



Simulating ALMA observations

Sandra Burkutean, Italian ARC node



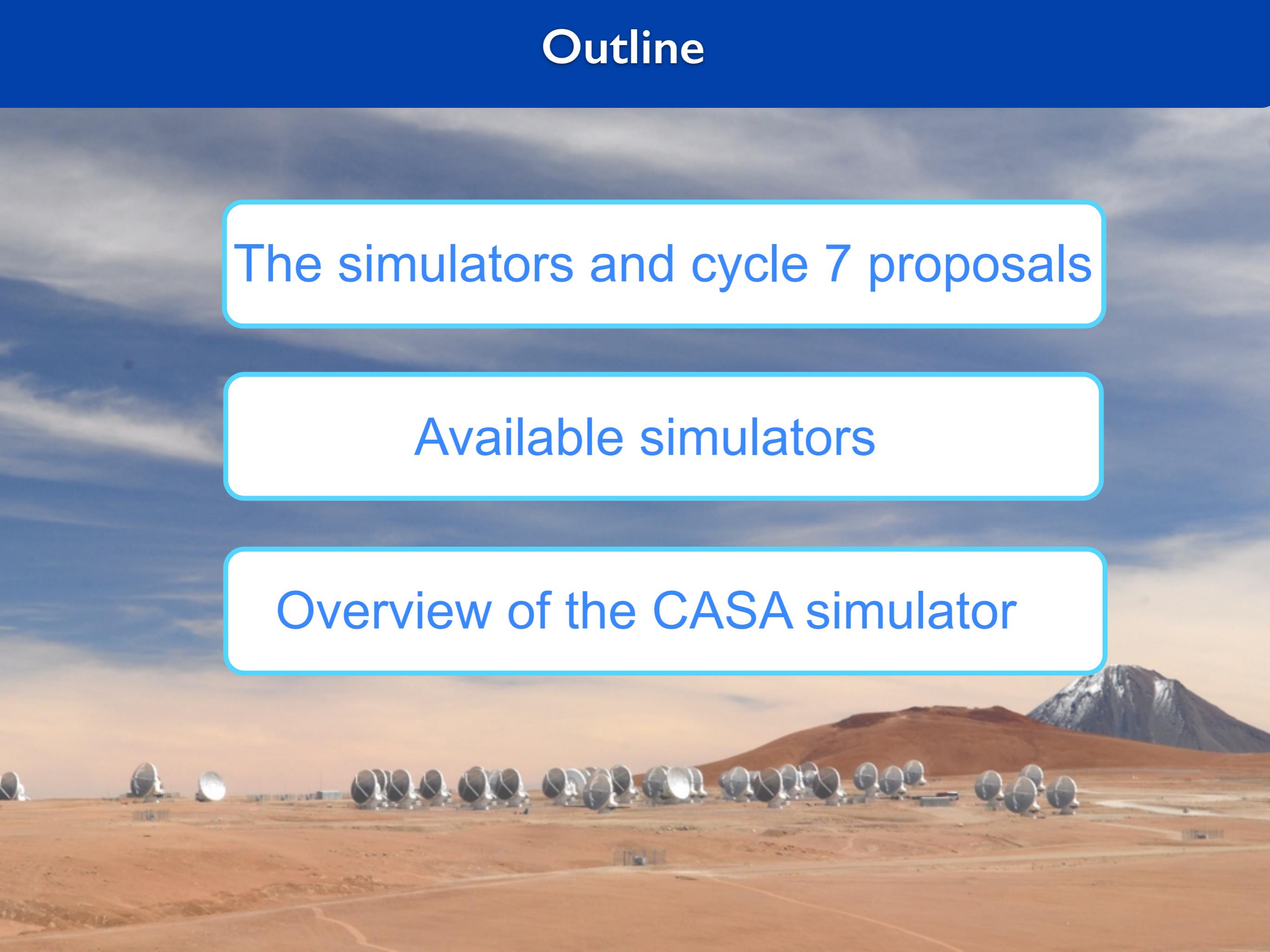
ALMA Science and Proposals Workshop,
26.02.2019

Outline

The simulators and cycle 7 proposals

Available simulators

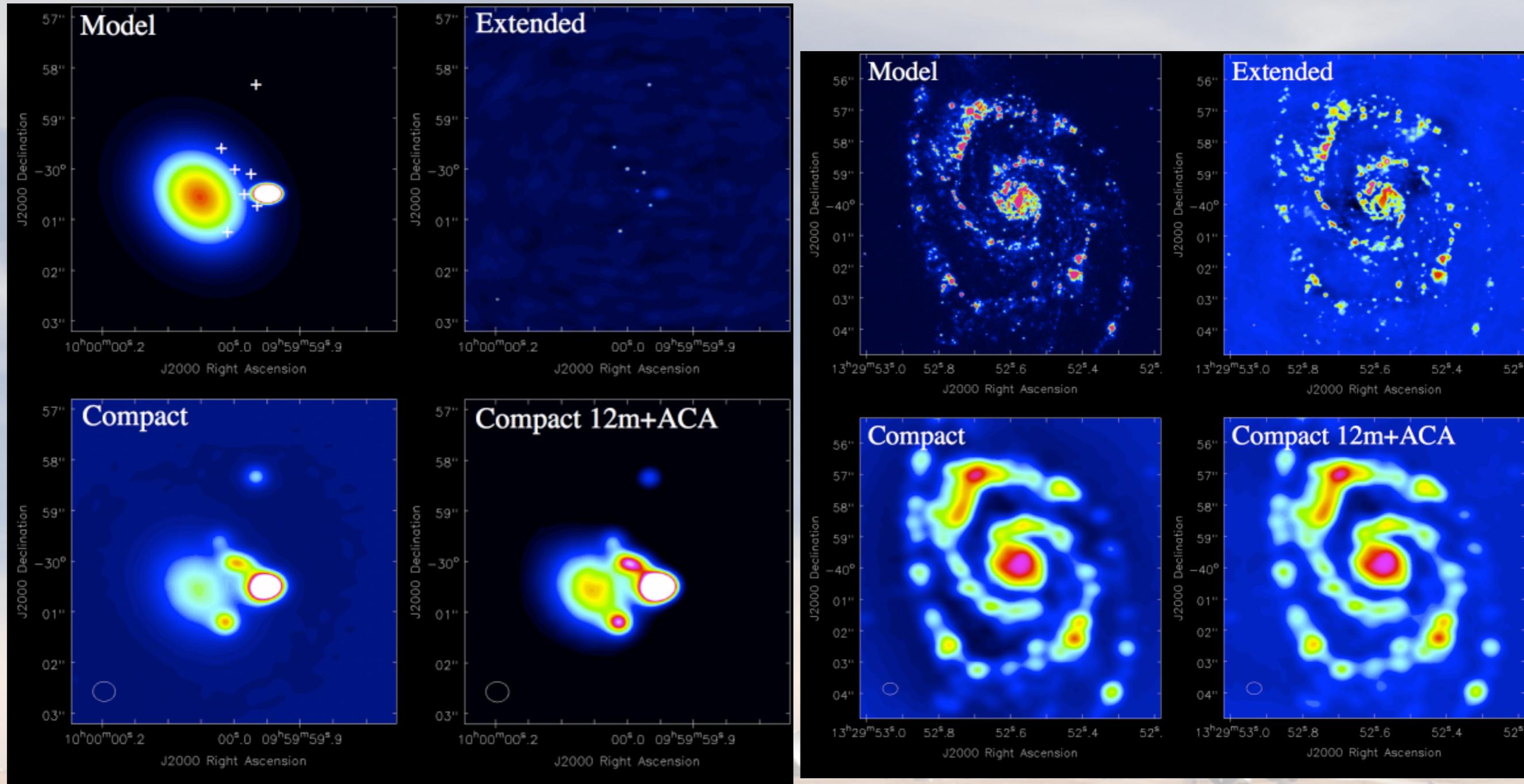
Overview of the CASA simulator



Why should I do an ALMA simulation ?

- You can try out different ALMA configurations and, for a given model, this might help you decide on the required angular resolution and maximum angular scale.
- A simulation can help support the required technical set-up in your proposal
- In particular, it can help you justify the need for complementary ACA/TP or, indeed, ACA-only observations.

Why should I do an ALMA simulation ?



source: CASA simulator pages

Available simulators

CASA simulator

```
# simobserve :: mosaic simulation task:  
project      = 'sim'          # root prefix for output file names  
skymodel    = '30dor.fits'    # model image to observe  
inbright    = '0.06mJy/pixel' # scale surface brightness of brightest pixel e.g. "1.2Jy/pixel"  
indirection = 'J2000 10h00m00 -40d00m00' # set new direction e.g. "J2000 19h00m00 -40d00m00"  
incell      = '0.15arcsec'   # set new cell/pixel size e.g. "0.1arcsec"  
incenter    = '230GHz'       # set new frequency of center channel e.g. "89GHz" (required even for 2D model)  
inwidth     = '2GHz'         # set new channel width e.g. "10MHz" (required even for 2D model)  
  
complist    = ''             # componentlist to observe  
setpointings = True          # integration (sampling) time  
integration = '600s'         # "J2000 19h00m00 -40d00m00" or "" to center on model  
direction   = ''             # angular size of map or "" to cover model  
mapsize     = ['', '']        # hexagonal, square, etc  
maptype    = 'topographic'  # spacing in between pointings or "0.25PB" or "" for 0.5 PB  
pointingspacing = ''  
  
obsmode     = 'int'          # observation mode to simulate  
# [int(interferometer)|sd(singledish)|"(none)"  
antennalist = '/usr/lib64/casapy/stable/data/alma/simmos/alma.cycle0.compact.cfg' # interferometer  
# antenna position file  
refdate     = '2012/05/21'    # date of observation - not critical unless concatenating simulations  
hourangle   = 'transit'      # hour angle of observation center e.g. -3:00:00, or "transit"  
totaltime   = '7200s'        # total time of observation or number of repetitions  
caldirection = ''            # pt source calibrator [experimental]  
calflux     = '1Jy'          #  
  
thermalnoise = ''  
leakage      = 0.0  
graphics     = 'both'  
verbose      = False  
overwrite    = True  
async        = False
```



m|tsys-manual|""]
rometer only)
age to [screen|file|both|none]

th \$project
e started using simobserve(...)

ALMA Observation Support tool

ALMA Observation Support Tool

OST NEWS HELP QUEUE LIBRARY ALMA HELPDESK

Important information on the new OST version.

Array Setup:

Instrument: ALMA

Select the desired ALMA antenna configuration.

Sky Setup:

Source model: OST Library: Central point source

Upload: Choose File no file selected

Choose a library source model or supply your own.

You may upload your own model here (max 10MB).

Declination: -35d00m00.0s

Image peak / point flux in mJy : 0.0

Ensure correct formatting of this string (+/-0d00m00s).

Rescale the image data with respect to new peak value.

Set to 0.0 for no rescaling of source model.

Observation Setup:

Observing mode: Spectral Continuum

Central frequency in GHz: 93.7

The value entered must be within an ALMA band.

Bandwidth in MHz : 32

Select the total bandwidth for continuum observations.

Enter 7.5 GHz to select ALMA recommend full continuum.

Number of polarizations: 2

This affects the noise in the final map.

Required resolution in arcseconds: 1.0

OST will choose array config based on this value if no array is specified.

Pointing strategy: Mosaic

Selecting single will apply primary beam attenuation.

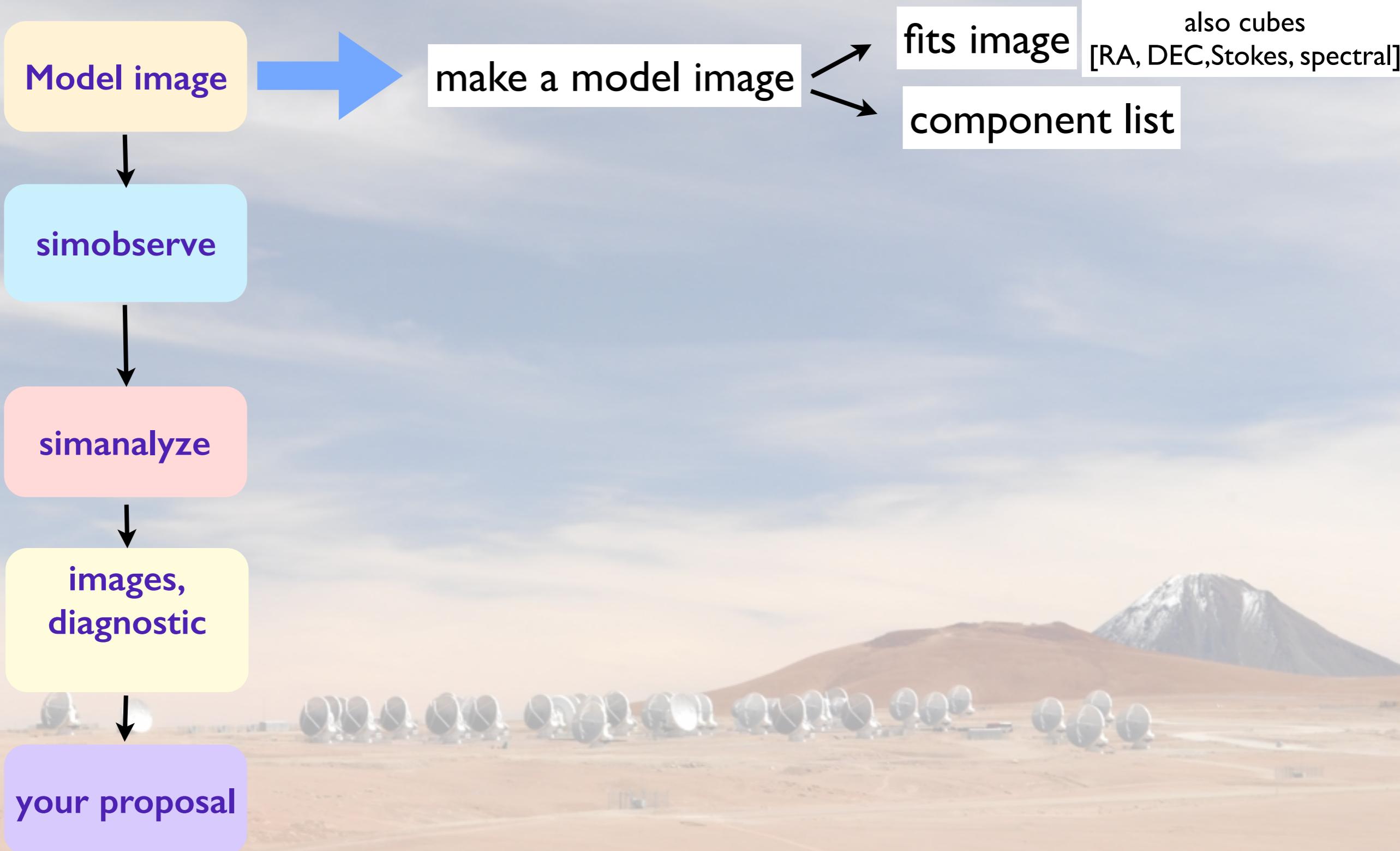
On-source time in hours : 3

Per pointing for mosaics.

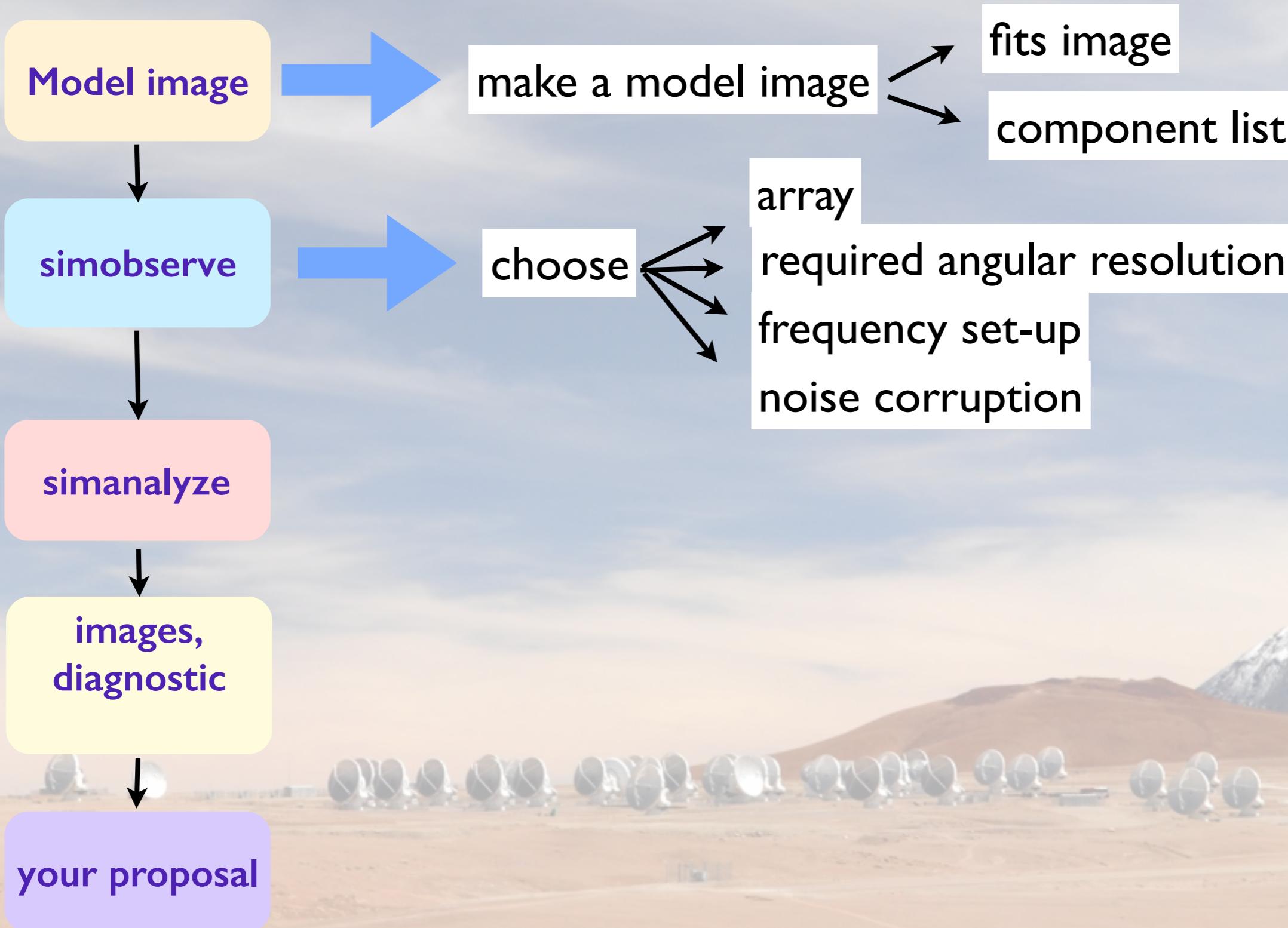
Start hour angle: 0.0

Deviation of start of observation from transit.

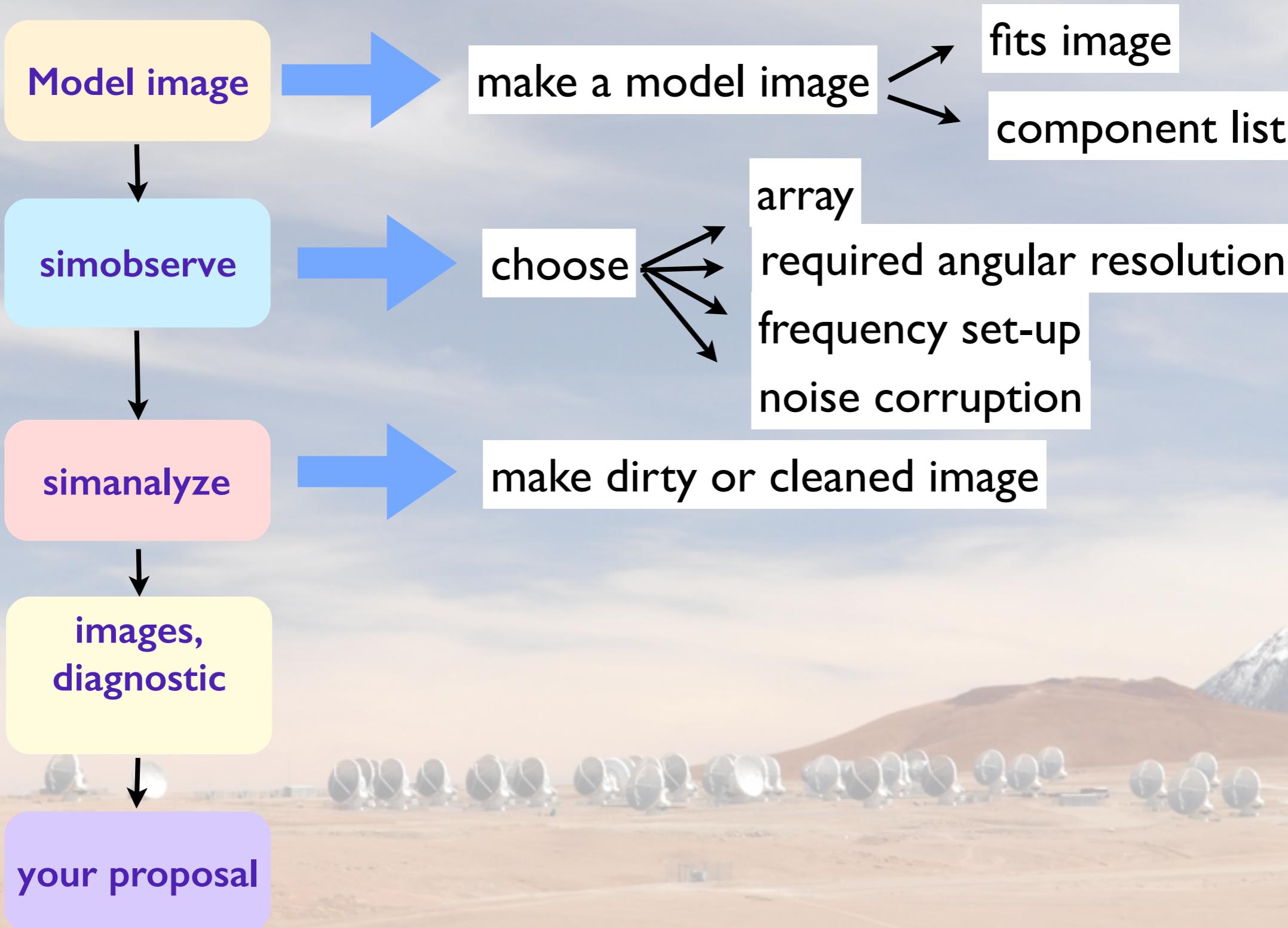
CASA simulator and cycle 7



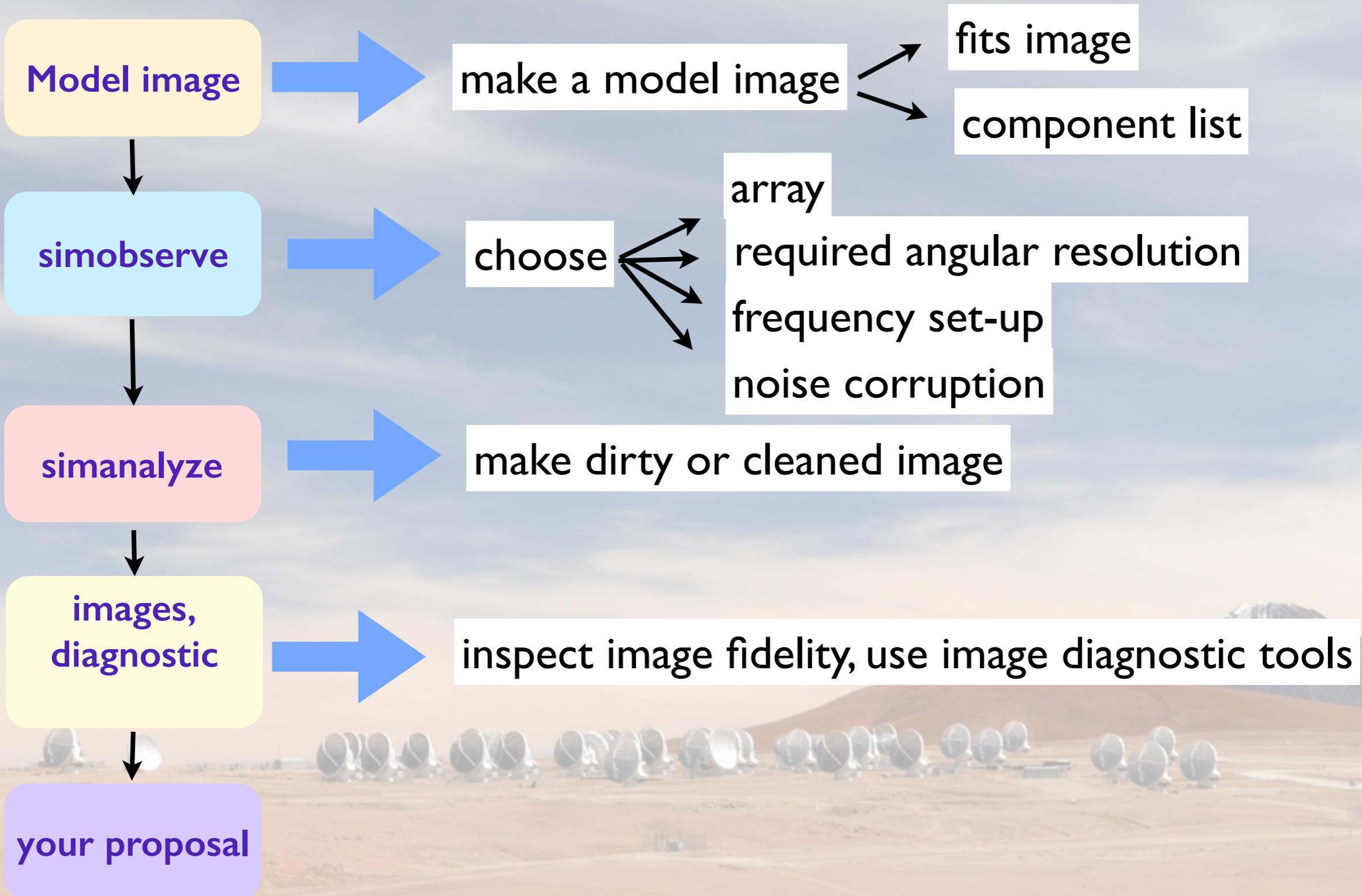
CASA simulator and cycle 7



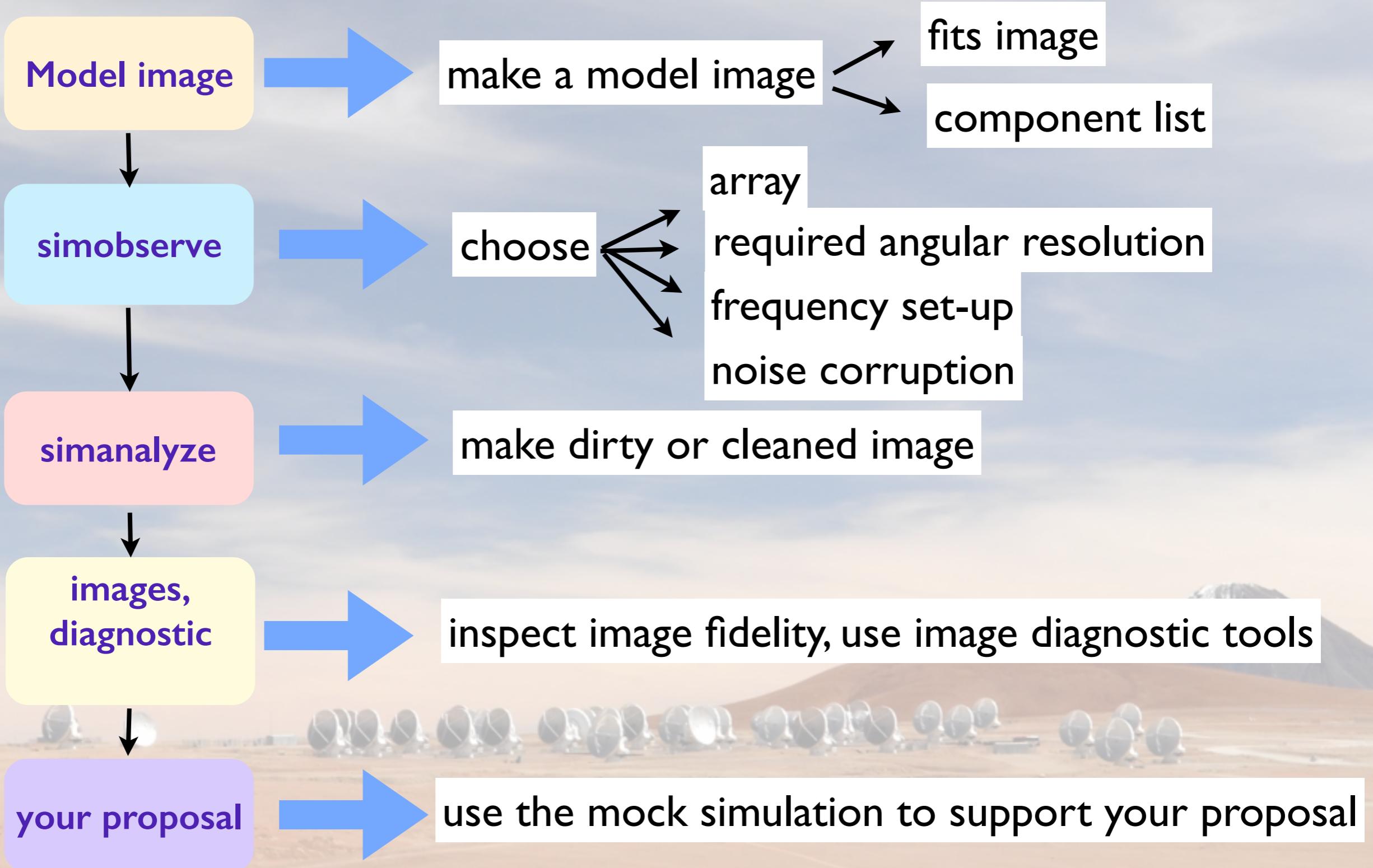
CASA simulator and cycle 7



CASA simulator and cycle 7



CASA simulator and cycle 7



Simobserve

```
# simobserve :: mosaic simulation task:
project          = 'sim'                      # root prefix for output file names
skymodel         = '30dor.fits'                 # model image to observe
inbright         = '0.06mJy/pixel'              # scale surface brightness of brightest pixel e.g. "1.2Jy/pixel"
indirection     = 'J2000 10h00m00 -40d00m00' # set new direction e.g. "J2000 19h00m00 -40d00m00"
incell           = '0.15arcsec'                # set new cell/pixel size e.g. "0.1arcsec"
incenter         = '230GHz'                    # set new frequency of center channel e.g. "89GHz" (required even for
                                              # model)
inwidth          = '2GHz'                     # set new channel width e.g. "10MHz" (required even for 2D model)

complist         = ''                         # componentlist to observe
setpointings     = True                       # 
integration      = '600s'                     # integration (sampling) time
direction        = ''                         # "J2000 19h00m00 -40d00m00" or "" to center on model
mapsize          = ['', '']                   # angular size of map or "" to cover model
maptype          = 'topographic'             # hexagonal, square, etc
pointingspacing = ''                         # spacing in between pointings or "0.25PB" or "" for 0.5 PB

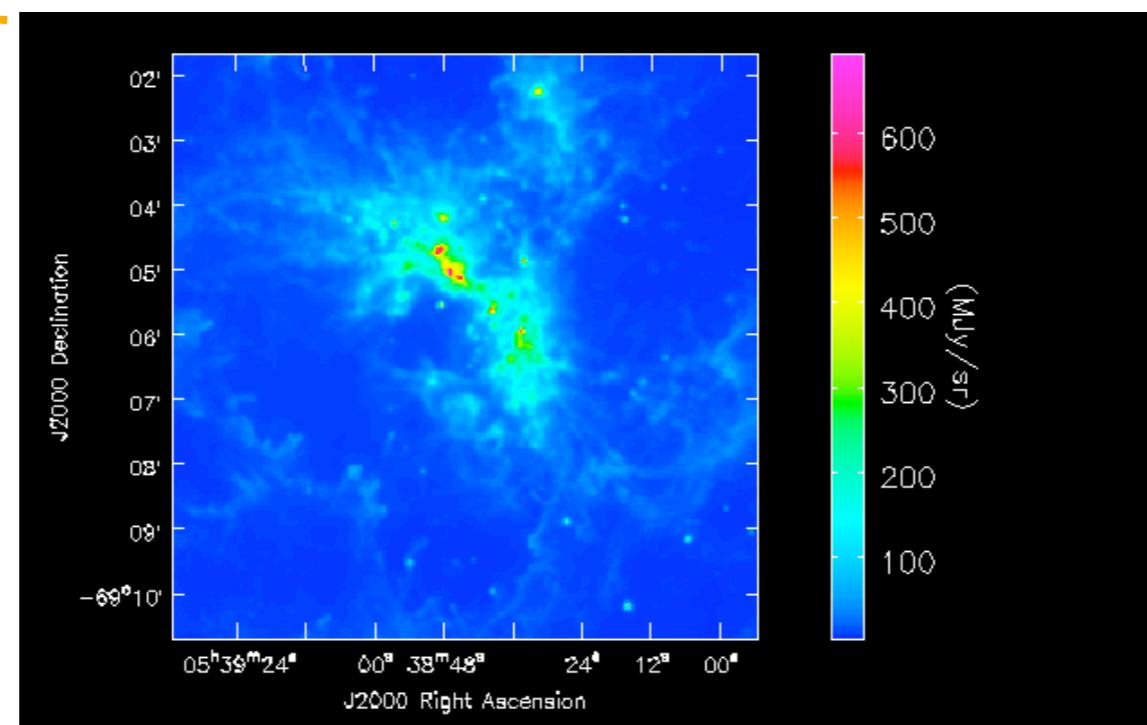
obsmode          = 'int'                      # observation mode to simulate
                                              # [int(interferometer)|sd(singledish)|""(none)]
antennalist      = '/usr/lib64/casapy/stable/data/alma/simmos/alma.cycle0.compact.cfg' # interferometer
                                              # antenna position file
refdate          = '2012/05/21'                # date of observation - not critical unless concatting simulations
hourangle         = 'transit'                  # hour angle of observation center e.g. -3:00:00, or "transit"
totaltime         = '7200s'                    # total time of observation or number of repetitions
caldirection     = ''                         # pt source calibrator [experimental]
calflux          = '1Jy'                     

thermalnoise      = ''                         # add thermal noise: [tsys-atm|tsys-manual|""]
leakage           = 0.0                        # cross polarization (interferometer only)
graphics          = 'both'                     # display graphics at each stage to [screen|file|both|none]
verbose           = False                      # 
overwrite         = True                       # overwrite files starting with $project
async             = False                      # If true the taskname must be started using simobserve(...)
```

Simobserve

```
# simobserve :: mosaic simulation task:  
project = 'sim' # root prefix for  
skymodel = '30dor.fits' # model image to  
inbright = '0.06mJy/pixel' # scale surface  
indirection = 'J2000 10h00m00 -40d00m00' # set  
incell = '0.15arcsec' # set new cell  
incenter = '230GHz' # set new freq  
# model)  
inwidth = '2GHz' # set new chan  
  
complist = '' # componentlist
```

Model
image



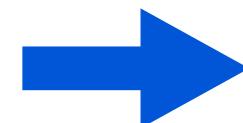
Spitzer IRAC 89 micron
image of 30 Doradus

Simobserve

interferometer (int) or single-dish (sd)

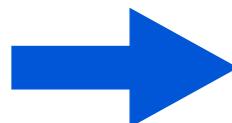


```
obsmode          = 'int'           # observation mode to simulate
                                # [int(interferometer)|sd(singledish)|""(none)]
antennalist      = '/usr/lib64/casapy/stable/data/alma/simmos/alma.cycle0.compact.cfg'
```



Cycle 7 representative configurations files

<https://almascience.eso.org/documents-and-tools/cycle7/alma-configuration-files>



or, give the required angular resolution

"alma;0.5arcsec"

(only works for full ALMA !)

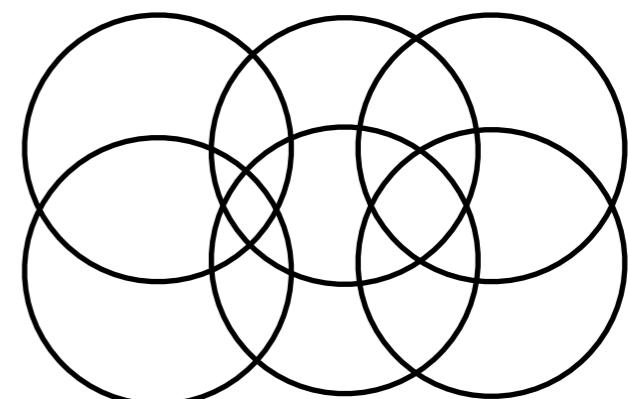


Simobserve

```
incenter      = '230GHz'  
inwidth       = '2GHz'
```

spectral set-up

```
complist      = ''  
setpointings  = True  
integration   = '600s'  
direction     = ''  
mapsize       = ['', '']  
maptype       = rectangle or hexagonal  
pointingspacing = ''
```



the simulator allows you to
test
several pointing scenarios

observing
time and
pointing

Simobserve

```
# simobserve :: mosaic simulation task:  
project          = 'sim'  
skymodel       = '30dor.fits'  
    inbright      = '0.06mJy/pixel'  
    indirection   = 'J2000 10h00m00 -40d00m00'  
    incell        = '0.15arcsec'  
    incenter      = '230GHz'  
    inwidth       = '2GHz'  
  
complist       = ''  
setpointings   = True  
    integration   = '600s'  
    direction     = ''  
    mapsize       = ['', '']  
    maptype       = 'topographic'  
    pointingspacing = ''  
  
obsmode        = 'int'  
  
    antennalist   = '/usr/lib64/casapy/stable/data/alma/simmos/alma.cycle0.compact.cfg' # inter  
  
    refdate       = '2012/05/21'  
hourangle      = 'transit'  
    totaltime     = '7200s'  
    caldirection  = ''  
    calflux       = '1Jy'  
  
thermalnoise  = 'tsys-atm'  
    user_pvw     = 1.0  
    t_ground     = 269.0  
    seed         = 11111  
  
leakage          = 0.0  
graphics         = 'both'  
verbose          = False  
overwrite        = True  
async            = False
```

model image

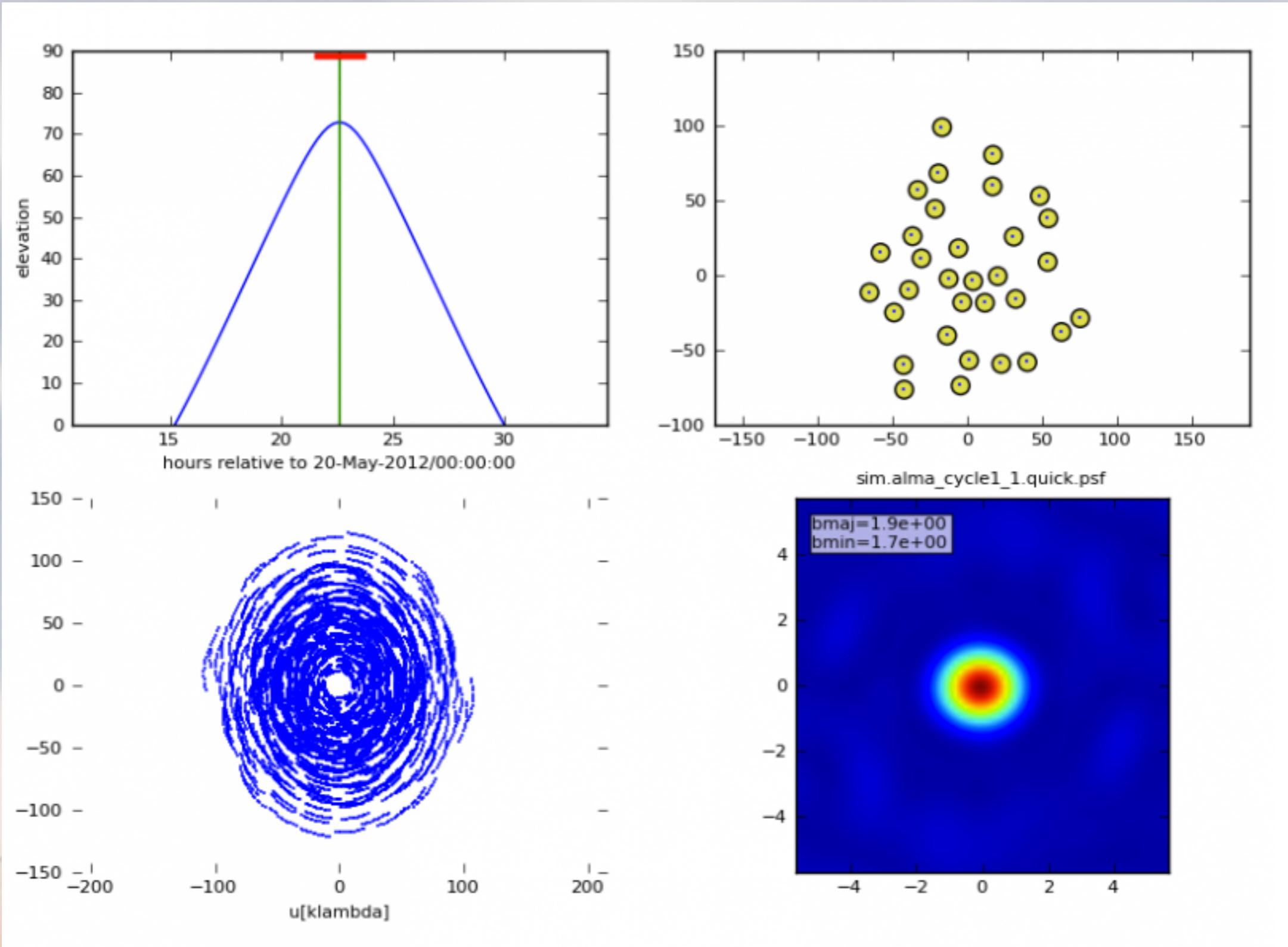
spectral set-up

observing
time and
pointing

array choice
or
required
angular
resolution

noise
addition

Simobserve



Imaging step

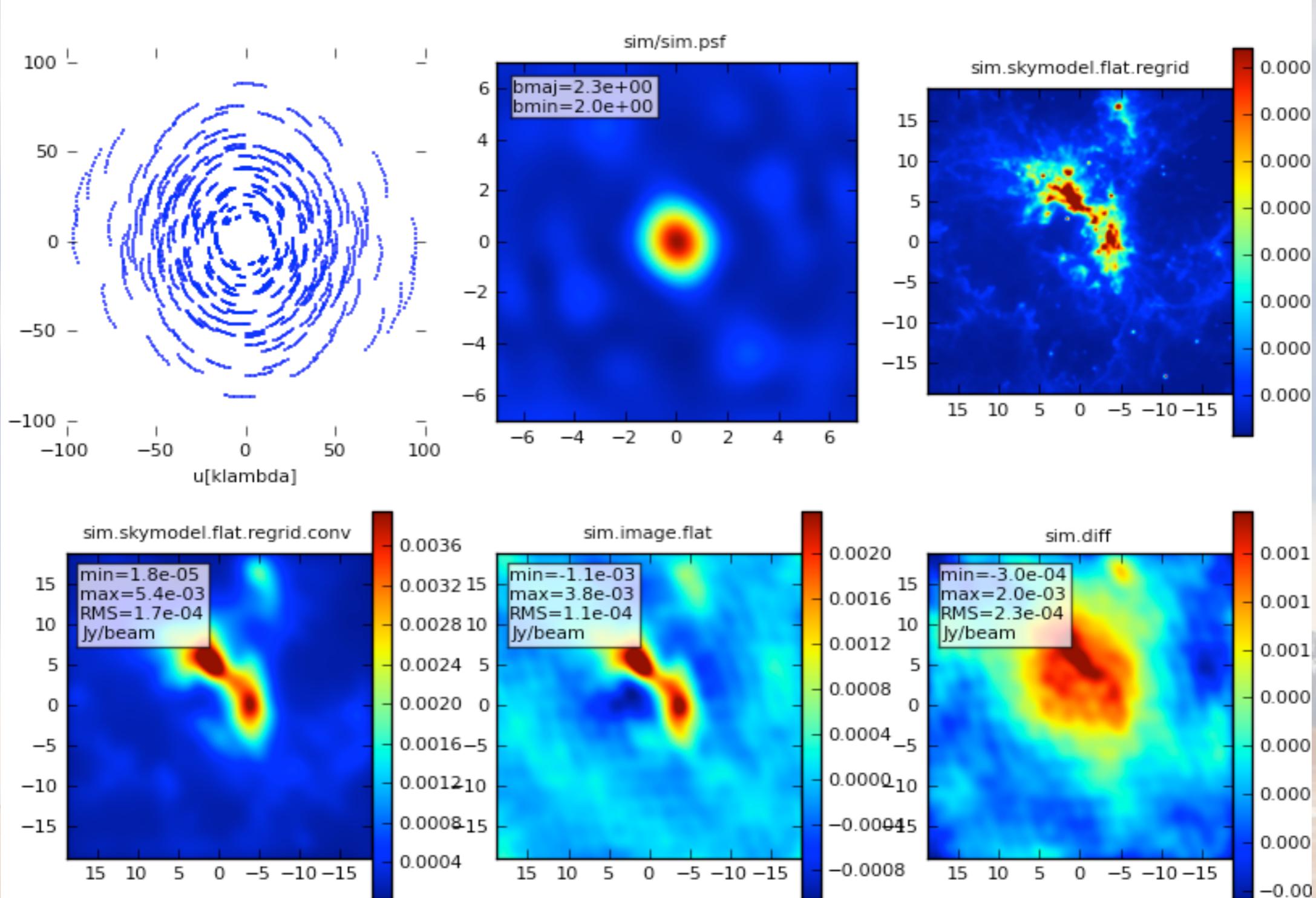
simanalyze

```
# simanalyze :: image and analyze simulated datasets
project          =      'sim'          # root prefix for output file names
image            =      True           # (re)image $project.*.ms to $project.image
vis              =      'default'       # Measurement Set(s) to image
modelimage       =      ''             # prior image to use in clean e.g. existing single dish image
imsize           =      0              # output image size in pixels (x,y) or 0 to match model
imdirection     =      ''             # set output image direction, (otherwise center on the model)
cell              =      ''             # cell size with units or "" to equal model
niter             =      500            # maximum number of iterations (0 for dirty image)
threshold        =      '0.1mJy'        # flux level (+units) to stop cleaning
weighting         =      'natural'       # weighting to apply to visibilities
mask              =      []             # Cleanbox(es), mask image(s), region(s), or a level
outertaper        =      []             # uv-taper on outer baselines in uv-plane
stokes            =      'I'             # Stokes params to image

analyze           =      True            # (only first 6 selected outputs will be displayed)
showuv           =      True            # display uv coverage
showpsf          =      True            # display synthesized (dirty) beam (ignored in single dish simulation)
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 (ignored in single dish simulation)
showdifference   =      True            # display difference image
showfidelity     =      True            # display fidelity

graphics          =      'both'          # display graphics at each stage to [screen|file|both|none]
verbose           =      False           # verbose output
overwrite         =      True            # overwrite files starting with $project
async             =      False           # If true the taskname must be started using simanalyze(...)
```

Simanalyze imaging output

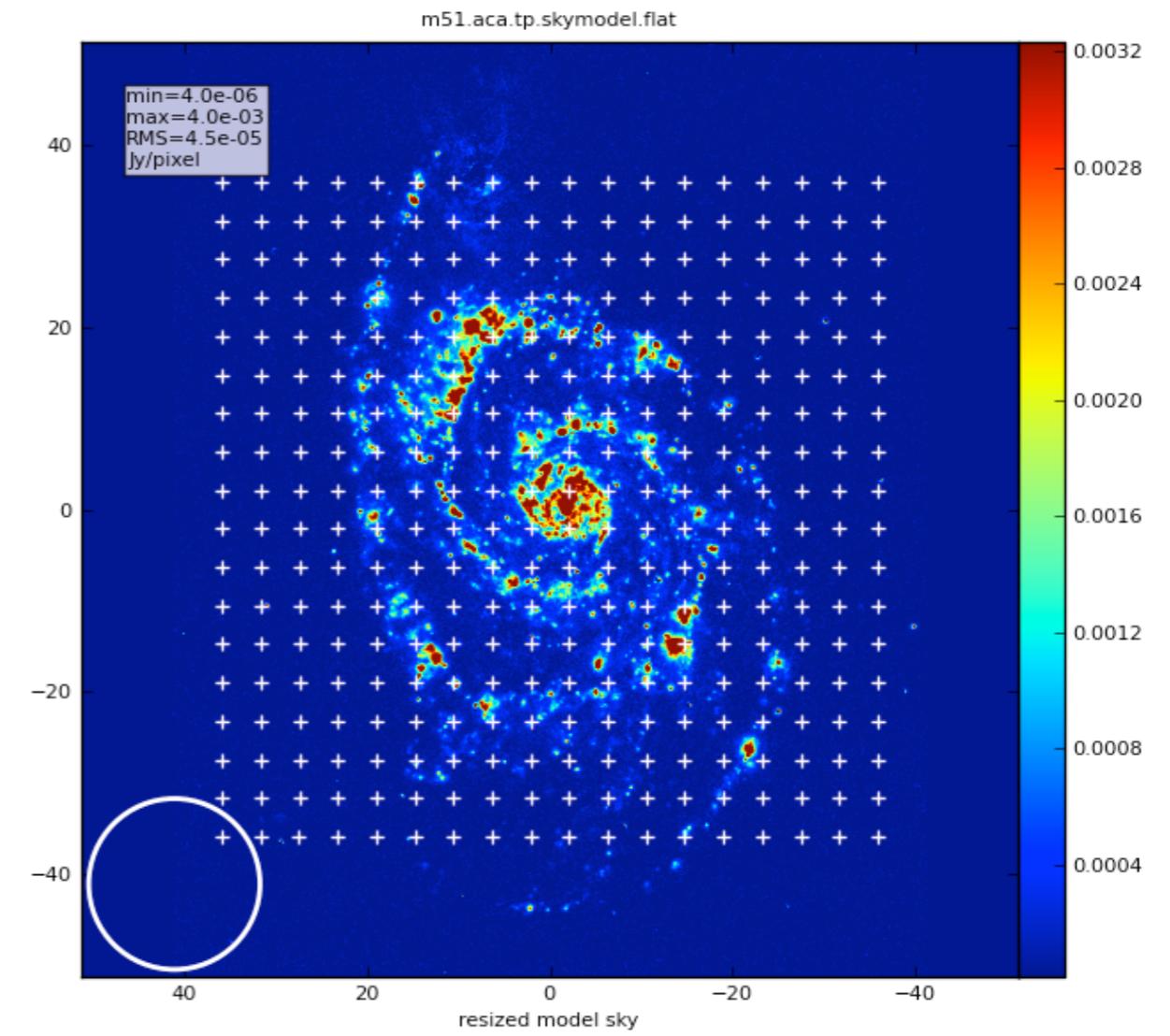
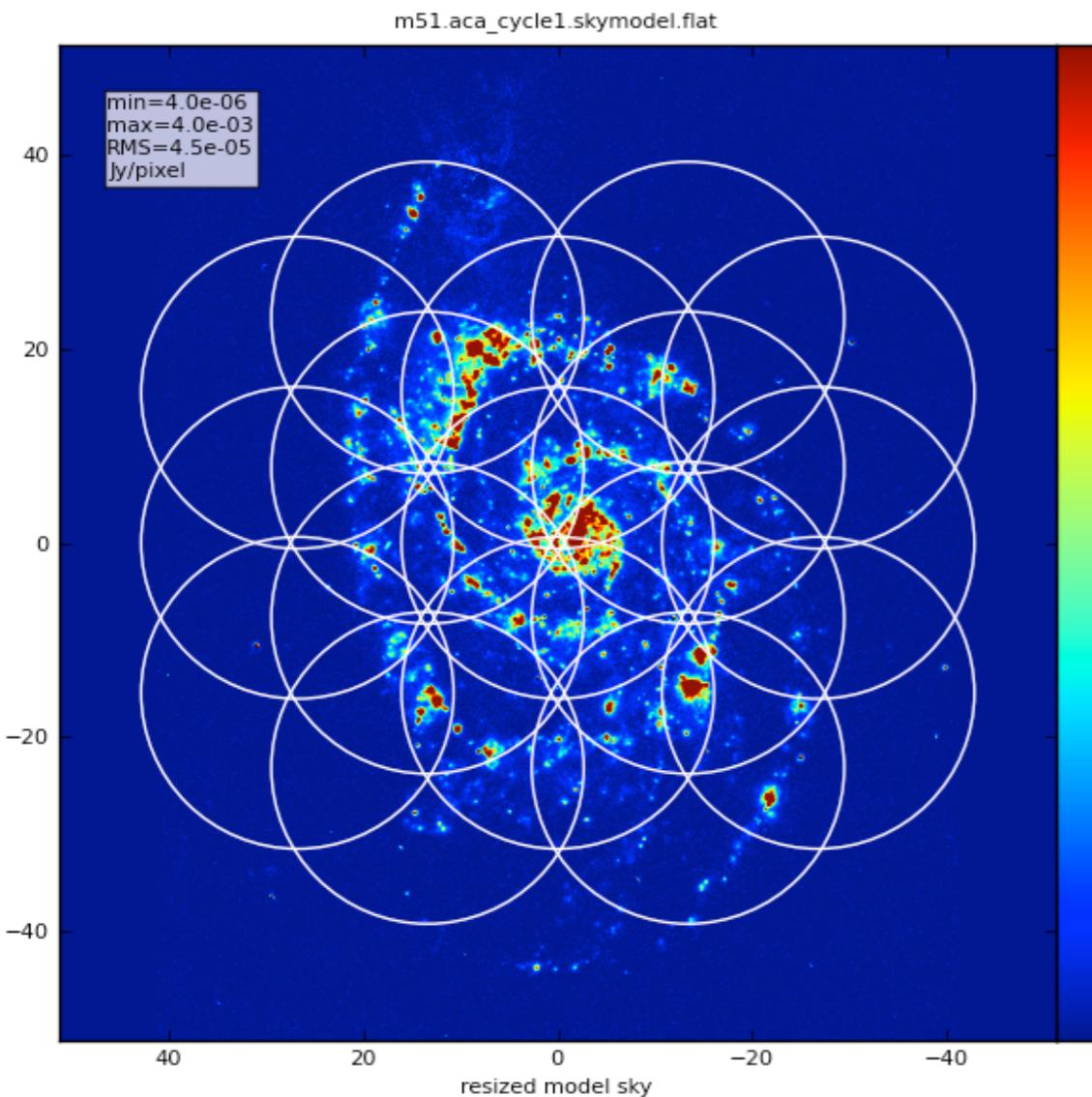


Simalma - combination of SD/ACA/ALMA during de-convolution

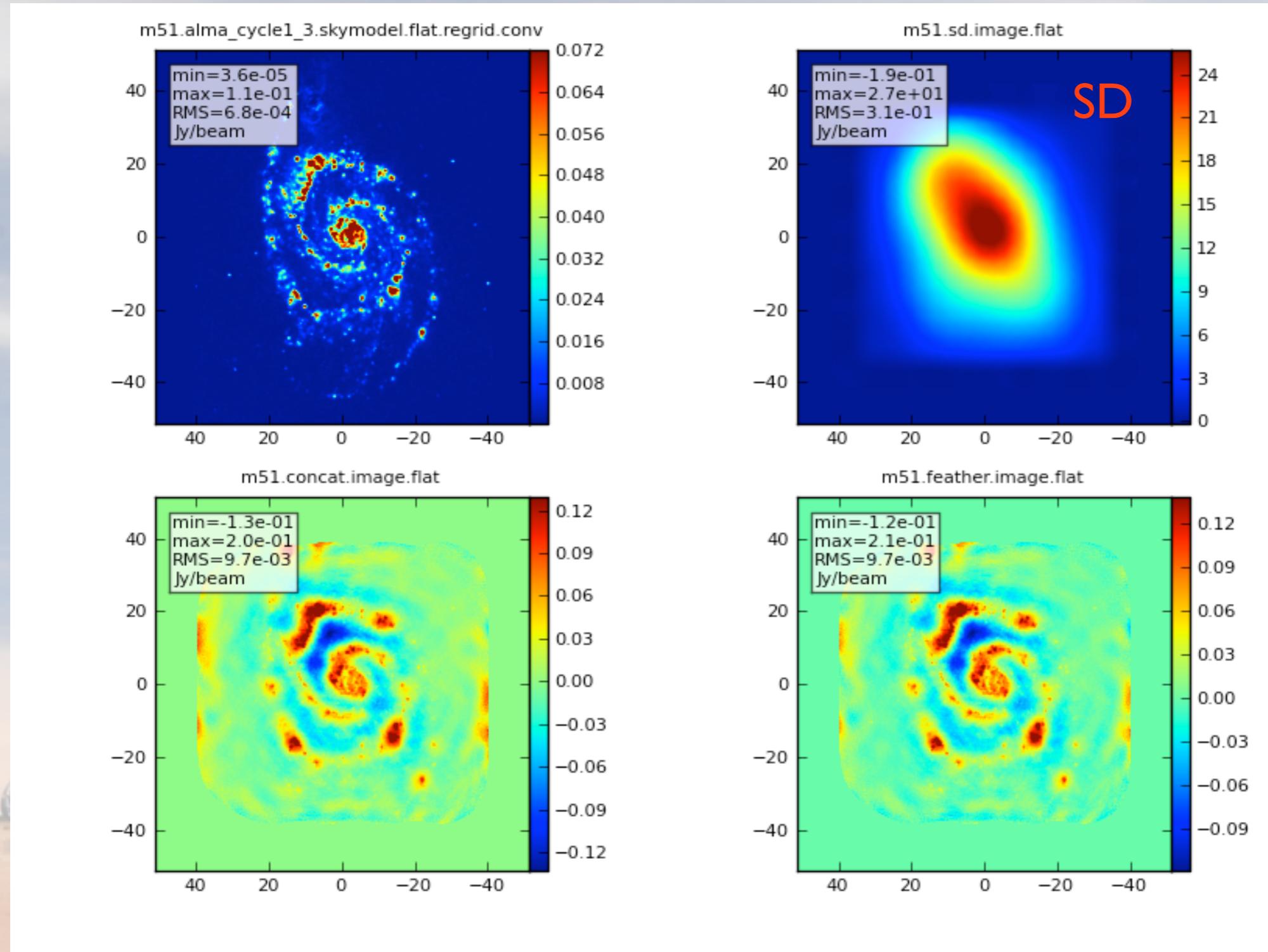
input model and pointing



M51 input

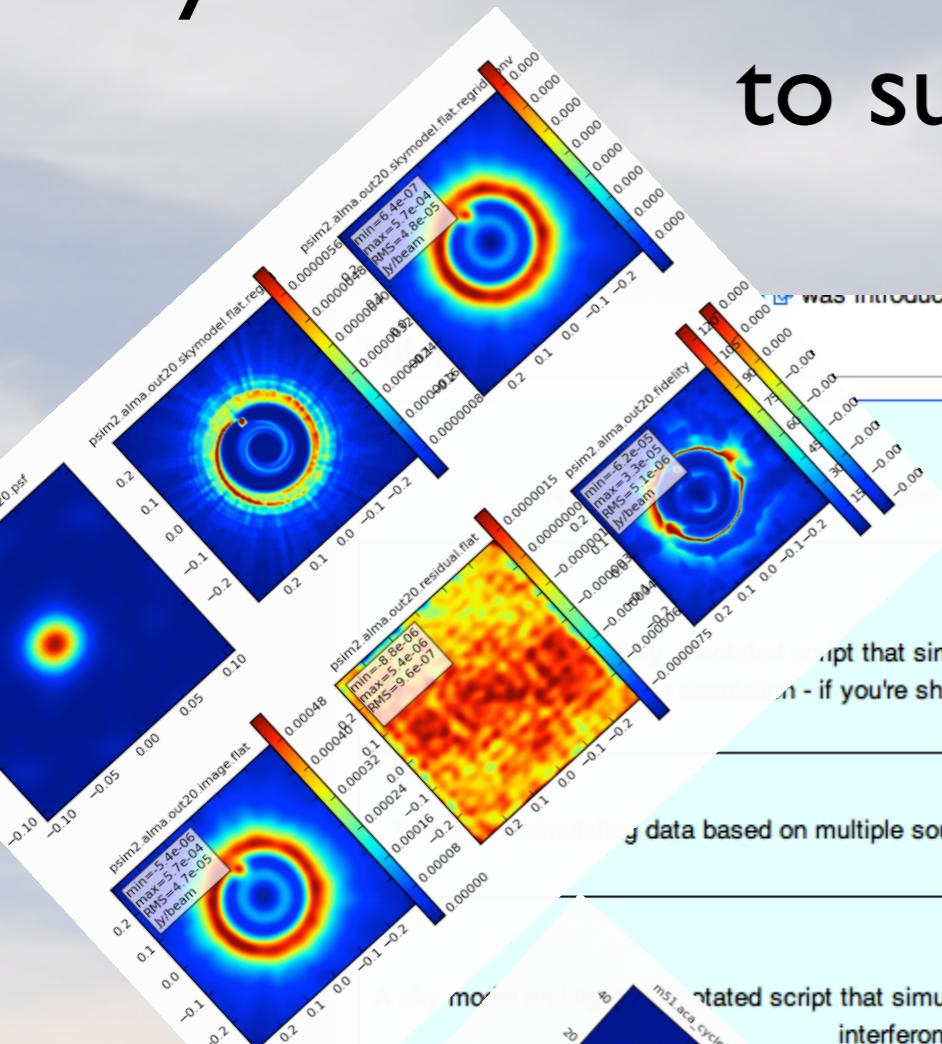


Simalma - combination of SD/ACA/ALMA during de-convolution



Useful examples

Try out the simulators and have fun using the simulations to support your proposals !



Simulation Guide for New Users (CASA 4.3)

A Spitzer SAGE 8 micron continuum image of 30 Doradus and scales it to greater resolution.

Protoplanetary Disk Simulation (CASA 4.3)

A script that simulates a protoplanetary disk. Uses a theoretical model of dust continuum from Sebastian Viala et al. - if you're short on time, you probably don't need to go through this one and the New Users guide, but it can be useful.

Simulation Guide Component Lists (CASA 4.3)

Simulating data based on multiple sources (using both a FITS image and a component list). If you are interested in simulating from a list of sources rather than or in addition to a sky model image, then read the considerations here.

Einstein-Face (CASA 4.3)

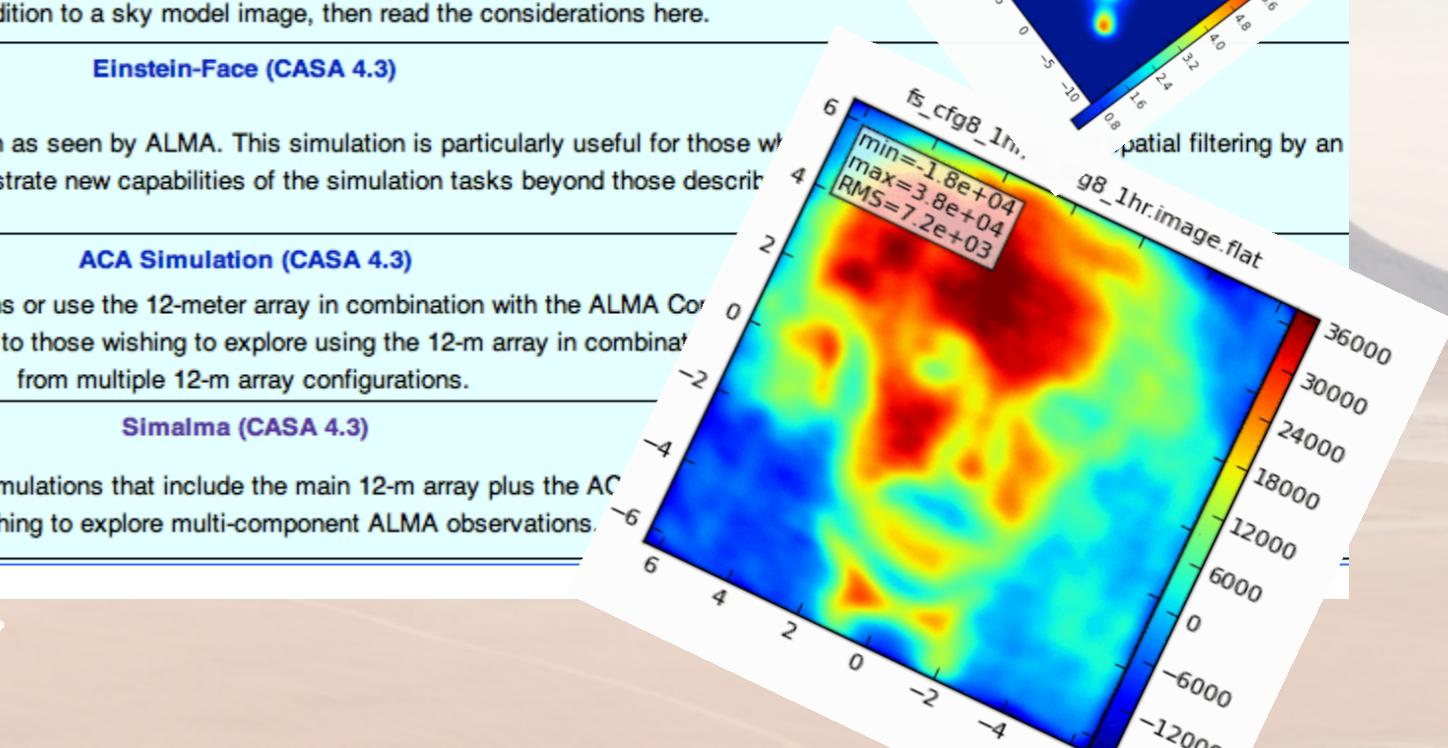
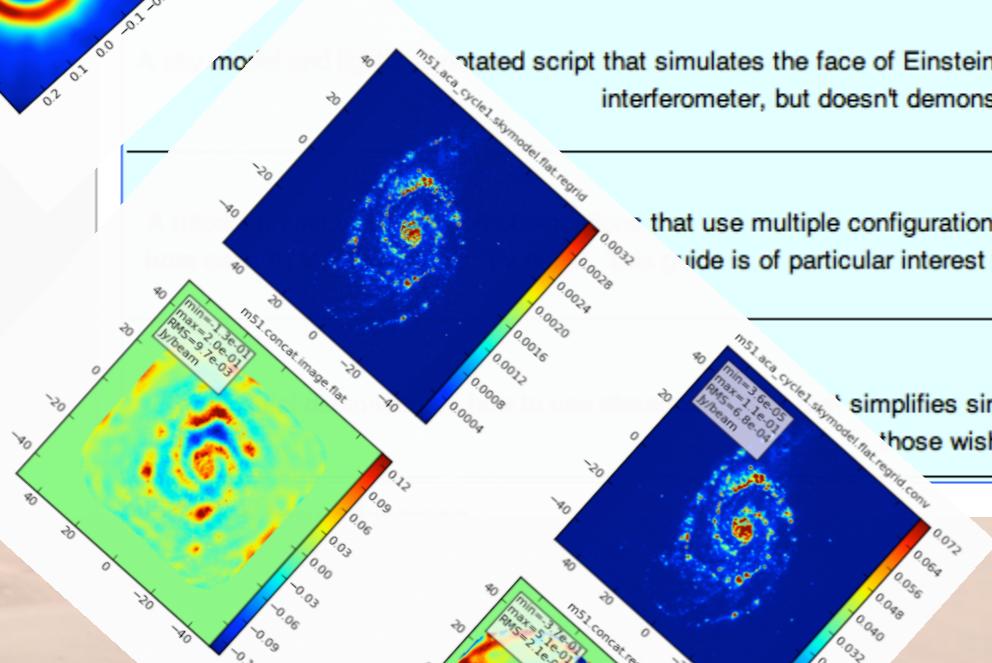
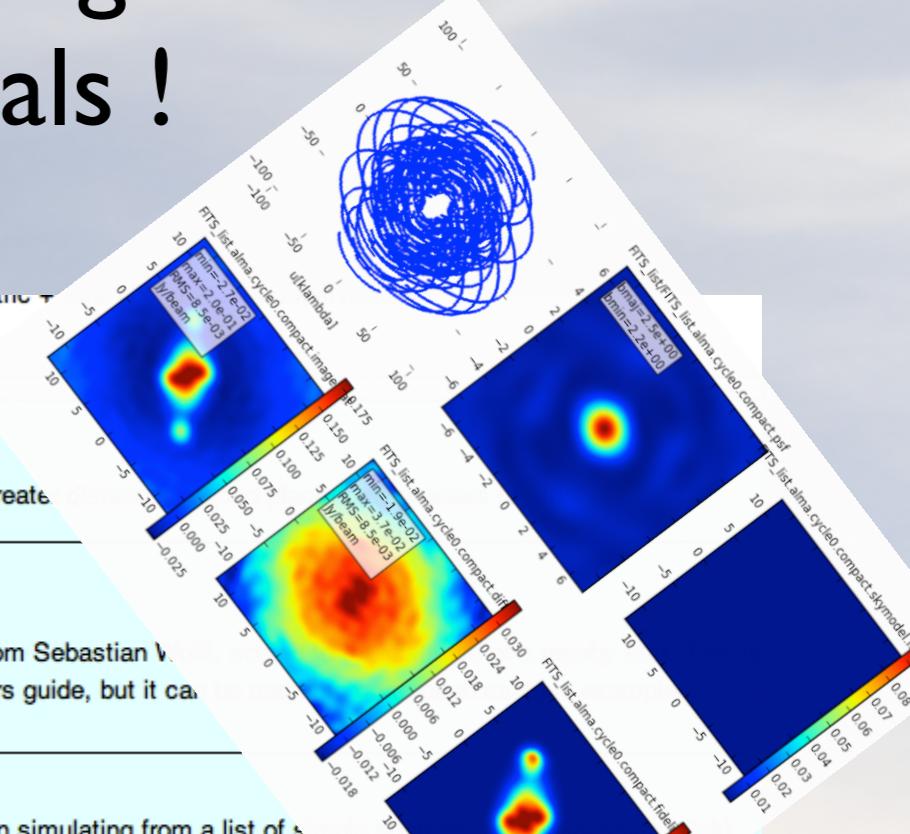
A rotated script that simulates the face of Einstein as seen by ALMA. This simulation is particularly useful for those who want to use the 12-m array as an interferometer, but doesn't demonstrate new capabilities of the simulation tasks beyond those described in the New Users guide.

ACA Simulation (CASA 4.3)

Simulations that use multiple configurations or use the 12-meter array in combination with the ALMA Compact Array. This guide is of particular interest to those wishing to explore using the 12-m array in combination with the ACA to obtain observations from multiple 12-m array configurations.

Simalma (CASA 4.3)

A simplified simulation task that simplifies simulations that include the main 12-m array plus the ACA. It is of particular interest to those wishing to explore multi-component ALMA observations.



more complicated sims

Use the simulator tool for:

phase delay variations

gain fluctuations and drift

cross-polarization

(soon also bandpass and pointing errors)

more flexibility in adding thermal noise

Thermal noise addition

<https://safe.nrao.edu/wiki/pub/ALMA/SimulatorCookbook/corruptguide.pdf>

simple: specify **simplenoise="1Jy"** to get random Gaussian noise with 1Jy RMS

- **tsys-atm:** use environment temperatures, antenna parameters, and the aatm library to create a model of the troposphere and add random noise of the appropriate magnitude to the visibilities
- **tsys-manual:** specify atmospheric brightness temperature and optical depth yourself (rather than let aatm calculate it for you) and apply noise of the corresponding magnitude.