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

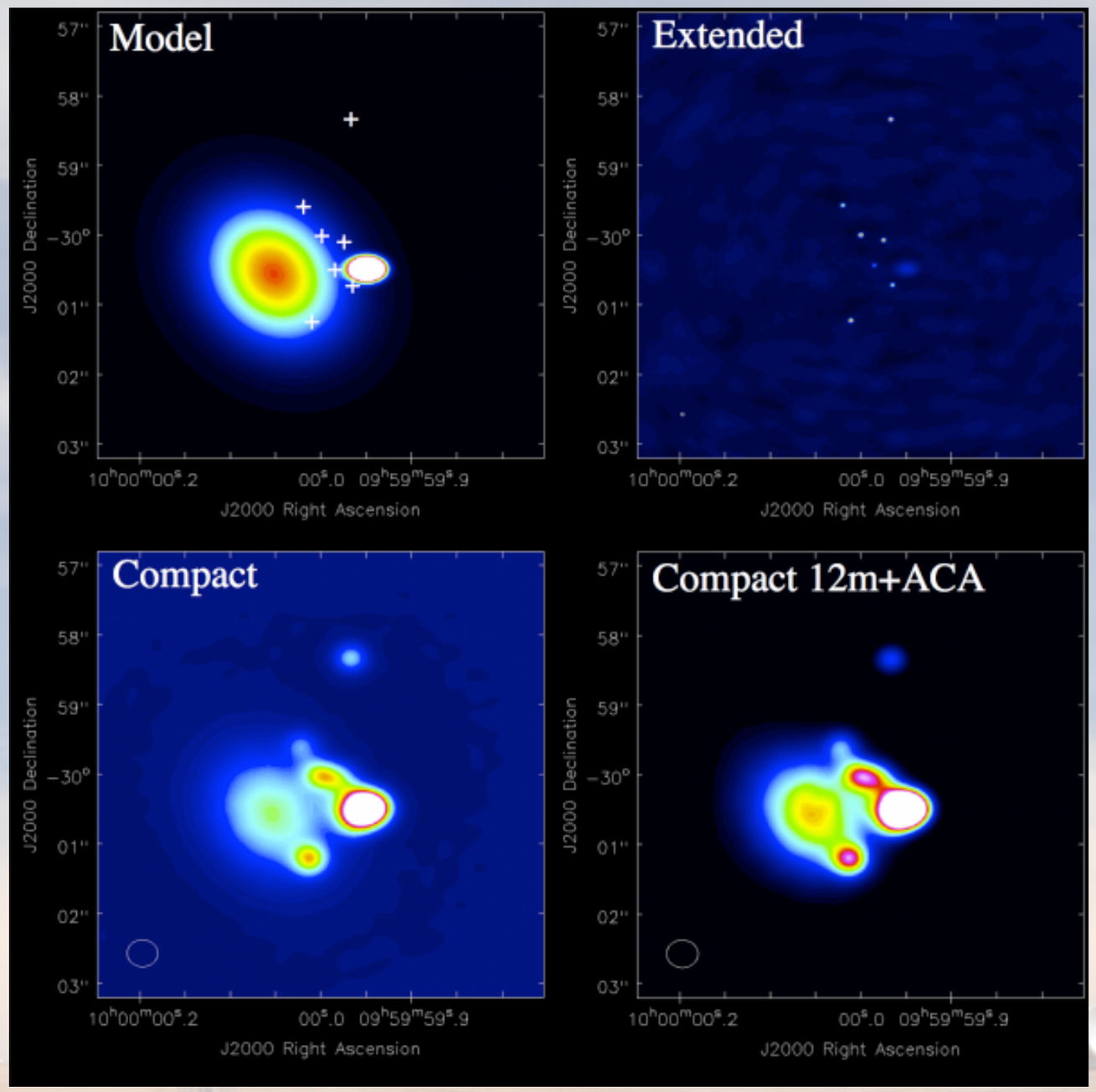
Why should I do an ALMA simulation ?

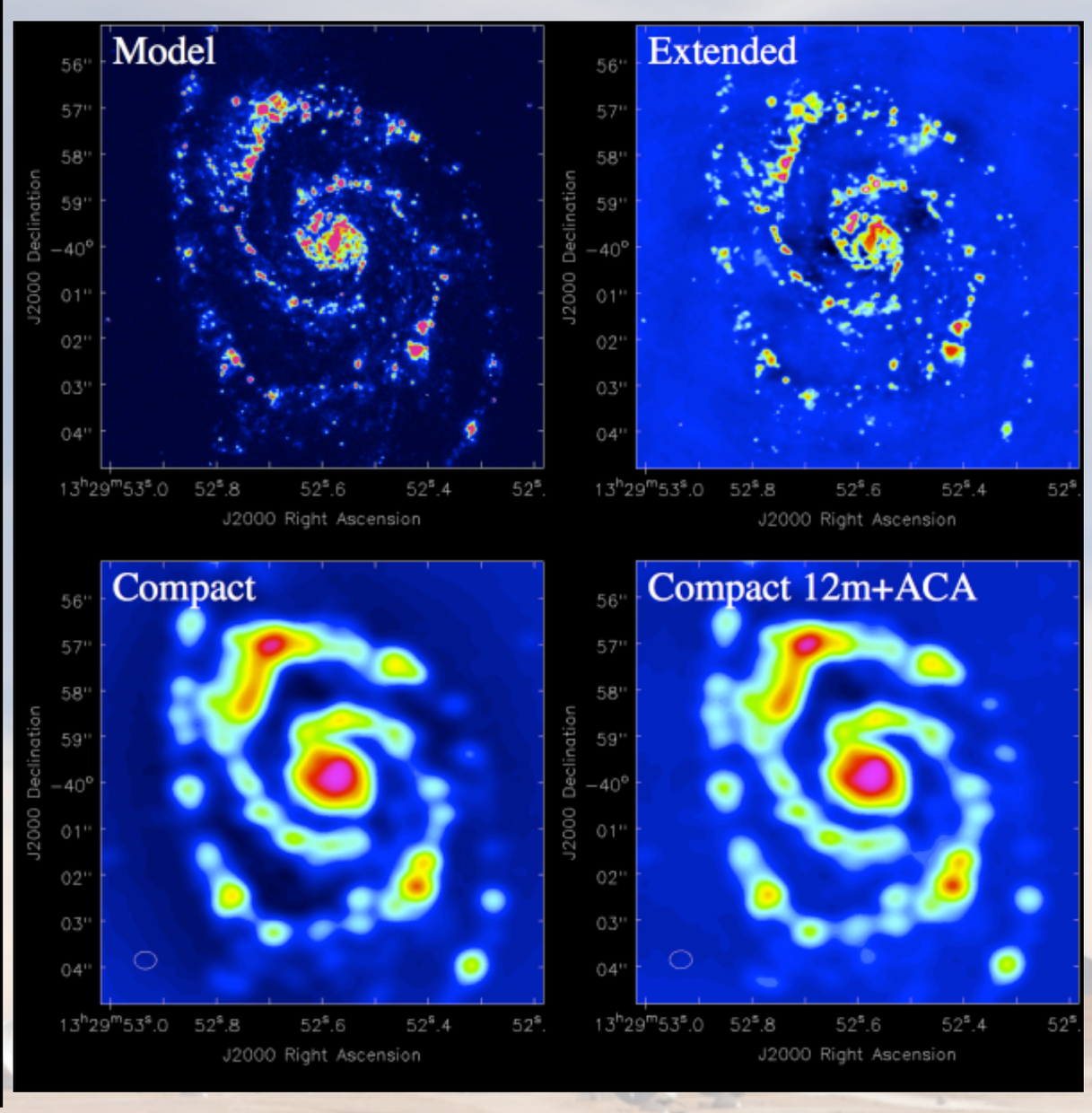
- You can try out different ALMA configurations and, for a given model, this might help you decide on the required angular resolution and maximum angular scale.

- 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

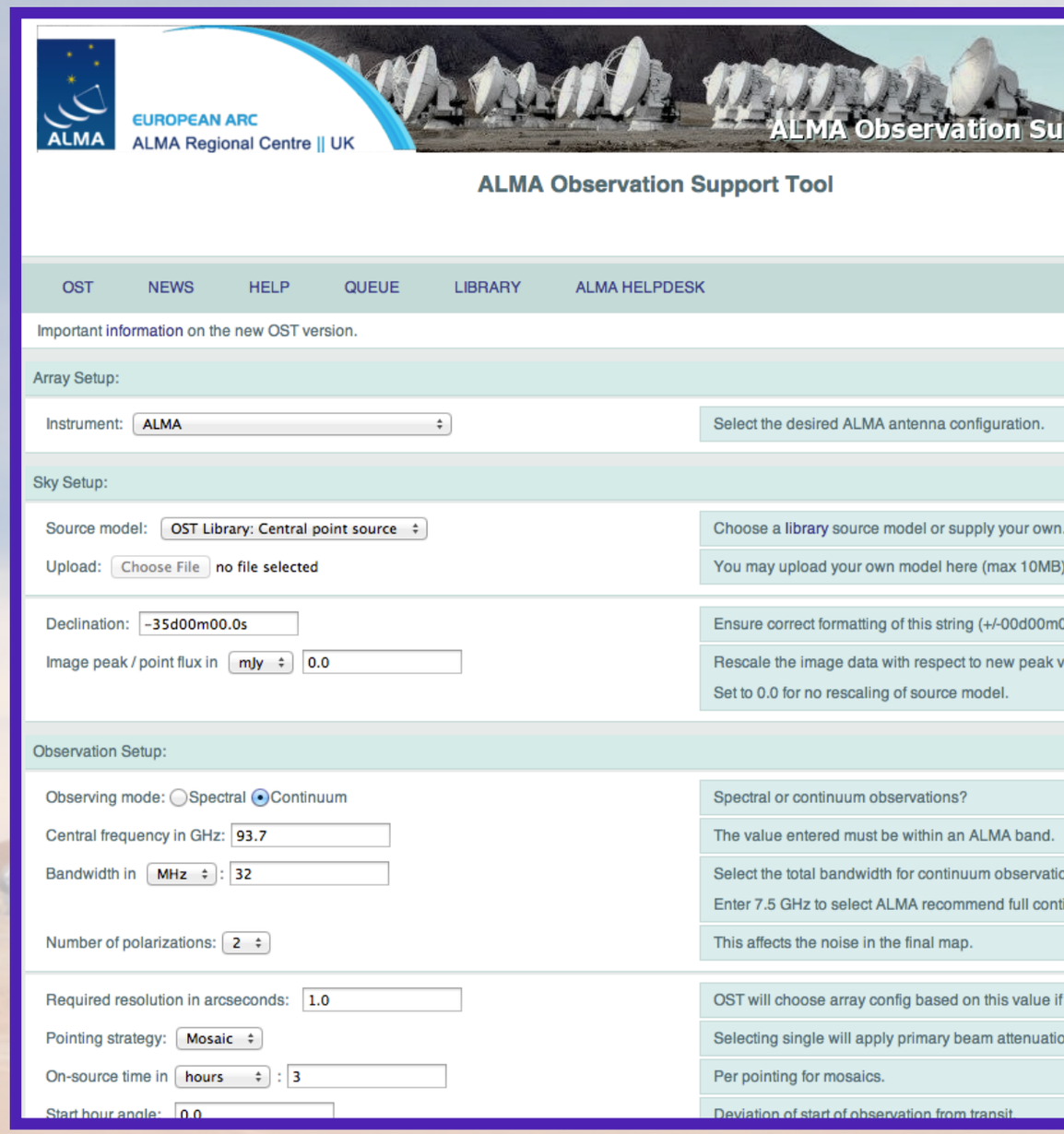
Available simulators

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    =     '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        =         ''                m|tsys-manual|""]
leakage             =        0.0                rometer only)
graphics            =      'both'               age to [screen|file|both|none]
verbose             =      False
overwrite           =       True               th $project
async               =      False               e started using simobserve(...)
```

CASA
Common Astronomy Software Applications

ALMA Observation Support tool



CASA simulator and cycle 7

Model image

simobserve

images, diagnostic

your proposal

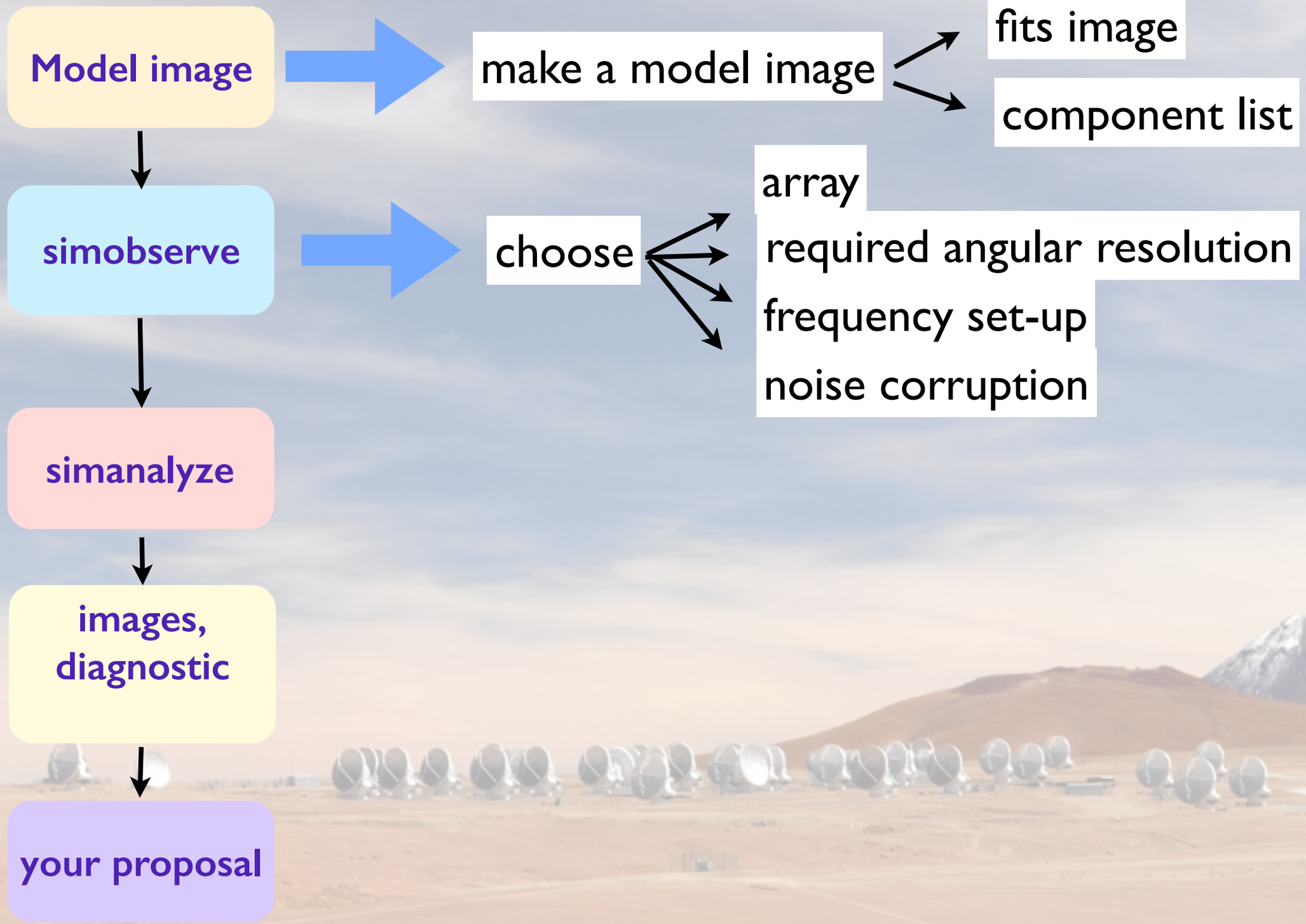
make a model image

fits image

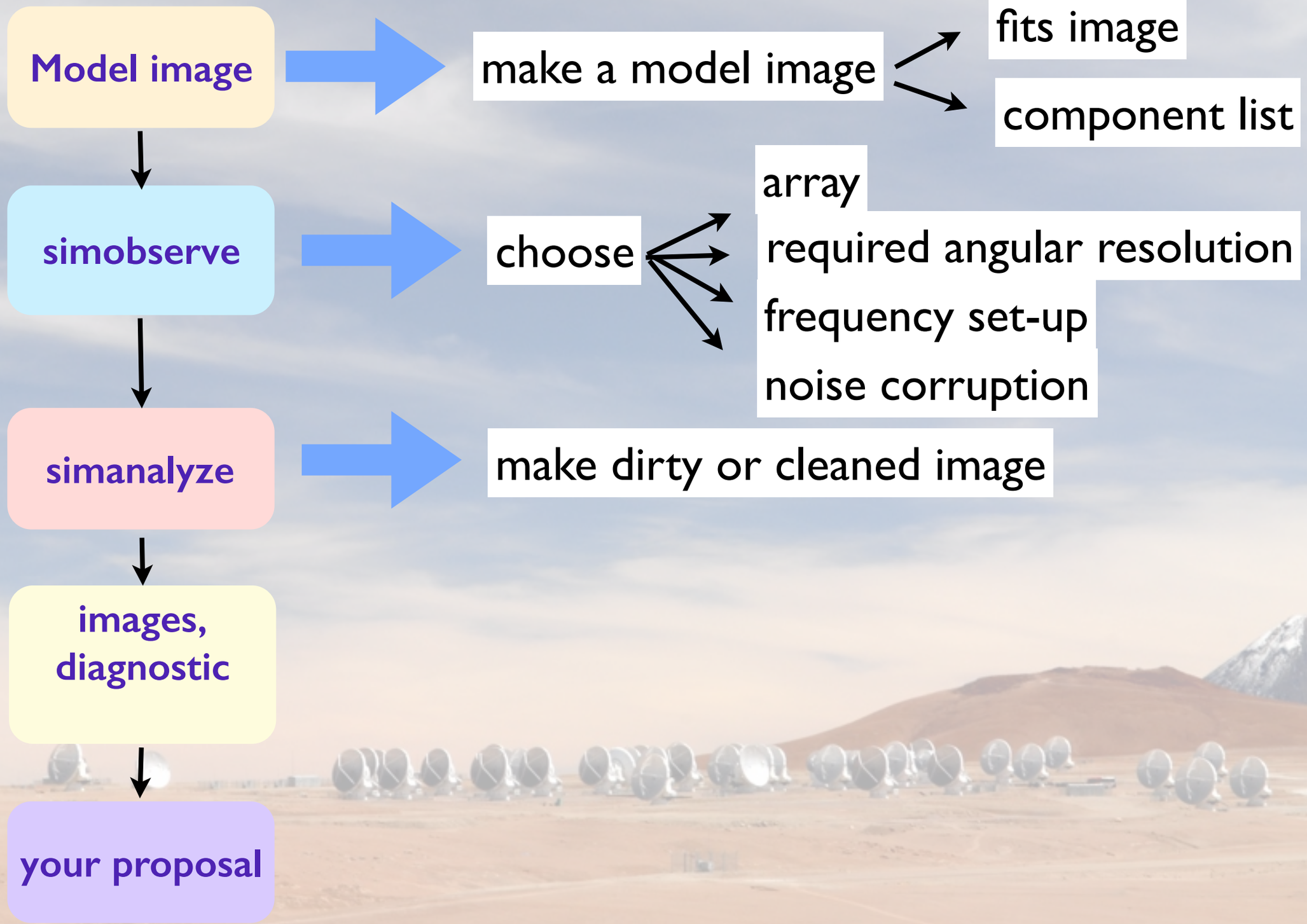
also cubes
[RA, DEC,Stokes, spectral]

component list

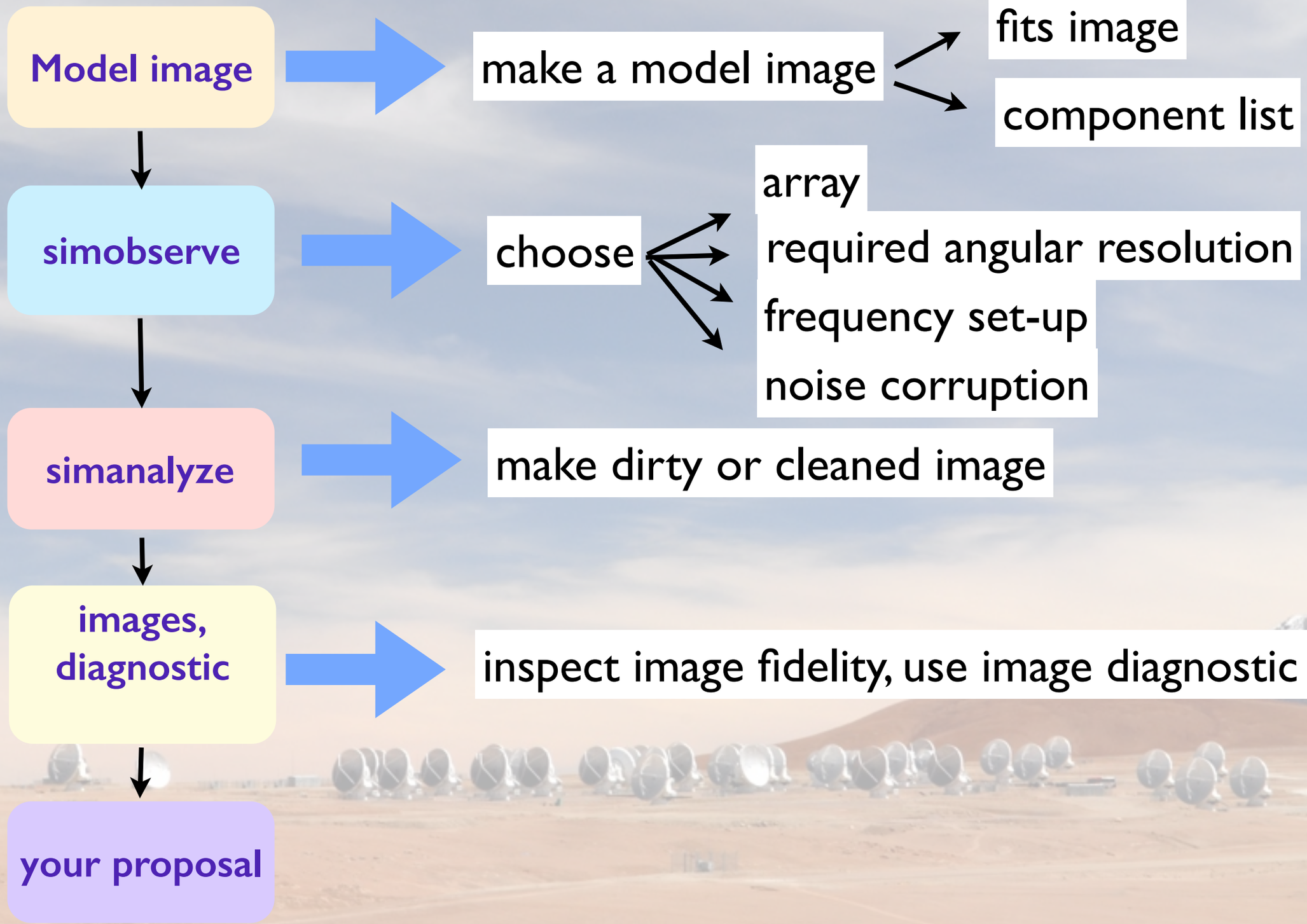
CASA simulator and cycle 7



CASA simulator and cycle 7



CASA simulator and cycle 7



CASA simulator and cycle 7

Model image → make a model image → fits image, component list

simobserve → choose → array, required angular resolution, frequency set-up, noise corruption

simanalyze → make dirty or cleaned image

images, diagnostic → inspect image fidelity, use image diagnostic tools

your proposal → use the mock simulation to support your proposal

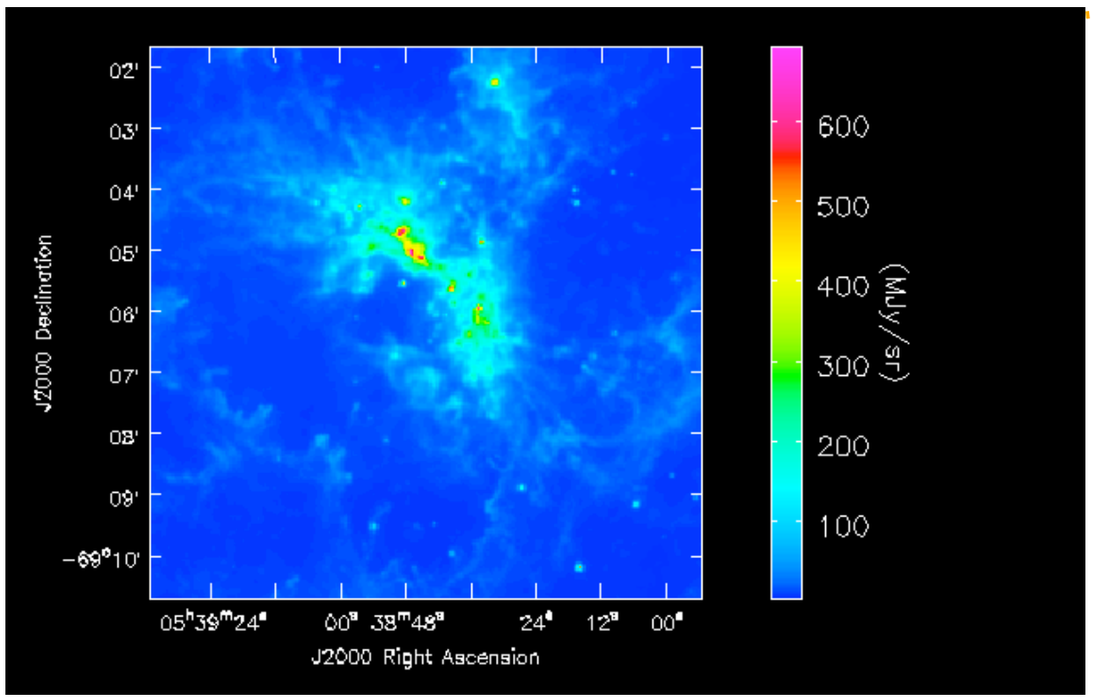
Simobserve

```
#  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 fo
                                        #   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 fo
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                =                ''    #  componentlis
```

Model image



Spitzer IRAC 89 micron image of 30 Doradus

https://casaguides.nrao.edu/index.php/Simulation_Guide_for_New_Users_(CASA_4.4)

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"`

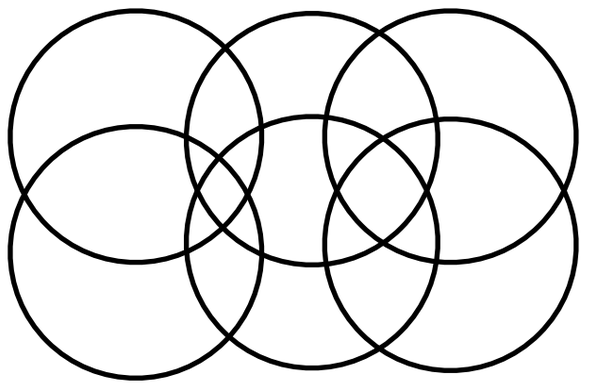
(only works for full ALMA !)

Simobserve

```
    incenter         =    '230GHz'

    inwidth          =     '2GHz'
```

spectral set-up

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



observing time and pointing

the simulator allows you to test
several pointing scenarios

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_pwv        =         1.0
     t_ground        =       269.0
     seed            =       11111

leakage              =         0.0
graphics             =      'both'
verbose              =       False
overwrite            =        True
async                =       False
```

model image

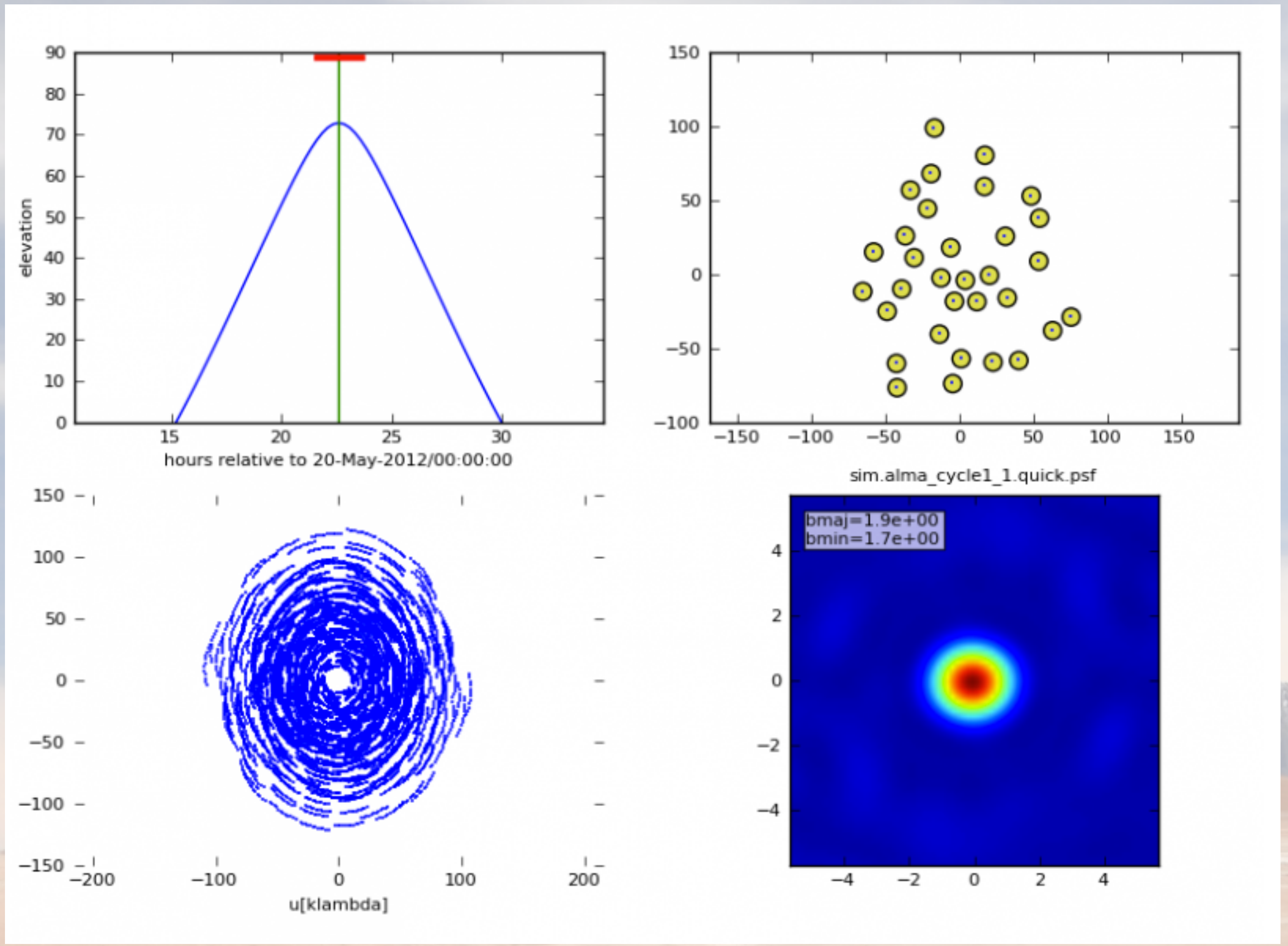
spectral set-up

observing time and pointing

array choice or required angular resolution

noise addition

Simobserve



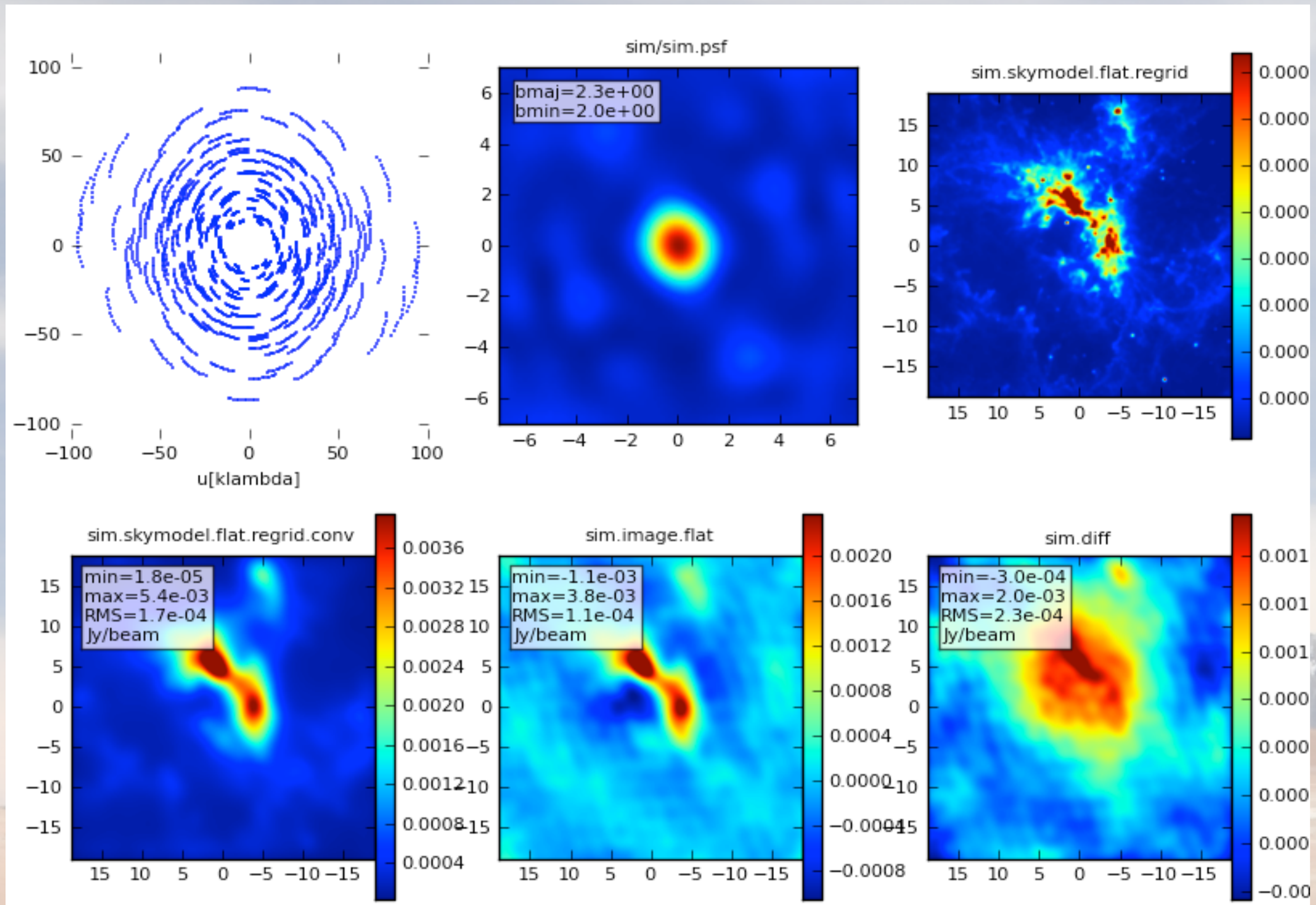
https://casaguides.nrao.edu/index.php/Simulation_Guide_for_New_Users_(CASA_4.4)

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
overwrite           =        True        #  overwrite files starting with $project
async               =       False        #  If true the taskname must be started using simanalyze(...)
```

Simanalyze imaging output

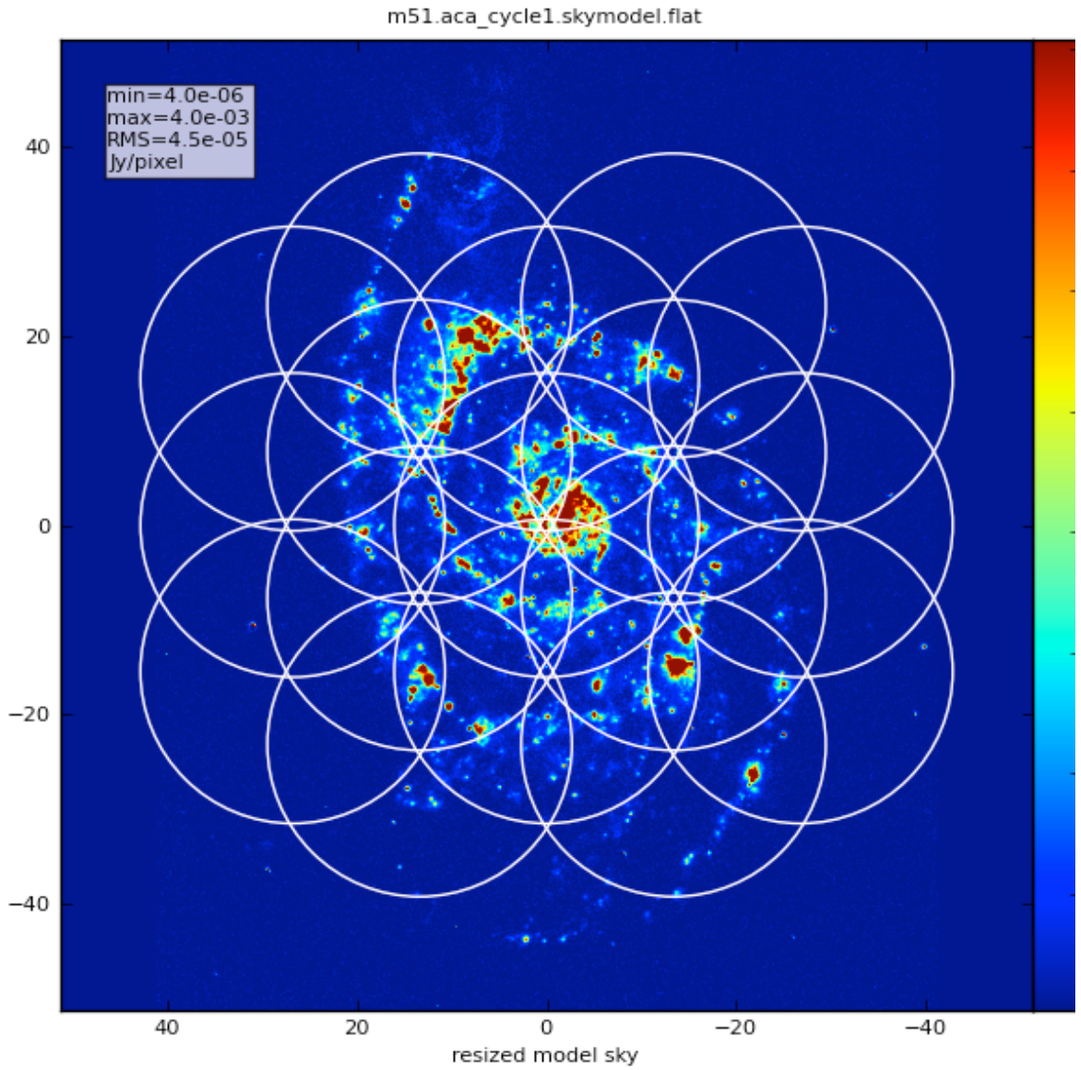


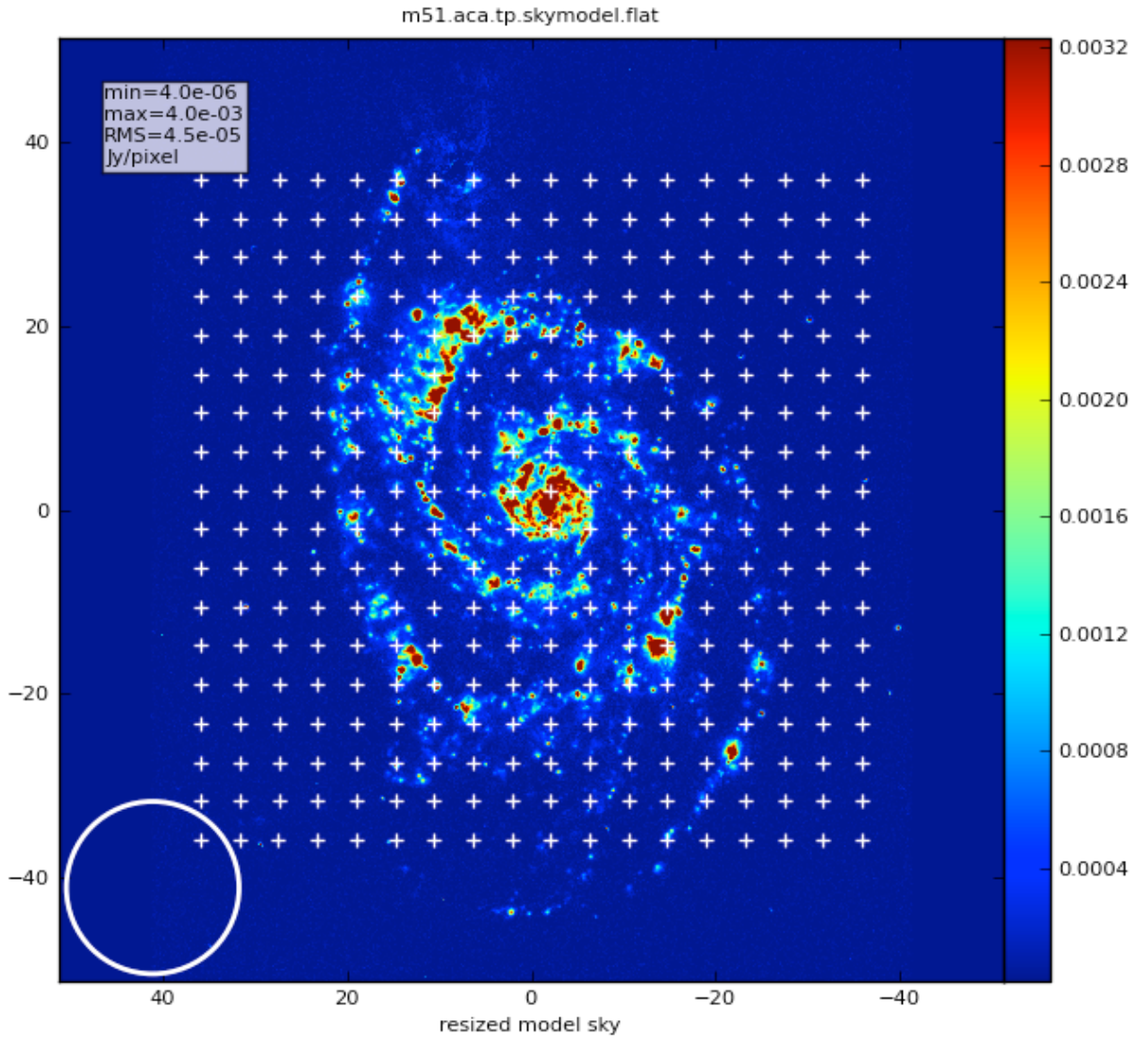
https://casaguides.nrao.edu/index.php/Simulation_Guide_for_New_Users_(CASA_4.4)

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

input model and pointing

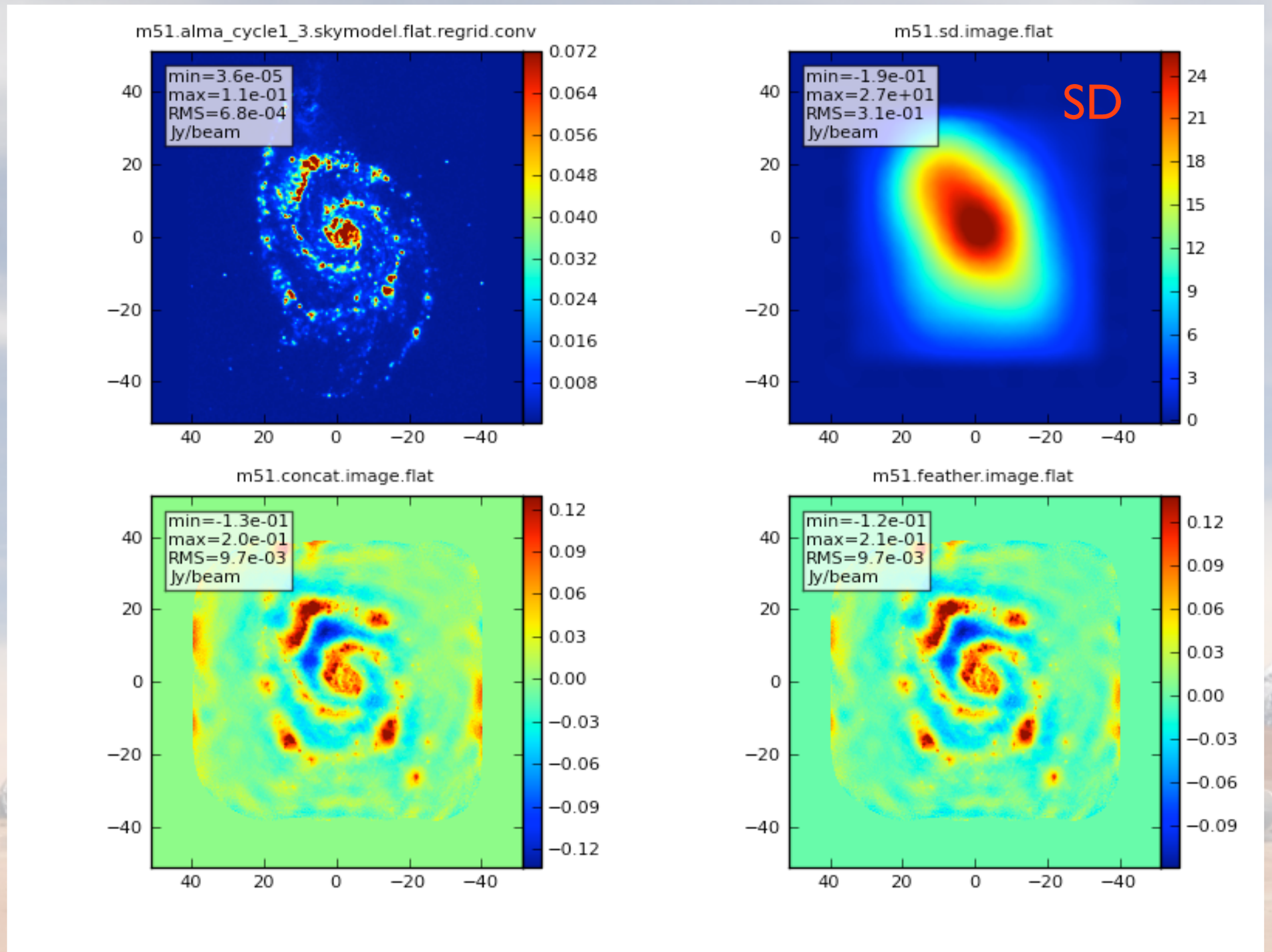
M51 input





https://casaguides.nrao.edu/index.php/Simalma_(CASA_4.4)

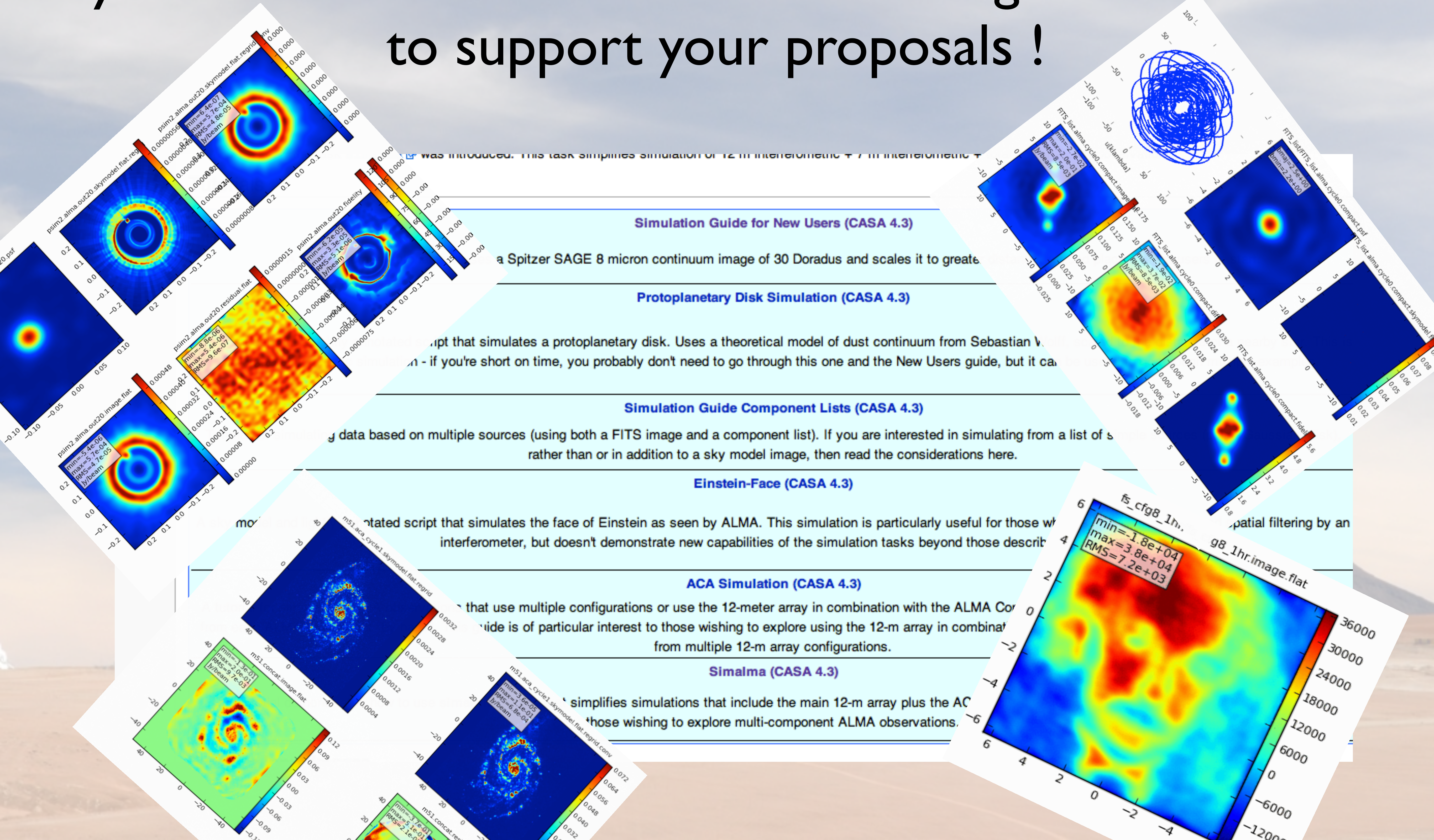
Simalma - combination of SD/ACA/ALMA during de-convolution



https://casaguides.nrao.edu/index.php/Simalma_(CASA_4.4)

Useful examples

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



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.