EUROPEAN ARC
ALMA Regional Centre || Allegro

ALMA CASA Self-Calibration

Allegro - CASA Tutorial Day

Daniel Harsono

2 Nov 2018

Self cal — what is it? why?

Please ask questions throughout if anything is unclear

Self cal — what is it? why?

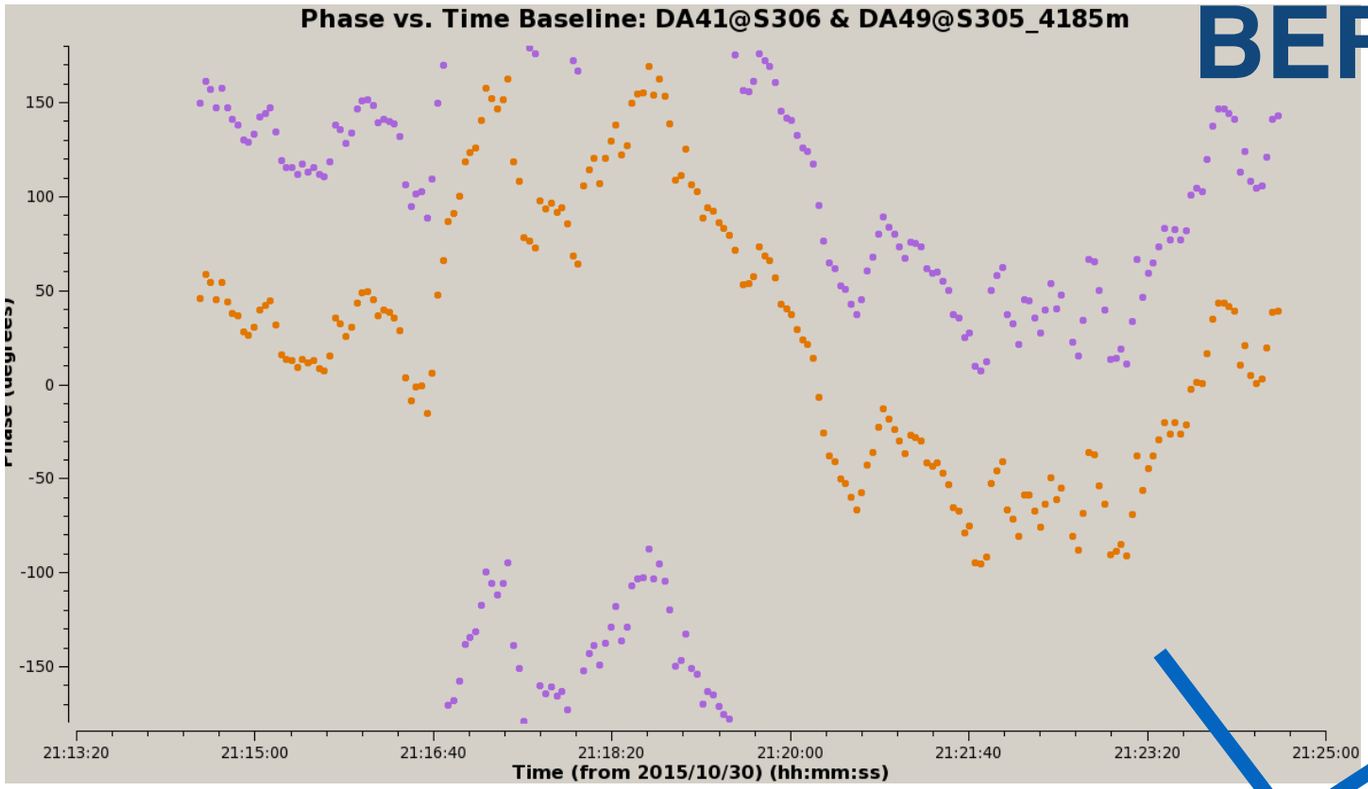
- **Calibration of the amplitudes and phases by using the source 'itself'**
  - **Calibration performs self-cal - on the calibrators!**
  - **these have 'known' visibilities so you find antenna solutions to match accordingly**

Self cal — what is it? why?

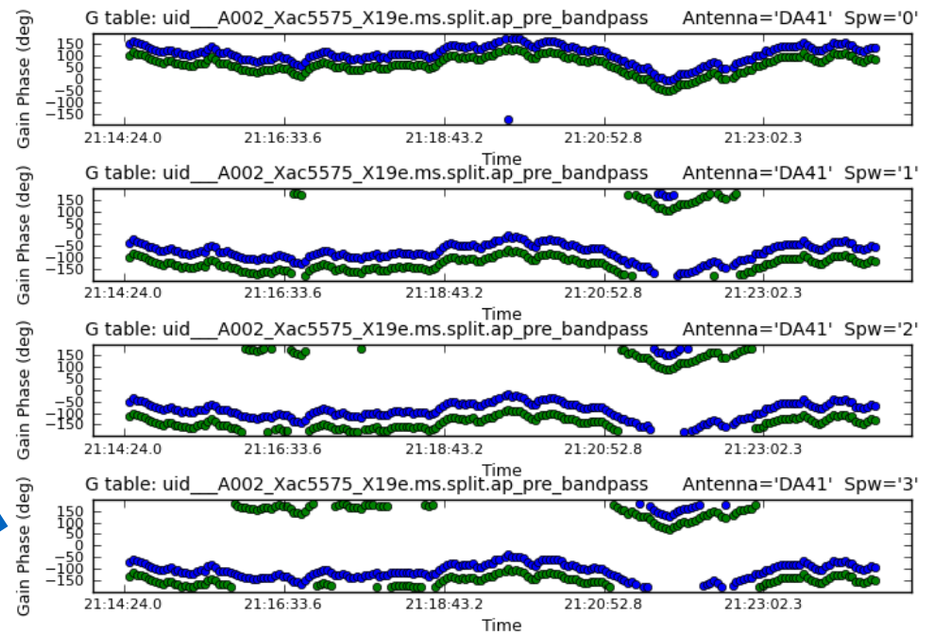
- **Calibration of the amplitudes and phases by using the source 'itself'**
  - **Calibration performs self-cal - on the calibrators!**
  - **these have 'known' visibilities so you find antenna solutions to match accordingly**
    - **QSO: point source - constant amp / zero phase**
    - **SSO: amp model - zero phase (if unresolved)**

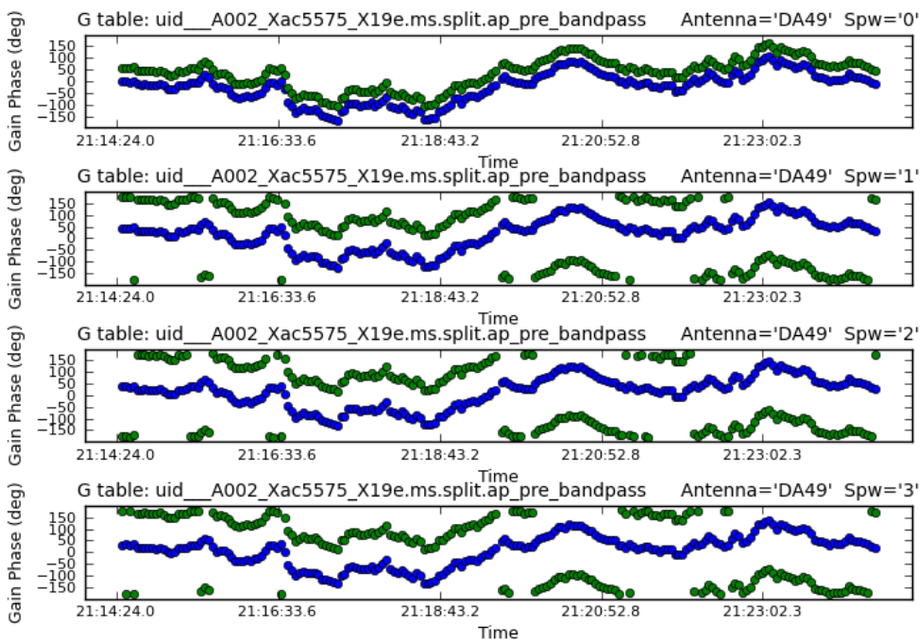
Recall - phase-up for bandpass

BEFORE

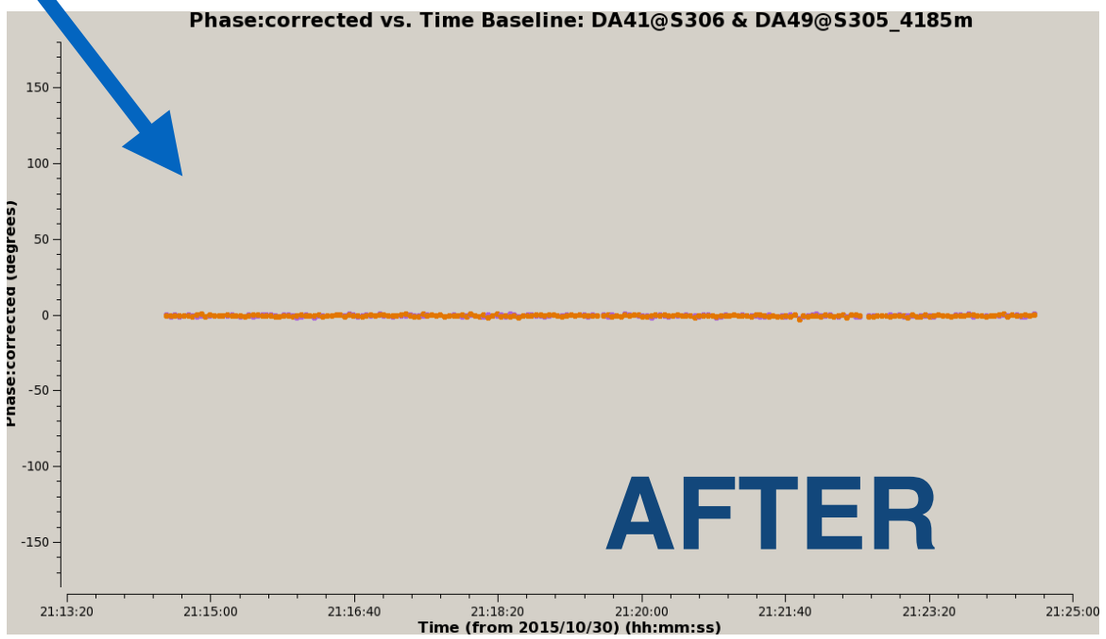


DA41 - Soln.





DA49 - Soln.



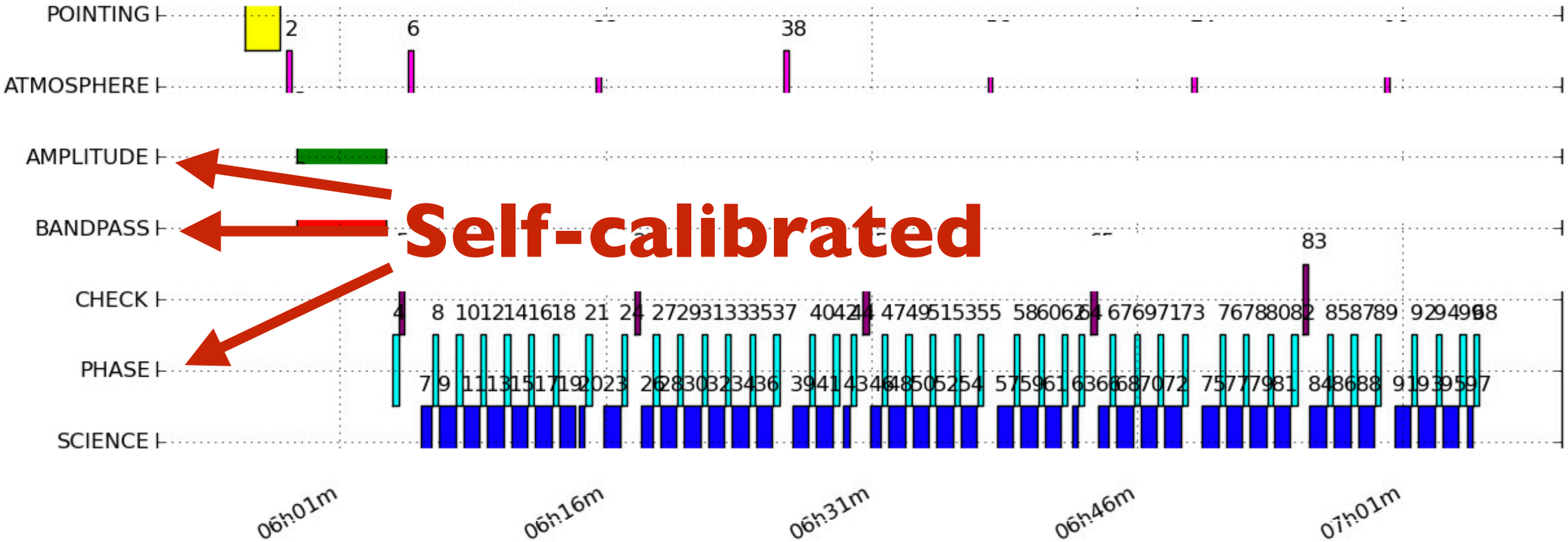
AFTER

Self cal — what is it? why?

- **Calibration of the amplitudes and phases by using the source 'itself'**
  - **Calibration performs self-cal - on the calibrators!**
  - **these have 'known' visibilities so you find antenna solutions to match accordingly**
  - **for a science target you rely on the model made during cleaning**

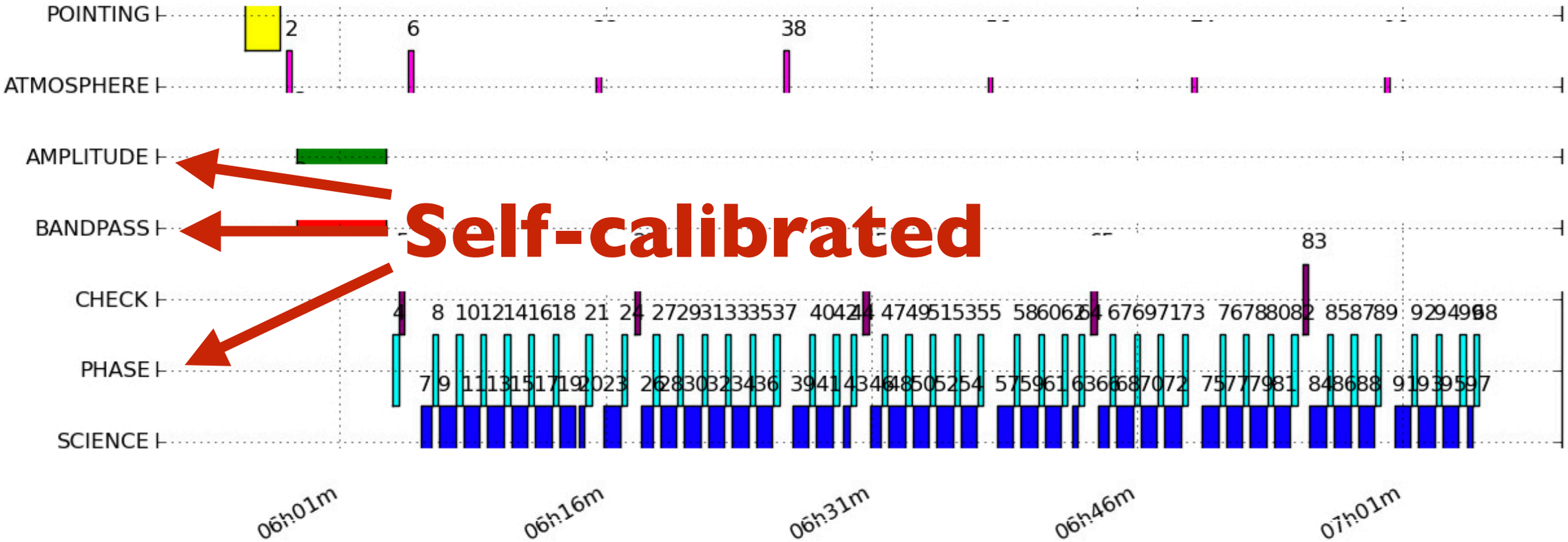
Self cal — what is it? why?

- **Further correction of the atmospheric phases**
  - **Calibration interpolates phase solutions from a gain calibrator**



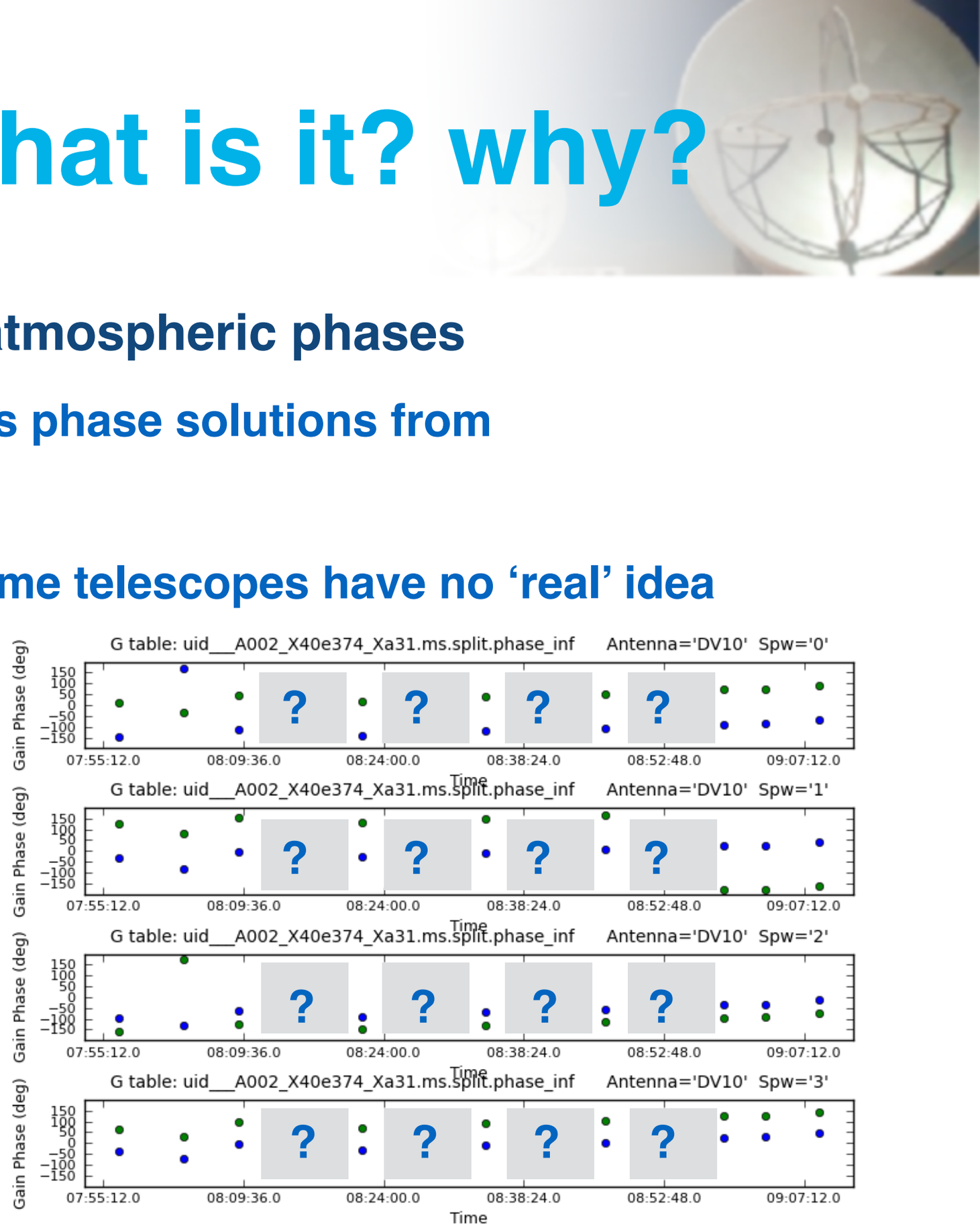
Self cal — what is it? why?

- **Further correction of the atmospheric phases**
  - **Calibration issues: the solutions from self-calibration were not good enough**



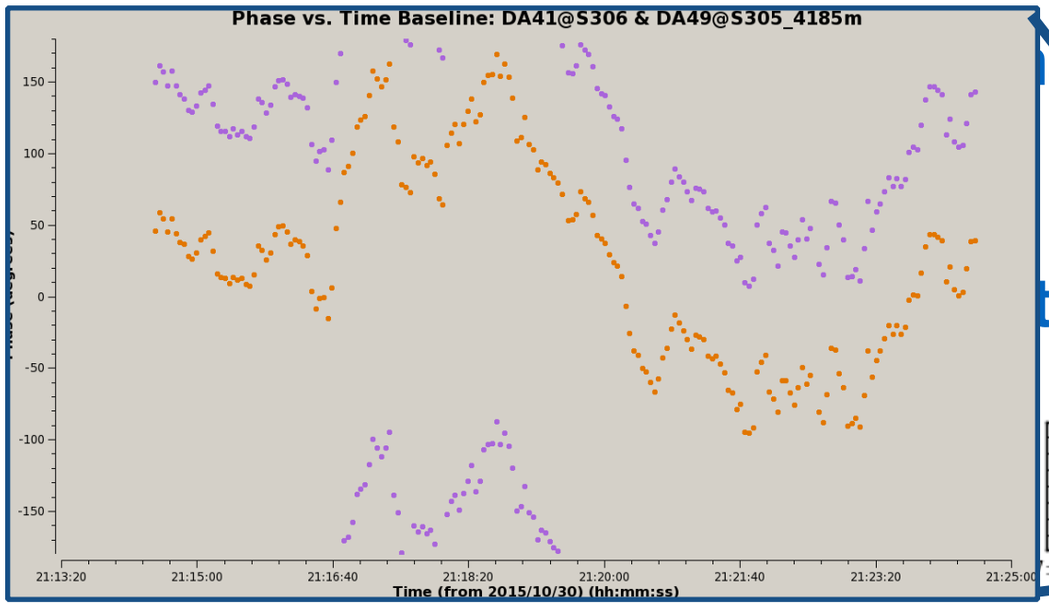
Self cal — what is it? why?

- **Further correction of the atmospheric phases**
  - **Calibration interpolates phase solutions from a gain calibrator**
  - **during this on-target time telescopes have no 'real' idea about the atmosphere**



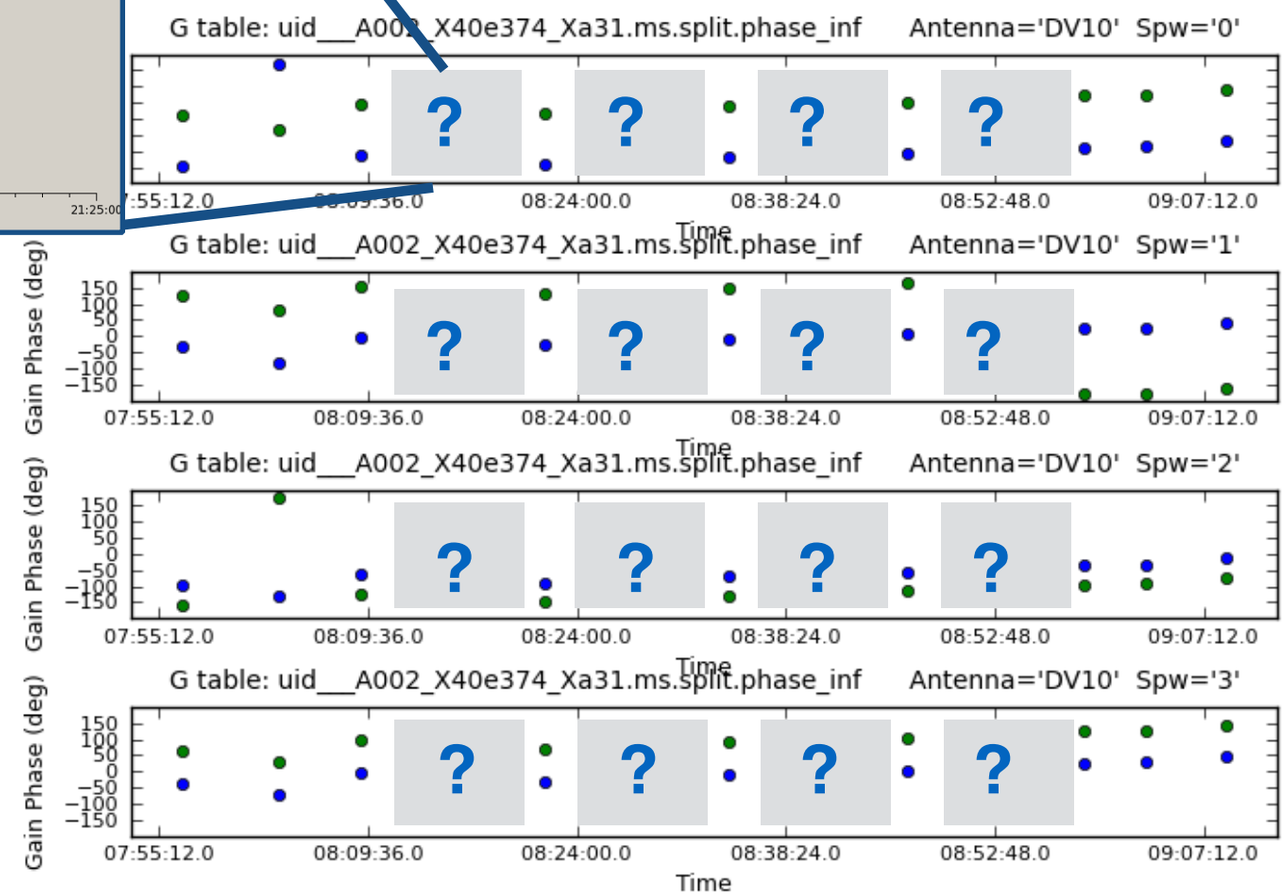
Self cal — what is it? why?

- **Further correction of the atmospheric phases**



ase solutions from

telescopes have no 'real' idea



Imagine ???

Self cal — what is it? why?

- **Further correction of the atmospheric phases**
  - **with enough S/N on target and a 'good' model you can further correct the phases**
  - **if there were significant phase fluctuations you can expect:**
    - **sharper images**
    - **increase peak flux**
    - **lower noise**

Self cal — what is it? why?

- **Phases relate to the 'position' of the flux in the images**
  - **incorrect phases can put the flux in the 'wrong' place**
  - **self-cal solves for the phase during the on-target time**
    - **puts the flux in the correct place: 'phased-up'**
    - **flux is no longer spread around the map**
    - **it is repositioned to the science source/s**
  - **phase noise also causes de-coherence - loss of signal**

$$\langle V \rangle = V_o \times \langle e^{i\phi} \rangle = V_o \times e^{-\phi_{rms}^2/2}$$

  - **self cal reduces the phase noise - increasing the flux**

Self cal — best practices

- Trying to match data with the model image
  - improve S/N by ‘combining’ SPW
  - select ONLY channels that made the image
    - i.e. continuum only
  - ensure image was cleaned ‘well’
  - start with long solution time - then reduce ‘solint’ as far as possible progressively (and conservatively…)

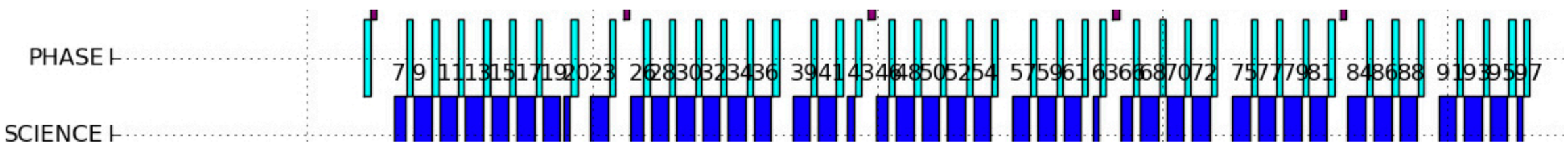
Self cal — best practices

- **Trying to match data with the model image**
  - **start with long solution time - then reduce ‘solint’ as far as possible**

‘Inf’, combine=‘scan’

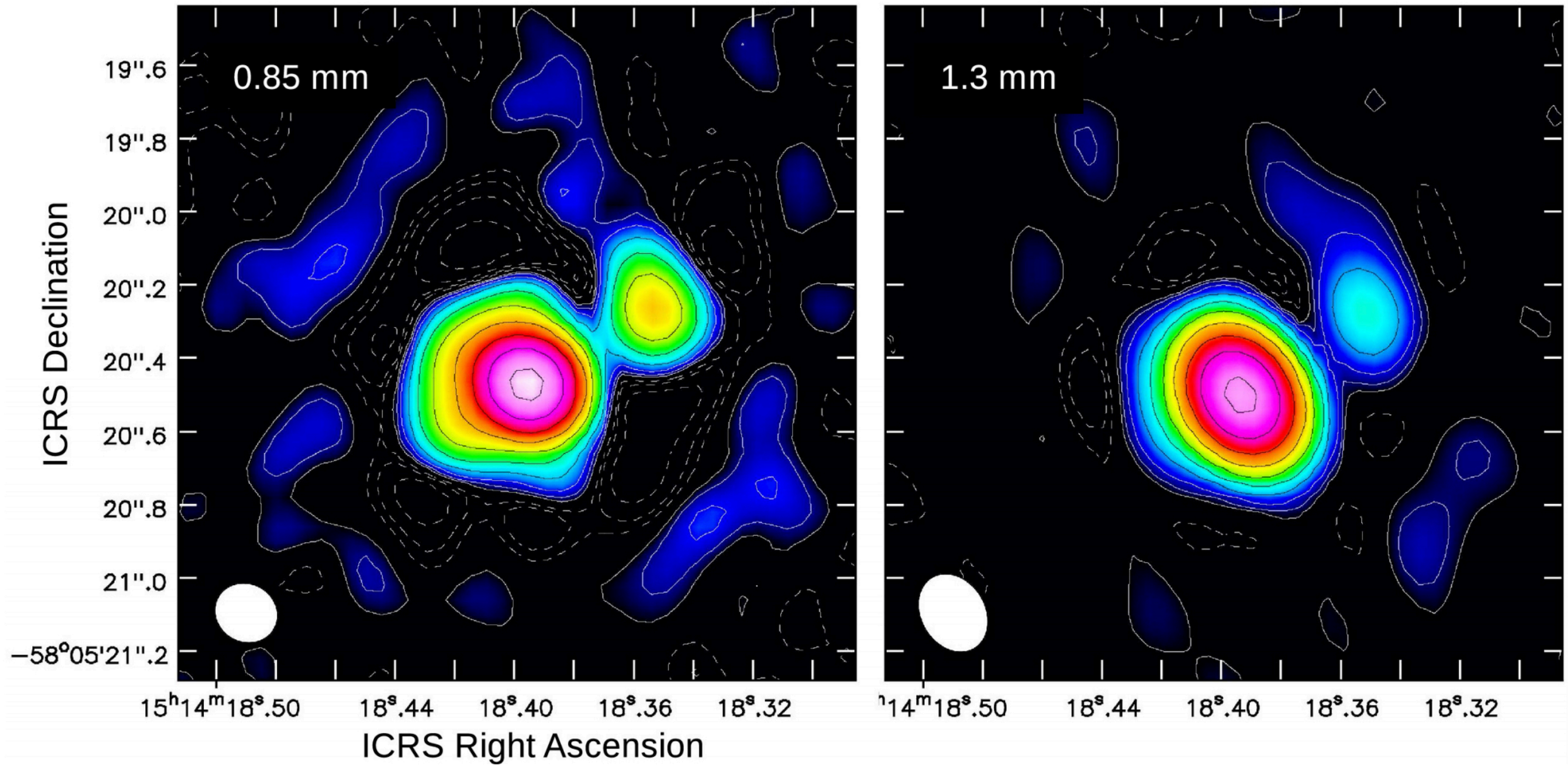
‘153s’, combine=‘scan’

‘int’



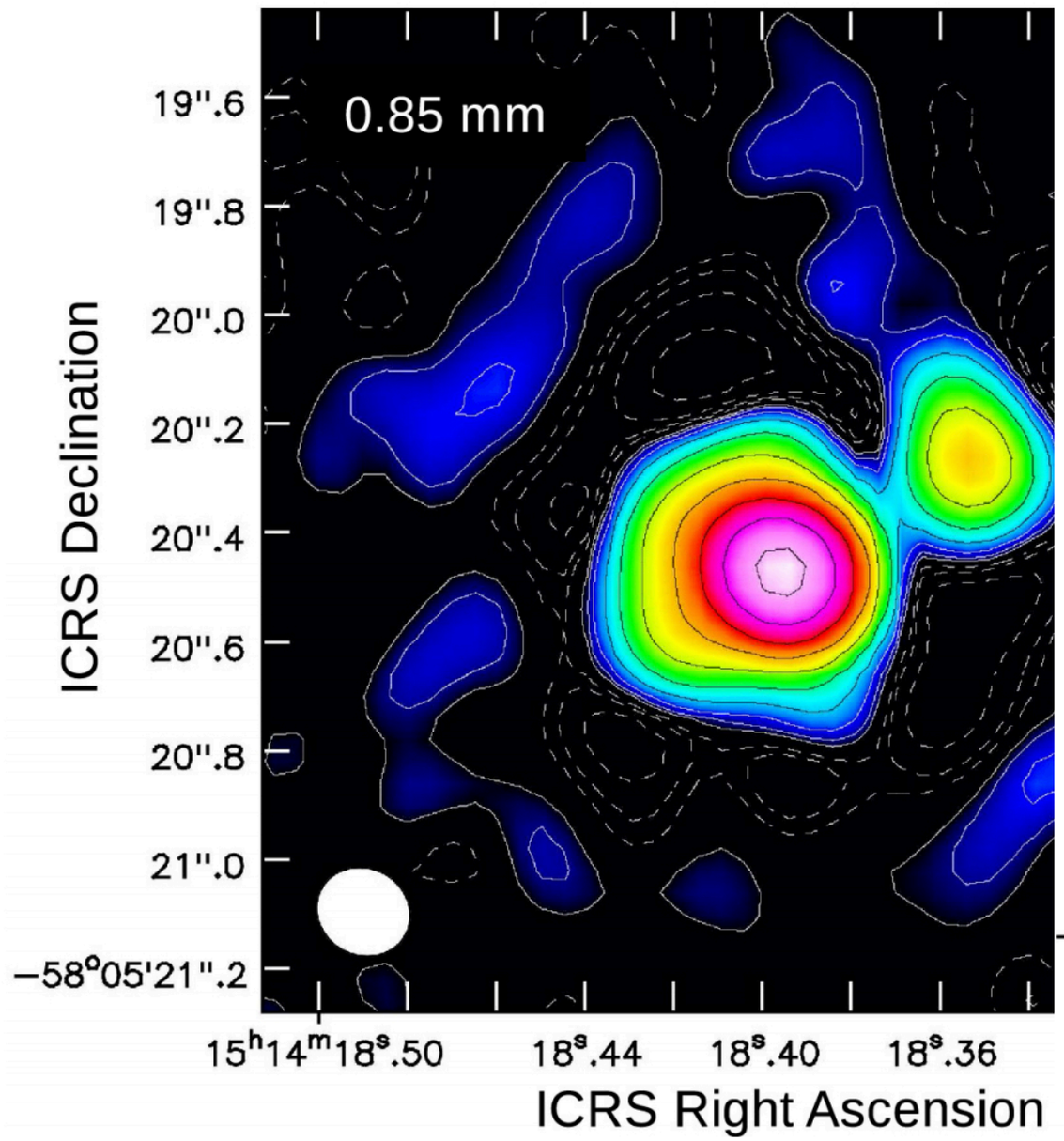
Self cal — examples

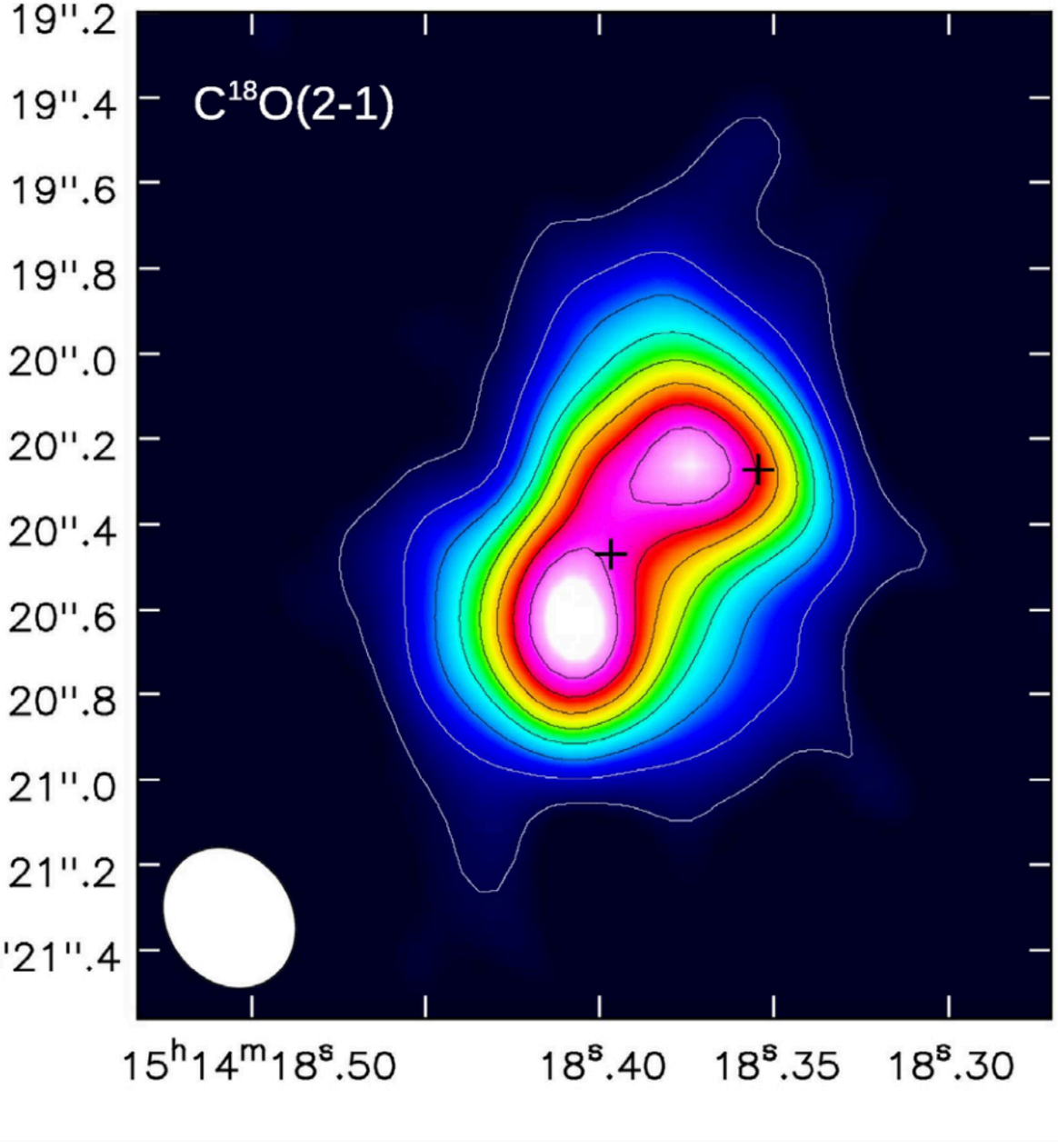
Planetary Nebula: Gomez et al. 2018



Self cal — examples

Planetary Nebula: Gomez et al. 2018





Self cal — examples

Disks: Long et al. 2018

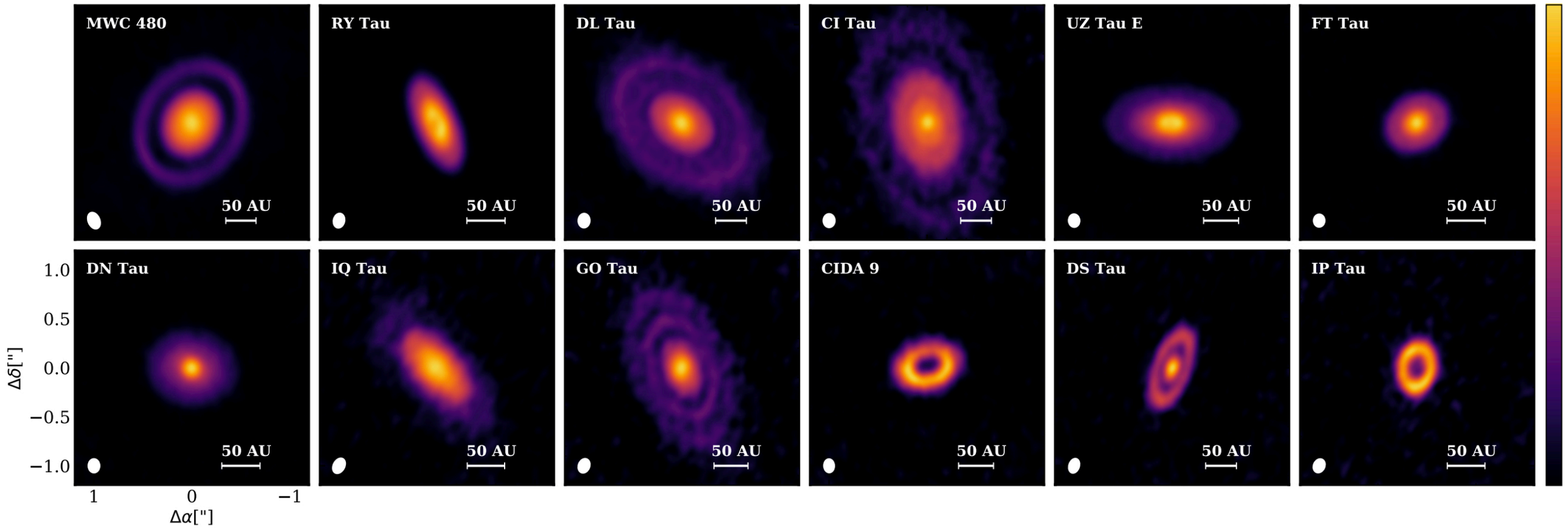
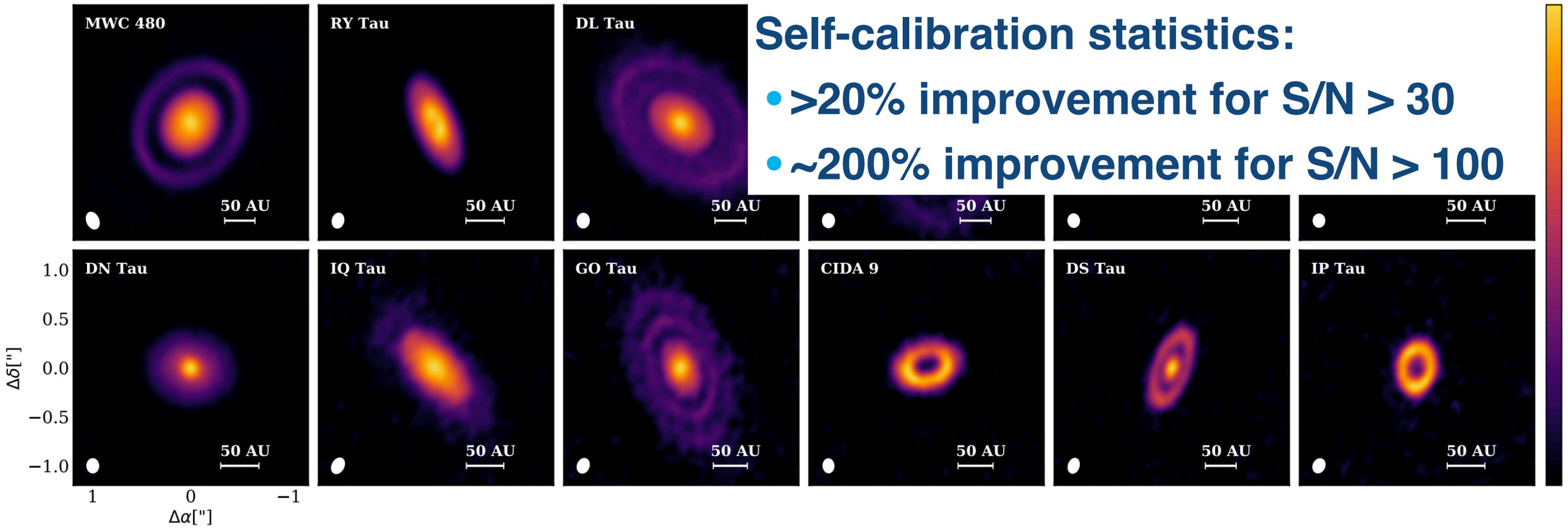


Figure 1. Synthesized images of the 1.33 mm continuum with a Briggs weighting of robust = 0.5. The images are displayed in order of decreasing mm flux, from the top left panel to the bottom right panel, and are scaled to highlight the weaker outer emission. The beam for each disk is shown in the left corner of each panel.

Self cal — examples

Disks: Long et al. 2018



Self-calibration statistics:

- **>20% improvement for S/N > 30**
- **~200% improvement for S/N > 100**

Self cal — the steps (1)

- Split/copy your ms
  - source only
- Select continuum SPW and channels
- Run an interactive clean

May need a separate folder

```
visname='calibrated_forselfcal.ms'  # or your MS filename
cell='0.015arcsec'  # set this for the imaging - 5 to 8 time less than beam
imagesize=2048      # enough to image at least ALL primary beam
souname='scienceTarget'  # source name

# Select the SPW and CHANNELS that comprise the continuum
# e.g. 4 SPW below
# select either side for 0 and 3
# select all of SPW 1 and 2
spwcont='0:0~250;750~1023,1,2,3:0~500;900~1919'

# continuum image
os.system('rm -rf '+souname+'.B6.cont.*')

  # clean continuum
delmod(vis=visname)  # remove any model that is in the dataset

clean(vis=visname,
      spw = spwcont,
      imagename = souname+'.B6.cont',
      field='0',
      cell=cell,
      imsize=imagesize,
      threshold='0.05mJy',
      mode='mfs',
      outframe='LSRK',
      niter=50000,
      weighting='briggs',
      robust=0.5,
      interactive=True)
```

some CASA version need - usescratch = True

Self cal — the steps (1)

- **Warning:**

The example script bins the data in channels and time

DO NOT bin in time ever!

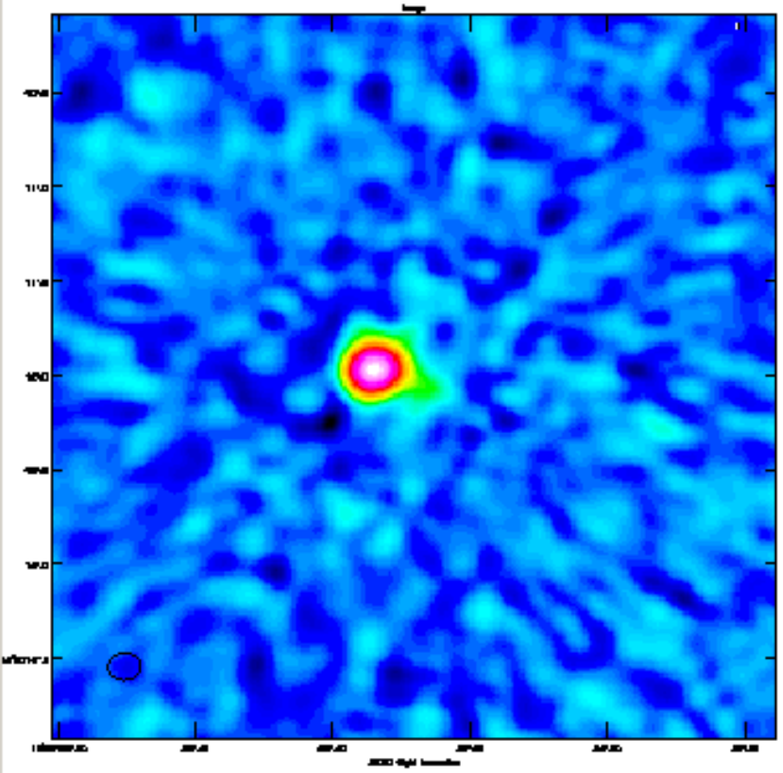
Bin in channels if your science is continuum

```
visname='calibrated_forselfcal.ms'  # or your MS filename
cell='0.015arcsec'  # set this for the imaging - 5 to 8 time less than beam
imagesize=2048      # enough to image at least ALL primary beam
souname='scienceTarget'  # source name

# Select the SPW and CHANNELS that comprise the continuum
# e.g. 4 SPW below
# select either side for 0 and 3
# select all of SPW 1 and 2
spwcont='0:0~250;750~1023,1,2,3:0~500;900~1919'

# continuum image

os.system('rm -rf '+souname+'.B6.cont.*')

  # clean continuum
delmod(vis=visname)  # remove any model that is in the dataset

clean(vis=visname,
      spw = spwcont,
      imagename = souname+'.B6.cont',
      field='0',
      cell=cell,
      imsize=imagesize,
      threshold='0.05mJy',
      mode='mfs',
      outframe='LSRK',
      niter=50000,
      weighting='briggs',
      robust=0.5,
      interactive=True)
```

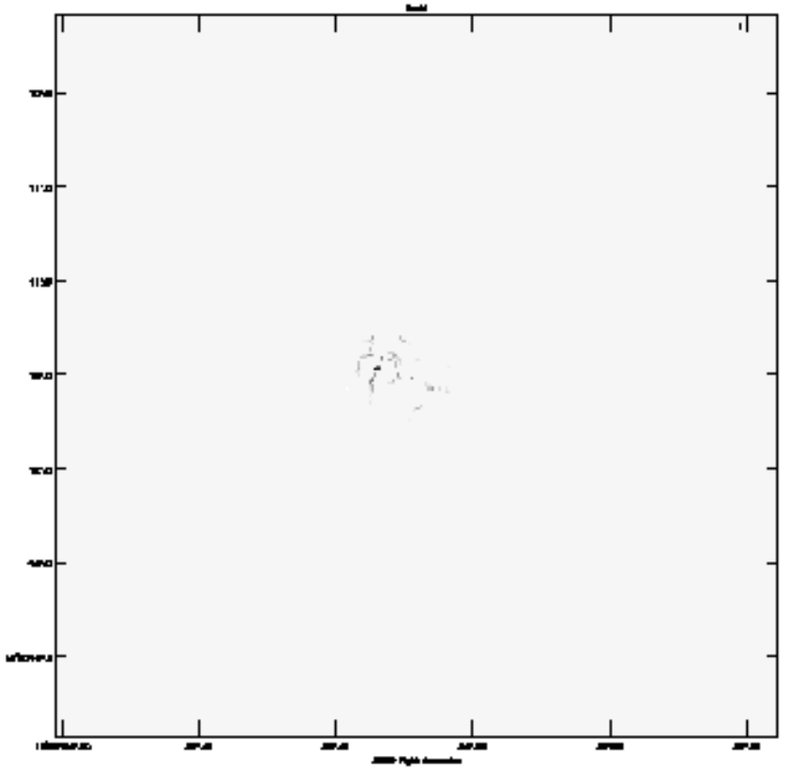
some CASA version need - usescratch = True

Self cal — the steps (1)

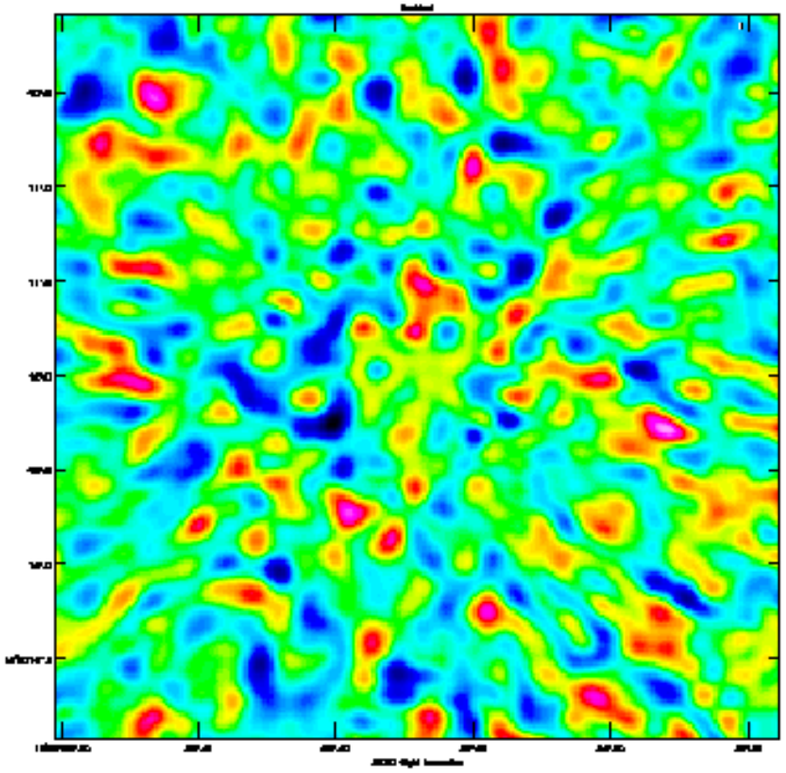
IMAGE



MODEL



RESIDUAL



- S/N ~ 120
- Compact model
- Reasonably deep

Self cal — the steps (2)

```
## inf time is ~5 mins
## find this roughly from listobs - time on source/target
os.system('rm -rf pcal1')
gaincal(vis=visname,
       caltable='pcal1',
       gaintype='T',
       refant='DA59',
       calmode='p',
       combine='spw',
       spw=spwcont,
       field='0',
       solint='inf',
       minsnr=3.0,
       minblperant=4)
```

- **Using gaincal - solve the phases of the selected SPW to match the image model - CASA 'knows' the model**

  caltable - the 1st calibration table (pcal1)

  refant - use same as calibration if possible

  calmode - p - phaseonly

  combine - 'spw' combine all inputs in spw into one

  spw= spwcont i.e. the previously selected continuum ONLY range

  solint - 'inf' - infinite time for a single scan - i.e. scan time on target

Self cal — the steps (2)

```
## inf time is ~5 mins
## find this roughly from listobs - time on source/target
os.system('rm -rf pcal1')
gaincal(vis=visname,
        caltable='pcal1',
        gaintype='T',
        refant='DA59',
        calmode='p',
        combine='spw',
        spw=spwcont,
        field='0',
        solint='inf',
        minsnr=3.0,
        minblperant=4)
```

Put in a print statement before and after incase any flagging occurs making the step easy to find

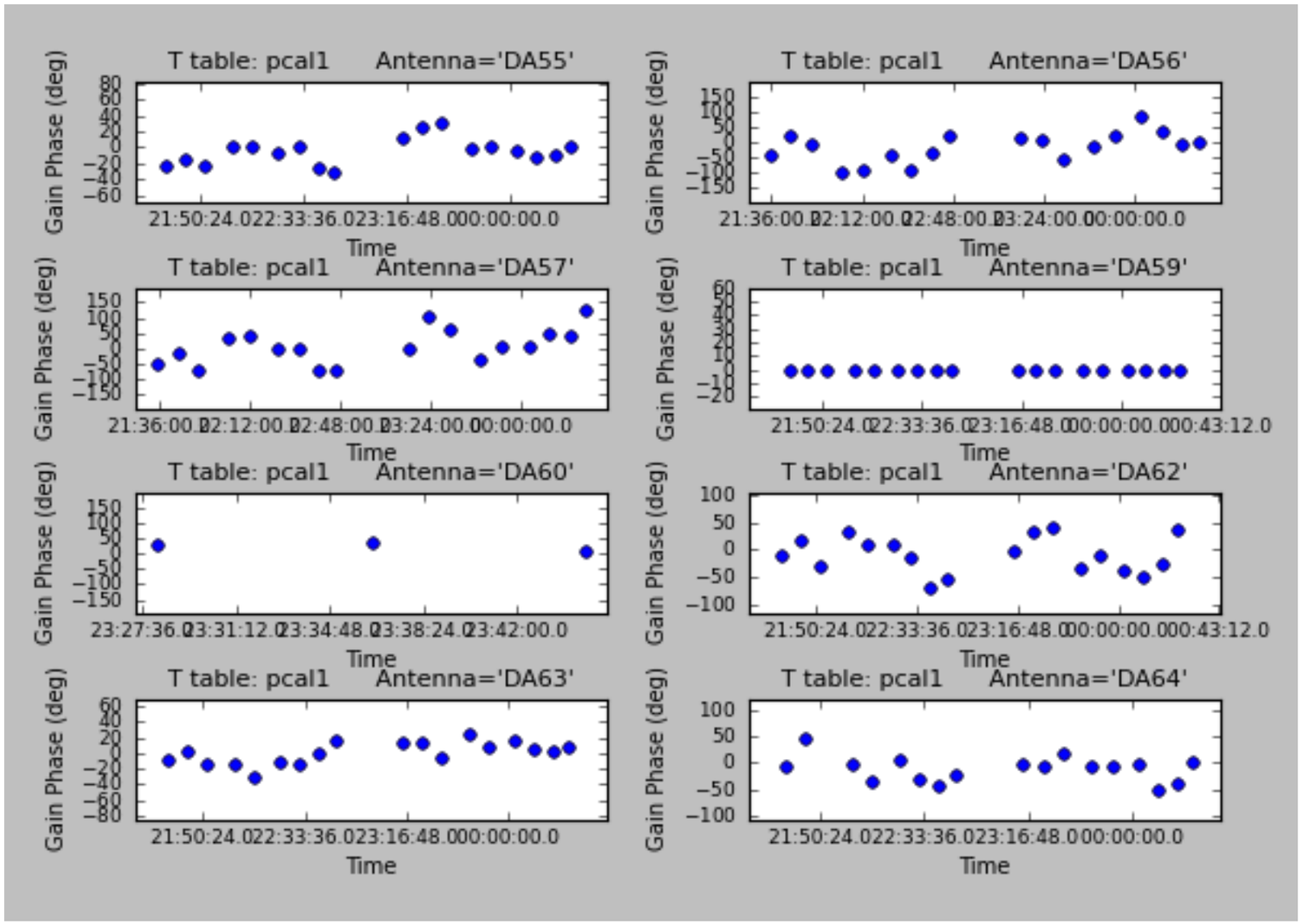
Flagging will be very detrimental and probably means you really DO NOT have enough S/N for doing self calibration (on some or all baselines)

- Using gaincal - solve the phases of the selected SPW to match the image model - CASA ‘knows’ the model
  - caltable - the 1st calibration table (pcal1)
  - refant - use same as calibration if possible
  - calmode - p - phaseonly
  - combine - ‘spw’ comtine all inputs in spw into one
  - spw= spwcont i.e. the previously selected continuum ONLY range
  - solint - ‘inf’ - infinite time for a single scan - i.e. scan time on target

Self cal — the steps (2)

```
plotcal(caltable='pcal1',xaxis='time',yaxis='phase',
        spw='',iteration='antenna',
         subplot=421,plotrange=[0,0,-180,180])
```

- **Using plotcal - make the plots of the solutions**



OPTIONAL:

showgui - default - True, False

means no screen pop-up

figfile - string - set to

produce a png, e.g.

‘pcal1.plots’

Self cal — the steps (2)

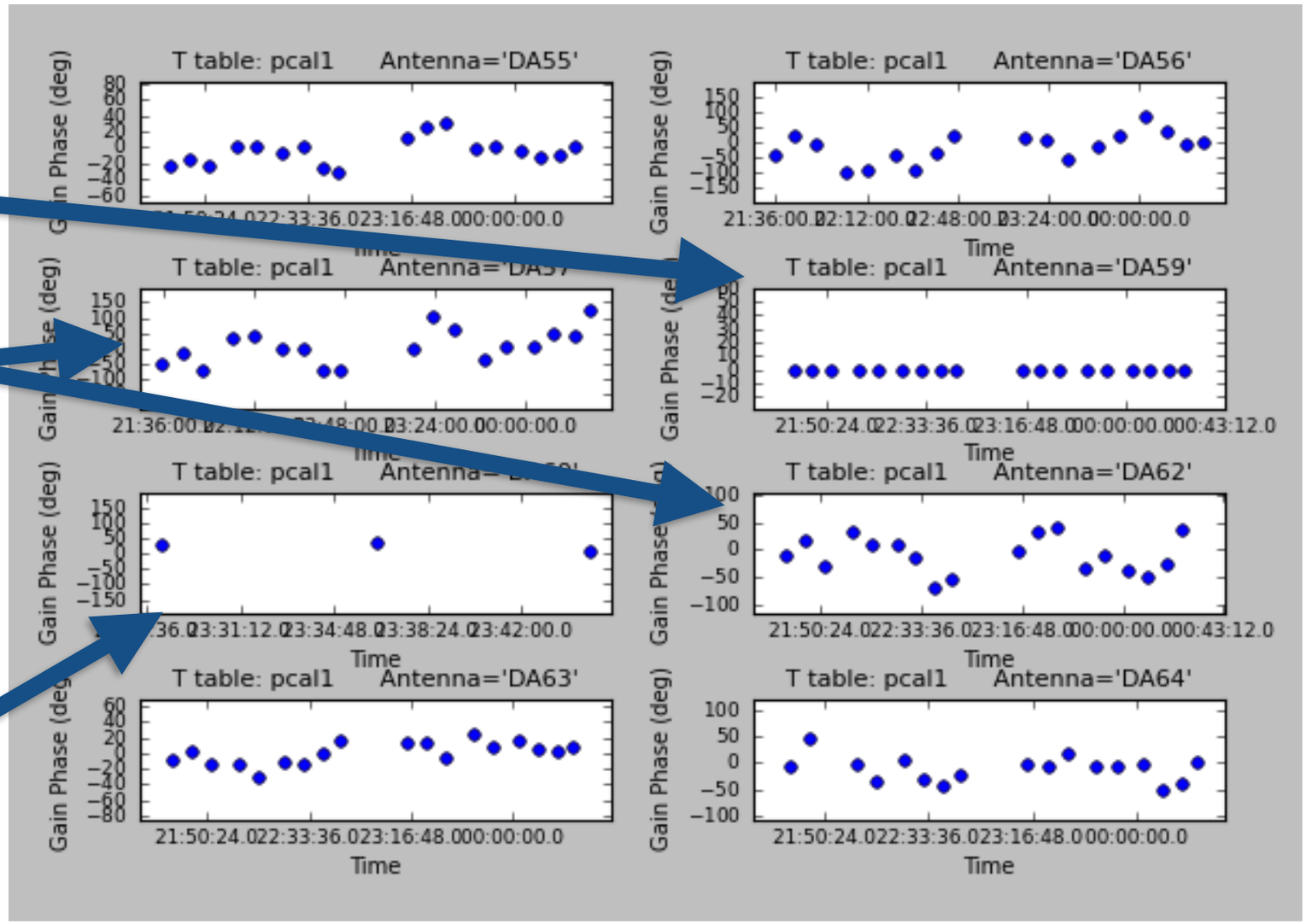
```
plotcal(caltable='pcal1',xaxis='time',yaxis='phase',
        spw='',iteration='antenna',
         subplot=421,plotrange=[0,0,-180,180])
```

- **Using plotcal - make the plots of the solutions**

refant - will always be zero

check the phases look ‘fluid’ not random noise

some antennas could be flagged - either during or previously in calibration



Self cal — the steps (3)

```
applycal(vis=visname,
     spwmap=[0,0,0,0],
     spw='',
     field='',
     gaintable=['pcal1'],
     gainfield='',
     calwt=F,
     flagbackup=T,
     applymode='calonly')  # or set as 'calflag'
```

- **Using applycal - apply the solutions to the dataset**

  this will make a ‘corrected’ data column

  spwmap - [0,0,0,0]

  maps the solutions created in SPW 0
  as a result of the combine=‘spw’ choice
  to spw = 0,1,2,3

  applymode - if all solutions ‘look’ reasonable - use ‘calonly’

  this avoids adding extra flagging to your source data if some self cal solutions were flagged out — it will not apply a solution where this isn't any, so doesn't change the data from the standard calibration

Self cal — the steps (3)

```
applycal(vis=visname,
     spwmap=[0,0,0,0],
     spw='',
     field='',
     gaintable=['pcal1'],
     gainfield='',
     calwt=F,
     flagbackup=T,
     applymode='calonly')  # or s
```

OPTIONAL:
can add an intermediate split step to copy out the data with the calibration applied

keep track of correct ‘vis’ files

- For long observations, split after a ‘inf, 30 min’ self-calibration may help before iterative self-calibration

- Using applycal -
  - thi
  - spwmap - maps the solutions created in SPW 0 as a result of the combine=‘spw’ choice to spw = 0,1,2,3
  - applymode - if all solutions ‘look’ reasonable - use ‘calonly’ this avoids adding extra flagging to your source data if some self cal solutions were flagged out — it will not apply a solution where this isn't any, so doesn't change the data from the standard calibration

Self cal — repeat step (1)

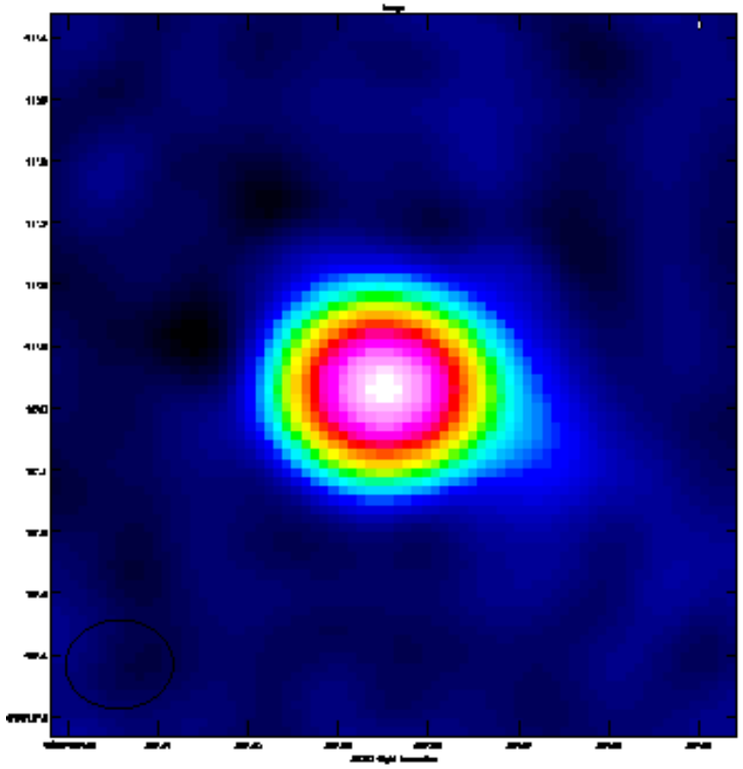
- Re-run an interactive clean
- Clean will *always* use the ‘corrected’ data column

```
os.system('rm -rf '+souname+'.B6.cont_pcal1*')
clean(vis=visname,
      spw = spwcont,
      imagename = souname+'.B6.cont_pcal1',
      field='0',
      cell=cell,
      imsize=imagesize,
      outframe='LSRK',
      niter=10000,
      interactive=True,
      threshold='0.05mJy',
      pbcor=False,
      weighting='briggs',
      robust=0.5,
      mode = 'mfs')
```

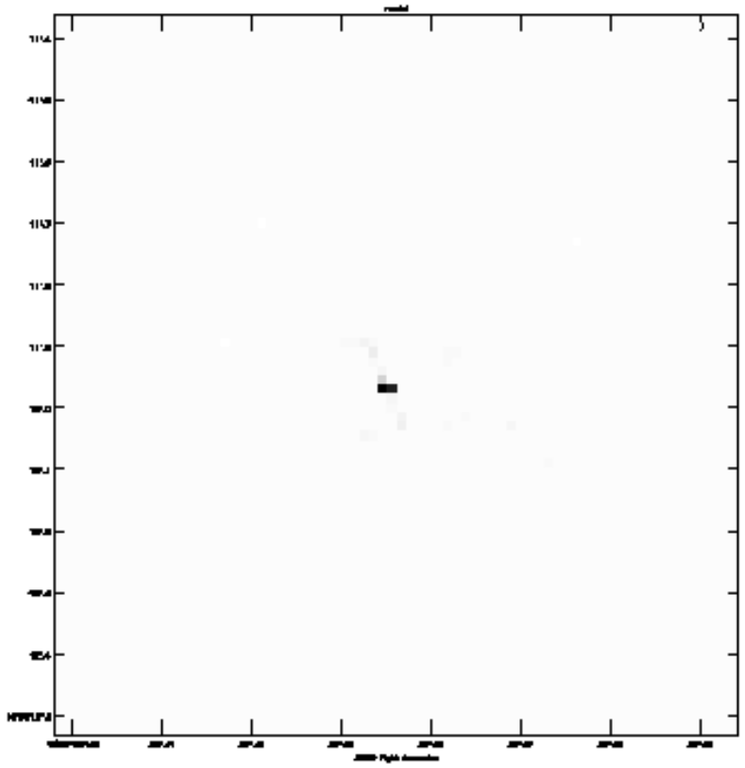
remember some CASA version need - usescratch = True

Self cal — repeat step (1)

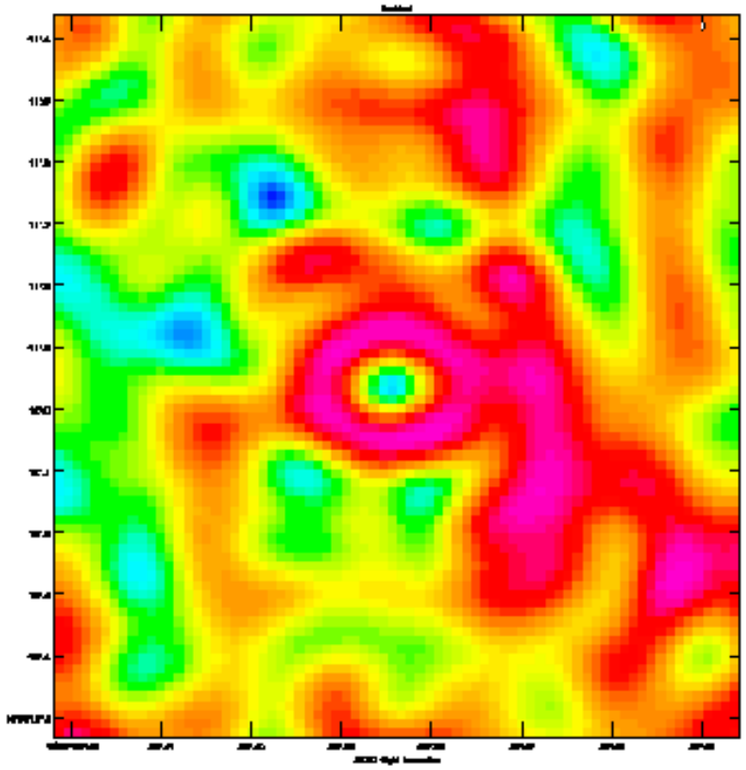
IMAGE



MODEL



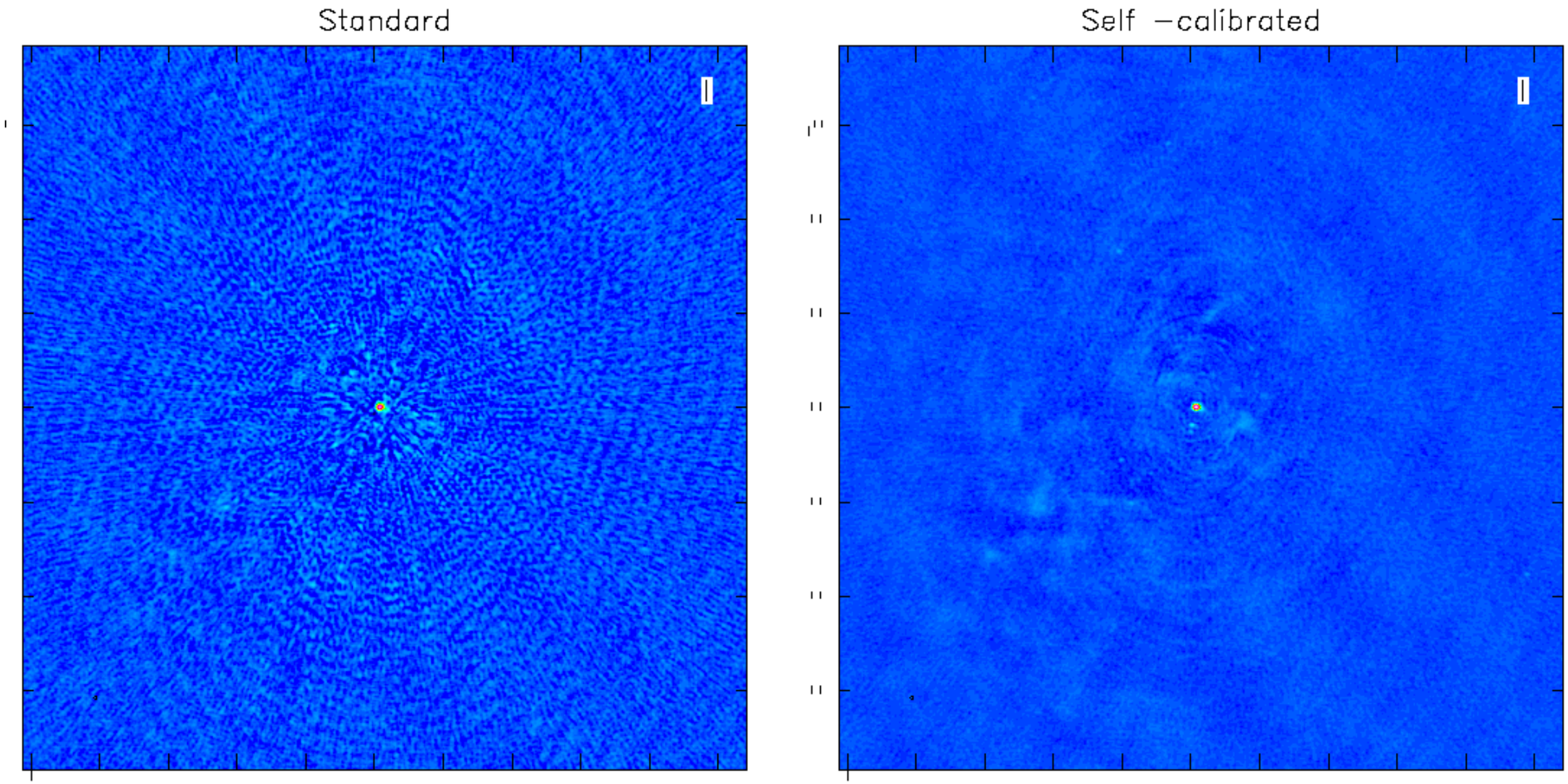
RESIDUAL



- **S/N ~ 850**
- **Compact model**
- **Reasonably deep**

- **Large improvement in this data**
  - **track share - multiple sources between gain cal scans**
  - **relatively large distance between calibrator and target (7 degrees)**

Self cal — repeat step (1)



S/N ~ 120

Pk ~ 50 mJy/bm

S/N ~ 850

Pk ~ 70 mJy/bm

Self cal — repeat step (2)

```
# now push to 30 sec intervals
os.system('rm -rf pcal2')
gaincal(vis=visname,
        caltable='pcal2',
        gaintype='T',
        refant='DA59',
        calmode='p',
        combine='spw',
        spw=spwcont,
        field='',
        solint='30s',
        minsnr=3.0,
        minblperant=4)
```

- **Using gaincal - solve the phases of the selected SPW to match the image model - CASA ‘knows’ the model**

  caltable - the 2nd calibration table (pcal2)

  refant - use same as calibration if possible

  calmode - p - phaseonly

  combine - ‘spw’ combine all inputs in spw into one

  spw= spwcont i.e. the previously selected continuum ONLY range

  solint - ‘30s’- use shorter value than before

  - don’t make too large a jump

Self cal — repeat step (2)

```
# now push to 30 sec intervals
os.system('rm -rf pcal2')
gaincal(vis=visname,
        caltable='pcal2',
        gaintype='T',
        refant='DA59',
        calmode='p',
        combine='spw',
        spw=spwcont,
        field='',
        solint='30s',
        minsnr=3.0,
        minblperant=4)
```

Note - the data has not been ‘split’
‘data’ column is the ‘original’ calibrated data
‘corrected’ column is one with pcal1 applied

gaincal always works on the ‘data’ to make a new gain table - as we re-imaged the new, more accurate image model is used

- Using gaincal - solve the phases of the selected SPW to match the image model - CASA ‘knows’ the model

  caltable - the 2nd calibration table (pcal2)

  refant - use same as calibration if possible

  calmode - p - phaseonly

  combine - ‘spw’ comtine all inputs in spw into one

  spw= spwcont i.e. the previously selected continuum ONLY range

  solint - ‘30s’- use shorter value than before

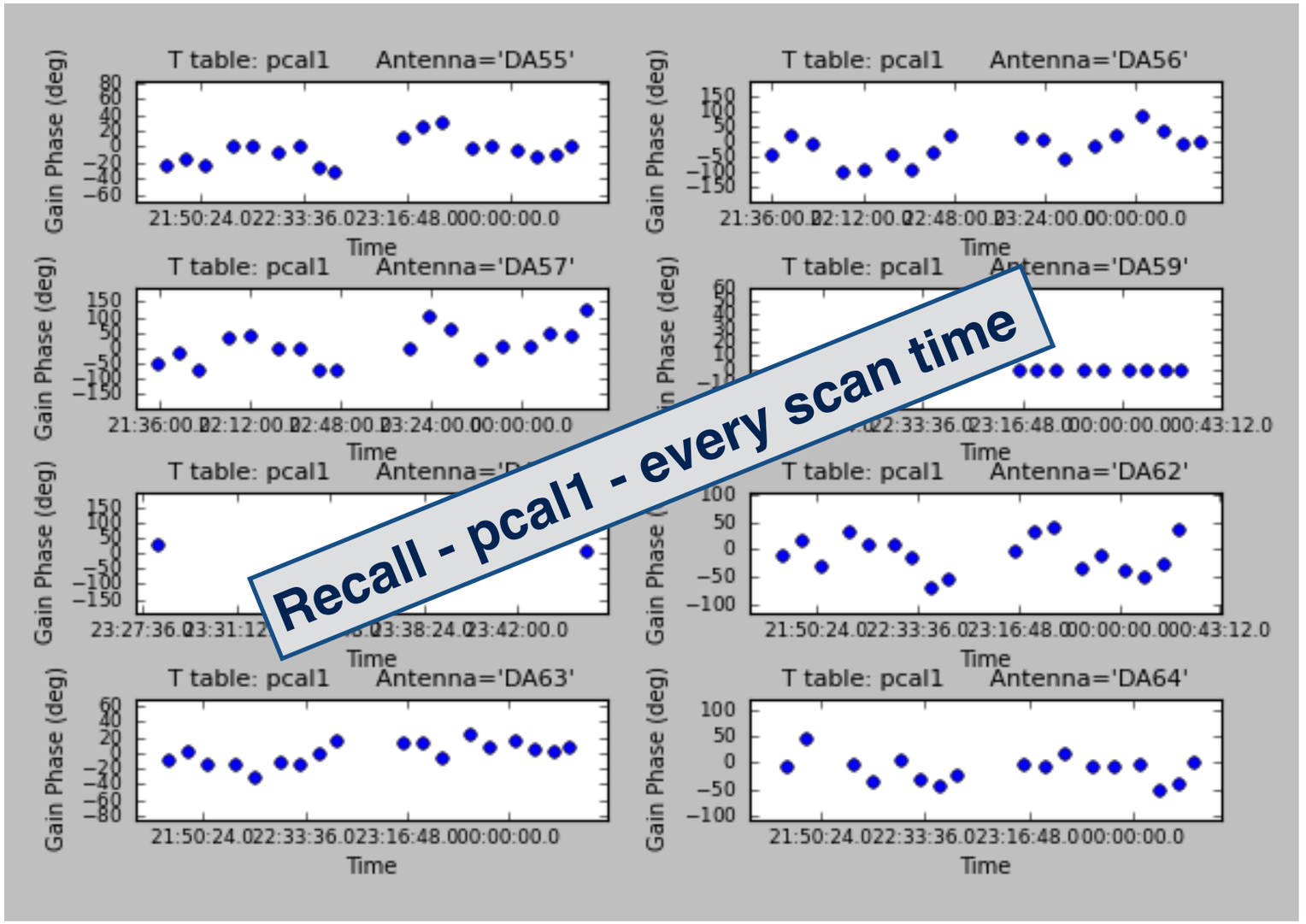
  - don’t make too large a jump

Self cal — repeat step (2)

```
plotcal(caltable='pcal2',
        xaxis='time',
        yaxis='phase',
        spw='',iteration='antenna',
        subplot=421,plotrange=[0,0,-180,180])
```

caltable = ‘pcal2’

- **Using plotcal - make the plots of the solutions**



OPTIONAL:

showgui - default - True, False

means no screen pop-up

figfile - string - set to

produce a png, e.g.

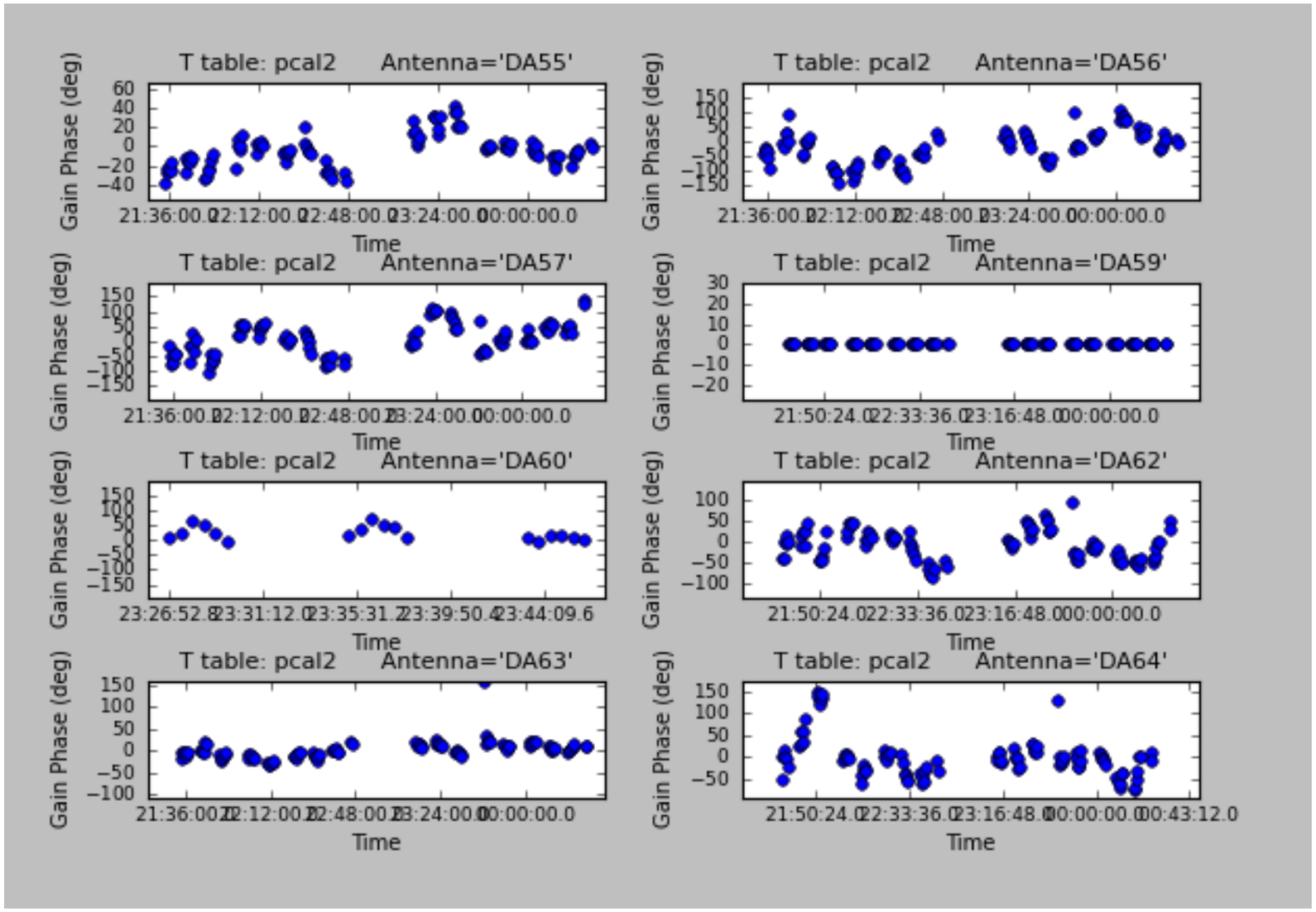
‘pcal2.plots’

Self cal — repeat step (2)

```
plotcal(caltable='pcal2',
        xaxis='time',
        yaxis='phase',
        spw='',iteration='antenna',
        subplot=421,plotrange=[0,0,-180,180])
```

caltable = ‘pcal2’

- **Using plotcal - make the plots of the solutions**



OPTIONAL:

showgui - default - True, False

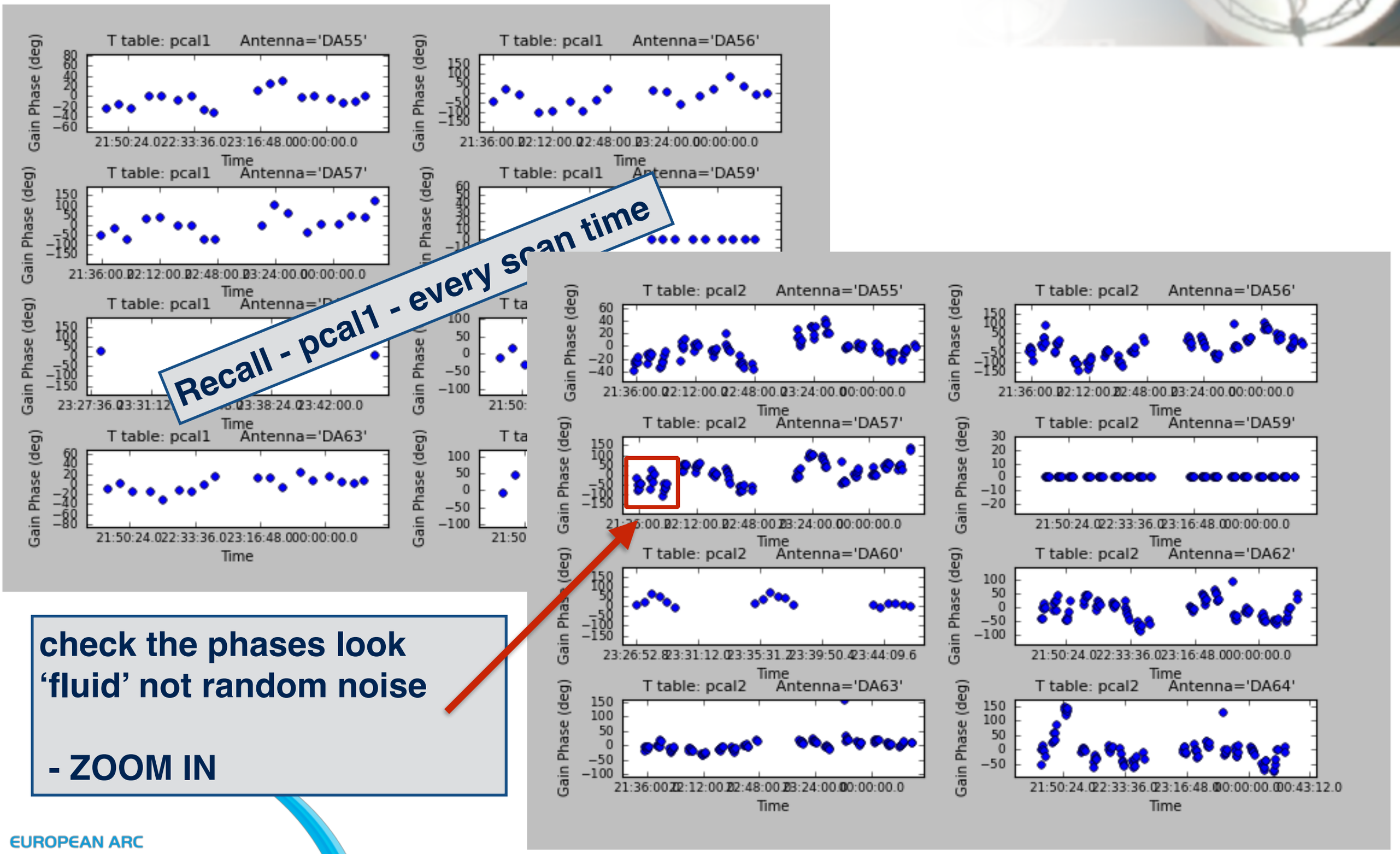
means no screen pop-up

figfile - string - set to

produce a png, e.g.

‘pcal2.plots’

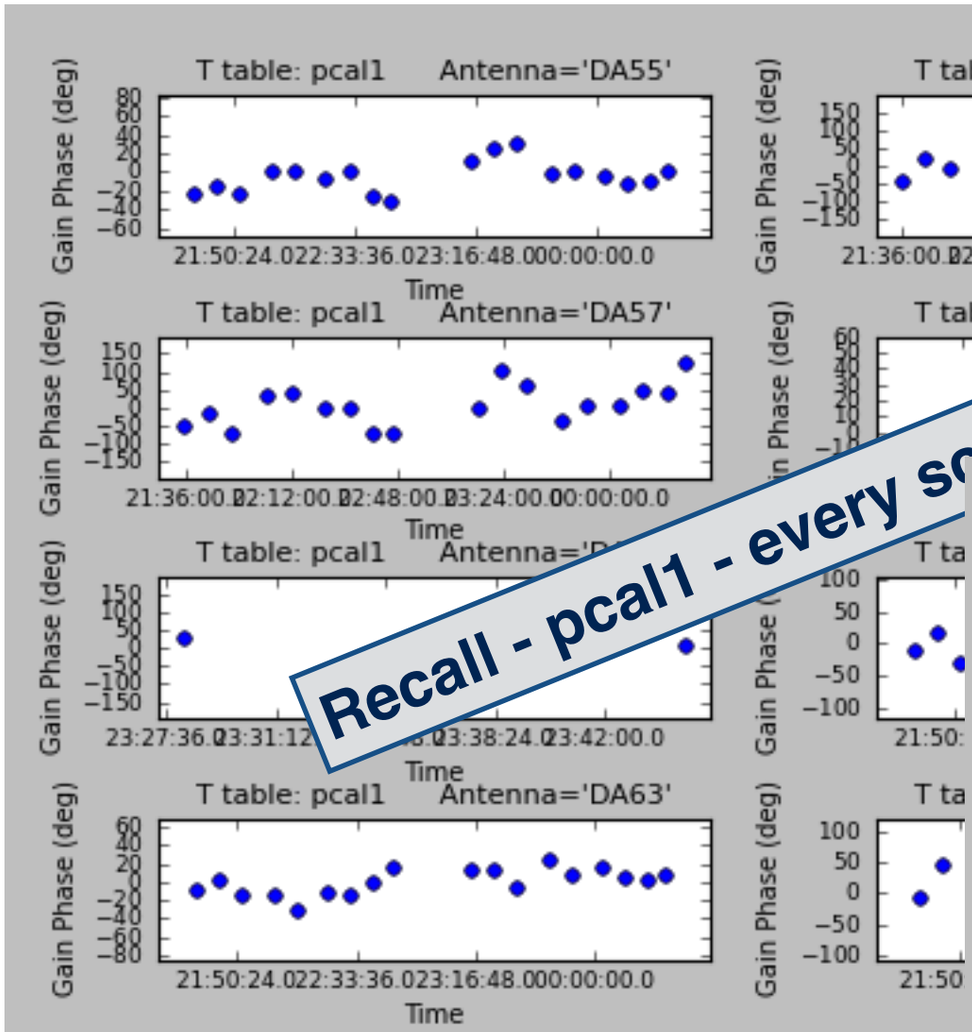
Self cal — repeat step (2)

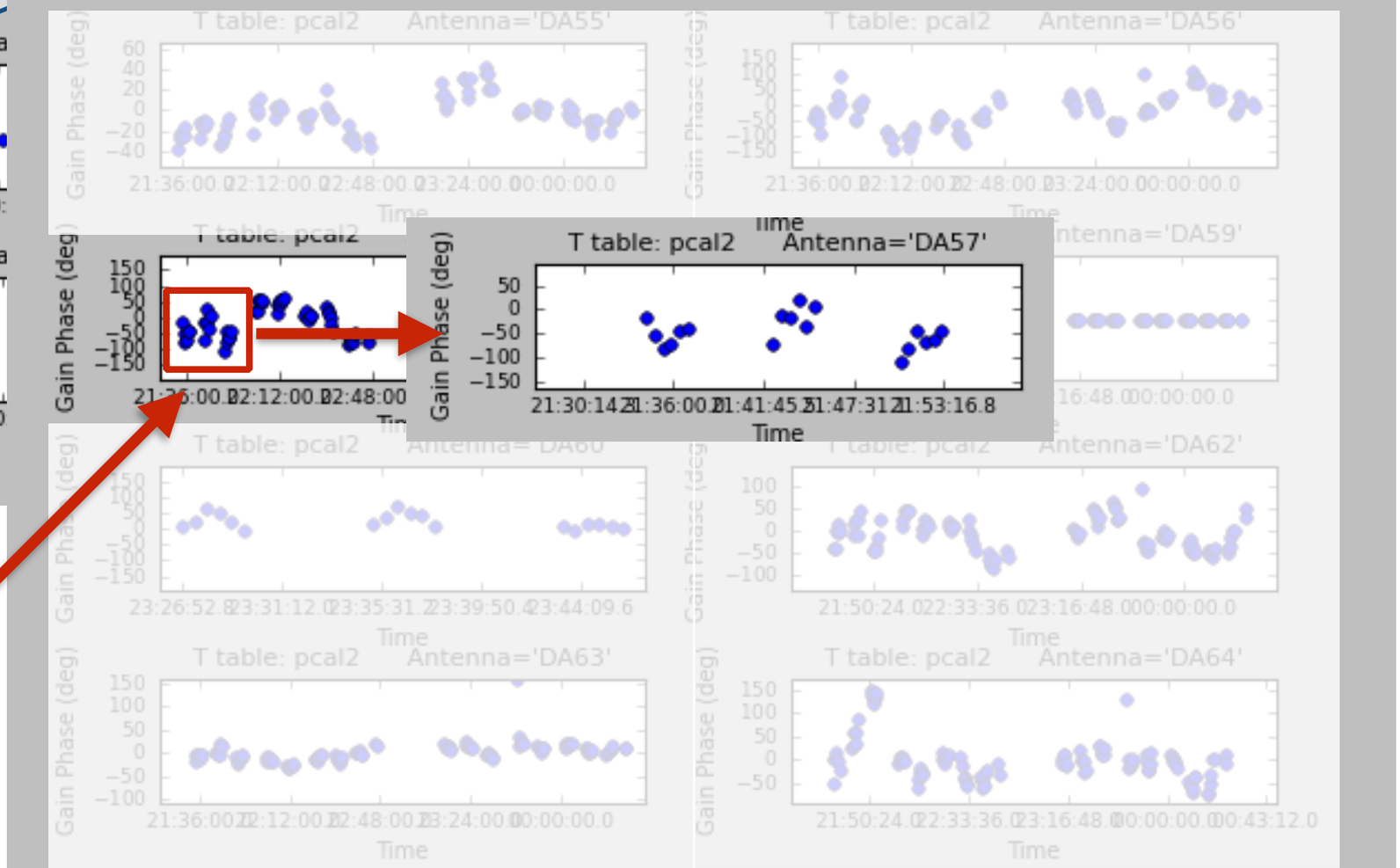


check the phases look ‘fluid’ not random noise

- ZOOM IN

Self cal — repeat step (2)





check the phases look ‘fluid’ not random noise

- ZOOM IN interactively

Self cal — repeat …….

```
applycal(vis=visname,
         spwmap=spwmap,
         spw='',
         field='',
         gaintable=['pcal2'],
         gainfield='',
         calwt=F,
         flagbackup=T,
         applymode='calonly')
```

applycal - will overwrite the old ‘corrected’ data applying now ‘pcal2’

```
os.system('rm -rf '+souname+'.B6.cont_pcal2*')
clean(vis=visname,
      spw = spwcont,
      imagename = souname+'.B6.cont_pcal2',
      field='0',
      cell=cell,
      imsize=imagesize,
      outframe='LSRK',
      niter=10000,
      interactive=True,
      threshold='0.05mJy',
      pbcor=False,
      weighting='briggs',
      robust=0.5,
      mode = 'mfs')
```

clean - will use the new ‘corrected’ data with best solutions for the image

New image S/N ~1100

Self cal — repeat …….

```
# now to int - 6 sec
os.system('rm -rf pcal3')
gaincal(vis=visname,
        caltable='pcal3',
        gaintype='T',
        refant='DA59',
        calmode='p',
        combine='spw',
        spw=spwcont,
        field='0',
        solint='int',
        minsnr=3.0,
        minblperant=4)
```

integration time - typically 6 seconds

- Using gaincal - solve the phases of the selected SPW to match the image model - CASA ‘knows’ to use the latest

  caltable - the 3rd phase calibration table (pcal3)

  refant - use same as calibration if possible

  calmode - p - phaseonly

  combine - ‘spw’ combine all inputs in spw into one

  spw= spwcont i.e. the previously selected continuum ONLY range

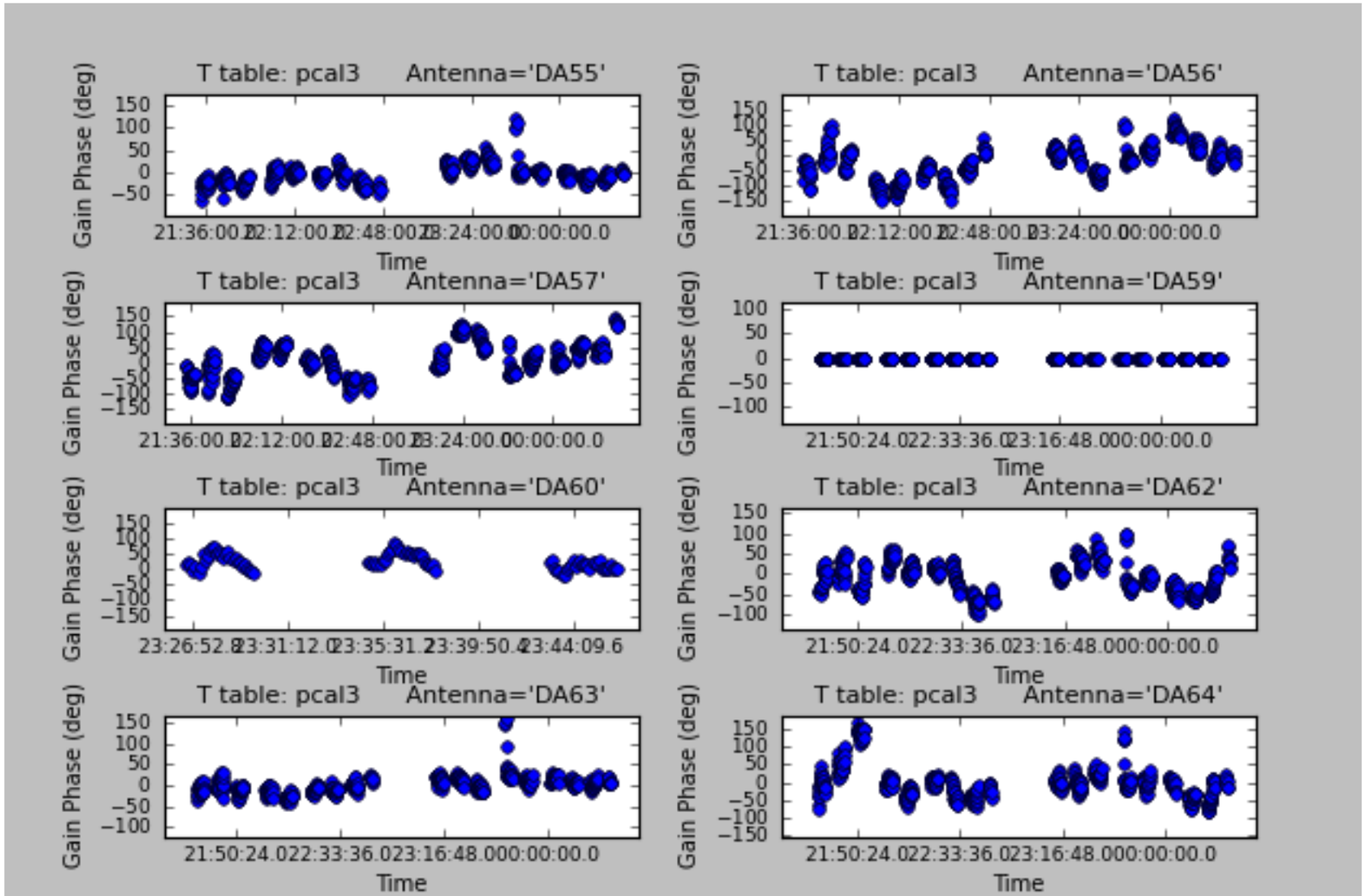
  solint - ‘int’ - integration time - each recorded data value

Self cal — repeat …….

```
plotcal(caltable='pcal3',
        xaxis='time',
        yaxis='phase',
        spw='',
        iteration='antenna',
        subplot=421,plotrange=[0,0,-180,180])
```

caltable = ‘pcal3’

- **Using plotcal - make the plots of the solutions**



OPTIONAL:

showgui - default - True, False

means no screen pop-up

figfile - string - set to

produce a png, e.g.

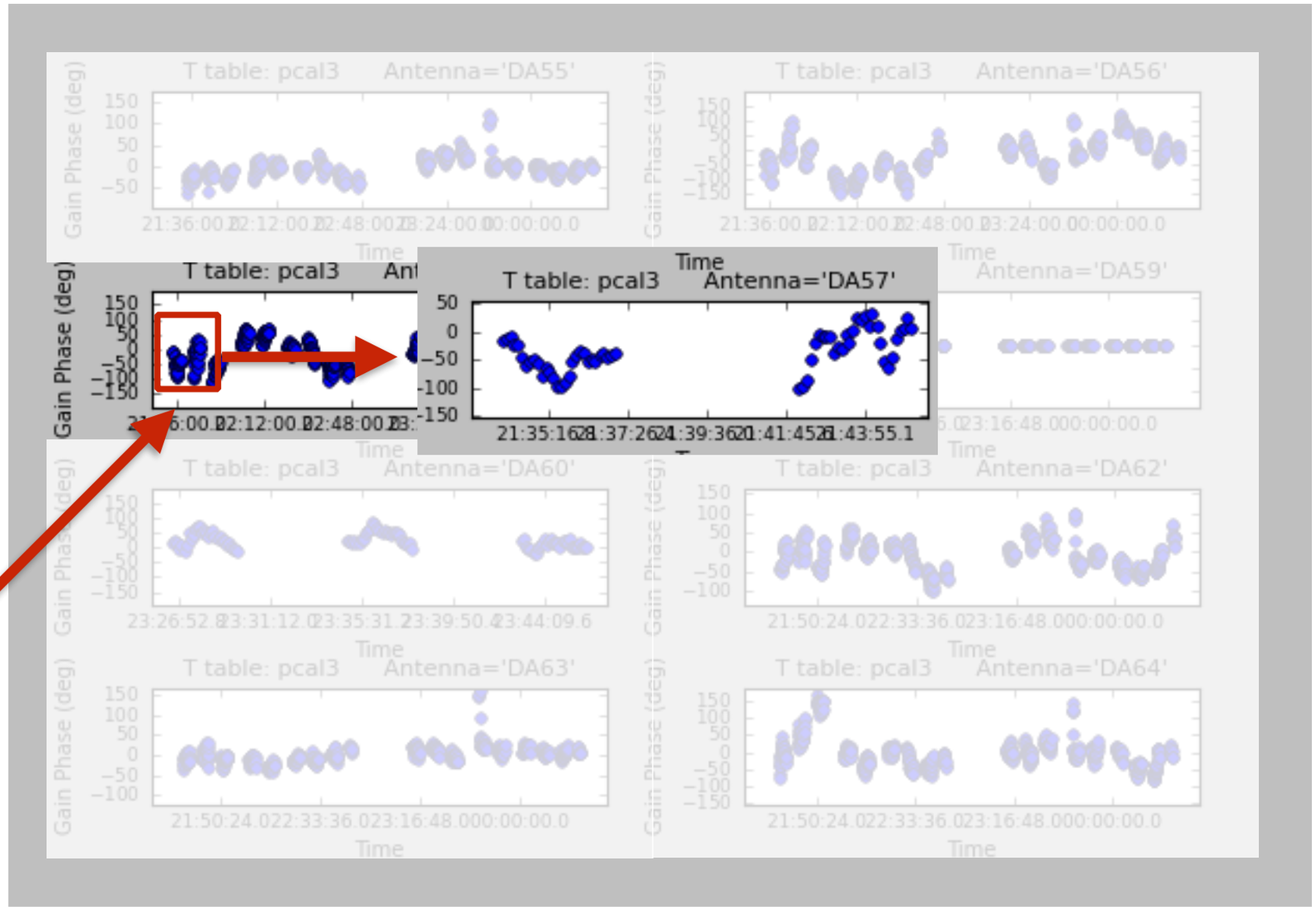
‘pcal3.plots’

Self cal — repeat …….

```
plotcal(caltable='pcal3',
        xaxis='time',
        yaxis='phase',
        spw='',
        iteration='antenna',
        subplot=421,plotrange=[0,0,-180,180])
```

caltable = ‘pcal3’

- **Using plotcal - make the plots of the solutions**



OPTIONAL:

showgui - default - True, False

means no screen pop-up

figfile - string - set to

produce a png, e.g.

‘pcal3.plots’

check the phases look ‘fluid’ not random noise

- ZOOM IN interactively

Self cal — repeat …….

applycal - will overwrite the old ‘corrected’ data applying now ‘pcal3’

clean - will use the new ‘corrected’ data with best solutions for the image

New image S/N ~1150 - smaller gain as solutions are only solving for very small phase fluctuation now

Self cal — final step - amp

```
## apcal here - applying the best phase cal
os.system('rm -rf apcal')
gaincal(vis=visname,caltable='apcal',
        gaintype='T',
        refant='DA59',
        calmode='ap',
        combine='spw',
        spw=spwcont,field='',
        solint='inf',
        minsnr=3.0,
        minblperant=4,
        gaintable=['pcal3'],
        spwmap=[0,0,0,0])
```

caltable = ‘apcal’

gaintable = ‘pcal3’ - apply BEST phase solutions

spwmap =[0,0,0,0] - as applying the phase solns. OTF

- **Using gaincal - solve the amplitude (and phases) of the selected SPW to match the image model**

  caltable - the final calibration table (apcal)

  refant - use same as calibration if possible

  calmode - ap - amp (and phase)

  combine - ‘spw’ combine all inputs in spw into one

  spw= spwcont i.e. the previously selected continuum ONLY range

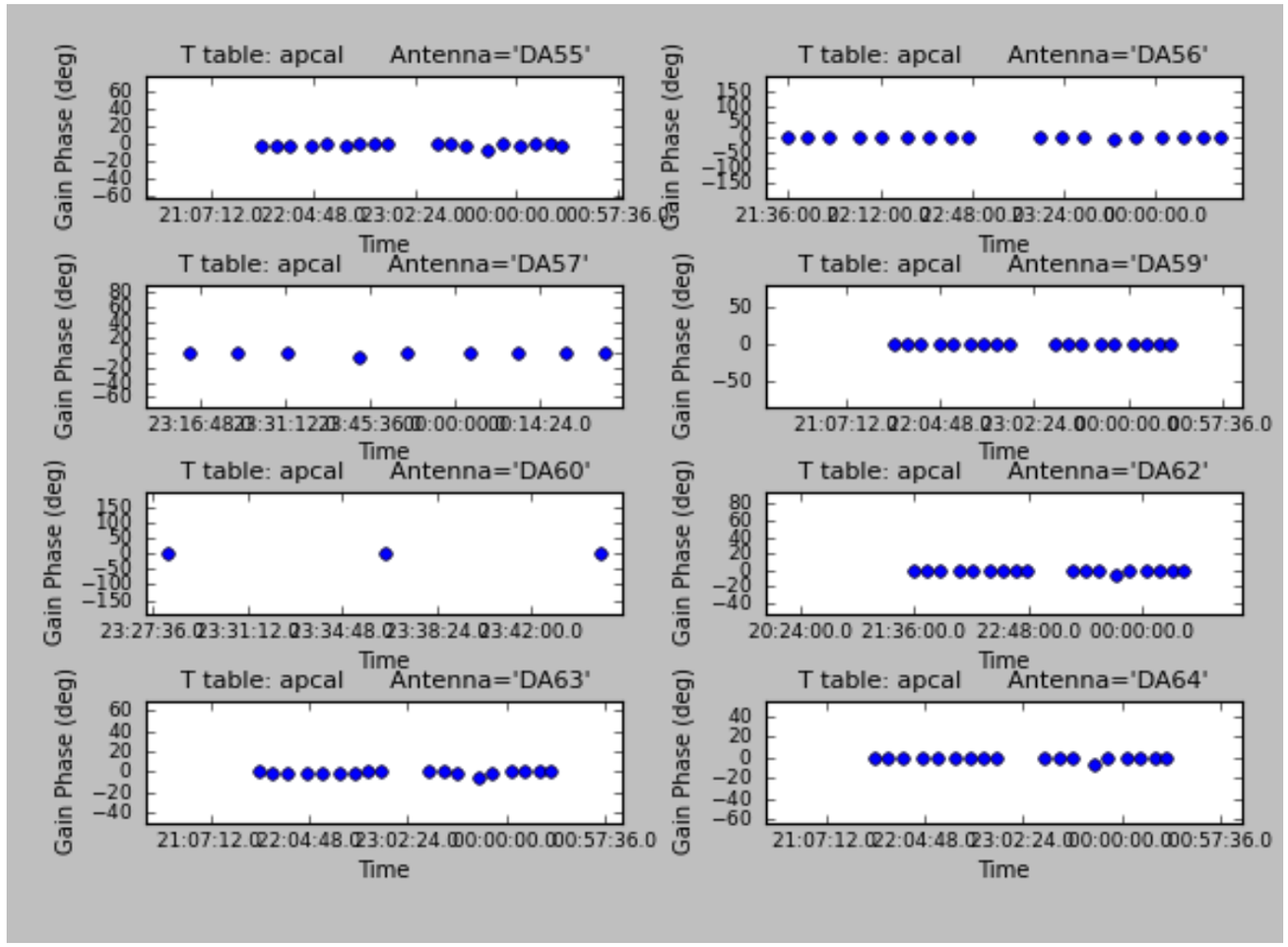
  solint - ‘inf’- infinite scan time - amplitude variations are slow in time

Self cal — final step - amp

```
plotcal(caltable='apcal',
        xaxis='time',
        yaxis='phase',
        spw='',
        iteration='antenna',
        subplot=421,plotrange=[0,0,-180,180])
```

yaxis = ‘phase’

- **Using plotcal - make the plots of the PHASE solutions**
  these should be residuals as the phase calibrations
  were applied during the gaincal solve step

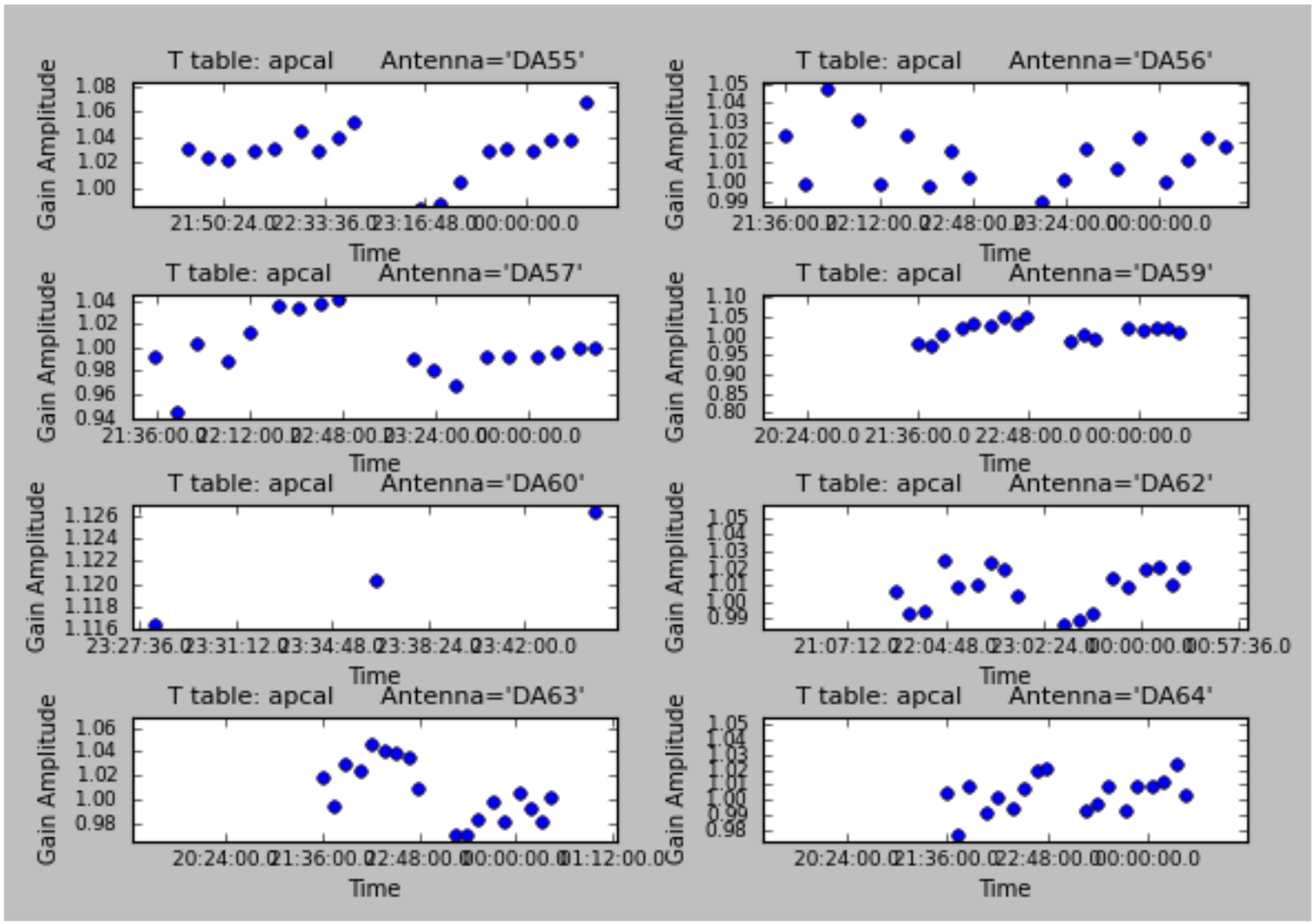


Self cal — final step - amp

```
plotcal(caltable='apcal',
        xaxis='time',
        yaxis='phase',
        spw='',
        iteration='antenna',
        subplot=421,plotrange=[0,0,-180,180])
```

yaxis = 'amp'

- **Using plotcal - make the plots of the AMP solutions**
  gains should not be wildly scattered - slightly offset from 1



Self cal — final step - amp

```
## apply apcal and int - phae cal
applycal(vis=visname,
         spwmap=[[0,0,0,0],[0,0,0,0]],
         spw='',field='0',
         gaintable=['pcal3','apcal'],
         gainfield='',
         calwt=F,
         flagbackup=T,
         applymode='calonly')
```

spwmap = [[0,0,0,0],[0,0,0,0]
for each gaintable

gaintable =[‘pcal3’,‘apcal’]

```
split(vis=visname, outputvis='Final_selfcal.ms')
```

split the data

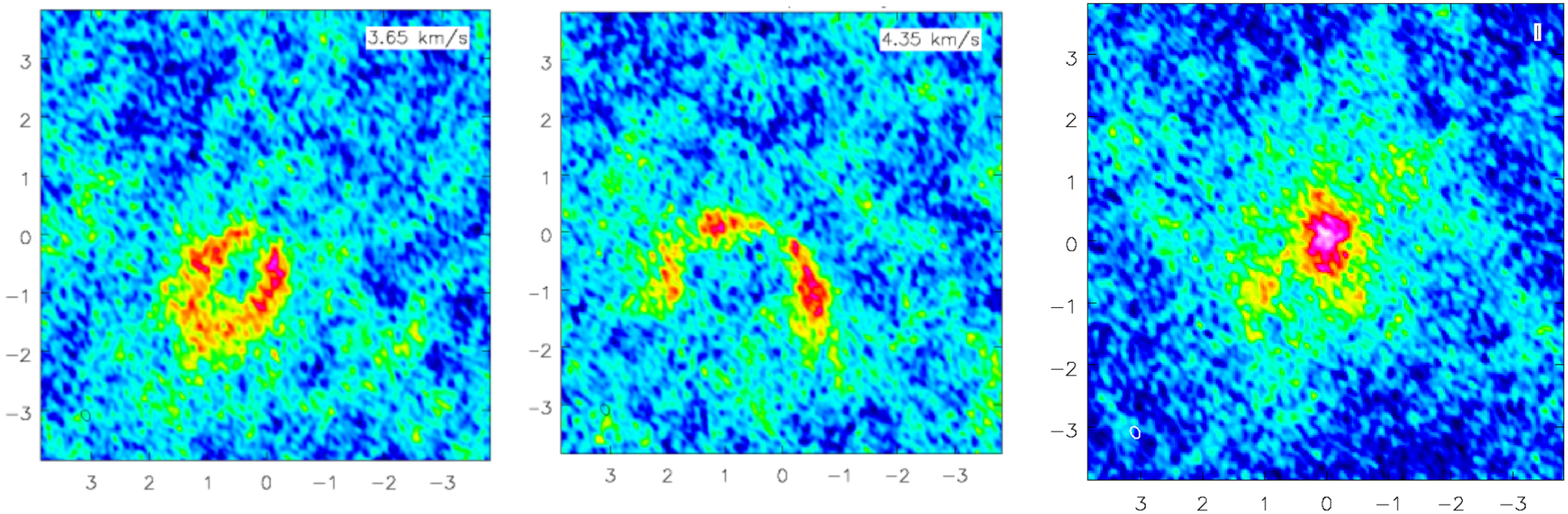
```
os.system('rm -rf '+souname+'.B6.cont_scfinal*')
clean(vis='Final_selfcal.ms',
      spw = spwcont,
      imagename = souname+'.B6.cont_scfinal',
      field='0',    cell=cell,
      imsize=imagesize,    outframe='LSRK',
      niter=10000,      interactive=True,
      threshold='0.05mJy',    pbcor=False,
      weighting='briggs',    robust=0.5,    mode = 'mfs')
```

clean -final clean of the continuum

Self cal — Lines

Self-calibration on molecular lines:

- **Works if the molecular line is strong (S/N > 100)**
- **Apply the continuum self-calibration tables on the molecular line windows - need to include edges of the window**



Self cal — Multiple datasets

Self-calibration across data sets:

- **Multiple configurations, combining with archival data, 7m array data, etc.**
- **Self-calibrated on each data down to ‘int’ (or shortest time domain possible) before combination**
- **Apply 1 or 2 phase self-calibration and amplitude after combination**

Self cal — final remarks

- Best results for longer baselines and/or higher frequencies
  - phase fluctuations increase with baseline and frequency
- Should have good improvement if: long cycle times and distant phase calibrators - i.e. interpolation not ideal
- Can opt to split out at after each apply cal step - easier to see phase residuals in each step
- Take care to ONLY select channels (i.e. continuum) which actually produced the image - self cal can go VERY wrong if you include line channels, or try using a composite bandwidth

Self cal — final remarks

If you think the data quality can be increased:
Ask ALLEGRO:
alma@strw.leidenuniv.nl