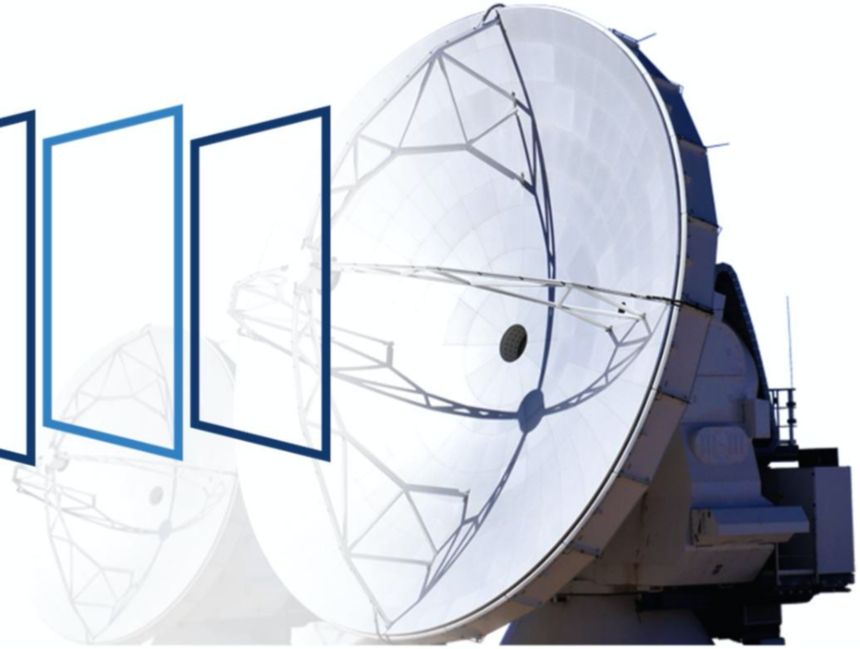
CASA

Common Astronomy Software Applications



Introduction to CASA

Bjorn Emonts

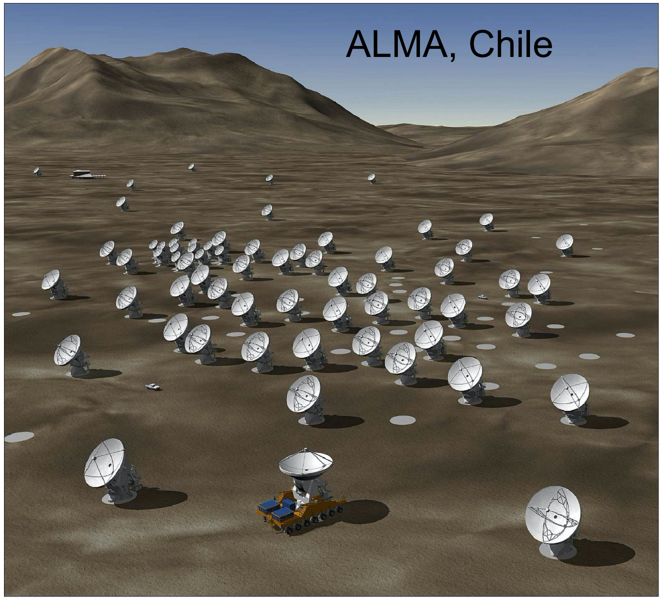
National Radio Astronomy Observatory
CASA User Community Liaison

Ryan Raba *(NRAO – CASA Lead)*

CASA

Common Astronomy Software Applications for Radio Astronomy

Primary data processing software *Karl G. Jansky Very Large Array (VLA)* and *Atacama Large Millimeter/submillimeter Telescope (ALMA),* but frequently used also for other radio telescopes.



NRAO ESO 国立天文台 NAOJ ASIAA ALMA CSIRO ASTRON JIVE Joint Institute for VLBI ERIC

CASA Team

Ryan Raba (NRAO-CV)	*CASA Lead*
Urvashi Rao Venkata (NRAO-SO)	*Deputy Lead, Lead Scientific development*
Ben Bean (NRAO-ALBQ)	*Scientific development, Verification testing*
Sandra Castro (ESO)	*Lead verification testing*
Bjorn Emonts (NRAO-CV)	*User Community Liaison*
Enrique Garcia (ESO)	*Infrastructure development*
Bob Garwood (NRAO-CV)	*Infrastructure, Verification testing*
Kumar Golap (NRAO-SO)	*Scientific development*
Justo Gonzalez Villalba (ESO)	*Scientific development*
Pam Harris (NRAO-SO)	*Data visualization*
Yohei Hayashi (NAOJ)	*Scientific development, Single Dish*
Josh Hoskins (NRAO-CV)	*Scientific development, Infrastructure*
Wataru Kawasaki (NAOJ)	*Scientific development, Single Dish*
Jorge Lopez (NRAO-CV)	*Infrastructure, Scientific development*
Josh Marvil (NRAO-SO)	*Lead scientific validation*
Andrew McNichols (NRAO-CV)	*Infrastructure, Scientific development*
Dave Mehringer (NRAO-CV)	*Scientific development, Verification testing*
Renaud Miel (NAOJ)	*Scientific development, Single Dish*
George Moellenbrock (NRAO-SO)	*Scientific development*
Federico Montesino (ESO)	*Infrastructure, Scientific development*
Takeshi Nakazato (NAOJ)	*Scientific development, Single Dish*
Dirk Petry (ESO)	*Scientific development*
Darrell Schiebel (NRAO-CV)	*Lead Visualization, Infrastructure development*
Neal Schweighart (NRAO-CV)	*Scientific development, Verification testing*
Kazuhiko Shimada (NAOJ)	*Scientific development, Single Dish*
Jan-Willem Steeb (NRAO-CV)	*Scientific, Infrastructure development*
Ville Suoranta (NRAO-CV)	*Infrastructure development*
Tak Tsutsumi (NRAO-SO)	*Scientific development, Verification testing*
Akeem Wells (NRAO-CV)	*Verification testing*
Wei Xiong (NRAO-ALBQ)	*Infrastructure, Scientific development*

JIVE
Joint Institute for VLBI
ERIC

Ilse van Bemmel (JIVE)	*VLBI, Project Scientist, pipeline/user training*
Mark Kettenis (JIVE)	*VLBI, development*
Des Small (JIVE)	*VLBI, development*
Arpad Szomoru (JIVE)	*VLBI, management*
Marjolein Verkouter (JIVE)	*VLBI, management*
Aard Keipema (JIVE)	*VLBI, Jupyter kernel*

CARTA visualization software team

Pipeline teams
(ALMA, VLA, Nobeyama)

CASA Stakeholders

- ALMA Users
- ALMA Pipeline
- VLA Users
- SRDP (Science Ready Data Products, NRAO)
- VLBA/VLBI
- ARDG (Algorithm Research and Development Group, NRAO)
- General Users

CASA Users Committee

- Feedback CASA capabilities, usability, reliability and performance
- Advise CASA development team from user perspective
- Inform development priorities

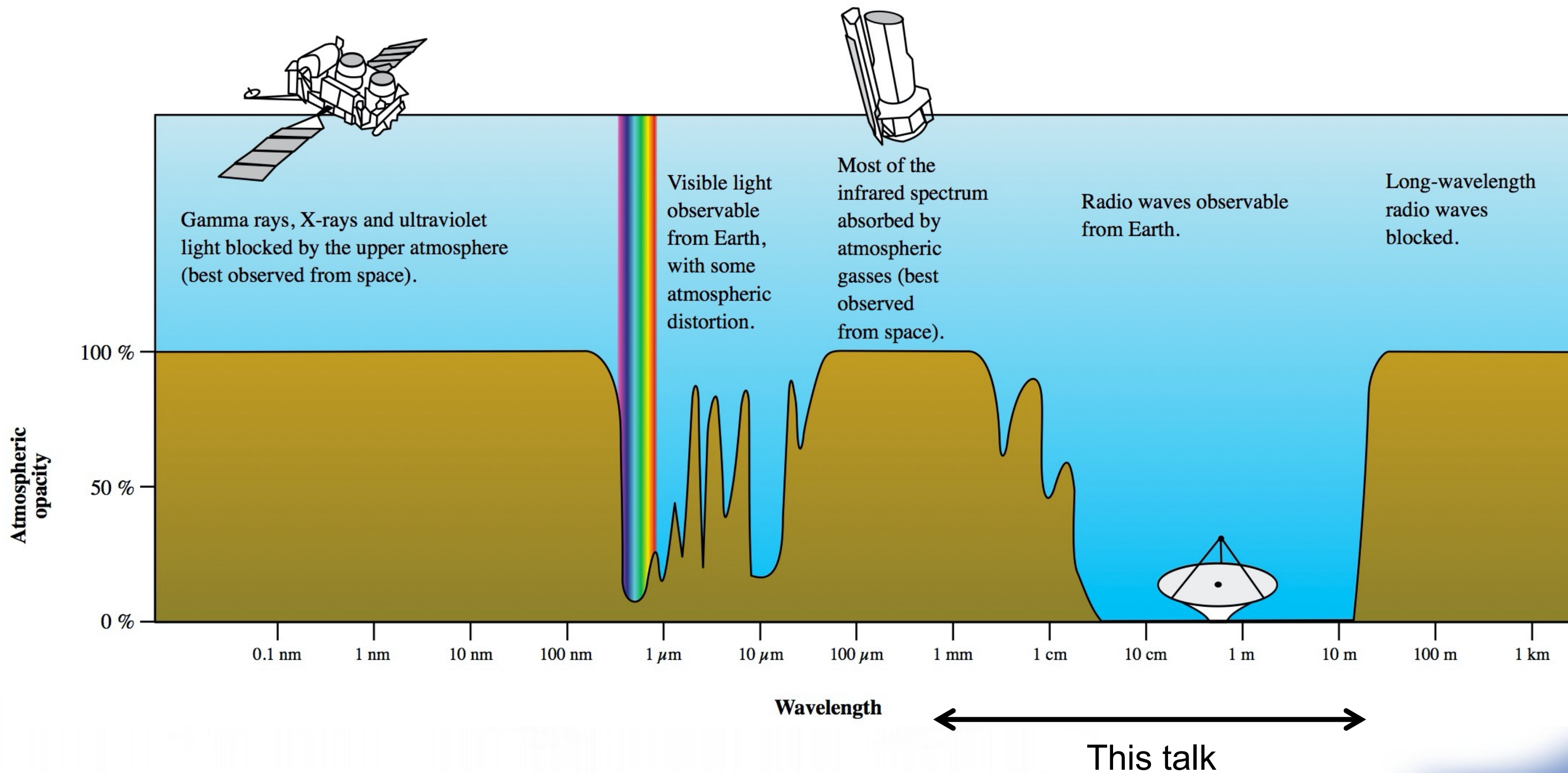
This Talk

- Radio astronomy
  - Basics of Radio Interferometry

- CASA
  - Basics
  - Installation
  - Calibration
  - Imaging
  - Visualization (CARTA)
  - CASA Documentation
  - Resources

- Future of radio astronomy
  - CASA Next Generation Infrastructure (CNGI)

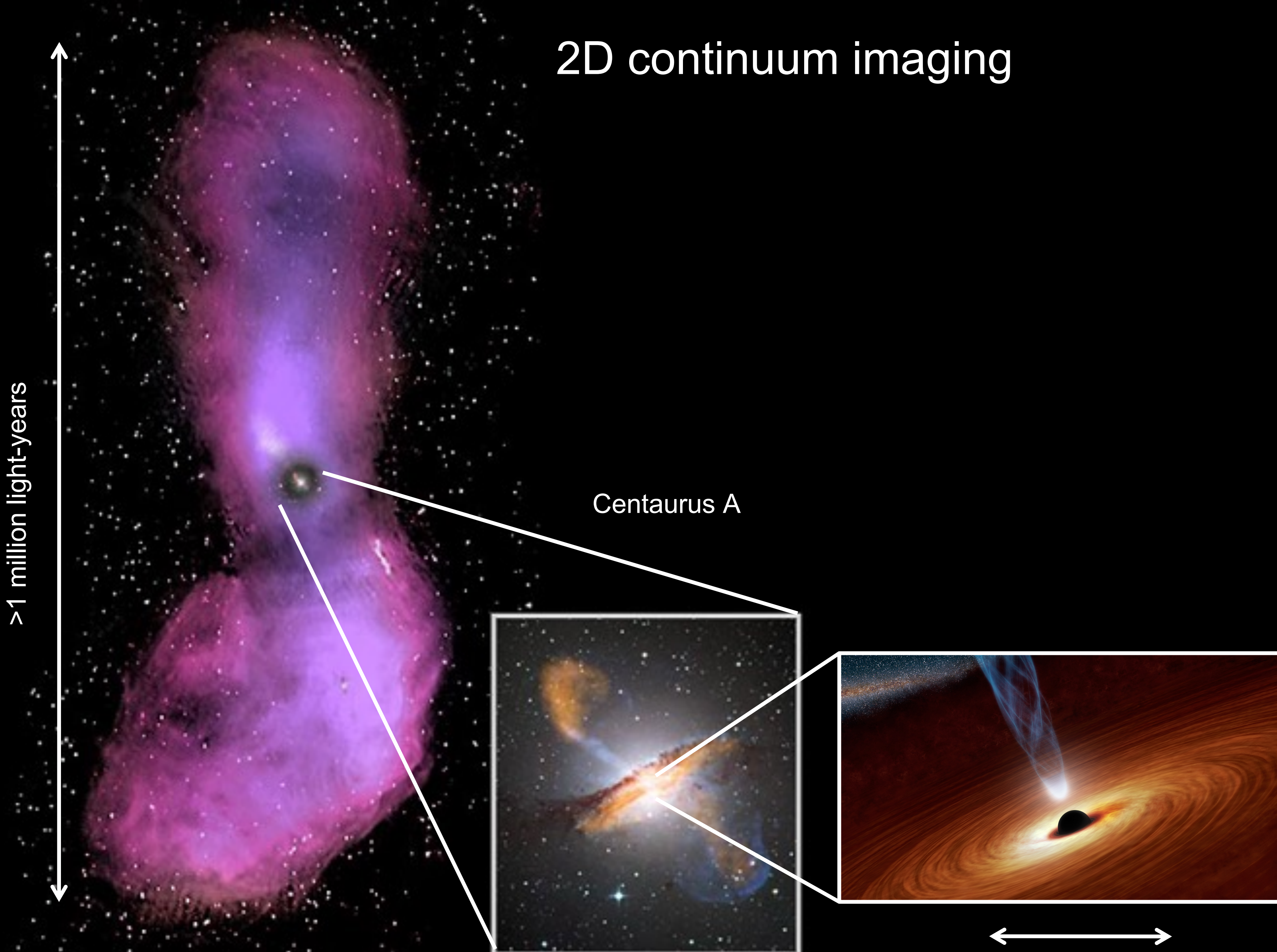
Radio astronomy



NRAO ESO NAOJ ASIAA ALMA CSIRO ASTRON JIVE Joint Institute for VLBI ERIC

Credit: I. Feain, T. Cornwell, R. Ekers (CSIRO/ATNF); ATCA middle lobe pointing courtesy R. Morganti (ASTRON); Parkes data courtesy N. Junkes (MPIfR); ATCA & Moon photo: S. Amy, CSIRO

2D continuum imaging



Feain⁺ (2011)

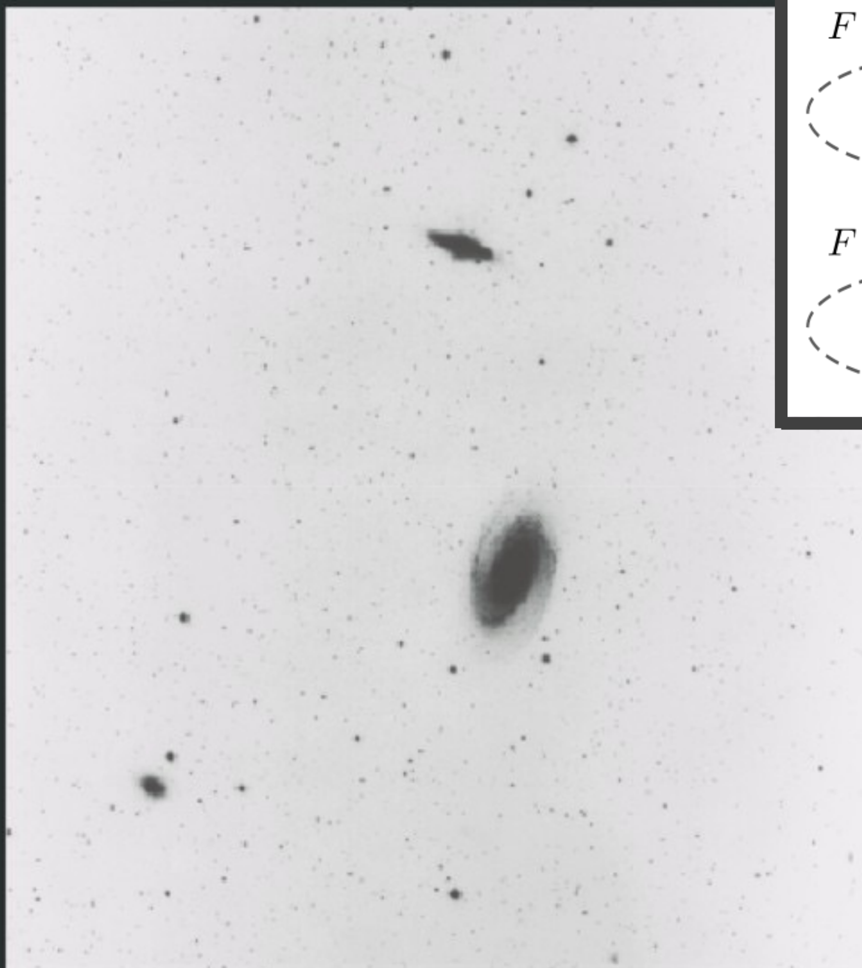
3D image cubes: Gas in the M81 group

Stellar Light Distribution

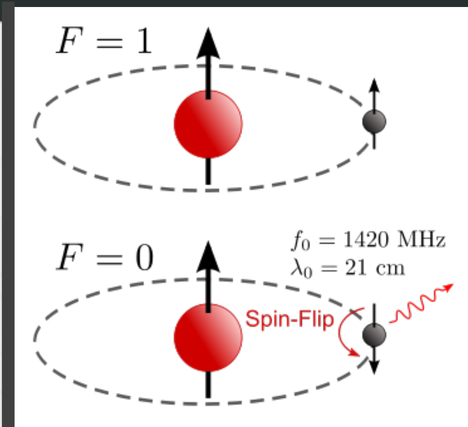
21 cm HI Distribution

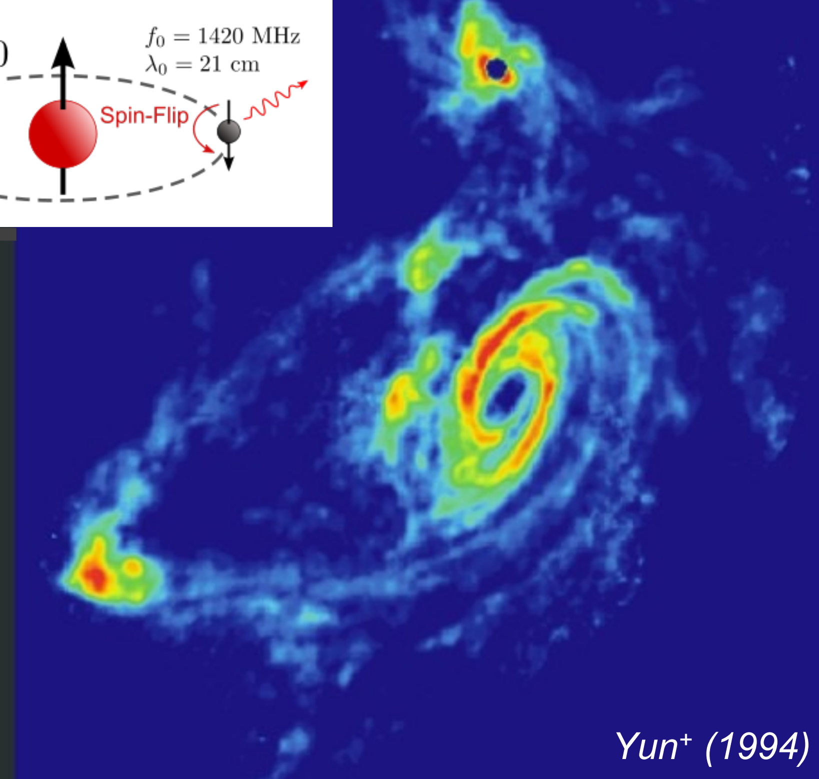
3D image cubes: Gas in the M81 group

Stellar Light Distribution

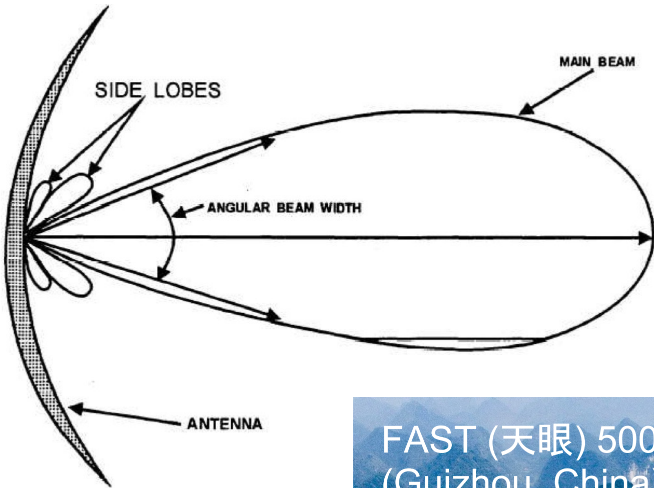


21 cm HI Distribution



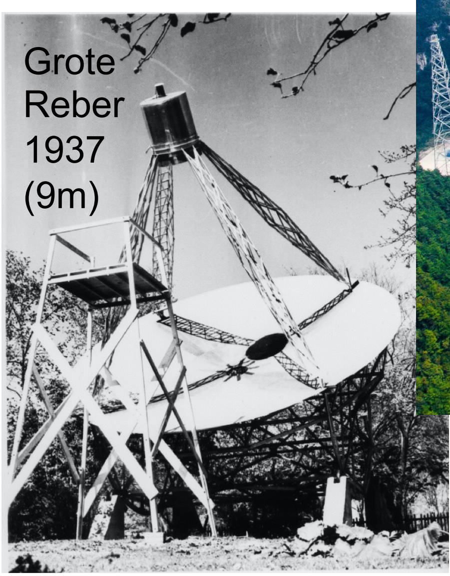


Single dish telescopes

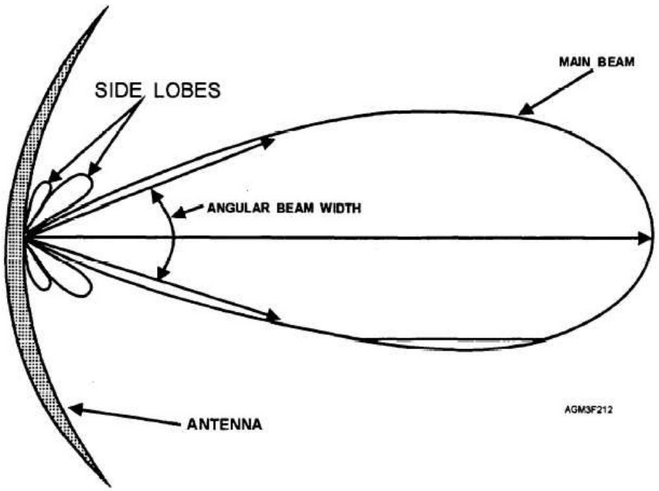


Resolution: **R ~ λ / Dish-size**

($\lambda_{radio} \sim 10^{3\text{-}7} \times \lambda_{optical}$)

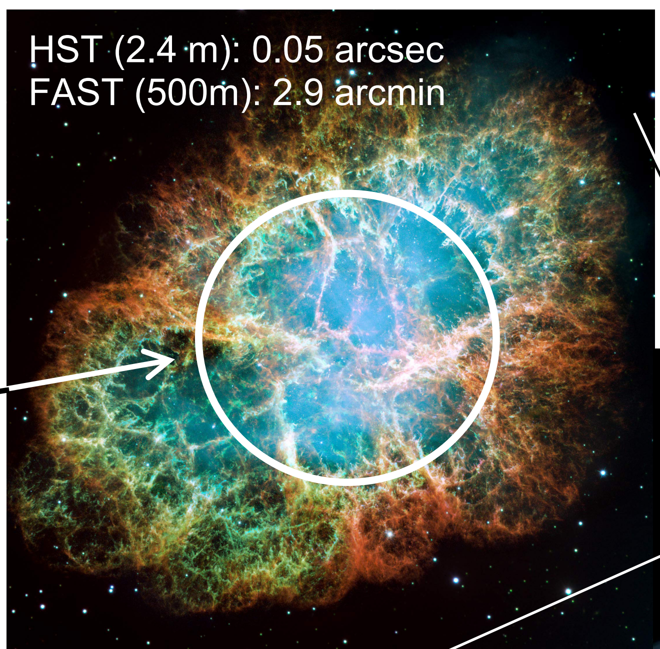


Single dish telescopes

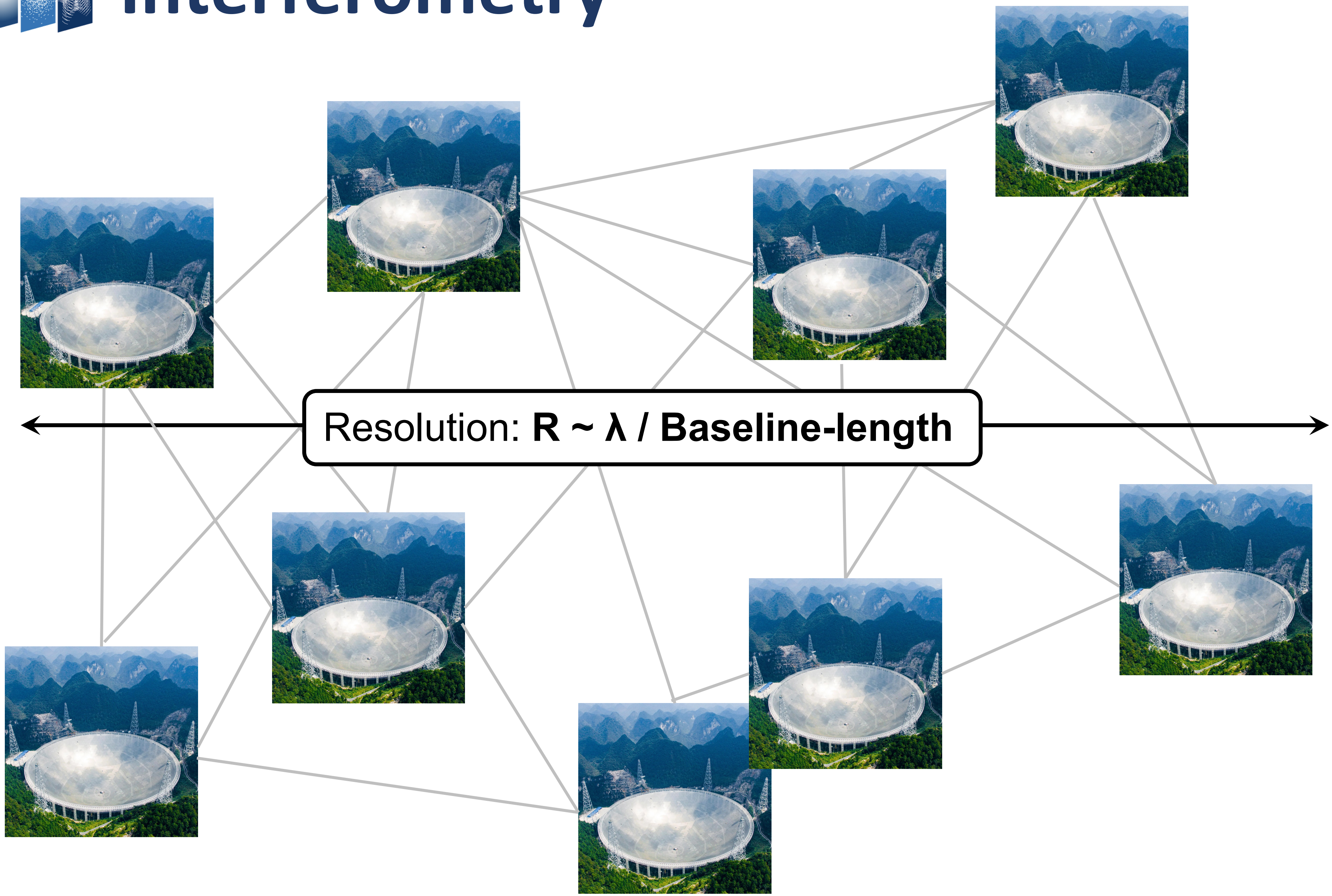


Resolution: R ~ λ / Dish-size

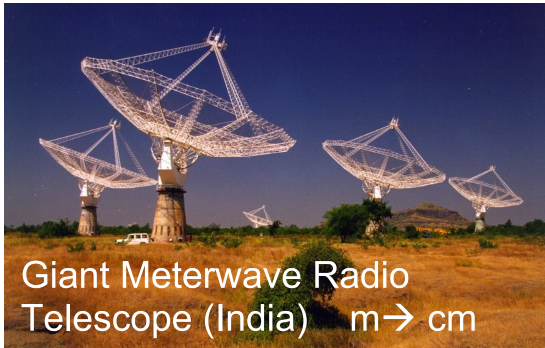
($\lambda_{radio} \sim 10^{3\text{-}7} \times \lambda_{optical}$)



Interferometry



Radio interferometers



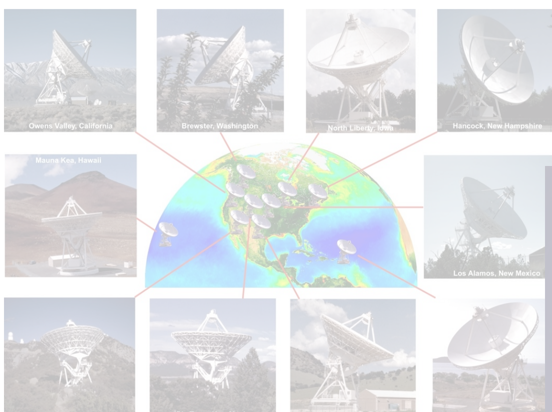
Giant Meterwave Radio Telescope (India) m→ cm

Human-made interference (phones, TV, satellite, etc)

Very Large Array (USA)
cm → 3 mm

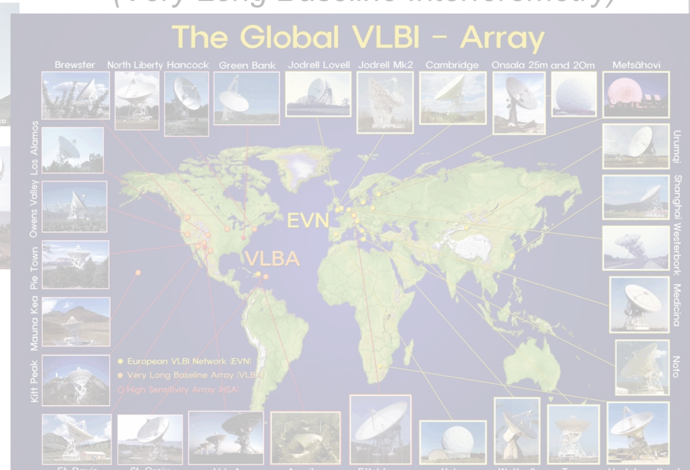
Atacama Large Millimeter Array (Chile)
3 mm → 300 micron

Atmosphere/weather (high sites)



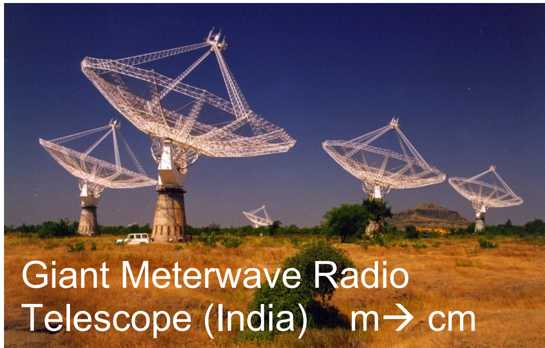
Very Long Baseline Array, VLBA (USA)

Global VLBI
(Very Long Baseline Interferometry)



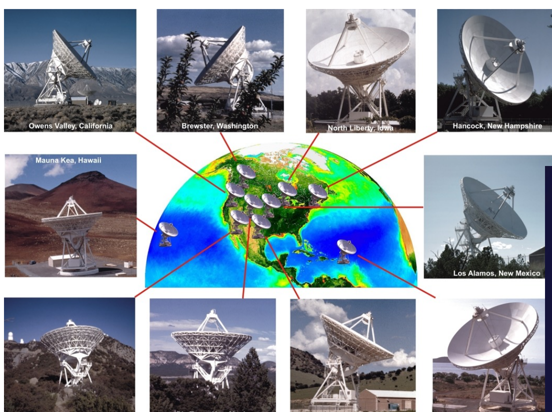
Space VLBI (D >20.000 km!)

Radio interferometers



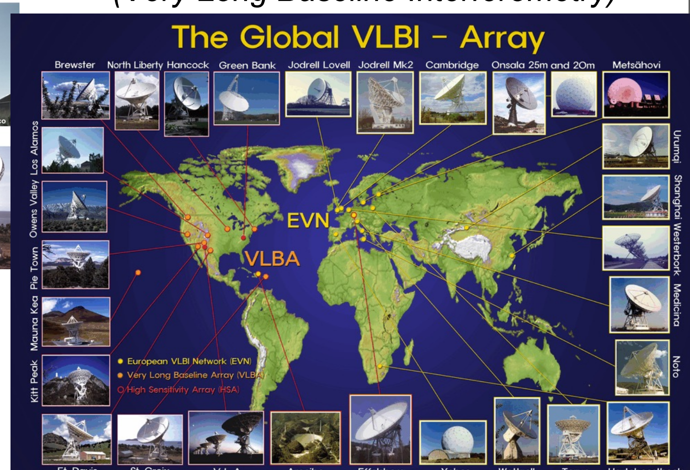
Human-made interference (phones, TV, satellite, etc)

Atmosphere/weather (high sites)



Very Long Baseline Array, VLBA (USA)

Global VLBI
(Very Long Baseline Interferometry)



Space VLBI (D >20.000 km!)

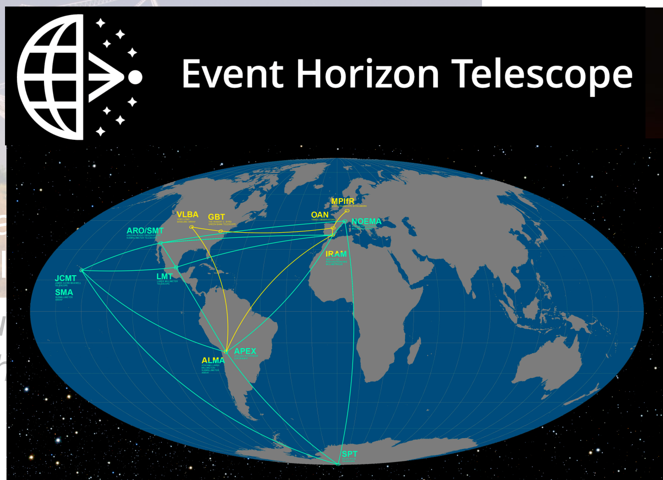
Radio interferometers

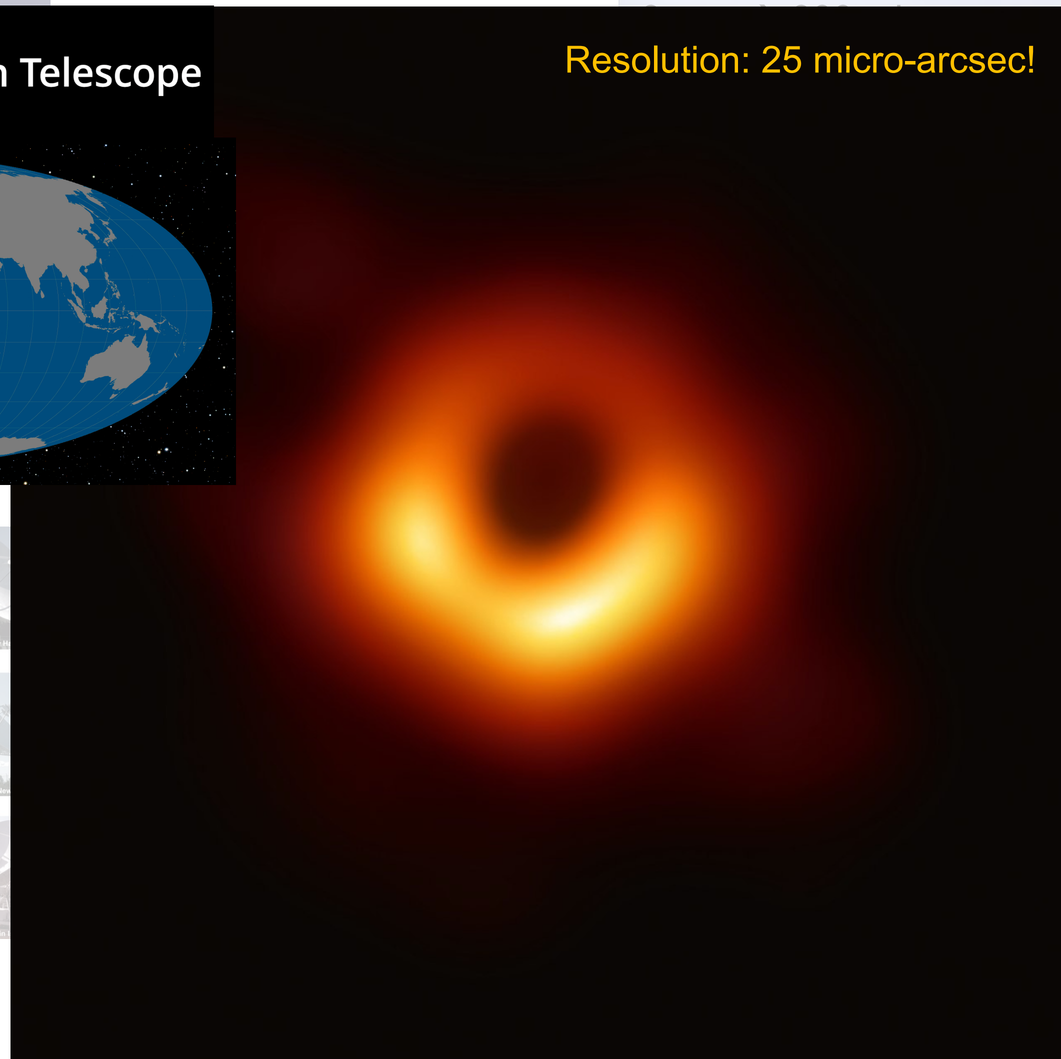
Gia
Tel

Hu
(ph

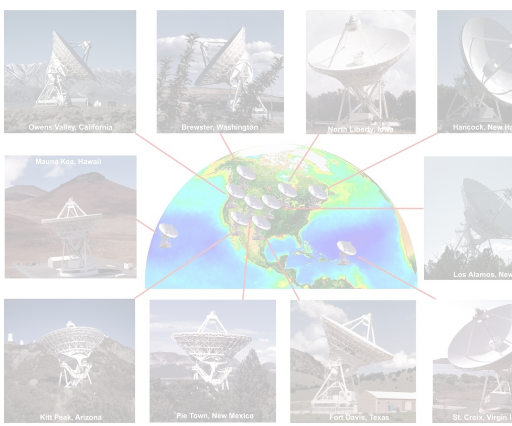
Atacama Large Millimeter Array (Chile)

r (high sites)



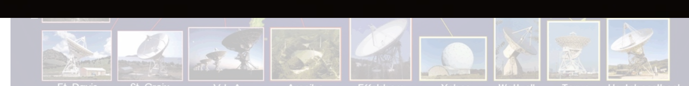


Credit: EHT Collaboration

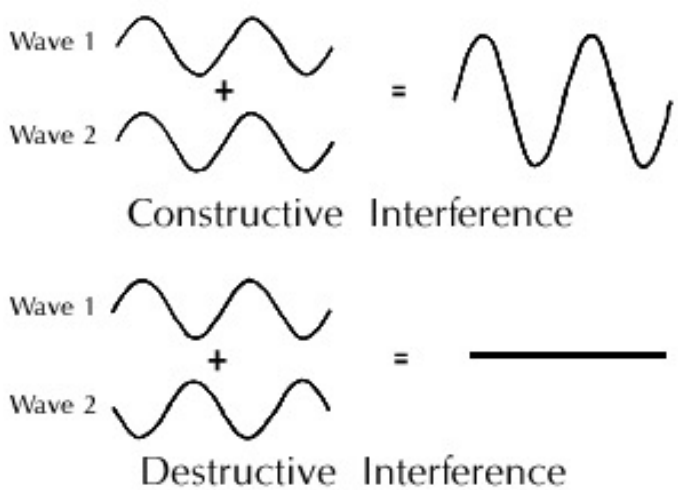


Very Long Baseline Array, VLBA (USA)

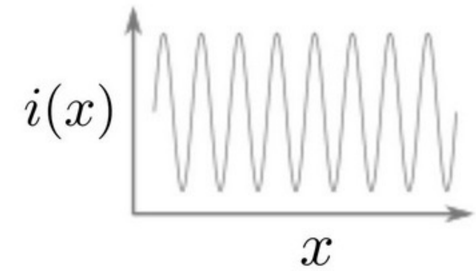
>20.000 km!)

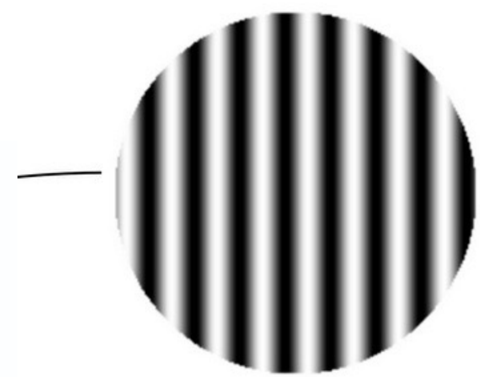


Interferometry

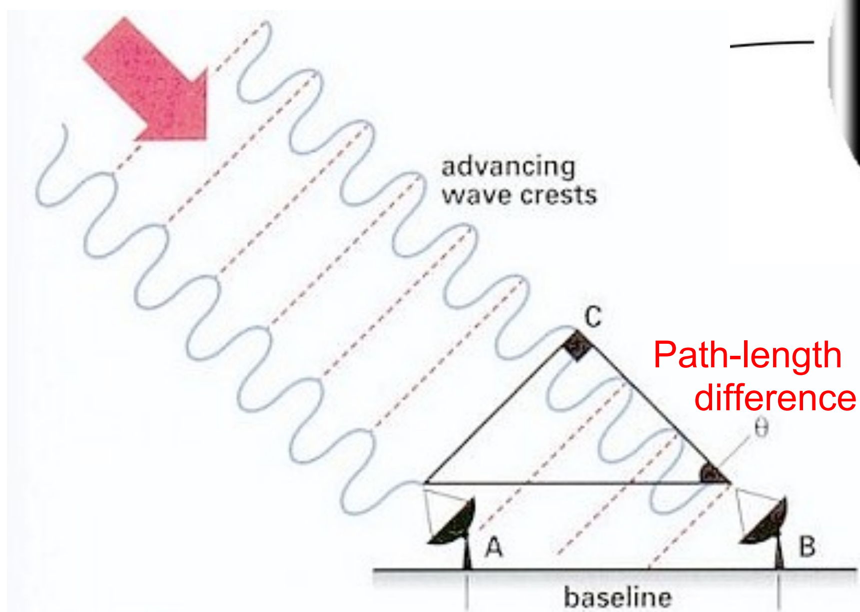


Track point-source on the sky:

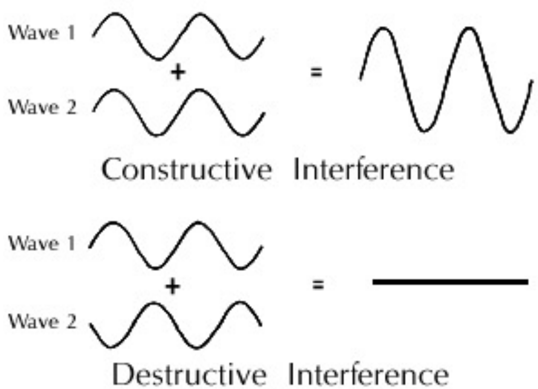


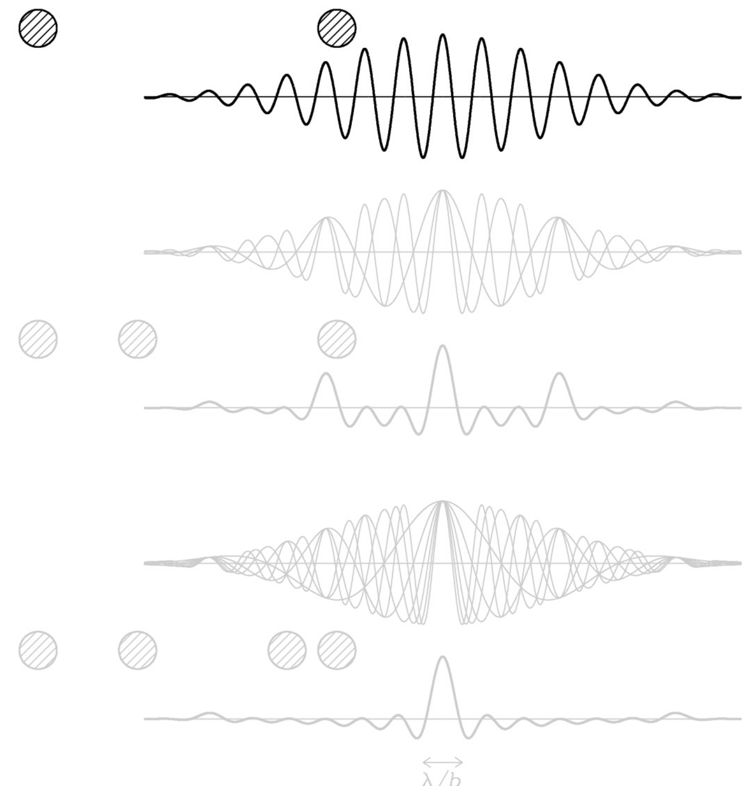


Resolution: **R ~ λ / B**



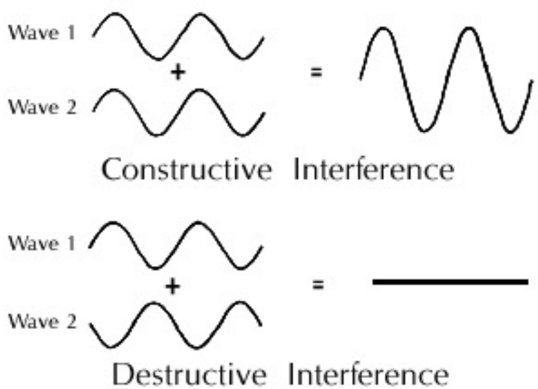
Interferometry

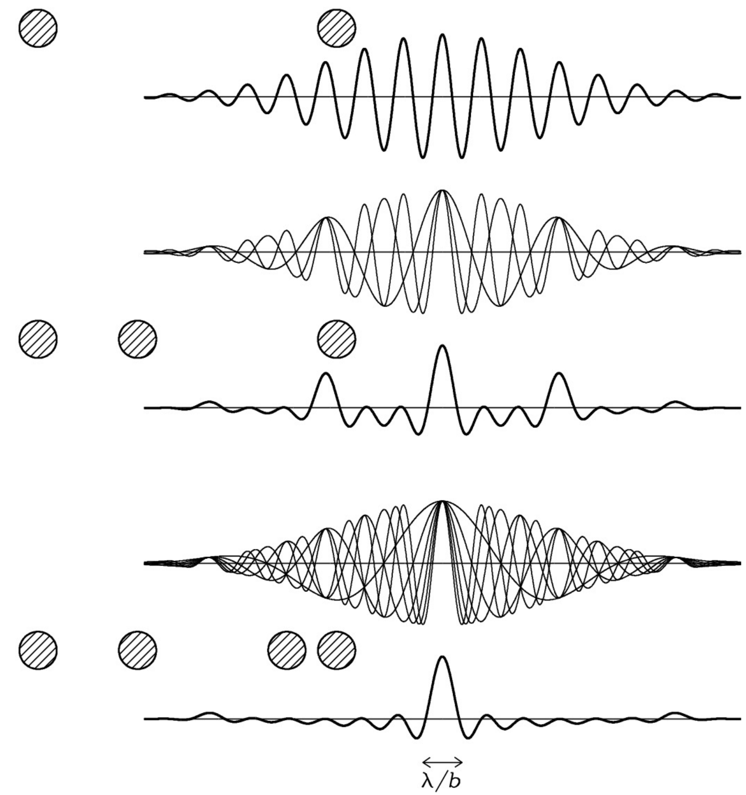




- Primary beam response single antenna (bandwidth)
- Add N antennas, i.e. N(N-1)/2 baselines
- Add complex source structure
- Scan through earth rotation

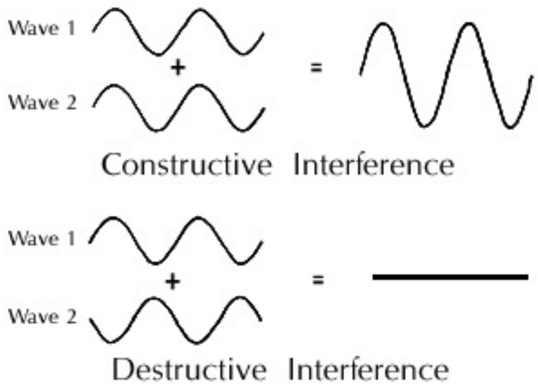
Interferometry

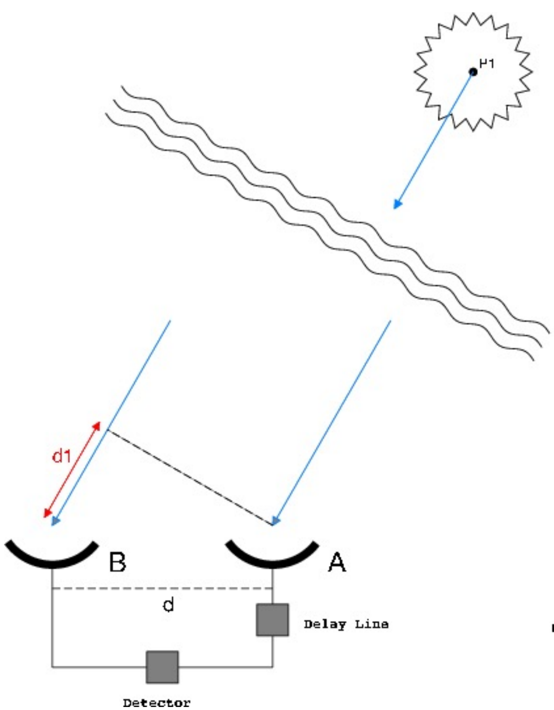


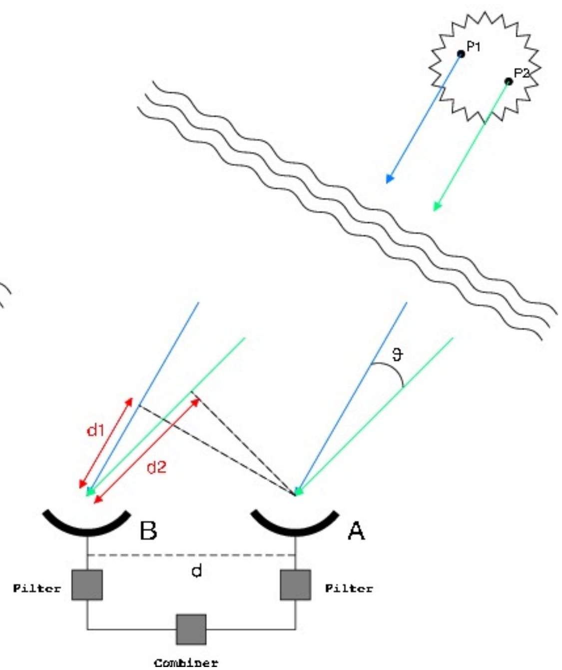


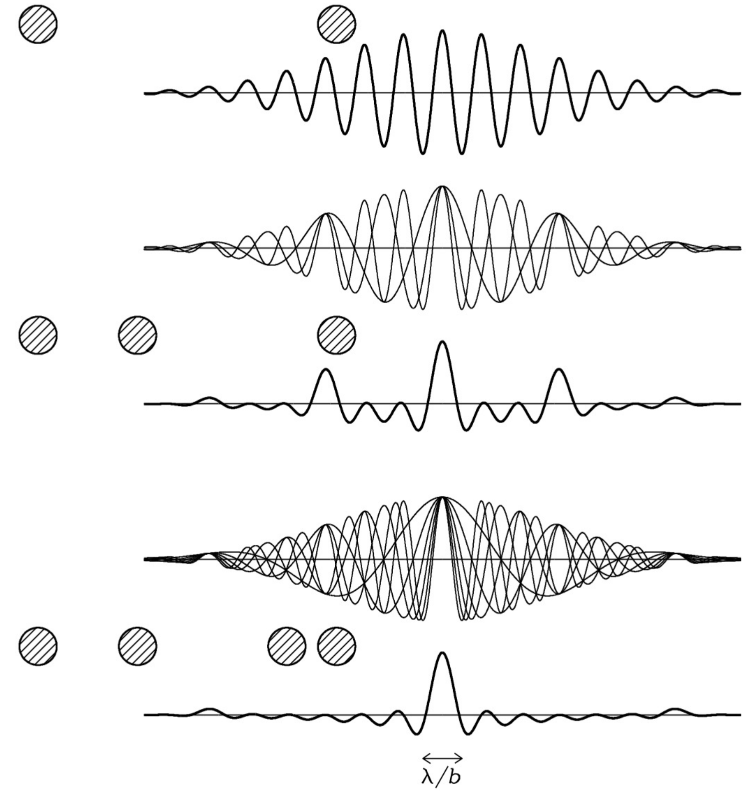
- Primary beam response single antenna (bandwidth)
- Add N antennas, i.e. N(N-1)/2 baselines
- Add complex source structure
- Scan through earth rotation

Interferometry



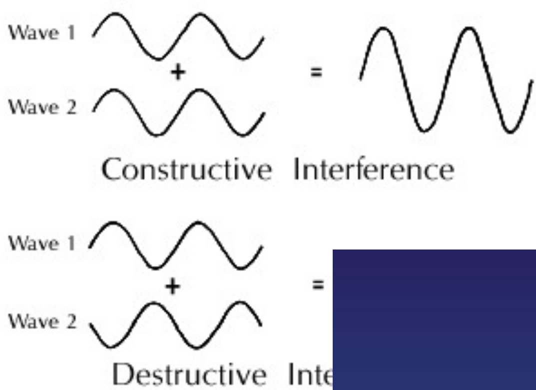


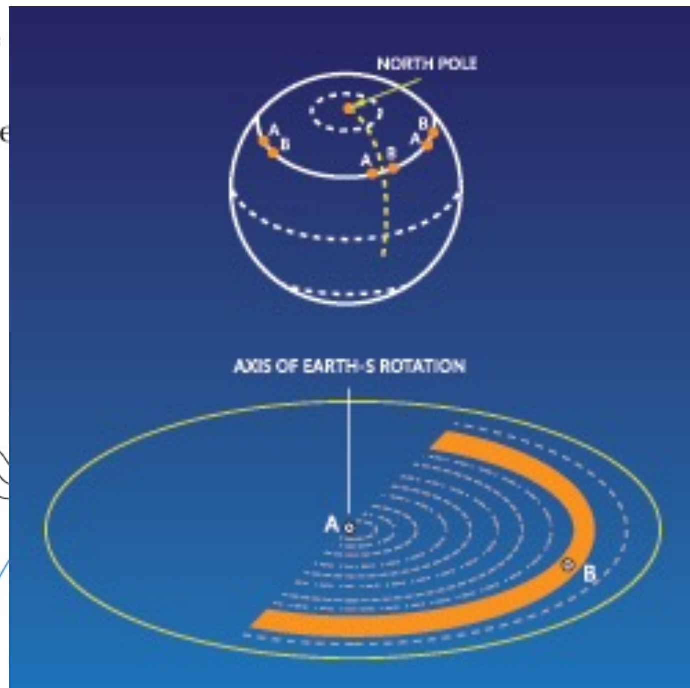


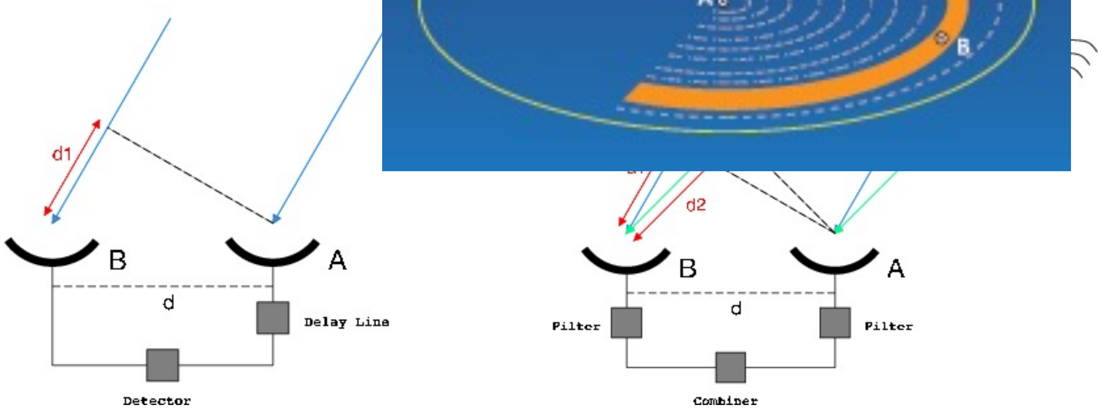


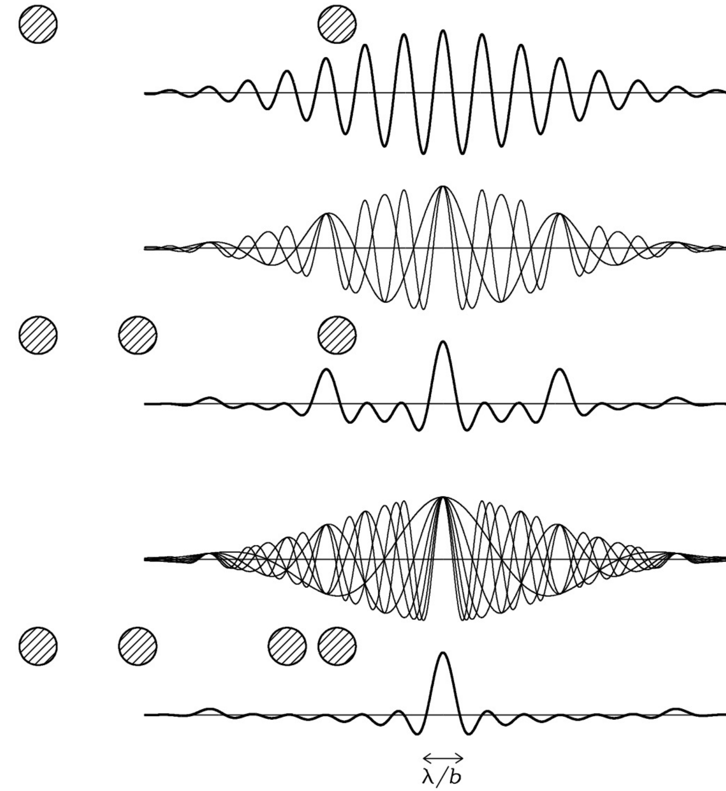
- Primary beam response single antenna (bandwidth)
- Add N antennas, i.e. N(N-1)/2 baselines
- Add complex source structure
- Scan through earth rotation

Interferometry









- Primary beam response single antenna (bandwidth)
- Add N antennas, i.e. N(N-1)/2 baselines
- Add complex source structure
- Scan through earth rotation

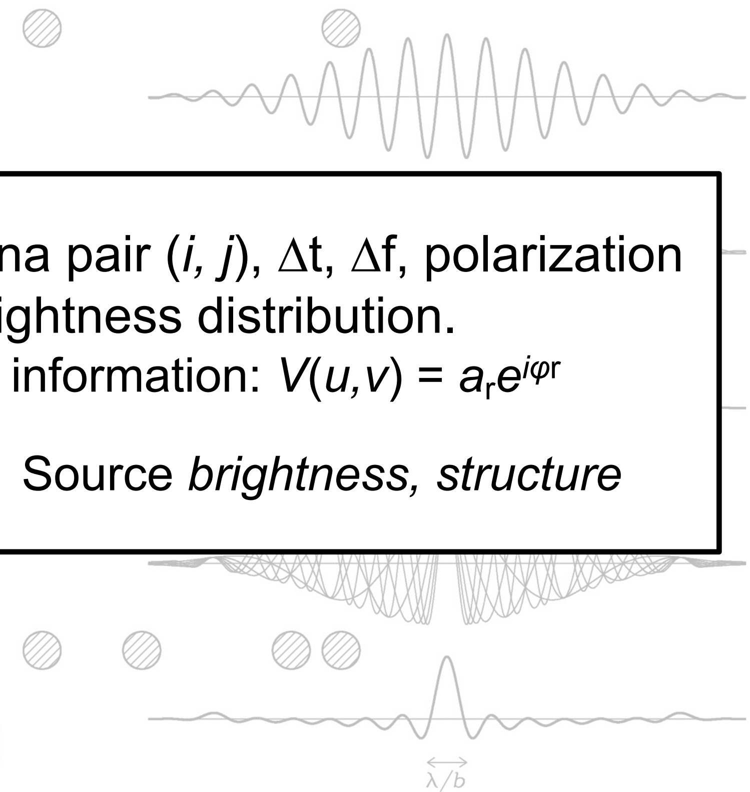
Interferometry

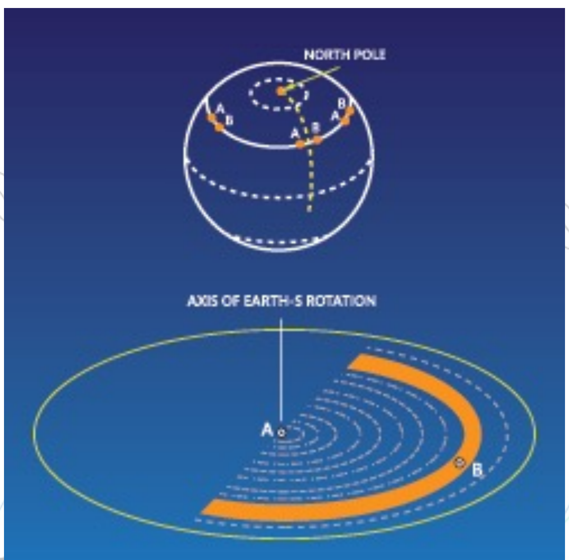
"Visibility": interferometer response per antenna pair (*i, j*), Δt, Δf, polarization

→ Fourier Transform of the sky brightness distribution.

→ "complex", with *amplitude, phase* information: $V(u,v) = a_r e^{i\varphi r}$

→ Source *brightness, structure*





- Primary beam response single antenna (bandwidth)
- Add N antennas, i.e. N(N-1)/2 baselines
- Add complex source structure
- Scan through earth rotation

Interferometry

"Visibility": interferometer response per antenna pair (*i, j*), Δt, Δf, polarization

→ Fourier Transform of the sky brightness distribution.

→ "complex", with *amplitude, phase* information: $V(u,v) = a_r e^{i\varphi r}$

→ Source *brightness, structure*

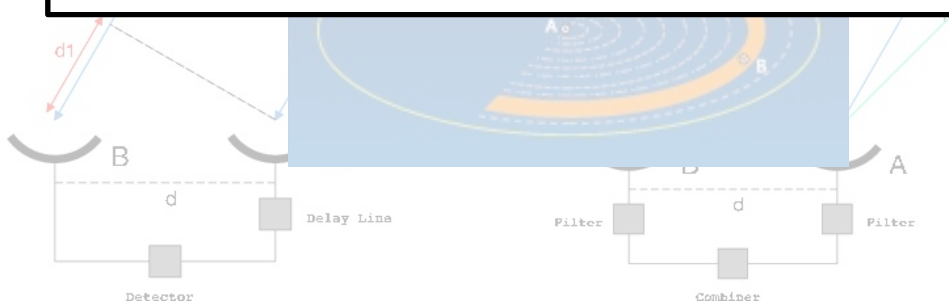
- Large data sets!

  Example: N(N-1) baselines, Δt ~ 3s over 12h, 2000 Δf channels, 4 polarizations, ….

- Discrete sampling ('broken mirror') → missing information

  Iteratively reconstruct model of sky

Computationally expensive!



- Add complex source structure
- Scan through earth rotation

CASA Basics

Single Dish & Interferometry

Casacore → original AIPS++ libraries, stable and nearly static platform

CASA → Implemented in C++, accessible through IPython

- **Tools**: basic C++ functions linked to Python interface that perform *basic operations* on data
- **Tasks**: bundle tools or Python functionality that perform *well-defined step in data processing* → user friendly, parameter input
- **GUIs**: Graphical User Interfaces to visualize and examine data/images
- **External**: Repository Measures Tables *(Earth Orientation Parameters, reference frames, ephemeris data, beam models, simulator configuration files, etc).* Minimal repo in CASA, update manually for more accuracy

Scripting & Pipelines *(ALMA calibration & imaging, VLA calibration, VLA Sky Survey)*

CASA

Import/export

Information

Manipulation

Calibration

Imaging

Analysis

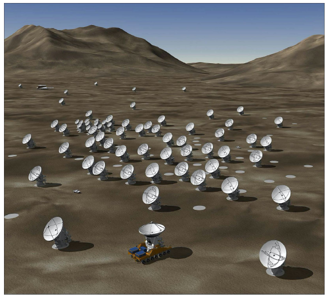
Single Dish

Simulations

CASA

Import/export
Information
Manipulation
Calibration
Imaging
Analysis

Single Dish
Simulations



(A)SDM, (ALMA) Science Data Model:
- Observing data: data from correlator (visibilities)
- Metadata: data that describes observations (source info, spectral setup, etc.)
- Auxiliary data: monitoring data observations (weather, pointing, etc.)

↓

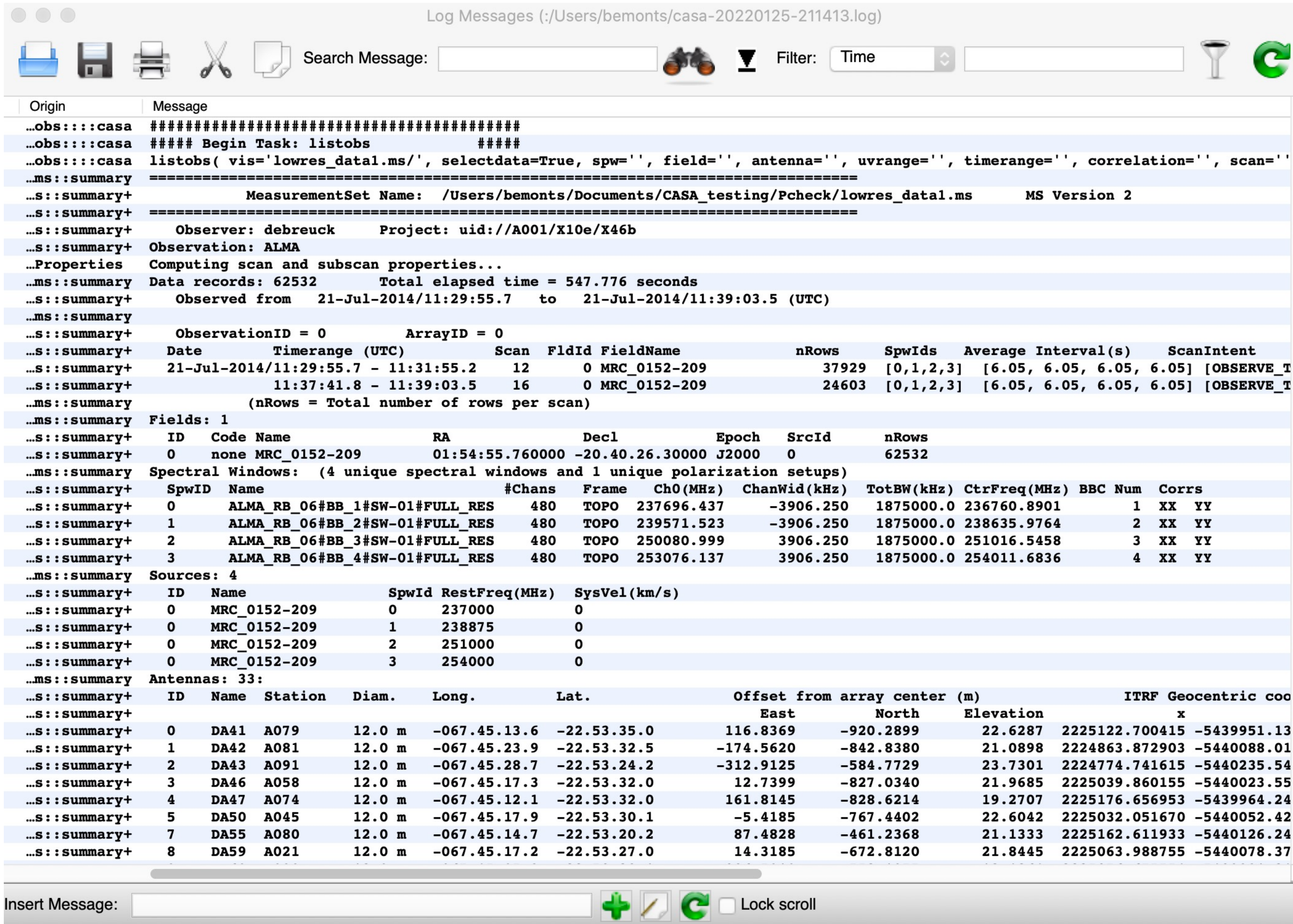
CASA MeasurementSet (MS):
Data directory with tables and subtables
- DATA column → data
- MODEL_DATA column → expected model values
- CORRECTED_DATA column → calibrated data
- Other optional columns: FLAG, SIGMA, WEIGHTS, etc.

NRAO ESO 国立天文台 NAOJ ASIAA ALMA CSIRO ASTRON JIVE Joint Institute for VLBI ERIC

CASA

- Import/export
- **Information**
- Manipulation
- Calibration
- Imaging
- Analysis

- Single Dish
- Simulations



CASA

Import/export

Information

Manipulation

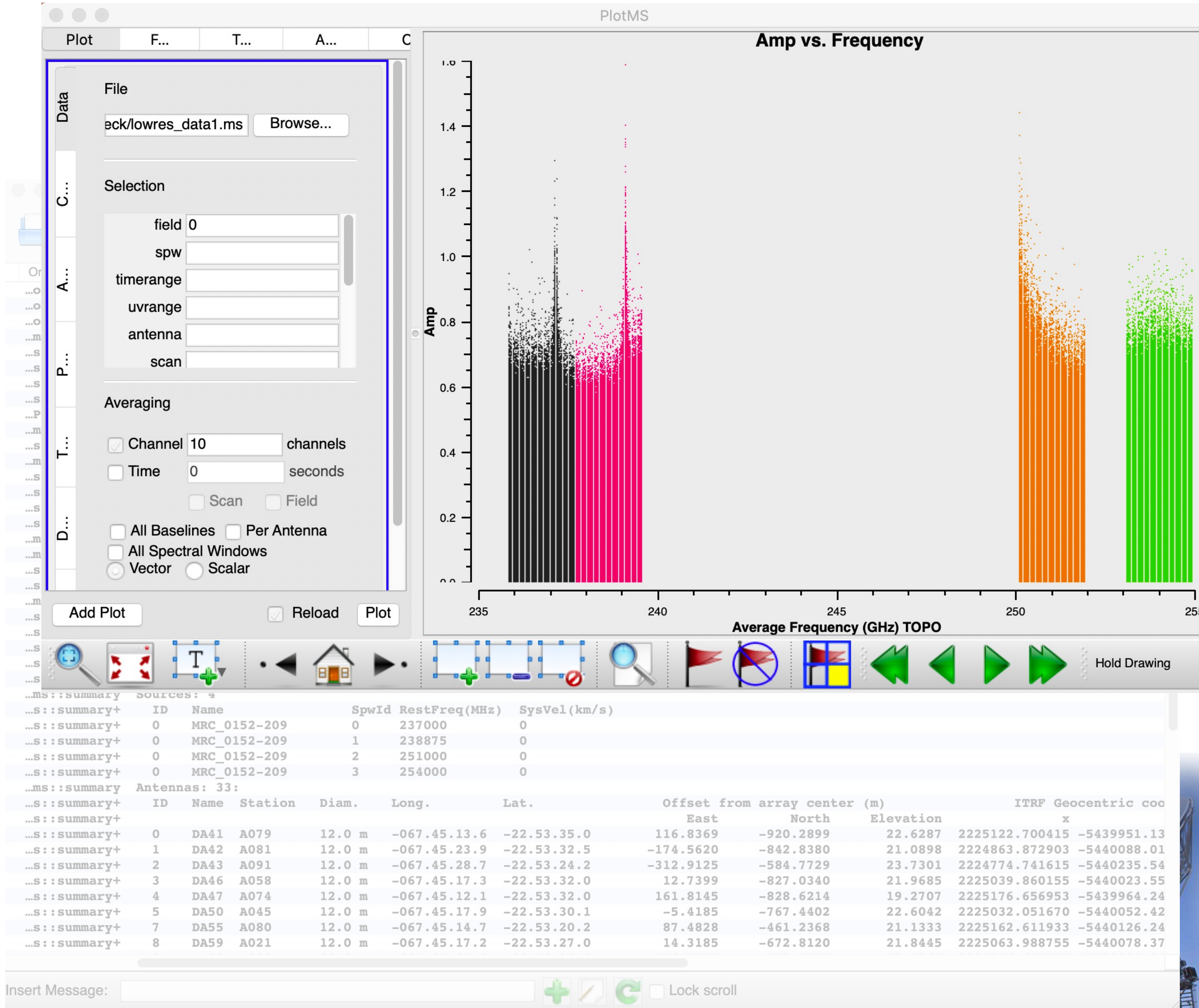
Calibration

Imaging

Analysis

Single Dish

Simulations

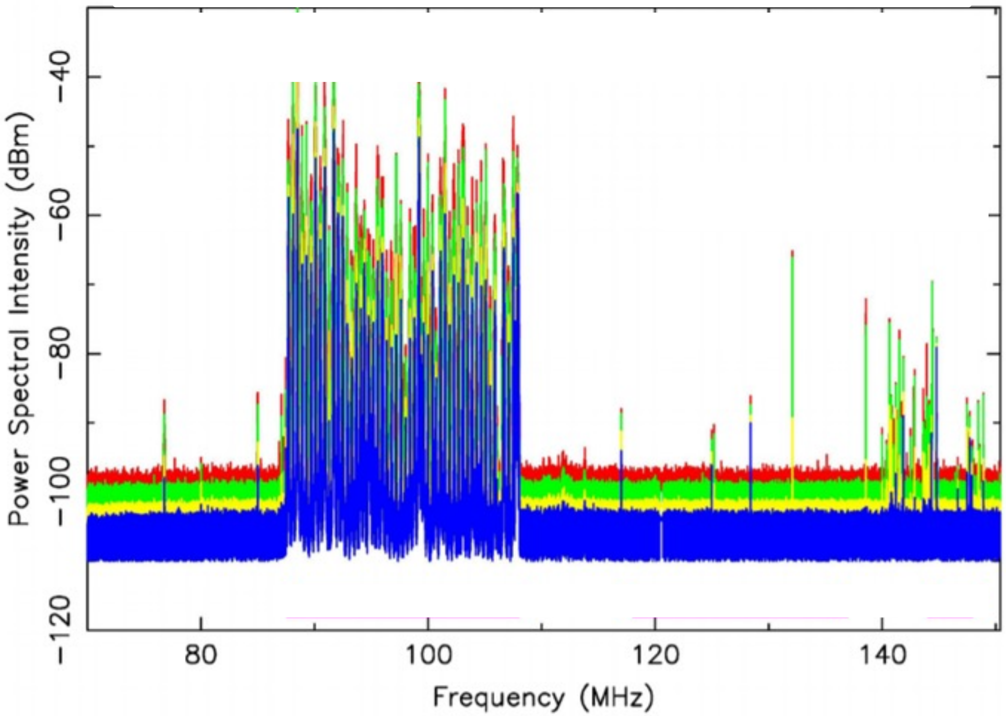


CASA

- Import/export
- Information
- **Manipulation**
- Calibration
- Imaging
- Analysis

- Single Dish
- Simulations

Radio Frequency Interference (RFI) → Flag!



CASA

Import/export

Information

Manipulation

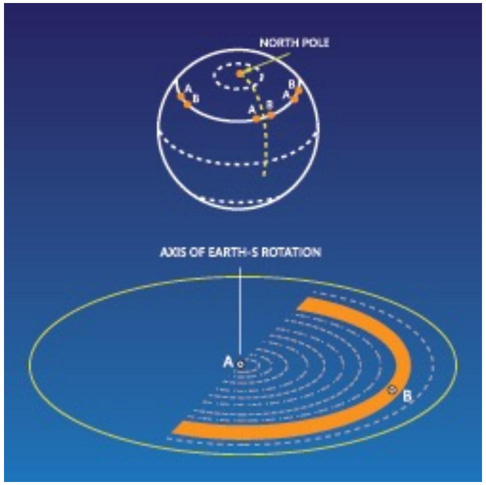
Calibration

Imaging

Analysis

Single Dish

Simulations



$$\vec{V}_{ij} = J_{ij}\,\vec{V}_{ij}^{IDEAL}$$

CASA

Import/export

Information

Manipulation

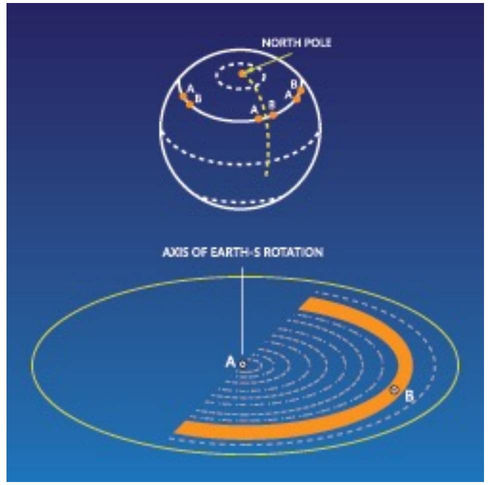
Calibration

Imaging

Analysis

Single Dish

Simulations



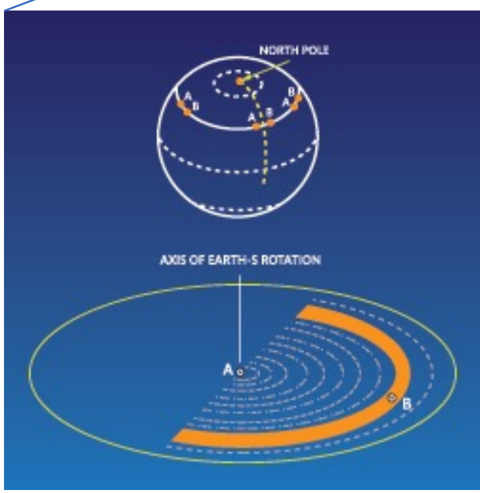
$$\vec{V}_{ij} = J_{ij}\,\vec{V}_{ij}^{IDEAL}$$

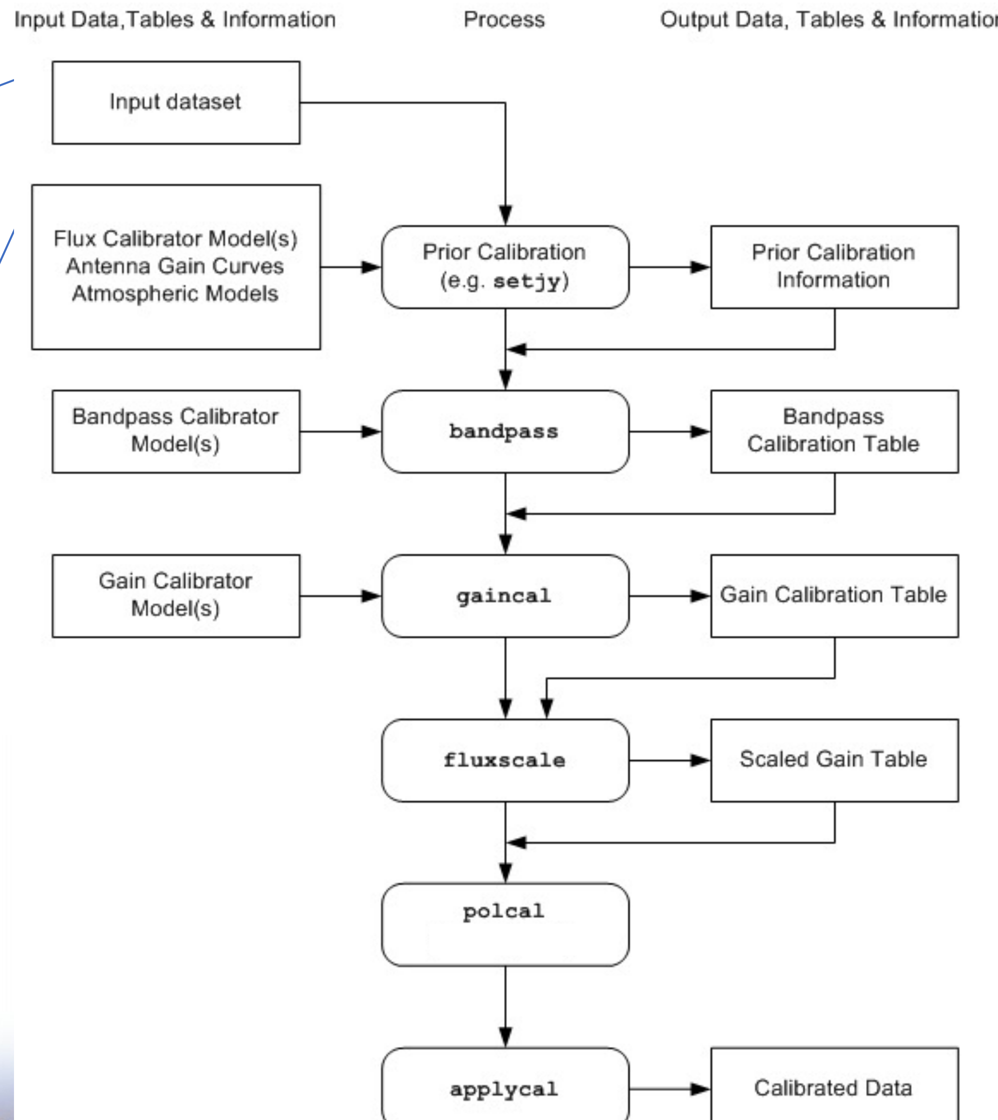
$$\vec{V}_{ij} = M_{ij}\,B_{ij}\,G_{ij}\,D_{ij}\,E_{ij}\,P_{ij}\,T_{ij}\,\vec{V}_{ij}^{IDEAL}$$

- T_{ij}: Troposphere
- P_{ij}: Parallactic rotation
- E_{ij}: Optical components tel.
- D_{ij}: Leakage
- G_{ij}: Time dependent gains
- B_{ij}: Frequency dep. gains (bandpass)
- M_{ij}: Baseline-based correlator (non-closing) errors

Calibration in CASA

$$\vec{V}_{ij} = J_{ij}\,\vec{V}_{ij}^{IDEAL}$$



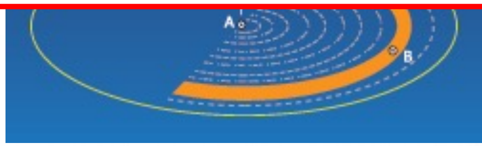


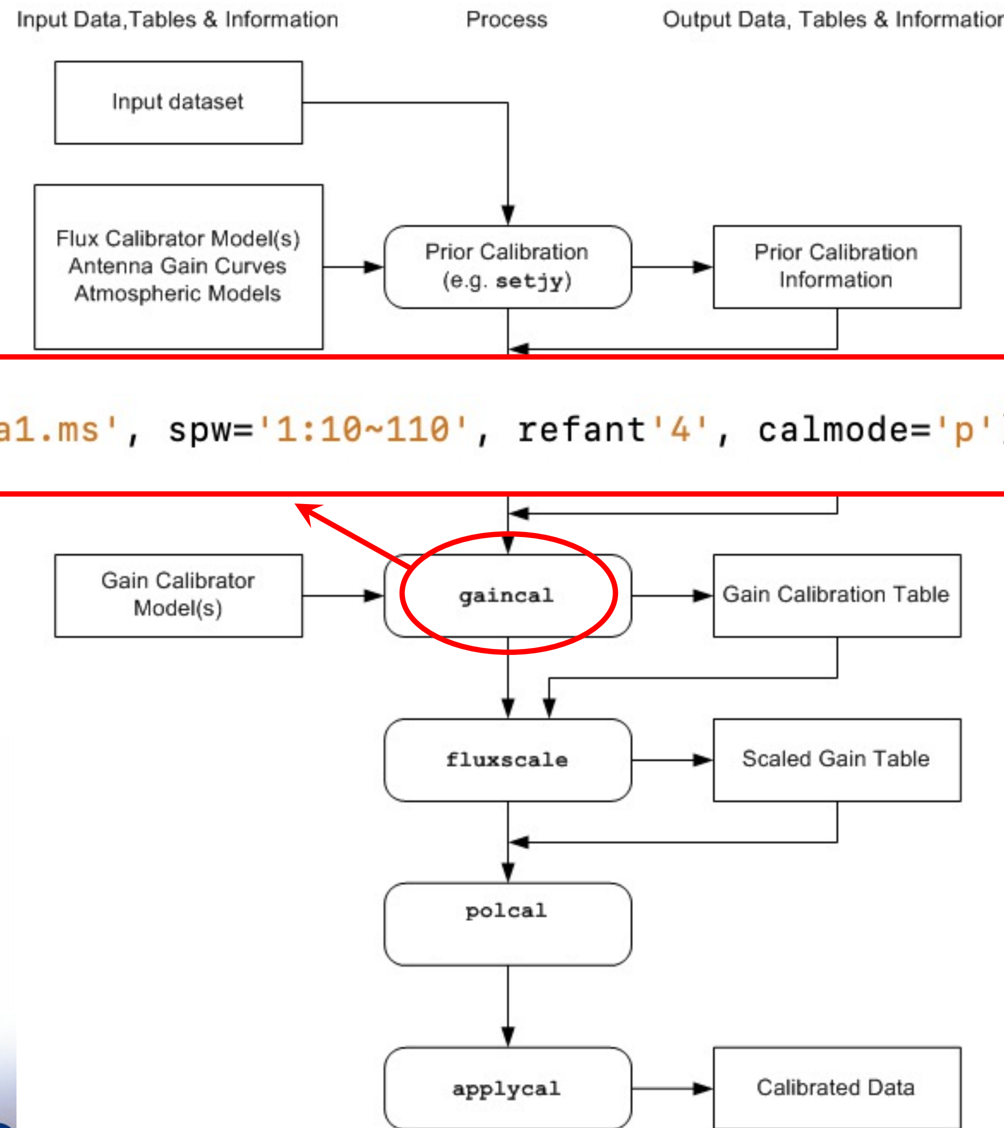
Calibration in CASA

$$\vec{V}_{ij} = J_{ij}\,\vec{V}_{ij}^{\,IDEAL}$$

```
CASA <31>: gaincal(vis='lowres_data1.ms', spw='1:10~110', refant'4', calmode='p')
```

- Command line
- Scripting
- Jupyter Notebook





Calibration in CASA

```
CASA <33>: inp
# gaincal -- Determine temporal gains from calibrator observations
vis                 = 'lowres_data1.ms/'     # Name of input visibility file
caltable            = ''                     # Name of output gain calibration table
field               = ''                     # Select field using field id(s) or field name(s)
spw                 = '1:10~110'             # Select spectral window/channels
intent              = ''                     # Select observing intent
selectdata          = True                   # Other data selection parameters
    timerange       = ''                     # Select data based on time range
    uvrange         = ''                     # Select data by baseline length.
    antenna         = ''                     # Select data based on antenna/baseline
    scan            = ''                     # Scan number range
    observation     = ''                     # Select by observation ID(s)
    msselect        = ''                     # Optional complex data selection (ignore for now)
solint              = 'inf'                  # Solution interval
combine             = ''                     # Data axes which to combine for solve (obs, scan, spw, and/or
                                             # field)
preavg              = -1.0                   # Pre-averaging interval (sec) (rarely needed)
refant              = '4'                    # Reference antenna name(s)
refantmode          = 'flex'                 # Reference antenna mode
minblperant         = 4                      # Minimum baselines _per antenna_ required for solve
minsnr              = 3.0                    # Reject solutions below this SNR
solnorm             = False                  # Normalize (squared) solution amplitudes (G, T only)
gaintype            = 'G'                    # Type of gain solution (G,T,GSPLINE,K,KCROSS)
smodel              = []                     # Point source Stokes parameters for source model.
calmode             = 'p'                    # Type of solution" ('ap', 'p', 'a')
solmode             = ''                     # Robust solving mode: ('', 'L1', 'R','L1R')
rmsthresh           = []                     # RMS Threshold sequence (for solmode='R' or 'L1R'; see help)
corrdepflags        = False                  # Respect correlation-dependent flags
append              = False                  # Append solutions to the (existing) table
docallib            = False                  # Use callib or traditional cal apply parameters
    gaintable       = []                     # Gain calibration table(s) to apply on the fly
    gainfield       = []                     # Select a subset of calibrators from gaintable(s)
    interp          = []                     # Interpolation parameters for each gaintable, as a list
    spwmap          = []                     # Spectral window mappings to form for gaintable(s)
parang              = False                  # Apply parallactic angle correction

CASA <34>: go
```

CASA

Import/export

Information

Manipulation

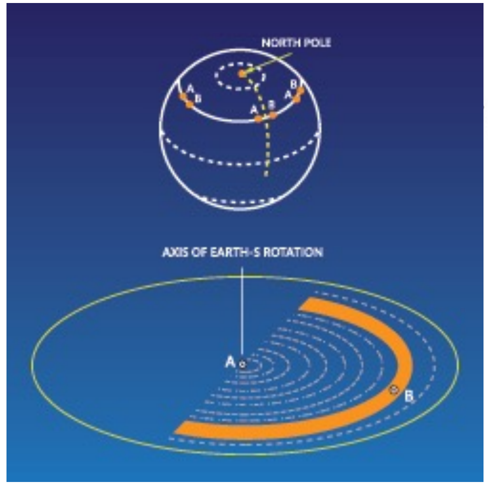
Calibration

Imaging

Analysis

Single Dish

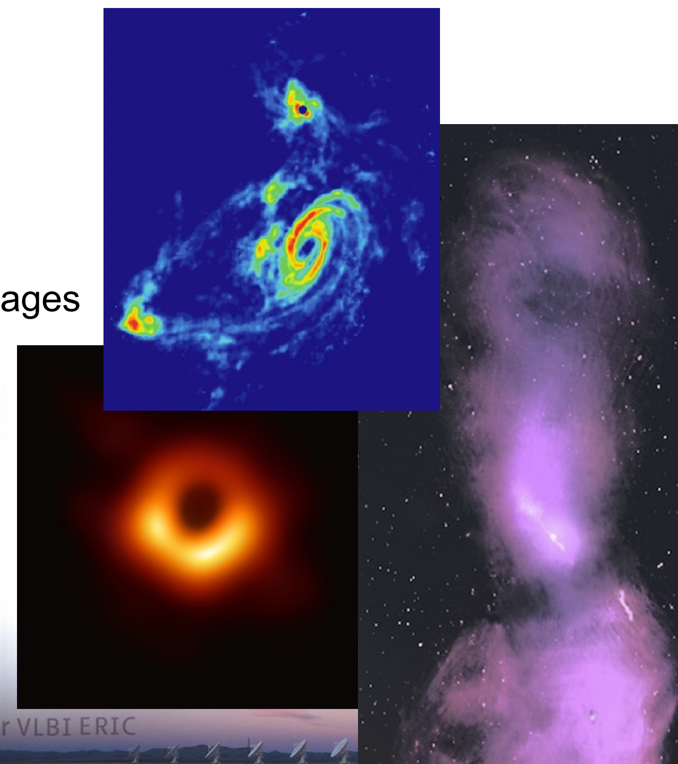
Simulations



$$\vec{V}_{ij}^{IDEAL}$$

→ Gridding data
→ Weighting data
→ Fourier transform
→ Deconvolution
→ Restoration

2D continuum images
3D image cubes



JIVE Joint Institute for VLBI ERIC

Imaging in CASA

Tclean: powerful imaging task

- Fast Fourier Transform (FFT)
- Gridding (incl. advanced imaging modes: mosaic, AW-project, etc)
- Deconvolution & Restoration
- Primary beam correction
- Data weighting

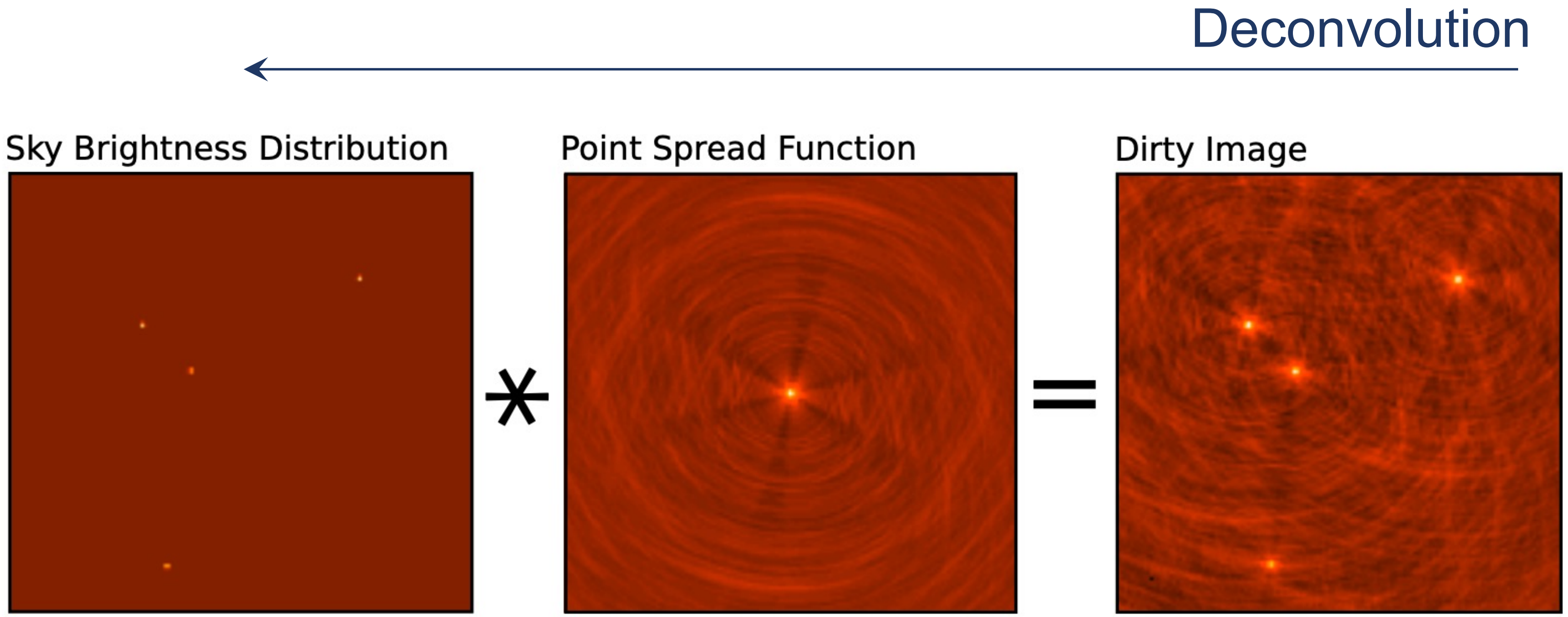
Imaging in CASA

Tclean: powerful imaging task

- Fast Fourier Transform (FFT)
- Gridding (incl. advanced imaging modes: mosaic, AW-project, etc)
- Deconvolution & Restoration
- Primary beam correction
- Data weighting

```
CASA <40>: inp
# tclean -- Radio Interferometric Image Reconstruction
vis                 = ''                   # Name of input visibility file(s)
selectdata          = True                 # Enable data selection parameters
     field          = ''                   # field(s) to select
     spw            = ''                   # spw(s)/channels to select
     timerange      = ''                   # Range of time to select from data
     uvrange        = ''                   # Select data within uvrange
     antenna        = ''                   # Select data based on antenna/baseline
     scan           = ''                   # Scan number range
     observation    = ''                   # Observation ID range
     intent         = ''                   # Scan Intent(s)
datacolumn          = 'corrected'          # Data column to image(data,corrected)
imagename           = ''                   # Pre-name of output images
imsize              = [100]                # Number of pixels
cell                = []                   # Cell size
phasecenter         = ''                   # Phase center of the image
stokes              = 'I'                  # Stokes Planes to make
projection          = 'SIN'                # Coordinate projection
startmodel          = ''                   # Name of starting model image
specmode            = 'mfs'                # Spectral definition mode (mfs,cube,cub
     reffreq        = ''                   # Reference frequency
gridder             = 'standard'           # Gridding options (standard, wproject,
     vptable        = ''                   # Name of Voltage Pattern table
     pblimit        = 0.2                  # PB gain level at which to cut off norm
deconvolver         = 'hogbom'             # Minor cycle algorithm (hogbom,clark,mu
restoration         = True                 # Do restoration steps (or not)
     restoringbeam  = []                   # Restoring beam shape to use. Default i
     pbcor          = False                # Apply PB correction on the output rest
outlierfile         = ''                   # Name of outlier-field image definition
weighting           = 'natural'            # Weighting scheme (natural,uniform,brig
                                           # briggsbwtaper[experimental])

     uvtaper        = []                   # uv-taper on outer baselines in uv-plan
niter               = 0                    # Maximum number of iterations
usemask             = 'user'               # Type of mask(s) for deconvolution: use
     mask           = ''                   # Mask (a list of image name(s) or regio
     pbmask         = 0.0                  # primary beam mask
fastnoise           = True                 # True: use the faster (old) noise calcu
                                           # calculations

restart             = True                 # True : Re-use existing images. False :
savemodel           = 'none'               # Options to save model visibilities (no
calcres             = True                 # Calculate initial residual image
calcpsf             = True                 # Calculate PSF
     psfcutoff      = 0.35                 # All pixels in the main lobe of the PSF
                                           # beam (the Clean beam).

parallel            = False                # Run major cycles in parallel

CASA <41>:
```

Imaging in CASA



Convolution of sky brightness distribution and instrumental PSF results in ‘dirty’ image

↓

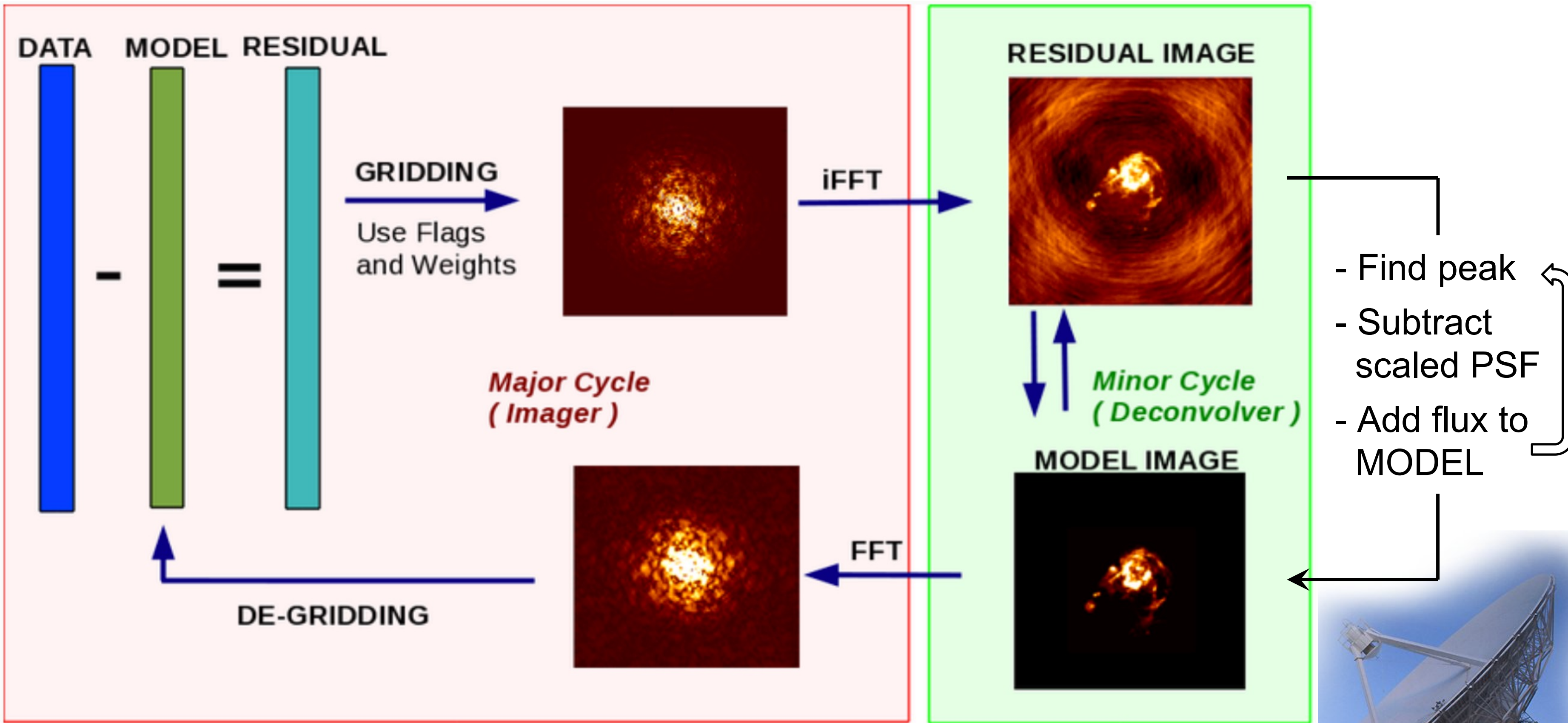
Deconvolution in tclean

↓

Reconstruct sky model by iteratively deconvolving the dirty image using the PSF

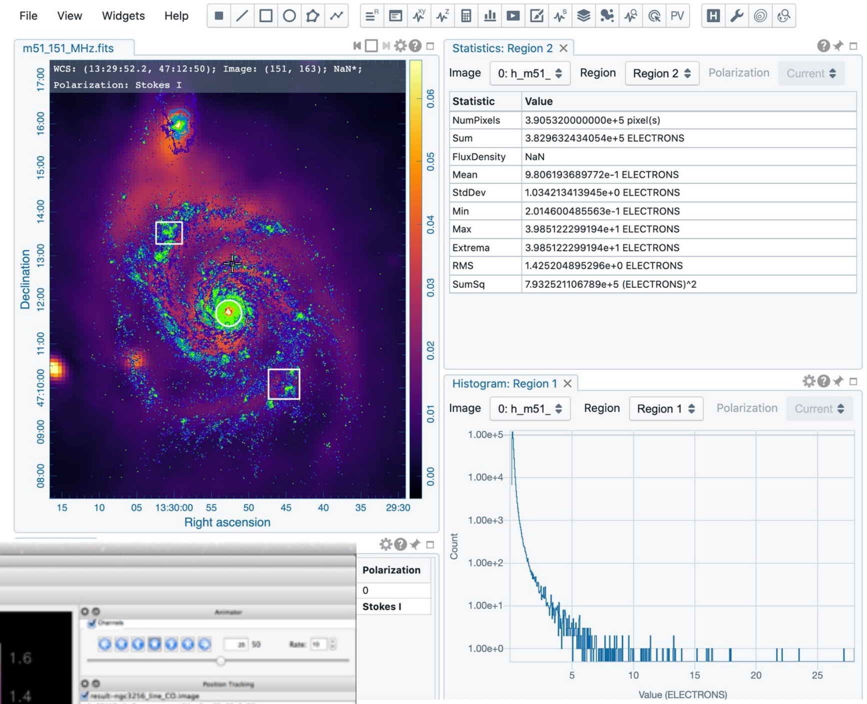
Imaging in CASA

Tclean: powerful imaging task



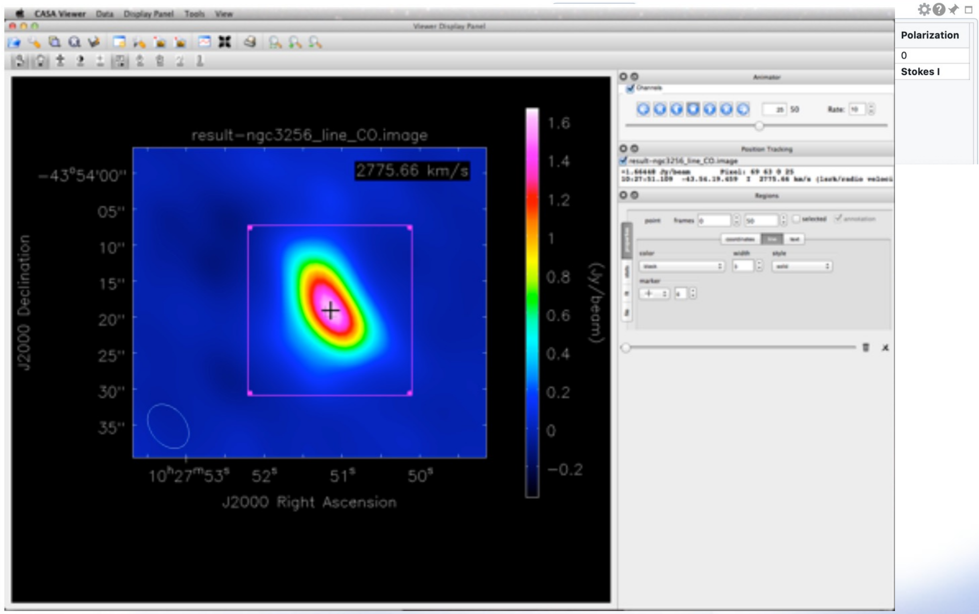
CASA

The new: CARTA *(ASIAA, IDIA, NRAO, Alberta)*



- Import/export
- Information
- Manipulation
- Calibration
- Imaging
- **Analysis/**
- **Visualization**

The old: CASA Viewer



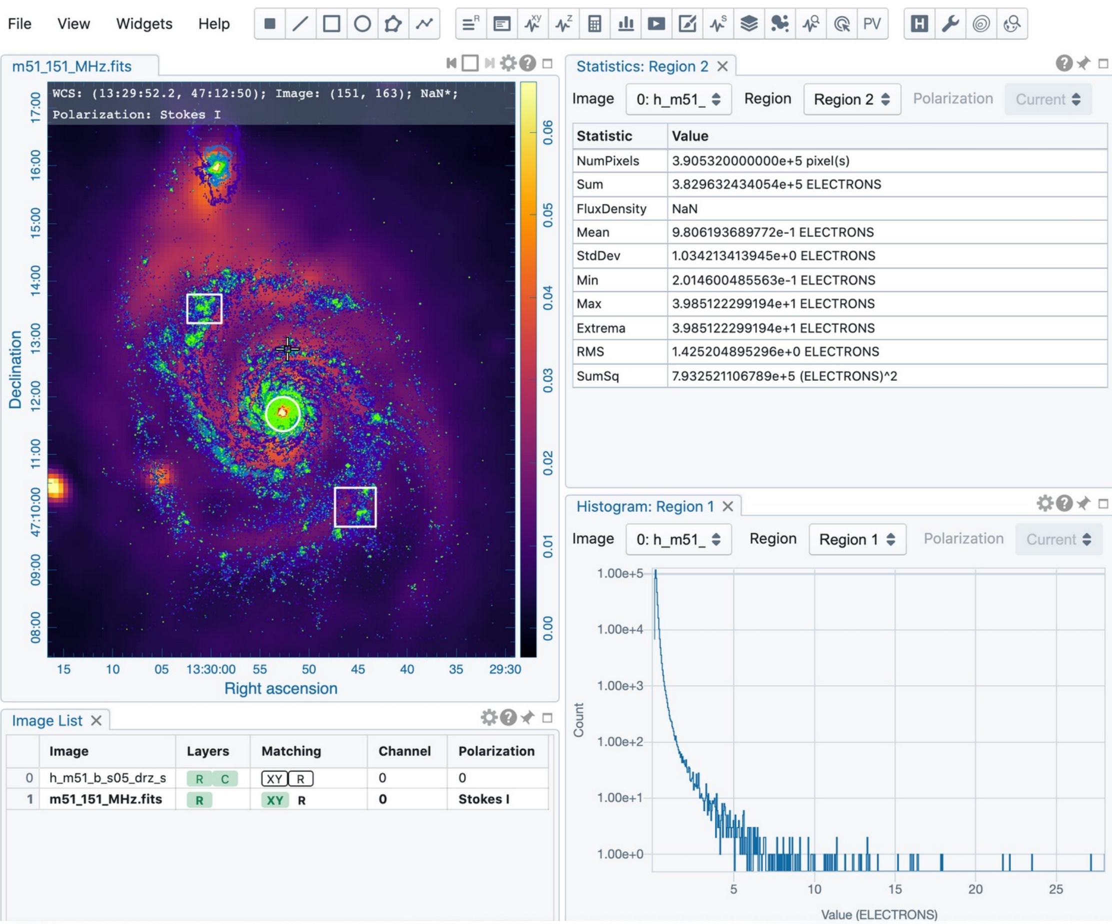
- Single Dish
- Simulations

NRAO ESO NAOJ ASIAA ALMA CSIRO ASTRON JIVE Joint Institute for VLBI ERIC

Visualization

Students, start using this!

CARTA: Cube Analysis and Rendering Tool for Astronomy



Next-generation radio telescopes
→ quick with large data volumes
(ALMA, VLA, SKA pathfinders)

https:cartavis.org

External software:

ASIAA (Taiwan)
IDIA (South Africa)
NRAO (USA)
Univ. Alberta (Canda)

NRAO ESO NAOJ ASIAA ALMA CSIRO ASTRON JIVE Joint Institute for VLBI ERIC

CASA

Import/export

Information

Manipulation

Calibration

Imaging

Analysis

Single Dish

Simulations

CASA download & installation

Website (casa.nrao.edu)

Monolithic (all-inclusive 'plug-and-play')

Pip-wheel (Pythonic, Jupyter Notebooks, Google Colab)

Pipelines (ALMA, VLA)

Compatibility Operating Systems

Parallel processing

(Message Passing Interface, MPI)

New release every ~2 months!

Latest version: CASA 6.4

The Release Notes and Known Issues of the 6.4 release are available in ...

CASA 6.4 is based on Python 3, and available either as a downloadable tar-file distribution with Python environment included, or as a modular version that can be installed with pip-wheels.

Manual processing can be done with any CASA version, but ALMA and VLA pipelines may differ and are not always included, so download the correct CASA version for pipeline use.

	Linux (RedHat 6, 7, 8)	Mac (OS 11, OSX 10.14, 10.15)
General Use (Notes)	CASA 6.4.0 (RH7/8 - Py 3.8) CASA 6.4.0 (RH7/8 - Py 3.6)	CASA 6.4.0 (OS11 - Py 3.8) CASA 6.4.0 (OS11 - Py 3.6) CASA 6.4.0 (10.15 - Py 3.8) CASA 6.4.0 (10.15 - Py 3.6)
ALMA Pipeline (Notes)	CASA 6.2.1 (RH6/7)	CASA 6.2.1 (10.15) CASA 6.2.1 (10.14)
VLA Pipeline (Notes)	CASA 6.2.1 (RH6/7)	CASA 6.2.1 (10.15) CASA 6.2.1 (10.14)

The above CASA versions can also be downloaded from our NAOJ CASA mirror site and NAOJ CASA-pipeline mirror site, or via Google Drive.

CASA 6: pip-wheel installation

CASA 6 can optionally be installed through modular pip-wheels, with the flexibility to build CASA tools and tasks into a customized Python environment. Instructions on how to install the pip-wheel version of CASA 6 can be found in CASA Docs: CASA 6 Installation and Usage

The modular pip-wheel version is not yet used in production by ALMA and VLA, and does not include any pipelines.

CASA download & installation

Website (casa.nrao.edu)

Monolithic (all-inclusive 'plug-and-play')

Pip-wheel (Pythonic, Jupyter Notebooks, Google Colab)

Pipelines (ALMA, VLA)

Compatibility Operating Systems

Parallel processing

(Message Passing Interface, MPI)

Latest version: CASA 6.4

The Release Notes and Known Issues of the 6.4 release are available in CASA Docs

CASA 6.4 is based on Python 3, and available either as a downloadable tar-file distribution with Python environment included, or as a modular version that can be installed with pip-wheels.

Manual processing can be done with any CASA version, but ALMA and VLA pipelines may differ and are not always included, so download the correct CASA version for pipeline use.

	Linux (RedHat 6, 7, 8)	Mac (OS 11, OSX 10.14, 10.15)
General Use (Notes)	CASA 6.4.0 (RH7/8 - Py 3.8) CASA 6.4.0 (RH7/8 - Py 3.6)	CASA 6.4.0 (OS11 - Py 3.8) CASA 6.4.0 (OS11 - Py 3.6) CASA 6.4.0 (10.15 - Py 3.8) CASA 6.4.0 (10.15 - Py 3.6)
ALMA Pipeline (Notes)	CASA 6.2.1 (RH6/7)	CASA 6.2.1 (10.15) CASA 6.2.1 (10.14)
VLA Pipeline (Notes)	CASA 6.2.1 (RH6/7)	CASA 6.2.1 (10.15) CASA 6.2.1 (10.14)

The above CASA versions can also be downloaded from our NAOJ CASA mirror site and NAOJ CASA-pipeline mirror site, or via Google Drive.

CASA 6: pip-wheel installation

CASA 6 can optionally be installed through modular pip-wheels, with the flexibility to build CASA tools and tasks into a customized Python environment. Instructions on how to install the pip-wheel version of CASA 6 can be found in CASA Docs: CASA 6 Installation and Usage

The modular pip-wheel version is not yet used in production by ALMA and VLA, and does not include any pipelines.

CASA download & installation

Website (casa.nrao.edu)

Monolithic (all-inclusive 'plug-and-play')

Pip-wheel (Pythonic, Jupyter Notebooks, Google Colab)

Pipelines (ALMA, VLA)

Compatibility Operating Systems

Parallel processing

(Message Passing Interface, MPI)

Latest version: CASA 6.4

The Release Notes and Known Issues of the 6.4 release are available in CASA Docs

CASA 6.4 is based on Python 3, and available either as a downloadable tar-file distribution with Python environment included, or as a modular version that can be installed with pip-wheels.

Manual processing can be done with any CASA version, but ALMA and VLA pipelines may differ and are not always included, so download the correct CASA version for pipeline use.

	Linux (RedHat 6, 7, 8)	Mac (OS 11, OSX 10.14, 10.15)
General Use (Notes)	CASA 6.4.0 (RH7/8 - Py 3.8) CASA 6.4.0 (RH7/8 - Py 3.6)	CASA 6.4.0 (OS11 - Py 3.8) CASA 6.4.0 (OS11 - Py 3.6) CASA 6.4.0 (10.15 - Py 3.8) CASA 6.4.0 (10.15 - Py 3.6)
ALMA Pipeline (Notes)	CASA 6.2.1 (RH6/7)	CASA 6.2.1 (10.15) CASA 6.2.1 (10.14)
VLA Pipeline (Notes)	CASA 6.2.1 (RH6/7)	CASA 6.2.1 (10.15) CASA 6.2.1 (10.14)

The above CASA versions can also be downloaded from our NAOJ CASA mirror site and NAOJ CASA-pipeline mirror site, or via Google Drive.

CASA 6: pip-wheel installation

CASA 6 can optionally be installed through modular pip-wheels, with the flexibility to build CASA tools and tasks into a customized Python environment. Instructions on how to install the pip-wheel version of CASA 6 can be found in CASA Docs: CASA 6 Installation and Usage

The modular pip-wheel version is not yet used in production by ALMA and VLA, and does not include any pipelines.

CASA download & installation

Website (casa.nrao.edu)

Monolithic (all-inclusive 'plug-and-play')

Pip-wheel (Pythonic, Jupyter Notebooks, Google Colab)

Pipelines (ALMA, VLA)

Compatibility Operating Systems

Parallel processing

(Message Passing Interface, MPI)

Full Monolithic Distribution

	Python 2.7	Python 3.6	Python 3.7	Python 3.8
RHEL 6	5.8	6.1, 6.2		
RHEL 7	5.8	>=6.1		>=6.4
RHEL 8				>=6.4
Ubuntu 18.04		>=6.2		>=6.4
Ubuntu 20.04		>=6.2		>=6.4
Mac OS 10.14	5.8	>=6.1		>=6.3
Mac OS 10.15	5.8	>=6.1		>=6.3
Mac OS 11 x86		>=6.3		>=6.3
Mac OS 11 ARM		TBD		TBD

Modular CASA

	Python 2.7	Python 3.6	Python 3.7	Python 3.8
RHEL 6		6.0, 6.1, 6.2	6.2	6.2
RHEL 7		>=6.0	>=6.2	>=6.2
RHEL 8		>=6.0	>=6.4	>=6.4
Ubuntu 18.04		>=6.0	>=6.2	>=6.2
Ubuntu 20.04		>=6.0	>=6.2	>=6.2
Mac OS 10.14		>=6.1		>=6.3
Mac OS 10.15		>=6.1		>=6.3
Mac OS 11 x86		>=6.3		>=6.3
Mac OS 11 ARM		TBD		TBD

CASA download & installation

Website (casa.nrao.edu)

Monolithic (all-inclusive 'plug-and-play')

Pip-wheel (Pythonic, Jupyter Notebooks, Google Colab)

Pipelines (ALMA, VLA)

Compatibility Operating Systems

Parallel processing

(Message Passing Interface, MPI)

Full Monolithic Distribution

	Python 2.7	Python 3.6	Python 3.7	Python 3.8
RHEL 6	5.8	6.1, 6.2		
RHEL 7	5.8	>=6.1		>=6.4
RHEL 8				>=6.4
Ubuntu 18.04		>=6.2		>=6.4
Ubuntu 20.04		>=6.2		>=6.4
Mac OS 10.14	5.8	>=6.1		>=6.3
Mac OS 10.15	5.8	>=6.1		>=6.3
Mac OS 11 x86		>=6.3		>=6.3
Mac OS 11 ARM		TBD		TBD

Modular CASA

	Python 2.7	Python 3.6	Python 3.7	Python 3.8
RHEL 6		6.0, 6.1, 6.2	6.2	6.2
RHEL 7		>=6.0	>=6.2	>=6.2
RHEL 8		>=6.0	>=6.4	>=6.4
Ubuntu 18.04		>=6.0	>=6.2	>=6.2
Ubuntu 20.04		>=6.0	>=6.2	>=6.2
Mac OS 10.14		>=6.1		>=6.3
Mac OS 10.15		>=6.1		>=6.3
Mac OS 11 x86		>=6.3		>=6.3
Mac OS 11 ARM		TBD		TBD

CASA Docs (casadocs.readthedocs.io)

CASA

Search docs

Read the Docs v: stable

» Common Astronomy Software Applications

Edit on GitHub

Common Astronomy Software Applications

CASA, the *Common Astronomy Software Applications*, is the primary data processing software for the Atacama Large Millimeter/submillimeter Array (ALMA) and Karl G. Jansky Very Large Array (VLA), and is often used also for other radio telescopes.

6.4.0 Release

CASA 6.4.0 can now be (downloaded) for general use. CASA 6.4.0 is available either as a downloadable tar-file, or through pip-wheel installation, which gives flexibility to integrate CASA into a customized Python environment.

Highlights:

- OS Support: CASA now supports RedHat 8, and Mac OS with Python 3.8, for both monolithic and modular versions. Note the Linux tarballs with different Python versions will extract to the same directory name.
- plotcal/plotms: Funtionality for plotcal has been migrated to plotms, and plotcal was deprecated.
- plotms: calibration table averaging with channel selection is now supported.
- fringefit: memory usage of fringefit has been reduced, allowing larger datasets to be processed.
- imhead: updated to display microsecond precision.
- caltables: the storage of frequency meta information in caltables improved, making certain frequency-dependent calibration solutions more accurate.
- sdintimaging: now adds information to the history of produced images
- T+dT timerange selection improved in accuracy.

CASA Docs (casadocs.readthedocs.io)

CASA

Search docs

Release Information

Index

Read the Docs v: stable

Versions

latest stable v6.4.0 v6.3.0

v6.2.1 v6.2.0

Downloads

HTML

On Read the Docs

Project Home Builds Downloads

On GitHub

View Edit

Search

Search docs

Hosted by Read the Docs · Privacy Policy

» Common Astronomy Software Applications

Edit on GitHub

Common Astronomy Software Applications

CASA, the *Common Astronomy Software Applications*, is the primary data processing software for the Atacama Large Millimeter/submillimeter Array (ALMA) and Karl G. Jansky Very Large Array (VLA), and is often used also for other radio telescopes.

6.4.0 Release

CASA 6.4.0 can now be (downloaded) for general use. CASA 6.4.0 is available either as a downloadable tar-file, or through pip-wheel installation, which gives flexibility to integrate CASA into a customized Python environment.

Highlights:

- OS Support: CASA now supports RedHat 8, and Mac OS with Python 3.8, for both monolithic and modular versions. Note the Linux tarballs with different Python versions will extract to the same directory name.
- plotcal/plotms: Funtionality for plotcal has been migrated to plotms, and plotcal was deprecated.
- plotms: calibration table averaging with channel selection is now supported.
- fringefit: memory usage of fringefit has been reduced, allowing larger datasets to be processed.
- imhead: updated to display microsecond precision.
- caltables: the storage of frequency meta information in caltables improved, making certain frequency-dependent calibration solutions more accurate.
- sdintimaging: now adds information to the history of produced images
- T+dT timerange selection improved in accuracy.

CASA Docs (casadocs.readthedocs.io)

Known Issues

Summary Most Important Issues

- The task **clean** is no longer being actively maintained; instead, **tclean** is now the recommended task for imaging.
- **CASA 6 startup** may fail on some **Mac OS** where users have set up a file system that is *case-sensitive*.
- There are generic problems putting multiple MSs into **tclean** that have mismatches in their shape.
- Wideband and widefield imaging in **tclean** are only partially validated - please use at own risk and read wideband and widefield documentation.
- In **tclean**, *uvtaper* does not work with *natural* weighting.
- When imaging large mosaics with *mosweight* in **tclean**, an error *"too many open files"* may occur that may require to increase the limit for open files.
- **stawt** may fail when the correlator integration time changes within an MS and statwt is run with timebin set to an integer value.
- CASA is not using *LD_LIBRARY_PATH* anymore but *CASALD_LIBRARY_PATH* to avoid confusion.
- **cvel** is calculating the velocity incorrectly for ephemeris objects. We recommend to use **mstransform** or its offspring **cvel2**, although the latter should be used with care as it is not fully commissioned yet.
- **fixvis** uses the small angle approximation and may be incorrect for large phase shifts. Use the new task **phaseshift** instead, or use **tclean** for phase center shifts during imaging when applicable.
- With parallel calibration on MMS files, **fixvis** does not write out the the new MMS specified in *outputvis* correctly, hence fixvis solutions are not applied when writing to a new MMS.
- In **fringefit**, calibration tables created with CASA 5.5 and before cannot be used with CASA 5.6 and later.
- In **tclean**, defining image cubes in optical velocity in some cases is known not to work.
- In **tclean**, using the mosaic gridder with the default *nchan=-1* is in some cases known to produce errors.
- Ionospheric TEC corrections are currently validated in CASA only for VLA data.
- *ephemeris* objects are not correctly supported by *virtual model columns*.
- In **tclean**, the combination of *specmode='cube'* and *gridder="awproject"* has not been commissioned for use and may result in errors.

CASA Docs (casadocs.readthedocs.io)

Monolithic Distribution

On Linux:

1. Download the .tar file and place it in a work directory (e.g. ~/casa)
2. From a Linux terminal window, expand the file:

```
$ tar -xvf casa-xyz.tar.xz
```

3. Start CASA

```
$ ./casa-xyz/bin/casa
```

4. The one caveat is that CASA on Linux currently will not run if the Security-Enhanced Linux option of the linux operating system is set to enforcing. For the non-root install to work, SElinux must be set to disabled or permissive (in `/etc/selinux/config`) or you must run (as root):

```
setsebool -P allow_execheap=1
```

Otherwise, you will encounter errors like:

```
error while loading shared libraries: /opt/casa/casa-20.0.5653-001/lib/liblapack.so.3.1.1: cannot r
```

On Macintosh:

1. Download the .dmg disk image file
2. Double click on the disk image file (if your browser does not automatically open it).
3. Drag the CASA application to the *Applications* folder of your hard disk.
4. Eject the CASA disk image.
5. Double click the CASA application to run it for the first time. If the OS does not allow you to install apps from non-Apple sources, please Change the settings in “System Preferences-> Security & Privacy -> General” and “Allow applications downloaded from: Mac App store and identified developers”.
6. Optional: Create symbolic links to the CASA version and its executables (Administrator privileges are

CASA Doc

Full Monolithic Distribution

	Python 2.7	Python 3.6	Python 3.7	Python 3.8
RHEL 6	5.8	6.1, 6.2		
RHEL 7	5.8	>=6.1		>=6.4
RHEL 8				>=6.4
Ubuntu 18.04		>=6.2		>=6.4
Ubuntu 20.04		>=6.2		>=6.4
Mac OS 10.14	5.8	>=6.1		>=6.3
Mac OS 10.15	5.8	>=6.1		>=6.3
Mac OS 11 x86		>=6.3		>=6.3
Mac OS 11 ARM		TBD		TBD

Modular CASA

	Python 2.7	Python 3.6	Python 3.7	Python 3.8
RHEL 6		6.0, 6.1, 6.2	6.2	6.2
RHEL 7		>=6.0	>=6.2	>=6.2
RHEL 8		>=6.0	>=6.4	>=6.4
Ubuntu 18.04		>=6.0	>=6.2	>=6.2
Ubuntu 20.04		>=6.0	>=6.2	>=6.2
Mac OS 10.14		>=6.1		>=6.3
Mac OS 10.15		>=6.1		>=6.3
Mac OS 11 x86		>=6.3		>=6.3
Mac OS 11 ARM		TBD		TBD

of the

nnot r

l apps

General" and "Allow applications downloaded from: Mac App store and identified developers".

6. Optional: Create symbolic links to the CASA version and its executables (Administrator privileges are

CASA Docs (casadocs.readthedocs.io)

API

External Interface definition of CASA. This section is verified prior to each release

- casatasks
- casatools
- casashell
- casadata
- casalith
- configuration

CASA Docs

```
CASA <1>: doc('phaseshift')
```

API

⊟ Task List

- Input / Output
- Information
- Flagging
- Calibration
- Imaging
- Single Dish
- ⊟ Manipulation
  - clearstat
  - concat
  - conjugatevis
  - cvel
  - cvel2
  - fixplanets
  - fixvis
  - hanningsmooth
  - mstransform
  - msuvbin
  - oldstatwt
  - partition
  - phaseshift
  - rmtables
  - split

Read the Docs v: stable

» API » casatasks » phaseshift Edit on GitHub

phaseshift

phaseshift(*vis, outputvis='', keepmms=True, field='', spw='', scan='', intent='', array='', observation='', datacolumn='all', phasecenter=''*) [source]

Rotate a Measurement Set to a new phase-center

[Description] [Examples] [Development] [Details]

Parameters

- vis (string) - Name of input visibility file
- outputvis (string='') - Name of output visibility file
- keepmms (bool=True) - Create a Multi-MS as the output if the input is a Multi-MS.
- field (string='') - Select field using field id(s) or field name(s)
- spw (string='') - Select spectral window/channels
- scan (string='') - Scan number range
- intent (string='') - Select observing intent
- array (string='') - Select (sub)array(s) by array ID number.
- observation (string='') - Select by observation ID(s)
- datacolumn (string='all') - Which data column(s) to process
- phasecenter (string='') - Direction coordinates of the desired center. MUST BE SPECIFIED

Description

This task changes the phase center of an MS by modifying the *UVW* coordinates and the specified data column(s) (via the **datacolumn** parameter) of the input MS and creating an output MS with these changes. The *PHASE_DIR* column of the *FIELD* subtable of the new MS is updated with the new phase center. Many MS selection parameters are supported (see Visibility Data Selections for details).

CASA Docs (casadocs.readthedocs.io)

Search docs

Release Information
Index
API
Task List
Using CASA
CASA Fundamentals
External Data
Calibration & Visibilities
Imaging & Analysis
CARTA
Pipeline
Simulations
Parallel Processing
Memo Series & Knowledgebase
⊟ **Community Examples**
CASA 6.2 Notebook Demo
PlotMS with Jupyter Notebooks and X11

Read the Docs v: stable

Community Examples

A collection of community provided scripts covering various CASA tutorials, examples, demonstrations, tips, tricks and general best practices.

- CASA 6.2 Notebook Demo
    - Description
    - Installation
    - Getting Started
    - tclean Example
    - View Images with Astropy
    - View Images with CARTA
- PlotMS with Jupyter Notebooks and X11
    - Description
    - Setup Virtual Frame Buffer
    - Get Data
    - Plot Data
    - Display the Exported Raster Image
- Simulation in CASA
    - Description
    - Installation
    - Make an empty MS with the desired uvw/scan/field/ddid setup
    - Make a True Sky Model (component list and/or image)
    - Simulate visibilities from the sky model into the DATA column of the MS
    - Add Noise and other errors to the simulated visibilities

CASA resources

- CASA Docs → official CASA documentation
  https://casadocs.readthedocs.io
- CASA Website → official CASA website (downloads)
  https://casa.nrao.edu
- CASA Guides → data reduction strategies (telescope-specific)
  https://casaguides.nrao.edu
- NRAO/ALMA Helpdesks → VLA / ALMA data reduction questions.
  https://help.nrao.edu ; https://help.almascience.org

  (coming soon: CASA Bug Report & Feature Request system)
- casa-announce → announcements, releases
- CASA Newsletter → 2x per year
- casa-feedback@nrao.edu → general feedback

First resources!

ALMA / VLA teams

Subscribe!
(casa.nrao.edu)

NRAO ESO 国立天文台 NAOJ ASIAA ALMA CSIRO ASTRON JIVE Joint Institute for VLBI ERIC

CASA resources

- CASA Docs → official CASA documentation
  https://casadocs.readthedocs.io
- CASA Website → official CASA website (downloads)
  https://casa.nrao.edu
- CASA Guides → data reduction strategies (telescope-specific)
  https://casaguides.nrao.edu
- NRAO/ALMA Helpdesks → VLA / ALMA data reduction questions.
  https://help.nrao.edu ; https://help.almascience.org

  (coming soon: CASA Bug Report & Feature Request system)
- casa-announce → announcements, releases
- CASA Newsletter → 2x per year
- casa-feedback@nrao.edu → general feedback

First resources!

ALMA / VLA teams

Subscribe! (casa.nrao.edu)

NRAO ESO NAOJ ASIAA ALMA CSIRO ASTRON JIVE Joint Institute for VLBI ERIC

CASA resources

- CASA Docs → official CASA documentation
  https://casadocs.readthedocs.io
- CASA Website → official CASA website (downloads)
  https://casa.nrao.edu
- CASA Guides → data reduction strategies (telescope-specific)
  https://casaguides.nrao.edu
- NRAO/ALMA Helpdesks → VLA / ALMA data reduction questions.
  https://help.nrao.edu ; https://help.almascience.org

  (coming soon: CASA Bug Report & Feature Request system)
- casa-announce → announcements, releases
- CASA Newsletter → 2x per year
- casa-feedback@nrao.edu → general feedback

First resources!

ALMA / VLA teams

Subscribe!
(casa.nrao.edu)

JIVE
Joint Institute for VLBI ERIC

CASA resources

- CASA Docs → official CASA documentation
  https://casadocs.readthedocs.io
- CASA Website → official CASA website (downloads)
  https://casa.nrao.edu
- CASA Guides → data reduction strategies (telescope-specific)
  https://casaguides.nrao.edu
- NRAO/ALMA Helpdesks → VLA / ALMA data reduction questions.
  https://help.nrao.edu ; https://help.almascience.org

  (coming soon: CASA Bug Report & Feature Request system)
- casa-announce → announcements, releases
- CASA Newsletter → 2x per year
- casa-feedback@nrao.edu → general feedback

First resources!

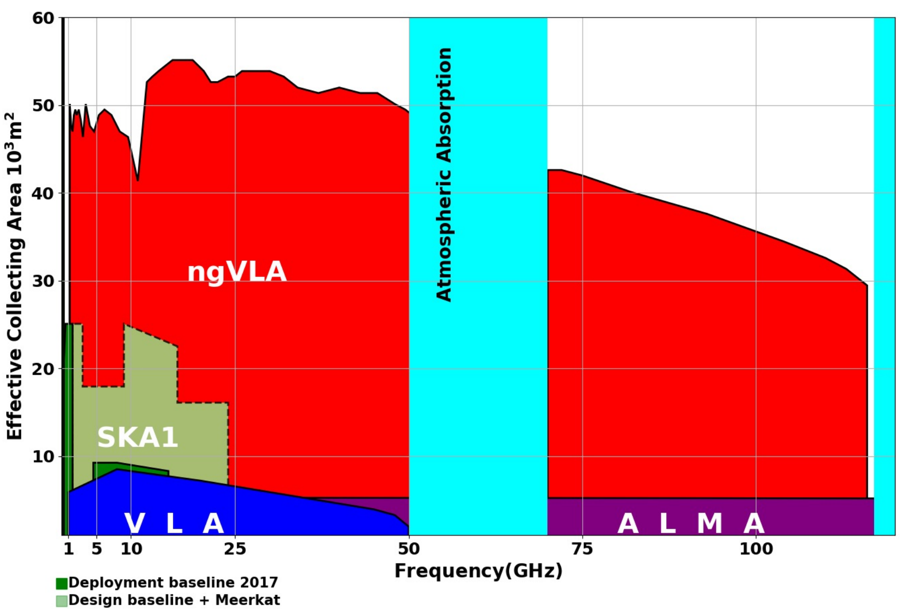
ALMA / VLA teams

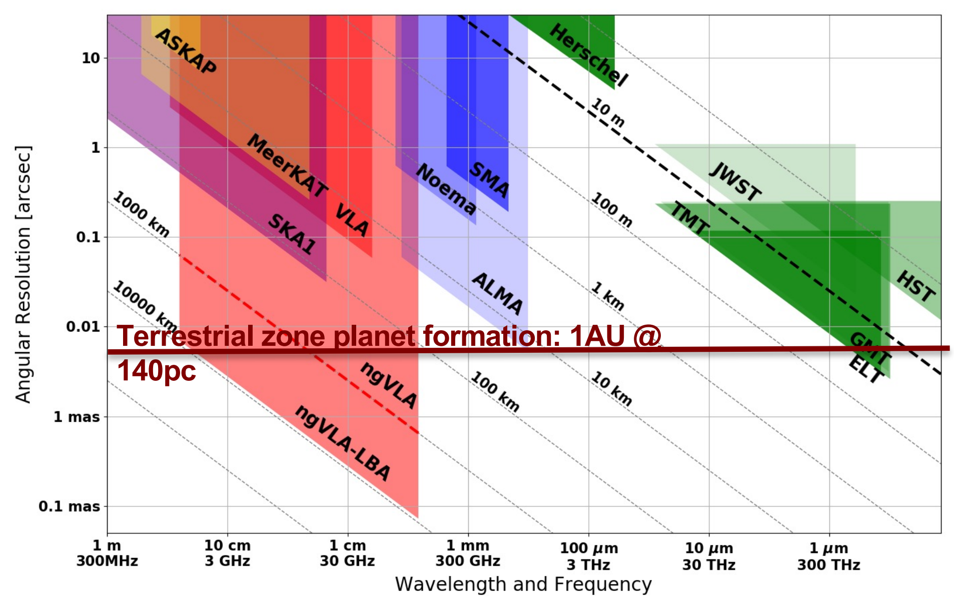
Subscribe!
(casa.nrao.edu)

Future of radio astronomy

Next-generation VLA

- 1.2 – 116 GHz (SKA ← ngVLA → ALMA)
- 244 antennas x 18m diameter (+ 19 x 6m)
- Maximum baseline ~9000 km (0.5-50 milli-arcsec), dense core





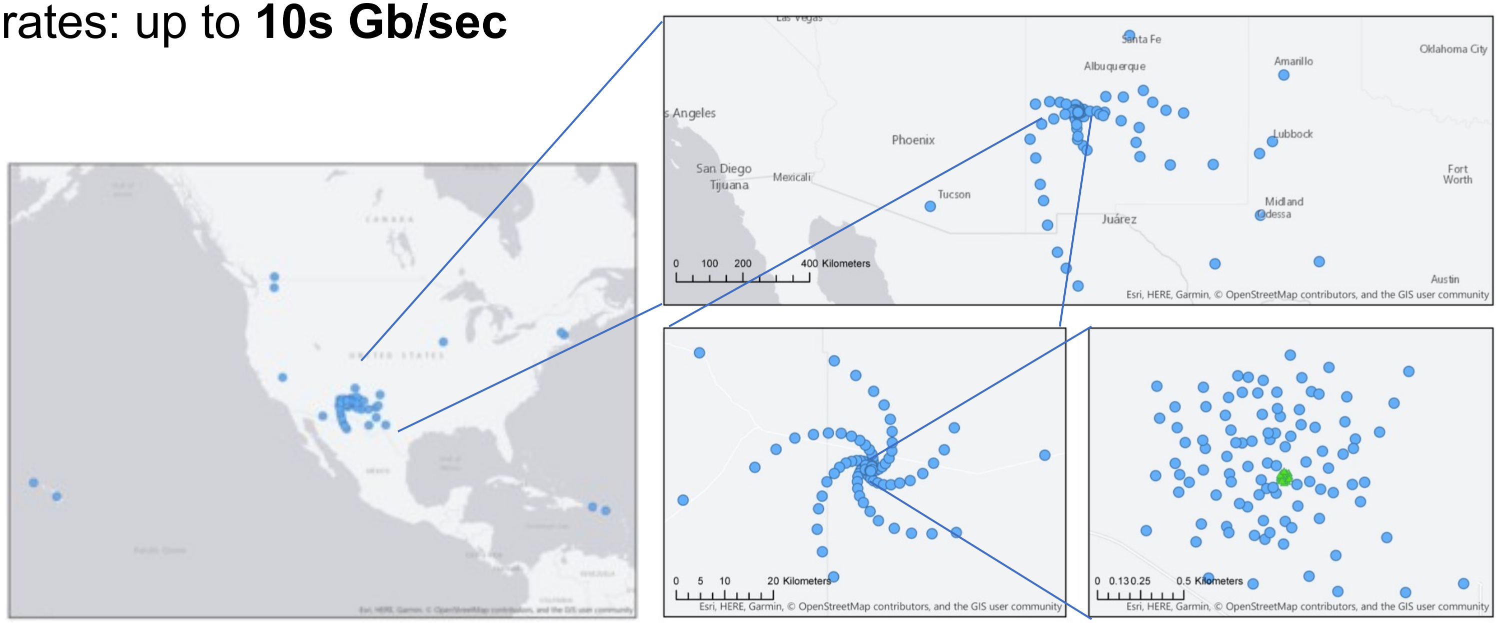
Murphy et al. 2019 – Science with a next-generation VLA

(terrestrial planet formation, star formation, molecular gas across Universe)

Future of radio astronomy

Next-generation VLA

- 1.2 – 116 GHz (SKA ← ngVLA → ALMA)
- 244 antennas x 18m diameter (+ 19 x 6m)
- Maximum baseline ~9000 km (0.5-50 milli-arcsec), dense core
- Data rates: up to **10s Gb/sec**

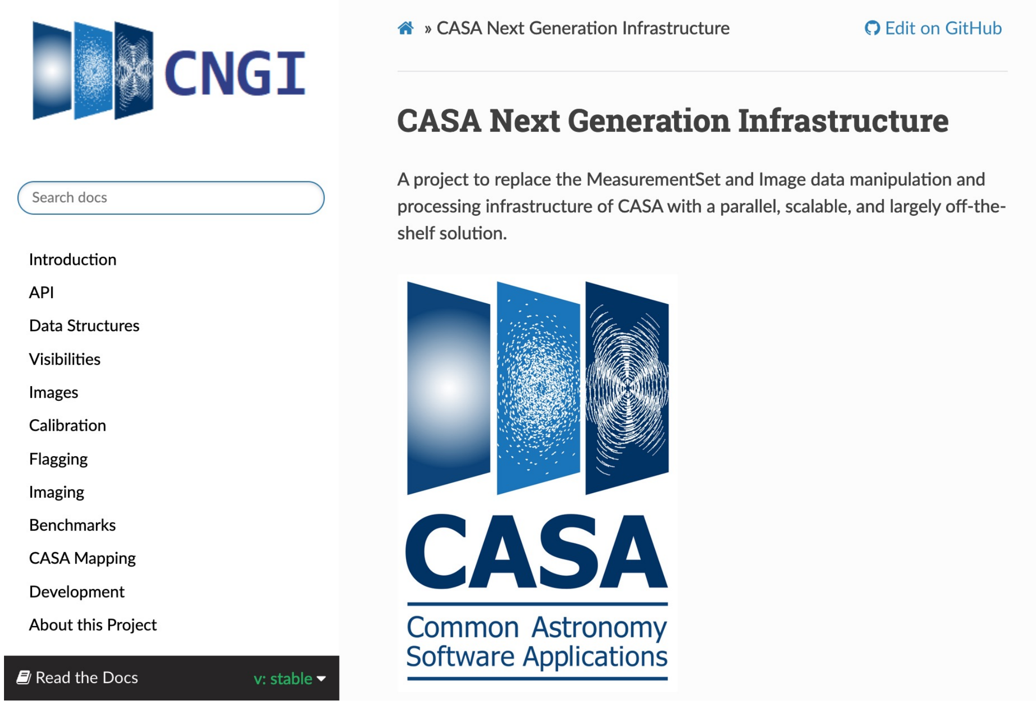


Note: Square Kilometre Array → data rates of global internet traffic

CASA Next Generation Infrastructure

CASA Next Generation Infrastructure (CNGI)

- Infrastructure for next-generation CASA software to meet growing demands of radio telescopes
- New data structures for MeasurementSet and Image contents → built in Python, atop popular technology stack *(numpy, dask, xarray)*
- Prototyping completed in 2021 and made available as a demonstration package to the community → https://cngi-prototype.readthedocs.io/en/stable/

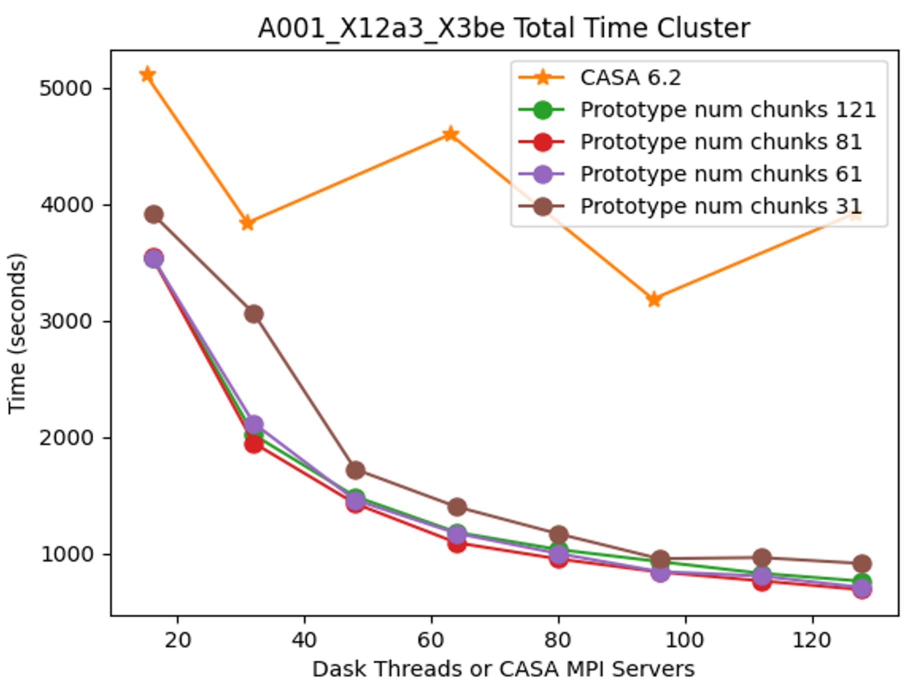


CNGI Demonstration Package Highlights

Credit: Jan-Willem Steeb, Ryan Raba, Andrew McNichols

https://cngi-prototype.readthedocs.io/en/stable/

1. Visibility Data Structure and Image Data Structures
    a. Implemented as Xarray Datasets (XDS)
    b. N-dimensional data variables with named dimensions, coordinate values, and free form attributes
    c. Natively parallelized through Dask
    d. Attached to on-disk storage (in Zarr format), not actually loaded in memory
2. Base set of selection, manipulation, and mathematics developed (natively parallel)
3. Standard and Mosaic Gridders
    a. Written in Python
    b. Numerically equivalent output to CASA6
    c. Greater potential for scalability
4. Benchmarking
    a. Four ALMA (30 GB - 300 GB) and one CHILES (2.5 TB) dataset.
    b. ALMA benchmark completed on a node and cluster (2 to 128 cores)
    c. 2-4x (or more) speedup, same hardware
    d. CHILES benchmark achieved 6.7x speedup on 8 node cluster
    e. Speedup due to CASA not supporting parallel writes



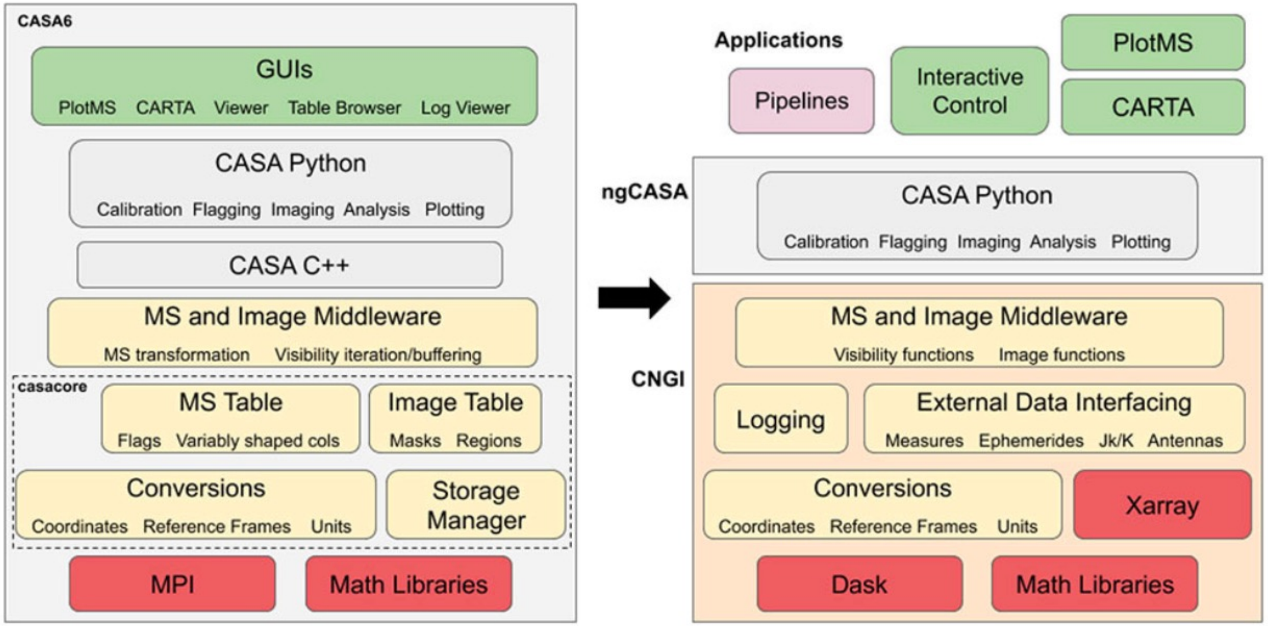
CASA Next Generation Infrastructure

CASA Next Generation Infrastructure (CNGI)

- Infrastructure for next-generation CASA software to meet growing demands of radio telescopes
- New data structures for MeasurementSet and Image contents → built in Python, atop popular technology stack *(numpy, dask, xarray)*
- Prototyping completed in 2021 and made available as a demonstration package to the community → https://cngi-prototype.readthedocs.io/en/stable/

Next-generation CASA (ngCASA)

- design and develop scientific package on top CNGI, aligned with ngVLA requirements and performance (but also serving ALMA and VLA)



Conclusions

- CASA is versatile and leading radio data processing software
- Publicly available - downloadable as all-inclusive tarfile, or pip-wheel installation *(casa.nrao.edu)*
- Supports large number of use-cases *(manual processing, scripting, pipelines, Notebooks)*
- CASA Docs primary resource for users *(casadocs.readthedocs.io)*
- Stable software package, ongoing efforts next-generation CASA

→ *bright future for radio astronomy!*

Thank you!

谢谢

We welcome feedback: casa-feedback@nrao.edu

NRAO ESO 国立天文台 NAOJ ASIAA ALMA CSIRO ASTRON JIVE Joint Institute for VLBI ERIC