ALMA imaging basics - 2nd Day

Archive School - Bologna 5-7 October

Luke Maud
MCarmen Toribio
Daniel Tafoya
Marcella Massardi

EUROPEAN ARC
ALMA Regional Centre

OPTICON RadioNet Pilot

Overview

- Basic interferometric concepts
- Basics imaging parameter selection
- Manual Imaging
- Pipeline Imaging
- Brief mention on image analysis
- Brief mention of Self-calibration

What we hope to achieve here

- You have some grasp on concepts of interferometer synthesis imaging
- You will understand how to image data you got from the ASA
  - Manually
  - Using quick Pipeline commands
- You can begin some basic analysis
  - You can *also* use the ALMA products directly
- You have an understanding of the caveats and some important points when considering to combine non-homogeneous data

What this is not:

A detailed synthesis imaging guide/overview

A detailed inspection of all CASA `tclean` parameters

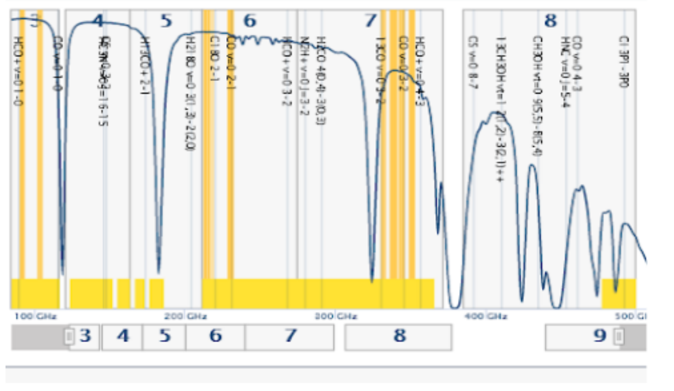
Basic interferometric concepts *(for ALMA archive use)*

- Band
- Spectral Resolution
- Field of View
- Primary Beam Correction
- Angular Resolution
- Maximum recoverable scale
- Sensitivity to spatial scales - U,V coverage
- Array type
- Image units
- Merging data

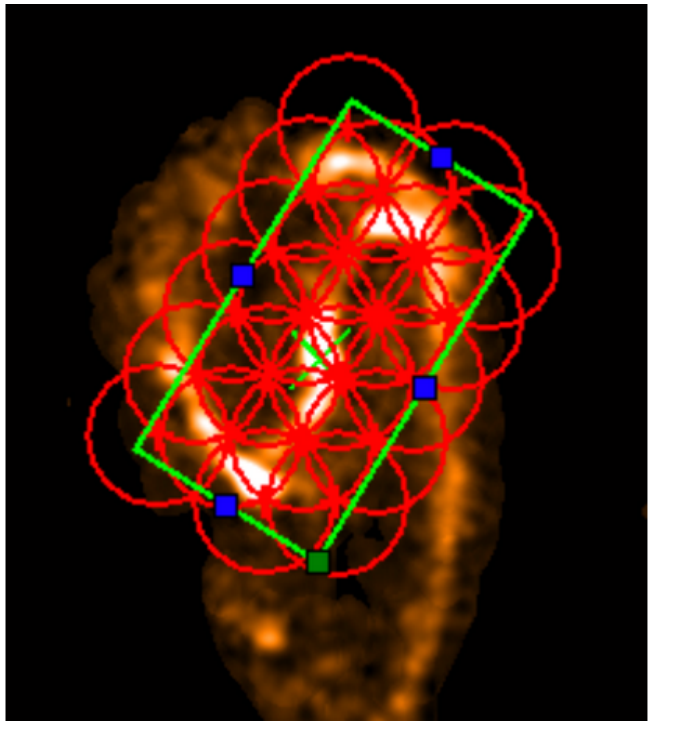
Also see session tomorrow - “Tips and Tricks on the ASA”

Basic interferometric concepts *(for ALMA archive use)*

- Band & Spectral Resolution
  - Frequency of interest for continuum or lines
  - Science you want might need multiple images from various datasets, spectral resolution is better (smaller) for same channel width at a higher frequency
  - Non-homogenous projects need to consider other parameters, e.g. re-binning of channels (lose information)



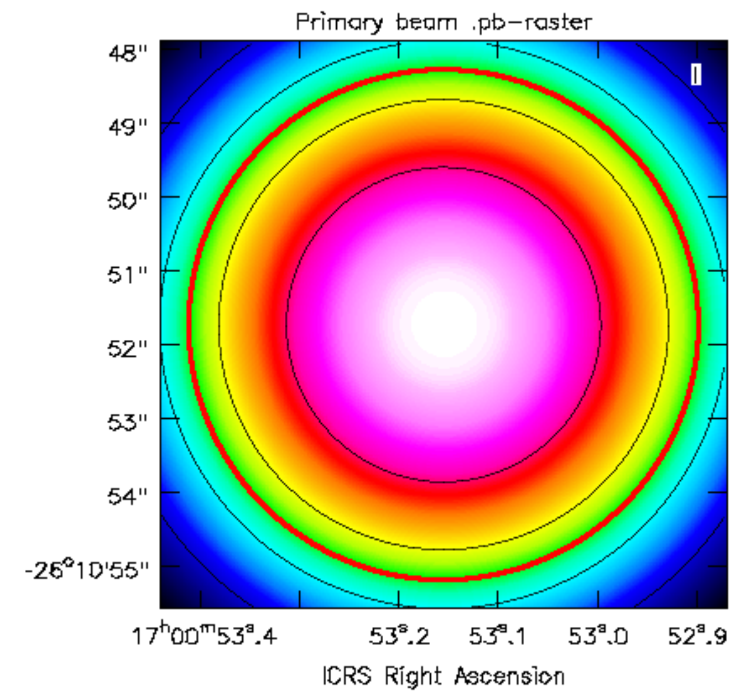
- Field of View (FOV)
  - Coverage of the observation
  - A single pointing has a primary beam (HPBW) of ~1.13 λ/d, where λ is wavelength and d is the antenna diameter, such that sensitivity to a signal is at the 50% response.
    - Angular size between nulls in the single dish response is ~2.44 λ/d
  - Mosaics built up of multiple pointings and cover a large area

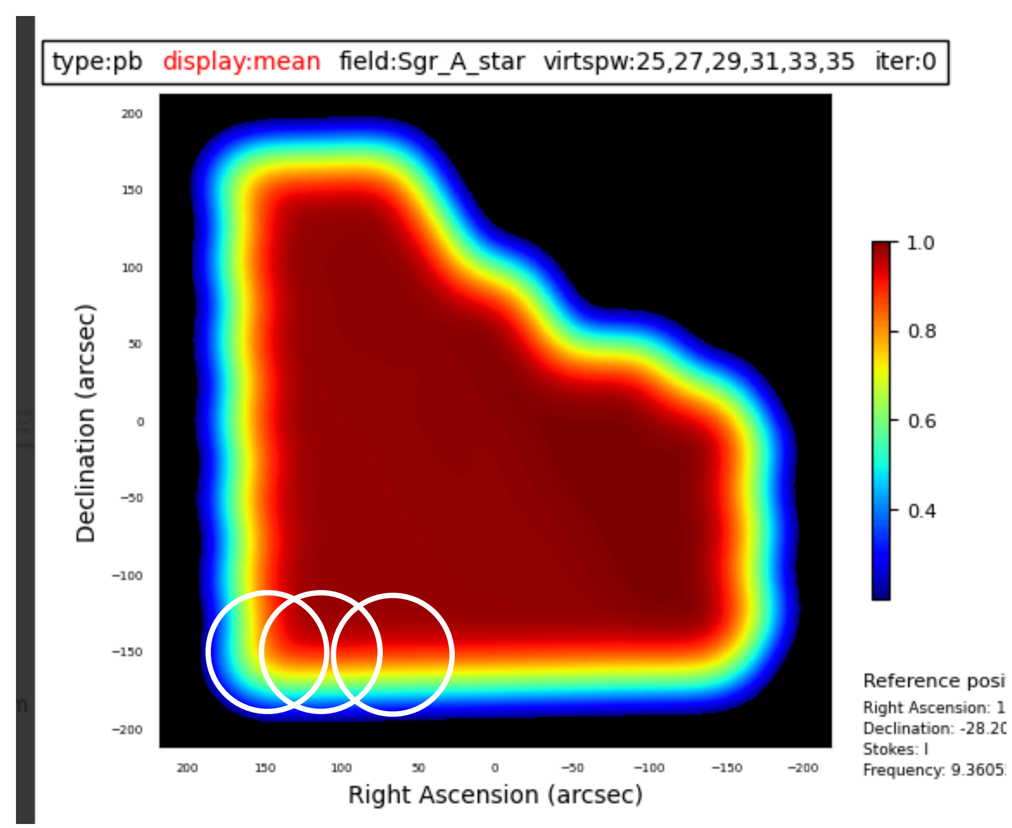


Basic interferometric concepts *(for ALMA archive use)*

- Primary Beam Correction
  - ALMA images are primary beam corrected - intrinsically nosier at the edges of the field of view
  - All archive products are PB corrected
  - Mosaics are Nyquist sampled - “uniform” noise over the well sampled area and noisier at the edges of the map
  - If measuring map noise use uncorrected PB image, or stay within the HPBW

Contours - 20,40,50,60,80%





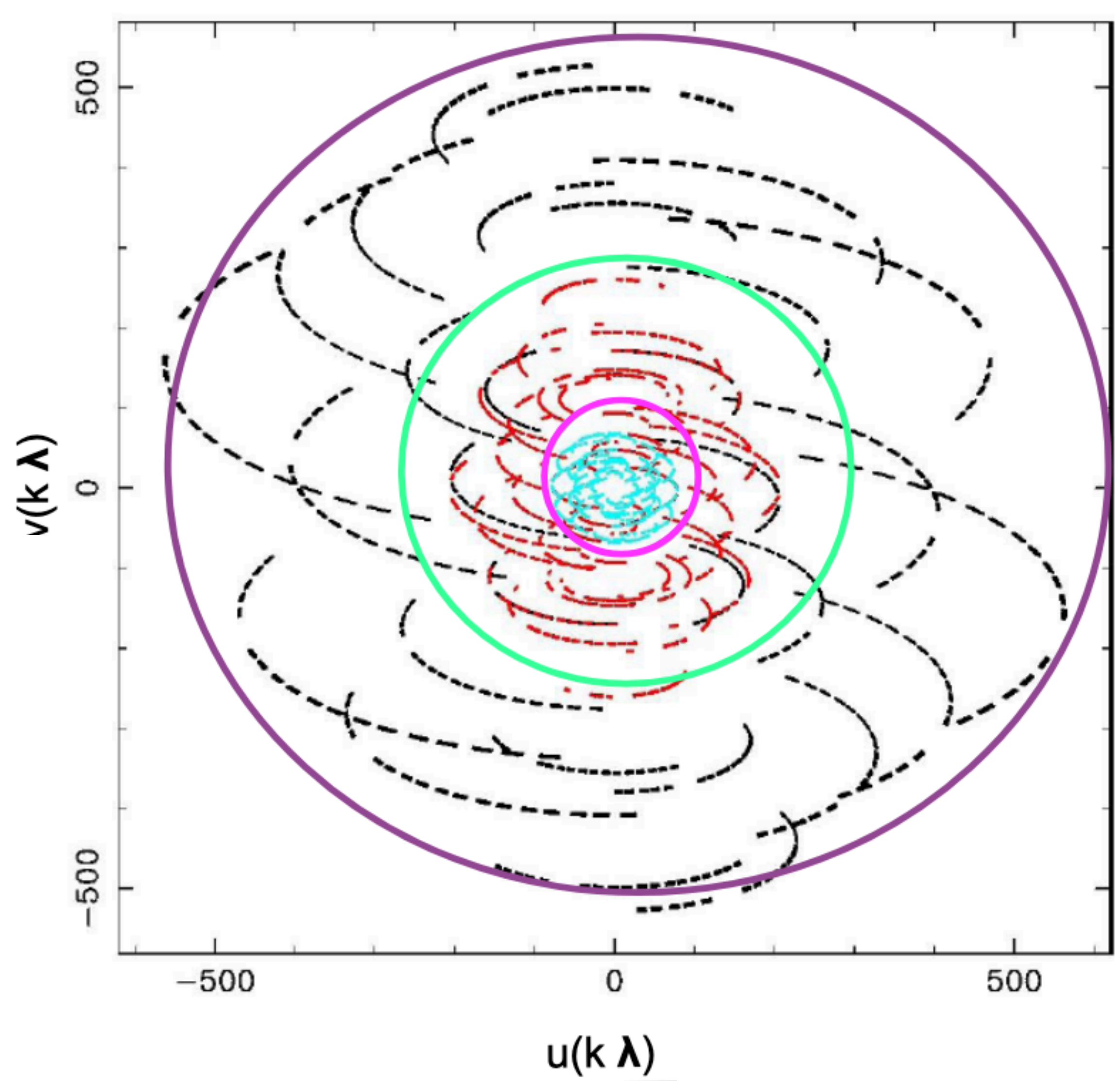
Basic interferometric concepts *(for ALMA archive use)*

- Angular Resolution (AR)
  - Will usually be ~λ/D, where λ is wavelength and D is the **maximal baseline length**
  - Calibration must be good otherwise depending on conditions, longer baselines (of any array) could suffer from decoherence (worse phase noise) and have a reduced signal
    - the signal from the smallest scales is not actually reliable
- Maximum recoverable scale (MRS)
  - Sensitivity to large scale structure, interferometers are otherwise insensitive (do not “see”) larger scales as they do not sample them (based on configuration)
  - MRS ~ 0.6 λ/D_{min}, where D_{min} is the shortest baseline in the array
    - Caution using the ‘single’ shortest baseline - *take a representative value*
  - Flux will be missing from large scale extended sources that are poorly sampled even if you have some short baselines and can cause imaging problems **- *ripples and striping***

Basic interferometric concepts *(for ALMA archive use)*

- Sensitivity to spatial scales - U,V coverage
  - What do you want to image or need for the science
  - Ties directly with AR and MRS - caution for using different data and about merging
  - Length of time - snapshot or longer - better U,V coverage, better representation of the true sky
  - Imaging parameters ideal for one array/ observation might not be the same as for a different one

Elisabetta's introduction talk
From D. Wilner - SMA imaging school



Basic interferometric concepts *(for ALMA archive use)*

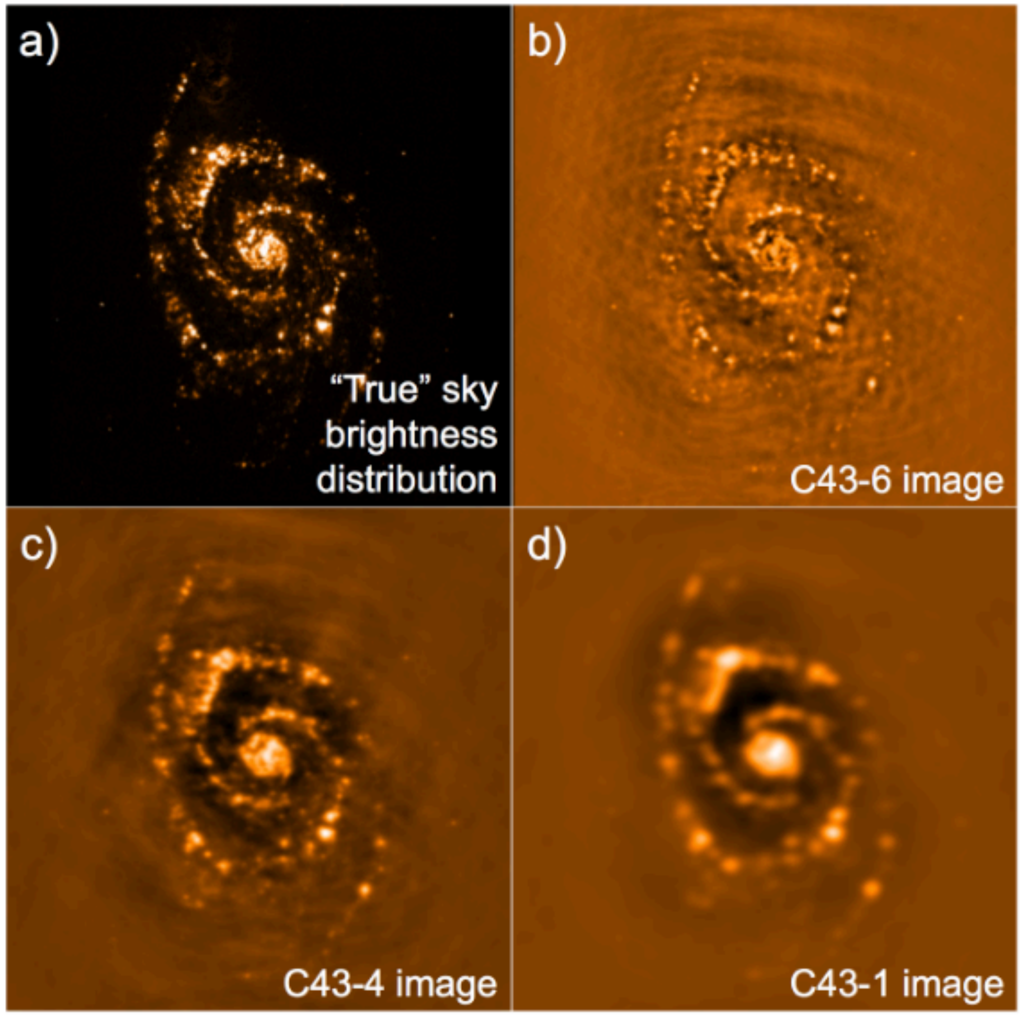
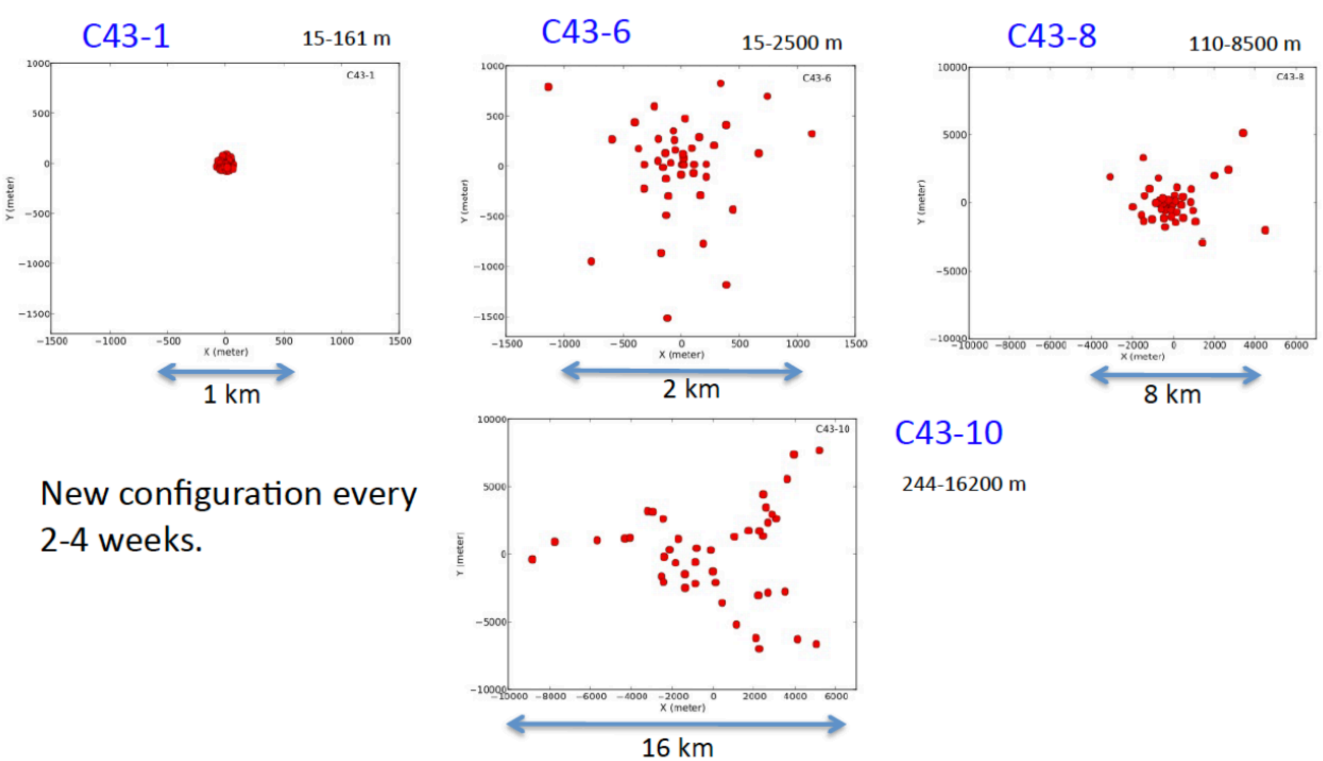


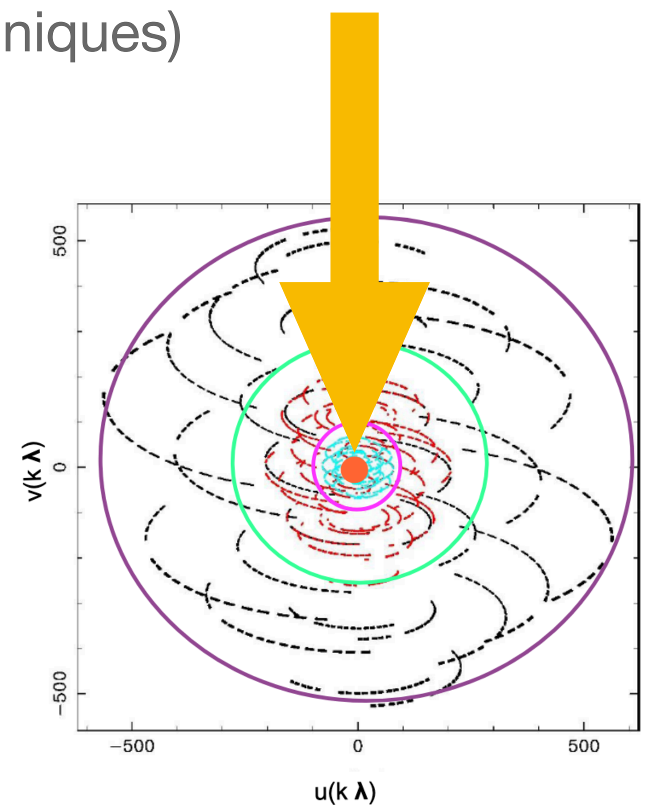
Figure 3.6: Examples of spatial filtering using the CASA task *simobserve* on *notional* ALMA configurations for Cycle 9. *Panel a (upper left)*: An optical image of the galaxy M51 used as a template for a true sky brightness distribution for the simulations. The frequency of the emission has been changed to 100 GHz, the image size has been scaled to $\sim 3' \times 3'$, and its declination has been changed to -40° to allow ALMA observations to be simulated. For the simulations, the galaxy was "observed" over a mosaic of 33 pointings, for ∼10 hours in total. The resulting dirty images were CLEANed. *Panel b (upper right)*: The high-resolution image of the galaxy obtained when observed in the ALMA C43-6 notional configuration with maximum baseline of 2516.9 m, respectively. The resulting synthesized beam is $\sim 0.47''$ and the maximum recoverable scale is $\sim 4''$. *Panel c (lower left)*: Medium-resolution image of the galaxy when observed in the ALMA C43-4 notional configuration with maximum baseline of 783.5 m, respectively. The resulting synthesized beam is $\sim 1.3''$ and the maximum recoverable scale is $\sim 11''$. *Panel d (lower right)*: Low-resolution image of the galaxy when observed in ALMA notional configuration C43-1 with maximum baselines of 160.7 m, respectively. The resulting synthesized beam is $\sim 4.3''$ and the maximum recoverable scale is $\sim 28''$.

Basic interferometric concepts *(for ALMA archive use)*

- Array type
  - Is the archive data selected suitable, i.e. intended for merging or not - i.e. C43-1 will not match with C43-10 - missing mid-length-baselines
    - Arrays should overlap in baseline lengths - continuous in U,V
  - ACA and Total-Power for short spacing -> large spatial scales
    - Caution with weights, and blending of TP (SDint / Feather techniques)



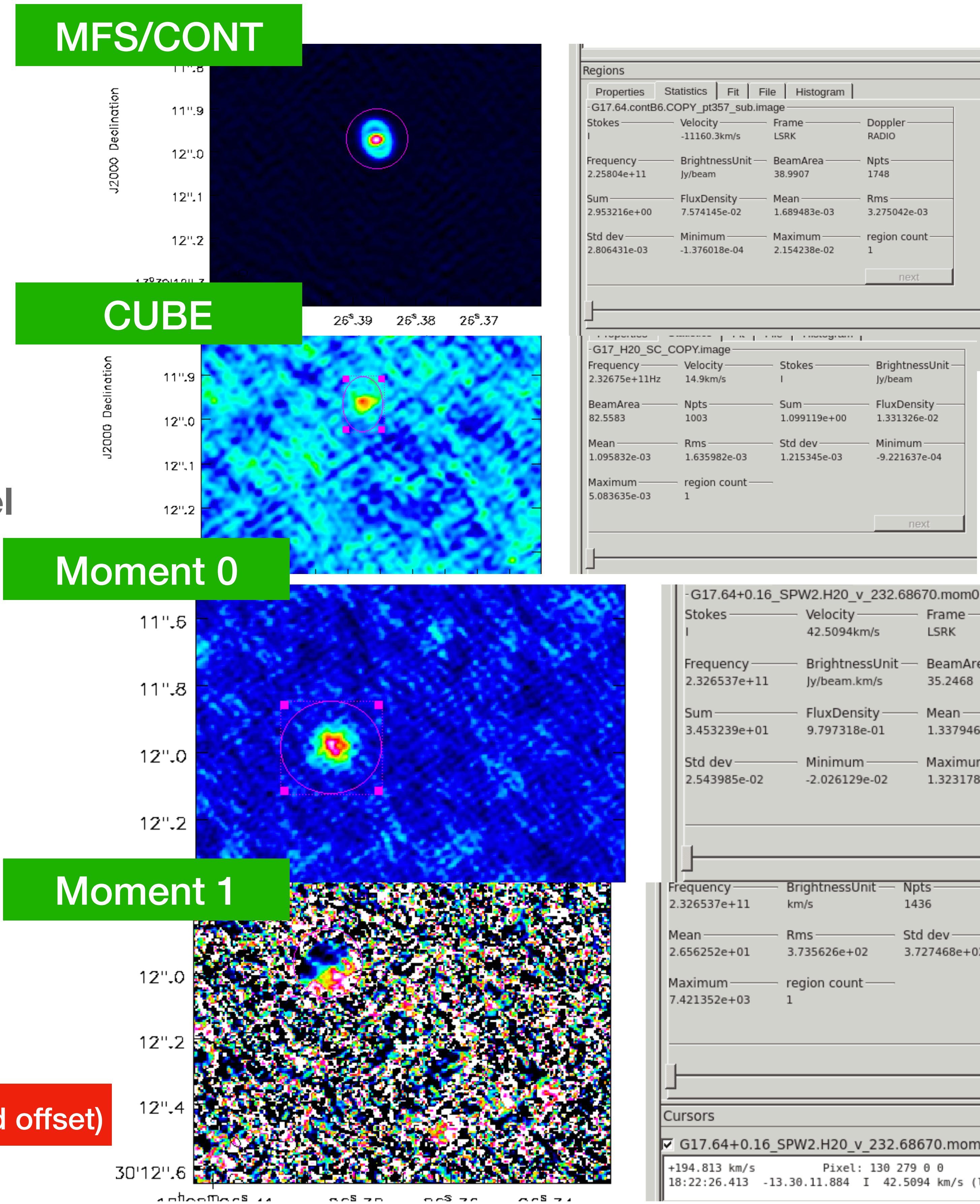
New configuration every 2-4 weeks.



Recap Introductory Talk

Basic interferometric concepts *(for ALMA archive use)*

- Imaging units (recap)
  - Continuum images:
    - ‘mfs’ or ‘cont’ are a single plane
    - Units are **Jy/beam**
    - Flux Density is **Jy**
  - Caution with models - **Jy / pixel**
    - If you input images into the CASA task `simobserve` make sure they are in **Jy/pixel**
  - Cube images:
    - Units are **Jy/Beam** per **CHANNEL**
      - Channel unit **km/s** or **mHz (kHz)**
  - Moment Zero:
    - Units are **Jy/Beam.Km/s**
    - Integrated over velocity
  - Moment One:
    - Units are **km/s**



Axes - J2000(FK5) != ICRS (10s milli-arcsecond offset)

Basic interferometric concepts *(for ALMA archive use)*

- Merging data
  - Required for the science?
  - Aforementioned array combinations - are they suitable ?
  - Same **Band**, same **spectral coverage** and **resolution**?
  - Same **field of view**?
  - Similar **sensitivities**?
    - Usually limited by longer baselines - brightness sensitivity in terms of temperature (K) is worse for high angular resolutions (beam dependent) vs. point source sensitivity (Jy/beam)
    - Older data *(Cycle 0,1,2)* has data weighting issues between 12m and ACA - needs manual adjustments - https://casaguides.nrao.edu/index.php/DataWeightsAndCombination
  - Coordinate units - **J2000 (FK5) to ICRS**

Specific data merging tutorials: https://casaguides.nrao.edu/index.php?title=M100_Band3
Also data combination group work: https://ui.adsabs.harvard.edu/abs/2021AAS...23735307K/abstract

Questions so far?

Basics of imaging synthesis

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

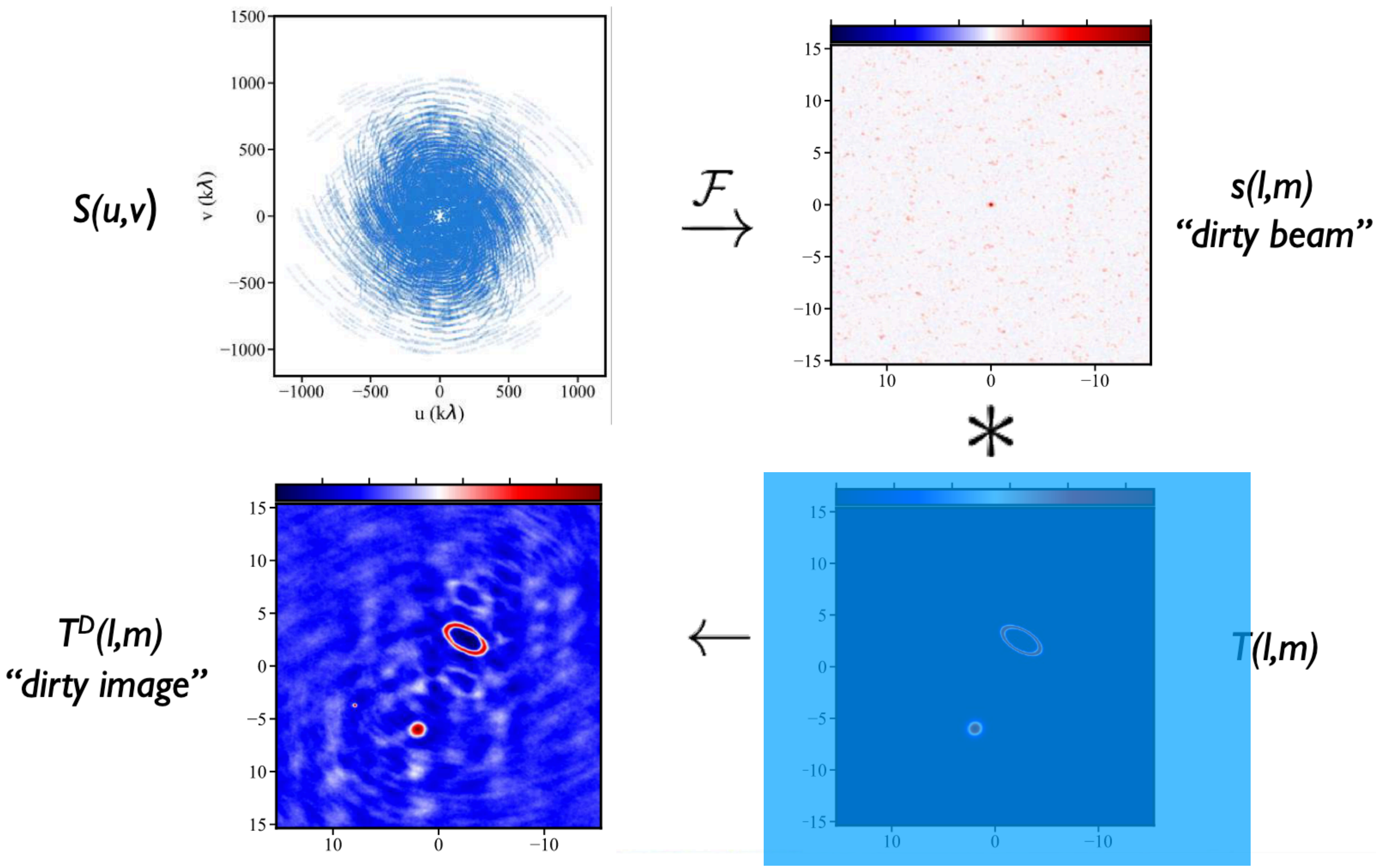
- Interferometers sample the sky in the Fourier Domain (the 'Visibilities') which are complex quantities (amplitude and phase) -> *think "flux and position"*
- Imaging is an inverse Fourier transform
  - We have sampled particular U, V coordinates with given baselines at given times. These must be 'converted' into physical parameter space onto an ***l, m*** grid
  - Mathematical transforms change U,V into image plane
    - $\boldsymbol{V}(U,V) = 2D\ FT\ \{\boldsymbol{B}_{primary}.\boldsymbol{I}_{source}\}$ (Visibilities)
    - $\boldsymbol{S}(U,V) = 1$ where U,V are sampled, $= 0$ if not (Sampling function)
    - $\boldsymbol{B}_{dirty}(l,m) = 2D\ FT^{-1}\ \{S\}$ (Dirty Beam)
    - $\boldsymbol{I}_{meas}(l,m) = 2D\ FT^{-1}\ \{S.V\}$ (Measured image)
      - $\longrightarrow \boldsymbol{I}_{meas} = \boldsymbol{B}_{dirty} * \{\boldsymbol{B}_{primary}.\boldsymbol{I}_{source}\}$
  - So we doing a Fourier Transform, then must deconvolve to 'remove' the dirty beam

Credits- J.Pety IRAM

Recommended (i) : https://science.nrao.edu/science/meetings/2018/16th-synthesis-imaging-workshop/talks/Wilner_Imaging.pdf

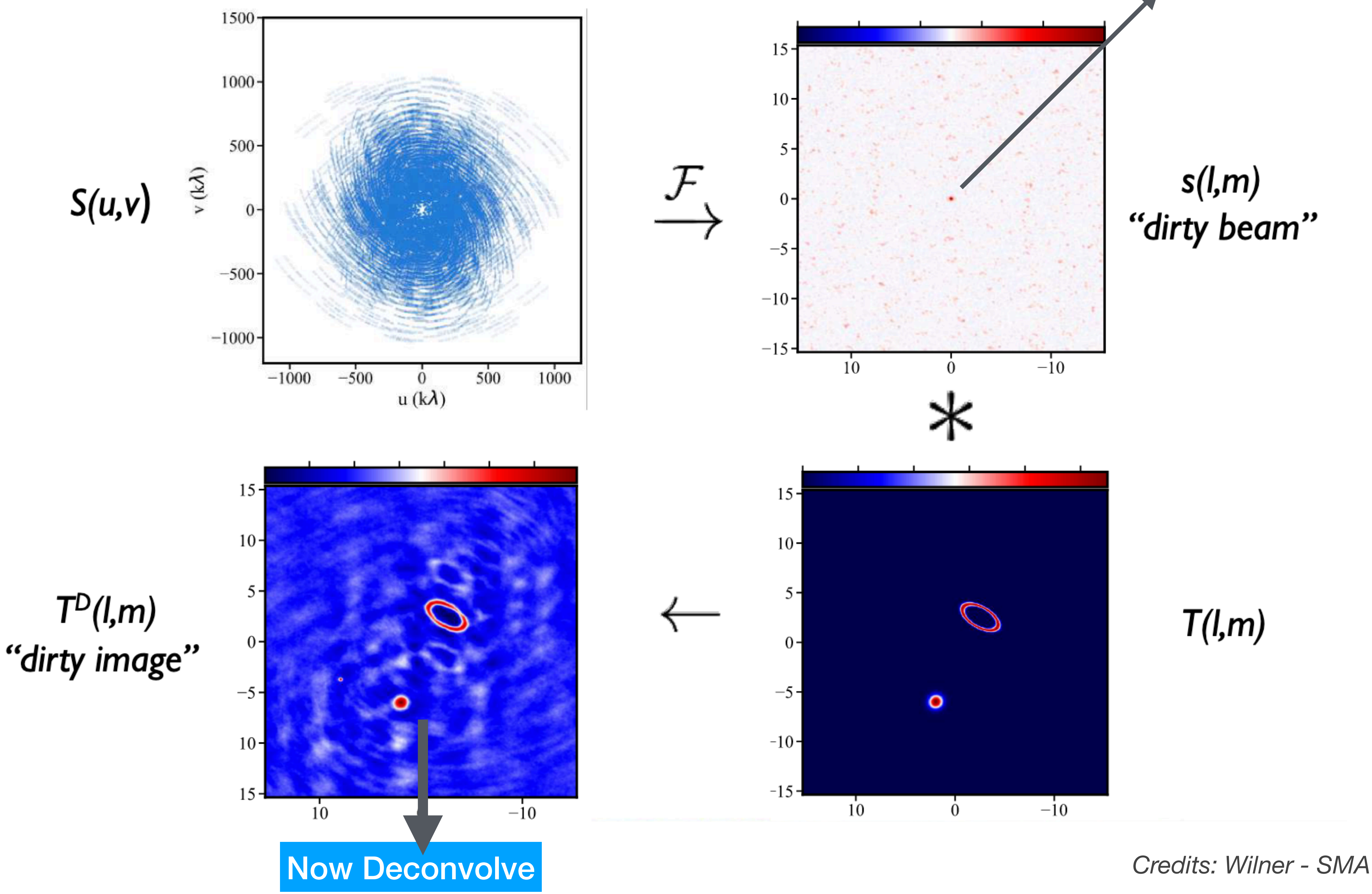
Recommended (ii) : https://www.iram-institute.org/medias/uploads/file/PDFs/IS-2018/pety-single-field.pdf

Basics of imaging synthesis



Credits: Wilner - SMA

Basics of imaging synthesis



Credits: Wilner - SMA

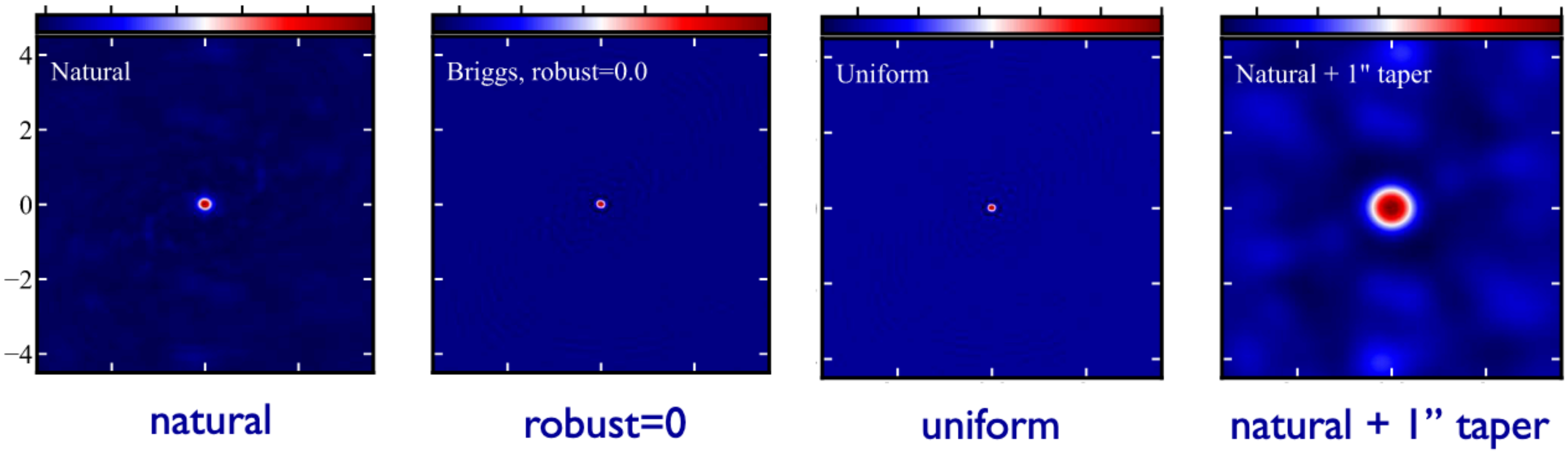
Imaging parameter selection

- When imaging a number of parameters need to be used:
  - **Cell (pixel) size**: clean beam / 5 - i.e. at least 5 cells (pixels) per clean beam *(you know AR)*
    - Required to grid correctly, ascribe flux to 'correct' locations within a 'clean beam'
    - Pixels too large - blocky image flux build up in 'wrong' places, poor clean beam 'fit'
    - Pixels too small - hard for the Fourier transform, cells 'empty', could affect weightings
  - **Image Size**: Cover the Primary Beam
    - If emission is not large scale, image to HPBW or smaller *(long baselines which can be huge images)*, for mosaics *always* extend past the edges
  - **Specmode**: 'mfs' (multi-frequency synthesis) or 'cube' spectral line cube
    - ALMA/CASA Pipeline specific 'cont' - merges all SPWs
  - **Cleaning type**: `CLEAN`ing - **Hogbom**, Clark, Multi-Scale; (but also Max Entropy MEM)
    - **niter** / **threshold** - how much to clean by before stopping

Imaging parameter selection

- When imaging a number of parameters need to be used:
  - **Robust**: Numerical Value from +2 *Natural* to -2 *Uniform (between is 'Briggs Robust')*
    - **+ve**, weights towards shorter baselines (each baselines is equally weighted and more shorter ones are always sampled - lower AR, but maximised sensitivity)
    - **-ve**, weights towards longer baselines (gives more power to least sampled visibilities - inversely proportioned, increasing noise but best AR)
    - Default = 0.5, 'middle-ground' between resolution and sensitivity
  - **Taper**: Make the beam larger (worse AR) by Apodizing the U,V by a Gaussian
    - Like smoothing the image with a Gaussian but not 'exactly' the same
  - **uvrange**: optional method to limit the range of visibilities in the image, e.g. if a few shorter baseline are causing a striping, you can exclude them from the time (uvrange = '800~16000' - default meters, or specify klambda - obeys list rule for multiple MS)

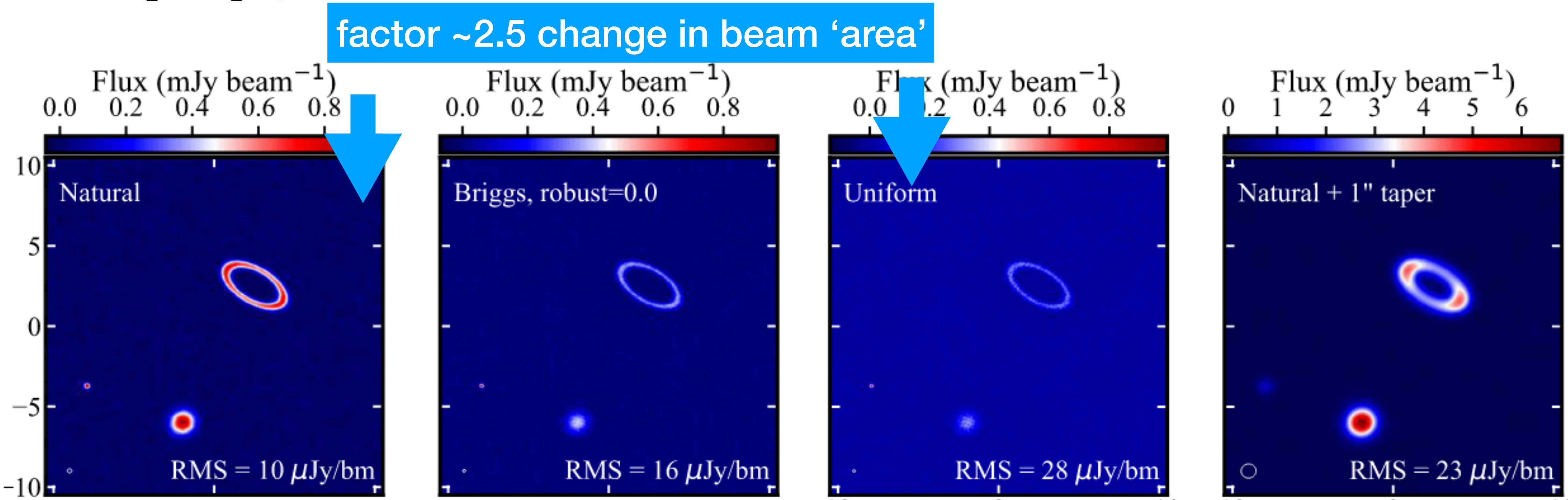
Imaging parameter selection



	Robust/Uniform	Natural	Taper
resolution	higher	medium	lower
sidelobes	lower	higher	depends
point source sensitivity	lower	maximum	lower
extended source sensitivity	lower	medium	higher

Credits: Wilner - SMA

Imaging parameter selection



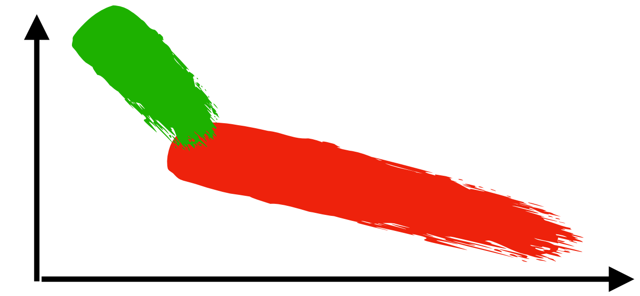
natural	robust=0	uniform	natural + 1" taper
0.29x0.25 p.a. -81	0.19x0.17 p.a. -78	0.17x0.15 p.a. -87	0.93x0.88 p.a. -86

	Robust/Uniform	Natural	Taper
resolution	higher	medium	lower
sidelobes	lower	higher	depends
point source sensitivity	lower	maximum	lower
extended source sensitivity	lower	medium	higher

EUROPEAN ARC
ALMA Regional Centre

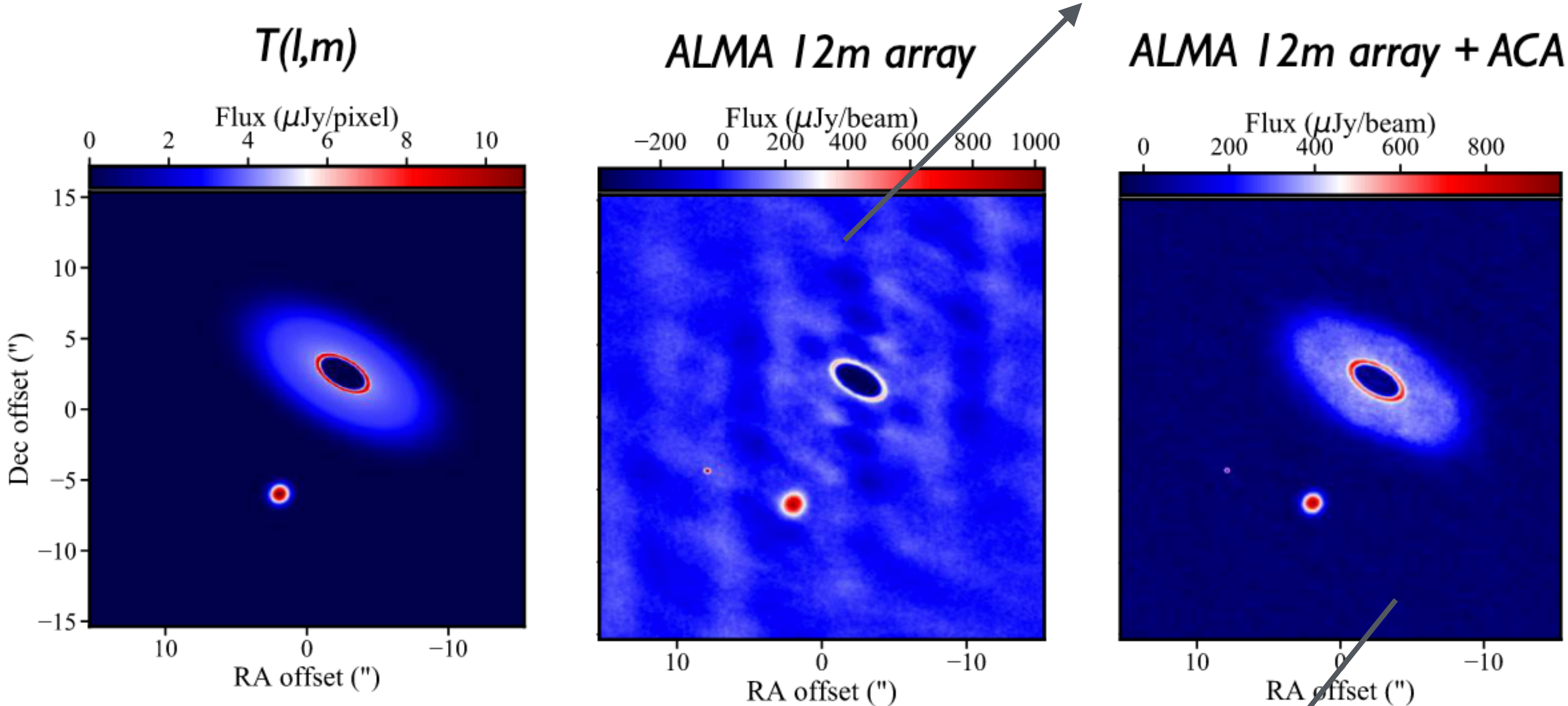
Credits: Wilner - SMA

Imaging parameter selection



Merging ACA and 12m to recover extended scales

Misses all extended structure, no U,V to sample this - so cannot image it



Recovers much of the extended structure

Credits: Wilner - SMA

EUROPEAN ARC
ALMA Regional Centre

Imaging parameter selection

- Brief aside: Image parameters are permitted to vary to reach a PI's main goal *(ACA 7m QA is not beam assessed - fixed array)*
  - Manual calibration values chosen by an analyst
  - Pipeline calculated by heuristics
    - Mainly robust is adjusted to meet the beam, within 0.0 to 2.0. Negative are not use as large scales are notably down-weighted
    - Pipeline mitigation for 'large' images
      - 3x cell per beam, smaller map (0.5 PB)

Goals From OT:

Representative Target: LkCa15

Representative Frequency: 663.9863 GHz (SPW 37)

Bandwidth for Sensitivity: 1.5e+04 MHz

Min / Max Acceptable Resolution: 0.146 arcsec / 0.179 arcsec

Maximum expected beam axial ratio (from OT): Not available

Goal PI sensitivity: Not available

Single Continuum: False

Estimated Synthesized Beam and Sensitivities for the Representative Target/Frequency

Estimates are given for four possible values of the tclean robust weighting parameter: robust = 0.0, +0.5 (default), +1.0, and +2.0. **If the "Min / Max Acceptable Resolution" is available (>=Cycle 5 12-m Array data)**, the robust value closest to the default (+0.5) that predicts a beam area (defined as simply major x minor) that is in the range of the PI requested beam areas according to the table row
Sensitivity) is chosen. If none of these robust values predict a beam area that is in range, robust=+2.0 is chosen if the predicted beam area is too small, and robust=0.0 is chosen if the predicted beam area is too large. The chosen robust value is highlighted in green and used for all science target imaging. In addition to an estimate for the repBW, an estimate for the aggregate continuum bandwidth (
NO line contamination but accounting for spw frequency overlap. If the Bandwidth for Sensitivity (repBW) is > the bandwidth of the spw containing the representative frequency (repSPW), then the beam is predicted using all spws, otherwise the beam is predicted for the repSPW alone. A message appears on the "By Task" view if a non-default value of robust (i.e., not +0.5) is chosen. Additionally, if
within the PI requested range using one of the four robust values, Warning messages appear on this page.

These estimates should always be considered as the BEST CASE SCENARIO. These estimates account for Tsys, the observed uv-coverage, and prior flagging. The estimates DO NOT account for (1) subsequent science target flagging; (2) loss of continuum bandwidth due to the hif_findcont process (i.e. removal of lines and other spectral features from the data used to image the continuum); (3
quality like (a) poor match of uv-coverage to image complexity; (b) dynamic range effects; (c) calibration deficiencies (poor phase transfer, residual baseline based effects, residual antenna position errors, etc.). *It is also important to note that both the repBW and aggBW beam calculations are intrinsically multi-frequency synthesis continuum calculations, using the relevant spws as described above. T*
single channel in a cube will typically be larger and can be significantly larger depending on the details of uv-coverage and channel width.

robust	uvtaper	Synthesized Beam	Cell	Beam Ratio	Bandwidth	BW Mode	Effective Sensitivity
0.0	[]	0.176 x 0.154 arcsec @ -15.0 deg	0.031 x 0.031 arcsec	1.14	8169 MHz	repBW	0.000373 Jy/beam
0.0	[]	0.176 x 0.154 arcsec @ -15.0 deg	0.031 x 0.031 arcsec	1.14	8169 MHz	aggBW	0.000373 Jy/beam
0.5	[]	0.204 x 0.175 arcsec @ -20.2 deg	0.035 x 0.035 arcsec	1.17	8169 MHz	repBW	0.000298 Jy/beam
0.5	[]	0.204 x 0.175 arcsec @ -20.2 deg	0.035 x 0.035 arcsec	1.17	8169 MHz	aggBW	0.000298 Jy/beam
1.0	[]	0.245 x 0.206 arcsec @ -22.6 deg	0.041 x 0.041 arcsec	1.19	8169 MHz	repBW	0.00027 Jy/beam
1.0	[]	0.245 x 0.206 arcsec @ -22.6 deg	0.041 x 0.041 arcsec	1.19	8169 MHz	aggBW	0.00027 Jy/beam
2.0	[]	0.266 x 0.222 arcsec @ -23.5 deg	0.044 x 0.044 arcsec	1.20	8169 MHz	repBW	0.000268 Jy/beam
2.0	[]	0.266 x 0.222 arcsec @ -23.5 deg	0.044 x 0.044 arcsec	1.20	8169 MHz	aggBW	0.000268 Jy/beam

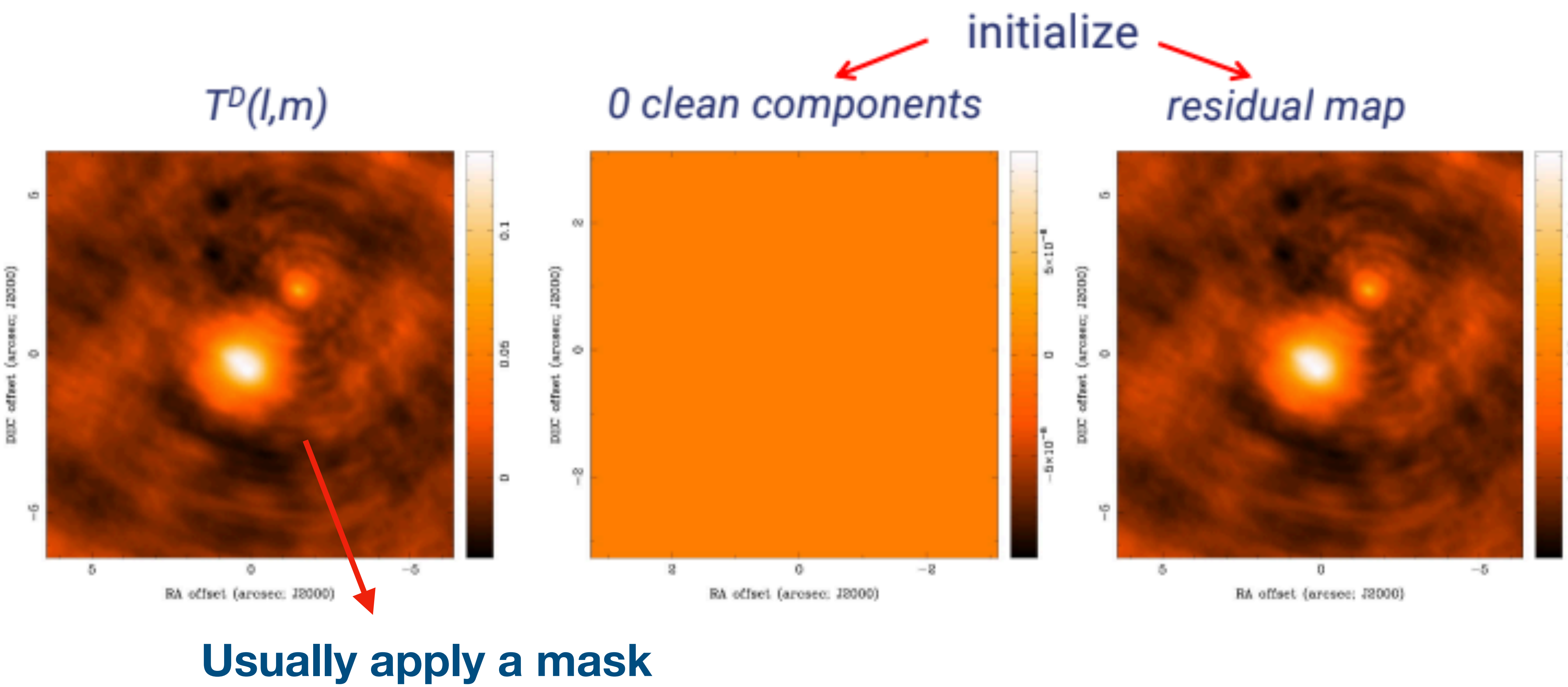
Brief aside on Cleaning

- Colloquial overview
  - Takes components out of the 'dirty image' and saves to a clean model
  - Can have major/minor cycles, visibility plane/ image plane 'cleaning' - step wise deconvolving
  - Stops at a stopping criteria *(niter or threshold)*
  - Should use 'masked' regions to avoid selection of artefacts or negatives
  - Clean model is convolved with clean beam → "representative" image

- Different algorithms behind the scenes:
  - **Hogbom** - default
  - Clark
  - Multi-scale

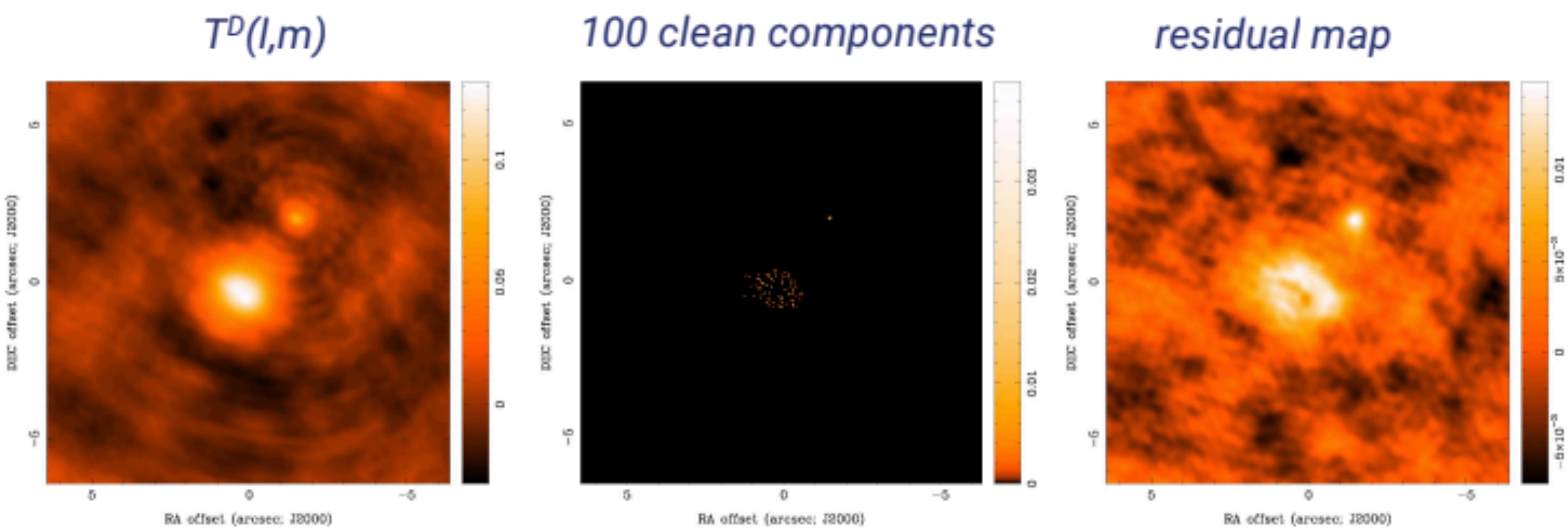
CASA docs.: https://casa.nrao.edu/casadocs/casa-6.1.0/imaging/synthesis-imaging/deconvolution-algorithms

Brief aside on Cleaning



Credits: Wilner - SMA

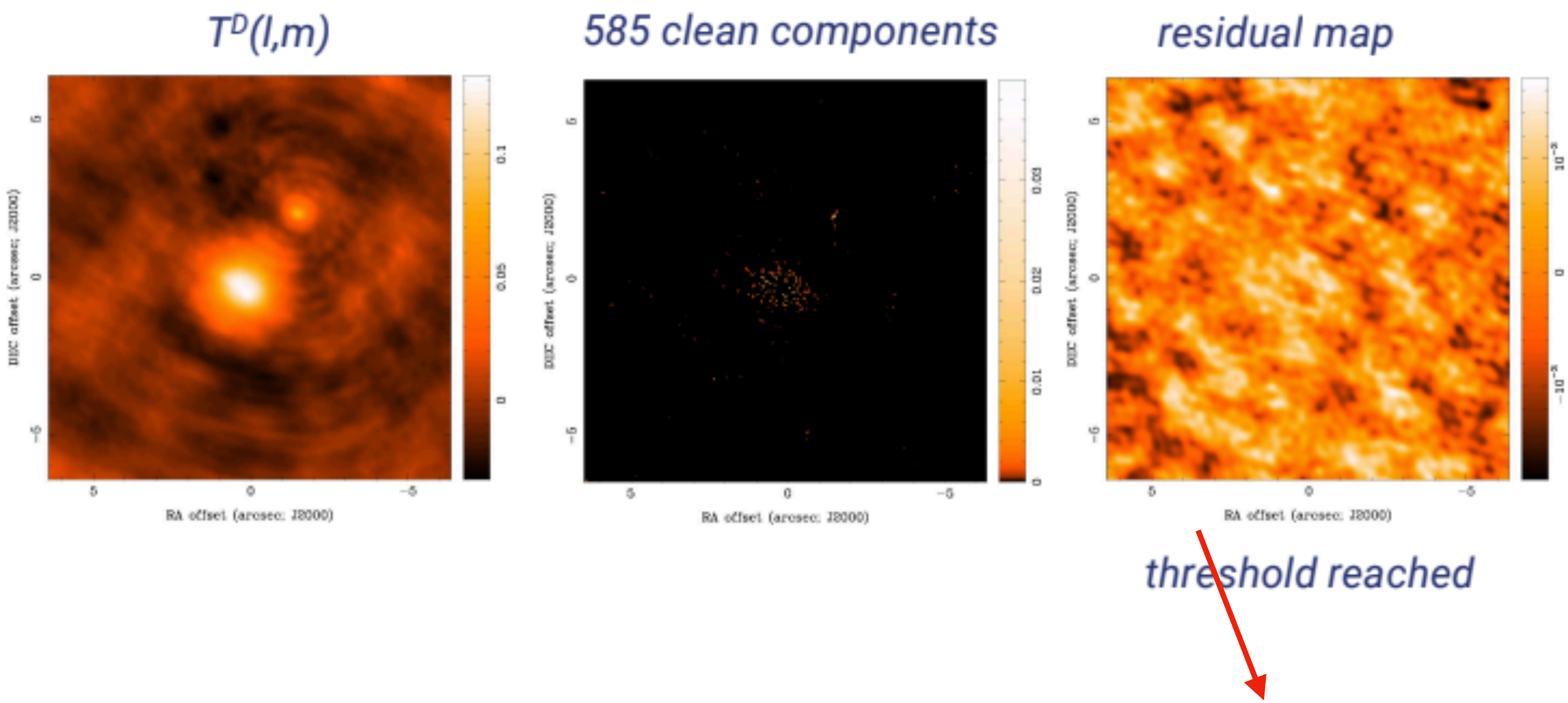
Brief aside on Cleaning



Could ‘interactively’ refine the mask

Credits: Wilner - SMA

Brief aside on Cleaning

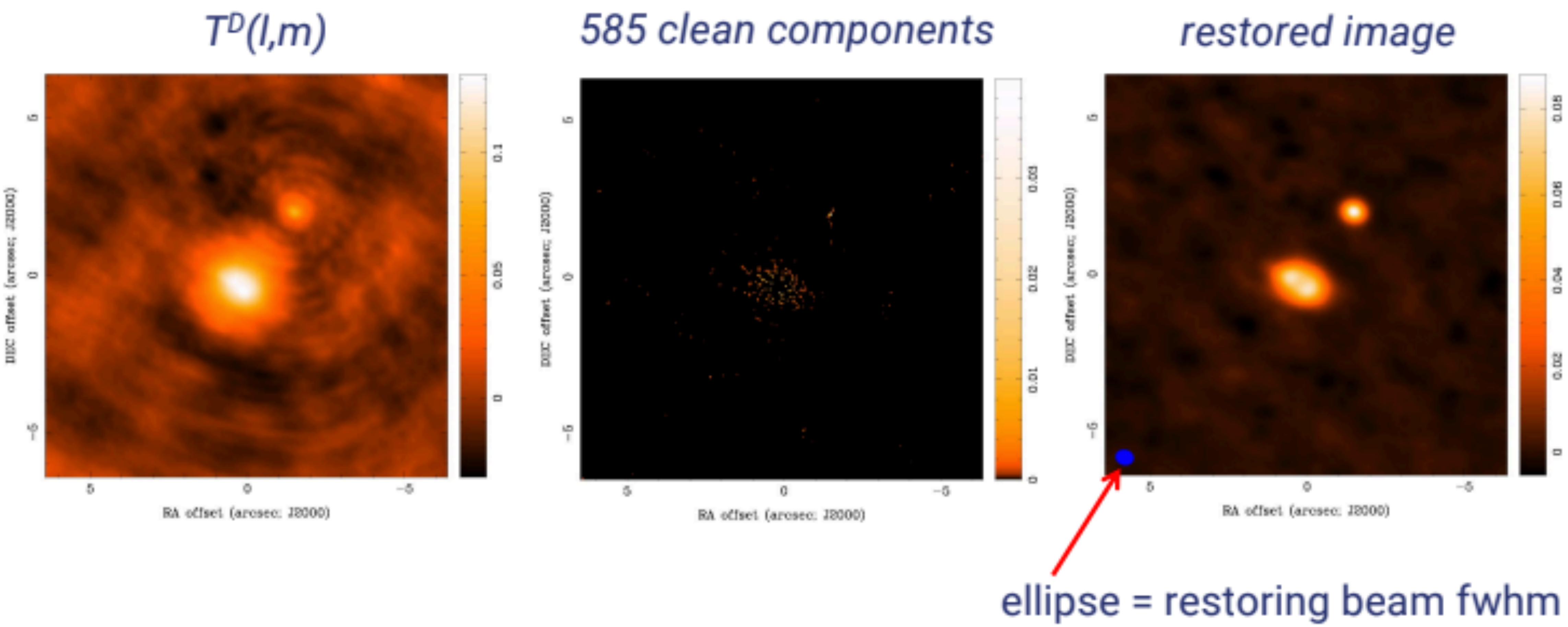


Should look like noise - then we are done

Credits: Wilner - SMA

EUROPEAN ARC
ALMA Regional Centre

Brief aside on Cleaning



'clean' beam

Credits: Wilner - SMA

Some Schools and Material *(not exhaustive)*

- IRAM summer school: https://www.iram-institute.org/EN/content-page-399-7-67-367-399-0.html
- European Radio interferometry school - ERIS: https://www.jive.eu/eris2022/
- SMA interferometry school: https://lweb.cfa.harvard.edu/sma-school/program/
- NRAO synthesis imaging workshop: https://science.nrao.edu/science/meetings/2018/16th-synthesis-imaging-workshop/16th-synthesis-imaging-workshop-lectures
- Myers - Imaging in CASA: https://slideplayer.com/slide/7964345/
- UK ARC node - line imaging tutorial: https://www.alma.ac.uk/index.php/meeting-supplemental-material/286-spectral-line-imaging-tutorial
- ALMA primer series basic concepts videos: https://www.youtube.com/channel/UCwTfiIlYuUQr4sRc5iSJaRg/videos
- Lecture course: https://www.astron.nl/astrowiki/doku.php?id=uva_msc_radioastronomy_2013

Questions so far?

Preparing for imaging

- Downloaded products can be from **manual** or **pipeline** imaging
- It is important to note:
    - There is no reason to assume better/worse quality based on method
    - Manual images (in earlier cycles) are more specific to make a quality assessment based on the science goal - *i.e. lower level of completeness* - only **mfs** per SPW, some cubes - focussed on line emission only (not full SPW) **-> hence ARI-L**
    - Pipeline is systematically making mfs per SPW, combined cont, full SPW cubes
        - If mitigation occurs products then become limited - less sources, smaller images, 'missing' cubes
    - ***Can do 'science' but images not assessed for scientific 'correctness'***
- **REMEMBER** - you do not download calibrated data, you need to remake it before imaging
    - or use the calMS service - ARCs run the calibration and deliver a calibrated dataset as a queue based system
    - *Cycle 8+ scriptForPI.py also runs some preparation for starting imaging for PL data*

https://almascience.eso.org/tools/eu-arc-network/the-european-arc-calms-service

Manually imaged data

- Recall "ALMA science Archive Content" talk
- After unpacking the download, e.g.:
    - 2018.1.01131.S

```
science_goal.uid___A001_X135b_X60
    group.uid___A001_X13b_X68
        member.uid___A001_X135b_X6b
            calibration
            log
            product
            qa
            raw
            script
```

Manually imaged data

- Recall “ALMA science Archive Content” talk
- After unpacking the download, e.g.:
  - 2018.1.01131.S

        science_goal.uid___A001_X135b_X60
            group.uid___A001_X13b_X68
                member.uid___A001_X135b_X6b
                    calibration
                    log
                    product
                    qa
                    raw
                    script

script → move here and in case run > **execfile(‘scriptForPI.py’)**

→ this runs manual calibration scripts actively for each EB

Manually imaged data can have been either manually or pipeline calibrated, later imaging can be made regardless to the calibration method

Manually imaged data

- Recall "ALMA science Archive Content" talk
- After unpacking the download, e.g.:
  - 2018.1.01131.S
    - science_goal.uid___A001_X135b_X60
      - group.uid___A001_X13b_X68
        - member.uid___A001_X135b_X6b
          - calibration
          - log
          - product
          - qa
          - raw
          - script
          - calibrated → is created - imaging can take place here, scripts will have an imaging script (not run automatically)

Manually imaged data

- Usually older cycles, or difficult for Pipeline to image
- EU delivered data will usually have a step-wise imaging script:

```
# ALMA Data Reduction Script

# Imaging

thesteps = []
step_title = {0: 'Transform SPW 0 for all MSs to common largest LSRK grid before concat',
              1: 'Transform SPW 1 for all MSs to common largest LSRK grid before concat',
              2: 'Transform SPW 2 for all MSs to common largest LSRK grid before concat',
              3: 'Transform SPW 3 for all MSs to common largest LSRK grid before concat',
              4: 'Transform SPW 4 for all MSs to common largest LSRK grid before concat',
              5: 'Transform SPW 5 for all MSs to common largest LSRK grid before concat',
              6: 'Concatenation of mstransformed MSs',
              7: 'Agg. bandwidth image for non-science target J1650-5044 (intent OBSERVE_CHECK_SOURCE), spws [0, 1, 2, 3, 4, 5]',
              8: 'Continuum image for target G333.6-0.2, spws [0, 1, 2, 3, 4, 5]',
              9: 'Continuum subtraction for field G333.6-0.2',
              10: 'Cube for target G333.6-0.2, spw 0',
              11: 'Cube for target G333.6-0.2, spw 1',
              12: 'Cube for target G333.6-0.2, spw 2',
              13: 'Cube for target G333.6-0.2, spw 3',
              14: 'Cube for target G333.6-0.2, spw 4',
              15: 'Cube for target G333.6-0.2, spw 5',
              16: 'Export images to FITS format'}

if 'applyonly' not in globals(): applyonly = False
try:
  print 'List of steps to be executed ...', mysteps
  thesteps = mysteps
except:
  print 'global variable mysteps not set.'
if (thesteps==[]):
  thesteps = range(0,len(step_title))
  print 'Executing all steps: ', thesteps

# The Python variable 'mysteps' will control which steps
# are executed when you start the script using
#   execfile('scriptForCalibration.py')
# e.g. setting
#   mysteps = [2,3,4]
# before starting the script will make the script execute
# only steps 2, 3, and 4
# Setting mysteps = [] will make it execute all steps.


thevis = ['uid___A002_Xc1d834_Xda3.ms.split.cal',
          'uid___A002_Xc1d834_X1186.ms.split.cal',
          'uid___A002_Xc1e2be_X401.ms.split.cal']

# put restfreqs in this dictionary,
# one for each SPW ID, e.g. {17: '350GHz', 19: '356GHz'}
therestfreqs = {0: '241.016088GHz',
                1: '244.935556GHz',
```

list of all steps to do

align data in LSRK frame for multi-EBs before concat

continuum image

continuum subtraction

cube images

Initial parameters

Manually imaged data

- Usually older cycles, or difficult for Pipeline to image
- EU delivered data will usually have a step-wise imaging script:

```
# put restfreqs in this dictionary,
# one for each SPW ID, e.g. {17: '350GHz', 19: '356GHz'}
therestfreqs = {0: '241.016088GHz',
                1: '244.935556GHz',
                2: '243.915826GHz',
                3: '257.35GHz',
                4: '256.302035GHz',
                5: '260.35GHz'}

thevislsrk = ['uid___A002_Xc1d834_Xda3.ms.split.cal.lsrk.spw0',
              'uid___A002_Xc1d834_Xda3.ms.split.cal.lsrk.spw1',
              'uid___A002_Xc1d834_Xda3.ms.split.cal.lsrk.spw2',
              'uid___A002_Xc1d834_Xda3.ms.split.cal.lsrk.spw3',
              'uid___A002_Xc1d834_Xda3.ms.split.cal.lsrk.spw4',
              'uid___A002_Xc1d834_Xda3.ms.split.cal.lsrk.spw5',
              'uid___A002_Xc1d834_X1186.ms.split.cal.lsrk.spw0',
              'uid___A002_Xc1d834_X1186.ms.split.cal.lsrk.spw1',
              'uid___A002_Xc1d834_X1186.ms.split.cal.lsrk.spw2',
              'uid___A002_Xc1d834_X1186.ms.split.cal.lsrk.spw3',
              'uid___A002_Xc1d834_X1186.ms.split.cal.lsrk.spw4',
              'uid___A002_Xc1d834_X1186.ms.split.cal.lsrk.spw5',
              'uid___A002_Xc1e2be_X401.ms.split.cal.lsrk.spw0',
              'uid___A002_Xc1e2be_X401.ms.split.cal.lsrk.spw1',
              'uid___A002_Xc1e2be_X401.ms.split.cal.lsrk.spw2',
              'uid___A002_Xc1e2be_X401.ms.split.cal.lsrk.spw3',
              'uid___A002_Xc1e2be_X401.ms.split.cal.lsrk.spw4',
              'uid___A002_Xc1e2be_X401.ms.split.cal.lsrk.spw5']


# Transform SPW 0 for all MSs to common largest LSRK grid before concat
mystep = 0
if(mystep in thesteps):
  casalog.post('Step '+str(mystep)+' '+step_title[mystep],'INFO')
  print 'Step ', mystep, step_title[mystep]

  for myvis in thevis:
    os.system('rm -rf '+myvis+'.lsrk.spw0')
    mstransform(vis = myvis,
                outputvis = myvis+'.lsrk.spw0',
                outframe = 'LSRK',
                spw = '0',
                mode = 'frequency',
                nchan = 3837,
                width = '122.082kHz',
                start = '240.827180449GHz',
                regridms = True,
                datacolumn = 'data',
                reindex = True
                )
```

Important: `tclean` **can handle input of multiple EBs and ensures the EBs velocities are correctly aligned. For the continuum step this is fine and the User can also simply select the SPW and CHANNELS required to image:**

e.g. spw='0:10~495,1:10~200;300~450,2:50~…….'

However, prior to continuum subtraction and line imaging it is easier to concatenate the EBs and just have one. In this case rather than 'forcing' together the EBs - where the channels might be very slightly shifted due to TOPO observing frame to LSRK transition, it is useful to convert all EBs to LSRK then combine. This way ever SPW and CHANNEL are aligned per EB, and you only ever have to track the original SPW numbering.

Manually imaged data

- Continuum image

The setting for mask **parameterization, is** usemask **= 'user' (default)**

For auto-masking add parameter usemask **= 'auto-multithresh'**

```
# Continuum image for target G333.6-0.2, spws [0, 1, 2, 3, 4, 5]
mystep = 8
if(mystep in thesteps):
  casalog.post('Step '+str(mystep)+' '+step_title[mystep],'INFO')
  print 'Step ', mystep, step_title[mystep]


  os.system('rm -rf G333.6-0.2_sci.spw0_1_2_3_4_5.cont.I.manual*')
  tclean(vis = thevis,
         imagename = 'G333.6-0.2_sci.spw0_1_2_3_4_5.cont.I.manual',
         field = 'G333.6-0.2',
         stokes = 'I',
         spw = '0:10~495,1:10~200;300~450,2:50~500,3,4,5:10~250',
         outframe = 'LSRK',
         specmode = 'cont',
         nterms = 1,
         imsize = [1372, 1372],
         cell = '0.026arcsec',
         deconvolver = 'hogbom',
         niter = 100,
         weighting = 'briggs',
         robust = 0.5,
         mask = '',
         gridder = 'standard',
         pbcor = True,
         threshold = '',
         interactive = True
         )
```

remove, in case it already exists (CAUTION)

list of EBs from top of script (ok for cont. images)

channels with only 'continuum' (can be frequency)

'cont' or 'mfs'

image parameters

deconvolution parameters

weighting parameters

set to true, while mask = '' (none)
- means User can (re)draw and clean each major cycle

Manually imaged data

- Continuum subtraction

```
# Continuum subtraction for field G333.6-0.2
mystep = 9
if(mystep in thesteps):
  casalog.post('Step '+str(mystep)+' '+step_title[mystep],'INFO')
  print 'Step ', mystep, step_title[mystep]

  uvcontsub(vis = 'concat.ms',
            field = 'G333.6-0.2',
            spw = '0,1,2,3,4,5',
            fitspw = '0:10~495,1:10~200;300~450,2:50~500,3,4,5:10~250',
            fitorder = 1,
            solint = 'int',
            combine = '',
            excludechans = False,
            want_cont = False
            )

  os.system('rm -rf concat.ms_G333.6-0.2.contsub')
  os.system('mv concat.ms.contsub concat.ms_G333.6-0.2.contsub')
```

→ vis = 'concat.ms': using the concat file (all EBs are merged)

→ spw: SPW to include in output

→ fitspw: SPW selected as continuum - ***to fit***

→ excludechans: if `False` - fitspw selection is the continuum, if `True`, fitspw should be the lines (to exclude)

→ want_cont: make an output file that has only continuum in or not

→ os.system: remove what you want to call the output and move the '.contsub' to that name

Manually imaged data

- Line Cube imaging

```
# Cube for target G333.6-0.2, spw 0
mystep = 10
if(mystep in thesteps):
  casalog.post('Step '+str(mystep)+' '+step_title[mystep],'INFO')
  print 'Step ', mystep, step_title[mystep]

  os.system('rm -rf G333.6-0.2_sci.spw0.cube.I.manual*')
  tclean(vis = 'concat.ms_G333.6-0.2.contsub',
         imagename = 'G333.6-0.2_sci.spw0.cube.I.manual',
         field = 'G333.6-0.2',
         stokes = 'I',
         spw = '0',
         outframe = 'LSRK',
         restfreq = therestfreqs[0],
         specmode = 'cube',
         imsize = [1372, 1372],
         cell = '0.026arcsec',
         deconvolver = 'hogbom',
         niter = 100,
         weighting = 'briggs',
         robust = 0.5,
         mask = '',
         gridder = 'standard',
         pbcor = True,
         threshold = '1.0mJy',
         width = '1.0km/s',
         start = '-80km/s',
         nchan = 160,
         interactive = True
         )
```

remove, in case it already exists (CAUTION)

concat dataset after contsub

preset at start of script - line of interest

'cube' image

image, deconvolver, weight parameters

cube parameter setup - km/s in combination with correct restfreq setting

set to true, while mask = '' (none)
- means User can (re)draw and clean each major cycle
- each channel could need a mask

Manually imaged data

- Line Cube imaging

```
# Cube for target G333.6-0.2, spw 0
mystep = 10
if(mystep in thesteps):
  casalog.post('Step '+str(mystep)+' '+step_title[mystep],'INFO')
  print 'Step ', mystep, step_title[mystep]

  os.system('rm -rf G333.6-0.2_sci.spw0.cube.I.manual*')
  tclean(vis = 'concat.ms_G333.6-0.2.contsub',
         imagename = 'G333.6-0.2_sci.spw0.cube.I.manual',
         field = 'G333.6-0.2',
         stokes = 'I',
         spw = '0',
         outframe = 'LSRK',
         restfreq = therestfreqs[0],
         specmode = 'cube',
         imsize = [1372, 1372],
         cell = '0.026arcsec',
         deconvolver = 'hogbom',
         niter = 100,
         weighting = 'briggs',
         robust = 0.5,
         mask = '',
         gridder = 'standard',
         pbcor = True,
         threshold = '1.0mJy',
         width = '1.0km/s',
         start = '-80km/s',
         nchan = 160,
         interactive = True
         )
```

can set to whatever you want

km/s or freq can be used

The setting for mask **parameterization, is** usemask **= 'user' (default)**

For auto-masking add parameter usemask **= 'auto-multithresh'**

User can image a full cube and then cut out channel ranges later, or just set to image a few 10s of channels per line of interest

Manually imaged data

- The previous full script can be generated with the EU Image Script Generator

  https://confluence.alma.cl/pages/viewpage.action?pageId=72165197

- Other possible templates, e.g. NRAO

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

- **User can copy, paste, edit existing commands**
  - use in the CASA command line
  - pre-set variables for cell or image size etc.
  - run clean loops and non-interactive imaging changing e.g. robust values
  - can set auto-masking

Automasking parameters

- setting **usemask** = 'auto-multithresh' activates the automatic masking
- 'unlocks' the extra variables:

```
usemask              = 'auto-multithresh'     # Type of mask(s) for deconvolution: user, pb, or auto-multithresh
   pbmask            = 0.2                    # primary beam mask
   sidelobethreshold = 3.0                    # sidelobethreshold * the max sidelobe level * peak residual
   noisethreshold    = 5.0                    # noisethreshold * rms in residual image + location(median)
   lownoisethreshold = 1.5                    # lownoisethreshold * rms in residual image + location(median)
   negativethreshold = 0.0                    # negativethreshold * rms in residual image + location(median)
   smoothfactor      = 1.0                    # smoothing factor in a unit of the beam
   minbeamfrac       = 0.3                    # minimum beam fraction for pruning
   cutthreshold      = 0.01                   # threshold to cut the smoothed mask to create a final mask
   growiterations    = 75                     # number of binary dilation iterations for growing the mask
   dogrowprune       = True                   # Do pruning on the grow mask
   minpercentchange  = -1.0                   # minimum percentage change in mask size (per channel plane) to trigger updating of ma
   verbose           = False                  # True: print more automasking information in the logger
```

- Generally, the **sidelobethreshold**, **noisethreshold**, **lownoisethrehsold** are those to change (slightly) for different behaviour in the masking
- NOTE: standard parameter of **niter** and **threshold, interactive** must be set accordingly to allow auto-masking to work:
  - **niter** = 99999 (so this doesn't stop clean)
  - **threshold** = ~2-3x sigma (of the map noise away from real/strong signal - in a dirty image)
  - **interactive** = False *(or True to steer the masking, and leave* threshold *unset)*

MUST READ - NRAO guide: https://casaguides.nrao.edu/index.php/Automasking_Guide

Questions so far?

Pipeline imaged data

https://almascience.eso.org/euarcdata/itrain01/I-TRAIN-01_Imaging_Pipeline_Tafoya.pdf

https://www.youtube.com/watch?v=TqqI8IhvyPE

- Recall “ALMA science Archive Content” talk
- After unpacking the download, e.g.:
  - 2018.1.01201.S

    science_goal.uid___A001_X133d_X2c85

    group.uid___A001_X133d_X2c86

    member.uid___A001_X133d_X2c8b

    calibration

    log

    product

    qa

    raw

    script

Pipeline imaged are always pipeline calibrated

move here and in case run > **execfile(‘scriptForPI.py’)**

you need “CASA -- pipeline” version

this runs a pipeline **RESTORE**, not the actual Pipeline again

Pipeline imaged data

> casapy -- pipeline

```
optional configuration file config.py not found, continuing CASA startup without it
Using user-supplied startup.py at /home/lmaud/.lmtst.mycasa/startup.py

IPython 7.15.0 -- An enhanced Interactive Python.

Creating a new telemetry file
Telemetry initialized. Telemetry will send anonymized usage statistics to NRAO.
You can disable telemetry by adding the following line to the config.py file in your rcdir (e.g. ~/.casa/config.py):
telemetry_enabled = False
--> CrashReporter initialized.
2022-09-30 07:31:04 INFO: Environment is not MPI enabled. Pipeline operating in single host mode
casaVersion =  6.4.1.12
2022-09-30 07:31:05 INFO: Environment variable FLUX_SERVICE_URL not defined.  Switching to backup url.
2022-09-30 07:31:05 INFO: Environment variable FLUX_SERVICE_URL_BACKUP not defined.
2022-09-30 07:31:06 INFO: Pipeline version 2022.2.0.64 running on arcp22.hq.eso.org
2022-09-30 07:31:06 INFO: Host environment: 251.6 GiB memory, 12 x Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz running
)
2022-09-30 07:31:06 INFO: Initializing cli...
2022-09-30 07:31:06 INFO: Loaded CASA tasks from package: h
2022-09-30 07:31:06 INFO: Loaded CASA tasks from package: hif
2022-09-30 07:31:06 INFO: Loaded CASA tasks from package: hifa
2022-09-30 07:31:06 INFO: Loaded CASA tasks from package: hifas
2022-09-30 07:31:06 INFO: Loaded CASA tasks from package: hifv
2022-09-30 07:31:06 INFO: Loaded CASA tasks from package: hsd
2022-09-30 07:31:06 INFO: Loaded CASA tasks from package: hsdn
*** You are not working in a subdirectory of /lustre/opsw/work ... ***
*** startup.py: QA2-relevant modules will be imported ***
imported casatasks and casatools individually
$Id: analysisUtils.py,v 1.5325 2022/09/29 15:58:38 thunter Exp $
CASA 6.4.1.12 -- Common Astronomy Software Applications [6.4.1.12]

CASA <1>:
```

Pipeline has correctly loaded

Pipeline imaged data

- Recall "ALMA science Archive Content" talk
- After unpacking the download, e.g.:
  - 2018.1.01201.S

    science_goal.uid___A001_X133d_X2c85

    group.uid___A001_X133d_X2c86

    member.uid___A001_X133d_X2c8b

    calibration

    log

    product

    qa

    raw

    script

    calibrated

Be aware

cal & image combined

*…*hifa_calimage.casa_piperestorescript.py*

*…*hifa_calimage.casa_pipescirpt.py*

cal & image separate:

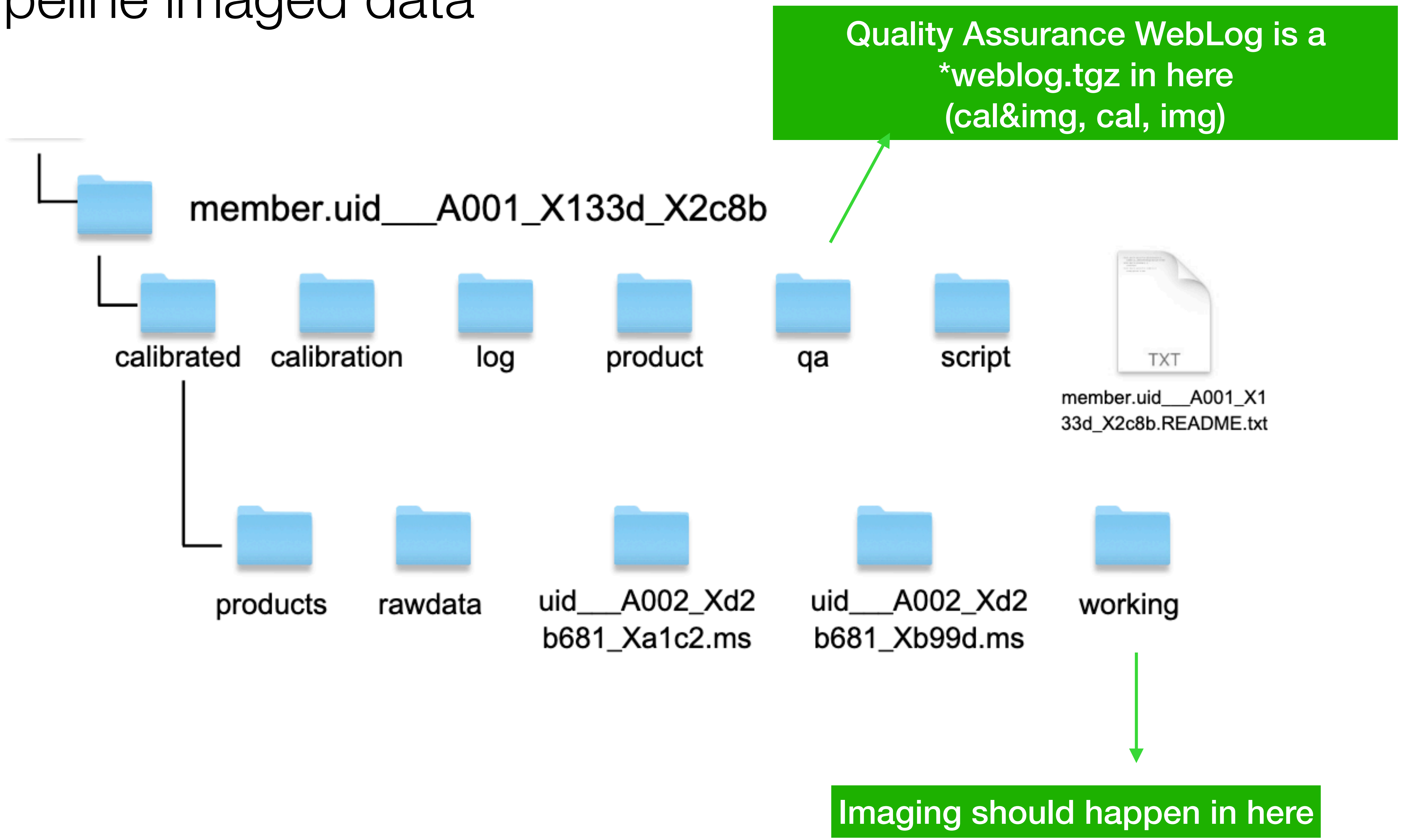
*…*hifa_cal.casa_piperestorescript.py*

*…*hifa_cal.casa_pipescirpt.py*

*…*hifa_image.casa_pipescirpt.py*

is created - and directories within

Pipeline imaged data



Pipeline imaged data

Calibration & Image PL run combined

```
from casarecipes.almahelpers import fixsyscaltimes # SACM/JAO - Fixes
from casatasks import fixplanets
__rethrow_casa_exceptions = True
context = h_init()
context.set_state('ProjectSummary', 'proposal_code', '2019.1.00097.S')
context.set_state('ProjectSummary', 'proposal_title', 'unknown')
context.set_state('ProjectSummary', 'piname', 'unknown')
context.set_state('ProjectStructure', 'ous_entity_id', 'uid://A001/X13b9/Xbf')
context.set_state('ProjectStructure', 'ous_part_id', 'X928641954')
context.set_state('ProjectStructure', 'ous_title', 'Undefined')
context.set_state('ProjectStructure', 'ps_entity_id', 'uid://A001/X13b9/Xc3')
context.set_state('ProjectStructure', 'ousstatus_entity_id', 'uid://A001/X1528/X2a2')
context.set_state('ProjectStructure', 'ppr_file', '/opt/pipelinedriver/2021FEB/mnt/dataproc/2019.1.00097.S_2021_
__A001_X1528_X2a0/GOUS_uid___A001_X1528_X2a1/MOUS_uid___A001_X1528_X2a2/working/PPR_uid___A001_X1528_X2a3.xml')
context.set_state('ProjectStructure', 'recipe_name', 'hifa_calimage')
try:
    hifa_importdata(dbservice=True, vis=['uid___A002_Xed07bd_X15e'], session=['session_1'])
    fixsyscaltimes(vis='uid___A002_Xed07bd_X15e.ms') # SACM/JAO - Fixes
    h_save() # SACM/JAO - Finish weblog after fixes
    h_init() # SACM/JAO - Restart weblog after fixes
    hifa_importdata(dbservice=True, vis=['uid___A002_Xed07bd_X15e'], session=['session_1'])
    hifa_flagdata(pipelinemode="automatic")
    hifa_fluxcalflag(pipelinemode="automatic")
    hif_rawflagchans(pipelinemode="automatic")
    hif_refant(pipelinemode="automatic")
    h_tsyscal(pipelinemode="automatic")
    hifa_tsysflag(pipelinemode="automatic")
    hifa_antpos(pipelinemode="automatic")
    hifa_wvrgcalflag(pipelinemode="automatic")
    hif_lowgainflag(pipelinemode="automatic")
    hif_setmodels(pipelinemode="automatic")
    hifa_bandpassflag(pipelinemode="automatic")
    hifa_bandpass(pipelinemode="automatic")
    hifa_spwphaseup(pipelinemode="automatic")
    hifa_gfluxscaleflag(pipelinemode="automatic")
    hifa_gfluxscale(pipelinemode="automatic")
    hifa_timegaincal(pipelinemode="automatic")
    hifa_targetflag(pipelinemode="automatic")
    hif_applycal(pipelinemode="automatic")
    hif_makeimlist(intent='PHASE,BANDPASS,AMPLITUDE')
    hif_makeimages(pipelinemode="automatic")
    hif_makeimlist(intent='CHECK', per_eb=True)
    hif_makeimages(pipelinemode="automatic")
    hifa_imageprecheck(pipelinemode="automatic")
    hif_checkproductsize(maxcubesize=40.0, maxcubelimit=60.0, maxproductsize=350.0)
    hifa_exportdata(pipelinemode="automatic")
    hif_mstransform(pipelinemode="automatic")
    hifa_flagtargets(pipelinemode="automatic")
    hif_makeimlist(specmode='mfs')
    hif_findcont(pipelinemode="automatic")
    hif_uvcontfit(pipelinemode="automatic")
    hif_uvcontsub(pipelinemode="automatic")
    hif_makeimages(pipelinemode="automatic")
    hif_makeimlist(specmode='cont')
    hif_makeimages(pipelinemode="automatic")
    hif_makeimlist(specmode='cube')
```

Calibration & Image PL run SEPARATE

CAL

```
__rethrow_casa_exceptions = True
context = h_init()
context.set_state('ProjectSummary', 'proposal_code', '2018.1.00526.S')
context.set_state('ProjectSummary', 'piname', 'unknown')
context.set_state('ProjectSummary', 'proposal_title', 'unknown')
context.set_state('ProjectStructure', 'ous_part_id', 'X446126180')
context.set_state('ProjectStructure', 'ous_title', 'Undefined')
context.set_state('ProjectStructure', 'ppr_file', '/opt/dared/opt/dared.2018AUG/mnt/dataproc/2
2019_01_03T12_52_58.189/SOUS_uid___A001_X133d_X1ec3/GOUS_uid___A001_X133d_X1ec7/MOUS_uid___A00
working/PPR_uid___A001_X133d_X1ec9.xml')
context.set_state('ProjectStructure', 'ps_entity_id', 'uid://A001/X12eb/X270')
context.set_state('ProjectStructure', 'recipe_name', 'hifa_cal')
context.set_state('ProjectStructure', 'ous_entity_id', 'uid://A001/X12eb/X26c')
context.set_state('ProjectStructure', 'ousstatus_entity_id', 'uid://A001/X133d/X1ec8')
try:
    hifa_importdata(vis=['uid___A002_Xd704f8_Xf10b'], dbservice=False, session=['session_1'])
    hifa_flagdata(pipelinemode="automatic")
    hifa_fluxcalflag(pipelinemode="automatic")
    hif_rawflagchans(pipelinemode="automatic")
    hif_refant(pipelinemode="automatic")
    h_tsyscal(pipelinemode="automatic")
    hifa_tsysflag(pipelinemode="automatic")
    hifa_antpos(pipelinemode="automatic")
    hifa_wvrgcalflag(pipelinemode="automatic")
    hif_lowgainflag(pipelinemode="automatic")
    hif_setmodels(pipelinemode="automatic")
    hifa_bandpassflag(pipelinemode="automatic")
    hifa_spwphaseup(pipelinemode="automatic")
    hifa_gfluxscaleflag(pipelinemode="automatic")
    hifa_gfluxscale(pipelinemode="automatic")
    hifa_timegaincal(pipelinemode="automatic")
    hif_applycal(pipelinemode="automatic")
    hif_makeimlist(intent='PHASE,BANDPASS')
    hif_makeimages(pipelinemode="automatic")
    hif_makeimlist(per_eb=True, intent='CHECK')
    hif_makeimages(pipelinemode="automatic")
    hifa_imageprecheck(pipelinemode="automatic")
    hif_checkproductsize(maxproductsize=350.0, maxcubesize=40.0, maxcubelimit=60.0)
finally:
    h_save()
```

IMAGING

```
__rethrow_casa_exceptions = True
context = h_init()
context.set_state('ProjectSummary', 'proposal_code', '2018.1.00526.S')
context.set_state('ProjectSummary', 'piname', 'unknown')
context.set_state('ProjectSummary', 'proposal_title', 'unknown')
context.set_state('ProjectStructure', 'ous_part_id', 'X446126180')
context.set_state('ProjectStructure', 'ous_title', 'Undefined')
context.set_state('ProjectStructure', 'ppr_file', '/opt/dared/opt/dared.2018OCT/mnt/dat
01_X133d_X1ec9.xml')
context.set_state('ProjectStructure', 'ps_entity_id', 'uid://A001/X12eb/X270')
context.set_state('ProjectStructure', 'recipe_name', 'hifa_image')
context.set_state('ProjectStructure', 'ous_entity_id', 'uid://A001/X12eb/X26c')
context.set_state('ProjectStructure', 'ousstatus_entity_id', 'uid://A001/X133d/X1ec8')
try:
    hifa_restoredata(vis=['/opt/dared/opt/dared.2018OCT/mnt/dataproc/2018.1.00526.S_201
['session_1'], copytoraw=False)
    hif_mstransform(pipelinemode="automatic")
    hifa_flagtargets(pipelinemode="automatic")
    hifa_imageprecheck(pipelinemode="automatic")
    hif_checkproductsize(maxproductsize=350.0, maxcubesize=40.0, maxcubelimit=60.0)
    hif_makeimlist(specmode='mfs')
    hif_findcont(pipelinemode="automatic")
    hif_uvcontfit(pipelinemode="automatic")
    hif_uvcontsub(pipelinemode="automatic")
    hif_makeimages(pipelinemode="automatic")
    hif_makeimlist(specmode='cont')
    hif_makeimages(pipelinemode="automatic")
    hif_makeimlist(specmode='cube')
    hif_makeimages(pipelinemode="automatic")
    hif_makeimlist(specmode='repBW')
    hif_makeimages(pipelinemode="automatic")
finally:
    h_save()
```

Pipeline imaged data

Calibration & Image PL run combined

```
from casarecipes.almahelpers import fixsyscaltimes # SACM/JAO - Fixes
from casatasks import fixplanets
__rethrow_casa_exceptions = True
context = h_init()
context.set_state('ProjectSummary', 'proposal_coc
context.set_state('ProjectSummary', 'proposal_tit
context.set_state('ProjectSummary', 'piname', 'ur
context.set_state('ProjectStructure', 'ous_entity
context.set_state('ProjectStructure', 'ous_part_i
context.set_state('ProjectStructure', 'ous_title'
context.set_state('ProjectStructure', 'ps_entity_
context.set_state('ProjectStructure', 'ousstatus_
context.set_state('ProjectStructure', 'ppr_file',
__A001_X1528_X2a0/GOUS_uid___A001_X1528_X2a1/MOUS
context.set_state('ProjectStructure', 'recipe_nam
try:
    hifa_importdata(dbservice=True, vis=['uid___A
    fixsyscaltimes(vis='uid___A002_Xed07bd_X15e.m
    h_save() # SACM/JAO - Finish weblog after fix
    h_init() # SACM/JAO - Restart weblog after fi
    hifa_importdata(dbservice=True, vis=['uid___A002_Xed07bd_X15e'], session=['session_1'])
    hifa_flagdata(pipelinemode="automatic")
    hifa_fluxcalflag(pipelinemode="automatic")
    hif_rawflagchans(pipelinemode="automatic")
    hif_refant(pipelinemode="automatic")
    h_tsyscal(pipelinemode="automatic")
    hifa_tsysflag(pipelinemode="automatic")
    hifa_antpos(pipelinemode="automatic")
    hifa_wvrgcalflag(pipelinemode="automatic")
    hif_lowgainflag(pipelinemode="automatic")
    hif_setmodels(pipelinemode="automatic")
    hifa_bandpassflag(pipelinemode="automatic")
    hifa_bandpass(pipelinemode="automatic")
    hifa_spwphaseup(pipelinemode="automatic")
    hifa_gfluxscaleflag(pipelinemode="automatic")
    hifa_gfluxscale(pipelinemode="automatic")
    hifa_timegaincal(pipelinemode="automatic")
    hifa_targetflag(pipelinemode="automatic")
    hif_applycal(pipelinemode="automatic")
    hif_makeimlist(intent='PHASE,BANDPASS,AMPLITUDE')
    hif_makeimages(pipelinemode="automatic")
    hif_makeimlist(intent='CHECK', per_eb=True)
    hif_makeimages(pipelinemode="automatic")
    hifa_imageprecheck(pipelinemode="automatic")
    hif_checkproductsize(maxcubesize=40.0, maxcubelimit=60.0, maxproductsize=350.0)
    hifa_exportdata(pipelinemode="automatic")
    hif_mstransform(pipelinemode="automatic")
    hifa_flagtargets(pipelinemode="automatic")
    hif_makeimlist(specmode='mfs')
    hif_findcont(pipelinemode="automatic")
    hif_uvcontfit(pipelinemode="automatic")
    hif_uvcontsub(pipelinemode="automatic")
    hif_makeimages(pipelinemode="automatic")
    hif_makeimlist(specmode='cont')
    hif_makeimages(pipelinemode="automatic")
    hif_makeimlist(specmode='cube')
```

Calibration & Image PL run SEPARATE

CAL

```
__rethrow_casa_exceptions = True
context = h_init()
context.set_state('ProjectSummary', 'proposal_code', '2018.1.00526.S')
context.set_state('ProjectSummary', 'piname', 'unknown')
context.set_state('ProjectSummary', 'proposal_title', 'unknown')
context.set_state('ProjectStructure', 'ous_part_id', 'X446126180')
context.set_state('ProjectStructure', 'ous_title', 'Undefined')
context.set_state('ProjectStructure', 'ppr_file', '/opt/dared/opt/dared.2018AUG/mnt/dataproc/2
2019_01_03T12_52_58.189/SOUS_uid___A001_X133d_X1ec3/GOUS_uid___A001_X133d_X1ec7/MOUS_uid___A00
working/PPR_uid___A001_X133d_X1ec9.xml')
context.set_state('ProjectStructure', 'ps_entity_id', 'uid://A001/X12eb/X270')
context.set_state('ProjectStructure', 'recipe_name', 'hifa_cal')
context.set_state('ProjectStructure', 'ous_entity_id', 'uid://A001/X12eb/X26c')
context.set_state('ProjectStructure', 'ousstatus_entity_id', 'uid://A001/X133d/X1ec8')
try:
    hifa_importdata(vis=['uid___A002_Xd704f8_Xf10b'], dbservice=False, session=['session_1'])
    hifa_flagdata(pipelinemode="automatic")
    ...de="automatic")
    ...de="automatic")
    ...tomatic")
    ...omatic")
    ..."automatic")
    ...utomatic")
    ...de="automatic")
    ...e="automatic")
    ..."automatic")
    ...ode="automatic")
    ...e="automatic")
    ...emode="automatic")
    ...e="automatic")
    ...de="automatic")
    ...automatic")
    ...E,BANDPASS')
    ...="automatic")
    ... intent='CHECK')
    ...="automatic")
    ...mode="automatic")
    ...ductsize=350.0, maxcubesize=40.0, maxcubelimit=60.0)
finally:
    h_save()
```

IMAGING

```
__rethrow_casa_exceptions = True
context = h_init()
context.set_state('ProjectSummary', 'proposal_code', '2018.1.00526.S')
context.set_state('ProjectSummary', 'piname', 'unknown')
context.set_state('ProjectSummary', 'proposal_title', 'unknown')
context.set_state('ProjectStructure', 'ous_part_id', 'X446126180')
context.set_state('ProjectStructure', 'ous_title', 'Undefined')
context.set_state('ProjectStructure', 'ppr_file', '/opt/dared/opt/dared.2018OCT/mnt/dat
01_X133d_X1ec9.xml')
context.set_state('ProjectStructure', 'ps_entity_id', 'uid://A001/X12eb/X270')
context.set_state('ProjectStructure', 'recipe_name', 'hifa_image')
context.set_state('ProjectStructure', 'ous_entity_id', 'uid://A001/X12eb/X26c')
context.set_state('ProjectStructure', 'ousstatus_entity_id', 'uid://A001/X133d/X1ec8')
try:
    hifa_restoredata(vis=['/opt/dared/opt/dared.2018OCT/mnt/dataproc/2018.1.00526.S_201
['session_1'], copytoraw=False)
    hif_mstransform(pipelinemode="automatic")
    hifa_flagtargets(pipelinemode="automatic")
    hifa_imageprecheck(pipelinemode="automatic")
    hif_checkproductsize(maxproductsize=350.0, maxcubesize=40.0, maxcubelimit=60.0)
    hif_makeimlist(specmode='mfs')
    hif_findcont(pipelinemode="automatic")
    hif_uvcontfit(pipelinemode="automatic")
    hif_uvcontsub(pipelinemode="automatic")
    hif_makeimages(pipelinemode="automatic")
    hif_makeimlist(specmode='cont')
    hif_makeimages(pipelinemode="automatic")
    hif_makeimlist(specmode='cube')
    hif_makeimages(pipelinemode="automatic")
    hif_makeimlist(specmode='repBW')
    hif_makeimages(pipelinemode="automatic")
finally:
    h_save()
```

Pipeline running is tracked in a 'Context', so it knows the stage and what to do.

Must always be initialised! (and saved)

Task order is important, Pipeline won't work by running out of sequence or forcing 'modified' data

Pipeline imaged data

Calibration & Image PL run combined

```
from casarecipes.almahelpers import fixsyscaltimes # SACM/JAO - Fixes
from casatasks import fixplanets
__rethrow_casa_exceptions = True
context = h_init()
context.set_state('ProjectSummary', 'proposal_code', '2019.1.00097.S')
context.set_state('ProjectSummary', 'proposal_title', 'unknown')
context.set_state('ProjectSummary', 'piname', 'unknown')
context.set_state('ProjectStructure', 'ous_entity_id', 'uid://A001/X13b9/Xbf')
context.set_state('ProjectStructure', 'ous_part_id', 'X928641954')
context.set_state('ProjectStructure', 'ous_title', 'Undefined')
context.set_state('ProjectStructure', 'ps_entity_id', 'uid://A001/X13b9/Xc3')
context.set_state('ProjectStructure', 'ousstatus_entity_id', 'uid://A001/X1528/X2a2')
context.set_state('ProjectStructure', 'ppr_file', '/opt/pipelinedriver/2021FEB/mnt/dataproc/2019.1.00097.S_2021_
__A001_X1528_X2a0/GOUS_uid___A001_X1528_X2a1/MOUS_uid___A001_X1528_X2a2/working/PPR_uid___A001_X1528_X2a3.xml')
context.set_state('ProjectStructure', 'recipe_name', 'hifa_calimage')
try:
    hifa_importdata(dbservice=True, vis=['uid___A002_Xed07bd_X15e'], session=['session_1'])
    fixsyscaltimes(vis='uid___A002_Xed07bd_X15e.ms') # SACM/JAO - Fixes
    h_save() # SACM/JAO - Finish weblog after fixes
    h_init() # SACM/JAO - Restart weblog after fixes
    hifa_importdata(dbservice=True, vis=['uid___A002_Xed07bd_X15e'], session=['session_1'])
    hifa_flagdata(pipelinemode="automatic")
    hifa_fluxcalflag(pipelinemode="automatic")
    hif_rawflagchans(pipelinemode="automatic")
    hif_refant(pipelinemode="automatic")
    h_tsyscal(pipelinemode="automatic")
    hifa_tsysflag(pipelinemode="automatic")
    hifa_antpos(pipelinemode="automatic")
    hifa_wvrgcalflag(pipelinemode="automatic")
    hif_lowgainflag(pipelinemode="automatic")
    hif_setmodels(pipelinemode="automatic")
    hifa_bandpassflag(pipelinemode="automatic")
    hifa_bandpass(pipelinemode="automatic")
    hifa_spwphaseup(pipelinemode="automatic")
    hifa_gfluxscaleflag(pipelinemode="automatic")
    hifa_gfluxscale(pipelinemode="automatic")
    hifa_timegaincal(pipelinemode="automatic")
    hifa_targetflag(pipelinemode="automatic")
    hif_applycal(pipelinemode="automatic")
    hif_makeimlist(intent='PHASE,BANDPASS,AMPLITUDE')
    hif_makeimages(pipelinemode="automatic")
    hif_makeimlist(intent='CHECK', per_eb=True)
    hif_makeimages(pipelinemode="automatic")
    hifa_imageprecheck(pipelinemode="automatic")
    hif_checkproductsize(maxcubesize=40.0, maxcubelimit=60.0, maxproductsize=350.0)
    hifa_exportdata(pipelinemode="automatic")
    hif_mstransform(pipelinemode="automatic")
    hifa_flagtargets(pipelinemode="automatic")
    hif_makeimlist(specmode='mfs')
    hif_findcont(pipelinemode="automatic")
    hif_uvcontfit(pipelinemode="automatic")
    hif_uvcontsub(pipelinemode="automatic")
    hif_makeimages(pipelinemode="automatic")
    hif_makeimlist(specmode='cont')
    hif_makeimages(pipelinemode="automatic")
    hif_makeimlist(specmode='cube')
```

Imaging related

CAUTION - this default script contains a restore cal - i.e. ***don't*** re-run if you already did the "scriptForPI.py"

Calibration & Image PL run SEPARATE

CAL

```
__rethrow_casa_exceptions = True
context = h_init()
context.set_state('ProjectSummary', 'proposal_code', '2018.1.00526.S')
context.set_state('ProjectSummary', 'piname', 'unknown')
context.set_state('ProjectSummary', 'proposal_title', 'unknown')
context.set_state('ProjectStructure', 'ous_part_id', 'X446126180')
context.set_state('ProjectStructure', 'ous_title', 'Undefined')
context.set_state('ProjectStructure', 'ppr_file', '/opt/dared/opt/dared.2018AUG/mnt/dataproc/2
2019_01_03T12_52_58.189/SOUS_uid___A001_X133d_X1ec3/GOUS_uid___A001_X133d_X1ec7/MOUS_uid___A00
working/PPR_uid___A001_X133d_X1ec9.xml')
context.set_state('ProjectStructure', 'ps_entity_id', 'uid://A001/X12eb/X270')
context.set_state('ProjectStructure', 'recipe_name', 'hifa_cal')
context.set_state('ProjectStructure', 'ous_entity_id', 'uid://A001/X12eb/X26c')
context.set_state('ProjectStructure', 'ousstatus_entity_id', 'uid://A001/X133d/X1ec8')
try:
    hifa_importdata(vis=['uid___A002_Xd704f8_Xf10b'], dbservice=False, session=['session_1'])
    hifa_flagdata(pipelinemode="automatic")
    hifa_fluxcalflag(pipelinemode="automatic")
    hif_rawflagchans(pipelinemode="automatic")
    hif_refant(pipelinemode="automatic")
    h_tsyscal(pipelinemode="automatic")
    hifa_tsysflag(pipelinemode="automatic")
    hifa_antpos(pipelinemode="automatic")
    hifa_wvrgcalflag(pipelinemode="automatic")
    hif_lowgainflag(pipelinemode="automatic")
    hif_setmodels(pipelinemode="automatic")
    hifa_bandpassflag(pipelinemode="automatic")
    hifa_spwphaseup(pipelinemode="automatic")
    hifa_gfluxscaleflag(pipelinemode="automatic")
    hifa_gfluxscale(pipelinemode="automatic")
    hifa_timegaincal(pipelinemode="automatic")
    hif_applycal(pipelinemode="automatic")
    hif_makeimlist(intent='PHASE,BANDPASS')
    hif_makeimages(pipelinemode="automatic")
    hif_makeimlist(per_eb=True, intent='CHECK')
    hif_makeimages(pipelinemode="automatic")
    hifa_imageprecheck(pipelinemode="automatic")
    hif_checkproductsize(maxproductsize=350.0, maxcubesize=40.0, maxcubelimit=60.0)
finally:
    h_save()
```

IMAGING

```
__rethrow_casa_exceptions = True
context = h_init()
context.set_state('ProjectSummary', 'proposal_code', '2018.1.00526.S')
context.set_state('ProjectSummary', 'piname', 'unknown')
context.set_state('ProjectSummary', 'proposal_title', 'unknown')
context.set_state('ProjectStructure', 'ous_part_id', 'X446126180')
context.set_state('ProjectStructure', 'ous_title', 'Undefined')
context.set_state('ProjectStructure', 'ppr_file', '/opt/dared/opt/dared.2018OCT/mnt/dat
01_X133d_X1ec9.xml')
context.set_state('ProjectStructure', 'ps_entity_id', 'uid://A001/X12eb/X270')
context.set_state('ProjectStructure', 'recipe_name', 'hifa_image')
context.set_state('ProjectStructure', 'ous_entity_id', 'uid://A001/X12eb/X26c')
context.set_state('ProjectStructure', 'ousstatus_entity_id', 'uid://A001/X133d/X1ec8')
try:
    hifa_restoredata(vis=['/opt/dared/opt/dared.2018OCT/mnt/dataproc/2018.1.00526.S_201
['session_1'], copytoraw=False)
    hif_mstransform(pipelinemode="automatic")
    hifa_flagtargets(pipelinemode="automatic")
    hifa_imageprecheck(pipelinemode="automatic")
    hif_checkproductsize(maxproductsize=350.0, maxcubesize=40.0, maxcubelimit=60.0)
    hif_makeimlist(specmode='mfs')
    hif_findcont(pipelinemode="automatic")
    hif_uvcontfit(pipelinemode="automatic")
    hif_uvcontsub(pipelinemode="automatic")
    hif_makeimages(pipelinemode="automatic")
    hif_makeimlist(specmode='cont')
    hif_makeimages(pipelinemode="automatic")
    hif_makeimlist(specmode='cube')
    hif_makeimages(pipelinemode="automatic")
    hif_makeimlist(specmode='repBW')
    hif_makeimages(pipelinemode="automatic")
finally:
    h_save()
```

Pipeline imaged data

- Imaging interactively in the 'working' directory after restoring the data

```
CASA <1>: __rethrow_casa_exceptions = True
CASA <2>: h_init()
Out[2]: <Context(name='pipeline-20201118T095710')>
CASA <3>:import glob
CASA <4>:MyVis=glob.glob('*ms')
CASA <5>: MyVis
Out[4]: ['uid___A002_Xd2b681_Xa1c2.ms', 'uid___A002_Xd2b681_Xb99d.ms']
```

Context initialised

define a global variable that includes all the calibrated ms

NOTE: There are some intricacies in loading data - typically from Pipeline restore this is an ".ms" and the CORECTED_DATA column is populated and later used to 'split-out' the target

Pipeline imaged data

- Imaging interactively in the ‘working’ directory after restoring the data

Default commands as from the imaging script:
- Run as script (defaults) will recreate images that Pipeline made
- Run step-by-step (defaults) also recreates what Pipeline made
- Run step-by-step with some changes - track also in WebLog → Image Parameters

```
hifa_importdata(vis=MyVis,dbservice=False)
hif_mstransform(pipelinemode="automatic")
hifa_flagtargets(pipelinemode="automatic")
hifa_imageprecheck(pipelinemode="automatic")
hif_makeimlist(specmode='mfs')
hif_findcont(pipelinemode="automatic")
hif_uvcontfit(pipelinemode="automatic")
hif_uvcontsub(pipelinemode="automatic")
hif_makeimages(pipelinemode="automatic")
hif_makeimlist(specmode='cont')
hif_makeimages(pipelinemode="automatic")
hif_makeimlist(specmode='cube')
hif_makeimages(pipelinemode="automatic")
hifa_exportdata(imaging_products_only=True)
```

Pipeline imaged data

- Imaging interactively in the 'working' directory after restoring the data

```
hifa_importdata(vis=MyVis,dbservice=False)
hif_mstransform(pipelinemode="automatic")
hifa_flagtargets(pipelinemode="automatic")
hifa_imageprecheck(pipelinemode="automatic")
hif_makeimlist(specmode='mfs')
hif_findcont(pipelinemode="automatic")
hif_uvcontfit(pipelinemode="automatic")
hif_uvcontsub(pipelinemode="automatic")
hif_makeimages(pipelinemode="automatic")
hif_makeimlist(specmode='cont')
hif_makeimages(pipelinemode="automatic")
hif_makeimlist(specmode='cube')
hif_makeimages(pipelinemode="automatic")
hifa_exportdata(imaging_products_only=True)
```

import and prepare the data

Imports into Pipeline
Makes MS with targets only
Precheck of image parameters
Image parameter command for MFS after contsub

find and subtract the continuum

Makes cubes and gets spectra from masked region - *(or use cont.dat)*
LINE only is saved to CORRECTED_DATA column in MS

make images and cubes

Loops of imlist and make images

exports to fits files

CAUTION - changes from Cycle 9 - files made for imaging will be:
(1) targets - cont+line (2) targets_line - line only (after cont-sub)

Pipeline imaged data

- Imaging interactively in the 'working' directory after restoring the data

As of late 2022 - Cycle 8 "scriptForPI.py" has an option to 'docontsub' which prepares Pipeline calibrated data ready for imaging loops

```
hifa_importdata(vis=MyVis,dbservice=False)
hif_mstransform(pipelinemode="automatic")
hifa_flagtargets(pipelinemode="automatic")
hifa_imageprecheck(pipelinemode="automatic")
hif_makeimlist(specmode='mfs')
hif_findcont(pipelinemode="automatic")
hif_uvcontfit(pipelinemode="automatic")
hif_uvcontsub(pipelinemode="automatic")
hif_makeimages(pipelinemode="automatic")
hif_makeimlist(specmode='cont')
hif_makeimages(pipelinemode="automatic")
hif_makeimlist(specmode='cube')
hif_makeimages(pipelinemode="automatic")
hifa_exportdata(imaging_products_only=True)
```

imp nd
re the data

Imports into Pipeline
Makes MS with targets only
Precheck of image parameters
Image parameter command for MFS after contsub

find and subtract the continuum

Makes cubes and gets spectra from masked region - *(or use cont.dat)*
LINE only is saved to CORRECTED_DATA column in MS

make images and cubes

Loops of imlist and make images

exports to fits files

CAUTION - changes from Cycle 9 - files made for imaging will be:
(1) targets - cont+line (2) targets_line - line only (after cont-sub)

Pipeline imaged data

- Recall the image precheck - setup image parameters *(records inside context for automatic mode imaging)* - what setting to use that the PI wanted
- Archive users - this can be a guide, but image parameter can be changed later

Goals From OT:

Representative Target: LkCa15
Representative Frequency: 663.9863 GHz (SPW 37)
Bandwidth for Sensitivity: 1.5e+04 MHz
Min / Max Acceptable Resolution: 0.146 arcsec / 0.179 arcsec
Maximum expected beam axial ratio (from OT): Not available
Goal PI sensitivity: Not available
Single Continuum: False

Estimated Synthesized Beam and Sensitivities for the Representative Target/Frequency

Estimates are given for four possible values of the tclean robust weighting parameter: robust = 0.0, +0.5 (default), +1.0, and +2.0. **If the "Min / Max Acceptable Resolution" is available (>=Cycle 5 12-m Array data)**, the robust value closest to the default (+0.5) that predicts a beam area (defined as simply major x minor) that is in the range of the PI requested beam areas acc
Sensitivity) is chosen. If none of these robust values predict a beam area that is in range, robust=+2.0 is chosen if the predicted beam area is too small, and robust=0.0 is chosen if the predicted beam area is too large. The chosen robust value is highlighted in green and used for all science target imaging. In addition to an estimate for the repBW, an estimate for the aggregate c
NO line contamination but accounting for spw frequency overlap. If the Bandwidth for Sensitivity (repBW) is > the bandwidth of the spw containing the representative frequency (repSPW), then the beam is predicted using all spws, otherwise the beam is predicted for the repSPW alone. A message appears on the "By Task" view if a non-default value of robust (i.e., not +0.5) is c
within the PI requested range using one of the four robust values, Warning messages appear on this page.

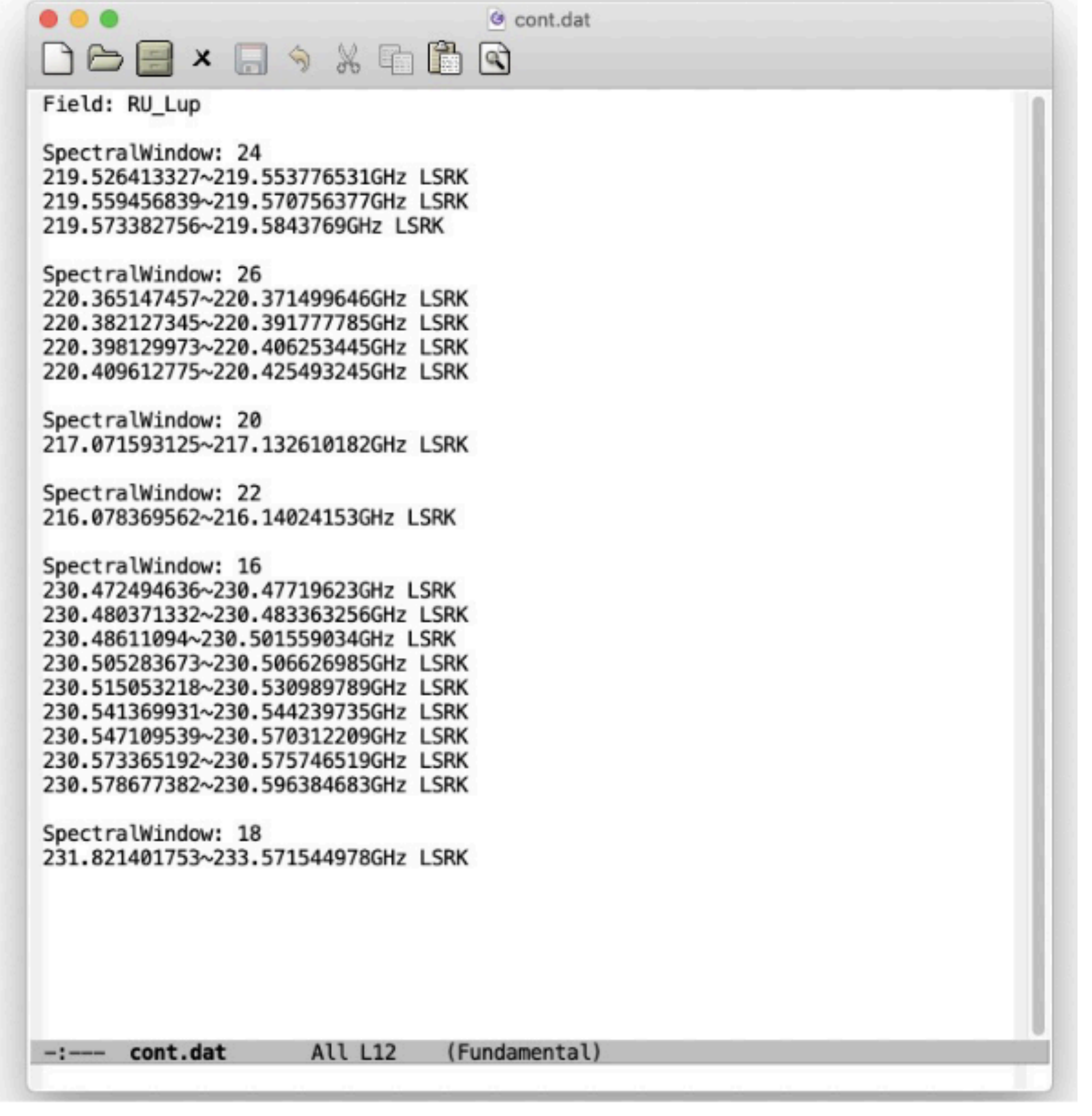
These estimates should always be considered as the BEST CASE SCENARIO. These estimates account for Tsys, the observed uv-coverage, and prior flagging. The estimates DO NOT account for (1) subsequent science target flagging; (2) loss of continuum bandwidth due to the hif_findcont process (i.e. removal of lines and other spectral features from the data used to im
quality like (a) poor match of uv-coverage to image complexity; (b) dynamic range effects; (c) calibration deficiencies (poor phase transfer, residual baseline based effects, residual antenna position errors, etc.). *It is also important to note that both the repBW and aggBW beam calculations are intrinsically multi-frequency synthesis continuum calculations, using the relevant spws*
single channel in a cube will typically be larger and can be significantly larger depending on the details of uv-coverage and channel width.

robust	uvtaper	Synthesized Beam	Cell	Beam Ratio	Bandwidth	BW Mode	Effective Sen
0.0	[]	0.176 x 0.154 arcsec @ -15.0 deg	0.031 x 0.031 arcsec	1.14	8169 MHz	repBW	0.000373 Jy/b
0.0	[]	0.176 x 0.154 arcsec @ -15.0 deg	0.031 x 0.031 arcsec	1.14	8169 MHz	aggBW	0.000373 Jy/b
0.5	[]	0.204 x 0.175 arcsec @ -20.2 deg	0.035 x 0.035 arcsec	1.17	8169 MHz	repBW	0.000298 Jy/b
0.5	[]	0.204 x 0.175 arcsec @ -20.2 deg	0.035 x 0.035 arcsec	1.17	8169 MHz	aggBW	0.000298 Jy/b
1.0	[]	0.245 x 0.206 arcsec @ -22.6 deg	0.041 x 0.041 arcsec	1.19	8169 MHz	repBW	0.00027 Jy/be
1.0	[]	0.245 x 0.206 arcsec @ -22.6 deg	0.041 x 0.041 arcsec	1.19	8169 MHz	aggBW	0.00027 Jy/be
2.0	[]	0.266 x 0.222 arcsec @ -23.5 deg	0.044 x 0.044 arcsec	1.20	8169 MHz	repBW	0.000268 Jy/b
2.0	[]	0.266 x 0.222 arcsec @ -23.5 deg	0.044 x 0.044 arcsec	1.20	8169 MHz	aggBW	0.000268 Jy/b

Pipeline imaged data

- Using or modifying the continuum subtraction

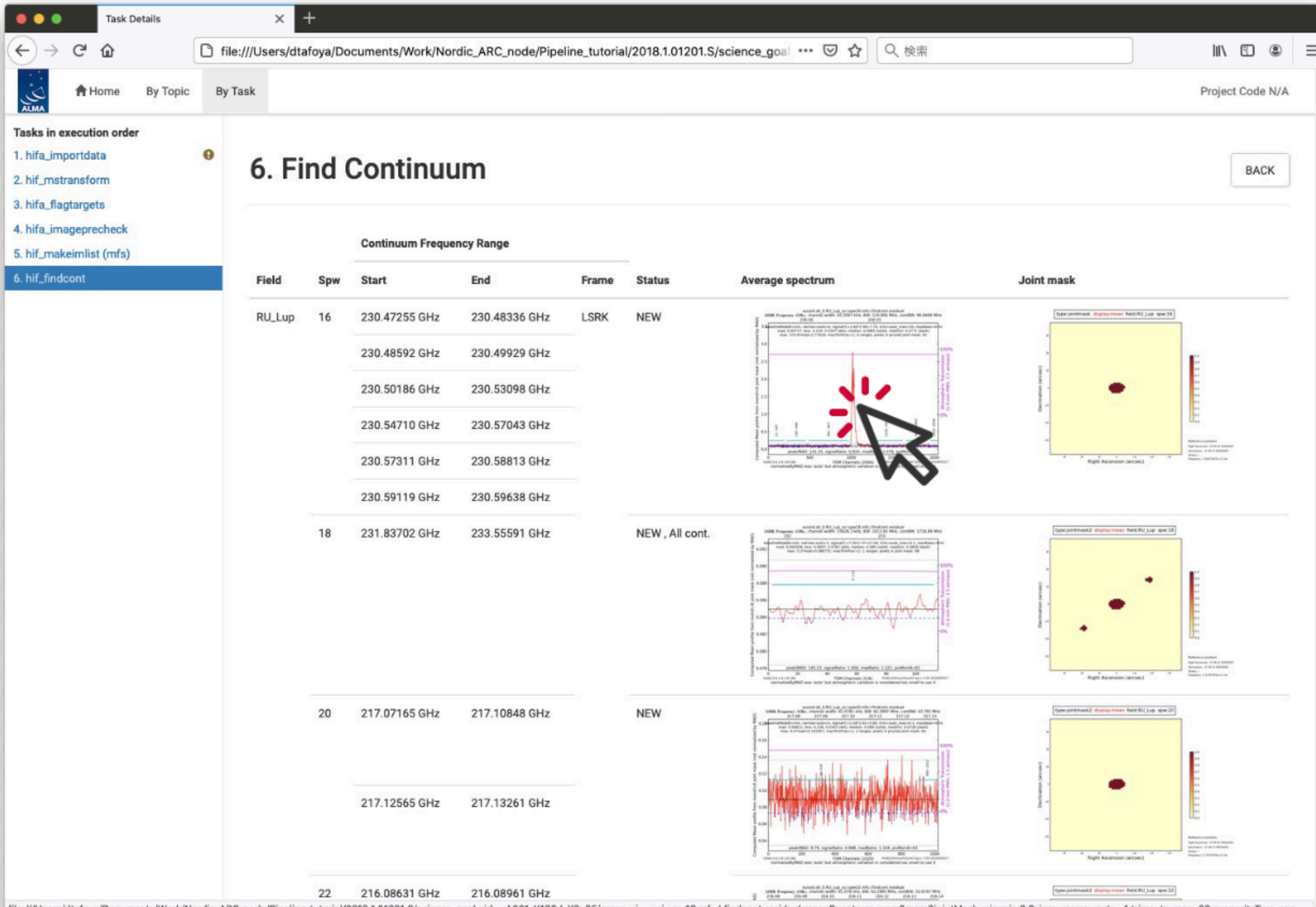
- Cont.dat was Pipeline run listing LSRK frequencies of the continuum



```
Field: RU_Lup

SpectralWindow: 24
219.526413327~219.553776531GHz LSRK
219.559456839~219.570756377GHz LSRK
219.573382756~219.5843769GHz LSRK

SpectralWindow: 26
220.365147457~220.371499646GHz LSRK
220.382127345~220.391777785GHz LSRK
220.398129973~220.406253445GHz LSRK
220.409612775~220.425493245GHz LSRK

SpectralWindow: 20
217.071593125~217.132610182GHz LSRK

SpectralWindow: 22
216.078369562~216.14024153GHz LSRK

SpectralWindow: 16
230.472494636~230.47719623GHz LSRK
230.480371332~230.483363256GHz LSRK
230.48611094~230.501559034GHz LSRK
230.505283673~230.506626985GHz LSRK
230.515053218~230.530989789GHz LSRK
230.541369931~230.544239735GHz LSRK
230.547109539~230.570312209GHz LSRK
230.573365192~230.575746519GHz LSRK
230.578677382~230.596384683GHz LSRK

SpectralWindow: 18
231.821401753~233.571544978GHz LSRK

-:---  cont.dat       All L12     (Fundamental)
```

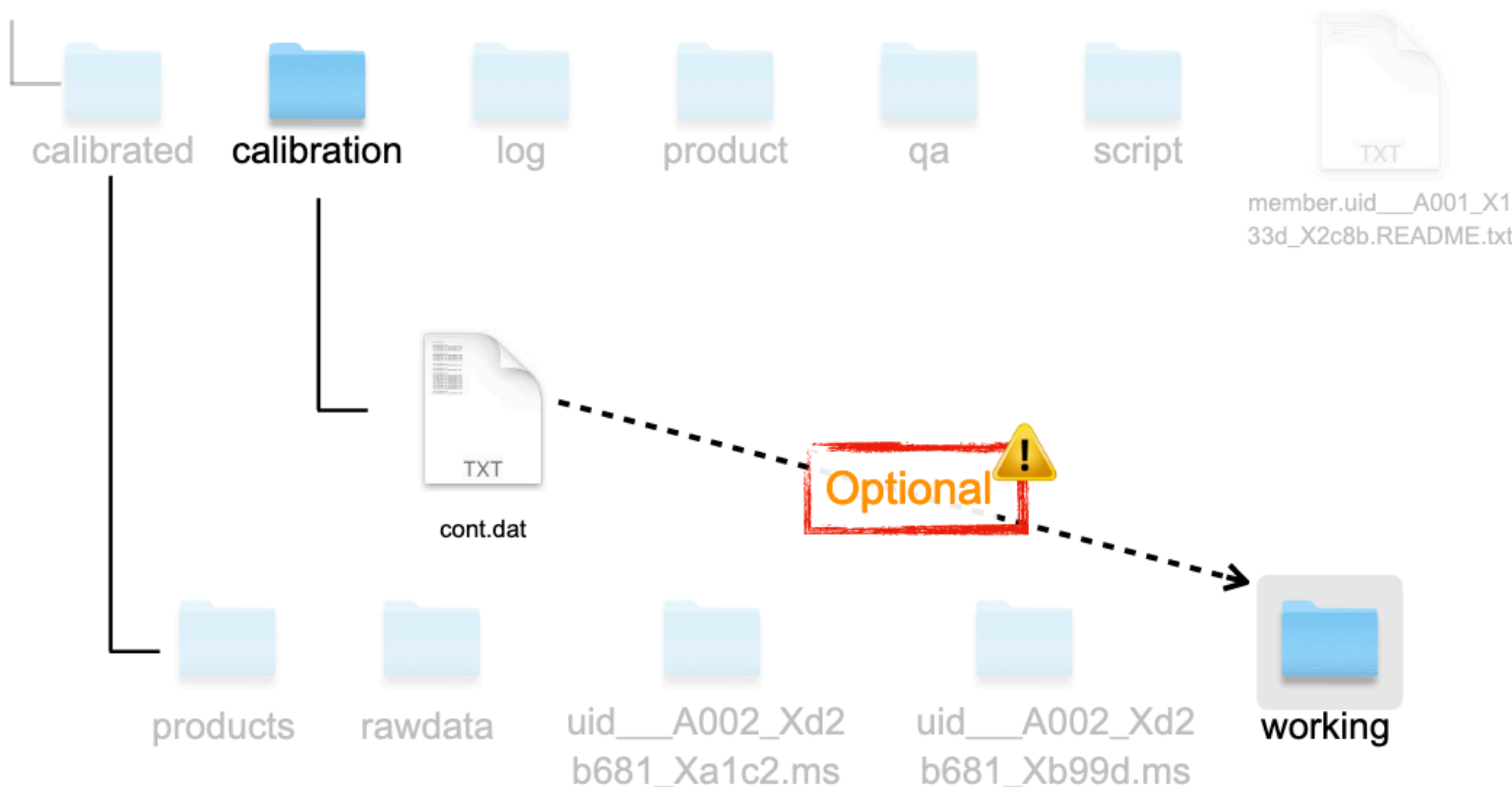
Pipeline imaged data

In the weblog (if `hif_findcont` was run without *cont.dat*):



Pipeline imaged data

- Using or modifying the continuum subtraction
- Cont.dat was Pipeline run listing LSRK frequencies of the continuum
  - If you re-run hifa_findcont without a cont.dat file it will simply replicate it *(taking time to make image cube and check the spectra)*
  - Instead copy the cont.dat to the working area
  - You can inspect the image cubes/products to see if you need to refine the ranges - and edit cont.dat *(also check the QA report for more information)*



Pipeline imaged data

- Imaging interactively in the ‘working’ directory after restoring the data

```
hifa_importdata(vis=MyVis,dbservice=False)
hif_mstransform(pipelinemode="automatic")
hifa_flagtargets(pipelinemode="automatic")
hifa_imageprecheck(pipelinemode="automatic")
hif_makeimlist(specmode='mfs')
hif_findcont(pipelinemode="automatic")
hif_uvcontfit(pipelinemode="automatic")
hif_uvcontsub(pipelinemode="automatic")
hif_makeimages(pipelinemode="automatic")
hif_makeimlist(specmode='cont')
hif_makeimages(pipelinemode="automatic")
hif_makeimlist(specmode='cube')
hif_makeimages(pipelinemode="automatic")
hifa_exportdata(imaging_products_only=True)
```

Best not to change any parameters for this first call

Pipeline imaged data

- Imaging interactively in the ‘working’ directory after restoring the data

```
hifa_importdata(vis=MyVis,dbservice=False)
hif_mstransform(pipelinemode="automatic")
hifa_flagtargets(pipelinemode="automatic")
hifa_imageprecheck(pipelinemode="automatic")
hif_makeimlist(specmode='mfs')
hif_findcont(pipelinemode="automatic")
hif_uvcontfit(pipelinemode="automatic")
hif_uvcontsub(pipelinemode="automatic")
hif_makeimages(pipelinemode="automatic")
hif_makeimlist(specmode='cont')
hif_makeimages(pipelinemode="automatic")
hif_makeimlist(specmode='cube')
hif_makeimages(pipelinemode="automatic")
hifa_exportdata(imaging_products_only=True)
```

Let this auto be automatic

Add a h_save() to save all work to this point

Pipeline imaged data

- Imaging interactively in the ‘working’ directory after restoring the data

```
hifa_importdata(vis=MyVis,dbservice=False)
hif_mstransform(pipelinemode="automatic")
hifa_flagtargets(pipelinemode="automatic")
hifa_imageprecheck(pipelinemode="automatic")
hif_makeimlist(specmode='mfs')
hif_findcont(pipelinemode="automatic")
hif_uvcontfit(pipelinemode="automatic")
hif_uvcontsub(pipelinemode="automatic")
hif_makeimages(pipelinemode="automatic")
hif_makeimlist(specmode='cont')
hif_makeimages(pipelinemode="automatic")
hif_makeimlist(specmode='cube')
hif_makeimages(pipelinemode="automatic")
hifa_exportdata(imaging_products_only=True)
```

Let this auto be automatic

Add a h_save() to save all work to this point

If you just run these as defaults, the will simply replicate what images got made and are in the products

Pipeline imaged data

- Imaging interactively in the ‘working’ directory after restoring the data
  - Editing the parameters - loops of **hifa_makeimlist** and **hif_makeimages**

```
CASA <20>: hif_makeimlist(specmode='mfs',robust=-0.5,spw='24',hm_imsize=60))
```

- only image spw 24 with robust parameter set to -0.5 and image size = 60pixels.

```
CASA <21>: hif_makeimages(pipelinemode="automatic")
```

Pipeline imaged data

- Imaging interactively in the ‘working’ directory after restoring the data
  - Editing the parameters - loops of hifa_makeimlist and hif_makeimages

dtafoya — -zsh — 143×53

```
CASA <22>: hif_makeimlist(specmode='cube',spw='24',nchan=200,start='219.553GHz')
```

- makes a cube of 200 channels of the spw 24, starting ar freq=219.553GHz.

dtafoya — -zsh — 143×53

```
CASA <23>: hif_makeimages(pipelinemode="automatic")
```

EUROPEAN ARC
ALMA Regional Centre

Pipeline imaged data

- Imaging interactively in the ‘working’ directory after restoring the data
  - Editing the parameters - loops of hifa_makeimlist and hif_makeimages

dtafoya — -zsh — 143×53

```
CASA <24>: help(hif_makeimlist)

 Help on instance of hif_makeimlist_cli_ in module hif_makeimlist_cli:

 hif_makeimlist = class hif_makeimlist_cli_
  |  Methods defined here:
  |
  |  __call__(self, vis=None, imagename=None, intent=None, field=None,
 spw=None, contfile=None, linesfile=None, uvrange=None, specmode=None,
 outframe=None, hm_imsize=None, hm_cell=None, calmaxpix=None,
 phasecenter=None, nchan=None, start=None, width=None, nbins=None,
 robust=None, uvtaper=None, clearlist=None, per_eb=None, calcsb=None,
 parallel=None, pipelinemode=None, dryrun=None, acceptresults=None)
  |      Compute list of clean images to be produced
  |
  |              Detailed Description:
  |
  |      Create a a list of images to be cleaned.
  |
  |              Arguments :
  |                      vis: List of input MeasurementSets
  |                         Default Value:
```

EUROPEAN ARC
ALMA Regional Centre

Pipeline imaged data

- Imaging interactively in the 'working' directory after restoring the data
  - Editing the parameters - loops of **hifa_makeimlist** and **hif_makeimages**

```
class _hif_makeimages(builtins.object)
 |  hif_makeimages ---- Compute clean map
 |
 |
 |  Compute clean results from a list of specified targets.
 |
 |  Output:
 |
 |  results -- If pipeline mode is 'getinputs' then None is returned. Otherwise
 |  the results object for the pipeline task is returned.
 |
 |  --------- parameter descriptions ---------------------------------------------
 |
 |  vis                     The list of input MeasurementSets. Defaults to the list of
 |                          MeasurementSets specified in the h_init or hif_importdata task.
 |                          '': use all MeasurementSets in the context
 |
 |                          Examples: 'ngc5921.ms', ['ngc5921a.ms', ngc5921b.ms', 'ngc5921c.ms']
 |  target_list             Dictionary specifying targets to be imaged; blank will read list from context
 |  hm_masking              Clean masking mode. Options are 'centralregion', 'auto',
 |                          'manual' and 'none'
 |  hm_sidelobethreshold    sidelobethreshold * the max sidelobe level
 |  hm_noisethreshold       noisethreshold * rms in residual image
 |  hm_lownoisethreshold    lownoisethreshold * rms in residual image
 |  hm_negativethreshold    negativethreshold * rms in residual image
 |  hm_minbeamfrac          Minimum beam fraction for pruning
 |  hm_growiterations       Number of binary dilation iterations for growing the mask
 |  hm_dogrowprune          Do pruning on the grow mask
 |
 |                          Defaults to '' to enable the automatic heuristics calculation.
 |                          Can be set to True or False manually.
 |  hm_minpercentchange     Mask size change threshold
 |  hm_fastnoise           Faster noise calucation for automask or nsigma stopping
 |
 |                          Defaults to '' to enable the automatic heuristics calculation.
 |                          Can be set to True or False manually.
 |  hm_nsigma               Multiplicative factor for rms-based threshold stopping
 |  hm_perchanweightdensity Calculate the weight density for each channel independently
 |
 |                          Defaults to '' to enable the automatic heuristics calculation.
 |                          Can be set to True or False manually.
 |  hm_npixels              Number of pixels to determine uv-cell size for super-uniform weighting
 |  hm_cyclefactor          Scaling on PSF sidelobe level to compute the minor-cycle stopping threshold
 |  hm_minpsffraction       PSF fraction that marks the max depth of cleaning in the minor cycle
 |  hm_maxpsffraction       PSF fraction that marks the minimum depth of cleaning in the minor cycle
 |  hm_weighting            Weighting scheme (natural,uniform,briggs,briggsabs[experimental],briggsbwtaper[experimental])
 |  hm_cleaning             Pipeline cleaning mode
 |  tlimit                  Times the sensitivity limit for cleaning
 |  masklimit               Times good mask pixels for cleaning
 |  cleancontranges         Clean continuum frequency ranges in cubes
 |  calcsb                  Force (re-)calculation of sensitivities and beams
 |  hm_mosweight            Mosaic weighting
 |
 |                          Defaults to '' to enable the automatic heuristics calculation.
 |                          Can be set to True or False manually.
 |  overwrite_on_export     Replace existing image products when h/hifa/hifv_exportdata is
```

for masking

for cleaning

Automasking parameters for Pipeline

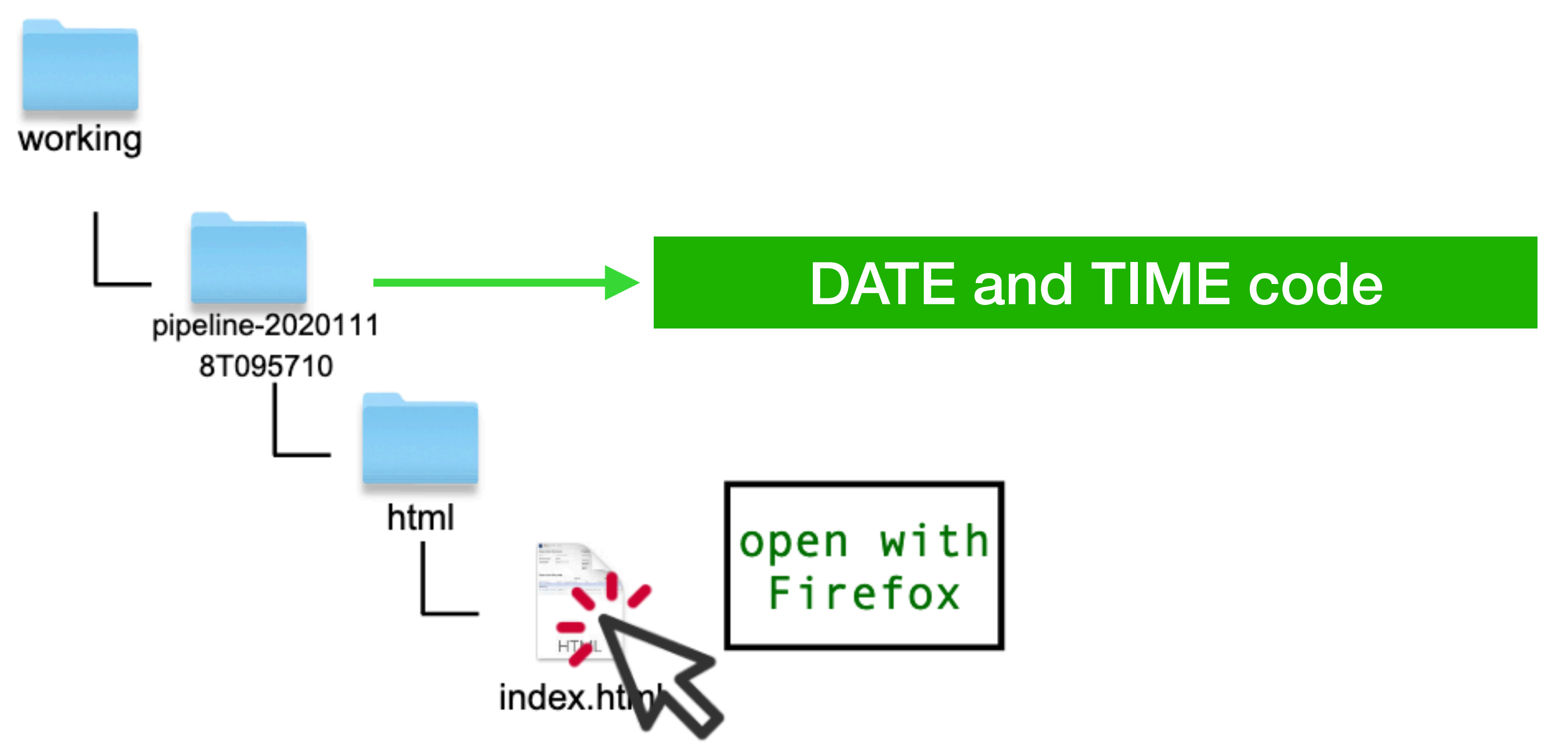
Array	*sidelobethreshold*	*noisethreshold*	*minbeamfrac*	*lownoisethreshold*	*negativethreshold*
12m (short) b75<300m	**2.0**	**4.25**	**0.3**	**1.5**	**0.0 (continuum) 15.0 (line)**
12m (long) b75>300m	**3.0**	**5.0**	**0.3**	**1.5**	**0.0 (continuum) 7.0 (line)**
7m (continnum/line)	**1.25**	**5.0**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

MUST READ - NRAO guide: https://casaguides.nrao.edu/index.php/Automasking_Guide

Pipeline imaged data

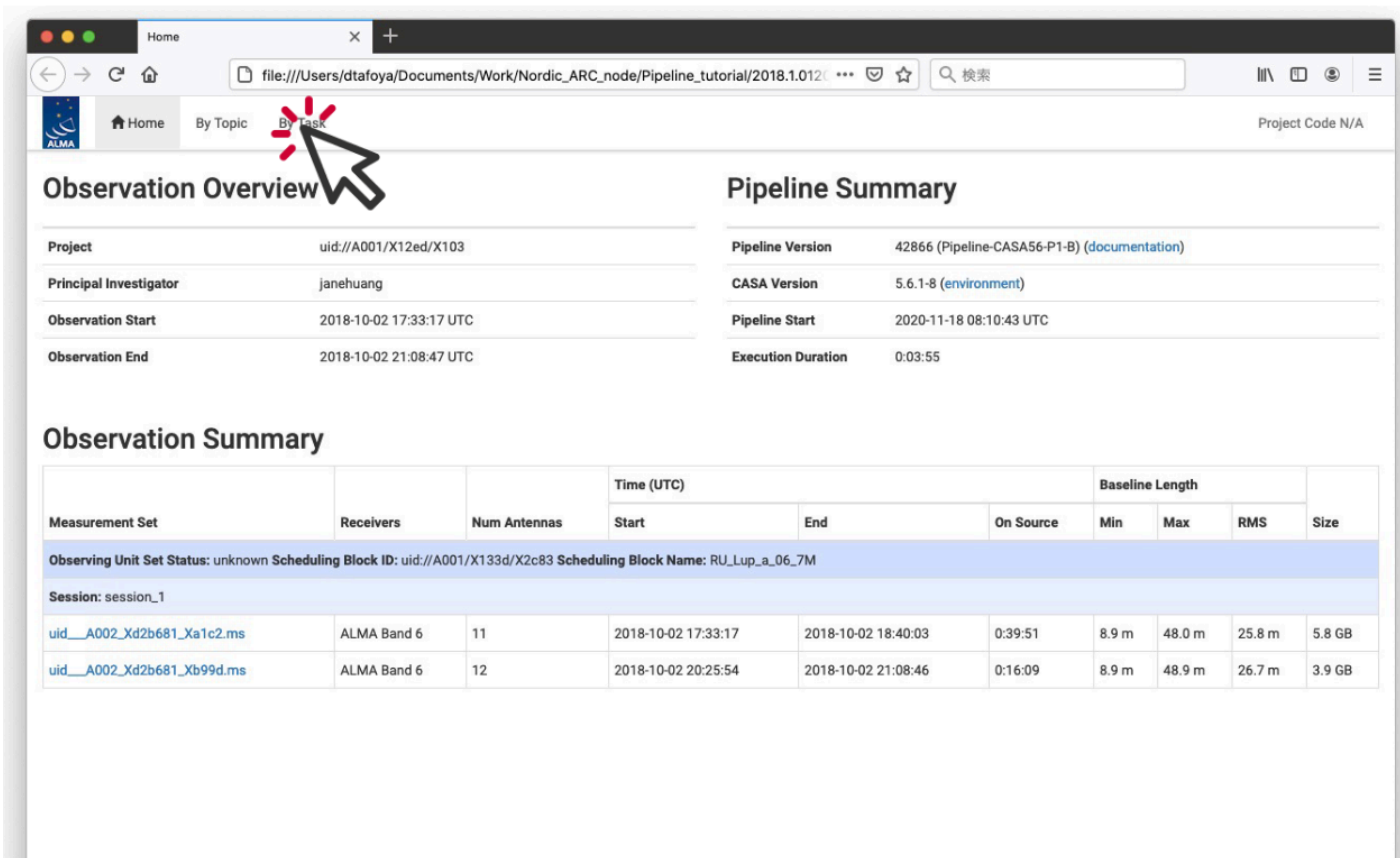
- Power of the pipeline imaging is easy to access image ‘quick look’ using the WebLog
- Initialised with your Pipeline run *(care as one was also made for the restore)*

- A html weblog will be created in the working folder:



Pipeline imaged data

- Power of the pipeline imaging is easy to access image ‘quick look’ using the WebLog
- Initialised with your Pipeline run *(care as one was also made for the restore)*



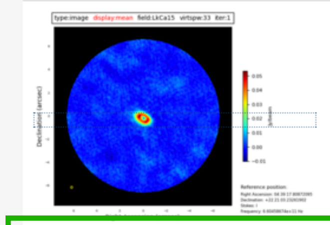
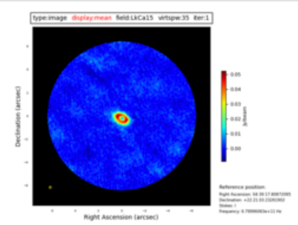
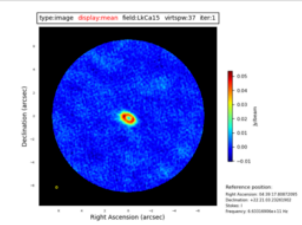
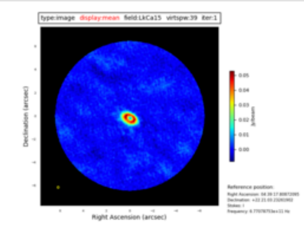
Pipeline imaged data

- Power of the pipeline imaging is easy to access image 'quick look' using the WebLog

34. Tclean/MakeImages

Make target per-spw continuum images

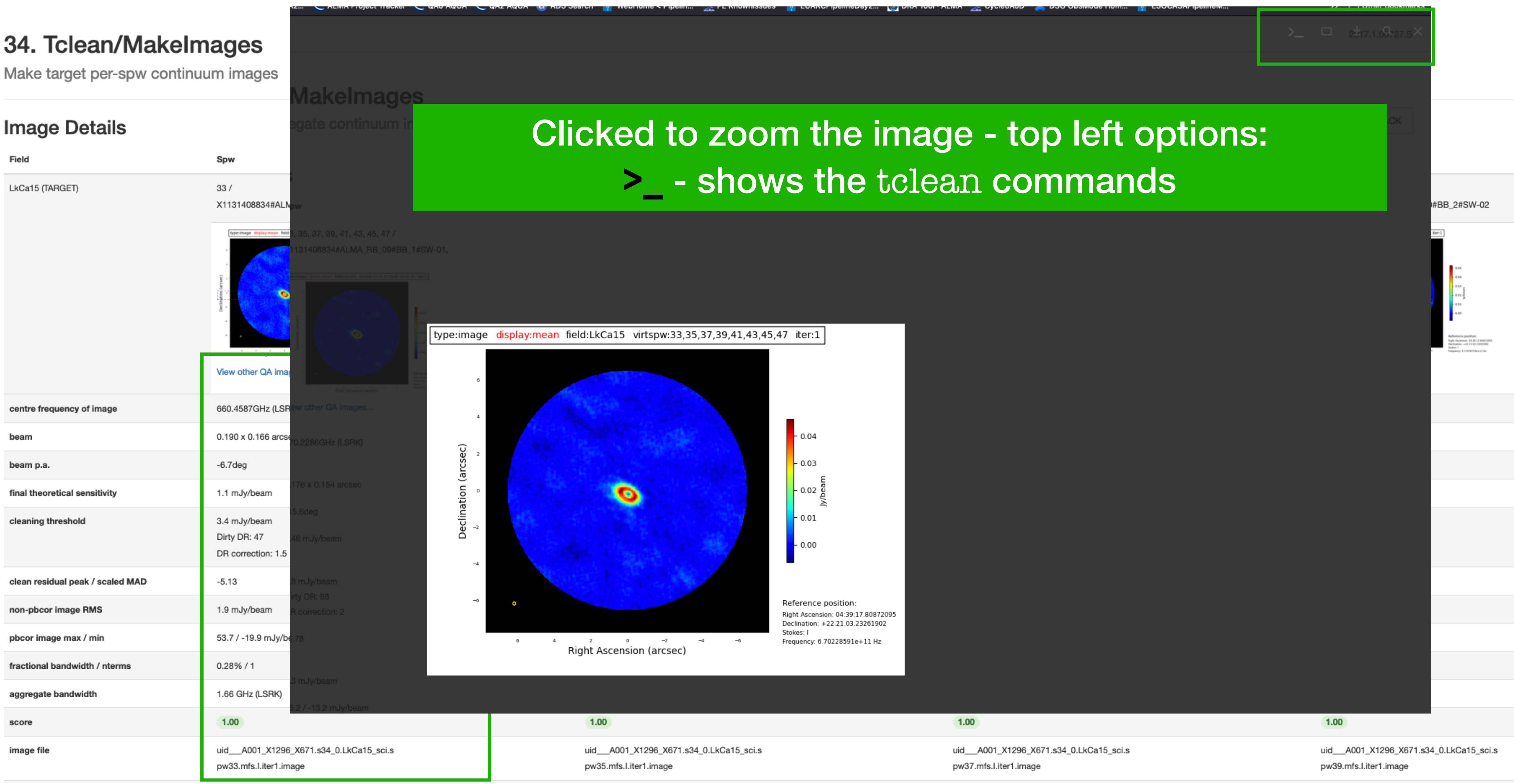
Image Details

Field	Spw			
LkCa15 (TARGET)	33 / X1131408834#ALMA_RB_09#BB_1#SW-01	35 / X1131408834#ALMA_RB_09#BB_1#SW-02	37 / X1131408834#ALMA_RB_09#BB_2#SW-01	39 / X1131408834#ALMA_RB_09#BB_2#SW-02
	 View other QA images...	 View other QA images...	 View other QA images...	 View other QA images...
centre frequency of image	660.4587GHz (LSRK)	679.9961GHz (LSRK)	663.3169GHz (LSRK)	677.0788GHz (LSRK)
beam	0.190 x 0.166 arcsec	0.185 x 0.161 arcsec	0.190 x 0.165 arcsec	0.185 x 0.162 arcsec
beam p.a.	-6.7deg	-8.3deg	-6.9deg	-7.4deg
final theoretical sensitivity	1.1 mJy/beam	1.4 mJy/beam	1.8 mJy/beam	1.2 mJy/beam
cleaning threshold	3.4 mJy/beam Dirty DR: 47 DR correction: 1.5	4.2 mJy/beam Dirty DR: 36 DR correction: 1.5	5.5 mJy/beam Dirty DR: 29 DR correction: 1.5	3.6 mJy/beam Dirty DR: 44 DR correction: 1.5
clean residual peak / scaled MAD	-5.13	4.80	4.22	4.15
non-pbcor image RMS	1.9 mJy/beam	2.3 mJy/beam	2.5 mJy/beam	2 mJy/beam
pbcor image max / min	53.7 / -19.9 mJy/beam	52.4 / -22 mJy/beam	53.9 / -28 mJy/beam	53.1 / -20.3 mJy/beam
fractional bandwidth / nterms	0.28% / 1	0.27% / 1	0.26% / 1	0.27% / 1
aggregate bandwidth	1.66 GHz (LSRK)	1.46 GHz (LSRK)	0.465 GHz (LSRK)	1.85 GHz (LSRK)
score	1.00	1.00	1.00	1.00
image file	uid___A001_X1296_X671.s34_0.LkCa15_sci.spw33.mfs.I.iter1.image	uid___A001_X1296_X671.s34_0.LkCa15_sci.spw35.mfs.I.iter1.image	uid___A001_X1296_X671.s34_0.LkCa15_sci.spw37.mfs.I.iter1.image	uid___A001_X1296_X671.s34_0.LkCa15_sci.spw39.mfs.I.iter1.image

NOTE: slight changes in viewing pages per Cycle (new PL features added)

Pipeline imaged data

- Power of the pipeline imaging is easy to access image 'quick look' using the WebLog



Clicked to zoom the image - top left options:
>_ - shows the `tclean` commands

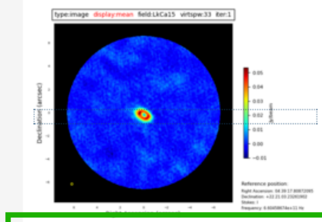
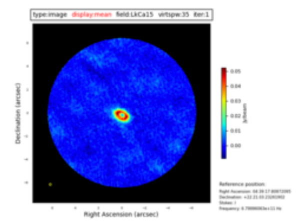
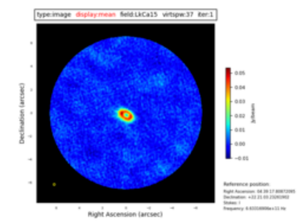
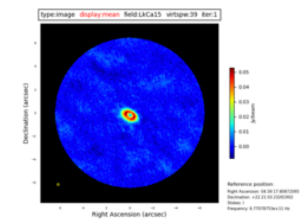
NOTE: slight changes in viewing pages per Cycle (new PL features added)

EUROPEAN ARC
ALMA Regional Centre

34. Tclean/MakeImages

Make target per-spw continuum images

Image Details

Field	Spw			
LkCa15 (TARGET)	33 / X1131408834#ALMA_RB_09#BB_1#SW-01	35 / X1131408834#ALMA_RB_09#BB_1#SW-02	37 / X1131408834#ALMA_RB_09#BB_2#SW-01	39 / X1131408834#ALMA_RB_09#BB_2#SW-02
	 View other QA images...	 View other QA images...	 View other QA images...	 View other QA images...
centre frequency of image	660.4587GHz (LSRK)	679.9961GHz (LSRK)	663.3169GHz (LSRK)	677.0788GHz (LSRK)
beam	0.190 x 0.166 arcsec	0.185 x 0.161 arcsec	0.190 x 0.165 arcsec	0.185 x 0.162 arcsec
beam p.a.	-6.7deg	-8.3deg	-6.9deg	-7.4deg
final theoretical sensitivity	1.1 mJy/beam	1.4 mJy/beam	1.8 mJy/beam	1.2 mJy/beam
cleaning threshold	3.4 mJy/beam Dirty DR: 47 DR correction: 1.5	4.2 mJy/beam Dirty DR: 36 DR correction: 1.5	5.5 mJy/beam Dirty DR: 29 DR correction: 1.5	3.6 mJy/beam Dirty DR: 44 DR correction: 1.5
clean residual peak / scaled MAD	-5.13	4.80	4.22	4.15
non-pbcor image RMS	1.9 mJy/beam	2.3 mJy/beam	2.5 mJy/beam	2 mJy/beam
pbcor image max / min	53.7 / -19.9 mJy/beam	52.4 / -22 mJy/beam	53.9 / -28 mJy/beam	53.1 / -20.3 mJy/beam
fractional bandwidth / nterms	0.28% / 1	0.27% / 1	0.26% / 1	0.27% / 1
aggregate bandwidth	1.66 GHz (LSRK)	1.46 GHz (LSRK)	0.465 GHz (LSRK)	1.85 GHz (LSRK)
score	1.00	1.00	1.00	1.00
image file	uid___A001_X1296_X671.s34_0.LkCa15_sci.spw33.mfs.I.iter1.image	uid___A001_X1296_X671.s34_0.LkCa15_sci.spw35.mfs.I.iter1.image	uid___A001_X1296_X671.s34_0.LkCa15_sci.spw37.mfs.I.iter1.image	uid___A001_X1296_X671.s34_0.LkCa15_sci.spw39.mfs.I.iter1.image

Pipeline QA

Input Parameters

Tasks Execution Statistics

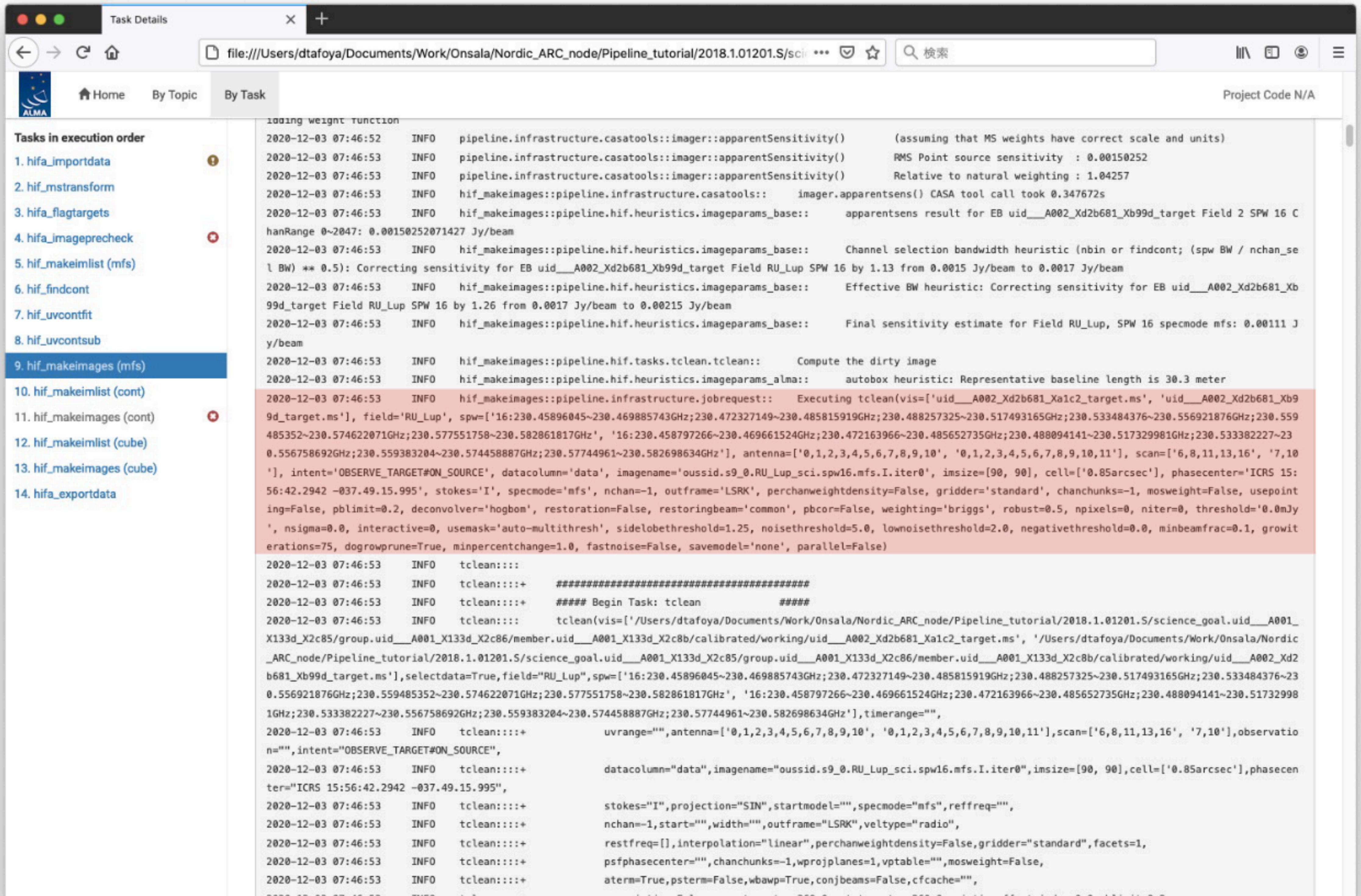
CASA logs for stage 9

- View or download stage9/casapy.log (344.3 KB)

Click for log at the bottom and can see the CASA call

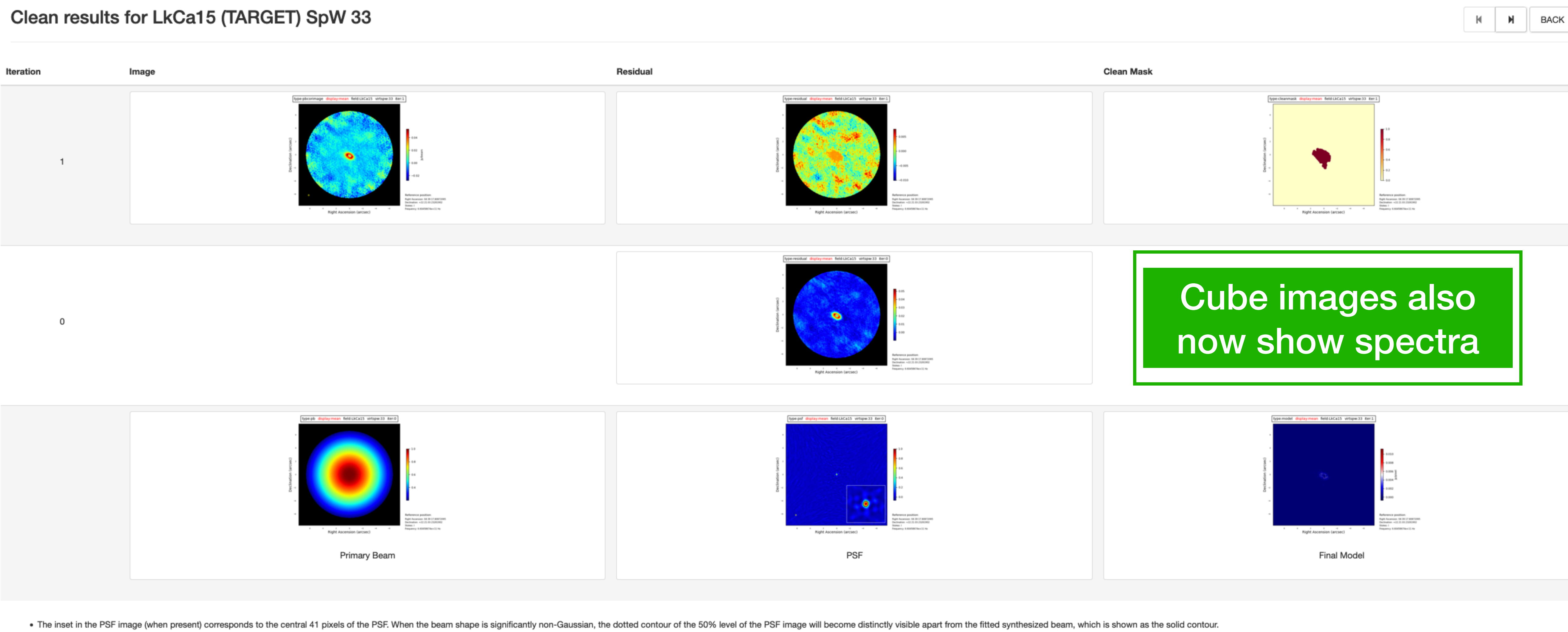
Pipeline imaged data

- Power of the pipeline imaging is easy to access image ‘quick look’ using the WebLog



Pipeline imaged data

- Power of the pipeline imaging is easy to access image ‘quick look’ using the WebLog



Cube images also now show spectra

NOTE: slight changes in viewing pages per Cycle (new PL features added)

Pipeline imaged data

- "Context" concepts:
    - The context is a trace of a given job and can sometimes be restored, depending on what actions occurred.
    - Can only start from the last action saved, everything else is lost. WebLog renders can get confused
        - If you ran many stages and something crashed then:
            - restart with a resume, but first you should probably delete everything since the last save (cumbersome/error prone) - if you don't delete products will be similarly named and the PL run becomes confused - context does not know they exist
            - restart again, in a clean directory - safer but can be more time consuming

Questions so far?

Brief image analysis

- Some analysis can be done in the CARTA **This afternoon CARTA session !!**

Image Information
- **imhead** — summarize and manipulate the “header” information in a CASA image

Image Manipulation
- **imsubimage** — Create a (sub)image from a region of the image
- **imcontsub** — perform continuum subtraction on a spectral-line image cube
- **imfit** — image plane Gaussian component fitting
- **immath** — perform mathematical operations on or between images
- **immoments** — compute the moments of an image cube
- **impv** — generate a position-velocity diagram along a slit

Image Information
- **imstat** — calculate statistics on an image or part of an image
- **imval** — extract the data and mask values from a pixel or region of an image

Image Reformatting
- **imtrans** — reorder the axes of an image or cube
- **imcollapse** — collapse image along one or more axes by aggregating pixel values along that axis
- **imregrid** — regrid an image onto the coordinate system of another image
- **imreframe** — change the frame in which the image reports its spectral values
- **imrebin** — rebin an image
- **specsmooth** — 1-dimensional smooth images in the spectral and angular directions
- **imsmooth** — 2-dimensional smooth images in the spectral and angular directions

Spectral line related
- **specfit** — fit 1-dimensional Gaussians, polynomial, and/or Lorentzians models to an image or image region
- **specflux** — Report details of an image spectrum.
- **plotprofilemap** — Plot spectra at their position
- **rmfit** — Calculation of rotation measures
- **spxfit** — Calculation of Spectral Indices and higher order polynomials
- **makemask** — image mask handling
- **slsearch** — query a subset of the Splatalogue spectral line catalog
- **splattotable** — convert a file exported from Splatalogue to a CASA table

Image Import/Export
- **importfits** — import a FITS image into a CASA image format table
- **exportfits** — write out an image in FITS format

CREDITS: Imaging Analysis in CASA by Feng Long - SMA school 2022 (older viewer commands): https://lweb.cfa.harvard.edu/sma-school/program/

Brief image analysis

- **imval**(imagename, region=‘’, box=‘’, chans=‘’, stokes=‘’)
- **imstat**(imagename, region=‘’, box=‘’ chans=‘’, stokes=‘’)

box = ‘512,512’ - one pixel
box = ‘blcX, blcY, trcX, trcY’ - box region

region = ‘circle[[512pix, 512pix], 50pix]’
region = ‘circle[[04h35m28.15s, +22d32m14.24s], 1.5arcsec’

- **immath**(imagename=‘name’, expr=‘IM0^2’, outfiile=‘’)

will square all data in image 0

- **immath**(imagename=[‘name1’, ‘name2’], expr=‘IM0-IM1’, outfile=‘’)

will subtract image 0 from image 1 (of input list)

EUROPEAN ARC
ALMA Regional Centre

CREDITS: Imaging Analysis in CASA by Feng Long - SMA school 2022 (older viewer commands): https://lweb.cfa.harvard.edu/sma-school/program/

Brief image analysis

- myfit = **imfit**(imagename, region=‘’, chans=‘’, stokes=‘’)

>print(myfit)

- **imsubimage**(imagename, outfile=‘’, chans=‘5~10’, region=‘’, box=‘’)

takes channels 5 to 10 out of a cube

- **immoments**(imagename=‘name’, moments=[0], chans=‘10~50, outfile=‘image_mom0’)

integrated intensity using channels 10 to 50 (Jy/beam.km/s units)

- **immoments**(imagename=‘name’, moments=[1], chans=‘10~50, outfile=‘image_mom1’, includepix=[3*0.05])

if sigma = 0.05Jy/bm, this is >3sigma

(‘range=[-20km/s,10km/s], restfreq=230.5GHz’)

EUROPEAN ARC
ALMA Regional Centre

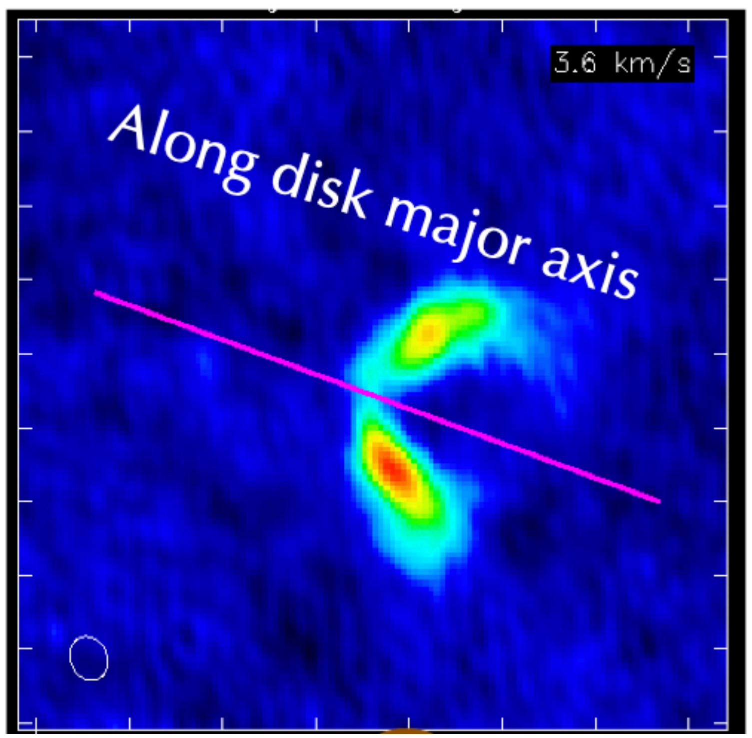
CREDITS: Imaging Analysis in CASA by Feng Long - SMA school 2022 (older viewer commands): https://lweb.cfa.harvard.edu/sma-school/program/

Brief image analysis

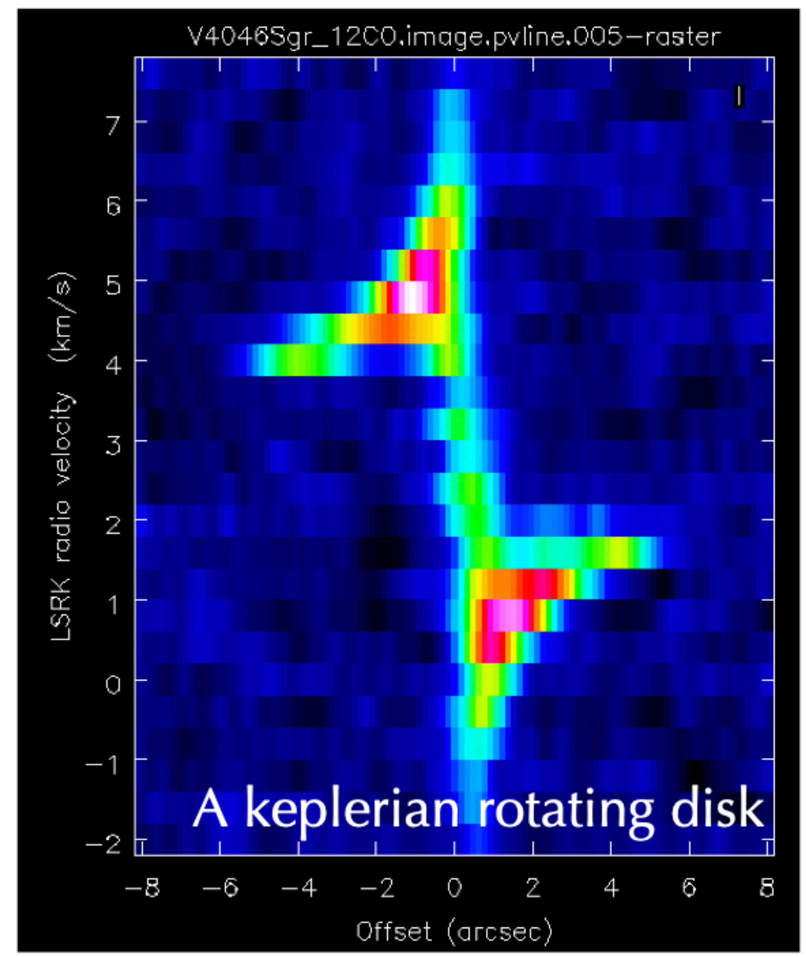
- **impv**(imagename, outfile='source_line_PV.image', chans='', mode='length', center=['18h14m10.5s, '-32d47m35.27s'],length='15arcsec', pa='70deg')

pix units also

Single channel



PV image - rotating disk



CREDITS: Imaging Analysis in CASA by Feng Long - SMA school 2022 (older viewer commands): https://lweb.cfa.harvard.edu/sma-school/program/

Brief notes on selfcal

- Interferometry uses phase referencing calibration - transfers solutions from a phase calibrator in position and time
  - Inexact
- If the data has enough SNR, and the image is ‘good’ then the clean model - of ‘real’ sources - can be used to ‘self-calibrate’ the data
  - The phases vary due to the atmosphere and the only way to correct for the duration of the observation on the target is with self-calibration
    - Short term - faster than the repetition of phase calibrator visits
    - Phase offset can also be solved with longer term solutions - imparted due to distant calibrator - longer term (10s minute timescales)
  - Imaging needs to have saved the source model, self-calibration usually iterates from longer to short times if images improve. Worse conditions and bright source will show the most improvement

EXTRA READING: https://ui.adsabs.harvard.edu/abs/2022arXiv220705591R/abstract
https://ui.adsabs.harvard.edu/abs/2018arXiv180505266B/abstract

Brief notes on selfcal

- Typically use the "strong" continuum
  - in **gaincal** selection of only the continuum SPWs and channels:
    - spw = '5:0~10;50~490;7:20~200;350~1023' -like the selection for imaging (manually)
- Can use very strong lines, i.e. maser emission
  - make sure an image has been made for these channels only (and a model saved)
  - in **gaincal** selection of only the **maser** SPW and channels

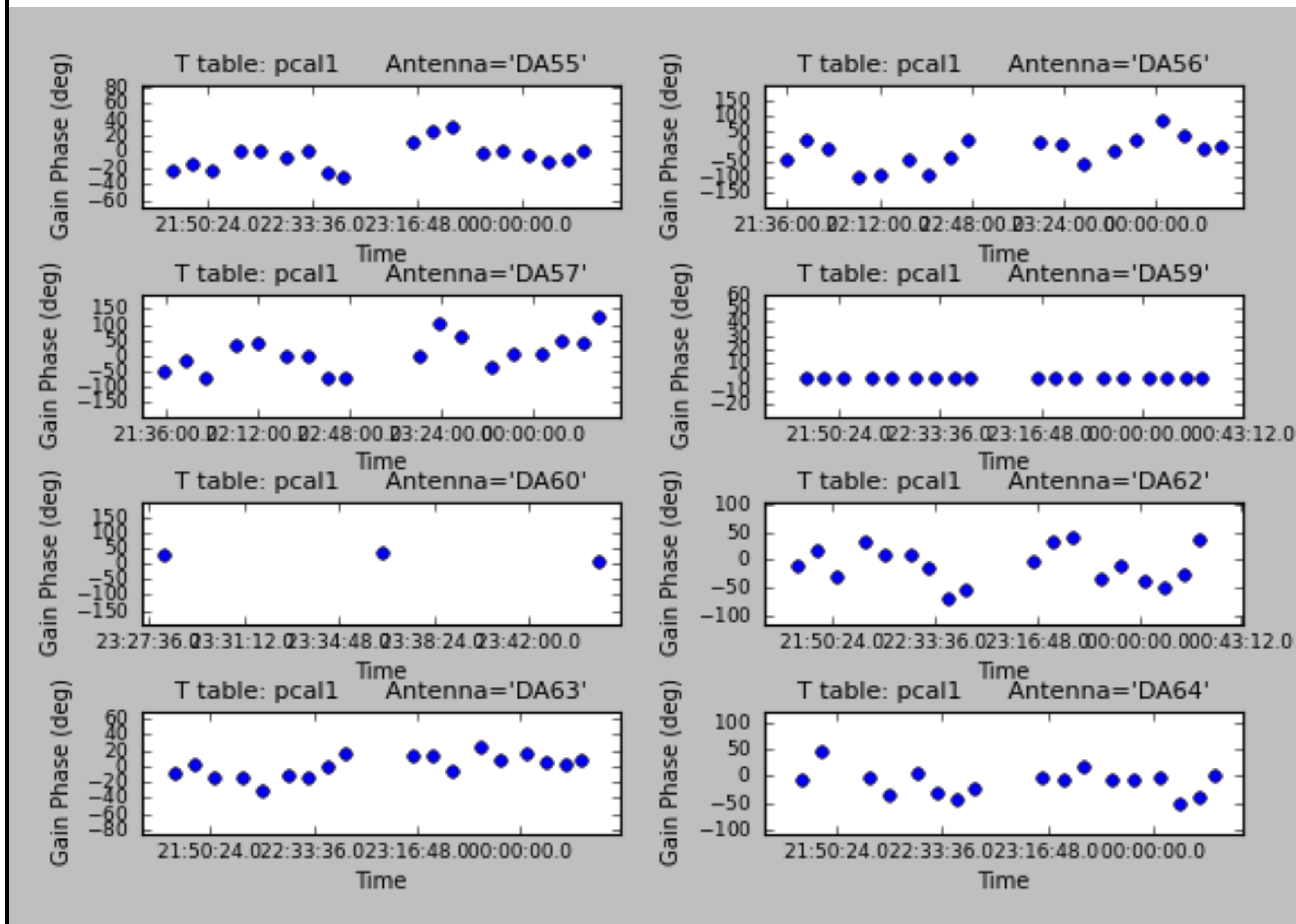
- In **gaincal**, start with a per SPW, and timescale 'inf', then apply, then image
  - can combine SPWs for SNR - they should have been aligned by the standard calibration process
  - if the image improves, and solutions look to track the atmospheric phase variations, try with shorter interval. Lowest is 'int' - integration time
  - if the phase solutions look like noise - ***you don't have enough SNR***

EXTRA READING: https://ui.adsabs.harvard.edu/abs/2022arXiv220705591R/abstract
https://ui.adsabs.harvard.edu/abs/2018arXiv180505266B/abstract

Brief notes on selfcal

```
## 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)
```

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



- then apply

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

- then re-image

Brief notes on selfcal

```
## 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)
```

per scan

```
# 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)
```

1/3 of a scan

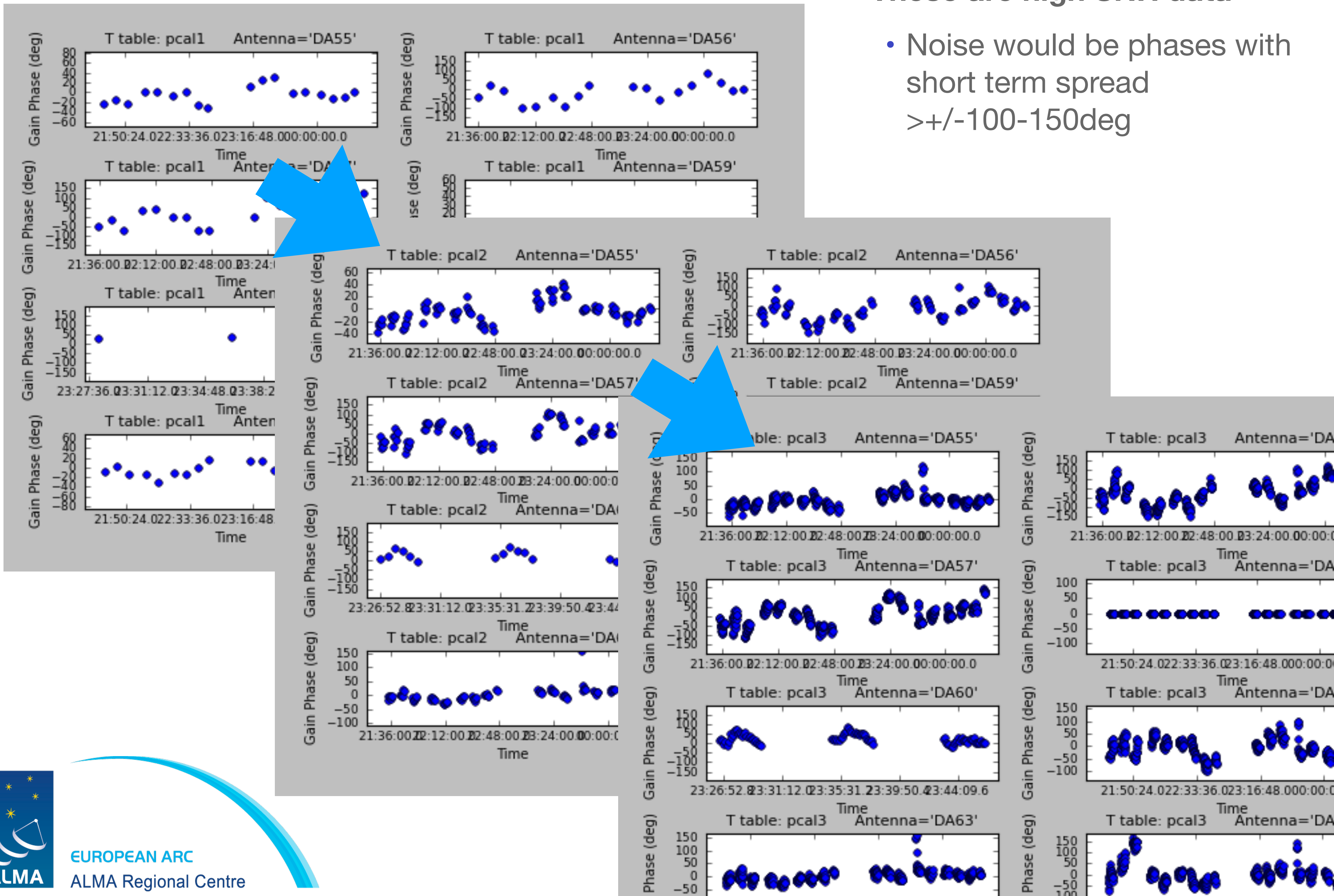
```
# now to int - 6 sec
os.system('rm -rf pcal3')
gaincal(vis=visname,
        caltable='pcal3',
        gaintype='T',
        refant='DA59',
        calmode='p',
        combine='spw',
        spw=spwcont,
        field='0',
        solint='int',
        minsnr=3.0,
        minblperant=4)
```

integration time

- Loops:
  - gaincal
  - applycal
  - image

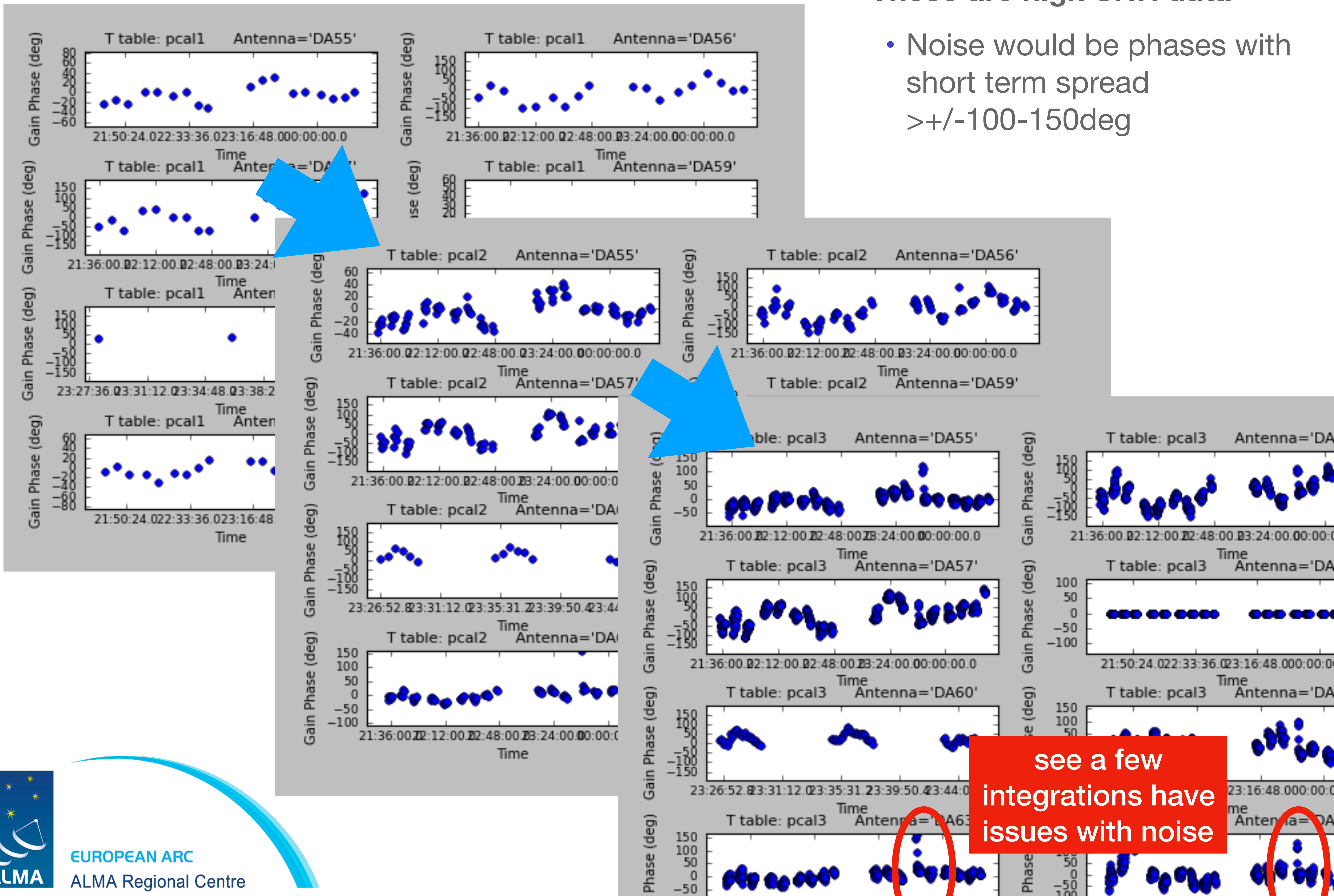
Brief notes on selfcal

- **These are high SNR data**
  - Noise would be phases with short term spread >+/-100-150deg



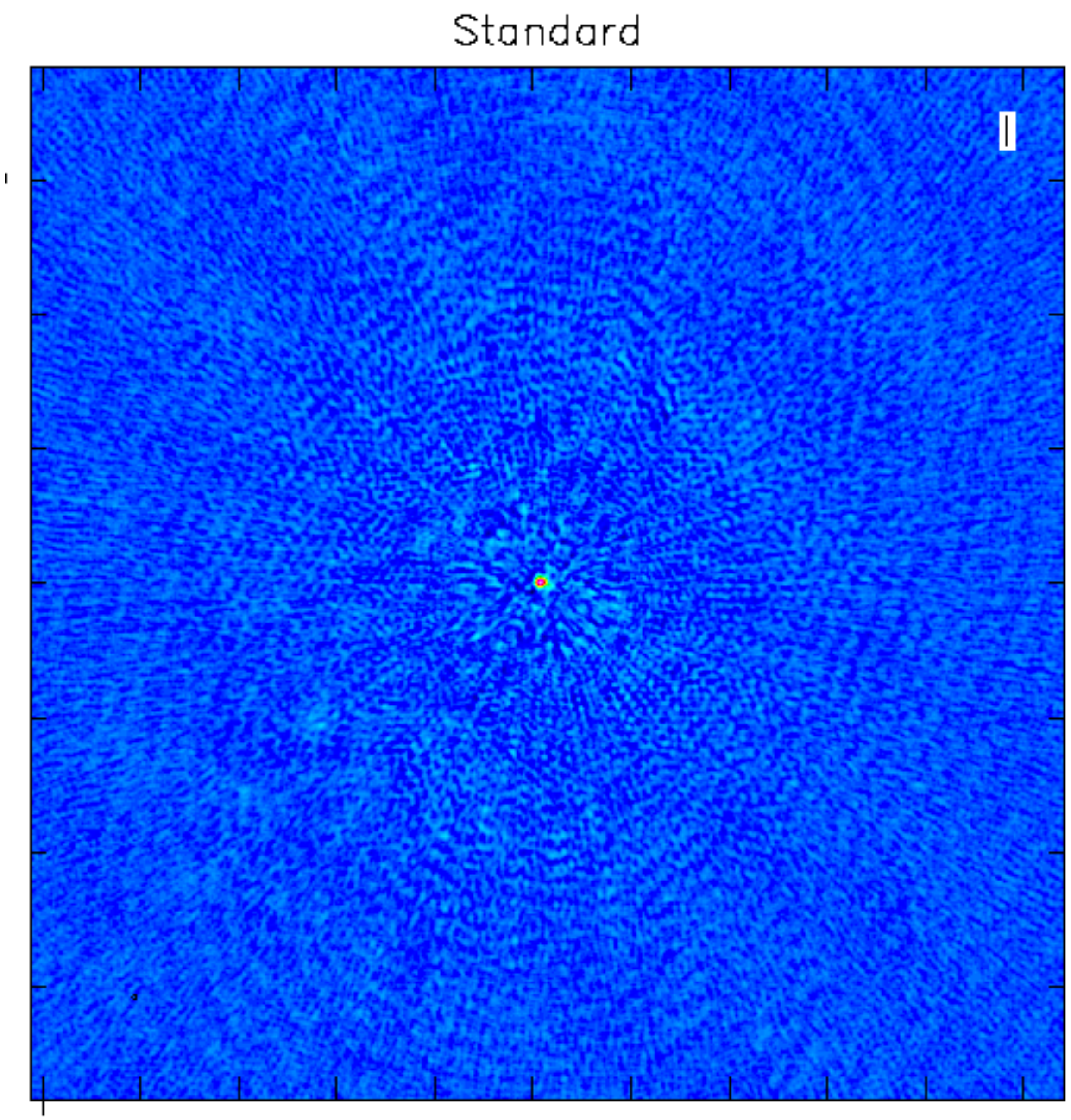
Brief notes on selfcal

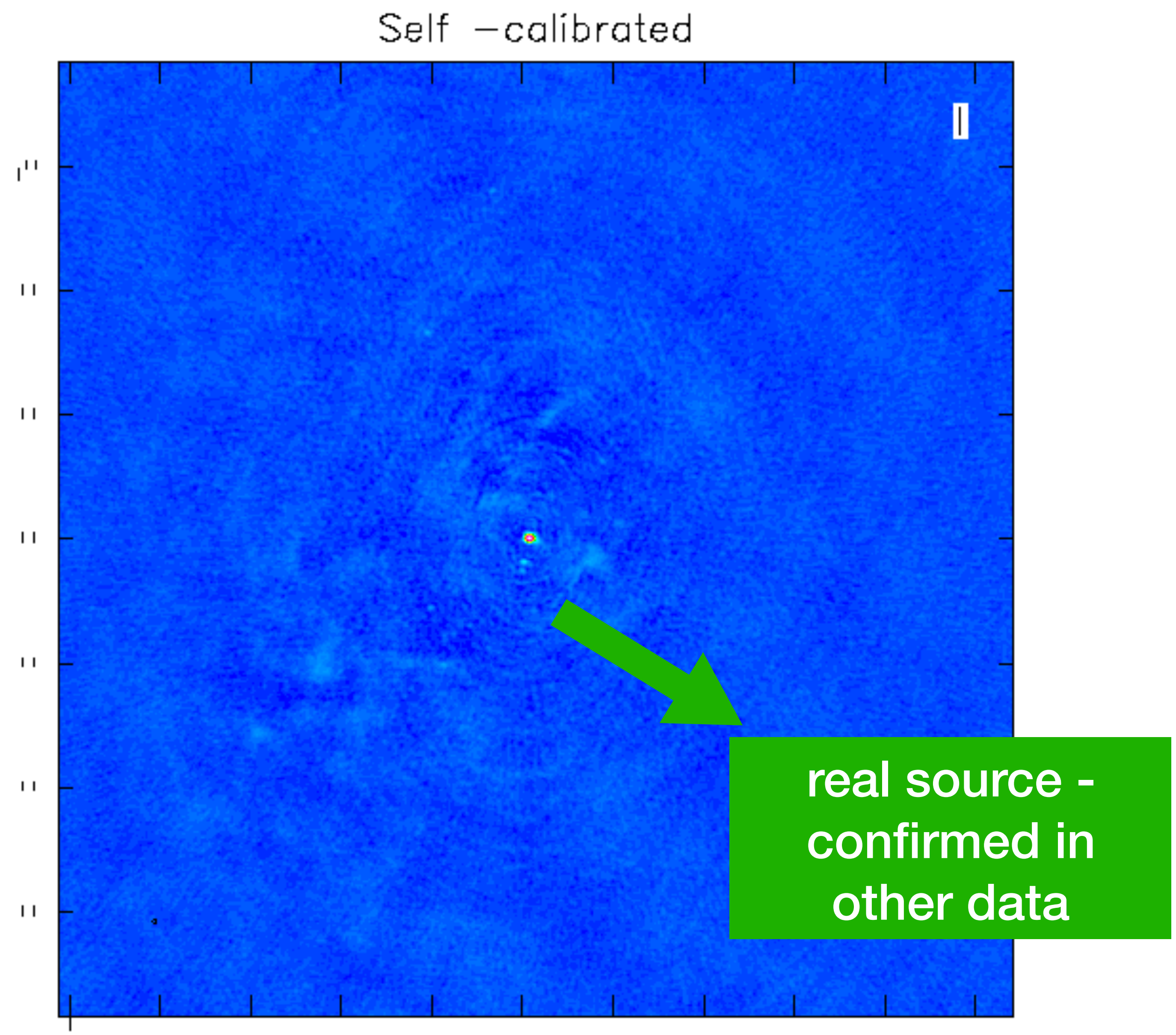
- **These are high SNR data**
  - Noise would be phases with short term spread >+/-100-150deg



EUROPEAN ARC
ALMA Regional Centre

Brief notes on selfcal





S/N ~ 120

Pk ~ 50 mJy/bm

S/N ~ 850

Pk ~ 70 mJy/bm

EUROPEAN ARC
ALMA Regional Centre

Final Questions?

EUROPEAN ARC
ALMA Regional Centre

OPTICON
RadioNet
Pilot