

# Workshop

ALMA, CASA, DALI, etc.  
in (transition) disks

Nienke van der Marel

# ALMA/CASA

- Interferometry data
  - Basics interferometry: spatial scales, uv-coverage, high-frequency
  - Calibration and flagging
- A new dataset
  - Delivery contents
  - Measurement sets
  - Basic CASA commands
  - QA2: what to check
- CASA viewer: tricks and tools

# ALMA/CASA

- Data reduction
  - Cleaning
  - Weighting
  - Reduction:
    - Continuum
    - Line
    - Parameters
    - Tips/tricks
    - Output: export, moment maps, inspection
  - Serendipitous lines
  - Self-calibration
  - Not really discussed

# DALI

- Motivation
- Basic setup: input/output
- Physics
- DALI structure/run-through
- Different modeling codes available
- Applications/literature so far

# DALI

- Using DALI
  - Careful checks output
  - Output tools
  - Analysis approach
  - Time (in)dependence
  - Chemistry (in)completeness
- DALI specials

# ALMA/CASA

- Visibility curves and analysis
- Simulating
  - Simobserve
  - Mapping on observed visibilities
- Analysis of ALMA data
- ALMA archive
- Proposals
  - The OT
  - A successful proposal

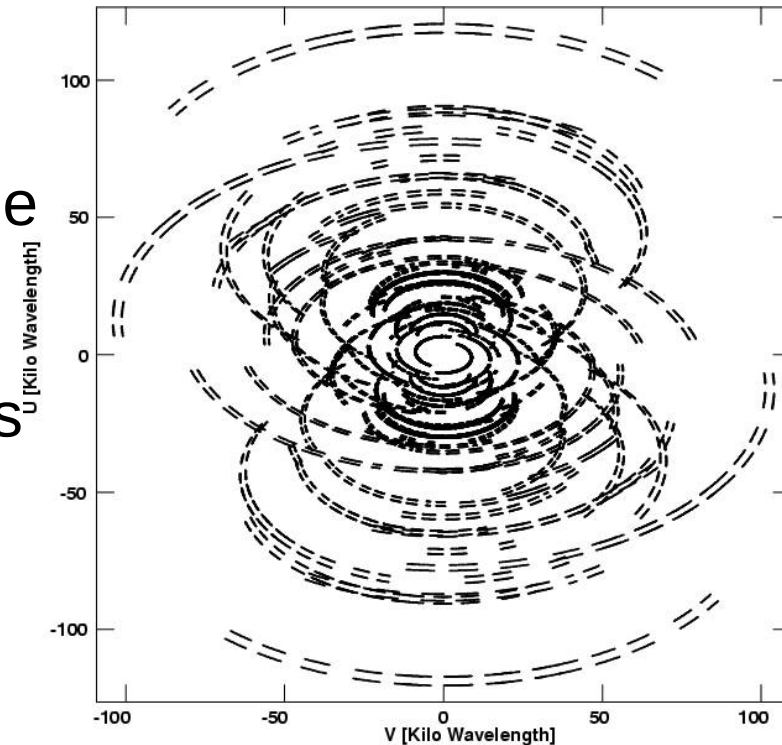
*ALMA/CASA*

# ALMA/CASA

- Available online:
  - Allegro workshops: [www.alma-allegro.nl](http://www.alma-allegro.nl)
  - NRAO:  
<https://science.nrao.edu/science/meetings/2014/14th-synthesis-imaging-workshop/lectures>
  - CARMA summer school (Mel Wright):  
[http://w.astro.berkeley.edu/~wright/school\\_2014.pdf](http://w.astro.berkeley.edu/~wright/school_2014.pdf)
  - CASA guides:  
[https://casaguides.nrao.edu/index.php/Main\\_Page](https://casaguides.nrao.edu/index.php/Main_Page)
  - ALMA/CASA tutorials
  - ALMA Technical handbook (each Cycle)

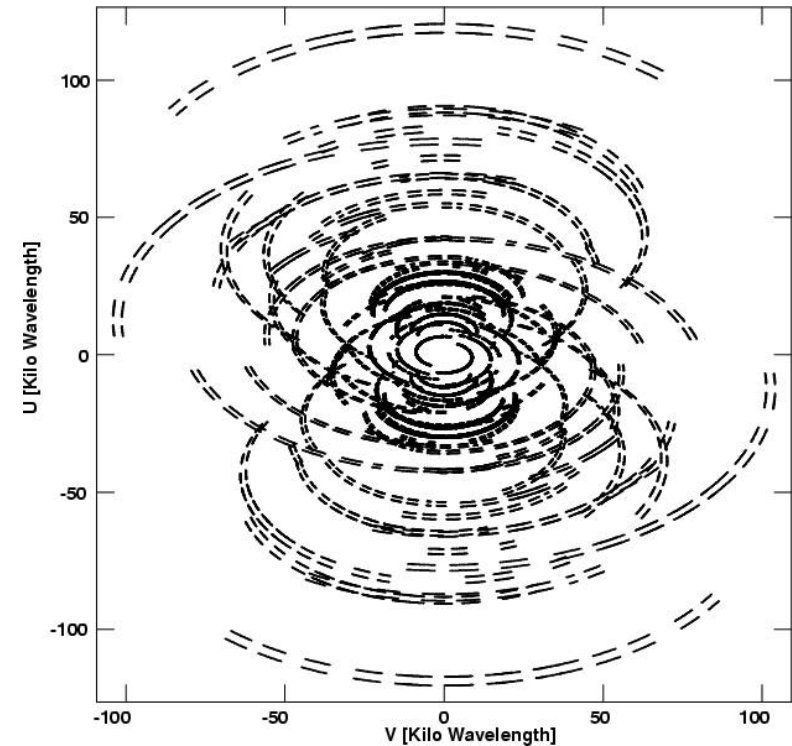
# Basic interferometry

- Combine observations from multiple antennas to obtain high spatial resolution
- uv-plane:
  - Plane in which we observe
  - reverse Fourier-transform of x,y-plane
- Baseline:
  - projected distance between antennas (source Dec/RA vs latitude/obs time)
  - changes with time (Earth rotation)



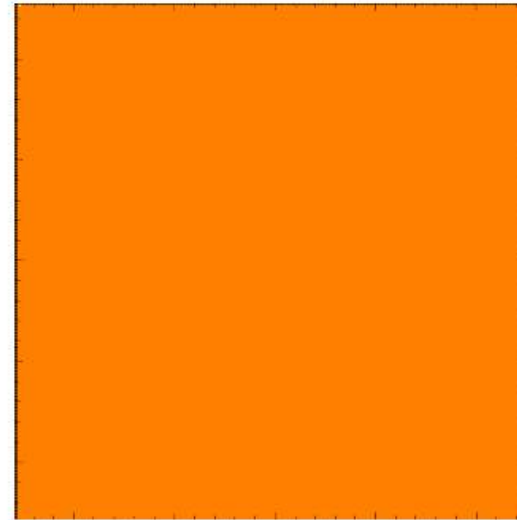
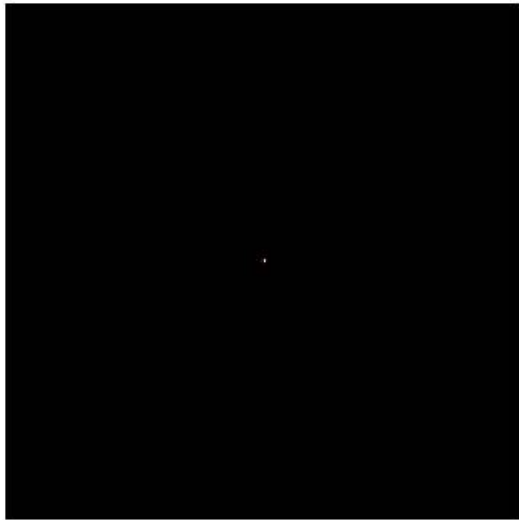
# Basic interferometry

- uv-coverage:
  - Part of the uv-plane sampled
  - Proper sampling required for FT
- uv-distance:
  - baseline length
  - reversely prop. to spatial scales
  - spatial filtering: small/large scale variations
- Beam size:
  - Resolution element
  - Elliptical, depends on uv-coverage (in 2D)



# Basic interferometry: FT

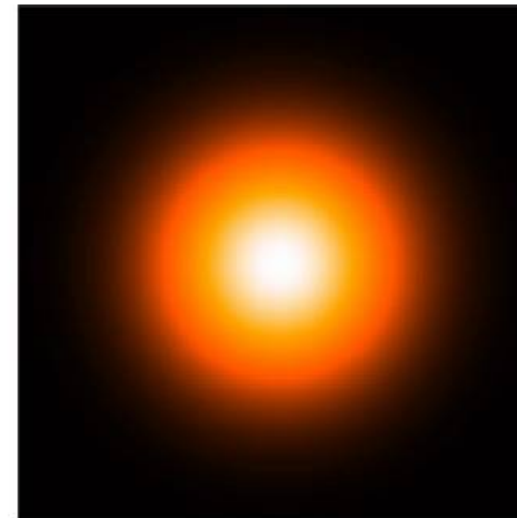
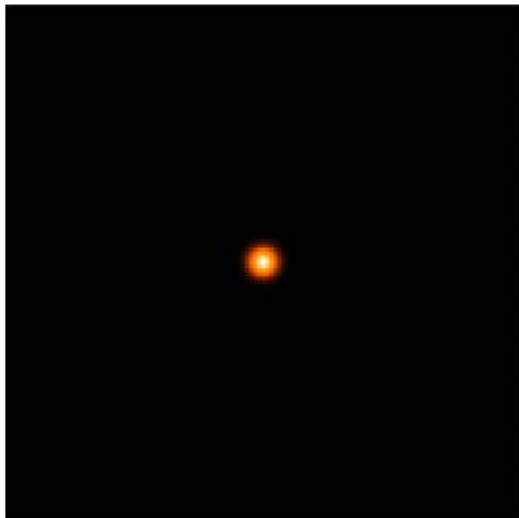
$\delta$  Function



$\text{Amp}\{V(u,v)\}$

Constant

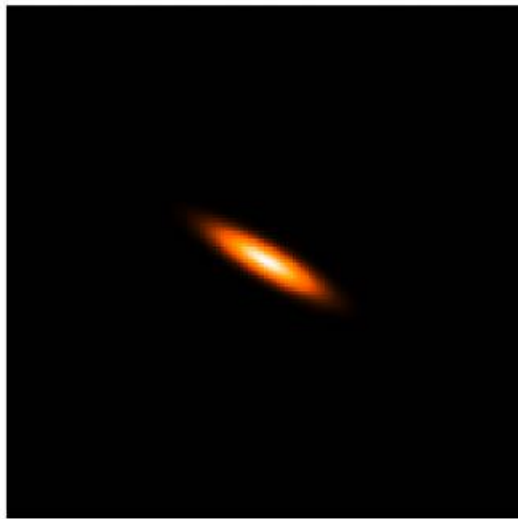
Gaussian



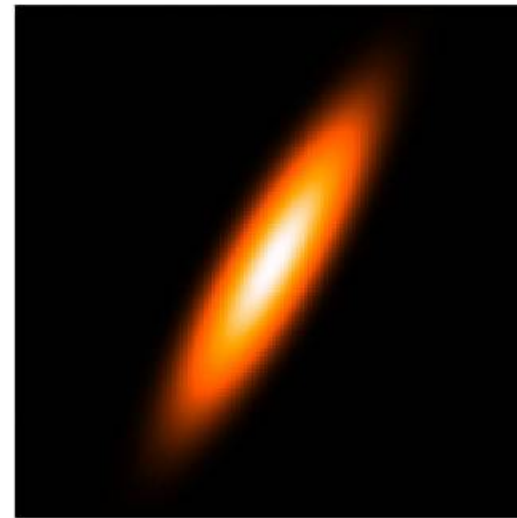
Gaussian

# Basic interferometry: FT

elliptical  
Gaussian



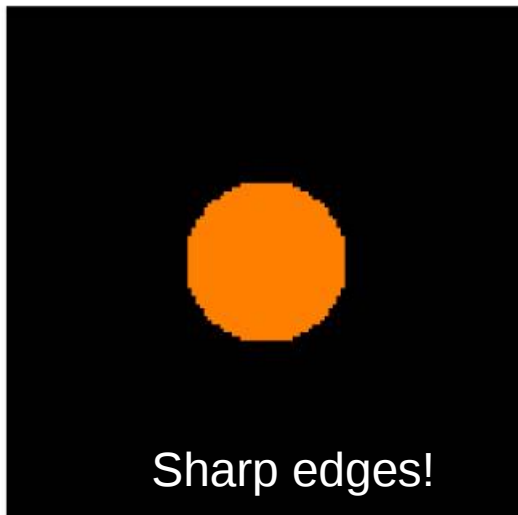
$\Downarrow$



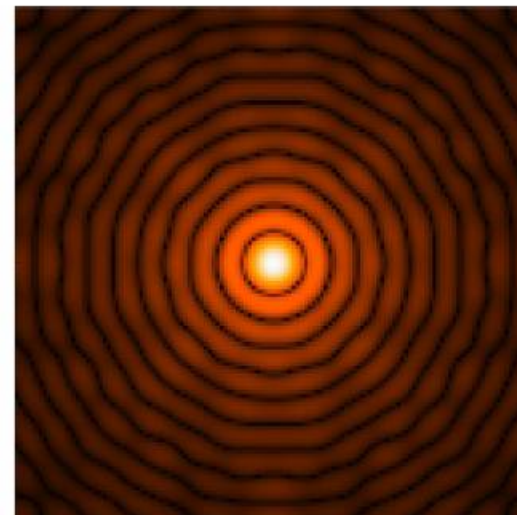
$\text{Amp}\{V(u,v)\}$

elliptical  
Gaussian

Disk



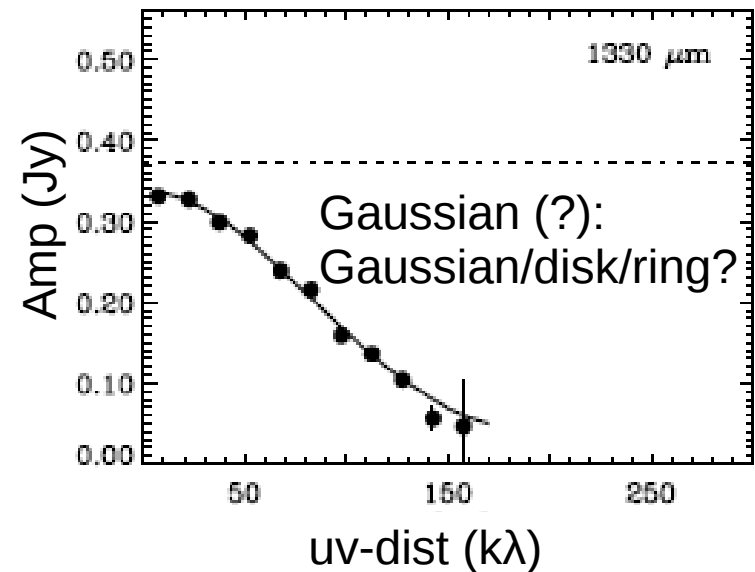
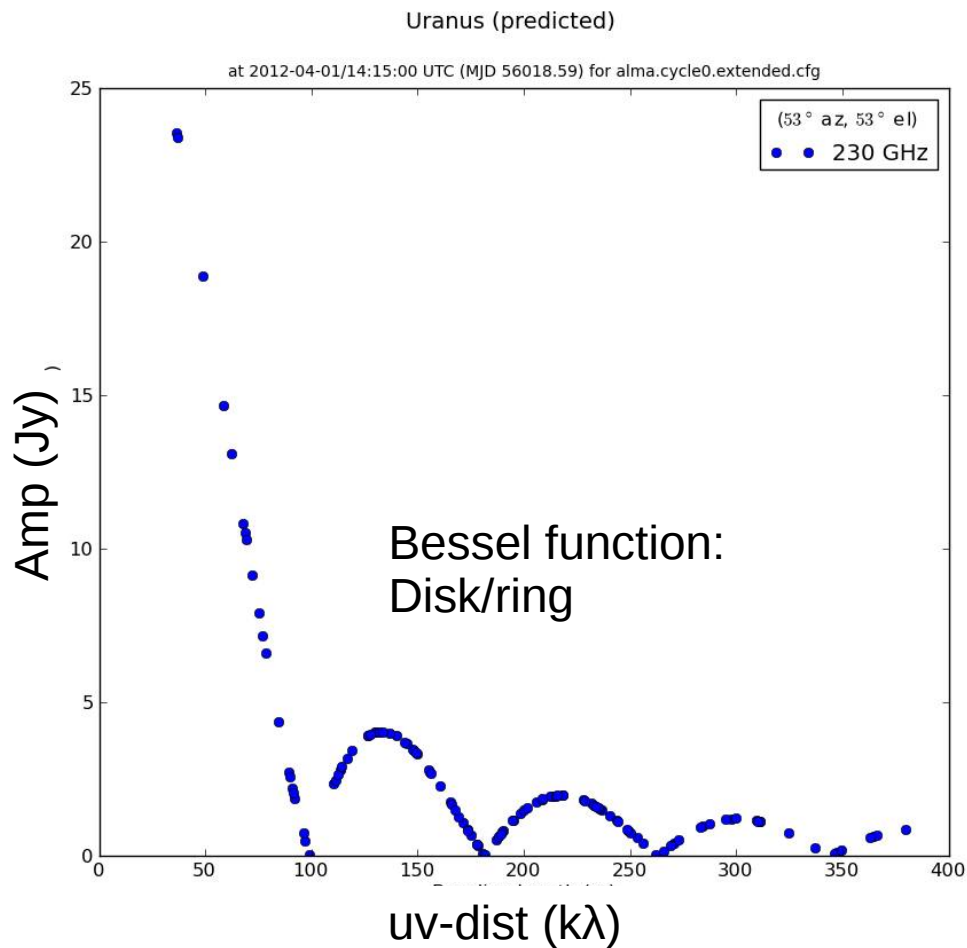
$\Downarrow$



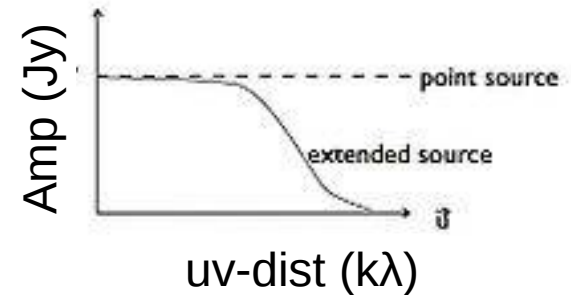
Bessel

# Basic interferometry

- Visibility curve: amp vs uv-distance

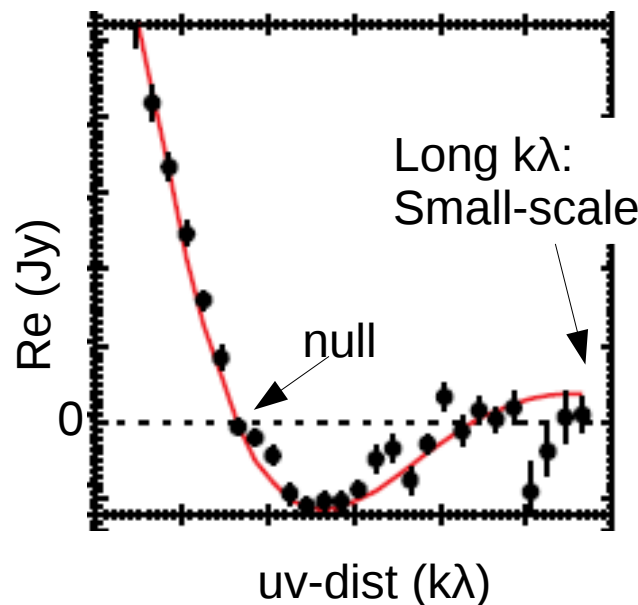


Constant: unresolved point source

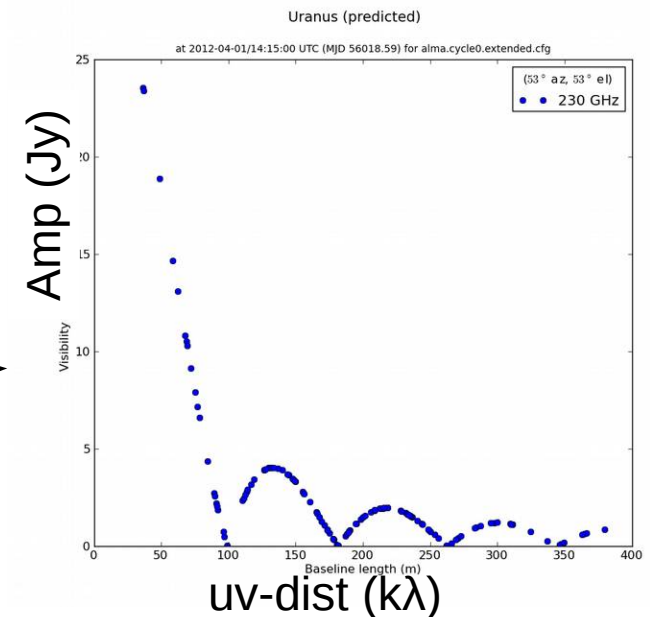


# Basic interferometry

- Visibility curve:  $\text{amp} = \sqrt{\text{Real}^2 + \text{Imag}^2}$ 
  - Real: radial axisymmetric structure wrt phase center
  - Imag: azimuthally asymmetric structure
- Transition disks: ring  $\Rightarrow$  Real = Bessel function  
Imag = 0 (not true for asymmetries)

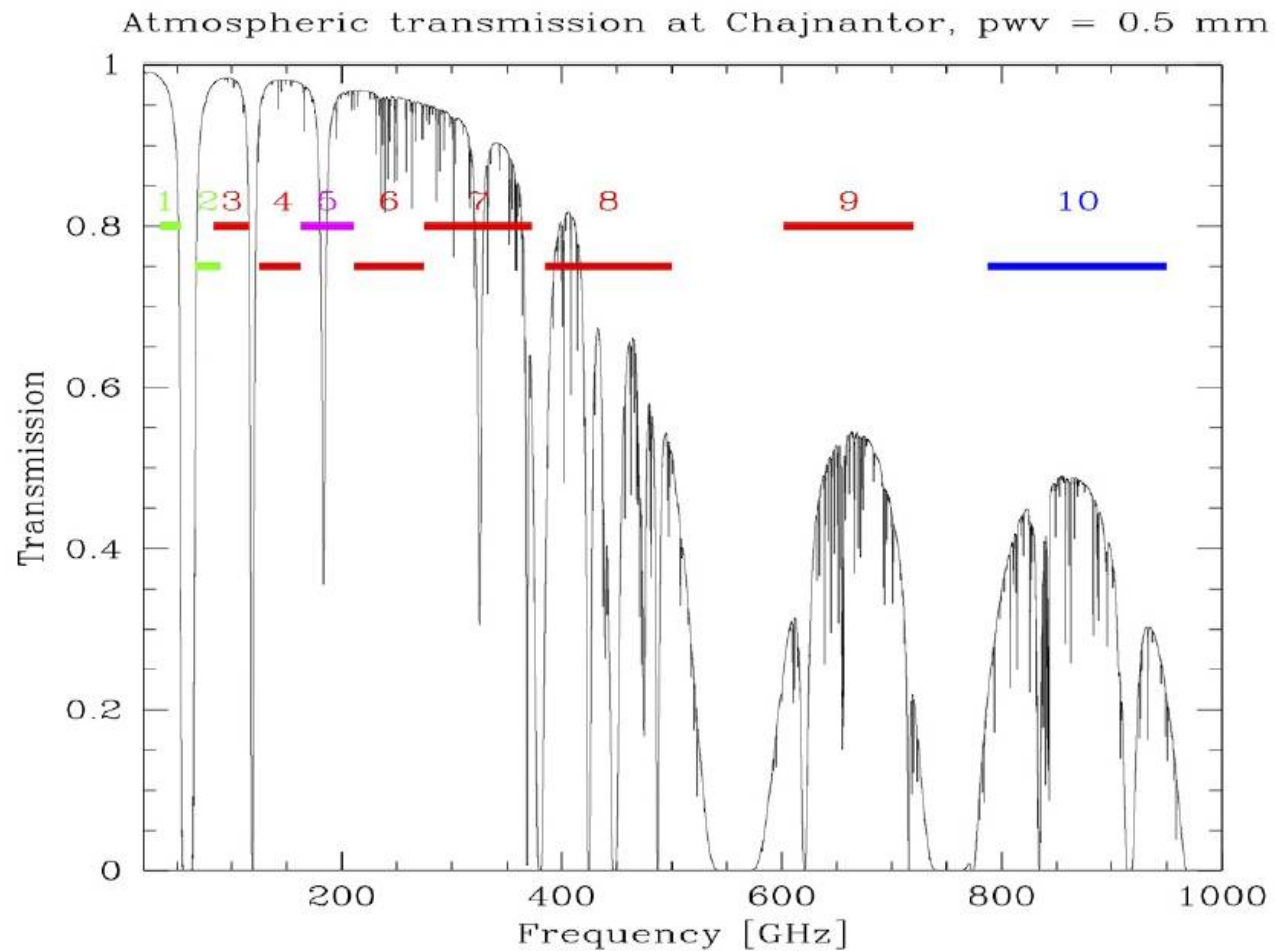


squared



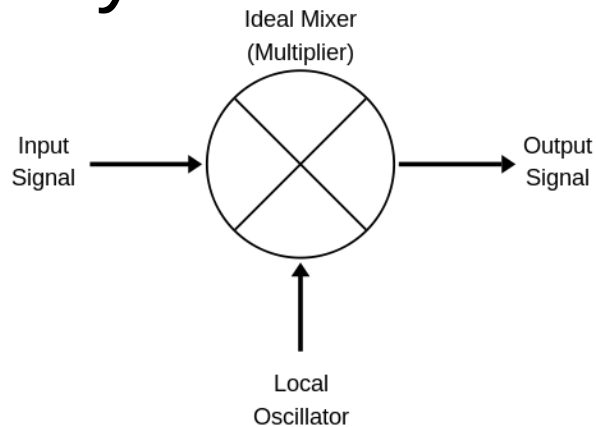
# Basic interferometry

- ALMA Bands: atmosphere



# Basic interferometry

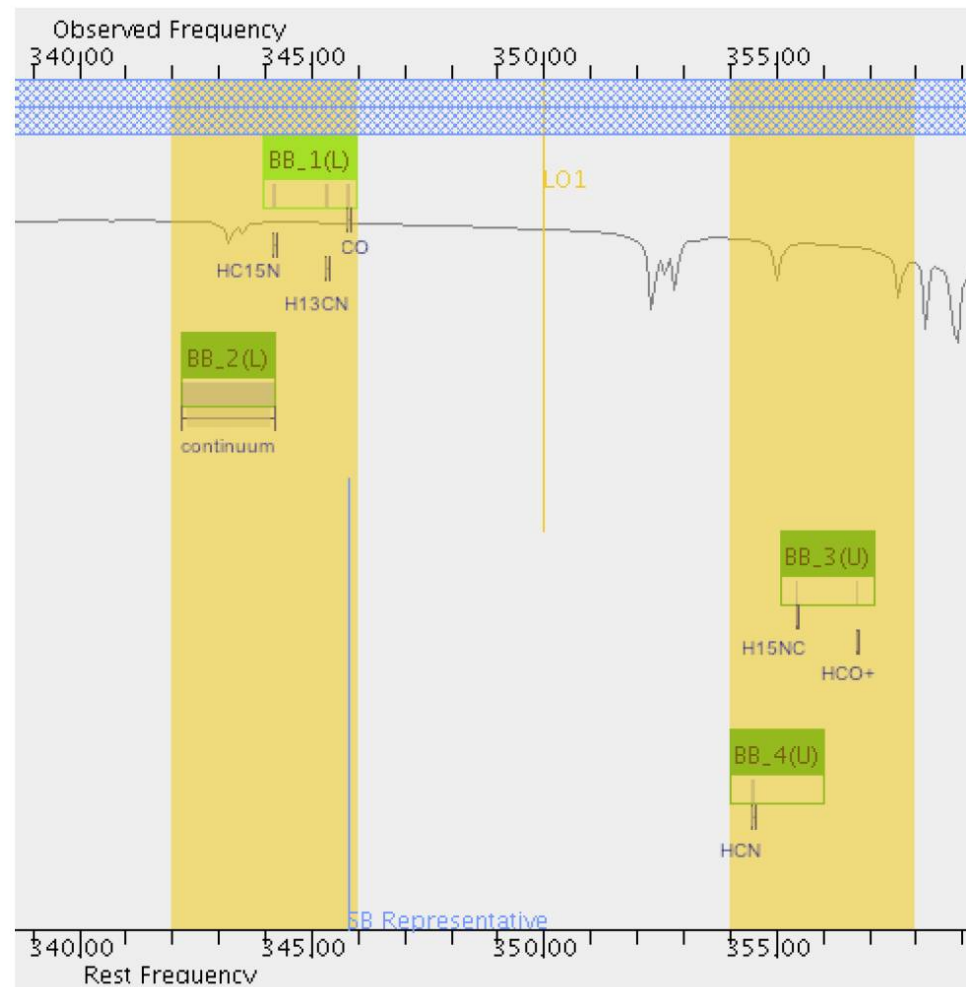
- Heterodyne receiver: mixing local oscillator



- Frequency spectral windows always within USB and LSB:  $LO \pm IF$  ( $IF \sim 4-8$  GHz)
- ALMA: B9 double, other bands dual SB: freedom in frequency range

# Basic interferometry

- Setup spectral windows: 4 basebands
- Multiple spws within baseband
- Bandwidth vs spectral res.
- FDM and TDM windows
  - We like serendipitous lines: FDM
- $d\nu \sim dv$ : depends on  $v$  (Band
- Observing frequency  
~ Earth velocity at obstime:  
sometimes multiple spws  
in data set due to multiple  
obstimes



# Calibration

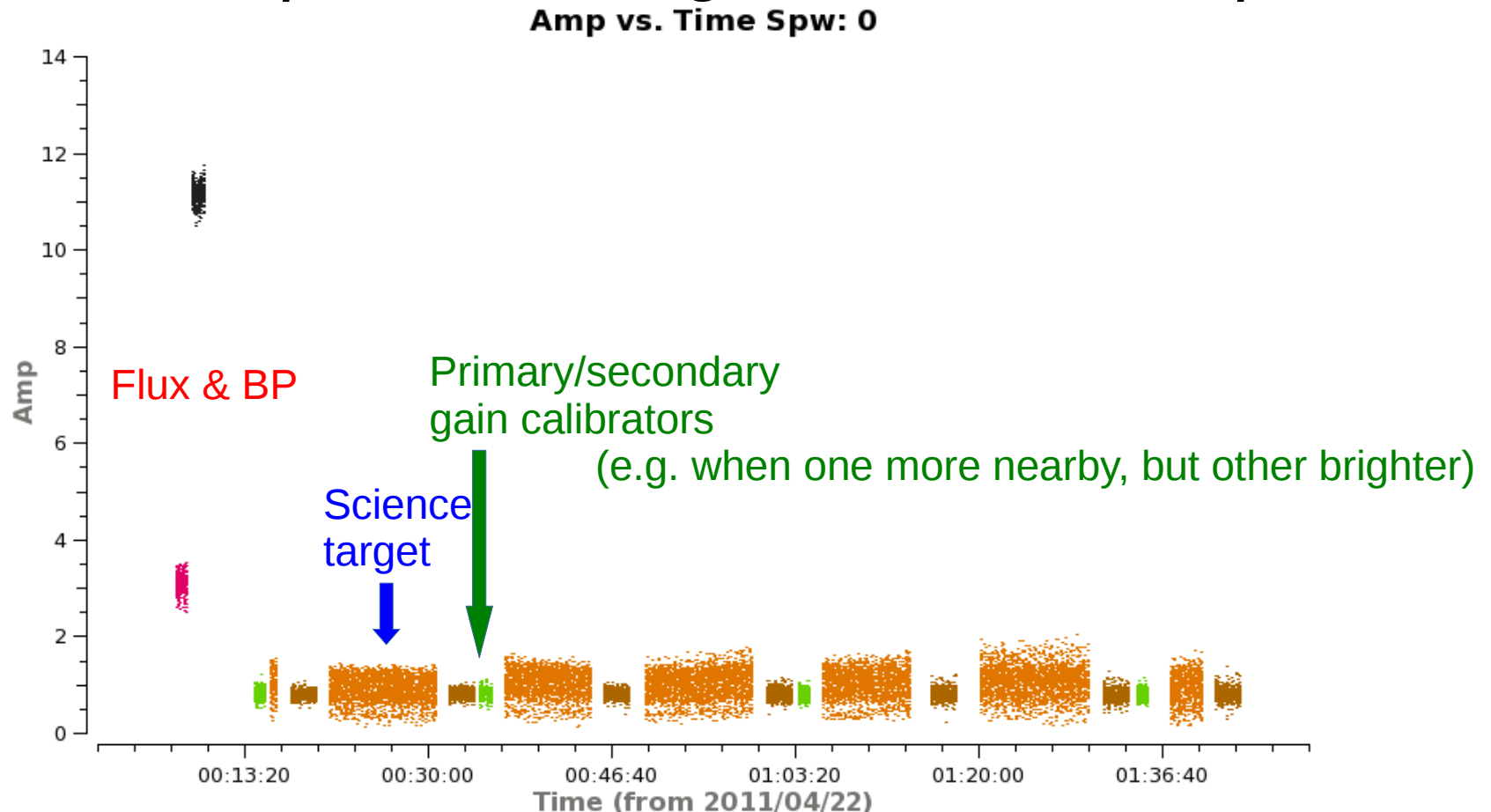
- We measure amplitudes and phases of sources
- Time dependent:
  - Weather
  - Atmosphere
  - Antenna response
  - Instrument response
  - Moving antennas
  - Etc.
- So we have to calibrate for **each** antenna **all** the time
- ALMA calibration is done for you (QA2) so basics only now

# Calibration

- A priori (handled by observatory):
  - Antenna positions
  - $T_{\text{sys}}$
  - WVR (phase)
  - Pointing/focus
- Physical calibrators:
  - Flux calibrator (often solar system)
  - Bandpass calibrator
  - Gain calibrator (quasars): atmosphere
  - (Polarization calibrator: not further discussed)

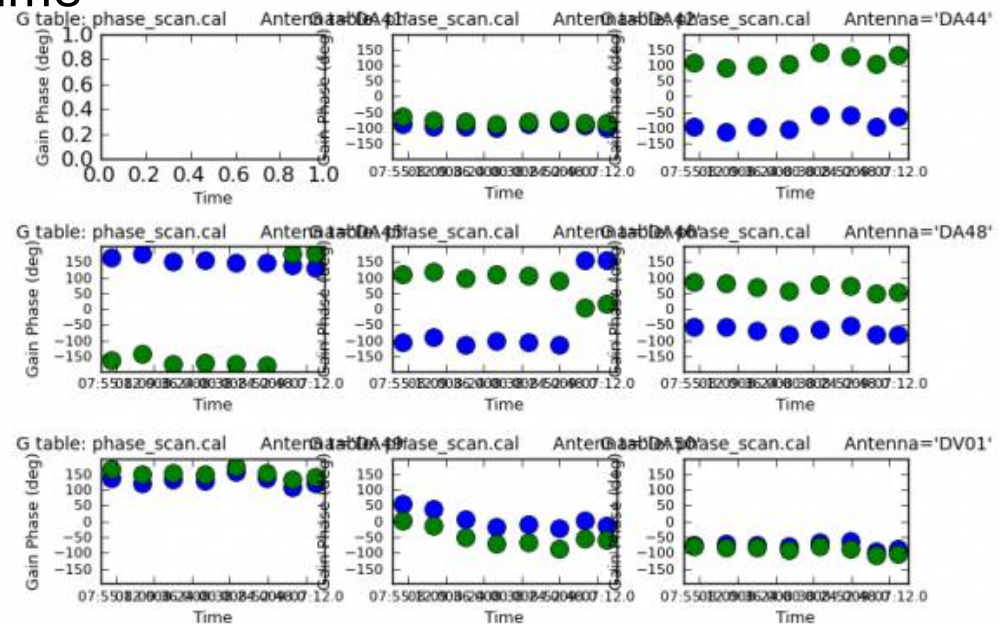
# Calibration

- Gain calibrator: observe regularly a bright target nearby your science target to measure phase/amplitude changes in the atmosphere



# Calibration

- Gain calibrator:
  - Phase/amp varying smoothly with time
  - During calibration (QA2):
    - Measure phase/amp of gaincal
    - Weird outliers are flagged: data points no longer included
    - Interpolate in between gaincal measurements: the phase/amp during science target
    - Mostly automatic, with output plots
  - Flagging:
    - No removal, just a mark that time data point is excluded in further processing
    - Most important part of calibration: only calibrate with what you trust!
    - Also bad antennas, weird results BP/Flux, edge channels



# ALMA datasets

- Dataset:
  - QA1: first check at telescope; QA2: fully reduced (pipeline/person)
  - completely calibrated, basic imaged and checked (QA2) **but often room for improvement**
  - Measurement set (calibrated.ms) contains all targets observed, including calibrators, in case you want to look at them
  - After Cycle 0: always have to run the full calibration script: only uncalibrated data and final products included

# ALMA dataset: folder with subfolders

- README file
  - Comments of the QA2 (rms noise, problems)
  - CASA version to run
- Calibration: calibration tables used
- Log: CASA logs
- Product:
  - Output fits data: X.flux.fits (primary beam used) and X.pbcor.fits (primary beam corrected)  
=> latter is the science data
  - Only doing the parts mentioned in the technical part of the original proposal: specific lines/continuum
  - Sometimes includes self-cal and calibrator data: depends on QA2-er
- QA: webpage (>Cycle 2) or separate plots (<Cycle 2) calibration
- Script: scripts for calibration, flux calibration and imaging
- Raw: raw data
- Calibrated: includes calibrated measurement sets (is created during calibration)

Name	Size	Type
▶ calibrated	10 items	Folder
▶ calibration	12 items	Folder
▶ log	7 items	Folder
▶ product	4 items	Folder
▶ qa	24 items	Folder
▶ raw	0 items	Folder
▶ script	8 items	Folder
▶ selfcal	110 items	Folder
▶ README	6.3 kB	Text

# ALMA dataset

- ALMA science archive: <https://almascience.eso.org/aq/>
- Make sure to have large amount of disk space: >500 Gb or use helada/chaxa/tebinquiche
- Download data (product only or including raw)  
=> go to sleep
- Unpack data (tarballs)  
=> go to sleep
- Check your product files, the QA2, the comments in README
- If happy: run the scriptForPI.py in the **exact correct** CASA version  
=> go to sleep
- Split out your science target from calibrated.ms  
=> have a long lunch/coffee break
- Get started with your new data!

# Measurement set

- MS file contains all observational visibility data
- Calibrated.ms sometimes consists of multiple observing blocks, taken at different times, which have been calibrated separately
- MS “file”: complex folder structure  
(use `cp -r` and `chmod -R`: slow due to # files)
- Fields: different targets
- Columns:
  - Data: raw data
    - When split, corrected data is put into data column new MS
  - Corrected: data on which calibration solutions are applied
    - Often taken as default column to read from: error saying it's empty
  - Model: column of synthesized model, e.g. when setting flux calibrator, simulating observations or self-calibration
- Inspection: `listobs`, `plotms`

# ALMA dataset

- Take a look at B7-isot of IRS48 (Cycle2)
- `listobs(vis='IRS48/B7-CO-isot/calibrated.ms')`
  - Output in logger
  - Two observation dates
  - 5 calibrators: picked automatically, apparently some difference between the two dates
  - 12 spws: twice the same setup of 6 spws, different velocity
  - Note Ch0 frequency and sign Chanwidth!
  - 57 antennas: apparently moved between observation sets

Ex 1a

# ALMA dataset

- `plotms(vis='IRS48/B7-CO-isot/calibrated.ms')`
- Plot any parameter against any parameter in the dataset
- Slow: too much data => narrow down within what you want to check
  - Select 1 field/1 channel/1 spw
  - Averaging: channels, time (1e10), baselines, spws
- Choose parameters to plot in Axes
- Error column: choose 'data' in Axes
- Canvas: for output in PDF/graphic
- Export: possible to output data to text file for further processing
- Display: for multiple color labeling

Ex 1b

# Basic CASA

- Open CASA in the directory where you want to work (logging)
- Terminal commands within CASA: `> os.system('rm -rf target*')`
- Copy your task commands to a script file, even when run manually: often necessary to return or check what you have done
- The last used parameters of each task are saved in `task.last` (working directory)
- Careful with Ctrl+C!
- Logger displays what is being done
- Two log files (working directory)
  - `ipython*.log` => input (tasks, command line)
  - `casapy*.log` => output logger

SHOW

# Basic CASA

- `inp(task)` => display task parameters
- `inp` => display current task parameters
- Careful with parameters in different tasks: sometimes wrongly set because of the same name
- `task()` => run task with currently set parameters
- `task(par1=A,par2=B,par3=C)` => run task with the A,B,C parameters set
- `go` => after parameters are set, run the current task
- `default(task)` => set default task parameters
- `tget(task)` => get the parameters from the `task.last` in current directory
- Parameter value is almost always a string

# Basic CASA

- Useful commands until now
  - Listobs
  - Plotms (outside CASA: casaplotms)
  - Tget
  - Inp
  - Default
  - Concat: for concatenating multiple observing blocks
  - Split: for taking out part of data and/or averaging data
  - Viewer (outside CASA: casaviewer)

# Basic CASA

- Concat (slow)
  - Freqtol and dirtol sometimes required
- Split (slow)
  - Width: averaging channels in each spw: safer to use no larger than #channels in spw
  - Timebin: averaging data in time: saves a lot of space, but no more than a few minutes, and after self-calibration
  - Spw notation: '0:100~300,1:200~1800' etc.

Ex 1c

# QA2

- In principle this is done and checked, but always good to double-check, certainly with older datasets
- Open in Firefox:  
[IRS48/qa/pipeline-20150810T213106/html/index.html](https://IRS48/qa/pipeline-20150810T213106/html/index.html)
- Home
  - 2 measurement sets included: # antennas, baselines, dates
- By topic: check amount of flagging
- By task
  1. Importdata
    - ASDM = raw data
    - Fields: different calibrators
    - Spw numbers: still includes lots of other data
    - Flux density: from catalog, only for double-check afterwards: possible to check ALMA calibrator catalog for time variation

# QA2

Bottom: Logs, QA (!), input, time

2. Flagdata: initial flagging, typically 10-20%
3. Spectral lines in flux calibrators flagged: write down
4. Selection reference antenna: useful for self-calibration
- 5-6. System temperature plots and flagging: edgechans, BP, spikes
7. WVR radiometer results used to correct phase delay (not in VLA/SMA!) => only applied if rms improves
8. Low-gain antennas (not working properly) are flagged
9. Calculates models for flux calibrator on each baseline
10. Shows bandpass solutions amp/phase for each spw: notice noisiness depends on spectral resolution. Check for weird things

# QA2

11. Shows flagging results of BP

12. Flux scaling: flux calibrators are resolved, so some baselines have zero flux => gain calibrators used to bootstrap: flux models (check if fluxes are consistent with 1)

13. Shows gain solutions phase/amp (use detailed plots)

- Note that phase goes 'around' at 180 degrees
- Phase jumps in the beginning expected: flux/BP other part of sky
- Amp solutions are not normalized
- Solutions should be more or less smooth: otherwise, consider flagging certain antennas/time ranges/spws in final dataset

14. Solutions applied: check outcome if anything weird

15-16. Automatic cleaning continuum of calibrators: small/point sources

# Product files

- Products delivered based on proposal: cleaned images in fits format
- Primary beam correction:
  - sensitivity decreases radially, so this is corrected in every image
  - huge noise in outer parts
  - X.flux.fits is the primary beam, X.pbcor.fits is the science data
- Fits headers sometimes difficulties: use `importfits()` to convert to image in that case
- Check `scriptForImaging.py` to see how products were obtained

# CASA viewer

- Open/manage/close: display multiple images
- Animator: scan through images/channels
- Raster vs contour: overlay
- Data display: change the appearance of the image: color range, axes, labels, colors, fonts, colorwedge, beam
- Spectral profile: select a region (circle/rectangle/free-form/dot) and change to Flux density units (default: mean)
- Statistics: select a region and double-click (result in Terminal) or check in panel Regions to get flux, max, min, rms, etc.
- Intensity/moment map: integrate over velocities
- Output: print to PDF/PS/graphics in working directory (can not be changed)

# CASA viewer tips

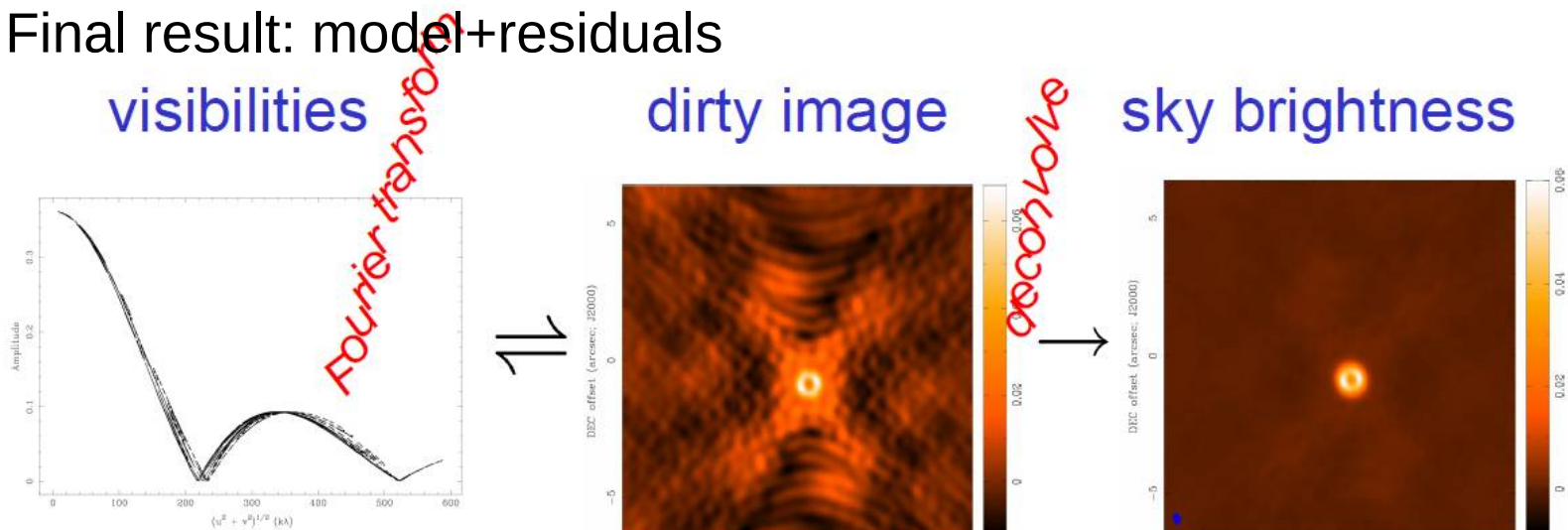
- Zooming in and out will sometimes mess up the overlay: rezoom to fix it
- Click inside selection on Esc to remove it
- Color scale is adjusted to min,max in image, which is dominated by the edges: zoom in and set the color scale smaller (or clean a smaller image!)
- Intensity map units are per channel: multiply with channel width in km/s
- Lines are always too thin, and color is foreground/background color, which gives different results in PDF/EPS/PNG
- The upper right corner displays the velocity for a cube, but also for continuum (a constant velocity or Stokes): remove it in axis label properties, movie axis label type
- Multiple panels possible with Panel setup
- Viewer can be partially scripted using `imview()`
- Save settings using Save panel state (Save image only for moments)
- Sometimes it just crashes...

# Data reduction

- If calibration is done, next step is imaging (cleaning): most of the work in ALMA data
- QA2 product files often leave room for improvement, also depending on your science
- Sometimes improvement by self-calibration
- Always script your steps!

# Cleaning

- Non-complete uv-sampling: FT leads to image with 'dirty beam' (point spread function)
- Dirty beam is PSF with sidelobes: depends on uv-coverage (**PDF-W**)
- Cleaning: removing sidelobes by iteration on dirty image (**PDF-C**):
  - Take clean components as 'model image' (bright region within mask)
  - Convolve with dirty beam
  - Subtract from data
  - Take clean components in residual and repeat until side lobes < noise levels
  - Final result: model+residuals



# Data reduction

- Measurement set
  - Split out science data => split()
  - Continuum reduction:
    - Check for bright lines in plotms and flag them (backup!)
      - Plotms(), flagmanager(), flagdata()
    - Average continuum for quick reduction
      - Split()
    - Cleaning in mode 'mfs'
  - Line reduction
    - Subtract continuum in uv-plane with fitorder=1
      - Plotms(), uvcontsub()
    - Cleaning in mode 'velocity' (or channel/frequency)

# Weighting

- Adjustment of dirty beam: weighting parameter

## (Super-uniform)

- Extreme uniform
- Careful with image quality

## Uniform

- Based on uv-coverage
- Smaller beam
- Lower PS sensitivity
- Problems sampling
- Suitable for high S/N

## Briggs Robust = +0.5 (-2 ... +2)

- Compromise between resolution and sensitivity
- 'Default'

## Natural

- Based on noise-level
- Larger beam
- Higher PS sensitivity
- Suitable for low S/N (weak lines!)

## (Tapering)

- Less weight to long baselines: extended structures
- Less relevant in disk studies until  $<0.1''$  data

```

# clean :: Invert and deconvolve images
vis = 'measurement.ms'
imagename = 'myimage.clean'
outlierfile = ''
field = ''
spw = ''
selectdata = True
  timerange = ''
  uvrange = ''
  antenna = ''
  scan = ''
  observation = ''
  intent = ''

mode = 'velocity'
  nchan = -1
  start = ''
  width = ''
  interpolation = 'linear'
  resmooth = False
  chaniter = False
  outframe = ''
  veltype = 'radio'

gridmode = ''
niter = 500
gain = 0.1
threshold = '0.0mJy'
psfmode = 'clark'
imagermode = 'csclean'
  cyclefactor = 1.5
  cyclespeedup = -1

multiscale = []
interactive = False
mask = []
imsize = [256, 256]
cell = ['1.0arcsec']
phasecenter = ''
restfreq = ''
stokes = 'I'
weighting = 'natural'
uvtaper = False
modelimage = ''
restoringbeam = []
pbcor = False
minpb = 0.2
uscratch = False
allowchunk = False
async = False

```

# Cleaning

Image parameters:

- Input/output file names

- imsize = FOV in pixels [400]

=> cover your science target

=> QA2 FOV is large enough for PB, but in disk studies this can be set much smaller, unless other targets nearby

- cell = pixel size ['0.025arcsec']

=> ~5 pixels per beam dimension

- weighting = 'natural'/'uniform'/'briggs'/etc.

- restoringbeam = fix beam size ['0.3','0.2','40deg']

=> Only if you want to compare with other data and only with realistic beam size

- pbcor = primary beam correction [True]

=> QA2 applies afterwards to measure noise but not necessary anymore

# Cleaning

```
# clean :: Invert and deconvolve images
vis                    = 'measurement.ms'
imagenam              = 'myimage.clean'
outlierfile           = ''
field                 = ''
spw                   = ''
selectdata            = True
  timerange           = ''
  uvrange             = ''
  antenna             = ''
  scan                = ''
  observation         = ''
  intent              = ''
mode                  = 'velocity'
  nchan               = -1
  start               = ''
  width               = ''
  interpolation        = 'linear'
  resmooth            = False
  chaniter            = False
  outframe            = ''
  veltype             = 'radio'

gridmode              = ''
niter                 = 500
gain                  = 0.1
threshold             = '0.0mJy'
psfmode               = 'clark'
imagermode            = 'csclean'
  cyclefactor         = 1.5
  cyclespeedup       = -1

multiscale            = []
interactive           = False
mask                  = []
imsize                = [256, 256]
cell                  = ['1.0arcsec']
phasecenter           = ''
restfreq              = ''
stokes                 = 'I'
weighting              = 'natural'
uvtaper               = False
modelimage            = ''
restoringbeam         = []
pbcor                 = False
minpb                 = 0.2
usescratch            = False
allowchunk            = False
async                 = False
```

Line cleaning parameters:

- mode = 'velocity'/'frequency'/'channel'

=> for continuum, mode='mfs'

- nchan = # channels [50]

=> line width!

- start = start-velocity ['-5 km/s']

- width = velocity bin ['0.5 km/s']

=> S/N!

- outframe = velocity frame ['LSRK']

=> important, especially with multiple spws!

- restfreq = frequency molecule

['345.79600GHz']

=> accurate! Check databases

- stokes = Stokes parameter ['I']

=> sometimes accidentally set blank

# Cleaning

```
# clean :: Invert and deconvolve images
vis = 'measurement.ms'
imagename = 'myimage.clean'
outlierfile = ''
field = ''
spw = ''
selectdata = True
  timerange = ''
  uvrange = ''
  antenna = ''
  scan = ''
  observation = ''
  intent = ''

mode = 'velocity'
  nchan = -1
  start = ''
  width = ''
  interpolation = 'linear'
  resmooth = False
  chaniter = False
  outframe = ''
  veltype = 'radio'

gridmode = ''
niter = 500
gain = 0.1
threshold = '0.0mJy'
psfmode = 'clark'
imagermode = 'csclean'
  cyclefactor = 1.5
  cyclespeedup = -1

multiscale = []
interactive = False
mask = []
imsize = [256, 256]
cell = ['1.0arcsec']
phasecenter = ''
restfreq = ''
stokes = 'I'
weighting = 'natural'
uvtaper = False
modelimage = ''
restoringbeam = []
pbcor = False
minpb = 0.2
usescratch = False
allowchunk = False
async = False
```

Cleaning parameters:

- niter = # iterations of cleaning [10000]  
=> set to large number, but interactive
- threshold = level where cleaning should stop  
=> typically 2-3 sigma, but interactive
- interactive = cleaning mode [True]  
=> unless mask and threshold well-defined, always clean interactively
- mask = mask used for cleaning  
=> can be taken from previous cleaning, using the filename myimage.clean.mask, otherwise created in the beginning

SHOW

# Cleaning tips

- Clean interactively to adjust the mask and check the noise levels
- Check your residual image
- Set the mask narrow (extend it with iterations): real emission outside mask will not disappear, but fake emission (side lobes) inside mask can be enlarged
- Line cleaning: especially for low signal, it is generally better to set the mask for each channel separately instead of All channels
- Careful with cleaning too deep to avoid fake structure
- Use existing masks as starting point for weak lines (but if nothing visible, do not push too hard)
- Make a dirty image by setting niter=0 to measure noise level
- Remove files instead of overwriting them: problematic
- Check rms/peak/flux levels of resulting image, but remember that small differences are irrelevant: flux calibration uncertainty is 10% (20% for B9)

# Output

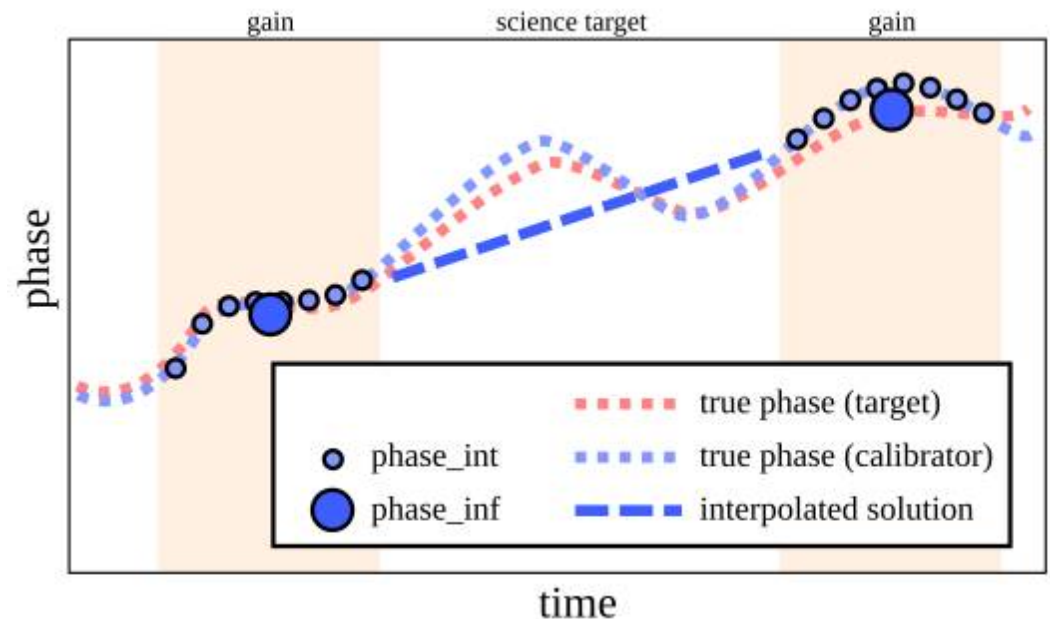
- `Exportfits()` => export your image to fits cube
- `Imstat()` => get statistics about your image
- `Immoments()`  
=> create moment maps of your image:
  - zeroth moment (intensity)
  - First moment (velocity map)
  - Second moment (velocity dispersion)
  - Several other options (3-11)
- Use clipping for moment maps: >3 sigma pixels only

# Serendipitous lines

- Bandwidth is huge: lots of other possible lines within spw
- Useful website: Splatalogue  
<http://www.cv.nrao.edu/php/splat/> (Advanced)
- Take frequency range, a reasonable limit for  $E_U$  (K) (<300 K) and check the  $A_{ij}$  Einstein coeff.
- Two approaches:
  - Clean at specific rest frequency in narrow velocity range
  - Clean entire spectral window blindly (use mask of known line) and scroll through it in the image plane:
    - Keplerian motion visible?
    - Any sign of signal in the spectrum?
- Careful cleaning of low S/N data
- If no detection: upper limits also provide useful information!

# Self-calibration

- Using science target as gain calibrator
- Why? Phase/amp vary during science target observations
- Only possible for high S/N science target (continuum): trying to solve signal per antenna



- Solutions of continuum can be applied to line data, but improvement in lines is not always visible if  $\text{rms noise} > \text{phase noise}$

# Self-calibration

- Basic steps
  - Backup flagging state before you start (flagmanager)
  - Clean the continuum image, image will be put in model column used for the self-cal
    - Use max a few 100 iterations to make sure only real structure is included
  - Check S/N using imstat()
  - Find phase solutions using the longest time interval in gaincal()
    - During this task baselines may be flagged due to insufficient S/N
    - Save the caltable as pcal1, pcal2, etc.
    - Use a reference antenna close to the center (check the QA2 or plotants()) that is not flagged
    - Set the minsnr to at least 3
    - Note that sometimes data gets flagged due to S/N

# Self-calibration

- Basic steps (continued)
  - Check solutions with `plotcal()` and flag if necessary (sometimes plot multiple time ranges)
  - Apply solutions with `applycal()`
  - Repeat clean and following steps with a shorter time interval
    - Time intervals example: 'inf', '120s', '60s', '30s', 'int': decrease slowly ((integration time from listobs))
  - When S/N does not change any more, find amplitude solutions with time interval at least 2 times larger than shortest phase interval and apply
  - Apply solutions to line data

# Not really discussed

- Large scale structure
  - Image quality
  - Absorption
  - Getting rid of cloud (embedded disks)
  - Tapering
- Mosaicking
- Polarization
- Data weights and rms
- Atmospheric limitations
- Other interferometers: non-CASA scripts (SMA, CARMA)
- Etc. etc. etc.

=> Allegro!

Tuesday

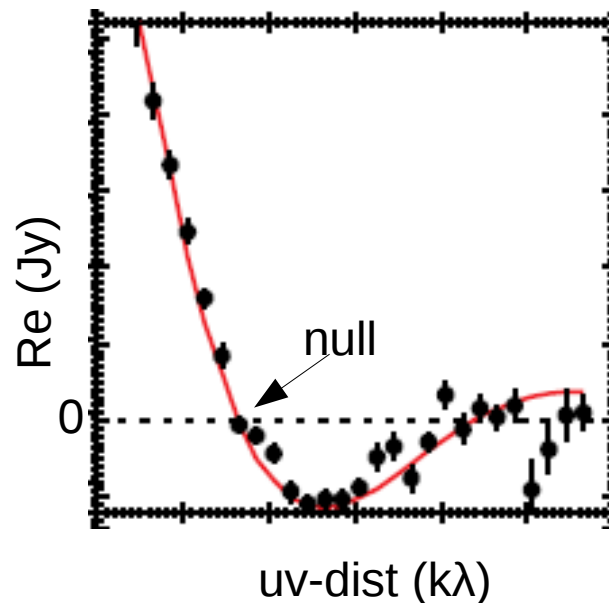
# ALMA/CASA

- Visibility curves and analysis
- Simulating
  - Simobserve
  - Mapping on observed visibilities
- Analysis of ALMA data
- ALMA archive
- Proposals
  - A successful proposal
  - The OT

ALMA/CASA: analyzing and simulating data

# Visibility curves and analysis

- Due to incomplete uv-coverage, fitting of interferometry data often performed in the uv-plane: in the visibilities
- Visibilities also give information on 'smoothness' of radial profile, small scale structures (long baselines) and asymmetries (imaginary)
- For axisymmetric disks,  $\text{Imag} = 0$ , so only Real.

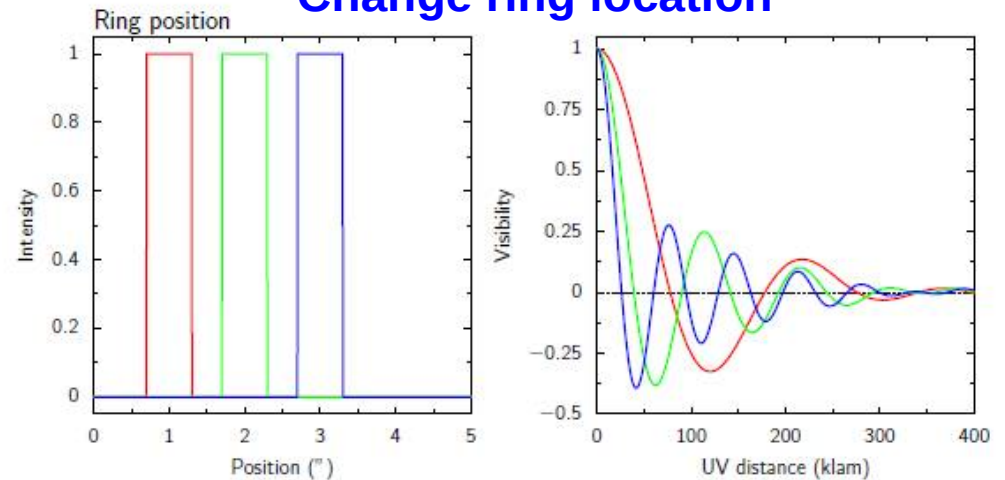


Detection of transition disk cavities used to be limited by lack of long baselines!

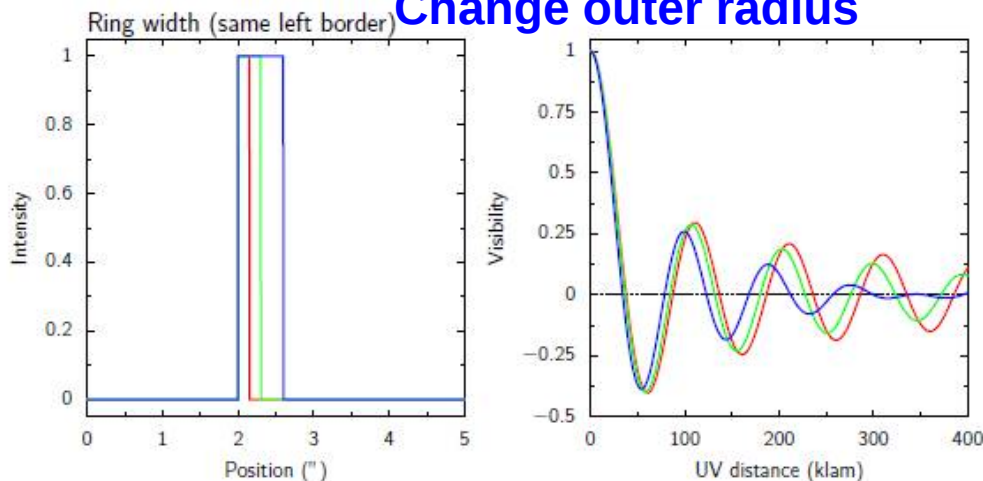
# Visibility curves and analysis

- Location of the null tells you about the ring location, not (only) the cavity size
- Amplitude of ringing tells you about the ring width

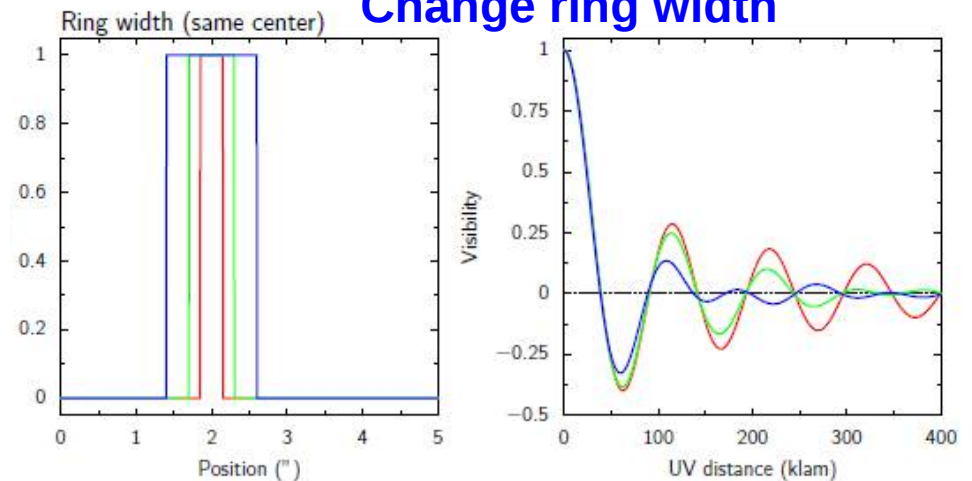
**Change ring location**



**Change outer radius**



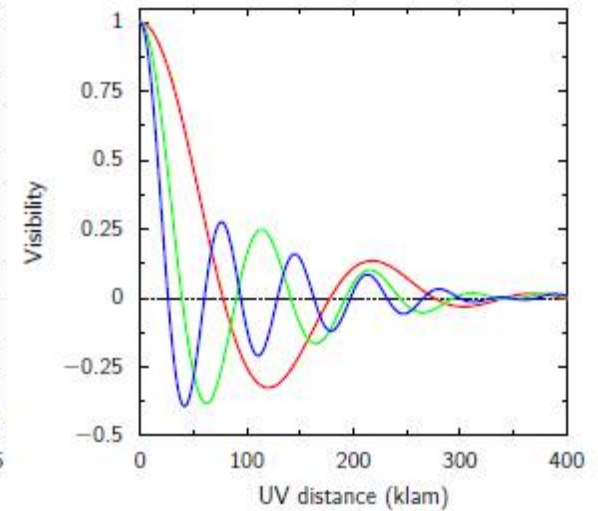
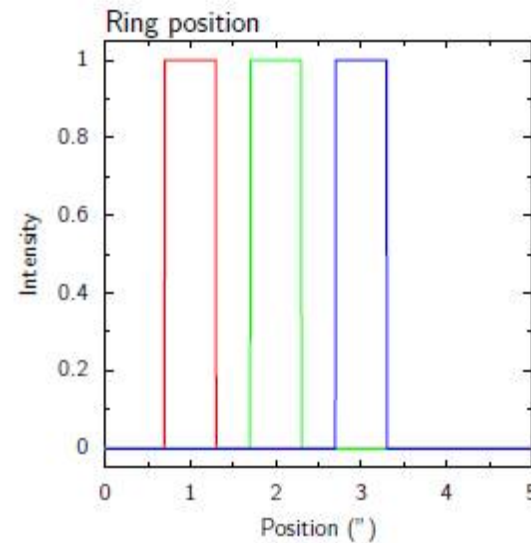
**Change ring width**



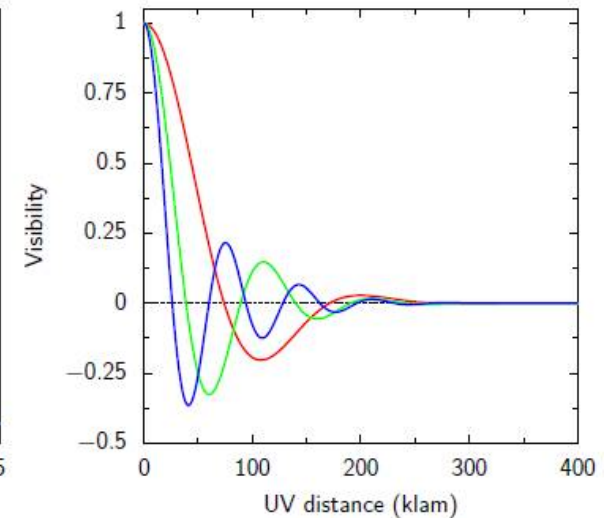
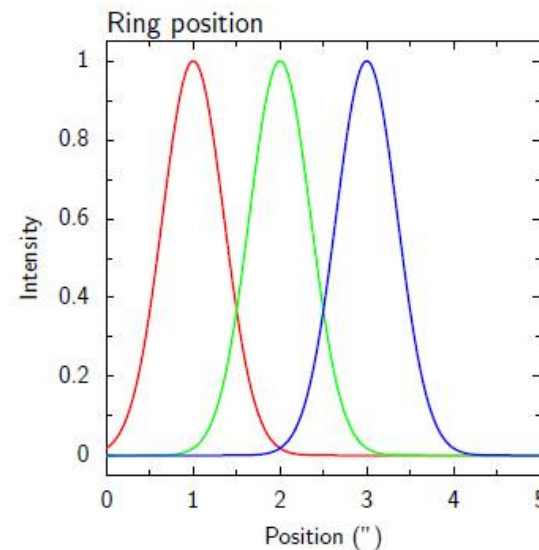
# Visibility curves and analysis

- Amount of 'ringing' tells you about the presence of sharp edges

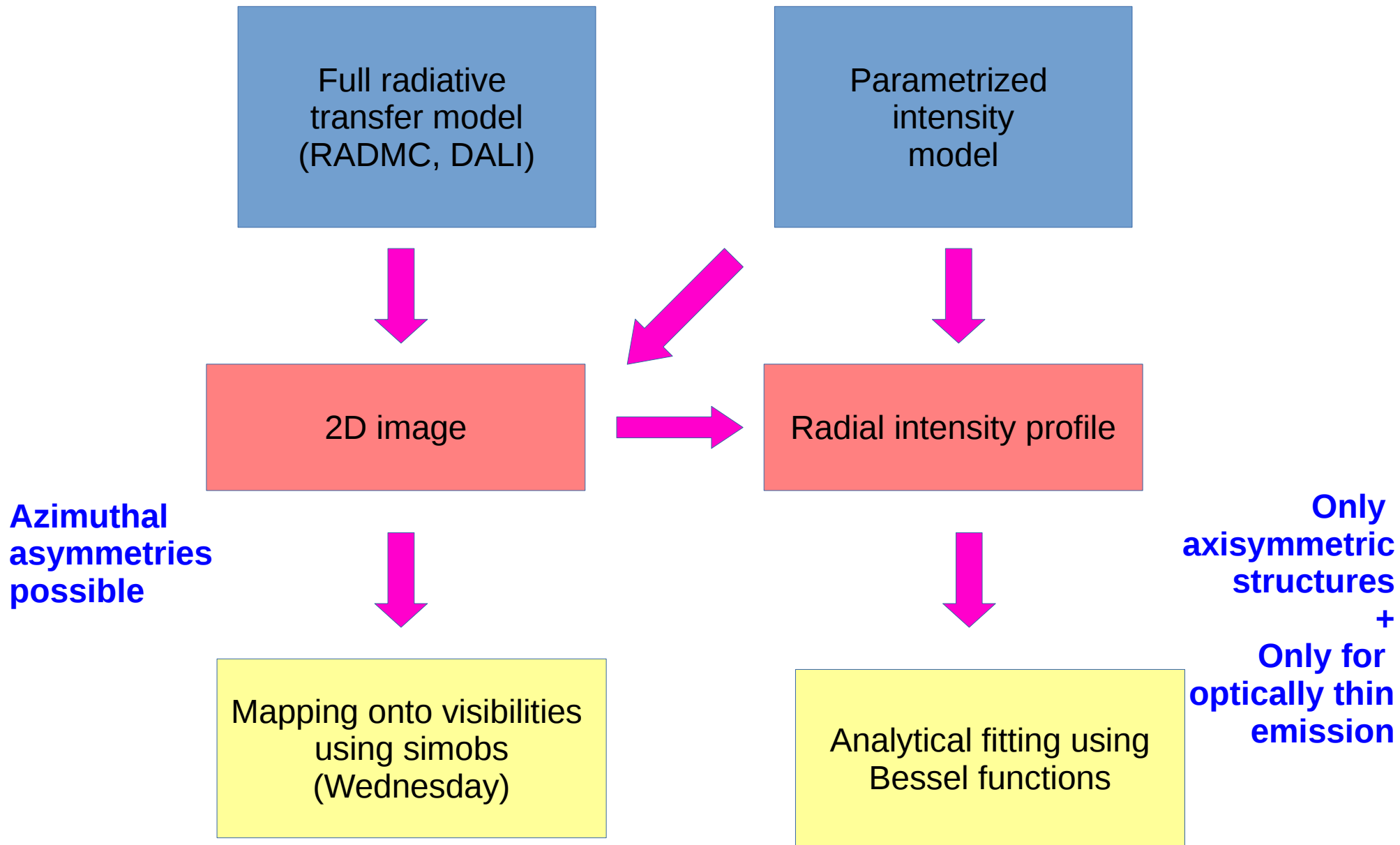
**Sharp**



**Smooth**



# Visibility curves and analysis

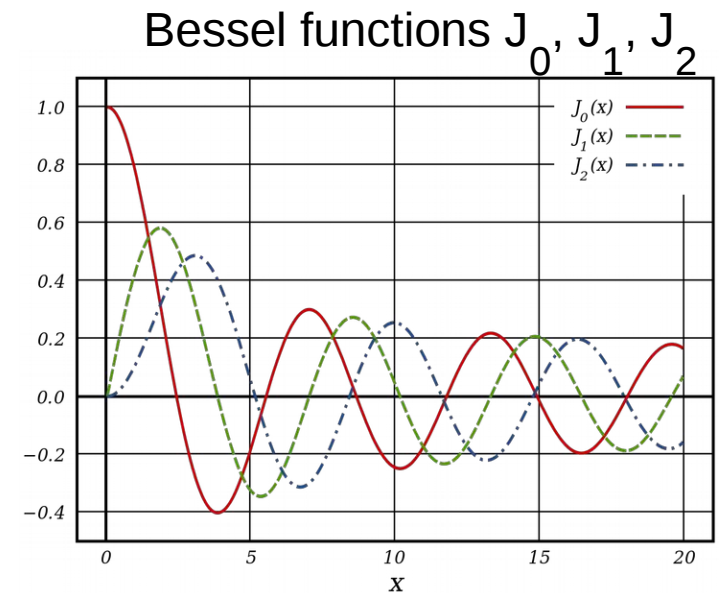


# Visibility curves and analysis

- Analytical Real: 
$$V_{\text{Re}}(r_{uv}) = 2\pi \int_0^\infty I(\theta) J_0(2\pi r_{uv} \theta) \theta d\theta$$
- Zeroth order Bessel function  $J_0$ : 
$$J_0(x) = \frac{1}{2\pi} \int_0^{2\pi} \exp(-ix \cos \Theta) d\Theta$$
- Intensity function ring example (Walsh et al. 2014)

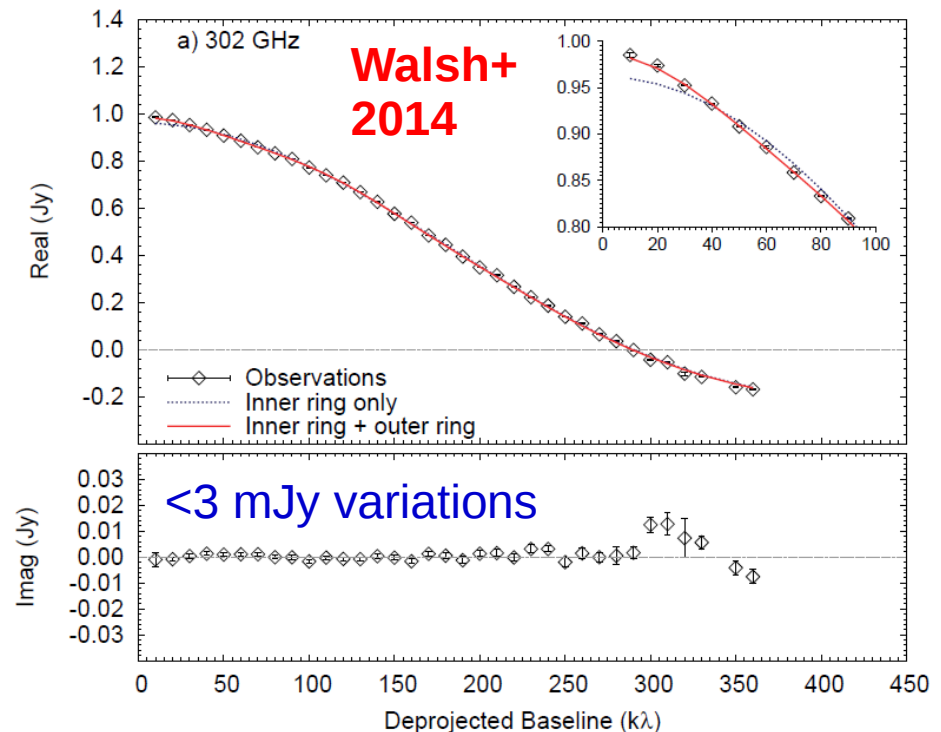
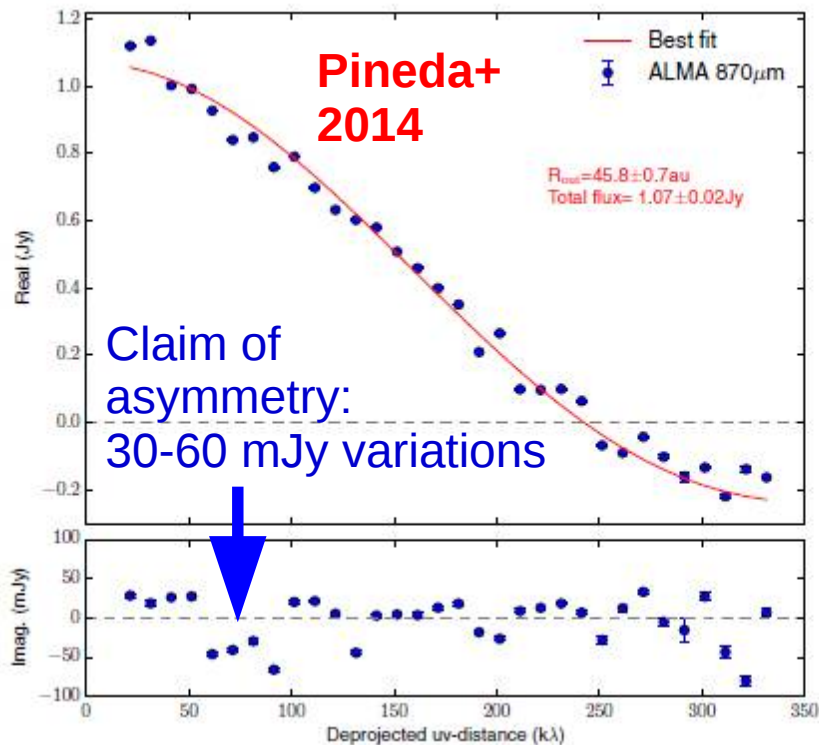
$$I(\theta) = C \exp\left(\frac{-(\theta - \theta_{\text{peak}})^2}{2\theta_{\text{width}}^2}\right)$$

- Fast MCMC fitting possible
- Imaginary part: analytical prescription derived in Isella et al. 2013 due to small asymmetry in LkHalpha-330



# Visibility curves and analysis

- Real and imaginary visibilities are measured with respect to the phase center: center of disk
- Center = center of gravity (Keplerian motion): taking center of continuum can be wrong if asymmetric (IRS 48!)
- Setting phasecenter correctly is very important: a wrong PC results in non-zero imaginary signal, which indicates asymmetries => Walsh vs Pineda on HD100546 data



# Visibility curves and analysis

- Visibility curves so far: *deprojected* uv-distances (corrected for PA and inclination)
- Formulas deprojection: 
$$\begin{cases} u_{\Theta} = u \cos \Theta + v \sin \Theta \\ v_{\Theta} = -u \sin \Theta + v \cos \Theta \end{cases}$$
$$r_{uv\Theta i} = \sqrt{u_{\Theta}^2 + v_{\Theta}^2 \cos(i)^2}$$
- Careful with the signs:  
SMA uses opposite definition of direction of u,v

# Visibility curves and analysis

- Step-by-step
  - Set phase center correctly using `fixvis()`
  - Extract visibility data using script, `execfile(extract.py)`
  - Correct  $u,v$  for PA and inclination
  - Bin  $u,v$  data points and calculate errors
  - Calculate visibilities based on an intensity profile using Bessel functions and compare
  - In fitting procedure: also fit PC, PA and incl.

# Visibilities

- Alternative: `uvmodelfit()` task
- Very simple approach to fit point source, 2D gaussian or 2D disk in uv-plane
- Can be useful as zeroth order estimate of parameters or to check how well a target is resolved

# Simulating

- Due to uv-coverage, convolving with the beam may not always be enough to compare models with observations: potential missing spatial scales
- Use simobserve (for proposal predictions with/without noise) or simobs\_custom (for comparing model with dataset) to map model onto visibilities and subtract and/or image it

# Simobserve

- Main parameters
  - Project name (always new name or delete old directory)
  - Model fits (proper header!)
  - Antenna list (can be generated from dataset)
  - Total time
  - Thermal noise (default)
- Frequency observations determined by input model

```
# simobserve :: visibility simulation task
project = 'simc6' #
skymodel = 'image_001.cont.fits' #
  inbright = '' #
  indirection = '' #
  incell = '' #
  incenter = '' #
  inwidth = '' #
  complist = '' #
  setpointings = True #
  integration = '10s' #
  direction = '' #
  mapsize = ['', ''] #
  maptype = 'ALMA' #
  pointingspacing = '' #
  obsmode = 'int' #
  antennalist = 'alma.cycle3.6.cfg' #
  refdate = '2014/05/21' #
  hourangle = 'transit' #
  totaltime = '7200s' #
  caldirection = '' #
  calflux = '1Jy' #
  thermalnoise = 'tsys-atm' #
  user_pvw = 0.5 #
  t_ground = 269.0 #
  seed = 11111 #
leakage = 0.0 #
graphics = 'both' #
verbose = False #
overwrite = True #
async = False #
```

# Simobserve

- Fits file:
  - Intensity in Jy/pixel (see slides Tuesday)
  - Header must contain following information:
    - RESTFREQ (in Hz, e.g. 345.579599E9)
    - RADESYS ('FK5')
    - SPECSYS ('LSRK')
    - CRVAL1 (RA in decimal degrees)
    - CRVAL2 (DEC in decimal degrees)
    - Third axis contains velocity information/units for line data and frequency information/units for continuum (bandwidth irrelevant)

# Simobserve

- Scaling

- Change frequency, coordinates, brightness (distance!) of model
- Can also be done on model itself a priori

- Speed up

- Increase sampling time (not recommended)

- For comparison with dataset:

- Total time of obs
- Reference date of obs
- Central hour angle of obs (e.g. -3h)

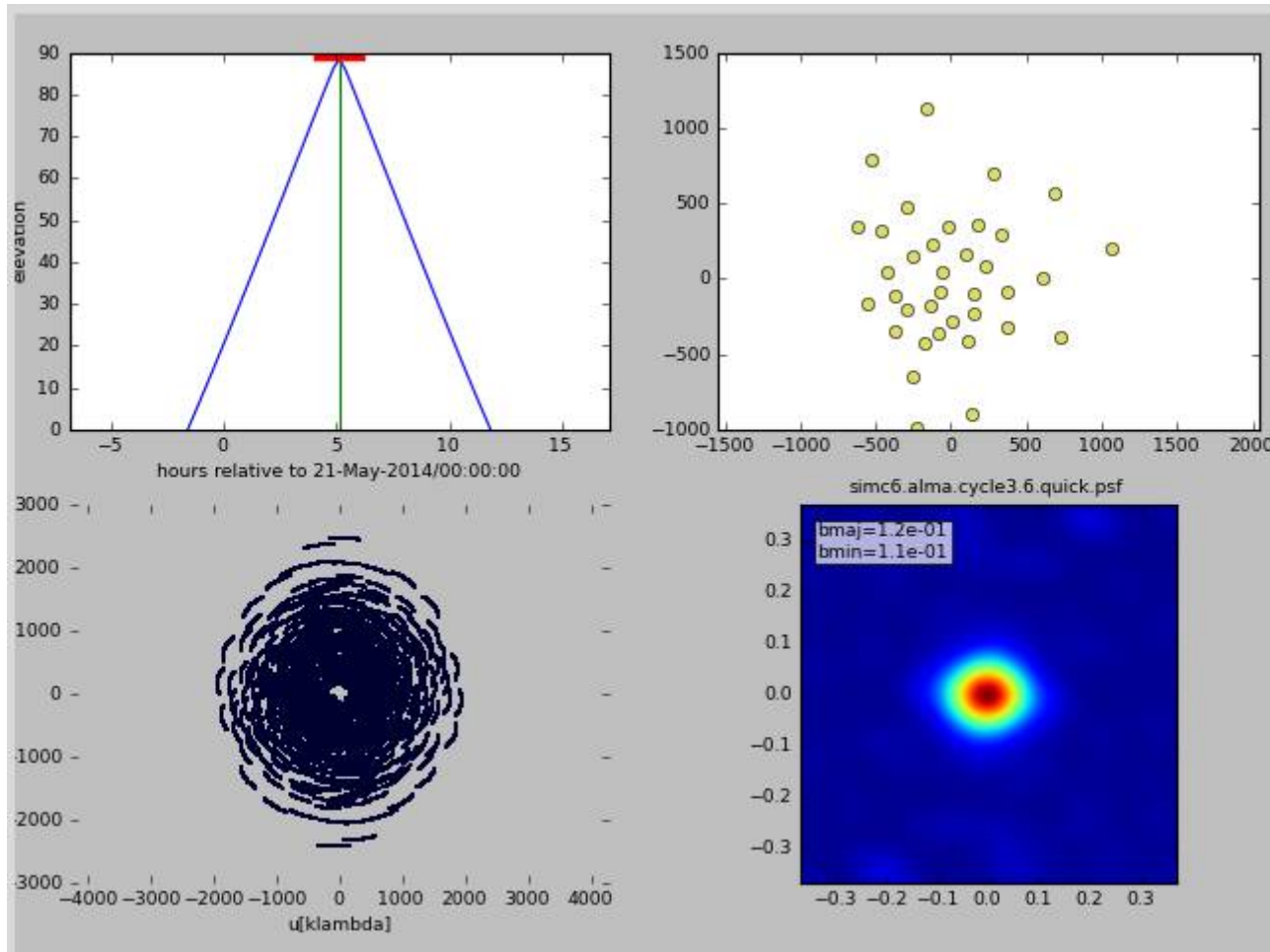
```
# simobserve :: visibility simulation task
project = 'simc6' #
skymodel = 'image_001_cont.fits' #
  inbright = '' #
  indirection = '' #
  incell = '' #
  incenter = '' #
  inwidth = '' #
complist = '' #
setpointings = True #
  integration = '10s' #
  direction = '' #
  mapsize = ['', ''] #
  maptype = 'ALMA' #
  pointingspacing = '' #
obsmode = 'int' #
  antennalist = 'alma_cycle3.6.cfg' #
  refdate = '2014/05/21' #
  hourangle = 'transit' #
  totaltime = '7200s' #
  caldirection = '' #
  calflux = '1Jy' #
thermalnoise = 'tsys-atm' #
  user_pwv = 0.5 #
  t_ground = 269.0 #
  seed = 11111 #
leakage = 0.0 #
graphics = 'both' #
verbose = False #
overwrite = True #
async = False #
```

# Simulating

- Due to uv-coverage, convolving with the beam may not always be enough to compare models with observations: potential missing spatial scales
- Use `simobserve` (for proposal predictions with/without noise) or `simobs_custom` (for comparing model with dataset) to map model onto visibilities and subtract and/or image it

# Simobserve

Hour angle observation



Antenna positions

uv coverage

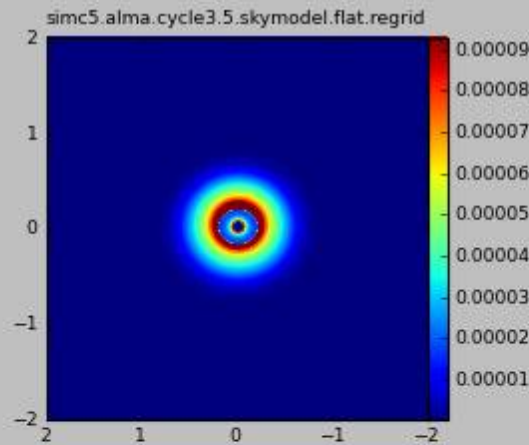
PSF

# Simanalyze

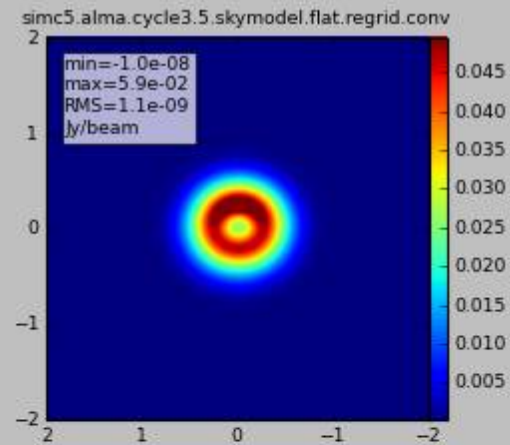
- Simobserve creates project.ms and project.noisy.ms (if thermalnoise set)
- Next step: imaging
- Simanalyze() does an automatic clean of project.(noisy).ms with natural weighting
- Either adjust parameters in simanalyze() or just run clean in the normal way
- Advantage of simanalyze(): some parameters already set correctly (cell size) and comparison with beam convolved image is made

# Simanalyze

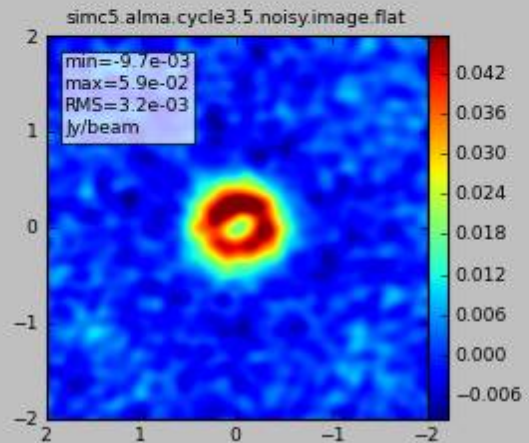
Input model



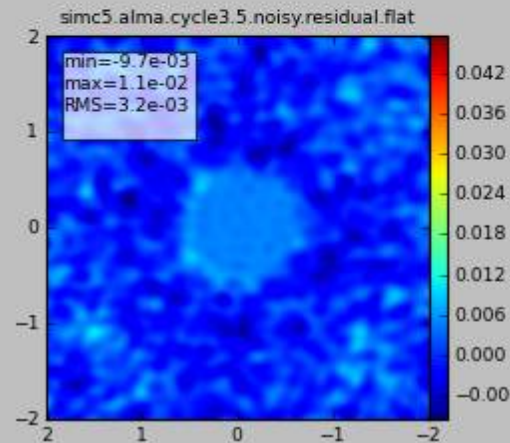
Input model convolved with psf



Simulated image:  
Do not trust rms!  
Use ALMA  
sensitivity  
calculator



Difference  
between  
convolved and  
simulated image



And lines work exactly the same way...

# Simobs\_custom

- Tool developed by Attila Juhasz (ex Allegro, now Cambridge)
- Allows you to map a model on the measured visibilities of a measurement set
- It copies the measurement set to working directory, so can be slow (and takes lot of disk space!)
- Two versions: v0.2, v0.3 (including subtraction)
- Already installed on helada (test: >tasklist())
- Installation
  - Put task\_simobs\_custom.py & simobs\_custom.xml in directory X
  - Go to X, start CASA and run >os.system(('buildmytasks'))
  - Open ~/.casa/init.py and add >execfile('X/mytasks.py')

# Simobs\_custom

- Parameters:
  - Project
  - input vis (will be copied to project dir)
  - skymodel (header!)
  - Mode='cont'/'line'
  - subtract\_model=T
- In project/project.ms: original data in 'data', model data in 'model' and subtraction in 'corrected\_data'

```
# simobs_custom :: Simulates observation u
vis = 'HD135344B.cont.ms' #
project = 'simobsc' #
skymodel = 'modell.cont.fits' #
  inbright = '' #
  indirection = '' #
  incell = '' #
  incenter = '' #
  inwidth = '' #
spw = '' #
mode = 'cont' #
  user_powex = 0.0 #
subtract_model = True #
EXPECTED_DATA column)
thermalnoise = '' #
phasenoise = '' #
overwrite = False #
async = False #
```

# Simobs\_custom

- Steps for continuum simulation
  - `simobs_custom()` with continuum measurement set
  - Split out model (or residual) data column with `split()`
  - Clean the model/residual measurement set using the same mask/threshold as the observational data

# Simobs\_custom

- Steps for line simulation
  - Split out measurement set line part that covers (a bit more than) the model line data ( $dv = dv * c / v_{rest}$ )
  - `simobs_custom()` with `mode='line'`
  - Split out model (or residual) data column with `split()`
  - Subtract the continuum (`uvcontsub`)
  - Clean the model/residual measurement set using the same mask/threshold as the observational data

# Analysis of ALMA data

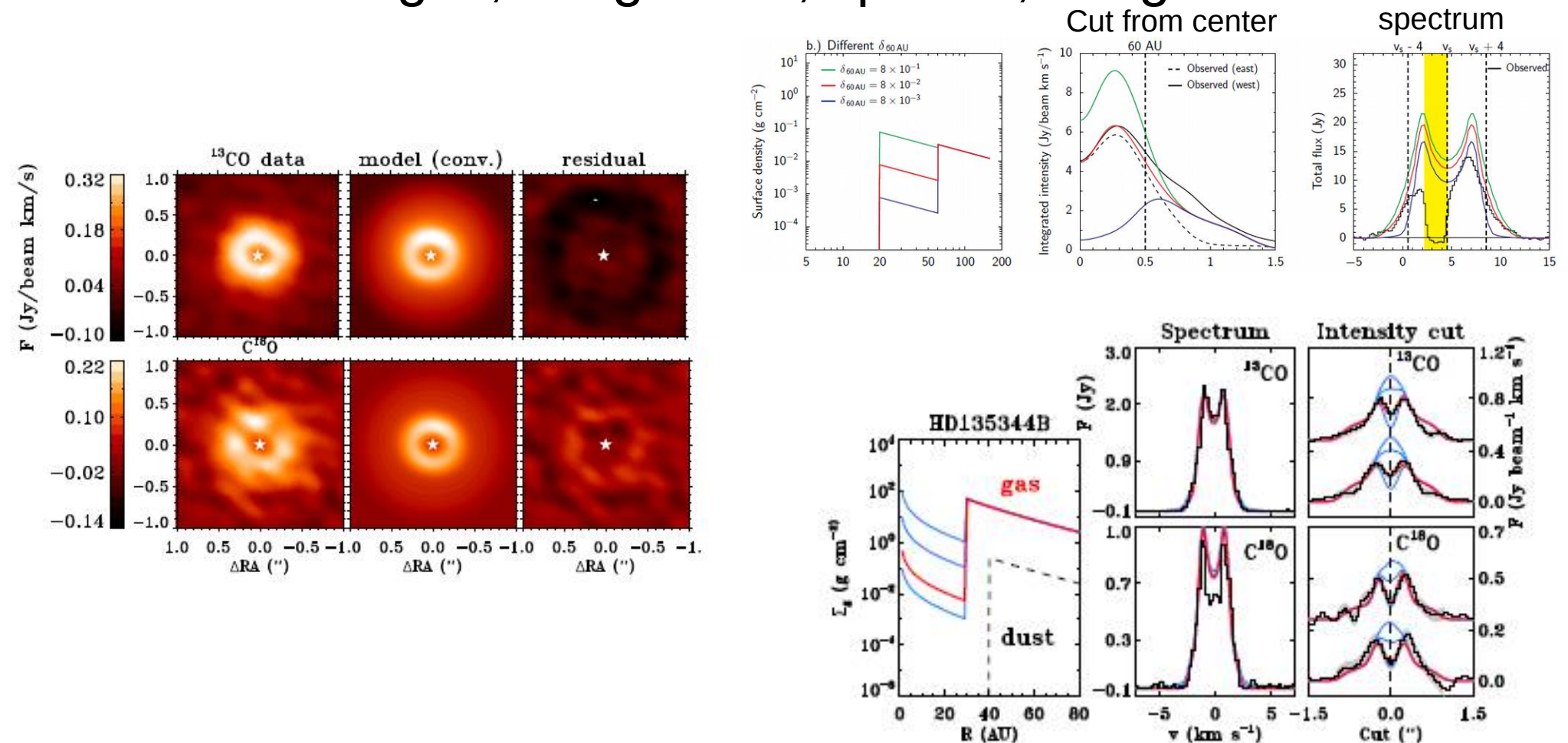
- Check calibration/QA2
- Perform self-calibration
- Clean lines you are interested in
- Determine your science question: what do I want to analyze/what models need to be run?
  - Simple intensity models?
  - Full physical-chemical models? (DALI)
  - Excitation models? (LIME)
    - Simple constant abundance model?
    - Parametrized abundance model?

# Analysis of ALMA data

- Running DALI/LIME models: keep your science question in mind
  - Constrain your dust structure (SED/mm image) before your gas structure
  - Check output models: do they make sense?
  - Check trends in models: what happens with parameter changes?
  - Keep an eye on resolution/size
  - Do NOT fit blindly

# Analysis of ALMA data

- Comparison (after convolution):  
=> images, image cuts, spectra, integrated flux



# Analysis of ALMA data

- Comparison with data
  - Convolve with beam  
(remember disk PA and inclination, beam size and orientation)
  - Simulate with simobserve
  - Simulate with simobs\_custom
- Simulating takes a lot of time: usually enough to check if you get a similar answer for one model as with convolution
- Remember that both flux calibration uncertainty and uncertainties in modeling itself (including the physics!) introduce uncertainties that are much higher than the rms levels!

How do we get ALMA data?

ALMA archive

ALMA proposals

# ALMA archive

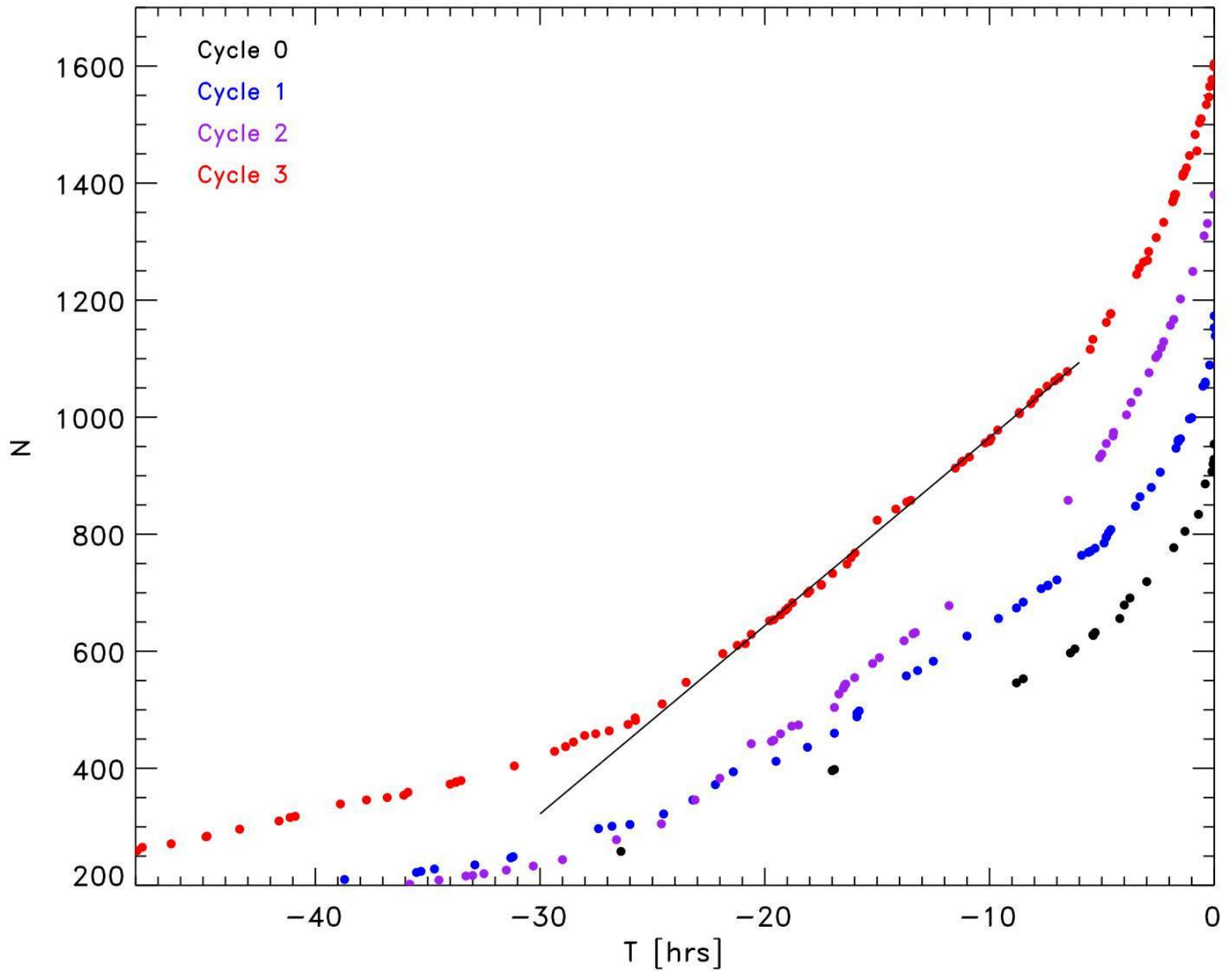
- Science verification data (immediately public) and PI data (public after one year): lots that can be done with the archival data!
- For PI data it is polite and useful (but not mandatory) to check with the PI to ask about the status of their publications, rather than 'stealing' it; may prevent problems.
- Delivery list of each Cycle (including proposal abstract) available online: useful to see what is coming
- Data science archive: query on coordinates, settings, PI, etc.
- <https://almascience.eso.org/> (create account!)

# ALMA proposals

- Proposal deadline once every 1-1.5 years
- Oversubscription huge! Factor 5-10
- PI = principal investigator, co-I = coinvestigator
- Four regions + Open skies (PI from region!)
  - Europe (EU) 33.75% (ESO)
  - North-America (NA) 33.75% (NRAO)
  - East-Asia (EA) 22.5% (NAOJ)
  - Chile 10%
  - Open Skies <5%
- Grading proposals (in combination with ranking per region):
  - A = very likely (and transferred)
  - B = likely (not transferred)
  - C = filler (possible)
  - U = rejected

# ALMA proposals

- Jes Jorgensen plot



# ALMA proposals

- Background information each Cycle
  - Technical handbook
  - Proposer's guide
  - Sensitivity calculator
  - CASA: simulator!
  - Many workshops, tutorials online
  - Support groups/ARC nodes:  
Netherlands has Allegro in Leiden

# ALMA proposals

- Content proposal
  - Abstract & title
  - Scientific justification (4 pages including figures/refs)
  - In Observing Tool
    - Science goal: one combination of sources (within 15 degrees) and spectral settings
    - Sources
    - Spectral settings
    - Angular resolution (min and max angular scale: determines configurations for you automatically!)
    - Technical justification (justify choices above and connect with your scientific goals)
- Open OT (installed on chaxa): `>ALMA-OT-Cycle3.sh &`

# ALMA proposals

- Start early (~4 weeks) with discussion/draft
- Run simulations (script)
- State very clearly what we learn: different scenarios, different outcomes
- Limit scientific goals: even if you get them, no need to mention all of them: confusing
- A figure can say more than a thousand words: schematics, cartoons, plots, indication missing data
- Organize a mini-TAC: read each other's proposals

# ALMA proposals

- Considerations
  - Configuration: resolution and maximum angular scale
  - Timeline (sometimes only night-time observing!)
  - Observing frequency/weather
  - Amount of observing time (calibration!)
  - Spectral settings: getting as much info as possible (shift spectral window to cover multiple lines, bandwidth/resolution)
  - Data rate
  - High-risk: long baselines/high frequency
  - Available data in archive
  - Collaborations, other teams

Questions?