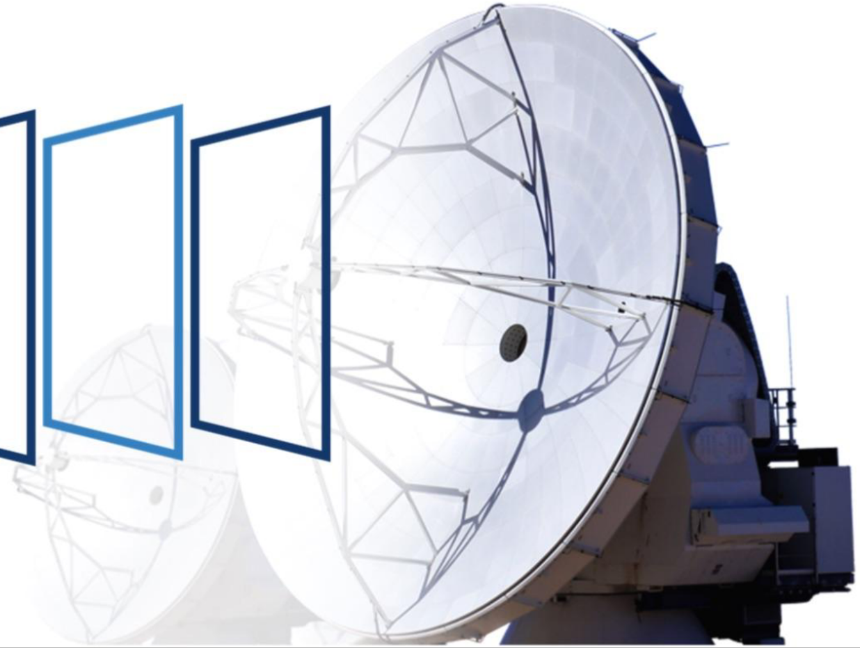
CASA

Common Astronomy Software Applications



Introduction to CASA

Bjorn Emonts

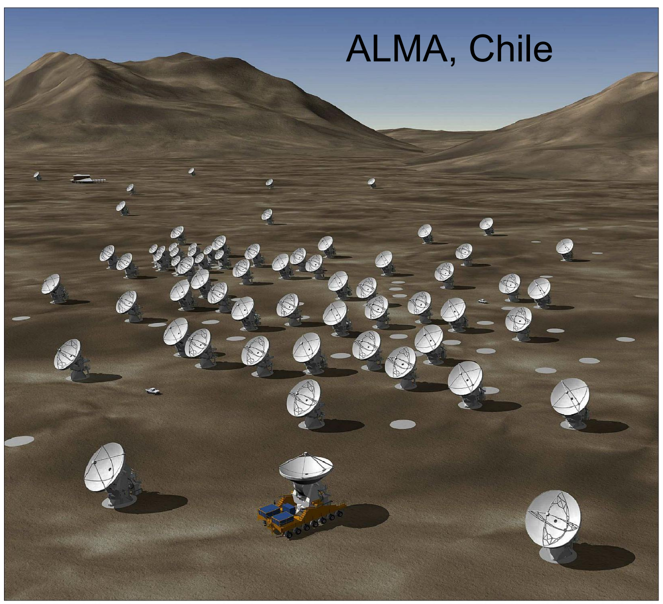
National Radio Astronomy Observatory

CASA User Community Liaison

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.



CASA Team

Urvashi Rao Venkata (NRAO-SO)	*CASA Lead, Lead scientific development*
Sandra Castro (ESO)	*Lead verification testing*
Darrell Schiebel (NRAO-CV)	*Lead visualization, Infrastructure development*
Takeshi Nakazato (NAOJ)	*Lead Single Dish, Scientific development*
Josh Marvil (NRAO-SO)	*Lead scientific validation*

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*
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*
Dirk Petry (ESO)	*Scientific development*
Neal Schweighart (NRAO-CV)	*Scientific development, Verification testing*
Kazuhiko Shimada (NAOJ)	*Scientific development, Single Dish*
Jan-Willem Steeb (NRAO-CV)	*Scientific, Infrastructure development*
Takeshi Shakunaga (NAOJ)	*Scientific development, Single Dish*
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*

ARDG (Algorithm Research & Development Group)

Sanjay Bhatnagar (NRAO) - ARDG Lead
Mingyu (Genie) Hsieh (NRAO)
Martin Pokorny (NRAO)
Preshanth Jagannathan (NRAO)
Srikrishna Sekhar (NRAO, IDIA)

JIVE
Joint Institute for VLBI
ERIC

CASA-VLBI

Ilse van Bemmel (JIVE)	*VLBI, Project Scientist*
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

Adam Leroy (Ohio State Univ - Chair)
Yoshimasa Watanabe (Shibaura I.T. Japan)
Yu-Nung Su (ASIAA Taiwan)
Abhijeet Borkar (ASCR Chech)
Kristina Nyland (NRL Washington DC)

Ruta Kale (NCRA India)
Jihyun Kang (KASI Korea)
Olga Bayandina (INAF Italy)
Imke de Pater (UC Berkeley)
Jane Huang (Univ. Michigan)

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

Adam Leroy (Ohio State Univ - Chair)
Yoshimasa Watanabe (Shibaura I.T. Japan)
Yu-Nung Su (ASIAA Taiwan)
Abhijeet Borkar (ASCR Chech)
Kristina Nyland (NRL Washington DC)
Ruta Kale (NCRA India)
Jihyun Kang (KASI Korea)
Olga Bayandina (INAF Italy)
Imke de Pater (UC Berkeley)
Jane Huang (Univ. Michigan)

Using CASA - This Talk

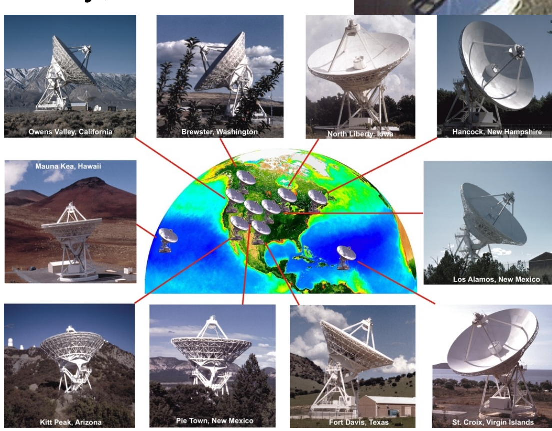
- Radio Interferometry & CASA
- CASA Basics
- Data Processing in CASA
  - Import/export
  - Information
  - Manipulation
  - Calibration
  - Imaging
  - Analysis / Visualization (CARTA)
- Download & Installation
- Documentation & Resources

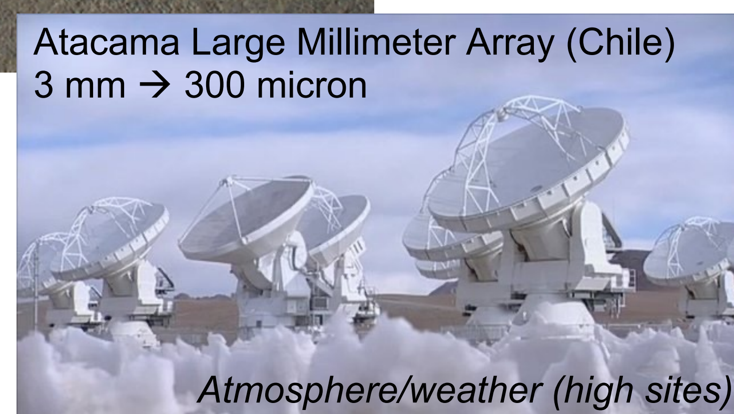
(Not in this talk: initiatives for a next-generation CASA

→ *Talk by Jan-Willem Steeb*)

Radio interferometry

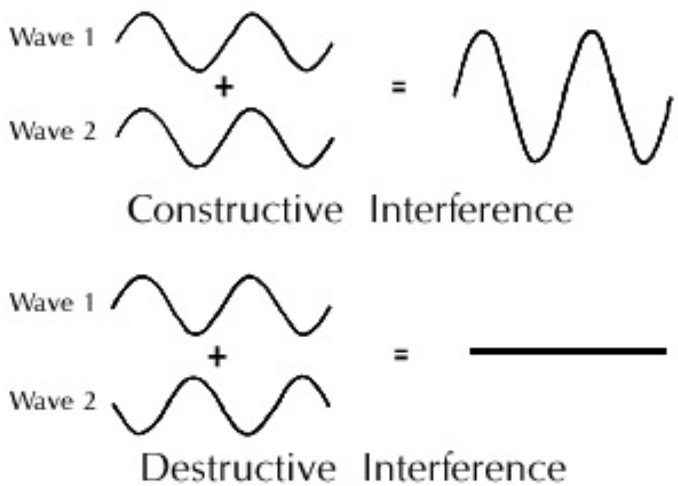
Very Long Baseline Array, VLBA

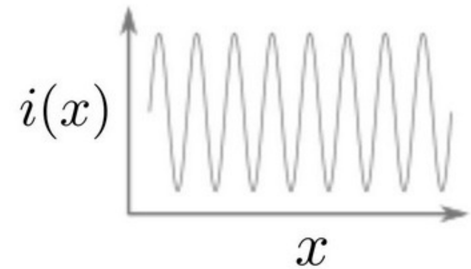


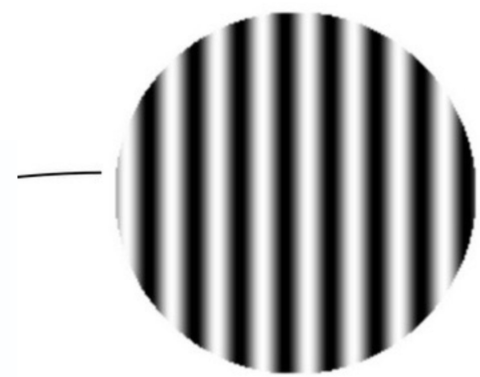


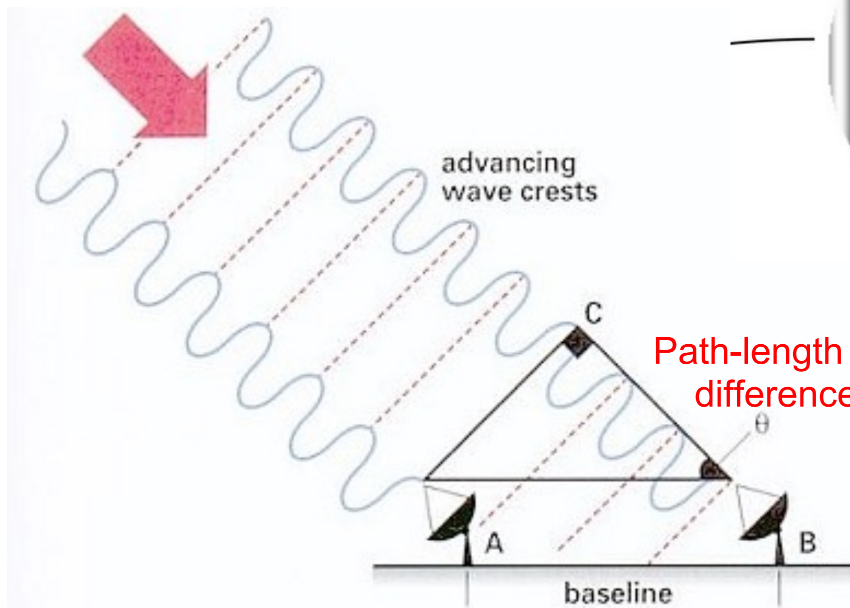
Interferometry

Track point-source
on the sky:



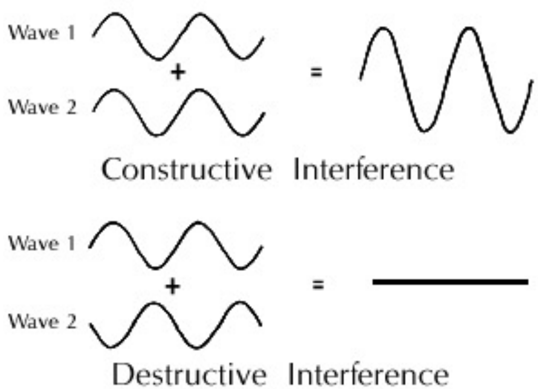


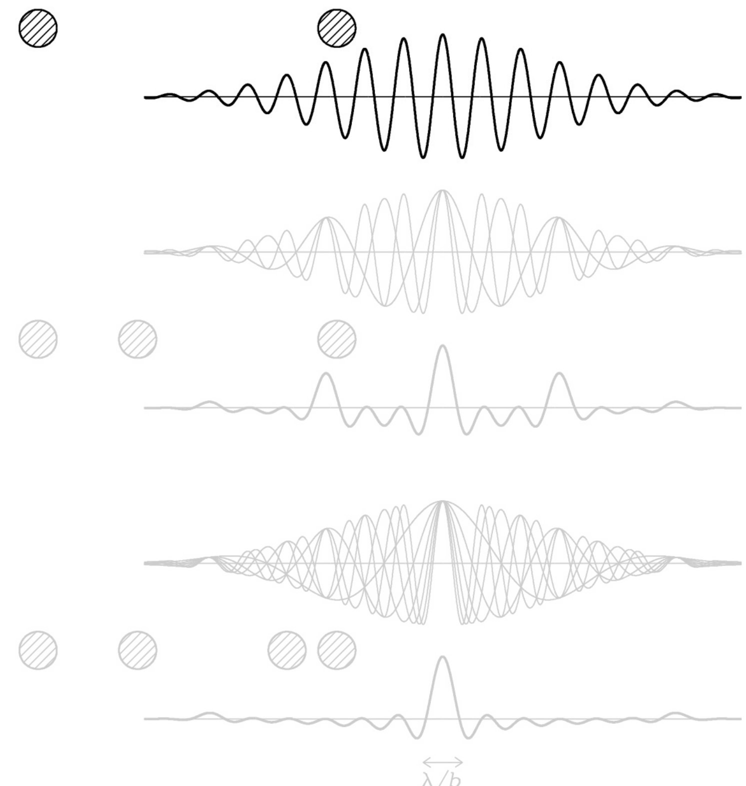




Resolution: **R ~ λ / B**

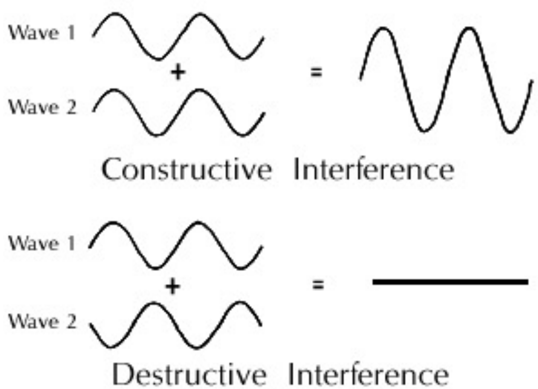
Interferometry

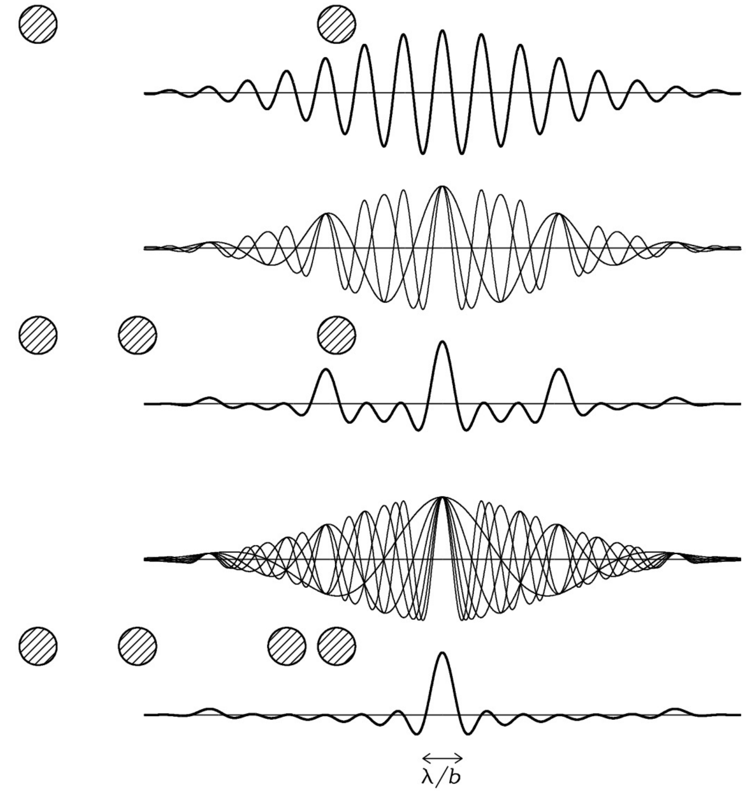




- Primary beam response single antenna (bandwidth)
- Add N antennas, i.e. N(N-1)/2 baselines
- Add complex source structure
- Scan through earth rotation

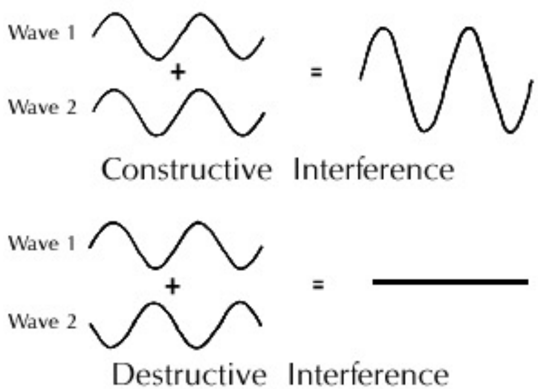
Interferometry

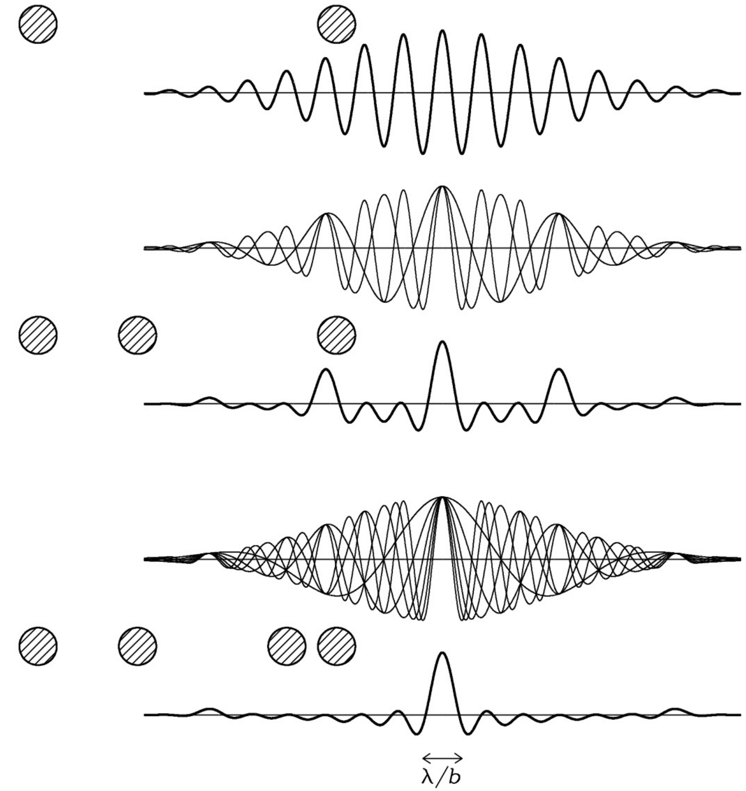




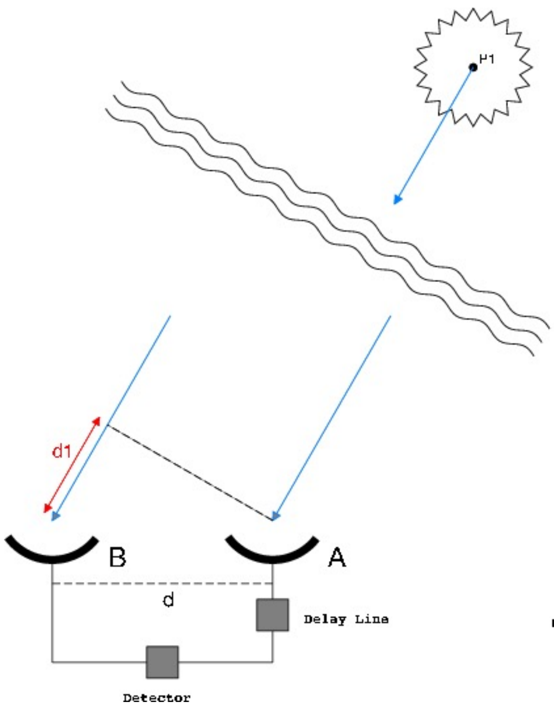
- Primary beam response single antenna (bandwidth)
- Add N antennas, i.e. N(N-1)/2 baselines
- Add complex source structure
- Scan through earth rotation

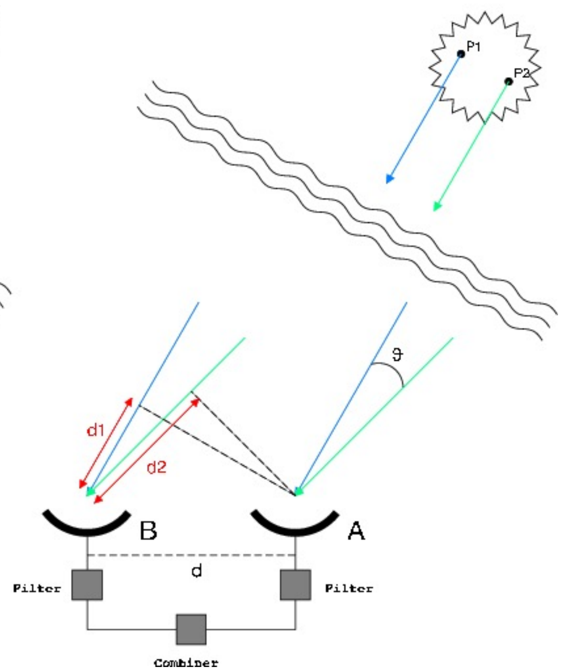
Interferometry



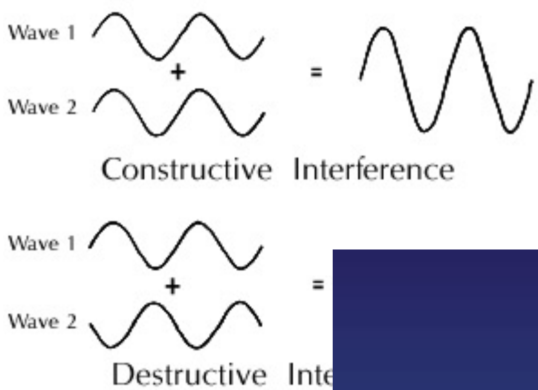


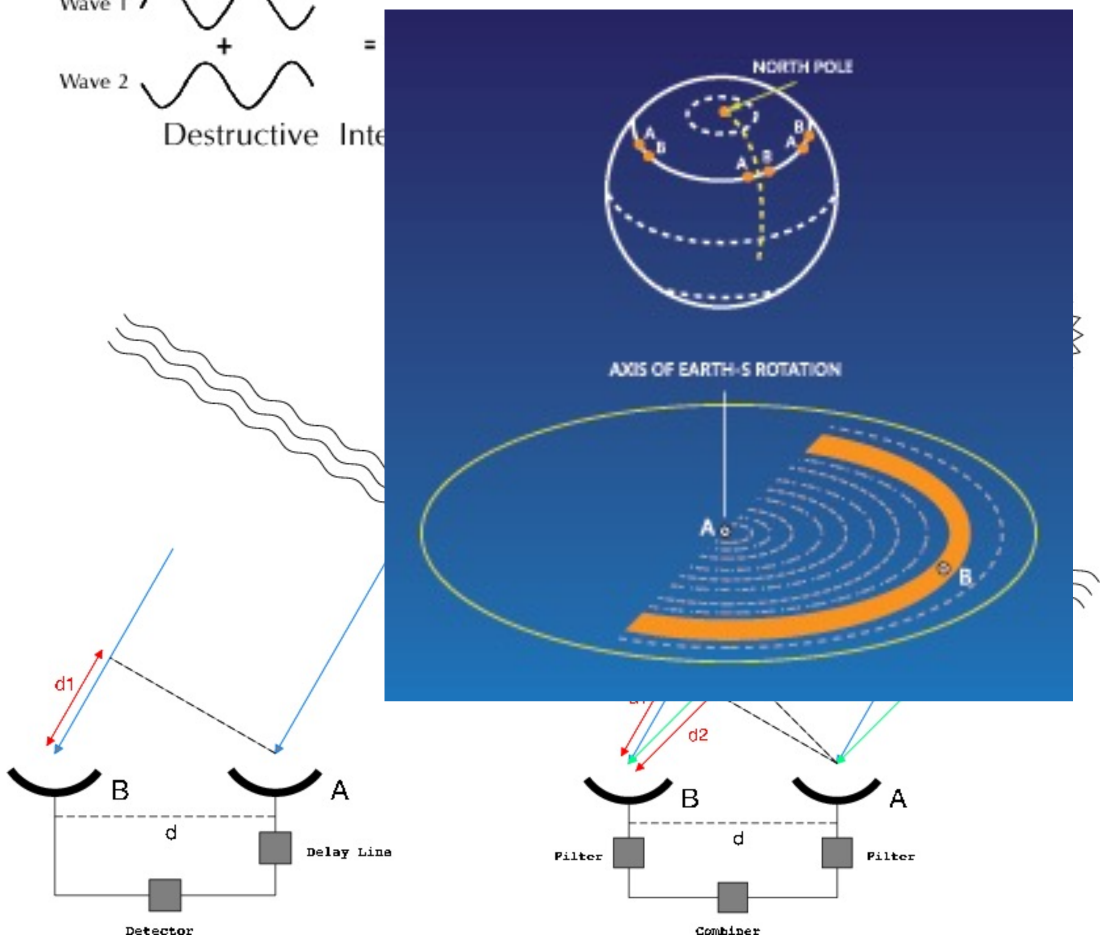
- Primary beam response single antenna (bandwidth)
- Add N antennas, i.e. N(N-1)/2 baselines
- Add complex source structure
- Scan through earth rotation

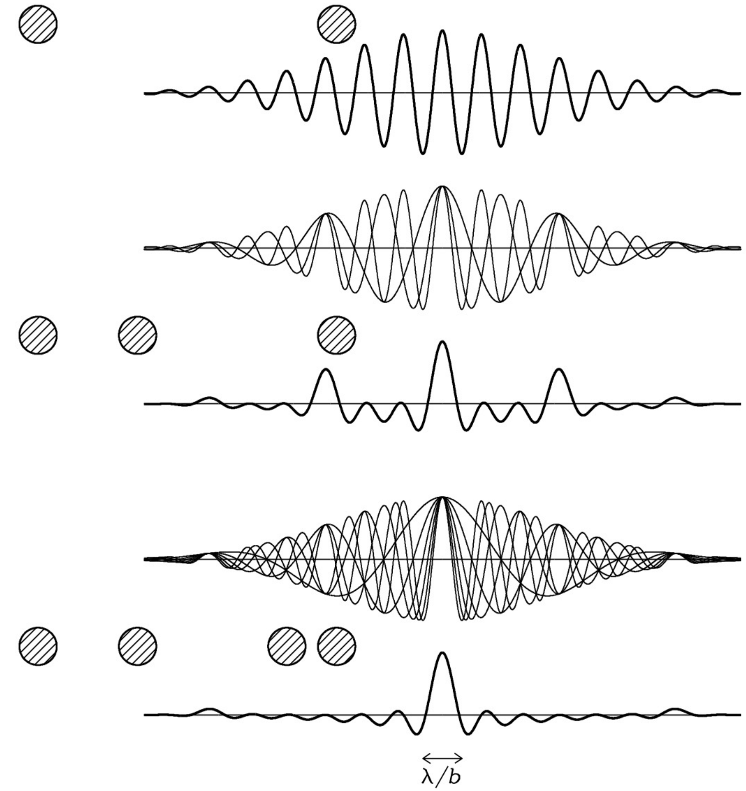




Interferometry

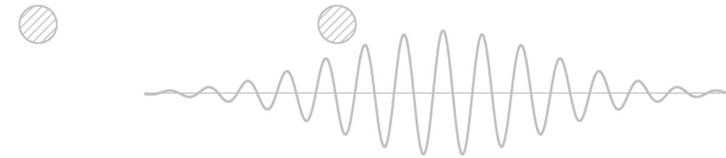






- Primary beam response single antenna (bandwidth)
- Add N antennas, i.e. N(N-1)/2 baselines
- Add complex source structure
- Scan through earth rotation

Interferometry



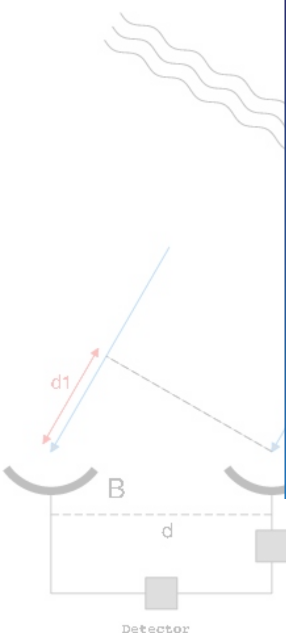
“Visibility”: interferometer response per antenna pair (*i, j*), Δt, Δf, polarization

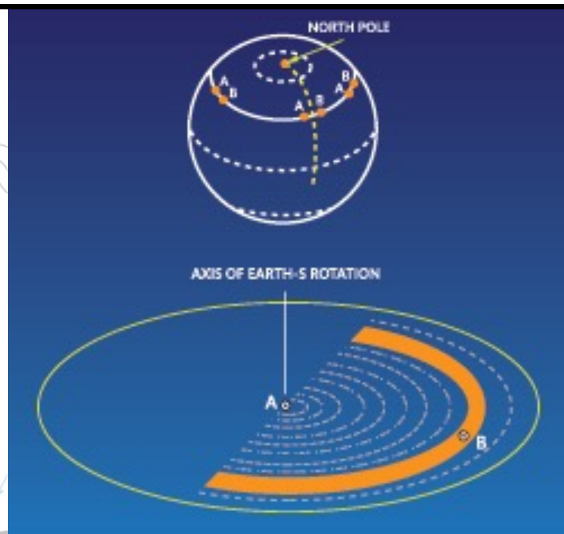
→ Fourier Transform of the sky brightness distribution.

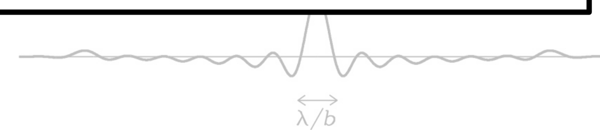
→ “complex”, with *amplitude, phase* information: $V(u,v) = a_r e^{i\varphi r}$

→ Source *brightness, structure*

CASA: raw visibility data → science products

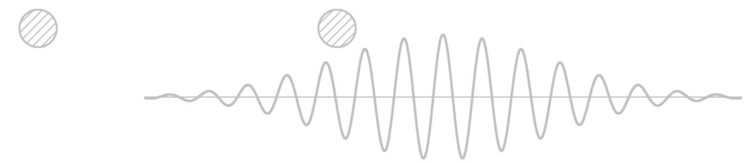






- Primary beam response single antenna (bandwidth)
- Add N antennas, i.e. N(N-1)/2 baselines
- Add complex source structure
- Scan through earth rotation

Interferometry



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

→ Fourier Transform of the sky brightness distribution.

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

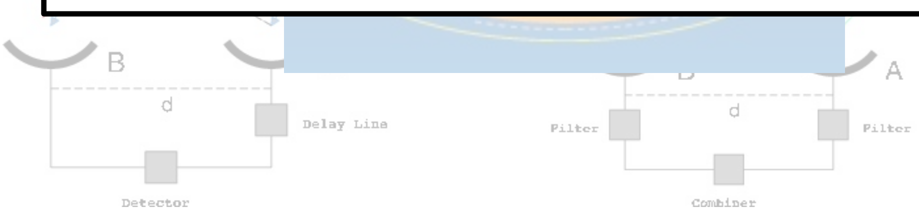
→ Source *brightness, structure*

CASA: raw visibility data → science products

- 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!



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 tasks

Well-defined step in data processing
→ user friendly, parameter input

Example: determine complex time-dependent gains for each antenna and spw

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

- ***CASA task*** – gaincal
- *Input MS* – vis='lowres_data1.ms'
- *Spectral Window selection* – spw='1:10~110'
- *Reference antenna* – refant'4'
- *Phase-only Calibration mode* – calmode='p'

Other parameters, use default!

CASA tasks

Exam

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 (Visualization)

Single Dish

Simulations

importasdm

Import/export
Information
Manipulation
Calibration
Imaging
Analysis

Single Dish
Simulations

(A)SDM, (Astronomy) 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.

CASA

listobs

Import/export

Information

Manipulation

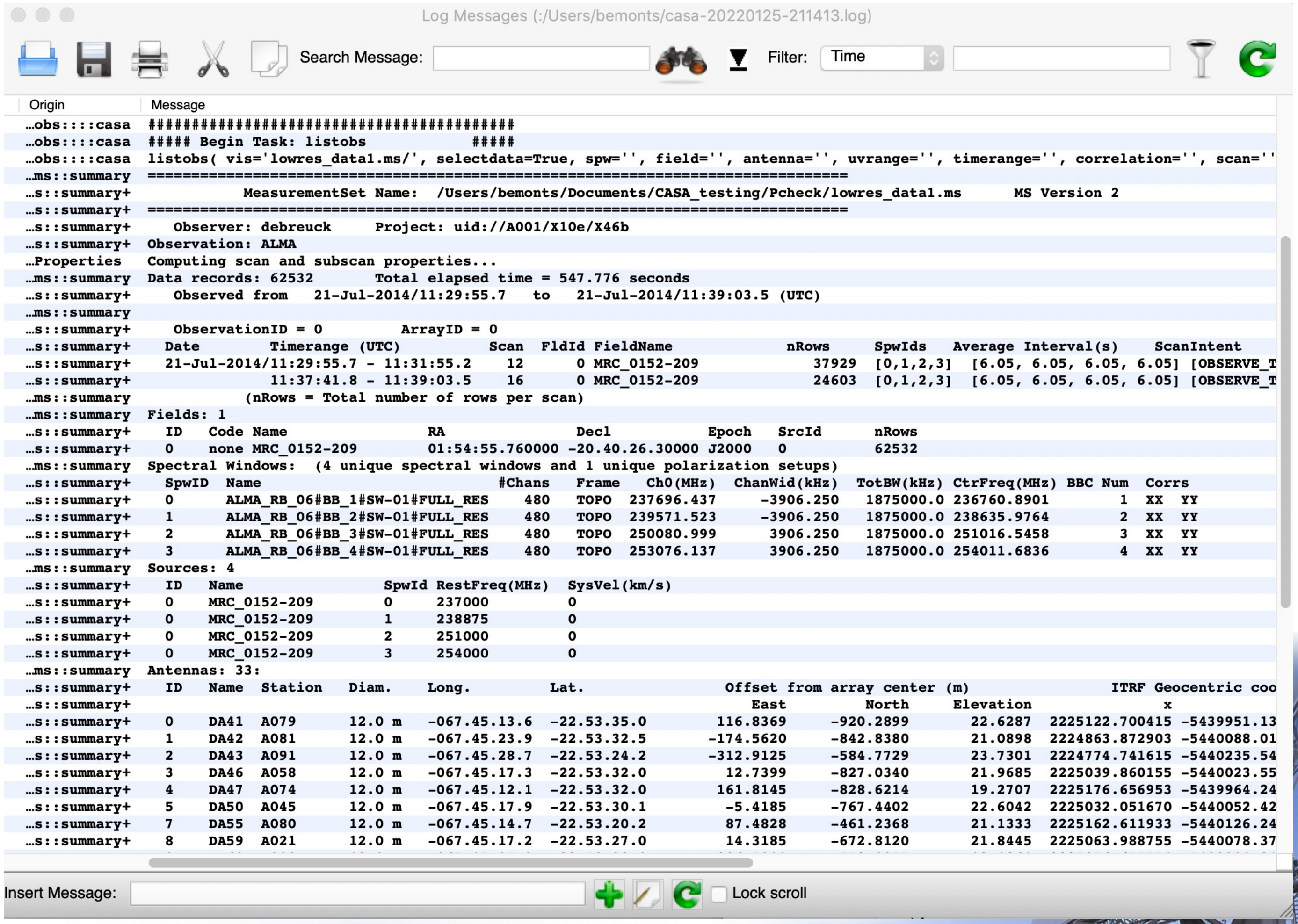
Calibration

Imaging

Analysis

Single Dish

Simulations



CASA

Import/export

Information

Manipulation

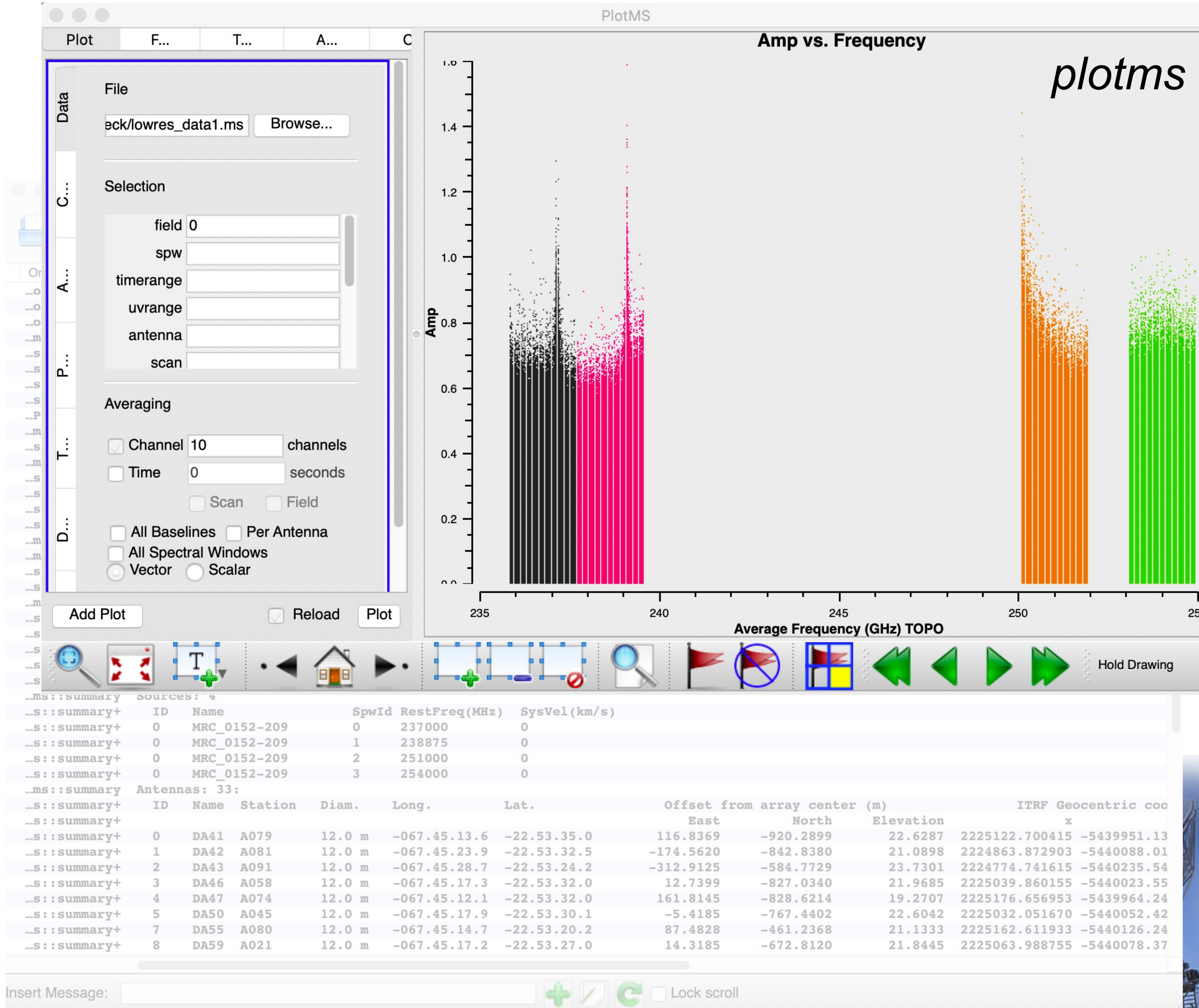
Calibration

Imaging

Analysis

Single Dish

Simulations



CASA

Import/export

Information

Manipulation

Calibration

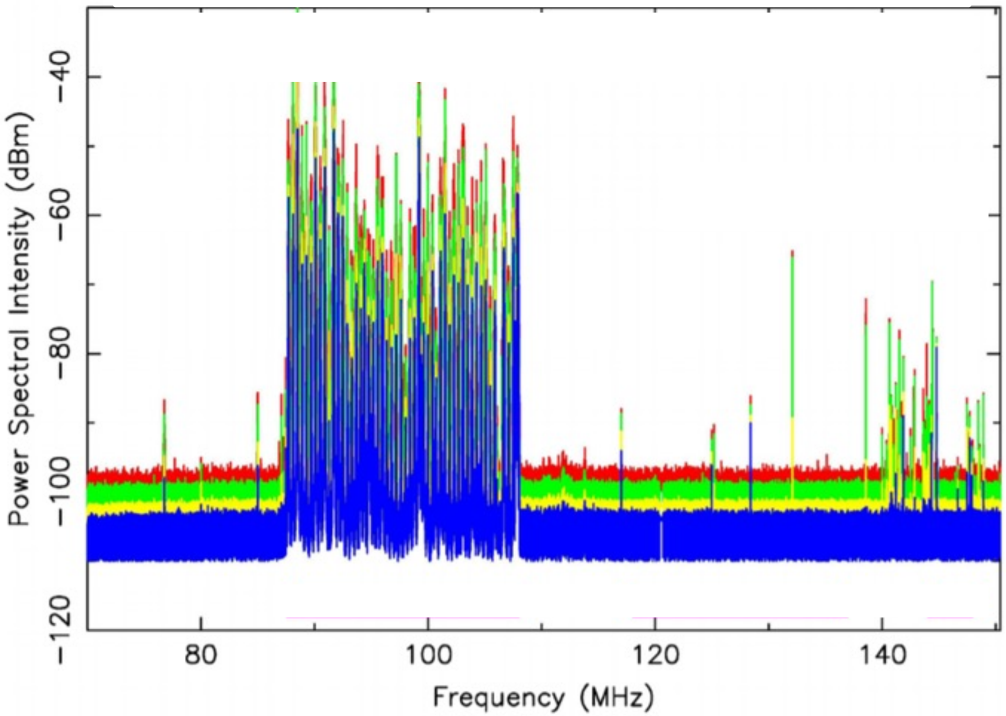
Imaging

Analysis

Single Dish

Simulations

Radio Frequency Interference (RFI) → Flag!



See talk by EvangeliaTremou

CASA

Import/export

Information

Manipulation

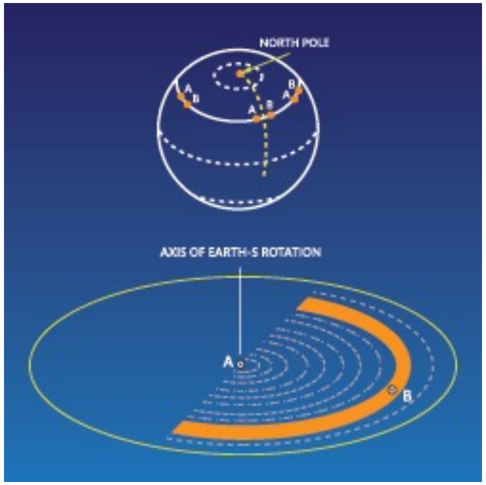
Calibration

Imaging

Analysis

Single Dish

Simulations

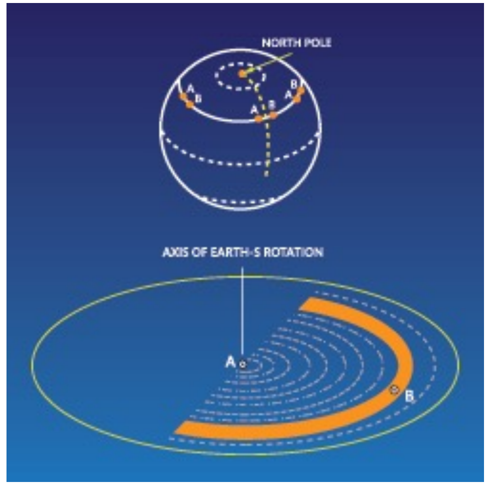


$$\vec{V}_{ij} = J_{ij}\,\vec{V}_{ij}^{IDEAL}$$

CASA

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

- Single Dish
- Simulations



Various talks this week

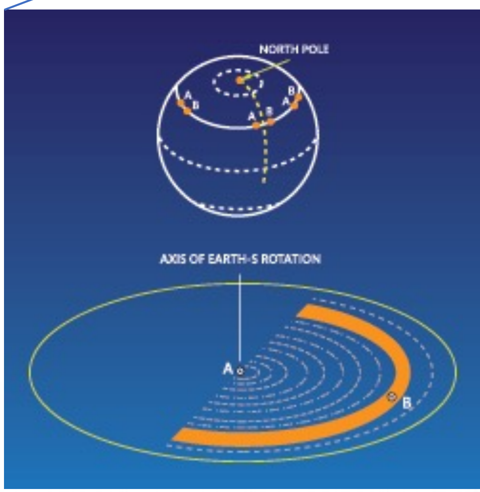
$$\vec{V}_{ij} = J_{ij}\,\vec{V}_{ij}^{IDEAL}$$

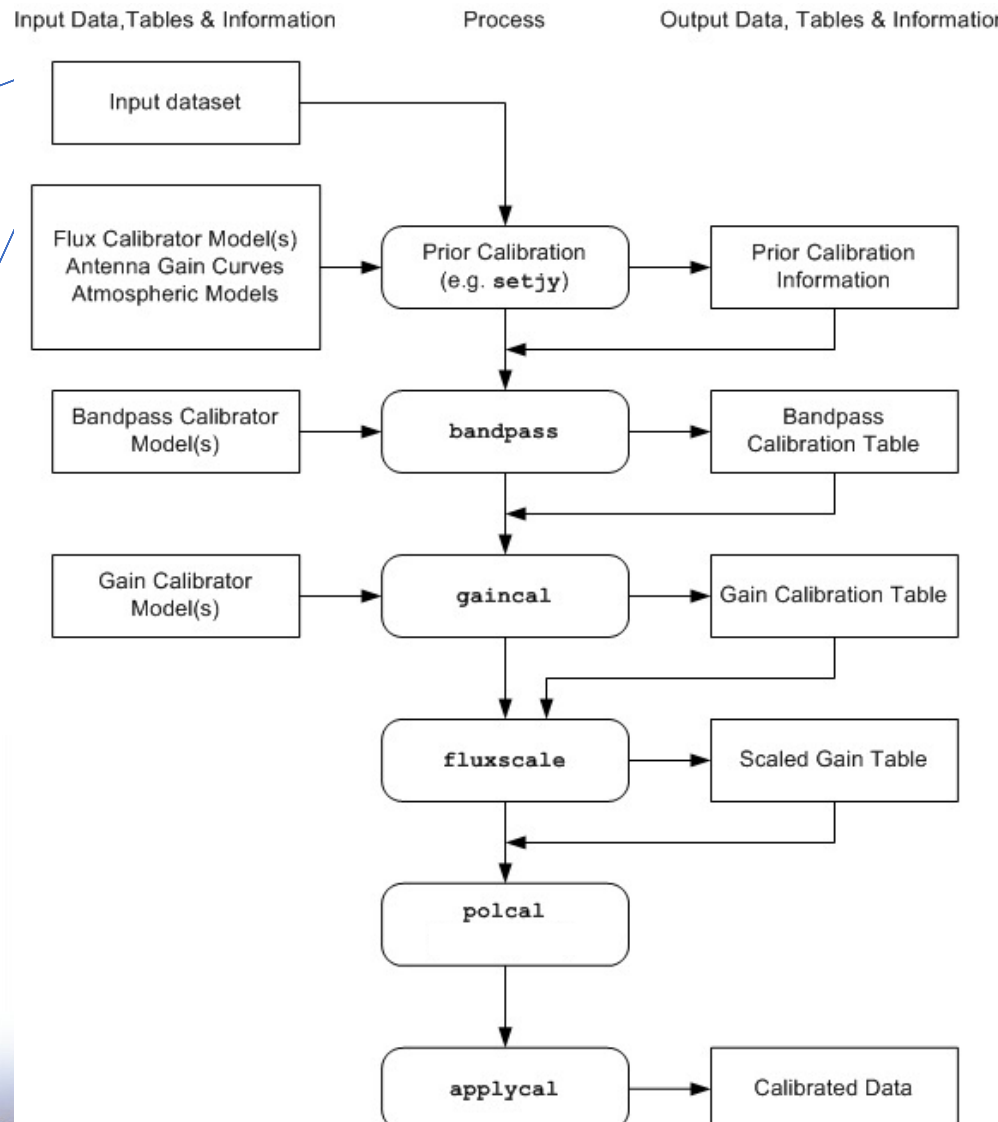
$$\vec{V}_{ij} = M_{ij}\,B_{ij}\,G_{ij}\,D_{ij}\,E_{ij}\,P_{ij}\,T_{ij}\,\vec{V}_{ij}^{IDEAL}$$

- 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

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





CASA

Import/export

Information

Manipulation

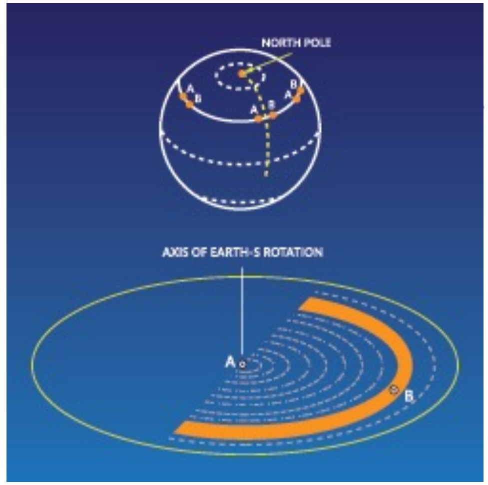
Calibration

Imaging

Analysis

Single Dish

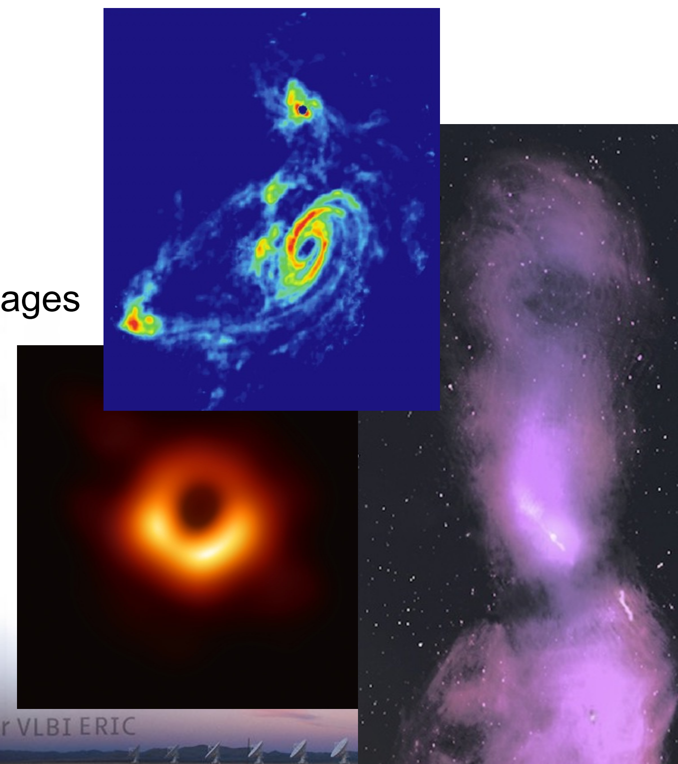
Simulations



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

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

2D continuum images
3D image cubes

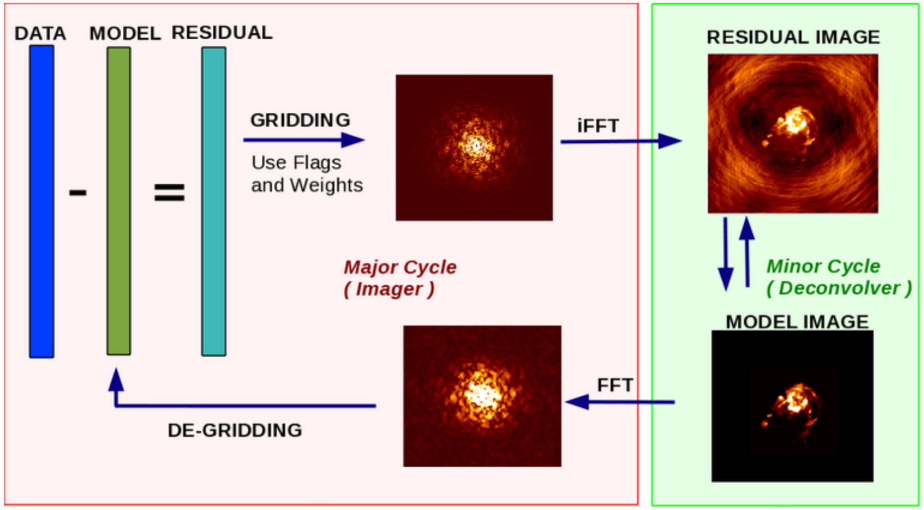


NRAO ESO 国立天文台 NAOJ ASIAA ALMA CSIRO ASTRON JIVE Joint Institute for VLBI ERIC

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



See talk by Preshanth Jagannathan

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 definitior
weighting           = 'natural'             # Weighting scheme (natural,uniform,brig
                                            # briggsbwtaper[experimental])
   uvtaper          = []                    # uv-taper on outer baselines in uv-plar
niter               = 0                     # Maximum number of iterations
usemask             = 'user'                # Type of mask(s) for deconvolution: use
   mask             = ''                    # Mask (a list of image name(s) or regic
   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 (nc
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>:
```

CASA

Import/export

Information

Manipulation

Calibration

Imaging

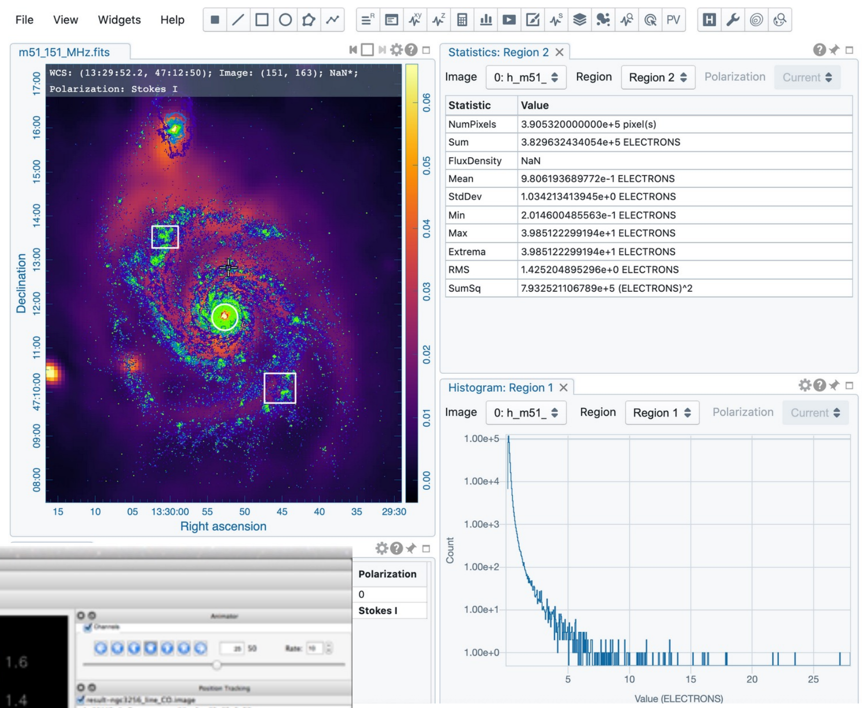
Analysis/

Visualization

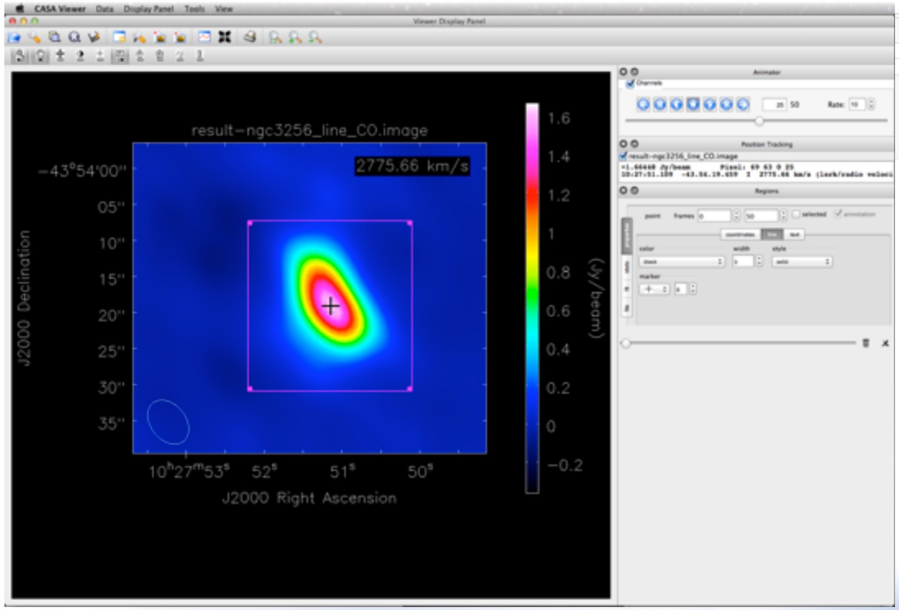
Single Dish

Simulations

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



The old: CASA Viewer



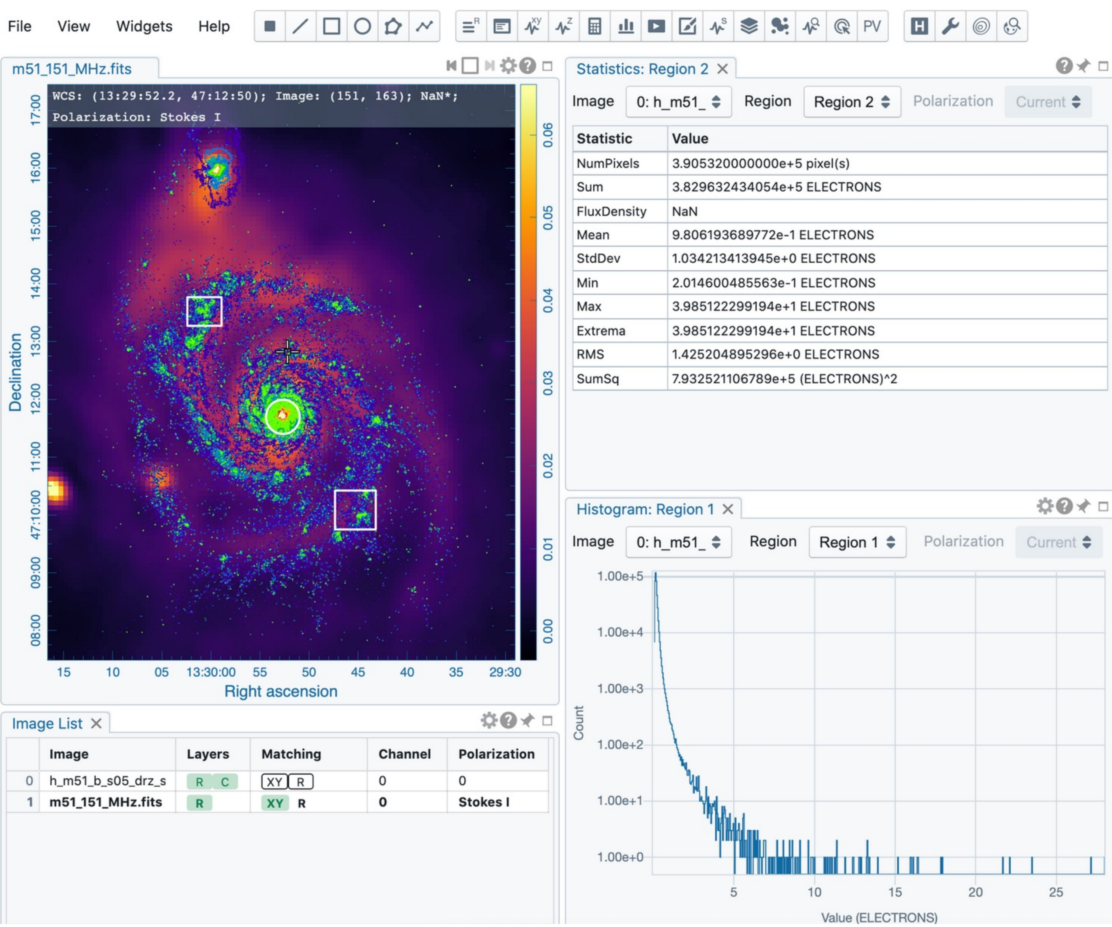
NRAO ESO NAOJ ASIAA ALMA CSIRO ASTRON JIVE Joint Institute for VLBI ERIC

Visualization

Start using this!

See talk by Juergen Ott

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)

CASA

Import/export

Information

Manipulation

Calibration

Imaging

Analysis / Visualization

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

New release every ~2 months!

Latest version: CASA 6.5

The Release Notes and Known Issues of the 6.5 release are a...

CASA 6.5 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 10.15)
General Use (Notes)	CASA 6.5.2 (RH7/8 - Py 3.8) CASA 6.5.2 (RH7 - Py 3.6)	CASA 6.5.2 (OS11 - Py 3.8) CASA 6.5.2 (OS11 - Py 3.6)
ALMA Pipeline (Notes)	CASA 6.4.1 (RH7/8)	CASA 6.4.1 (OS11) CASA 6.4.1 (10.15)
VLA Pipeline (Notes)	CASA 6.4.1 (RH7/8)	CASA 6.4.1 (OS11) CASA 6.4.1 (10.15)

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

Latest version: CASA 6.5

The Release Notes and Known Issues of the 6.5 release are available in CASA Docs

CASA 6.5 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 10.15)
General Use (Notes)	CASA 6.5.2 (RH7/8 - Py 3.8) CASA 6.5.2 (RH7 - Py 3.6)	CASA 6.5.2 (OS11 - Py 3.8) CASA 6.5.2 (OS11 - Py 3.6)
ALMA Pipeline (Notes)	CASA 6.4.1 (RH7/8)	CASA 6.4.1 (OS11) CASA 6.4.1 (10.15)
VLA Pipeline (Notes)	CASA 6.4.1 (RH7/8)	CASA 6.4.1 (OS11) CASA 6.4.1 (10.15)

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 (ca

Monolithic

Pip-wheel

Pipelines (

Compatibil

We execute tasks just like normal Python functions. Many times they will write information to the log or a specified output file, which we then must display.

```
[ ]: from casatasks import listobs

     rc = listobs(vis='sis14_twhya_calibrated_flagged.ms', listfile='obslist.txt', verbose=False, overwrite=True)
     !cat obslist.txt
```

```
================================================================================
           MeasurementSet Name:  /content/sis14_twhya_calibrated_flagged.ms      MS Version 2
================================================================================
   Observer: cqi     Project: uid://A002/X327408/X6f
Observation: ALMA(26 antennas)
Data records: 80563       Total elapsed time = 5647.68 seconds
   Observed from   19-Nov-2012/07:36:57.0   to   19-Nov-2012/09:11:04.7 (UTC)

Fields: 5
  ID   Code Name                RA               Decl           Epoch   SrcId      nRows
  0    none J0522-364           05:22:57.984648 -36.27.30.85128 J2000   0           4200
  2    none Ceres               06:10:15.950590 +23.22.06.90668 J2000   2           3800
  3    none J1037-295           10:37:16.079736 -29.34.02.81316 J2000   3          16000
  5    none TW Hya              11:01:51.796000 -34.42.17.36600 J2000   4          53161
  6    none 3c279               12:56:11.166576 -05.47.21.52464 J2000   5           3402
Spectral Windows:  (1 unique spectral windows and 1 unique polarization setups)
  SpwID  Name                           #Chans   Frame   Ch0(MHz)  ChanWid(kHz)  TotBW(kHz) CtrFreq(MHz) BBC Num  Corrs
  0      ALMA_RB_07#BB_2#SW-01#FULL_RES    384   TOPO  372533.086       610.352    234375.0 372649.9688        2  XX  YY
Antennas: 21 'name'='station'
   ID=   1-4: 'DA42'='A050', 'DA44'='A068', 'DA45'='A070', 'DA46'='A067',
   ID=   5-9: 'DA48'='A046', 'DA49'='A029', 'DA50'='A045', 'DV02'='A077',
   ID= 10-15: 'DV05'='A082', 'DV06'='A037', 'DV08'='A021', 'DV10'='A071',
   ID= 16-19: 'DV13'='A072', 'DV15'='A074', 'DV16'='A069', 'DV17'='A138',
   ID= 20-24: 'DV18'='A053', 'DV19'='A008', 'DV20'='A020', 'DV22'='A011',
   ID= 25-25: 'DV23'='A007'
```

Another example, lets do channel averaging with MSTransform. Here we need to make sure we've deleted the previous output file if/when running multiple times. Since this task doesn't return anything, we can look at the end of the log file to see what happened.

```
[ ]: from casatasks import mstransform

     os.system("rm -fr chanavg.ms")
     mstransform(vis='sis14_twhya_calibrated_flagged.ms', outputvis='chanavg.ms',
                 datacolumn='DATA', chanaverage=True, chanbin=3)
     !tail casa-202*.log
```

```
2021-10-14 17:43:24     INFO    MSTransformManager::parseMsSpecParams   Tile shape is [0]
2021-10-14 17:43:24     INFO    MSTransformManager::parseChanAvgParams  Channel average is activated
2021-10-14 17:43:24     INFO    MSTransformManager::parseChanAvgParams  Channel bin is [3]
2021-10-14 17:43:24     INFO    MSTransformManager::colCheckInfo        Adding DATA column to output MS from input DATA column
2021-10-14 17:43:24     INFO    MSTransformManager::open        Select data
2021-10-14 17:43:24     INFO    MSTransformManager::createOutputMSStructure     Create output MS structure
2021-10-14 17:43:24     INFO    ParallelDataHelper::::casa      Apply the transformations
2021-10-14 17:43:29     INFO    mstransform::::casa     Task mstransform complete. Start time: 2021-10-14 17:43:23.610120 End time: 2021-10-14 17:43:29.323998
2021-10-14 17:43:29     INFO    mstransform::::casa     ##### End Task: mstransform          #####
2021-10-14 17:43:29     INFO    mstransform::::casa     ##########################################
```

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

Latest version: CASA 6.5

The Release Notes and Known Issues of the 6.5 release are available in CASA Docs

CASA 6.5 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 10.15)
General Use (Notes)	CASA 6.5.2 (RH7/8 - Py 3.8) CASA 6.5.2 (RH7 - Py 3.6)	CASA 6.5.2 (OS11 - Py 3.8) CASA 6.5.2 (OS11 - Py 3.6)
ALMA Pipeline (Notes)	CASA 6.4.1 (RH7/8)	CASA 6.4.1 (OS11) CASA 6.4.1 (10.15)
VLA Pipeline (Notes)	CASA 6.4.1 (RH7/8)	CASA 6.4.1 (OS11) CASA 6.4.1 (10.15)

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

Full Monolithic Distribution

	Python 2.7	Python 3.6	Python 3.7	Python 3.8
RHEL 6	5.8	<=6.3		
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 12 ARM*				>=6.4

Modular CASA

	Python 2.7	Python 3.6	Python 3.7	Python 3.8
RHEL 6		<=6.3	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 12 ARM				>=6.4

CASA at NRAO

Pre-installed on NRAO machines:

> casa → 6.5.2 (default latest CASA release)
> casa-vla → 6.2.1 (latest for VLA pipeline)
> casa-pipe → 6.2.1 (recommended for workshop, incl. pipeline)
> casa-alma → 6.4.1 (latest for ALMA pipeline)

> casa -ls → list all CASA versions
> casa -r 6.4.0-16 → load specific version “6.4.0-16”
> casa-vla --pipeline → load also the pipeline-specific tasks

See: https://casa.nrao.edu/CASANMandCV.shtml

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.

6.5.2 Release

CASA 6.5.2 can now be downloaded for general use. CASA 6.5.2 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:

- deconvolve: new task for image-domain deconvolution.
- uvcontsub: new implementation, old uvcontsub task deprecated.
- fringefit: support added for 'uvrange' parameter.
- tclean: new iteration control parameter 'nmajor'.
- sdimaging: new parameter 'enablecache' for improved performance.
- mstransform: parameter 'douvcontsub' deprecated.
- flagdata: mode='shadow' now uses the uvw values from the UVW column.
- tclean/tsdimaging: improved runtime performance of ephemeris imaging.
- simulator tool: new parameter 'simint' in sm.settrop() to control time granularity, down to 0.1s.
- ImageAnalysis tool: new string 'mbret' parameter added to 'image.restoringbeam()'.
- casalog tool: new method 'getOrigin()' implemented to retrieve origin of messages.

For more details on these and other new features, see the CASA 6.5.2 Release Notes.

CASA is developed by an international consortium of scientists based at the National Radio Astronomical Observatory (NRAO), the European Southern Observatory (ESO), the National Astronomical Observatory of Japan (NAOJ), the Academia Sinica Institute of Astronomy and

CASA Docs (casadocs.readthedocs.io)

CASA

Search docs

Release Information
Index
API
Task List
Using CASA

Read the Docs v: stable

Versions
latest stable v6.5.2 v6.5.1 v6.5.0
v6.4.4 v6.4.3 v6.4.1 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.

6.5.2 Release

CASA 6.5.2 can now be downloaded for general use. CASA 6.5.2 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:

- deconvolve: new task for image-domain deconvolution.
- uvcontsub: new implementation, old uvcontsub task deprecated.
- fringefit: support added for 'uvrange' parameter.
- tclean: new iteration control parameter 'nmajor'.
- sdimaging: new parameter 'enablecache' for improved performance.
- mstransform: parameter 'douvcontsub' deprecated.
- flagdata: mode='shadow' now uses the uvw values from the UVW column.
- tclean/tsdimaging: improved runtime performance of ephemeris imaging.
- simulator tool: new parameter 'simint' in sm.settrop() to control time granularity, down to 0.1s.
- ImageAnalysis tool: new string 'mbret' parameter added to 'image.restoringbeam()'.
- casalog tool: new method 'getOrigin()' implemented to retrieve origin of messages.

For more details on these and other new features, see the CASA 6.5.2 Release Notes.

CASA is developed by an international consortium of scientists based at the National Radio Astronomical Observatory (NRAO), the European Southern Observatory (ESO), the National Astronomical Observatory of Japan (NAOJ), the Academia Sinica Institute of Astronomy and

CASA Docs (casadocs.readthedocs.io)

CASA

Search docs

⊟ Release Information

Highlights

Release Notes

⊟ Known Issues

General

Installation

Scripting

statwt

mstransform

cvel

bandpass

polcal

setjy

uvcontsub

cal library

VLA Switched Power

fringefit

fixvis

Read the Docs v: stable

Known Issues

Summary Most Important Issues

- The **Adaptive Scale Pixel (asp)** deconvolution algorithm in **tclean** is experimental, and we welcome user feedback.
- 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. *(fixed in CASA 6.5.1)*
- 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.
- **sdimaging** will crash or create incorrect images if there exist some spectra taken at a time t that fall outside all pointing intervals of a specific antenna.

General

CASA Docs (casadocs.readthedocs.io)

Modular Packages

Pip wheels for casatools and casatasks are available as Python 3 modules. This allows simple installation and import into standard Python environments. The casatools wheel is necessarily a binary wheel so there may be some compatibility issues for some time as we work toward making wheels available for important Python configurations.

Make sure you have set up your machine with the *necessary prerequisite libraries* first. Then a la carte installation of desired modules (from a Linux terminal window) as follows:

```
$: python3 -m venv myvenv
$: source myvenv/bin/activate
(myvenv) $: pip install --upgrade pip wheel
```

Now pick whichever subset of the available CASA packages you are interested in. Package dependencies are handled automatically by pip, with the exception of **casadata** which must be explicitly installed and updated by the user (see External Data). The following packages are available:

```
(myvenv) $: pip install casatools==6.5.2.26
(myvenv) $: pip install casatasks==6.5.2.26
(myvenv) $: pip install casaplotms==1.8.7
(myvenv) $: pip install casaviewer==1.6.6
(myvenv) $: pip install casampi==0.5.01
(myvenv) $: pip install casashell==6.5.2.26
(myvenv) $: pip install casadata==2022.9.5
(myvenv) $: pip install casaplotserver==1.4.6
(myvenv) $: pip install almatasks==1.5.2
(myvenv) $: pip install casatestutils==6.5.2.26
```

Note for Mac M1 users: For macOS 12 on an ARM-based M1 chip, users will need to install the wheels of CASA version 11 for x86 architecture. For that, we recommend to use the following command to pip install the CASA wheels:

```
(myvenv) $: arch -x86_64 python3 -m pip install ...
```

Users are advised to use a Python virtual environment (venv) and specific module version numbers as shown above. Giving an invalid number (like 999) to the pip install command is an effective way to list all available version numbers.

List all available versions of a module (a hack):

CASA Doc

Search docs

⊟ **Release Information**

- Highlights
- Release Notes
- ⊞ Known Issues
- Compatibility
- ⊟ **Installation**
  - **Monolithic Distribution**
  - Modular Packages

Index

API

Task List

Using CASA

CASA Fundamentals

External Data

Calibration & Visibilities

Imaging & Analysis

CARTA

Pipeline

Simulations

Parallel Processing

Memo Series & Knowledgebase

Community Examples

Change Log

Read the Docs v: stable

Full Monolithic Distribution

	Python 2.7	Python 3.6	Python 3.7	Python 3.8
RHEL 6	5.8	<=6.3		
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 12 ARM*				>=6.4

Modular CASA

	Python 2.7	Python 3.6	Python 3.7	Python 3.8
RHEL 6		<=6.3	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 12 ARM				>=6.4

ption of the

et to

`1: cannot r`

install apps

vacy ->

".

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

CASA Docs (casadocs.readthedocs.io)

CASA

Search docs

Safari d the Docs v: stable

API

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

- almatasks
- casadata
- casalith
- casaplotms
- casashell
- casatasks
- casatools
- casaviewer
- configuration

CASA Docs

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


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
  - A few Imaging and Calibration examples

CASA Reference Paper

PASP in press.
arXiv: 2210.02276

CASA, the Common Astronomy Software Applications for Radio Astronomy

The CASA Team

Ben Bean,[1] Sanjay Bhatnagar,[2] Sandra Castro,[3] Jennifer Donovan Meyer,[4] Bjorn Emonts,[4] Enrique Garcia,[3] Robert Garwood,[4] Kumar Golap,[2] Justo Gonzalez Villalba,[3] Pamela Harris,[2] Yohei Hayashi,[5] Josh Hoskins,[4] Mingyu Hsieh,[2] Preshanth Jagannathan,[2] Wataru Kawasaki,[5] Aard Keimpema,[6] Mark Kettenis,[6] Jorge Lopez,[4] Joshua Marvil,[2] Joseph Masters,[4] Andrew McNichols,[4] David Mehringer,[4] Renaud Miel,[5] George Moellenbrock,[2] Federico Montesino,[3] Takeshi Nakazato,[5] Juergen Ott,[2] Dirk Petry,[3] Martin Pokorny,[2] Ryan Raba,[4] Urvashi Rau,[2] Darrell Schiebel,[4] Neal Schweighart,[4] Srikrishna Sekhar,[7,2] Kazuhiko Shimada,[5] Des Small,[6] Jan-Willem Steeb,[4] Kanako Sugimoto,[5] Ville Suoranta,[4] Takahiro Tsutsumi,[2] Ilse M. van Bemmel,[6] Marjolein Verkouter,[6] Akeem Wells,[4] Wei Xiong,[1] Arpad Szomoru,[6] Morgan Griffith,[4] Brian Glendenning[2] and Jeff Kern[4]

[1]*National Radio Astronomy Observatory, 800 Bradbury Dr., SE Ste 235, Albuquerque, NM 87106, USA*
[2]*National Radio Astronomy Observatory, P.O. Box O, Socorro, NM 87801, USA*
[3]*European Southern Observatory, Karl Schwarzschild Strasse 2, D-85748 Garching, Germany*
[4]*National Radio Astronomy Observatory, 520 Edgemont Road, Charlottesville, VA 22903*
[5]*National Astronomical Observatory of Japan, 2-21-1 Osawa, Mitaka, Tokyo 181-8588, Japan*
[6]*Joint Institute for VLBI ERIC, Oude Hoogeveensedijk 4, 7991 PD Dwingeloo, The Netherlands*
[7]*Inter-University Institute for Data Intensive Astronomy, University of Cape Town, Rondebosch, Cape Town, 7701, South Africa*

(Accepted by PASP on 27 Sept 2022)

ABSTRACT

CASA, the *Common Astronomy Software Applications*, is the primary data processing software for the Atacama Large Millimeter/submillimeter Array (ALMA) and the Karl G. Jansky Very Large Array (VLA), and is frequently used also for other radio telescopes. The CASA software can handle data from single-dish, aperture-synthesis, and Very Long Baseline Interferometery (VLBI) telescopes. One of its core functionalities is to support the calibration and imaging pipelines for ALMA, VLA, VLA Sky Survey (VLASS), and the Nobeyama 45m telescope. This paper presents a high-level overview of the basic structure of the CASA software, as well as procedures for calibrating and imaging astronomical radio data in CASA. CASA is being developed by an international consortium of scientists and software engineers based at the National Radio Astronomical Observatory (NRAO), the European Southern Observatory (ESO), the National Astronomical Observatory of Japan (NAOJ), and the Joint Institute for VLBI European Research Infrastructure Consortium (JIV-ERIC), under the guidance of NRAO.

CASA Reference Paper

PASP in press.
arXiv: 2210.02276
arXiv: 2210.02275

CASA, the Common Astronomy Software Applications for Radio Astronomy

The CASA Team

Ben Bean,[1] Sanjay Bhatnagar,[2] Sandra Castro,[3] Jennifer Donovan Meyer,[4] Bjorn Emonts,[4] Enrique Garcia,[3] Robert Garwood,[4] Kumar Golap,[2] Justo Gonzalez Villalba,[3] Pamela Harris,[2] Yohei Hayashi,[5] Josh Hoskins,[4] Mingyu Hsieh,[2] Preshanth Jagannathan,[2] Wataru Kawasaki,[5] Aard Keimpema,[6] Mark Kettenis,[6] Jorge Lopez,[4] Joshua Marvil,[2] Joseph Masters,[4] Andrew McNichols,[4] David Mehringer,[4] Renaud Miel,[5] George Moellenbrock,[2] Federico Montesino,[3] Takeshi Nakazato,[5] Juergen Ott,[2] Dirk Petry,[3] Martin Pokorny,[2] Ryan Raba,[4] Urvashi Rau,[2] Darrell Schiebel,[4] Neal Schweighart,[4] Srikrishna Sekhar,[7,2] Kazuhiko Shimada,[5] Des Small,[6] Jan-Willem Steeb,[4] Kanako Sugimoto,[5] Ville Suoranta,[4] Takahiro Tsutsumi,[2] Ilse M. van Bemmel,[6] Marjolein Verkouter,[6] Akeem Wells,[4] Wei Xiong,[1] Arpad Szomoru,[6] Morgan Griffith,[4] Brian Glendenning[2] and Jeff Kern[4]

[1] *National Radio Astronomy Observatory, 800 Bradbury Dr., SE Ste 235, Albuquerque, NM 87106, USA*

[2] *National Radio Astronomy Observatory, P.O. Box O, Socorro, NM 87801, USA*

[3] *European Southern Observatory, Karl Schwarzschild Strasse 2, D-85748 Garching, Germany*

[7] *Inter-Universit*

CASA
the Atac
(VLA),
from sin
of its cor
Survey (
basic str
radio dat
engineers
Observat
for VLBI

CASA on the fringe – Development of VLBI processing capabilities for CASA

Ilse M. van Bemmel,[1] Mark Kettenis,[1] Des Small,[1] Michael Janssen,[2] George A. Moellenbrock,[3] Dirk Petry,[4] Ciriaco Goddi,[5,6] Justin D. Linford,[3] Kazi L. J. Rygl,[7] Elisabetta Liuzzo,[7] Benito Marcote,[1] Olga S. Bayandina,[1,8] Neal Schweighart,[3] Marjolein Verkouter,[1] Aard Keimpema,[1] Arpad Szomoru,[1] and Huib Jan van Langevelde[1,9,10]

[1] *Joint Institute for VLBI ERIC (JIVE), Oude Hoogeveensedijk 4, 7991 PD Dwingeloo, The Netherlands*

[2] *Max-Planck-Institut für Radioastronomie, Auf dem Hügel 69, D-53121 Bonn, Germany*

[3] *National Radio Astronomy Observatory, P.O. Box O, Socorro, NM 87801, USA*

[4] *European Southern Observatory, Karl-Schwarzschild-Strasse 2, 85748 Garching, Germany*

[5] *Dipartimento di Fisica, Universitá degli Studi di Cagliari, SP Monserrato-Sestu km 0.7, I-09042 Monserrato, Italy*

[6] *INAF - Osservatorio Astronomico di Cagliari, via della Scienza 5, I-09047 Selargius (CA), Italy*

[7] *INAF-Istituto di Radioastronomia & Italian ALMA Regional Centre, Via P. Gobetti 101, I-40129 Bologna, Italy*

[8] *INAF – Osservatorio Astrofisico di Arcetri, Largo E. Fermi 5, 50125 Firenze, Italy*

[9] *Leiden Observatory, Leiden University, Postbus 2300, 9513 RA Leiden, The Netherlands*

[10] *University of New Mexico, Department of Physics and Astronomy, Albuquerque, NM 87131, USA*

ABSTRACT

New functionality to process Very Long Baseline Interferometry (VLBI) data has been implemented in the CASA package. This includes two new tasks to handle fringe fitting and VLBI-specific amplitude calibration

CASA resources

- **CASA Docs:** official CASA documentation https://casadocs.readthedocs.io
- **CASA Website:** official CASA website (downloads) https://casa.nrao.edu
- **CASA Reference:** paper on arXiv:2210.02276
- **CASA email lists:** casa-announce → announcements, releases
  casa-news → CASA Newsletter

Subscribe!
(casa.nrao.edu)

VLA / ALMA instrument teams:

- **CASA Guides:** data reduction strategies (telescope-specific)
  https://casaguides.nrao.edu
- **Helpdesks:** VLA / ALMA data reduction questions
  https://help.nrao.edu https://help.almascience.org

  (coming: CASA Bug Report System)

Contact CASA Team:

- **casa-feedback@nrao.edu:** general feedback