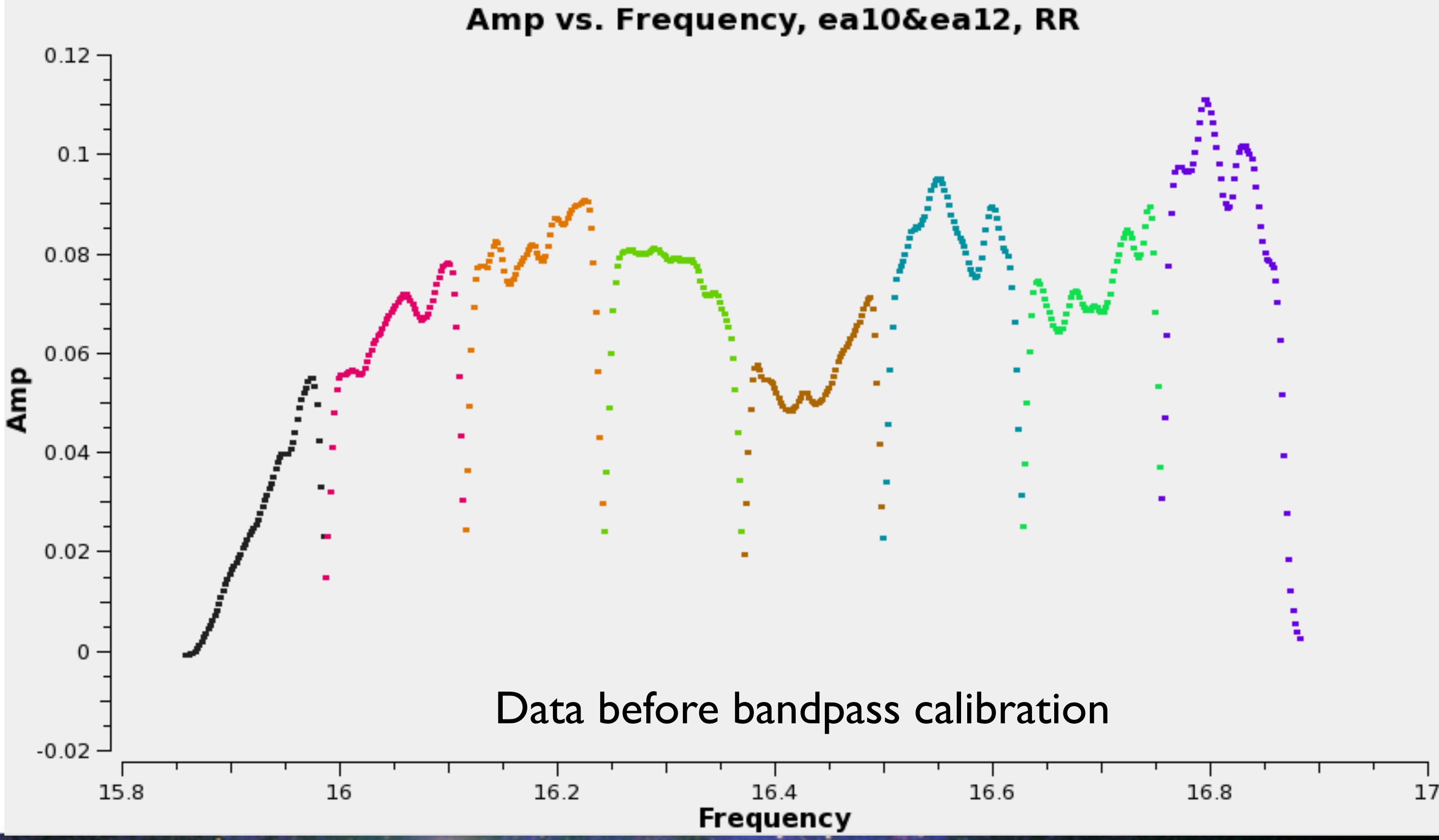
Initial Phase only calibration

Plotting the solutions: *plotcal*



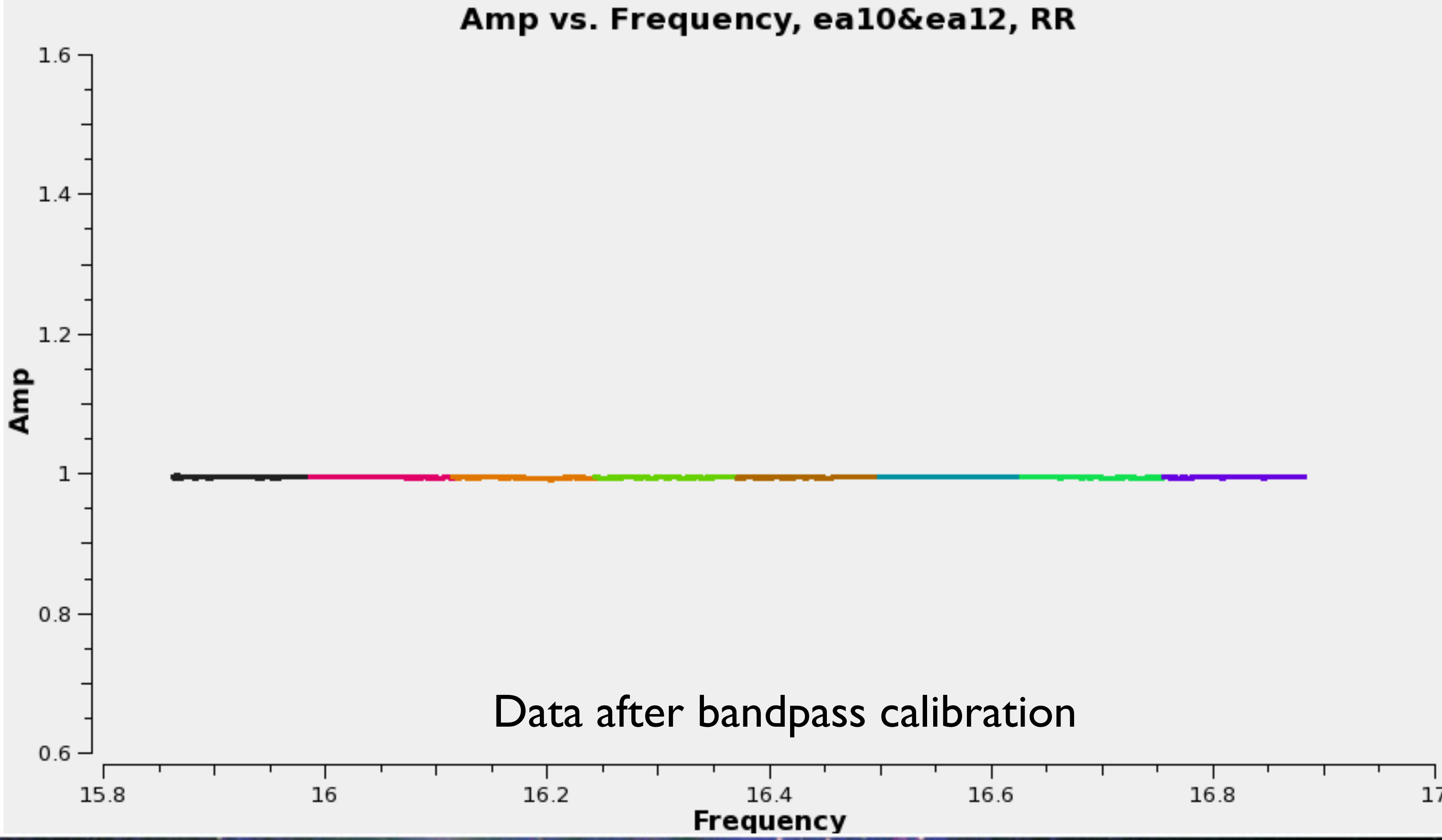
Bandpass Calibration: *bandpass*

- Needed for continuum observations too.



Bandpass Calibration: *bandpass*

- Needed for continuum observations too.



Bandpass Calibration: *bandpass*

```
caltable     =    'bandpass.bcal'
field        =    '?'
solint       =    '?'   [time and optionally frequency]
refant       =    'ea??'
solnorm      =    False
bandtype     =    B    or   BPOLY
gaintable    = various calibration tables
```

- If using a source other than the flux calibrator, the spectral index (and the spectral curvature) should be accounted for.
- CASA will report these while bootstrapping the flux densities, and store the numbers in a dictionary.
  - Use *setjy* to make use of these values.

Bandpass Calibration

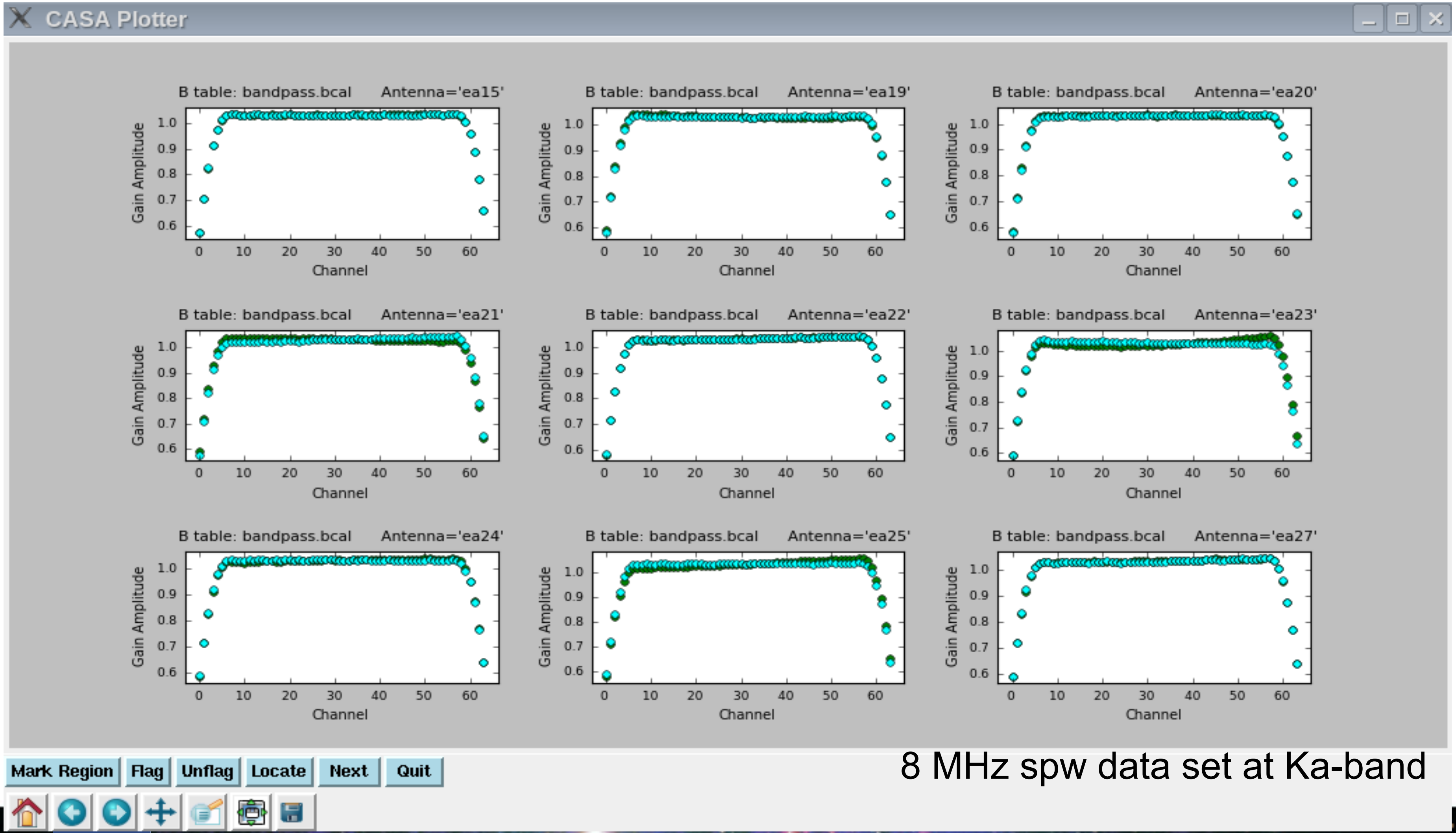
Plotting the solutions: *plotcal*

```
caltable        = ‘bandpass.bcal'
xaxis           =     ‘chan'
yaxis           =     ‘amp'   or  ‘phase’
spw             =        '1'
subplot         =        331
iteration       =  'antenna'
```

NRAO

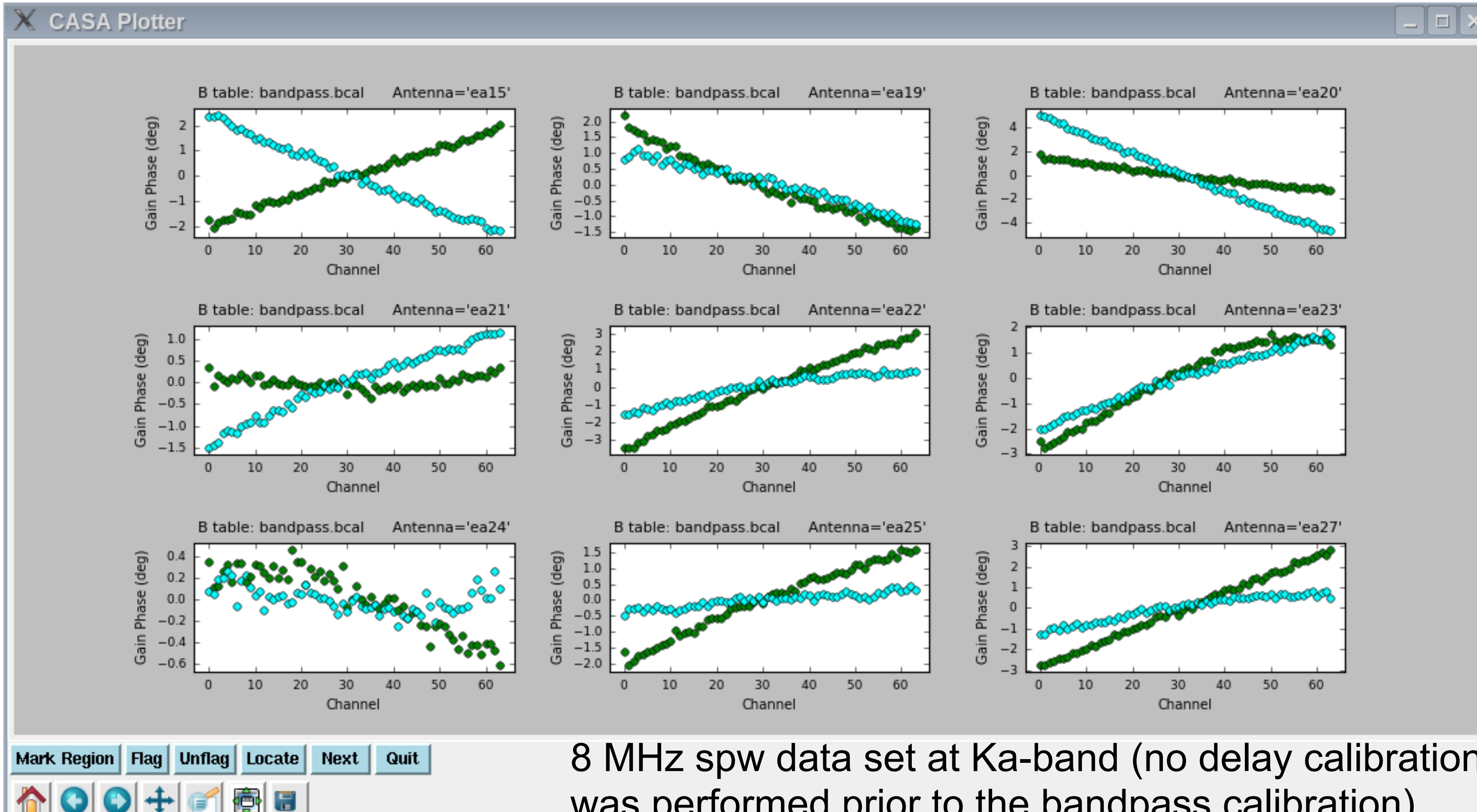
Bandpass Calibration

Plotting the solutions: *plotcal*



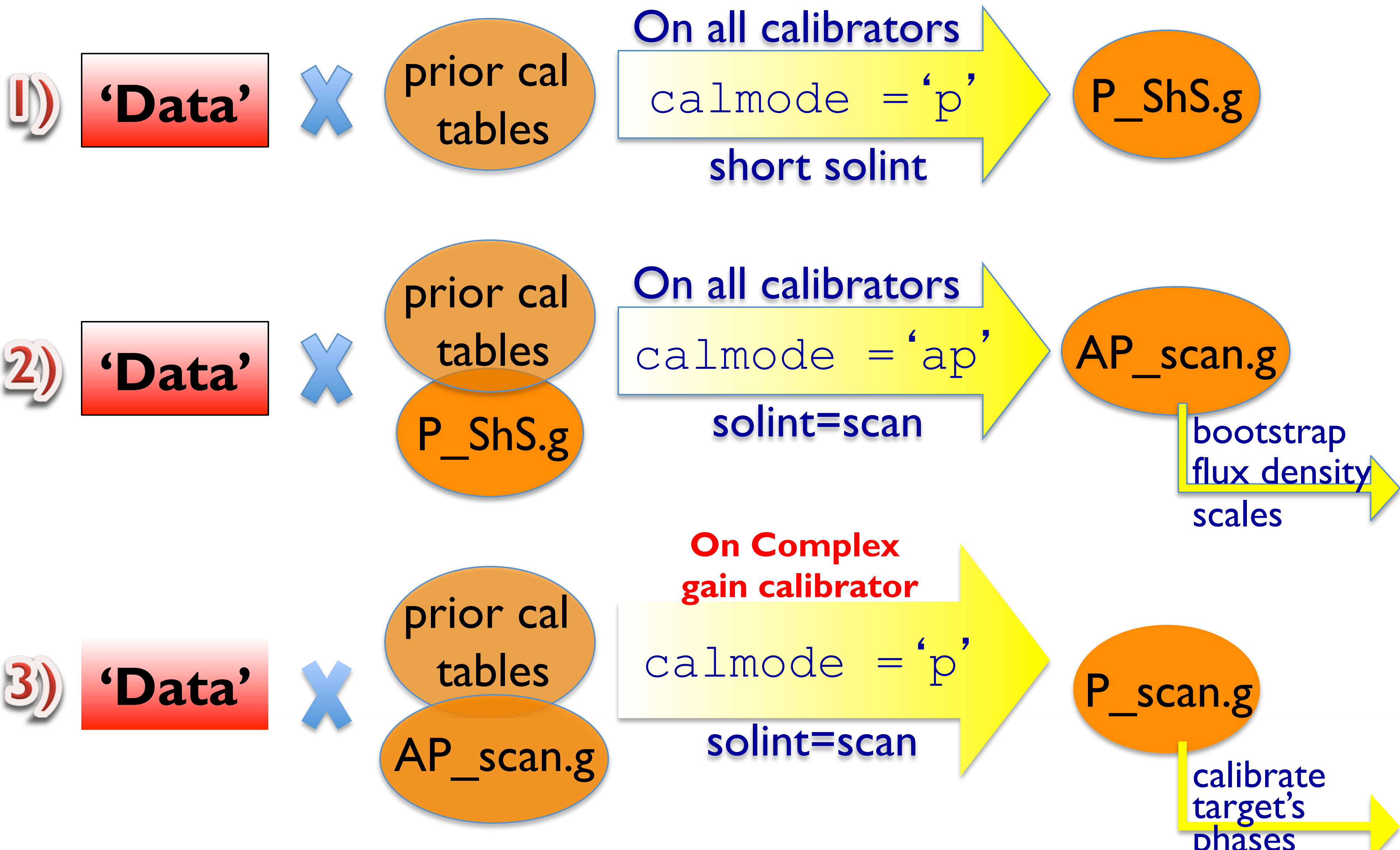
8 MHz spw data set at Ka-band

Bandpass Calibration
Plotting the solutions: *plotcal*



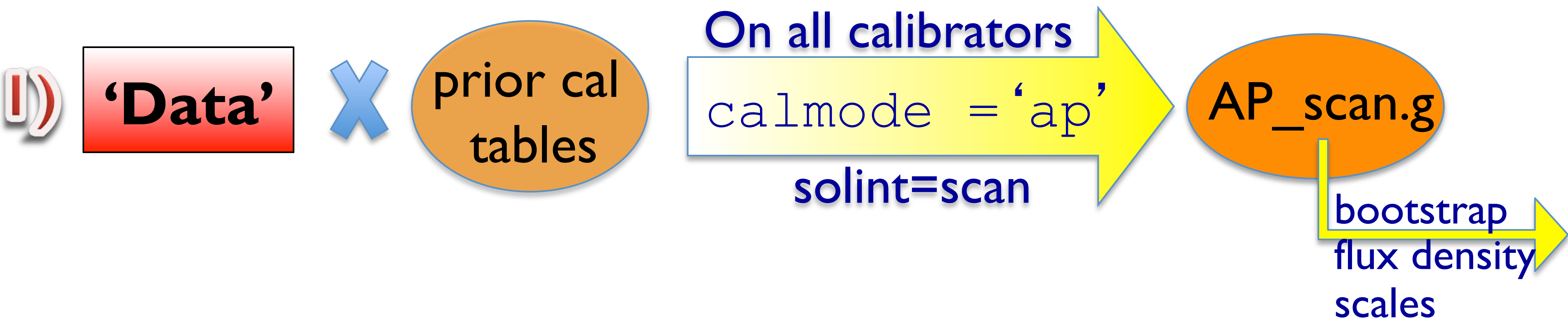
8 MHz spw data set at Ka-band (no delay calibration was performed prior to the bandpass calibration).

Complex Gain Calibration: *gaincal*, High Freq



Examine the resulting tables with *plotcal*

Complex Gain Calibration: *gaincal*, Low Freq



- Examine the resulting table with *plotcal.*
- If the phases show rapid variations (e.g., due to ionosphere), use the method outlined for high frequencies.
- The VLA calibration pipeline uses the HF approach.

Polarization Calibration

- *Gaincal*
  - solving for the cross-hand delays
- *Polcal*
  1. solving for the leakage terms
  2. solving for the R-L polarization position angle

➢ For VLA observations, and particularly for wide bandwidth observations: have channel based solutions for the leakage terms and for the R-L polarization position angle.

➢ Both CASA and AIPS allow solving for these per spectral channel.

Polarization Calibration: *gaincal*

- For polarization calibration, you will typically observe
  - A source to calibrate the leakage terms (this can be a polarized or an unpolarized source), and
  - A source with very well known polarization characteristics to calibrate the polarization position angle.
- Before running *polcal*, calibrate the cross hand delays (critical if your leakage calibrator is polarized):
  - Use the (polarized) position angle calibrator (use *setjy* to provide its Q and U values).
  - Run *gaincal* with `gaintype = 'KCROSS'`
  - Examine the resulting table with *plotcal*.
  - Apply the resulting table in subsequent steps.

Polarization Calibration: *polcal*
1. Solving for the leakage (D) terms (instrumental pol.)

- For an unpolarized calibrator (*Q=U=0*):
  - Use `poltype = 'Df'` to solve for the leakage terms (D) on per channel (f) basis.
    - Single scan
- For a polarized calibrator with unknown polarization:
  - Use `poltype = 'Df+QU'` to solve for channel base leakage terms & apparent source polarization.
    - Several scans, and a good parallactic angle coverage

Examine the resulting tables with *plotcal*.

Polarization Calibration: *polcal*
2. Solving for the R-L polarization position angle

- To obtain an accurate polarization position angle, the R-L phase needs to be calibrated.
- In *polcal*, use `poltype = 'Xf'` for a frequency dependent polarization position angle calibration.
- Requires the use of a (polarized) source with known polarization angle (use *setjy* to set its Q and U values).
- Examine the resulting table with *plotcal*.

Scale flux density: *fluxscale*

- Bootstraps the flux density scale of the secondary calibrators.
- Uses the scan based `'ap'` gain table

AP_scan.g

```
vis          =  'my.ms'
caltable     =  'input ap table'
reference    =  'field # of the flux cal'
fitorder     =  1 or  2
fluxtable    =  'output table'
incremental  =   True or False
```

- Reports the flux density values per calibrator per spw.
- Fits across the spw's of each calibrator to report a *spectral index and curvature* (can be supplied through *setjy* if needed).

Scale flux density: *fluxscale*

- Bootstraps the flux density scale of the secondary calibrators.
- Uses the scan based `'ap'` gain table AP_scan.g

```
vis          =  'my.ms'
caltable     =  'input ap table'
reference    =  'field # of the flux cal'
fitorder     =  1 or  2
fluxtable    =  'output table'
incremental  =   True or False
```

- Reports the flux density values per calibrator per spw.
- Fits across the spw's of each calibrator to report a *spectral index and curvature* (can be supplied through *setjy* if needed).

Scale flux density: *fluxscale*

```
fluxtable      =   'output table'
incremental    =      T or F
```

- If `incremental = F`

  The `output table` replaces the input `'ap'` table.

- If `incremental = T`

  The `output table` contains only the scaling factors, and should be used alongside the input `'ap'` table.

Apply Calibration: *applycal*

```
field         =        '?'
interp        = nearest or linear
gaintable     = various calibration tables
gainfield     = fields corresponding to the
                    above tables
parang        = False   (True if polcal was run)
calwt         =       False
```

- One field at a time, but targets with the same calibrators can be grouped together.
- Use the appropriate tables for each source.
- Make sure to match the gainfield entries with the gaintables.
- calwt can also be a list, with each element corresponding to a calibration table. Make sure it is not an empty list.

Examine the calibrated data (the corrected column) with *plotms*.

Flag, if needed, and re-calibrate.

The VLA Calibration Pipeline

- Performs basic flagging and calibration using CASA.
- Primarily designed for Stokes *I* continuum data.
- To run successfully, the scan intents in the scheduling block must be set correctly.
- Information are at:

https://science.nrao.edu/facilities/vla/data-processing/pipeline

- Many more details in talks scheduled during this workshop.

Split the target(s): *split (currently split2)*

- Split the target source(s) using the corrected column.
- Optionally:
  - apply time averaging
  - apply frequency averaging
  - choose spectral windows/channels
  - choose certain antennas
  - choose a certain UV range
  - choose particular scans
  - choose polarization
- The *split*-ed data will occupy the 'data column' in the resulting MS.
- Self-calibration can be performed if the target is strong enough.
- Self-calibrated data will be placed in the corrected column (upon running applycal).

Weights:

- The weights are initialized to be channel bandwidth and time dependent ($2\Delta\nu\Delta\tau$).
- Weights are per spw.
- CASA 4.4 and later versions also support spectral weights:
  - Weights are per channel.
  - Not used in VLA data (TBD channelized *statwt*)

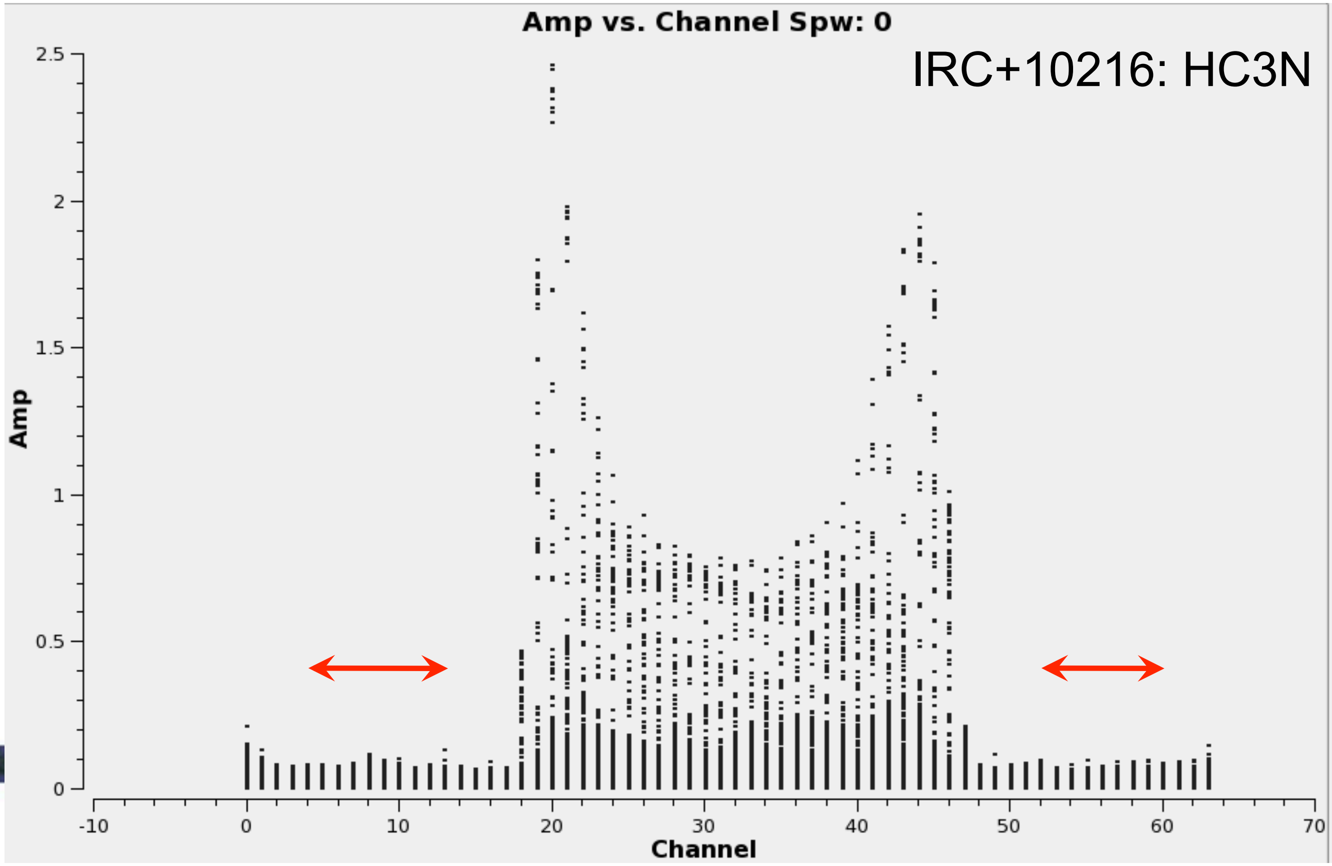
(Re)Weighting visibilities: *statwt*

- *statwt* reweights the visibilities according to their scatter.
- Needed to down-weight underperforming antennas, or spw's affected by RFI.
- How/when to use it:
  - The data should be fully calibrated.
  - Recommended to *split* the data (source) of interest first (some time averaging might be helpful).
- Channelized version of *statwt* is coming soon.

Continuum Subtraction: *uvcontsub*

```
vis            =  'my.ms'
fitspw         =  '0:4~13;52~60'
excludechans   =   False    or    True
want_cont      =        False
```

can choose multiple spw's



Doppler Correction: *cvel*

- The VLA does not offer Doppler Tracking, but only Doppler setting.
- The line of interest may shift over one or more channels during the observations.
- If adding different observing blocks, one can choose to first Doppler correct (*cvel*) each block, concatenate (*concat*) and then image (*clean*). However, the imaging task *clean* can do this on-the-fly.
- *cvel* should be run if one needs/wants to do self-calibration using a (narrow) strong spectral line.

NRAO

The End of Part II