Imaging ALMA data with CASA

Gerald Schieven

NRC Herzberg Astronomy & Astrophysics

ALMA Casaguides are available here:

https://casaguides.nrao.edu/index.php/ALMAguides

Imaging ALMA Data

- Very few people will need or want to re-calibrate ALMA data
- Nearly everyone will want to re-image
  - Pipeline images are just enough to verify that the data meet the resolution/rms requirements
  - Many options for re-imaging
- Archive data are available!
  - Hundreds of data sets already in the archive which have never been published, plus one can always do new science with "old" data
- Japanese Virtual Observatory
  - http://jvo.nao.ac.jp/portal/alma/archive.do

JVO: http://jvo.nao.ac.jp/portal/alma/archive.do

JVO Top|Search|VOServices|Subaru|ALMA|Analysis|Bookmark|JVOSpace Login

p00 ver.170419 News | FAQ(J) | Help(J) | Bugs(J) I am a guest

=> Location: Top Page > ALMA > ALMA FITS Archive

ALMA FITS Archive

Using the data for publication

The following statement should be included in the acknowledgment of papers using the ALMA datasets obtained from the JVO portal:

"This paper makes use of the following ALMA data: ADS/JAO.ALMA#<Project code>. ALMA is a partnership of ESO (representing its member states), NSF (USA) and NINS (Japan), together with N (Canada) , NSC and ASIAA (Taiwan), and KASI (Republic of Korea), in cooperation with the Republic of Chile. The Joint ALMA Observatory is operated by ESO, AUI/NRAO and NAOJ."

You can find the project code (e.g. 2011.0.01234.S) on the dataset info page where you download the data.

Please also include the following sentence on the title page as a footnote to the title or in the acknowledgment of the paper.

"[Part of] the data are retrieved from the JVO portal (http://jvo.nao.ac.jp/portal) operated by the NAOJ"

Target Name | Simbad Name | **Project Code** | Coords | Frequency | Desktop Viewer

Numbre of Projects found : 826

#	Project Code	# of Data	Title ?	Last Upate
1	2015.1.01286.S	3	Molecular tracers of hidden AGN: Spatially resolved chemistry in Circinus	2017-04-24
2	2015.1.01232.S	15	Revealing the Shock-Interacting Molecular Gas toward the Magellanic Superbubble 30 Doradus C	2017-04-24
3	2015.1.01425.S	17	Lensing through Cosmic Time II: Mapping the Remaining Fronter Fields	2017-04-24
4	2015.1.00388.S	147	A Spectral Line Snapshot Proposal for ALMA: Characterizing Star Formation Rates and Surface Densities of High-redshift Galaxies	2017-04-24
5	2015.1.01151.S	30	The first glimpse of CO 2-1 in z=1.6 cluster galaxies	2017-04-24
6	2015.1.01572.S	18	AGN Feedback and its Role in Galaxy Evolution: gas and stellar kinematics of radio-loud early-type galaxies	2017-04-20
7	2015.1.01271.S	25	Circumstellar chemistry in carbon stars: How unique is IRC+10216?	2017-04-20
8	2013.1.01091.S	47	Sub-parsec scale structure of quiescent molecular clouds possibly pre-forming clusters	2017-04-20
9	2015.1.01344.S	34	The initial conditions of Galactic Center Super-Cluster precursors: an ALMA survey of A_V >50 massive clumps along the 100-pc Ring.	2017-04-20
10	2015.1.00997.S	54	Extreme quasar feedback in the early Universe	2017-04-20
11	2015.A.00025.S	17	Identifying the Host of the Fast Radio Burst 121102	2017-04-20
12	2015.1.01289.S	610	Bulge Asymmetries and Dynamical Evolution (BAaDE) II	2017-04-20
13	2015.1.01593.S	24	Multi CO line imaging of the nearby galaxy M83: Variation of cloud properties across the galactic structures	2017-04-20

NRAO

CNRC·NRC Canada

JVO: http://jvo.nao.ac.jp/portal/alma/archive.do

"[Part of] the data are retrieved from the JVO portal (http://jvo.nao.ac.jp/portal) operated by the NAOJ"

Target Name | Simbad Name | Project Code | **Coords** | Frequency | Desktop Viewer

1. Center Coords or Target Name J2000 (FK5)

Sample Format : M83 (object name)

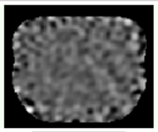
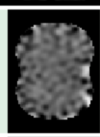
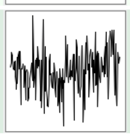
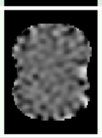
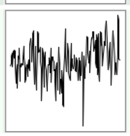
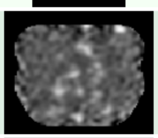
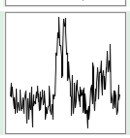
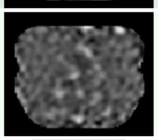
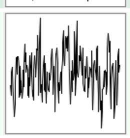
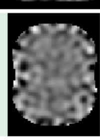
2. Search Radius 10 arcsec

Search

JVOSky

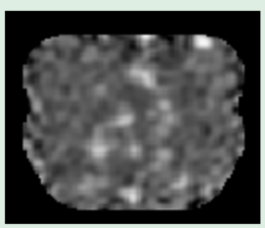
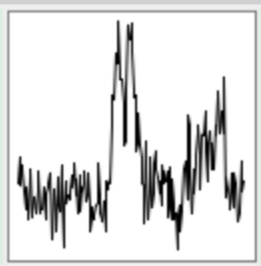
JVO: http://jvo.nao.ac.jp/portal/alma/archive.do

Ordered by dataset_id (desc)

#	dataset id	image	spect	ra/dec (J2000)	target name	Cube size (XxYxF) ?	Image size (arcmin2)	pixel scale, beam size (arcsec)	band	freq. range (GHz)	freq. scale per pix (MHz)
1	ALMA01040445			05h35m51.2 -69d09m52	30_Dor_C	108 x90 x1 x1	5.40 x4.50	3.000, 14.484 x13.569	Band3	100.406 -- 102.903	2,497.526
2	ALMA01040442			05h36m17.5 -69d13m21	MCSNR_J0536-6913	60 x80 x201 x1	3.00 x4.00	3.000, 12.669 x11.419	Band3	115.156 -- 115.180	.122
3	ALMA01040439			05h36m17.5 -69d13m21	MCSNR_J0536-6913	60 x80 x201 x1	3.00 x4.00	3.000, 12.669 x11.419	Band3	115.156 -- 115.180	.122
4	ALMA01040438			05h35m51.2 -69d09m52	30_Dor_C	108 x90 x201 x1	5.40 x4.50	3.000, 12.836 x11.508	Band3	115.156 -- 115.180	.122
5	ALMA01040436			05h35m51.2 -69d09m52	30_Dor_C	108 x90 x201 x1	5.40 x4.50	3.000, 12.836 x11.508	Band3	115.156 -- 115.180	.122
6	ALMA01040434			05h36m17.5 -69d13m21	MCSNR_J0536-6913	60 x80 x1 x1	3.00 x4.00	3.000, 14.367 x13.177	Band3	100.406 -- 102.903	2,497.522

JVO: http://jvo.nao.ac.jp/portal/alma/archive.do

- **Target**
  30_Dor_C
- **Coord. (RA/DEC J2000)**
  05h35m51.2-69d09m52
- **Image Size (arcmin2)**
  5.40x4.50
- **Band Name**
  Band3
- **Freq. Range. (GHz)**
  115.156 -- 115.180
- **Data Type**
  intensity cube
- **Cube Pix ?**
  108x90x201x1
- **3rd(4th) Axis**
  frequency
- **Deprecated / Duplicated**
  false / false
- **Dataset ID**
  ALMA01040438
- **Date of Observations**
  2016-03-18
- **Image Scale and Beam Size. (arcsec)**
  3.000, 12.836x11.508
- **Spectrum Scale per pix. (MHz)**
  .122
- **Project Code**
  2015.1.01232.S
- **science goal UID**
  A001_X2fb_X18a
- **group UID**
  A001_X2fb_X18b
- **member UID**
  A001_X2fb_X18e
- **Original Filename**
  target3DrC_co10b5.cl01.pbcor.fits

data id	image	spect	file size (byte)	Download	WebQL	Readme
ALMA01040438			8,043,840	Download	WebQLv2	Readme

Simbad objects related to the data

2MASS_J05352118-6909101(IR) , 2MASS_J05352612-6909005(Star) , 2MASS_J05352782-6911459(LPV*) , 2MASS_J05352853-6911156(LPV*) , 2MASS_J05353124-6908242(LPV*) , 2MASS_J05353239-6910166(Star) , 2MASS_J05353270-6909012(LPV*) ,

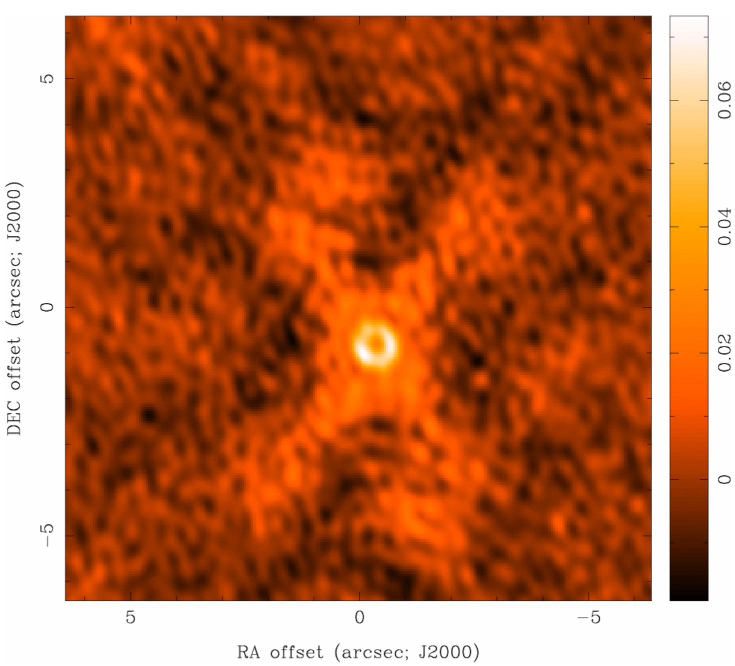
ALMA Guides and Tutorials casaguides.nrao.edu
Navigate to ALMA Guides/Tutorials

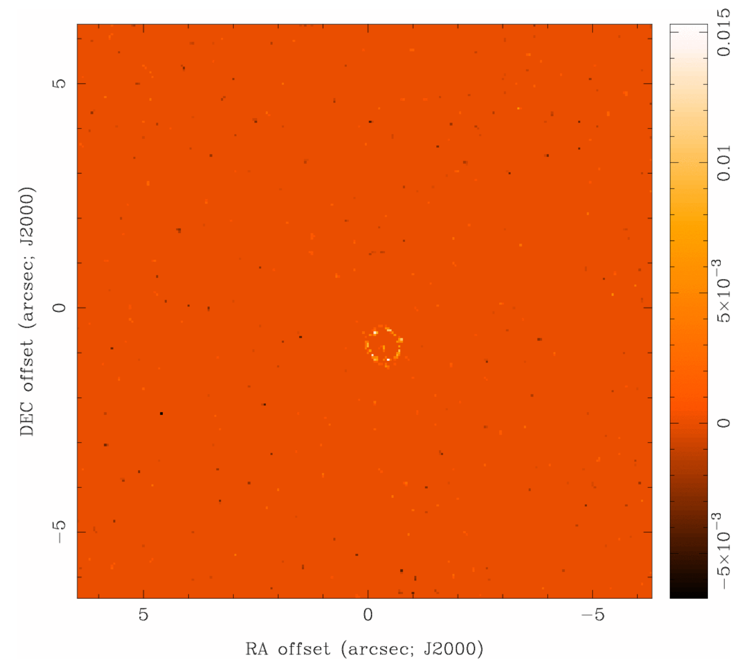
- First Look Imaging Guide: Beginners to CASA should start here. Covers CASA commands, CLEAN, TCLEAN, and a brief section on calibration.
- First Look at Spectral Line Imaging: Start here once you know the basics of CASA imaging. Covers continuum subtraction (UVCONTSUB) and CLEAN.
- First Look at Self Calibration: Start learning how to improve images with high dynamic range or at high frequency by using your source as a calibrator.

  Interested in more? Check out the TWHydraBand7 Imaging guide
- First Look at Image Analysis: Learn the tools available once you have created images. Introduces IMSTAT and IMMOMENTS.

CLEAN

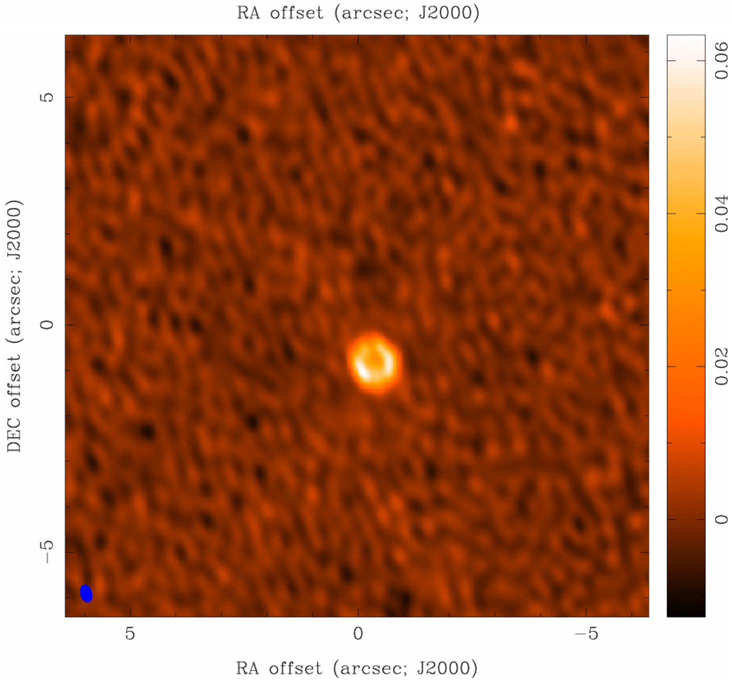
TD(x,y)

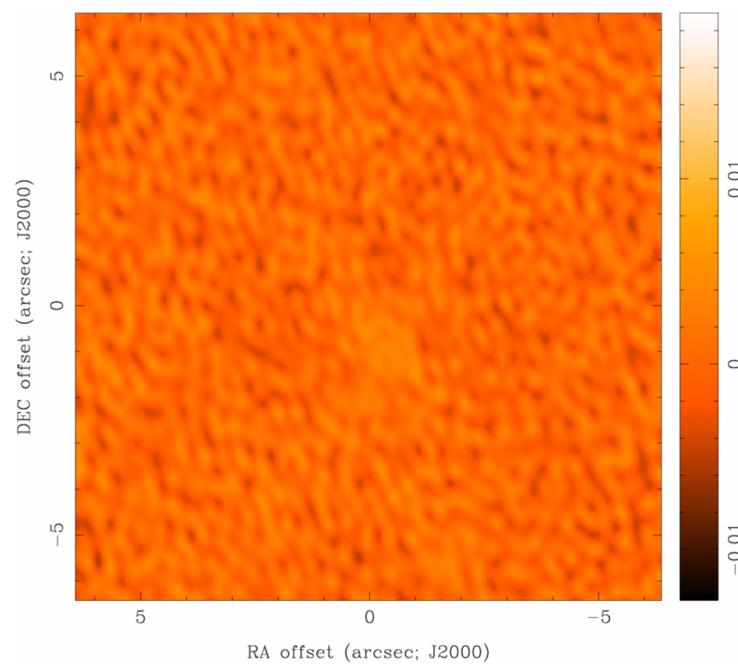




CLEAN model

restored image

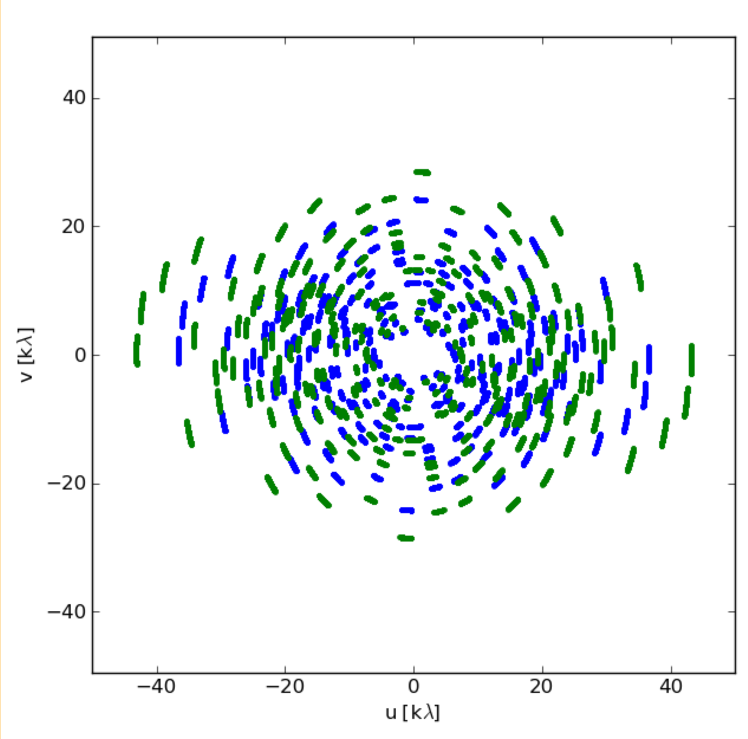




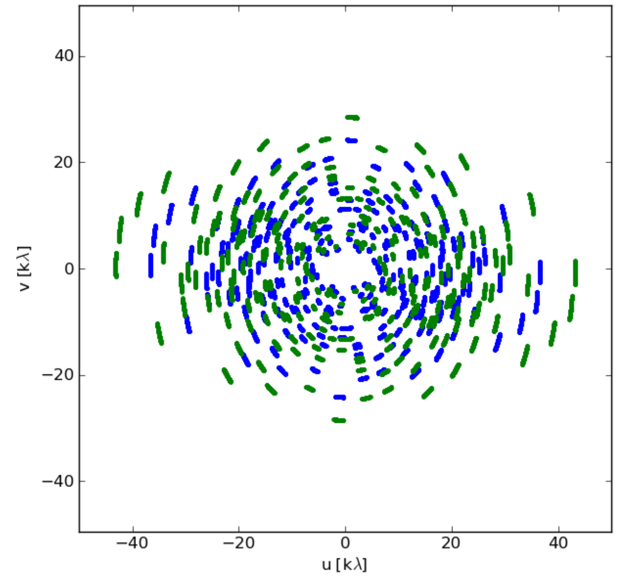
residual map

Select weighting

```
# In CASA
tclean(vis='sis14_twhya_calibrated_flagged.ms',
imagename='secondary_tclean',
field='3',
spw='',
specmode='mfs',
gridder='standard',
deconvolver='hogbom',
nterms=1,
imsize=[128,128],
cell=['0.1arcsec'],
weighting='briggs',
robust=-1.0,
threshold='0mJy',
interactive=True)

imview("secondary_tclean.image")
```



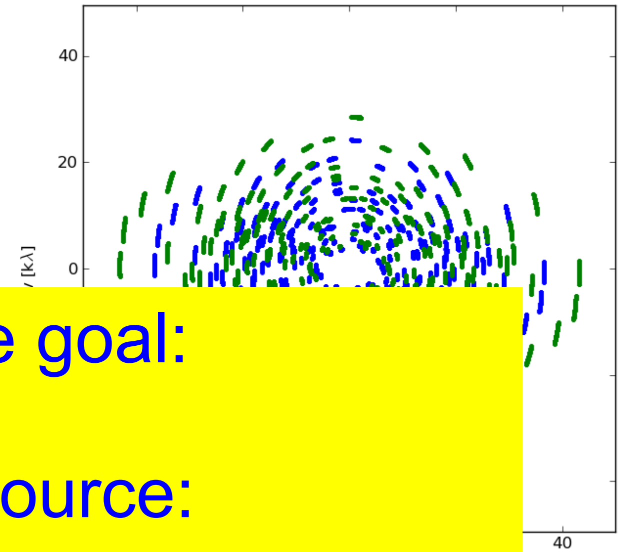
Dirty Beam Shape and Weighting



- Each visibility point is given a weight in the imaging step
- First piece: weight given by Tsys, integration time, etc.
- **Natural**
  - Each sample is given the same weight
  - There are many samples at short baselines, so natural weighting will give the largest beam and the best surface brightness sensitivity (and sometimes pronounced wings in the dirty beam)
- **Uniform**
  - each visibility is given a weight inversely proportional to the sample density
  - Weighs down short baselines, long baselines are more pronounced. Best resolution; poorer noise characteristics
- **Briggs (Robust)**
  - A graduated scheme using the parameter *robust;* compromise of noise and resolution
  - In CASA, set *robust* from -2 (~ uniform) to +2 (~ natural)
  - *robust* = 0 often a good choice

- **Taper:** additional weight function to be applied (typically a Gaussian to suppress the weights of the outer visibilities – be careful, however, not to substantially reduce the collecting area)

Dirty Beam Shape and Weighting



- Each visibility point is given a weight in the imaging step
- First piece: weight given by Tsys, integration time, etc.
- **Natural**
  - Each sample is given the same weight

Adjust the weighting to match your science goal:

→ Detection experiment/weak extended source: **natural** (maybe even with a taper)

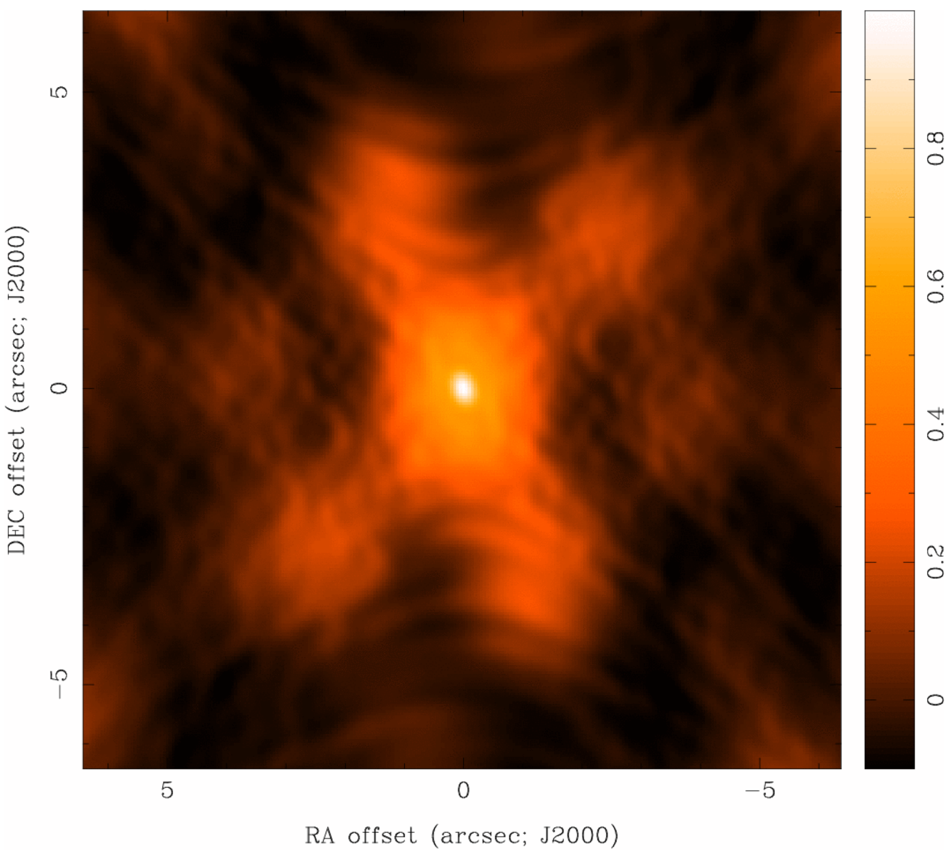
→ Finer detail of strong sources: **robust** or even **uniform**

  - A graduated scheme using the parameter *robust*; compromise of noise and resolution
  - In CASA, set *robust* from -2 (~ uniform) to +2 (~ natural)
  - *robust* = 0 often a good choice

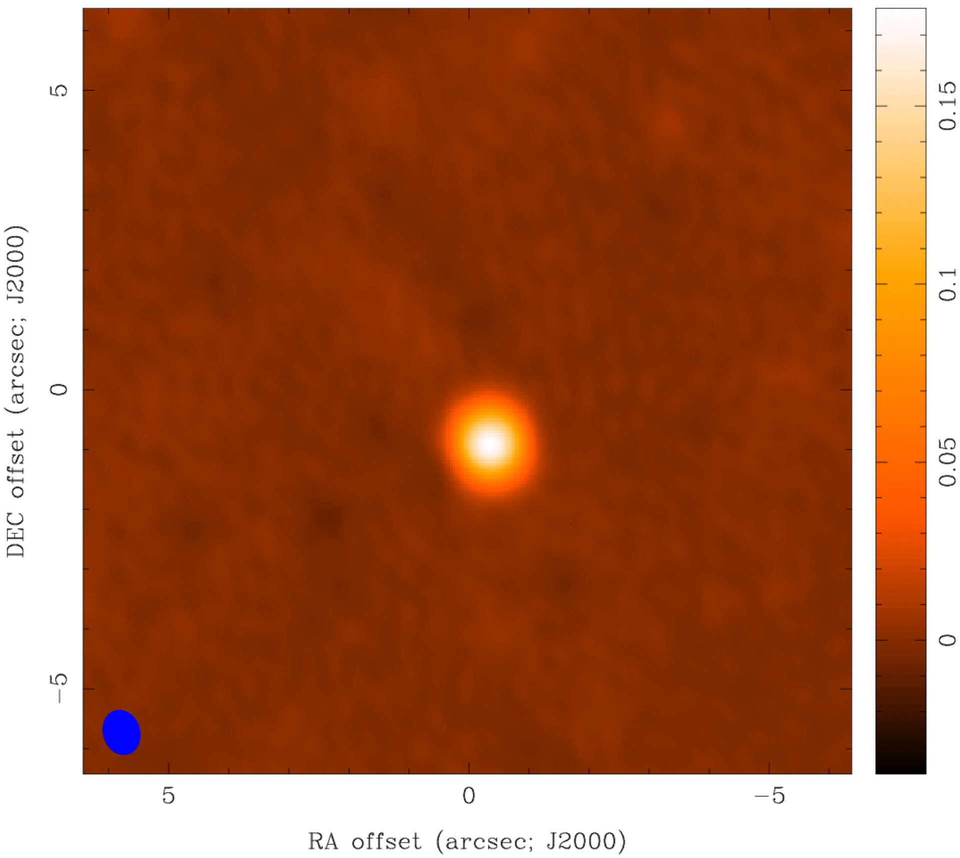
Taper: additional weight function to be applied (typically a Gaussian to suppress the weights of the outer visibilities – be careful, however, not to substantially reduce the collecting area)

Imaging Results

Natural Weight Beam

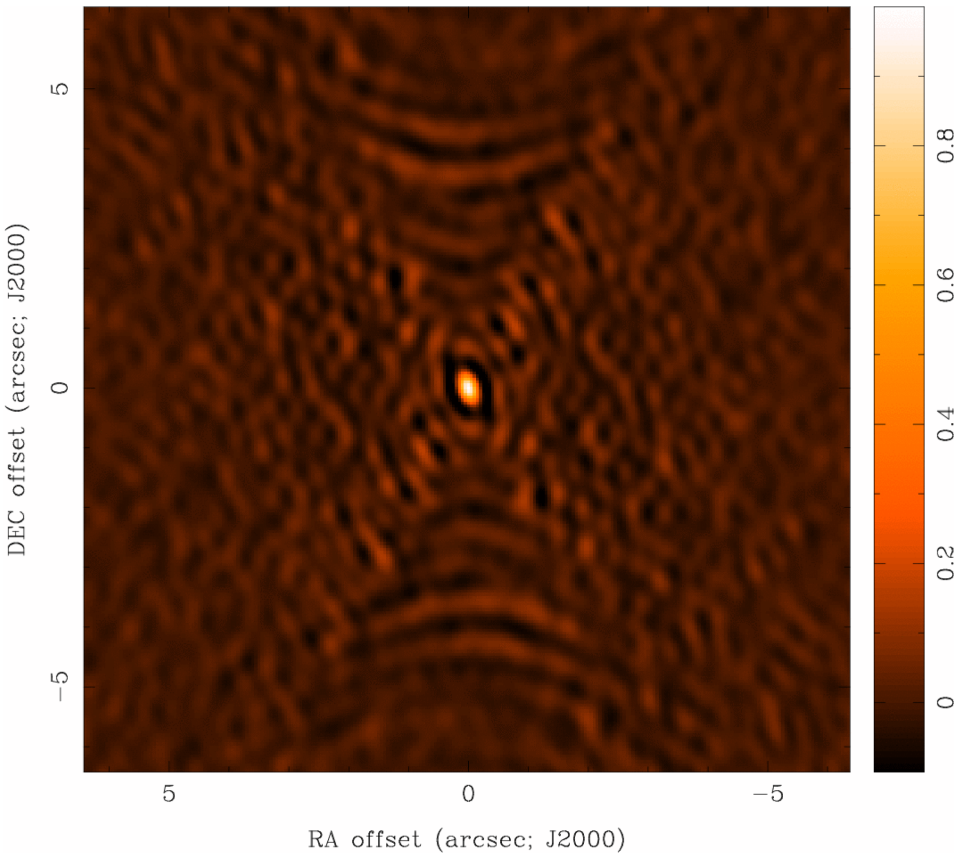


CLEAN image

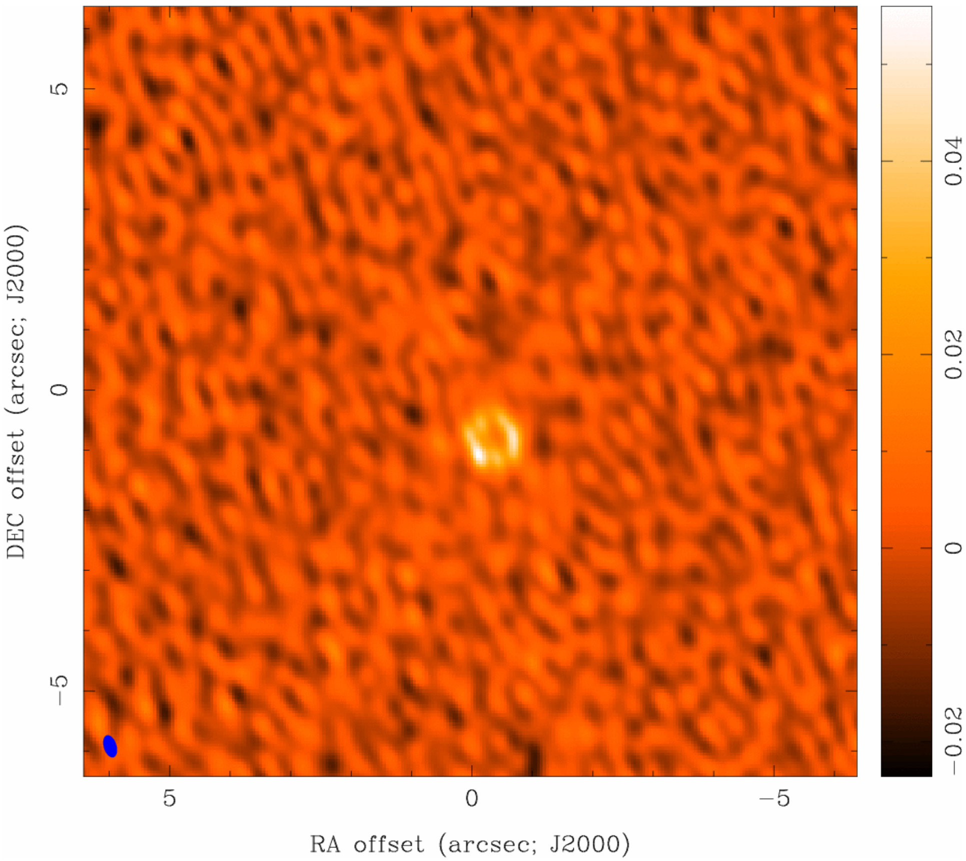


Imaging Results

Uniform Weight Beam

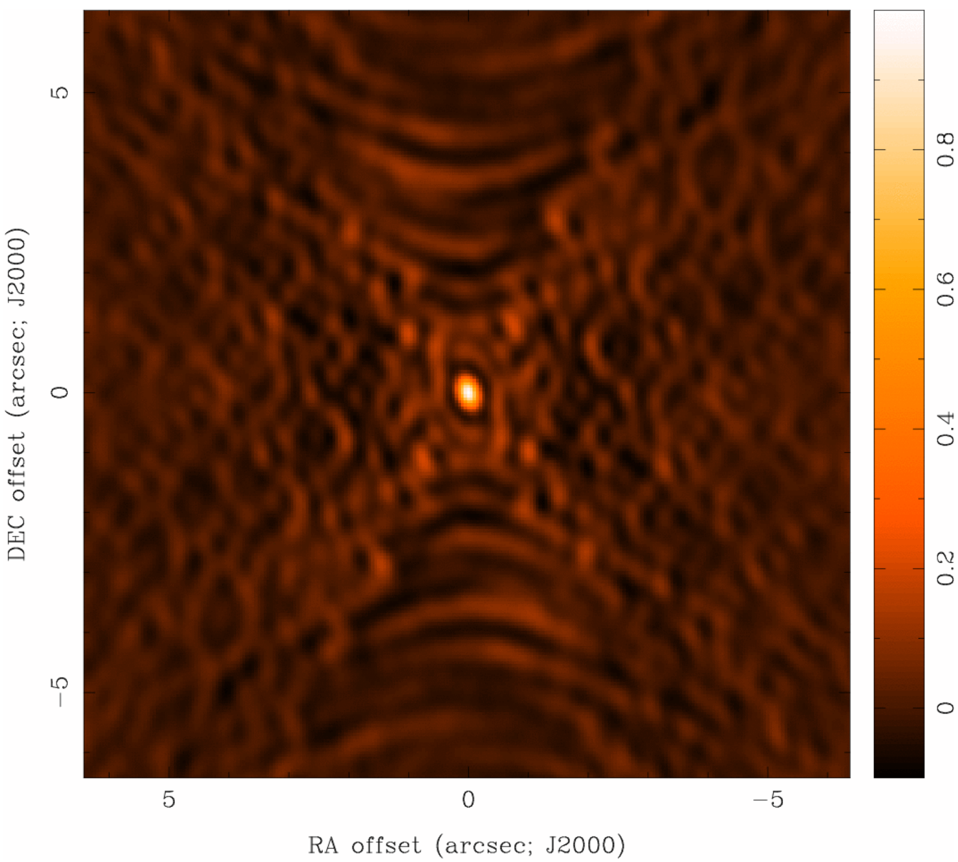


CLEAN image

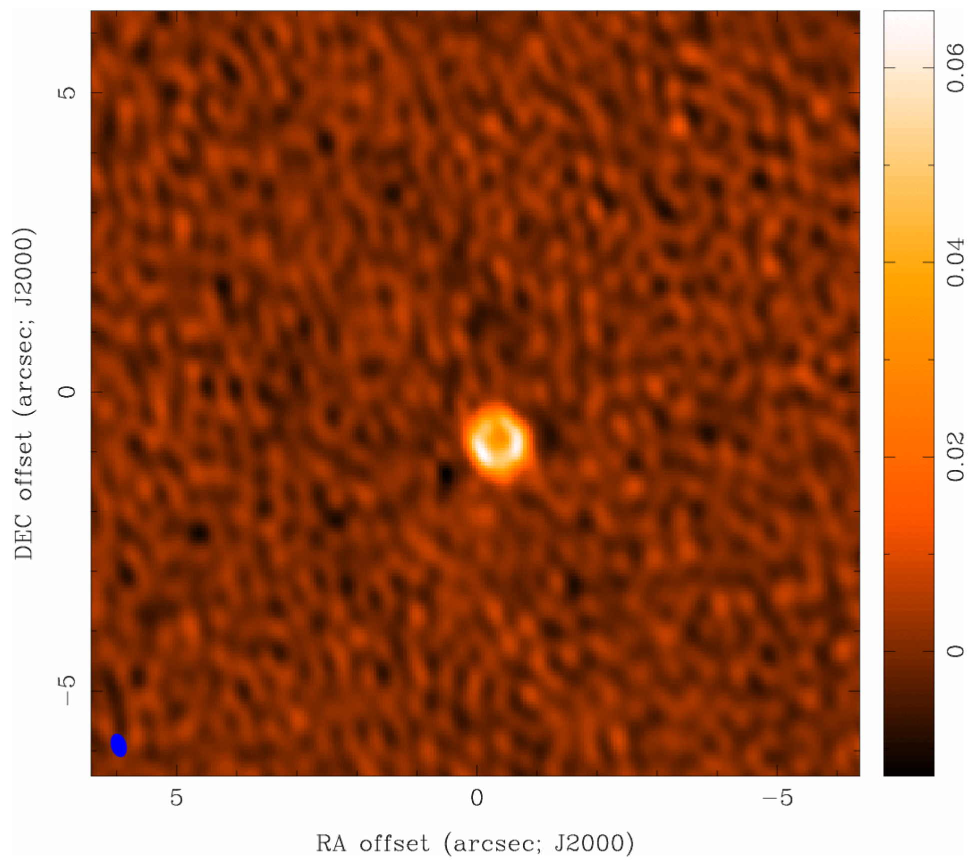


Imaging Results

Robust=0 Beam



CLEAN image

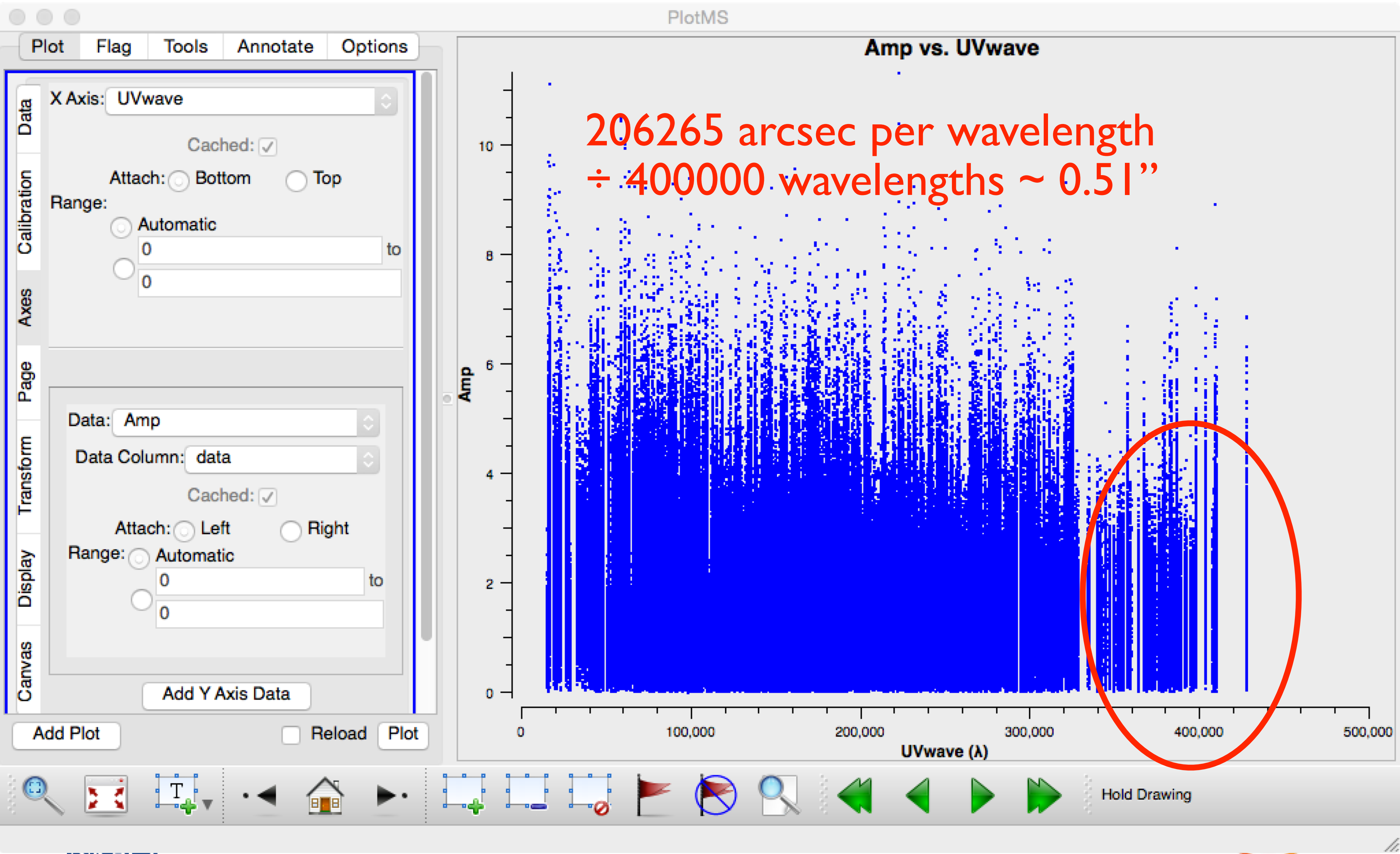


Select pixel scale and image size

```
# In CASA
tclean(vis='sis14_twhya_calibrated_flagged.ms',
imagename='secondary_tclean',
field='3',
spw='',
specmode='mfs',
gridder='standard',
deconvolver='hogbom',
nterms=1,
imsize=[128,128],
cell=['0.1arcsec'],
weighting='briggs',
robust=-1.0,
threshold='0mJy',
interactive=True)

imview("secondary_tclean.image")
```

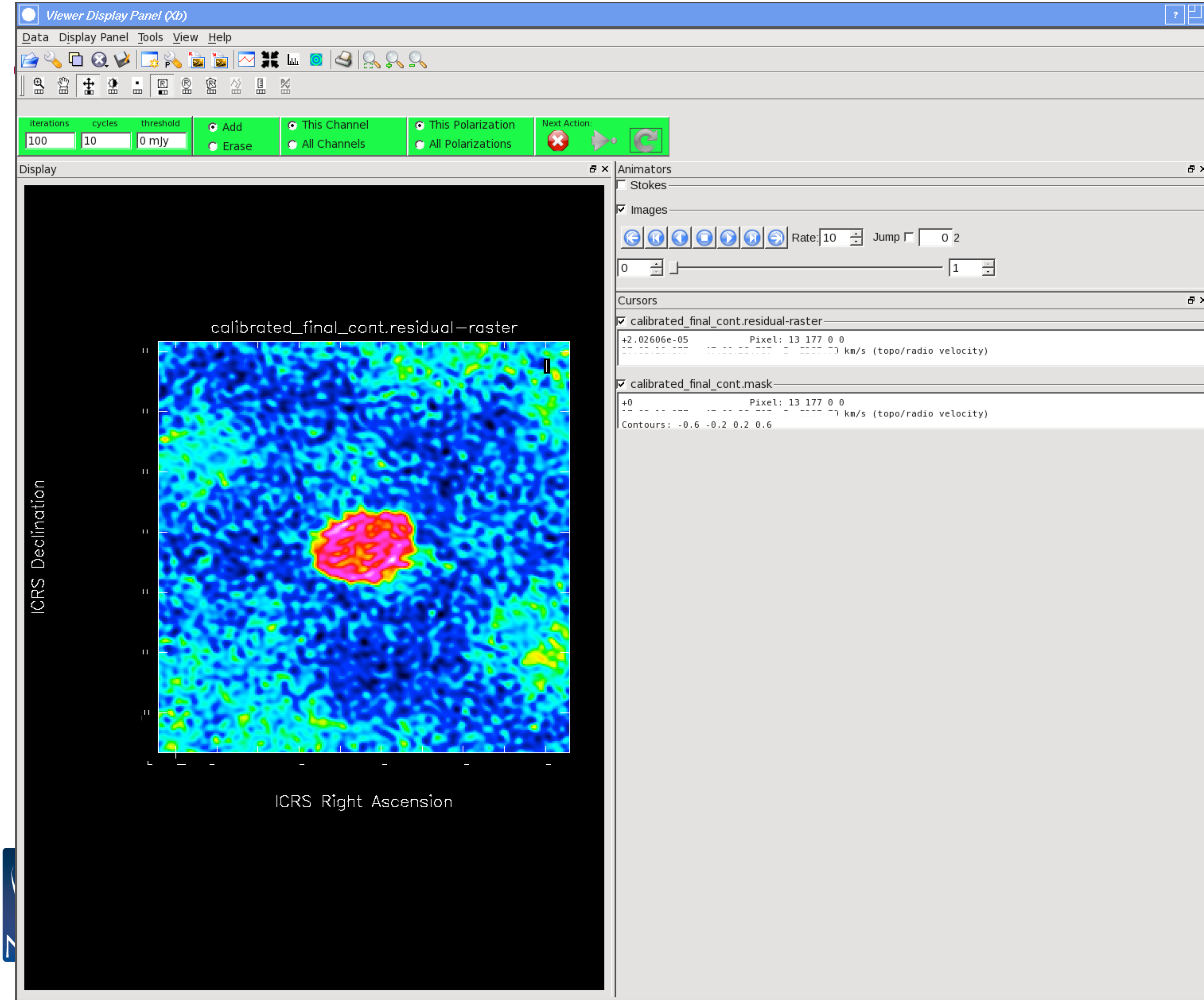
- Want ~5 pixels per angular resolution
- cell*imsize should at least cover the primary beam (field-of-view) or size of mosaic

Select pixel scale and image size



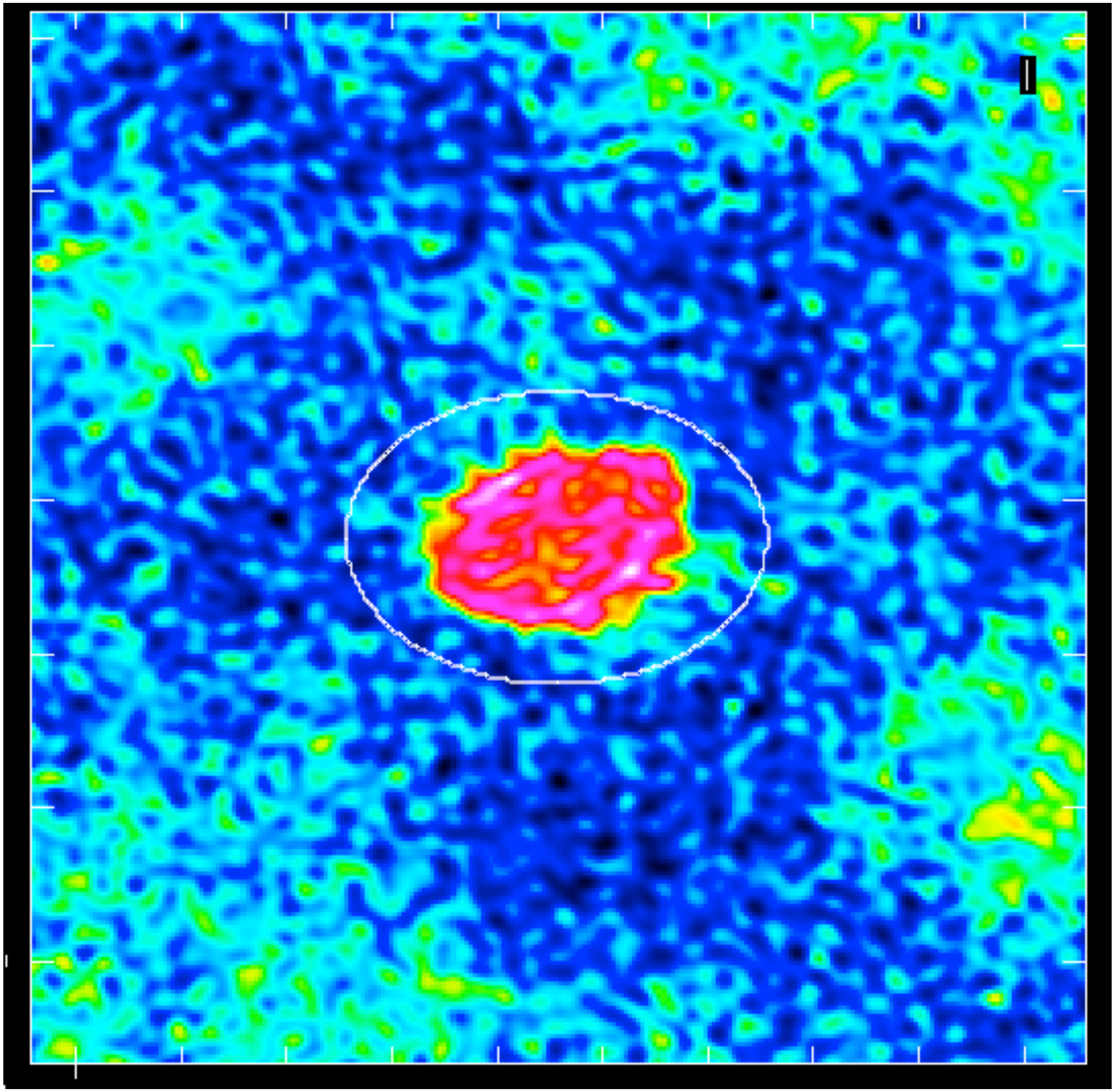
Interactive Clean

```
     scan              =         ''        #  Scan number range
     observation       =         ''        #  Observation ID range
     intent            =         ''        #  Scan Intent(s)

mode                   =      'mfs'        #  Spectral gridding type (mfs, channel, velo
     nterms            =          1        #  Number of Taylor coefficients to model the
     reffreq           =         ''        #  Reference frequency (nterms > 1),'' uses c

gridmode               =         ''        #  Gridding kernel for FFT-based transforms,
niter                  =       1000        #  Maximum number of iterations
gain                   =        0.1        #  Loop gain for cleaning
threshold              =   '0.0mJy'        #  Flux level to stop cleaning, must include
psfmode                =    'clark'        #  Method of PSF calculation to use during mi
imagermode             =  'csclean'        #  Options: 'csclean' or 'mosaic', '', uses p
     cyclefactor       =        1.5        #  Controls how often major cycles are done.
     cyclespeedup      =         -1        #  Cycle threshold doubles in this number of

multiscale             =         []        #  Deconvolution scales (pixels); [] = standa
interactive            =       True        #  Use interactive clean (with GUI viewer)
     npercycle         =        100        #  Clean iterations before interactive prompt

mask                   =         []        #  Cleanbox(es), mask image(s), region(s), or
imsize                 = [256, 256]        #  x and y image size in pixels.  Single valu
cell                   = '0.08arcsec'      #  x and y cell size(s). Default unit arcsec.
phasecenter            =         ''        #  Image center: direction or field index
restfreq               =         ''        #  Rest frequency to assign to image (see hel
stokes                 =        'I'        #  Stokes params to image (eg I,IV,IQ,IQUV)
weighting              =   'briggs'        #  Weighting of uv (natural, uniform, briggs,
     robust            =        0.5        #  Briggs robustness parameter
     npixels           =          0        #  number of pixels to determine uv-cell size

uvtaper                =      False        #  Apply additional uv tapering of visibiliti
modelimage             =         ''        #  Name of model image(s) to initialize clean
restoringbeam          =       ['']        #  Output Gaussian restoring beam for CLEAN i
pbcor                  =      False        #  Output primary beam-corrected image
minpb                  =        0.2        #  Minimum PB level to use
usescratch             =      False        #  True if to save model visibilities in MODE
allowchunk             =      False        #  Divide large image cubes into channel chun

CASA <25>:
```

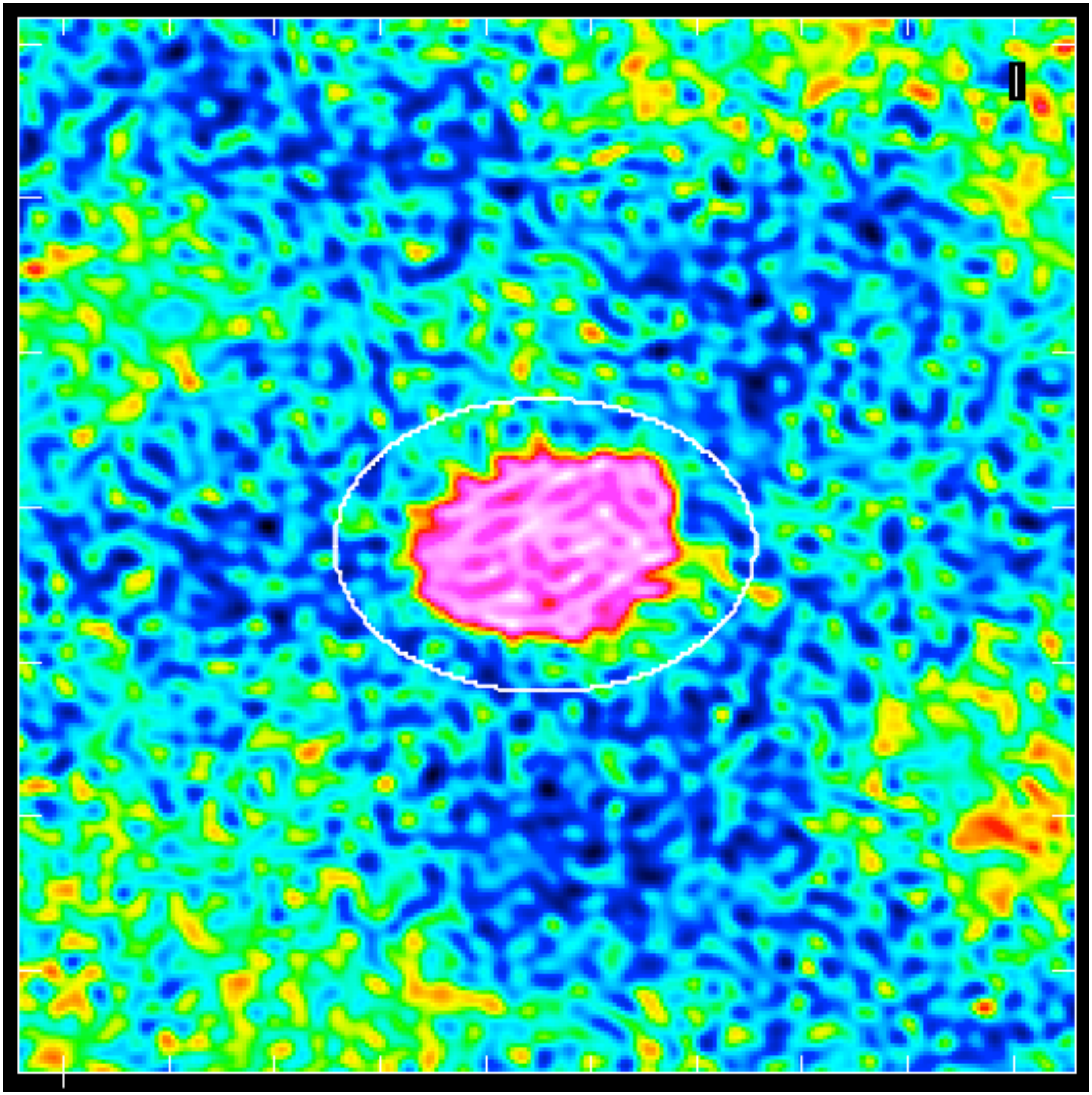


Viewer Display Panel (Xb)
Data Display Panel Tools View Help
iterations cycles threshold
100 10 0 mJy
Add
Erase
This Channel
All Channels
This Polarization
All Polarizations
Next Action:
Display
calibrated_final_cont.residual−raster
ICRS Declination
ICRS Right Ascension
Animators
Stokes
Images
Rate: 10
Jump
0 2
0
1
Cursors
calibrated_final_cont.residual-raster
+2.02606e-05 Pixel: 13 177 0 0
km/s (topo/radio velocity)
calibrated_final_cont.mask
+0 Pixel: 13 177 0 0
km/s (topo/radio velocity)
Contours: -0.6 -0.2 0.2 0.6

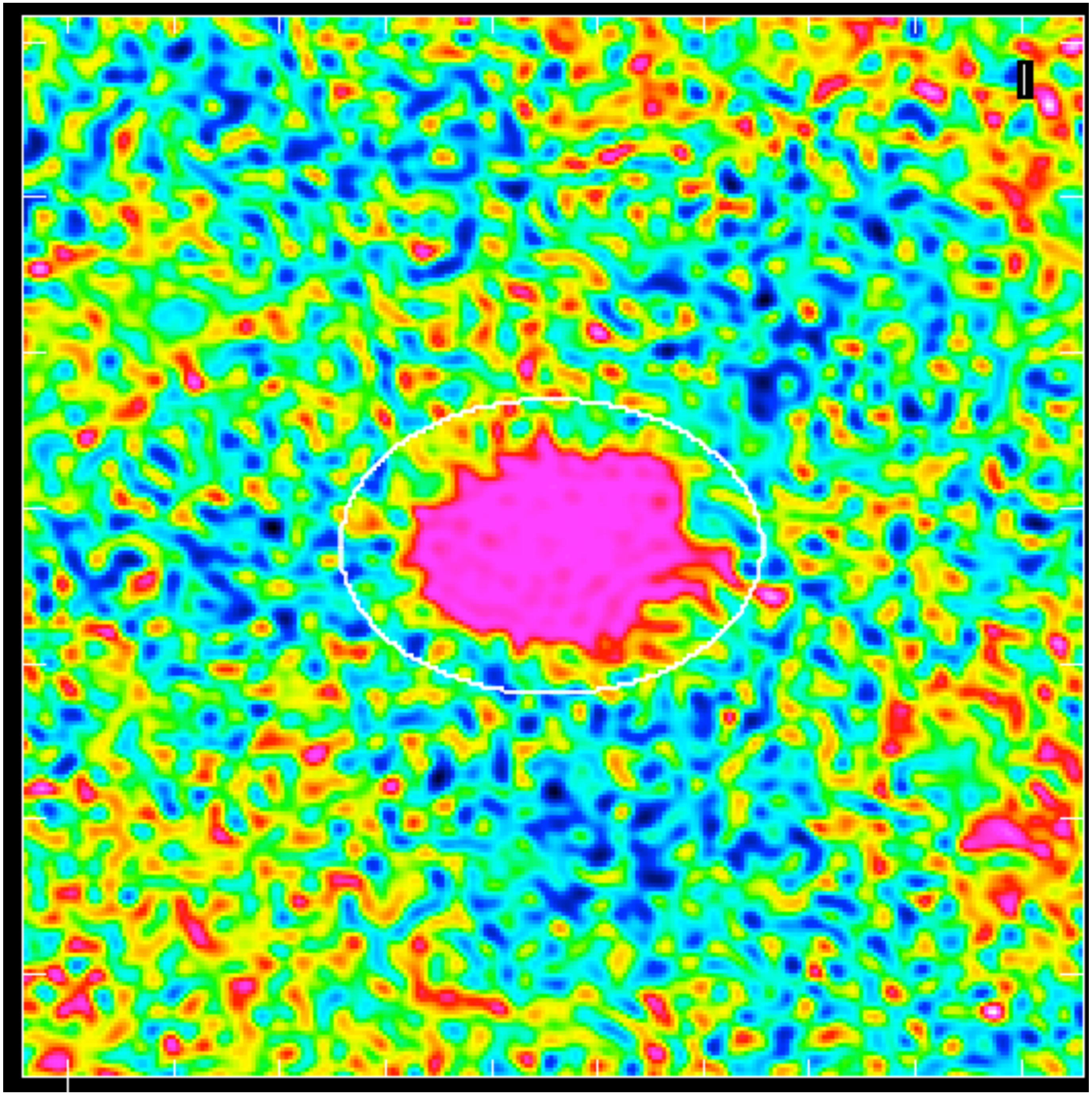
CLEAN



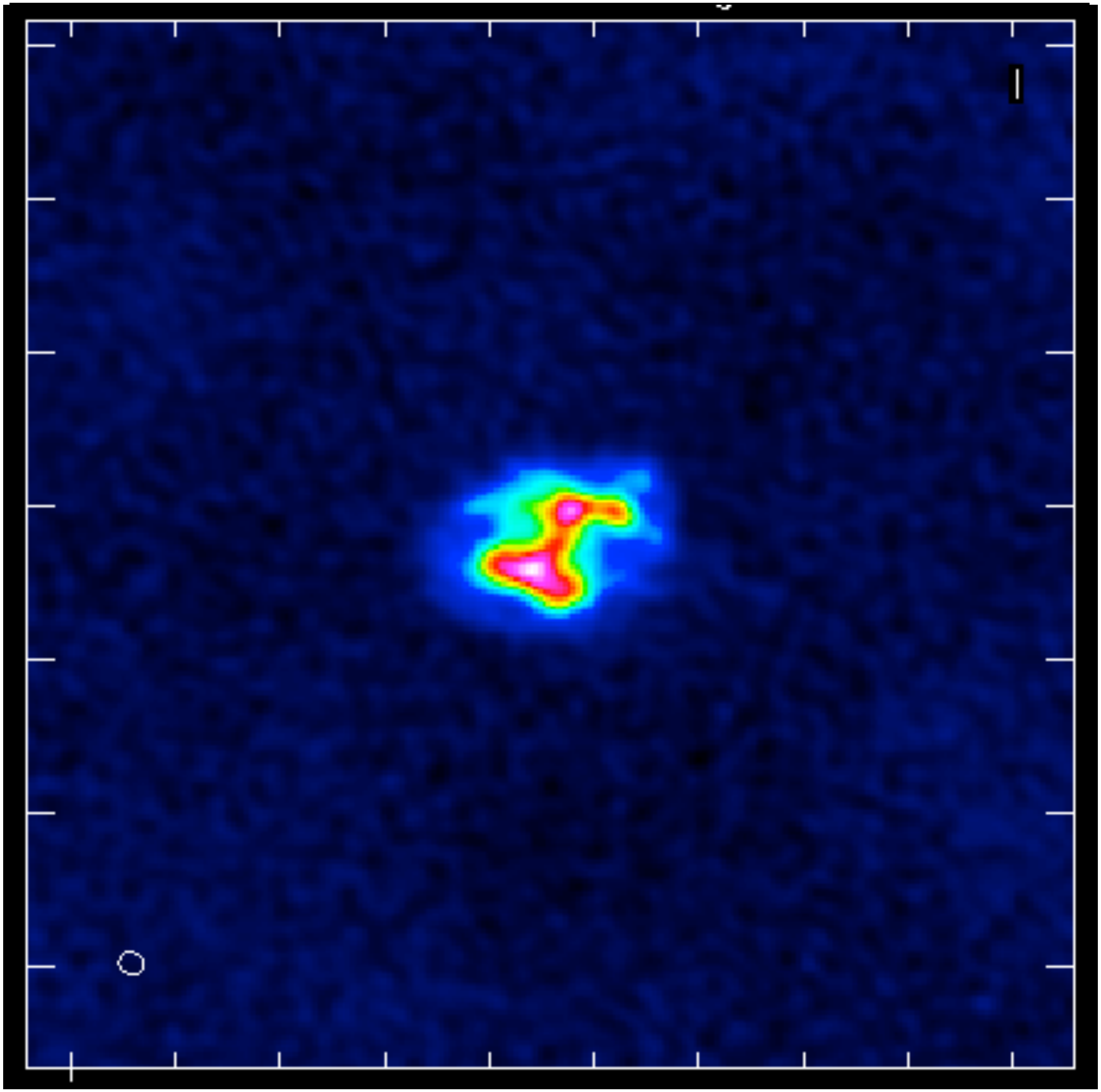
CLEAN



CLEAN



CLEAN



Imaging spectral lines: continuum subtraction

- Generally would like to subtract continuum emission (we will see how to identify line-free channels in hands-on session)
- Use `uvcontsub` to do the subtraction in uv plane.

```
CASA <11>: inp
---------> inp()
#  uvcontsub :: Continuum fitting and subtraction in the uv plane
vis                 = 'ngc3256_co.ms'   #  Name of input MS.  Output goes to vis + ".contsub"
field               =        ''         #  Select field(s) using id(s) or name(s)
fitspw              = '0:20~53;71~120'  #  Spectral window:channel selection for fitting the continuum
combine             =        ''         #  Data axes to combine for the continuum estimation (none, or spw and/or scan)
solint              =      'int'        #  Continuum fit timescale (int recommended!)
fitorder            =         0         #  Polynomial order for the fits
spw                 =        ''         #  Spectral window selection for output
want_cont           =      False        #  Create vis + ".cont" to hold the continuum estimate.
async               =      False        #  If true the taskname must be started using uvcontsub(...)
```

Imaging spectral lines

```
mode             = 'velocity'       #  Spectral gridding type (mfs, channel,
                                    #   velocity, frequency)
  nchan          =        100       #  Number of channels (planes) in output
                                    #   image; -1 = all
  start          =  '300km/s'       #  Velocity of first channel: e.g
                                    #   '0.0km/s'(''=first channel in first
                                    #   SpW of MS)
  width          =   '10km/s'       #  Channel width e.g '-1.0km/s'
                                    #   (''=width of first channel in first
                                    #   SpW of MS)
  interpolation  =   'linear'       #  Spectral interpolation (nearest,
                                    #   linear, cubic).
  resmooth       =      False       #  Re-restore the cube image to a common
                                    #   beam when True
  chaniter       =      False       #  Clean each channel to completion
                                    #   (True), or all channels each cycle
                                    #   (False)
  outframe       =     'LSRK'       #  Velocity reference frame of output
                                    #   image; '' =input
  veltype        =    'radio'       #  Velocity definition of output image
restfreq           = '115.271201800GHz' #  Rest frequency to assign to image (see help)
```

mode="velocity"

→ Set the dimensions of the cube

→ Set Rest frequency

→ Set Velocity Frame (LSRK, BARY, …)

→ Set Doppler definition (optical/radio)

Clean will calculate the Doppler corrections for you! No need to realign beforehand. (but **cvel** will do it for you if needed, e.g. when self-calibrating)

Self-Calibration: Motivation

JVLA and ALMA have such impressive sensitivity that what you achieve is often limited by residual calibration errors!

To surpass this, many objects have enough Signal-to-Noise (S/N) that they can be used to calibrate ***themselves*** to obtain a better image. This is self-calibration.

- Sometimes, the increase in effective sensitivity may be an *order of magnitude!*

It is not a circular trick to produce the image that you want. It works because the number of baselines is much larger than the number of antennas so that an approximate source image does not stop you from determining a better temporal gain calibration which leads to a better source image.

Self-calibration may not be included in the data pipelines

…so, it's best you learn how to do it

But remember: self calibration is just **regular calibration**.

What Sensitivities do I need for Self-Calibration?

- **For phase only self-cal:** Need to detect the target with a S/N > 3 in a solution time (**solint**) less than the time for significant phase variations for all baselines to **a single antenna.** For 25 antennas, S/N > 3 will lead to < 15 deg error.

- Make an initial image, cleaning it conservatively
  - Measure rms in emission free region
  - rms_{Ant} = rms x sqrt(N-3) where N is # of antennas
  - rms_{self} = rms_{Ant} x sqrt(total time/solint)
  - Measure Peak flux density = Signal
  - If S/N_{self} = $Peak/rms_{Self}$ >3 try phase only self-cal

Rule of thumb:

For an array with 20+ antennas, if S/N in image >20 its worth trying phase-only self-cal

- CAVEAT 1: If dominated by extended emission, estimate what the flux will be on the longer baselines (by plotting the uv-data) instead of the image
  - If the majority of the baselines in the array cannot "see" the majority of emission in the target field (i.e. emission is resolved out) at a S/N of about 3, the self-cal will fail in extreme cases (though bootstrapping from short to longer baselines is possible, it can be tricky).

- CAVEAT 2: If severely dynamic range limited (poor uv-coverage), it can also be helpful to estimate the rms noise from uv-plots

What Sensitivities do I need for Self-Calibration?

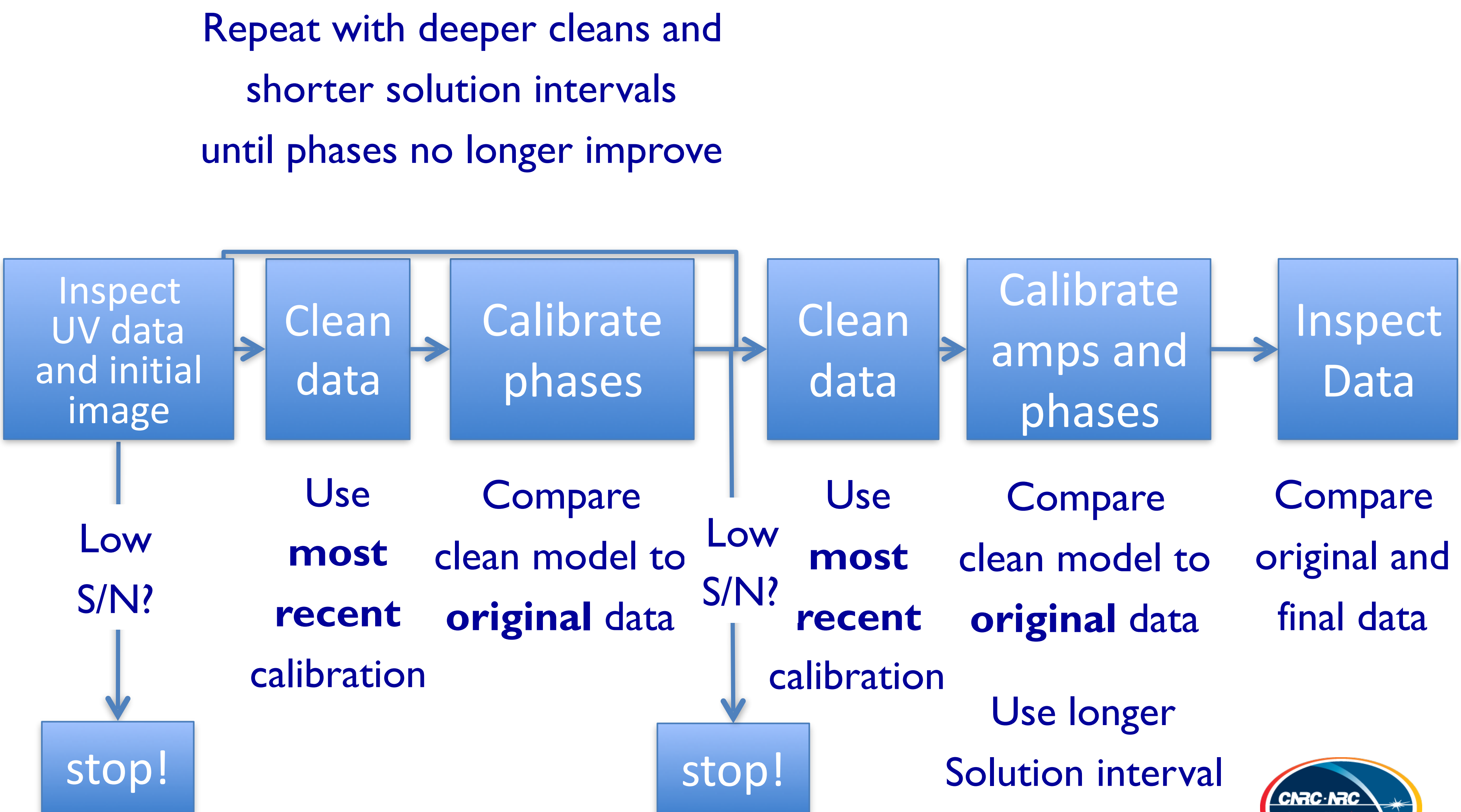
- **For amplitude self-cal:** Need to detect the target with a S/N > 10 with only the baselines to a single antenna in a solution time (solint) less than the time for significant amplitude variations. For 25 antennas, an antenna based S/N > 10 will lead to a 10% amplitude error.
  - Amplitude corrections are more subject to deficiencies in the model image. Check results carefully!
  - For example, if clean model is missing significant flux compared to uv-data, give uvrange for amplitude solution that excludes short baselines.

Additional S/N for self-cal can be obtained by:

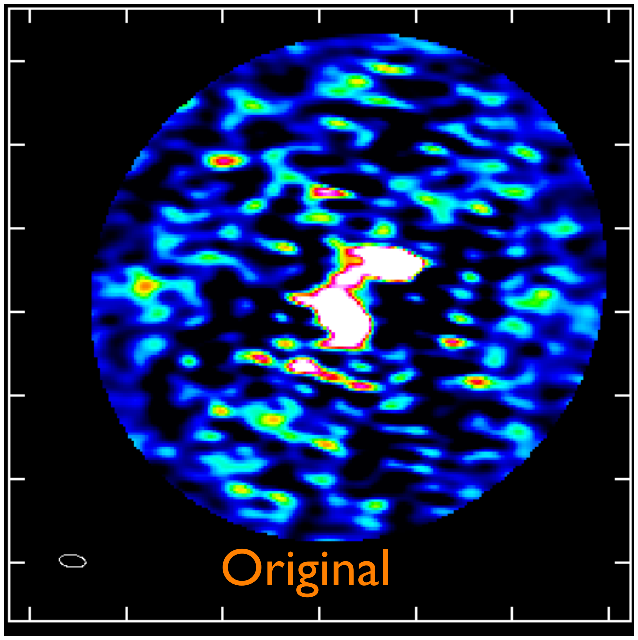
- Increase solint (solution interval)
  - If errors that are directional rather than time dependent, self-calibration solutions can yield surprising improvement even for solints that span the entire observation. Antenna position (aka baseline) errors are a good example.
- gaintype= 'T' to average polarizations
  - Caveat: Only if your source is unpolarized
- Combine = 'spw' to average spw's (assumes prior removal of spw to spw offsets)
  - Caveat: If source spectral index/morphology changes significantly across the band, do not combine spws, especially for amplitude self-cal
- Combine = 'fields' to average fields in a mosaic (use with caution, only fields with strong signal)

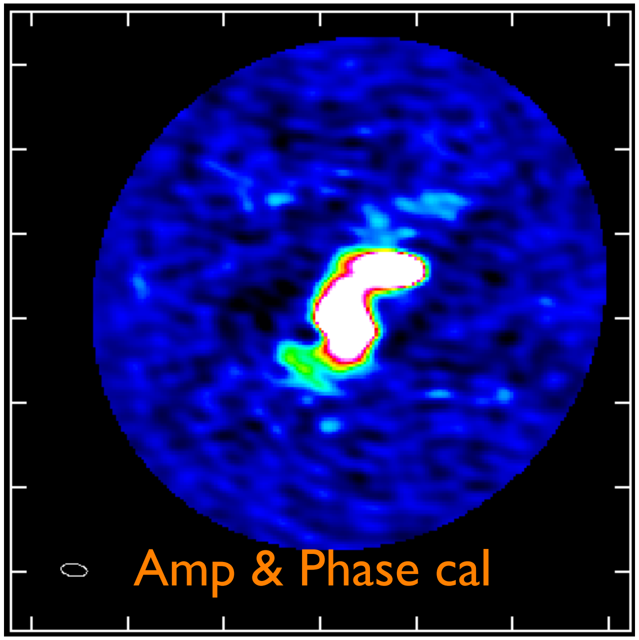
NRAO

Outline of Self-Calibration Process



Self-calibration Example: see workshop page ALMA SV Data for IRAS16293 Band 6



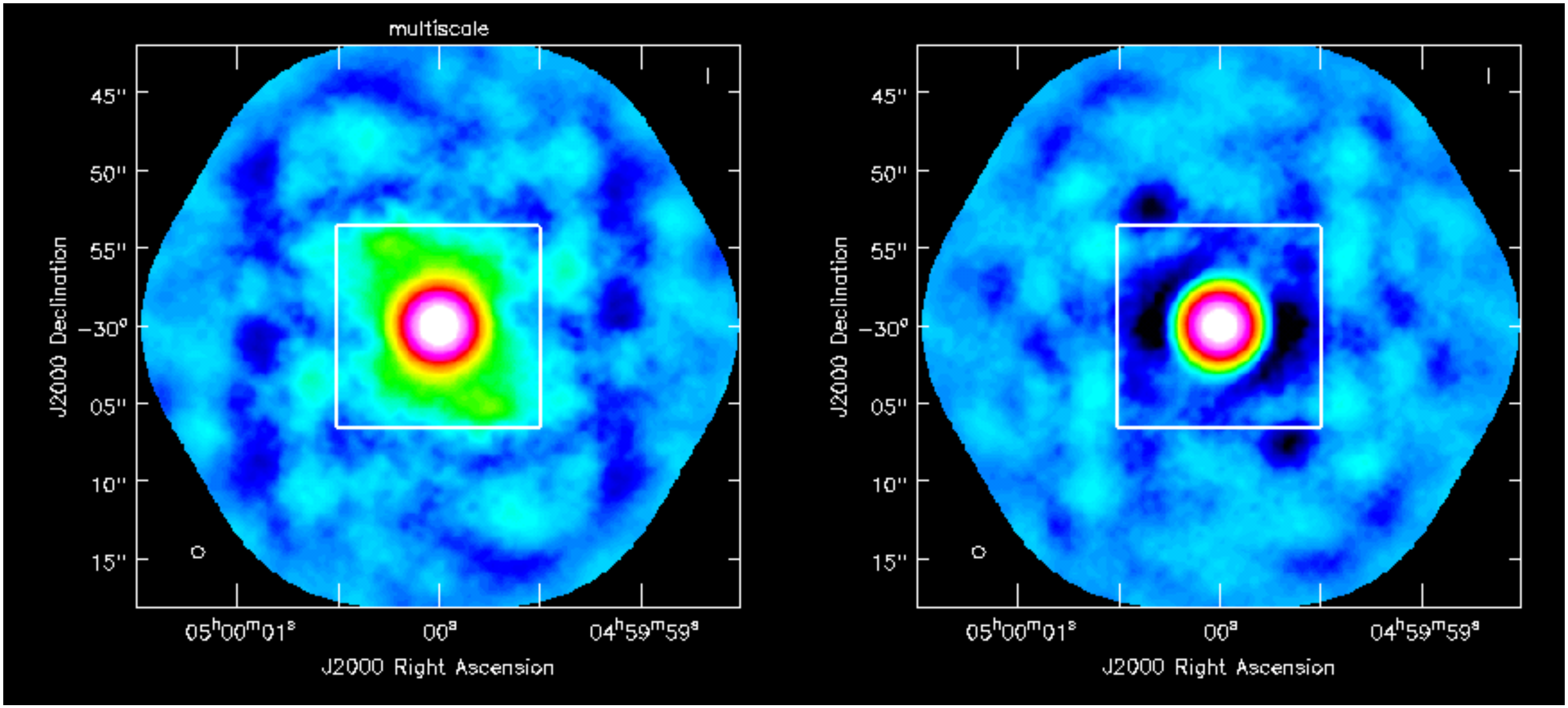


See [IRAS16293 CASA guide]for detailed commands.

http://casaguides.nrao.edu/index.php?title=IRAS16293_Band9_-_Imaging_for_CASA_4.3

Multi-scale CLEAN

multi-scale — “classic” scale



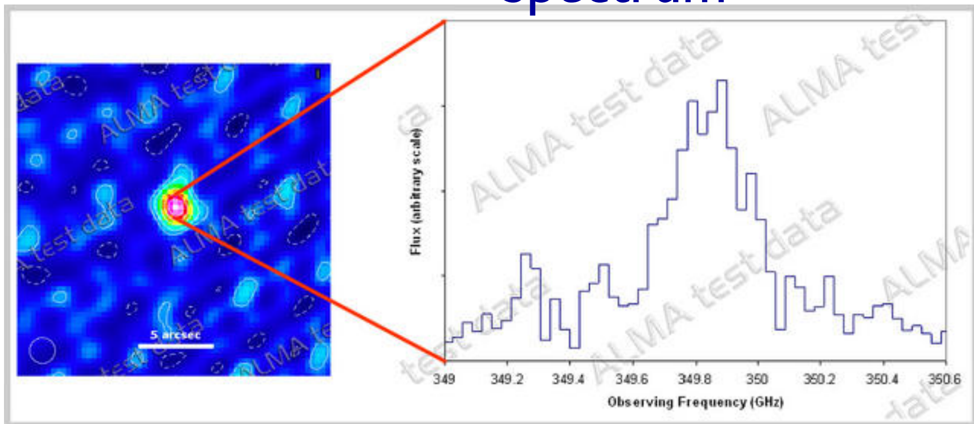

```
multiscale          = [0, 5, 12, 24, 50] #  Deconvolution scales (pixels); [] =
                                         #  standard clean
```

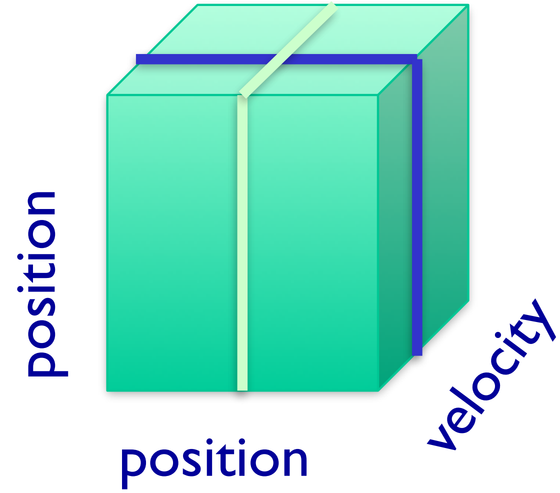
Instead of delta functions, one can use extended clean components to better match emission scales (multiscales, typically paraboloids)

Pick delta function, half the largest emission and a few in between

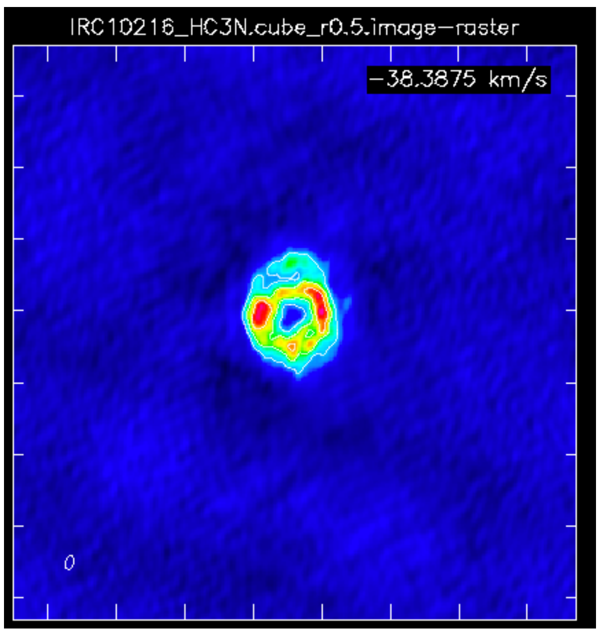
Imaging spectral lines

- Spectrum



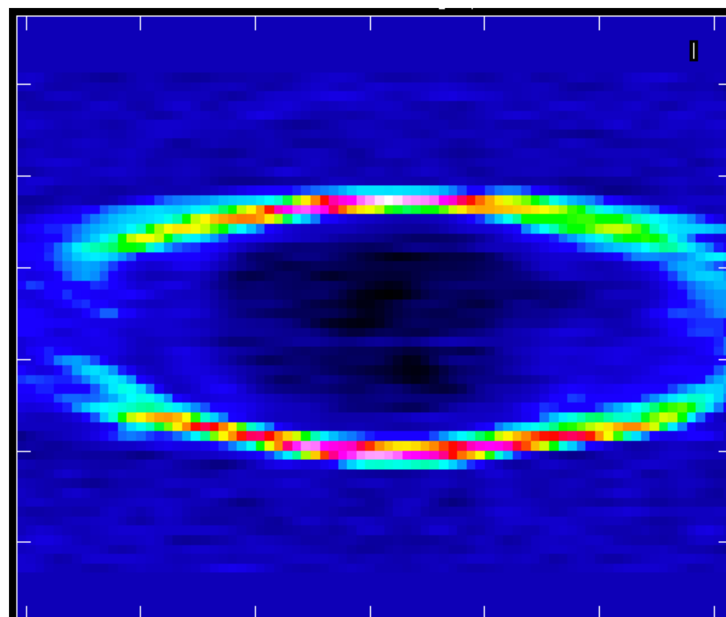


- Channel map



Fixed velocity, polarization, etc.

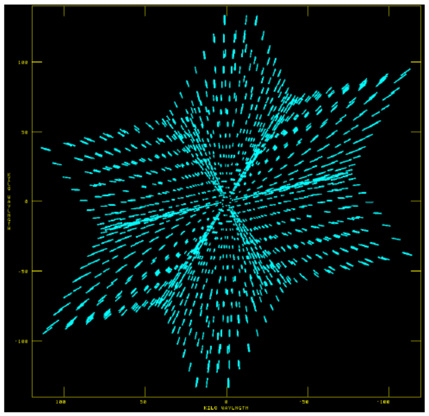
- Position-velocity map



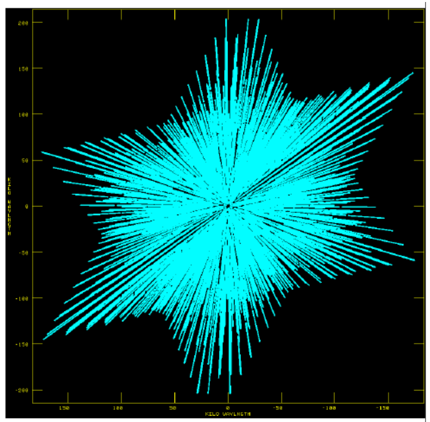
One fixed position, polarization, etc.

Continuum Imaging

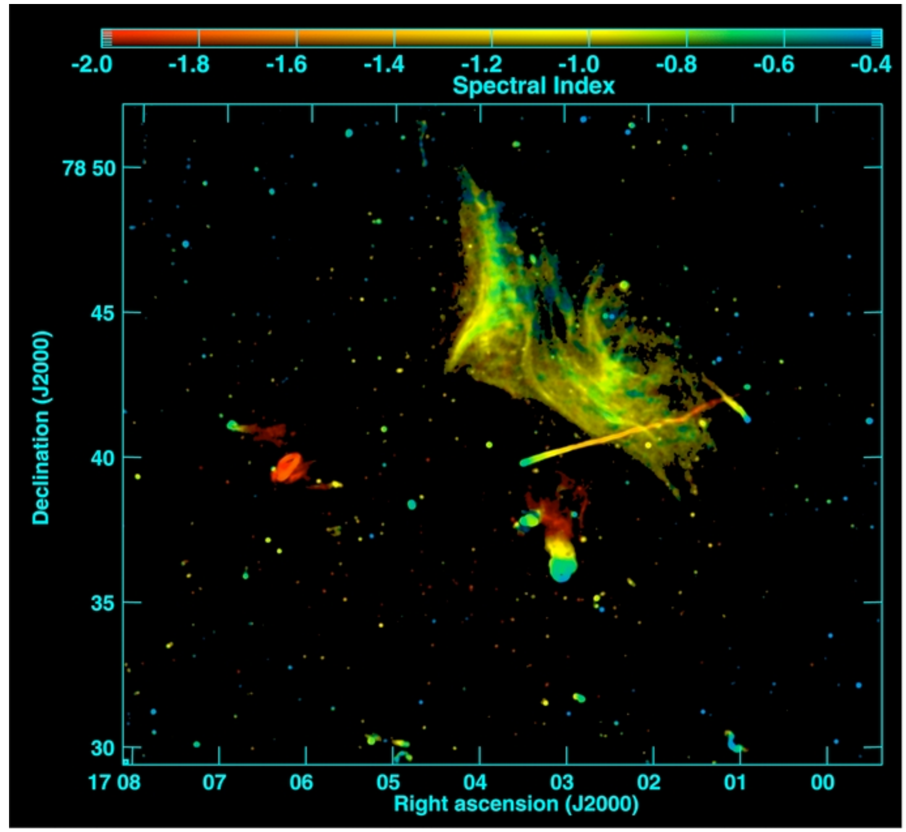
- Multi-scale Multi-Frequency Taylor Term expansion



Narrow BW



wide BW
(better uv-coverage)

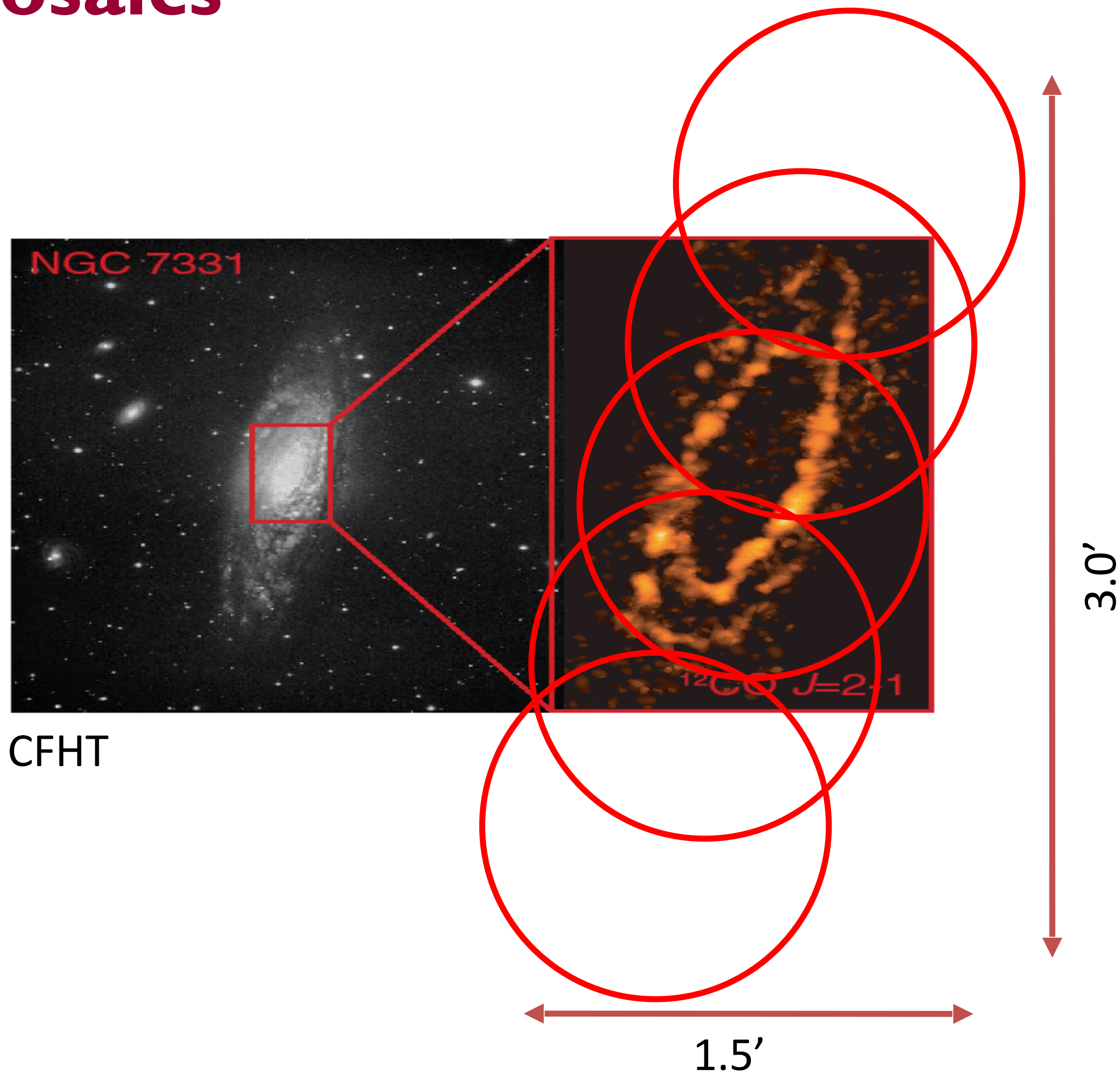


Abell 2256; Owen et al. (2014)

- Plus spectral index:

- MFS (mode mfs)
  - nterm=2 compute spectral index, 3 for curvature etc.
  - needed for bandwidths ~5% or more (S/N dependent)
  - tt0 average intensity, tt1 alpha*tt0, alpha images output
  - takes at least nterms longer (image size dependent)

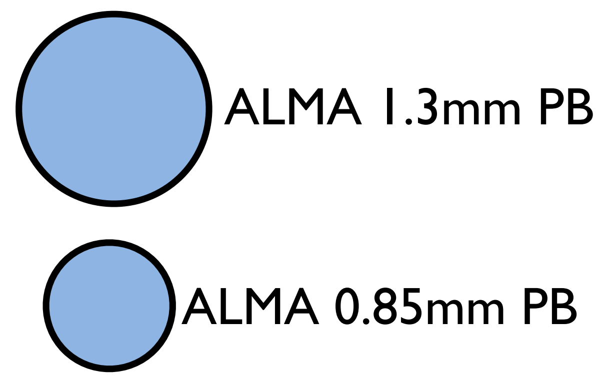
Mosaics



CFHT

Example: SMA 1.3 mm observations: 5 pointings

- Primary beam ~1’
- Resolution ~3”



Petitpas et al.

Imaging mosaics

```
imagermode      =   'mosaic'   #  Options: 'csclean' or 'mosaic', '', uses psfmode
  mosweight     =      False   #  Individually weight the fields of the mosaic
  ftmachine     =       'ft'   #  Gridding method for the image
  scaletype     =    'SAULT'   #  Controls scaling of pixels in the image plane. default='SAULT'; example:
                               #   scaletype='PBCOR' Options: 'PBCOR','SAULT'
  cyclefactor   =        1.5   #  Controls how often major cycles are done. (e.g. 5 for frequently)
  cyclespeedup  =         -1   #  Cycle threshold doubles in this number of iterations
  flatnoise     =       True   #  Controls whether searching for clean components is done in a constant noise
                               #   residual image (True) or in an optimal signal-to-noise residual image
                               #   (False)
```

ftmachine = “mosaic” : add in uv plane and invert together, Use *csclean* for deconvolution.

ftmachine = “ft” : shift and add in image plane

There’s a tool (“ia.linearmosaic”) to linear mosaic after cleaning each pointing and to stitch all pointings together entirely in the image domain

Coming in Cycle 5: TCLEAN will replace CLEAN

Why replace clean? Clean has become difficult to maintain, trace down issues, and most importantly to extend to new capabilities, like combining different algorithms together and parallelizing the code efficiently.

Benefits of TCLEAN:

- A more straightforward interface that is more logical, usable and reliable to the users including significant improvements to the logging output and has been made much more homogeneous across algorithms.
- more combinations of imaging algorithms - most important of these for ALMA is the ability to account for spatial spectral index variations (nterms >1) for mosaics.
- Includes algorithms for autoboxing - refactored code has simplified development of a fully integrated autoboxing capability (available for the first time in CASA 5.0, and for deployment in the pipeline for CASA 5.1, Cycle 5)
- Add significant improvements in divergence checks for safeguarding against divergence in the cleaning process

Coming in Cycle 5: TCLEAN

TCLEAN also offers a more straightforward user interface inside CASA and clearer logging output

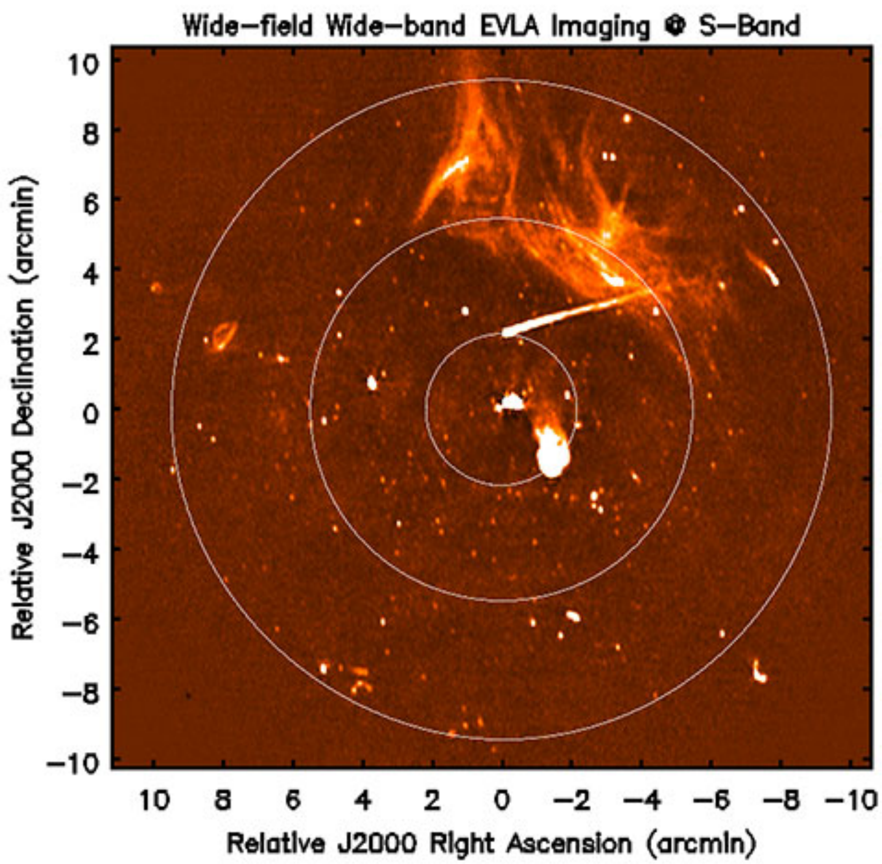
Example Syntax Changes:

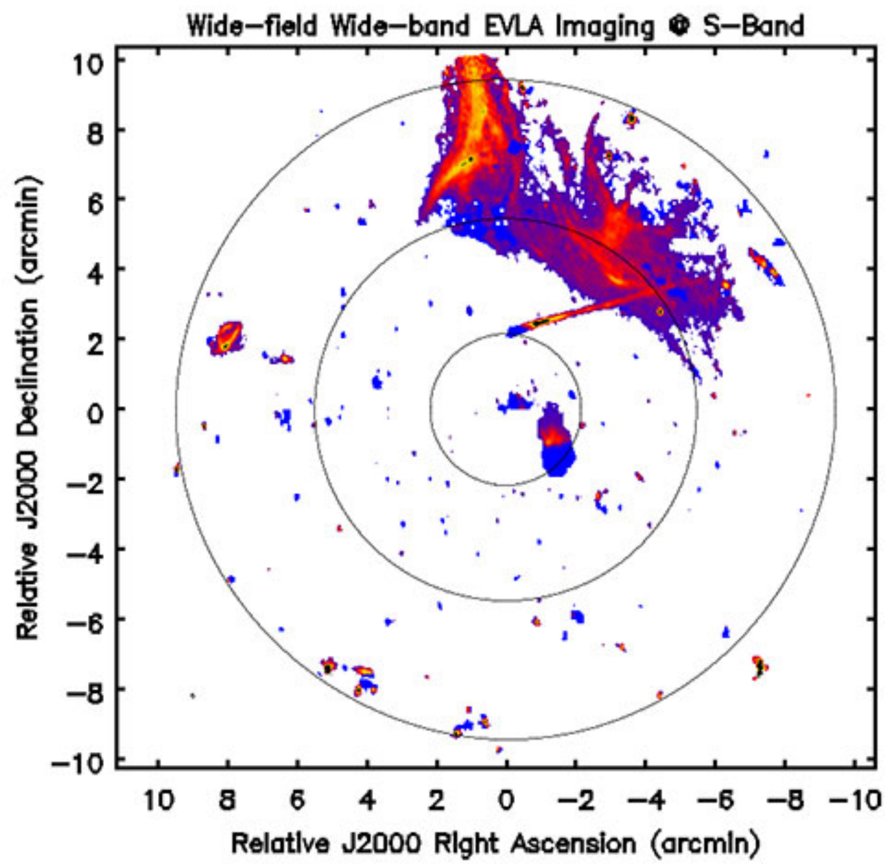
clean	tclean
mode	specmode
resmooth=True	*restoringbeam='common'*
minpb	*pblimit*
gridmode, imagermode	gridder
psfmode	deconvolver
modelimage	startmodel
.flux (output)	.pb (output)

Major syntax and usage changes from clean → tclean are summarized here:
https://casaguides.nrao.edu/index.php/TCLEAN_and_ALMA

Coming in Cycle 5: TCLEAN

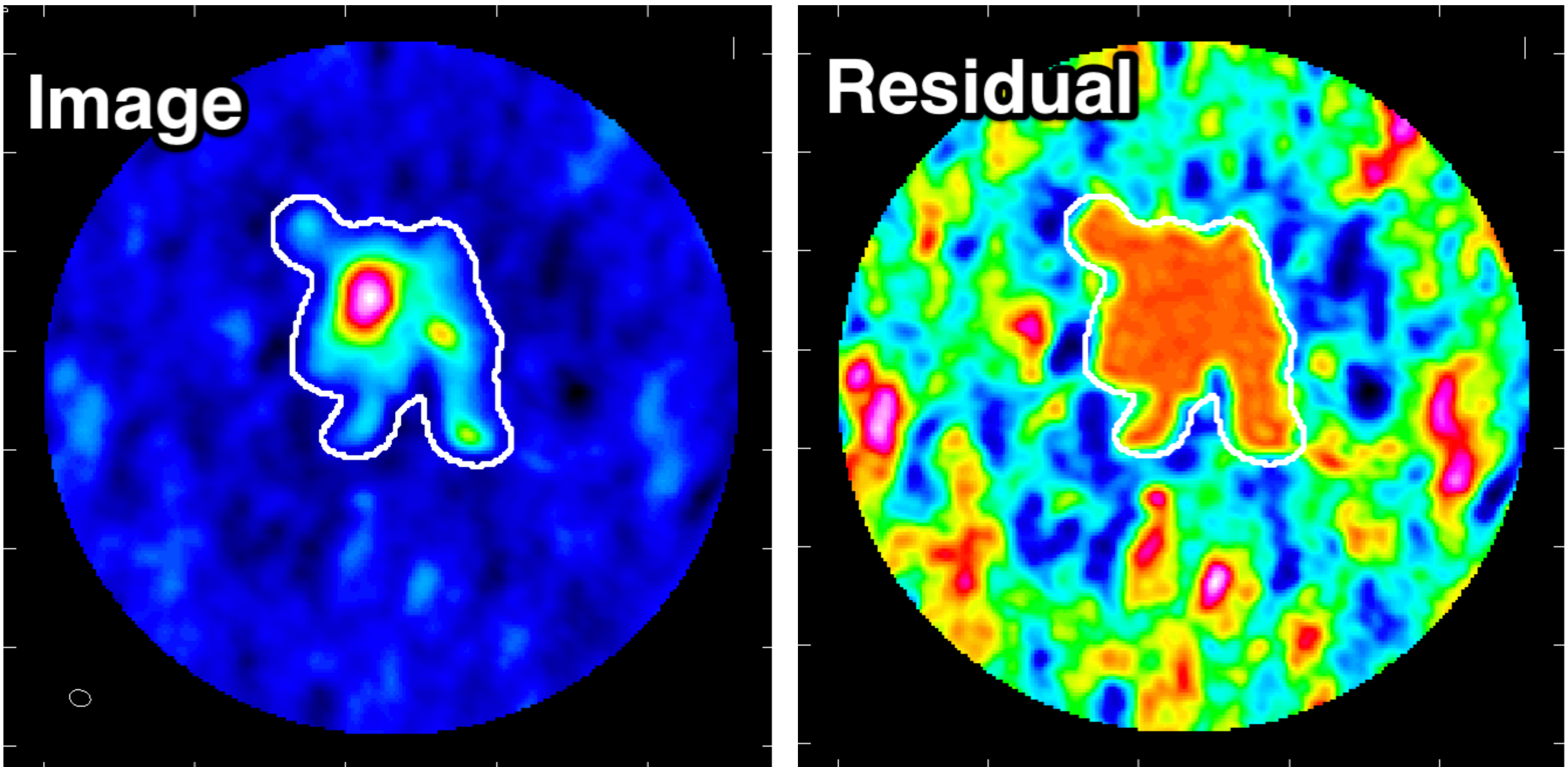
Example (JVLA) application of TCLEAN: Wide field, wide band mosaic imaging with A projection





Coming in Cycle 5: TCLEAN

Also included in TCLEAN: Robust Autoboxing algorithms!



Coming in Cycle 5: TCLEAN

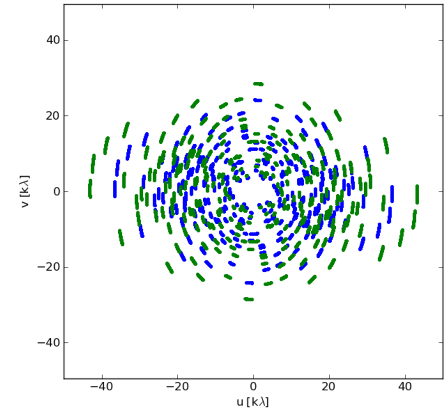
Should I use CLEAN or TCLEAN?

As we transition from clean to tclean, we will be updating all the relevant documentation including the CASA Guides, Scripts and Tutorials… Etc.

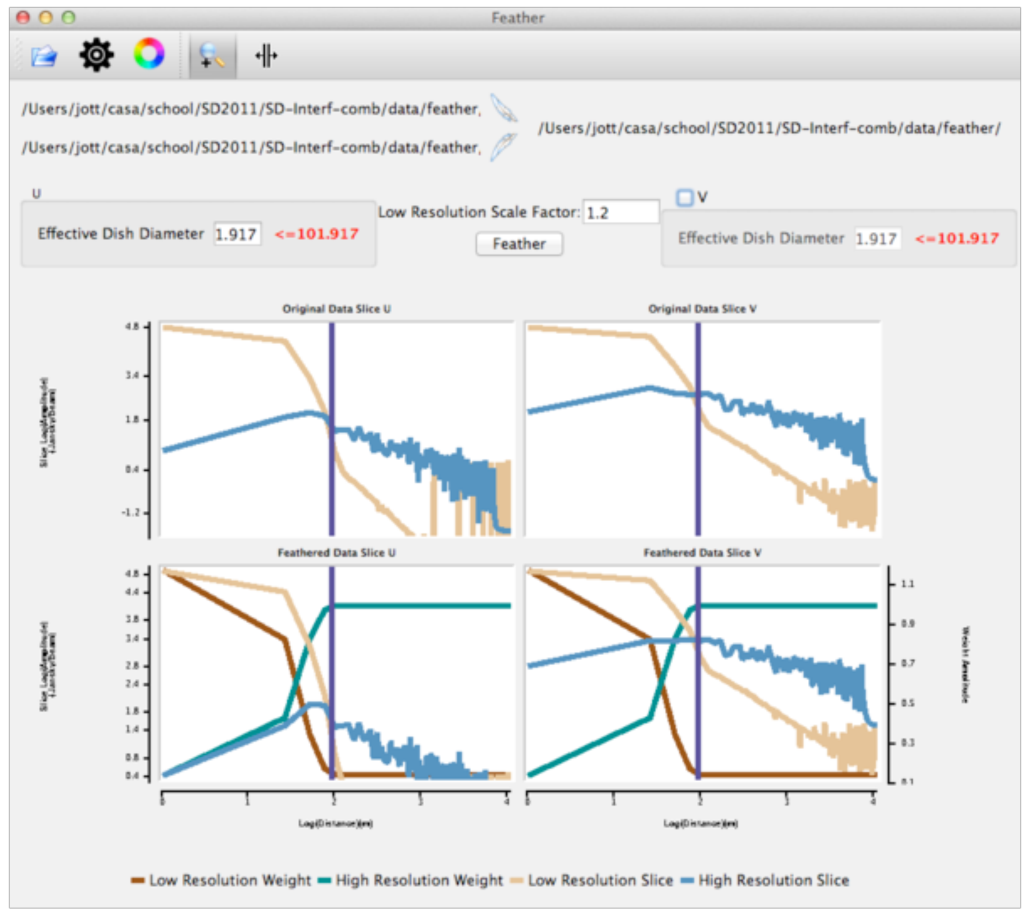
The ALMA Imaging Pipeline uses tclean and calls to tclean are in all the pipeline "hif_" routines.

So we strongly advise people that tclean should start to be used especially as we are getting closer to Cycle 5. There will be no further development or bug fixes in clean!

Combining with single-dish or other interferometric maps

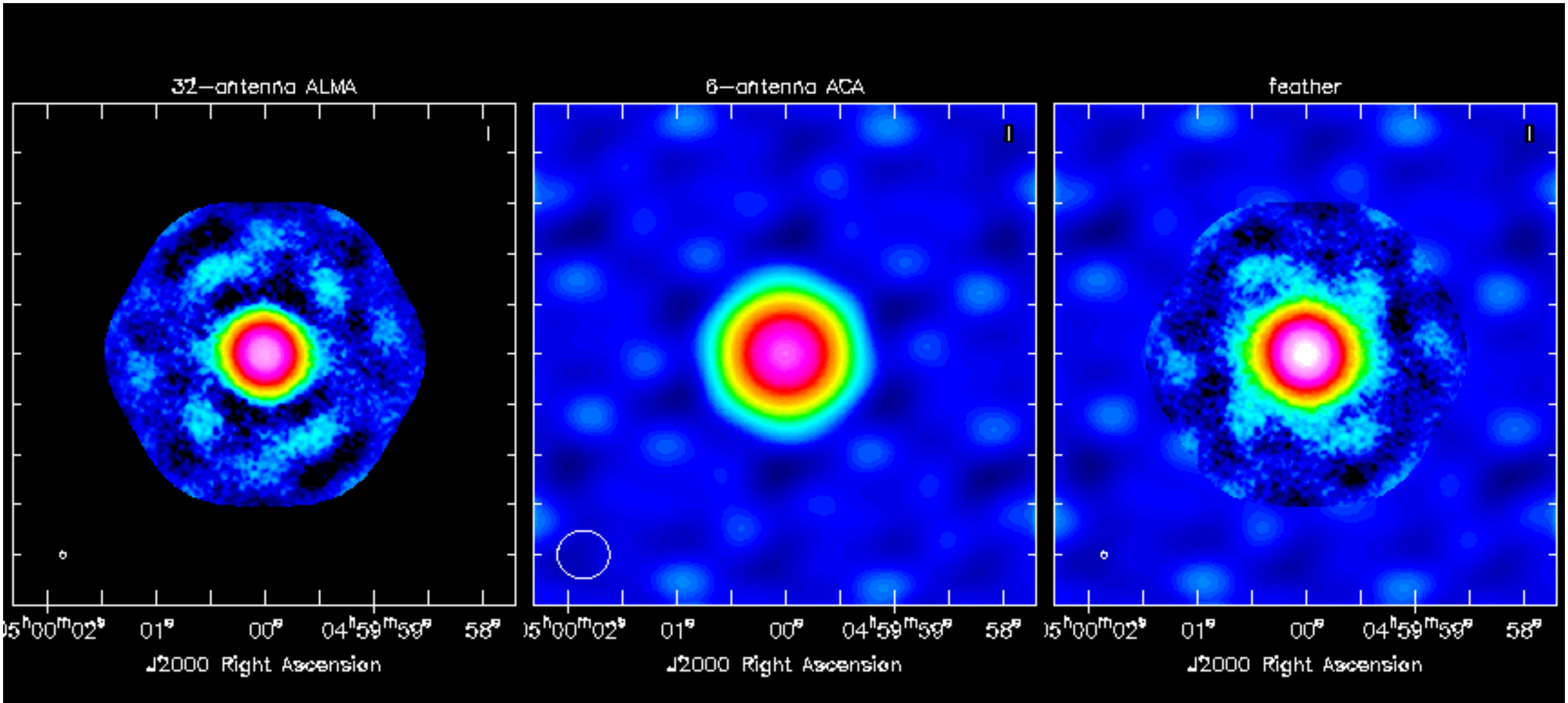


- If you have only images:
  - feather (or "casafeather")
- If you have an image and an MS:
  - use CLEAN with the image as "modelimage"
  - and/or feather
- If you have multiple MS plus an image:
  - Same as above, input to clean will be all the MS



Combining with other data: feather

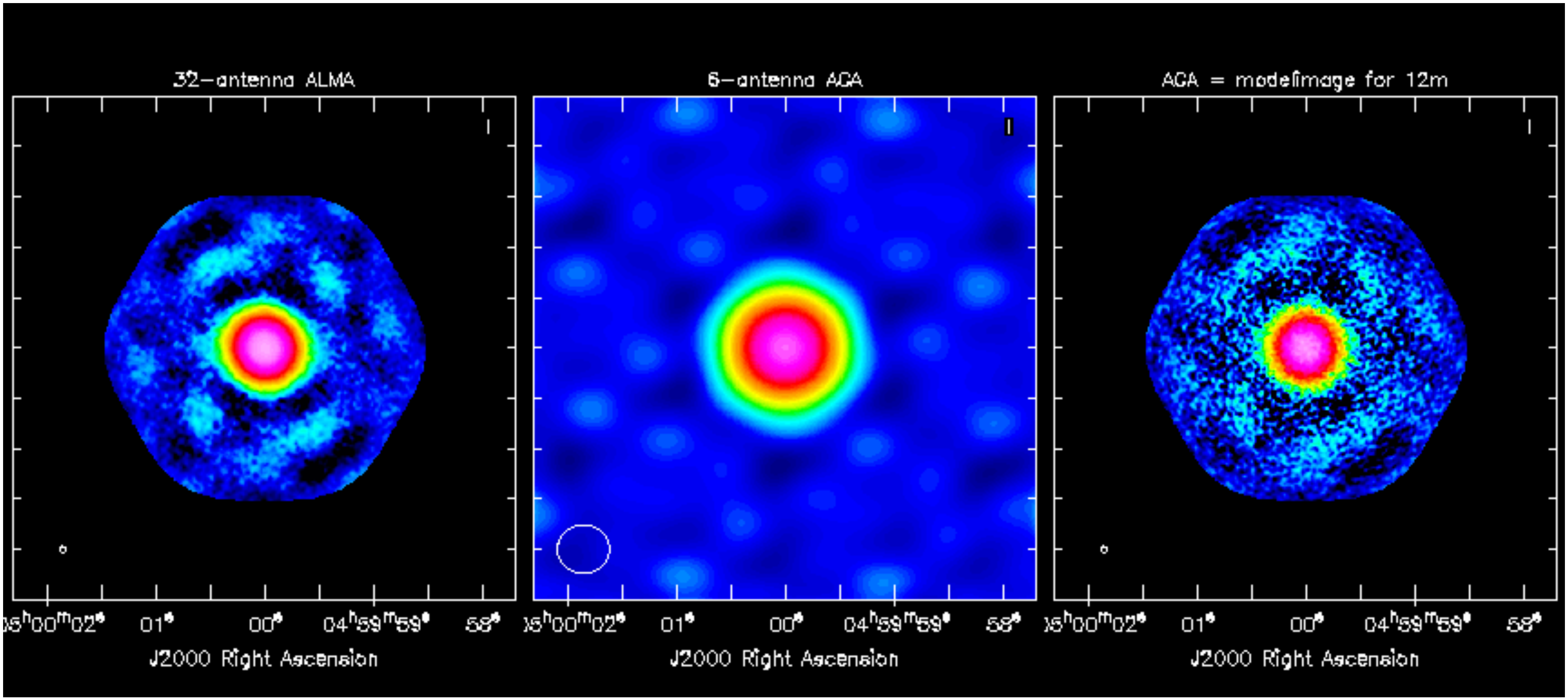
```
#  feather :: Combine two images using their Fourier transforms
imagename          =        ''          #  Name of output feathered image
highres            =        ''          #  Name of high resolution (interferometer) image
lowres             =        ''          #  Name of low resolution (single dish) image
async              =     False          #  If true the taskname must be started using feather(...)
```



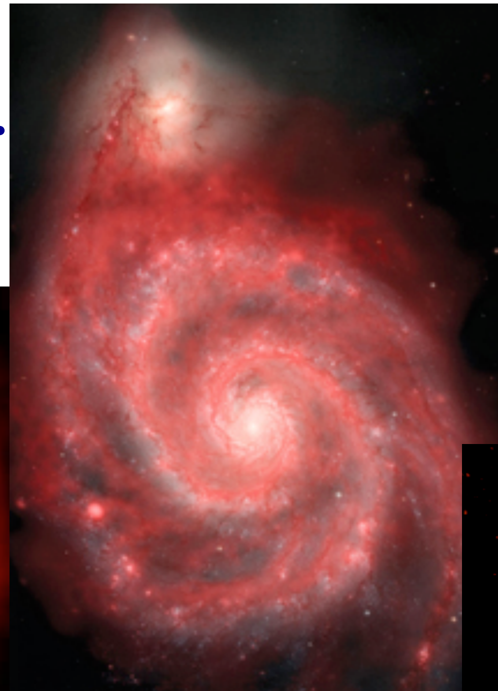
We also have a graphical tool: CASAfeather

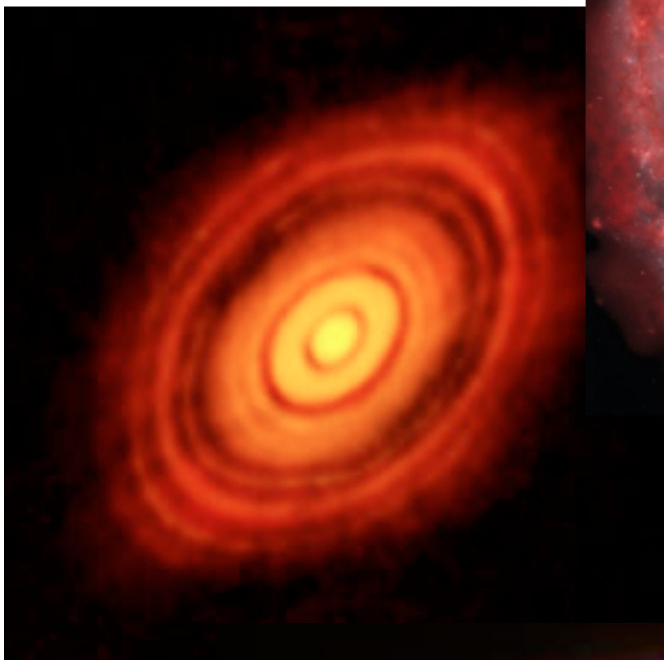
Combining with other data: modelimage

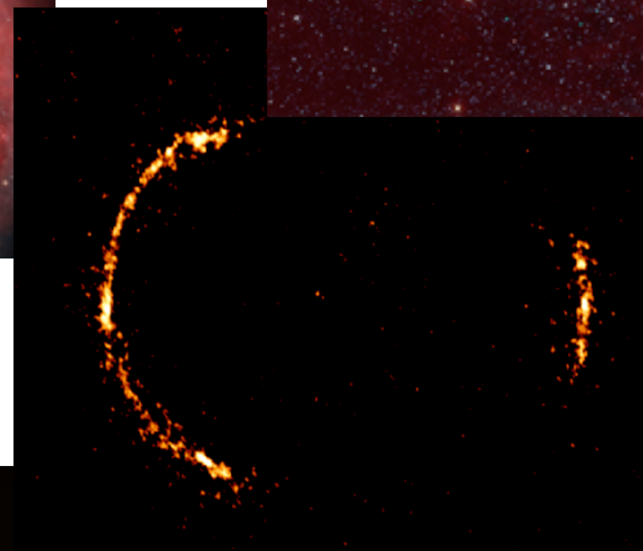
```
--------> inp(clean)
#  clean :: Invert and deconvolve images with selected algorithm
modelimage          =        ''       #  Name of model image(s) to initialize cleaning
```

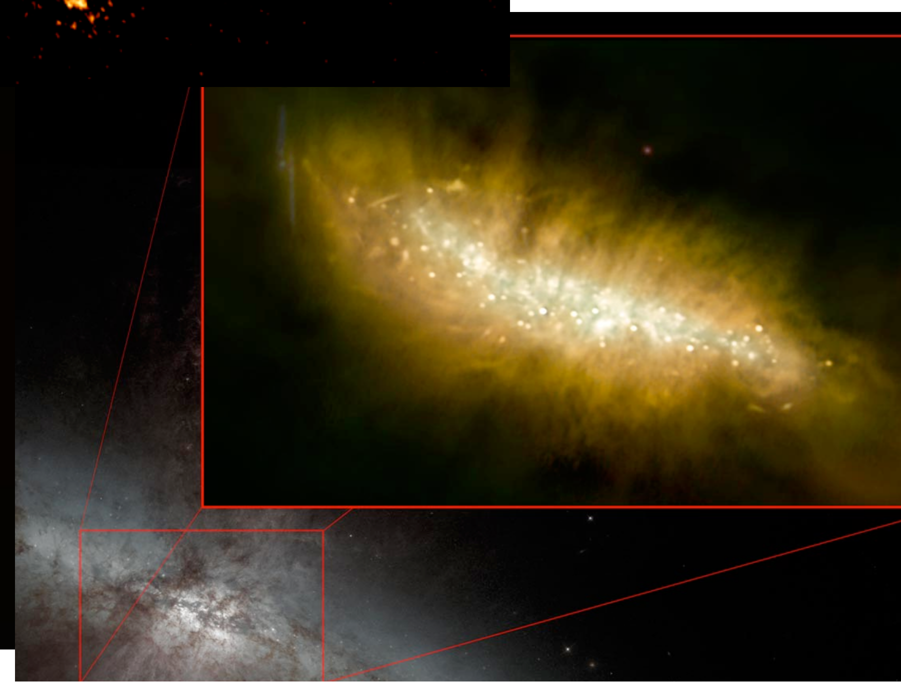


… some CASA images…









Now go get your hands dirty!

ALMAguides

How to use these CASA Tutorials

Imaging Tutorials for CASA beginners

If you are new to CASA, start with the following tutorials. ALMA data are delivered with standard calibrations applied and they are ready for imaging. These guides cover the basic steps required for imaging and self-calibration.

- A first look at imaging in CASA This guide gives a first look at imaging and image analysis in CASA.
- A first look at self-calibration in CASA This guide demonstrates continuum self-cal.
- A first look at spectral line imaging in CASA This guide shows imaging of a spectral line.
- A first look at image analysis in CASA This guide demonstrates moment creation and basic image analysis.

Guides for reducing ALMA Science Verification data

The links below lead to overview pages for each science verification observation. The guides themselves are linked from the overview pages. These guides are a useful tools for those who would like to learn the process of calibration and imaging in detail.

The following ALMA science verification guides have been validated for CASA version 4.3. They should also work for CASA version 4.4, and they will be validated for version 4.4 soon.

- TWHydraBand7: The protoplanetary disk source TW Hya at Band 7 (0.87 mm)
- NGC3256Band3: The galaxy merger NGC 3256 at Band 3 (3 mm)
- AntennaeBand7: Mosaic of the galaxy merger NGC 4038/4039 (Antennae) at Band 7 (0.87 mm)
- IRAS16293Band9: Mosaic of the protostellar cluster IRAS16293-2422 at Band 9 (0.45 mm)
- File:BR1202 SV Band7 Calibration notes.pdf: Supplemental notes on the calibration of Science Verification target BR1202-0725 in CASA 3.3
- ALMA2014_LBC_SVDATA: Imaging scripts and details for the 2014 ALMA Long Baseline Campaign science verification data for Juno, Mira, HL Tau, and SDP.81.
- M100_Band3: Demonstration of combining 12m-array, 7m-array, and Total Power data for M100 using CASA 4.3.1
- 3C286_Polarization: Demonstration of the reduction of ALMA continuum polarization toward the quasar 3C286

A Guide to CASA Data Weights and How to Ensure They are Correct for Data Combination

A Guide to Processing ALMA Data for Cycle 0

This page takes you through the steps of processing Cycle 0 data from the ALMA data archive. The guide describes some helpful hints for downloading the data, and describes the process all the way through imaging and self-calibration, and image analysis.

You can also get a look at example data calibration scripts used for Cycle 0 data at the following links. These were written for CASA version 3.4.

- TDM (128 channels/spw) File:TDM.example.ms.scriptForCalibration.py
- FDM (3840 channels/spw) File:FDM.example.ms.scriptForCalibration.py
- If you need to update 3.4 scripts to 4.2, see more information here

A Tutorial for Simulating ALMA Data.

Start here to learn about simulations. The CASA 4.3 simulation examples in the above tutorial should also work for version 4.4, and they will be validated for version 4.4 soon. Jump directly to the simulations examples with the following links.

- Simulation Examples in CASA 4.3
- Examples for older versions of CASA: 4.2 4.1 4.0 3.4 3.3