Reducing ALMA data in CASA

Eric Villard (ALMA JAO)

Python and tasks

- CASA is built on Python environment
  - Scripts can be run with execfile()
  - System commands can be run with a !
- The task is the high-level interface to CASA
  - A task is a Python script, with inputs and outputs
  - *tasklist*, to obtain a list of all tasks
  - *help taskname*, to obtain help on a task
  - *inp taskname*, to list the input parameters of a task
  - *tget taskname*, to load the input parameters of a task at the previous successful execution
  - *go* to run active task

The four data columns

- DATA, CORRECTED, MODEL, RESIDUAL
  - Same dimensions
  - DATA contains the raw visibilities
  - CORRECTED contains the calibrated visibilities
  - MODEL contains the model visibilities
  - RESIDUAL contains the difference between the calibrated and model visibilities

applycal

- CASA is never modifying the raw visibilities (the DATA column)
- applycal takes the DATA column, the calibration table, and puts the result in the CORRECTED column.
- This strategy means that one cannot apply calibration tables subsequently. All calibration tables must be applied in one go.

gaincal and setjy

- gaincal is used to calibrate out the time variations of phase and amplitude.
- In case of a resolved calibrator (e.g. Titan for flux calibration), we don't want to calibrate out phase/ amplitude variations coming from the extension of the object.
- The strategy is to have model visibilities in the MODEL column, and gaincal divides DATA by MODEL. (In other words: gaincal assumes a 1Jy point source.)
- setjy is used to fill in the MODEL column.
  - It has a model for most planets and moons.
  - It can take an image.

Main steps of ALMA data calibration

- Conversion ASDM to MS
- Tsys and WVR corrections, applycal
- Split of channelized spectral windows (not required)
- A priori flagging (autocorr, shadow, pointing and atm cal scans)
- setjy, if using any resolved calibrators
- Initial phase solution, bandpass
- Complete phase solution
- Complete amplitude solution
- Flux scale
- Clean

Example:

NGC3256 (SV target), band 3

March 9-17, 2011

Tsys solution

taskname = 'gencal’

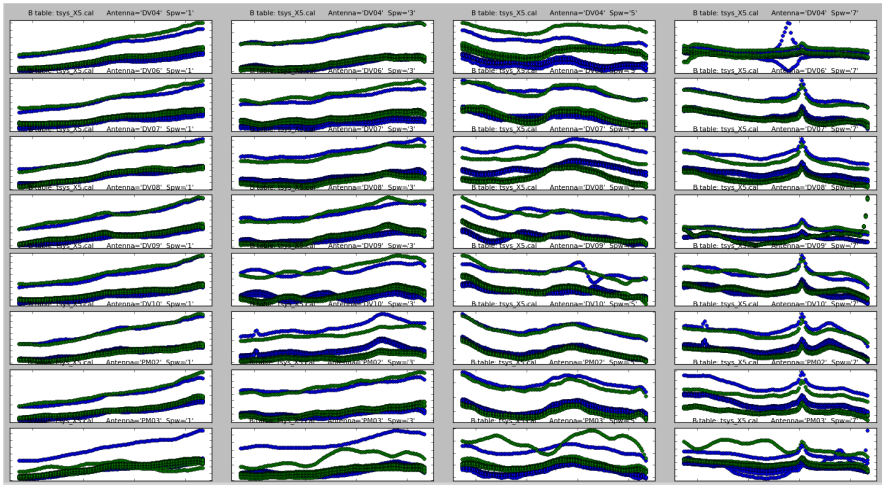
default(taskname)

vis = 'uid___A002_X1d5a20_X5.ms’

caltype = 'tsys’

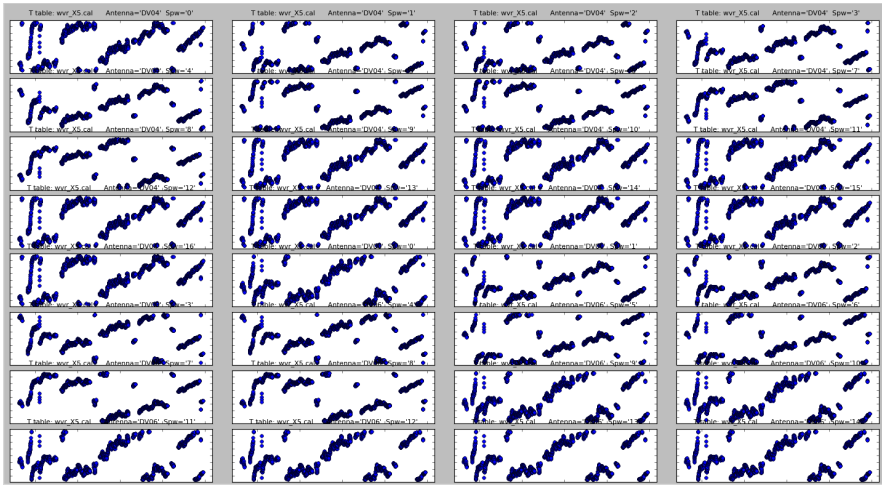
caltable = 'tsys_X5.cal’

gencal()



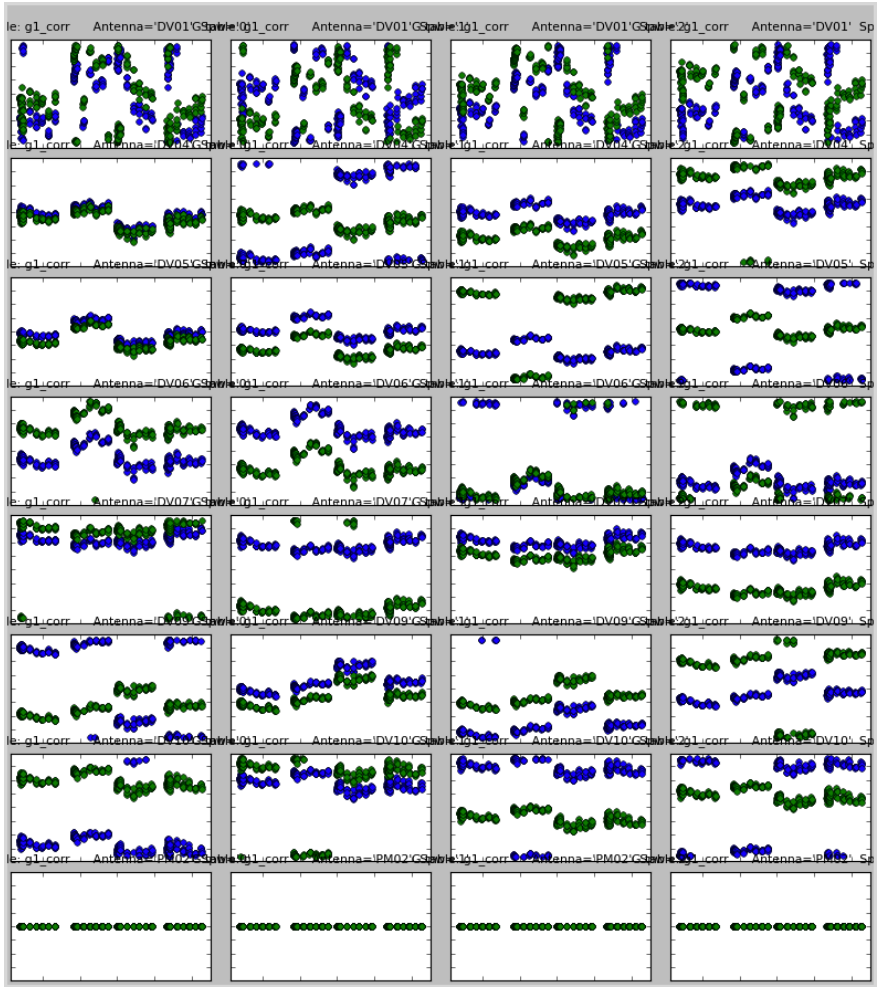
WVR solution

```
os.system('wvrgcal --ms
    uid___A002_X1d5a20_X5.ms --
    output wvr_X5.cal --toffset -1 --
    segfield')
```



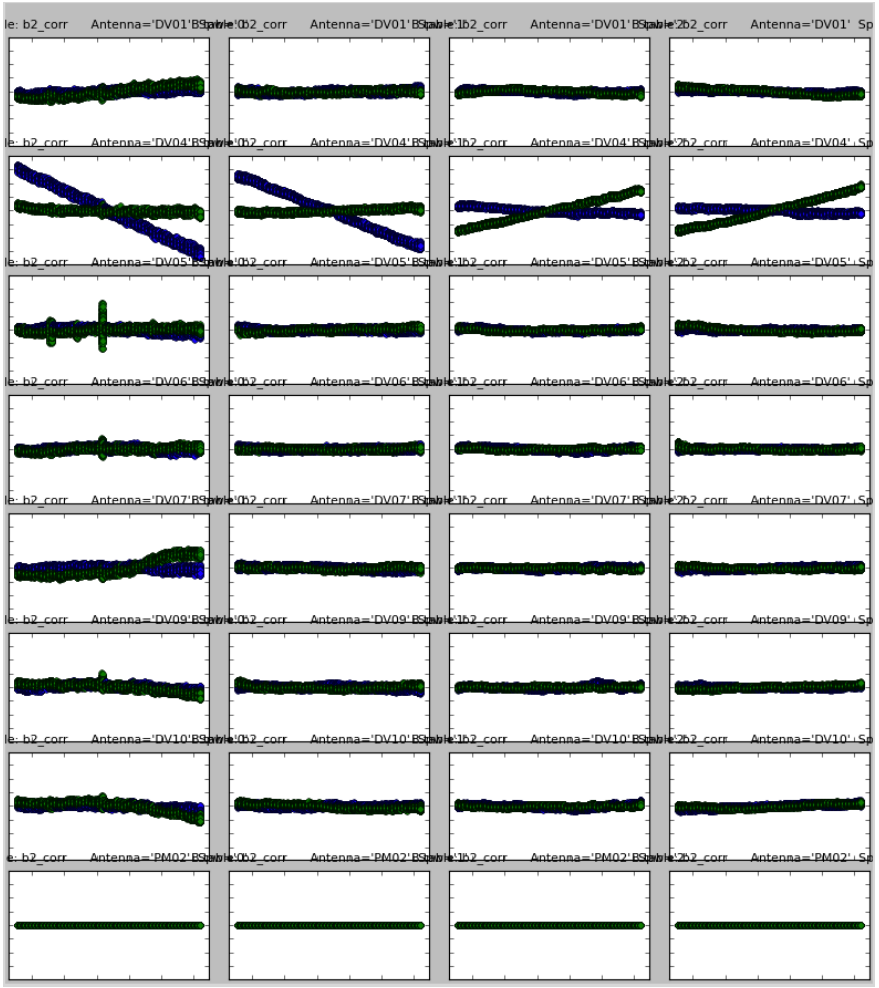
Bandpass solution (1)

```
taskname = 'gaincal'
default(taskname)
vis = 'ngc3256.ms'
caltable = 'g1_corr'
spw = '*:50~80'
field = '0'
solint = 'int'
refant = 'PM02'
calmode = 'p'
gaincal()
```



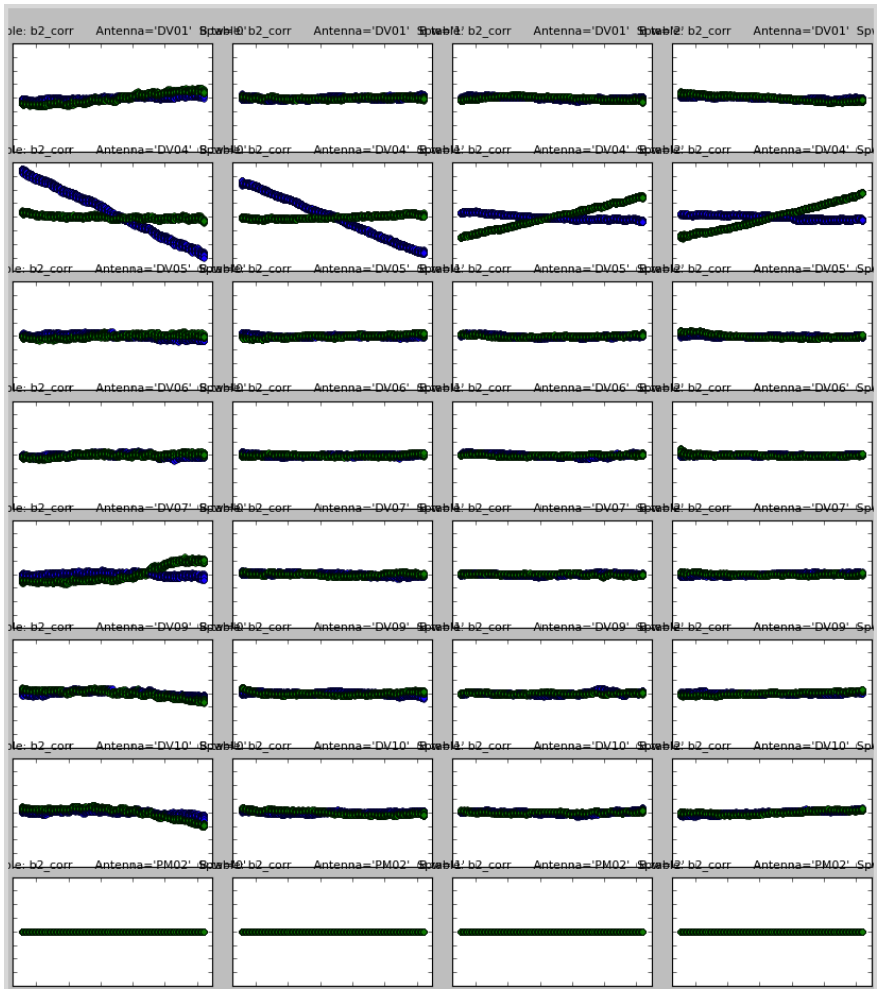
Bandpass solution (2)

```
taskname = 'bandpass'
default(taskname)
vis = 'ngc3256.ms'
caltable = 'b2_corr'
field = '0'
refant = 'PM02'
solint = 'int'
solnorm = T
gaintable = 'g1_corr'
bandpass()
```



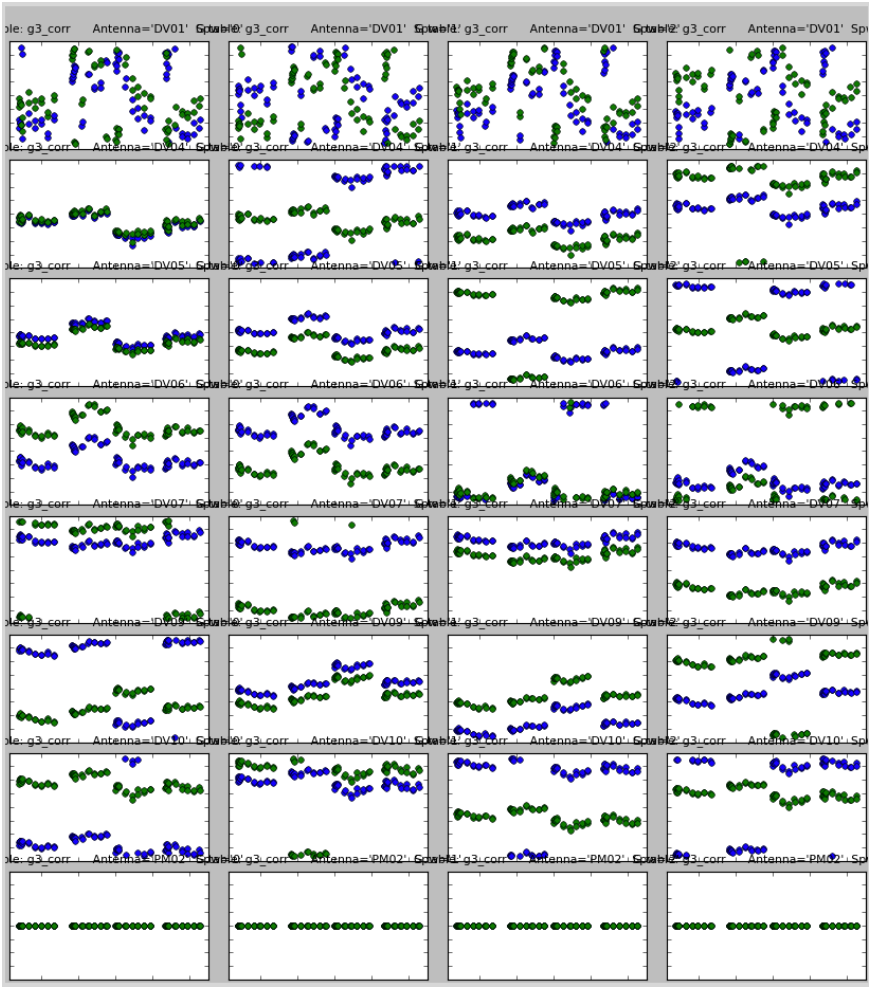
Bandpass solution (3)

```
taskname = 'flagdata’
default(taskname)
vis = 'ngc3256.ms’
flagbackup = F
spw = ['0:31~32;47~48;63~64']
field = '0’
correlation = 'YY’
flagdata()
```



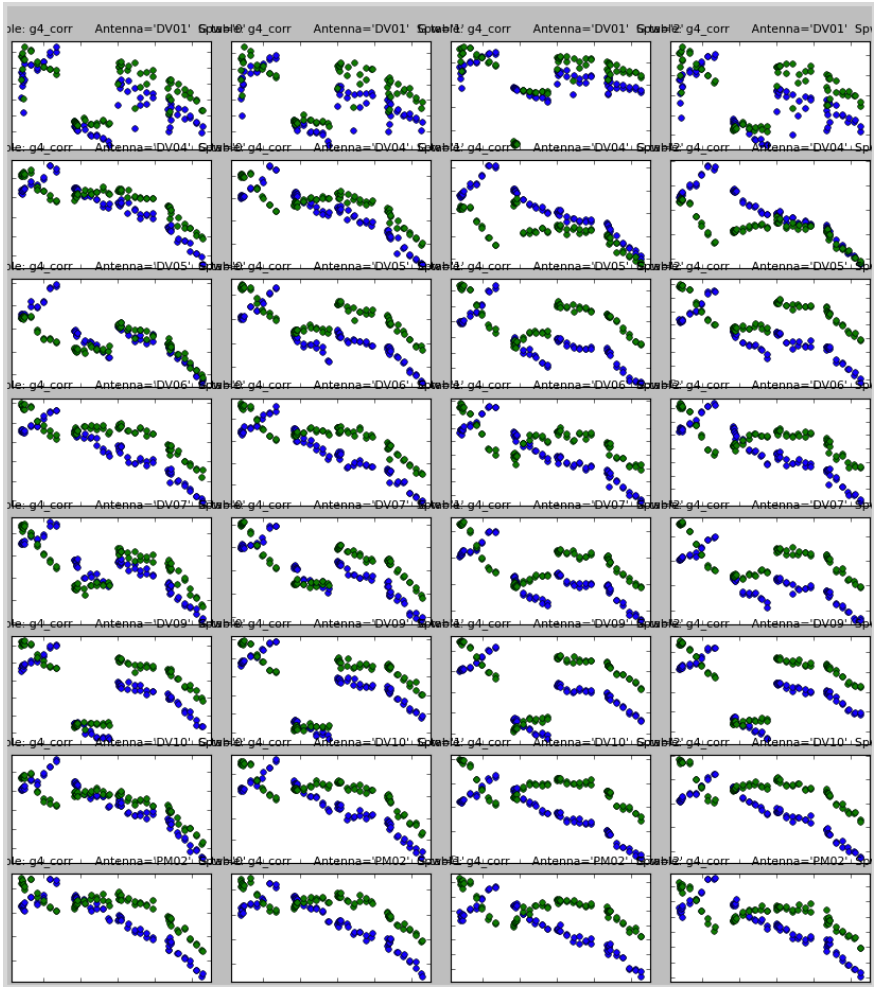
Complete phase solution

```
taskname = 'gaincal'
default(taskname)
vis = 'ngc3256.ms'
caltable = 'g3_corr'
field = '0'
solint = '30s'
refant = 'PM02'
calmode = 'p'
gaintable = 'b2_corr'
gaincal()
```

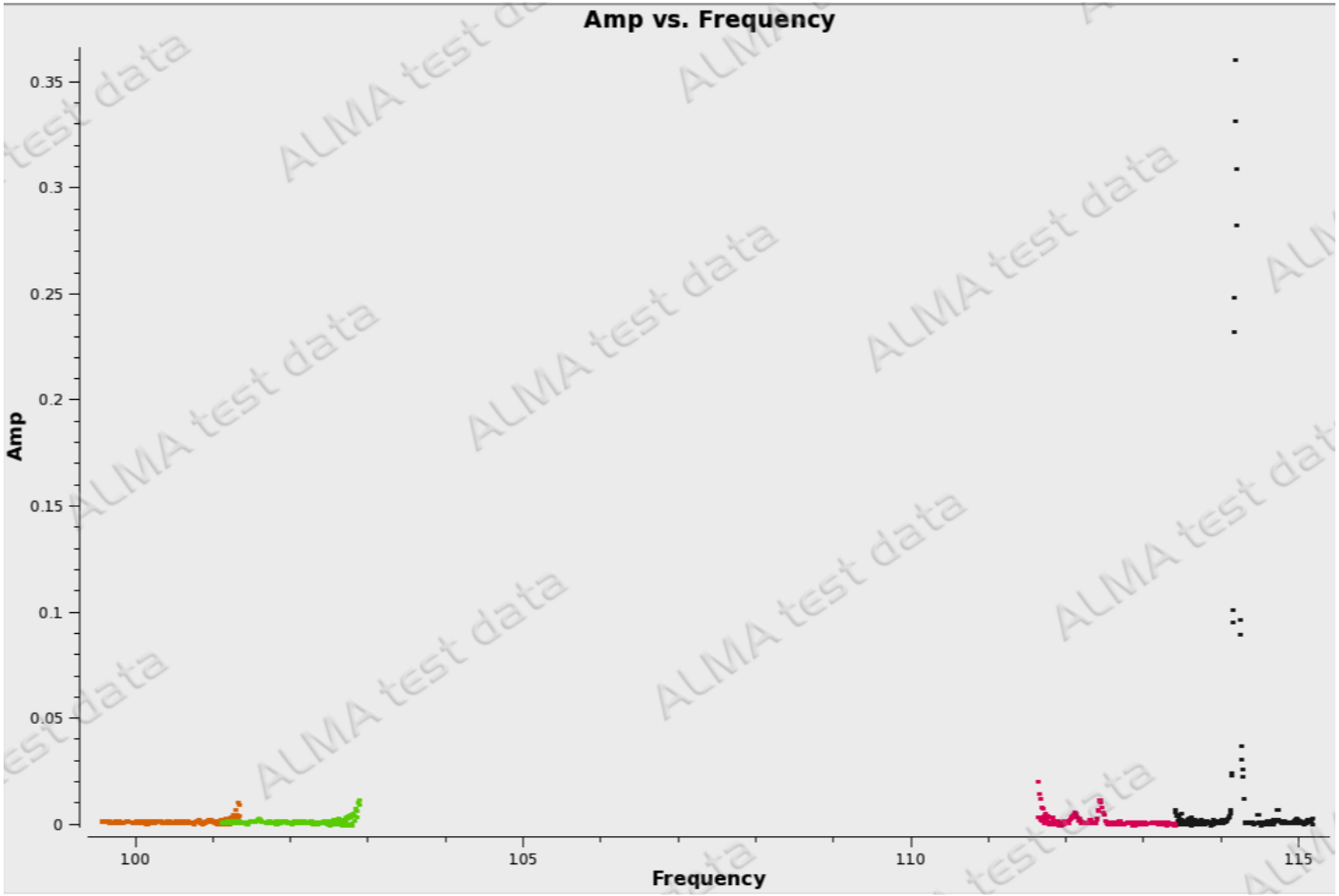


Complete amplitude solution

```
taskname = 'gaincal'
default(taskname)
vis = 'ngc3256.ms'
caltable = 'g4_corr'
field = '0'
solint = '30s'
refant = 'PM02'
calmode = 'ap'
gaintable = ['b2_corr', 'g3_corr']
gaincal()
```

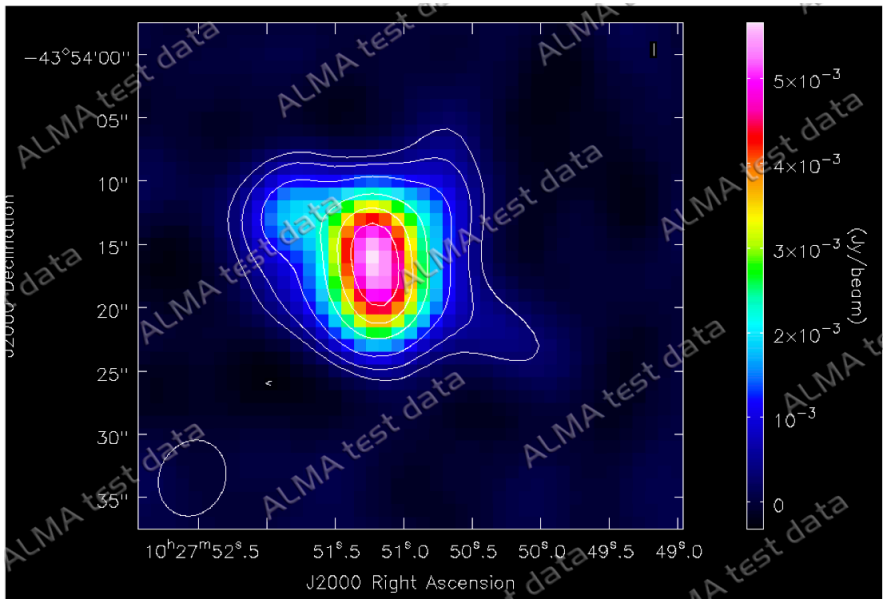


Amp vs freq, after full calibration



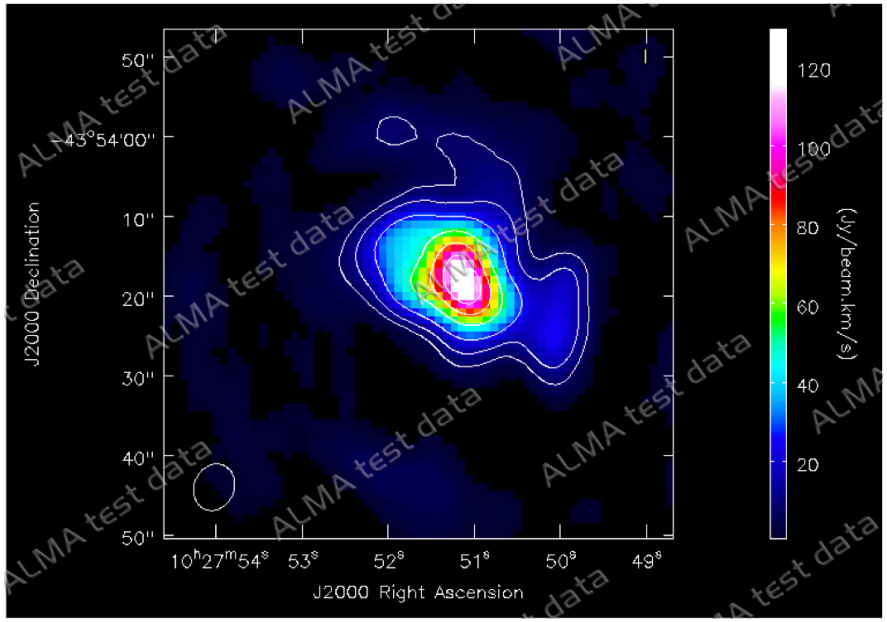
Continuum map (with one round of selfcal)

```
taskname = 'clean'
default(taskname)
vis = 'ngc3256_cont.ms'
imagename = 'ngc3256_cont'
field = '0'
cell = '1arcsec'
imsize = 512
weighting = 'briggs'
interactive = T
niter = 1000
clean()

taskname = 'gaincal'
default(taskname)
vis = 'ngc3256_cont.ms'
caltable = 'g5_corr'
field = '0'
solint = '600s'
refant = 'PM02'
calmode = 'p'
gaincal()
```



CO (1-0) moment 0 map

```
taskname = 'uvcontsub'
default(taskname)
vis = 'ngc3256_specline.ms'
field = '0'
fitspw =
      '0:16~55;71~127,1:16~55;71~127,2:16~127,
      3:16~127'
solint = 'int'
fitorder = 1
fitmode = 'subtract'
uvcontsub()

taskname = 'immoments'
default(taskname)
imagename = 'ngc3256_specline.image'
moments = [0]
includepix = [0.01, 1]
outfile = 'ngc3256_specline_moment0'
immoments()
```



CO (1-0) moment 1 map

```
taskname = 'immoments'
default(taskname)
imagename =
    'ngc3256_specline.image'
moments = [1]
includepix = [0.04, 10]
outfile =
    'ngc3256_specline_moment1'
immoments()
```