NAASC

An Introduction to CASA and Simulations with Simdata

NRAO

May 25, 2011

General Description

- CASA is the post-processing package for ALMA (and EVLA) both interferometric and single dish
- The ALMA pipeline is is being built from CASA *toolkit*
- Toolkit packaged into most commonly used *tasks* for users
- Designed with parallelization in mind
- CASA in active community use since October 2007
- Release 3.2.0 available since May 18 2011
  - Most recent linux flavors, Mac OSX for Leopard & Snow Leopard
  - Available to anyone from CASA homepage:

http://casa.nrao.edu

CASA Documentation and Web Resources

NAASC

- Casaguides are fully annotated scripts including screen shots
- There is also a complete CASA user manual
- Currently there are millimeter guides for EVLA, CARMA, and SMA that serve as important learning tools for future ALMA data
- CASAguides for ALMA Science Verification data on near horizon

http://casaguides.nrao.edu

NRAO

Inside CASA

> Tasklist
Will show an organized list of all currently available tasks

Opportunities for full ALMA related CASA tutorials will be available after the proposal call deadline (June 30)

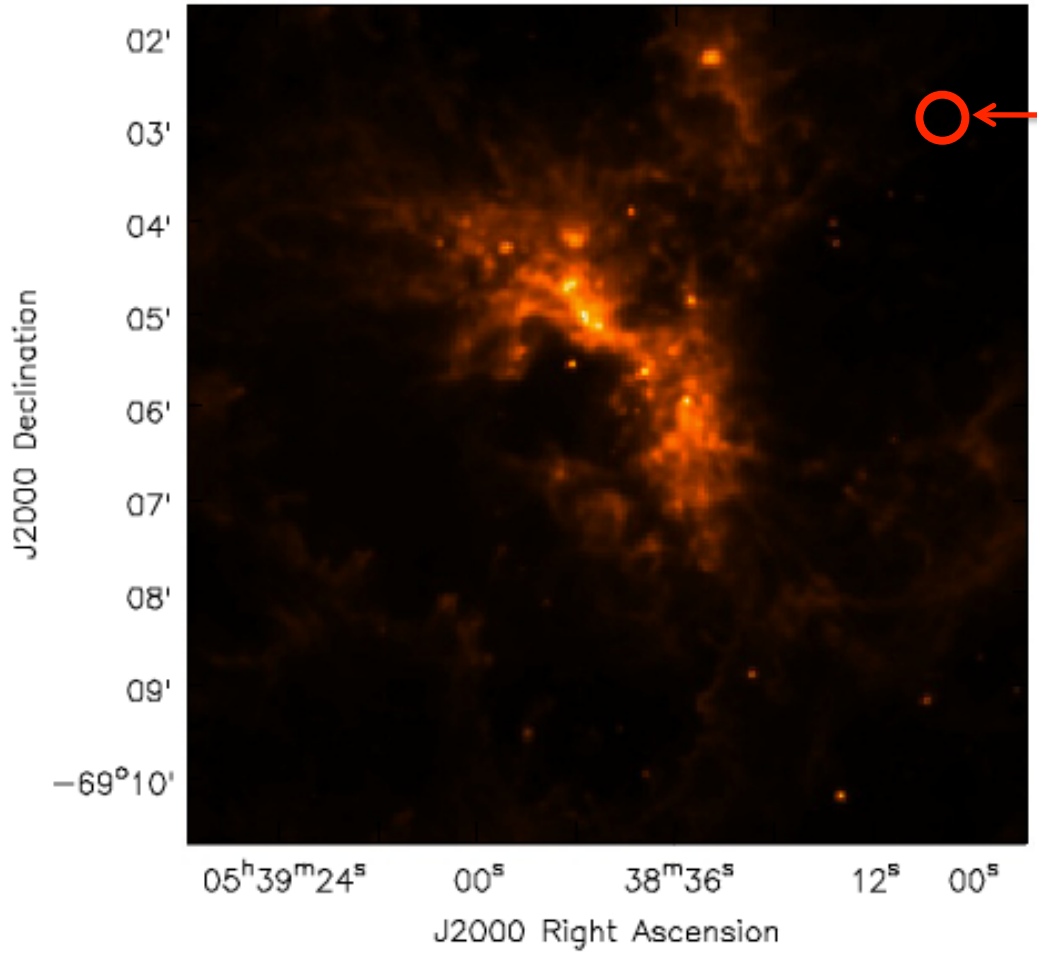
What is Simdata Good For?

Take a model image and find out how it would look if observed with ALMA

- Number of antennas
- Antenna configuration
- Length of observation
- Noise* ⟵
  - Thermal Noise
  - Phase Noise

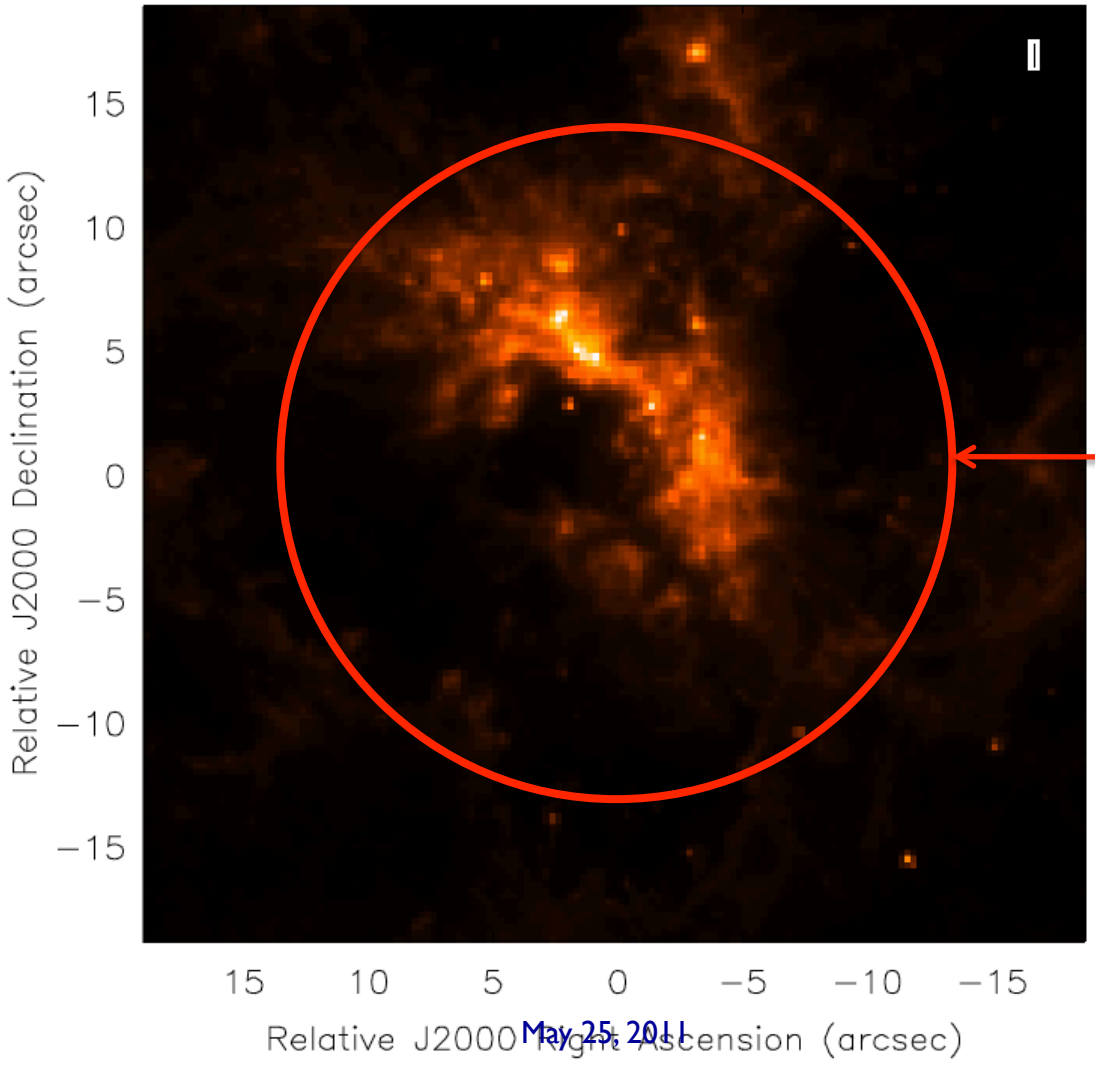
Model Image

30Dor Spitzer IRAC 8um image from SAGE



Field of view at 230 GHz

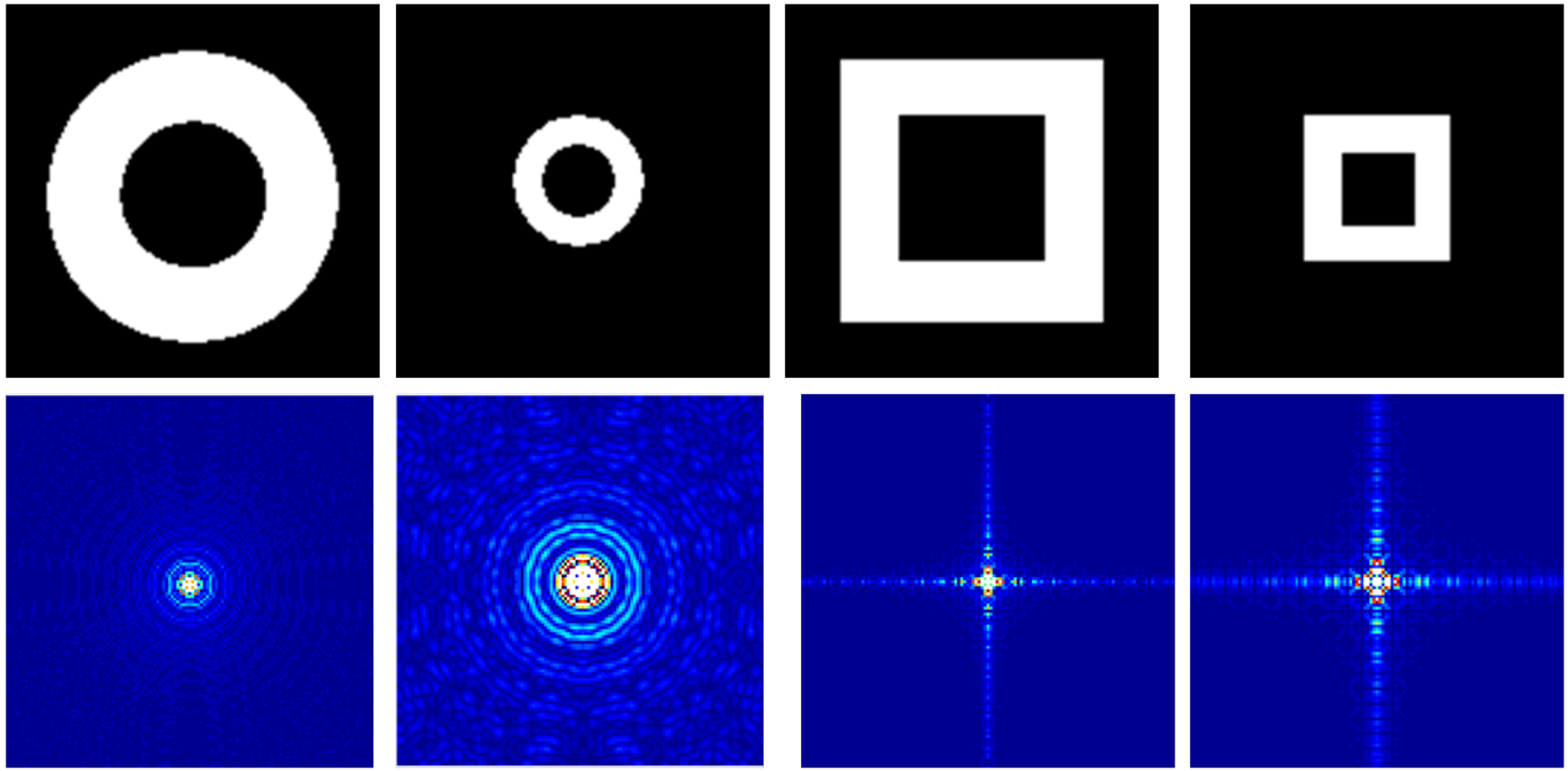
Model Image
Resized and now at 230 GHz



Now ~15 times more distant!

Field of view at 230 GHz

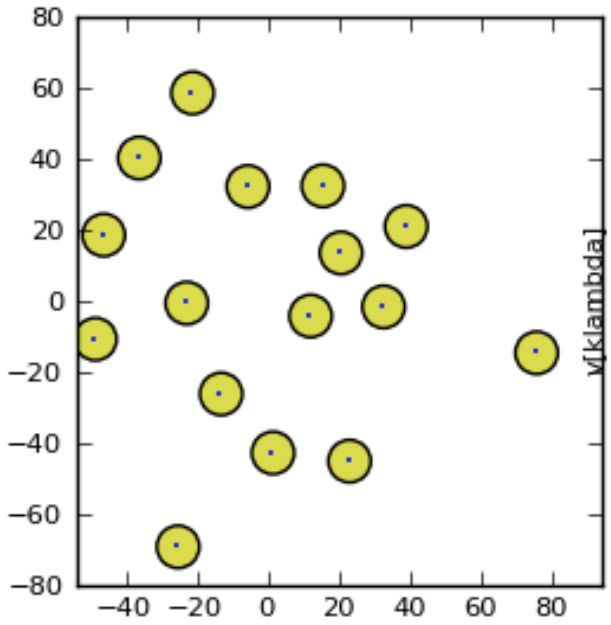
Fourier Transforms of Images



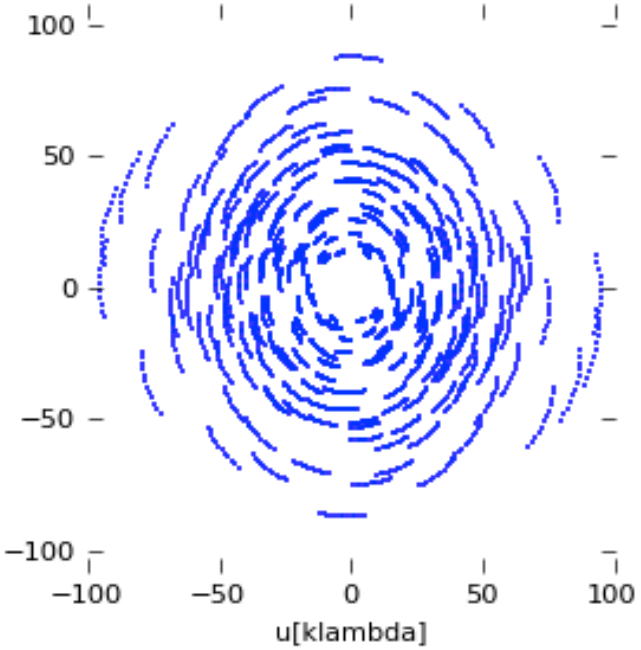
From http://carmilumban-ap186.blogspot.com

Scales Measured in Compact Early Science

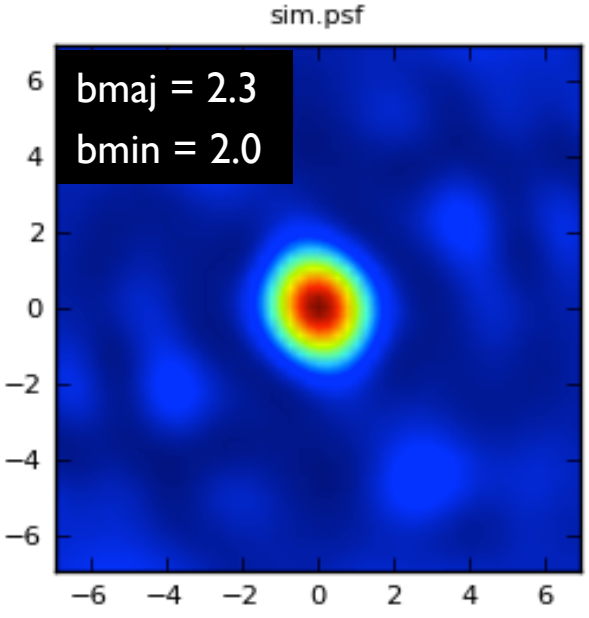
Antenna Placement



uv-coverage



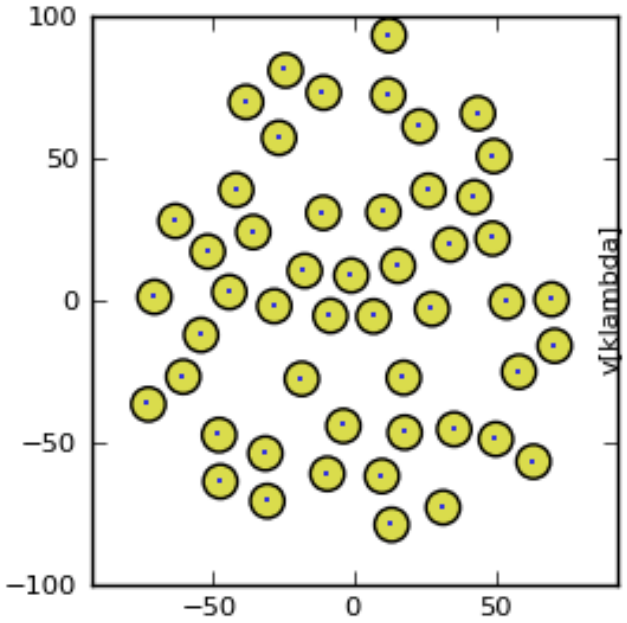
Point Spread Function



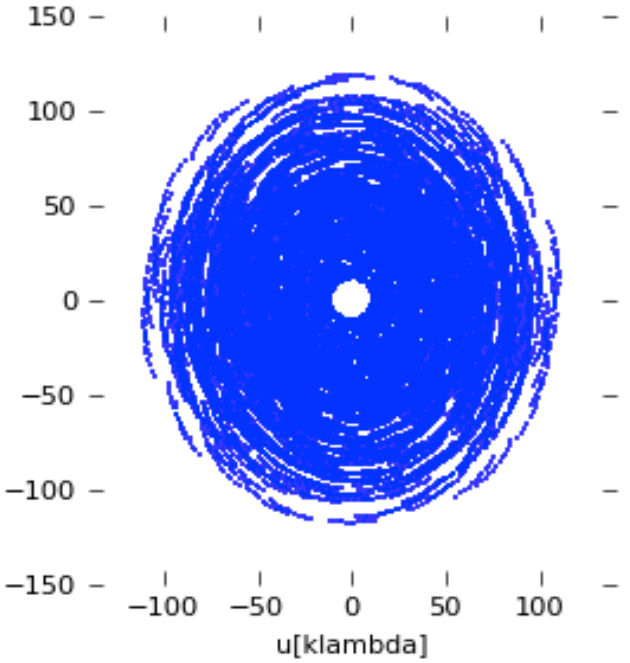
2 hour observation in Compact Configuration

Full Science 12m Array - Compact

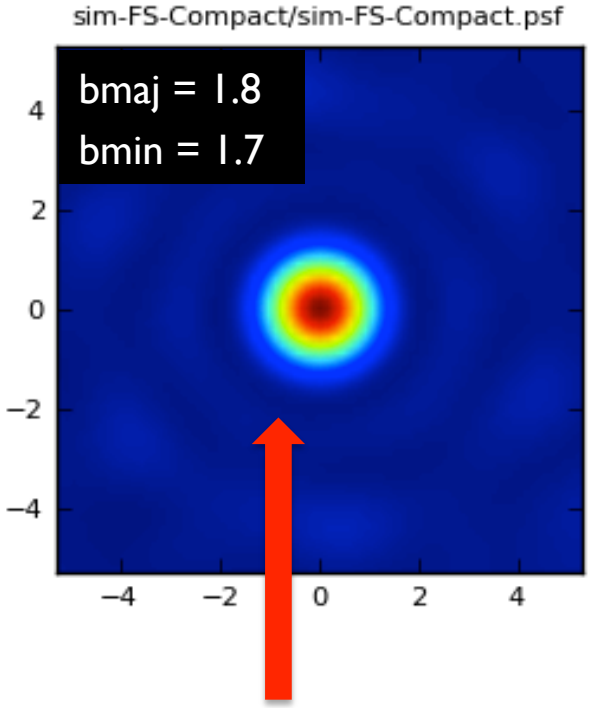
Antenna Placement



uv-coverage



synthesized beam

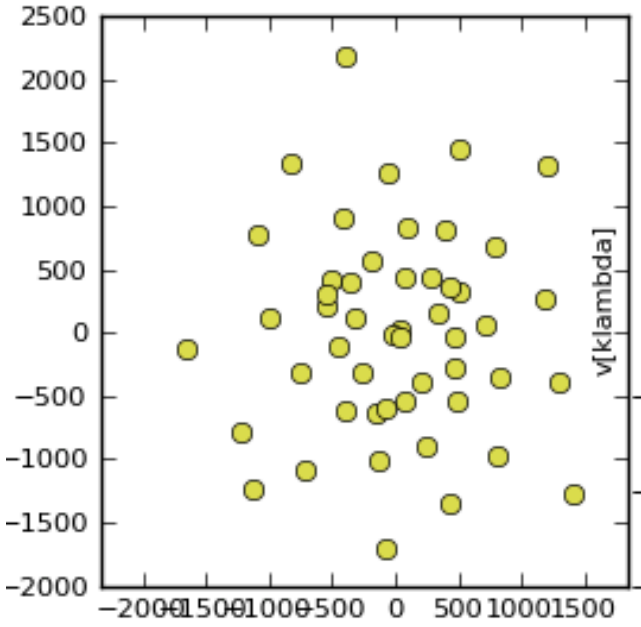


2 hour observation

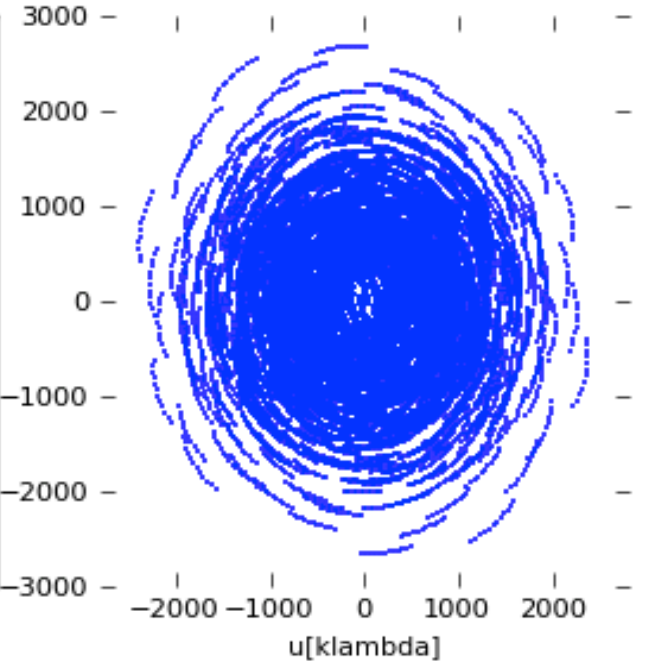
Note lower sidelobes

Full Science 12m Array - Extended

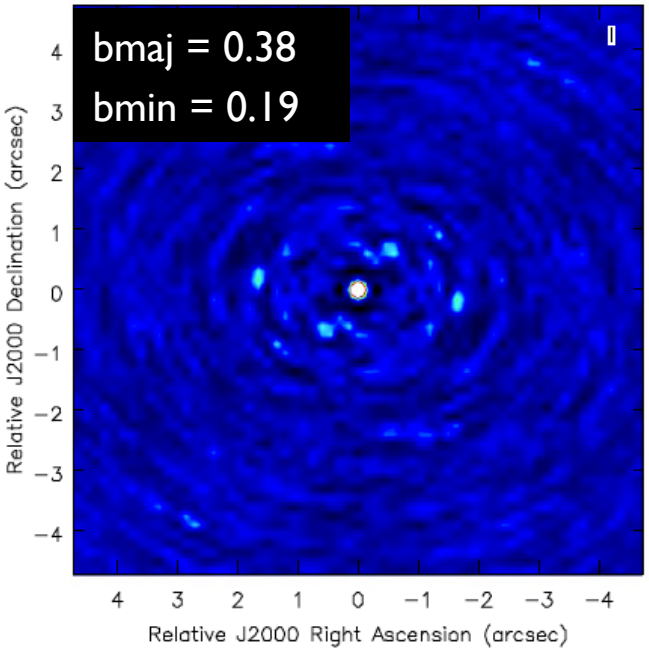
Antenna Placement



uv-coverage



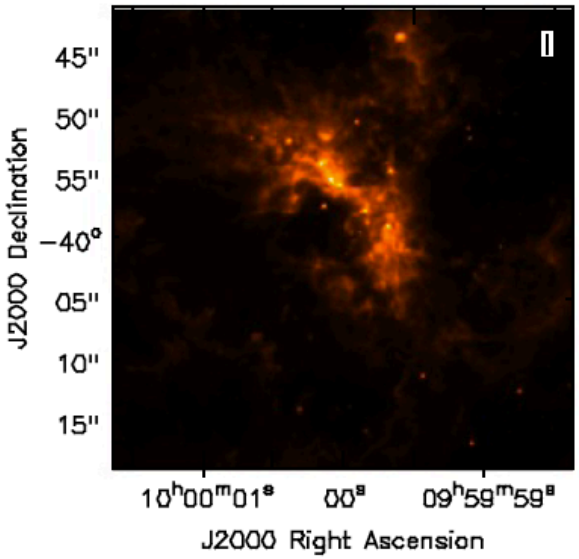
synthesized beam



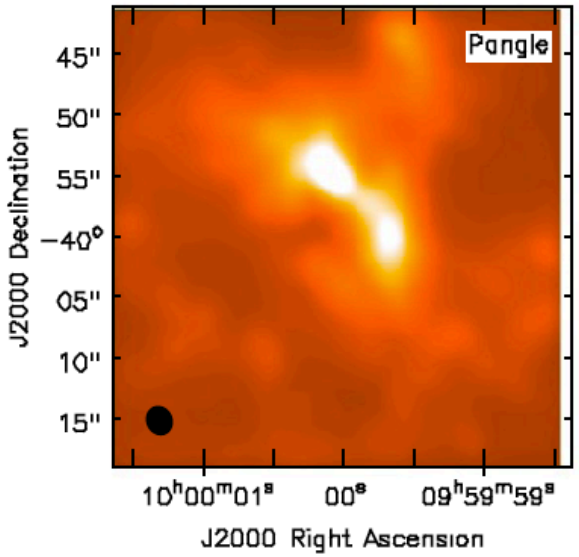
2 hour observation

Model: Early Science Compact Configuration

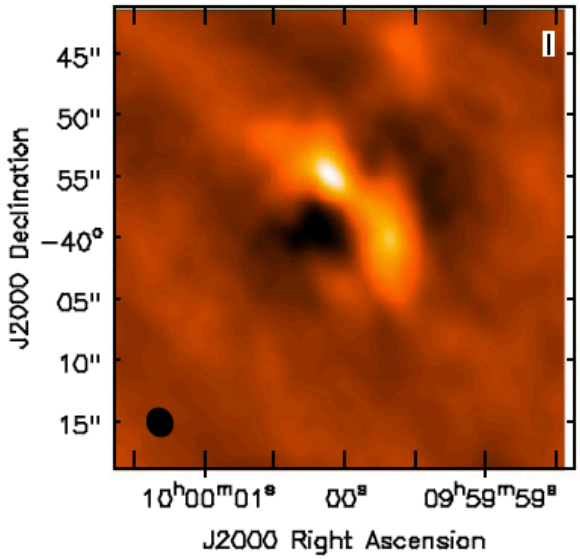
Model Image



Convolved Model



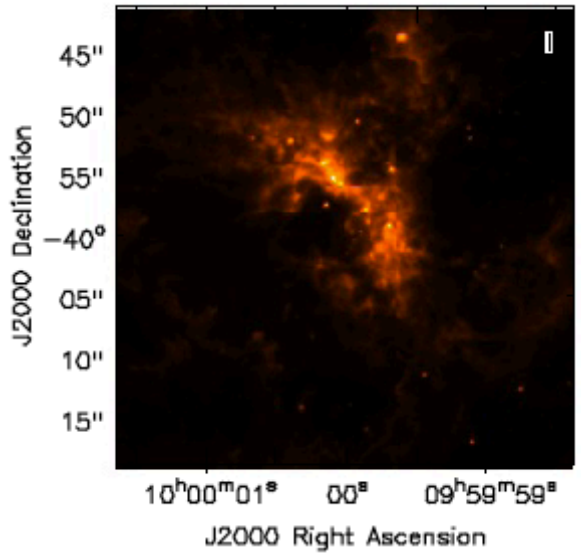
"Observed" Image



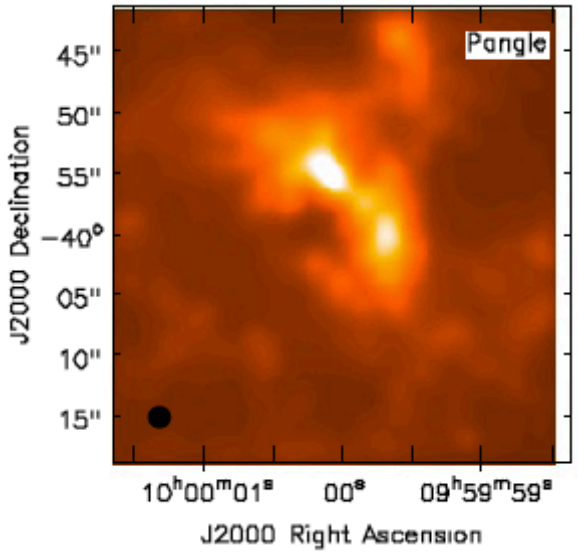
2 hour observation

Model: Full Science Main Array - Compact

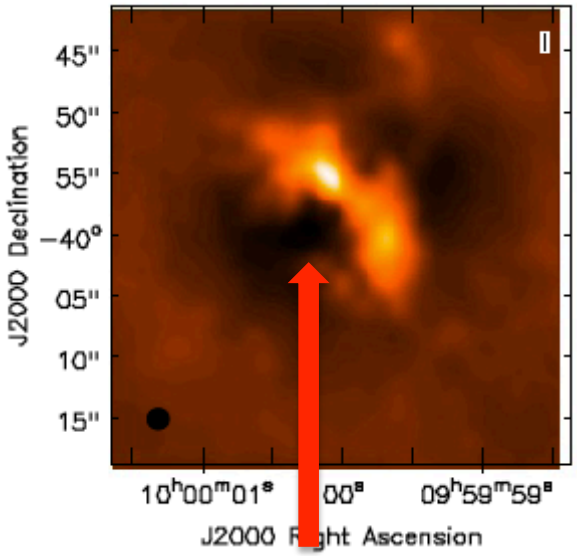
Model Image



Convolved Model



"Observed" Image

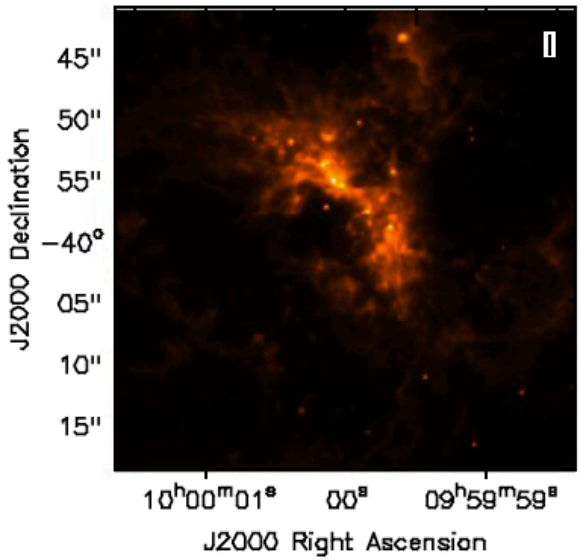


2 hour observation

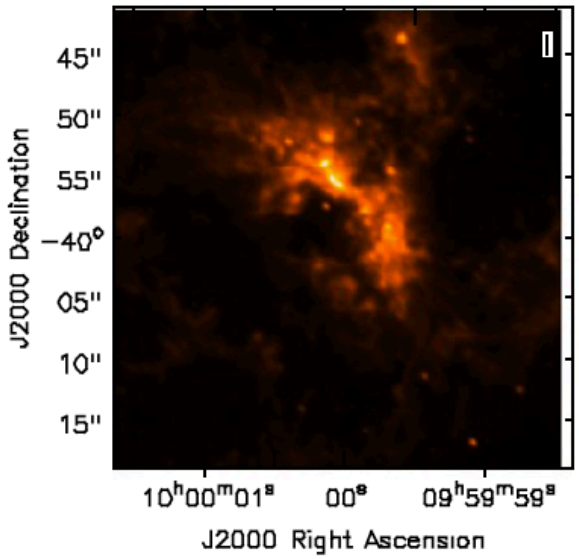
Large scale emission: Observe with ACA and possibly TPA

Model: Full Science Main Array - Extended

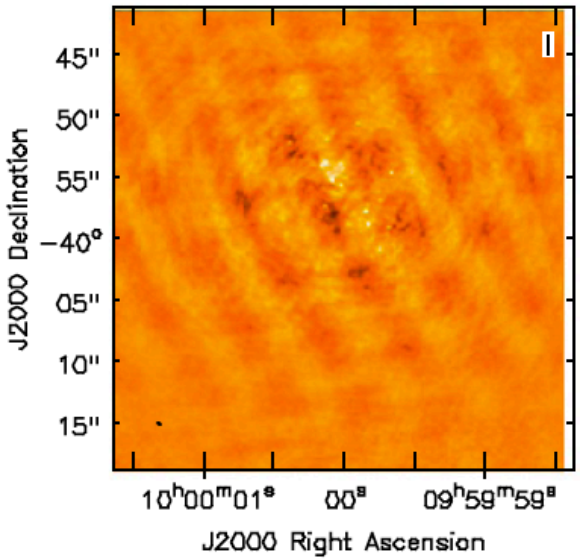
Model Image



Convolved Model



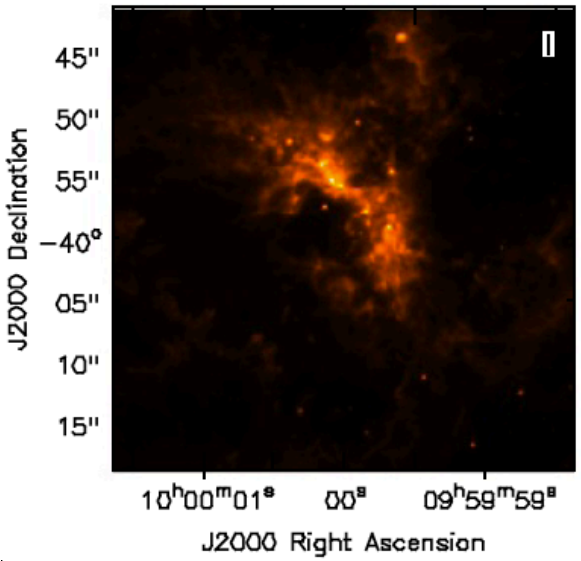
"Observed" Image



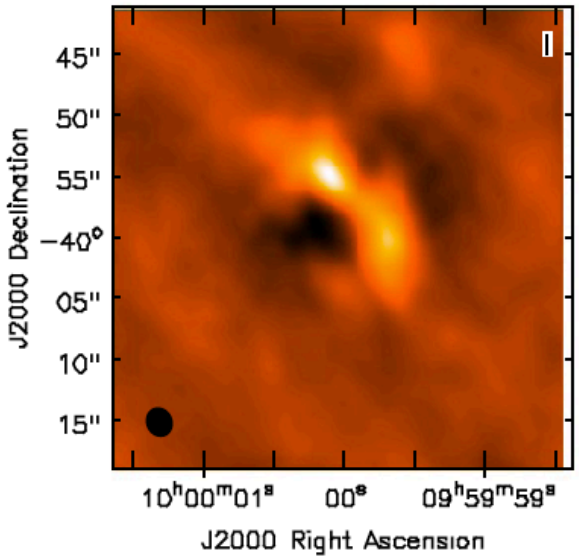
2 hour observation

Model: Early Science Compact & Extended

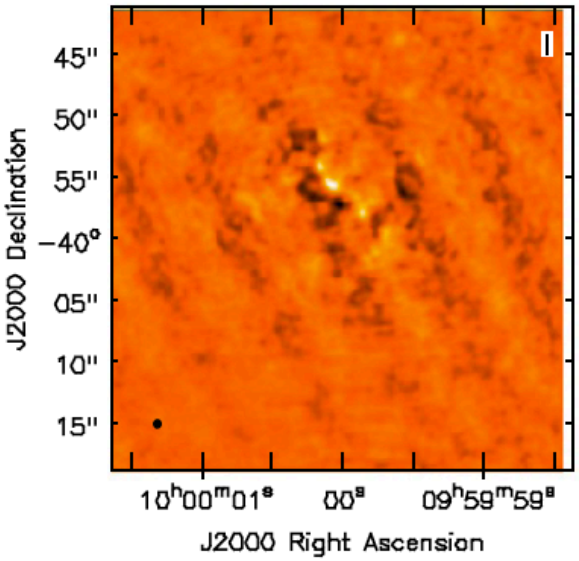
Model Image



Compact Array



Extended Array



2 hour observation

Basic Simdata Workflow

- Start CASA
- Input image file into Simdata
- Predict what ALMA would see using Simdata
- Add noise (optional)
- Compare ALMA image with input image

CASA Basics

- To start > casapy
- To look at task inputs
  > inp clean
- Grey: expandable parameters
- Red: invalid value
- Blue: accepted value but not the default
- To see full help > help clean
- Reset defaults > default clean
- To run > go

```
CASA <71>: inp
---------> inp()
#  clean :: Invert and deconvolve images with selected algorithm
vis                 =         ''            #  Name of input visibility file
imagename           =         ''            #  Pre-name of output images
outlierfile         =         ''            #  Text file with image names, sizes, ce
field               =         ''            #  Field Name or id
spw                 =         ''            #  Spectral windows e.g. '0~3', '' is al
selectdata          =  'chicken'            #  Other data selection parameters
mode                =      'mfs'            #  Spectral gridding type (mfs, channel,
     nterms         =          1            #  Number of terms used to model the sky
                                            #   development)
     reffreq        =         ''            #  Reference frequency for MFS (relevant
                                            #   frequency

gridmode            =         ''            #  Gridding kernel for FFT-based transfo
niter               =        500            #  Maximum number of iterations
gain                =        0.1            #  Loop gain for cleaning
threshold           =    '0.0mJy'           #  Flux level to stop cleaning, must inc
psfmode             =     'clark'           #  Method of PSF calculation to use duri
imagermode          =    'mosaic'           #  Options: 'csclean' or 'mosaic', '', u
     mosweight      =      False            #  Individually weight the fields of the
     ftmachine      =    'mosaic'           #  Gridding method for the image
     scaletype      =     'SAULT'           #  Controls scaling of pixels in the ima
                                            #   Options: 'PBCOR','SAULT'
     cyclefactor    =        1.5            #  change depth in between of  csclean c
     cyclespeedup   =         -1            #  Cycle threshold doubles in this numbe
     flatnoise      =       True            #  Controls whether searching for clean
                                            #   (True) or in an optimal signal-to-no

multiscale          =         []            #  Deconvolution scales (pixels); [] = s
interactive         =      False            #  Use interactive clean (with GUI viewe
mask                =         []            #  Cleanbox(es), mask image(s), region(s
imsize              = [256, 256]            #  x and y image size in pixels. Single
cell                = ['1.0arcsec']         #  x and y cell size(s). Default unit ar
phasecenter         =         ''            #  Image center: direction or field inde
restfreq            =         ''            #  Rest frequency to assign to image (se
stokes              =        'I'            #  Stokes params to image (eg I,IV,IQ,IQ
weighting           =  'natural'            #  Weighting of uv (natural, uniform, br
uvtaper             =      False            #  Apply additional uv tapering of visib
modelimage          =         ''            #  Name of model image(s) to initialize
restoringbeam       =       ['']            #  Output Gaussian restoring beam for CL
pbcor               =      False            #  Output primary beam-corrected image
minpb               =        0.2            #  Minimum PB level to use
```

Or use script mode - only need to set non-default values:
> clean(vis='visname',field='2',mode='channel')

Basic Simdata Inputs

- Image of target
- Observing time
- Antenna configuration
- Optional noise parameters

Basic Simdata Inputs

```
CASA <4>: inp simdata
--------> inp(simdata)
#  simdata :: mosaic simulation task:
project             =       'sim'          #  root for output file names
modifymodel         =       False          #  modify model image
     skymodel       = '$project.skymodel' #  model image to observe or modify

setpointings        =       False
     ptgfile        = '$project.ptg.txt' #  list of pointing positions

predict             =        True          #  calculate visibilites using ptgfile
     complist       =          ''          #  optional componentlist to observe
                                           #   with skymodel
     compwidth      =      '2GHz'          #  optional bandwidth if simulating from
                                           #   components only
     antennalist    = 'alma.out10.cfg'     #  antenna position file or "" for no
                                           #   interferometric MS
     refdate        = '2012/05/21/22:05:00' #  time/date of observation *see
                                           #   help
     totaltime      =    '7200s'           #  total time of observation
```

Current values of parameters

Parameters that can be changed

Explanation of what the parameters are

Basic Simdata Inputs

```
CASA <4>: inp simdata
--------> inp(simdata)
#  simdata :: mosaic simulation task:
project             =       'sim'          #  root for output file names
modifymodel         =       False          #  modify model image
     skymodel       = '$project.skymodel' #  model image to observe or modify

setpointings        =       False
     ptgfile        = '$project.ptg.txt' #  list of pointing positions

predict             =        True          #  calculate visibilites using ptgfile
     complist       =          ''          #  optional componentlist to observe
                                           #   with skymodel
     compwidth      =      '2GHz'          #  optional bandwidth if simulating from
                                           #   components only
     antennalist    = 'alma.out10.cfg'    #  antenna position file or "" for no
                                           #   interferometric MS
     refdate        = '2012/05/21/22:05:00' #  time/date of observation *see
                                           #   help
     totaltime      =    '7200s'          #  total time of observation
```

Model Input FITS File

Header must include:

- Coordinates
- Brightness units
- Observing frequency
- Pixel Scale (angular and spectral)
- Polarization (if needed)
- OR: Modify FITS image within Simdata

Change your input file: modifymodel

```
modifymodel         =       True        #  modify model image
     skymodel       = '$project.skymodel' #  model image to observe or modify
     inbright       =         ''        #  scale surface brightness of brightest pixel e.g
                                        #   ""
     indirection    =         ''        #  "J2000 19h00m00 -40d00m00" or ""
     incell         =         ''        #  cell/pixel size e.g. "0.1arcsec" or ""
     incenter       =         ''        #  frequency of center channel e.g. "89GHz" or ""
     inwidth        =         ''        #  channel width e.g. "10MHz" or ""
```

- modifymodel = True
- skymodel = “30dor.fits”
- inbright = ”0.06mJy/pixel”
- indirection = "J2000 10h00m00 -40d00m00”
- incell = "0.15arcsec”
- incenter = "230GHz”
- inwidth = '2GHz'

setpointings

```
setpointings          =       True
     integration      =      '10s'         #  integration (sampling) time
     direction        =         ''         #  "J2000 19h00m00 -40d00m00" or "" to center on model
     mapsize          =   ['', '']         #  angular size of map or "" to cover model
     maptype          = 'hexagonal'        #  hexagonal, square, etc
     pointingspacing =          ''         #  spacing in between pointings or "0.25PB" or "" for 0.5 PB
```

- integration = '600s’ ← Time average for each data point

predict

```
predict             =       True        #  calculate visibilites using ptgfile
     complist       =         ''        #  optional componentlist to observe with
                                        #   skymodel
     compwidth      =     '2GHz'        #  optional bandwidth if simulating from
                                        #   components only
     antennalist    = 'alma.out10.cfg'  #  antenna position file or "" for no
                                        #   interferometric MS
     refdate        = '2012/05/21/22:05:00'  help
     totaltime      =    '7200s'        #
     caldirection   =         ''        #                                  al]
     calflux        =      '1Jy'
     sdantlist      =         ''        #                                  or "" for
                                        #
     sdant          =          0        #  single dish antenna index in file
```

Total time for the observation (must be ≥ *integration* times the number of pointings)

To use the Early Science Configurations

- repodir = os.getenv("CASAPATH").split(' ')[0]
- antennalist = repodir+"/data/alma/simmos/alma.cycle0.compact.cfg"
- antennalist = repodir+"/data/alma/simmos/alma.cycle0.extended.cfg"

predict

```
predict             =       True        #  calculate visibilites using ptgfile
     complist       =         ''        #  optional componentlist to observe with
                                        #   skymodel
     compwidth      =     '2GHz'        #  optional bandwidth if simulating from
                                        #   components only
     antennalist    = 'alma.out10.cfg'  #  antenna position file or "" for no
                                        #   interferometric MS
     refdate        = '2012/05/21/22:05:00' #  time/date of observation *see help
     totaltime      =    '7200s'        #  total time of observation
     caldirection   =         ''        #  pt source calibrator [experimental]
     calflux        =      '1Jy'
     sdantlist      =         ''        #  single dish antenna position file or "" for
                                        #   no total power MS
     sdant          =          0        #  single dish antenna index in file
```

To use the most compact Full Science Configuration

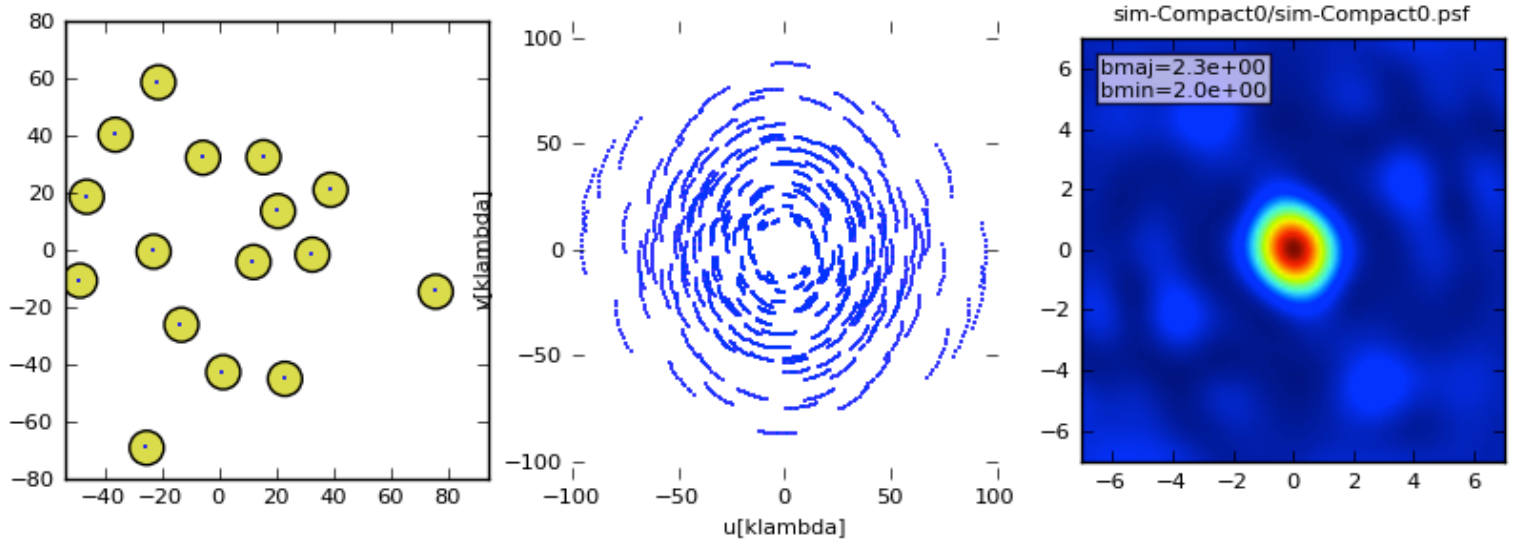
- repodir = os.getenv("CASAPATH").split(' ')[0]
- antennalist = repodir+"/data/alma/simmos/alma.out01.cfg"

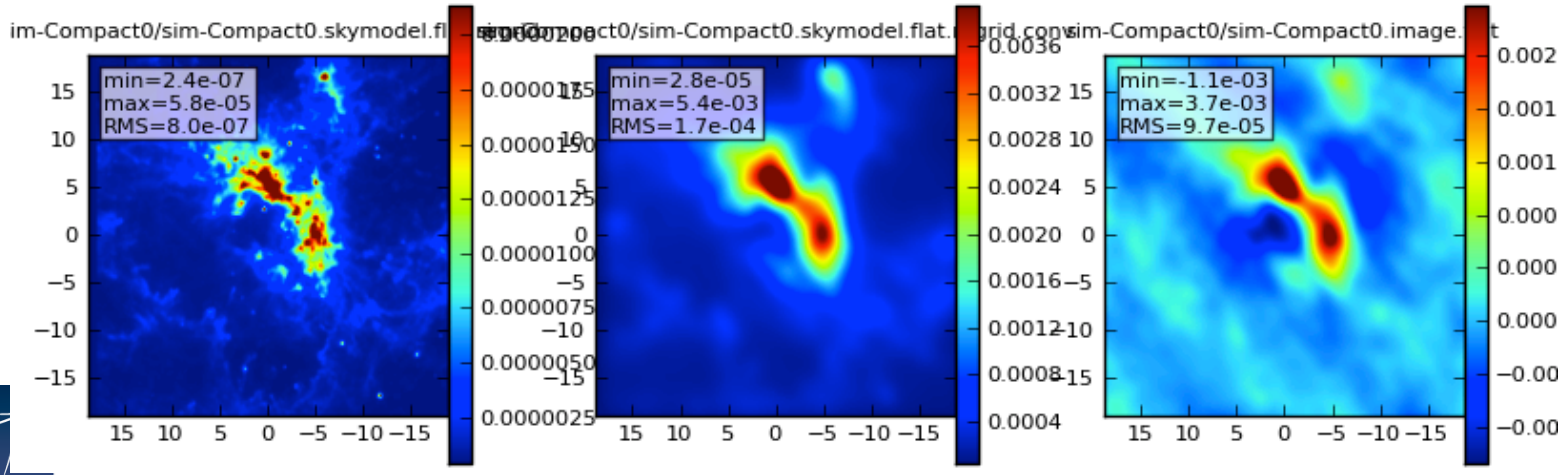
Simdata Output

```
analyze            =         True       #  (only first 6 selected outputs will be
                                        #   displayed)
    showarray      =        False       #  like plotants
    showuv         =         True       #  display uv coverage
    showpsf        =         True       #  display synthesized (dirty) beam
    showmodel      =         True       #  display sky model at original resolution
    showconvolved  =        False       #  display sky model convolved with output
                                        #   beam
    showclean      =         True       #  display the synthesized image
    showresidual   =        False       #  display the clean residual image
    showdifference =         True       #  display difference image
    showfidelity   =         True       #  display fidelity
```

- analyze = True
- showarray = True
- showconvolved = True
- showdifference = False
- showfidelity = False

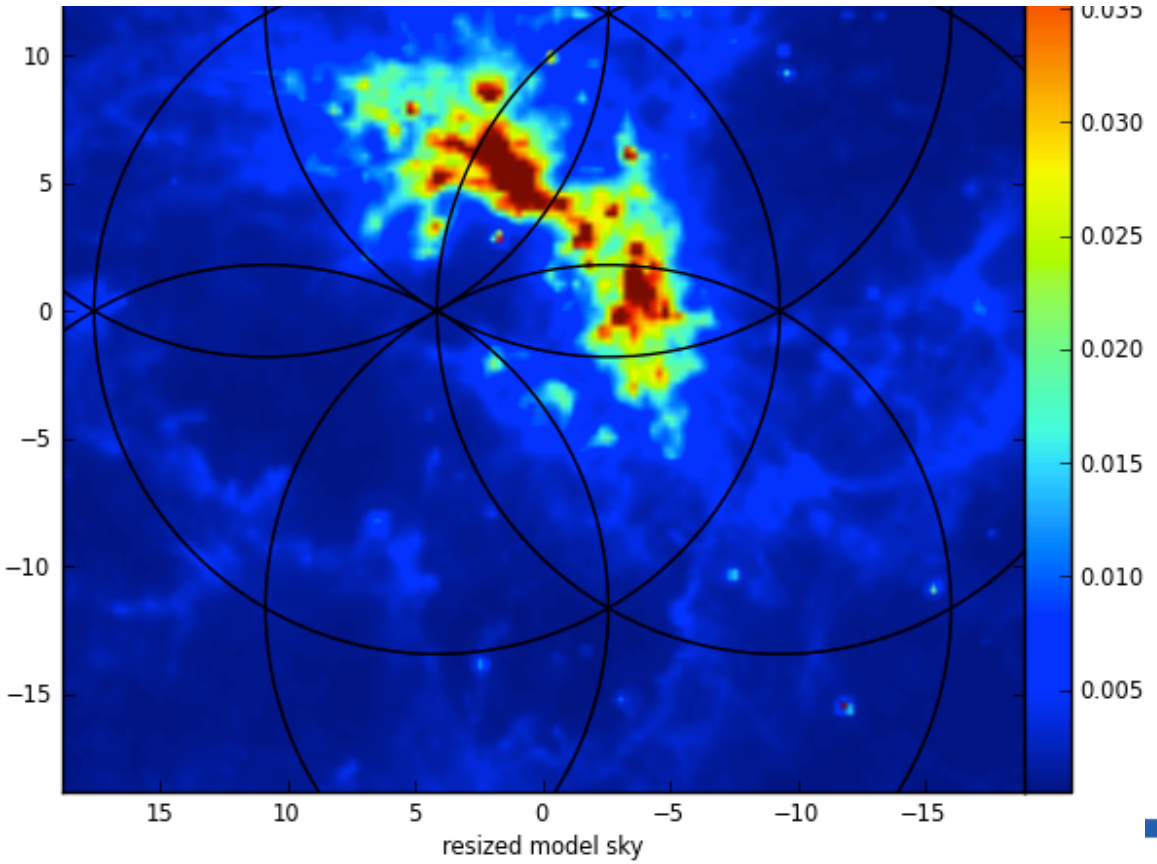
Simdata Output – ES Compact Array





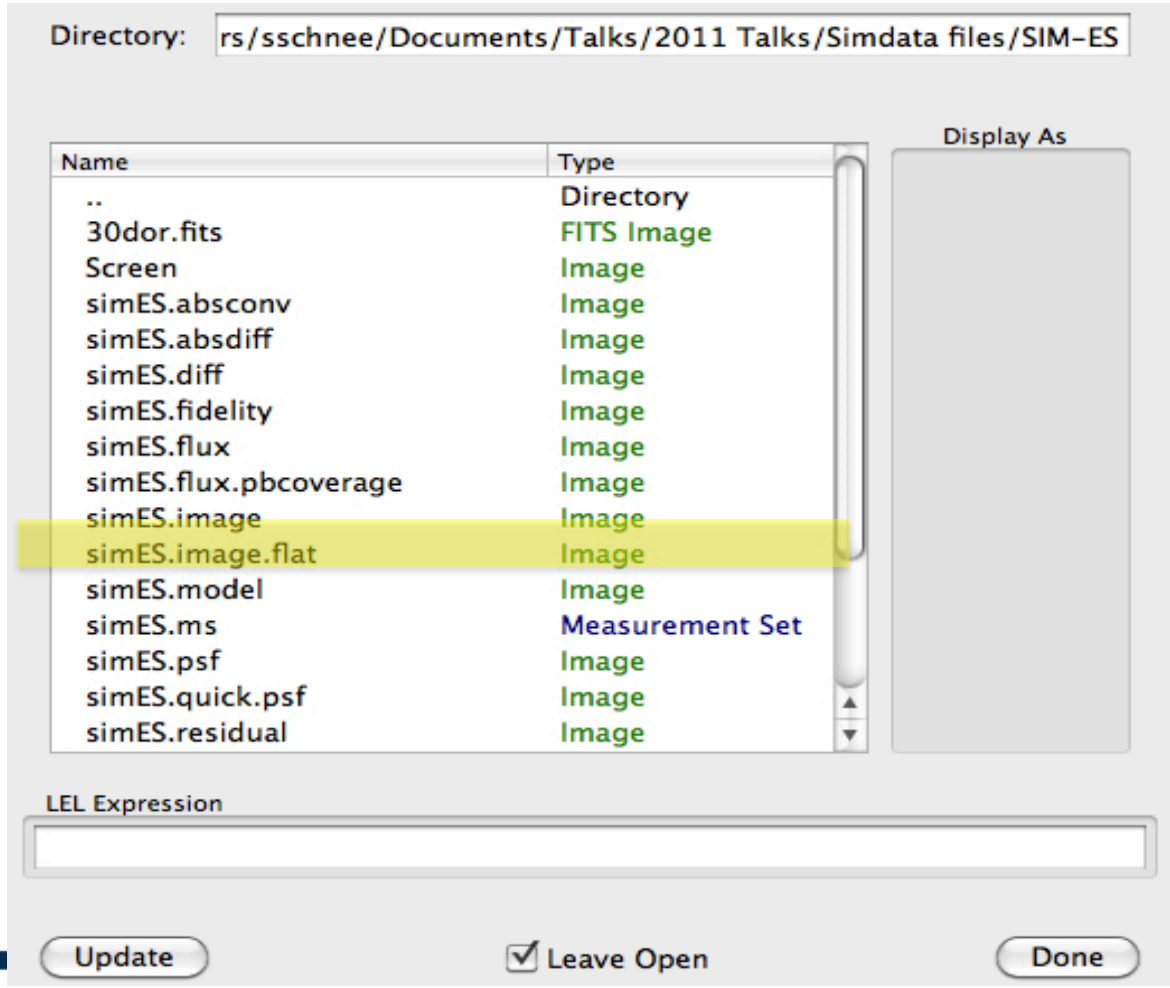
Simdata Output

Note that there are six pointings, and we previously set *integration* time to 600 seconds, so this product is less than the 7200 seconds set in *totaltime*



Simdata Output – Viewer

http://casa.nrao.edu/CasaViewerDemo/casaViewerDemo.html



Other Simdata Options
Thermal Noise

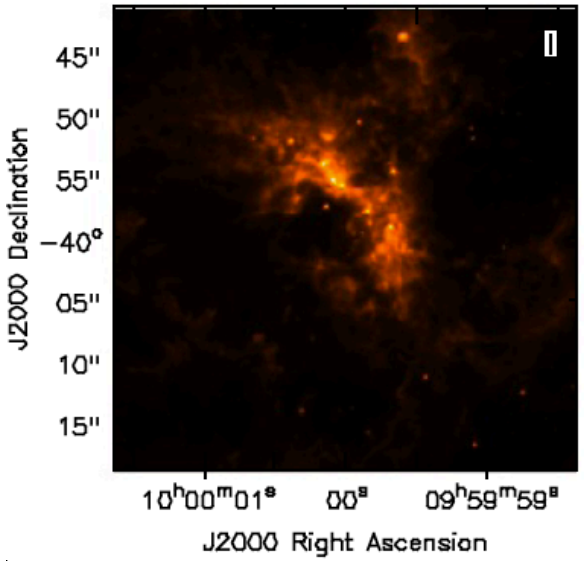
```
thermalnoise          = 'tsys-atm'          #  add thermal noise: [tsys-atm|tsys-
                                            #   manual|""]
     user_pwv         =          1.0        #  Precipitable Water Vapor in mm
     t_ground         =        270.0        #  ambient temperature

leakage               =          0.0        #  cross polarization
image                 =         True        #  (re)image $project.ms to
                                            #   $project.image
     vis              = '$project.noisy.ms' #  Measurement Set(s) to image
```

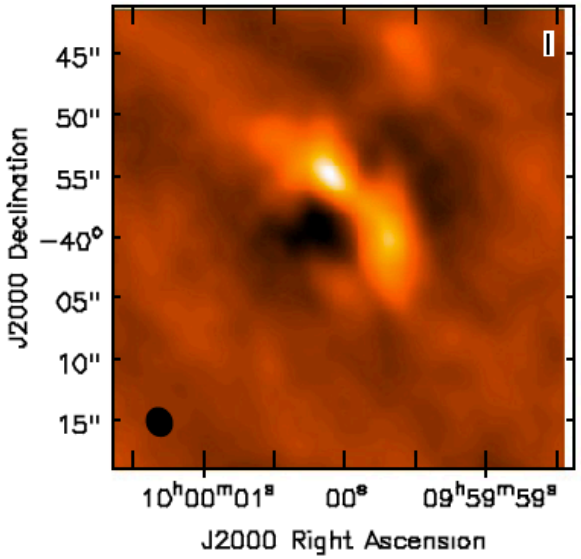
- thermalnoise = 'tsys-atm'
- vis='$project.noisy.ms'

Other Simdata Options
Thermal Noise

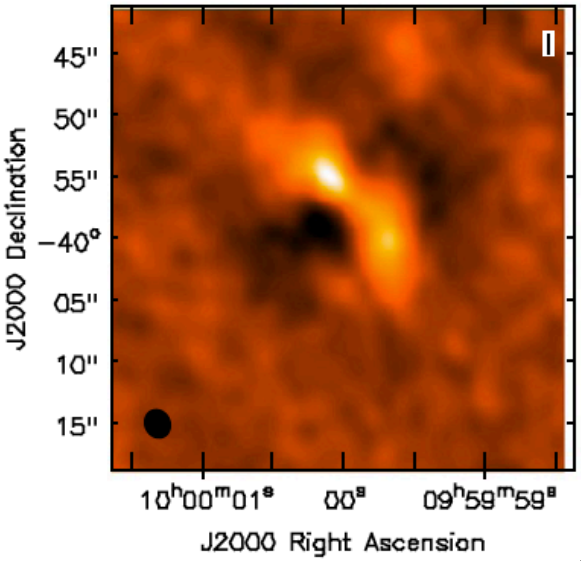
Sky Model



No Thermal Noise



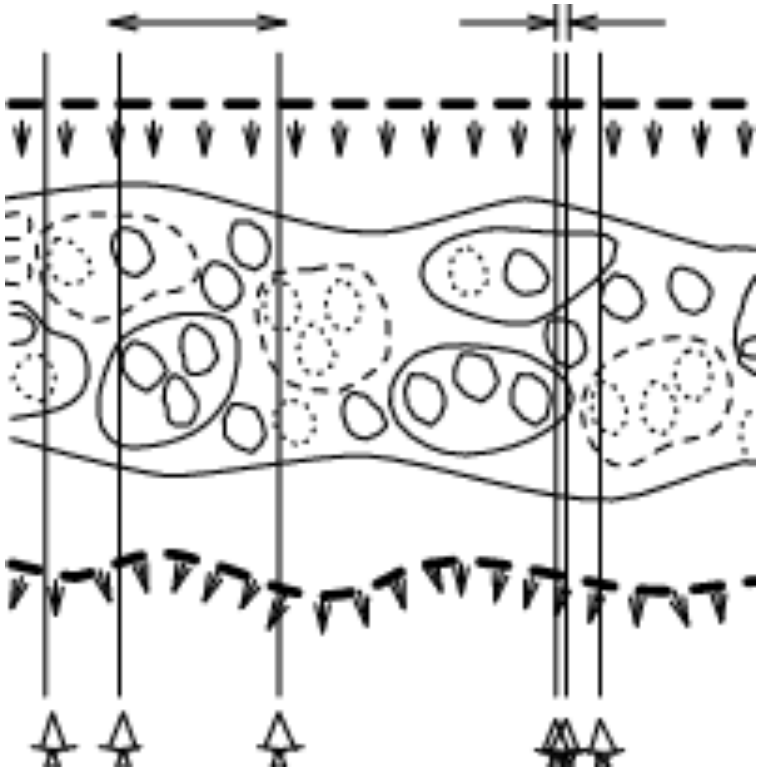
Thermal Noise



Atmospheric phase fluctuations

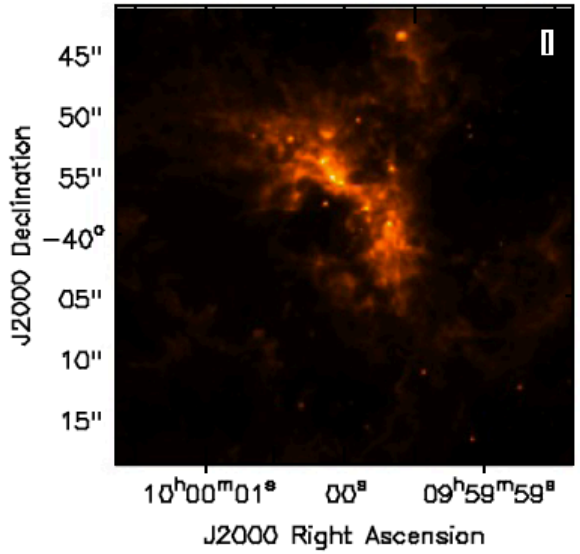
- Variations in the amount of precipitable water vapor (PWV) cause phase fluctuations and result in
  - Low coherence (loss of sensitivity)
  - Radio "seeing", typically 1″ at 1 mm
  - Anomalous pointing offsets
  - Anomalous delay offsets

  Patches of air with different water vapor content (and hence index of refraction) affect the incoming wave front differently.

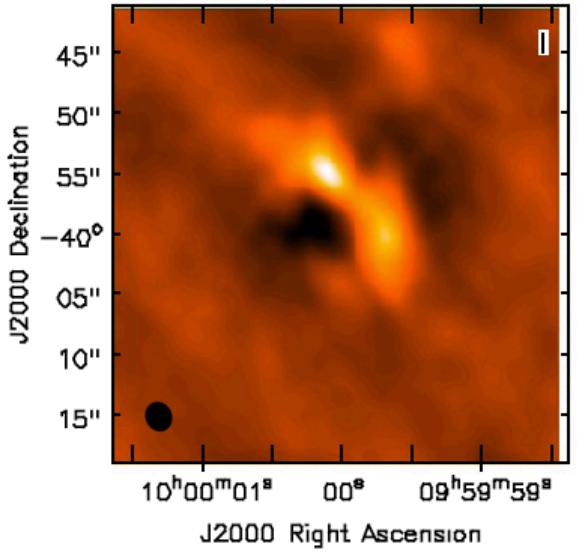


Other Simdata Options
Phase Noise

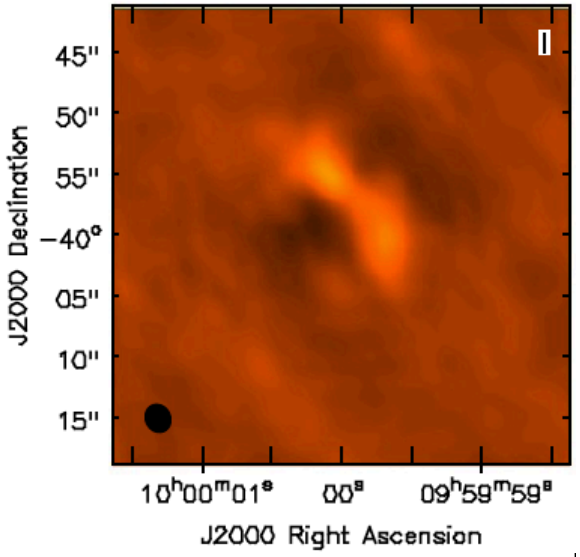
Sky Model



No Phase Noise



Phase Noise



Other Simdata Options
Phase Noise

- Start with the visibilities (measurement set) created by simdata
- Use the "toolkit" to add phase noise
  - http://casa.nrao.edu/docs/CasaRef/CasaRef.html
  - http://casaguides.nrao.edu/index.php?title=Corrupt
  - sm.openfromms("simPN.ms")
  - sm.settrop(mode='screen',pwv=1.0,deltapwv=0.15)
  - sm.corrupt()
  - sm.done()
- Use `clean` (in CASA) to make the image

Your Turn

- Find a fits file of an object you want to image
- Use simdata to see how it would look if observed with ALMA
  - Rescale image if necessary
  - Try different antenna configurations
  - Add noise if interested
- Don't be afraid to ask for help!

Your Turn

- Sample images can be found at:
  - http://casaguides.nrao.edu/index.php?title=Sim_Inputs
- Simdata walk-throughs available at:
  - http://casaguides.nrao.edu
    - Simulating observations in CASA 3.2

Extra Slides

Simdata output images

- Most important images are:
  - $project.image – output simulated cube/plane
  - $project.image.flat – moment 0 of simulated cube/plane
  - $project.skymodel – input cube/plane
  - $project.psf – synthesized beam (point spread function)
  - $project.residual – residuals after cleaning

Simdata hands-on quickstart

- Start CASA and look at simdata
  - casapy, inp simdata, help simdata
- Go through simdata tutorial
  - http://casaguides.nrao.edu
- Get image for your own simulation
  - http://casaguides.nrao.edu/index.php?title=Sim_Inputs

Simdata FAQ

- Find the antenna configuration files
  - repodir = os.getenv("CASAPATH").split(' ')[0]
  - antennalist = repodir+"/data/alma/simmos/alma.cycle0.compact.cfg"
- Problems with FITS file?
  - modifymodel = True
  - Set all parameters manually
- "image detached" error message?
  - image = True
  - analyze = True

Simdata FAQ

- Simulation running too long?
  - Increase value of "integration"
  - Decrease number of required pointings
  - Clean for fewer iterations
- Can't see effect of added thermal noise?
  - thermalnoise = 'tsys-atm'
  - vis = '$project.noisy.ms'
  - Check brightness of input image vs expected noise