Introduction to CASA

Jennifer Donovan Meyer

CASA
Common Astronomy Software Applications

NRAO NSF AUI

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 5.7/6.1 [5.8/6.2 to be released soon]
  - CASA 5.X = python 2, CASA 6.X = python 3
- But we use CASA 6.1.2 for this workshop (a patch of 6.1: 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., 6.1.2)
  - ALMA: October 1, VLA a bit later

- Latest version CASA 5.7/6.1 is internally tested on:
  - Red Hat Linux 6 and 7
  - macOS 10.14 and 10.15
  - also works on other Linux systems, but not tested in house

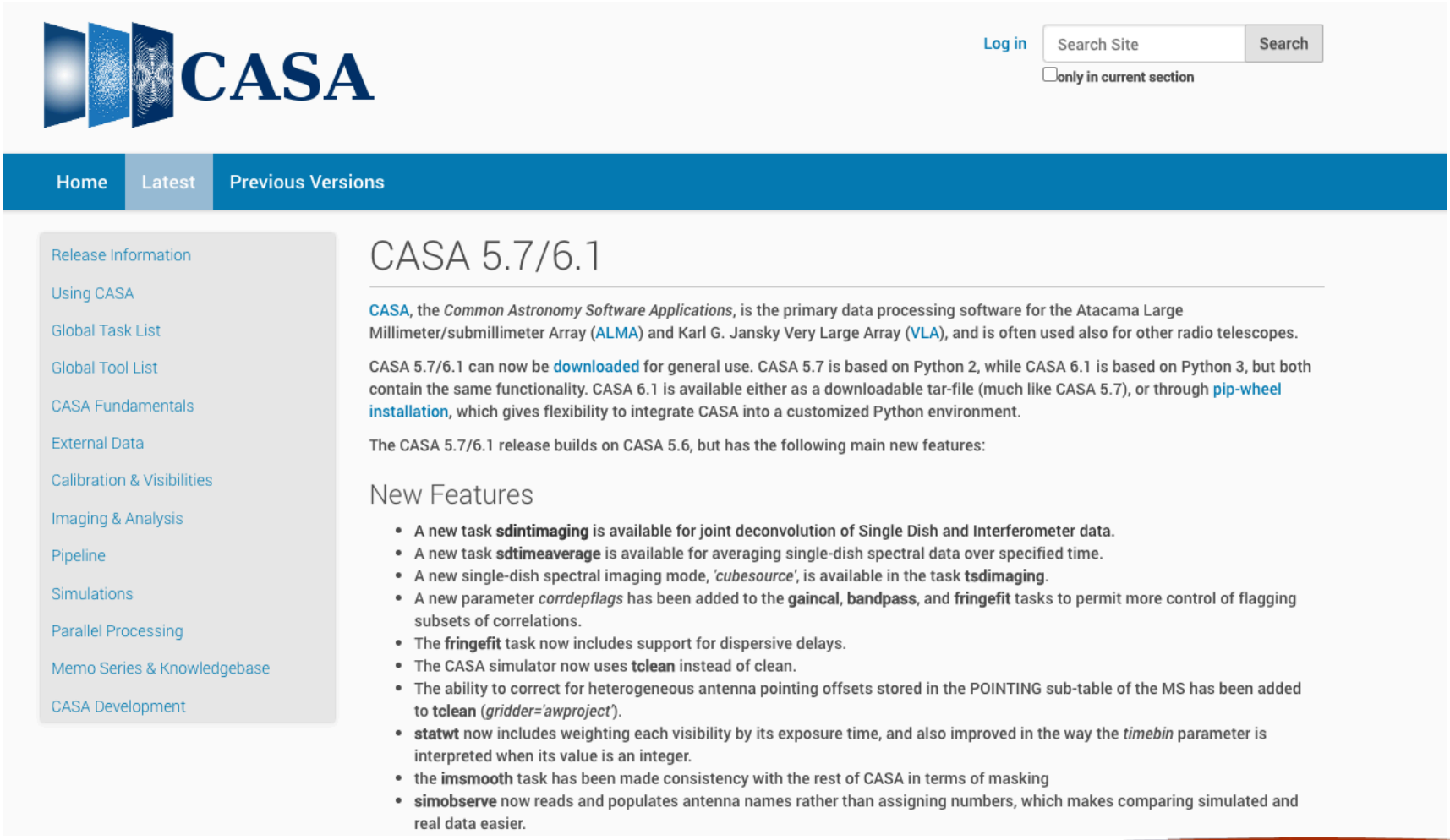
CASA documentation and web resources

CASA Homepage http://casa.nrao.edu

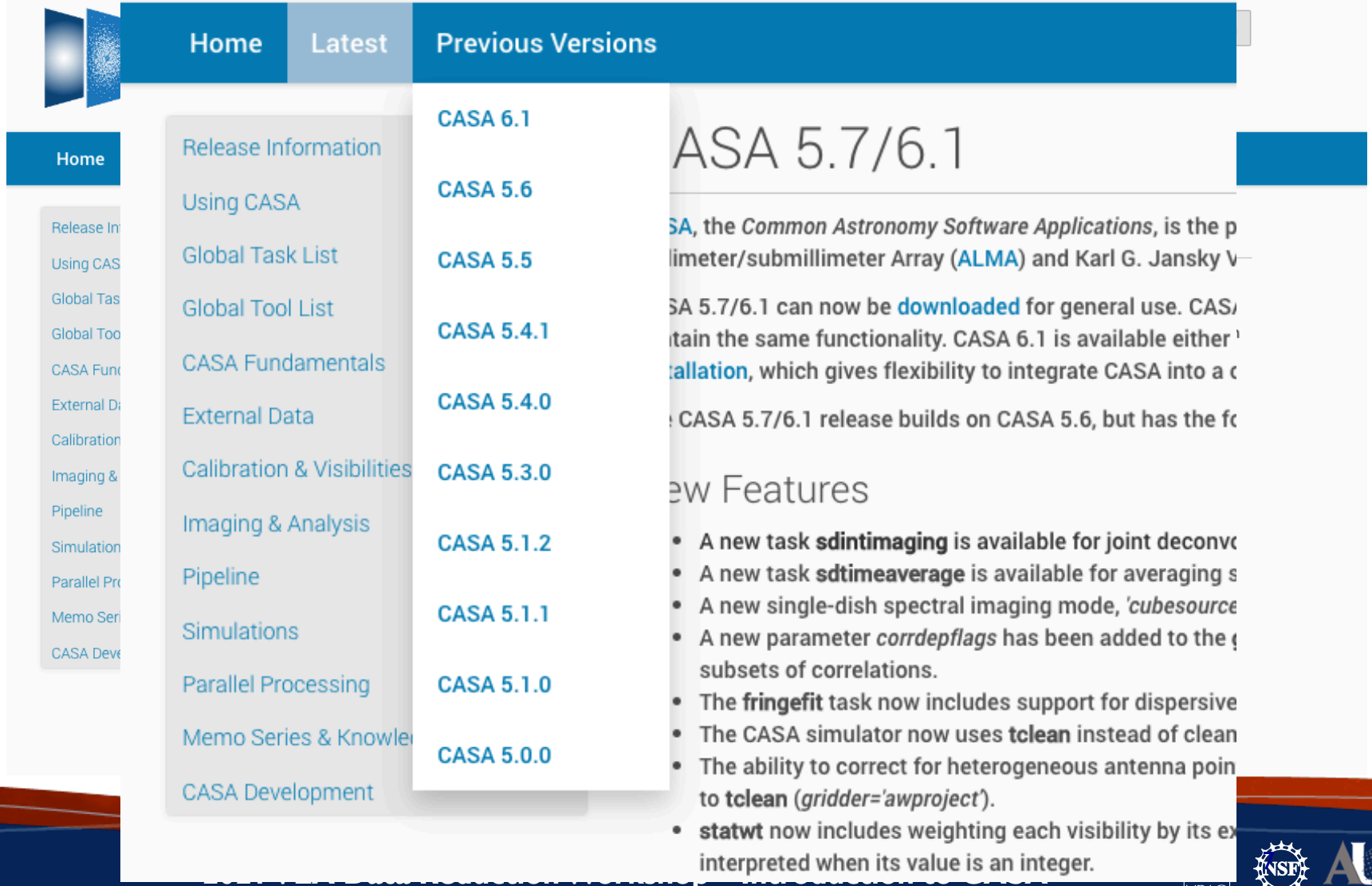
CASA, the *Common Astronomy Software Applications* package, is the primary data processing software for the Atacama Large Millimeter/submillimeter Array (ALMA) and NSF's Karl G. Jansky Very Large Array (VLA), and is frequently used also for other radio telescopes. The CASA software can process data from 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://casa.nrao.edu/casadocs



CASA


Home
Latest
Previous Versions
Release Information
Using CASA
Global Task List
Global Tool List
CASA Fundamentals
External Data
Calibration & Visibilities
Imaging & Analysis
Pipeline
Simulations
Parallel Processing
Memo Series & Knowle
CASA Development
CASA 6.1
CASA 5.6
CASA 5.5
CASA 5.4.1
CASA 5.4.0
CASA 5.3.0
CASA 5.1.2
CASA 5.1.1
CASA 5.1.0
CASA 5.0.0
ASA 5.7/6.1
SA, the Common Astronomy Software Applications, is the p
limeter/submillimeter Array (ALMA) and Karl G. Jansky V
SA 5.7/6.1 can now be downloaded for general use. CASA
tain the same functionality. CASA 6.1 is available either
allation, which gives flexibility to integrate CASA into a c
CASA 5.7/6.1 release builds on CASA 5.6, but has the fo
ew Features
A new task sdintimaging is available for joint deconvo
A new task sdtimeaverage is available for averaging s
A new single-dish spectral imaging mode, 'cubesource
A new parameter corrdepflags has been added to the g
subsets of correlations.
The fringefit task now includes support for dispersive
The CASA simulator now uses tclean instead of clean
The ability to correct for heterogeneous antenna poin
to tclean (gridder='awproject').
statwt now includes weighting each visibility by its ex
interpreted when its value is an integer.

CASA

Log in | Search Site | Search | only in current section

Home | Latest | Previous Versions

Release Information
- Release Notes CASA 5.7 / 6.1
- Known Issues

Using CASA
Global Task List
Global Tool List
CASA Fundamentals
External Data
Calibration & Visibilities
Imaging & Analysis
Pipeline
Simulations
Parallel Processing
Memo Series & Knowledgebase
CASA Development

Release Notes CASA 5.7 / 6.1

Summary of new features in this release of CASA

This is the documentation for CASA 5.7/6.1. Changes compared to the CASA 5.6/6.0 release are listed below.

CASA 6.1.2 (5.7.2)

CASA 6.1.2 includes a pipeline that has been validated for the VLA. CASA 6.1.2 and included pipeline are functionally equivalent to CASA 6.1.1, and have the following added features:

- A script for porting the **tec_maps** for ionospheric calibration of VLA data has been included in version 6.1.2. To follow new server conventions at NASA's Crustal Dynamics Data Information System (CDDIS), the TEC date retrieval mechanism within the tec_maps script has been updated. See the Ionospheric Corrections section of CASA Docs for details.
- In the task **polcal**: (a) poltypes *'Df'* and *'Df+QU'* have been restored to operability (they were accidentally disabled by the introduction of correlation-dependent flagging in gain-like calibration in 5.7.0/6.1.0); (b) In *poltype='Xfparang+QU'*, the solution process is now less vulnerable to fully-flagged scans defeating the solution.

Warning! The **simalma** bug-fix implemented in CASA 6.1.1 is **not** available in CASA 6.1.2 or 5.7.2. Please use CASA 6.1.1 for running task simalma.

A CASA 5.7.2 version that includes all the 6.1.2 and 6.1.1 updates is also available, but does not include any pipeline.

CASA 6.1.1

05 March 2021: The following information on CASA 6.1.1 is in anticipation of the upcoming 6.1.1 release. The information is not final until 6.1.1 has been officially released, which will be *after* the CASA 6.1.2 release.

CASA 6.1.1 includes a pipeline that has been validated for ALMA. CASA 6.1.1 and its included pipeline is functionally equivalent to CASA 6.1.0, and has the following added features:

CASA documentation and web resources

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

main page | discussion | view source | history

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

- Main Page

search

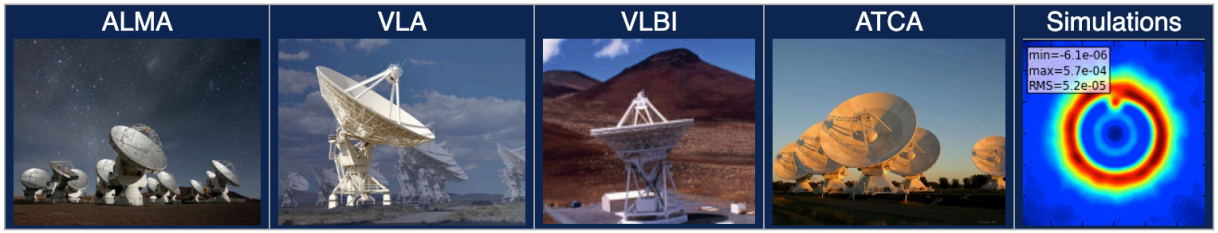
Search

Go | Search

tools

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

CASA Tutorials



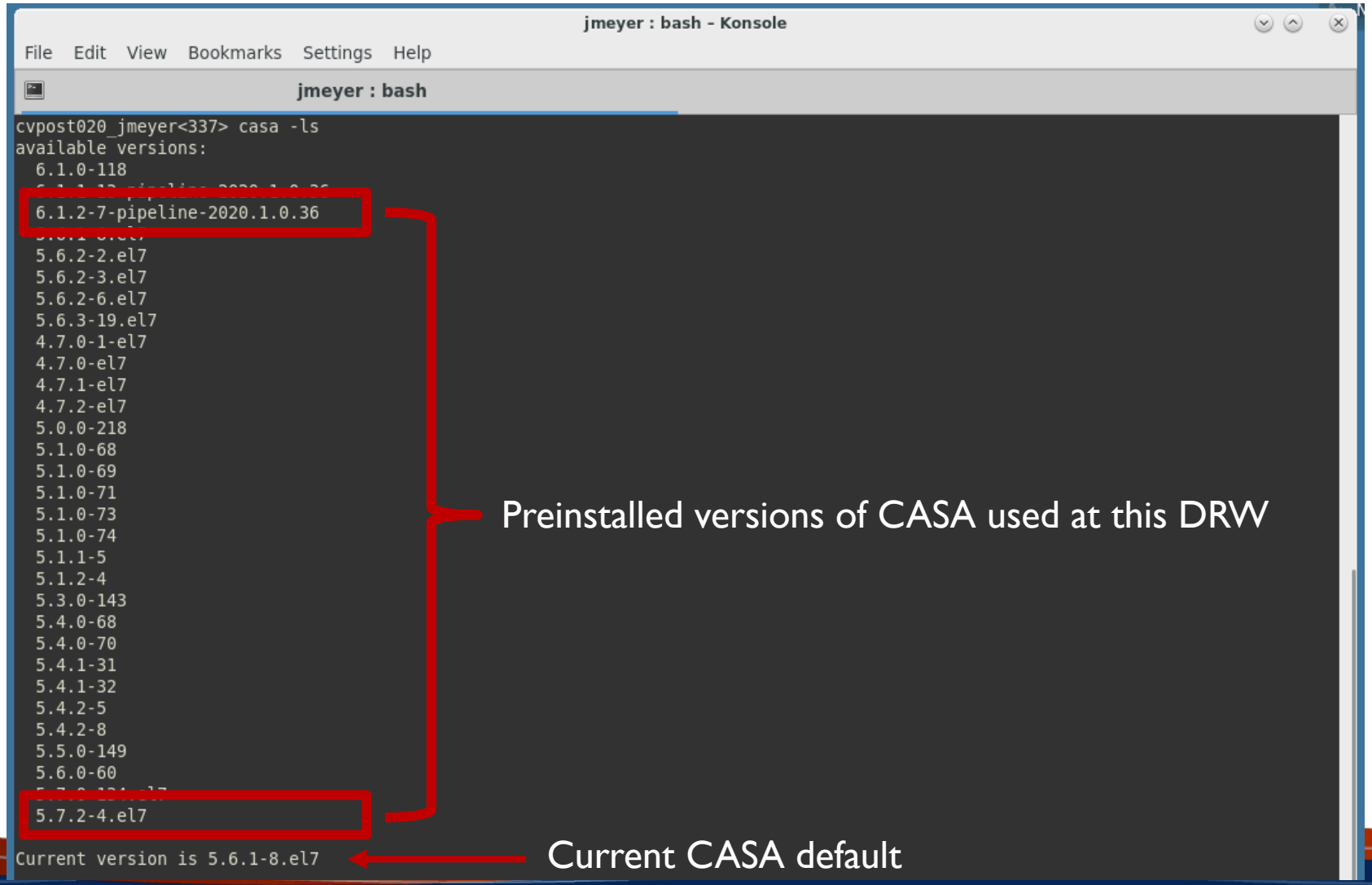
Extracting Scripts from Tutorials

Information for authors: MediaWiki markup language CASAGuides Instructions for Authors

537 articles since July 2009.

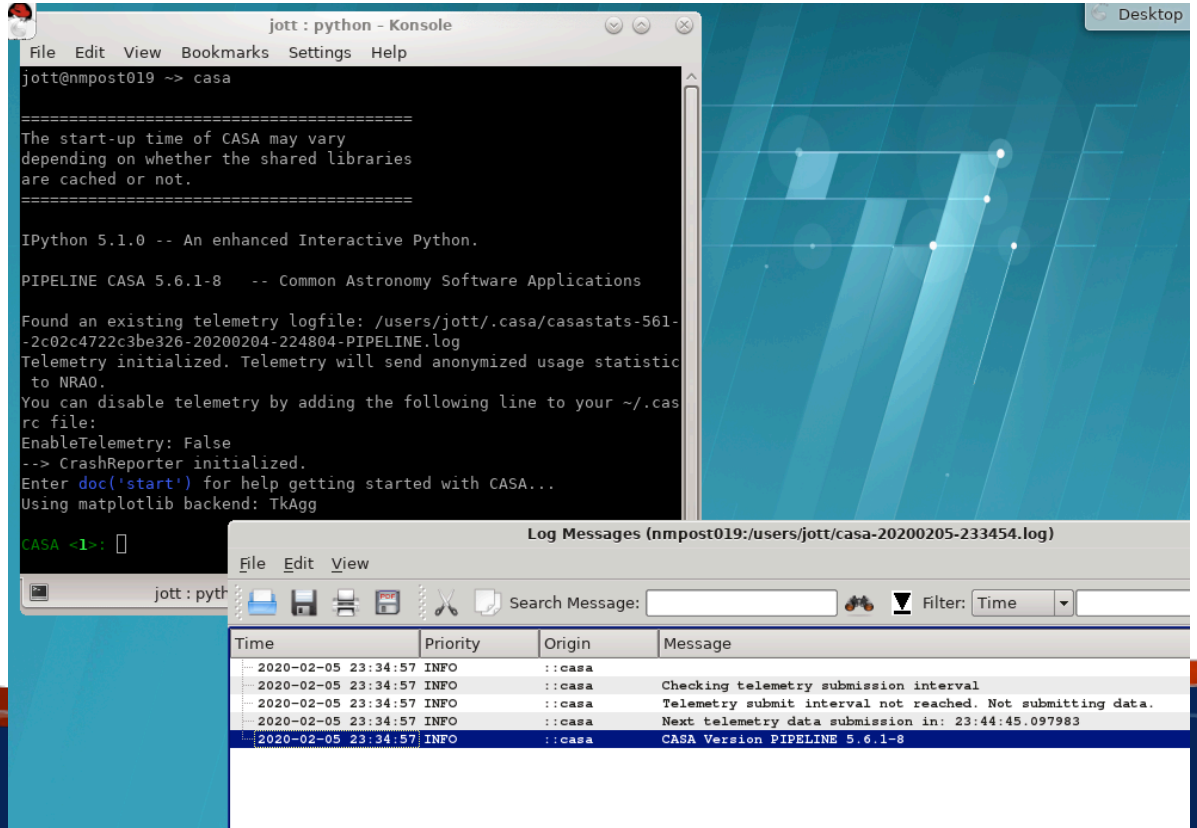
Starting Monolithic CASA

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



Starting Monolithic CASA

- Start CASA from the UNIX shell: **casa –r 6.1.2-7-pipeline-2020.1.0.36**
- (with pipeline: casa –r 6.1.2-7-pipeline-2020.1.0.36 --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



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!
  - 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 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

CASA

Log in | Search Site | Search | only in current section

Home | Latest | Previous Versions

- Release Information
- **Global Task List**
- Global Tool List
- CASA Fundamentals
- Using CASA
- Calibration & Visibilities
- Imaging & Analysis
- Pipeline
- Simulations
- Parallel Processing
- Memo Series
- CASA Development

tclean | Description | Parameters | Changelog | Examples | Developer | Planning

Description

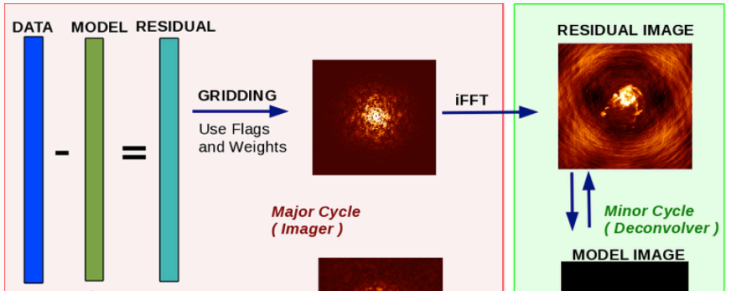
task tclean description

Overview

The **tclean** task forms images from visibilities and reconstructs a sky model.

tclean handles continuum images and spectral line cubes, supports outlier fields, contains point-source CLEAN based algorithms as well as options 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). Parallelization of the major cycle is also available.

Image reconstruction in CASA typically comprises an outer loop of *major cycles* and an inner loop of *minor cycles*. The major cycle implements transforms between the data and image domains and the minor cycle operates purely in the image domain. Together, they implement an iterative weighted χ^2 minimization that solves the measurement equation. Minor cycle algorithms can have their own (different) optimization schemes and the imaging framework and task interface allow for considerable freedom in choosing options separately for each step of the process.



Task help

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

Release Information
Global Task List
Global Tool List
CASA Fundamentals
Using CASA
Calibration & Visibilities
Imaging & Analysis
Pipeline
Simulations
Parallel Processing
Memo Series
CASA Development

tclean | Description | Parameters | Changelog | Examples | Developer | Planning

Parameters

`vis : string stringArray`

Name(s) of input visibility file(s)
default: none;
example: vis='ngc5921.ms'
vis=['ngc5921a.ms','ngc5921b.ms']; multiple MSes

`field : string stringArray`

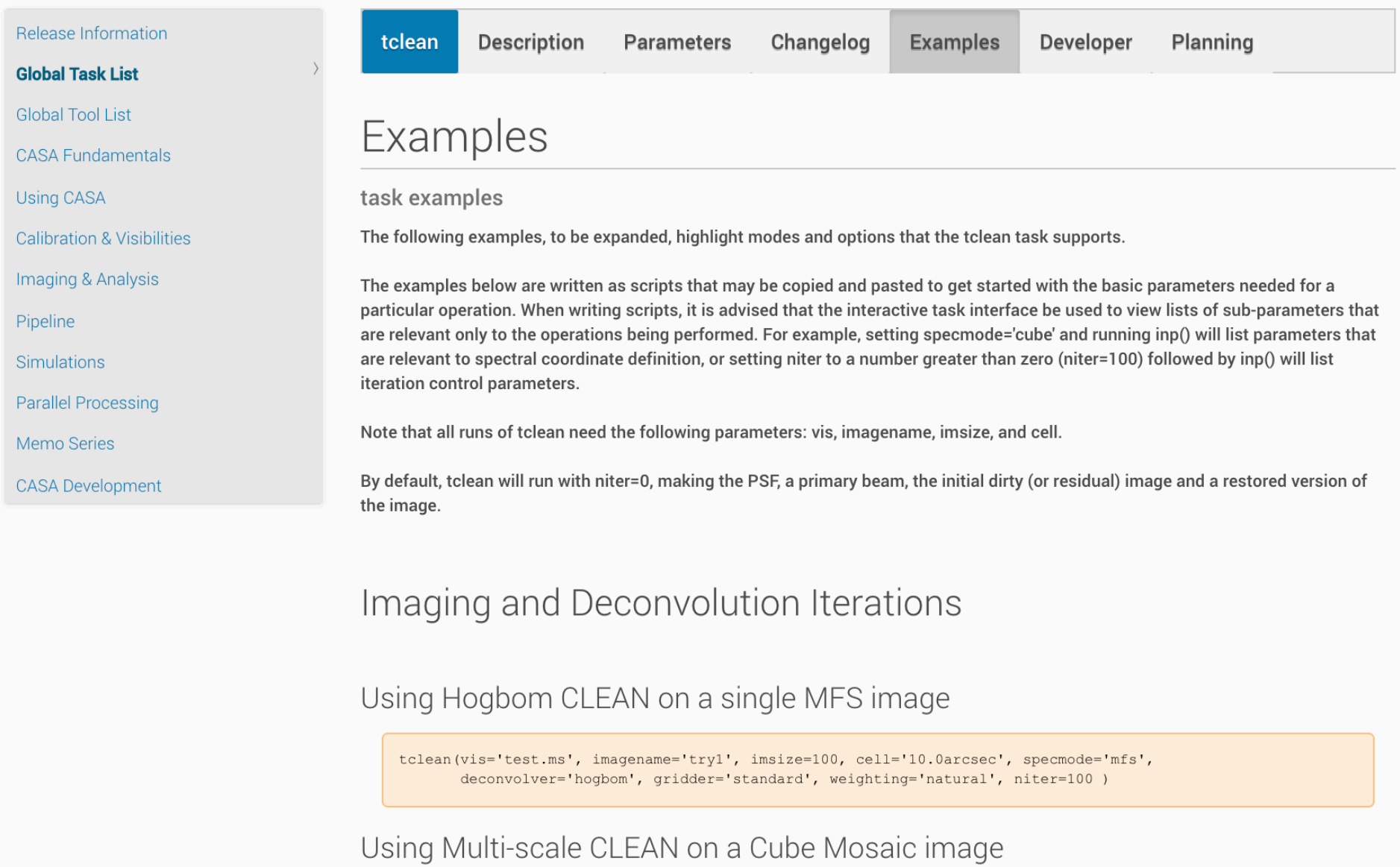
Select fields to image or mosaic. Use field id(s) or name(s).
['go listobs' to obtain the list id's or names]
default: '= all fields
If field string is a non-negative integer, it is assumed to
be a field index otherwise, it is assumed to be a
field name
field='0~2'; field ids 0,1,2
field='0,4,5~7'; field 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
for the second
field = '0~2'; field ids 0-2 for all input MSes

`spw : string stringArray`

Select spectral window/channels
NOTE: channels de-selected here will contain all zeros if
selected by the parameter mode subparameters.
default: '=all spectral windows and channels
spw='0~2,4'; spectral windows 0,1,2,4 (all channels)

Task help

- Examples in CASAdocs



Release Information
Global Task List
Global Tool List
CASA Fundamentals
Using CASA
Calibration & Visibilities
Imaging & Analysis
Pipeline
Simulations
Parallel Processing
Memo Series
CASA Development

tclean | Description | Parameters | Changelog | Examples | Developer | Planning

Examples

task 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.

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

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 (functional 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)

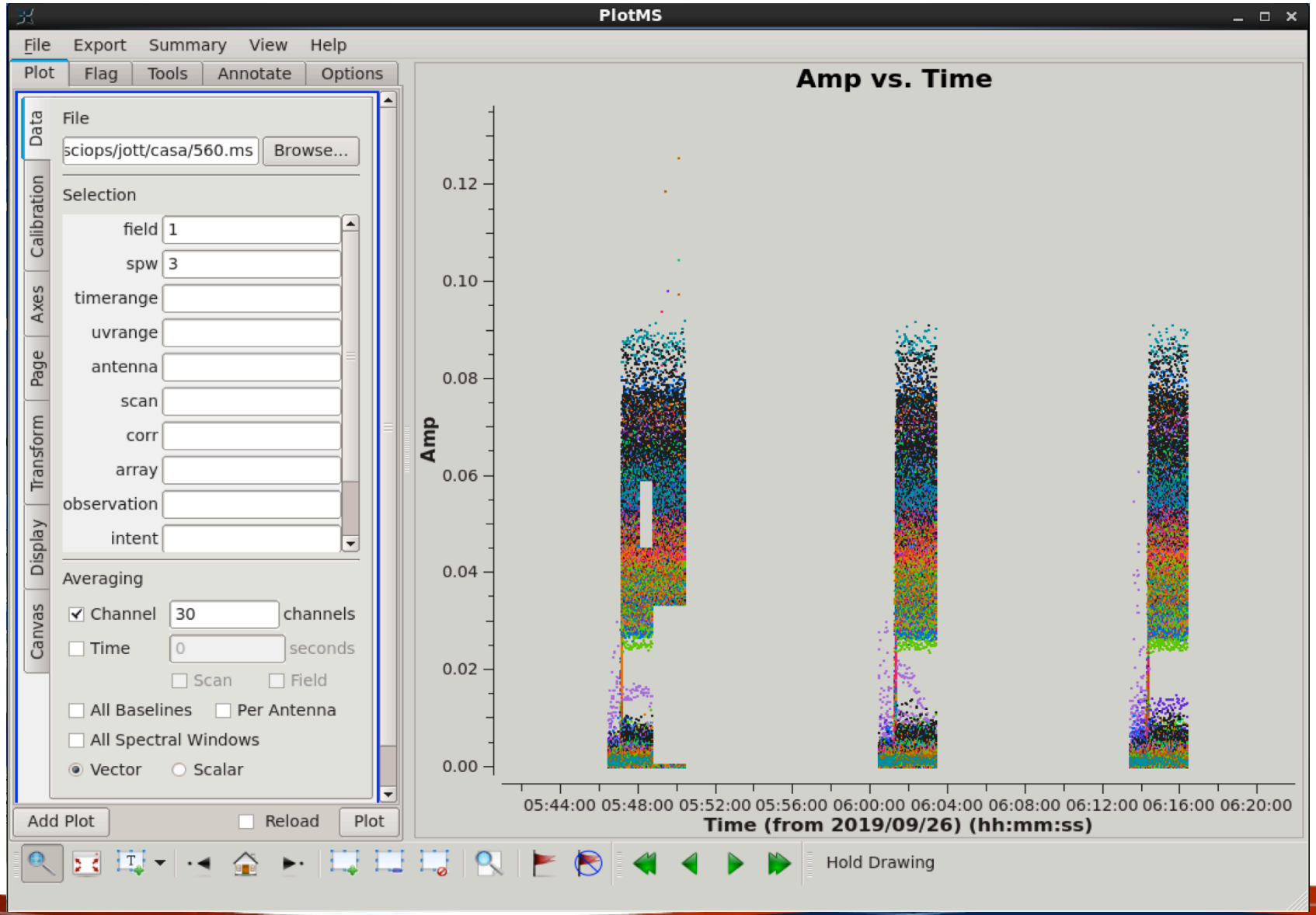
- 3rd parties like the Nordic ALMA ARC node, etc.

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

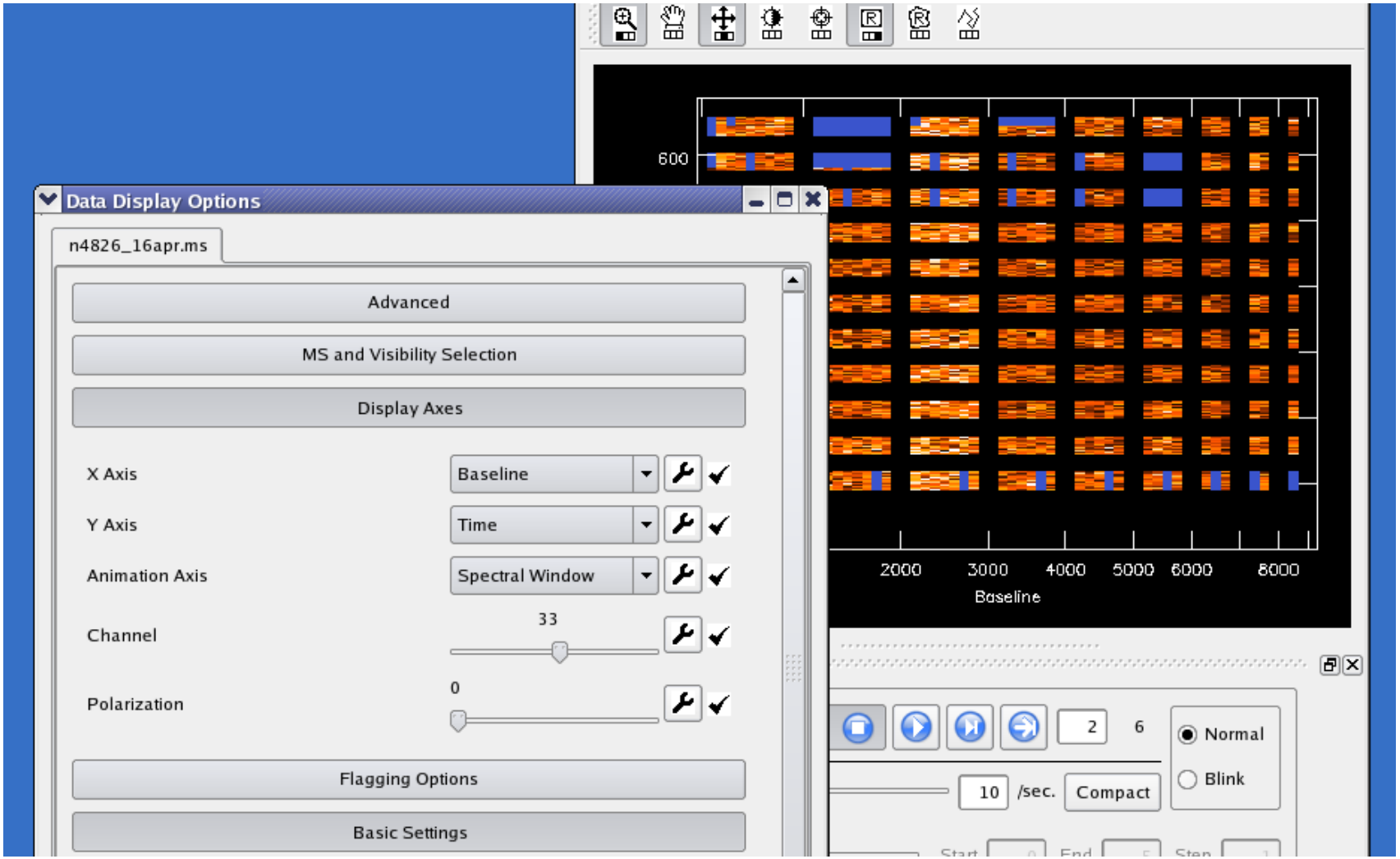
Applications

- 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, viewer, feather, plotants, plotbandpass
- Any CASA (table) data: browsetable
- Images: imview, viewer
- Don't forget about full functionality of python! e.g. matplotlib, astropy, …

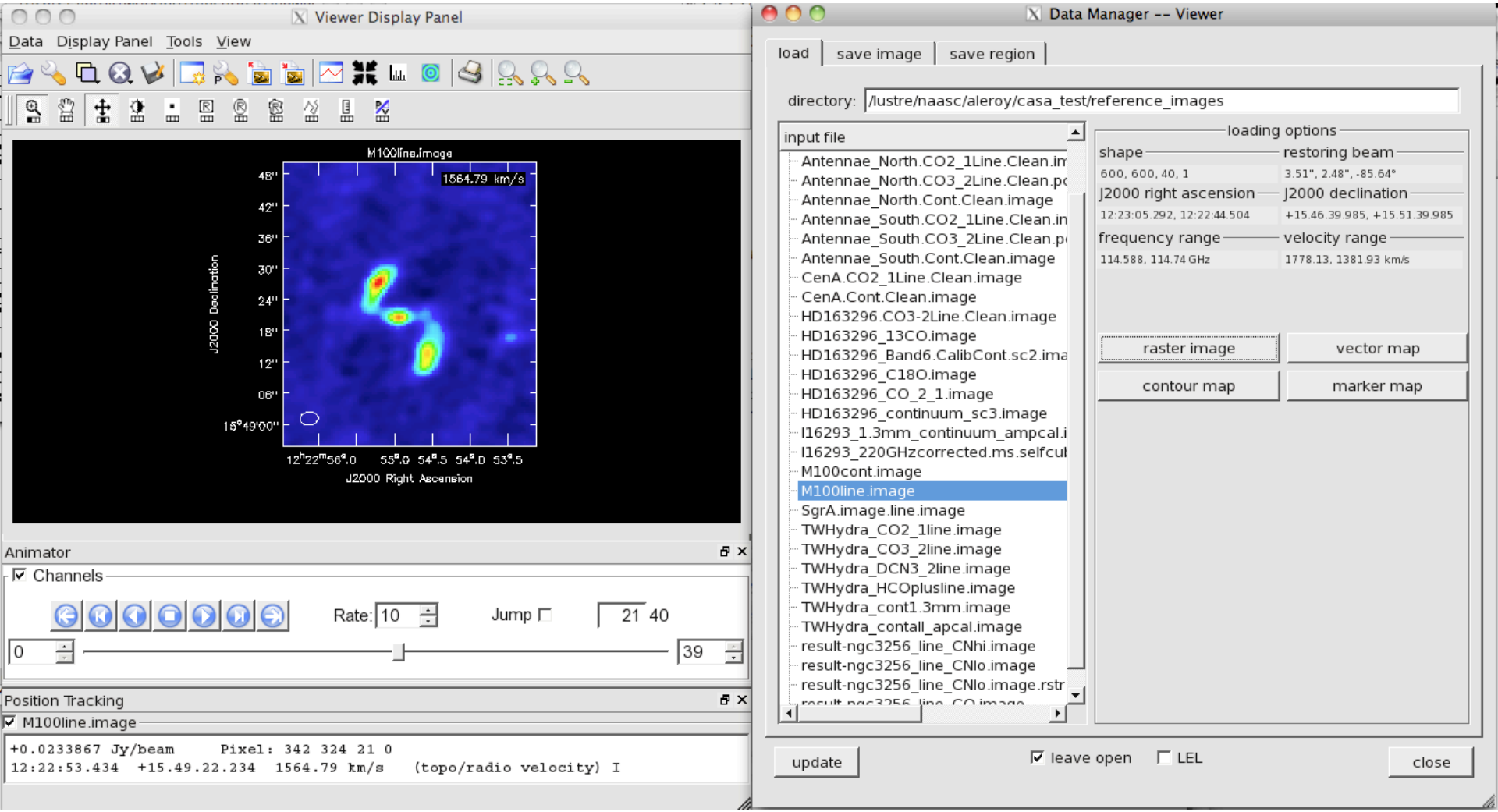
PlotMS


PlotMS
File Export Summary View Help
Plot Flag Tools Annotate Options
Data Calibration Axes Page Transform Display Canvas
File
sciops/jott/casa/560.ms
Browse...
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
Add Plot
Reload
Plot
Amp vs. Time
Amp
0.12
0.10
0.08
0.06
0.04
0.02
0.00
05:44:00 05:48:00 05:52:00 05:56:00 06:00:00 06:04:00 06:08:00 06:12:00 06:16:00 06:20:00
Time (from 2019/09/26) (hh:mm:ss)
Hold Drawing

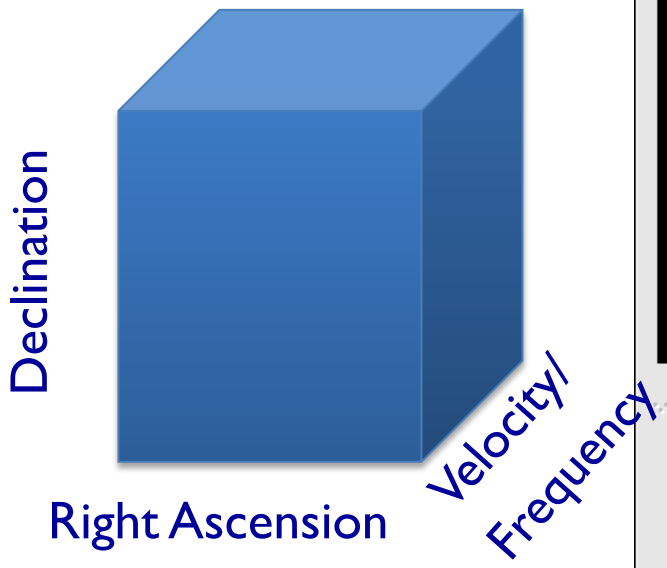
Viewer (msview)

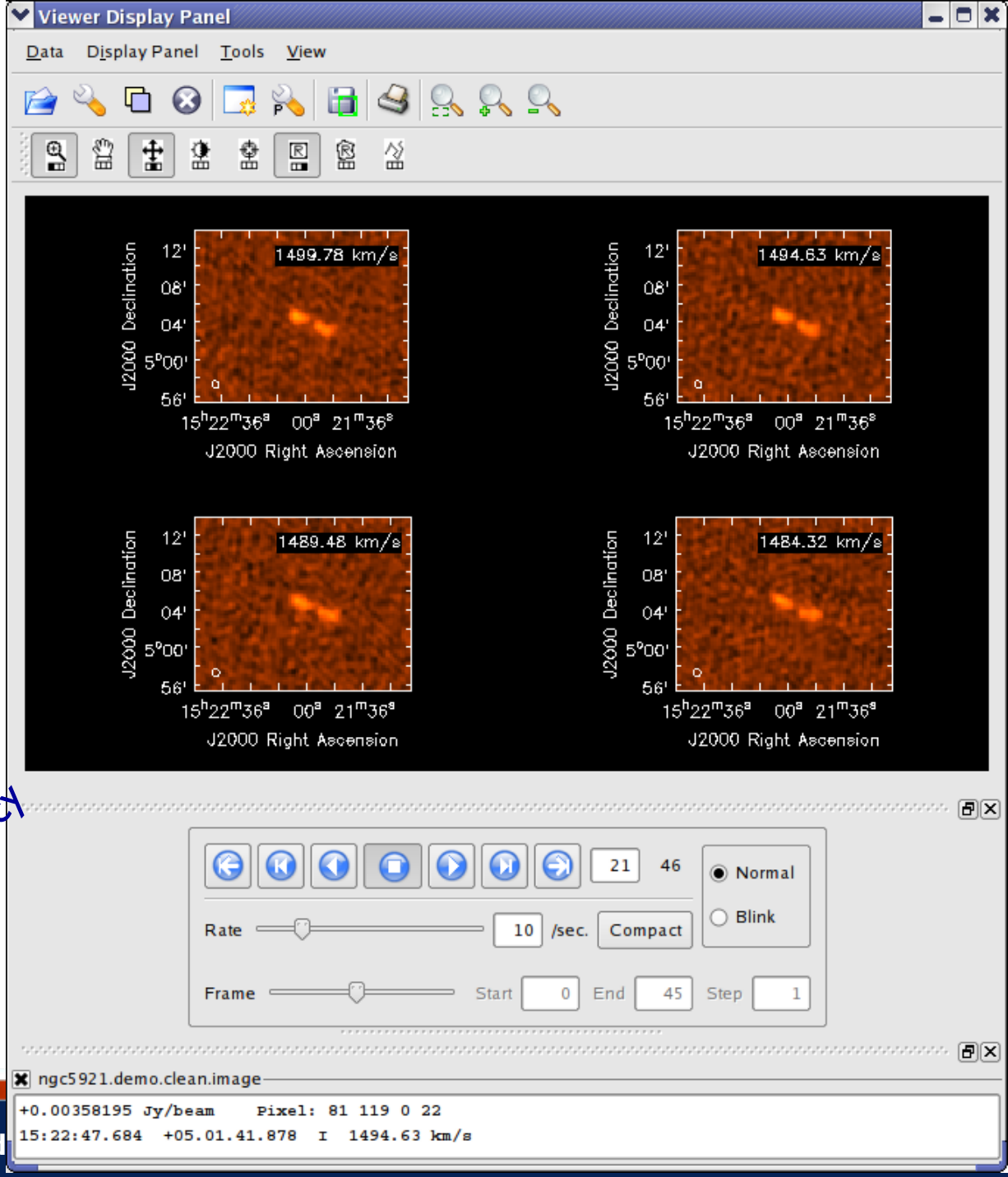


Viewer (imview)

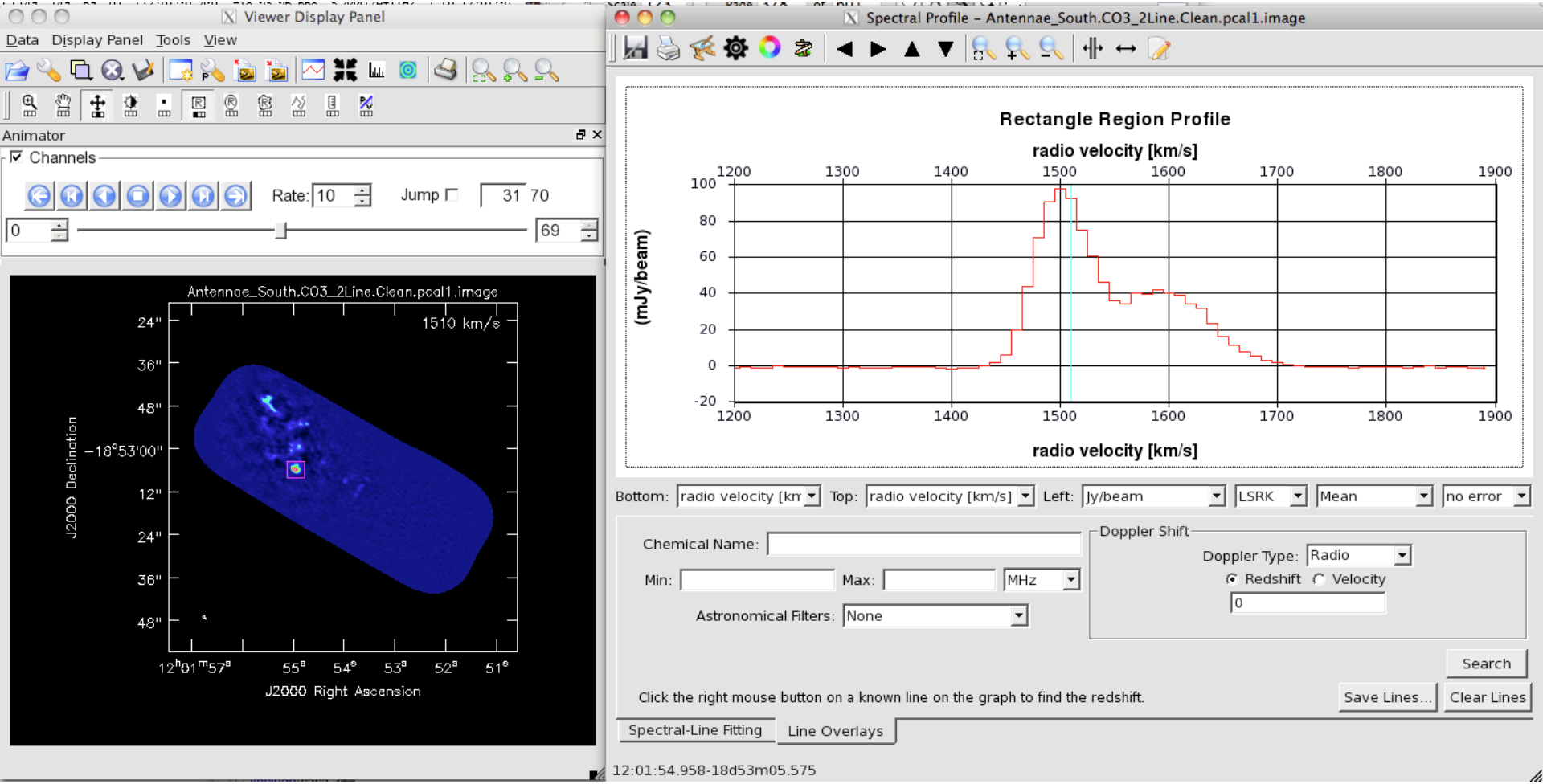


- Cubes
- Movies
- Channel maps

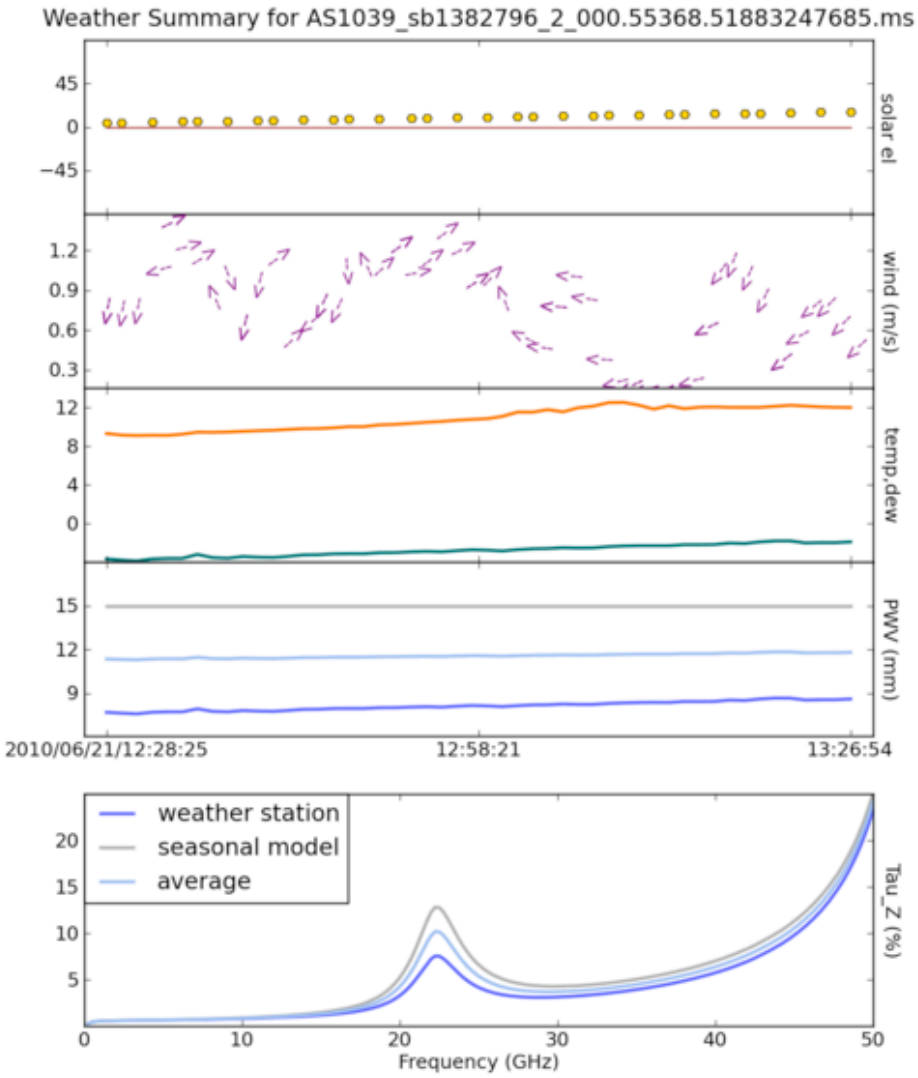




Viewer (imview)



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 **os.system('rm –rf 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
```

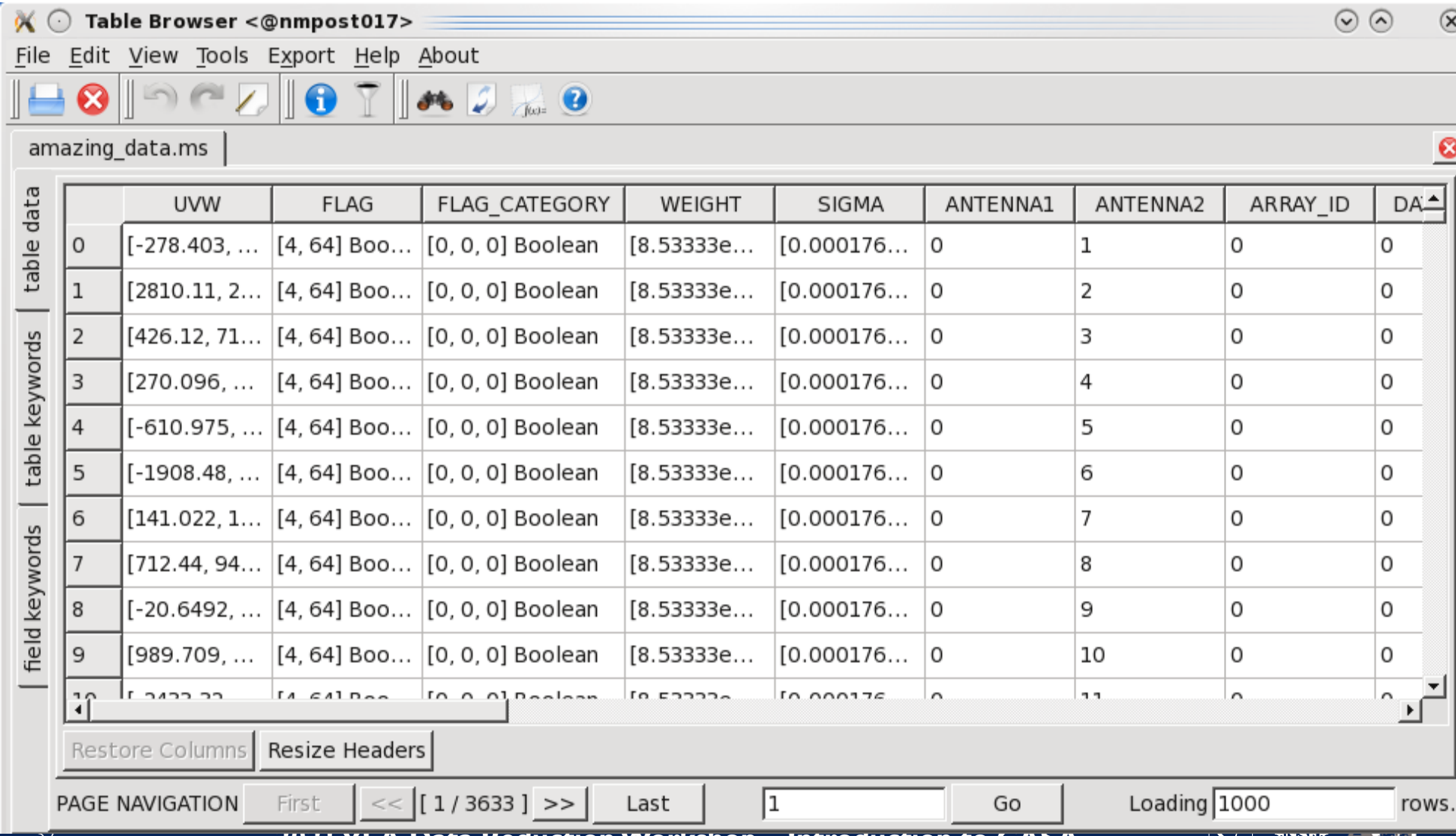
- Also contains sub-tables, e.g. FIELD, SOURCE, WEATHER, …

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

- More on the MS: https://casa.nrao.edu/casadocs/casa-6.1.0/casa-fundamentals/the-measurementset

MS MAIN table contents

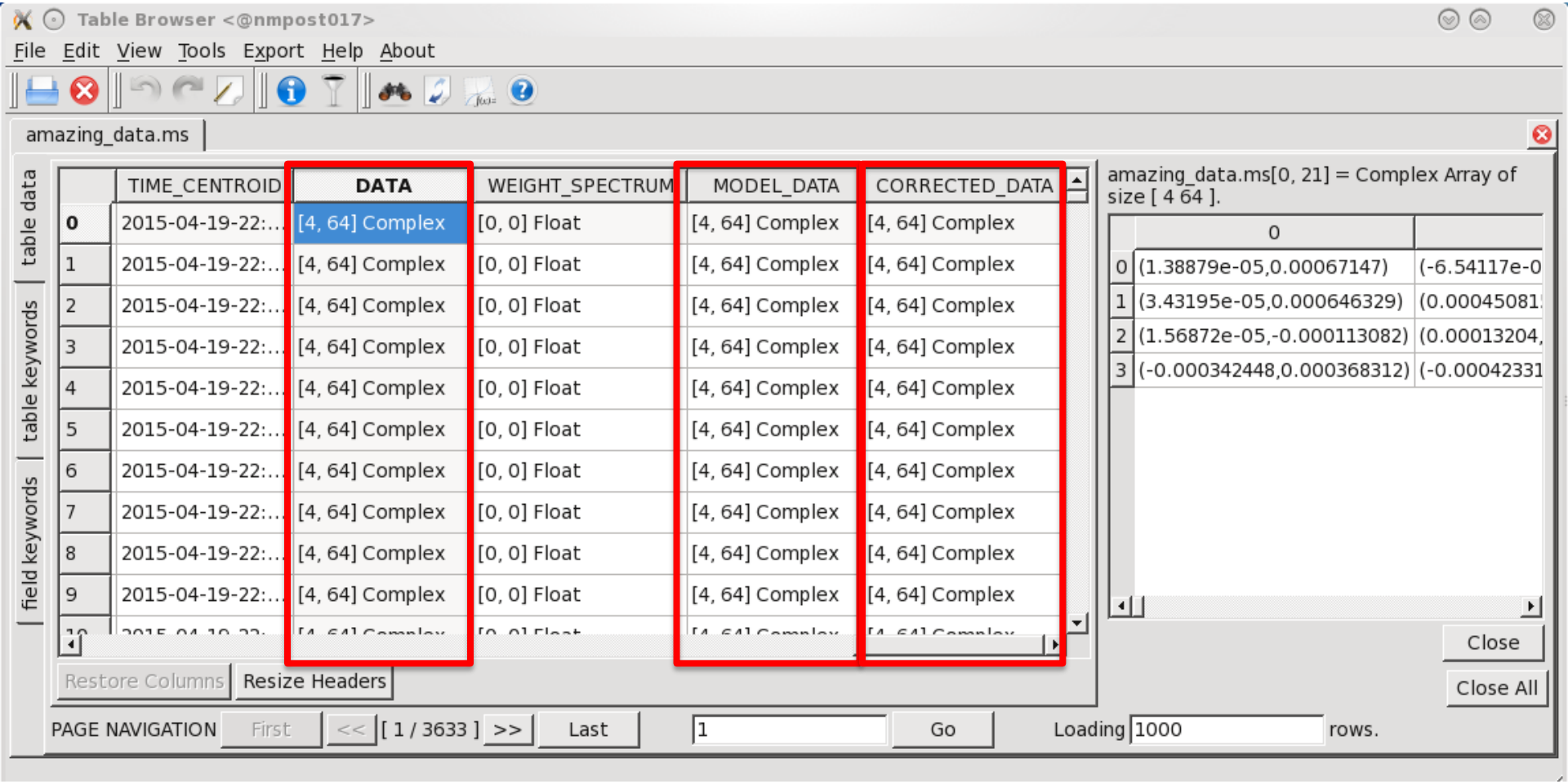
Inspect with task **browsetable**



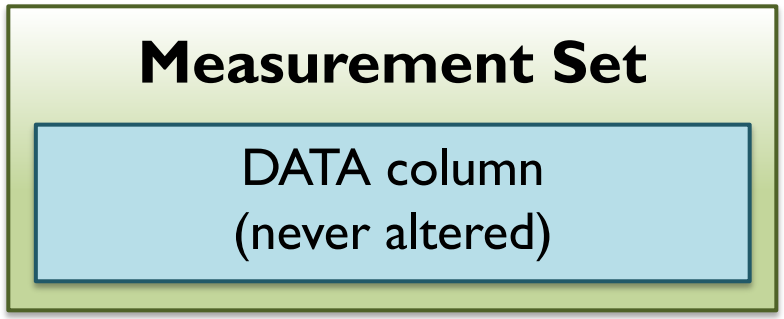
	UVW	FLAG	FLAG_CATEGORY	WEIGHT	SIGMA	ANTENNA1	ANTENNA2	ARRAY_ID	DA
0	[-278.403, ...	[4, 64] Boo...	[0, 0, 0] Boolean	[8.53333e...	[0.000176...	0	1	0	0
1	[2810.11, 2...	[4, 64] Boo...	[0, 0, 0] Boolean	[8.53333e...	[0.000176...	0	2	0	0
2	[426.12, 71...	[4, 64] Boo...	[0, 0, 0] Boolean	[8.53333e...	[0.000176...	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

MS MAIN table contents

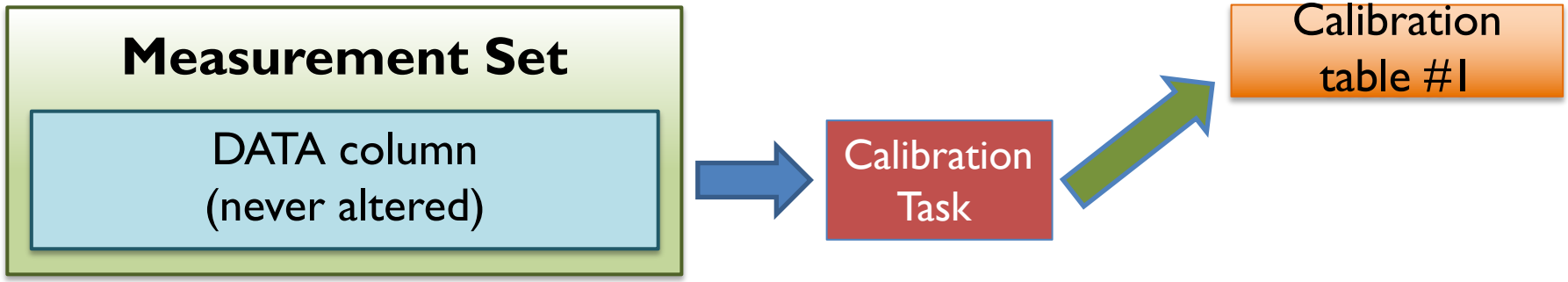
Inspect with task **browsetable**



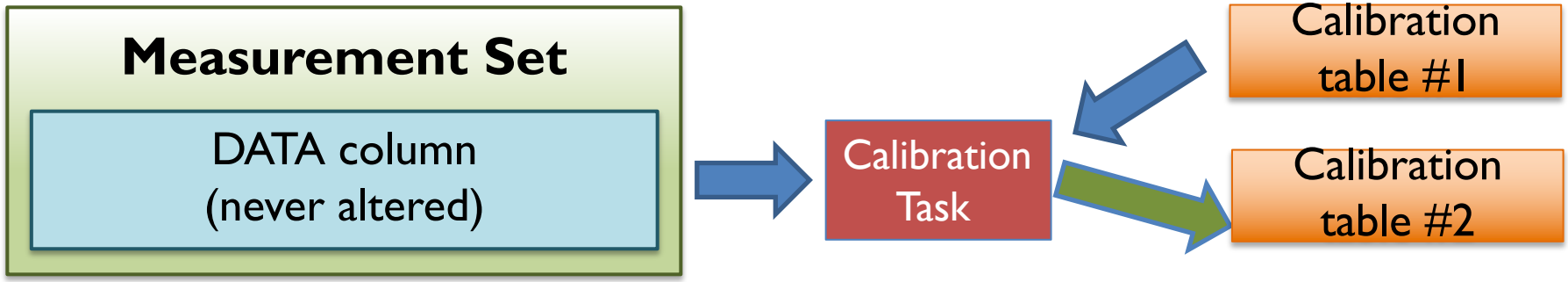
MS columns & calibration tables



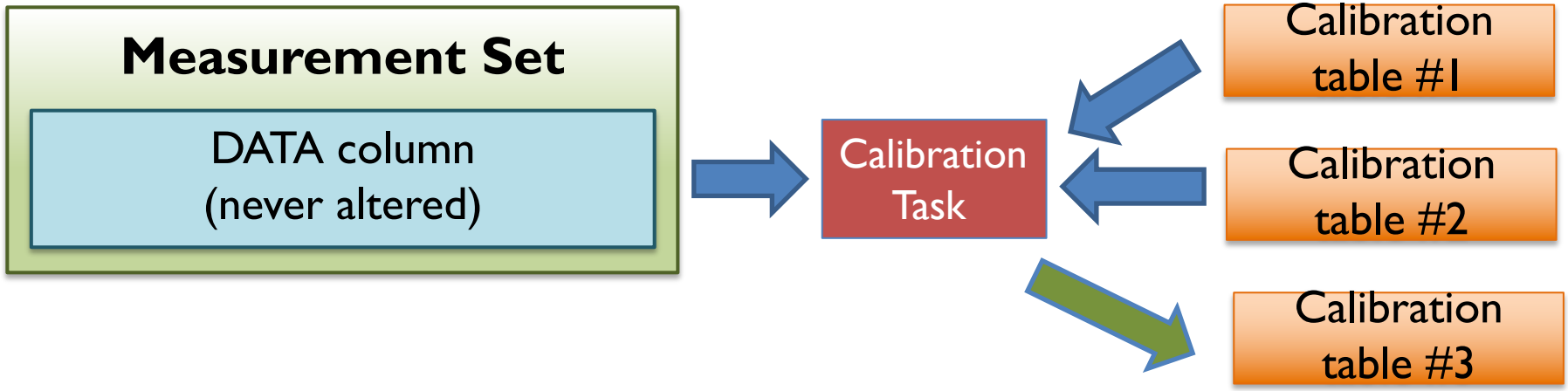
MS columns & calibration tables



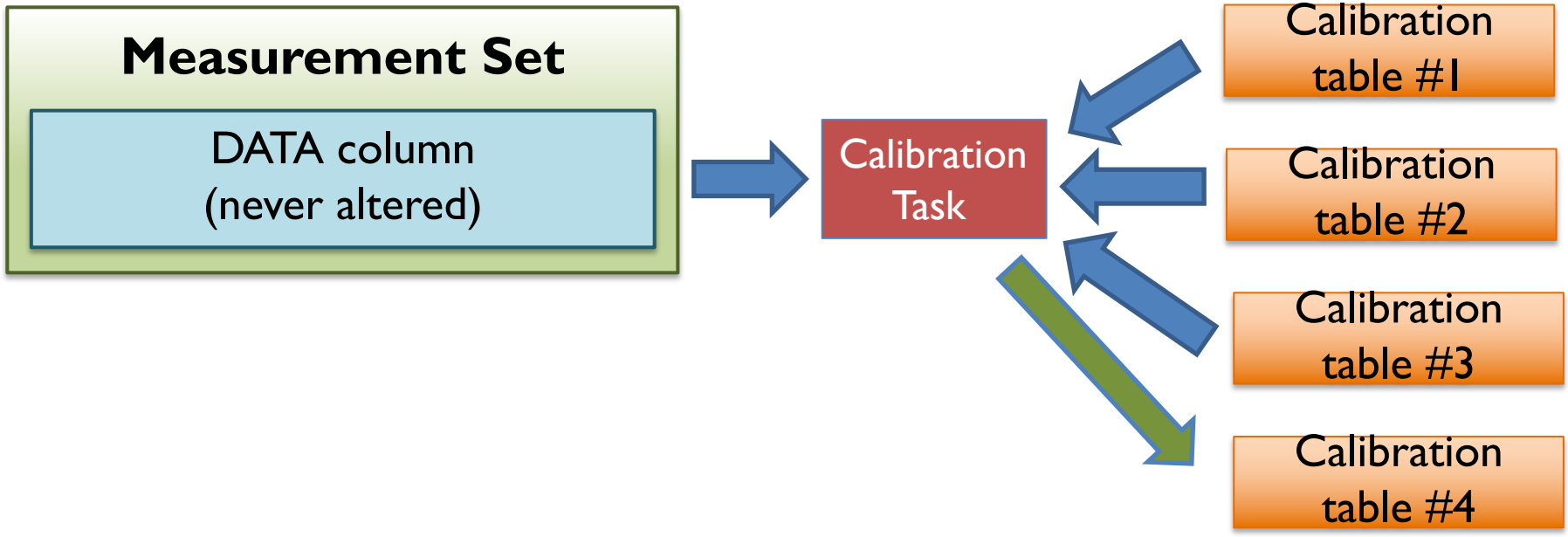
MS columns & calibration tables



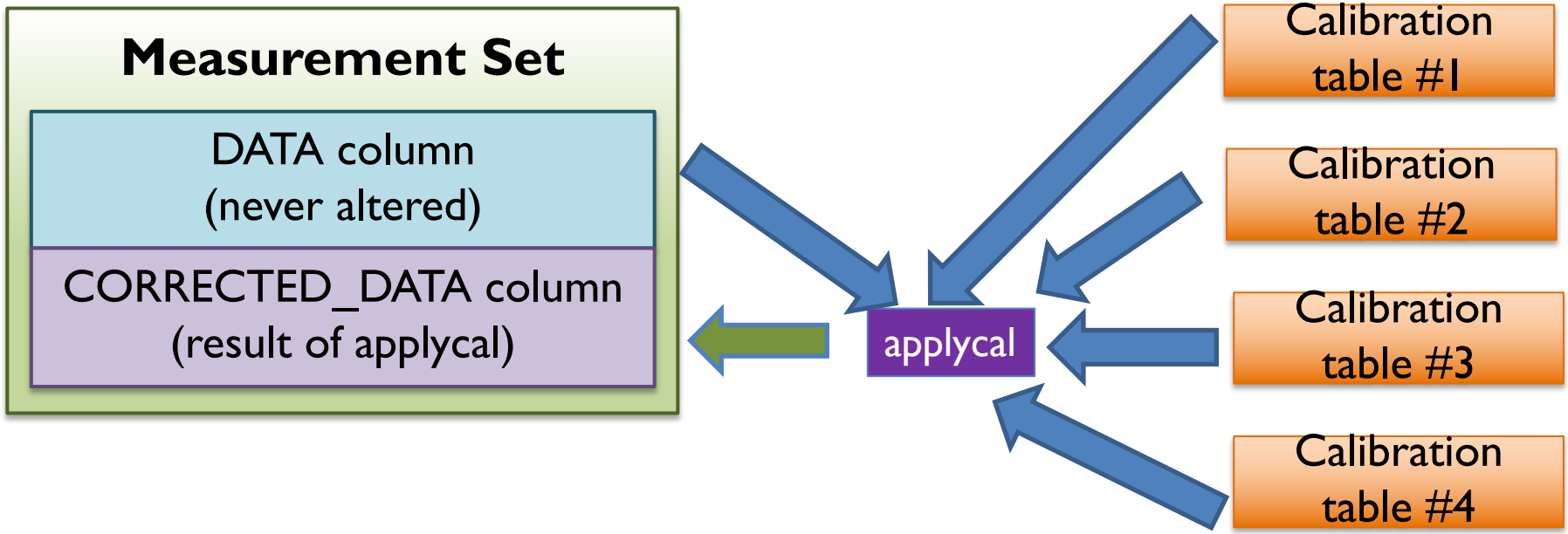
MS columns & calibration tables



MS columns & calibration tables

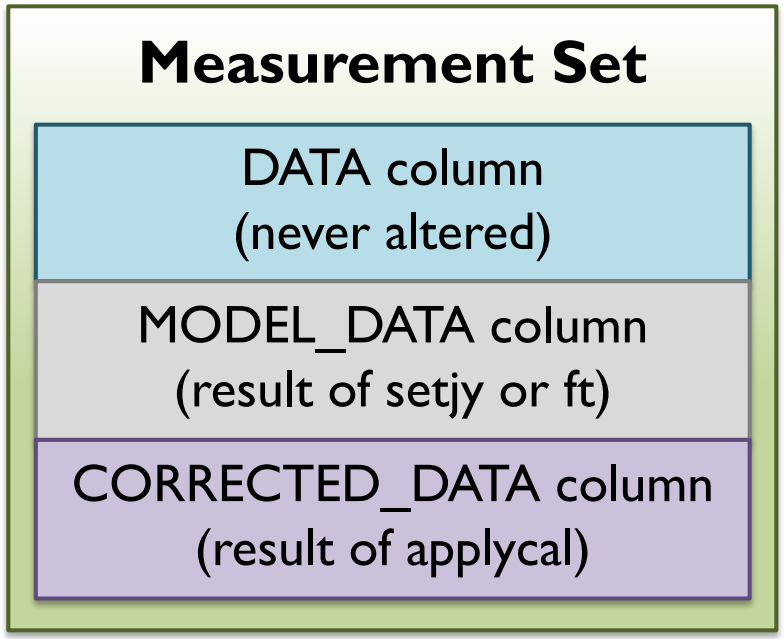


MS columns & calibration tables



If CORRECTED_DATA exists,
applycal will overwrite

MS columns & calibration tables

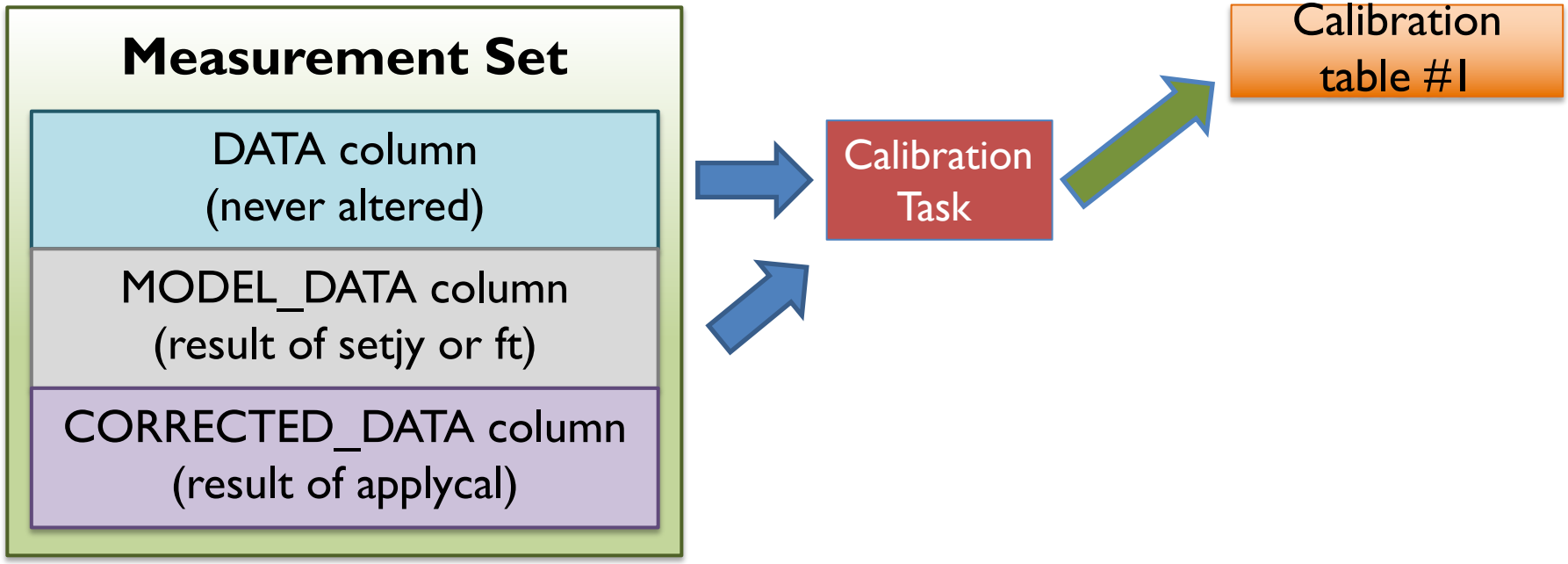


(note: data size tripled)

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)

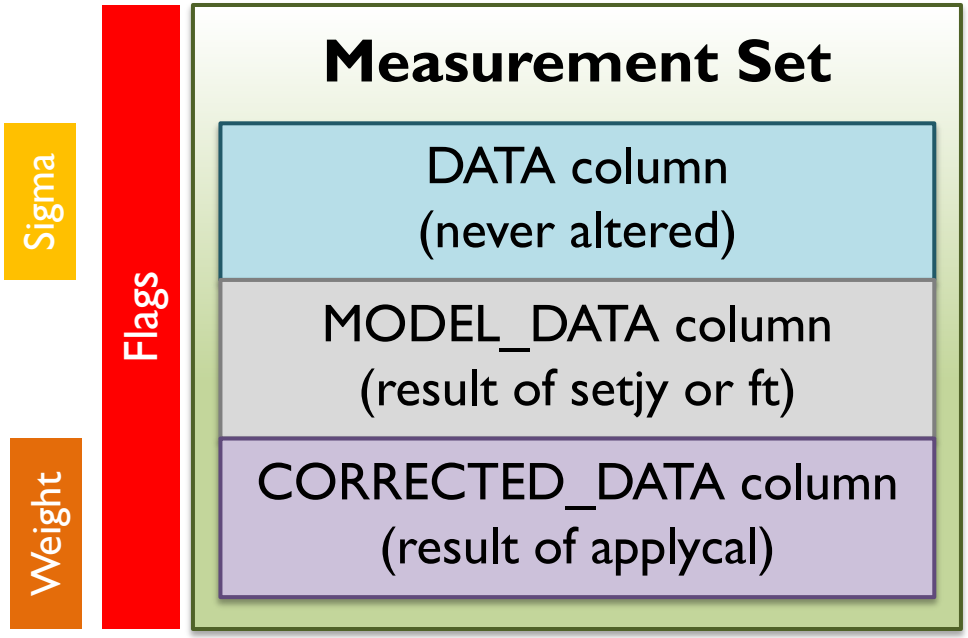
Model supplied by
setjy
ft
tclean

MS columns & calibration tables



(note: data size tripled)

MS columns & calibration tables



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://casa.nrao.edu/casadocs/casa-6.1.0/calibration-and-visibility-data/data-weights

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)

Listobs

Antennas: 27:

ID	Name	Station	Diam.	Long.	Lat.	Offset from array center (m) East	Offset from array center (m) North	Offset from array center (m) Elevation	ITRF Geocentric coordinates (m) x	ITRF Geocentric coordinates (m) y	ITRF Geocentric coordinates (m) z
0	ea01	W06	25.0 m	-107.37.15.6	+33.53.56.4	-275.8278	-166.7360	-2.0595	-1601447.195400	-5041992.497600	3554739.694800
1	ea02	W04	25.0 m	-107.37.10.8	+33.53.59.1	-152.8711	-83.7955	-2.4675	-1601315.900500	-5041985.306670	3554808.309400
2	ea03	W07	25.0 m	-107.37.18.4	+33.53.54.8	-349.9804	-216.7527	-1.7877	-1601526.383100	-5041996.851000	3554698.331400
3	ea04	N04	25.0 m	-107.37.06.5	+33.54.06.1	-42.6260	132.8521	-3.5428	-1601173.981600	-5041902.657800	3554987.528200
4	ea05	E05	25.0 m	-107.36.58.4	+33.53.58.8	164.9709	-92.7908	-2.5361	-1601014.465100	-5042086.235700	3554800.804900
5	ea06	N06	25.0 m	-107.37.06.9	+33.54.10.3	-54.0745	263.8800	-4.2325	-1601162.598500	-5041828.990800	3555095.895300
6	ea07	E04	25.0 m	-107.37.00.8	+33.53.59.7	102.8035	-63.7671	-2.6299	-1601068.794800	-5042051.918100	3554824.842700
7	ea08	E01	25.0 m	-107.37.05.7	+33.53.59.2	-23.8867	-81.1272	-2.5808	-1601192.486700	-5042022.840700	3554810.460900
8	ea09	N05	25.0 m	-107.37.06.7	+33.54.08.0	-47.8569	192.6072	-3.8789	-1601168.794400	-5041869.042300	3555036.937000
9	ea10	E08	25.0 m	-107.36.48.9	+33.53.55.1	407.8379	-206.0064	-3.2255	-1600801.917500	-5042219.370600	3554706.449200
10	ea11	N07	25.0 m	-107.37.07.2	+33.54.12.9	-61.1072	344.2424	-4.6414	-1601155.630600	-5041783.816000	3555162.366400
11	ea12	E07	25.0 m	-107.36.52.4	+33.53.56.5	318.0401	-164.1704	-2.6834	-1600888.422300	-5042170.386600	3554741.476400

MS data selection syntax

- 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, @ for pads e.g. **antenna='ea09&ea11@W51'**
  - Use less than (<) or greater than (>) for selection, e.g. **uvrange='<1000m'**

- For full syntax (and limitations) see CASAdocs
- https://casa.nrao.edu/casadocs/casa-6.1.0/calibration-and-visibility-data/data-selection-in-a-measurementset

MS data selection syntax: Examples

- **field** (spatial)
  - String with source name or field ID (checks former first)
  - Examples: **field='1331+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='1421MHz:10~20;50,5:1.6~1.7GHz'**

- **timerange** (temporal)
  - 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'**

MS data selection syntax: Examples

- **antenna**
  - String with antenna name or ID (checks former first)
    - Beware VLA name ea1-ea28, these have IDs 0-27 (but not necessarily consecutive, ids are numbered through the antenna table)
  - & = CC only , && = CC+AC , &&& = AC only
  - @ used for pad specification
  - Examples: **antenna = '1~5,8' ; antenna='!ea01&ea10' ; antenna='ea05&&&'; antenna='ea03@N43&&@E22'**

- **scan** – the scan numbers (an execution sequence)
  - e.g. **scan='3~14'**

MS data selection syntax: Examples

- **correlation** – polarization products
  - e.g. **correlation='RR,LL,LR'**

- **uvrange** – select on uv range
  - e.g. **uvrange='30~3000m' ; uvrange='<1000m'**

- **observation** – ID of the observation day when different observations are combined
  - e.g. **observation='0'**

- **intent** - intent of the scan
  - e.g. **intent='CALIBRATE_FLUX'**

Ongoing CASA Developments

- CARTA: new alternative image cube viewer, excellent especially for large images
  - developed by NRAO, IDIA, ASIAA, U Alberta
  - v1.4 release available on **http://cartavis.github.io/**)
- New Imaging Algorithms
  - including single dish + interferometric combination, **sdintimaging**
  - https://casa.nrao.edu/casadocs/casa-6.1.0/global-task-list/task_sdintimaging/about
- Parallelization (mpicasa) for calibration and imaging
- Pipeline improvements (ALMA and VLA)
- Fringe fitting/VLBI (developed by JIVE and NRAO)
- CASAdocs transition for tools
  - Starting with CASA 5.8/6.2: https://casadocs.readthedocs.io/en/latest/index.html
- CASA 6 python 3, and modularization of CASA (**https://casa-pip.nrao.edu**)
- Nextgen CASA infrastructure for massive high performance computing applications

Modular CASA

- Not for this week…

- pip wheels for *casatasks* and *casatools* are available as Python 3 modules via the public PyPI server **casa-pip.nrao.edu**
  - Supported for Linux (not yet OSX)
  - pre-requisites on host machine: Python 3.6 installation, libgfortran3

```
$ python3.6 -m venv casa6

$ source casa6/bin/activate

(casa6) $ pip install --index-url https://casa-pip.nrao.edu/repository/pypi-casa-release/simple casatools

(casa6) $ pip install --index-url https://casa-pip.nrao.edu/repository/pypi-casa-release/simple casatasks
```

- https://casadocs.readthedocs.io/en/latest/notebooks/usingcasa.html#Modular-Installation-of-CASA-6
- and stay tuned for Josh Marvil's talk on Wednesday on "Advanced CASA"

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
  - Logger (and terminal) output
  - 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
- CASA Newsletter
- Sign up to mailing lists to receive updates

www.nrao.edu
science.nrao.edu
public.nrao.edu

The National Radio Astronomy Observatory is a facility of the National Science Foundation operated under cooperative agreement by Associated Universities, Inc.