ALMA Data Reduction Training Day, 27 Nov. 2023

Imaging

Ko-Yun (Monica) Huang
ALMA Local Expertise Group (Allegro)

EUROPEAN ARC
ALMA Regional Centre

ALMA

ESO

Universiteit Leiden
Sterrewacht Leiden

Software used in this tutorial

- This tutorial uses **CASA 6.6.0**
- Obtain CASA at casa.nrao.edu
    - CASA 6.6 as “Latest Release”
- CASA documentation & general help available in the CASA Docs:
  casadocs.readthedocs.io
- **CASA Paper:**
  “CASA, the Common Astronomy Software Applications for Radio Astronomy,” the CASA Team
  arXiv: 2210.02276
- AnalysisUtils package, check Analysis Utils CASA Guide for more information:
  https://casaguides.nrao.edu/index.php/Analysis_Utilities

```
CASA <x>: sys.path.append('/lustre2/allegro/lib/jao-mirror/AIV/science/analysis_scripts/')

CASA <x>: import analysisUtils as au
```

Data used in this tutorial

- TW Hya from the "**First Look at Line Imaging CASA 6**" guide, **N_2H^+ J=4-3**
- ALMA Project 2011.0.00340.S, "Searching for H2D+ in the disk of TW Hya v1.5", PI Chunhua Qi

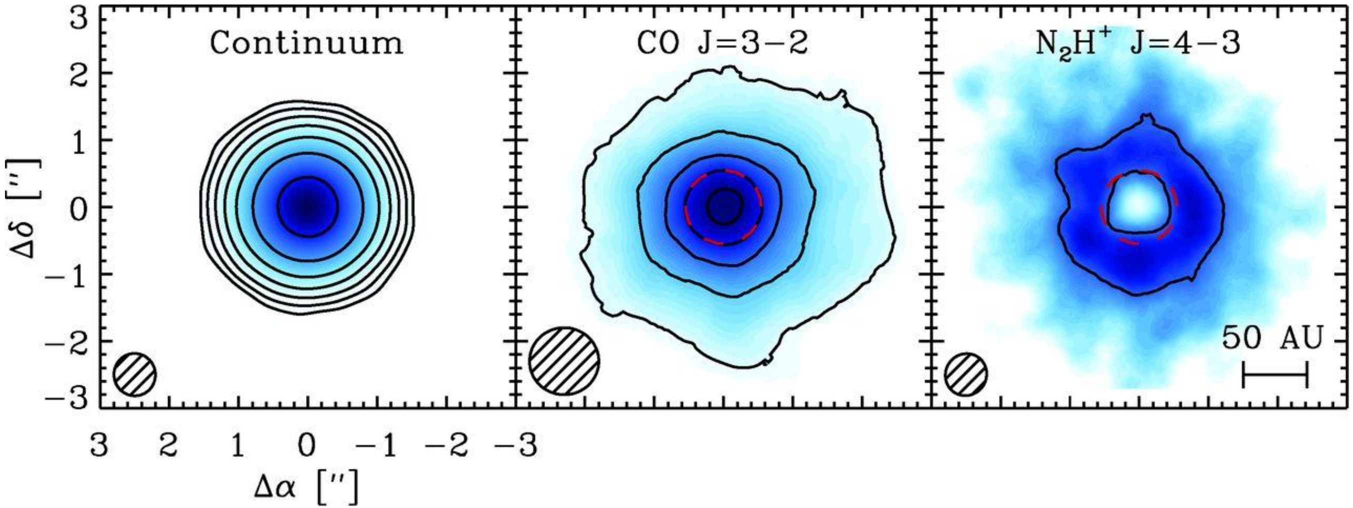


Fig. 1 from *Qi et al., 2013, Science,* **341**, 6146, 630

tclean: CASA task for Radio Interferometric Image Reconstruction

```
CASA <x>: inp tclean
vis                = ''               # Name of input visibility file(s)
selectdata         = True             # Enable data selection parameters
    ...
specmode           = 'mfs'            # Spectral definition mode
    ...
gridder            = 'standard'       # Gridding options
    ...
deconvolver        = 'hogbom'         # Minor cycle algorithm
    ...
weighting          = 'natural'        # Weighting scheme
    ...
niter              = 0                # Maximum number of iterations
    ...
usemask            = 'user'           # Type of mask(s) for deconvolution
    ...
```

Synthesis Imaging

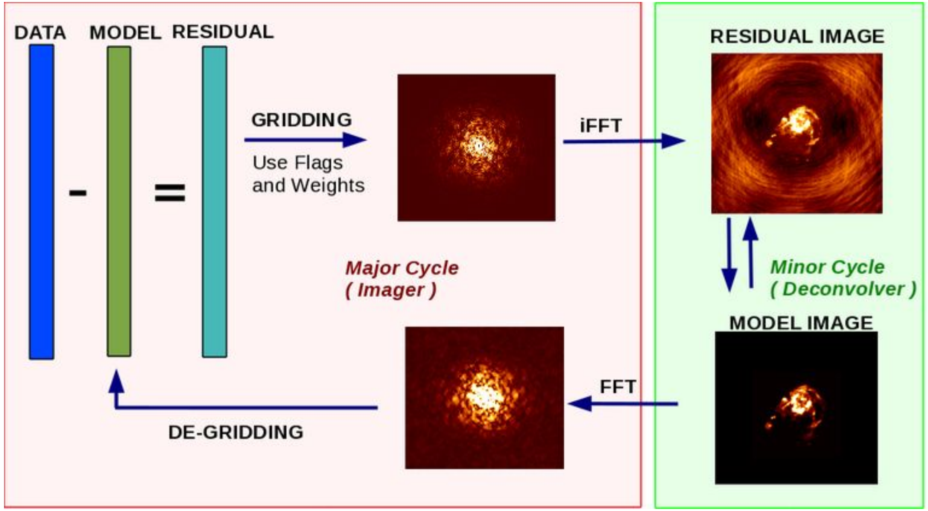


Figure from CASA Docs

Major Cycle: *visibility* frame

1. Subtract model from data to generate residual visibilities
2. Grid residual visibilities, iFFT into image frame to produce residual image

Minor Cycle: image frame

Use specified clean algorithm (e.g. Högbom, multiscale, etc.) to generate model of source

Source model components convolved with PSF and subtracted from residual image

Back to Major Cycle: *visibility* frame

3. FFT model image into *uv* frame, de-grid visibilities

Before you clean: Imaging Preparation

1. Inspect your data
   a. **Inspect the weblog** - see I-TRAIN #4: ALMA WebLog inspection
   b. `listobs`: spw & field information
   c. `plotms`: check *uv* coverage, check for spectral lines, telluric lines, etc.
2. Prepare your measurement set
   a. If needed, `split` science source from calibrated measurement set(s)
   b. If continuum imaging, optional: `split` out continuum-only MS
   c. If line imaging: *uv* continuum subtraction with `uvcontsub` if continuum > 3 sigma per channel

Resources:

- ALMAGuides & General Imaging Tutorials: casaguides.nrao.edu
- Video tutorial on imaging: https://youtu.be/yuLKAfroHu4

Material preparation & Let’s launch CASA!

In YOUR analysis directory (analysis/USERNAME/):

mkdir imaging

cd imaging

cp -r ../../../scripts/Imaging_*.py ./

Scripts involved in this tutorial:

(a) Imaging_1_basic.py - setting up all tclean parameter in interactive style

(b) Imaging_2_basic_scripted.py - scripted version of (a)

(c) Imaging_3_uvcontsub.py - basic steps to perform continuum subtraction

(d) Imaging_4_uvtaper_scripted.py - tclean script including tapering specification (uvtaper and the associated parameters)

(you can copy below from the script “Imaging_1_basic.py”)

cp -r ../../../archive/DRT2023/TW_hydra/sis14_twhya_calibrated_flagged.ms.contsub/ ./

nice +10 env -u PYTHONPATH -u LD_LIBRARY_PATH casapy-660

tclean: data selection

```
CASA <x>: inp tclean
vis                     = 'sis14_twhya_calibrated_flagged.ms.contsub'
selectdata              = True          # Enable data selection parameters
   field                = 'TW Hya'      # field(s) to select
   spw                  = '0'           # spw(s)/channels to select
   timerange            = ''            # Range of time to select from data
   uvrange              = ''            # Select data within uvrange
   antenna              = ''            # Select data based on antenna/baseline
   scan                 = ''            # Scan number range
   observation          = ''            # Observation ID range
   intent               = ''            # Scan Intent(s)
     ...
```

tclean: get spw & field information from `listobs`

```
CASA <x>: vis = 'sis14_twhya_calibrated_flagged.ms.contsub'
CASA <x>: listobs(vis)
CASA <x>: listobs(vis, listfile='listobs.txt')
```

- View `listobs` output in logger or in file
- Information includes:
  - Observation information (time, field observed, intents)
  - Field information (field IDs, coordinates)
  - Spectral window information (spw IDs, frequencies, bandwidth, spectral resolution)
  - Antenna information (Names, stations, dish diameter, coordinates)

tclean: get spw & field information from `listobs`

Output:

```
================================================================================
          MeasurementSet Name:  /yourDirectory/sis14_twhya_calibrated_flagged.ms.contsub      MS Version 2
================================================================================
   Observer: cqi     Project: uid://A002/X327408/X6f
Observation: ALMA
Data records: 53161       Total elapsed time = 4268.11 seconds
   Observed from   19-Nov-2012/07:56:23.5   to   19-Nov-2012/09:07:31.6 (UTC)

   ObservationID = 0         ArrayID = 0
  Date        Timerange (UTC)          Scan  FldId FieldName             nRows     SpwIds   Average Interval(s)    ScanIntent
  19-Nov-2012/07:56:23.5 - 08:02:11.3    12      0 TW Hya                     8514  [0]  [6.05] [OBSERVE_TARGET#ON_SOURCE]
              08:08:09.6 - 08:13:57.3    16      0 TW Hya                    10360  [0]  [6.05] [OBSERVE_TARGET#ON_SOURCE]
              08:19:53.9 - 08:25:41.7    20      0 TW Hya                    10321  [0]  [6.05] [OBSERVE_TARGET#ON_SOURCE]
              08:32:00.5 - 08:37:48.2    24      0 TW Hya                    10324  [0]  [6.05] [OBSERVE_TARGET#ON_SOURCE]
              08:43:45.6 - 08:49:33.4    28      0 TW Hya                     9462  [0]  [6.05] [OBSERVE_TARGET#ON_SOURCE]
              09:05:15.6 - 09:07:31.6    36      0 TW Hya                     4180  [0]  [6.05] [OBSERVE_TARGET#ON_SOURCE]
           (nRows = Total number of rows per scan)
Fields: 1
  ID   Code Name               RA               Decl           Epoch   SrcId      nRows
  0    none TW Hya             11:01:51.796000 -34.42.17.36600 J2000   0          53161
Spectral Windows:  (1 unique spectral windows and 1 unique polarization setups)
  SpwID  Name                           #Chans   Frame   Ch0(MHz)  ChanWid(kHz)  TotBW(kHz) CtrFreq(MHz) BBC Num  Corrs
  0      ALMA_RB_07#BB_2#SW-01#FULL_RES    384   TOPO  372533.086       610.352    234375.0 372649.9688        2  XX  YY

      ...
```

tclean: get spw & field information from `listobs`

Output:

```
================================================================================
           MeasurementSet Name:  /yourDirectory/sis14_twhya_calibrated_flagged.ms.contsub      MS Version 2
================================================================================
   Observer: cqi     Project: uid://A002/X327408/X6f
Observation: ALMA
Data records: 53161       Total elapsed time = 4268.11 seconds
   Observed from   19-Nov-2012/07:56:23.5   to   19-Nov-2012/09:07:31.6 (UTC)

   ObservationID = 0         ArrayID = 0
  Date        Timerange (UTC)          Scan  FldId FieldName             nRows     SpwIds   Average Interval(s)    ScanIntent
  19-Nov-2012/07:56:23.5 - 08:02:11.3    12      0 TW Hya                 8514  [0]  [6.05] [OBSERVE_TARGET#ON_SOURCE]
              08:08:09.6 - 08:13:57.3    16      0 TW Hya                10360  [0]  [6.05] [OBSERVE_TARGET#ON_SOURCE]
              08:19:53.9 - 08:25:41.7    20      0 TW Hya                10321  [0]  [6.05] [OBSERVE_TARGET#ON_SOURCE]
              08:32:00.5 - 08:37:48.2    24      0 TW Hya                10324  [0]  [6.05] [OBSERVE_TARGET#ON_SOURCE]
              08:43:45.6 - 08:49:33.4    28      0 TW Hya                 9462  [0]  [6.05] [OBSERVE_TARGET#ON_SOURCE]
              09:05:15.6 - 09:07:31.6    36      0 TW Hya                 4180  [0]  [6.05] [OBSERVE_TARGET#ON_SOURCE]
           (nRows = Total number of rows per scan)
Fields: 1
  ID   Code Name                RA               Decl           Epoch   SrcId      nRows
  0    none TW Hya              11:01:51.796000 -34.42.17.36600 J2000   0          53161
Spectral Windows:  (1 unique spectral windows and 1 unique polarization setups)
  SpwID  Name                           #Chans   Frame   Ch0(MHz)  ChanWid(kHz)  TotBW(kHz) CtrFreq(MHz) BBC Num  Corrs
  0      ALMA_RB_07#BB_2#SW-01#FULL_RES    384   TOPO  372533.086       610.352    234375.0 372649.9688        2  XX  YY
```

...

tclean: data selection

```
CASA <x>: inp tclean
vis                 = 'sis14_twhya_calibrated_flagged.ms.contsub'
selectdata          = True         # Enable data selection parameters
   field            = 'TW Hya'     # field(s) to select
   spw              = '0'          # spw(s)/channels to select
   timerange        = ''           # Range of time to select from data
   uvrange          = ''           # Select data within uvrange
   antenna          = ''           # Select data based on antenna/baseline
   scan             = ''           # Scan number range
   observation      = ''           # Observation ID range
   intent           = ''           # Scan Intent(s)
    ...
```

tclean: image parameters

```
CASA <x>: inp tclean
    ...
datacolumn             = 'data'            # Data column
imagename              = 'twhya_n2hp43'    # Pre-name of output images
imsize                 = [240,240]         # Number of pixels
cell                   = '0.1arcsec'       # Cell size
phasecenter            = 0                 # Phase center of the image
stokes                 = 'I'               # Stokes Planes to make
projection             = 'SIN'             # Coordinate projection
startmodel             = ''                # Name of starting model
    ...
```

tclean: determine `cell` size

- Rule of thumb: ~5-8 cells across beam

1. Use `plotms`:

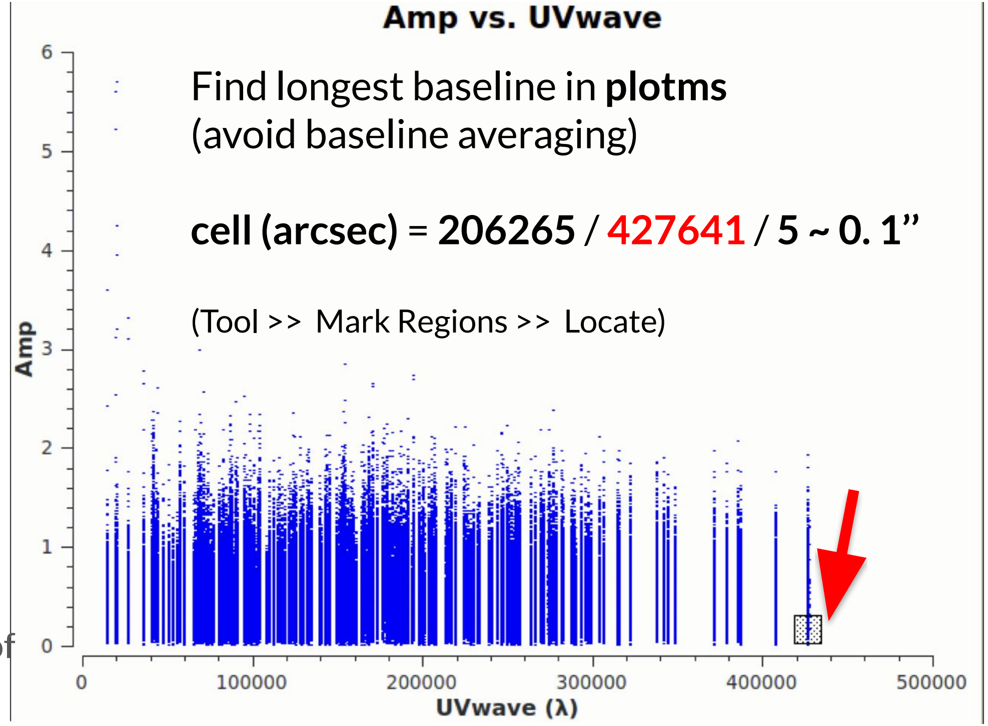
```
CASA <x>: plotms(vis = vis,
      xaxis = 'UVwave',
      field = 'TW Hya', spw = '0',
      avgtime = '1e6', avgscan = True)
```

Calculate: **cell (arcsec)** = **206265** /
(longest baseline in wavelengths) /
(# pixels across beam)

Use averaging in `plotms`, esp. for large datasets!

2. Use AnalysisUtils - does not account for projection of baselines:

```
CASA <x>: au.pickCellSize(vis)
Out[x]:  0.12
```



tclean: determine `imsize`

Rules of thumb:

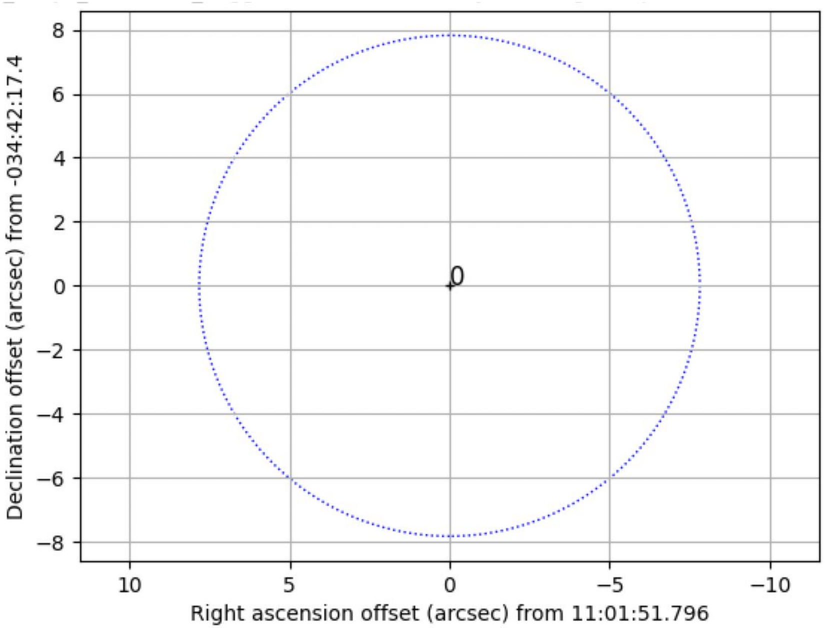
- image > PB extent (especially non-point sources)
- Select symmetrical values
- Round up to nearest 10 or 100 pixels
- CASA will tell (in logger) if it doesn't like choice
- **HPBW Primary Beam (FOV) = 1.02 λ / D**

1. **Get imsize in pixels (note cell size):**

```
CASA <x>: au.pickCellSize(vis,imsize=True)
Out[x]: [0.12, [200, 200], 0]
```

2. **Plot mosaic:**

```
CASA <x>: au.plotmosaic(vis,
        sourceid='TW Hya',
        coord='rel',
        figfile='twhya_pointings_rel.png')
```



tclean: image parameters

```
CASA <x>: inp tclean
vis                 = 'sis14_twhya_calibrated_flagged.ms.contsub'
selectdata          = True
   field            = 'TW Hya'                  # field(s) to select
   spw              = '0'                       # spw(s)/channels to select
   timerange        = ''                        # Range of time to select from data
   uvrange          = ''                        # Select data within uvrange
   antenna          = ''                        # Select data based on antenna/baseline
   scan             = ''                        # Scan number range
   observation      = ''                        # Observation ID range
   intent           = ''                        # Scan Intent(s)
datacolumn          = 'data'                     # Data column to image(data,corrected)
imagename           = 'twhya_n2hp43'             # Pre-name of output images
imsize              = [240,240]                  # Number of pixels
cell                = '0.1arcsec'                # Cell size
phasecenter         = 0                          # Phase center of the image
stokes              = 'I'                       # Stokes Planes to make
projection          = 'SIN'                     # Coordinate projection
startmodel          = ''                        # Name of starting model image
     ...
```

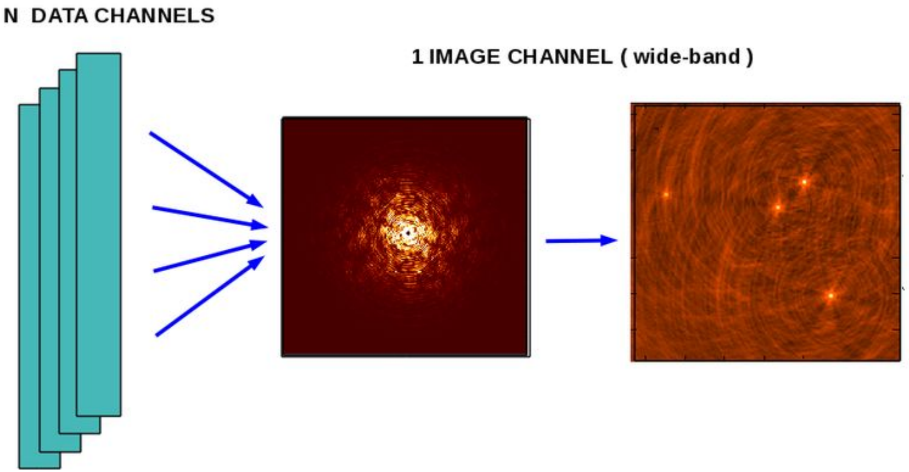
Spectral Mode

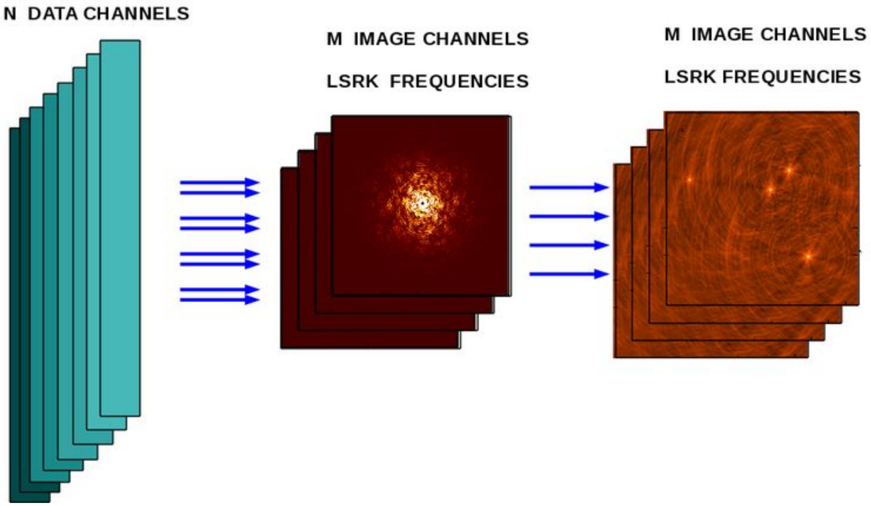
The spectral mode determines whether all channels are:

I. synthesized into a single image using **`specmode = 'mfs'`** (multi-frequency synthesis), i.e. continuum imaging

or

II. imaged separately using **`specmode = 'cube'`**, i.e. spectral line imaging





Figures from CASA Docs

Spectral Modes

For this tutorial, we use:

```
CASA <x>: specmode = 'cube'
```

- `nchan, start,` and **`width`** can be in terms of channel number, frequency, or velocity

```
CASA <x>: nchan = 30
CASA <x>: start = 230
CASA <x>: width = 1
```

- for ***z<0.2***, can use rest frequency of line (look up with e.g. Splatalogue)

```
CASA <x>: restfreq = '372.67250900GHz' # N2H+ J=4-3
```

- Set velocity parameters:

```
CASA <x>: outframe = 'lsrk' # LSR as a kinematical (radio) definition
CASA <x>: veltype = 'radio' # produces channels of fixed velocity width
```

- See CASA Docs for more options and precise definitions

Spectral Modes

```
CASA <x>: inp tclean
    ...
specmode               = 'cube'              # Spectral definition mode (mfs...
   nchan               = 30                  # Number of channels...
   start               = 230                 # First channel (e.g. start=3...
   width               = 1                   # Channel width (e.g. width=2...
   outframe            = 'lsrk'              # Spectral reference frame...
   veltype             = 'radio'             # Velocity type (radio...
   restfreq            = '372.67250900GHz'   # List of rest frequencies
   interpolation       = 'linear'            # Spectral interpolation...
   perchanweightdensity = True               # whether to calculate weight…
    ...
```

Gridder

The gridder resamples imaging weights and weighted visibilities onto a uniform *uv* grid

Recommended:

- gridder = 'standard'
  - operations applied in image-domain to correct for direction-dependent effects
  - use for single pointings
- gridder = 'mosaic'
  - direction-dependent, time-variable and baseline-dependent corrections during gridding in the visibility-domain
  - use for mosaics

For this tutorial:

CASA <x>: gridder = 'standard'

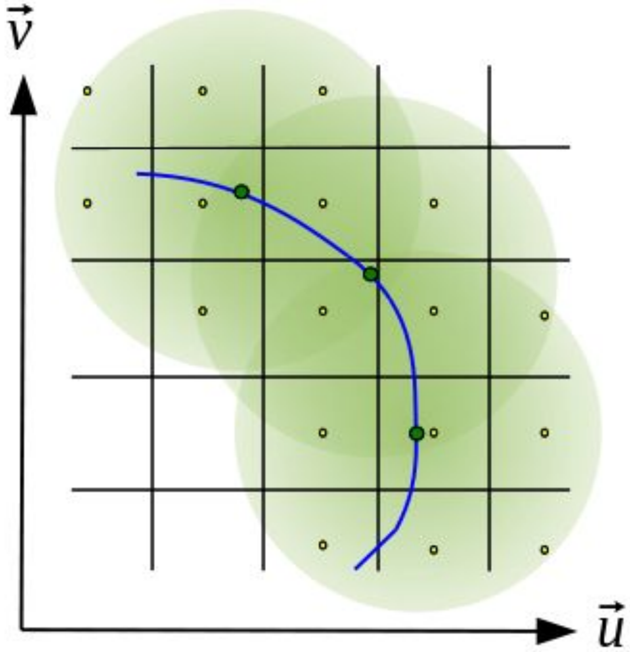


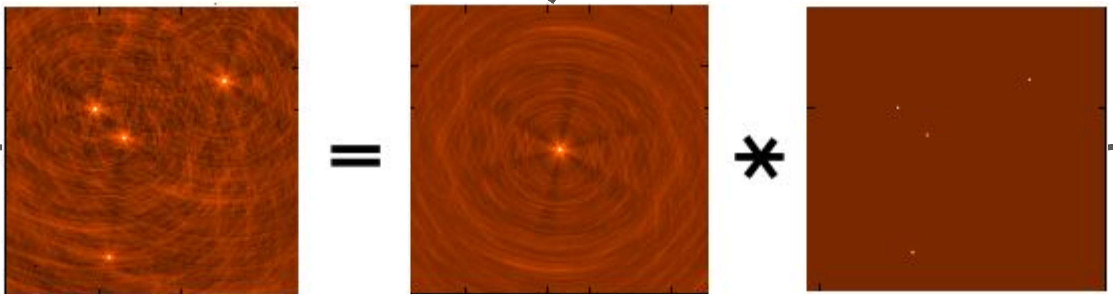
Figure from CASA Docs

Minor-cycle clean algorithms

tclean subparameter: `deconvolver`

'Dirty map' of source brightness distribution is convolved with telescope PSF or **'dirty beam'**

Assumption: sources in radio sky can be **modeled** by multiple **point sources** and/or **Gaussians**



Dirty map **Dirty beam** **Model**

Clean uses an iterative method to **deconvolve** dirty beam from dirty source brightness distribution

Figure from CASA Docs

Minor-cycle clean algorithms

Recommended:

- `deconvolver = 'hogbom'`: adapted version of Hogbom Clean [Hogbom, 1974]
  - assumes point source model of source brightness distribution
    → most appropriate for fields of isolated point sources
  - compute intensive
- `deconvolver = 'clark'` (or `'clarkstokes'`): adapted version of Clark Clean [Clark, 1980]
  - also assumes point source model of source brightness distribution
  - uses smaller patch of PSF in residual image updates → faster than Hogbom
- `deconvolver = 'multiscale'` (or `'mtmfs'`): MultiScale Clean [Cornwell, 2008]
  - scale-sensitive clean, can specify multiple scales
  - assuming sources extended, tapered ‘paraboloids’
  - `scales = []`: list of scales (in pixels)
    - use scales up to the smaller of the largest extent of the emission
    - recommended to include a point source scale (pixel size 0)
  - `smallscalebias = 0.0`: value from -1 (biases towards larger scales) to 1 (biases towards smaller scales)

For this tutorial:

```
CASA <x>: deconvolver = 'multiscale'
CASA <x>: scales = [0,5,10]
```

Weighting Schemes

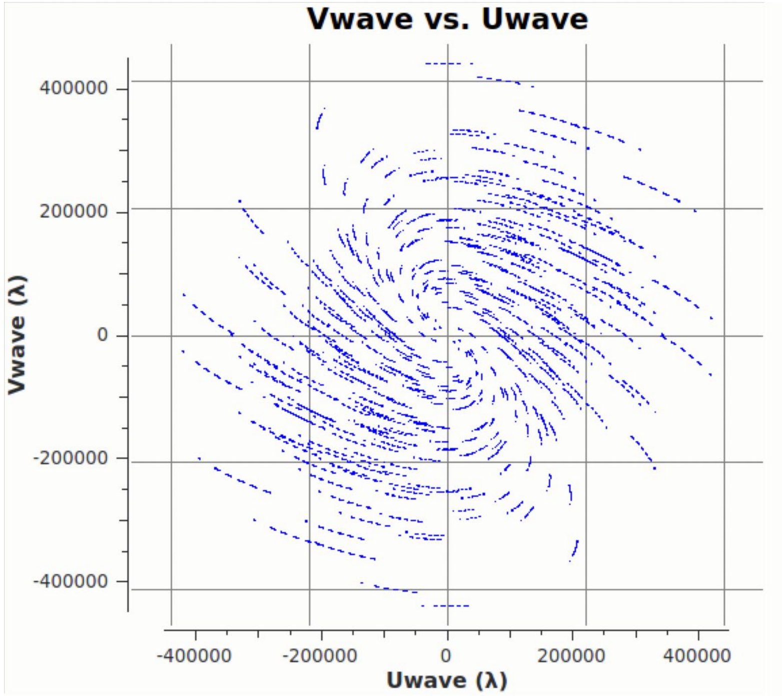
Visibility weights alter the synthesised beam and dynamic range of output image

`weighting = 'natural'`
- visibilities are weighted by data weights
- lower rms noise, lower resolution

`weighting = 'uniform'`
- Visibilities in same *uv* cell are weighted ‘uniformly’
- reduces sidelobes, higher rms noise

`weighting = 'briggs'`
- Compromise between natural & uniform
- **`robust`** parameter can be adjusted from -2 (uniform-like) to 2 (natural-like)

`uvtaper = []`
- Applies a Gaussian taper in addition to the weighting scheme
- Only outertaper → can clip inner *uv* data using **`uvrange`**
- Should use with **`natural`** or briggs with **`robust = 2`**



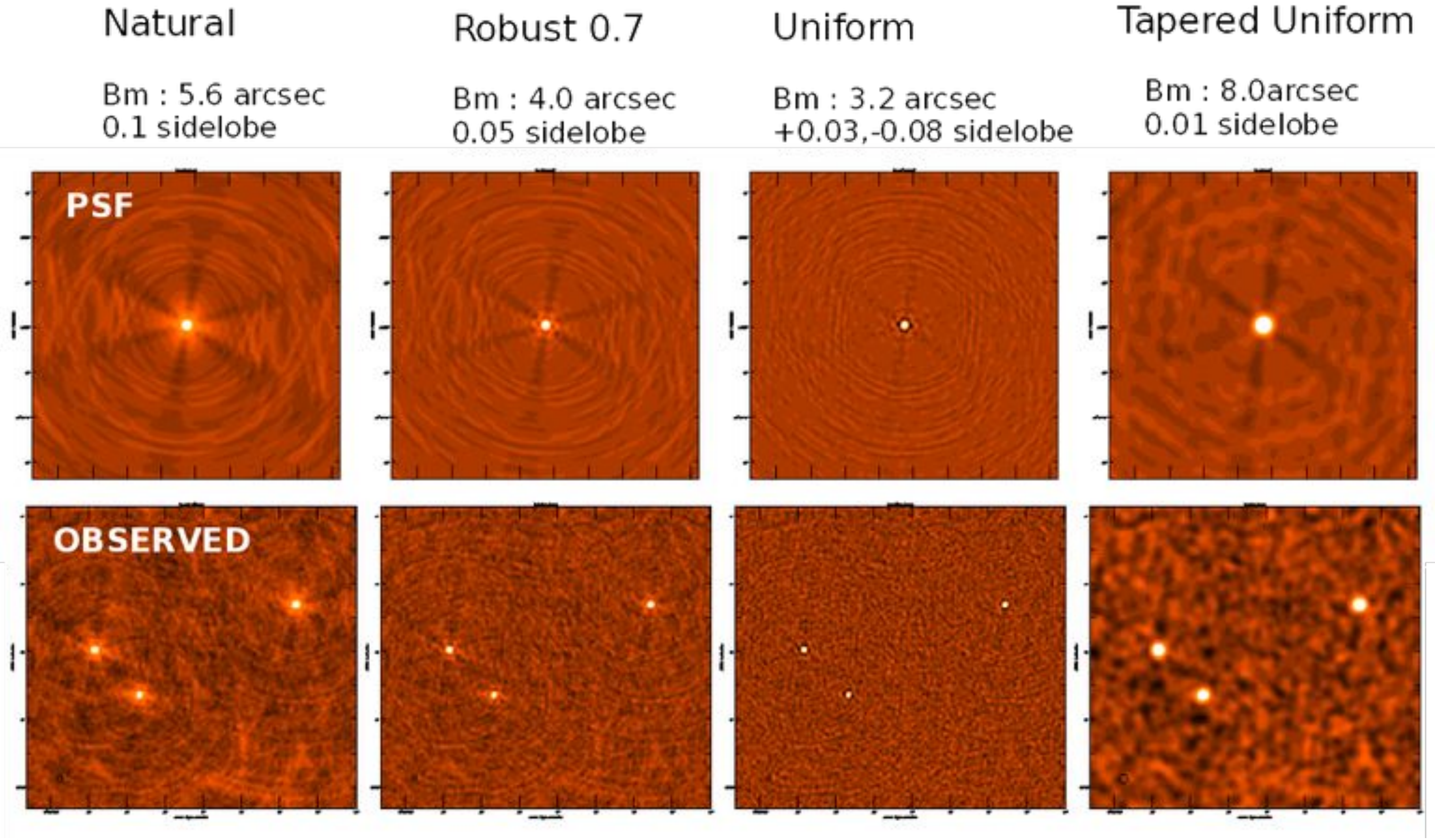


Figure from CASA Docs

Weighting Schemes

For this tutorial we will use:

```
CASA <x>: weighting = 'briggs'
CASA <x>: robust = 0.5

CASA <x>: inp tclean
gridder             = 'standard'       # Gridding options...
   vptable          = ''               # Name of Voltage Pattern table
   pblimit          = 0.2              # PB gain level…
deconvolver         = 'multiscale'     # Minor cycle algorithm...
   scales           = [0, 5, 10]       # List of scale sizes (in pixels)
   smallscalebias   = 0.0              # Biases the scale...
     ...
weighting           = 'briggs'         # Weighting scheme
   robust           = 0.5              # Robustness parameter
   npixels          = 0                # Number of pixels to determine uv-cell
   uvtaper          = []               # uv-taper on outer baselines in uv-plane
     ...
```

Masks for Deconvolution

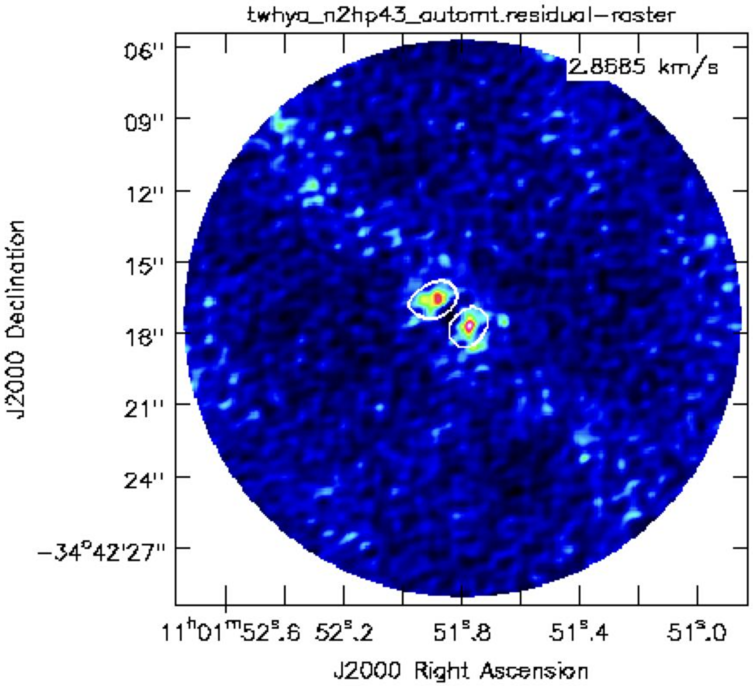
Masks are used to restrict the regions over which clean components are found

`usemask = 'user'`

- this option can be selected to define regions by hand in the GUI when using `interactive = True`
- Alternatively, the `mask` subparameter can be specified as an image file, a region file, or a region string

`usemask = 'auto-multithresh'`

- Available in CASA versions 5.1 and later
- Makes masking spectral line emission easier and faster
- **"AUTO-MULTITHRESH: A General Purpose Automasking Algorithm"**
  Kepley et al., 2020 PASP **132** 024505
- Automasking Guide:
  casaguides.nrao.edu/index.php/Automasking_Guide



Masks for Deconvolution

For this tutorial we will use:

```
CASA <x>: usemask = 'auto-multithresh'

CASA <x>: inp tclean
     ...
usemask                 = 'auto-multithresh'        # Type of mask(s)
   pbmask               = 0.2                       # primary beam mask
   sidelobethreshold    = 2.0                       # sidelobethreshold *...
   noisethreshold       = 4.25                      # noisethreshold * ...
   lownoisethreshold    = 1.5                       # lownoisethreshold * ...
   negativethreshold    = 0.0                       # negativethreshold * ...
   smoothfactor         = 1.0
   minbeamfrac          = 0.3                       # minimum beam fraction ...
   cutthreshold         = 0.01
   growiterations       = 75
   dogrowprune          = True
   minpercentchange     = -1.0
   verbose              = False
     ...
```

1. Set initial source mask

Input parameters:

- `noisethreshold`
- `sidelobethreshold`

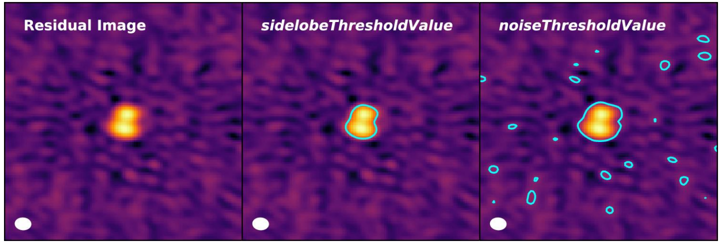


Fig. 2, Kepley+2020

- Automasking picks largest of the two thresholds
- In this example: sidelobeThresholdValue > noiseThresholdValue

2. Prune regions

Input parameters:

- `minbeamfrac`

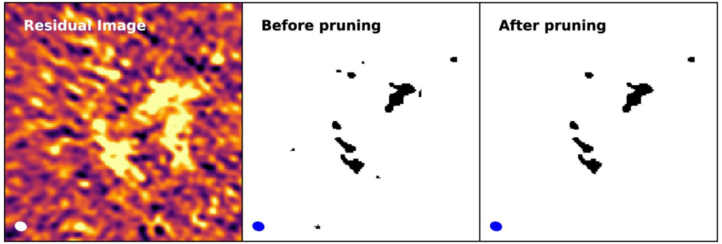


Fig. 3, Kepley+2020

- Regions smaller than the minimum beam fraction are pruned

3. Expand initial,pruned mask

Input parameters:

- `lownoisethreshold`

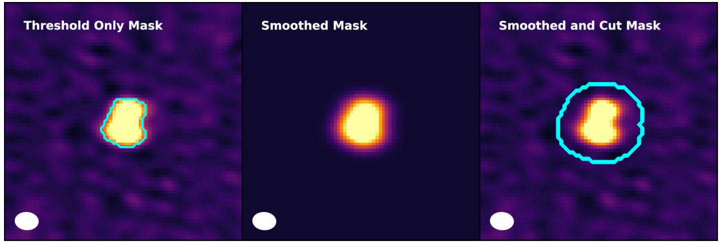


Fig. 4, Kepley+2020

- Mask is extended to include low signal-to-noise emission

usemask = 'auto-multithresh'

- Behavior of automasking depends on *uv* coverage
- Table of parameter values available on Automasking Guide: casaguides.nrao.edu/index.php/Automasking_Guide
- Can check 75th percentile baselines, b75, using AnalysisUtils

```
CASA <x>: au.getBaselineStats(vis)
```

- `b75 = 197.4m`, corresponding to 12m (short)
- Meant to be used as a guide, adjust as necessary

Array	*sidelobethreshold*	*noisethreshold*	*minbeamfrac*	*lownoisethreshold*	*negativethreshold*
12m (short) b75<300m	2.0	4.25	0.3	1.5	0.0 (continuum)/15.0 (line)
12m (long) b75>300m	3.0	5.0	0.3	1.5	0.0 (continuum)/7.0 (line)
7m (continnum/line)	1.25	5.0	0.1	2.0	0.0
12m + 7m combined **TENTATIVE**	2.0	4.25	0.3	1.5	0.0

usemask = 'auto-multithresh'

```
CASA <x>: inp tclean
    ...
usemask                = 'auto-multithresh'       # Type of mask(s)
   pbmask              = 0.2                      # primary beam mask
   sidelobethreshold   = 2.0                      # sidelobethreshold *...
   noisethreshold      = 4.25                     # noisethreshold * ...
   lownoisethreshold   = 1.5                      # lownoisethreshold * ...
   negativethreshold   = 0.0                      # negativethreshold * ...
   smoothfactor        = 1.0
   minbeamfrac         = 0.3                      # minimum beam fraction ...
   cutthreshold        = 0.01
   growiterations      = 75
   dogrowprune         = True
   minpercentchange    = -1.0
   verbose             = False
    ...
```

Setting clean stopping thresholds

```
CASA <x>: inp tclean
     ...
niter                = 100000                        # Maximum number of iterations
   gain              = 0.1                           # Loop gain
   threshold         = ''                            # Stopping threshold
   nsigma            = 2.0                           # rms-based threshold stopping
   cycleniter        = -1                            # Max minor-cycle iterations
   cyclefactor       = 1.0                           # Scaling on PSF sidelobe...
   minpsffraction    = 0.05                          # PSF fraction max depth...
   maxpsffraction    = 0.8                           # PSF fraction min depth...
   interactive       = True                          # Modify masks and parameters...
     ...
```

Summary of tclean inputs

```
CASA <x>: inp tclean
vis = 'sis14_twhya_calibrated_flagged.ms.contsub'
selectdata              = True
   field                = 'TW Hya'
   spw                  = '0'
      ...
datacolumn              = 'data'
imagename               = 'twhya_n2hp43'
imsize                  = [240,240]
cell                    = '0.1arcsec'
phasecenter             = 0
      ...
specmode                = 'cube'
   nchan                = 30
   start                = 230
   width                = 1
   outframe             = 'lsrk'
   restfreq             = '372.67250900GHz'
      ...
```

```
gridder                 = 'standard'
deconvolver             = 'multiscale'
   scales               = [0, 5, 10]
      ...
weighting               = 'briggs'
   robust                   = 0.5
      ...
usemask                 = 'auto-multithresh'
   sidelobethreshold    = 2.0
   noisethreshold       = 4.25
      ...
niter                   = 100000
   nsigma               = 2.0
   interactive          = True
      ...
```

Summary of tclean inputs (scripted ver.)

(Script file Imaging_2_basic_scripted.py)

```
tclean(vis='sis14_twhya_calibrated_flagged.ms.contsub',
       field = 'TW Hya',
       spw = '0',
       datacolumn = 'data',
       imagename = 'twhya_n2hp43_interactive',
       cell = '0.1arcsec',
       imsize = [240,240],
       phasecenter = 0,
       specmode = 'cube',
       start = 230,
       nchan = 30,
       width = 1,
       restfreq = '372.67250900GHz',
       outframe = 'lsrk',
       veltype = 'radio',
       gridder = 'standard',
           ...
```

```
...
deconvolver = 'multiscale',
scales = [0,5,10],
smallscalebias = 0.0,
weighting = 'briggs',
robust = 0.5,
usemask = 'auto-multithresh',
sidelobethreshold = 2.0,
noisethreshold = 4.25,
minbeamfrac = 0.3,
lownoisethreshold = 1.5,
negativethreshold = 0.,
niter=100000,
threshold = '',
nsigma = 2.,
pblimit = 0.2,
pbmask = 0.2,
interactive = True)
```

Running tclean

Run tclean:

`CASA <x>:  go`

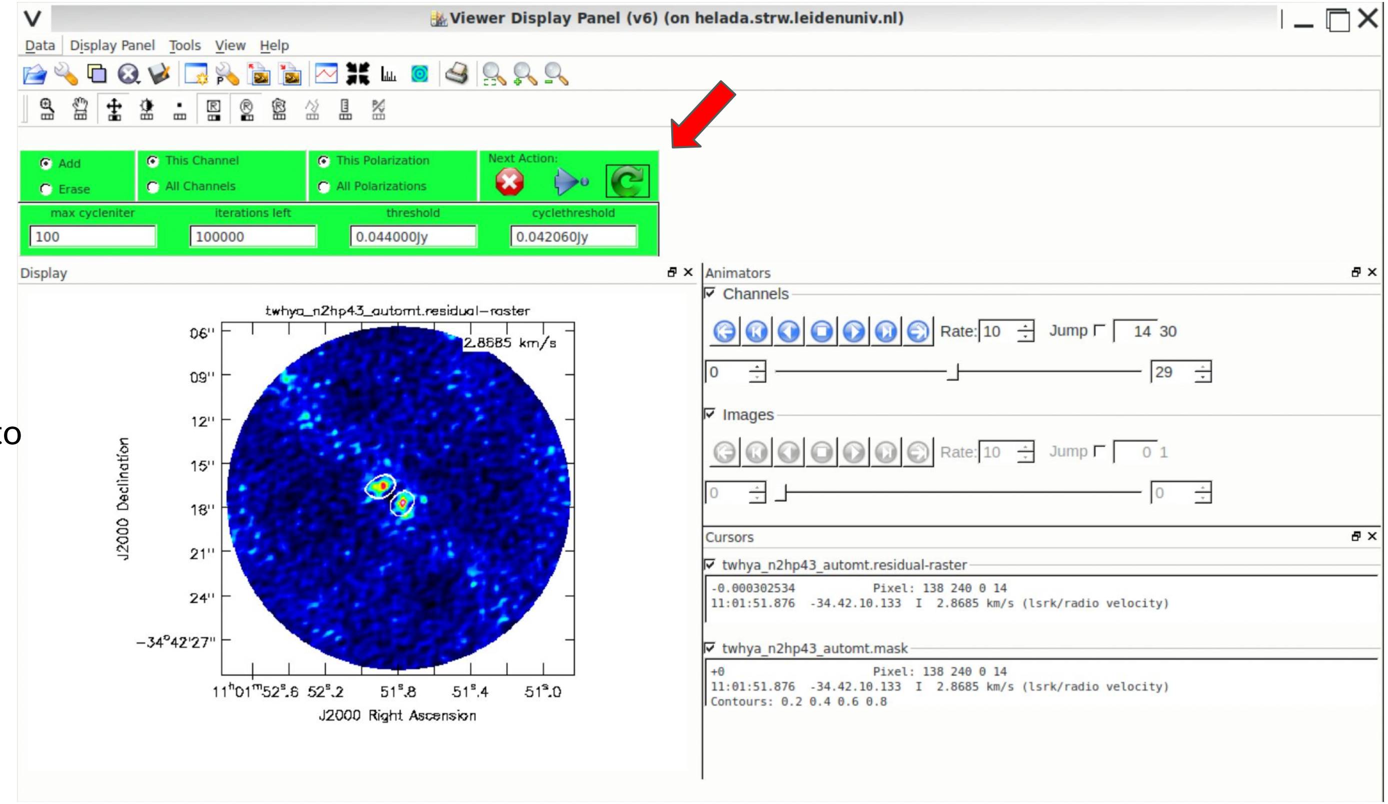
Inspect each channel:

Create / adjust masks using:

Use green clockwise arrow to continue cleaning & return interactive GUI

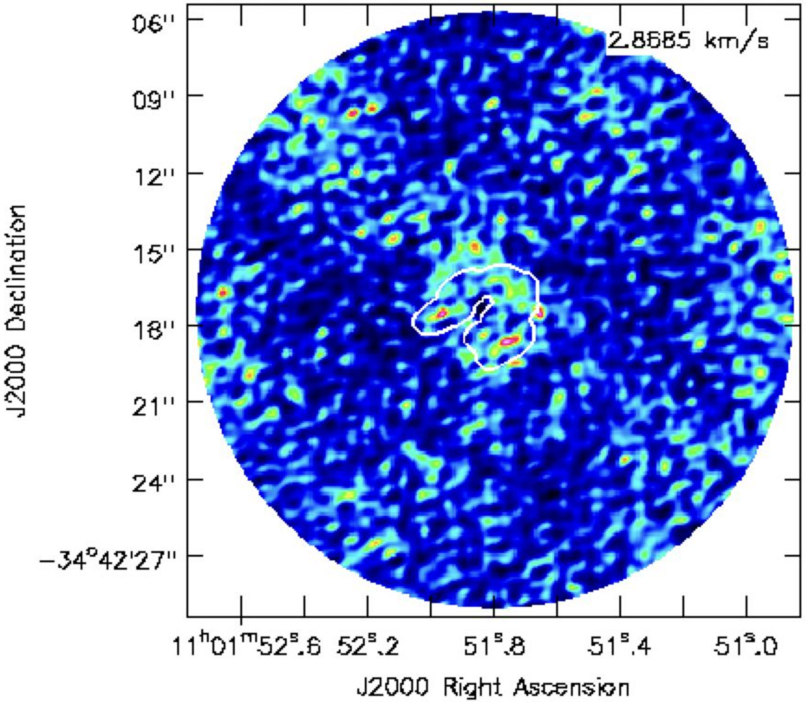
Use blue arrow to finish cleaning non-interactively

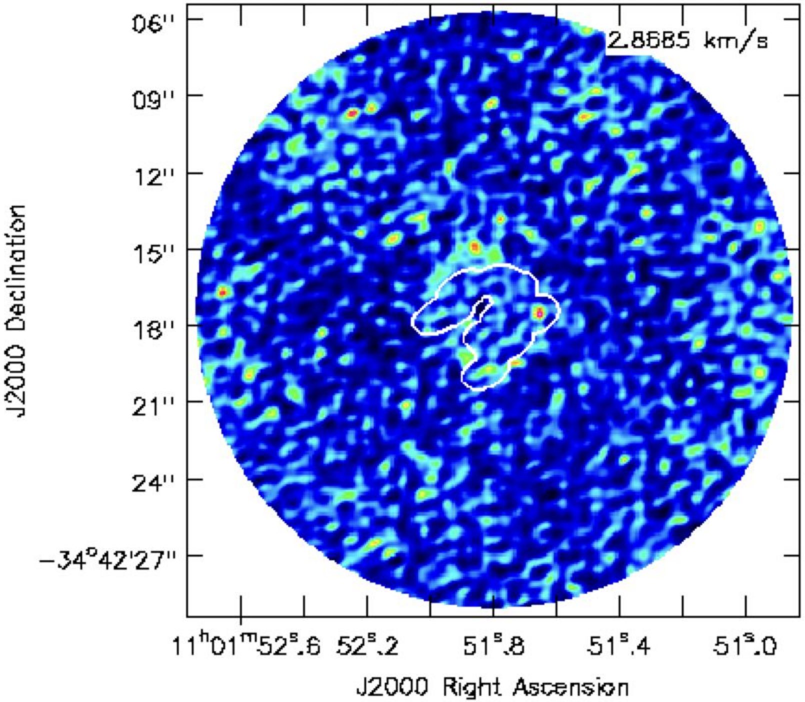
Use red button to terminate tclean



Running tclean

As cleaning progresses, source emission → residuals





Inspect output files

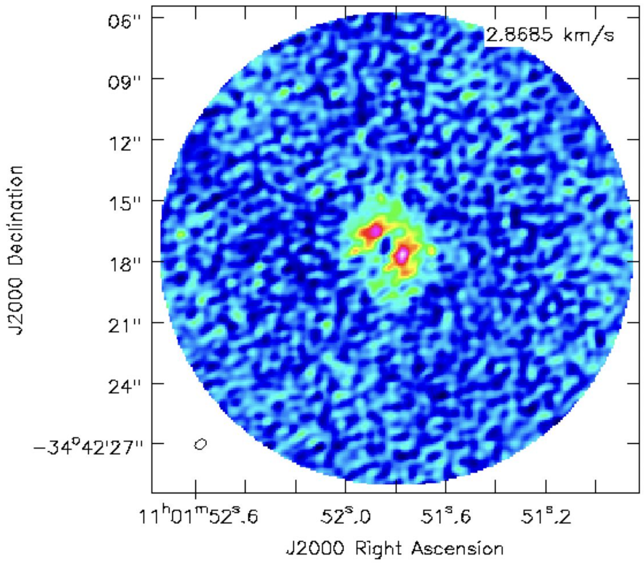
Output image extensions:

`.image`, `.mask`, `.model`,
`.pb`, `.psf`, `.residual`,
`.sumwt`

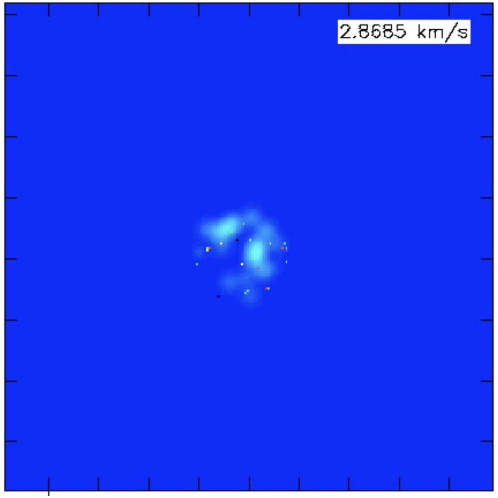
+others for different imaging setups

- Inspect all output files to make sure things look correct:
  - Inspect the clean components of the model
  - Check residuals to see if there is any "uncleaned" emission

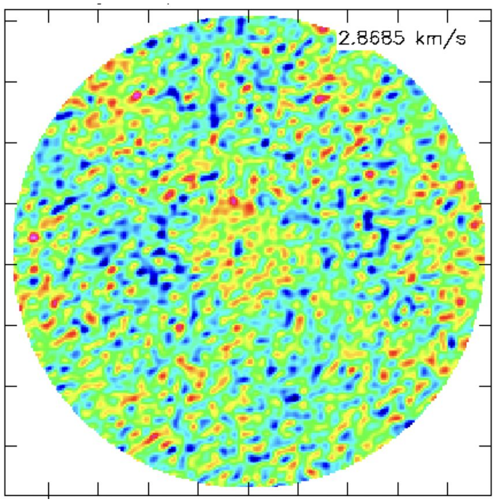
Image, channel 14



Model



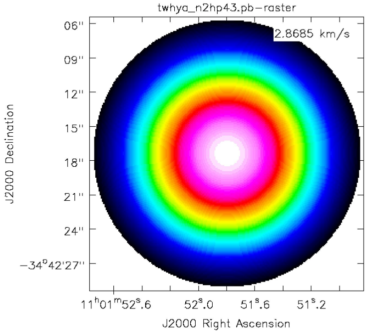
Residual

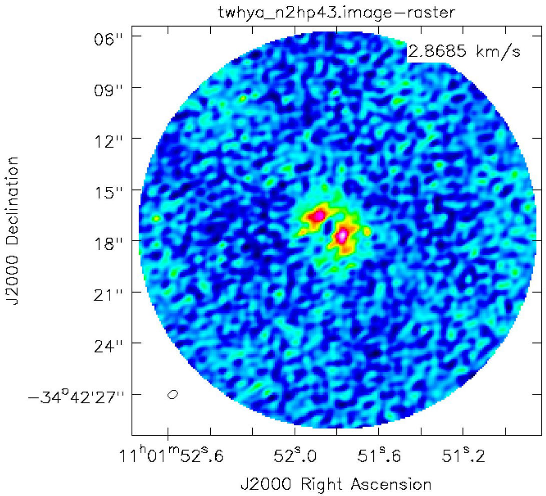


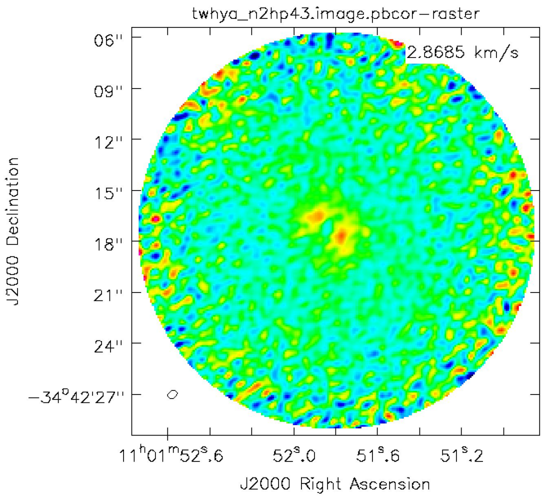
Additional bits: Primary beam correction (pbcor=True)

Output image extensions:

.image, .mask, .model,
.pb, .psf, .residual,
.sumwt
.pbcor







Additional bits: Continuum subtraction

(Script file Imaging_3_uvcontsub.py)

```
listobs(vis='sis14_twhya_calibrated_flagged.ms')
plotms(vis='sis14_twhya_calibrated_flagged.ms',
       xaxis='channel',
       yaxis='amp',
       field='5',
       avgspw=False,
       avgtime='1e9',
       avgscan=True,
       avgbaseline=True,
       showgui = True)

uvcontsub(vis = 'sis14_twhya_calibrated_flagged.ms',
          outputvis = 'sis14_twhya_calibrated_flagged.ms.contsub',
          field = '5',
          fitspec = '0:0~230;281~383',
          fitorder = 0)
```

Additional example: tapering

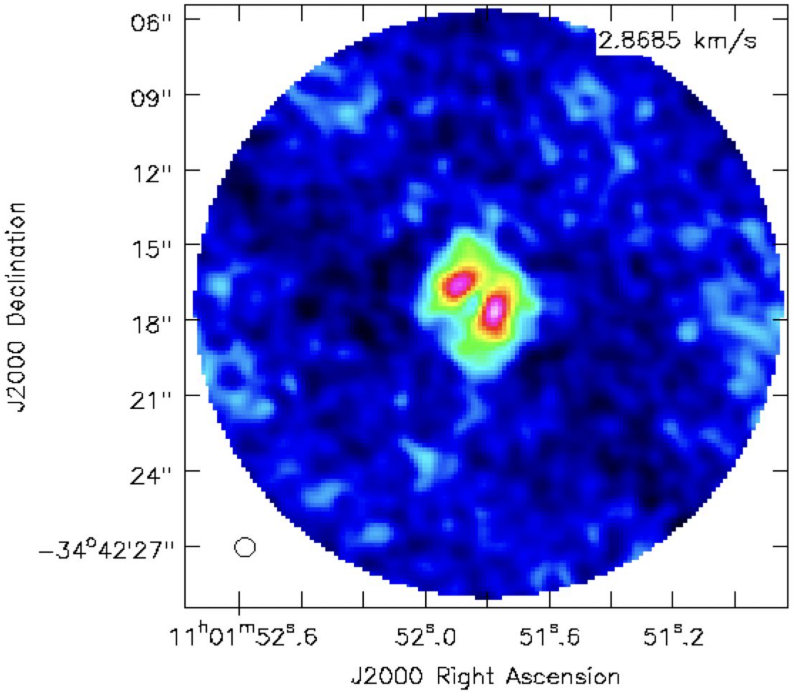
(Script file `Imaging_4_uvtaper_scripted.py`)

- Create new image with larger synthesized beam:
  `~0.4x0.6'' to ~0.8''`

- Estimate uvtaper with CASA tool:
  `ia.beamforconvolvedsize`

```
CASA <x>: inp tclean
   ...
   imagename              = 'twhya_n2hp43_taper'
   cell                   = '0.16arcsec'
   imsize                 = [150,150]
   ...
weighting                 = 'briggs'
   robust                 = 2
   uvtaper                = ['0.67arcsec',
                             '0.54arcsec',
                             '32.16deg']

   ...
```



Additional bits: parameter customization

`weighting = 'natural'`

lower rms noise, lower resolution

`= 'uniform'`

reduces sidelobes, higher rms noise

`= 'briggs'`

`robust` parameter can be adjusted from -2 (uniform-like) to 2 (natural-like): 0.5 in our original setup

`usemask = 'auto-multithresh'`

`= 'user'`

Happy cleaning!