Simulating data for ALMA

(and other interferometers)

Rémy Indebetouw,
NAASC staff, esp Crystal Brogan
CASA development team

Atacama Large Millimeter/submillimeter Array
Expanded Very Large Array
Robert C. Byrd Green Bank Telescope
Very Large Baseline Array

NRAO

Simulating data for ALMA

(and other interferometers)

- ✧ ALMA introduction and status
- ✧ features of ALMA data to simulate
- ✧ implementation

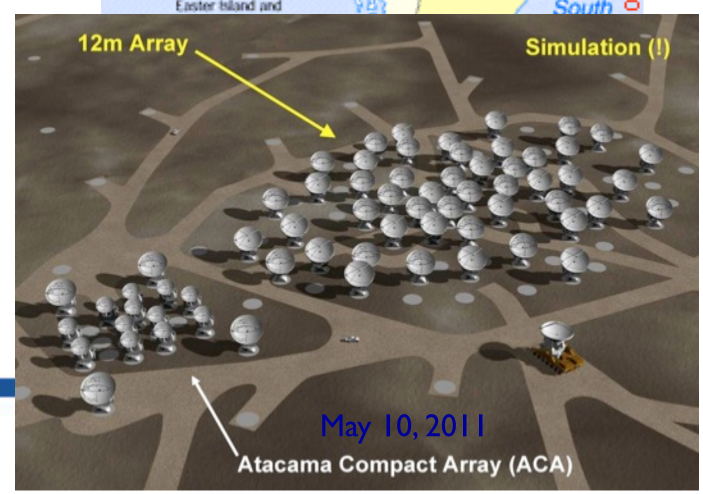
Rémy Indebetouw,
NAASC staff, esp Crystal Brogan
CASA development team

Atacama Large Millimeter/submillimeter Array
Expanded Very Large Array
Robert C. Byrd Green Bank Telescope
Very Large Baseline Array

NRAO

ALMA Overview

- A global partnership to deliver a transformational millimeter/submillimeter interferometer
  - North America (US, Canada, Taiwan)
  - Europe (ESO)
  - East Asia (Japan, Taiwan)
  - In collaboration with Chile
- 5000m (16,500 Ft) site in Chilean Atacama desert
- Main Array: 50 x 12m antennas
- \+ Atacama Compact Array (ACA): 12 x 7m antennas
- \+ Total Power Array 4 x 12m
- Total shared cost ~1.3 Billion ($US2006)

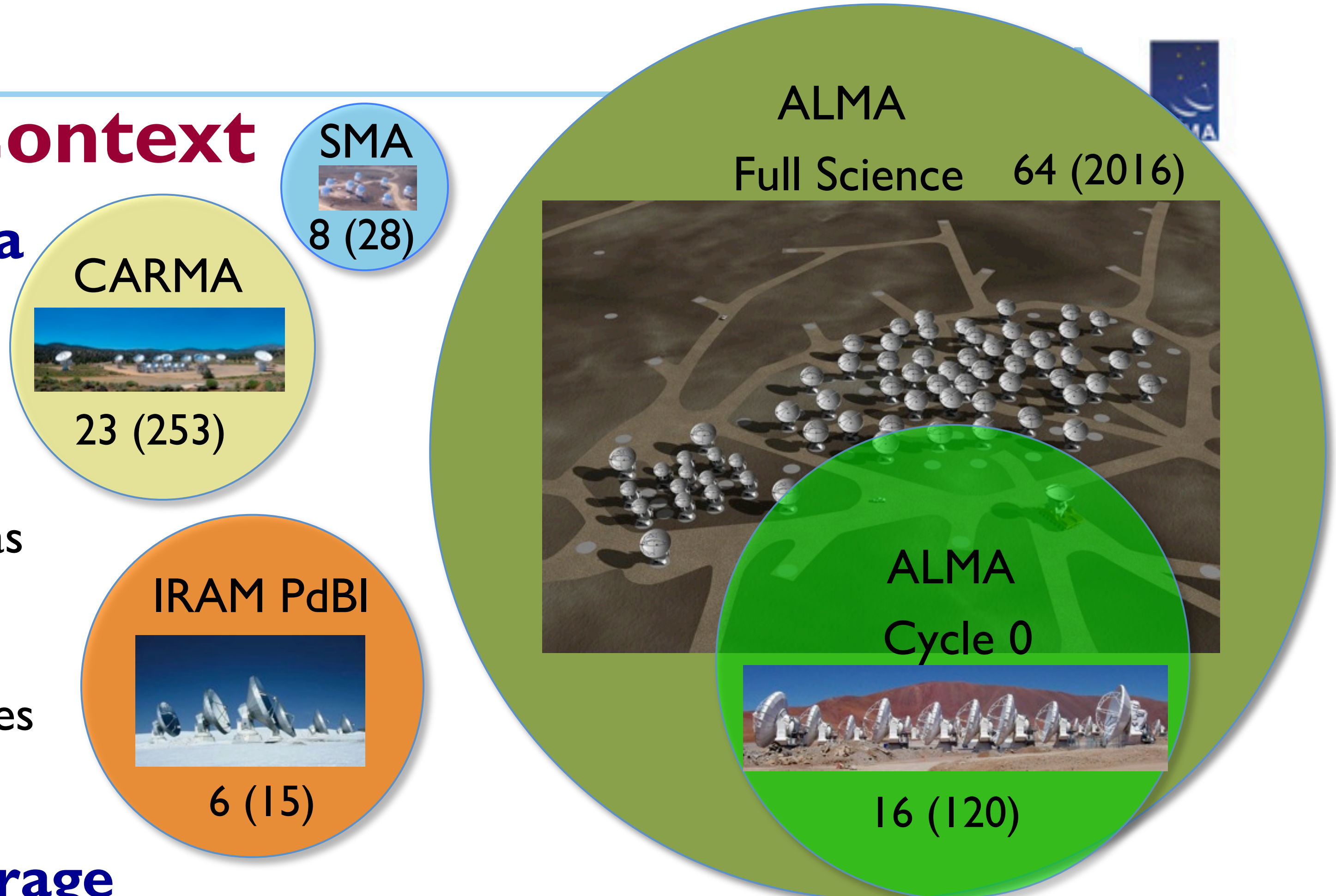


ALMA in Context

Collecting Area

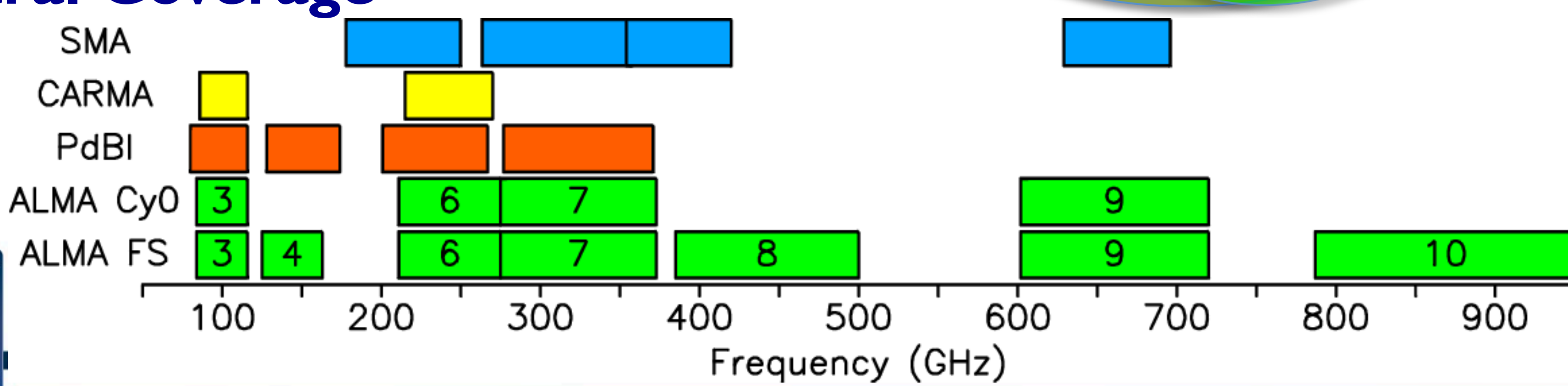
of Antennas

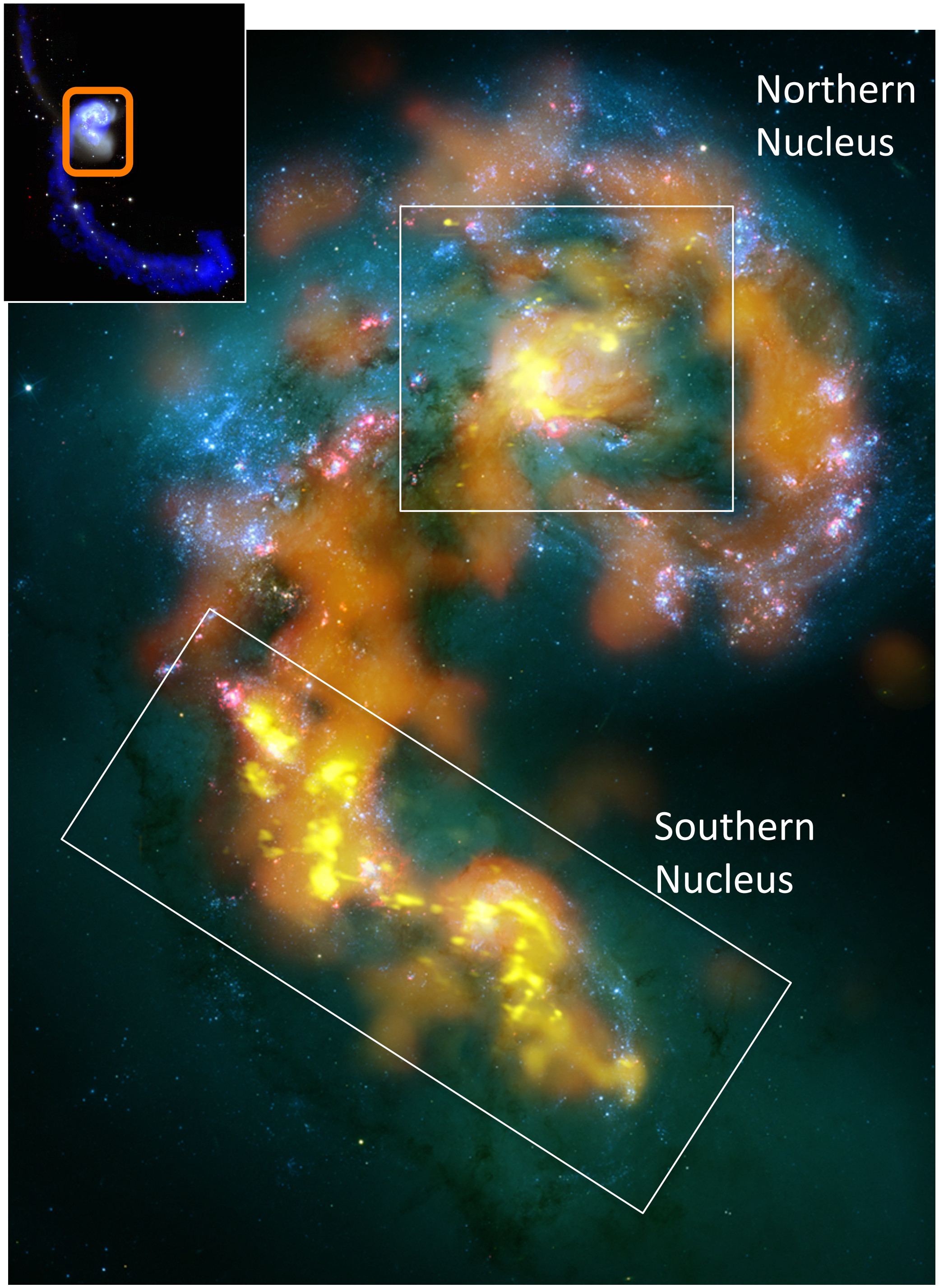
(# of baselines)



- Sensitivity goes as collecting area
- Image fidelity goes as # of baselines

Spectral Coverage





Zoom in on the Center of the Antennae Galaxies

Background (white, blue, red): *Hubble Space Telescope* optical image

- Traces adolescent and middle-aged massive clusters of stars.

Foreground (orange and yellow): **ALMA** test and science verification data showing emission from **carbon monoxide molecules**.

- Traces cold, dense, optically obscured clouds of molecular gas.
- Pinpoints where new generations of stars are born.
- Gradation of color from orange to yellow shows progressively denser gas. (Densest tracer only observed within the boxed regions.)

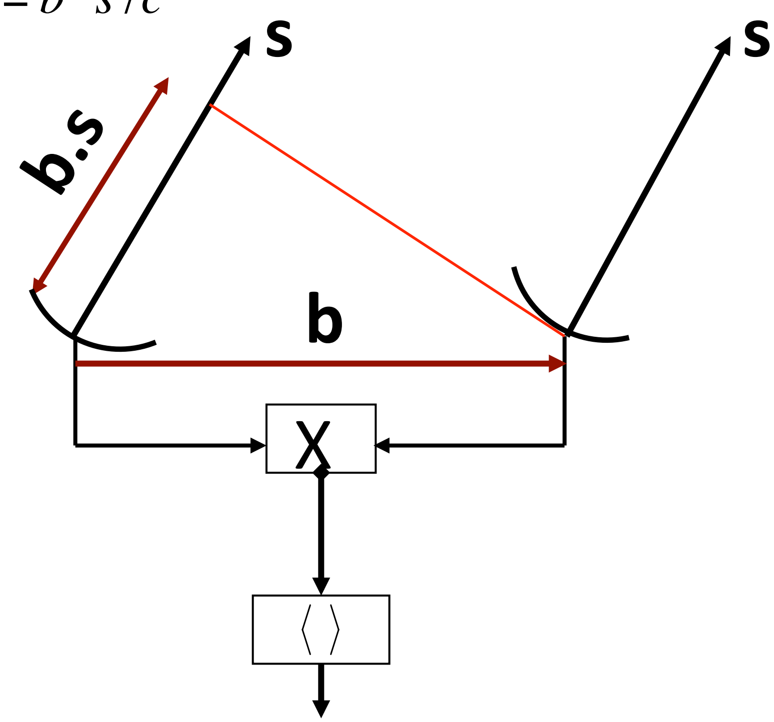
Data Issues

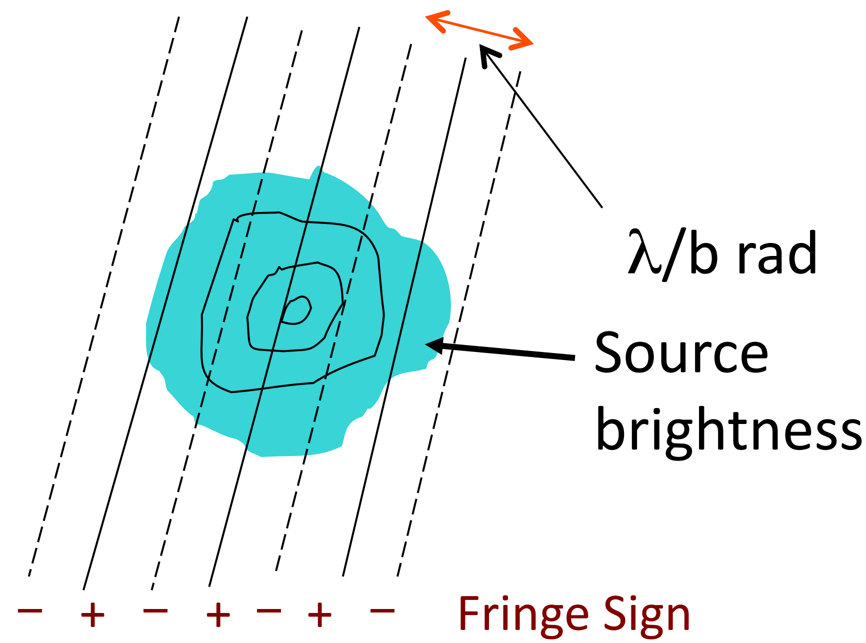
- volume: Cycle 0 ~50GB/dataset; Full rate 64GB/s
  - automated reduction pipeline
- sparsely filled aperture
  - image fidelity depends on observing details
- measurements made in Fourier space
  - mechanics of simulation more complicated
- small field of view
  - mosaics constructed from non-independent pointings
- aperture large compared to atmospheric fluctuations
  - correcting phase corruption is complex

Sparse Aperture Telescopes

Geometric
Time Delay

$\tau_g = \vec{b} \cdot \vec{s} / c$





>> angular sensitivity variation

$$R_C = [A^2 \cos(\omega\tau_g)]/2$$

figure c/o Rick Perley

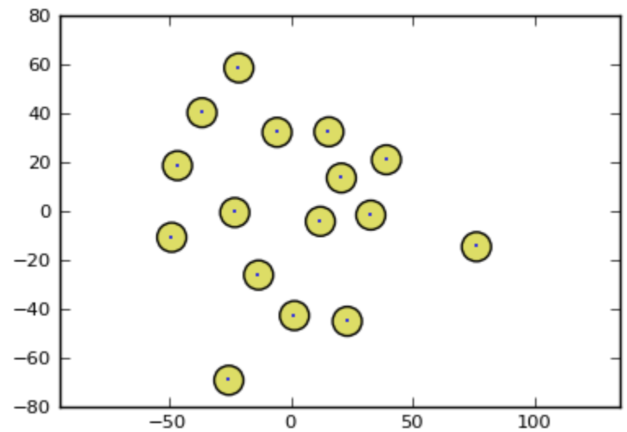
Fundamentally, we measure the coherence function or visibility at a discrete location in Fourier space

Sparse Aperture Telescopes

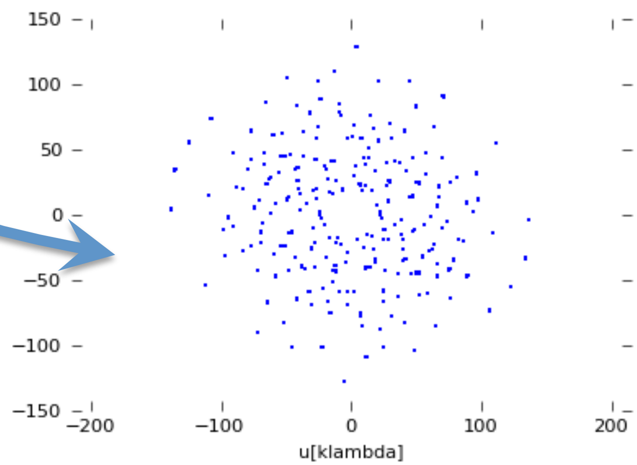
more baselines, either from more antennas, or earth rotation >
more information in Fourier space >
cleaner/tighter point spread function

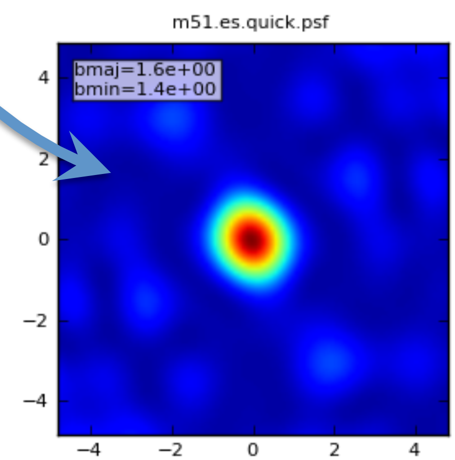
ALMA Cycle 0 “snapshot”:

modest coverage in uv space
results in
PSF sidelobes



Output from CASA simdata



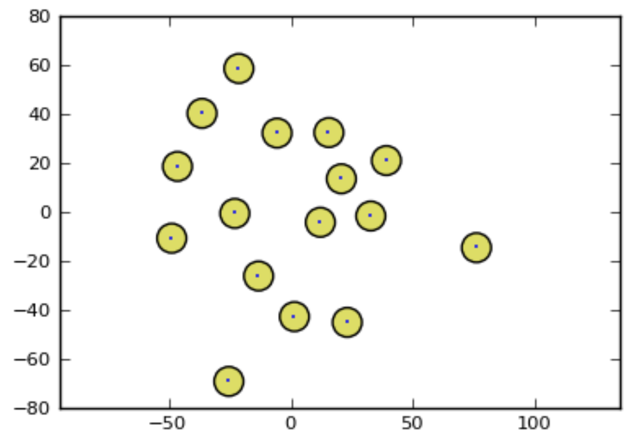


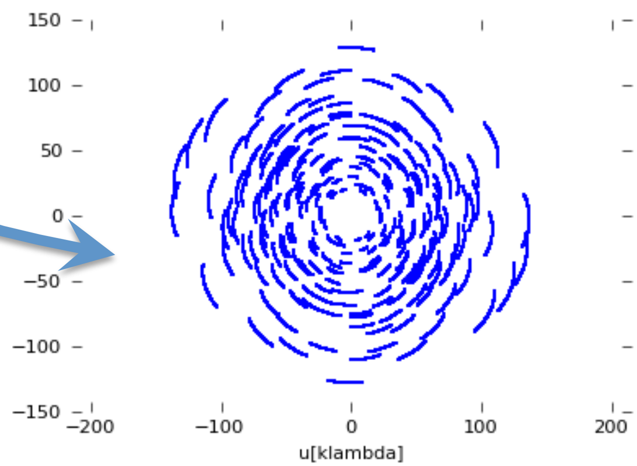
Sparse Aperture Telescopes

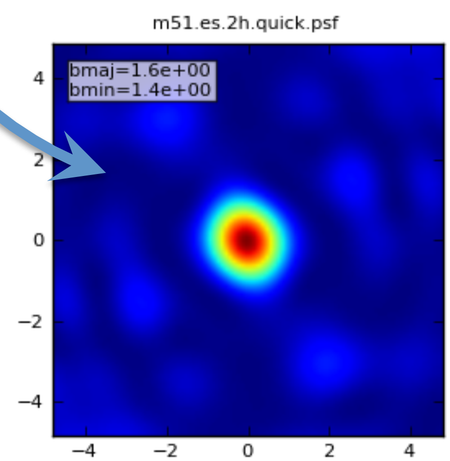
more baselines, either from more antennas, or earth rotation >
more information in Fourier space >
cleaner/tighter point spread function

ALMA Cycle 0 2h integration:

better coverage in uv space
results in
lower PSF sidelobes



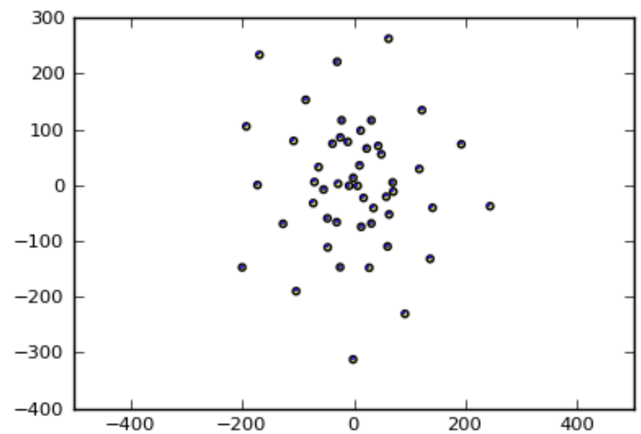




Sparse Aperture Telescopes

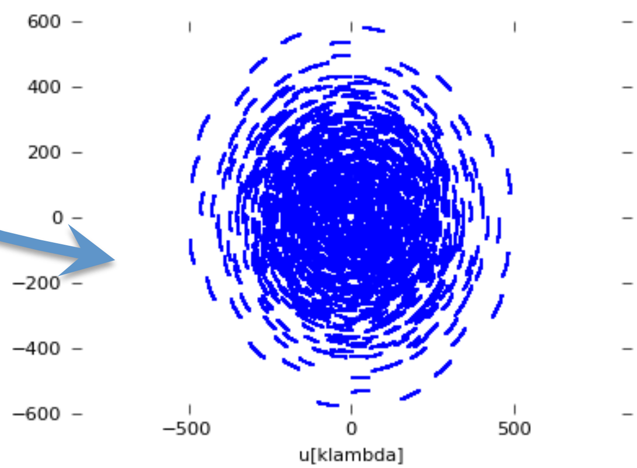
more baselines, either from more antennas, or ear
more information in Fourier space >
cleaner/tighter point spread function

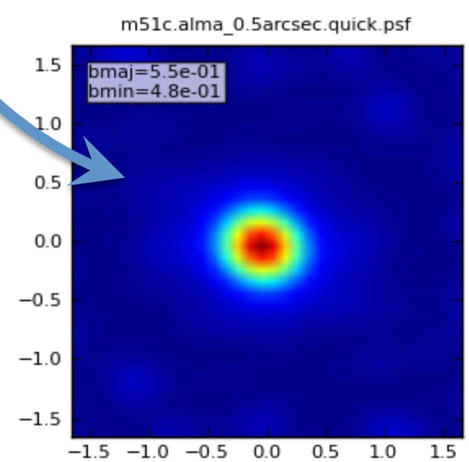
★ We used CASA simdata to determine the optimal antenna placement for ALMA Cycles 0 and 1



ALMA Full Science:

great coverage in uv space
results in
negligible PSF sidelobes

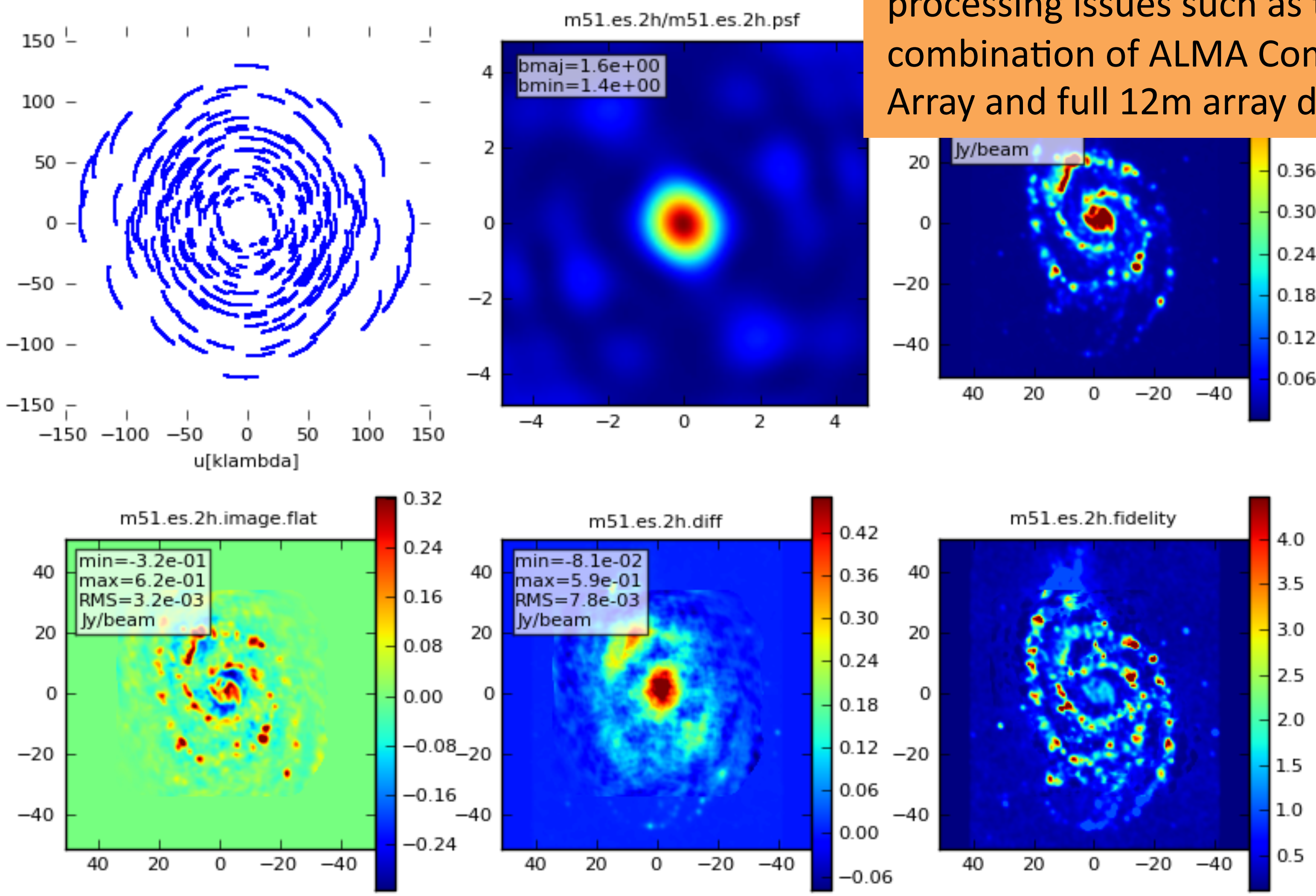




Sparse Aperture Telescopes

Spatial filtering of a realistic source

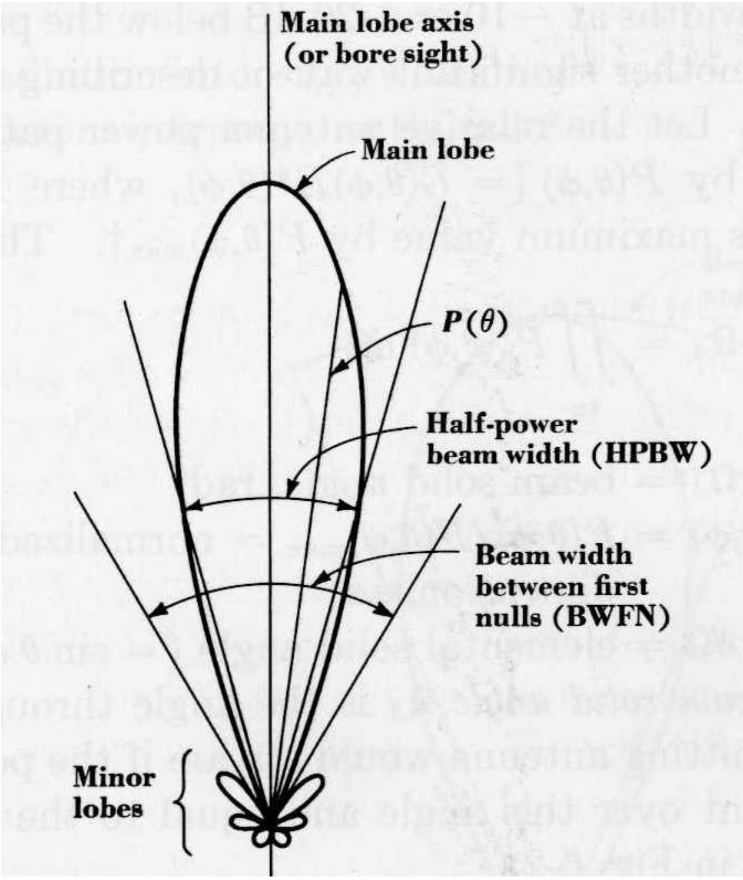
We're using simdata to explore image reconstruction and data processing issues such as the combination of ALMA Compact Array and full 12m array data

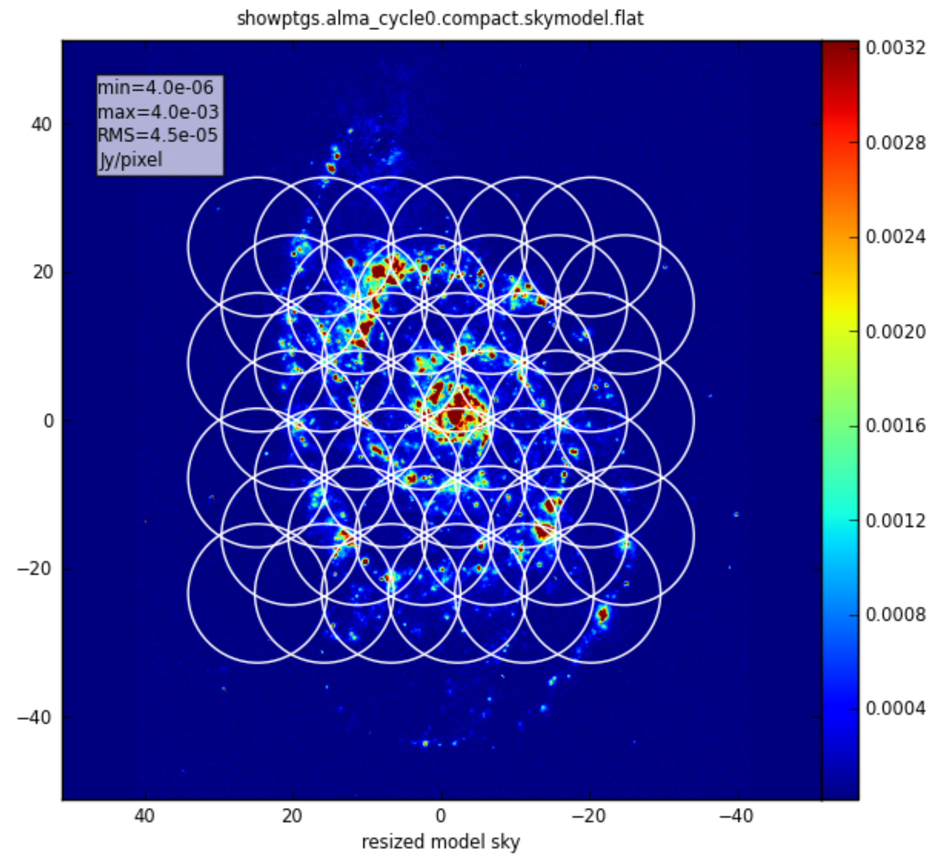


NRAO

ALMA
TACAMA LARGE MILLIMETER ARRAY

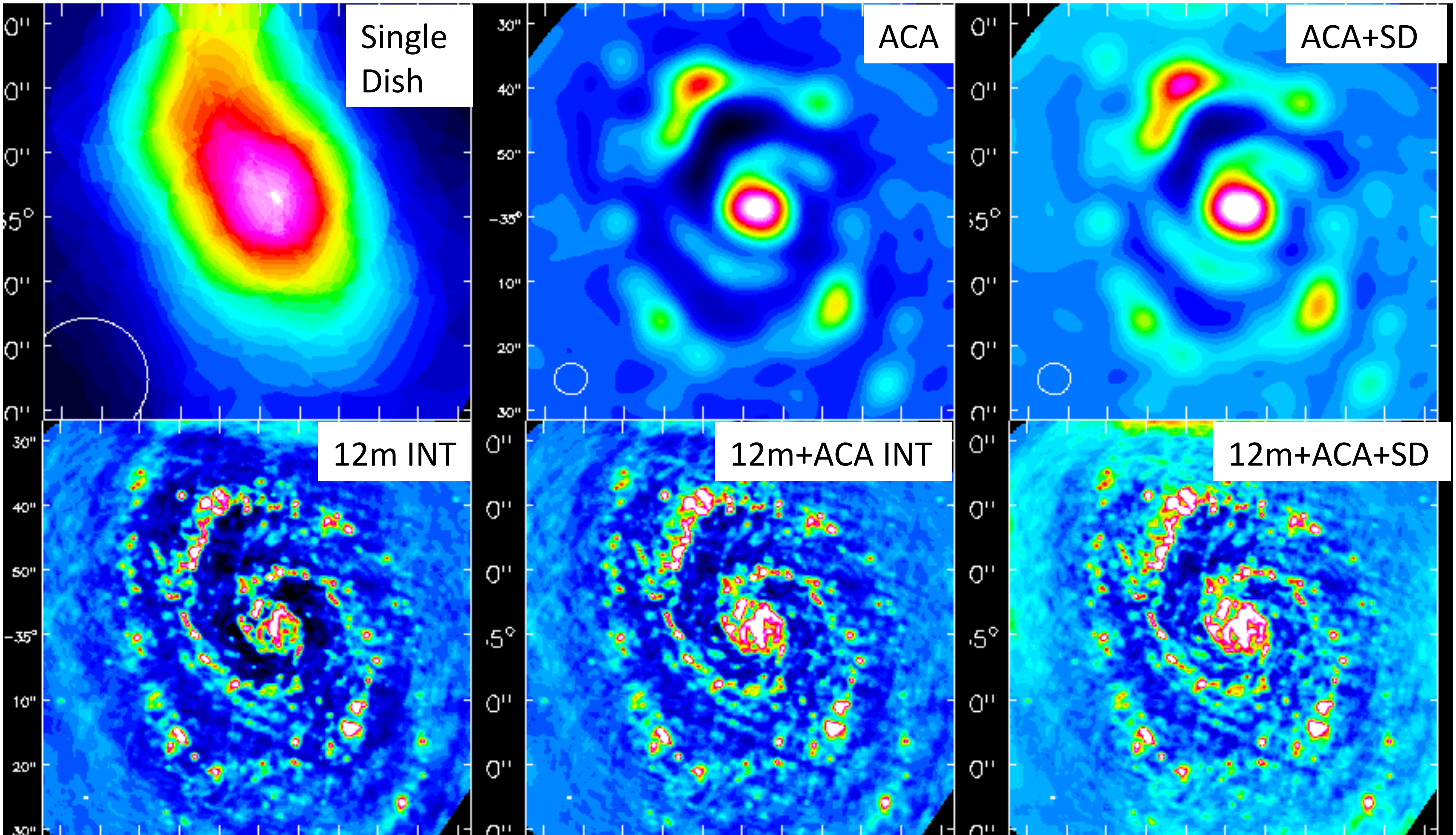
Small Field of View in Gaussian Regime



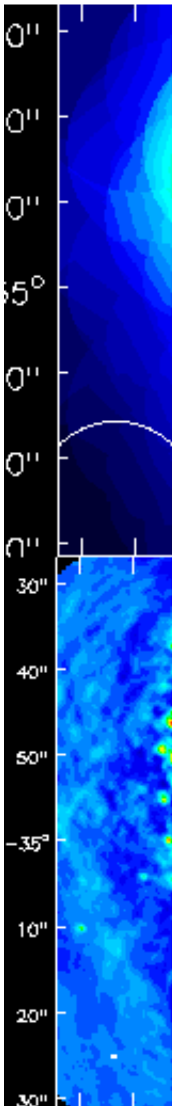


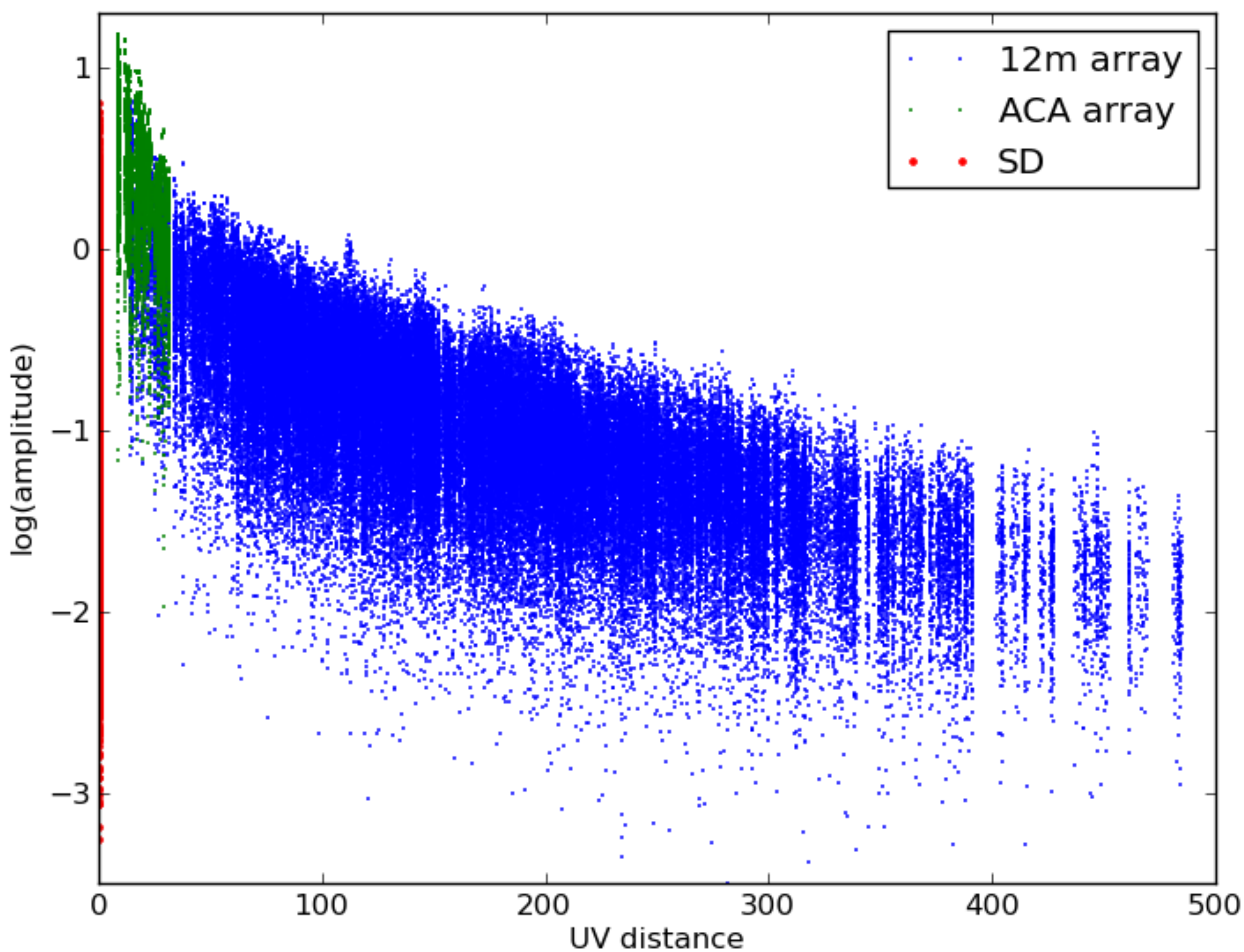
Kraus, 1966. Fig.6-1, p. 153.
ALMA 12m primary beam size at 345GHz (850µm) = 18”

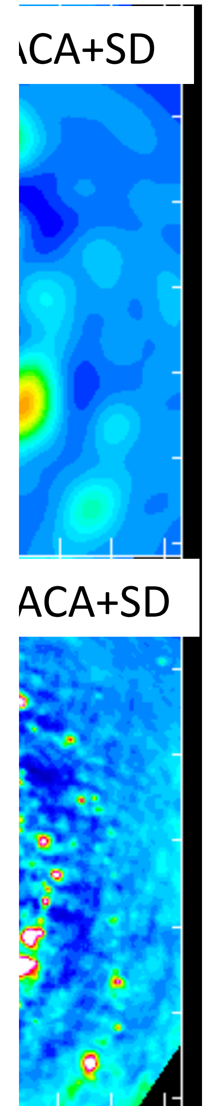
Multiple Arrays



Multiple Arrays – Fourier space







Thermal noise

$$F_\nu^*(\text{noise}) = \frac{4\sqrt{2}k10^{-23}}{\eta_a \eta_c \pi d_1 d_2 \sqrt{\Delta\nu \Delta t}} [T_{CMB} + \eta_s T_{atm}(e^{\tau_\nu} - 1) + (1-\eta_s) T_{amb} e^{\tau_\nu} + T_{RX} e^{\tau_\nu}]$$

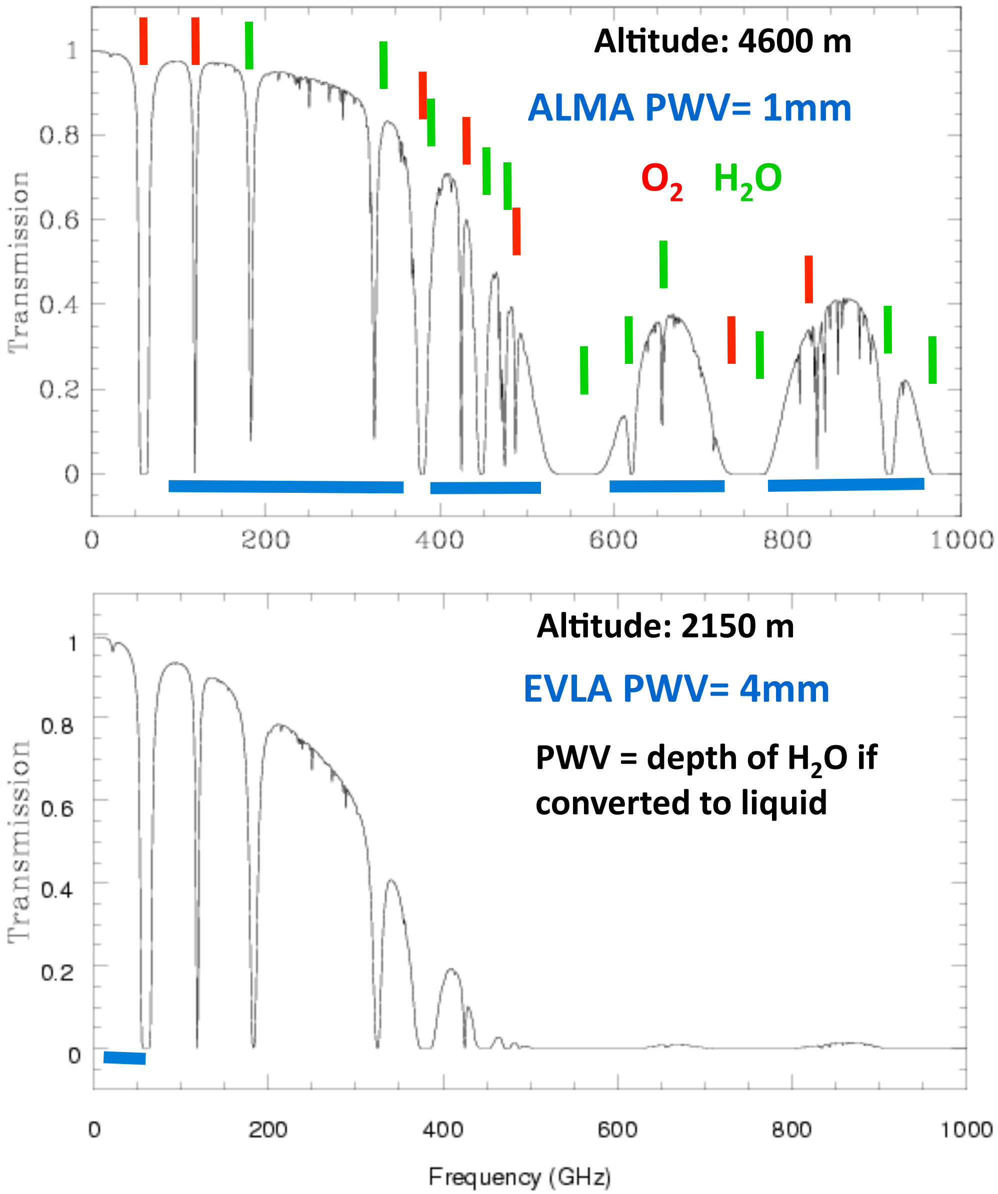
Geometric mean collecting area for a pair of antennas

Bandwidth * integration time

Proportional to the power coming from the atmosphere, sidelobes, and receiver (wave, not photon counting statistics)

simdata used to determine the relative importance of thermal noise and dynamic range limitations

The Atmosphere

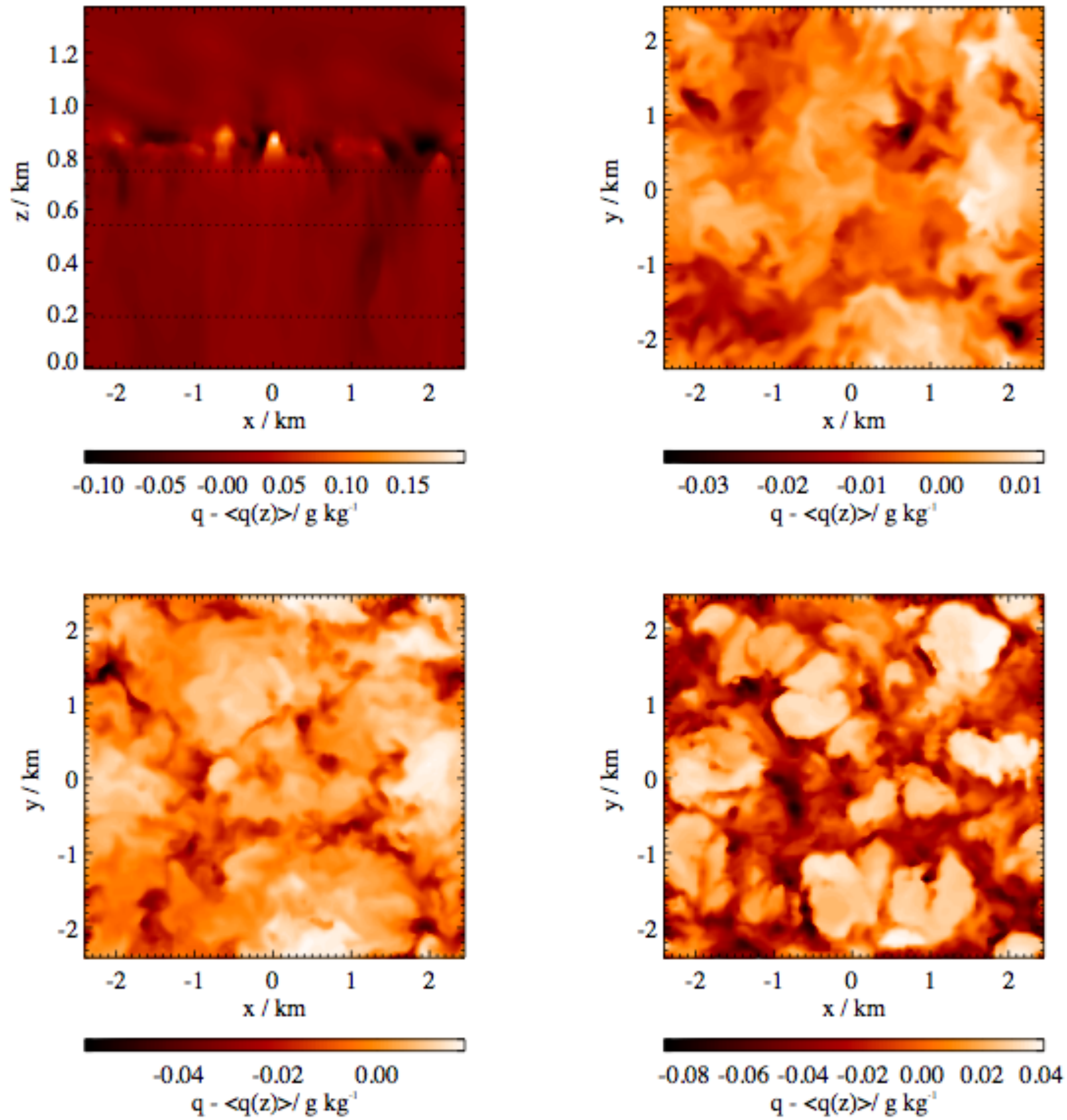


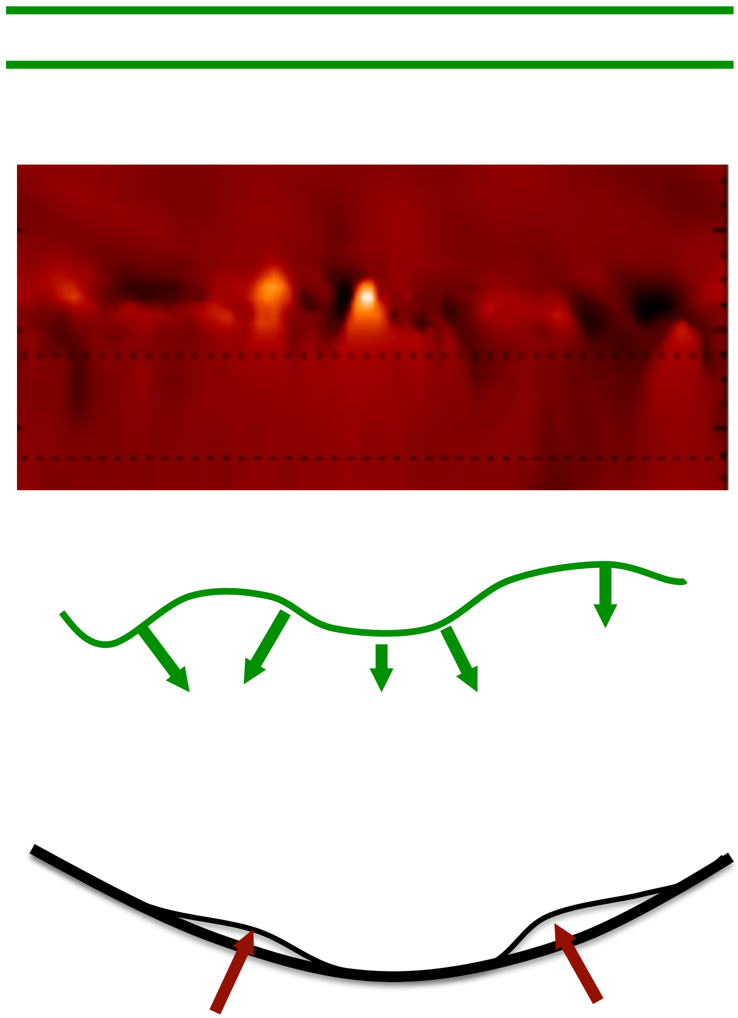
Models of atmospheric transmission from 0 to 1000 GHz for the ALMA site in Chile, and for the VLA site in New Mexico

$\Rightarrow$ Atmosphere transmission not a problem for λ > cm (most VLA bands)

The Atmosphere

Alison Stirling et al ALMA memo 517





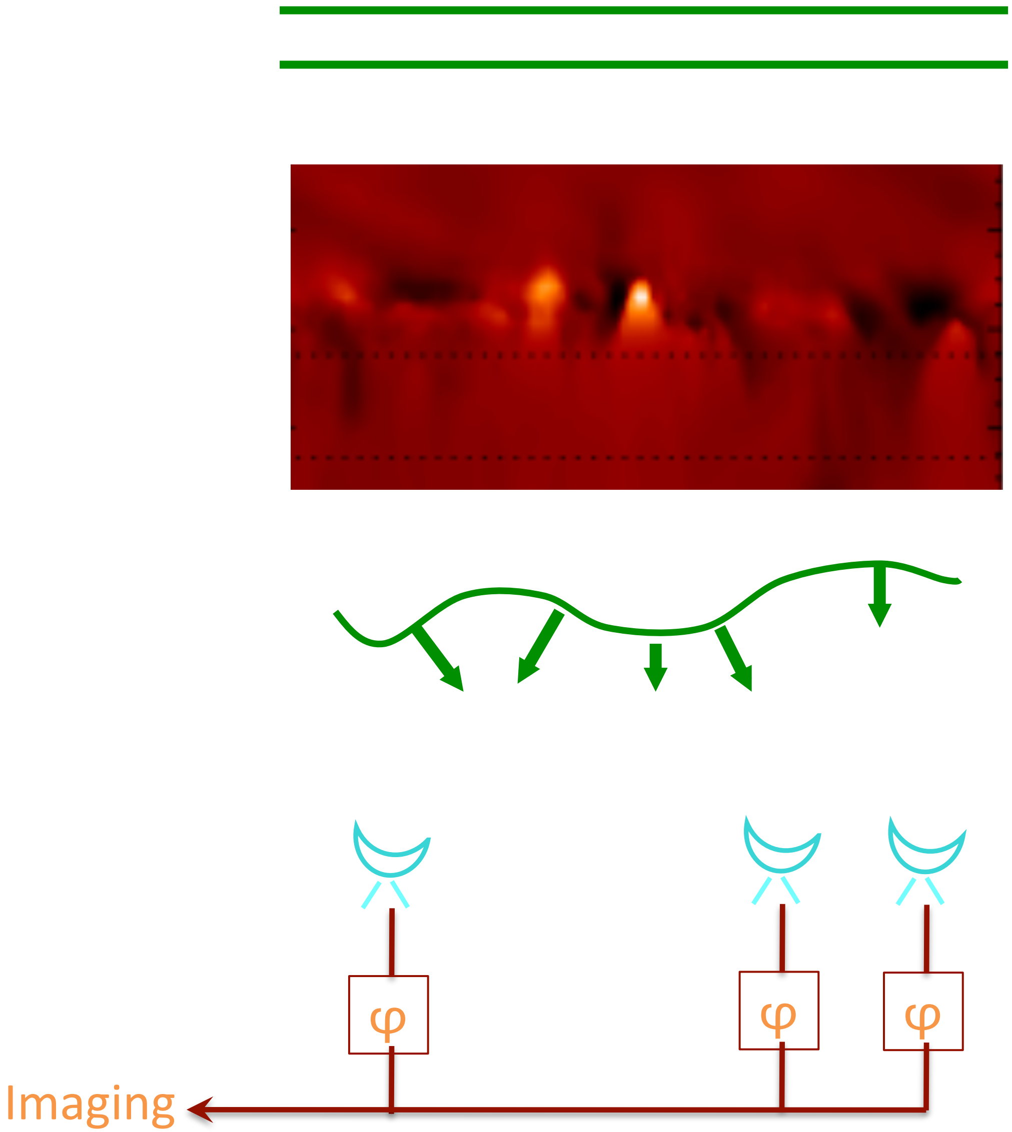
flat wave front

Atmosphere: fluctuating index of refraction

distorted wave front

filled aperture telescope

correction by distorting the optical surface in real time



flat wave front

Atmosphere: fluctuating index of refraction

distorted wave front

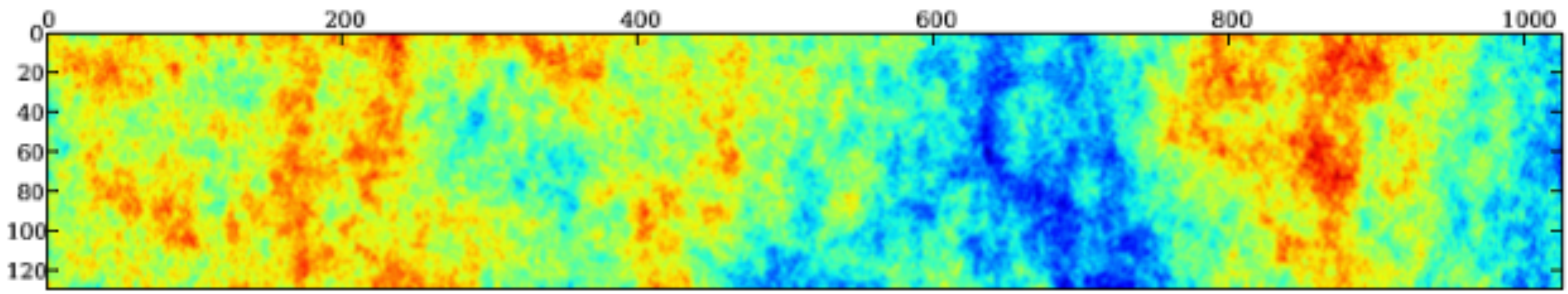
sparse aperture heterodyne telescope

correction by applying phase delay to each antenna (in real time or afterwards)

>> a.k.a. calibration

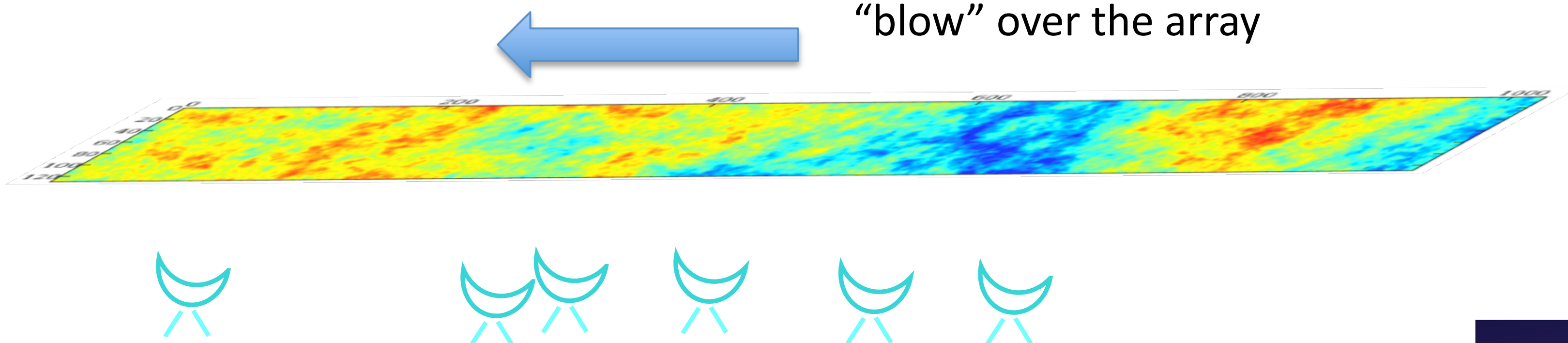
The Atmosphere

Bojan Nikolic et al. ALMA memos 573, 582, 587, 588, 590

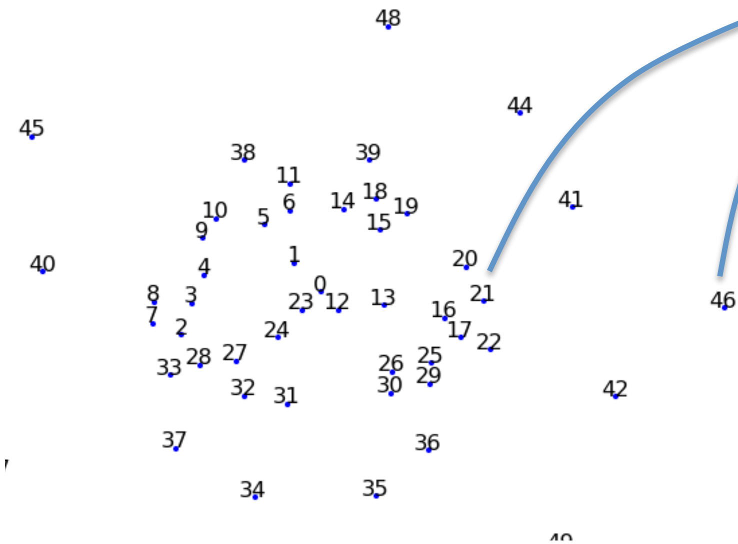


Water vapor fluctuation screen > phase delay screen

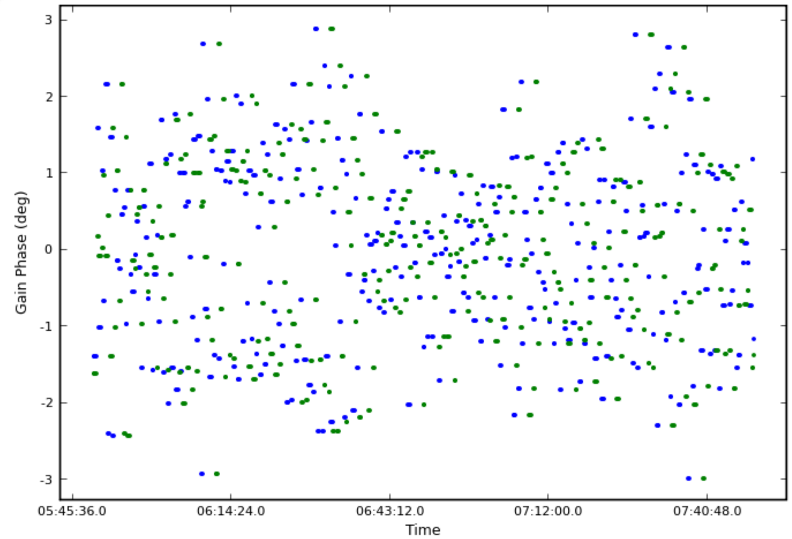
“blow” over the array



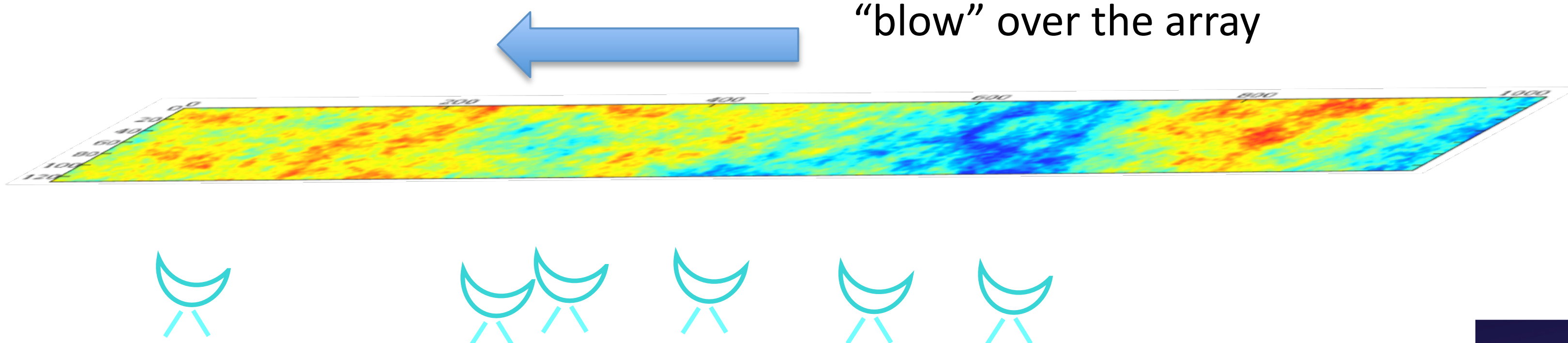
The Atmosphere



Phase delay for 2 antennas aligned with wind direction



“blow” over the array



simdata in CASApy

- Synthesis data reduction:
  - manipulate tables of complex visibilities (100s of GB)
  - calibrate = correct atmospheric phase and amp corruption
  - image, deconvolve, analyze, and repeat
- CASA
  - powerful C++ libraries bound to ipython user interface
  - most common steps arranged as menu-parameter driven "tasks"
- useful for building data analysis tools, e.g.
  - routines for coordinate transformation and geodesy
  - 4d image analysis and visualization
  - easy to import and combine CASA C++ tool objects with scipy, etc

CASA Common Astronomy Software Applications

NRAO

ALMA ATACAMA LARGE MILLIMETER ARRAY

simdata in CASApy

- Simulation steps
  - Define truth (model sky)
  - Define the observation (list of pointings)
  - Observe = calculate Fourier visibilities
  - Corrupt visibilities
  - Image and deconvolve
  - Analyze the output compared to the input

```
CASA <15>: inp
---------> inp()
#  sim_observe :: mosaic simulation task:
project             =      'sim'        #  root prefix for output file names
skymodel            = 'g41_I4.rd.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        = '1.5arcmin'       #  angular size of map or "" to cover model
     maptype        =     'ALMA'        #  hexagonal, square, etc
     pointingspacing =         ''       #  spacing in between pointings or "0.25PB" or "" for 0.5 PB

observe             =       True        #  calculate visibilities using ptgfile
     antennalist    = 'ALMA;0.5arcsec'  #  antenna position file or "" for no interferometric MS
     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'
     sdantlist      =         ''        #  single dish antenna position file or "" for no total power MS
     sdant          =          0        #  single dish antenna index in file

thermalnoise        = 'tsys-atm'        #  add thermal noise: [tsys-atm|tsys-manual|""]
     user_pwv       =        1.0        #  Precipitable Water Vapor in mm
     t_ground       =      269.0        #  ambient temperature
     seed           =      11111        #  random number seed

leakage             =        0.0        #  cross polarization
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 sim_observe(...)
```

```
CASA <15>: inp
---------> inp()
#  sim_observe :: mosaic simulation task:
project             =      'sim'      #  root prefix for output file names
skymodel            = 'g41_I4.rd.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
     integratio
     direction                                or "" to center on model
     mapsize                                  o cover model
     maptype
     pointingspacing                          s or "0.25PB" or "" for 0.5 PB

observe             =       True      #  calculate visibilities using ptgfile
     antennalist    = 'ALMA;0.5arcsec' #  antenna position file or "" for no interferometric MS
     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'
     sdantlist      =         ''      #  single dish antenna position file or "" for no total power MS
     sdant          =          0      #  single dish antenna index in file

thermalnoise        =  'tsys-atm'     #  add thermal noise: [tsys-atm|tsys-manual|""]
     user_pwv       =        1.0      #  Precipitable Water Vapor in mm
     t_ground       =      269.0      #  ambient temperature
     seed           =      11111      #  random number seed

leakage             =        0.0      #  cross polarization
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 sim_observe(...)
```

Model sky image can be rescaled in intensity, and its spatial and spectral coordinate information adjusted

```
CASA <15>: inp
---------> inp()
#  sim_observe :: mosaic simulation task:
project             =       'sim'       #  root prefix for output file names
skymodel            = 'g41_I4.rd.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        = '1.5arcmin'       #  angular size of map or "" to cover model
     maptype        =      'ALMA'       #  hexagonal, square, etc
     pointingspacing =          ''      #  spacing in between pointings or "0.25PB" or "" for 0.5 PB

observe             =        True       #  calculate visibilities using ptgfile
     antennalist    = 'ALMA;0.5arcsec'  #  antenna position file or "" for no interferometric MS
     refdate        =                      l unless concatting simulations
     hourangle      =                      e.g. -3:00:00, or "transit"
     totaltime      =                      er of repetitions
     caldirection   =                      l]
     calflux        =
     sdantlist      =                       or "" for no total power MS
     sdant          =           0       #  single dish antenna index in file

thermalnoise        =   'tsys-atm'      #  add thermal noise: [tsys-atm|tsys-manual|""]
     user_pwv       =         1.0       #  Precipitable Water Vapor in mm
     t_ground       =       269.0       #  ambient temperature
     seed           =       11111       #  random number seed

leakage             =         0.0       #  cross polarization
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 sim_observe(...)
```

coverage of a portion of the sky can be calculated automatically, or the user can define complicated patterns

```
CASA <15>: inp
---------> inp()
#  sim_observe :: mosaic simulation task:
project             =      'sim'         #  root prefix for output file names
skymodel            = 'g41_I4.rd.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        = '1.5arcmin'        #  angular size of map or "" to cover model
     maptype        =     'ALMA'         #  hexagonal, square, etc
     pointingspacing =        ''         #  spacing in between pointings or "0.25PB" or "" for 0.5 PB

observe             =       True         #  calculate visibilities using ptgfile
     antennalist    = 'ALMA;0.5arcsec'   #  antenna position file or "" for no interferometric MS
     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'
     sdantlist      =         ''         #  single dish antenna position file or "" for no total power MS
     sdant          =          0         #  single dish antenna index in file

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

Any array for which the antenna positions are known can be simulated, including ones like SKA that don't exist yet.
A phase calibrator can be observed with the science target, to test data reduction strategies

simdata in CASApy

```
CASA <18>: inp sim_analyze
---------> inp(sim_analyze)
#  sim_analyze :: 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 sim_analyze(...)
```

simdata in CASApy

```
CASA <18>: inp sim_analyze
---------> inp(sim_analyze)
#  sim_analyze :: 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
                                                                         sh simulation)
     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 sim_analyze(...)
```

Imaging allows combination of different array configurations as well as single-dish data.

simdata in CASApy

Numerous diagnostic plots can be generated along with the data.

```
CASA <18>: inp sim_analyze
---------> inp(sim_analyze)
#  sim_analyze :: 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)
     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 sim_analyze(...)
```

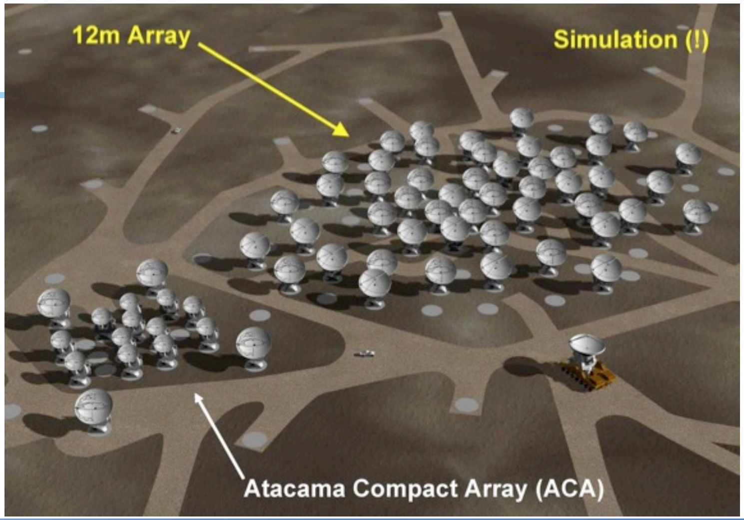
simdata in CASApy

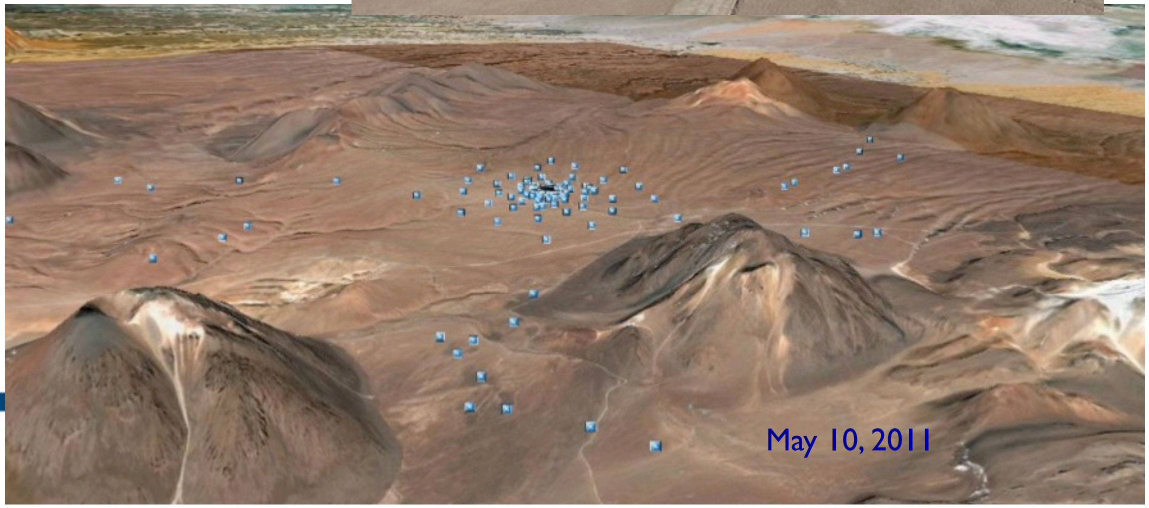
- Useful tools to understand and analyze interferometric data
- Used internally to ALMA project,
  and by numerous Cycle 0 proposers
- Simplified 2D web interface available,
  the "ALMA Observing Support Tool OST"
- Should continue to be useful through the next few years of ALMA, and to help design future arrays

ALMA Overview

- Baselines up to 15 km (0.015" at 300 GHz) in "zoom lens" configurations
- Sensitive, precision imaging 84 to 950 GHz (3 mm to 315 μm)
- State-of-the-Art low-noise, wide-band SIS receivers (8 GHz bandwidth)
- Flexible correlator with high spectral resolution at wide bandwidth
- Full polarization capabilities
- Estimate 1 TB/day archived

- A resource for ALL astronomers



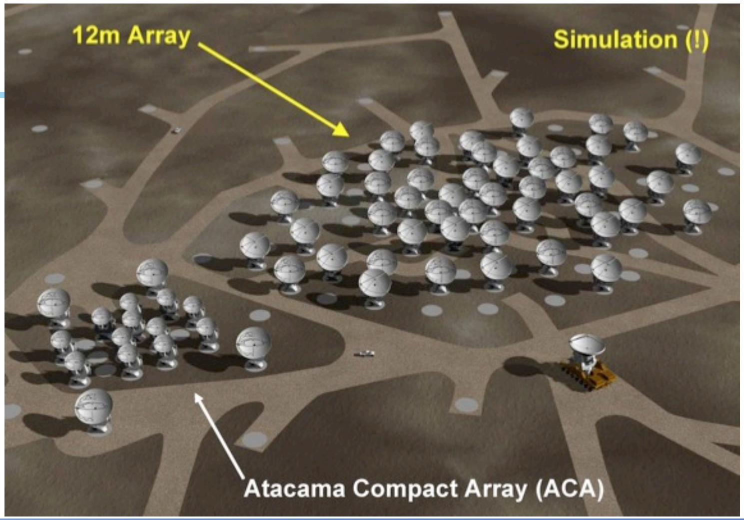


NRAO

ALMA Overview

- Baselines up to 15 km (0.015" at 300 GHz) in "zoom lens" configurations
- Sensitive, precision imaging 84 to 950 GHz (3 mm to 315 µm)
- State-of-the-Art low-noise, wide-band SIS receivers (8 GHz bandwidth)
- Flexible correlator with high spectral resolution at wide bandwidth
- Full polarization capabilities
- Estimate 1 TB/day archived

- A resource for ALL astronomers



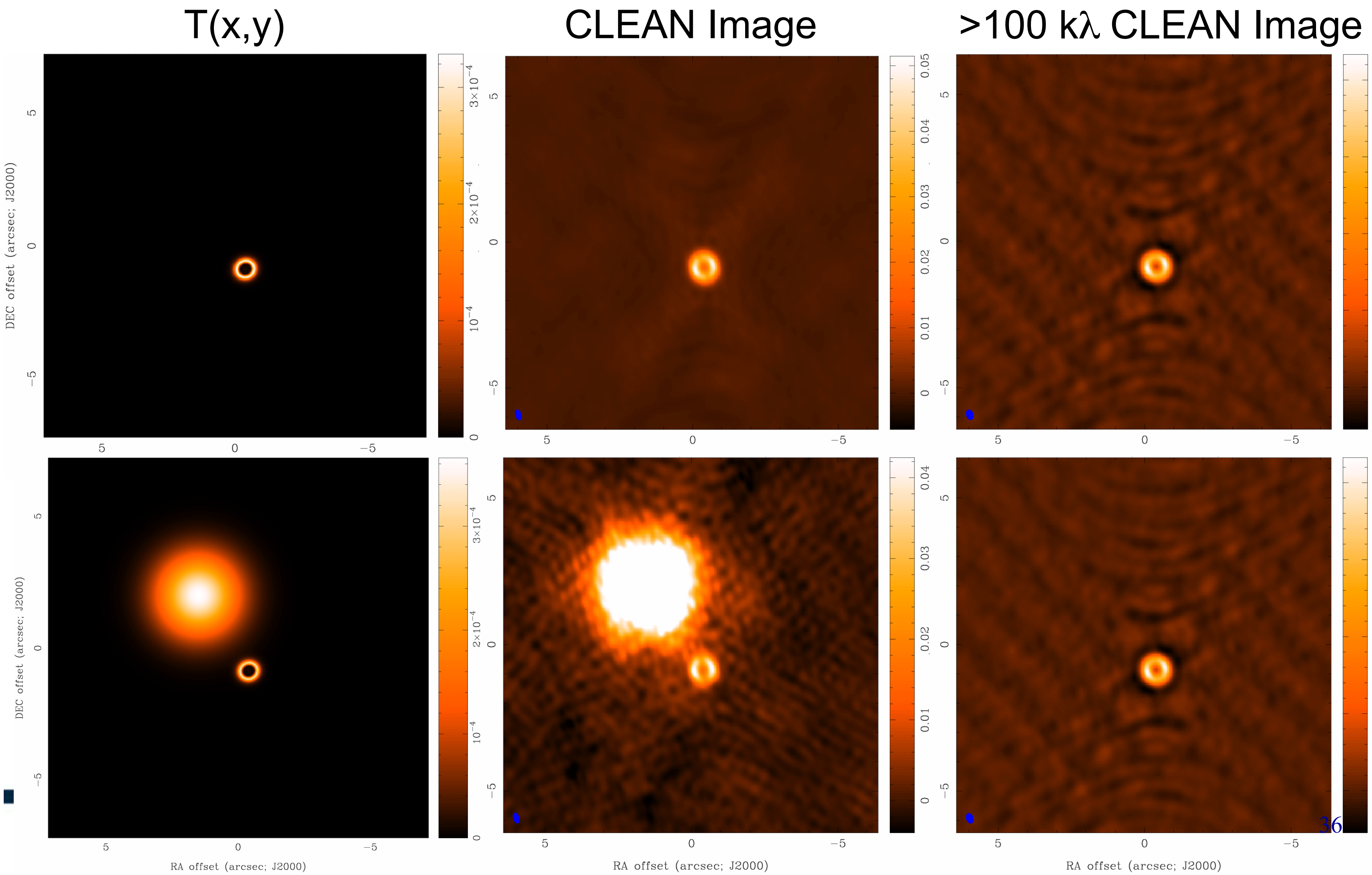
ALMA will be 10-100 times more sensitive and have 10-100 times better angular resolution compared to current millimeter interferometers

Data Volume

- Cycle 0 (16 antennas)
  - ~50G raw, few 100GB to process to a ~3600x3600x1024 cube
  - recommended workstation 4-8 cores w/24G RAM
- Full ALMA: up to 64 GB/s raw data
- Automated data reduction pipeline

- Simulated data is critically important to refine algorithms and increase efficiency

Missing Short Spacings: Demonstration

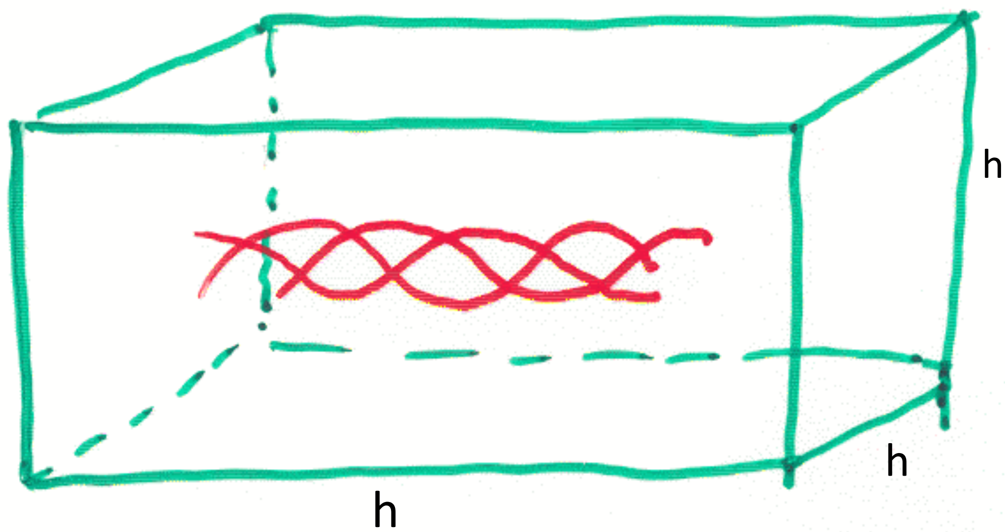


Tourist Guide to Interferometry Jargon

Optical/IR speak	Radio speak
Optical path difference (OPD)	Delay, lag
Differential piston	Delay residual
Beam combiner	Correlator
Strehl ratio	Antenna gain
Background level	System temperature
Fringe tracking	Phase referencing
Telescope	Antenna
Detector	Feed
Point spread function (PSF)	Dirty (or CLEAN) beam
Magnitudes	log (flux density)
Obscure band designations	Confusing band designations

From W. D. Cotton (in 'The Role of VLBI in Astrophysics, Astrometry, And Geodesy, ed Mantovani & Kus, Kluwer 2004)

Origin of wave noise: ‘Bunching of Bosons’ in phase space (time and frequency) allows for interference (ie. coherence).



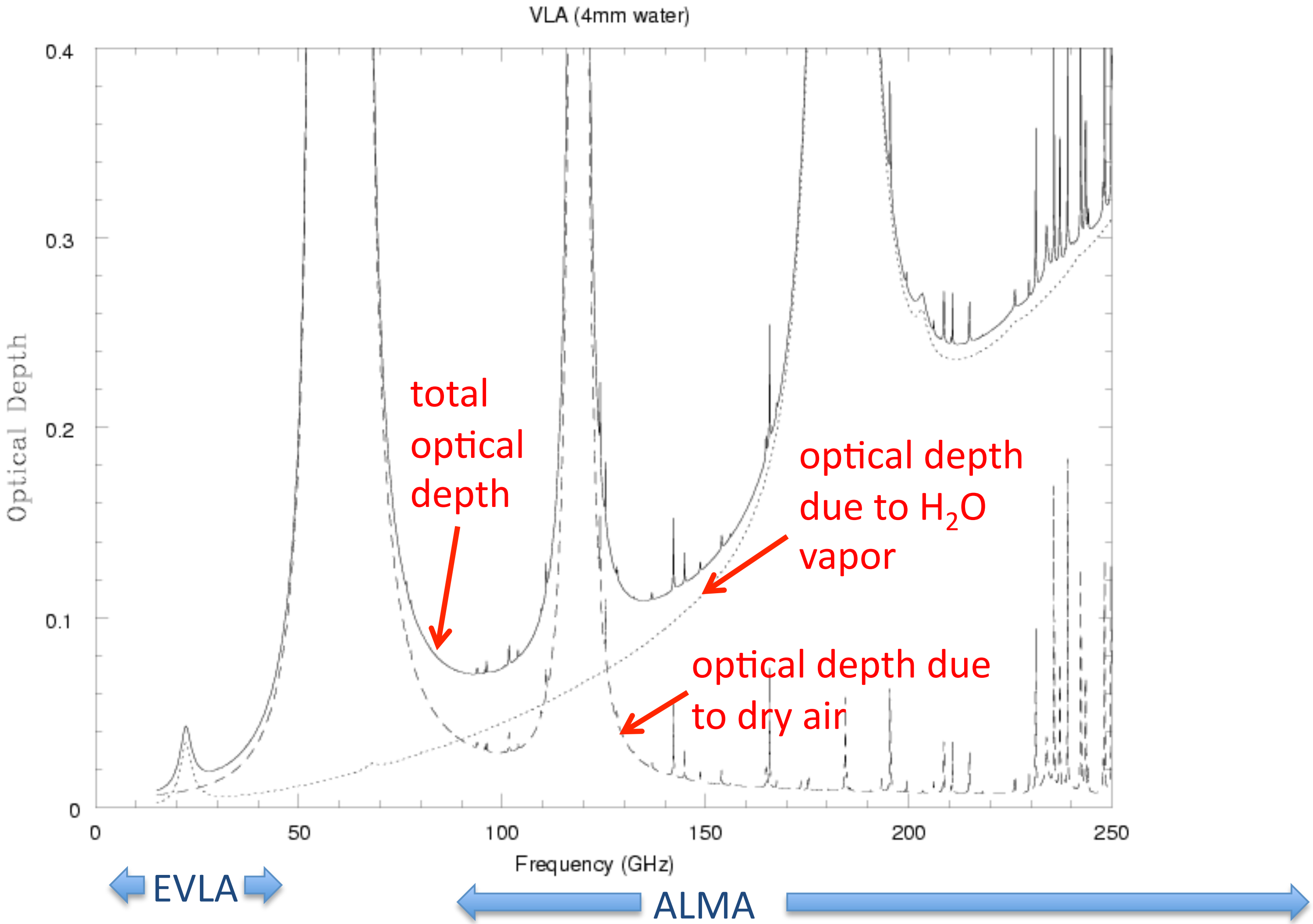
Bosons can, and will, occupy the exact same phase space if allowed, such that interference (destructive or constructive) will occur. Restricting phase space (ie. narrowing the bandwidth and sampling time) leads to interference within the beam. **This naturally leads to fluctuations that are proportional to intensity (= wave noise).**

Interferometric Radiometer Equation

$$S_{rms} = \frac{2kT_{sys}}{A_{eff}\sqrt{N_A(N_A - 1)t_{\mathrm{int}}\Delta\nu}}$$

- T_{sys} = wave noise for photons (RJ): rms $\propto$ total power
- A_{eff},k_B = Johnson-Nyquist noise + antenna temp definition
- $t\Delta\nu$ = # independent measurements of T_A/T_{sys} per pair of antennas
- N_A = # indep. meas. for array, or can be folded into A_{eff}

The Atmosphere

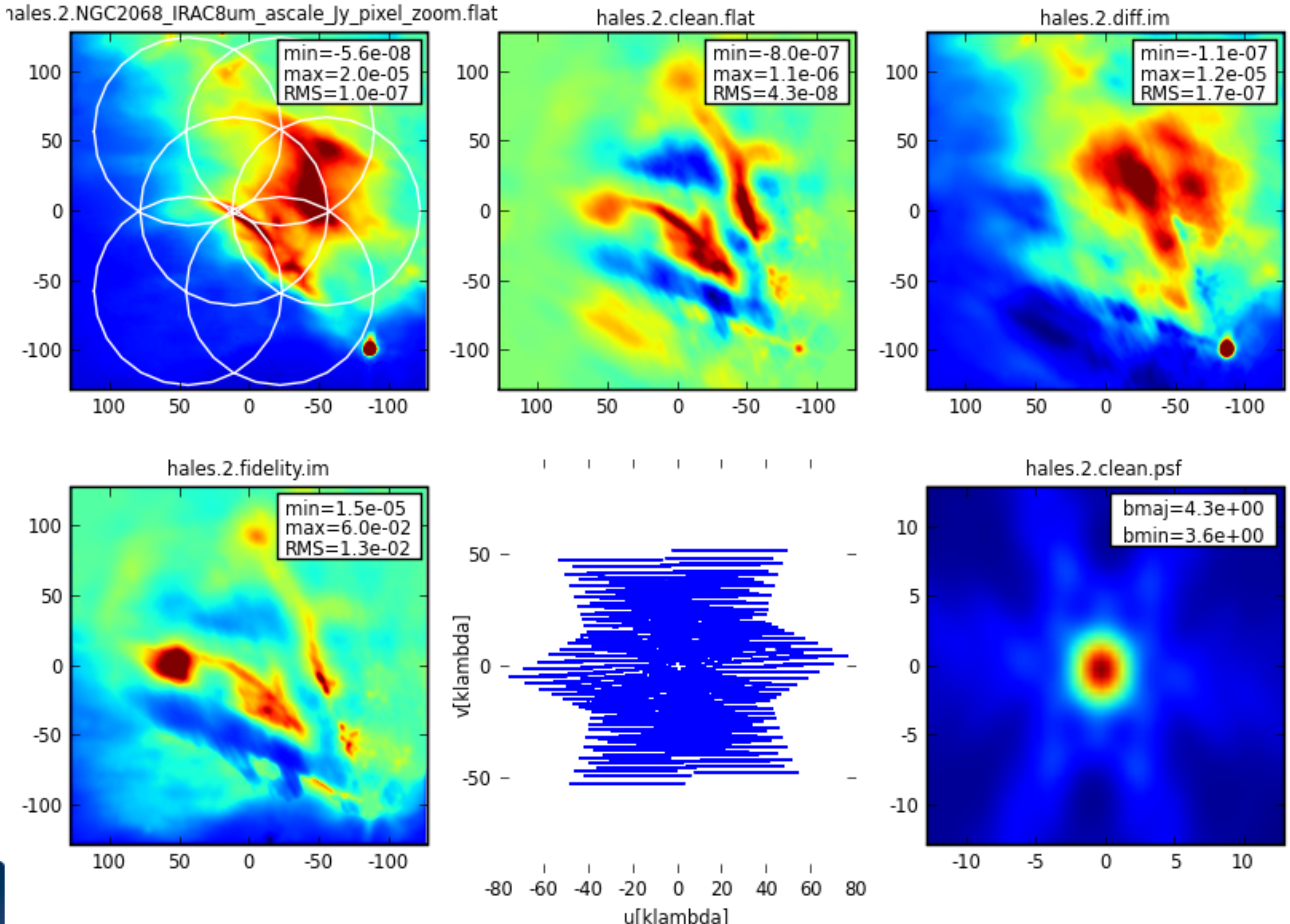


```
CASA <15>: inp
---------> inp()
#  sim_observe :: mosaic simulation task:
project             =      'sim'       #  root prefix for output file names
skymodel            = 'g41_I4.rd.fits' #  model image to observe
     inbright       =         ''       #  scale surface brightness of brightest pixel 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        =  '1.5arcmin'     #  angular size of map or "" to cover model
     maptype        =     'ALMA'       #  hexagonal, square, etc
     pointingspacing =        ''       #  spacing in between pointings or "0.25PB" or "" for 0.5 PB

observe             =       True       #  calculate visibilities using ptofile
     an                                                                      mulations
     re                                                                      nsit"
     ho
     to
     ca
     ca
     sd                                                                      wer MS
     sdant          =          0       #  single dish antenna index in file

thermalnoise        =  'tsys-atm'      #  add thermal noise: [tsys-atm|tsys-manual|""]
     user_pwv       =        1.0       #  Precipitable Water Vapor in mm
     t_ground       =      269.0       #  ambient temperature
     seed           =      11111       #  random number seed

leakage             =        0.0       #  cross polarization
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 sim_observe(...)
```

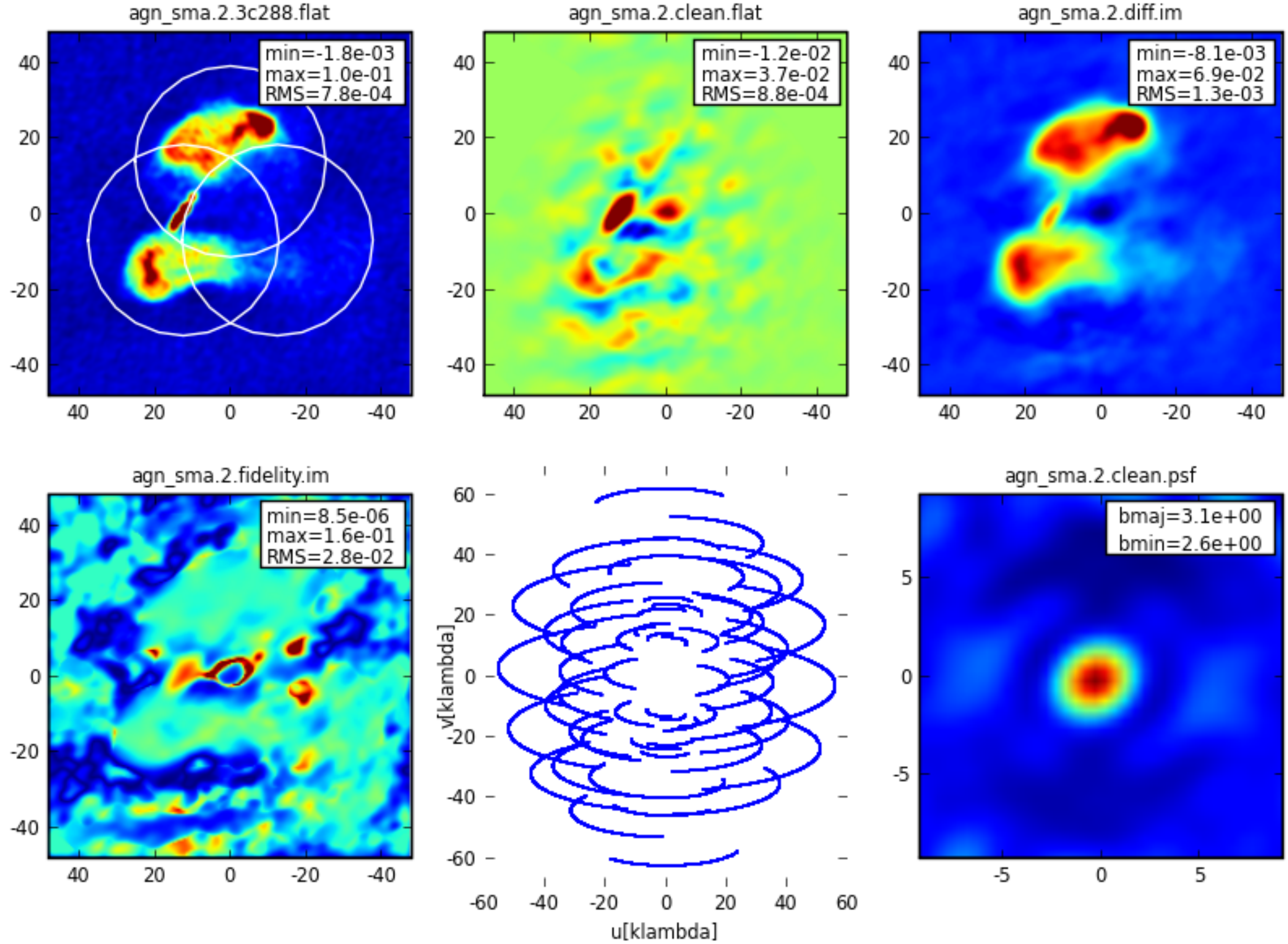
Any array for which the antenna positions are known can be simulated, including ones like SKA that don't exist yet.

A phase calibrator can be observed with the science target, to test data reduction strategies

EVLA


ɿales.2.NGC2068_IRAC8um_ascale_Jy_pixel_zoom.flat
min=-5.6e-08
max=2.0e-05
RMS=1.0e-07
hales.2.clean.flat
min=-8.0e-07
max=1.1e-06
RMS=4.3e-08
hales.2.diff.im
min=-1.1e-07
max=1.2e-05
RMS=1.7e-07
hales.2.fidelity.im
min=1.5e-05
max=6.0e-02
RMS=1.3e-02
v[klambda]
u[klambda]
hales.2.clean.psf
bmaj=4.3e+00
bmin=3.6e+00

SMA


agn_sma.2.3c288.flat
min=-1.8e-03
max=1.0e-01
RMS=7.8e-04
agn_sma.2.clean.flat
min=-1.2e-02
max=3.7e-02
RMS=8.8e-04
agn_sma.2.diff.im
min=-8.1e-03
max=6.9e-02
RMS=1.3e-03
agn_sma.2.fidelity.im
min=8.5e-06
max=1.6e-01
RMS=2.8e-02
v[klambda]
u[klambda]
agn_sma.2.clean.psf
bmaj=3.1e+00
bmin=2.6e+00