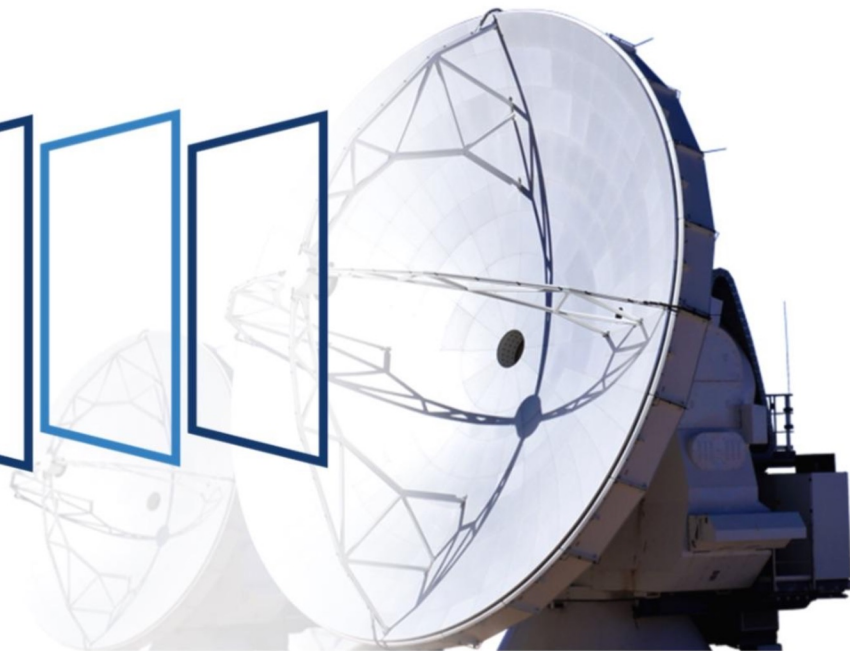


CASA

Common Astronomy
Software Applications



Introduction to CASA

Bjorn Emonts

National Radio Astronomy Observatory

CASA User Community Liaison

CASA team-members attending VLBI workshop:

George Moellenbrock

Neal Schweighart

Victor de Souza Magalhaes (remote)



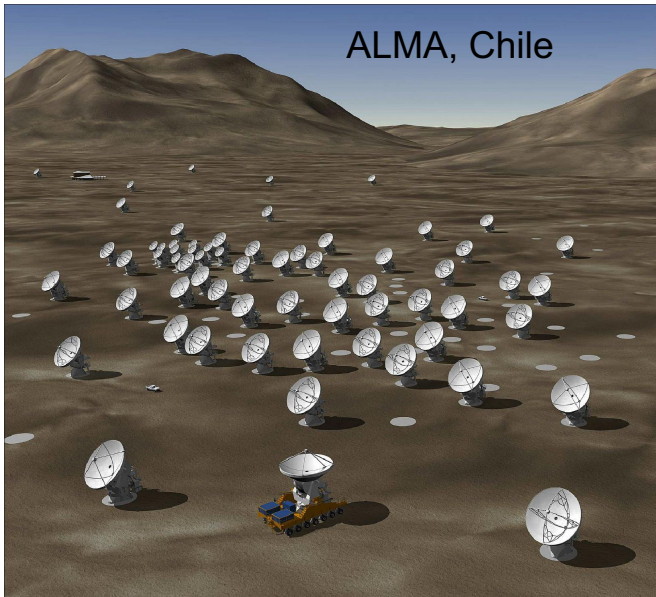
OPTICON
RadioNet
Pilot

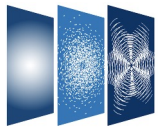


JIVE
Joint Institute for VLBI ERIC

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 (e.g., EVN, GMRT).



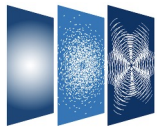


This Talk

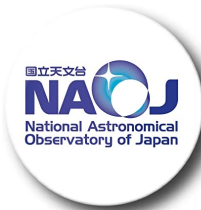
- CASA Team and Development process
- Radio astronomy
 - Basics of Radio Interferometry
- CASA
 - Software content
 - Data processing: calibration, imaging, visualization (CARTA)
 - Download & installation
 - CASA 6.5.5
 - CASA Documentation
 - Reference papers
- Future of radio astronomy & CASA
 - CASA Next Generation Infrastructure (CNGI)



JIVE
Joint Institute for VLBI ERIC



CASA Team



JIVE

Joint Institute for VLBI
ERIC

CASA-VLBI

Urvashi Rau (NRAO-SO)
 Sandra Castro (ESO)
 Josh Marvil (NRAO-SO)
 George Moellenbrock (NRAO-SO)
 Takeshi Nakazato (NAOJ)
 Darrell Schiebel (NRAO-CV)
 Jan-Willem Steeb (NRAO-CV)
 Ville Suoranta (NRAO-CV)

CASA Lead, Lead scientific development
Lead verification testing
Lead scientific validation
Lead Calibration and VLBI
Lead Single Dish, Scientific development
Lead visualization, Infrastructure development
Lead infrastructure development
Lead Release Engineering

Ilse van Bemmel (JIVE)
 Mark Kettenis (JIVE)
 Des Small (JIVE)
 Arpad Szomoru (JIVE)
 Marjolein Verkouter (JIVE)
 Aard Keipema (JIVE)

VLBI, Project Scientist
VLBI, development
VLBI, development
VLBI, management
VLBI, management
VLBI, Jupyter kernel

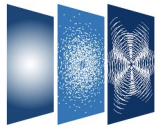
Victor de Souza Magalhaes (NRAO-ALBQ)
 Bjorn Emonts (NRAO-CV)
 Enrique Garcia (ESO)
 Bob Garwood (NRAO-CV)
 Kumar Golap (NRAO-SO)
 Justo Gonzalez Villalba (ESO)
 Pam Harris (NRAO-SO)
 Yohei Hayashi (NAOJ)
 Josh Hoskins (NRAO-CV)
 Wataru Kawasaki (NAOJ)
 Jorge Lopez (NRAO-CV)
 Andrew McNichols (NRAO-CV)
 Dave Mehringer (NRAO-CV)
 Renaud Miel (NAOJ)
 Federico Montesino (ESO)
 Dirk Petry (ESO)
 Neal Schweighart (NRAO-CV)
 Kazuhiko Shimada (NAOJ)
 Takeshi Shakunaga (NAOJ)
 Tak Tsutsumi (NRAO-SO)
 Akeem Wells (NRAO-CV)
 Wei Xiong (NRAO-ALBQ)

Scientific development
User Community Liaison
Infrastructure development
Infrastructure, Verification testing
Scientific development
Scientific development
Data visualization
Scientific development, Single Dish
Scientific development, Infrastructure
Scientific development, Single Dish
Infrastructure, Scientific development
Infrastructure, Scientific development
Scientific development, Verification testing
Scientific development, Single Dish
Infrastructure, Scientific development
Scientific development
Scientific development, Verification testing
Scientific development, Single Dish
Scientific development, Single Dish
Scientific development, Verification testing
Verification testing
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)





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

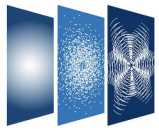
(A. Leroy, R. Kale, Y. Watanabe, J. Kang, O. Bayandina, Y-N. Su, A. Borkar, I. de Pater, J. Huang, K. Nyland)

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



JIVE
Joint Institute for VLBI ERIC





CASA Development process

Stakeholder/internal requests (~6 months) ——— *Bugs (any time)*



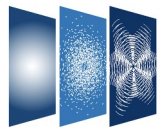
- Prioritization by CASA team
- Development

- Definition requirement / specification ——— *Documentation* ———
- Development
- Internal verification: testing → functional, unit, stakeholder
- External validation (pipeline testing)
- Documentation (benchmarking) ←

- Release: **General CASA releases** → ~2-3 months
(CASA+pipeline releases → ~yearly)

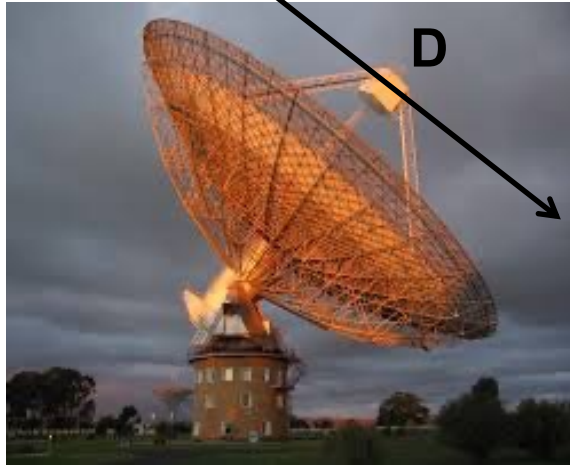


JIVE
Joint Institute for VLBI ERIC



Interferometry Basics

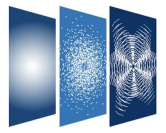
Single Dish & Interferometry



$$\text{Resolution} \approx \lambda / D$$

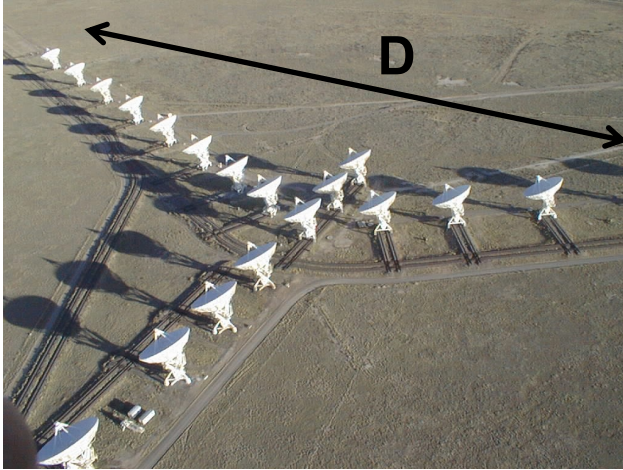


JIVE
Joint Institute for VLBI ERIC



Interferometry Basics

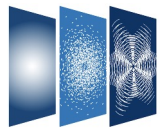
Single Dish & Interferometry



$$\text{Resolution} \approx \lambda / D$$

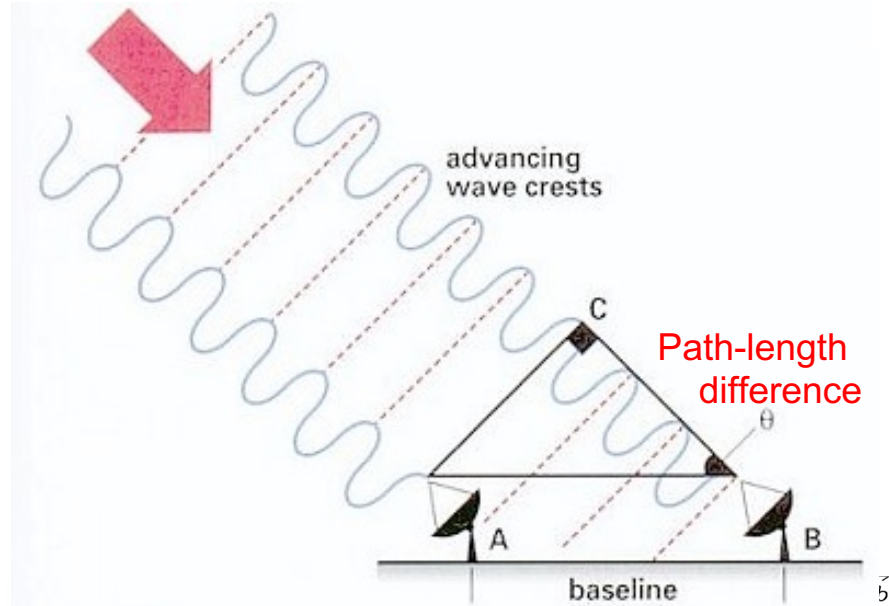
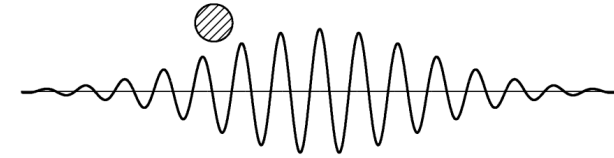


JIVE
Joint Institute for VLBI ERIC

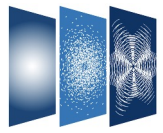


Interferometry Basics

Interferometry

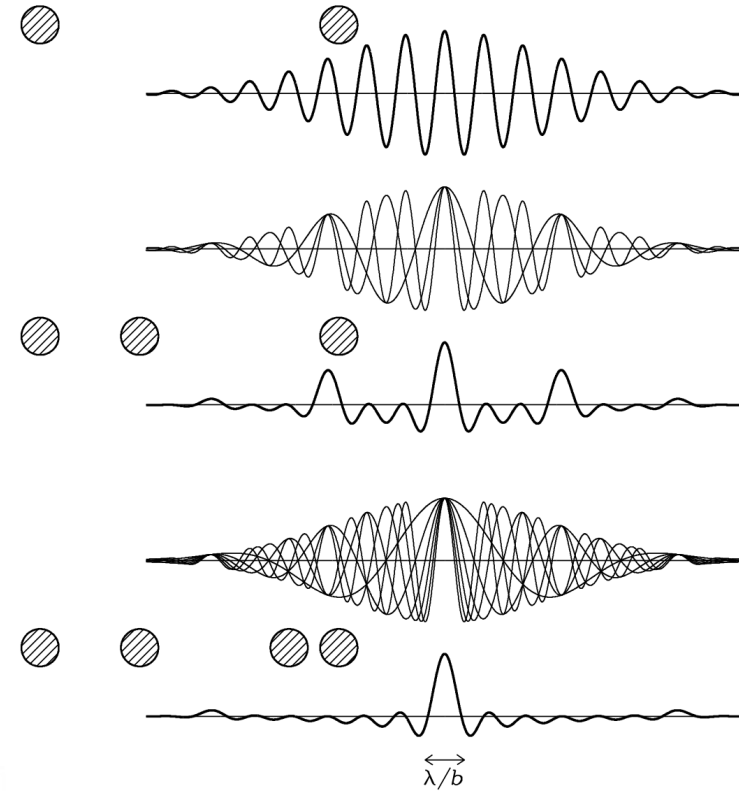
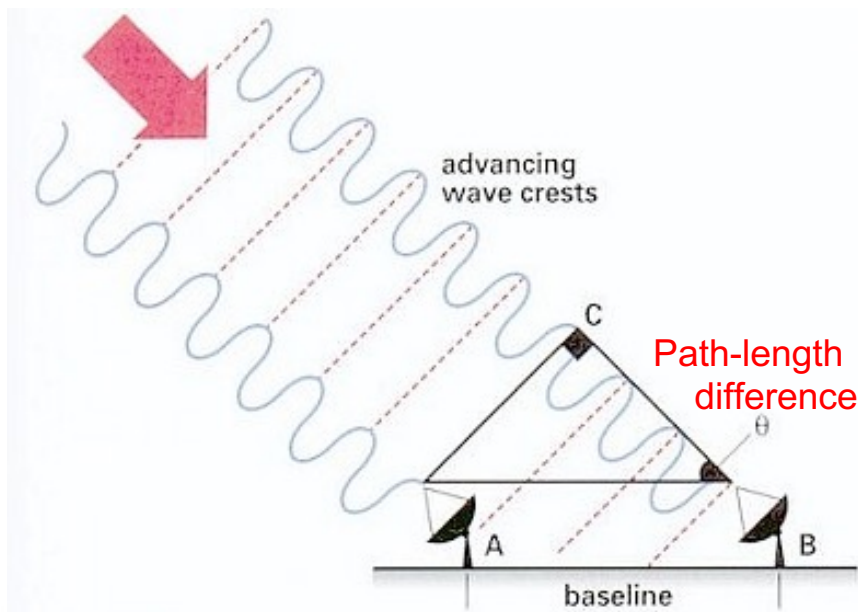


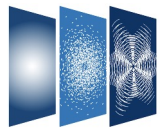
JIVE
Joint Institute for VLBI ERIC



Interferometry Basics

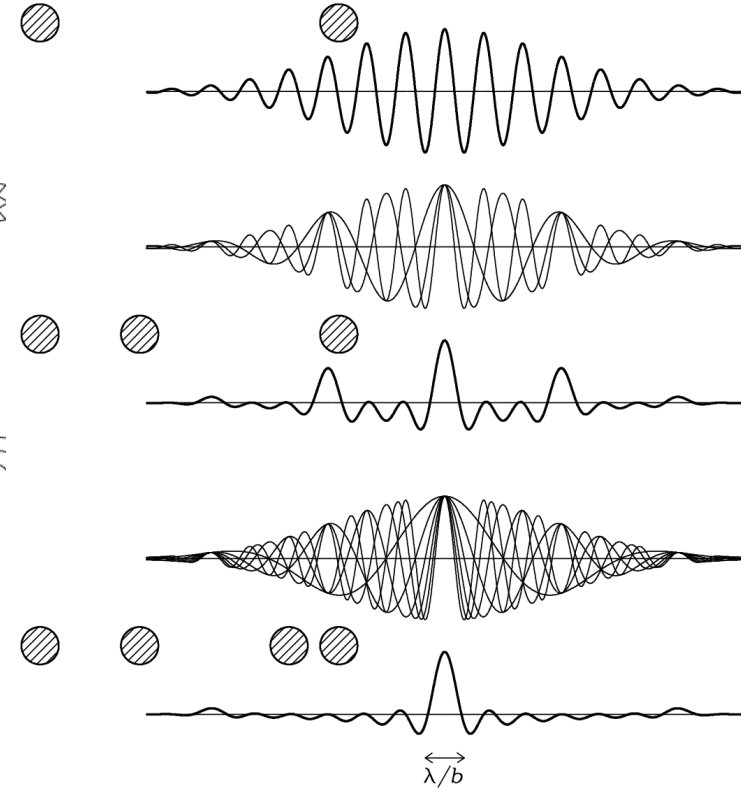
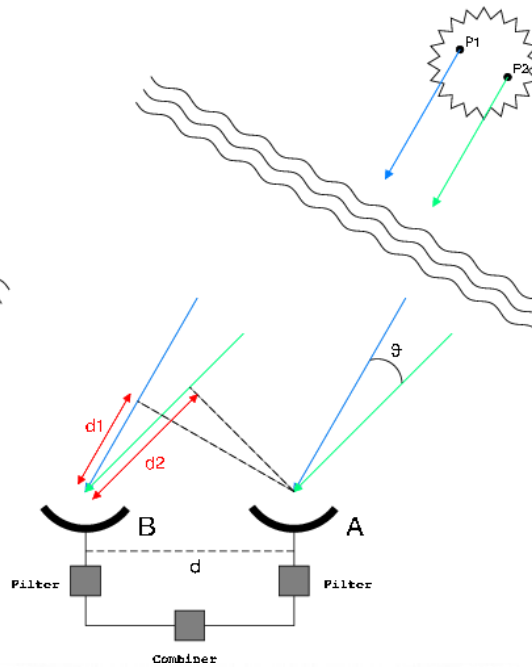
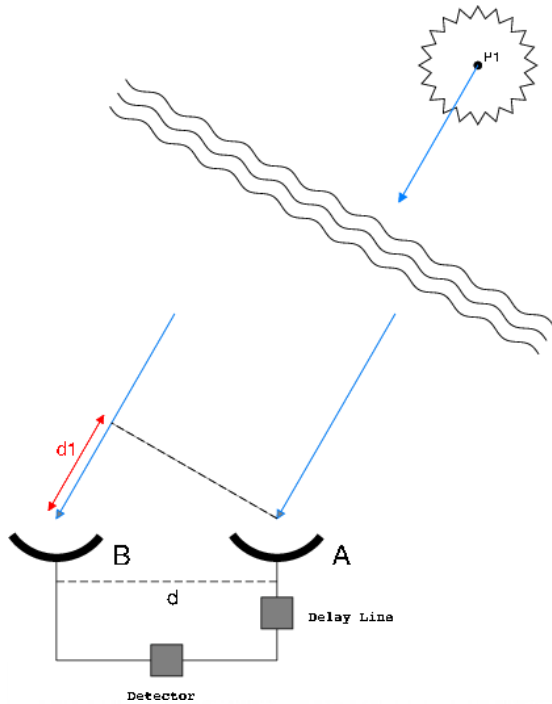
Interferometry

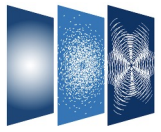




Interferometry Basics

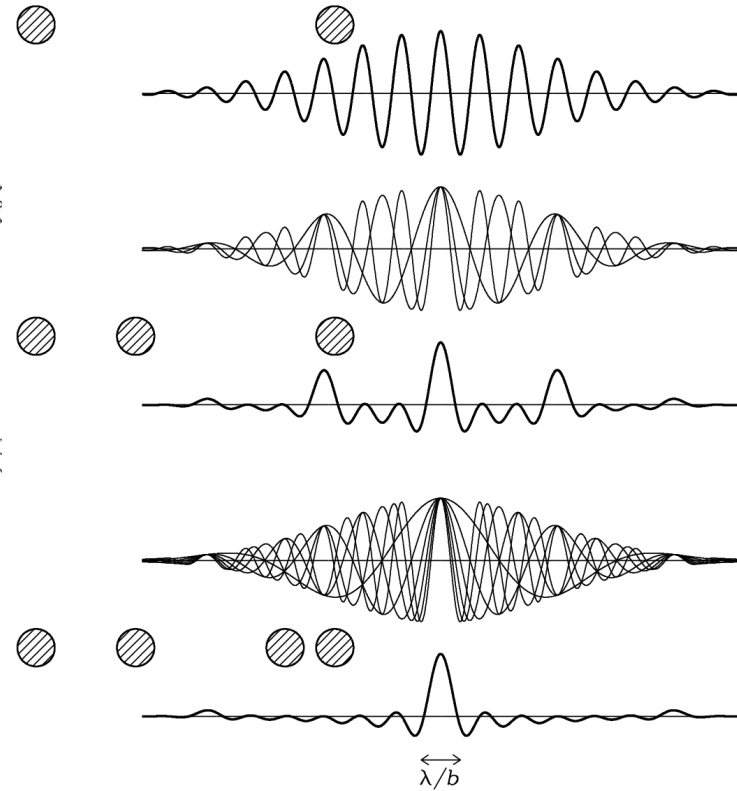
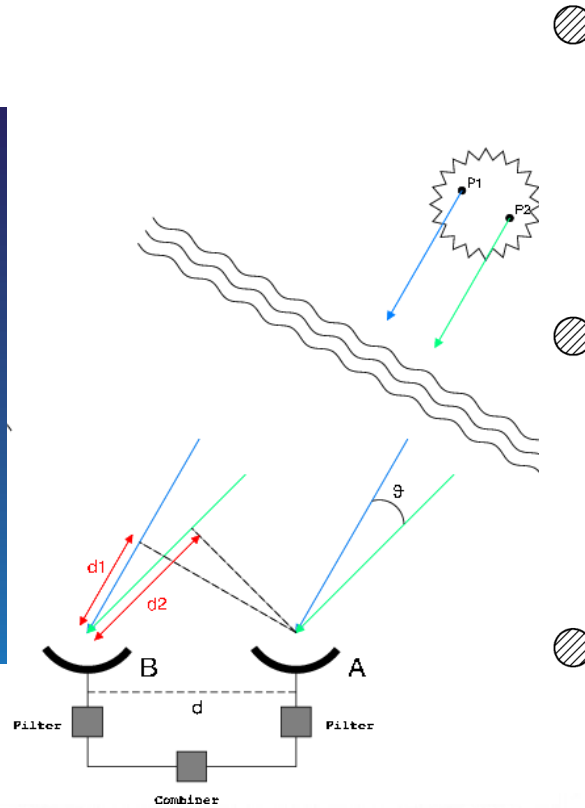
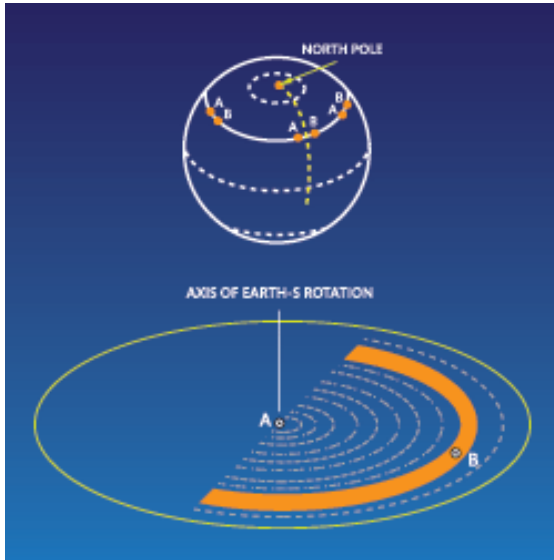
Interferometry

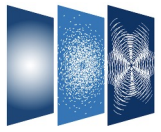




Interferometry Basics

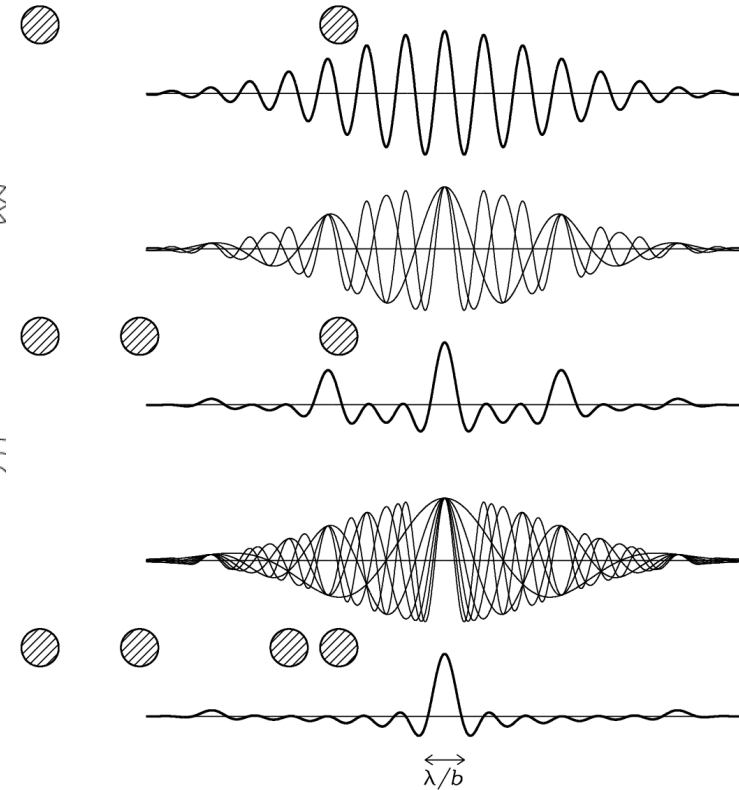
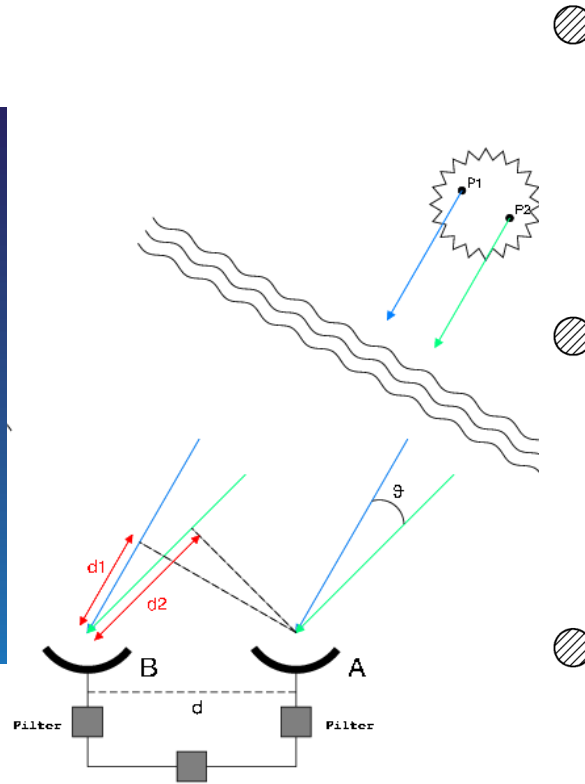
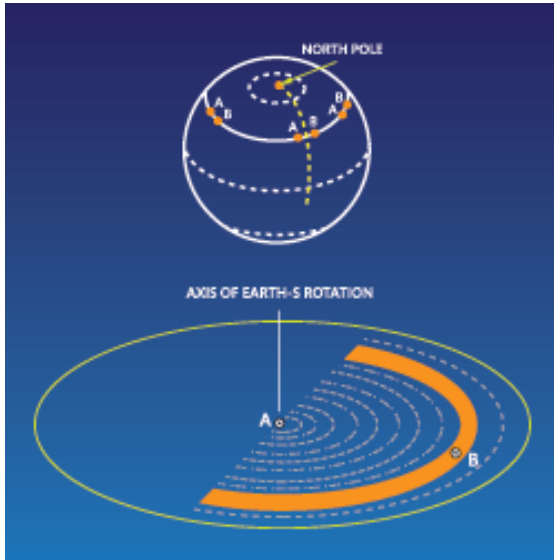
Interferometry





Interferometry Basics

Interferometry

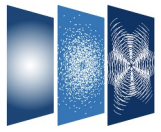


“Visibility”: interferometer response per antenna pair (i, j) , Δt , Δf , polarization

→ Fourier Transform of the sky brightness distribution.

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

→ Source *brightness*, *structure*



Interferometry Basics

CASA software: raw “visibility” data → science products

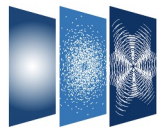
- Large data sets!
Example: $N(N-1)$ baselines, $\Delta t \sim 3s$ over 12h, 2000 Δf channels, 4 polarizations,
 - Discrete sampling (‘broken mirror’) → missing information!
Iteratively reconstruct model of sky
- **Computationally expensive**
- **Complex, advanced algorithms**

“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*



The CASA Software

Casacore:

original AIPS++ libraries, stable and nearly static platform, scripting & pipelines

CASA:

- **Tools:** basic C++ functions linked to Python interface → **basic operations**
- **Tasks:** bundle tools + Python functionality → **specific data reduction step**
→ *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

Manual, scripting & pipelines

→ ALMA calibration & imaging, VLA calibration, VLA Sky Survey,
Nobeyama 45-m single-dish, but also external (e.g., rPicard).



JIVE
Joint Institute for VLBI ERIC



Import/export

Information

Manipulation

Calibration

Imaging

Analysis

Single Dish

Simulations

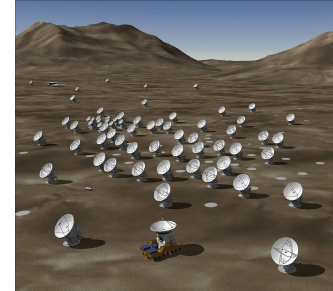


Import/export
 Information
 Manipulation
 Calibration
 Imaging
 Analysis

 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.)



For VLBI: importfitsidi!



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.



Import/export
 Information
 Manipulation
 Calibration
 Imaging
 Analysis
 Single Dish
 Simulations

Log Messages (~/Users/bemonts/casa-20220125-211413.log)

Search Message: Filter: Time

```

...obs:::casa #####
...obs:::casa ##### Begin Task: listobs #####
...obs:::casa listobs( vis='lowres_data1.ms/', selectdata=True, spw='', field='', antenna='', uvrange='', timerange='', correlation='', scan='
...ms::summary
...s::summary+
...s::summary+ MeasurementSet Name: /Users/bemonts/Documents/CASA_testing/Pcheck/lowres_data1.ms MS Version 2
...s::summary+
...s::summary+ Observer: debreuck Project: uid://A001/X10e/X46b
...s::summary+ Observation: ALMA
...Properties
...Computing scan and subscan properties...
...ms::summary Data records: 62532 Total elapsed time = 547.776 seconds
...s::summary+ Observed from 21-Jul-2014/11:29:55.7 to 21-Jul-2014/11:39:03.5 (UTC)
...ms::summary
...s::summary+ ObservationID = 0 ArrayID = 0
...s::summary+ Date Timerange (UTC) Scan FldId FieldName nRows SpwIds Average Interval(s) ScanIntent
...s::summary+ 21-Jul-2014/11:29:55.7 - 11:31:55.2 12 0 MRC_0152-209 37929 [0,1,2,3] [6.05, 6.05, 6.05, 6.05] [OBSERVE_T
...s::summary+ 11:37:41.8 - 11:39:03.5 16 0 MRC_0152-209 24603 [0,1,2,3] [6.05, 6.05, 6.05, 6.05] [OBSERVE_T
...ms::summary (nRows = Total number of rows per scan)
...s::summary+
...s::summary+ Fields: 1
...s::summary+ ID Code Name RA Decl Epoch SrcId nRows
...s::summary+ 0 none MRC_0152-209 01:54:55.760000 -20.40.26.30000 J2000 0 62532
...ms::summary Spectral Windows: (4 unique spectral windows and 1 unique polarization setups)
...s::summary+ SpwID Name #Chans Frame Ch0(MHz) ChanWid(kHz) TotBW(kHz) CtrFreq(MHz) BBC Num Corrs
...s::summary+ 0 ALMA_RB_06#BB_1#SW-01#FULL_RES 480 TOPO 237696.437 -3906.250 1875000.0 236760.8901 1 XX YY
...s::summary+ 1 ALMA_RB_06#BB_2#SW-01#FULL_RES 480 TOPO 239571.523 -3906.250 1875000.0 238635.9764 2 XX YY
...s::summary+ 2 ALMA_RB_06#BB_3#SW-01#FULL_RES 480 TOPO 250080.999 3906.250 1875000.0 251016.5458 3 XX YY
...s::summary+ 3 ALMA_RB_06#BB_4#SW-01#FULL_RES 480 TOPO 253076.137 3906.250 1875000.0 254011.6836 4 XX YY
...ms::summary Sources: 4
...s::summary+ ID Name SpwID RestFreq(MHz) SysVel (km/s)
...s::summary+ 0 MRC_0152-209 0 237000 0
...s::summary+ 1 MRC_0152-209 1 238875 0
...s::summary+ 2 MRC_0152-209 2 251000 0
...s::summary+ 3 MRC_0152-209 3 254000 0
...ms::summary Antennas: 33:
...s::summary+ ID Name Station Diam. Long. Lat. Offset from array center (m) ITRF Geocentric coo
...s::summary+ East North Elevation x
...s::summary+ 0 DA41 A079 12.0 m -067.45.13.6 -22.53.35.0 116.8369 -920.2899 22.6287 2225122.700415 -5439951.13
...s::summary+ 1 DA42 A081 12.0 m -067.45.23.9 -22.53.32.5 -174.5620 -842.8380 21.0898 2224863.872903 -5440088.01
...s::summary+ 2 DA43 A091 12.0 m -067.45.28.7 -22.53.24.2 -312.9125 -584.7729 23.7301 2224774.741615 -5440235.54
...s::summary+ 3 DA46 A058 12.0 m -067.45.17.3 -22.53.32.0 12.7399 -827.0340 21.9685 2225039.860155 -5440023.55
...s::summary+ 4 DA47 A074 12.0 m -067.45.12.1 -22.53.32.0 161.8145 -828.6214 19.2707 2225176.656953 -5439964.24
...s::summary+ 5 DA50 A045 12.0 m -067.45.17.9 -22.53.30.1 -5.4185 -767.4402 22.6042 2225032.051670 -5440052.42
...s::summary+ 7 DA55 A080 12.0 m -067.45.14.7 -22.53.20.2 87.4828 -461.2368 21.1333 2225162.611933 -5440126.24
...s::summary+ 8 DA59 A021 12.0 m -067.45.17.2 -22.53.27.0 14.3185 -672.8120 21.8445 2225063.988755 -5440078.37
  
```

Insert Message: Lock scroll

Import/export
 Information
 Manipulation
 Calibration
 Imaging
 Analysis
 Single Dish
 Simulations




Bug in CASA 6.5.5:
 header does not display

Log Messages (/Users/bemonts/casa-20220125-211413.log)

Search Message: Filter: Time

```

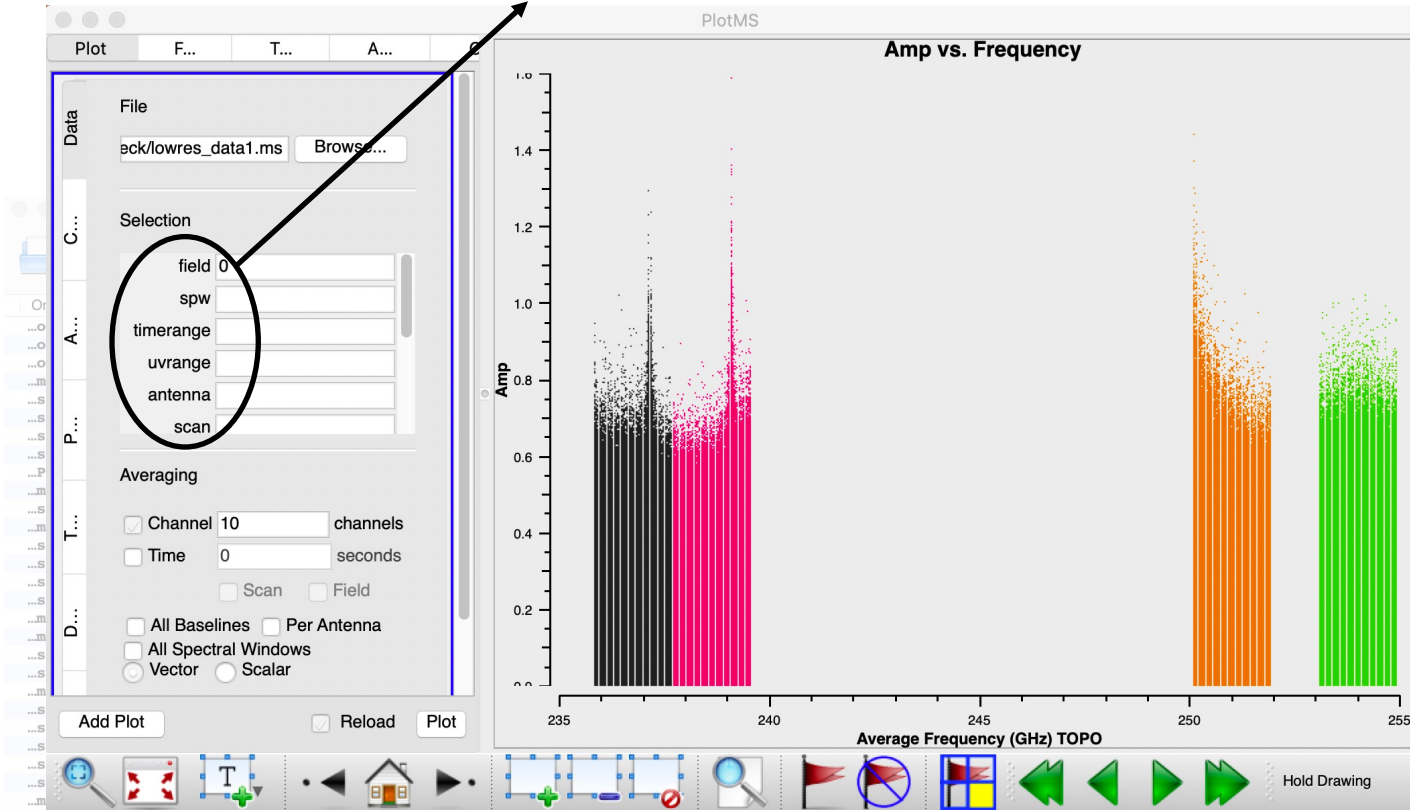
...obs:::casa #####
...obs:::casa ##### Begin Task: listobs #####
...obs:::casa listobs( vis='lowres_data1.ms/', selectdata=True, spw='', field='', antenna='', uvrange='', timerange='', correlation='', scan='
...ms::summary
...s::summary+ MeasurementSet Name: /Users/bemonts/Documents/CASA_testing/Pcheck/lowres_data1.ms MS Version 2
...s::summary+
...s::summary+ Observer: debreuck Project: uid://A001/X10e/X46b
...s::summary+ Observation: ALMA
...Properties
...Computing scan and subscan properties...
...ms::summary Data records: 62532 Total elapsed time = 547.776 seconds
...s::summary+ Observed from 21-Jul-2014/11:29:55.7 to 21-Jul-2014/11:39:03.5 (UTC)
...ms::summary
...s::summary+ ObservationID = 0 ArrayID = 0
...s::summary+ Date Timerange (UTC) Scan FldId FieldName nRows SpwIds Average Interval(s) ScanIntent
...s::summary+ 21-Jul-2014/11:29:55.7 - 11:31:55.2 12 0 MRC_0152-209 37929 [0,1,2,3] [6.05, 6.05, 6.05, 6.05] [OBSERVE_T
...s::summary+ 11:37:41.8 - 11:39:03.5 16 0 MRC_0152-209 24603 [0,1,2,3] [6.05, 6.05, 6.05, 6.05] [OBSERVE_T
...ms::summary (nRows = Total number of rows per scan)
...s::summary+
...s::summary+ Fields: 1
...s::summary+ ID Code Name RA Decl Epoch SrcId nRows
...s::summary+ 0 none MRC_0152-209 01:54:55.760000 -20.40.26.30000 J2000 0 62532
...ms::summary Spectral Windows: (4 unique spectral windows and 1 unique polarization setups)
...s::summary+ SpwID Name #Chans Frame Ch0(MHz) ChanWid(kHz) TotBW(kHz) CtrFreq(MHz) BBC Num Corrs
...s::summary+ 0 ALMA_RB_06#BB_1#SW-01#FULL_RES 480 TOPO 237696.437 -3906.250 1875000.0 236760.8901 1 XX YY
...s::summary+ 1 ALMA_RB_06#BB_2#SW-01#FULL_RES 480 TOPO 239571.523 -3906.250 1875000.0 238635.9764 2 XX YY
...s::summary+ 2 ALMA_RB_06#BB_3#SW-01#FULL_RES 480 TOPO 250080.999 3906.250 1875000.0 251016.5458 3 XX YY
...s::summary+ 3 ALMA_RB_06#BB_4#SW-01#FULL_RES 480 TOPO 253076.137 3906.250 1875000.0 254011.6836 4 XX YY
...ms::summary Sources: 4
...s::summary+ ID Name SpwID RestFreq(MHz) SysVel (km/s)
...s::summary+ 0 MRC_0152-209 0 237000 0
...s::summary+ 1 MRC_0152-209 1 238875 0
...s::summary+ 2 MRC_0152-209 2 251000 0
...s::summary+ 3 MRC_0152-209 3 254000 0
...ms::summary Antennas: 33:
...s::summary+ ID Name Station Diam. Long. Lat. Offset from array center (m) ITRF Geocentric coo
...s::summary+ East North Elevation x
...s::summary+ 0 DA41 A079 12.0 m -067.45.13.6 -22.53.35.0 116.8369 -920.2899 22.6287 2225122.700415 -5439951.13
...s::summary+ 1 DA42 A081 12.0 m -067.45.23.9 -22.53.32.5 -174.5620 -842.8380 21.0898 2224863.872903 -5440088.01
...s::summary+ 2 DA43 A091 12.0 m -067.45.28.7 -22.53.24.2 -312.9125 -584.7729 23.7301 2224774.741615 -5440235.54
...s::summary+ 3 DA46 A058 12.0 m -067.45.17.3 -22.53.32.0 12.7399 -827.0340 21.9685 2225039.860155 -5440023.55
...s::summary+ 4 DA47 A074 12.0 m -067.45.12.1 -22.53.32.0 161.8145 -828.6214 19.2707 2225176.656953 -5439964.24
...s::summary+ 5 DA50 A045 12.0 m -067.45.17.9 -22.53.30.1 -5.4185 -767.4402 22.6042 2225032.051670 -5440052.42
...s::summary+ 7 DA55 A080 12.0 m -067.45.14.7 -22.53.20.2 87.4828 -461.2368 21.1333 2225162.611933 -5440126.24
...s::summary+ 8 DA59 A021 12.0 m -067.45.17.2 -22.53.27.0 14.3185 -672.8120 21.8445 2225063.988755 -5440078.37
  
```

Insert Message:    Lock scroll

Import/export
Information
 Manipulation
 Calibration
 Imaging
 Analysis

Single Dish
 Simulations

Efficiency: make selection of data



The screenshot shows the CASA PlotMS interface. On the left, a 'Data' panel is visible with a 'Selection' section containing a list of parameters: field 0, spw, timerange, uvrange, antenna, and scan. An arrow points from the text 'Efficiency: make selection of data' to this selection panel. The main plot area displays 'Amp vs. Frequency' with a scatter plot of amplitude versus average frequency (GHz TOPO). The plot shows several distinct vertical clusters of data points, colored in black, pink, orange, and green. Below the plot, there is a toolbar with various icons for zooming, panning, and other plot controls.

Summary	ID	Name	Spwid	Restfreq (MHz)	sysvel (km/s)
...:summary+	0	MRC_0152-209	0	237000	0
...:summary+	1	MRC_0152-209	1	238875	0
...:summary+	2	MRC_0152-209	2	251000	0
...:summary+	3	MRC_0152-209	3	254000	0

Summary	ID	Name	Station	Diam.	Long.	Lat.	Offset from array center (m)			ITRF Geocentric coo	
Summary	ID	Name	Station	Diam.	Long.	Lat.	East	North	Elevation	x	y
...:summary+	0	DA41	A079	12.0 m	-067.45.13.6	-22.53.35.0	116.8369	-920.2899	22.6287	2225122.700415	-5439951.13
...:summary+	1	DA42	A081	12.0 m	-067.45.23.9	-22.53.32.5	-174.5620	-842.8380	21.0898	2224863.872903	-5440088.01
...:summary+	2	DA43	A091	12.0 m	-067.45.28.7	-22.53.24.2	-312.9125	-584.7729	23.7301	2224774.741615	-5440235.54
...:summary+	3	DA46	A058	12.0 m	-067.45.17.3	-22.53.32.0	12.7399	-827.0340	21.9685	2225039.860155	-5440023.55
...:summary+	4	DA47	A074	12.0 m	-067.45.12.1	-22.53.32.0	161.8145	-828.6214	19.2707	2225176.656953	-5439964.24
...:summary+	5	DA50	A045	12.0 m	-067.45.17.9	-22.53.30.1	-5.4185	-767.4402	22.6042	2225032.051670	-5440052.42
...:summary+	7	DA55	A080	12.0 m	-067.45.14.7	-22.53.20.2	87.4828	-461.2368	21.1333	2225162.611933	-5440126.24
...:summary+	8	DA59	A021	12.0 m	-067.45.17.2	-22.53.27.0	14.3185	-672.8120	21.8445	2225063.988755	-5440078.37

Import/export

Information

Manipulation

Calibration

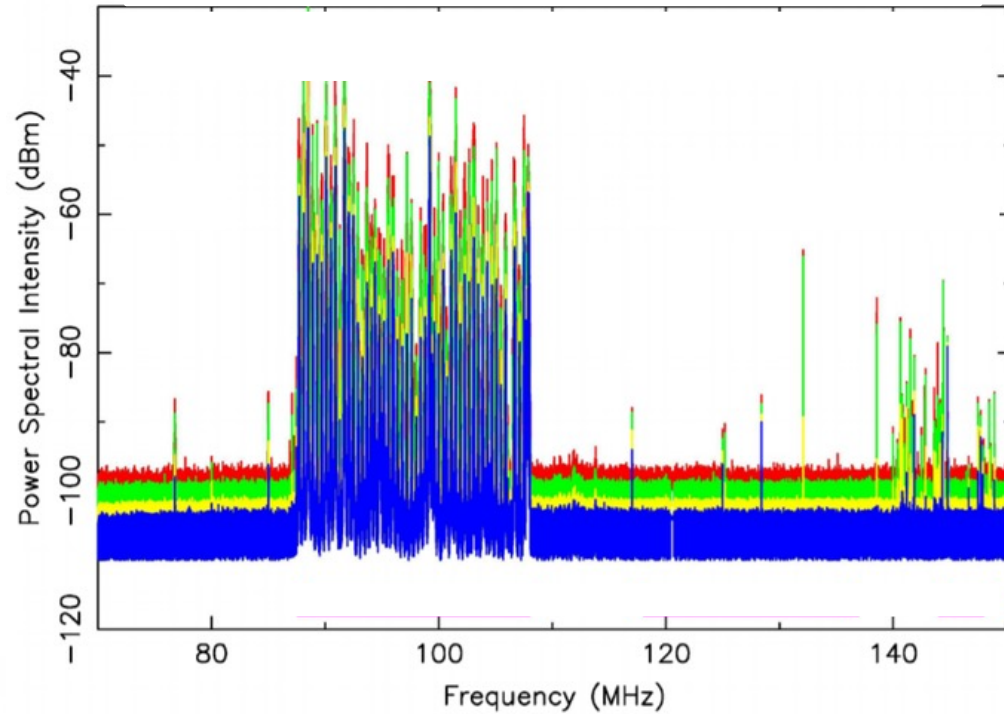
Imaging

Analysis

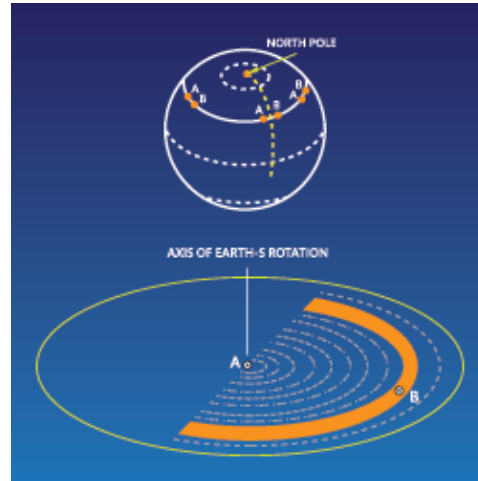
Single Dish

Simulations

Radio Frequency Interference (RFI) → Flag!



Import/export
 Information
 Manipulation
Calibration
 Imaging
 Analysis



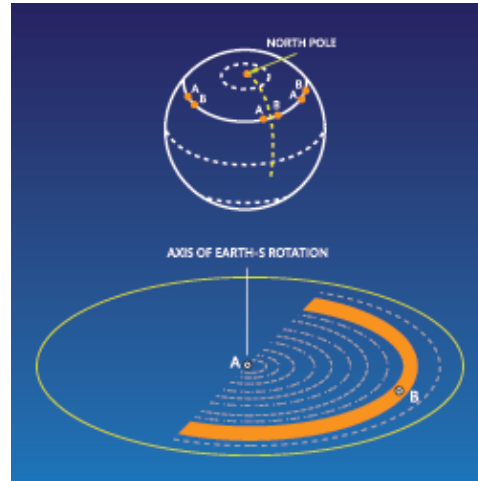
$$\vec{V}_{ij} = J_{ij} \vec{V}_{ij}^{IDEAL}$$

Single Dish
 Simulations



Import/export
 Information
 Manipulation
Calibration
 Imaging
 Analysis

 Single Dish
 Simulations



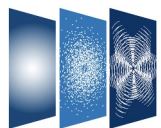
$$\vec{V}_{ij} = J_{ij} \vec{V}_{ij}^{IDEAL}$$

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

↓ Troposphere
 ↓ Parallax rotation
 ↓ Optical components tel.
 ↓ Leakage
 ↓ Time dependent gains
 ↓ Frequency dep. gains (bandpass)

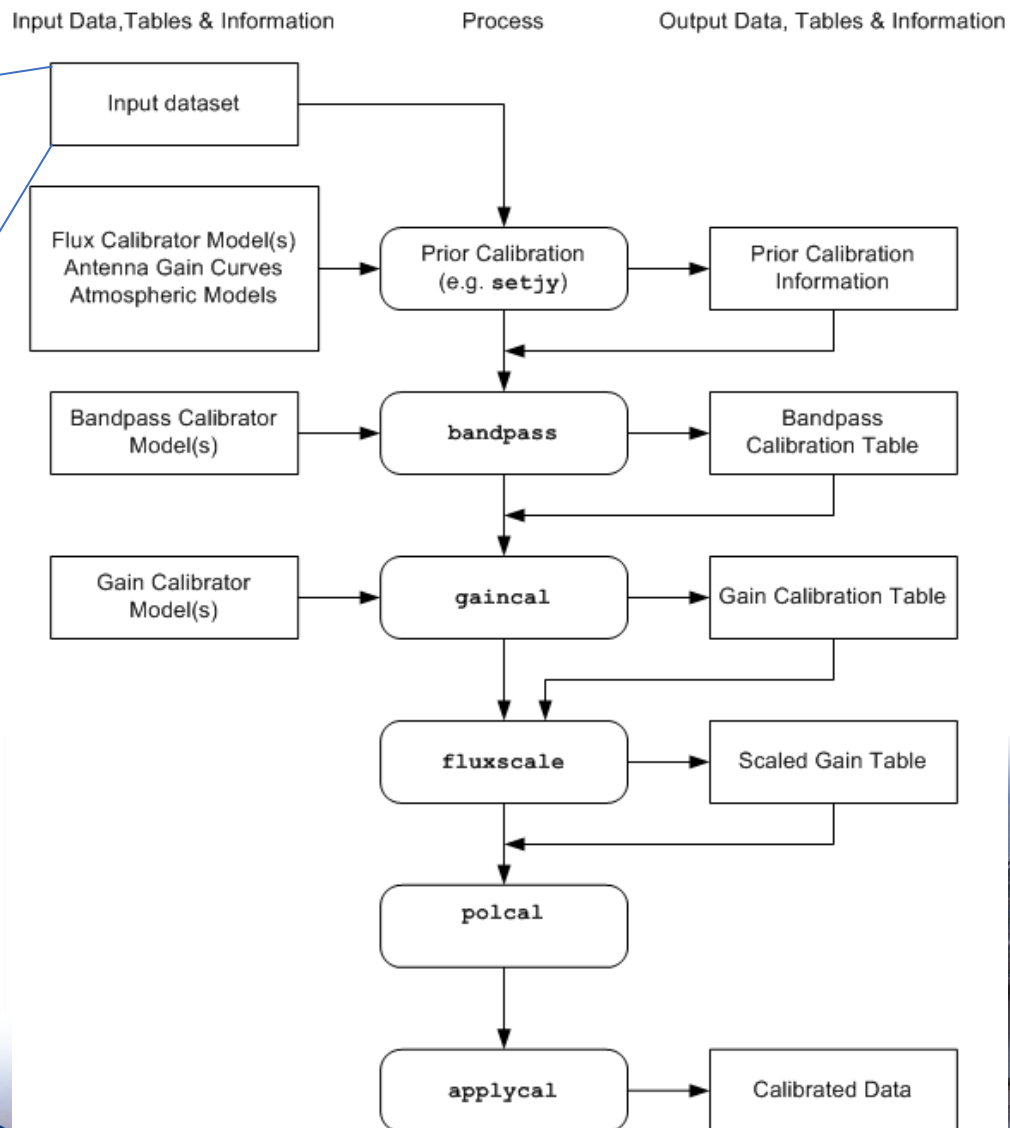
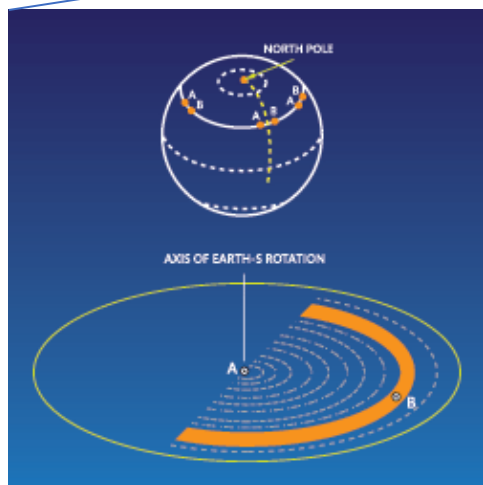
Baseline-based correlator (non-closing) errors

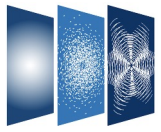




Calibration in CASA

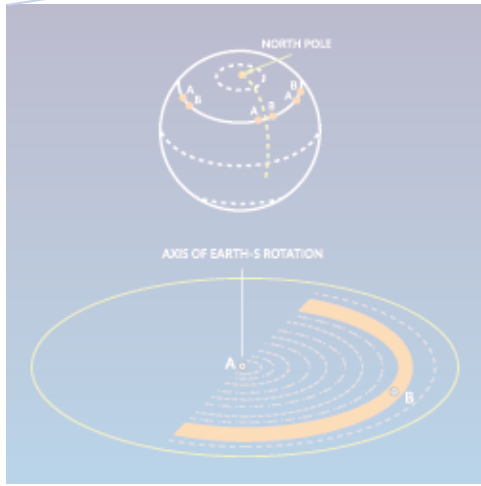
$$\vec{V}_{ij} = J_{ij} \vec{V}_{ij}^{IDEAL}$$



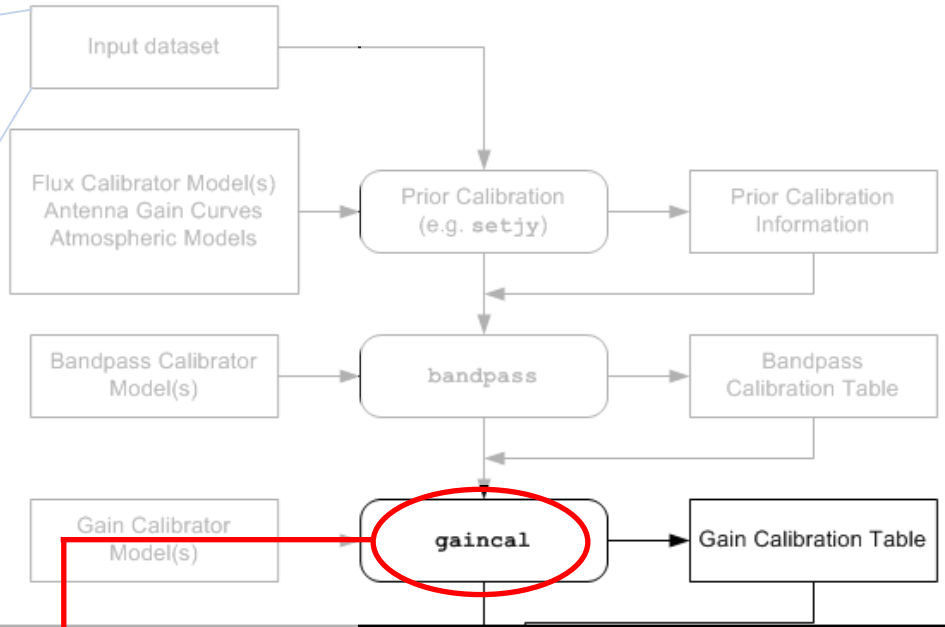


Calibration in CASA

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



Input Data, Tables & Information Process Output Data, Tables & Information



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

CASA task

Input Data

Spectral Window selection

Reference antenna

Phase-only Calibration mode



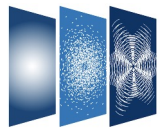
Calibration in CASA

\vec{V}_{ij}

```
[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)
corrdepflag = 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



Calibration in CASA

Import/export

Information

Manipulation

Calibration

Imaging

Analysis

Single Dish

Simulations

VLBI calibration in CASA

- Task **fringefit**

calculates fringe-fitting solutions
(phase/delay/rate/dispersive delay)

- Task **accor**

Normalization visibility amplitudes
based on auto-correlations

- Other tasks adapted for VLBI

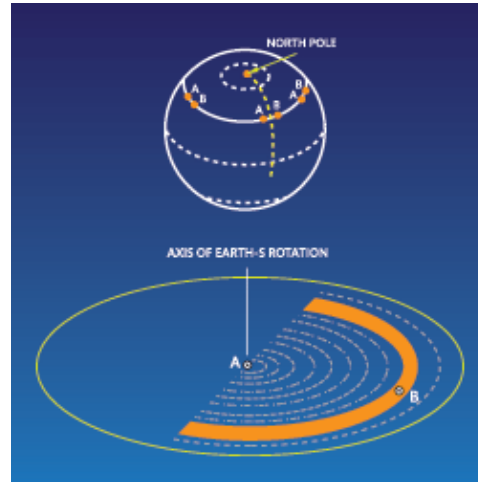
Talk to tutors!



JIVE
Joint Institute for VLBI ERIC

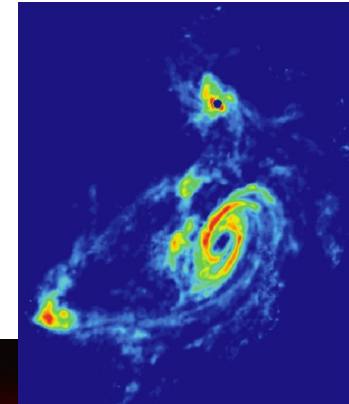
Import/export
 Information
 Manipulation
 Calibration
 Imaging
 Analysis

Single Dish
 Simulations

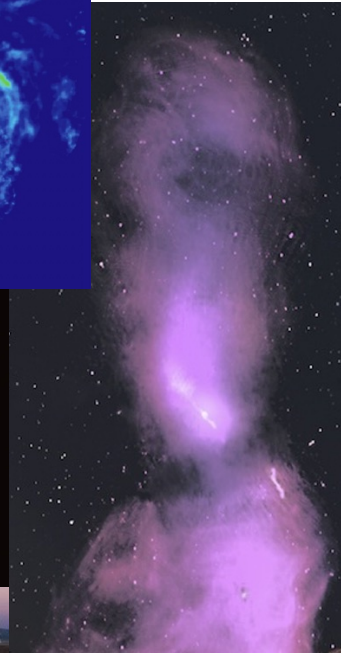
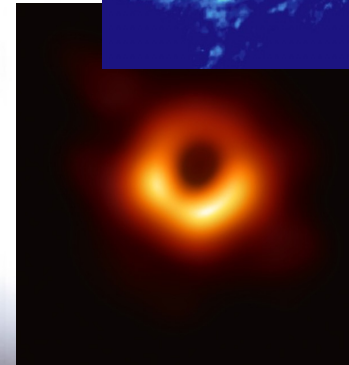


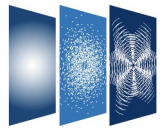
$\vec{V}_{ij} IDEAL$

- Gridding data
- Weighting data
- Fourier transform
- Deconvolution
- Restoration



2D continuum images
 3D image cubes





Imaging in CASA

Tclean: powerful imaging task

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



国立天文台

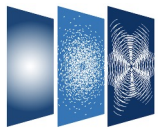


ASTRON



JIVE

Joint Institute for VLBI ERIC



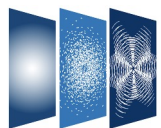
Imaging in CASA

Tclean: powerful imaging task

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

```
[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
gridding = '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 calcul
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>: []
```

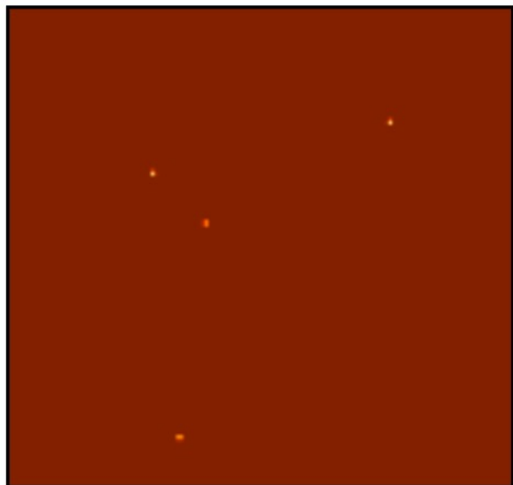


Imaging in CASA

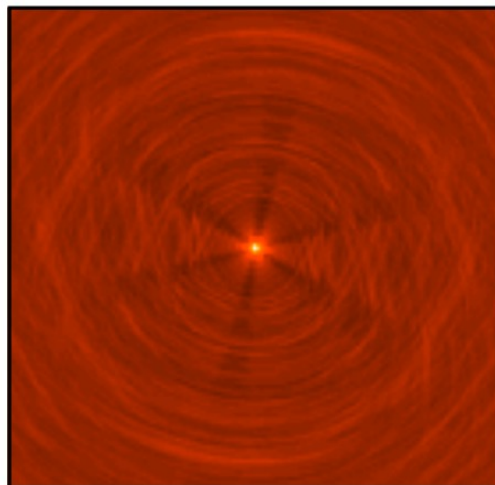
Deconvolution



Sky Brightness Distribution



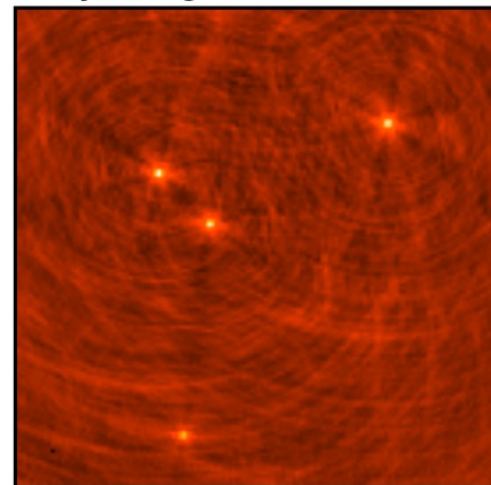
Point Spread Function



$*$

$=$

Dirty Image

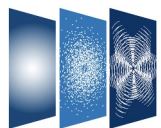


Convolution of sky brightness distribution and instrumental PSF results in 'dirty' image

Deconvolution in tclean

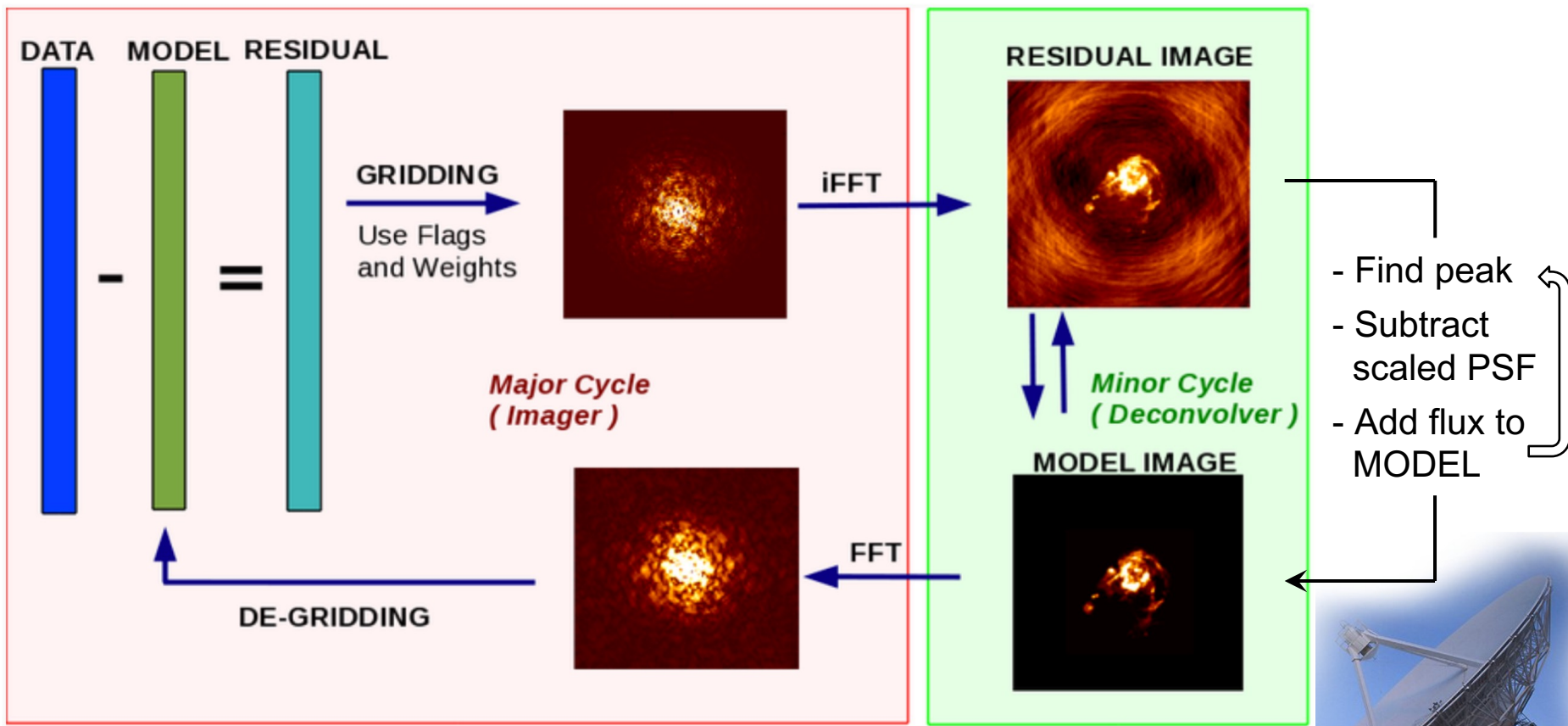
Reconstruct sky model by iteratively deconvolving the dirty image using the PSF





Imaging in CASA

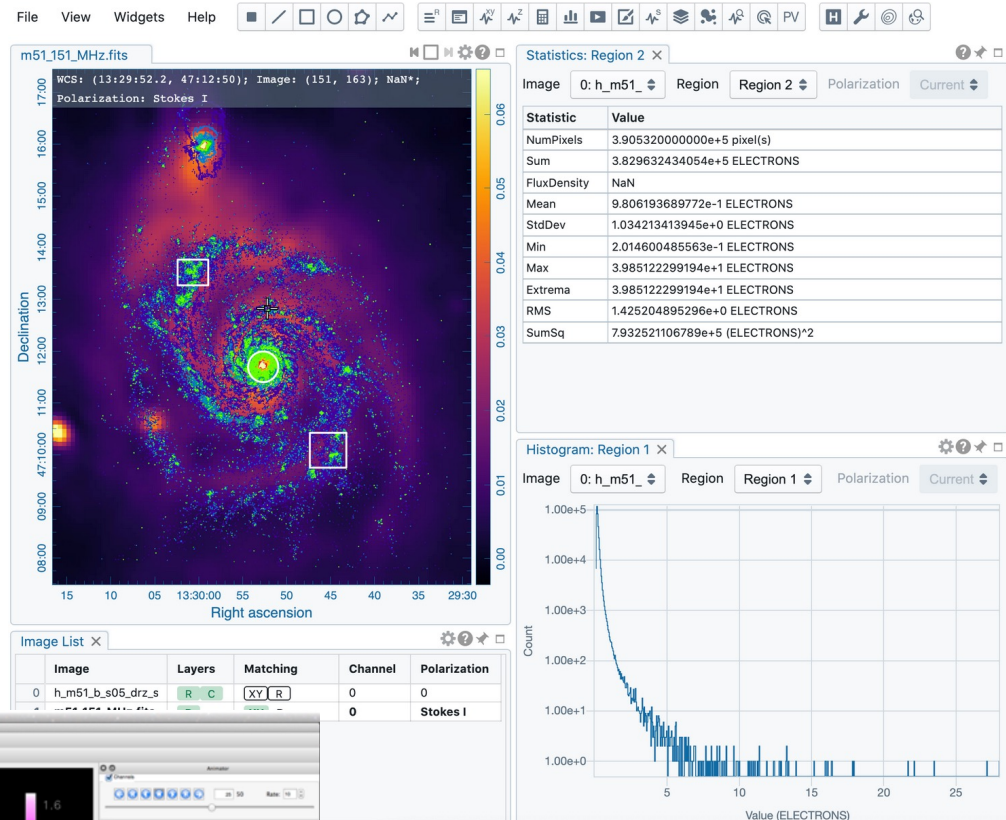
Tclean: deconvolution process



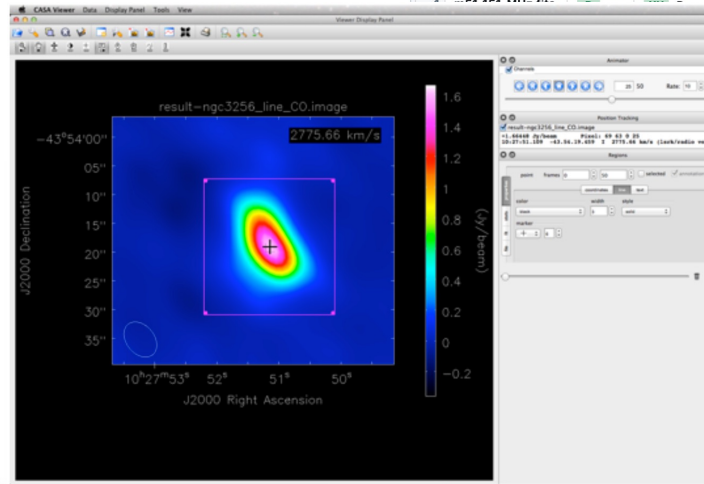
Import/export
 Information
 Manipulation
 Calibration
 Imaging
**Analysis/
 Visualization**

Single Dish
 Simulations

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



The old: CASA Viewer



<https://cartavis.org/>

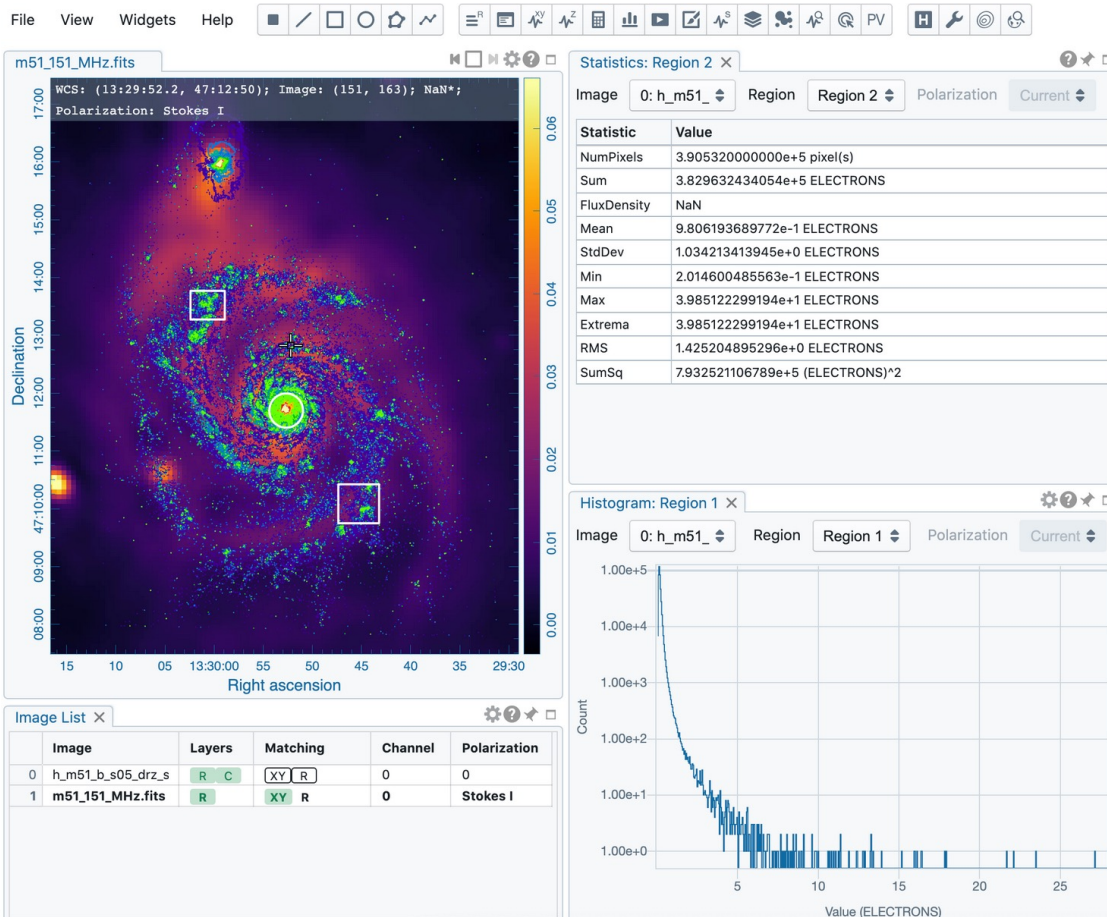




Visualization

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 (Canada)



Import/export

Information

Manipulation

Calibration

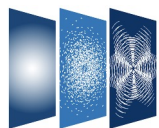
Imaging

Analysis/Visualization

Single Dish

Simulations





CASA download & installation



About Download Documentation Team Contact

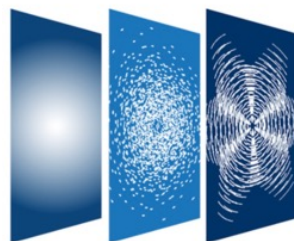
Website (casa.nrao.edu)

Pipelines (ALMA, VLA)

Monolithic (all-inclusive 'plug')

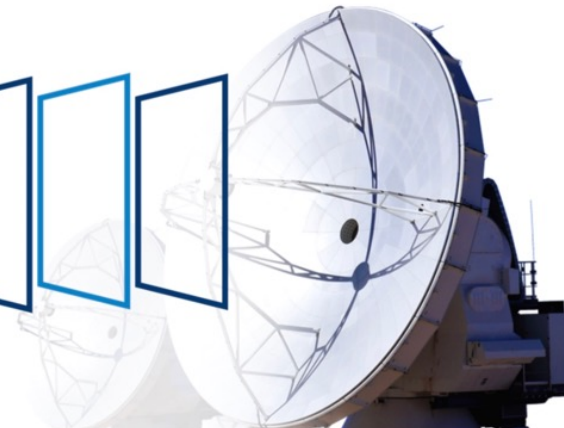
Pip-wheel (Pythonic, Jupyter, Google Colab)

Compatibility Operating



CASA

Common Astronomy Software Applications

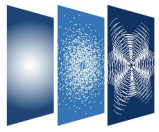


Latest CASA Release: 6.5*

- *CASA paper*: "CASA, the Common Astronomy Software Applications for Radio Astronomy", [PASP_134_114501](#)
- *Modular CASA*: separate python wheels for casafeather, casabrowser and casalogger GUIs.
- *deconvolve*: new task for image-domain deconvolution.
- *uvcontsub*: new implementation, old uvcontsub task deprecated.
- *setjy*: updated VLA flux calibrator model images at C, X and Ka bands; will catch unreasonable input spectral index.
- *fringeft*: support for 'uvrange' and 'corrcomb' parameters; new functionality 'concatspw' or 'combine=spw'.
- *tclean*: new iteration control parameter 'nmajor'.
- *tclean*: new parameter 'fullsummary' to avoid issues with large cubes and increased MPI records.
- *tclean*: more stable cube imaging with 'awproject'; numerical fixes w-term correction 'awproject'.
- *sdimaging*: new parameter 'enablecache' for improved performance.
- *applanal*: per scan interpolation.
- *mstransform*: parameter 'douvcontsub' deprecated.
- *flagdata*: mode='shadow' now uses the uvw values from the UVW column.
- *tclean/tsdimaging*: performance improvements of about 10-16 percent, and improved runtime performance of ephemeris imaging.
- *tclean*: corrections to the math implemented for the uvtaper weighting scheme.
- *simulator tool*: new parameter 'simint' in sm.settrop() to control time granularity, down to 0.1s.
- *simulator tool*: now works with primary beams and a component list with spectral structure.
- *imbaseline*: new task for image-based baseline subtraction for single-dish data.
- *msmetadata tool*: ALMA-specific methods 'rxband()' and 'subwindows()'.
- *corbitt*: new method in msmetadata tool for returning the value in SPECTRAL_WINDOW::SDM_CORR_BIT column.
- *plotms*: added support for additional axes of calibration tables.
- *setjy*: parameter "modimage" removed (use parameter "model" instead).
- *tclean*: return dictionary now includes additional information about the minor cycles.
- *Mac OS 12*: bug fixed that prevented the OS 11/Python 3.6 package to open on Mac OS 12.
- *tclean*: bug fixed that prevented UV taper to work with weighting='natural'.

For details and more, see [CASA Docs](#)





CASA download & installation

Website (casa.nrao.edu)

Pipelines (ALMA, VLA)

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

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

Compatibility Operating Systems

New release every ~2 months!

Latest version: CASA 6.4

The [Release Notes](#) and [Known Issues](#) of the 6.4 release are available

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)

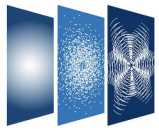
The above CASA versions can also be downloaded from our [NAOJ CASA mirror site](#) and [NAOJ CASA-pipeline mirror site](#), or via [Google Drive](#).

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)

Pipelines (ALMA, VLA)

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

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



Compatibility Operating Systems

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)

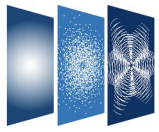
 The above CASA versions can also be downloaded from our [NAOJ CASA mirror site](#) and [NAOJ CASA-pipeline mirror site](#), or via [Google Drive](#).

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)

Pipelines (ALMA, VLA)

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

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



Compatibility Operating Systems

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)

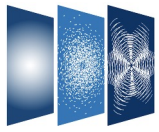
 The above CASA versions can also be downloaded from our [NAOJ CASA mirror site](#) and [NAOJ CASA-pipeline mirror site](#), or via [Google Drive](#).

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 (ca

Pipelines (

Monolithic

Pip-wheel

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 MSttransform. 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.

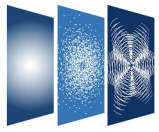
```
[ ]: from casatasks import mstransform

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 MSttransformManager::parseMsSpecParams Tile shape is [0]
2021-10-14 17:43:24 INFO MSttransformManager::parseChanAvgParams Channel average is activated
2021-10-14 17:43:24 INFO MSttransformManager::parseChanAvgParams Channel bin is [3]
2021-10-14 17:43:24 INFO MSttransformManager::colCheckInfo Adding DATA column to output MS from input DATA column
2021-10-14 17:43:24 INFO MSttransformManager::open Select data
2021-10-14 17:43:24 INFO MSttransformManager::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 #####
```



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



CASA download

Website (casa.nrao.edu)

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

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

Pipelines (ALMA, VLA)

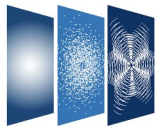
Compatibility Operating Systems

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 6.5.5

CASA 6.5.5-21: Released 14 April 2023

No pipeline included!

VLBI highlights:

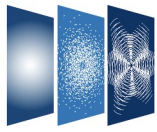
- **fringefit:** allows combined solving correlations via *corrcomb* parameter. (CASA 6.5.5)
- **fringefit:** new functionality with *concatspws* or *combine='spw'*. (CASA 6.5.5)
- **applycal** (and *apply* by solve tasks): now have per-scan interpolation. (CASA 6.5.5)
- **fringefit:** now supports the *uvrange* parameter. (CASA 6.5.2)

Talk to JIVE staff for details!



JIVE
Joint Institute for VLBI ERIC





CASA Docs (casadocs.readthedocs.io)



- 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
- Citing CASA
- Change Log

» 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.

6.5.5 Release

CASA 6.5.5 can now be [downloaded](#) for general use. CASA 6.5.5 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:

- **fringefit**: allows combined solving of correlations via the `corrcomb` parameter.
- **fringefit**: new functionality with `concatspws` or `combine='spw'`.
- **tclean**: enabled more stable cube imaging with the `awproject` gridded.
- **plotms**: exports text data with more sufficient precision.
- **setjy**: will catch an unreasonable input spectral index value.
- **msmetadata** tool: includes ALMA-specific methods `rxbands()` and `subwindows()`.
- **applycal**: now has per-scan interpolation.

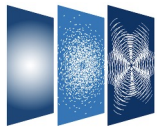
In addition, a number of bugs were fixed, including (but not limited to):

- **tclean**: numerical fixes with the w-term correction within `awproject`.
- **tclean**: not recognizing the observatory name.
- **gencal**: not always taking antenna position offsets properly into account.
- **sdfit/importasap**: invalid memory access.
- an MPI issue with Ubuntu.

Casacore was also updated from Aug 2022 - Mar 2023.

For more details on these and other new features, see the [CASA 6.5.5 Release Notes](#).

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 Docs (casadocs.readthedocs.io)



Search docs

Release Information

Index

API

Task List

Using CASA

CASA Fundamentals

External Data

Calibration & Visibilities

Imaging & Analysis

CARTA

Read the Docs v: stable

Versions

latest **stable** v6.5.5 v6.5.3 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.

6.5.5 Release

CASA 6.5.5 can now be [downloaded](#) for general use. CASA 6.5.5 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:

- fringefit: allows combined solving of correlations via the corrcomb parameter.
- fringefit: new functionality with concatspws or combine='spw'.
- tclean: enabled more stable cube imaging with the awproject gridded
- plotms: exports text data with more sufficient precision.
- setjy: will catch an unreasonable input spectral index value.
- msmetadata tool: includes ALMA-specific methods rxbands() and subwindows().
- applycal: now has per-scan interpolation.

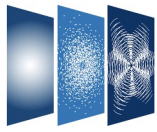
In addition, a number of bugs were fixed, including (but not limited to):

- tclean: numerical fixes with the w-term correction within awproject.
- tclean: not recognizing the observatory name.
- gencale: not always taking antenna position offsets properly into account.
- sdfit/importasap: invalid memory access.
- an MPI issue with Ubuntu.

Casacore was also updated from Aug 2022 - Mar 2023.

For more details on these and other new features, see the [CASA 6.5.5 Release Notes](#).

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 Docs (casadocs.readthedocs.io)



Search docs

Release Information

Highlights

Release Notes

Known Issues

Compatibility

Installation

Index

API

Task List

Using CASA

Read the Docs v: v6.5.0

Versions

latest	stable	v6.5.5	v6.5.3	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

Release Information

These are the release notes for CASA 6.5. Changes compared to the [CASA 6.4 release](#) are listed below.

Highlights

- **imbaseline**: a new task `imbaseline` was added for image-based baseline subtraction for single-dish data.
- **corrbit()**: a new method `corrbit()` was implemented in the `msmetadata` tool for returning the value in `SPECTRAL_WINDOW::SDM_CORR_BIT` column
- **setjy**: the parameter `modimage` was removed and parameter `model` should be used instead.
- **plotms**: support was added for additional axes of calibration tables.
- **tclean**: the return dictionary now includes additional information about the minor cycles.
- **Mac OS 12**: a bug was fixed which prevented the Mac OS 11 / Python 3.6 package to open on Mac OS 12.

Release Notes

Installation and operation

- **OS Support**: CASA is expected to be compatible with a range of Operating Systems for RedHat, Ubuntu and Mac OS. For information, please see CASA's [compatibility matrix](#)

Import/export

- **msmetadata tool**: a new method `corrbit()` was implemented to return the value in `SPECTRAL_WINDOW::SDM_CORR_BIT` column for the specified `spw(s)`, or a list representing these values for all spectral windows if `spw < 0`.

Calibration

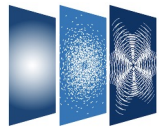
- **setjy**: the deprecated parameter `'modimage'` has been removed from the `setjy` task; the parameter `'model'` should be used instead.

Imaging

- **tclean**: the return dictionary now includes additional information about the minor cycles.

Analysis

- **imbaseline**: a new task `imbaseline` was added to do image-based baseline subtraction for single-dish data. Task `imbaseline` does, if necessary, smoothing of the spatial/spectral plane of an input image cube to improve S/N. then subtracts spectral baselines from it.



CASA Docs (casadocs.readthedocs.io)



Search docs

Release Information

Highlights

Release Notes

Known Issues

General

Installation

Scripting

statwt

mstransform

cvel

bandpass

polcal

setjy

uvcontsub

cal library

VLA Switched Power

fringeft

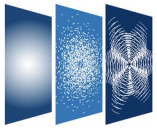
fixvis

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.
- **stwt** 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 **fringeft**, 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 gridded 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 **gridded='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)

- [-] Release Information
 - Highlights
 - Release Notes
- [-] Known Issues
 - Compatibility
 - Automated testing
- [-] Installation**
 - Prerequisite OS Libraries
 - Monolithic Distribution
 - Modular Packages**
- [-] Performance

- Index
- API
- Task List
- Using CASA
- CASA Fundamentals
- External Data
- Calibration & Visibilities
- Imaging & Analysis
- CARTA
- Pipeline
- Simulations
- Parallel Processing
- Memo Series & Knowledgebase
- Community Examples
- Citing CASA
- Change Log

Read the Docs v: stable

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 myenv
$: source myenv/bin/activate
(myenv) $: pip install --upgrade pip wheel
```

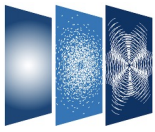
Now pick whichever subset of the available CASA packages you are interested in. Package dependencies are handled automatically by pip, with the exception of `casadata` which must be explicitly installed and updated by the user (see [External Data](#)). The following packages are available:

```
(myenv) $: pip install casatools==6.5.5.21
(myenv) $: pip install casatasks==6.5.5.21
(myenv) $: pip install casaplotms==2.0.1
(myenv) $: pip install casaviewer==1.7.1
(myenv) $: pip install casampi==0.5.3
(myenv) $: pip install casashell==6.5.5.21
(myenv) $: pip install casadata==2023.4.10
(myenv) $: pip install casaplotserver==1.5.1
(myenv) $: pip install almatasks==1.6.1
(myenv) $: pip install casatestutils==6.5.5.21
(myenv) $: pip install casatablebrowser==0.0.31
(myenv) $: pip install casalogger==1.0.15
(myenv) $: pip install casafeather==0.0.18
(myenv) $: pip install casatelemetry==1.3.4
```

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:

```
(myenv) $: 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



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

option of the

et to

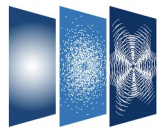
1: cannot n

install apps

vacancy ->

General and other applications submitted from this appstore and reviewed by developers.

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



CASA Docs (casadocs.readthedocs.io)



Search docs

Release Information

Index

API

- almataasks
- casadata
- casalith
- casaplotms
- casashell
- casataasks ←
- casatools
- casaviewer
- configuration

Task List

Using CASA

CASA Fundamentals

External Data

Calibration & Visibilities

Imaging & Analysis

CARTA

Pipeline

» API

[Edit on GitHub](#)

API

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

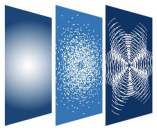
- [almataasks](#)
- [casadata](#)
- [casalith](#)
- [casaplotms](#)
- [casashell](#)
- [casataasks](#)
- [casatools](#)
- [casaviewer](#)
- [configuration](#)

← Previous

Next →

© Copyright 2021, Associated Universities, Inc Revision 2316c9b7.

Built with [Sphinx](#) using a [theme](#) provided by [Read the Docs](#).



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

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

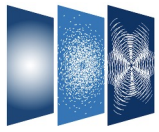
[\[Description\]](#) [\[Examples\]](#) [\[Development\]](#) [\[Details\]](#)

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)

Release Information

- Highlights
- Release Notes
- Known Issues
- Compatibility
- Automated testing

Installation

Performance

- casabench

Index

API

Task List

Using CASA

CASA Fundamentals

External Data

Calibration & Visibilities

Read the Docs

v: stable

Performance

CASA is now running performance benchmarks against a subset of the [casatasks](#) API to track various runtime metrics over development history of the project, starting in CASA 6. See the [Performance Benchmark](#) webpages for an interactive view of the latest test results.

casabench

The tests are implemented using [airspeed-velocity](#) with configuration and results tracked by a separate [repository](#). Automated deployment is coordinated with Bamboo and confined to a single computer node.

The dedicated testing machine has eight Intel(R) Xeon(R) CPUs (E5-2670 @ 2.60GHz), 264GB of DDR3 SDRAM (36KSF2G72PZ-1G6P1 @ 1600 MT/s), 500GB Crucial(R) internal SSD (MX500), and runs Red Hat Enterprise Linux Workstation release 7.9 against the Linux kernel version 3.10.0-1160.71.1.el7.x86_64.

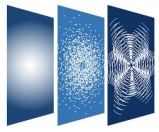
Tests currently run for each CASA6 pre-release package once other verification steps are complete. New tests and test cases will be added as development continues in CASA.

Previous

Next

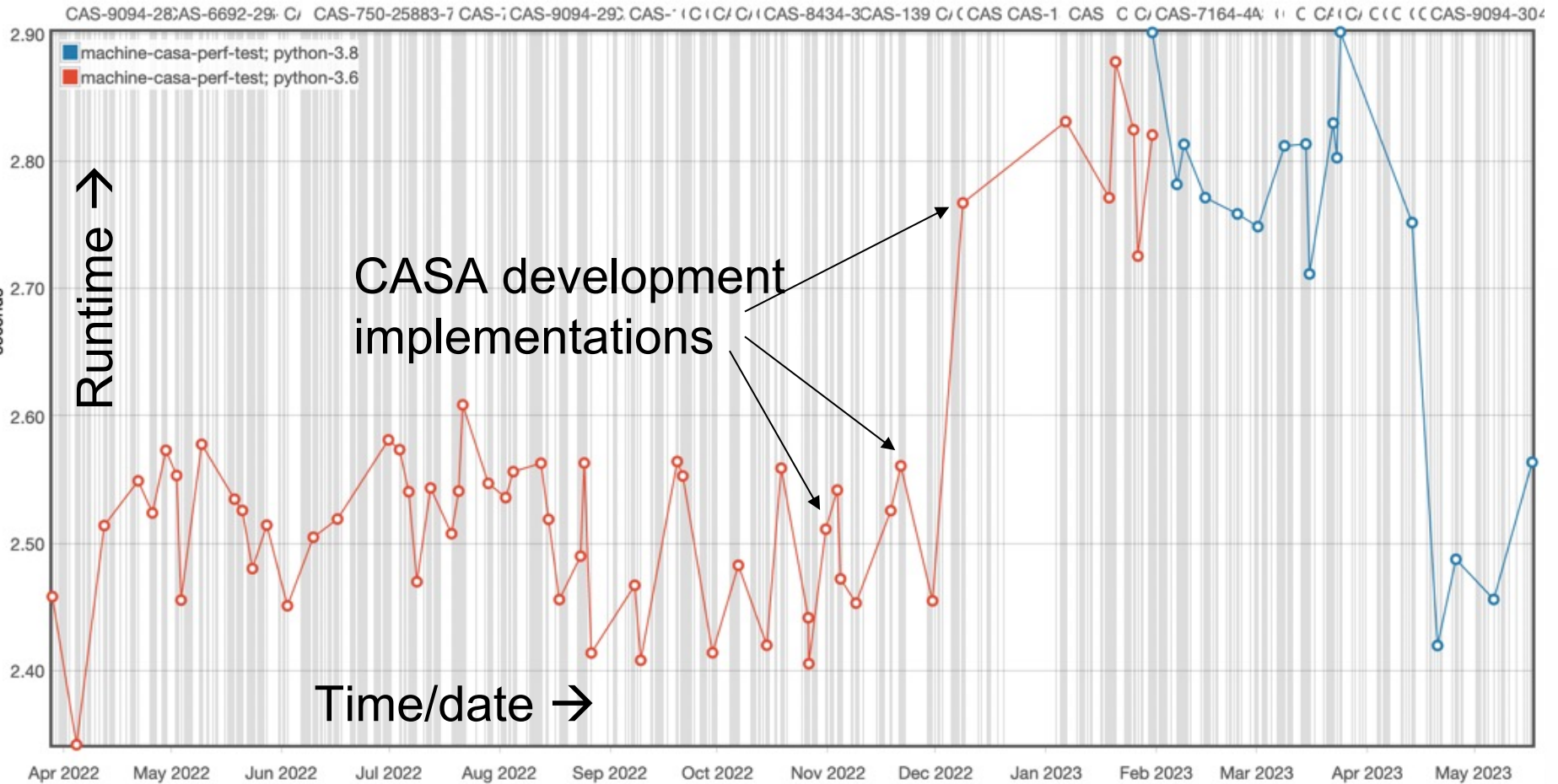
© Copyright 2021, Associated Universities, Inc Revision 1232d59b.

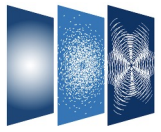
Built with [Sphinx](#) using a [theme](#) provided by [Read the Docs](#).



CASA Docs (casadocs.readthedocs.io)

CASA benchmarking





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

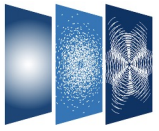
CASA 6.2 Notebook Demo

PlotMS with Jupyter Notebooks and X11

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 Docs (casadocs.readthedocs.io)

Modular CASA Demo

Installation

First the system must be configured with the appropriate pre-requisite libraries to create a virtual display necessary for later plots/images.

```
[ ]: # prerequisite system setup
import os
os.system('apt-get install xvfb')
os.system('pip install pyvirtualdisplay')

from pyvirtualdisplay import Display
display = Display(visible=0, size=(1024, 768))
display.start()

print('completed system setup')

completed system setup
```

Then we can choose from the available CASA packages to install:

```
casatools, casatasks, casaplotms, casaviewer, almatasks, casampi, casashell, casadata, casampi, casaplotserver
```

Running tclean

Tclean works in non-interactive mode only (interactive=False).

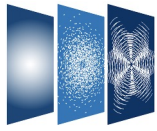
```
[ ]: from casatasks import tclean

print("running tclean, may take a bit...")

tclean(vis='sis14_twhya_calibrated_flagged.ms', imagename='first_image',
       field='5', spw='', specmode='mfs', deconvolver='hogbom', nterms=1,
       gridded='standard', imsize=[250,250], cell=['0.1arcsec'],
       weighting='natural', threshold='0mJy', niter=5000,
       interactive=False, savemodel='modelcolumn')

print("complete")

running tclean, may take a bit...
complete
```



CASA Reference Papers

CASA Team+ (2022), PASP, 134, 114501
van Bemmell+ (2022), PASP, 134, 114502

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 SEKSHAR,^{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 GLENDENNING² AND JEFF KERN⁴

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

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

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

⁴National Radio Astronomy Observatory, 520 Edgemont Road, Charlottesville, VA 22903

⁵National Astronomical Observatory, Tokyo, Japan

⁶Joint Institute for Data

⁷Inter-University Institute for Data

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}

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

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

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

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

⁵Dipartimento di Fisica, Università degli Studi di Cagliari, SP Monserrato-Sestu km 0.7, I-09042 Monserrato, Italy

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

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

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

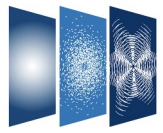
⁹Leiden Observatory, Leiden University, Postbus 2300, 9513 RA Leiden, The Netherlands

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

CASA, the Common Astronomy Software Applications for Radio Astronomy, is the software that runs on the Atacama Large Millimeter Array (ALMA), the Atacama Large Submillimeter Array (ALMA), and the Very Large Array (VLA), and is frequently used for processing VLBI data. CASA is the core of the VLBI software stack, and is frequently used for processing VLBI data. CASA is the core of the VLBI software stack, and is frequently used for processing VLBI data. CASA is the core of the VLBI software stack, and is frequently used for processing VLBI data.

ABSTRACT

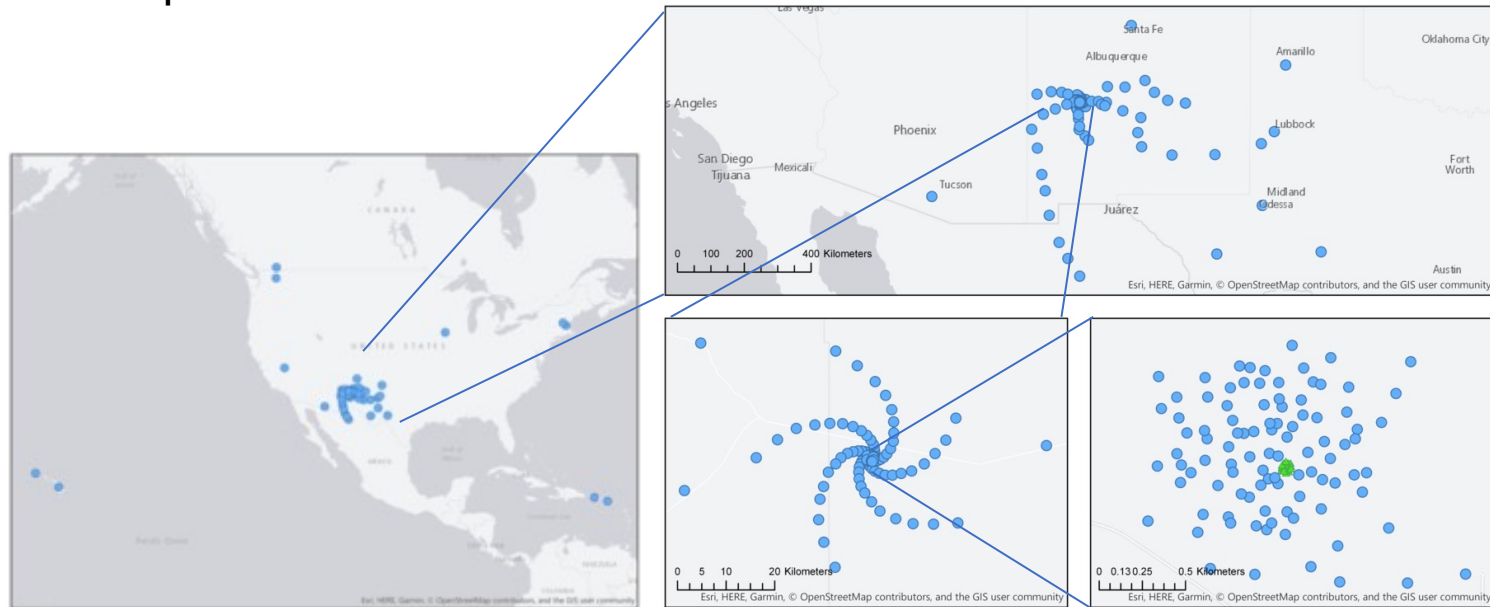
New functionality to process Very Long Baseline Interferometry (VLBI) data has been implemented in the CASA package. This includes two new tasks to handle fringe fitting and VLBI-specific amplitude calibration



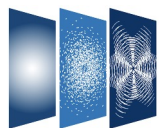
Future radio astronomy - NRAO/CASA

Next-generation VLA

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



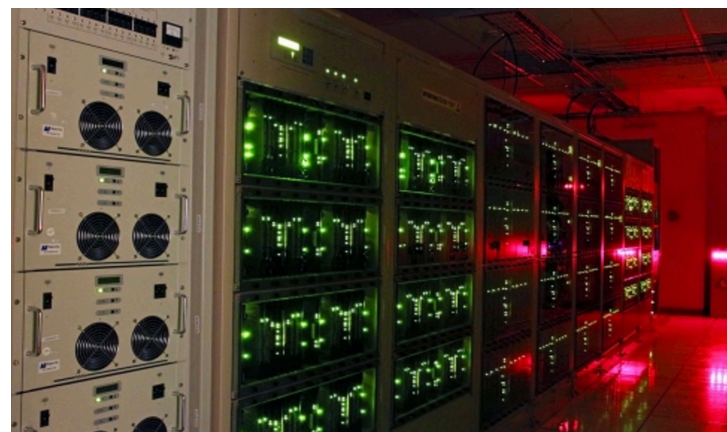
Murphy et al. 2019 – Science with a next-generation VLA
(*terrestrial planet formation, star formation, molecular gas across Universe*)



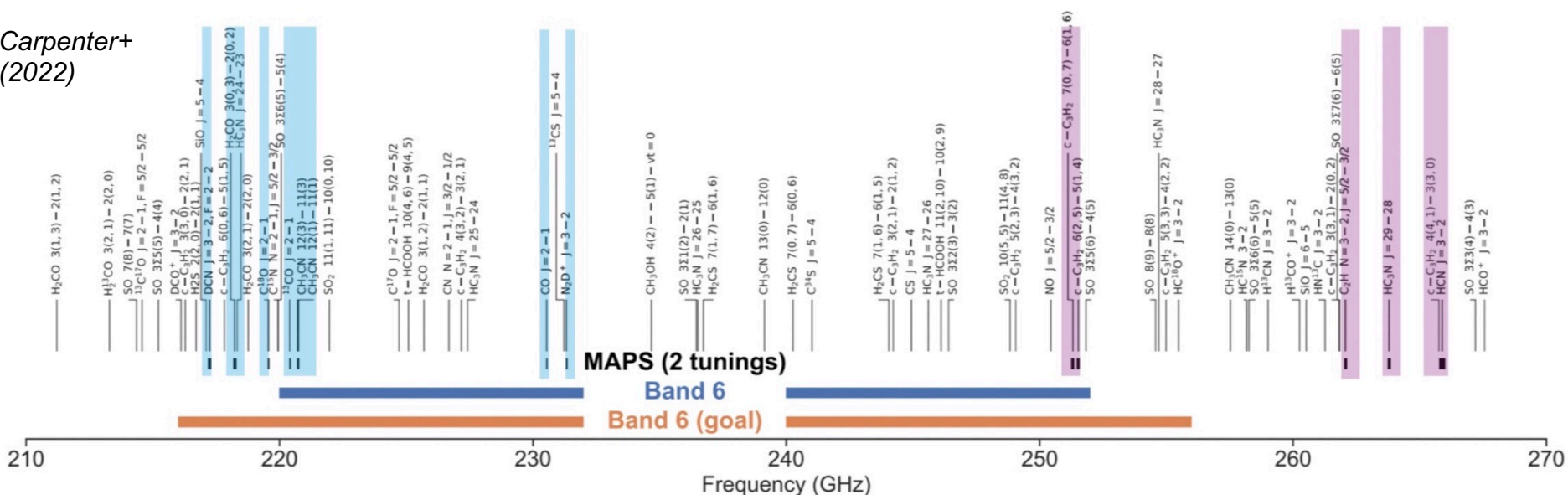
Future radio astronomy - NRAO/CASA

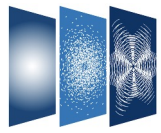
ALMA2030 Wideband Sensitivity Upgrade

- 2-4x current instantaneous bandwidth
- 1.2 million spectral channels (no trade-offs)
- Flexible sub-array
(12m and 7m simultaneous processing)
- 6-bit correlation (+13.4% sensitivity)



Carpenter+
(2022)





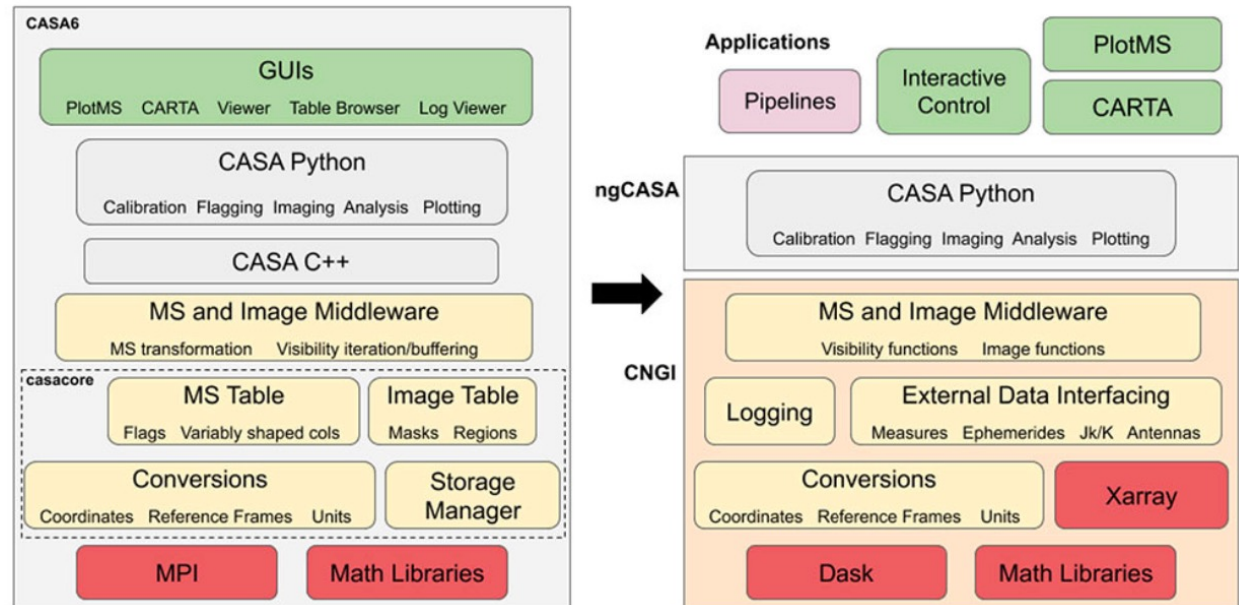
CASA Next Generation Infrastructure

CASA Next Generation Infrastructure (CNGI)

- Infrastructure next-generation CASA → processing demands ngVLA + ALMA WSU
- Requirements: efficient and easily scalable to large computing environments
Reduce code complexity/development time; increase reliability/flexibility/scalability
- Prototyping completed in 2021 and made available as a demonstration package
→ <https://cngi-prototype.readthedocs.io/en/stable/>.
Built in Python; off-the-shelf technologies (numpy, dask, xarray); natively parallel

Next-generation CASA (ngCASA)

- scientific package on top CNGI, serving ngVLA / ALMA WSU (but also current ALMA and VLA)





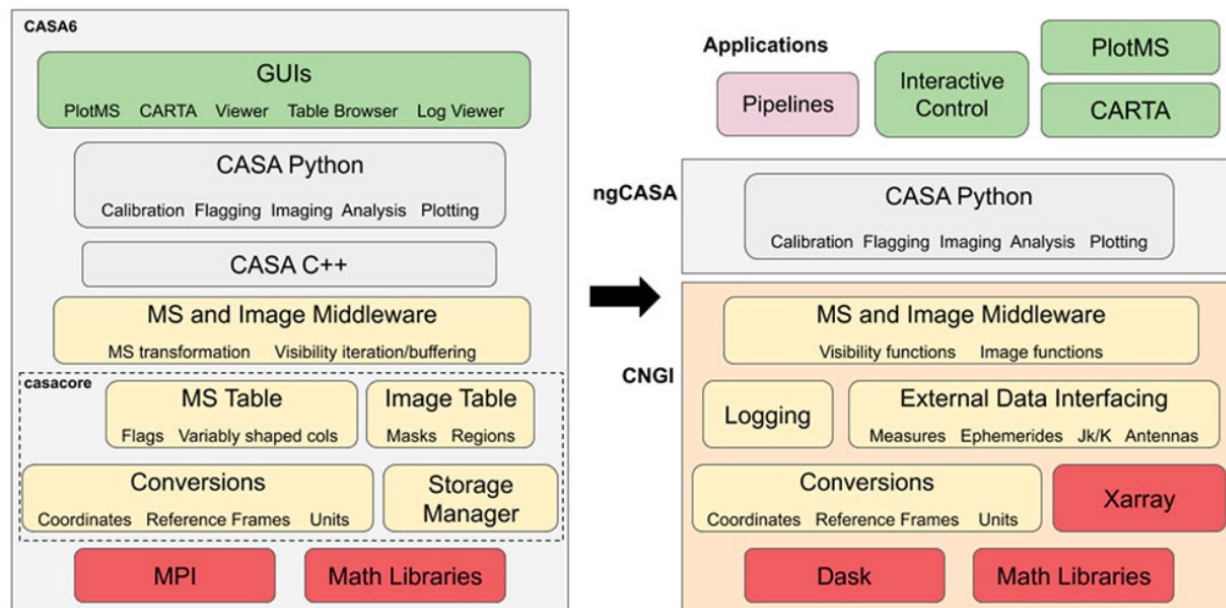
CASA Next Generation Infrastructure

CASA Next Generation Infrastructure (CNGI)

- Infrastructure next-generation CASA → processing demands ngVLA + ALMA WSU
- Requirements: efficient and easily scalable to large computing environments
Reduce code complexity/development time; increase reliability/flexibility/scalability
- Prototyping completed in 2021 and made available as a demonstration package
→ <https://cngi-prototype.readthedocs.io/en/stable/>.
Built in Python; off-the-shelf technologies (numpy, dask, xarray); natively parallel

Next-generation CASA (ngCASA)

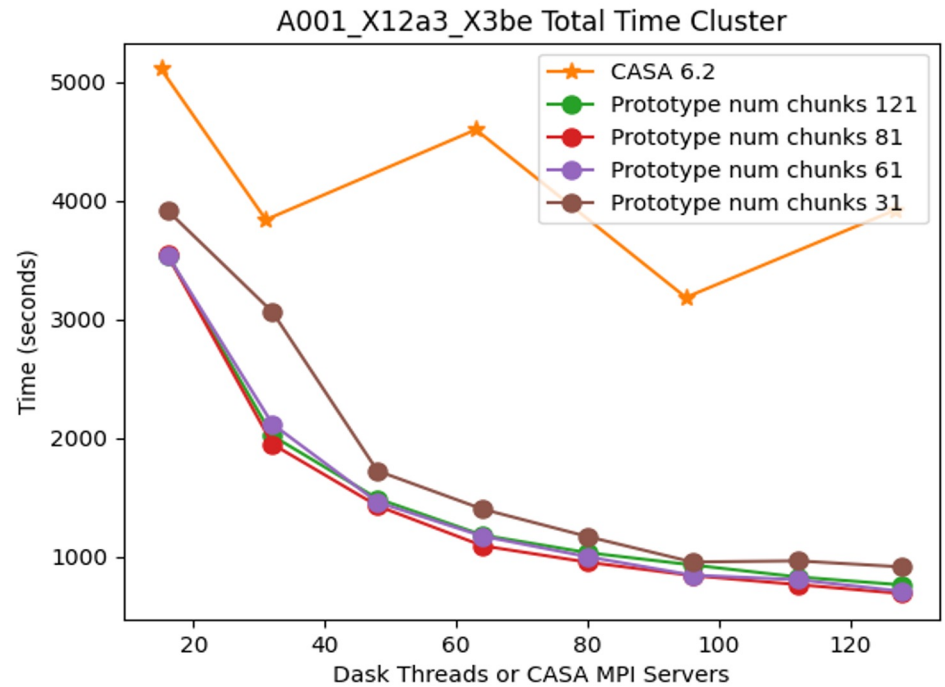
- scientific package on top CNGI, serving ngVLA / ALMA WSU (but also current ALMA and VLA)



CASA Next Generation

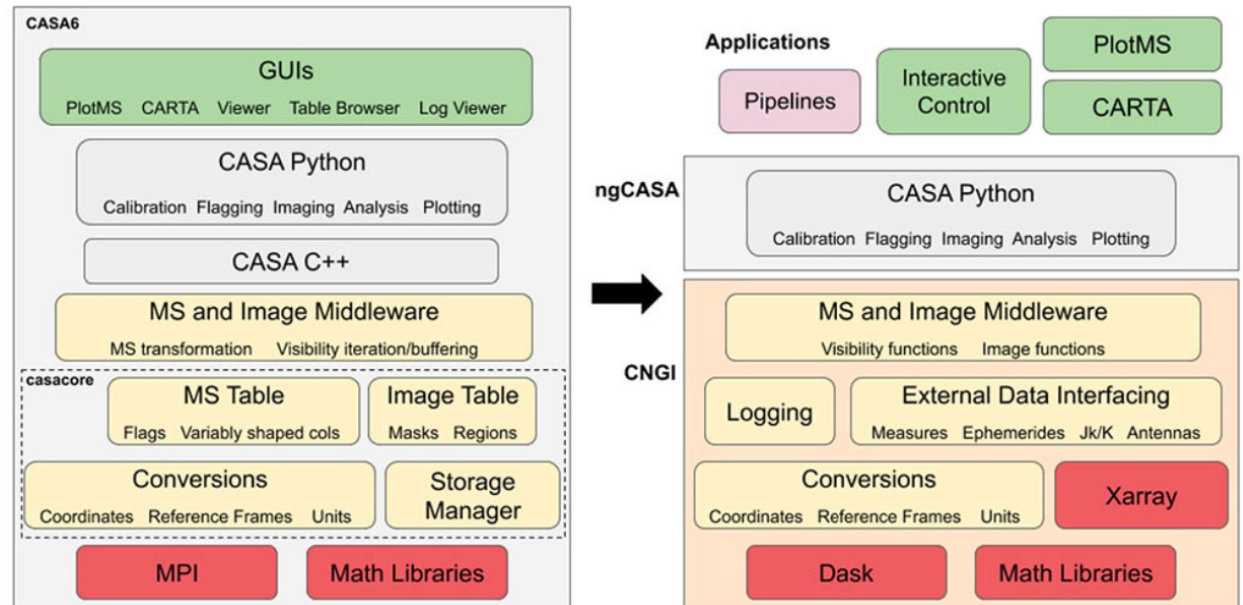
CASA Next Generation Infrastructure

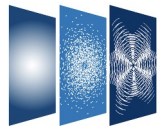
- Infrastructure next-generation CASA →
- Requirements: efficient and easily scaled
Reduce code complexity/development
- Prototyping completed in 2021 and more to come
→ <https://cngi-prototype.readthedocs.io/>
Built in Python; off-the-shelf technologies



Next-generation CASA (ngCASA)

- scientific package on top CNGI, serving ngVLA / ALMA WSU (but also current ALMA and VLA)





CASA resources

- **CASA Docs:** official CASA documentation <https://casadocs.readthedocs.io>
- **CASA Website:** official CASA website (downloads) <https://casa.nrao.edu>
- **CASA Reference:** papers PASP
- **CASA email lists:** casa-announce → announcements, releases [Subscribe!](#)
casa-news → CASA Newsletter (casa.nrao.edu)

Maintained by telescope instrument teams:

- **CASA Guides:** data reduction strategies (telescope-specific, incl. EVN © JIVE)
<https://casaguides.nrao.edu>

Coming soon:

- **CASA Bug Report system:** linked to NRAO Helpdesk

Contact CASA Team:

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

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 various use-cases (*manual, scripting, pipelines, Notebooks*)
- Matured for VLBI processing → this JIVE workshop!
- CASA Docs primary resource for users (*casadocs.readthedocs.io*)
- Ongoing efforts next-generation CASA

Thank you!



"This event has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101004719"