Introduction to CASA

Brian Svoboda (NRAO)

CASA
Common Astronomy Software Applications

Overview of this talk

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

General description

- CASA: Common Astronomy Software Applications http://casa.nrao.edu
  - Post-processing package for ALMA and VLA, both interferometric and single dish
  - Other telescopes also use it (e.g. Nobeyama, ATCA, VLBI)
  - Developed at NRAO (lead), ESO, NAOJ, CSIRO/ATNF, ASIAA, and ASTRON

- Code is C++ (fast) called by a Python interface (easy access & scripting)
- Many tasks and a lot of tools
- Automated calibration (and imaging) pipelines for ALMA and VLA
- Contributions from our Algorithm Research Development Group

- Latest CASA release is version 6.5 [6.6 next release]
  - CASA 5.X = python 2, CASA 6.X = python 3
- But we use CASA 6.4.1 for this workshop (a patch of 6.4: package which bundles CASA with the pipeline, validated for use by the VLA)

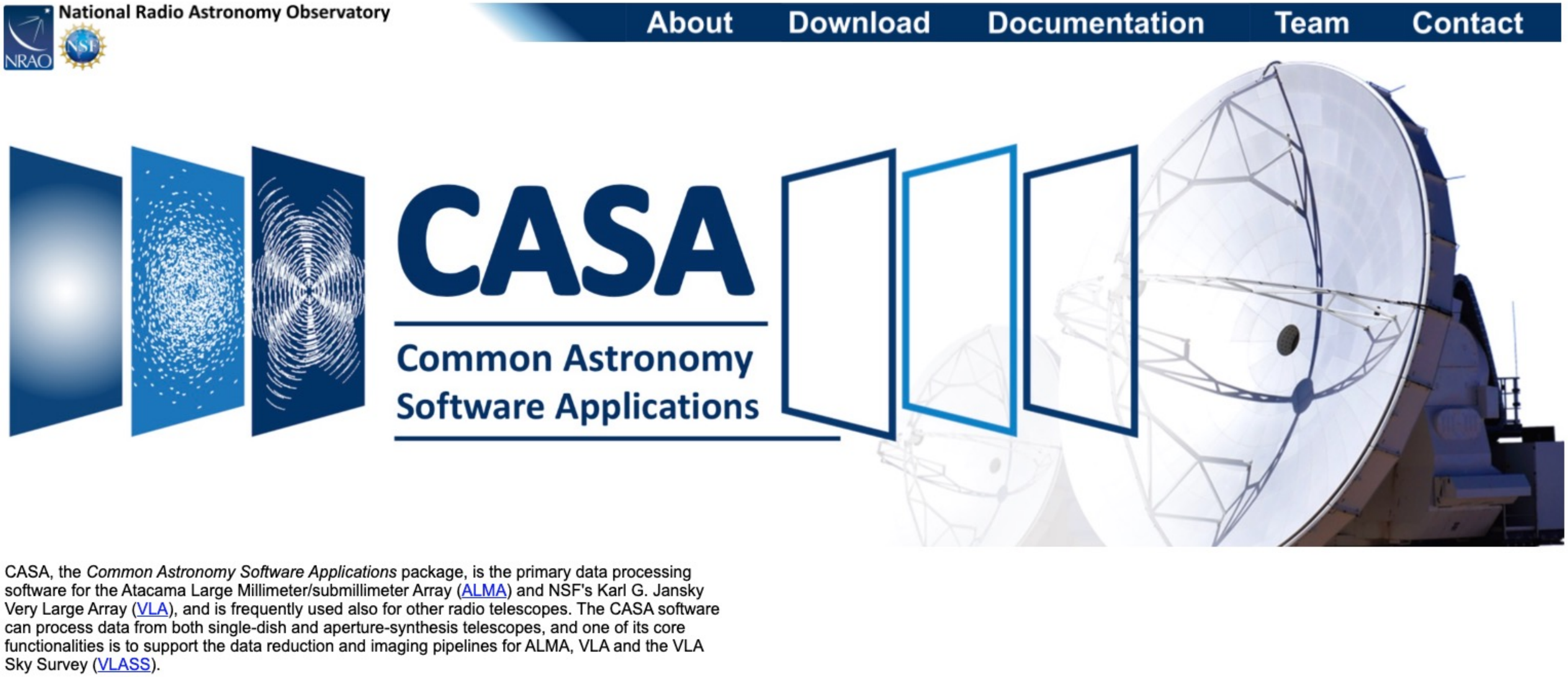
CASA releases

- New releases generally twice a year
- Pipelines usually released once a year, recently in CASA patches (i.e., 6.4.1)
  - ALMA: October 1, VLA a bit later

- Latest version CASA 6.6 is internally tested on:
  - Red Hat Linux 7 and 8
  - macOS 10 & 11 (Intel), 12 & 13 (Arm)
  - Ubuntu 18.04 and 20.04
  - also works on other Linux systems, but not tested in house

CASA documentation and web resources

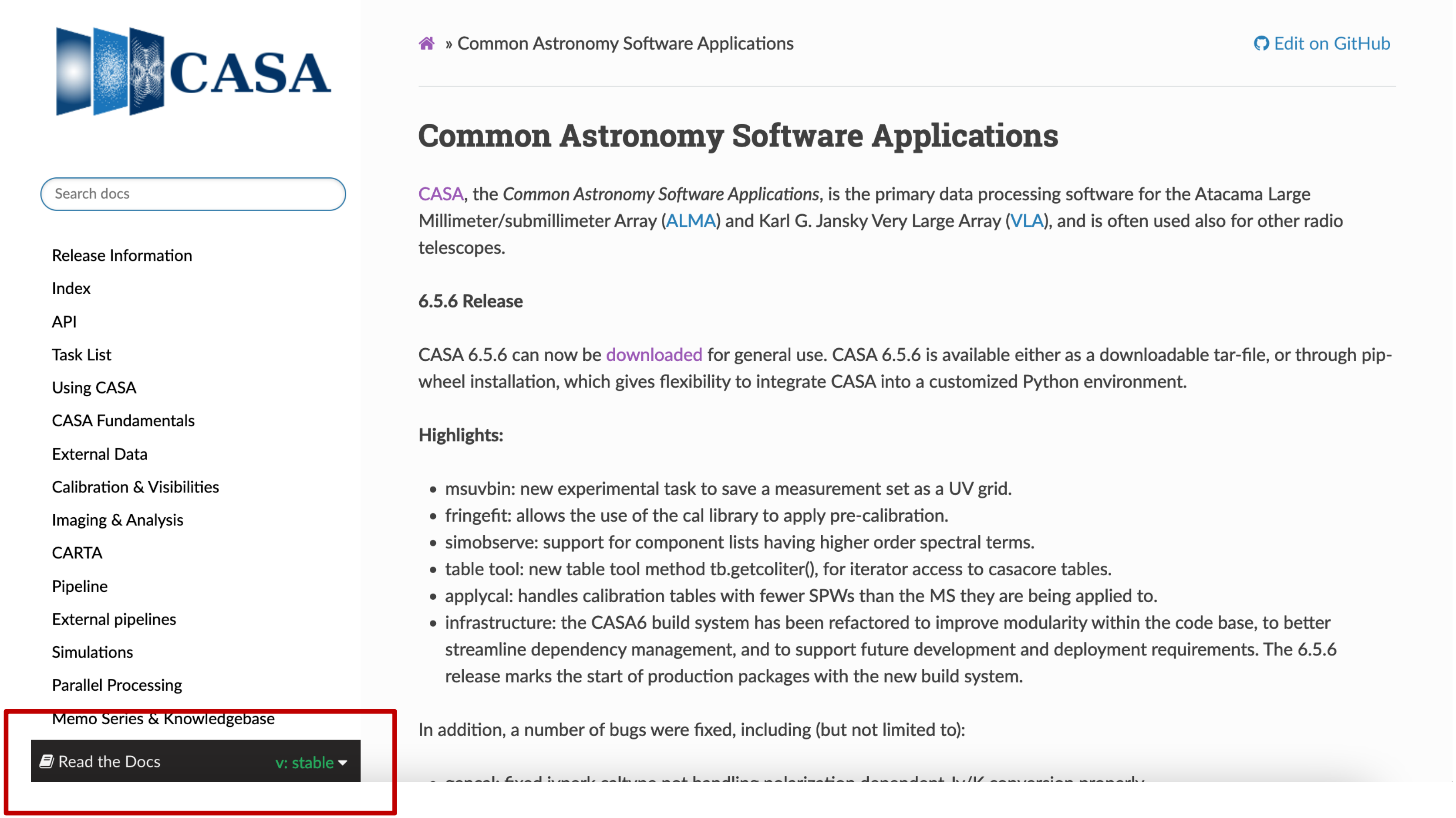
CASA Homepage http://casa.nrao.edu



CASA, the *Common Astronomy Software Applications* package, is the primary data processing software for the Atacama Large Millimeter/submillimeter Array (ALMA) and NSF's Karl G. Jansky Very Large Array (VLA), and is frequently used also for other radio telescopes. The CASA software can process data from both single-dish and aperture-synthesis telescopes, and one of its core functionalities is to support the data reduction and imaging pipelines for ALMA, VLA and the VLA Sky Survey (VLASS).

CASA documentation and web resources

Documentation: CASAdocs https://casadocs.readthedocs.io



CASA

Search docs

Release Information

Index

API

Task List

Using CASA

CASA Fundamentals

External Data

Calibration & Visibilities

Imaging & Analysis

CARTA

Pipeline

External pipelines

Simulations

Parallel Processing

Memo Series & Knowledgebase

Read the Docs v: stable

» Common Astronomy Software Applications

Edit on GitHub

Common Astronomy Software Applications

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

6.5.6 Release

CASA 6.5.6 can now be downloaded for general use. CASA 6.5.6 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:

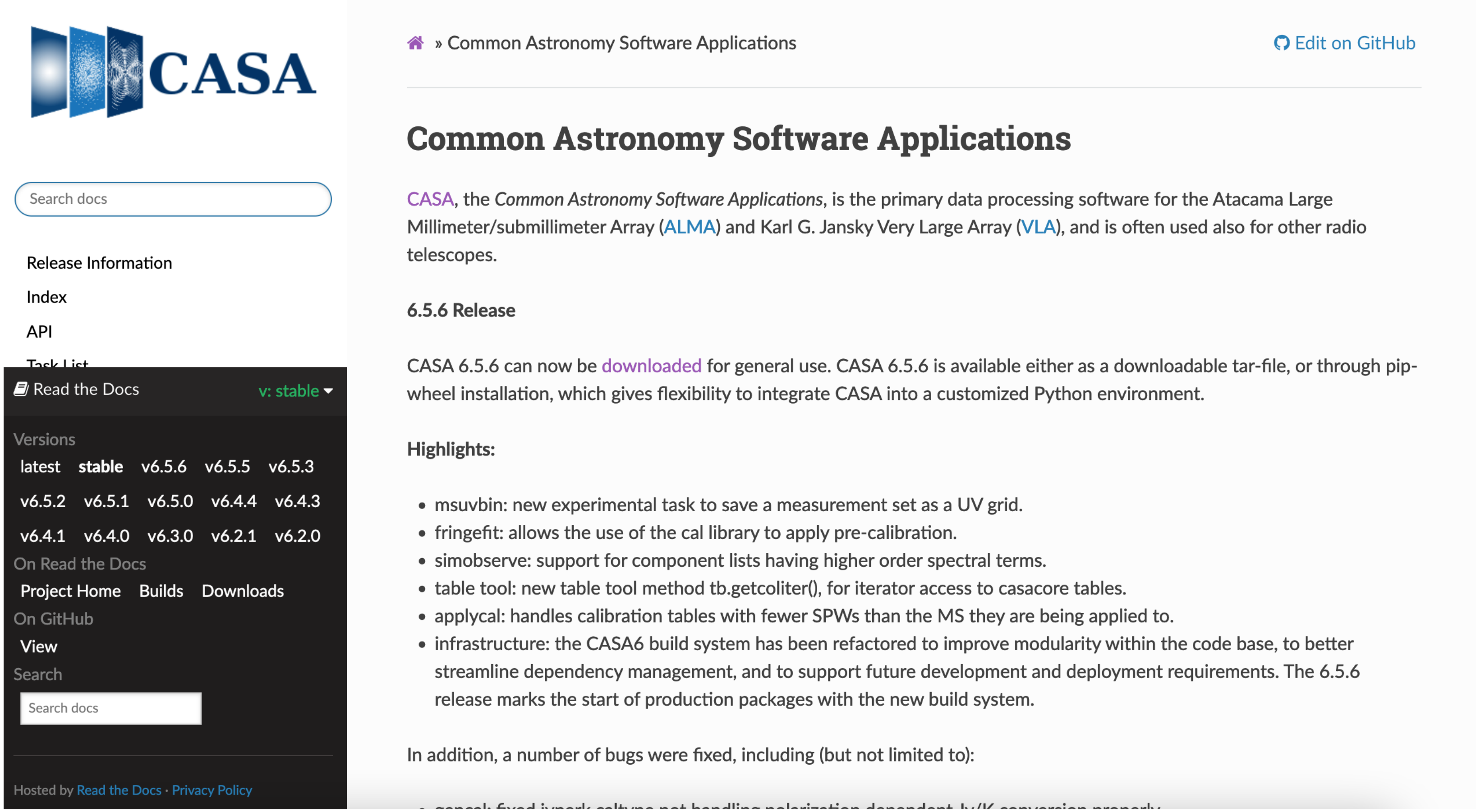
- msuvbin: new experimental task to save a measurement set as a UV grid.
- fringefit: allows the use of the cal library to apply pre-calibration.
- simobserve: support for component lists having higher order spectral terms.
- table tool: new table tool method tb.getcoliter(), for iterator access to casacore tables.
- applycal: handles calibration tables with fewer SPWs than the MS they are being applied to.
- infrastructure: the CASA6 build system has been refactored to improve modularity within the code base, to better streamline dependency management, and to support future development and deployment requirements. The 6.5.6 release marks the start of production packages with the new build system.

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

- gencal: fixed jyperk caltype not handling polarization-dependent Jy/K conversion properly

CASA documentation and web resources

Documentation: CASAdocs https://casadocs.readthedocs.io



CASA

Search docs

Release Information

Index

API

Task List

Read the Docs v: stable

Versions

latest stable v6.5.6 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

On Read the Docs

Project Home Builds Downloads

On GitHub

View

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.

6.5.6 Release

CASA 6.5.6 can now be downloaded for general use. CASA 6.5.6 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:

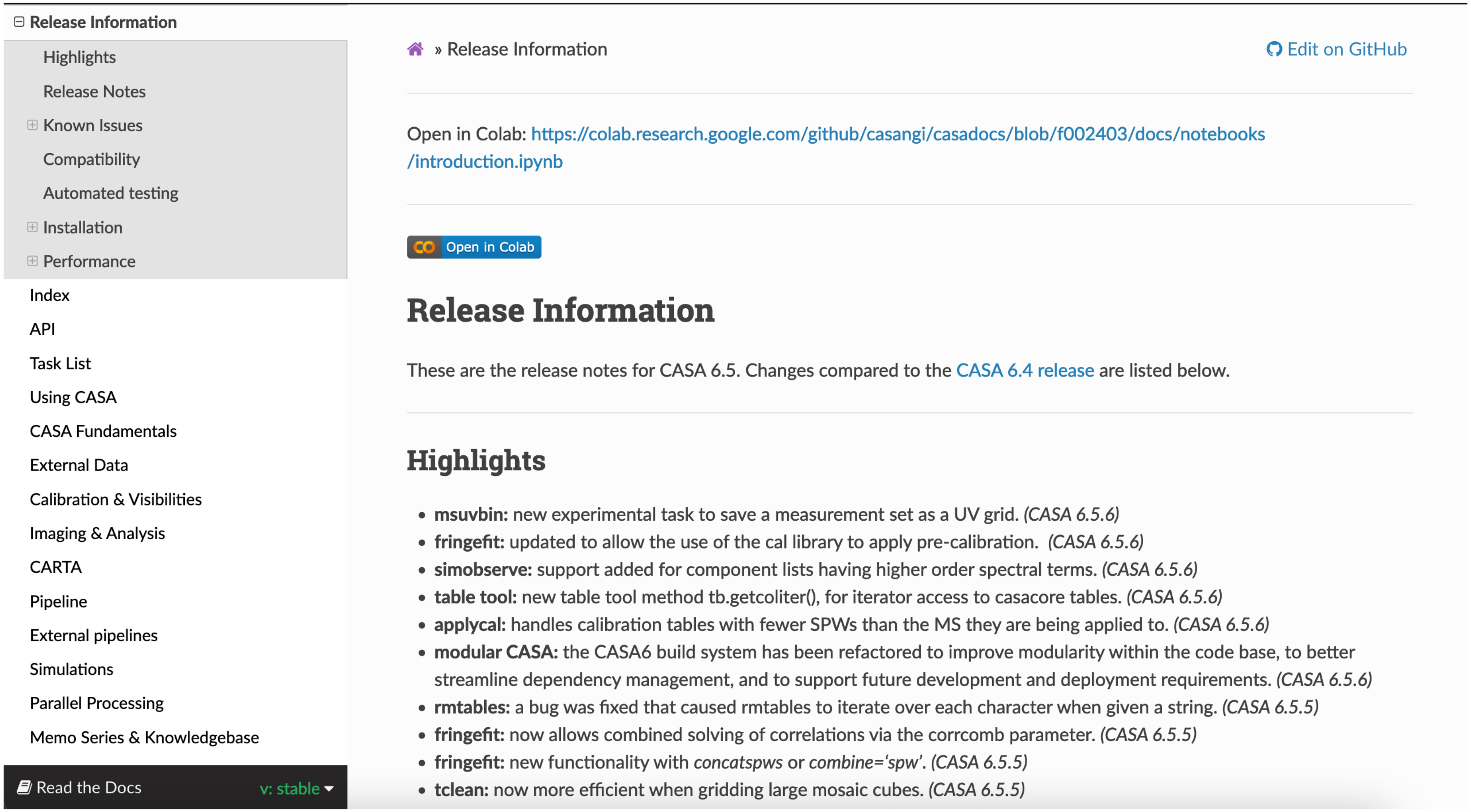
- msuvbin: new experimental task to save a measurement set as a UV grid.
- fringefit: allows the use of the cal library to apply pre-calibration.
- simobserve: support for component lists having higher order spectral terms.
- table tool: new table tool method tb.getcoliter(), for iterator access to casacore tables.
- applycal: handles calibration tables with fewer SPWs than the MS they are being applied to.
- infrastructure: the CASA6 build system has been refactored to improve modularity within the code base, to better streamline dependency management, and to support future development and deployment requirements. The 6.5.6 release marks the start of production packages with the new build system.

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

- gencal: fixed jyperk caltype not handling polarization dependent Jy/K conversion properly

CASA documentation and web resources

Documentation: CASAdocs https://casadocs.readthedocs.io



Open in Colab: https://colab.research.google.com/github/casangi/casadocs/blob/f002403/docs/notebooks/introduction.ipynb

Release Information

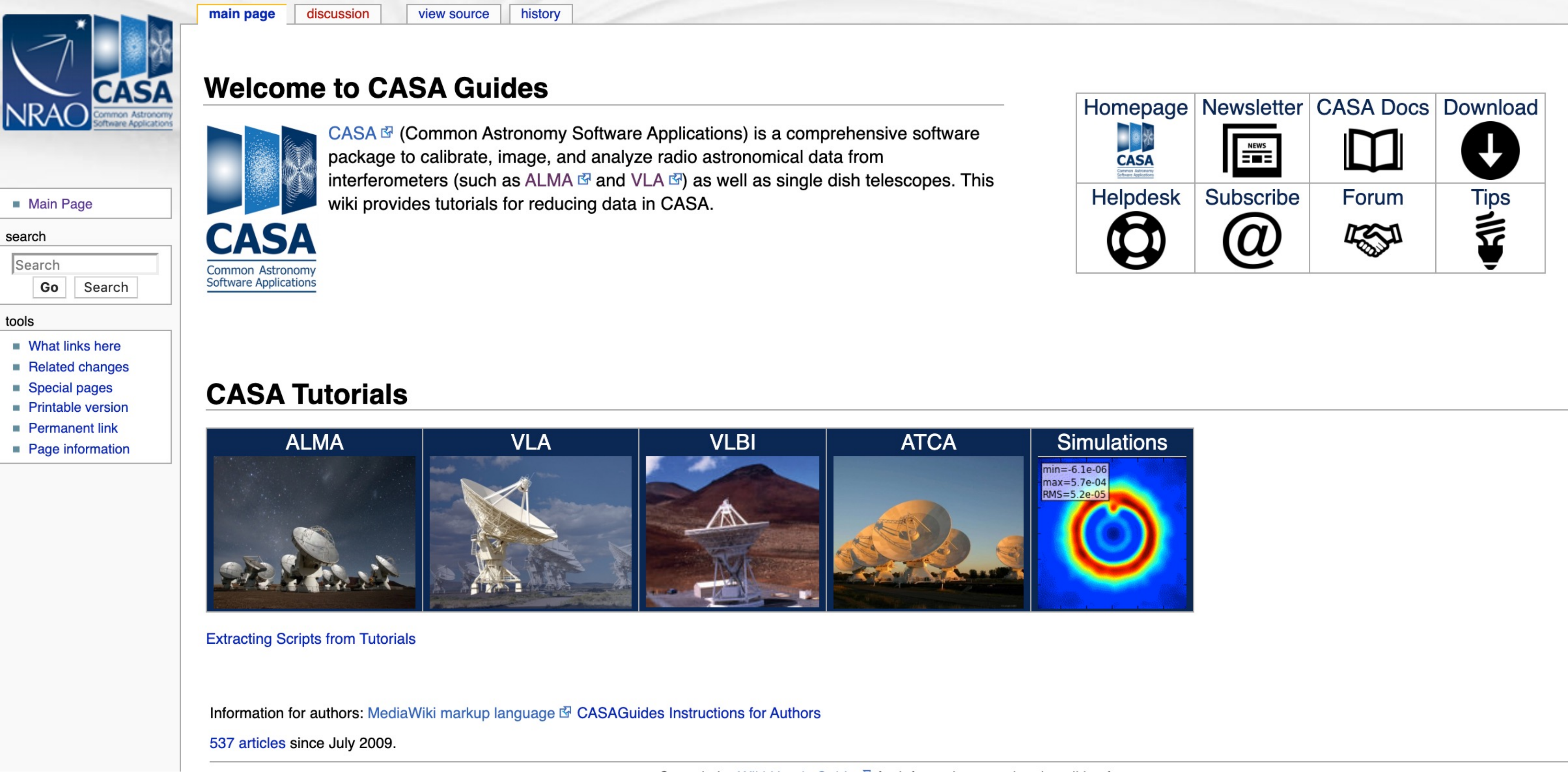
These are the release notes for CASA 6.5. Changes compared to the CASA 6.4 release are listed below.

Highlights

- **msuvbin:** new experimental task to save a measurement set as a UV grid. *(CASA 6.5.6)*
- **fringefit:** updated to allow the use of the cal library to apply pre-calibration. *(CASA 6.5.6)*
- **simobserve:** support added for component lists having higher order spectral terms. *(CASA 6.5.6)*
- **table tool:** new table tool method tb.getcoliter(), for iterator access to casacore tables. *(CASA 6.5.6)*
- **applycal:** handles calibration tables with fewer SPWs than the MS they are being applied to. *(CASA 6.5.6)*
- **modular CASA:** the CASA6 build system has been refactored to improve modularity within the code base, to better streamline dependency management, and to support future development and deployment requirements. *(CASA 6.5.6)*
- **rmtables:** a bug was fixed that caused rmtables to iterate over each character when given a string. *(CASA 6.5.5)*
- **fringefit:** now allows combined solving of correlations via the corrcomb parameter. *(CASA 6.5.5)*
- **fringefit:** new functionality with *concatspws* or *combine='spw'*. *(CASA 6.5.5)*
- **tclean:** now more efficient when gridding large mosaic cubes. *(CASA 6.5.5)*

CASA documentation and web resources

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



main page | discussion | view source | history

- Main Page

search

Search

Go Search

tools

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

Welcome to CASA Guides

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

Homepage	Newsletter	CASA Docs	Download
Helpdesk	Subscribe	Forum	Tips

CASA Tutorials

ALMA	VLA	VLBI	ATCA	Simulations

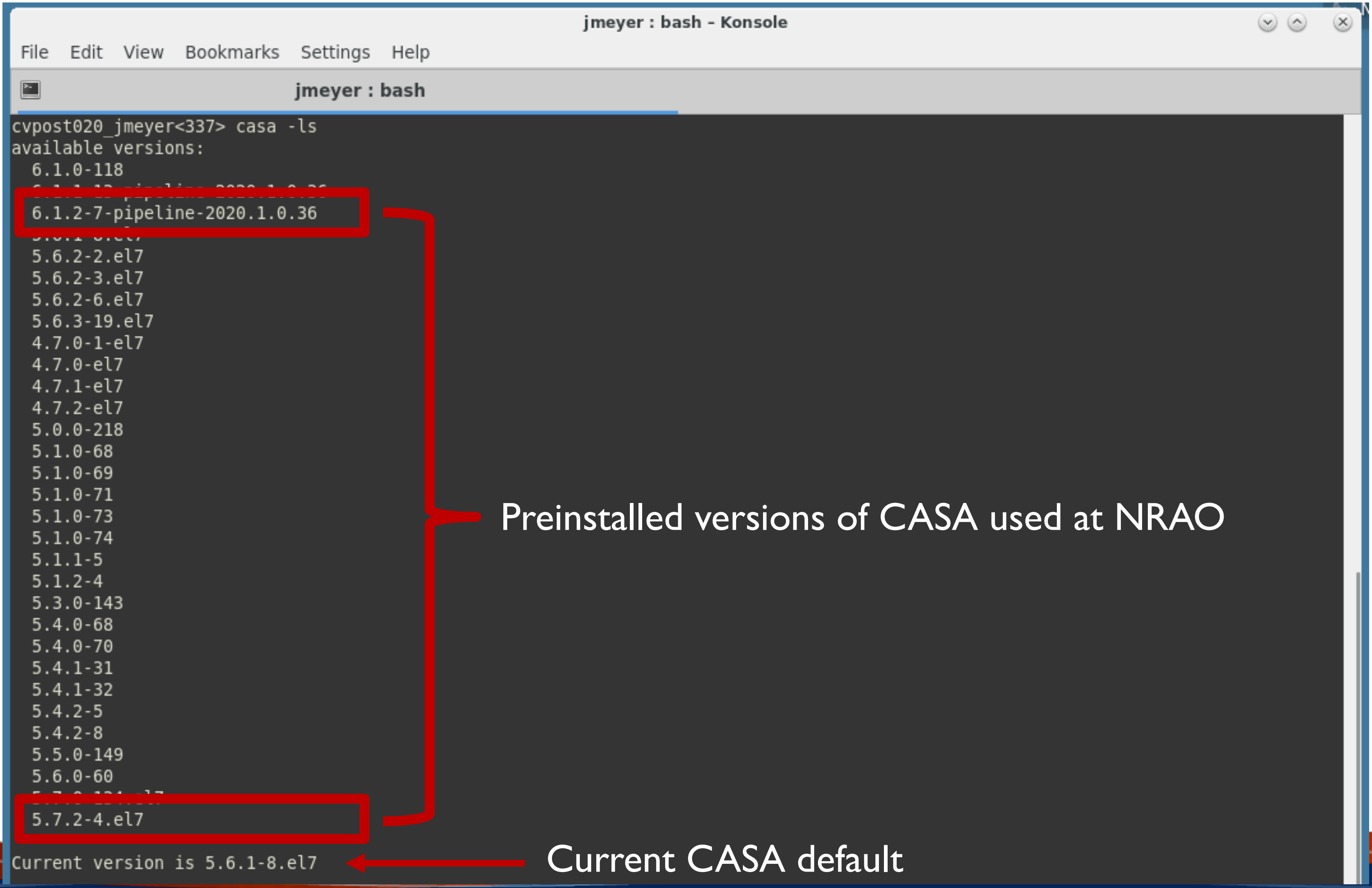
Extracting Scripts from Tutorials

Information for authors: MediaWiki markup language CASAGuides Instructions for Authors

537 articles since July 2009.

Starting Monolithic CASA

- List available pre-installed CASA options: **casa –ls**



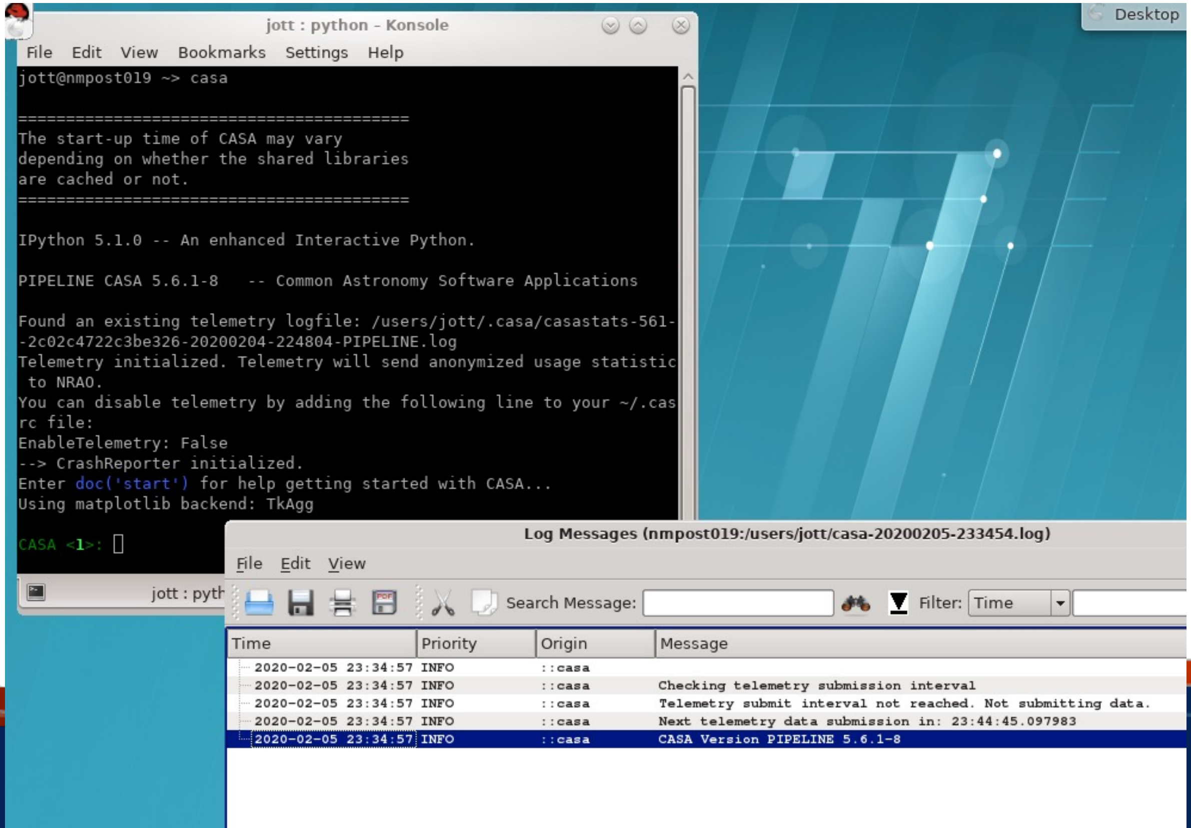
```
cvpost020_jmeyer<337> casa -ls
available versions:
  6.1.0-118
  6.1.1-12-pipeline-2020.1.0.36
  6.1.2-7-pipeline-2020.1.0.36
  5.6.1-8.el7
  5.6.2-2.el7
  5.6.2-3.el7
  5.6.2-6.el7
  5.6.3-19.el7
  4.7.0-1-el7
  4.7.0-el7
  4.7.1-el7
  4.7.2-el7
  5.0.0-218
  5.1.0-68
  5.1.0-69
  5.1.0-71
  5.1.0-73
  5.1.0-74
  5.1.1-5
  5.1.2-4
  5.3.0-143
  5.4.0-68
  5.4.0-70
  5.4.1-31
  5.4.1-32
  5.4.2-5
  5.4.2-8
  5.5.0-149
  5.6.0-60
  5.7.0-134.el7
  5.7.2-4.el7

Current version is 5.6.1-8.el7
```

Preinstalled versions of CASA used at NRAO

Current CASA default

Starting Monolithic CASA

- Start CASA from the UNIX shell: **casa –r 6.4.1-28-pipeline-2022.2.0.68**
- (with pipeline: casa –r 6.4.1-28-pipeline-2022.2.0.68 --pipeline; in parallel mpicasa <path>/casa)
- Session logging:
  - ipython-TIMESTAMP.log IPython command history
  - casapy-TIMESTAMP.log CASA logger messages (the content also appears in the Logger GUI)
- Crash reporter by default, opt-out options



```
jott@nmpost019 ~> 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.

PIPELINE CASA 5.6.1-8    -- Common Astronomy Software Applications

Found an existing telemetry logfile: /users/jott/.casa/casastats-561-
-2c02c4722c3be326-20200204-224804-PIPELINE.log
Telemetry initialized. Telemetry will send anonymized usage statistic
 to NRAO.
You can disable telemetry by adding the following line to your ~/.cas
rc file:
EnableTelemetry: False
--> CrashReporter initialized.
Enter doc('start') for help getting started with CASA...
Using matplotlib backend: TkAgg

CASA <1>: 
```

Log Messages (nmpost019:/users/jott/casa-20200205-233454.log)

```
Time                 Priority  Origin   Message
2020-02-05 23:34:57 INFO      ::casa
2020-02-05 23:34:57 INFO      ::casa   Checking telemetry submission interval
2020-02-05 23:34:57 INFO      ::casa   Telemetry submit interval not reached. Not submitting data.
2020-02-05 23:34:57 INFO      ::casa   Next telemetry data submission in: 23:44:45.097983
2020-02-05 23:34:57 INFO      ::casa   CASA Version PIPELINE 5.6.1-8
```

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)

- Python pitfalls
  - Indentation matters, used for loops & conditions (%cpaste)
  - Indices start from 0 and run to n-1

CASA tasks, tools, and applications

Tasks

- High-level functionality (set parameters and type go; script)
- These are what you will probably use the most

Tools

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

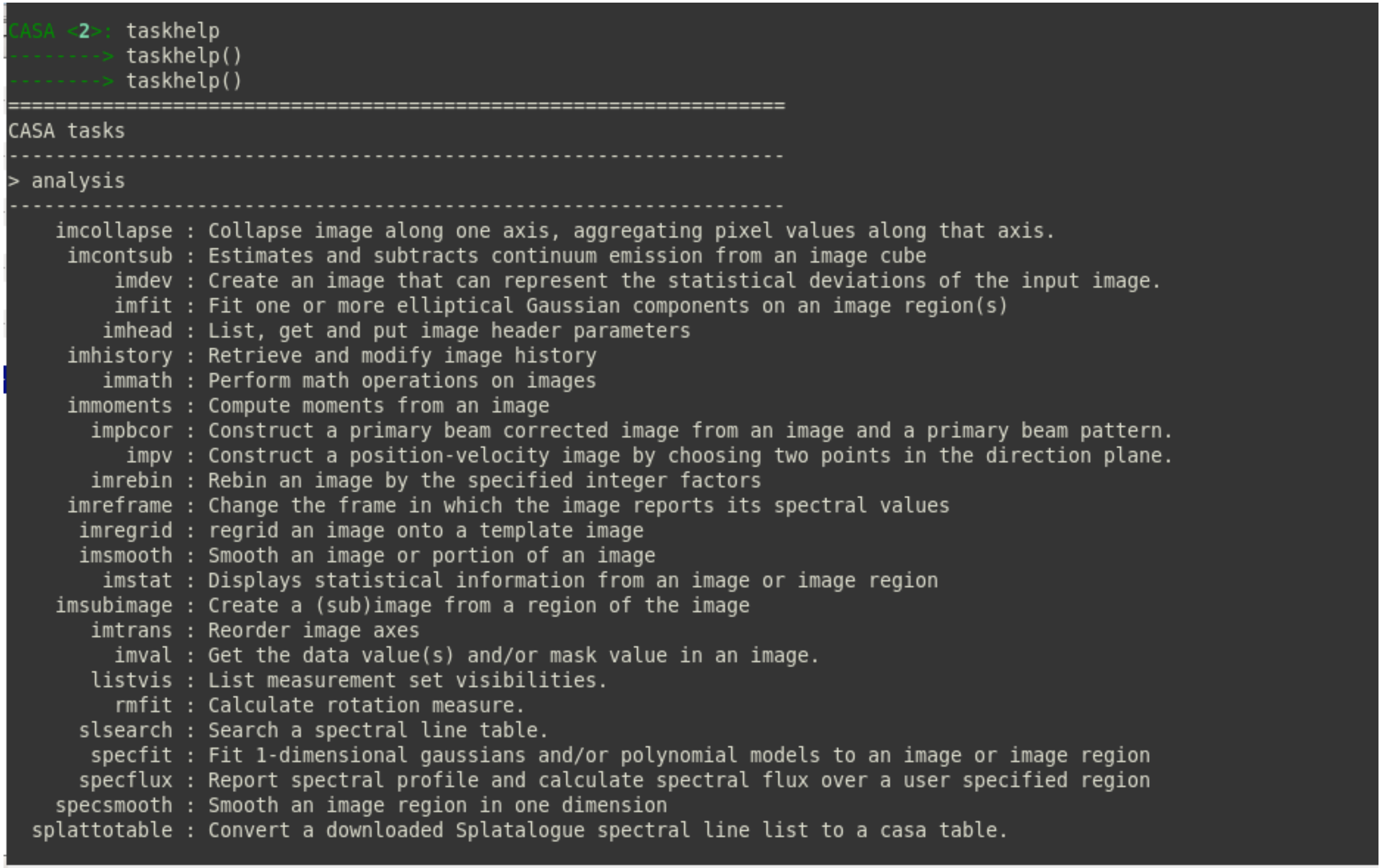
Applications

- Typically used to view, inspect, and edit data (MS, caltables, images)
- Can be invoked inside CASA (or as standalone programs, CASA 5)

Find the right task

To see an organized list with short summaries, type:

taskhelp



```
CASA <2>: taskhelp
--------> taskhelp()
--------> taskhelp()
=================================================================
CASA tasks
-----------------------------------------------------------------
> analysis
-----------------------------------------------------------------
    imcollapse : Collapse image along one axis, aggregating pixel values along that axis.
     imcontsub : Estimates and subtracts continuum emission from an image cube
         imdev : Create an image that can represent the statistical deviations of the input image.
         imfit : Fit one or more elliptical Gaussian components on an image region(s)
        imhead : List, get and put image header parameters
     imhistory : Retrieve and modify image history
        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.
          impv : Construct a position-velocity image by choosing two points in the direction plane.
       imrebin : Rebin an image by the specified integer factors
     imreframe : Change the frame in which the image reports its spectral values
      imregrid : regrid an image onto a template image
      imsmooth : Smooth an image or portion of an image
        imstat : Displays statistical information from an image or image region
    imsubimage : Create a (sub)image from a region of the image
       imtrans : Reorder image axes
         imval : Get the data value(s) and/or mask value in an image.
       listvis : List measurement set visibilities.
         rmfit : Calculate rotation measure.
      slsearch : Search a spectral line table.
       specfit : Fit 1-dimensional gaussians and/or polynomial models to an image or image region
      specflux : Report spectral profile and calculate spectral flux over a user specified region
    specsmooth : Smooth an image region in one dimension
  splattotable : Convert a downloaded Splatalogue spectral line list to a casa table.
```

Task help

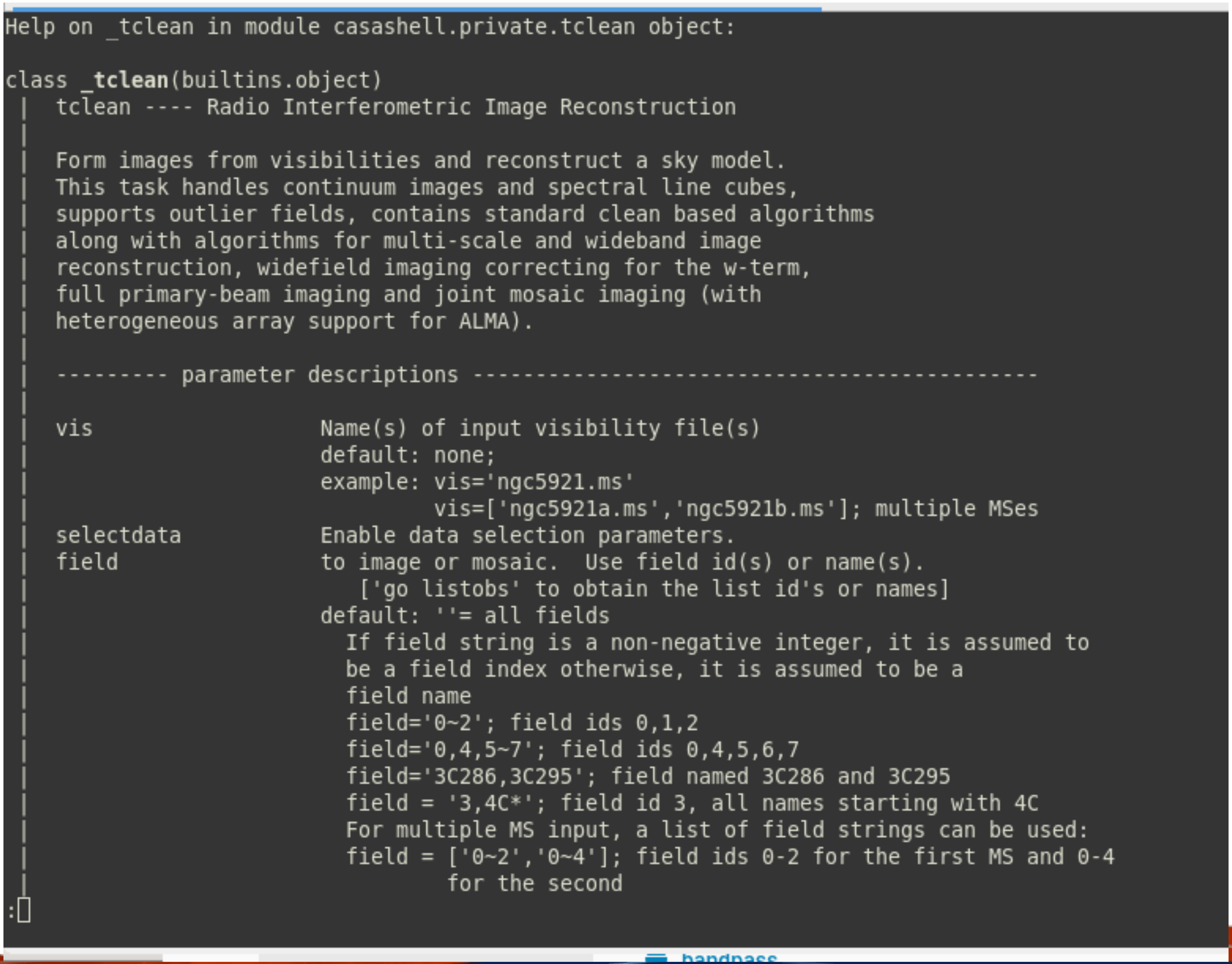
Type:

tclean?

Or

help tclean

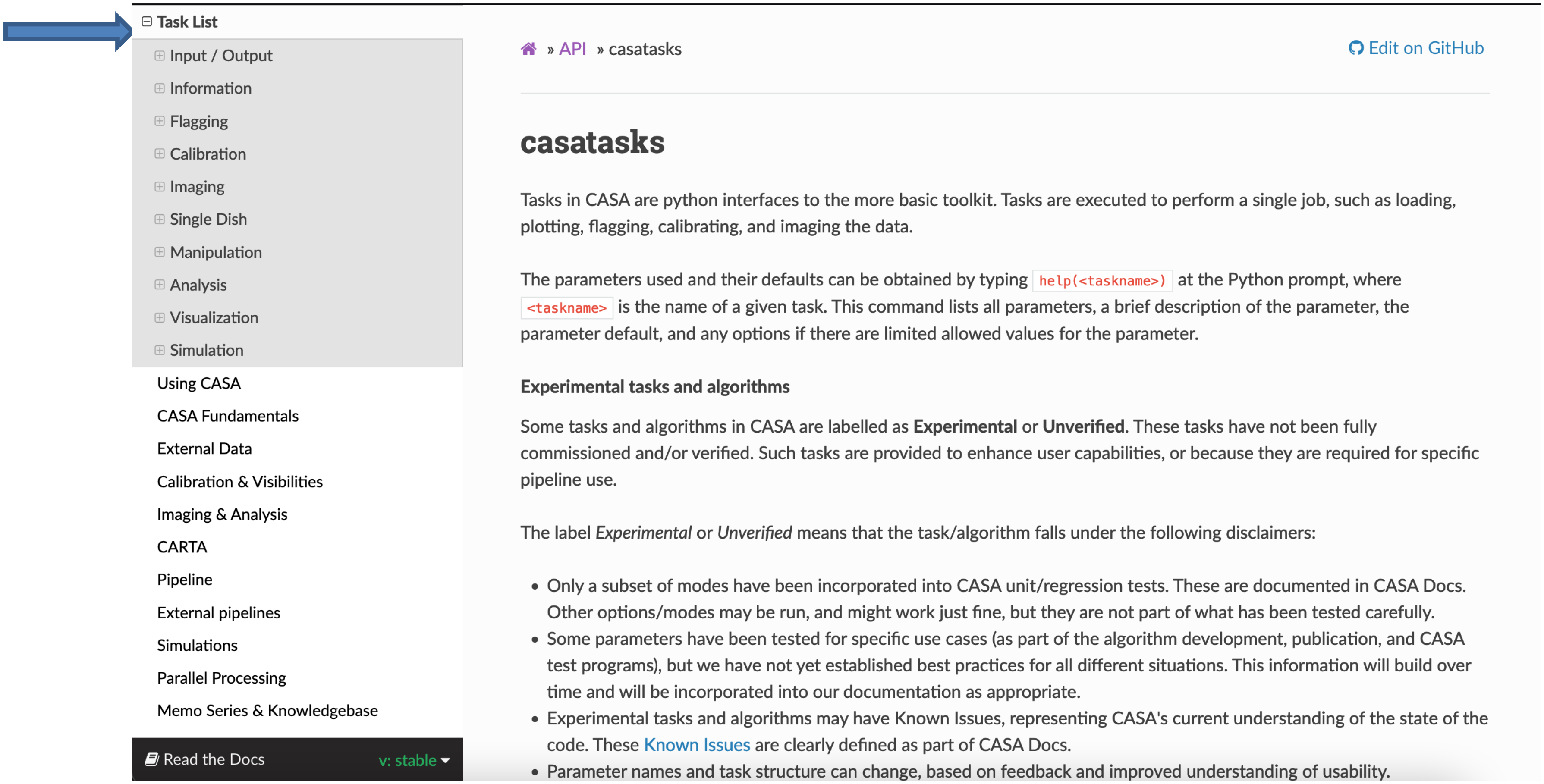
(note: a generic 'help' invokes python help, exit by <enter> or CTRL+D)



```
Help on _tclean in module casashell.private.tclean object:

class _tclean(builtins.object)
 |  tclean ---- Radio Interferometric Image Reconstruction
 |
 |  Form images from visibilities and reconstruct a sky model.
 |  This task handles continuum images and spectral line cubes,
 |  supports outlier fields, contains standard clean based algorithms
 |  along with algorithms for multi-scale and wideband image
 |  reconstruction, widefield imaging correcting for the w-term,
 |  full primary-beam imaging and joint mosaic imaging (with
 |  heterogeneous array support for ALMA).
 |
 |  --------- parameter descriptions ---------------------------------------------
 |
 |  vis                  Name(s) of input visibility file(s)
 |                       default: none;
 |                       example: vis='ngc5921.ms'
 |                                vis=['ngc5921a.ms','ngc5921b.ms']; multiple MSes
 |  selectdata           Enable data selection parameters.
 |  field                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
 |                                 for the second
:
```

Task help

- doc() or doc(tclean) brings up a browser pointed to the CASAdocs Task List
- Browse to find your complete task description



casatasks

Tasks in CASA are python interfaces to the more basic toolkit. Tasks are executed to perform a single job, such as loading, plotting, flagging, calibrating, and imaging the data.

The parameters used and their defaults can be obtained by typing `help(<taskname>)` at the Python prompt, where `<taskname>` is the name of a given task. This command lists all parameters, a brief description of the parameter, the parameter default, and any options if there are limited allowed values for the parameter.

Experimental tasks and algorithms

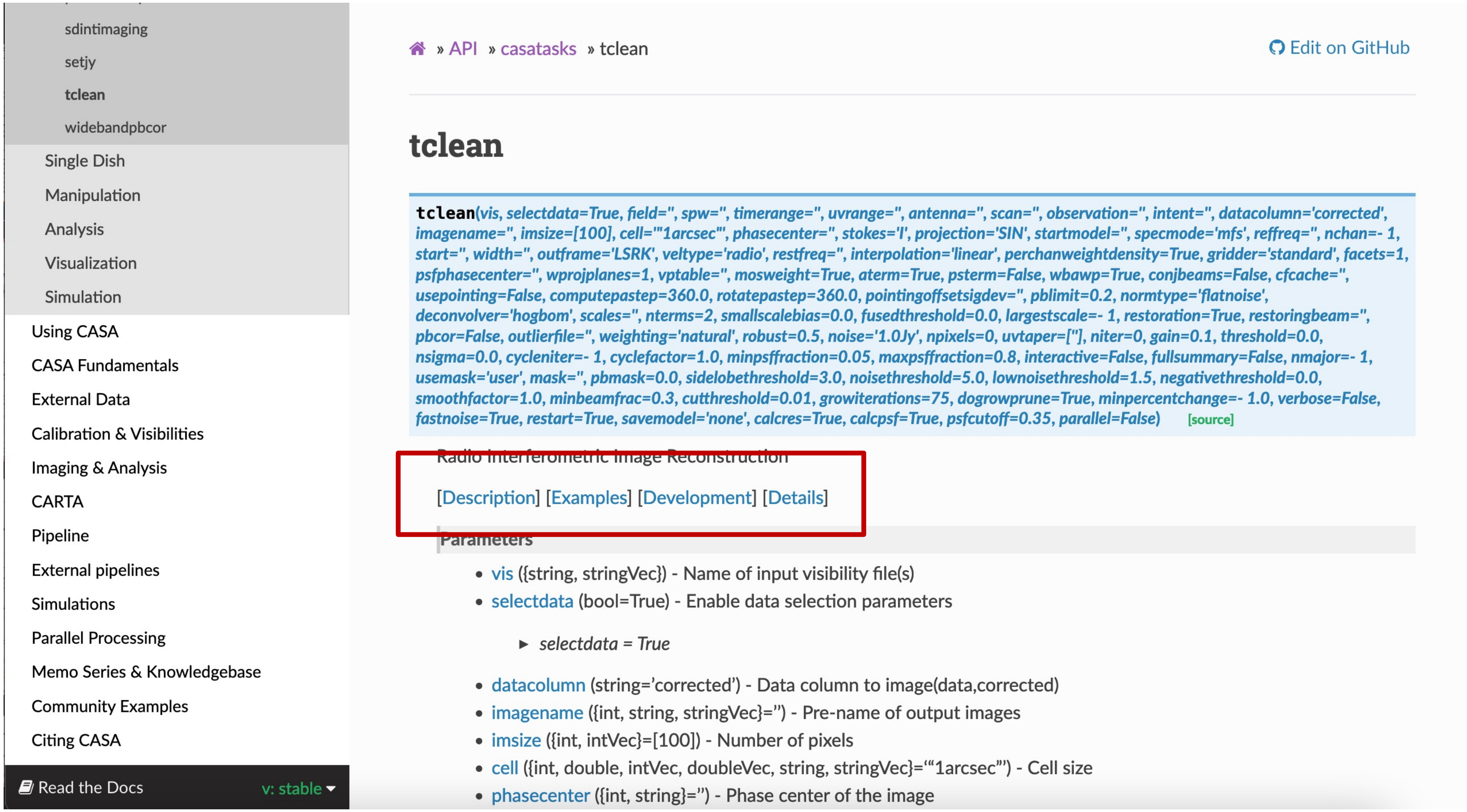
Some tasks and algorithms in CASA are labelled as **Experimental** or **Unverified**. These tasks have not been fully commissioned and/or verified. Such tasks are provided to enhance user capabilities, or because they are required for specific pipeline use.

The label *Experimental* or *Unverified* means that the task/algorithm falls under the following disclaimers:

- Only a subset of modes have been incorporated into CASA unit/regression tests. These are documented in CASA Docs. Other options/modes may be run, and might work just fine, but they are not part of what has been tested carefully.
- Some parameters have been tested for specific use cases (as part of the algorithm development, publication, and CASA test programs), but we have not yet established best practices for all different situations. This information will build over time and will be incorporated into our documentation as appropriate.
- Experimental tasks and algorithms may have Known Issues, representing CASA's current understanding of the state of the code. These Known Issues are clearly defined as part of CASA Docs.
- Parameter names and task structure can change, based on feedback and improved understanding of usability.

Task help

- Within CASAdocs for a task, the Parameters tab is identical to inline help



sdintimaging
setjy
tclean
widebandpbcor
Single Dish
Manipulation
Analysis
Visualization
Simulation
Using CASA
CASA Fundamentals
External Data
Calibration & Visibilities
Imaging & Analysis
CARTA
Pipeline
External pipelines
Simulations
Parallel Processing
Memo Series & Knowledgebase
Community Examples
Citing CASA

Read the Docs v: stable

» API » casatasks » tclean

Edit on GitHub

tclean

```
tclean(vis, selectdata=True, field=", spw=", timerange=", uvrange=", antenna=", scan=", observation=", intent=", datacolumn='corrected',
imagename=", imsize=[100], cell='"1arcsec"', phasecenter=", stokes='I', projection='SIN', startmodel=", specmode='mfs', reffreq=", nchan=- 1,
start=", width=", outframe='LSRK', veltype='radio', restfreq=", interpolation='linear', perchanweightdensity=True, gridder='standard', facets=1,
psfphasecenter=", wprojplanes=1, vptable=", mosweight=True, aterm=True, psterm=False, wbawp=True, conjbeams=False, cfcache=",
usepointing=False, computepastep=360.0, rotatepastep=360.0, pointingoffsetsigdev=", pblimit=0.2, normtype='flatnoise',
deconvolver='hogbom', scales=", nterms=2, smallscalebias=0.0, fusedthreshold=0.0, largestscale=- 1, restoration=True, restoringbeam=",
pbcor=False, outlierfile=", weighting='natural', robust=0.5, noise='1.0Jy', npixels=0, uvtaper=["], niter=0, gain=0.1, threshold=0.0,
nsigma=0.0, cycleniter=- 1, cyclefactor=1.0, minpsffraction=0.05, maxpsffraction=0.8, interactive=False, fullsummary=False, nmajor=- 1,
usemask='user', mask=", pbmask=0.0, sidelobethreshold=3.0, noisethreshold=5.0, lownoisethreshold=1.5, negativethreshold=0.0,
smoothfactor=1.0, minbeamfrac=0.3, cutthreshold=0.01, growiterations=75, dogrowprune=True, minpercentchange=- 1.0, verbose=False,
fastnoise=True, restart=True, savemodel='none', calcres=True, calcpsf=True, psfcutoff=0.35, parallel=False)    [source]
```

Radio Interferometric Image Reconstruction

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

Parameters

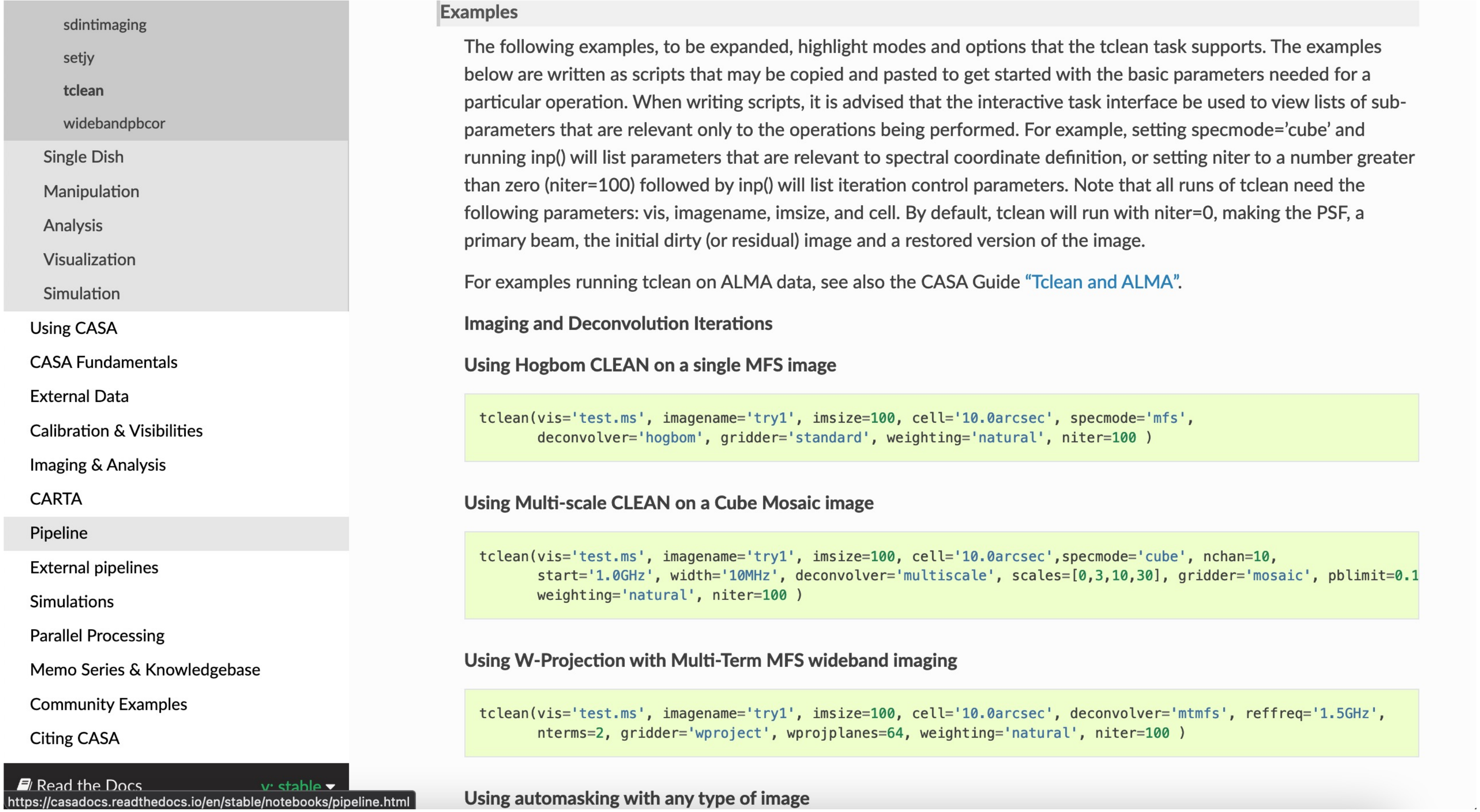
- vis ({string, stringVec}) - Name of input visibility file(s)
- selectdata (bool=True) - Enable data selection parameters

  ▸ *selectdata = True*

- datacolumn (string='corrected') - Data column to image(data,corrected)
- imagename ({int, string, stringVec}=") - Pre-name of output images
- imsize ({int, intVec}=[100]) - Number of pixels
- cell ({int, double, intVec, doubleVec, string, stringVec}="1arcsec") - Cell size
- phasecenter ({int, string}=") - Phase center of the image

Task help

- Examples in CASAdocs



sdintimaging
setjy
tclean
widebandpbcor
Single Dish
Manipulation
Analysis
Visualization
Simulation
Using CASA
CASA Fundamentals
External Data
Calibration & Visibilities
Imaging & Analysis
CARTA
Pipeline
External pipelines
Simulations
Parallel Processing
Memo Series & Knowledgebase
Community Examples
Citing CASA

Read the Docs v: stable

https://casadocs.readthedocs.io/en/stable/notebooks/pipeline.html

Examples

The following examples, to be expanded, highlight modes and options that the tclean task supports. The examples below are written as scripts that may be copied and pasted to get started with the basic parameters needed for a particular operation. When writing scripts, it is advised that the interactive task interface be used to view lists of sub-parameters that are relevant only to the operations being performed. For example, setting specmode='cube' and running inp() will list parameters that are relevant to spectral coordinate definition, or setting niter to a number greater than zero (niter=100) followed by inp() will list iteration control parameters. Note that all runs of tclean need the following parameters: vis, imagename, imsize, and cell. By default, tclean will run with niter=0, making the PSF, a primary beam, the initial dirty (or residual) image and a restored version of the image.

For examples running tclean on ALMA data, see also the CASA Guide "Tclean and ALMA".

Imaging and Deconvolution Iterations

Using Hogbom CLEAN on a single MFS image

```
tclean(vis='test.ms', imagename='try1', imsize=100, cell='10.0arcsec', specmode='mfs',
       deconvolver='hogbom', gridder='standard', weighting='natural', niter=100 )
```

Using Multi-scale CLEAN on a Cube Mosaic image

```
tclean(vis='test.ms', imagename='try1', imsize=100, cell='10.0arcsec',specmode='cube', nchan=10,
       start='1.0GHz', width='10MHz', deconvolver='multiscale', scales=[0,3,10,30], gridder='mosaic', pblimit=0.1
       weighting='natural', niter=100 )
```

Using W-Projection with Multi-Term MFS wideband imaging

```
tclean(vis='test.ms', imagename='try1', imsize=100, cell='10.0arcsec', deconvolver='mtmfs', reffreq='1.5GHz',
       nterms=2, gridder='wproject', wprojplanes=64, weighting='natural', niter=100 )
```

Using automasking with any type of image

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
  - default taskname resets all previous settings to default values

```
inp listobs
vis = ‘mydata.ms’
listfile = ‘outfile.txt’
inp
go
```

Writes to outfile.txt

```
listfile = ‘outfile.txt’
default listobs
inp listobs
vis = ‘mydata.ms’
inp
go
```

Won’t write to outfile.txt
listfile=” is the default

Task interface

Inspect task inputs:

inp tclean

Black/white: valid (default or non-default) value

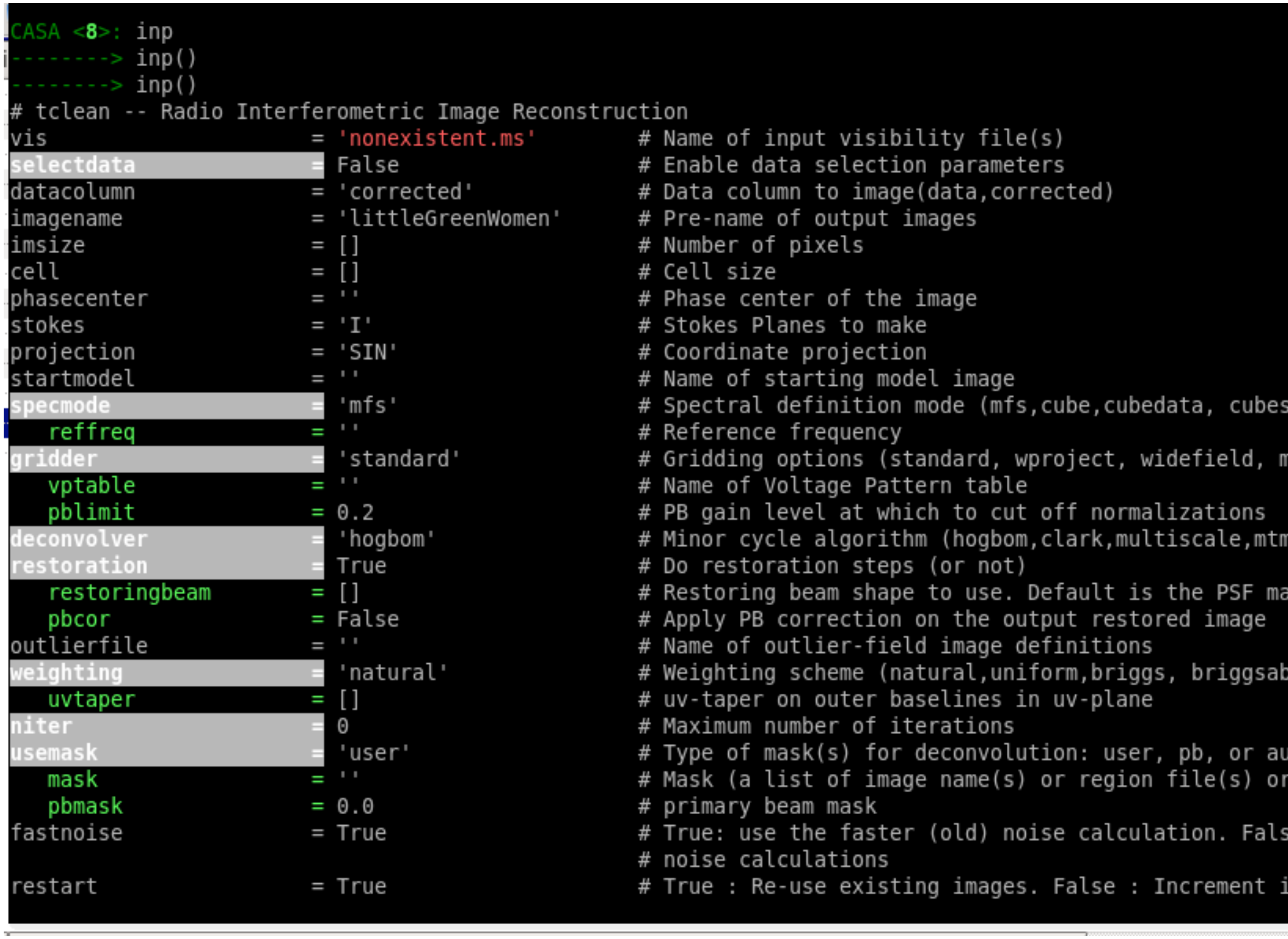
Red: invalid value

Grey: expandable

Green: sub-parameter

Reset defaults:

default tclean



```
CASA <8>: inp
--------> inp()
--------> inp()
# tclean -- Radio Interferometric Image Reconstruction
vis                 = 'nonexistent.ms'       # Name of input visibility file(s)
selectdata          = False                  # Enable data selection parameters
datacolumn          = 'corrected'            # Data column to image(data,corrected)
imagename           = 'littleGreenWomen'     # Pre-name of output images
imsize              = []                     # 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,cubedata, cubes
   reffreq          = ''                     # Reference frequency
gridder             = 'standard'             # Gridding options (standard, wproject, widefield, m
   vptable          = ''                     # Name of Voltage Pattern table
   pblimit          = 0.2                    # PB gain level at which to cut off normalizations
deconvolver         = 'hogbom'               # Minor cycle algorithm (hogbom,clark,multiscale,mtm
restoration         = True                   # Do restoration steps (or not)
   restoringbeam    = []                     # Restoring beam shape to use. Default is the PSF ma
   pbcor            = False                  # Apply PB correction on the output restored image
outlierfile         = ''                     # Name of outlier-field image definitions
weighting           = 'natural'              # Weighting scheme (natural,uniform,briggs, briggsab
   uvtaper          = []                     # uv-taper on outer baselines in uv-plane
niter               = 0                      # Maximum number of iterations
usemask             = 'user'                 # Type of mask(s) for deconvolution: user, pb, or au
   mask             = ''                     # Mask (a list of image name(s) or region file(s) or
   pbmask           = 0.0                    # primary beam mask
fastnoise           = True                   # True: use the faster (old) noise calculation. Fals
                                             # noise calculations

restart             = True                   # True : Re-use existing images. False : Increment i
```

Task interface

Inspect task inputs:

inp tclean

Black/white: valid (default or non-default) value

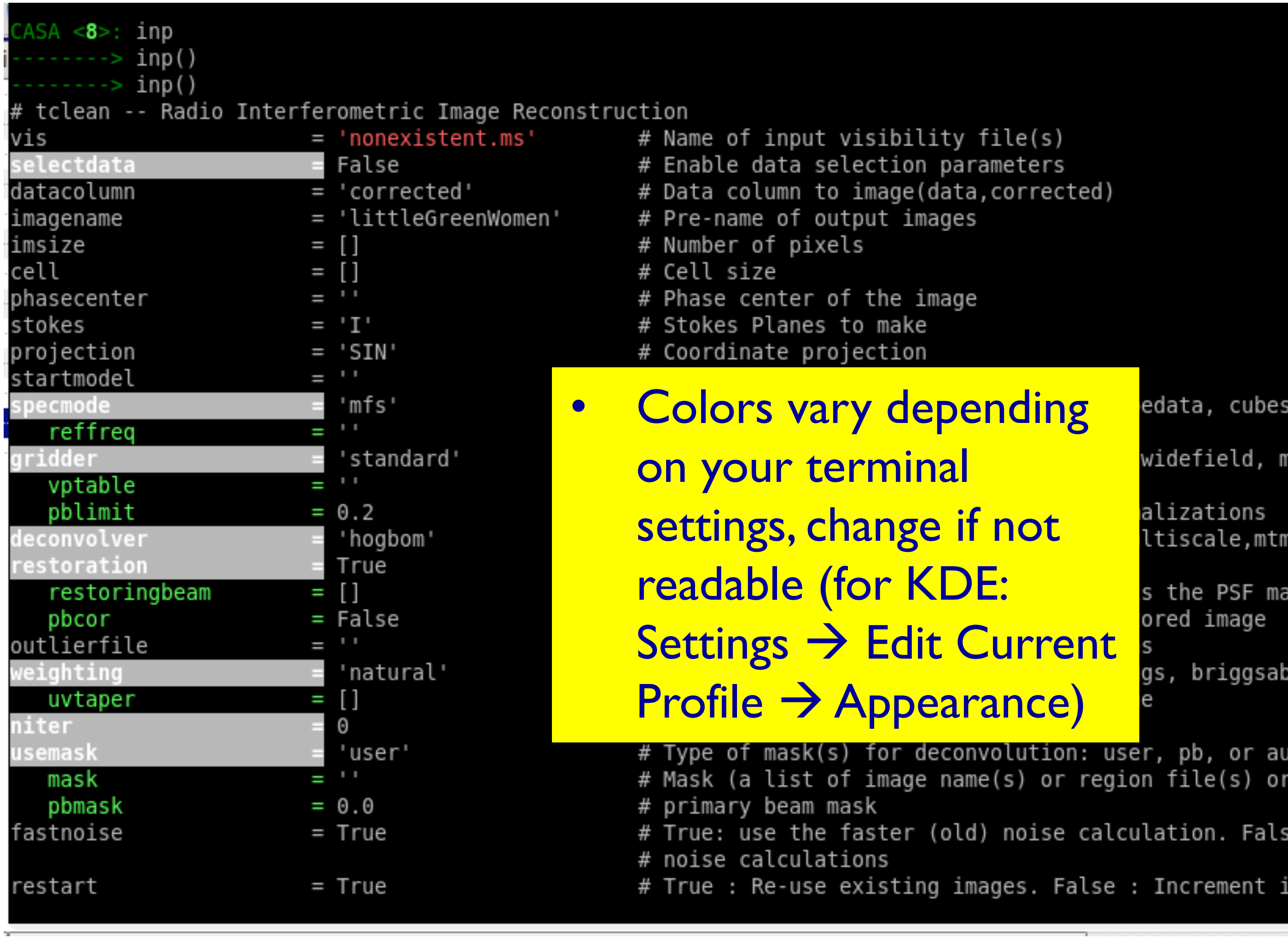
Red: invalid value

Grey: expandable

Green: sub-parameter

Reset defaults:

default tclean



```
CASA <8>: inp
---------> inp()
---------> inp()
# tclean -- Radio Interferometric Image Reconstruction
vis                     = 'nonexistent.ms'        # Name of input visibility file(s)
selectdata              = False                   # Enable data selection parameters
datacolumn              = 'corrected'             # Data column to image(data,corrected)
imagename               = 'littleGreenWomen'      # Pre-name of output images
imsize                  = []                      # Number of pixels
cell                    = []                      # Cell size
phasecenter             = ''                      # Phase center of the image
stokes                  = 'I'                     # Stokes Planes to make
projection              = 'SIN'                   # Coordinate projection
startmodel              = ''
specmode                = 'mfs'                                                           edata, cubes
   reffreq              = ''
gridder                 = 'standard'                                                      widefield, m
   vptable              = ''
   pblimit              = 0.2                                                             alizations
deconvolver             = 'hogbom'                                                        ltiscale,mtm
restoration             = True
   restoringbeam        = []                                                              s the PSF ma
   pbcor                = False                                                           ored image
outlierfile             = ''                                                              s
weighting               = 'natural'                                                       gs, briggsab
   uvtaper              = []                                                              e
niter                   = 0
usemask                 = 'user'                  # Type of mask(s) for deconvolution: user, pb, or au
   mask                 = ''                      # Mask (a list of image name(s) or region file(s) or
   pbmask               = 0.0                     # primary beam mask
fastnoise               = True                    # True: use the faster (old) noise calculation. Fals
                                                  # noise calculations

restart                 = True                    # True : Re-use existing images. False : Increment i
```

- Colors vary depending on your terminal settings, change if not readable (for KDE: Settings → Edit Current Profile → Appearance)

How to run a task

- IPython command line (Python function call)
  - Set all parameters at once
  - Values that are not specified will be defaulted
  - Unspecified values will be taken as listed in task help
  - Useful for pseudo-scripting
    - Copy-paste into a text or “.py” file to keep record of processing that can be easily changed an re-run if needed

```
listobs(vis=‘mydata.ms’, listfile=‘outfile.txt’)
```

```
listfile=‘outfile.txt’
listobs(vis=‘mydata.ms’)
```

Will not write to outfile.txt (listfile=“ is the default)

```
listobs(‘mydata.ms’)
```

vis is the first parameter, as shown in help:

```
listobs = class listobs_cli_
 |  Methods defined here:
 |  __call__(self, vis=None, selectdata=None, spw=None, field=None, antenna=None, uvrange=None, t
```

How to run a task

- Some tasks return a dictionary

```
results = imstat(imagename=‘pluto.im‘)
```

- Will also be shown on screen if not returned in a variable

- Dictionaries can be accessed through Python commands

```
CASA <13>: results
Out[13]:{'blc': array([0, 0, 0, 0], dtype=int32), 'blcf': '09:47:57.724, +13.16.35.660, I,
3.63124e+10Hz', 'max': array([ 0.00010101]),…..
CASA <11>: results['median'][0]
Out[11]: 0.77494734525680542
CASA <12>: fivesigma=5*results['rms'][0]
CASA <13>: fivesigma
Out[13]: 3.9262134213339852
```

How to run a task

- **Scripting**
  - Inside IPython: execfile(‘script.py’)
    - **Note** that the treatment of globals has changed in Python 3; to call execfile within a script run with execfile, make sure to call execfile(‘myscript.py’, globals())
  - Or %run -i ‘script.py’ (-i uses ipython namespace)
  - Or start casa non-interactively and run script right away: casa --nologger --nogui -c script.py

Content of script.py:

```
# functional calls
listobs(vis’mydata.ms’, listfile=‘outfile.txt’)
# full power of python
if (selectdata):
        # insist no ACs
        if len(msselect)>0:
            msselect='('+msselect+') && ANTENNA1!=ANTENNA2'
        else:
            msselect='ANTENNA1!=ANTENNA2'

        # pass all data selection parameters in as specified
        gaincal(time=timerange,spw=spw, scan=scan, field=field,
                intent=intent, observation=str(observation),
                baseline=antenna,uvrange=uvrange,chanmode='none',
                msselect=msselect);
```

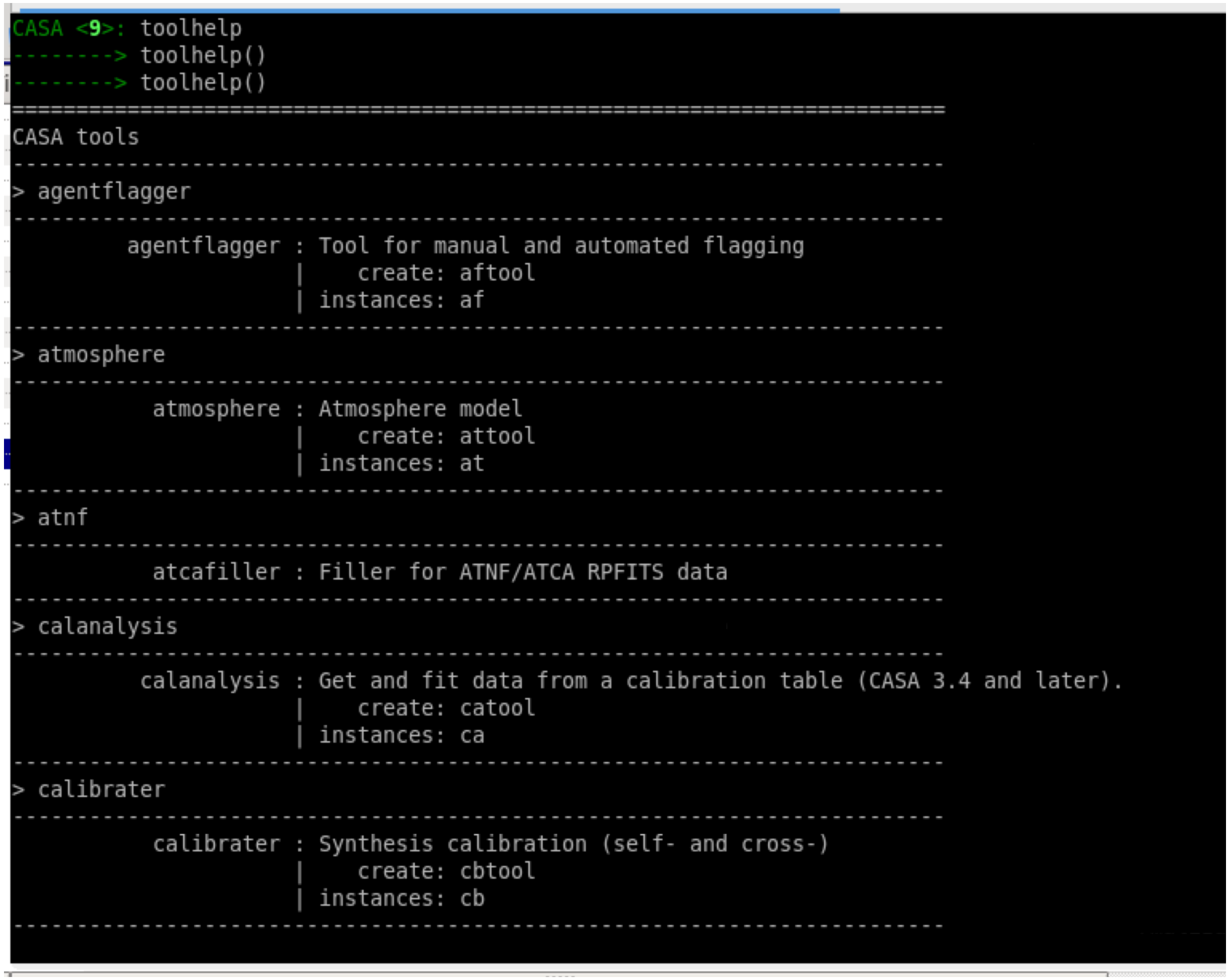
Tools

Tools (and their methods) are the building blocks of tasks

- Contain full functionality of CASA
- Used internally by tasks
- E.g. image analysis (ia), table utilities (tb), …

To see short summaries, type:

toolhelp



```
CASA <9>: toolhelp
--------> toolhelp()
--------> toolhelp()
==========================================================================
CASA tools
--------------------------------------------------------------------------
> agentflagger
--------------------------------------------------------------------------
       agentflagger : Tool for manual and automated flagging
                    |    create: aftool
                    | instances: af
--------------------------------------------------------------------------
> atmosphere
--------------------------------------------------------------------------
         atmosphere : Atmosphere model
                    |    create: attool
                    | instances: at
--------------------------------------------------------------------------
> atnf
--------------------------------------------------------------------------
         atcafiller : Filler for ATNF/ATCA RPFITS data
--------------------------------------------------------------------------
> calanalysis
--------------------------------------------------------------------------
        calanalysis : Get and fit data from a calibration table (CASA 3.4 and later).
                    |    create: catool
                    | instances: ca
--------------------------------------------------------------------------
> calibrater
--------------------------------------------------------------------------
         calibrater : Synthesis calibration (self- and cross-)
                    |    create: cbtool
                    | instances: cb
--------------------------------------------------------------------------
```

How to use the tools

- Tools contain a number of methods (>1k tool methods are available)
  - 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.fft(amp=‘imagefft.im’,…)
ia.close()
```

- PySynthesisImager scripting for tclean is a bit different (see examples in tclean CASAdocs)

Still searching for functionality?

- Look through contributed scripts and tasks at:

  http://casaguides.nrao.edu/ (e.g., analysisUtils)

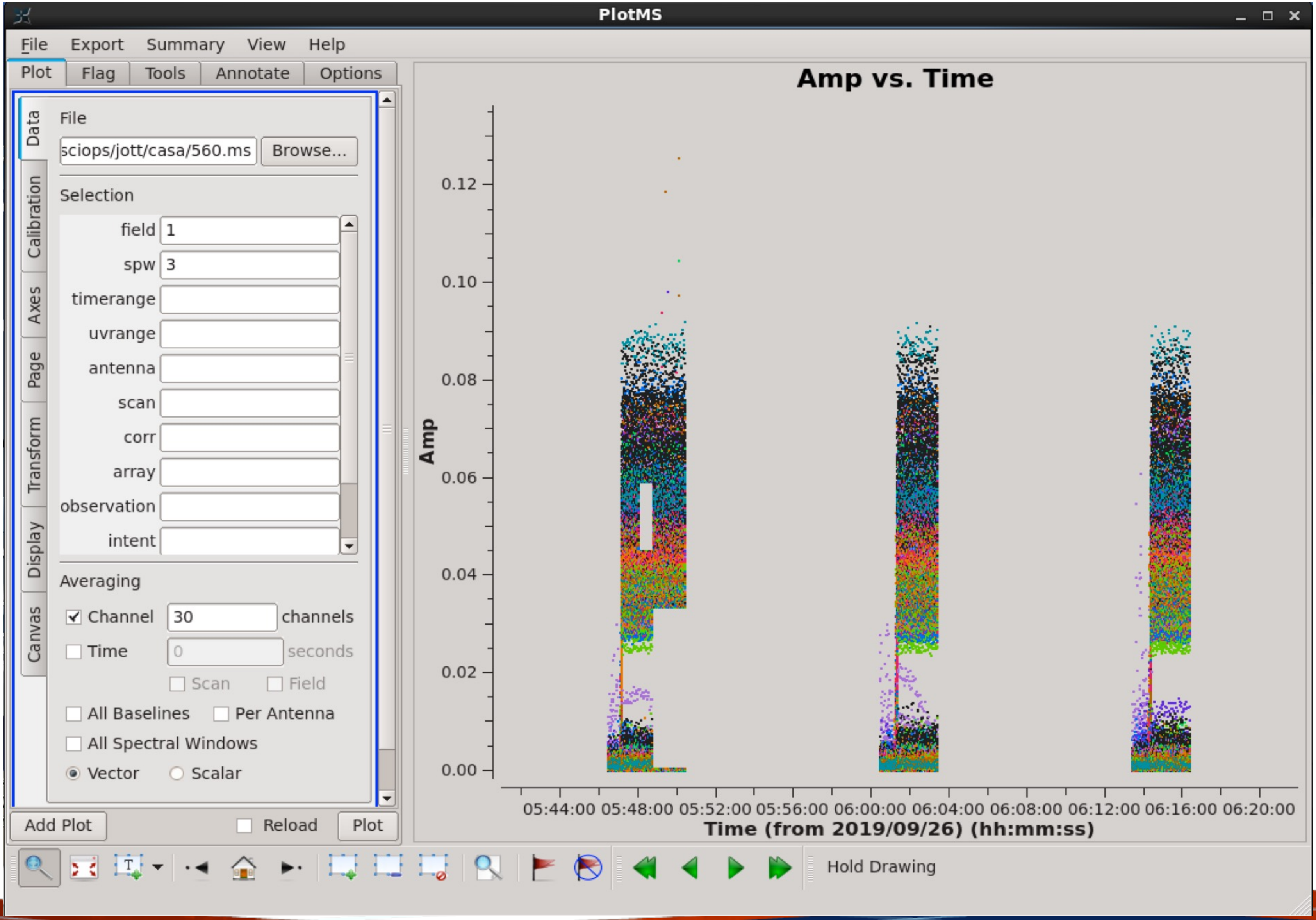
- 3rd parties like the Nordic ALMA ARC node (e.g., uvmultifit), etc.

- If you still can't find what you need, write your own task!
  - Combination of Python plus CASA toolkit is very powerful

Applications

- Used to display and edit data (visibilities, calibration tables, images)
- Can be invoked inside CASA or (currently only in CASA 5) as standalone programs from Linux shell
- Visibilities and calibration tables: plotms, msview, feather, plotants, plotbandpass
- Any CASA (table) data: browsetable
- Images: viewer, CARTA (affiliate package)
- Don't forget about full functionality of Python! e.g. matplotlib, astropy, …

PlotMS



PlotMS
File Export Summary View Help
Plot Flag Tools Annotate Options
Data Calibration Axes Page Transform Display Canvas
Browse...
Add Plot
Reload
Plot
Hold Drawing

File

sciops/jott/casa/560.ms

Selection

field 1

spw 3

timerange

uvrange

antenna

scan

corr

array

observation

intent

Averaging

☑ Channel 30 channels

□ Time 0 seconds

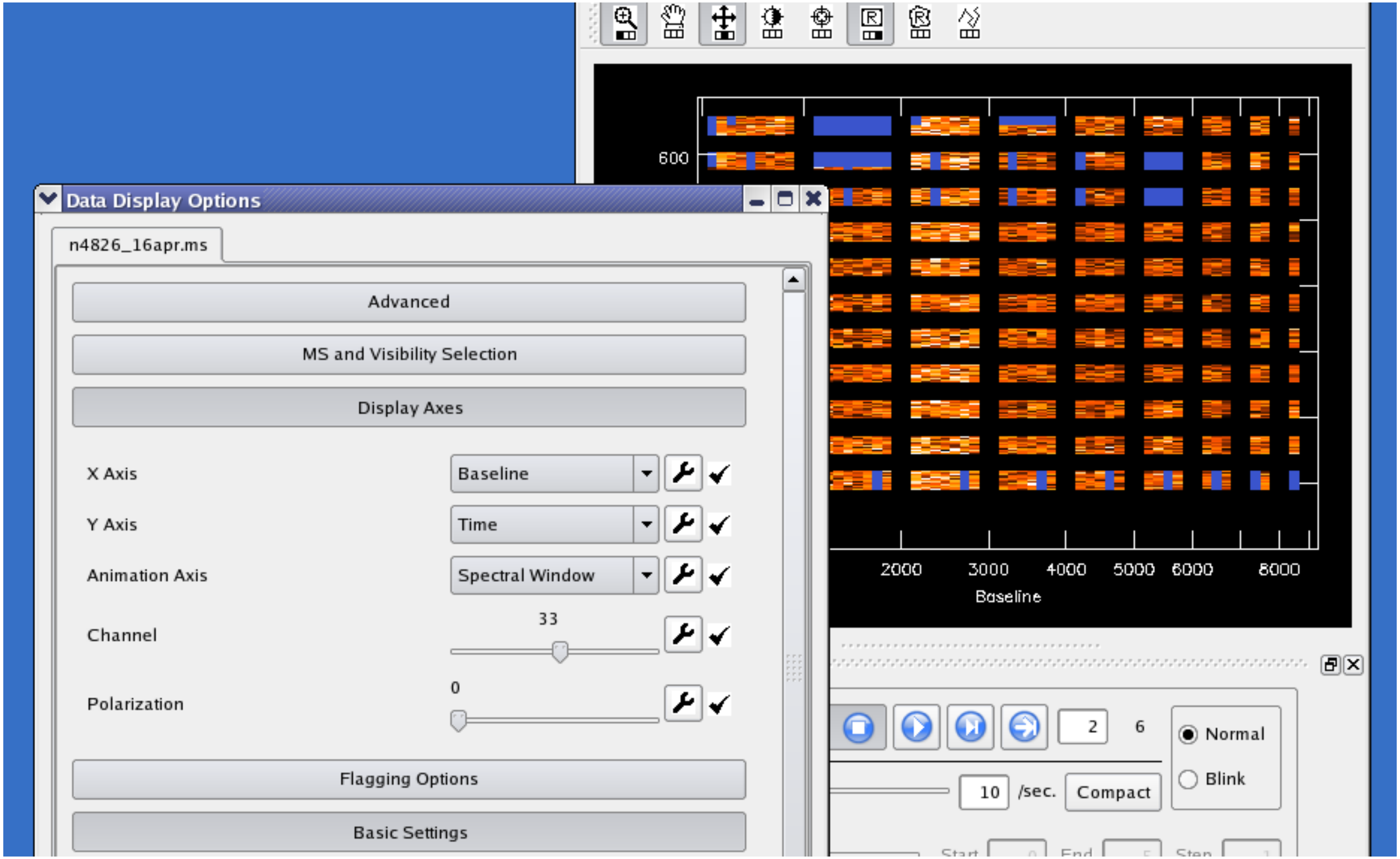
□ Scan □ Field

□ All Baselines □ Per Antenna

□ All Spectral Windows

● Vector ○ Scalar

Viewer (msview)



Data Display Options

n4826_16apr.ms

X Axis: Baseline

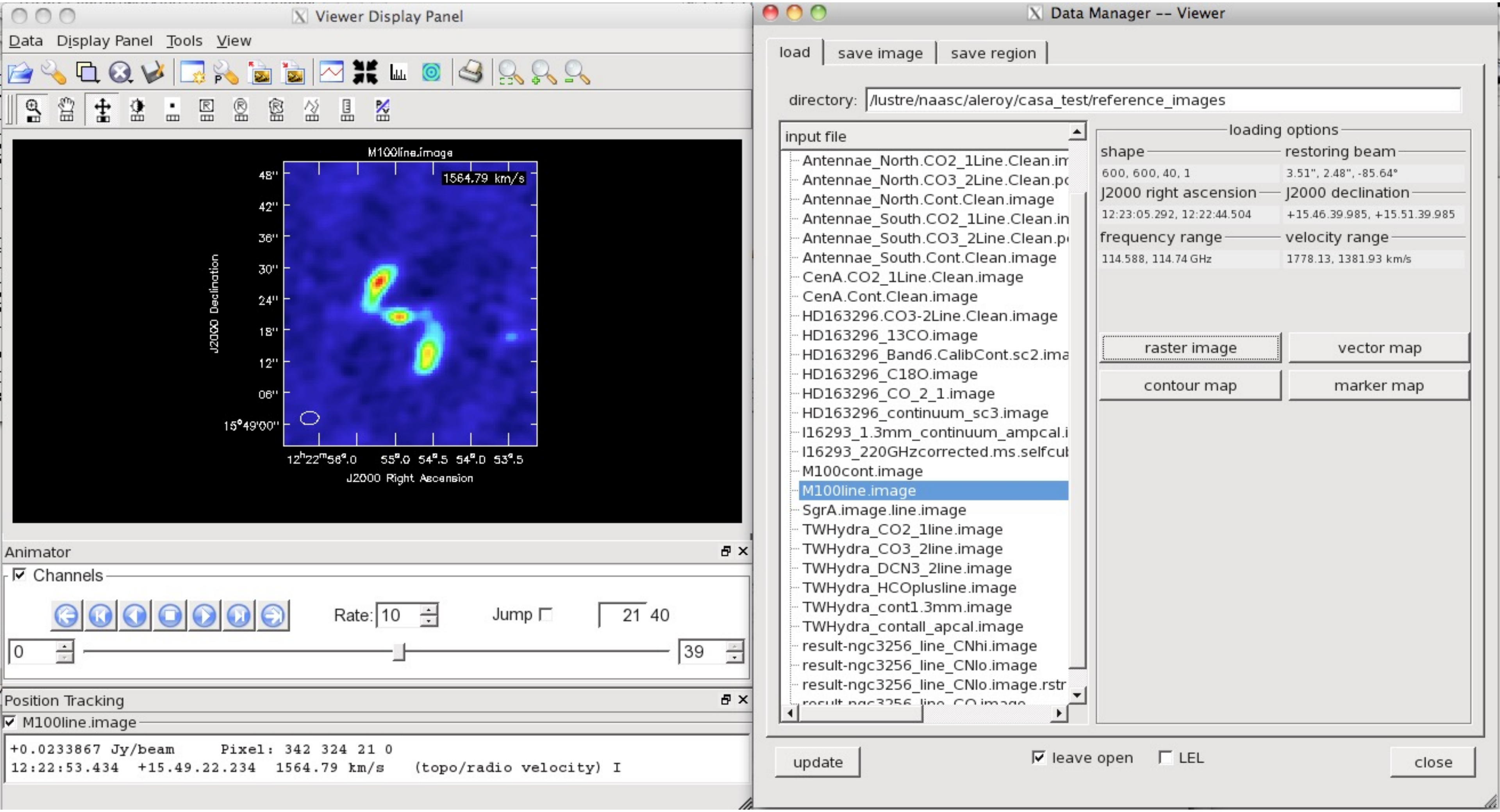
Y Axis: Time

Animation Axis: Spectral Window

Channel: 33

Polarization: 0

Viewer (imview)



```
+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
```

directory: /lustre/naasc/aleroy/casa_test/reference_images

- Antennae_North.CO2_1Line.Clean.im
- Antennae_North.CO3_2Line.Clean.pc
- Antennae_North.Cont.Clean.image
- Antennae_South.CO2_1Line.Clean.in
- Antennae_South.CO3_2Line.Clean.pi
- Antennae_South.Cont.Clean.image
- CenA.CO2_1Line.Clean.image
- CenA.Cont.Clean.image
- HD163296.CO3-2Line.Clean.image
- HD163296_13CO.image
- HD163296_Band6.CalibCont.sc2.ima
- HD163296_C18O.image
- HD163296_CO_2_1.image
- HD163296_continuum_sc3.image
- I16293_1.3mm_continuum_ampcal.i
- I16293_220GHzcorrected.ms.selfcut
- M100cont.image
- M100line.image
- SgrA.image.line.image
- TWHydra_CO2_1line.image
- TWHydra_CO3_2line.image
- TWHydra_DCN3_2line.image
- TWHydra_HCOplusline.image
- TWHydra_cont1.3mm.image
- TWHydra_contall_apcal.image
- result-ngc3256_line_CNhi.image
- result-ngc3256_line_CNlo.image
- result-ngc3256_line_CNlo.image.rstr

shape: 600, 600, 40, 1

restoring beam: 3.51", 2.48", -85.64°

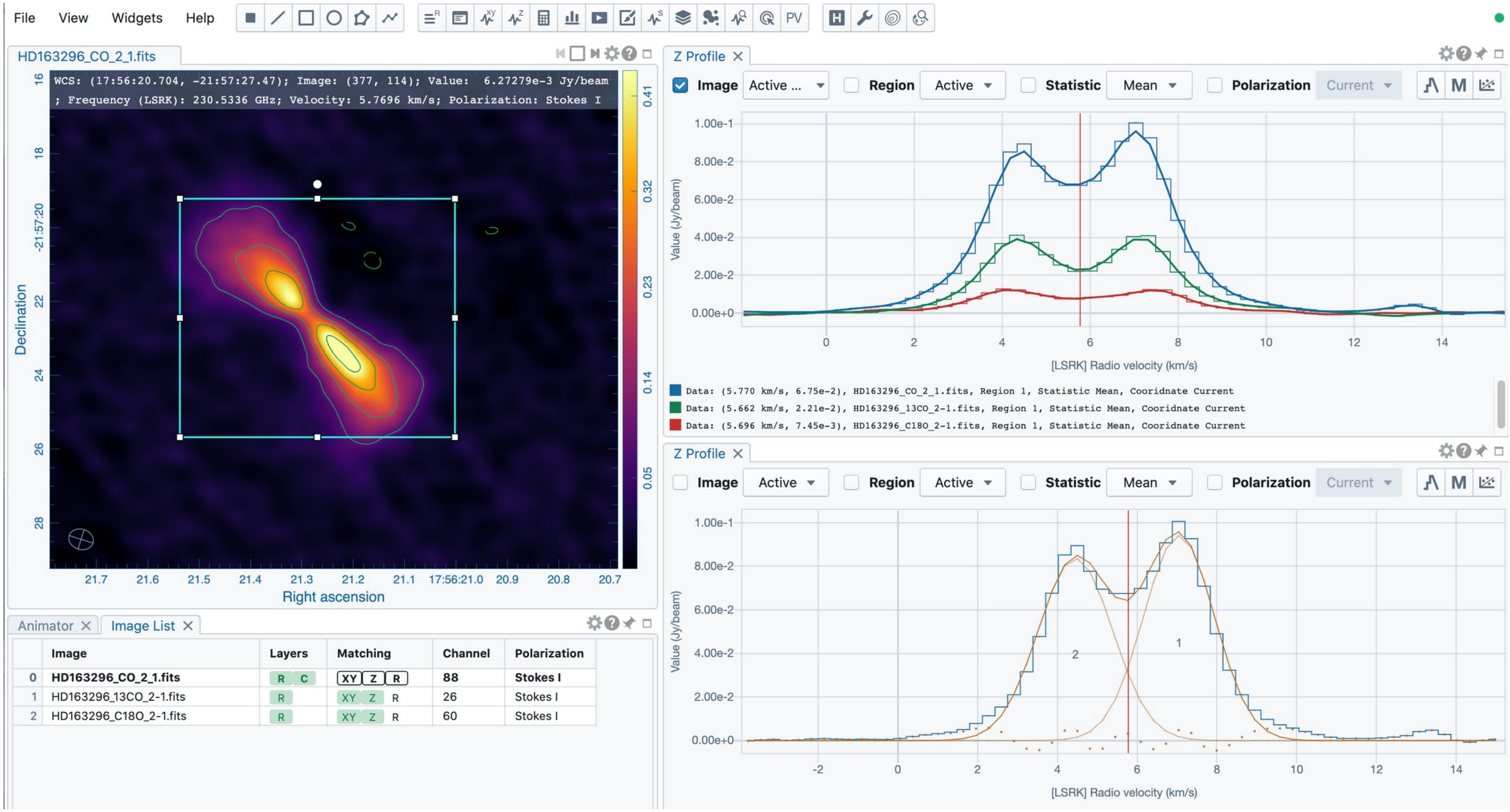
J2000 right ascension: 12:23:05.292, 12:22:44.504

J2000 declination: +15.46.39.985, +15.51.39.985

frequency range: 114.588, 114.74 GHz

velocity range: 1778.13, 1381.93 km/s

CARTA: line fitting

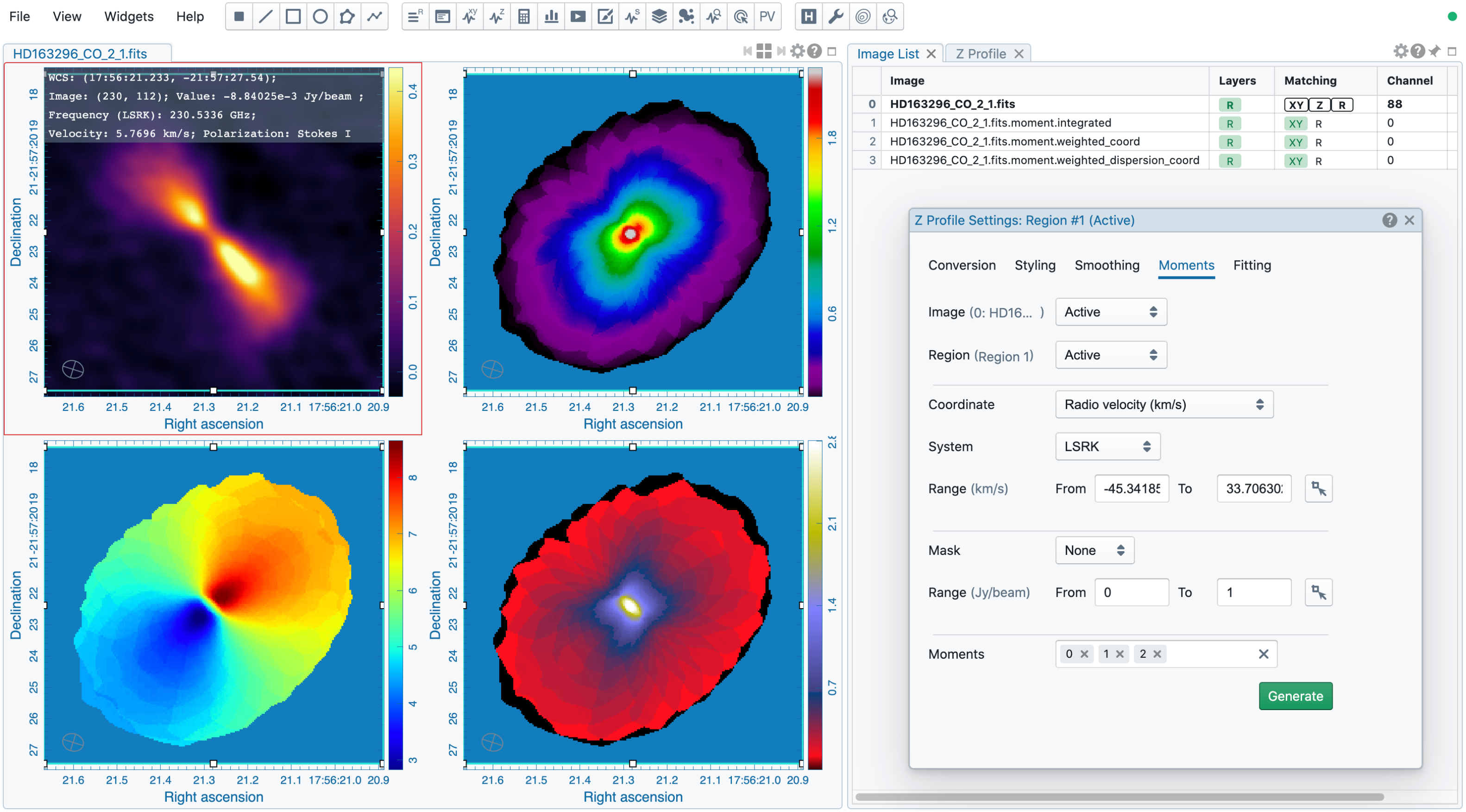


WCS: (17:56:20.704, -21:57:27.47); Image: (377, 114); Value: 6.27279e-3 Jy/beam
; Frequency (LSRK): 230.5336 GHz; Velocity: 5.7696 km/s; Polarization: Stokes I

🟦 Data: (5.770 km/s, 6.75e-2), HD163296_CO_2_1.fits, Region 1, Statistic Mean, Cooridnate Current
🟩 Data: (5.662 km/s, 2.21e-2), HD163296_13CO_2-1.fits, Region 1, Statistic Mean, Cooridnate Current
🟥 Data: (5.696 km/s, 7.45e-3), HD163296_C18O_2-1.fits, Region 1, Statistic Mean, Cooridnate Current

	Image	Layers	Matching	Channel	Polarization
0	**HD163296_CO_2_1.fits**	R C	XY Z R	**88**	**Stokes I**
1	HD163296_13CO_2-1.fits	R	XY Z R	26	Stokes I
2	HD163296_C18O_2-1.fits	R	XY Z R	60	Stokes I

CARTA: moment maps



```
WCS: (17:56:21.233, -21:57:27.54);
Image: (230, 112); Value: -8.84025e-3 Jy/beam ;
Frequency (LSRK): 230.5336 GHz;
Velocity: 5.7696 km/s; Polarization: Stokes I
```

	Image	Layers	Matching	Channel
0	**HD163296_CO_2_1.fits**	R	XY Z R	88
1	HD163296_CO_2_1.fits.moment.integrated	R	XY R	0
2	HD163296_CO_2_1.fits.moment.weighted_coord	R	XY R	0
3	HD163296_CO_2_1.fits.moment.weighted_dispersion_coord	R	XY R	0

Z Profile Settings: Region #1 (Active)

Conversion | Styling | Smoothing | Moments | Fitting

Image (0: HD16...): Active

Region (Region 1): Active

Coordinate: Radio velocity (km/s)

System: LSRK

Range (km/s): From -45.3418ξ To 33.70630:

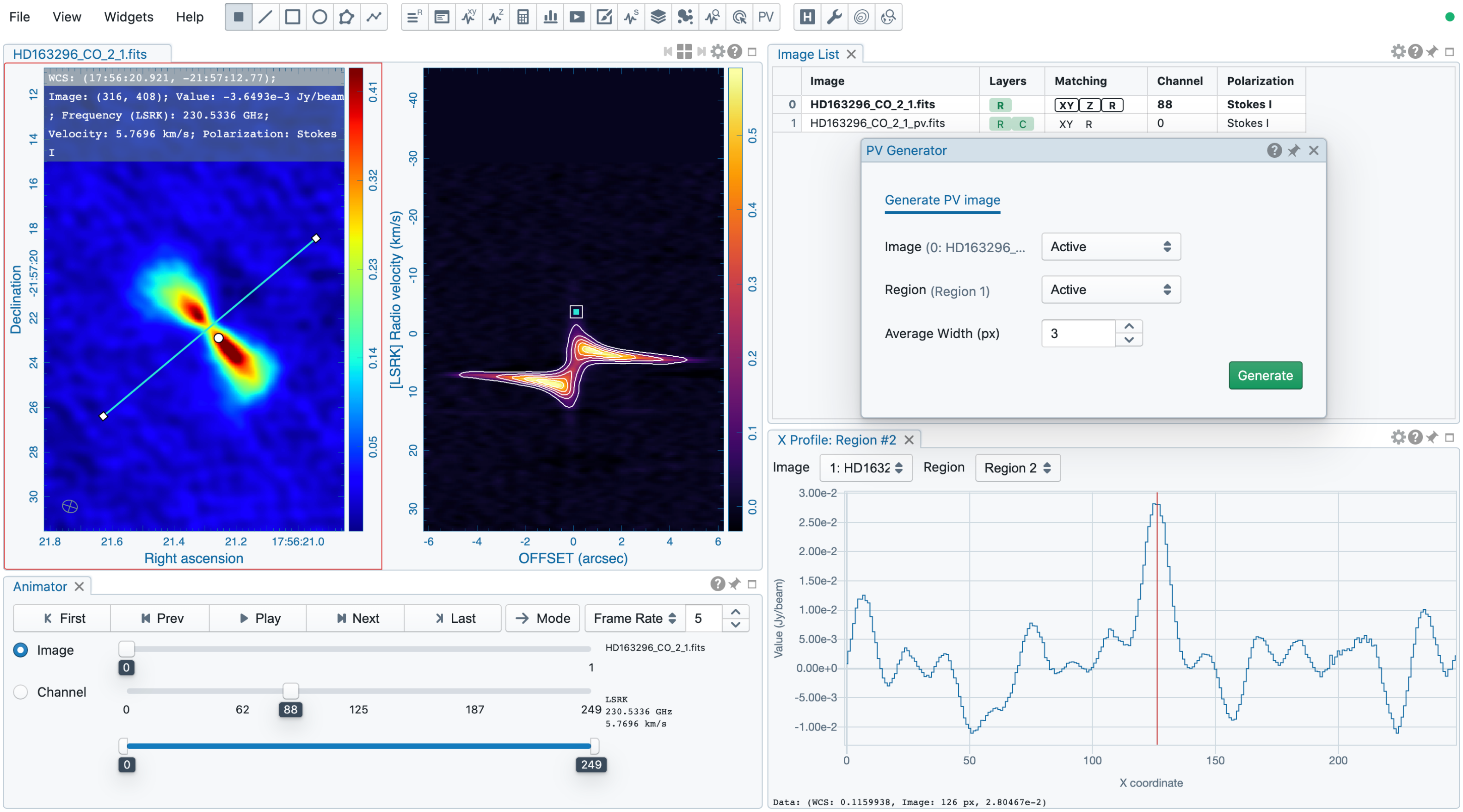
Mask: None

Range (Jy/beam): From 0 To 1

Moments: 0 × 1 × 2 ×

Generate

CARTA: position-velocity diagrams



```
WCS: (17:56:20.921, -21:57:12.77);
Image: (316, 408); Value: -3.6493e-3 Jy/beam
; Frequency (LSRK): 230.5336 GHz;
Velocity: 5.7696 km/s; Polarization: Stokes
I
```

	Image	Layers	Matching	Channel	Polarization
0	**HD163296_CO_2_1.fits**	R	XY Z R	**88**	**Stokes I**
1	HD163296_CO_2_1_pv.fits	R C	XY R	0	Stokes I

PV Generator

Generate PV image

Image (0: HD163296_... Active

Region (Region 1) Active

Average Width (px) 3

Generate

Animator: Frame Rate 5

● Image — 0 … 1 — HD163296_CO_2_1.fits

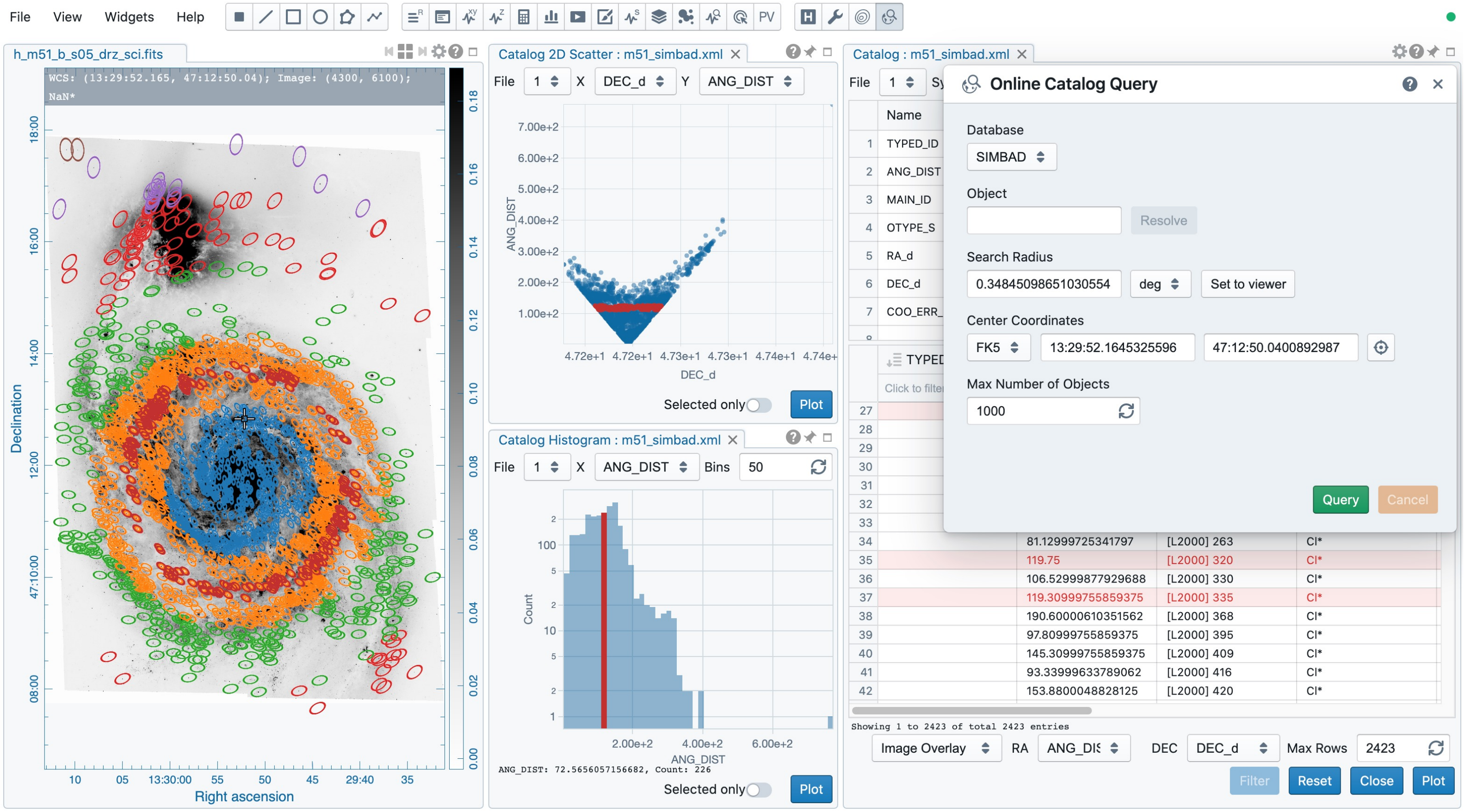
○ Channel — 0, 62, 88, 125, 187, 249 — LSRK 230.5336 GHz 5.7696 km/s

Range: 0 – 249

X Profile: Image 1: HD1632 — Region: Region 2

```
Data: (WCS: 0.1159938, Image: 126 px, 2.80467e-2)
```

CARTA: catalog query/selection



WCS: (13:29:52.165, 47:12:50.04); Image: (4300, 6100); NaN*

Catalog 2D Scatter: File 1, X DEC_d, Y ANG_DIST

Selected only ○ Plot

Catalog Histogram: File 1, X ANG_DIST, Bins 50

ANG_DIST: 72.5656057156682, Count: 226

Selected only ○ Plot

Catalog: File 1

	Name
1	TYPED_ID
2	ANG_DIST
3	MAIN_ID
4	OTYPE_S
5	RA_d
6	DEC_d
7	COO_ERR_

TYPED

Click to filter

Online Catalog Query

Database: SIMBAD

Object:

Resolve

Search Radius: 0.34845098651030554 deg

Set to viewer

Center Coordinates: FK5 13:29:52.1645325596 47:12:50.0400892987

Max Number of Objects: 1000

Query Cancel

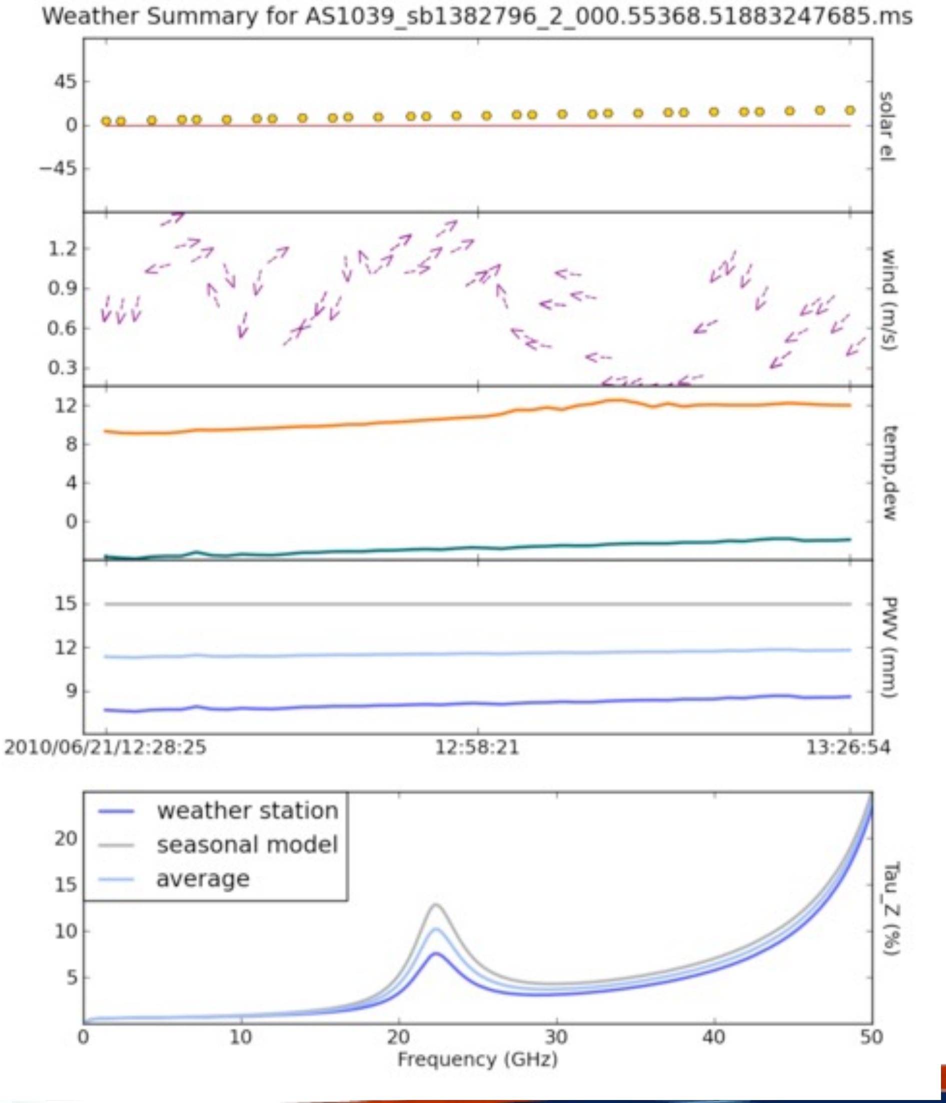
27				
28				
29				
30				
31				
32				
33				
34		81.12999725341797	[L2000] 263	Cl*
35		119.75	[L2000] 320	Cl*
36		106.52999877929688	[L2000] 330	Cl*
37		119.30999755859375	[L2000] 335	Cl*
38		190.60000610351562	[L2000] 368	Cl*
39		97.80999755859375	[L2000] 395	Cl*
40		145.30999755859375	[L2000] 409	Cl*
41		93.33999633789062	[L2000] 416	Cl*
42		153.8800048828125	[L2000] 420	Cl*

Showing 1 to 2423 of total 2423 entries

Image Overlay, RA ANG_DIS, DEC DEC_d, Max Rows 2423

Filter Reset Close Plot

Plot anything - matplotlib



Data structures

- JVLA and ALMA observatory raw data are stored in **(A)SDM** format (xml, binaries)
- CASA uses **MeasurementSets** (MS) for visibilities
  - Use **importasdm** for ALMA, EVLA/JVLA, **importvla** for historic VLA, etc.

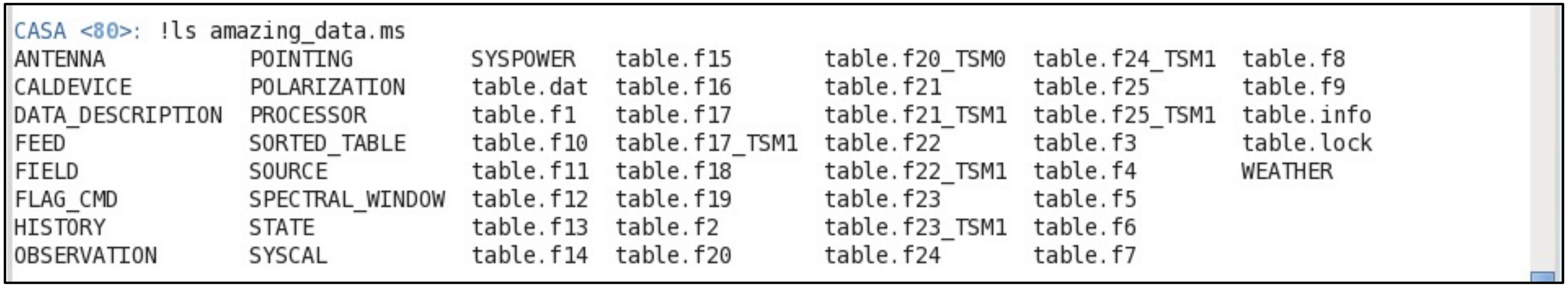
- Calibration information is stored in **calibration tables**
- Images are in **CASA image format**
  - Use **importfits** to convert a FITS file to CASA image format, **exportfits** to write out in FITS

- All of the CASA formats are *directories* with a table structure that contains the necessary information
  - Copying requires recursive option (**!cp –r**)

- Delete tables using **rmtables('mydata.ms')**
  - **!rm –rf** or **shutil.rmtree('mydata.ms')** may also work, but can leave traces in the cache

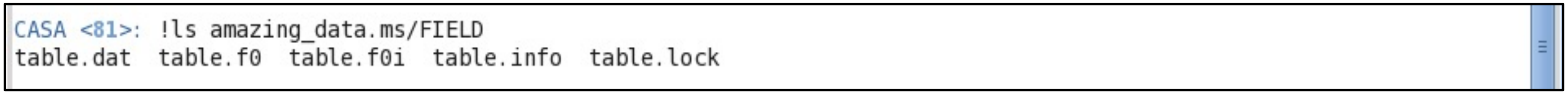
Inspect a MeasurementSet (MS)

- 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, …

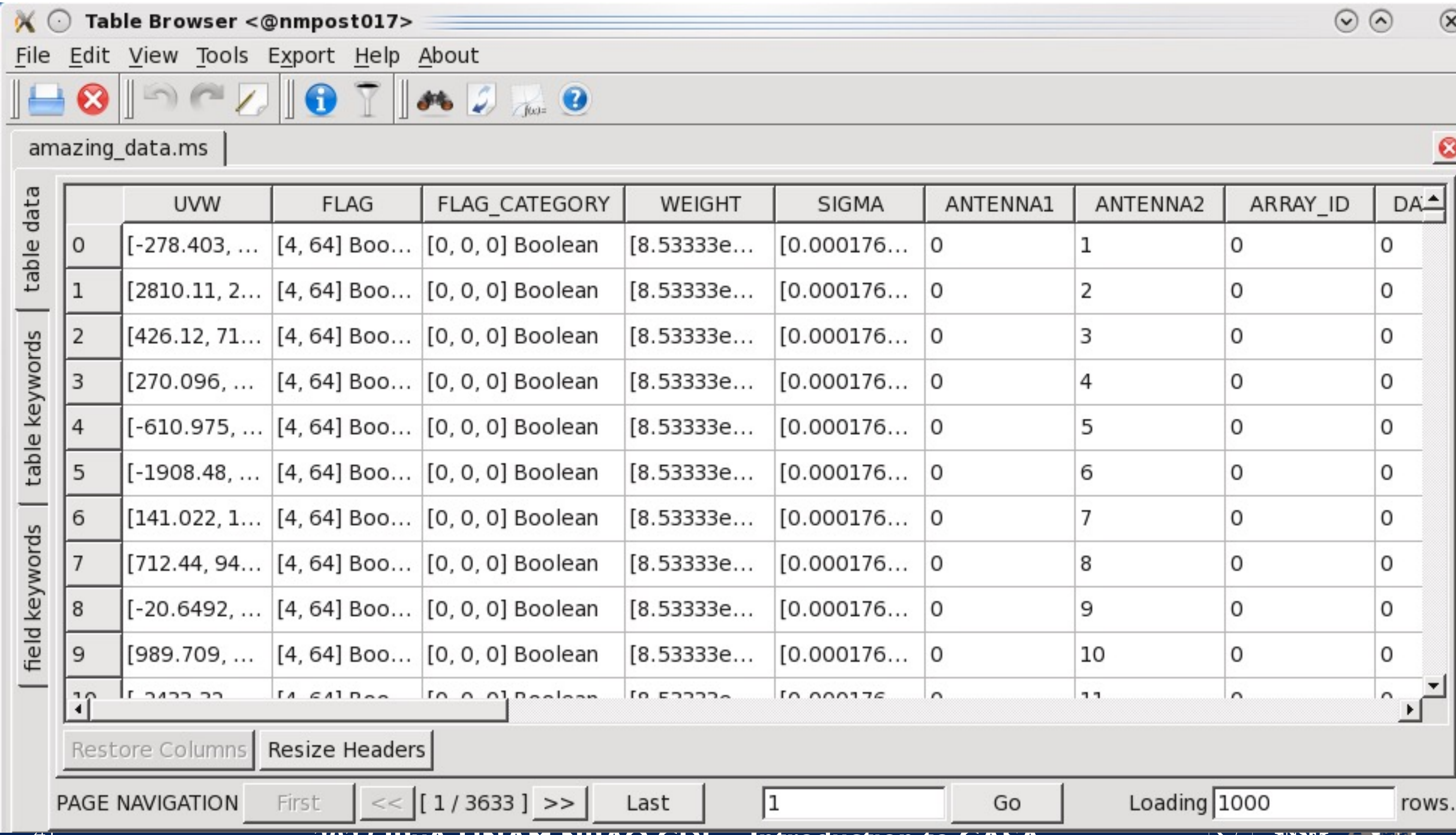


```
CASA <81>: !ls amazing_data.ms/FIELD
table.dat  table.f0  table.f0i  table.info  table.lock
```

- More on the MS: https://casadocs.readthedocs.io/en/stable/notebooks/casa-fundamentals.html#MeasurementSet-Basics

MS MAIN table contents

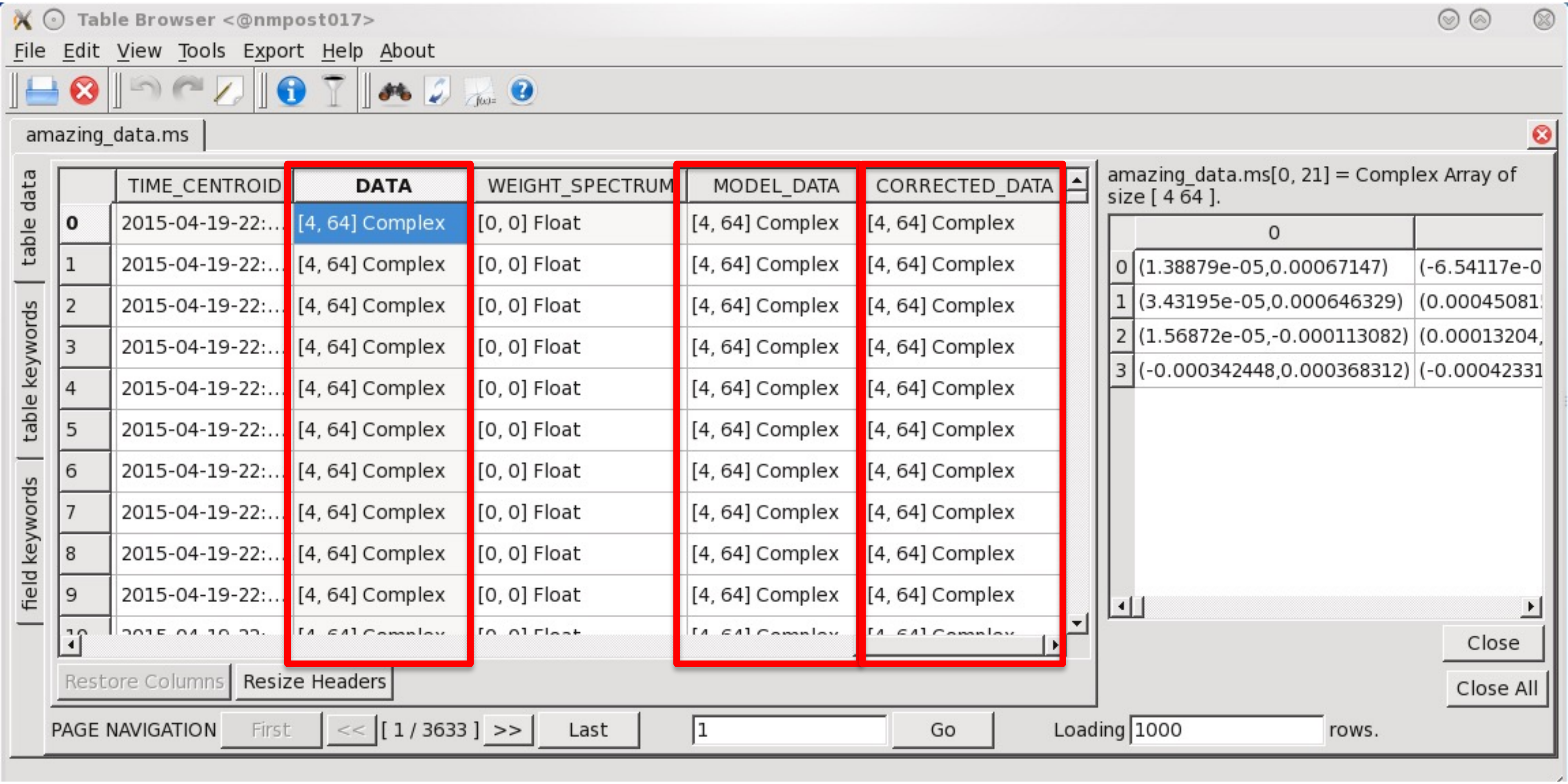
Inspect with task **browsetable**



	UVW	FLAG	FLAG_CATEGORY	WEIGHT	SIGMA	ANTENNA1	ANTENNA2	ARRAY_ID	DA
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	[-2423.22	[4, 64] Boo	[0, 0, 0] Boolean	[8.53333e	[0.000176	0	11	0	0

MS MAIN table contents

Inspect with task **browsetable**



	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

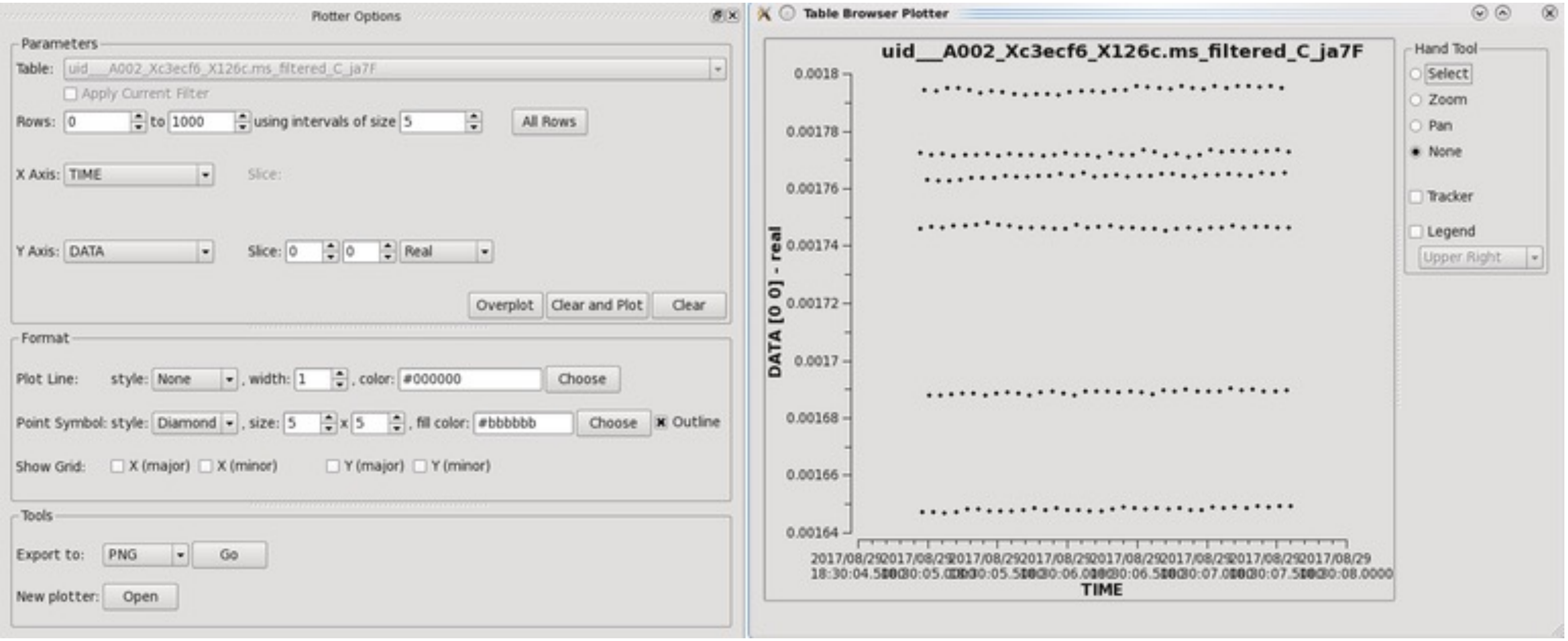
amazing_data.ms[0, 21] = Complex Array of size [4 64].

	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

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

2D plots with the table browser

Plot columns with **browsetable**



Plotter Options

Parameters

Table: uid___A002_Xc3ecf6_X126c.ms_filtered_C_ja7F

□ Apply Current Filter

Rows: 0 to 1000 using intervals of size 5 — All Rows

X Axis: TIME — Slice:

Y Axis: DATA — Slice: 0 0 Real

Overplot — Clear and Plot — Clear

Format

Plot Line: style: None, width: 1, color: #000000 — Choose

Point Symbol: style: Diamond, size: 5 x 5, fill color: #bbbbbb — Choose — ☒ Outline

Show Grid: □ X (major) □ X (minor) □ Y (major) □ Y (minor)

Tools

Export to: PNG — Go

New plotter: Open

Table Browser Plotter

Hand Tool

○ Select

○ Zoom

○ Pan

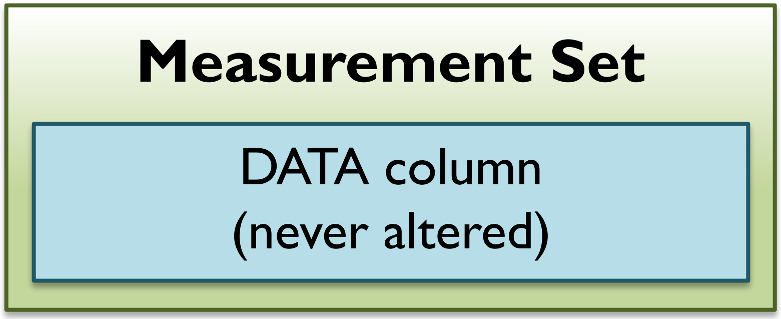
● None

□ Tracker

□ Legend

Upper Right

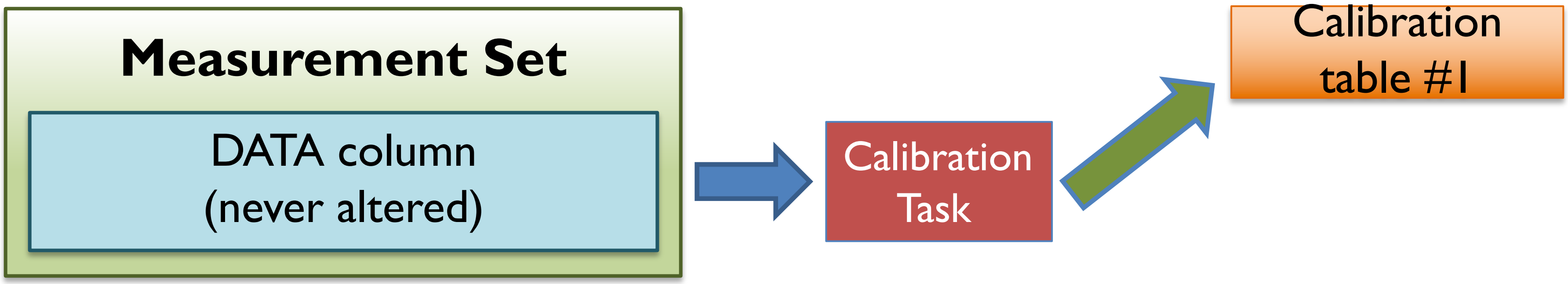
MS columns & calibration tables



Measurement Set

DATA column
(never altered)

MS columns & calibration tables



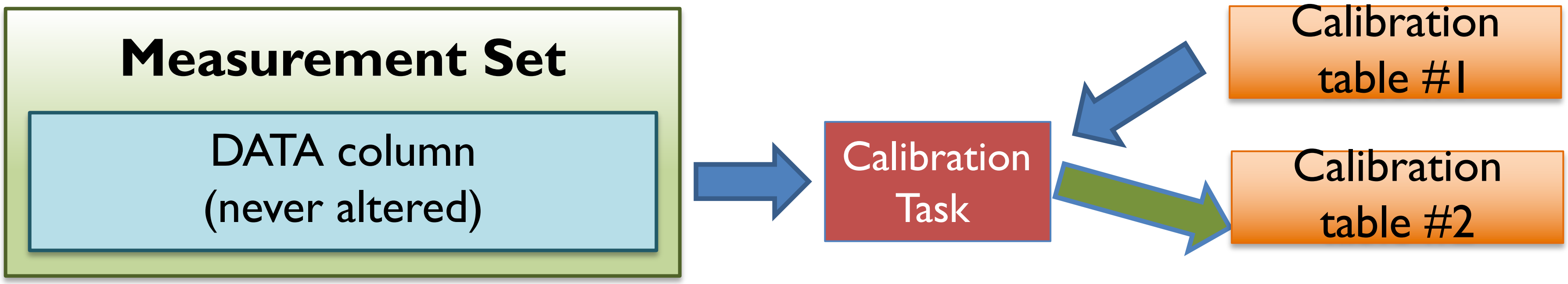
Measurement Set

DATA column
(never altered)

Calibration Task

Calibration table #1

MS columns & calibration tables



Measurement Set

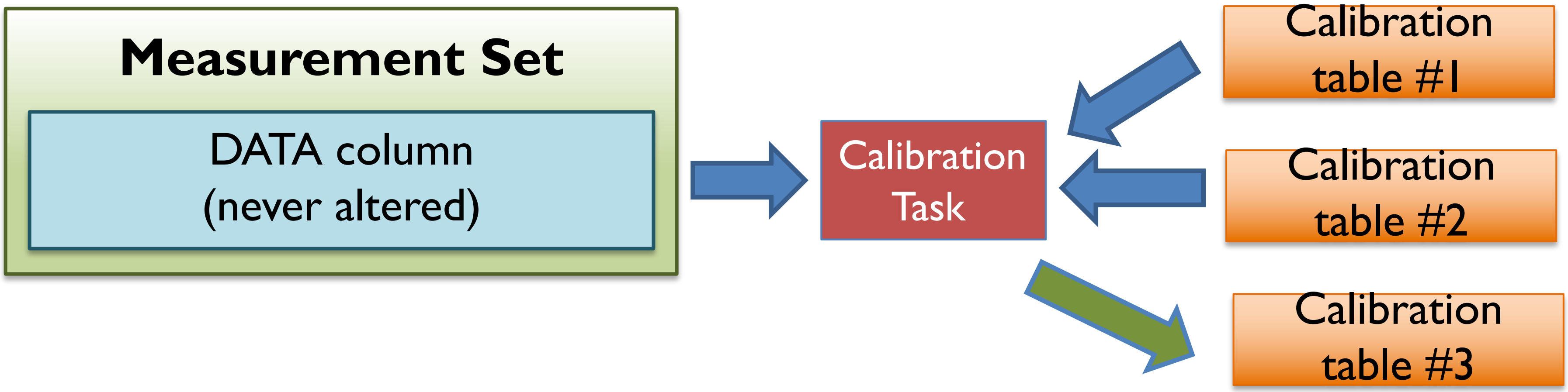
DATA column (never altered)

Calibration Task

Calibration table #1

Calibration table #2

MS columns & calibration tables



Measurement Set

DATA column
(never altered)

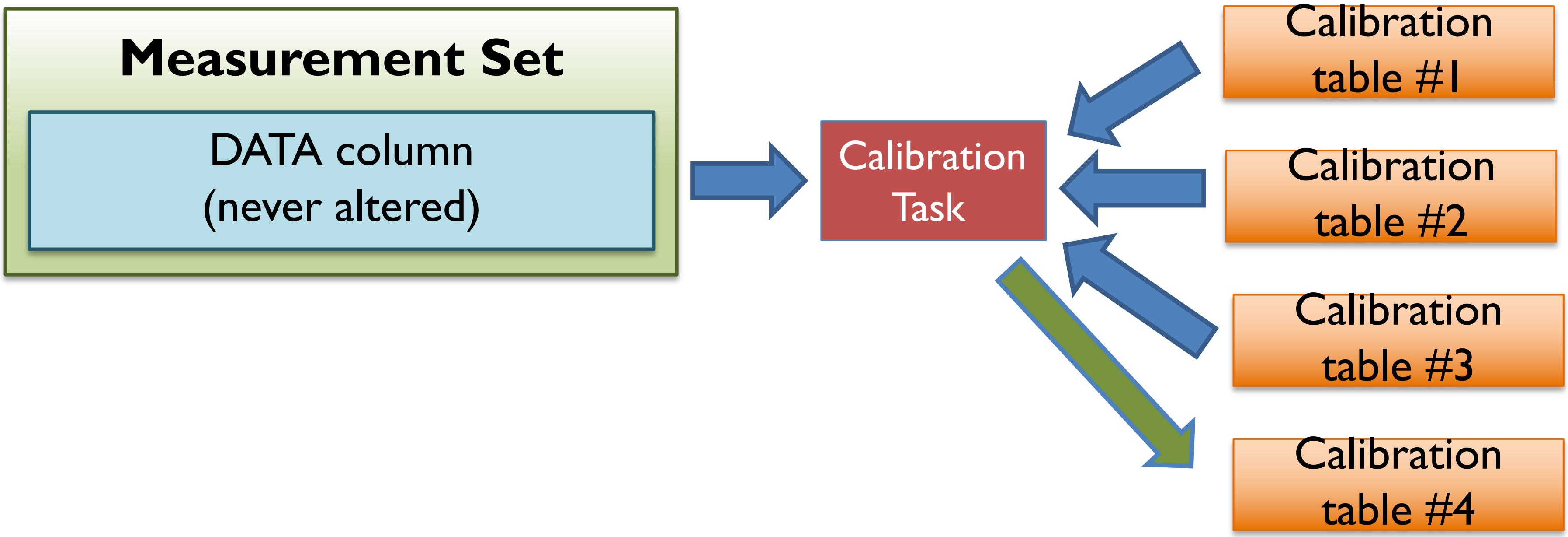
Calibration
Task

Calibration
table #1

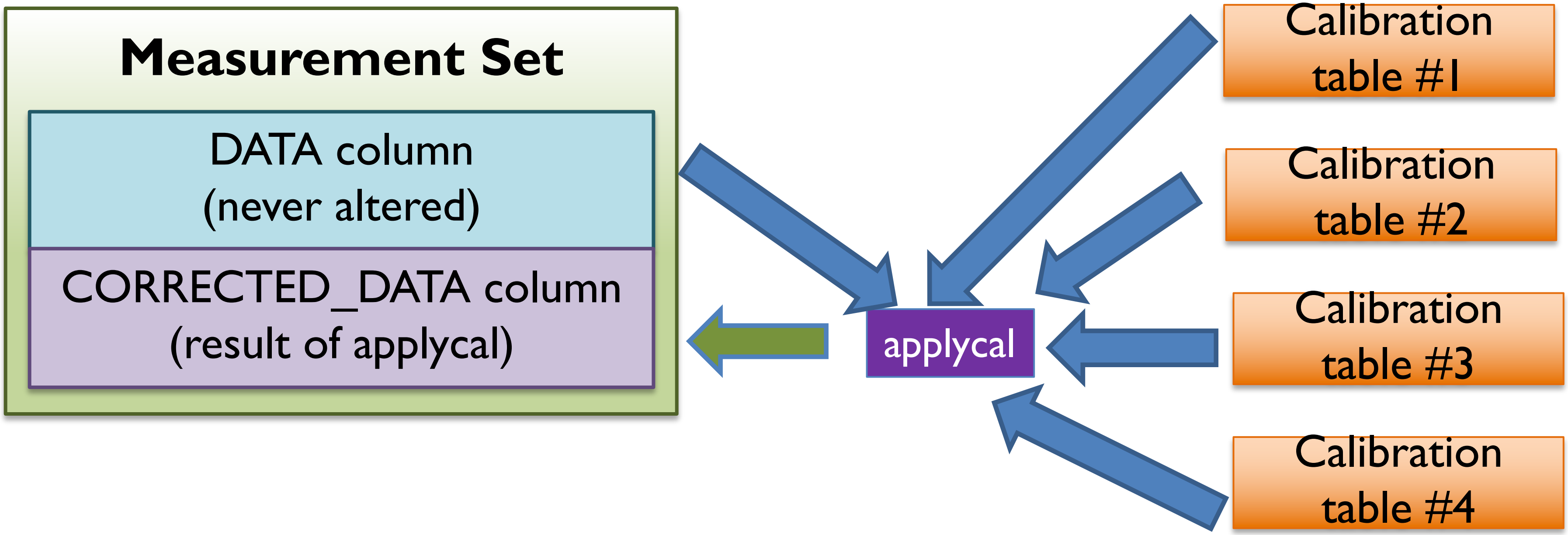
Calibration
table #2

Calibration
table #3

MS columns & calibration tables



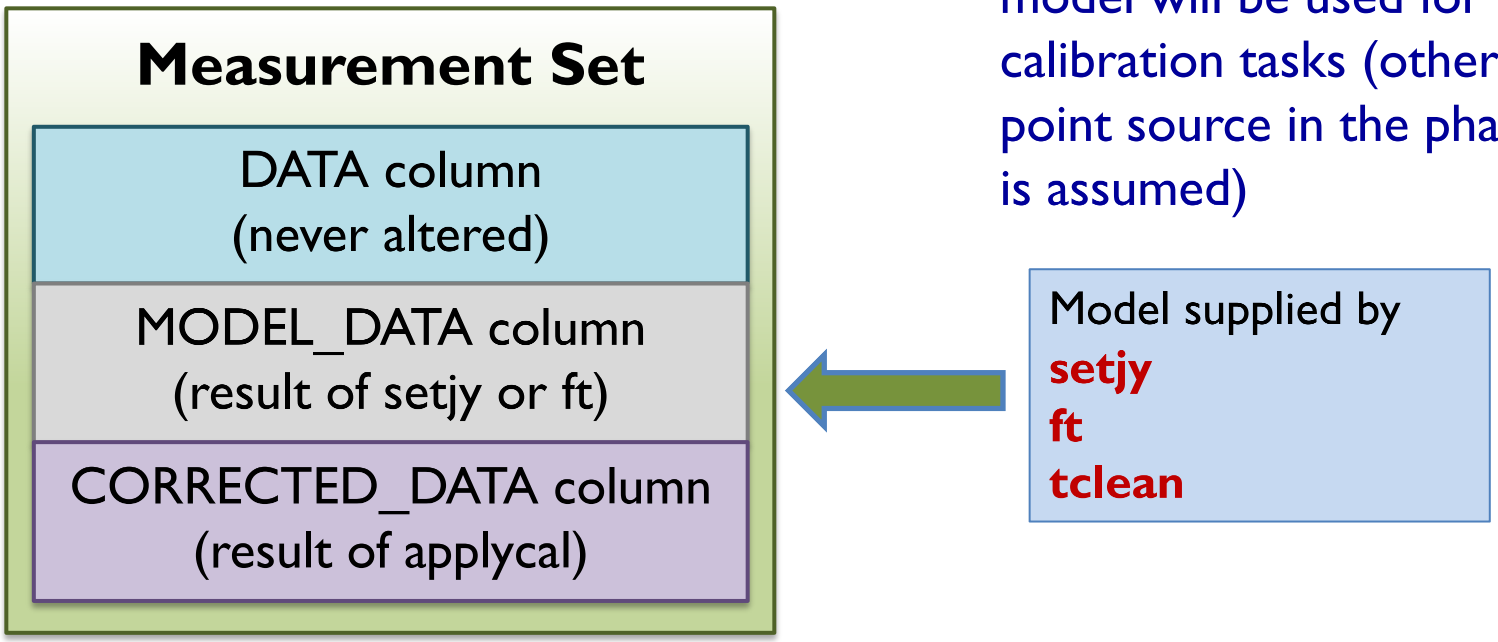
MS columns & calibration tables



If CORRECTED_DATA exists,
applycal will overwrite

MS columns & calibration tables

If a model is supplied in the MODEL_DATA column, the model will be used for the calibration tasks (otherwise a point source in the phase center is assumed)



Measurement Set

DATA column
(never altered)

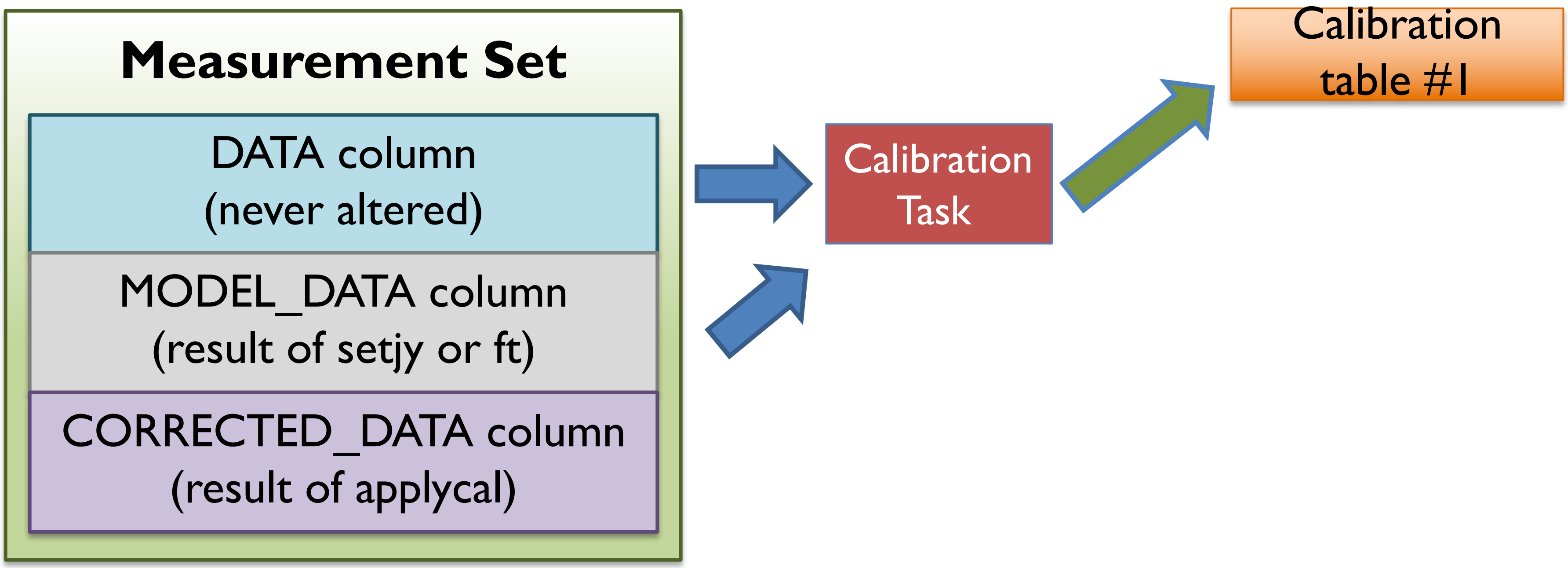
MODEL_DATA column
(result of setjy or ft)

CORRECTED_DATA column
(result of applycal)

Model supplied by
setjy
ft
tclean

(note: data size tripled)

MS columns & calibration tables



Measurement Set

DATA column
(never altered)

MODEL_DATA column
(result of setjy or ft)

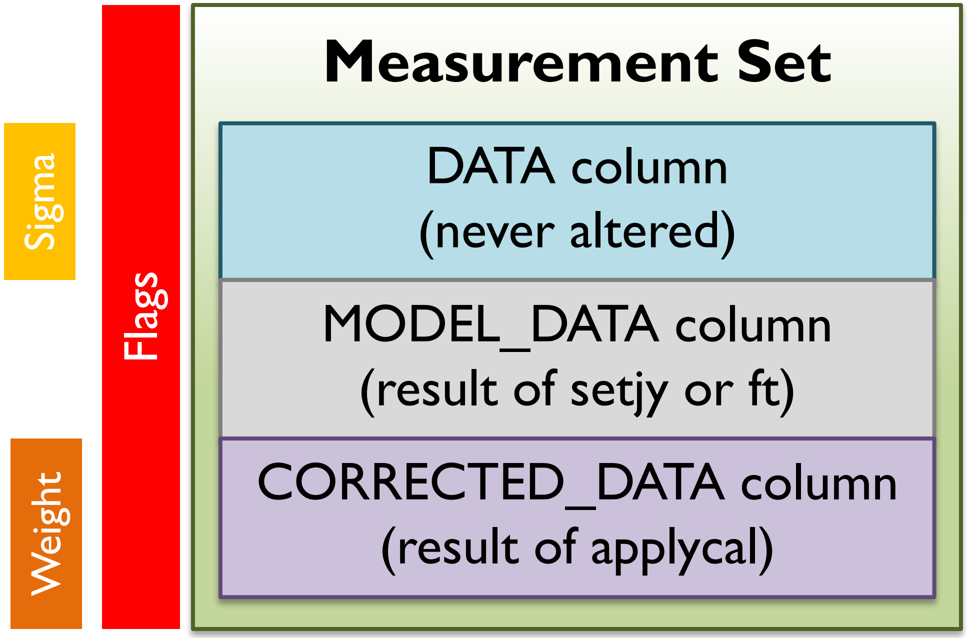
CORRECTED_DATA column
(result of applycal)

Calibration
Task

Calibration
table #1

(note: data size tripled)

MS columns & calibration tables



Sigma

Weight

Flags

Measurement Set

DATA column
(never altered)

MODEL_DATA column
(result of setjy or ft)

CORRECTED_DATA column
(result of applycal)

Data columns will be created by the tasks that need them. **clearcal** can reset them

Flags can also be saved in **<MS>.flagversions** (some tasks create flag backups there)

(note: data size tripled)

https://casadocs.readthedocs.io/en/stable/notebooks/data_weights.html

Listobs

- You can select subset of visibilities to perform actions on:
  - Antennas, baselines, frequencies, time, polarization, etc.
  - IDs are provided for almost all quantities, simply numbering through all items, starting with 0 (e.g. antennas, scans, …); check with **listobs**

```
    ================================================================================
      Observer: Dr. Alfred Nobel     Project: uid://evla/pdb/35621723
    Observation: EVLA
    Computing scan and subscan properties...
    Data records: 1137240       Total elapsed time = 8760 seconds
      Observed from   04-Oct-2018/06:04:00.0   to   04-Oct-2018/08:30:00.0 (UTC)

      ObservationID = 0         ArrayID = 0
     Date        Timerange (UTC)          Scan  FldId FieldName             nRows     SpwIds   Average Interval(s)
ScanIntent
     04-Oct-2018/06:04:00.0 - 06:18:45.0     8      0 3C75                   126360  [0,1,2,3,4,5,6,7]  [19.7, 19.7, 19.7, 19.7,
19.7, 19.7, 19.7, 19.7] [OBSERVE_TARGET#UNSPECIFIED]
                06:20:15.0 - 06:35:05.0    10      0 3C75               126360  [0,1,2,3,4,5,6,7]  [19.7, 19.7, 19.7, 19.7, 19.7,
19.7, 19.7, 19.7] [OBSERVE_TARGET#UNSPECIFIED]
                06:36:25.0 - 06:51:20.0    12      0 3C75               126360  [0,1,2,3,4,5,6,7]  [19.9, 19.9, 19.9, 19.9, 19.9,
19.9, 19.9, 19.9] [OBSERVE_TARGET#UNSPECIFIED]
                06:52:35.0 - 07:07:30.0    14      0 3C75               126360  [0,1,2,3,4,5,6,7]  [19.9, 19.9, 19.9, 19.9, 19.9,
19.9, 19.9, 19.9] [OBSERVE_TARGET#UNSPECIFIED]
                07:08:50.0 - 07:23:40.0    16      0 3C75               126360  [0,1,2,3,4,5,6,7]  [19.8, 19.8, 19.8, 19.8, 19.8,
19.8, 19.8, 19.8] [OBSERVE_TARGET#UNSPECIFIED]
                07:26:30.0 - 07:41:25.0    18      0 3C75               126360  [0,1,2,3,4,5,6,7]  [19.9, 19.9, 19.9, 19.9, 19.9,
19.9, 19.9, 19.9] [OBSERVE_TARGET#UNSPECIFIED]
```

Listobs

```
          (nRows = Total number of rows per scan)
Fields: 1
  ID   Code Name            RA              Decl          Epoch   SrcId      nRows
  0    NONE 3C75            02:57:42.630000 +06.01.04.80000 J2000   0        1137240
Spectral Windows:  (8 unique spectral windows and 1 unique polarization setups)
  SpwID  Name          #Chans   Frame   Ch0(MHz)  ChanWid(kHz)  TotBW(kHz) CtrFreq(MHz) BBC Num  Corrs
  0      EVLA_S#A0C0#2    13    TOPO    2503.000    8000.000    104000.0   2551.0000      12  RR  RL  LR  LL
  1      EVLA_S#A0C0#3    13    TOPO    2631.000    8000.000    104000.0   2679.0000      12  RR  RL  LR  LL
  2      EVLA_S#A0C0#4    13    TOPO    2759.000    8000.000    104000.0   2807.0000      12  RR  RL  LR  LL
  3      EVLA_S#A0C0#5    13    TOPO    2887.000    8000.000    104000.0   2935.0000      12  RR  RL  LR  LL
  4      EVLA_S#A0C0#6    13    TOPO    3015.000    8000.000    104000.0   3063.0000      12  RR  RL  LR  LL
  5      EVLA_S#A0C0#7    13    TOPO    3143.000    8000.000    104000.0   3191.0000      12  RR  RL  LR  LL
  6      EVLA_S#A0C0#8    13    TOPO    3271.000    8000.000    104000.0   3319.0000      12  RR  RL  LR  LL
  7      EVLA_S#A0C0#9    13    TOPO    3399.000    8000.000    104000.0   3447.0000      12  RR  RL  LR  LL
Sources: 8
  ID   Name              SpwId RestFreq(MHz)  SysVel(km/s)
  0    3C75             0    -            -
  0    3C75             1    -            -
  0    3C75             2    -            -
  0    3C75             3    -            -
  0    3C75             4    -            -
  0    3C75             5    -            -
  0    3C75             6    -            -
  0    3C75             7    -            -
Antennas: 27:
  ID   Name  Station   Diam.    Long.         Lat.              Offset from array center (m)             ITRF Geocentric
coordinates (m)
                                                    East         North      Elevation
```