ALMA Users Workshop 2015, (March 28, 2015)

ALMA Observing Simulations with CASA

Kuo-Song Wang
and
ARC Taiwan team

download at
www.asiaa.sinica.edu.tw/~kswang/casa_sim.zip

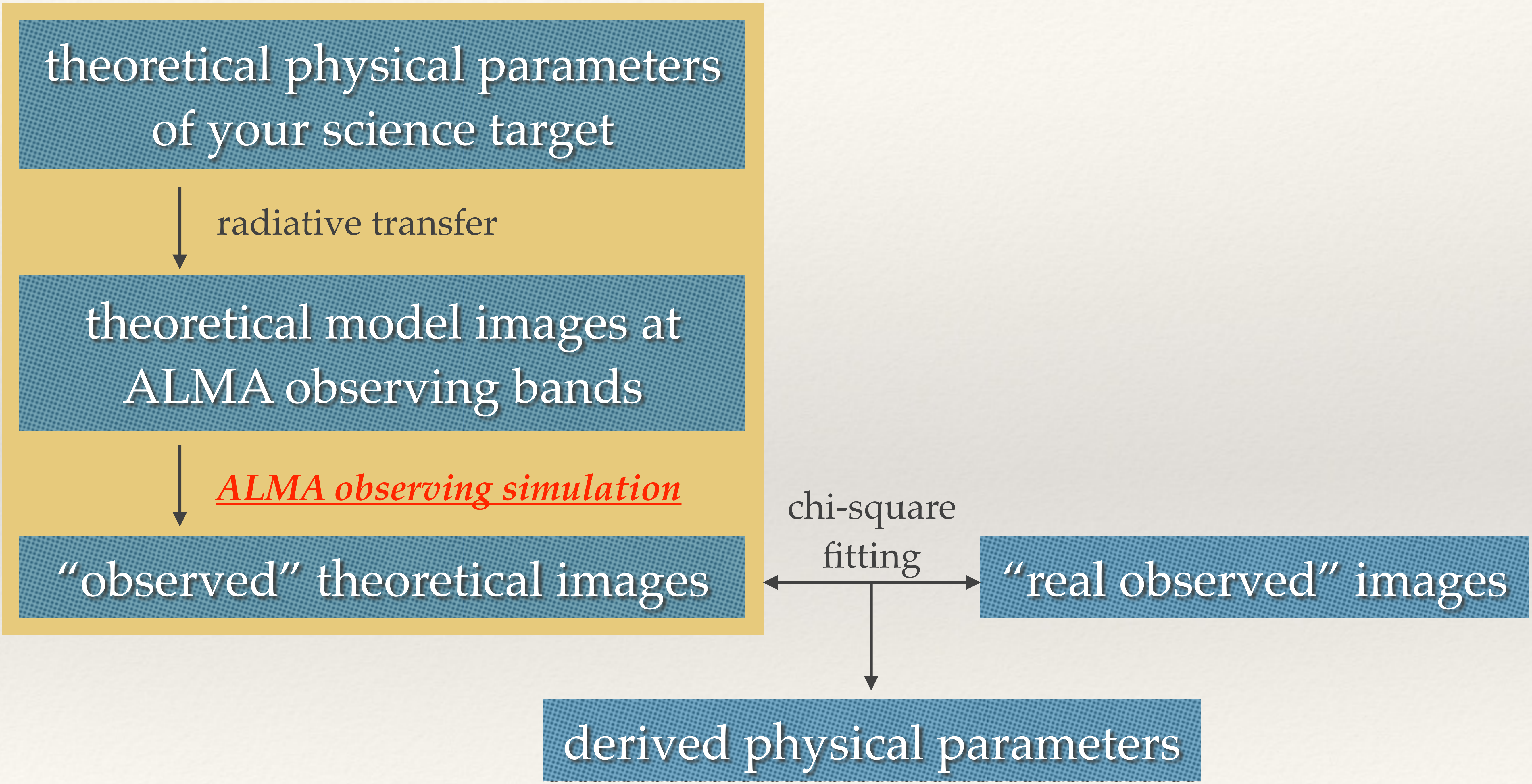
Why doing ALMA observing simulations?

- Preparing for your ALMA proposals
  - Can your proposed ALMA observations provide you what you want to see?
- Extracting physical parameters from your real ALMA data
  - Comparing physical models with ALMA data directly

ALMA components

- ALMA 12m array:
  - fifty 12m antennae (≥ 36 for cycle3)
  - *providing high-resolution and high-sensitivity images*
- Atacama Compact Array (ACA):
  - 7m array: twelve 7m antennae (10 for cycle3)
  - Total Power (TP) array: Four 12m antennae (2 for cycle3)
  - *imaging extended structures*

ALMA Observing Simulation



Interferometric observations

- What we can obtain from an interferometric observation are Visibilities ($V_{\mathrm{obs}}(u,v)$), i.e., Fourier-transformation of source images $I(x,y)$, sampled by the sampling function $S(u,v)$.

$$V_{\mathrm{obs}}(u,v) = S(u,v) \iint I(x,y) e^{-2\pi i(ux+vy)} dx dy$$

S(u,v): antenna tracks projected on the sky

$V(u,v)$ consists of *amplitude* and *phase*.

- Image reconstruction: $I_{\mathrm{obs}}{}^{\mathrm{dirty}}(x,y) = FT(V_{\mathrm{obs}}(u,v))$, i.e., dirty image.

- $I_{\mathrm{obs}}{}^{\mathrm{dirty}}(x,y)$ —CLEAN algorithms→ $I_{\mathrm{obs}}{}^{\mathrm{clean}}(x,y)$

Fidelity

- How to evaluate the imaging quantitatively?
- At each image pixel (i,j):

$$\text{Fidelity}(i,j) = \frac{|\,model(i,j)\,|}{|\,model(i,j) - simulated(i,j)\,|}$$

Pety et al. 2001

The fidelity image helps to identify which parts of your simulated image are better/poorly "observed" by the interferometer.

ALMA Observing Simulator

- ALMA Observation Support Tool (OST):
  - http://almaost.jb.man.ac.uk
- CASA 4.3.1
  - simobserve: create model visibilities from input sky image (cube) with user-specified observing parameters (frequency, mapping area, antenna configuration, etc.)
  - simanalyze: Fourier transform the model visibilities, CLEAN the image (cube) with user-specified imaging parameters (image size, weighting, etc.), and compute the Fidelity image for diagnostics.
  - simalma: a wrapper of simobserve and simanalyze (with less user control)

Useful links

- A web application of radio interferometer (VRI)
  - http://www.narrabri.atnf.csiro.au/astronomy/vri.html
- Guide to simulate ALMA observations
  - http://casaguides.nrao.edu/index.php?title=Guide_To_Simulating_ALMA_Data
  - http://casaguides.nrao.edu/index.php?title=Simulation_Guide_for_New_Users_(CASA_4.3)
  - http://casaguides.nrao.edu/index.php?title=Simulating_Observations_in_CASA_4.3
  - http://casaguides.nrao.edu/index.php?title=Simalma_(CASA_4.3)
- Antenna configuration (including the ones for cycle 3)
  - http://casaguides.nrao.edu/index.php?title=Antenna_Configurations_Models_in_CASA

Useful links

- Guide to create/download sky model images
  - http://casaguides.nrao.edu/index.php?title=Simulation_Guide_Component_Lists_%28CASA_4.1%29
  - http://casaguides.nrao.edu/index.php?title=Convert_jpg_to_fits
  - http://casaguides.nrao.edu/index.php?title=Sim_Inputs
- Guide to add noise to simulated observations (advanced)
- http://casaguides.nrao.edu/index.php?title=Corrupt
- https://safe.nrao.edu/wiki/pub/ALMA/SimulatorCookbook/corruptguide.pdf

CASA basics

- CASA 4.3.1
- >casapy — to start CASA
- >default(simobserve) — reset all control parameters of the task
- >inp — to see what control parameters can be modified
- >project = 'sim_cycle3' — to modify a control parameter
- >go — execute the task (remember to "inp" and check if everything is okay)
- >help simobserve — get help of the task

inp simobserve

```
#  simobserve :: visibility simulation task
project             =      'sim'        #  root prefix for output file names
skymodel            = 'M51ha.fits'      #  model image to observe
     inbright       =         ''        #  scale surface brightness of brightest
                                        #   pixel e.g. "1.2Jy/pixel"
     indirection    =         ''        #  set new direction e.g. "J2000
                                        #   19h00m00 -40d00m00"
     incell         =         ''        #  set new cell/pixel size e.g.
                                        #   "0.1arcsec"
     incenter       =         ''        #  set new frequency of center channel
                                        #   e.g. "89GHz" (required even for 2D
                                        #   model)
     inwidth        =         ''        #  set new channel width e.g. "10MHz"
                                        #   (required even for 2D model)

complist            =         ''        #  componentlist to observe
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        =     'ALMA'        #  hexagonal, square (raster), ALMA, etc
     pointingspacing =         ''       #  spacing in between pointings or
                                        #   "0.25PB" or "" for ALMA default
                                        #   INT=lambda/D/sqrt(3), SD=lambda/D/3

obsmode             =      'int'        #  observation mode to simulate [int(int
                                        #   erferometer)|sd(singledish)|""(none)
                                        #   ]
     antennalist    = 'alma.out10.cfg'  #  interferometer antenna position file
     refdate        = '2014/05/21'      #  date of observation - not critical
                                        #   unless concatting simulations
     hourangle      =  'transit'        #  hour angle of observation center e.g.
                                        #   "-3:00:00", "5h", "-4.5" (a number
                                        #   without units will be interpreted as
                                        #   hours), or "transit"
     totaltime      =    '7200s'        #  total time of observation or number
                                        #   of repetitions
     caldirection   =         ''        #  pt source calibrator [experimental]
     calflux        =      '1Jy'

thermalnoise        = 'tsys-atm'        #  add thermal noise: [tsys-atm|tsys-
                                        #   manual|""]
     user_pwv       =        0.5        #  Precipitable Water Vapor in mm
     t_ground       =      269.0        #  ambient temperature
     seed           =      11111        #  random number seed

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
```

simobserve

```
project          =       'sim'        #  root prefix for output file names
skymodel         = 'M51ha.fits'       #  model image to observe
     inbright    =          ''        #  scale surface brightness of brightest
                                      #   pixel e.g. "1.2Jy/pixel"
     indirection =          ''        #  set new direction e.g. "J2000
                                      #   19h00m00 -40d00m00"
     incell      =          ''        #  set new cell/pixel size e.g.
                                      #   "0.1arcsec"
     incenter    =          ''        #  set new frequency of center channel
                                      #   e.g. "89GHz" (required even for 2D
                                      #   model)
     inwidth     =          ''        #  set new channel width e.g. "10MHz"
                                      #   (required even for 2D model)
```

project: name of the output directory, all output files/directories will be put in
skymodel: a fits file of your sky model (can be image cube as well)
inbright: scale the brightness of the brightest pixel
indirection: a new sky position of your sky model
incell: a new pixel size (useful to scale your sky model to a new distance)
incenter: a new reference frequency at center channel
inwidth: a new channel width

simobserve

```
complist            =          ''          #  componentlist to observe
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        =       'ALMA'         #  hexagonal, square (raster), ALMA, etc
     pointingspacing =          ''         #  spacing in between pointings or
                                           #   "0.25PB" or "" for ALMA default
                                           #   INT=lambda/D/sqrt(3), SD=lambda/D/3
```

complist: a file to generate sky model if no user input sky model
integration: integration time per pointing (not total observing time!), set a longer time (e.g., 600s) to speed up the simulation if necessary
direction: center position of the observation (mosaic center)
mapsize: area to cover in the observation
maptype: how the mosaic is done
pointingspacing: user-specified spacing for mosaic observation

simobserve

```
obsmode             =       'int'       #  observation mode to simulate [int(int
                                        #   erferometer)|sd(singledish)|""(none)
                                        #   ]
     antennalist    = 'alma.out10.cfg'  #  interferometer antenna position file
     refdate        = '2014/05/21'      #  date of observation - not critical
                                        #   unless concatting simulations
     hourangle      =  'transit'        #  hour angle of observation center e.g.
                                        #   "-3:00:00", "5h", "-4.5" (a number
                                        #   without units will be interpreted as
                                        #   hours), or "transit"
     totaltime      =    '7200s'        #  total time of observation or number
                                        #   of repetitions
     caldirection   =          ''       #  pt source calibrator [experimental]
     calflux        =       '1Jy'
```

obsmode: interferometer of single dish
antennalist: user-specified antenna position file (http://casaguides.nrao.edu/index.php?title=Antenna_Configurations_Models_in_CASA)
refdate: a date to simulate observation (important if multiple execution)
hourangle: hour angle of observation center
totaltime: total time of observation or number of repetitions

simobserve

```
thermalnoise        =  'tsys-atm'        #  add thermal noise: [tsys-atm|tsys-
                                         #   manual|""]
     user_pwv       =        0.5         #  Precipitable Water Vapor in mm
     t_ground       =      269.0         #  ambient temperature
     seed           =      11111         #  random number seed

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
```

thermalnoise: to add thermal noise to the simulated data

in our demo, we will use thermalnoise = “”, i.e., noise free, to just focus on the spatial filtering effect

Note: don’t quote the measured noise level from your simulated image! Use ALMA sensitivity calculator (https://almascience.eso.org/proposing/sensitivity-calculator) or ALMA OT

inp simanalyze

```
#  simanalyze :: image and analyze measurement sets created with simobserve
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     =         ''        #  lower resolution prior image to use
                                        #   in clean e.g. existing total power
                                        #   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 e.g. "10arcsec"
                                        #   or "" to equal model
     interactive    =      False        #  interactive clean?  (make sure to set
                                        #   niter>0 also)
     niter          =          0        #  maximum number of iterations (0 for
                                        #   dirty image)
     threshold      =   '0.1mJy'        #  flux level (+units) to stop cleaning
     weighting      =  'natural'        #  weighting to apply to visibilities.
                                        #   briggs will use robust=0.5
     mask           =         []        #  Cleanbox(es), mask image(s),
                                        #   region(s), or a level
     outertaper     =         []        #  uv-taper on outer baselines in uv-
                                        #   plane
     pbcor          =       True        #  correct the output of synthesis
                                        #   images for primary beam response?
     stokes         =        'I'        #  Stokes params to image
     featherimage   =         ''        #  image (e.g. total power) to feather
                                        #   with new 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 clean beam
     showclean      =       True        #  display the synthesized image
     showresidual   =      False        #  display the clean residual image
                                        #   (ignored in single dish simulation)
     showdifference =       True        #  display difference between output
                                        #   cleaned image and input model sky
                                        #   image convolved with output clean
                                        #   beam
     showfidelity   =       True        #  display fidelity (see help)

graphics            =     'both'        #  display graphics at each stage to
                                        #   [screen|file|both|none]
verbose             =      False
overwrite           =       True        #  overwrite files starting with
                                        #   $project
dryrun              =      False        #  only print information [experimental;
                                        #   only for interferometric data]
logfile             =         ''
```

simanalyze

```
#  simanalyze :: image and analyze measurement sets created with simobserve
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     =         ''        #  lower resolution prior image to use
                                        #   in clean e.g. existing total power
                                        #   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 e.g. "10arcsec"
                                        #   or "" to equal model
     interactive    =      False        #  interactive clean?  (make sure to set
                                        #   niter>0 also)
     niter          =          0        #  maximum number of iterations (0 for
                                        #   dirty image)
     threshold      =   '0.1mJy'        #  flux level (+units) to stop cleaning
     weighting      =  'natural'        #  weighting to apply to visibilities.
                                        #   briggs will use robust=0.5
     mask           =         []        #  Cleanbox(es), mask image(s),
                                        #   region(s), or a level
     outertaper     =         []        #  uv-taper on outer baselines in uv-
                                        #   plane
     pbcor          =       True        #  correct the output of synthesis
                                        #   images for primary beam response?
     stokes         =        'I'        #  Stokes params to image
     featherimage   =         ''        #  image (e.g. total power) to feather
                                        #   with new image
```

project: root prefix for output file name, and the directory to look for measurement set
vis: input measurement set
modelimage: lower resolution prior image to use in clean
imsize: image size of the reconstructed image
cell: pixel size of the reconstructed image
niter: maximun iteration number in clean
threshold: when to stop clean process

simanalyze

```
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 clean beam
     showclean      =        True          #  display the synthesized image
     showresidual   =       False          #  display the clean residual image
                                           #   (ignored in single dish simulation)
     showdifference =        True          #  display difference between output
                                           #   cleaned image and input model sky
                                           #   image convolved with output clean
                                           #   beam
     showfidelity   =        True          #  display fidelity (see help)

graphics            =      'both'          #  display graphics at each stage to
                                           #   [screen|file|both|none]
verbose             =       False
overwrite           =        True          #  overwrite files starting with
                                           #   $project
dryrun              =       False          #  only print information [experimental;
                                           #   only for interfermetric data]
logfile             =          ''
```

analyze: to enable diagnostics plots
dryrun: only print information (no clean happened; clean can be time consuming!), inteferometric data only

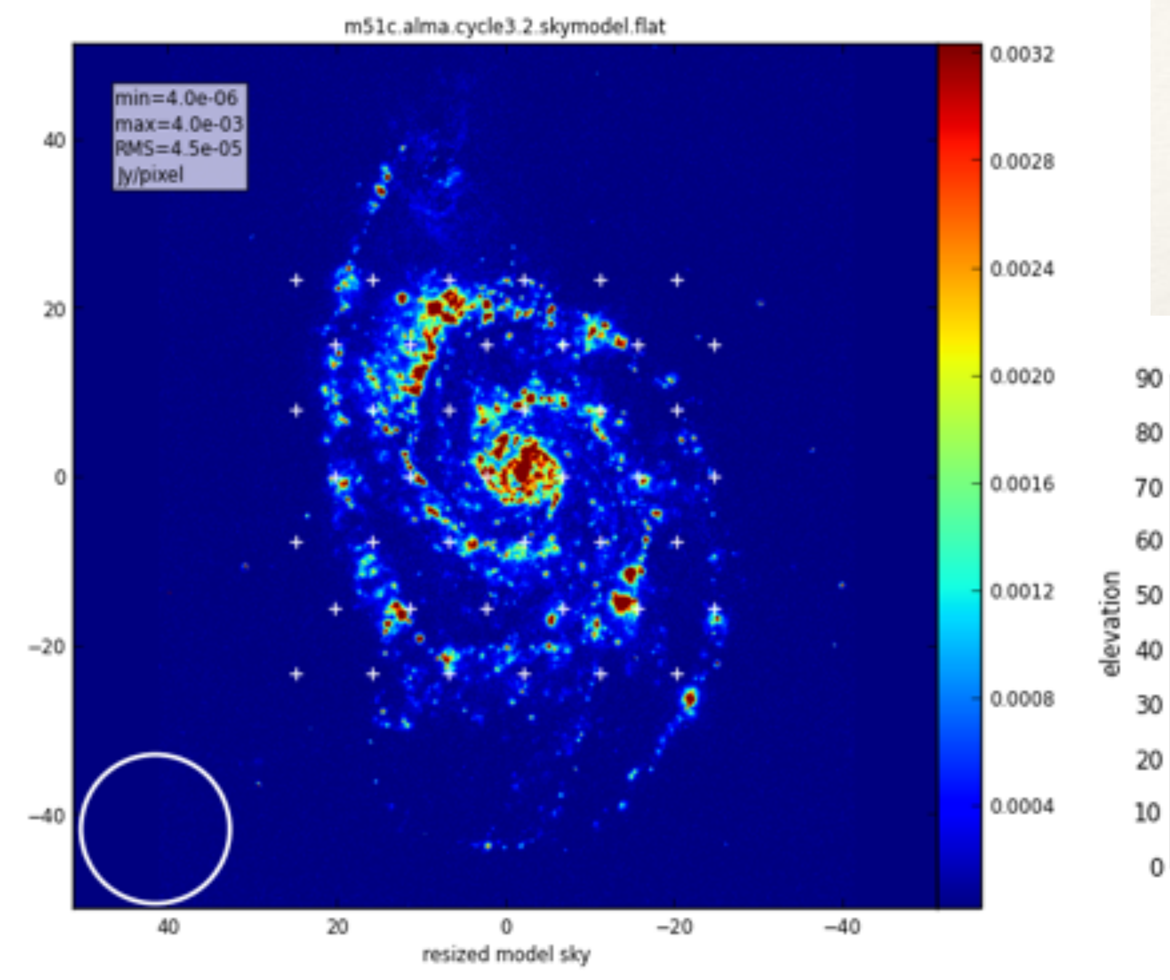
output files from simobserve/simanalyze

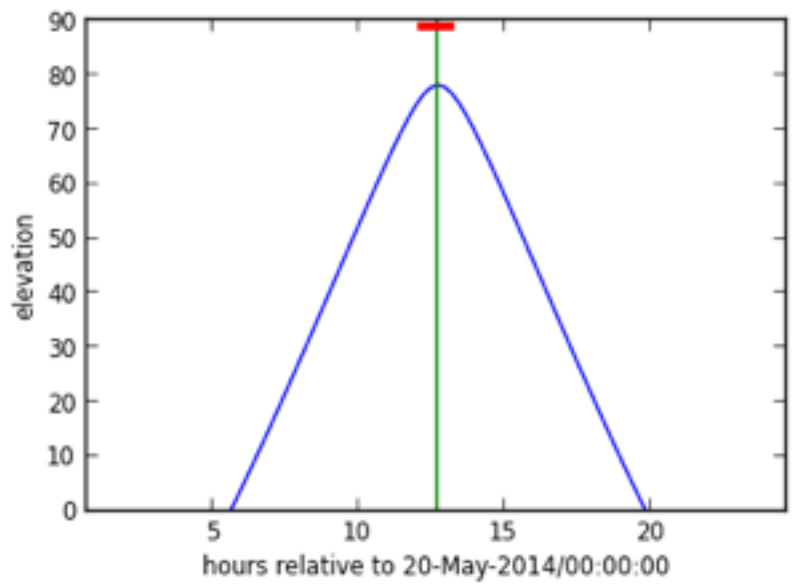
$project.$ant_cfg.xxxx

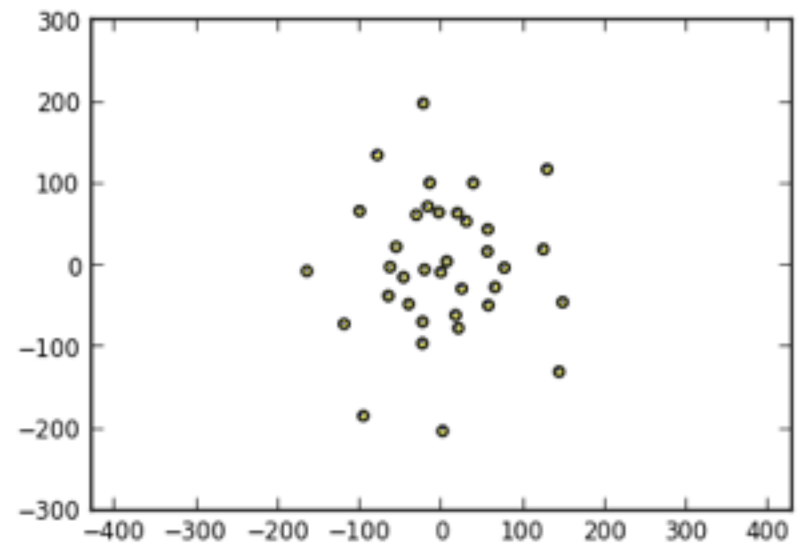
xxxx =

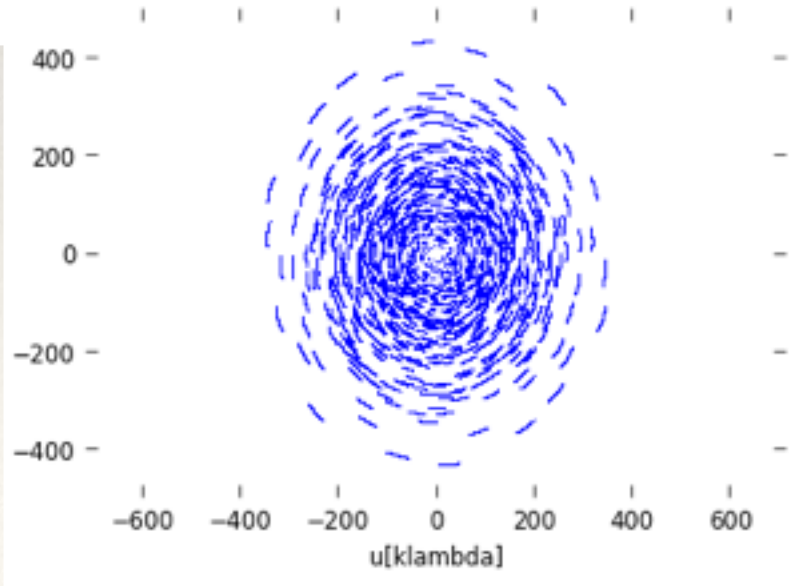
.diff	model image - simulated image
.fidelity	fidelity image
.flux	sky sensitivity image
.model	clean component image
.ms	measurement set (visibilities)
.psf	dirty beam (point source response)
.residual	residual image after clean
.skymodel	your input model sky image (original image size)
.skymodel.flat	flux-rescaled model sky image (original image size)
.skymodel.flat.regrid	regrid to the size of simulated image
.skymodel.flat.regrid.conv	and convolve with the clean beam
.ptg.txt	observed pointings

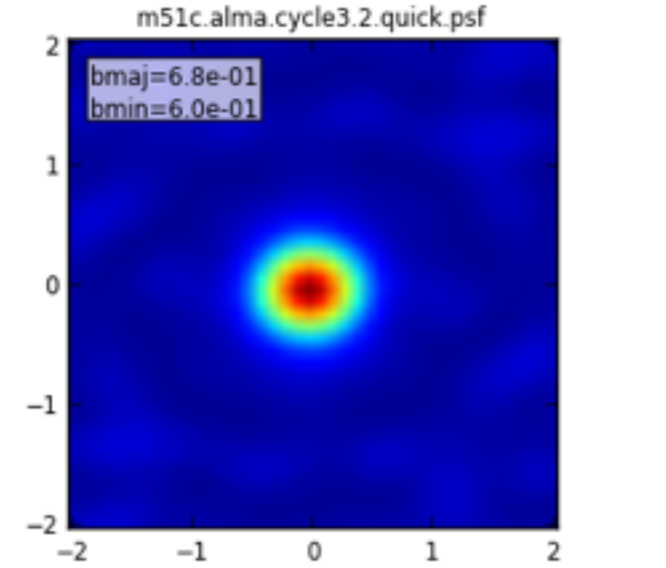
some output plots from simobserve



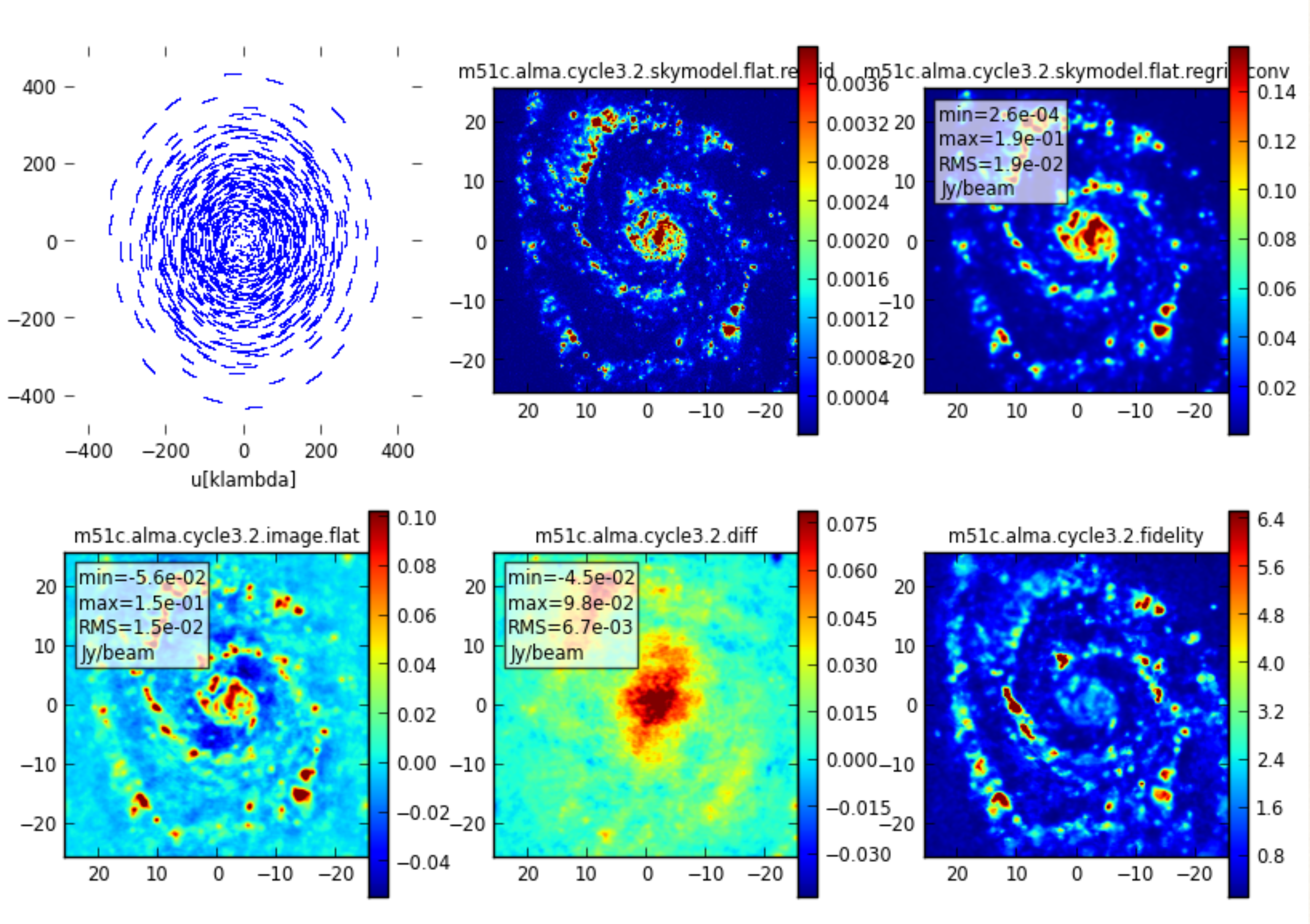








some output plots from simanalyze



full image synthesis

http://casaguides.nrao.edu/index.php?title=ACA_Simulation_(CASA_4.3)

Next we use **simanalyze** to combine the three measurement sets (12m,7m, and TP) and create a single image.

There are many ways to do this.
(1) We will use the total power image as a model when deconvolving the ACA image, and then use the result as a model when deconvolving the 12m interferometric image. This method tends to give low weight to the large spatial scales, but is simple to illustrate.

If given a total power and interferometric measurement set, **simanalyze** will automatically create the total power image, then use it as a model and deconvolve the interferometric image.

(2) It's possible to get better results if one used multiscale clean in the clean task (again using the lower resolution image as a model when deconvolving the higher resolution one).

(3) An alternative would be to create an image independently from each dataset, and then use the CASA feather task to combine them entirely in the image plane.

(4) If you decide to concatenate the two measurement sets and image all visibilities simultaneously, it is critical that the relative weights be set between the two different interferometric arrays. Simulated data has weights=1, since the thermal noise is generated uniformly per baseline. However, in reality the 7m baselines have lower sensitivity than the 12m baselines, and their weights must be decreased by that sensitivity ratio. `simalma` uses the `visweightscale` parameter of `concat` to apply that lower weight of (7./12)**2 to the 7m visibilities (See Simalma guide)

full image synthesis - simanalyze

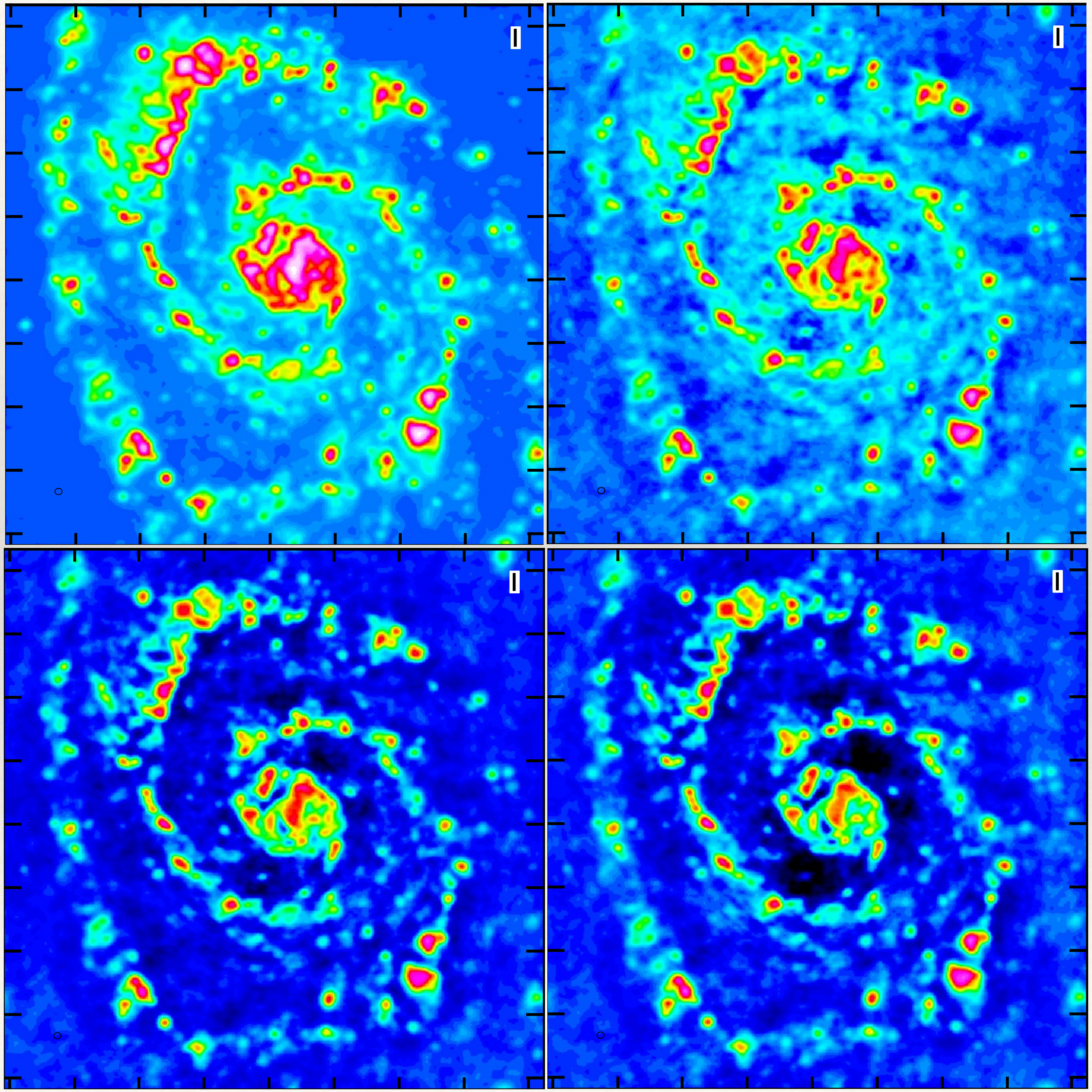
- Run simanalyze TWICE
- First image total power and 7m with total power as a model
  - vis = ‘$project.aca.cycle3.ms,$project.aca.tp.sd.ms’
- Next add the 12m data
  - vis = ‘$project.alma.cycle3.X.ms’
  - modelimage = ‘X.feather.image’

full image synthesis - simanalyze

- Image each ALMA component separately (run simanalyze three times) and run feather TWICE

```
#  feather :: Combine two images using their Fourier transforms
imagename          =        ''       #  Name of output feathered image
highres            =        ''       #  Name of high resolution
                                     #   (interferometer) image
lowres             =        ''       #  Name of low resolution (single dish)
                                     #   image
sdfactor           =       1.0       #  Scale factor to apply to Single Dish
                                     #   image
effdishdiam        =      -1.0       #  New effective SingleDish diameter to
                                     #   use in m
lowpassfiltersd    =     False       #  Filter out the high spatial
                                     #   frequencies of the SD image
```

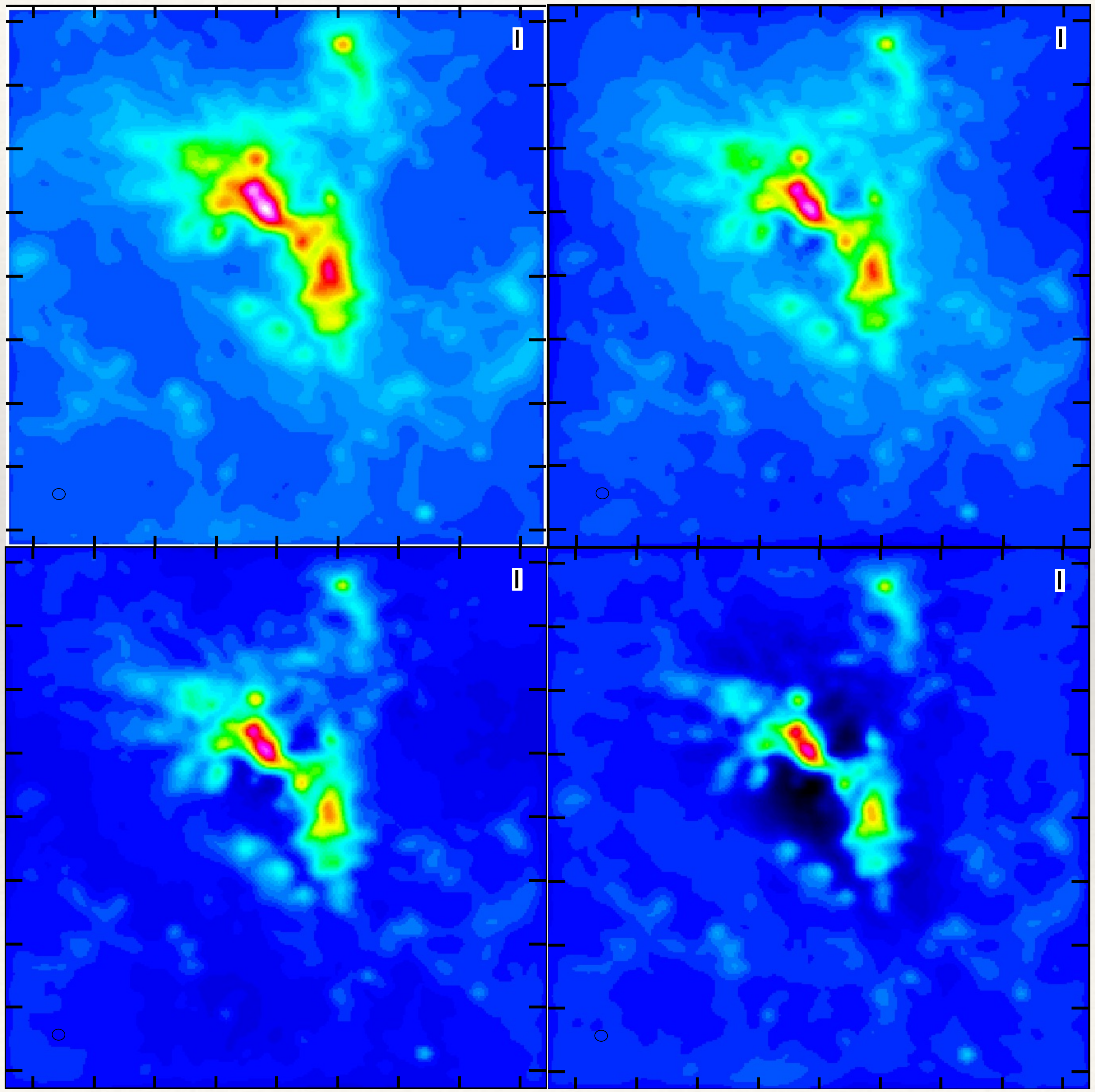
case study: a M51-like galaxy (snapshot)



case study: 30 Dor (snapshot)

sky model

12m+7m+TP



12m+7m

12m only

simalma

This task takes one set of parameters describing the region of the sky to observe, and makes the appropriate calls to **simobserve** and **simanalyze**.

The `simalma` task first calls `simobserve` to simulate the visibilities for each of the three array components: the 12 m Main Array, the 7 m Array, and the 12 m Total Power Array.

Next `simalma` generates an image from each of the three array components, separately. This step is not essential to getting the final result from the combined arrays, but it provides a useful diagnostic.

Note that the total power map covers the same region as the main array mosaic, with an extra pointing position added around the outside edge of the map so that the total power map is larger than the interferometric mosaic. (Total power maps usually have additional noise and artifacts at their edges). Furthermore, a square raster pattern is used instead of the hexagonal pattern of the interferometric array maps.

Next `simalma` uses **simanalyze** to combine the three measurement sets and create a single image. It accomplishes this in the following manner.

First, `simalma` concatenates the two sets of interferometric visibilities, and images them. Diagnostic graphics with "concat" in their names are generated.

Finally it combines the total power image with the concatenated interferometric image using the CASA task `feather`.

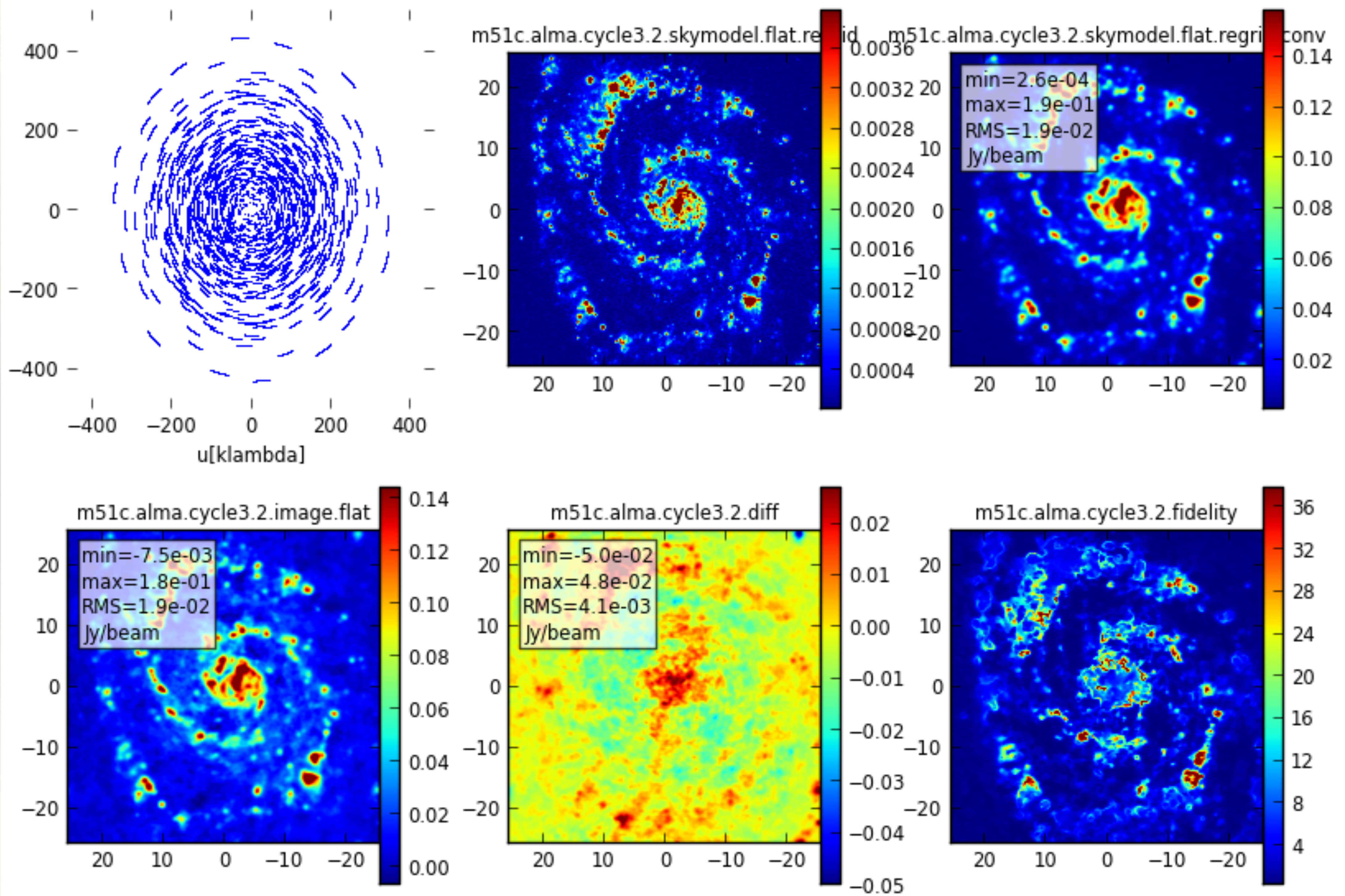
Some notes for combining data manually

1. When combining interferometric data from different arrays "manually", it is critical to set the relative data weights properly. Simulated data have weights=1, since the thermal noise is generated uniformly per baseline. However, in reality the 7m baselines have lower sensitivity than the 12m baselines, and their weights must be decreased by the sensitivity ratio. `simalma` uses the `visweightscale` parameter of `concat` to apply that lower weight of (7/12)**2 to the 7m visibilities. If you wish to combine data manually, you must do this step yourself.

2. When combining the single dish and interferometric maps in the image plane using the `feather` task, one must use the interferometric map *without* the primary beam correction, and first multiply the total power map by the interferometric sensitivity image (".flux") -- this ensures that noise effects are properly handled on the edges of each map. After running `feather`, the output is masked to 0.2 times the interferometric primary beam, since the total power map was created larger than the interferometric map on purpose, so the edges of the combined image do not contain any interferometric information.

Hands-on

- generate model observations of each ALMA component
  - >run simobserve_m51_X.last (X=12m,7m, and tp)
  - >inp (check input parameters)
  - >go
- image each component
  - >run simanalyze_m51_X.last (X=12m,7m, and tp)
  - >inp (check input parameters)
  - >go
- To combine components
- >run simanalyze_m51_tp_plus_7m.last
- >inp
- >go
- >run simanalyze_m51_tp7m_plus_12m.last
- >inp
- >go



u[klambda]
m51c.alma.cycle3.2.skymodel.flat.regrid
m51c.alma.cycle3.2.skymodel.flat.regrid.conv
min=2.6e-04
max=1.9e-01
RMS=1.9e-02
Jy/beam
m51c.alma.cycle3.2.image.flat
min=-7.5e-03
max=1.8e-01
RMS=1.9e-02
Jy/beam
m51c.alma.cycle3.2.diff
min=-5.0e-02
max=4.8e-02
RMS=4.1e-03
Jy/beam
m51c.alma.cycle3.2.fidelity