

Introduction to CASA



Jürgen Ott (NRAO Socorro)

6th VLA Data Reduction Workshop
Socorro NM
23 October 2017

Atacama Large Millimeter/submillimeter Array
Expanded Very Large Array

Overview of this talk

- General introduction to CASA
- Documentation and web resources
- Starting CASA
- Tasks, tools, and applications
- Structure of measurement sets and associated data
- CASA data selection syntax
- Current Developments

General description

- CASA: Common Astronomy Software Applications
 - Post-processing package for next gen facilities like ALMA and VLA, and both interferometric and single dish
 - Data from other telescopes also usually work
 - Developed at NRAO (lead), ESO, NAOJ, CSIRO/ATNF and ASTRON
 - Active in community since October 2007
- Code is C++ (fast) underneath iPython interface (easy access & scripting)
- Latest CASA release is **version 5.1.0/5.1.1**
 - Many tasks and a lot of tools
 - Contains automated calibration pipeline
 - Uses iPython 5.1.0 for Linux, 5.3.0 for Mac



<http://casa.nrao.edu>

CASA releases

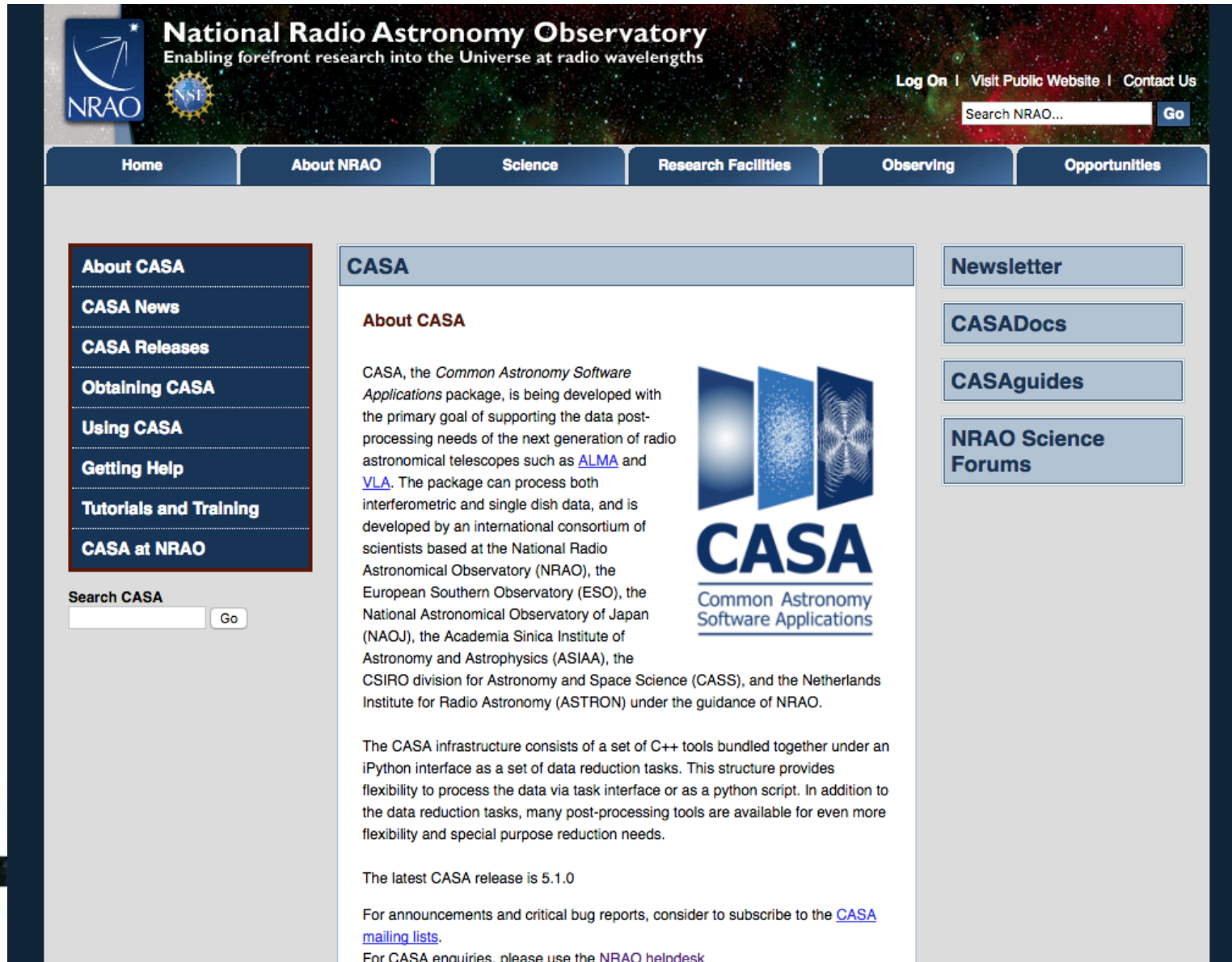
- New releases about every 6 months (typically May and November)
 - Also “prerelease” versions that are markers on path to next release with more functionality, but likely contain unfinished developments, less tested code, and less up-to-date documentation
- Latest version 5.1.0/5.1.1 runs on:
 - Red Hat Linux 6 and 7 (64-bit)
 - Mac OS X 10.11 (El Capitan) and 10.12 (Sierra)
 - May also work on other systems, see website for details
 - 5.3.0 (spring 2018) is planned to run on Mac OS X 10.13 (High Sierra), drop 10.11

Why use CASA?

- Import data, inspect, edit, calibrate, image, view, analyze
 - CASA has some of the most sophisticated imaging algorithms (multi-scale clean, Taylor term expansion for wide bandwidths, W-term projection, OTF mosaicking, etc.)
- We have an active Algorithm Research Group, so expect more features in future versions, → various talks in this workshop, see also CASA newsletter

CASA documentation and web resources

CASA Homepage <http://casa.nrao.edu>



The screenshot shows the CASA homepage on the NRAO website. The header features the NRAO logo and the text "National Radio Astronomy Observatory" with the tagline "Enabling forefront research into the Universe at radio wavelengths". Navigation links include "Log On", "Visit Public Website", and "Contact Us". A search bar is present with the text "Search NRAO..." and a "Go" button. The main navigation bar includes links for "Home", "About NRAO", "Science", "Research Facilities", "Observing", and "Opportunities".

About CASA


- CASA News
- CASA Releases
- Obtaining CASA
- Using CASA
- Getting Help
- Tutorials and Training
- CASA at NRAO

Search CASA

CASA

About CASA

CASA, the *Common Astronomy Software Applications* package, is being developed with the primary goal of supporting the data post-processing needs of the next generation of radio astronomical telescopes such as [ALMA](#) and [VLA](#). The package can process both interferometric and single dish data, and 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 Astrophysics (ASIAA), the CSIRO division for Astronomy and Space Science (CASS), and the Netherlands Institute for Radio Astronomy (ASTRON) under the guidance of NRAO.



The CASA infrastructure consists of a set of C++ tools bundled together under an iPython interface as a set of data reduction tasks. This structure provides flexibility to process the data via task interface or as a python script. In addition to the data reduction tasks, many post-processing tools are available for even more flexibility and special purpose reduction needs.

The latest CASA release is 5.1.0

For announcements and critical bug reports, consider to subscribe to the [CASA mailing lists](#).


For CASA enquiries, please use the [NRAO helndesk](#)

Newsletter

CASADocs

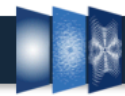
CASAguides

NRAO Science Forums



CASA documentation and web resources

CASAdocs <https://casa.nrao.edu/casadocs> Documentation



CASA Documentation Archives

Search Site



Directory

Release Information

Hardware Requirements

Using CASA

Visibility Data Import

Export

Visibility Data Selection

Data Examination and

Editing

UV Manipulation

Synthesis Calibration

Ephemeris Data

Synthesis Imaging

Single Dish Calibration

Single-Dish Imaging

Image Combination

Image Cube

Visualization

Image Analysis

Simulation

Parallel Processing

Pipeline

Home / CASA 5.1.0

CASA 5.1.0

Documentation for CASA 5.1.0

CASA, the *Common Astronomy Software Applications* package, is being developed with the primary goal of supporting the data post-processing needs of the next generation of radio astronomical telescopes such as [ALMA](#) and [VLA](#). The package can process both interferometric and single dish data, and 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 CSIRO Australia Telescope National Facility (CSIRO/ATNF), and the Netherlands Institute for Radio Astronomy (ASTRON) under the guidance of NRAO.

The CASA infrastructure consists of a set of C++ tools bundled together under an iPython interface as a set of data reduction tasks. This structure provides flexibility to process the data via task interface or as a python script. In addition to the data reduction tasks, many post-processing tools are available for even more flexibility and special purpose reduction needs.

Release Information

Information concerning this release of CASA

Hardware Requirements

Recommended CASA computing environments

Using CASA

Description of how CASA interacts with the python environment.

Visibility Data Import Export

Description of how to import and export Visibility Data to CASA.

Visibility Data Selection


Techniques to select data in the Visibility domain.

Data Examination and Editing



CASA documentation and web resources

CASAguides <https://casaguides.nrao.edu> Tutorials



■ Main Page

search

Go Search


tools










- What links here
- Related changes
- Upload file
- Special pages
- Printable version
- Permanent link
- Page information

main page discussion edit history delete move unwatch

Jott talk preferences watchlist contributions log out

Welcome to CASA Guides

 **CASA** (Common Astronomy Software Applications) is a comprehensive software package to calibrate, image, and analyze radioastronomical data from interferometers (such as [ALMA](#) and [VLA](#)) as well as single dish telescopes. This wiki provides tutorials and hints for reducing data in CASA.

CASA Home 	CASADocs 	Newsletter 
Forum 	Mailing Lists 	CASA Tips 
NRAO Helpdesk 	ALMA Helpdesk 	Download 



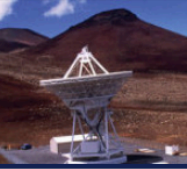
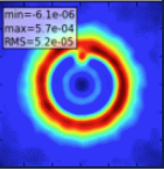

CASA News

- 18 July 2017: [CASA Release 5.0.0](#) is now available
- 11 October 2016: [CASA Newsletter #4](#)

Events

- 23-27 October 2017: [6th VLA Data Reduction Workshop](#) (NRAO Socorro, NM)

CASA Tutorials

ALMA 	Jansky VLA 	VLBI 	Simulations 	ATCA 
--	--	---	---	--

Extracting Scripts from Tutorials

Starting CASA

- Start CASA from the UNIX shell:

% casa

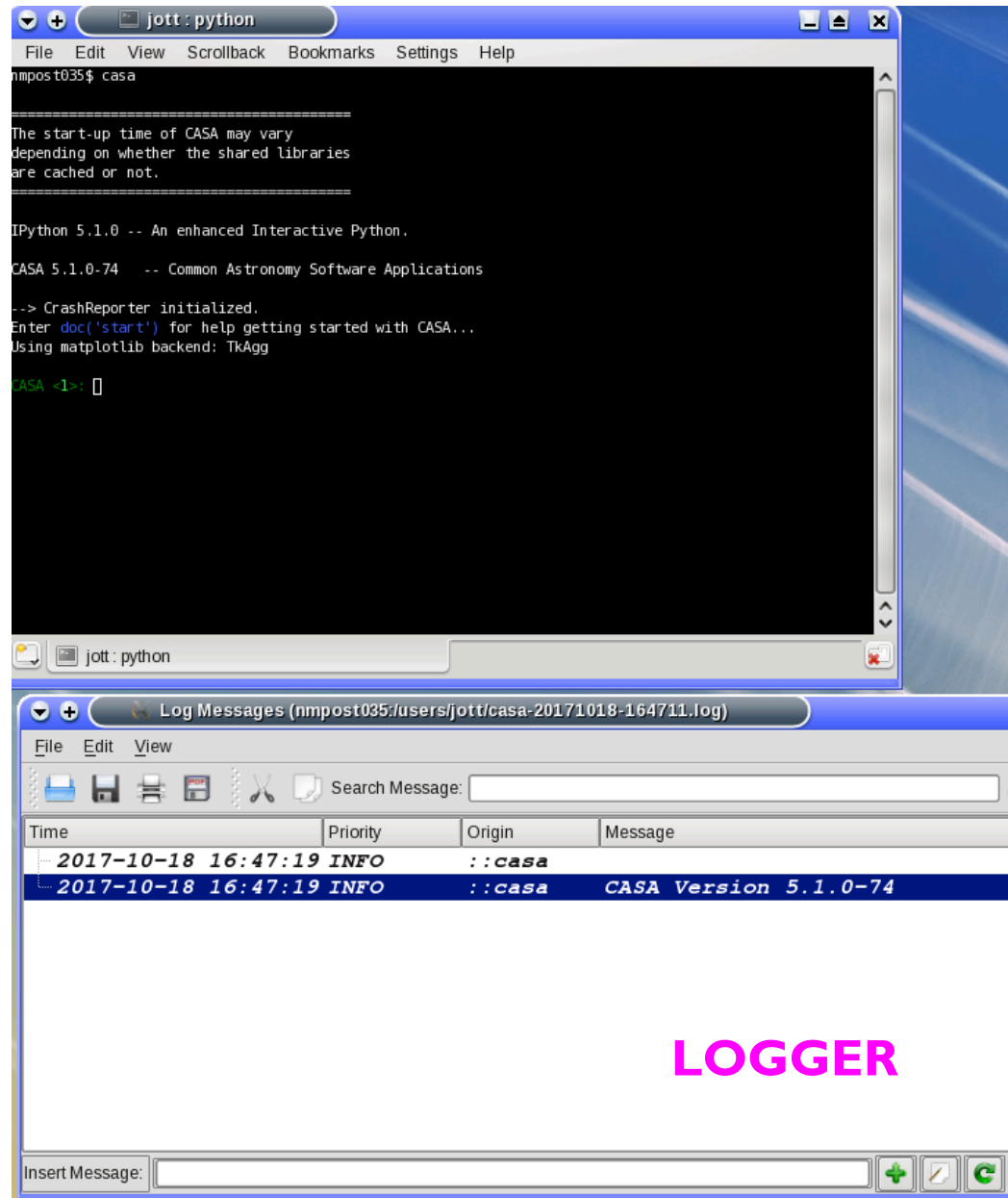
- Session logging:

- ipython-TIMESTAMP.log**

iPython command history

- casapy-TIMESTAMP.log**

CASA logger messages



The screenshot shows two windows. The top window is a terminal titled 'jott: python' showing the output of the 'casa' command. The bottom window is a log viewer titled 'Log Messages (nmpost035:/users/jott/casa-20171018-164711.log)' showing a table of log messages.

```
nmpost035$ casa

=====
The start-up time of CASA may vary
depending on whether the shared libraries
are cached or not.
=====

IPython 5.1.0 -- An enhanced Interactive Python.
CASA 5.1.0-74 -- Common Astronomy Software Applications

--> CrashReporter initialized.
Enter doc('start') for help getting started with CASA...
Using matplotlib backend: TkAgg

CASA <1>: []
```

Time	Priority	Origin	Message
2017-10-18 16:47:19	INFO	::casa	
2017-10-18 16:47:19	INFO	::casa	CASA Version 5.1.0-74

LOGGER

CASA interactive interface

- iPython interface (ipython.org) provides:
 - Numbered input/output
 - Shell access with leading exclamation mark, e.g. **!pwd** (or **os.system**)
 - Tab auto-completion
 - Auto-parenthesis
 - Command history (up-arrow or **hist [-n]**)
 - History/searching (start typing then use up-arrow, or use Ctrl-r)
- Some python tips:
 - Indentation matters, used for loops & conditions (beware copy paste)
 - Indices start from 0 and run to n-1

CASA tasks, tools, and applications

TASKS

- High-level functionality (set parameters and press go, or script)
- These are what you will probably use most

TOOLS

- Provide access to complete functionality of CASA
- Used internally by tasks
- Sometimes shown in tutorial scripts

APPLICATIONS

- Typically used to view data (tables, images)
- Can be invoked inside CASA or as standalone programs

Find the right task

To see an organized list, type:

> tasklist

```
Terminal <3>
File Edit View Search Terminal Help
CASA <18>: tasklist
-----> tasklist()
Available tasks, organized by category (experimental tasks in parenthesis ()
deprecated tasks in curly brackets {}).
```

Import/export	Information	Editing	Manipulation
exportasdm	imhead	fixplanets	concat
exportfits	imreframe	fixvis	conjugatevis
exportuvfits	imstat	flagcmd	cvel
importasdm	imval	flagdata	fixvis
importfits	listcal	flagmanager	hanningsmooth
importfitsidi	listfits	msview	imhead
importmiriad	listhistory	plotms	msmoments
importuvfits	listobs		mstransform
importvla	listpartition		partition
(importevla)	listvis		plotms
(importgmt)	plotms		split
	plotuv		testconcat
	vishead		uvcontsub
	visstat		virtualconcat
	(asdmsummary)		vishead
	(listsdm)		(cvel2)
	(makemask)		(hanningsmooth2)
			(split2)
			(statwt)
			(uvcontsub3)
Calibration	Modeling	Imaging	Analysis
accum	predictcomp	clean	imcollapse
applycal	setjy	deconvolve	imcontsub
bandpass	uvcontsub	feather	imfit
blcal	uvmodelfit	ft	imhead
calstat	uvsub	imcontsub	immath
clearcal	(uvcontsub3)	(boxit)	immoments
delmod		(csvclean)	impcor
fixplanets		(tclean)	impv
fluxscale		(widebandpbcpr)	imrebin
ft		{mosaic}	imreframe
gaincal		{widefield}	imregrid
gencal			imsmooth
initweights			imstat
listcal			imsubimage
plotants			imtrans
plotbandpass			imval

Find the right task

To see short summaries, type:

> taskhelp

```
Terminal <3>
File Edit View Search Terminal Help
CASA <12>: taskhelp
-----> taskhelp()
Available tasks:

accum          : Accumulate incremental calibration solutions into a calibration table
applycal       : Apply calibrations solutions(s) to data
asdmsummary    : Summarized description of an ASDM dataset.
autoclean      : CLEAN an image with automatically-chosen clean regions.
bandpass       : Calculates a bandpass calibration solution
blcal          : Calculate a baseline-based calibration solution (gain or bandpass)
boxit          : Box regions in image above given threshold value.
browsetable    : Browse a table (MS, calibration table, image)
calstat        : Displays statistical information on a calibration table
caltabconvert  : Convert old-style caltables into new-style caltables.
clean          : Invert and deconvolve images with selected algorithm
clearcal       : Re-initializes the calibration for a visibility data set
clearplot      : Clear the matplotlib plotter and all layers
clearstat      : Clear all autolock locks
concat         : Concatenate several visibility data sets.
conjugatevis   : Change the sign of the phases in all visibility columns.
csvclean       : This task does an invert of the visibilities and deconvolve in the image plane.
cvel           : regrid an MS to a new spectral window / channel structure or frame
cvel2          : Regrid an MS or MMS to a new spectral window, channel structure or frame
deconvolve     : Image based deconvolver
delmod         : Deletes model representations in the MS
exportasdm     : Convert a CASA visibility file (MS) into an ALMA or EVLA Science Data Model
exportfits     : Convert a CASA image to a FITS file
exportuvfits   : Convert a CASA visibility data set to a UVFITS file:
feather        : Combine two images using their Fourier transforms
find           : Find string in tasks, task names, parameter names:
fixplanets     : Changes FIELD and SOURCE table entries based on user-provided direction or POINTING table
fixvis         : Recalculates (u, v, w) and/or changes Phase Center
flagcmd        : Flagging task based on batches of flag-commands
flagdata       : All-purpose flagging task based on data-selections and flagging modes/algorithms.
flagmanager    : Enable list, save, restore, delete and rename flag version files.
fluxscale      : Bootstrap the flux density scale from standard calibrators
ft             : Insert a source model a visibility set:
gaincal        : Determine temporal gains from calibrator observations
gencal         : Specify Calibration Values of Various Types
hanningsmooth  : Hanning smooth frequency channel data to remove Gibbs ringing
hanningsmooth2 : Hanning smooth frequency channel data to remove Gibbs ringing
imcollapse     : Collapse image along one axis, aggregating pixel values along that axis.
imcontsub      : Estimates and subtracts continuum emission from an image cube
imfit          : Fit one or more elliptical Gaussian components on an image region(s)
imhead         : List, get and put image header parameters
immath         : Perform math operations on images
immoments      : Compute moments from an image
impbcor        : Construct a primary beam corrected image from an image and a primary beam pattern.
importasdm     : Convert an ALMA Science Data Model observation into a CASA visibility file (MS) or similar
importevla     : Convert an Science Data Model observation into a CASA Measurement Set
importfits     : Convert an image FITS file into a CASA image
importfitsidi  : Convert a FITS-IDI file to a CASA visibility data set
importgmt      : Convert a UVFITS file to a CASA visibility data set
importmiriad   : Convert a Miriad visibility file into a CASA MeasurementSet
importuvfits   : Convert a UVFITS file to a CASA visibility data set
```

Task interface

Inspect task inputs:

> **inp clean**

Black: default value

Red: invalid value

Blue: non-default value

Reset defaults:

> **default clean**

Grey: expandable

```

Terminal <3>
File Edit View Search Terminal Help
CASA <67>: inp clean
-----> inp(clean)
# clean :: Invert and deconvolve images with selected algorithm
vis          =      ''      # Name of input visibility file
imagename    =      ''      # Pre-name of output images
outlierfile   =      ''      # Text file with image names, sizes, centers for outliers
field        =      ''      # Field Name or id
spw          =      ''      # Spectral windows e.g. '0~3', '' is all
selectdata   =      True    # Other data selection parameters
  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)

mode         = 'burrito'    # Spectral gridding type (mfs, channel, velocity, frequency)
gridmode     =      ''      # Gridding kernel for FFT-based transforms, default='' None
niter        =      500     # Maximum number of iterations
gain         =      0.1     # Loop gain for cleaning
threshold    = '5.0mJy'    # Flux level to stop cleaning, must include units: '1.0mJy'
psfmode      = 'clark'     # Method of PSF calculation to use during minor cycles
imagermode   = 'csclean'   # Options: 'csclean' or 'mosaic', '', uses psfmode
  cyclefactor =      1.5     # Controls how often major cycles are done. (e.g. 5 for frequently)
  cyclespeedup =     -1     # Cycle threshold doubles in this number of iterations

multiscale   =      []      # Deconvolution scales (pixels); [] = standard clean
interactive  =      False   # Use interactive clean (with GUI viewer)
mask         =      []      # Cleanbox(es), mask image(s), region(s), or a level
imsize       = [256, 256]   # x and y image size in pixels. Single value: same for both
cell         = ['1.0arcsec'] # x and y cell size(s). Default unit arcsec.
phasecenter  =      ''      # Image center: direction or field index
restfreq     =      ''      # Rest frequency to assign to image (see help)
stokes       = 'I'         # Stokes params to image (eg I,IV,IQ,IQUV)
weighting    = 'natural'   # Weighting of uv (natural, uniform, briggs, ...)
uvtaper      =      False   # Apply additional uv tapering of visibilities
modelimage   =      ''      # Name of model image(s) to initialize cleaning
restoringbeam =      ['']   # Output Gaussian restoring beam for CLEAN image
pbcor        =      False   # Output primary beam-corrected image
minpb        =      0.2     # Minimum PB level to use
uscratch     =      False   # True if to save model visibilities in MODEL_DATA column
allowchunk   =      False   # Divide large image cubes into channel chunks for deconvolution

CASA <68>: 

```



Task interface

Grey: expandable

Green: sub-parameter

- Colors vary depending on your terminal settings, change if not readable



```

Terminal <3>
File Edit View Search Terminal Help
CASA <70>: inp
-----> inp()
# clean :: Invert and deconvolve images with selected algorithm
vis = '' # Name of input visibility file
imagename = '' # Pre-name of output images
outlierfile = '' # Text file with image names, sizes, centers for outliers
field = '' # Field Name or id
spw = '' # Spectral windows e.g. '0~3', '' is all
selectdata = True # Other data selection parameters
  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)

mode = 'burrito' # Spectral gridding type (mfs, channel, velocity, frequency)
gridmode = '' # Gridding kernel for FFT-based transforms, default='' None
niter = 500 # Maximum number of iterations
gain = 0.1 # Loop gain for cleaning
threshold = '5.0mJy' # Flux level to stop cleaning, must include units: '1.0mJy'
psfmode = 'clark' # Method of PSF calculation to use during minor cycles
imagermode = 'csclean' # Options: 'csclean' or 'mosaic', '', uses psfmode
  cyclefactor = 1.5 # Controls how often major cycles are done. (e.g. 5 for frequently)
  cyclespeedup = -1 # Cycle threshold doubles in this number of iterations

multiscale = [] # Deconvolution scales (pixels); [] = standard clean
interactive = False # Use interactive clean (with GUI viewer)
mask = [] # Cleanbox(es), mask image(s), region(s), or a level
imsize = [256, 256] # x and y image size in pixels. Single value: same for both
cell = ['1.0arcsec'] # x and y cell size(s). Default unit arcsec.
phasecenter = '' # Image center: direction or field index
restfreq = '' # Rest frequency to assign to image (see help)
stokes = 'I' # Stokes params to image (eg I,IV,IQ,IQUV)
weighting = 'briggs' # Weighting of uv (natural, uniform, briggs, ...)
  robust = 0.0 # Briggs robustness parameter
  npixels = 0 # number of pixels to determine uv-cell size 0=> field of view

uvtaper = False # Apply additional uv tapering of visibilities
modelimage = '' # Name of model image(s) to initialize cleaning
restoringbeam = [] # Output Gaussian restoring beam for CLEAN image
pbcor = False # Output primary beam-corrected image
minpb = 0.2 # Minimum PB level to use
usescratch = False # True if to save model visibilities in MODEL_DATA column
allowchunk = False # Divide large image cubes into channel chunks for deconvolution

CASA <71>:
  
```

Task help

Type:

> help clean

```
Terminal <3>
File Edit View Search Terminal Help

Example :

The clean task has many options:


1) Make 'dirty' image and 'dirty' beam (psf)
2) Multi-frequency-continuum images or spectral channel imaging
3) Full Stokes imaging
4) Mosaicking of several pointings
5) Multi-scale cleaning
6) Widefield cleaning
7) Interactive clean boxing
8) Use starting model (eg from single dish)

vis -- Name(s) of input visibility file(s)
      default: none;
      example: vis='ngc5921.ms'
               vis=['ngc5921a.ms','ngc5921b.ms']; multiple MSes
imagenam -- Pre-name of output images:
           default: none; example: imagename='m2'
           output images are:
             m2.image; cleaned and restored image
               With or without primary beam correction
             m2.psf; point-spread function (dirty beam)
             m2.flux; relative sky sensitivity over field
             m2.flux.pbcoverage; relative pb coverage over field
                           (gets created only for ft='mosaic')
             m2.model; image of clean components
             m2.residual; image of residuals
             m2.interactive.mask; image containing clean regions
           To include outlier fields:
             imagename=['n5921','outlier1','outlier2']
outlierfile --- Text file name which contains image names, sizes, field
                centers (See 'HINTS ON CLEAN WITH FLANKING FIELDS' below
                for the format of this outlier file.)
field -- Select fields to image or mosaic. Use field id(s) or name(s).
        ['go listobs' to obtain the list id's or names]
        default: ''= all fields
        If field string is a non-negative integer, it is assumed to
        be a field index otherwise, it is assumed to be a
        field name
        field='0~2'; field ids 0,1,2
        field='0,4,5~7'; field ids 0,4,5,6,7
        field='3C286,3C295'; field named 3C286 and 3C295
        field = '3,4C*'; field id 3, all names starting with 4C
        For multiple MS input, a list of field strings can be used:
        field = ['0~2','0~4']; field ids 0-2 for the first MS and 0-4

lines 301-348
```


Task help

- Use **doc('clean')** from now on → **launches casadocs**
- (but not yet available for all tasks/tools, and depends on default browser)

 CASA Documentation Archives

Log in Search Site

Subdirectory

[Back to Directory](#)

[applycal](#)

[bandpass](#)

[blcal](#)

[calstat](#)

[clean](#)

[concat](#)

[exportasdm](#)

[exportfits](#)

[exportuvfits](#)

[flagdata](#)

[fluxscale](#)

[gaincal](#)

[gencal](#)

[hanningsmooth](#)

[imfit](#)

[imhistory](#)

[immath](#)

[immoments](#)

[impbcor](#)

[importasap](#)

Home / CASA 5.1.1 / Global Task List / clean / Description

clean

DescriptionParametersChangelogExamples

Note that **clean** is no longer being actively developed; it has been refactored, and the new version of the task is called **tclean**. Much more information on the context for **clean**/**tclean** can be found in the [Sythesis Imaging](#) chapter.

Basics of clean

The CLEAN algorithm (more details available [here](#)) is the most popular and widely-studied method for reconstructing a model image based on interferometer data. It iteratively removes at each step a fraction of the flux in the brightest pixel in a defined region of the current “dirty” image, and places this in the model image. The **clean** task implements the CLEAN algorithm for single-field data. The user can choose from a number of options for the particular flavor of CLEAN to use. Often, the first step in imaging is to make a simple gridded Fourier inversion of the calibrated data to make a “dirty” image. This can then be examined to look for the presence of noticeable emission above the noise, and to assess the quality of the calibration by searching for artifacts in the image. This is done using **clean** with *niter*=0.

ALERT: For large fractional bandwidths, the psf in **clean** may vary considerably with frequency in data cubes. To accommodate this fact we have introduced a per-plane psf (dirty beam) when the change is larger than half the size of a pixel. Analysis tasks in CASA can deal with such beam variation. If a single beam size is requested, **imsmooth** can be invoked on the clean products to smooth to a common, uniform beam for all channels.

The **clean** task has many options:

- Make 'dirty' image and 'dirty' beam (PSF)
- Multi-frequency-continuum images or spectral channel imaging
- Full Stokes imaging
- Mosaicking of several pointings
- Multi-scale cleaning
- Widefield cleaning
- Interactive clean boxing
- Use starting model (e.g. from single-dish)

How to run a task

- Task interface

- Use **inp taskname** to see list of parameters
- Set (global) parameters one at a time
- Useful for interactive work, exploring parameters
- Recover previous parameters using **tget taskname** (antonym **tput**)

```
> inp listobs  
> vis = 'mydata.ms'  
> listfile = 'outfile.txt'  
> inp  
> go
```

- iPython command line

- Set all parameters at once
- Useful for scripting
 - Copy-paste into a text or .py file to keep record of processing

```
> listobs(vis='mydata.ms',listfile='outfile.txt')
```

```
> execfile('commands.py')
```

Some things to note about running tasks

- Some tasks return a dictionary `> results = imstat()`
- CASA <13>: results
- Out[13]:{'blc': array([0, 0, 0, 0], dtype=int32), 'blcf': '09:47:57.724, +13.16.35.660, 1, 3.63124e+10Hz', 'max': array([0.00010101]),.....
- Command line execution will ignore global parameters

- Unspecified parameters will be set to their default values

```
> field='3C286'  
> listobs(vis='mydata.ms',listfile='outfile.txt')
```

'field' will
be ignored
here (set
to default)

- Exception is if global parameter is explicitly referenced

```
> field='3C286'  
> listobs(vis='mydata.ms',listfile='outfile.txt',field=field)
```

Tools

- What if there's no task?
 - Use CASA tools!
- Tools (and their methods) are the building blocks of tasks
 - Contain full functionality of CASA
 - Used internally by tasks
 - E.g. imaging utilities (im), table utilities (tb), ...

Using tools and their methods

- Tools contain a number of methods
 - Access using **tool.method()**
 - Use tab-completion to see listing
- Typically, data must be opened and closed (unlike tasks)
 - Failure to close may block other tasks and clutter memory

```
> ia.open('image.im')  
> ia.regrid(outfile='out.im',...)  
> ia.close()
```

Find the right tool

To see short summaries, type:

> toolhelp

```
Terminal <3>
File Edit View Search Terminal Help
CASA <74>: toolhelp
-----> toolhelp()

Available tools:

af : Agent flagger utilities
at : Juan Pardo ATM library
ca : Calibration analysis utilities
cb : Calibration utilities
cl : Component list utilities
cp : Cal solution plotting utilities
cs : Coordinate system utilities
cu : Class utilities
dc : Deconvolver utilities
fi : Fitting utilities
fn : Functional utilities
ia : Image analysis utilities
im : Imaging utilities
lm: linear mosaic
me : Measures utilities
ms : MeasurementSet (MS) utilities
msmd : MS metadata accessors
mt : MS transformer utilities
qa : Quanta utilities
pm : PlotMS utilities
po : Imagepol utilities
rg : Region manipulation utilities
sl : Spectral line import and search
sm : Simulation utilities
tb : Table utilities (selection, extraction, etc)
tp : Table plotting utilities
vp : Voltage pattern/primary beam utilities
...
pl : pylab functions (e.g., pl.title, etc)
sd : Single dish utilities
...

CASA <75>: █
```


CASA toolkit reference manual

- There's a good chance your problem can be solved on the tool level, don't be afraid to use this resource!
- >1000 tool methods available
- See CASA toolkit reference manual:

<http://casa.nrao.edu/docs/CasaRef/CasaRef.html>

- → see Steve Myer's talk

[What imager produces:](#)

[What imager does not do:](#)

[What improvement to imager are in the works:](#)

[Advanced use of imager:](#)

[Overview of imager tool functions:](#)

2.4.1 imager - Tool

[imager.imager - Function](#)

[imager.advise - Function](#)

[imager.approximatepsf - Function](#)

[imager.boxmask - Function](#)

[imager.caluvvw - Function](#)

[imager.clean - Function](#)

[imager.clipimage - Function](#)

[imager.clipvis - Function](#)

[imager.close - Function](#)

[imager.defineimage - Function](#)

[imager.done - Function](#)

[imager.drawmask - Function](#)

[imager.exprmask - Function](#)

[imager.feather - Function](#)

[imager.filter - Function](#)

[imager.fitpsf - Function](#)

[imager.fixvis - Function](#)

[imager.ft - Function](#)

[imager.linearmosaic - Function](#)

[imager.make - Function](#)

[imager.makeimage - Function](#)

[imager.makemodelfromsd - Function](#)

[imager.mask - Function](#)

[imager.mem - Function](#)

[imager.nnls - Function](#)

[imager.open - Function](#)

[imager.pb - Function](#)

[imager.plotsummary - Function](#)

[imager.plotuv - Function](#)

[imager.plotvis - Function](#)

[imager.plotweights - Function](#)

[imager.regionmask - Function](#)

[imager.regiontoimagemask - Function](#)

[imager.residual - Function](#)

[imager.restore - Function](#)

[imager.sensitivity - Function](#)



CASA toolkit reference manual

- There's a good chance your problem can be solved on the tool level, don't be afraid to use this resource!
- >1000 tool methods available
- See CASA toolkit reference manual:

<http://casa.nrao.edu/Users/CasaRef/CasaRef.html>

- → see Steve Myer's talk

[What imager produces:](#)

[What imager does not do:](#)

[What improvement to imager are in the works:](#)

[Advanced use of imager:](#)

[Overview of imager tool functions:](#)

2.4.1 imager - Tool

[imager.imager - Function](#)

[imager.advise - Function](#)

[imager.approximatepsf - Function](#)

[imager.boxmask - Function](#)

[imager.caluvvw - Function](#)

[imager.clean - Function](#)

[imager.clipimage - Function](#)

[imager.clipvis - Function](#)

[imager.close - Function](#)

[imager.defineimage - Function](#)

[imager.done - Function](#)

[imager.drawmask - Function](#)

[imager.exprmask - Function](#)

[imager.feather - Function](#)

[imager.filter - Function](#)

[imager.fitpsf - Function](#)

[imager.fixvis - Function](#)

[imager.ft - Function](#)

[imager.linearmosaic - Function](#)

[imager.make - Function](#)

[imager.makeimage - Function](#)

[imager.makemodelfromsd - Function](#)

[imager.mask - Function](#)

[imager.mem - Function](#)

[imager.nnls - Function](#)

[imager.open - Function](#)

[imager.pb - Function](#)

[imager.plotsummary - Function](#)

[imager.plotuv - Function](#)

[imager.plotvis - Function](#)

[imager.plotweights - Function](#)

[imager.regionmask - Function](#)

[imager.regiontoimagemask - Function](#)

[imager.residual - Function](#)

[imager.restore - Function](#)

[imager.sensitivity - Function](#)



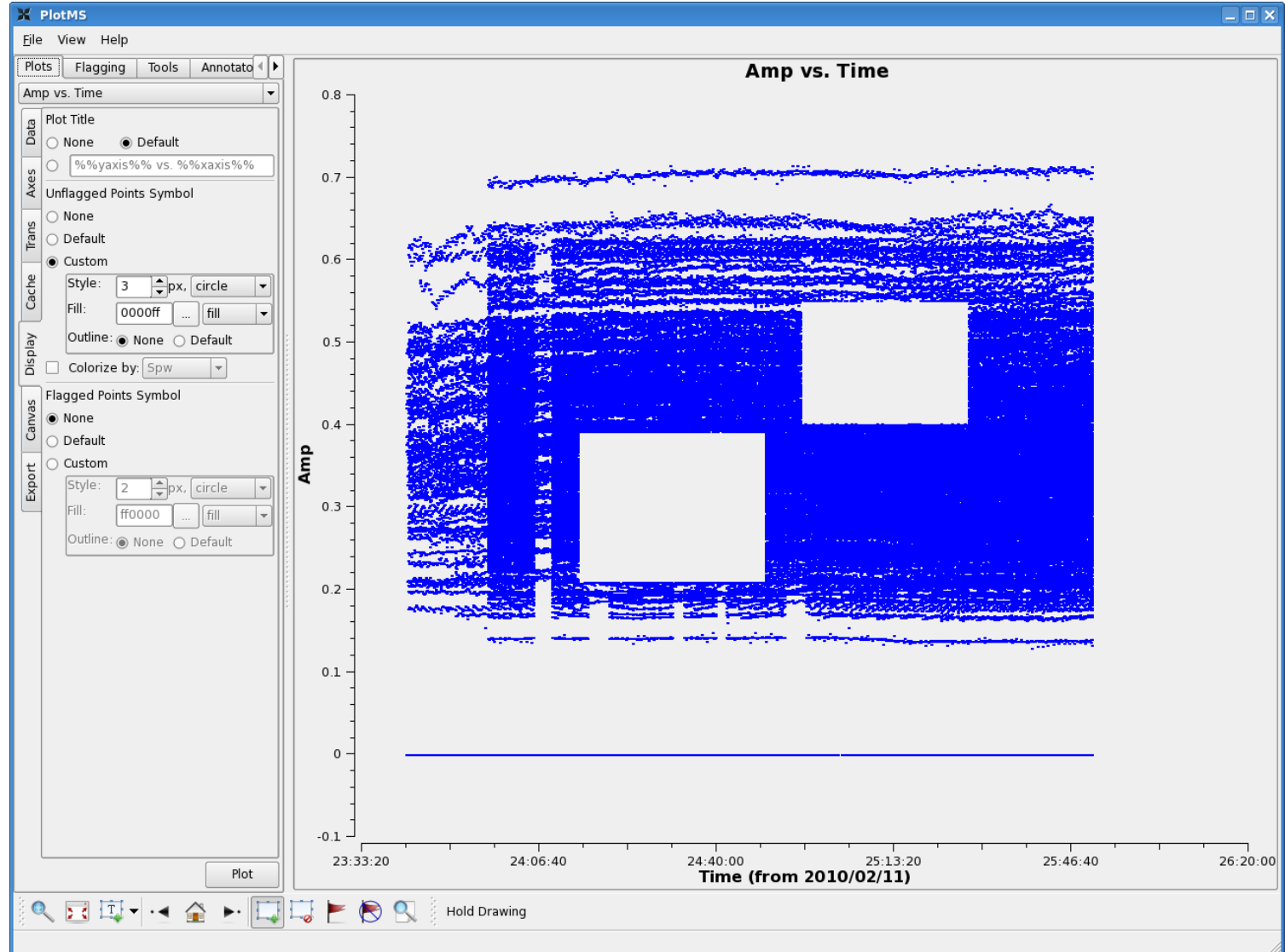
Still searching for functionality?

- Look through contributed scripts and tasks at:
<http://casaguides.nrao.edu/>
- If you still can't find what you need, write your own task!
 - Combination of Python plus CASA toolkit is very powerful
 - We encourage you to submit your own scripts to us through the helpdesk

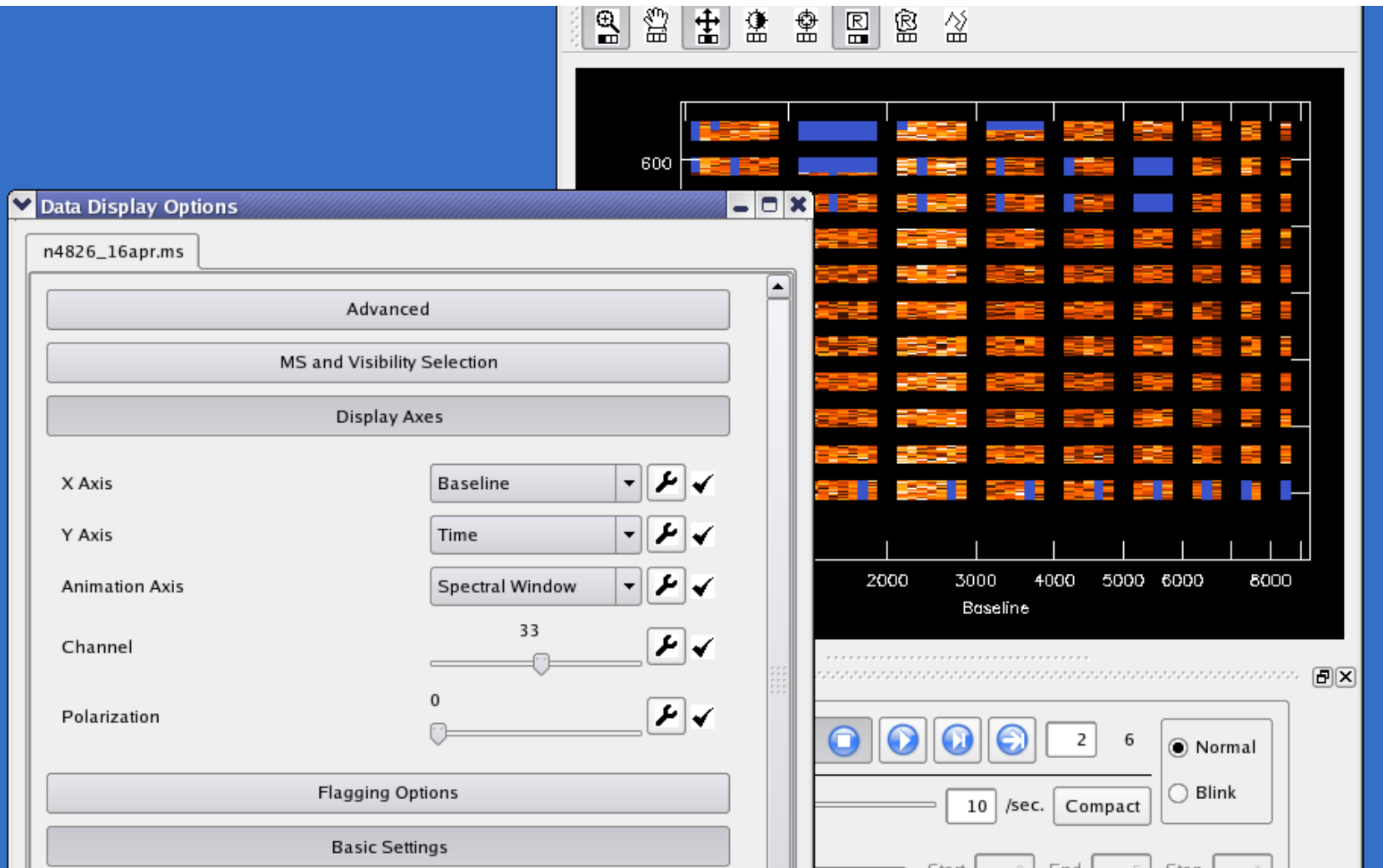
Applications

- Typically used to display data (tables, images)
 - Some editing capabilities
- Can be invoked inside CASA or as standalone programs from shell
- Visibilities: **plotms** (standalone **% casaplotms**), **msview**, **viewer**, **casafeather**
- Any table data: **browsetable** (standalone **% casabrowser**)
- Calibration tables: **plotcal** (eventually **plotms**)
- Images: **imview**, **viewer** (standalone **% casaviewer**)
- Don't forget about full functionality of python! e.g. matplotlib

PlotMS



MSView



Plotcal

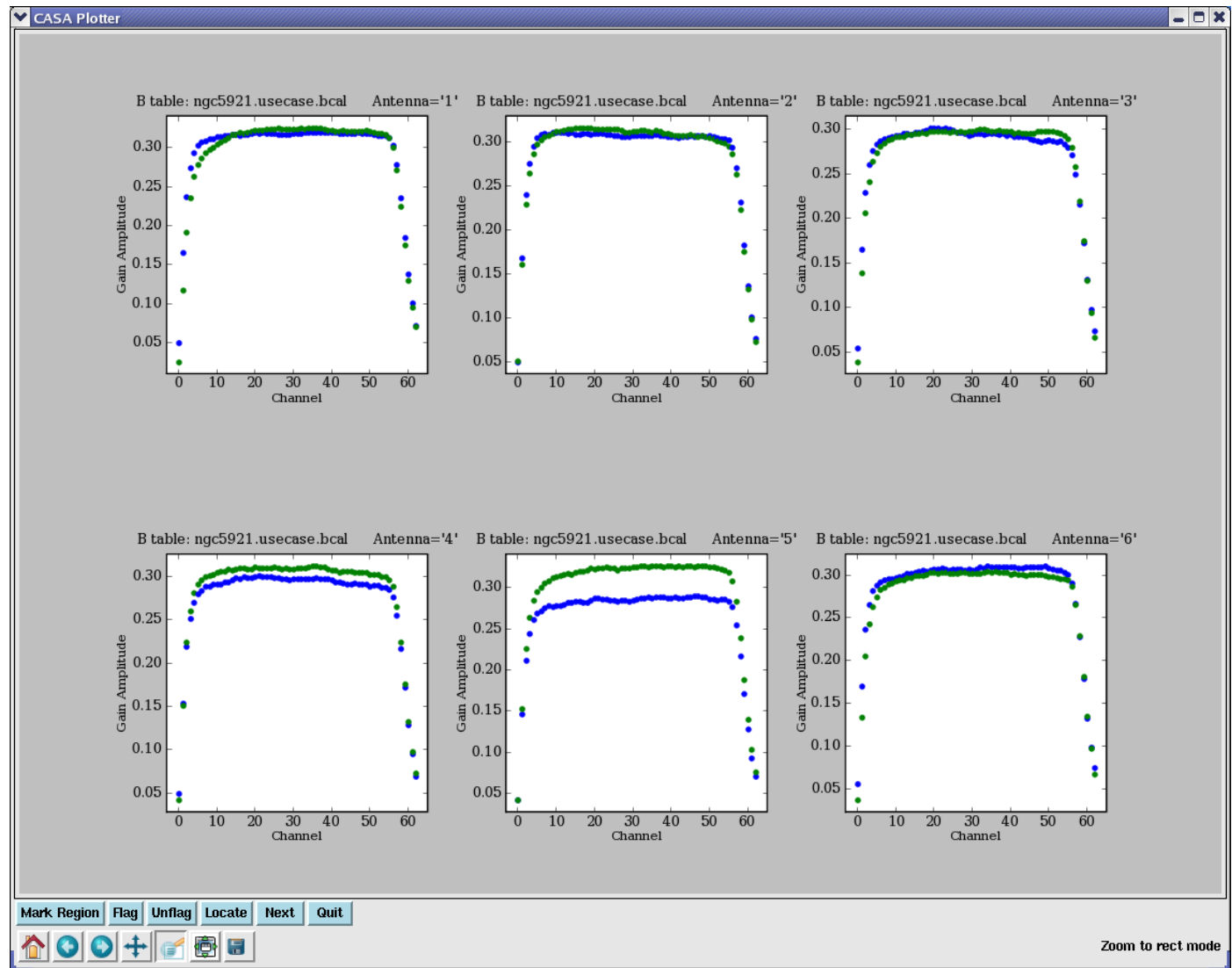


Image Viewer

Animator

☒ Channels

Navigation icons: Previous, Play, Next, First, Last, Home, End, Repeat, Stop, Zoom In, Zoom Out, Fit, Full Screen, Close.

Rate: 10 Jump ☐ 21 40

0 39

Position Tracking

☒ M100line.image

+0.0233867 Jy/beam Pixel: 342 324 21 0

12:22:53.434 +15:49:22.234 1564.79 km/s (topo/radio velocity) I

Image Viewer

- Cubes
- Movies
- Channel maps

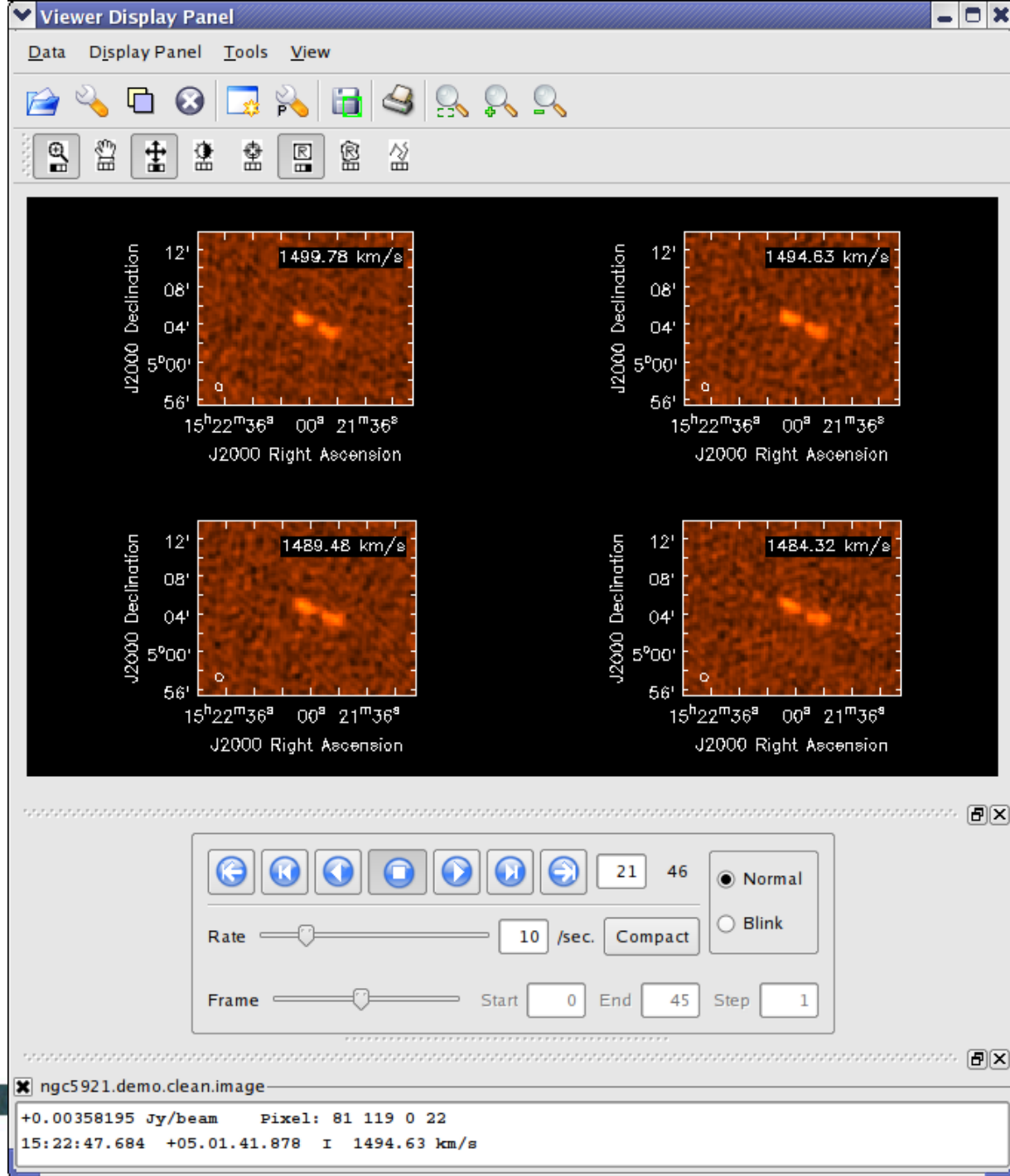
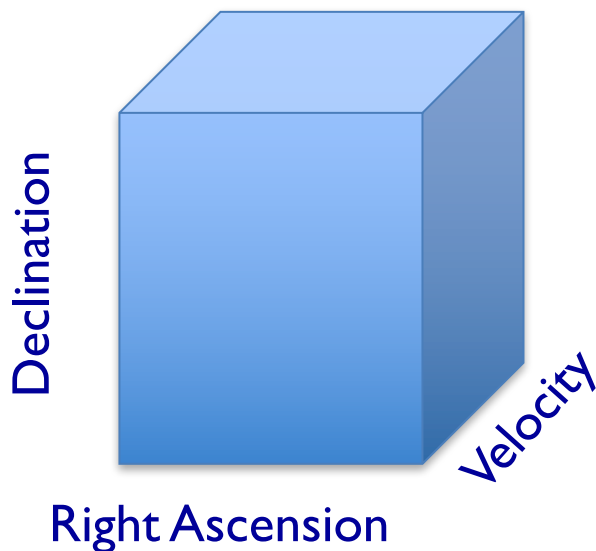
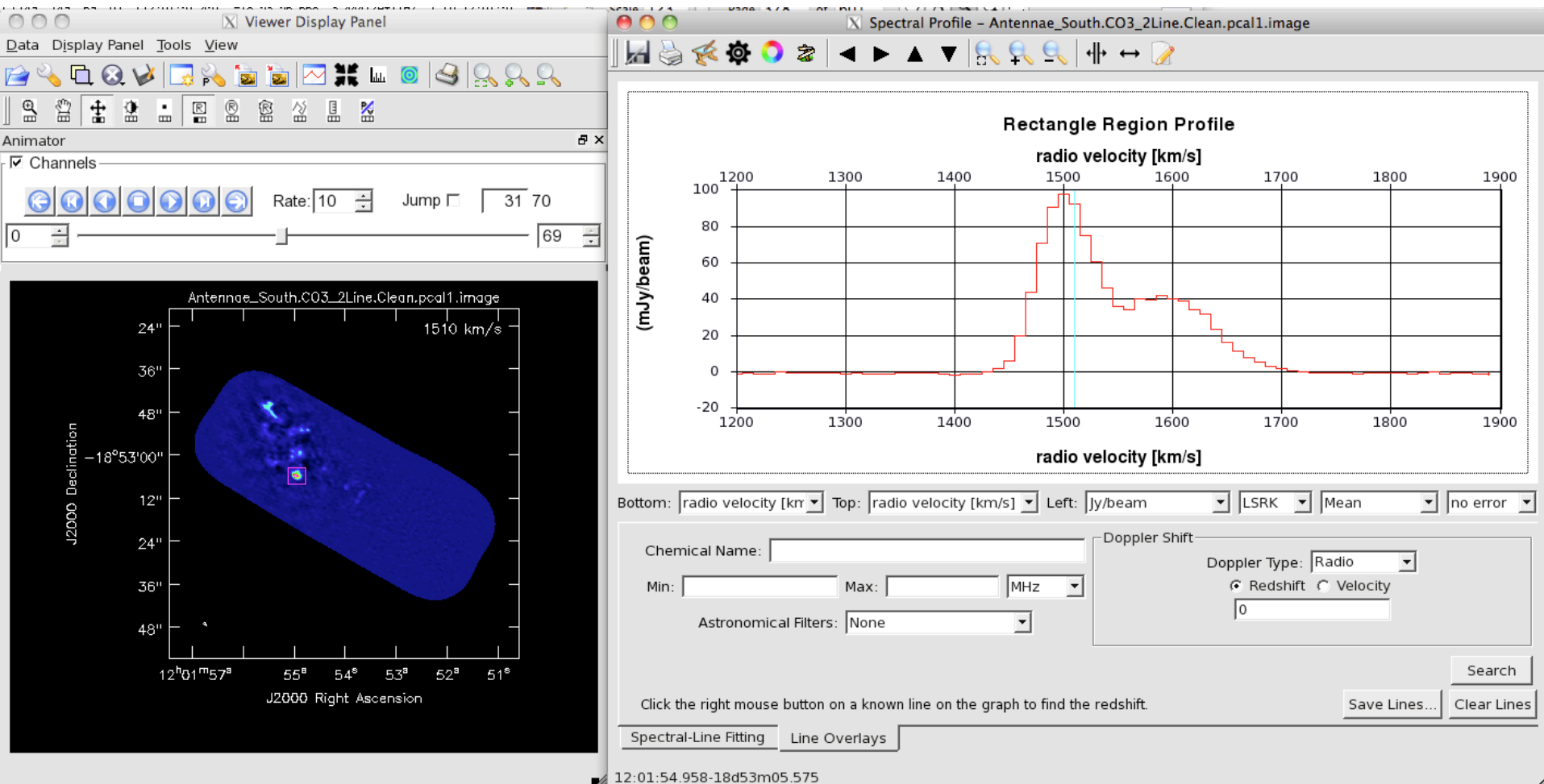
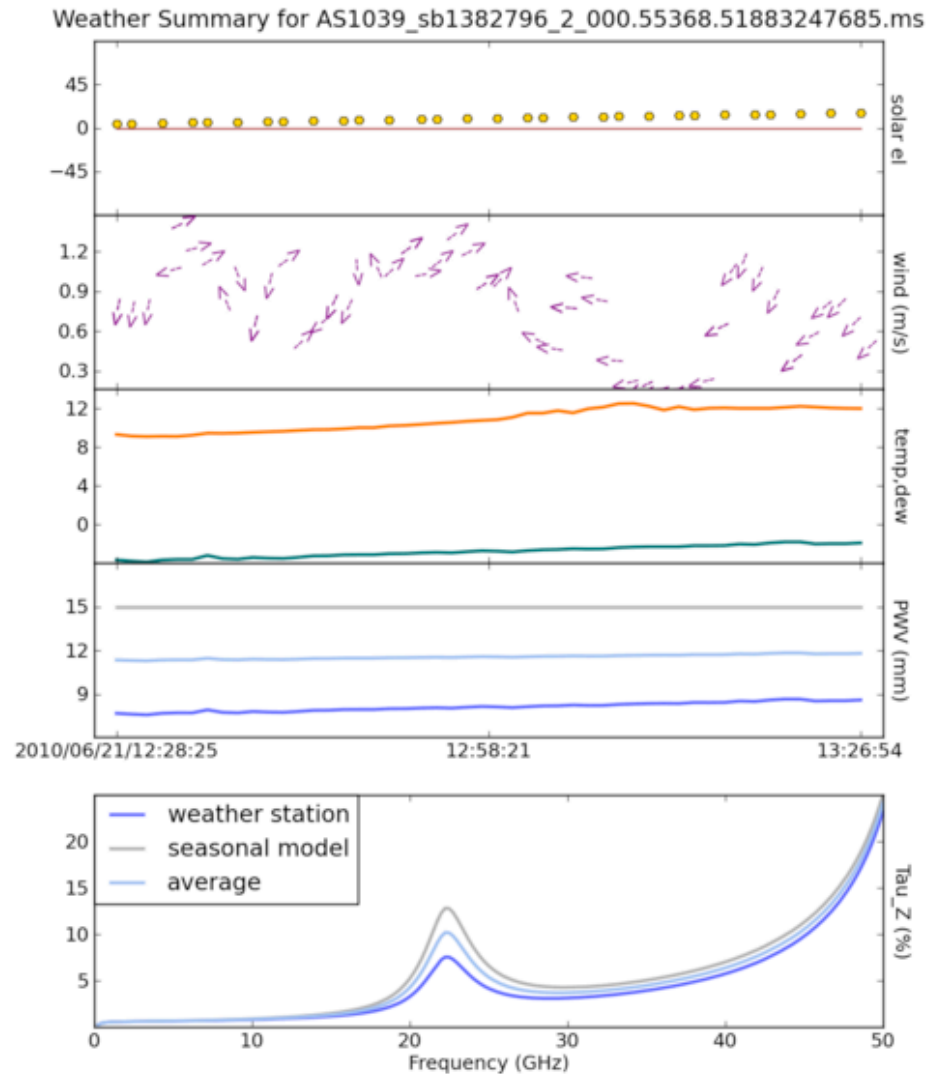


Image Viewer



Plot anything - matplotlib



Data structures

- Observatory data stored in (A)SDM format, CASA uses measurement sets
 - Use **importasdm** for ALMA, **importasdm** or **importevla** for EVLA/JVLA, **importvla** for old VLA, etc.
- Measurement set contains visibilities
Calibration information stored in calibration tables (not in ms)
Images are in CASA image format
- All of these are *directories* which contain the necessary information
 - Copying requires recursive option (**!cp -r**)
- Delete tables using **rmtables('mydata.ms')**
!rm -rf or **os.system('rm -rf mydata.ms')** may also work,
but could leave traces in the cache

Inspect a measurement set (ms)

- Contains visibilities (and flags) stored in MAIN table within table.* files

```
CASA <80>: !ls amazing_data.ms
ANTENNA          POINTING          SYSPower    table.f15      table.f20_TSM0  table.f24_TSM1  table.f8
CALDEVICE        POLARIZATION    table.dat   table.f16      table.f21       table.f25       table.f9
DATA_DESCRIPTION PROCESSOR       table.f1    table.f17      table.f21_TSM1  table.f25_TSM1  table.info
FEED             SORTED_TABLE    table.f10   table.f17_TSM1 table.f22       table.f3        table.lock
FIELD            SOURCE          table.f11   table.f18      table.f22_TSM1  table.f4        WEATHER
FLAG_CMD         SPECTRAL_WINDOW table.f12   table.f19      table.f23       table.f5
HISTORY          STATE           table.f13   table.f2       table.f23_TSM1  table.f6
OBSERVATION      SYSCAL          table.f14   table.f20      table.f24       table.f7
```

- Also contains sub-tables, e.g. FIELD, SOURCE, WEATHER, ...

```
CASA <81>: !ls amazing_data.ms/FIELD
table.dat  table.f0  table.f0i  table.info  table.lock
```


MS MAIN table contents

Inspect with task **browsetable** (or standalone % **casabrowser**)

Table Browser <@nmpost017>

File Edit View Tools Export Help About

amazing_data.ms

	UVW	FLAG	FLAG_CATEGORY	WEIGHT	SIGMA	ANTENNA1	ANTENNA2	ARRAY_ID	DATA
0	[-278.403, ...	[4, 64] Boo...	[0, 0, 0] Boolean	[8.53333e...	[0.000176...	0	1	0	0
1	[2810.11, 2...	[4, 64] Boo...	[0, 0, 0] Boolean	[8.53333e...	[0.000176...	0	2	0	0
2	[426.12, 71...	[4, 64] Boo...	[0, 0, 0] Boolean	[8.53333e...	[0.000176...	0	3	0	0
3	[270.096, ...	[4, 64] Boo...	[0, 0, 0] Boolean	[8.53333e...	[0.000176...	0	4	0	0
4	[-610.975, ...	[4, 64] Boo...	[0, 0, 0] Boolean	[8.53333e...	[0.000176...	0	5	0	0
5	[-1908.48, ...	[4, 64] Boo...	[0, 0, 0] Boolean	[8.53333e...	[0.000176...	0	6	0	0
6	[141.022, 1...	[4, 64] Boo...	[0, 0, 0] Boolean	[8.53333e...	[0.000176...	0	7	0	0
7	[712.44, 94...	[4, 64] Boo...	[0, 0, 0] Boolean	[8.53333e...	[0.000176...	0	8	0	0
8	[-20.6492, ...	[4, 64] Boo...	[0, 0, 0] Boolean	[8.53333e...	[0.000176...	0	9	0	0
9	[989.709, ...	[4, 64] Boo...	[0, 0, 0] Boolean	[8.53333e...	[0.000176...	0	10	0	0
10	[...]	[4, 64] Boo...	[0, 0, 0] Boolean	[8.53333e...	[0.000176...	0	11	0	0

Restore Columns Resize Headers

PAGE NAVIGATION First << [1 / 3633] >> Last 1 Go Loading 1000 rows.

MS MAIN table contents

Inspect with task **browsetable** (or standalone % **casabrowser**)

Table Browser <@nmpost017>

File Edit View Tools Export Help About

amazing_data.ms

	TIME_CENTROID	DATA	WEIGHT_SPECTRUM	MODEL_DATA	CORRECTED_DATA
0	2015-04-19-22:...	[4, 64] Complex	[0, 0] Float	[4, 64] Complex	[4, 64] Complex
1	2015-04-19-22:...	[4, 64] Complex	[0, 0] Float	[4, 64] Complex	[4, 64] Complex
2	2015-04-19-22:...	[4, 64] Complex	[0, 0] Float	[4, 64] Complex	[4, 64] Complex
3	2015-04-19-22:...	[4, 64] Complex	[0, 0] Float	[4, 64] Complex	[4, 64] Complex
4	2015-04-19-22:...	[4, 64] Complex	[0, 0] Float	[4, 64] Complex	[4, 64] Complex
5	2015-04-19-22:...	[4, 64] Complex	[0, 0] Float	[4, 64] Complex	[4, 64] Complex
6	2015-04-19-22:...	[4, 64] Complex	[0, 0] Float	[4, 64] Complex	[4, 64] Complex
7	2015-04-19-22:...	[4, 64] Complex	[0, 0] Float	[4, 64] Complex	[4, 64] Complex
8	2015-04-19-22:...	[4, 64] Complex	[0, 0] Float	[4, 64] Complex	[4, 64] Complex
9	2015-04-19-22:...	[4, 64] Complex	[0, 0] Float	[4, 64] Complex	[4, 64] Complex
10	2015-04-19-22:...	[4, 64] Complex	[0, 0] Float	[4, 64] Complex	[4, 64] Complex

amazing_data.ms[0, 21] = Complex Array of size [4 64].

	0	
0	(1.38879e-05,0.00067147)	(-6.54117e-0
1	(3.43195e-05,0.000646329)	(0.00045081
2	(1.56872e-05,-0.000113082)	(0.00013204,
3	(-0.000342448,0.000368312)	(-0.00042331

Restore Columns Resize Headers

PAGE NAVIGATION First << [1 / 3633] >> Last 1 Go Loading 1000 rows.

Close Close All

MS MAIN table contents

Inspect with task **browsetable** (or standalone % **casabrowser**)

Table Browser <@nmpost017>

File Edit View Tools Export Help About

amazing_data.ms

	TIME_CENTROID	DATA	WEIGHT_SPECTRUM	MODEL_DATA	CORRECTED_DATA
0	2015-04-19-22:...	[4, 64] Complex	[0, 0] Float	[4, 64] Complex	[4, 64] Complex
1	2015-04-19-22:...	[4, 64] Complex	[0, 0] Float	[4, 64] Complex	[4, 64] Complex
2	2015-04-19-22:...	[4, 64] Complex	[0, 0] Float	[4, 64] Complex	[4, 64] Complex
3	2015-04-19-22:...	[4, 64] Complex	[0, 0] Float	[4, 64] Complex	[4, 64] Complex
4	2015-04-19-22:...	[4, 64] Complex	[0, 0] Float	[4, 64] Complex	[4, 64] Complex
5	2015-04-19-22:...	[4, 64] Complex	[0, 0] Float	[4, 64] Complex	[4, 64] Complex
6	2015-04-19-22:...	[4, 64] Complex	[0, 0] Float	[4, 64] Complex	[4, 64] Complex
7	2015-04-19-22:...	[4, 64] Complex	[0, 0] Float	[4, 64] Complex	[4, 64] Complex
8	2015-04-19-22:...	[4, 64] Complex	[0, 0] Float	[4, 64] Complex	[4, 64] Complex
9	2015-04-19-22:...	[4, 64] Complex	[0, 0] Float	[4, 64] Complex	[4, 64] Complex
10	2015-04-19-22:...	[4, 64] Complex	[0, 0] Float	[4, 64] Complex	[4, 64] Complex

amazing_data.ms[0, 21] = Complex Array of size [4 64].

	0	
0	(1.38879e-05,0.00067147)	(-6.54117e-0
1	(3.43195e-05,0.000646329)	(0.00045081
2	(1.56872e-05,-0.000113082)	(0.00013204,
3	(-0.000342448,0.000368312)	(-0.00042331

Restore Columns Resize Headers

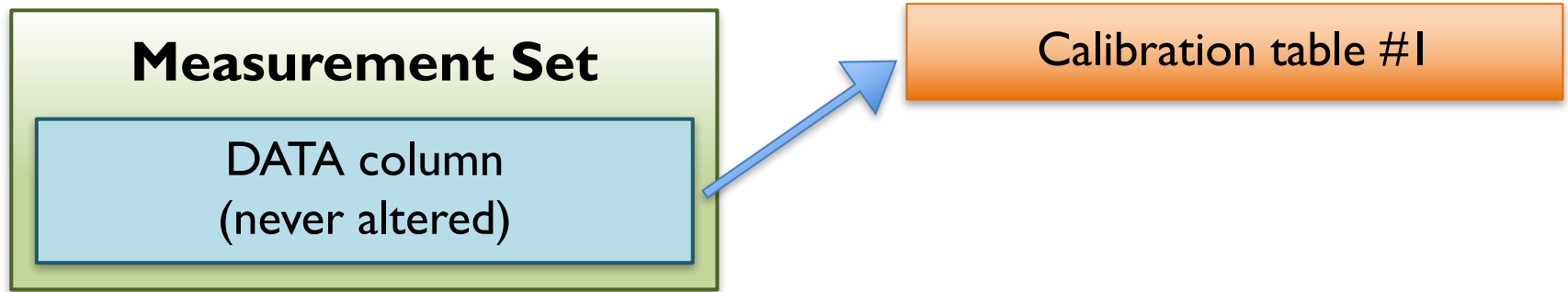
PAGE NAVIGATION First << [1 / 3633] >> Last 1 Go Loading 1000 rows.

MS columns & calibration tables

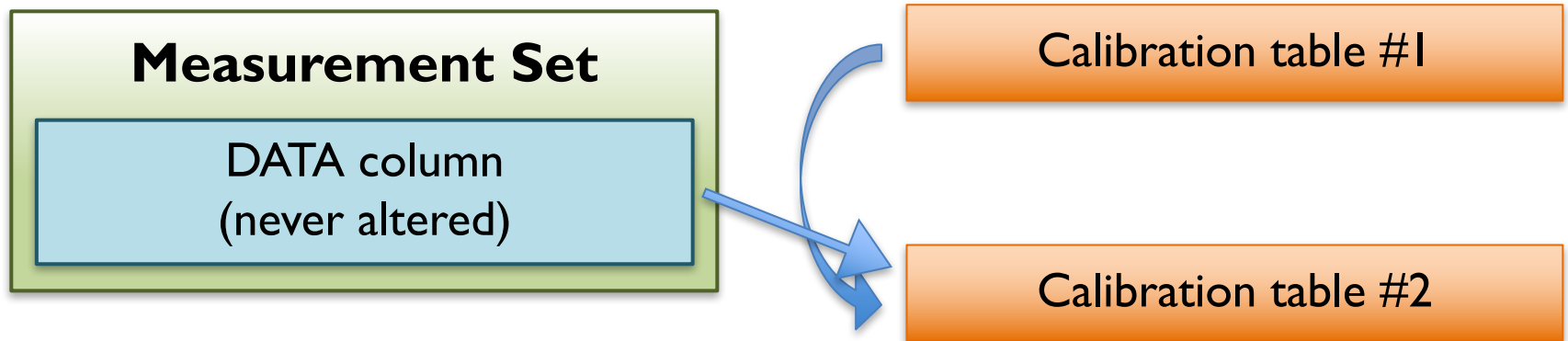
Measurement Set

DATA column
(never altered)

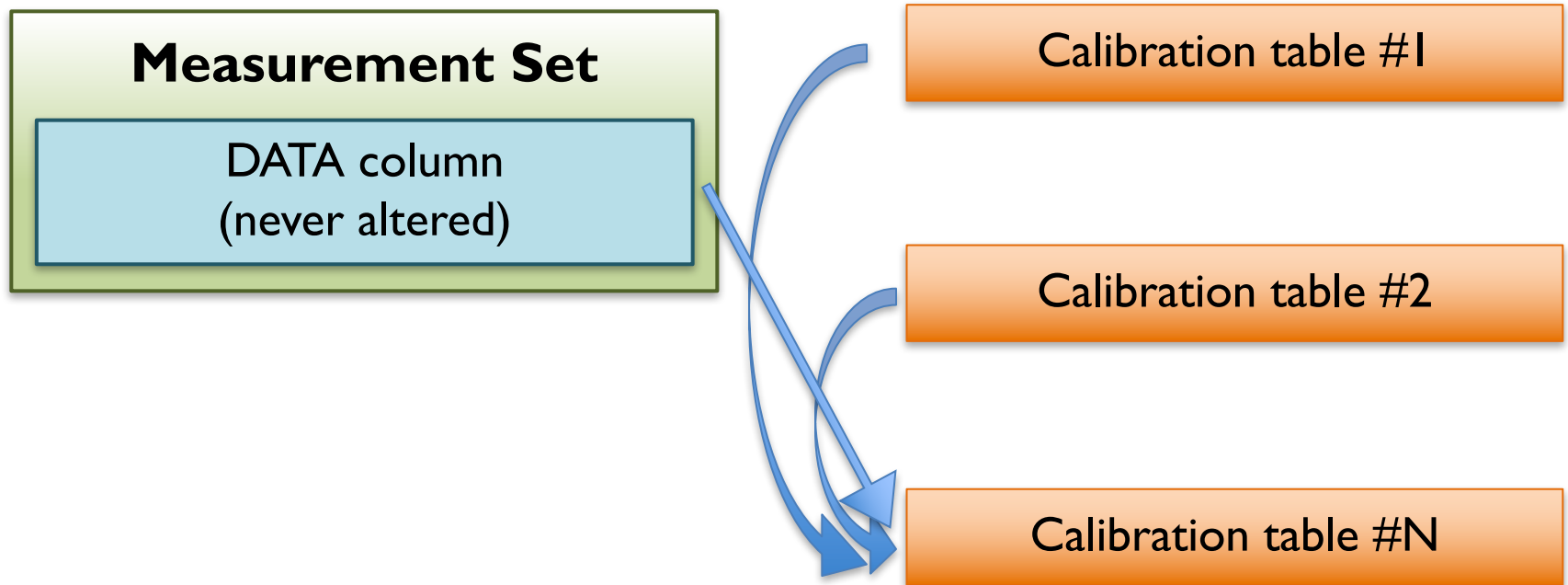
MS columns & calibration tables



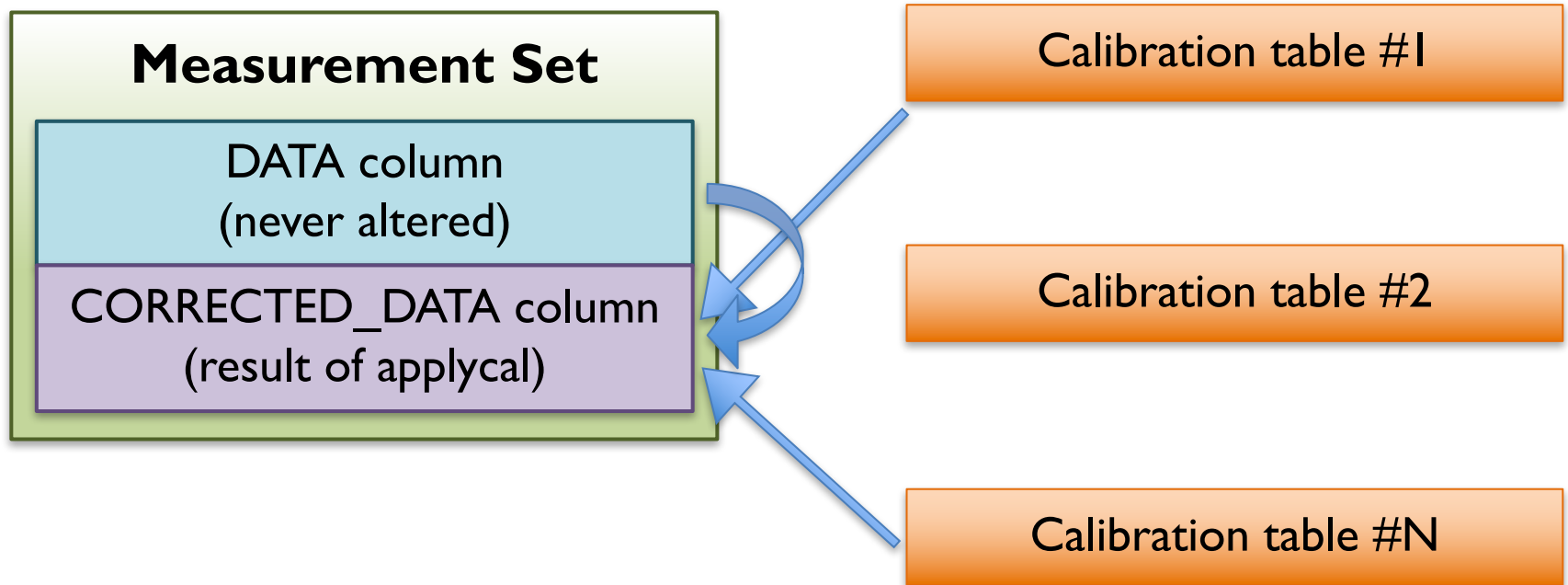
MS columns & calibration tables



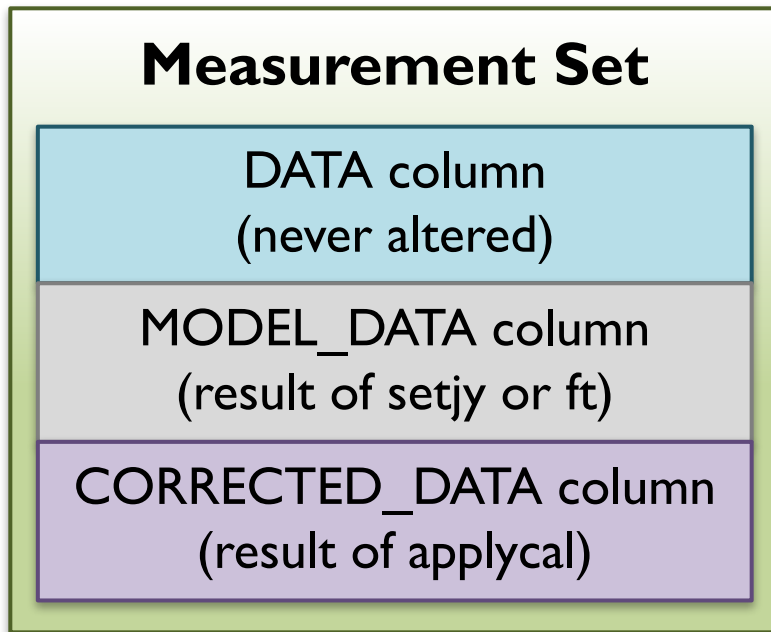
MS columns & calibration tables



MS columns & calibration tables



MS columns & calibration tables



(note: data size tripled)

Calibration table #1

Calibration table #2

Calibration table #N

MS data selection syntax

- You can select subset of visibilities to perform actions on:
 - Antennas, baselines, frequencies, time, polarization, etc.
- The standard CASA selection syntax is the following:
 - Use tilde (~) for inclusive range, e.g. **spw='0~3'**
 - Use comma (,) for separator, e.g. **spw='0~3,7,11'**
 - Use colon (:) for spw channelization, e.g. **spw='0:0~40,3:20~40'**
 - Use semicolon (;) for spw channel separator, e.g. **spw='0:0~10;20;25'**
 - Use asterisk (*) for wildcard, e.g. **field='3C*'**
 - Use exclamation mark (!) for omission, e.g. **antenna='!ea05'**
 - Use ampersands (&) for baselines, e.g. **antenna='ea09&ea11'**
 - Use less than (<) or greater than (>) for selection, e.g. **uvrange='<1000m'**



For full syntax see CASAdocs

MS data selection syntax: Examples

- **field** (*spatial*)
 - String with source name or field ID (checks former 1st)
 - Beware field names that are integers
 - Examples: **field='J331+305'** ; **field='3C*'** ; **field='0,1,4~5'**
- **spw** (*spectral*)
 - String with spectral window ID plus channels
 - Examples: **spw='0:10~20;45,4~5:35~45;50~70'** ; **spw='*:10~80'** ;
spw='J421MHz:10~20;50,5:1.6~1.7GHz'

MS data selection syntax: Examples

- **timerange** (*spatial*)
 - String with date/time range in format T0~T1
 - Can give T0+dT, where missing parts of T1 default to T0
 - Example: **timerange = '2014/10/21/01:00:00~06:30:00'**
- **antenna**
 - String with antenna name or ID (checks former 1st)
 - Beware VLA name I-27, these have ID's 0-26
 - & = CC only , && = CC+AC , &&& = AC only
 - Examples: **antenna = '1~5,8'** ; **antenna='!ea01&ea10'** ;
antenna='ea05&&&'

MS data selection syntax: Examples

- **scan** – the scan numbers (an execution sequence)
 - e.g. **scan='3~14'**
- **correlation** – polarization products
 - e.g. **correlation='RR,LL,LR'**
- **uvrange** – select on uv range
 - e.g. **uvrange='30~3000m' ; uvrange='<1000m'**

Ongoing CASA Developments

- **CARTA**: new image cube viewer (developed by ASIAA, beta release available now on <http://cartavis.github.io/>)
- **Calibration Library** (more flexible application of calibration tables through a provided file, single step applycal)
- **tclean** fully replacing clean (plus new development such as n - σ thresholds, autoboxing)
- **New Imaging Algorithms** → Sanjay Bhatnagar's talk
- **Parallelization** for calibration and imaging (using multi-MS and concatenated image cubes)
- **Imaging Pipeline for general VLA, polarization calibration** (developed now for VLASS)
- **New statwt** (channelized weights, robust statistics, time/frequency windows)
- **Fringe fitting/VLBI** (developed by ASTRON)
- **CASAdocs** transition
- **plotcal to plotms transition**
- **Continuum subtraction improvements** (spw combination, phase center shifts, better infrastructure)



Need a person to help?

- Helpdesk: <https://help.nrao.edu> <https://help.almascience.org>
- Questions can cover:
 - CASA problems
 - Calibration/imaging questions
 - Submit bug reports and suggestions
 - (Other issues, e.g. account/log-in info, proposal submission, etc.)
- When submitting a ticket, provide as much detail as possible:
 - CASA version
 - Operating system
 - Commands entered
 - Project ID (if relevant)
 - Scripts you followed from CASAGuides (if relevant)

CASA documentation and web resources

<http://casa.nrao.edu>

- CASAdocs (documentation)
- CASAguides (tutorials)
- Helpdesk
- NRAO science user forum
- Sign up to mailing lists to receive updates

