Introduction to CASA

Jürgen Ott

CASA
Common Astronomy Software Applications

NRAO NSF AUI

Overview of this talk

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

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, under development: 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 5.6.1-8 (incl. ALMA pipeline),
- But we use 5.4.2-5 for this workshop (latest release with a VLA pipeline)

CASA releases

- New releases about every 6 months (typically April and July)
- Pipelines usually released once a year (ALMA: October 1, VLA a bit later)
  - Also "prerelease" versions that are markers on path to next release with more functionality, but likely contain unfinished developments, less tested code, and less up-to-date documentation (casadocs-devel)

- Latest version 5.6 runs on:
  - Red Hat Linux 6 and 7
  - macOS 10.13 and 10.14
  - also works on other Linux systems, but less tested

CASA documentation and web resources

CASA Homepage http://casa.nrao.edu

Shortcuts to the most important sub-pages

CASA documentation and web resources

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

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

CASA 5.6

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.

The CASA 5.6 series includes two official CASA releases to date, CASA 5.6.0 and 5.6.1, which can be downloaded here. Both the 5.6.0 and 5.6.1 release are suited for manual data processing, while CASA 5.6.1 has been scientifically validated for the ALMA and includes the ALMA Cycle-7 pipeline. *[Editorial: CASA 5.6.1 will be officially released on 1 Oct 2019. CASA Docs updates reflecting CASA 5.6.1 are in preparation for the Oct 1st release.]*

The CASA 5.6 releases build on CASA 5.5, but have the following main new features:

New Features

- Task **bandpass** now supports relative frequency-dependent interpolation when applying bandpass tables.
- For **fringefit**, a new keyword *niter* determines a maximum number of iterations for the global least squares solver.
- In **tclean**, a new parameter *'smallscalebias'* for *deconvolver = 'mtmfs'* more efficiently cleans signal on different spatial scales.
- **plotms** can show atmospheric and sky temperature curves, enhanced pointing plots and more accurate axis labels.
- In **tclean**, the auto-multithresh now functions with polarization data.
- A new task **nrobeamaverage** is available Nobeyama data, to average on-source data over different beams.
- **tsdimaging** has new modes to track ephemeris objects.
- In **sdcal**, the *'otf'* calibration mode has improved.
- **simobserve** can create multi-channel MeasurementSets from a component list. ALMA receiver temperatures were updated.
- The **statwt** task log has been updated to provide explicitly the channels used in the statwt calculations.
- The ATM library of atmospheric models has been updated in CASA.

CASA 5.6

CASA, the *Common Astronomy Software Applications*, is the p
Millimeter/submillimeter Array (ALMA) and Karl G. Jansky V

The CASA 5.6 series includes two official CASA releases to
and 5.6.1 release are suited for manual data processing, wh
the ALMA Cycle-7 pipeline. *[Editorial: CASA 5.6.1 will be offici
preparation for the Oct 1st release.]*

The CASA 5.6 releases build on CASA 5.5, but have the follo

New Features

- Task **bandpass** now supports relative frequency-depe
- For **fringefit**, a new keyword *niter* determines a maxim
- In **tclean**, a new parameter *'smallscalebias'* for *deconv*
- **plotms** can show atmospheric and sky temperature c
- In **tclean**, the auto-multithresh now functions with pol
- A new task **nrobeamaverage** is available Nobeyama d
- **tsdimaging** has new modes to track ephemeris object
- In **sdcal**, the *'otf'* calibration mode has improved.
- **simobserve** can create multi-channel MeasurementSe

CASA documentation and web resources

CASAguides https://casaguides.nrao.edu Tutorials

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

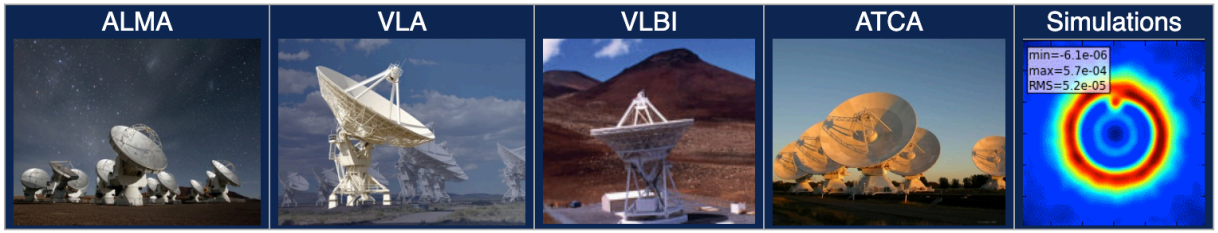
Search

Go | Search

tools

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

CASA Tutorials



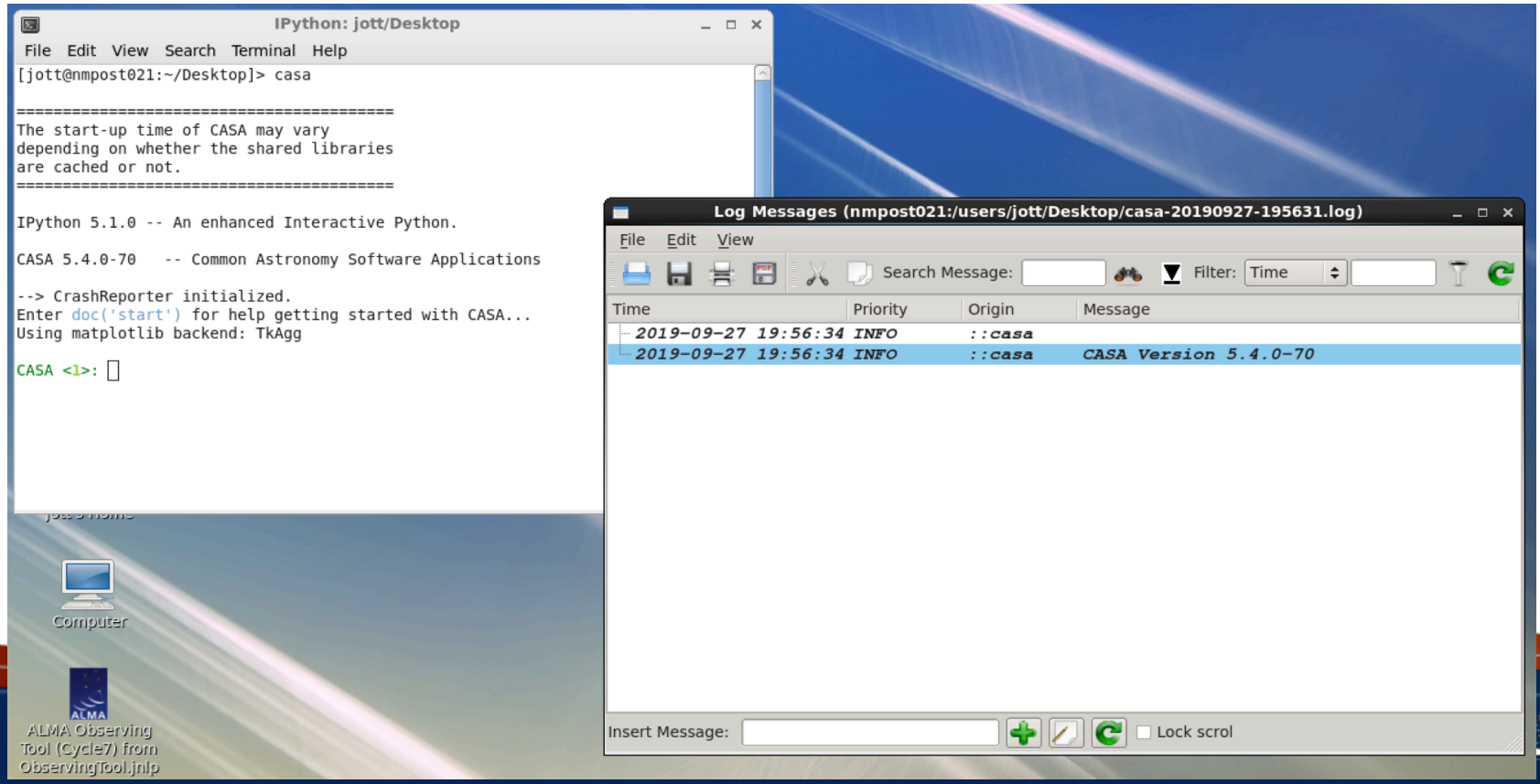
Extracting Scripts from Tutorials

Information for authors: MediaWiki markup language CASAGuides Instructions for Authors

537 articles since July 2009.

Starting CASA

- Start CASA from the UNIX shell: **casa**
- (with pipeline: casa --pipeline; 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



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

Find the right task

To see an organized list, type:

tasklist

IPython: jott/Desktop

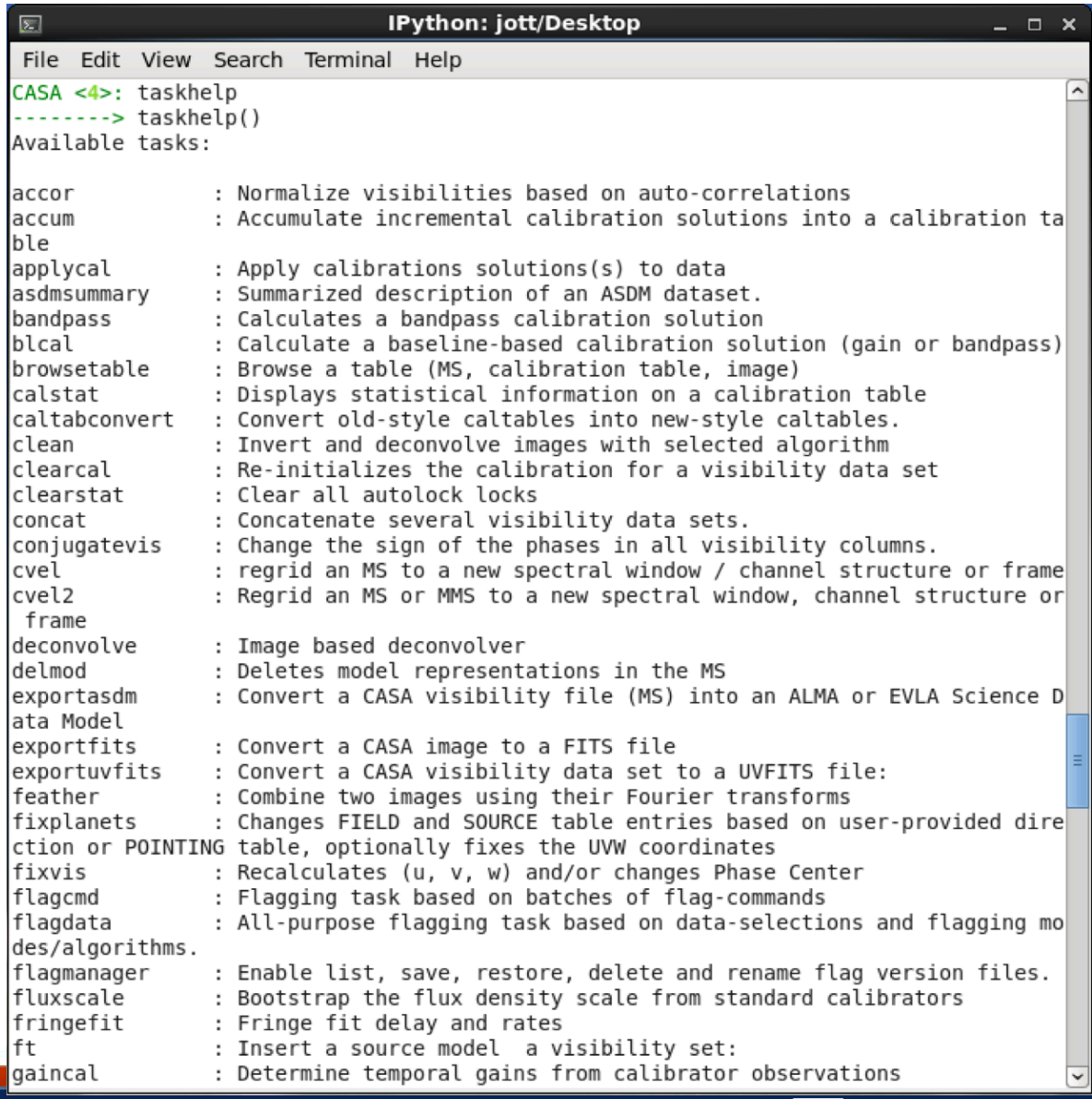
File Edit View Search Terminal Help

```
------------------

CASA <2>: tasklist
--------> tasklist()
Available tasks, organized by category (experimental tasks in parenthesis ()
  deprecated tasks in curly brackets {}).


Import/export       Information         Editing             Manipulation
------------------  ------------------  ------------------  ------------------
exportasdm          asdmsummary         fixplanets          concat
exportfits          imhead              fixvis              conjugatevis
exportuvfits        imreframe           flagcmd             cvel
importasap          imstat              flagdata            fixvis
importasdm          imval               flagmanager         hanningsmooth
importatca          listcal             msview              imhead
importfits          listfits            plotms              mstransform
importfitsidi       listhistory                             oldstatwt
importmiriad        listobs                                 partition
importnro           listpartition                           plotms
importuvfits        listsdm                                 split
importvla           listvis                                 testconcat
(importgmrt)        makemask                                uvcontsub
                    plotms                                  virtualconcat
                    vishead                                 vishead
                    visstat                                 (cvel2)
                                                            (uvcontsub3)

Calibration         Modeling            Imaging             Analysis
------------------  ------------------  ------------------  ------------------
accum               predictcomp         clean               imcollapse
applycal            setjy               deconvolve          imcontsub
bandpass            uvcontsub           feather             imdev
blcal               uvmodelfit          ft                  imfit
calstat             uvsub               imcontsub           imhead
clearcal            (uvcontsub3)        tclean              imhistory
delmod                                  (widebandpbcor)     immath
fixplanets                                                  immoments
fluxscale                                                   impbcor
ft                                                          impv
```

Find the right task

To see short summaries, type:

taskhelp



Task help

Type:

tclean?

Or

help tclean

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

IPython: jott/Desktop

File Edit View Search Terminal Help

```
Radio Interferometric Image Reconstruction

        Detailed Description:
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 algo
rithms
                        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).


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

                    Default Value:

                 field:   Select fields to image or mosaic.  Use field id(s) or name(
s).
                   ['go listobs' to obtain the list id's or names]
                default: ''= all fields
                  If field string is a non-negative integer, it is assumed to
                  be a field index otherwise, it is assumed to be a
                  field name
                  field='0~2'; field ids 0,1,2
                  field='0,4,5~7'; field ids 0,4,5,6,7
                  field='3C286,3C295'; field named 3C286 and 3C295
                  field = '3,4C*'; field id 3, all names starting with 4C
                  For multiple MS input, a list of field strings can be used:
                  field = ['0~2','0~4']; field ids 0-2 for the first MS and 0-4
```

Task help

- doc(‘’) brings up a browser and launches CASAdocs.
- Browse to ‘task list’ and 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

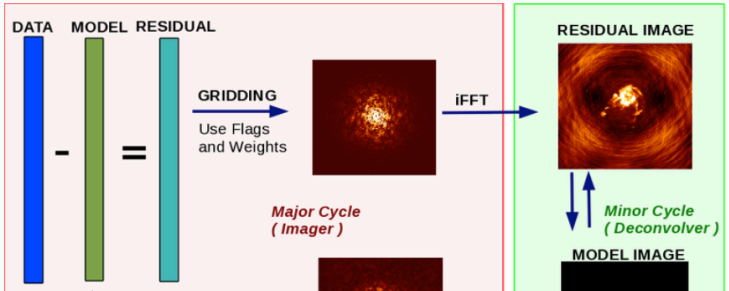
task tclean description

Overview

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

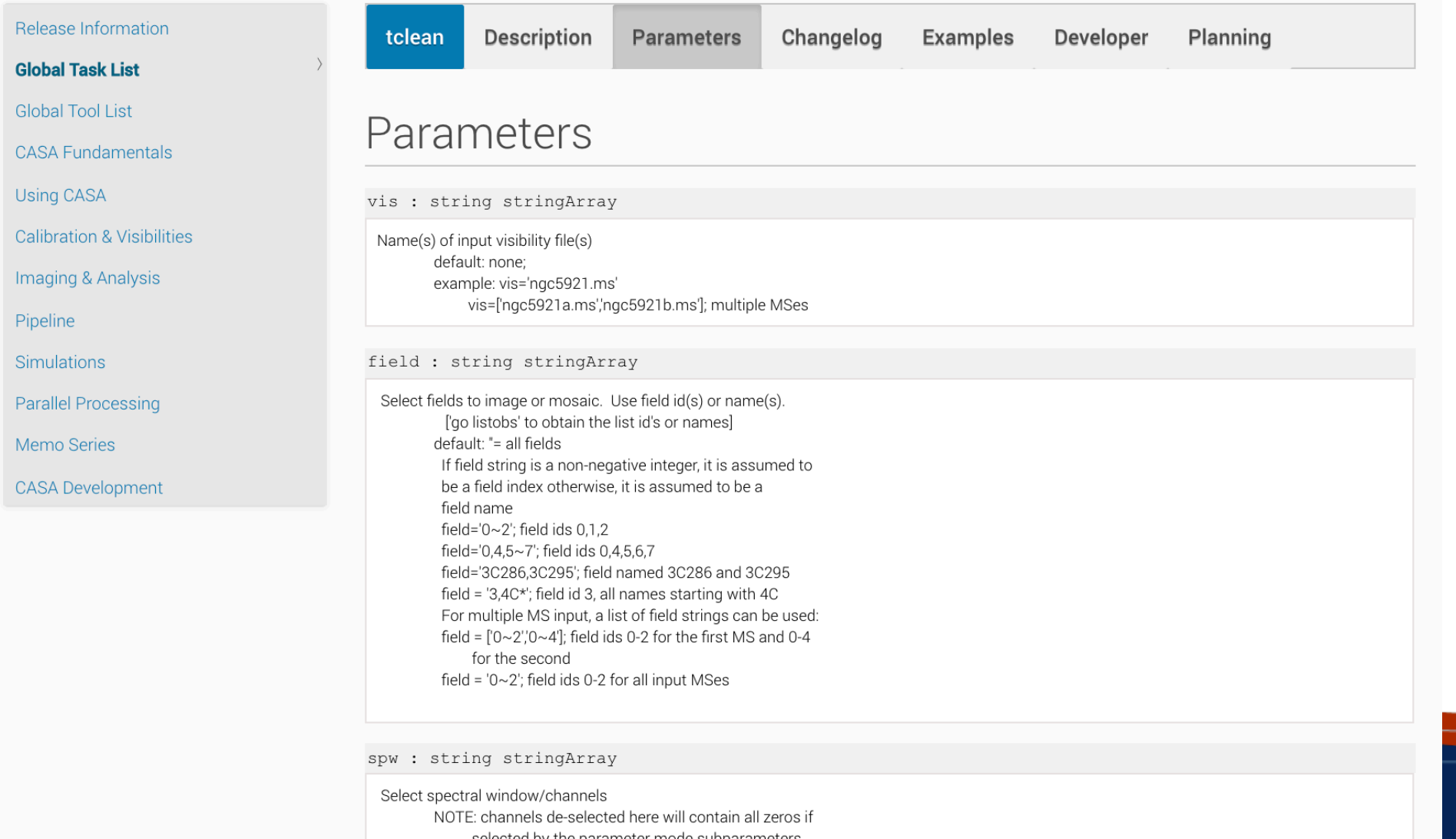
tclean handles continuum images and spectral line cubes, supports outlier fields, contains point-source CLEAN based algorithms as well as options for multi-scale and wideband image reconstruction, widefield imaging correcting for the w-term, full primary-beam imaging and joint mosaic imaging (with heterogeneous 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.



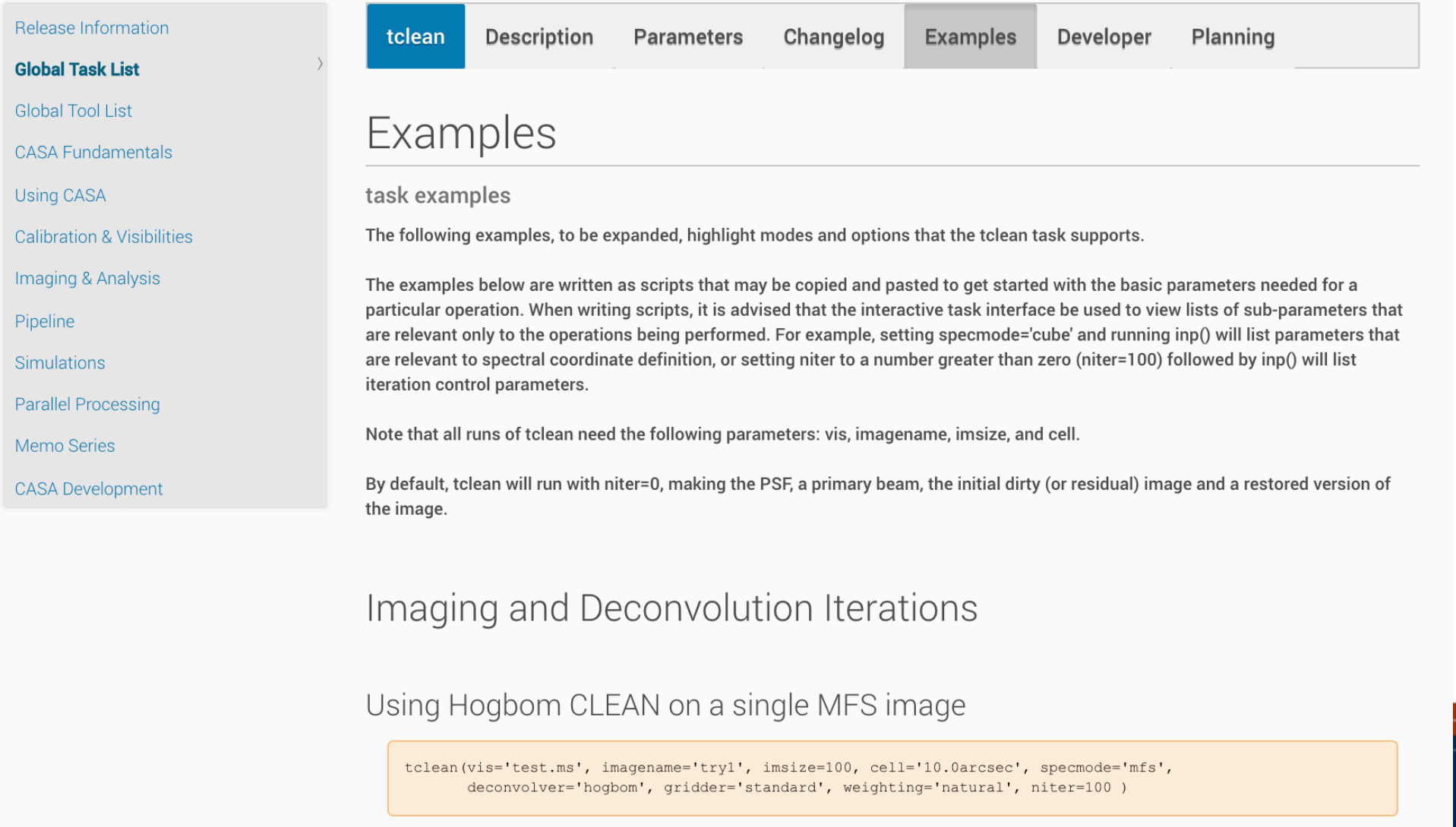
Task help

- doc('taskname') brings up a browser and launches CASAdocs for the task, the Parameters tab (identical to inline help)



Task help

- Examples in CASAdocs



How to run a task

- Task interface
  - Use inp taskname to see list of parameters
  - Set (global) parameters one at a time
  - Useful for interactive work, exploring parameters
  - Recover previous parameters using tget taskname
    antonym tput saves without running the task; go and tput saves parameters in **<task>.last** files (as parameters and as functional call)
  - 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

```
inp listobs
listfile = 'outfile2.txt'
tput
listfile='outfile.txt'
tget listobs
go
```

Will write to outfile2.txt

Task interface

Inspect task inputs:

inp tclean

Black: default value

Red: invalid value

Blue: non-default value

Reset defaults:

default tclean

Grey: expandable

IPython: jott/Desktop

File Edit View Search Terminal Help

```
CASA <11>: inp
---------> inp()
#  tclean :: Radio Interferometric Image Reconstruction
vis                 =  'seti.ms'        #  Name of input visibility file(s)
selectdata          =      False        #  Enable data selection parameters
datacolumn          = 'corrected'       #  Data column to image(data,corrected)
imagename           = 'littleGreenMen'  #  Pre-name of output images
imsize              =      [100]        #  Number of pixels
cell                = ['1arcsec']       #  Cell size
phasecenter         =         ''        #  Phase center of the image
stokes              =        'I'        #  Stokes Planes to make
projection          =      'SIN'        #  Coordinate projection (SIN, HPX)
startmodel          =         ''        #  Name of starting model image
specmode            =      'mfs'        #  Spectral definition mode
                                        #   (mfs,cube,cubedata, cubesource)
     reffreq        =         ''        #  Reference frequency

gridder             = 'standard'        #  Gridding options (standard, wproject,
                                        #   widefield, mosaic, awproject)
     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,mul
                                        #   scale,mtmfs,mem,clarkstokes)
restoration         =       True        #  Do restoration steps (or not)
     restoringbeam  =         []        #  Restoring beam shape to use. Default is
                                        #   the PSF main lobe
     pbcor          =      False        #  Apply PB correction on the output
                                        #   restored image

outlierfile         =         ''        #  Name of outlier-field image definitions
weighting           =  'natural'        #  Weighting scheme (natural,uniform,brigg
     uvtaper        =         []        #  uv-taper on outer baselines in uv-plane
```

Task interface

Grey: expandable

Green: sub-parameter

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

IPython: jott/Desktop

File Edit View Search Terminal Help

```
CASA <9>: inp
--------> inp()
#  tclean :: Radio Interferometric Image Reconstruction
vis                 =  'seti.ms'        #  Name of input visibility file(s)
selectdata          =       True        #  Enable data selection parameters
     field          =         ''        #  field(s) to select
     spw            =         ''        #  spw(s)/channels to select
     timerange      =         ''        #  Range of time to select from data
     uvrange        =         ''        #  Select data within uvrange
     antenna        =         ''        #  Select data based on antenna/baseline
     scan           =         ''        #  Scan number range
     observation    =         ''        #  Observation ID range
     intent         =         ''        #  Scan Intent(s)

datacolumn          = 'corrected'       #  Data column to image(data,corrected)
imagename           = 'littleGreenMen'  #  Pre-name of output images
imsize              =     [100]         #  Number of pixels
cell                = ['1arcsec']       #  Cell size
phasecenter         =         ''        #  Phase center of the image
stokes              =        'I'        #  Stokes Planes to make
projection          =      'SIN'        #  Coordinate projection (SIN, HPX)
startmodel          =         ''        #  Name of starting model image
specmode            =      'mfs'        #  Spectral definition mode
                                        #   (mfs,cube,cubedata, cubesource)
     reffreq        =         ''        #  Reference frequency

gridder             = 'standard'        #  Gridding options (standard, wproject,
                                        #   widefield, mosaic, awproject)
     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,mul
                                        #   scale,mtmfs,mem,clarkstokes)
restoration         =       True        #  Do restoration steps (or not)
```

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')
  - 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);

.....
```

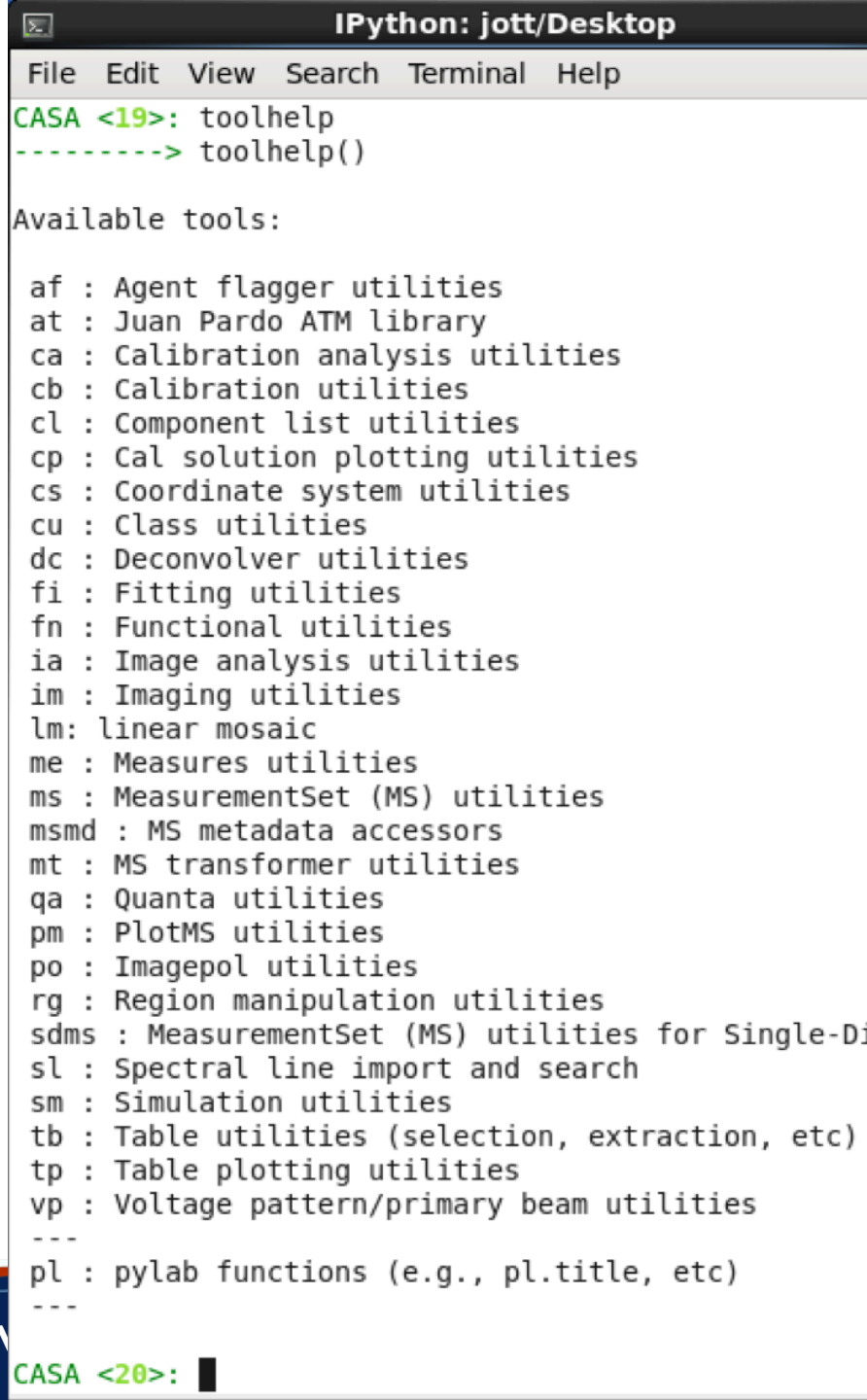
Tools

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



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)

See also Josh Marvil's talk

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.

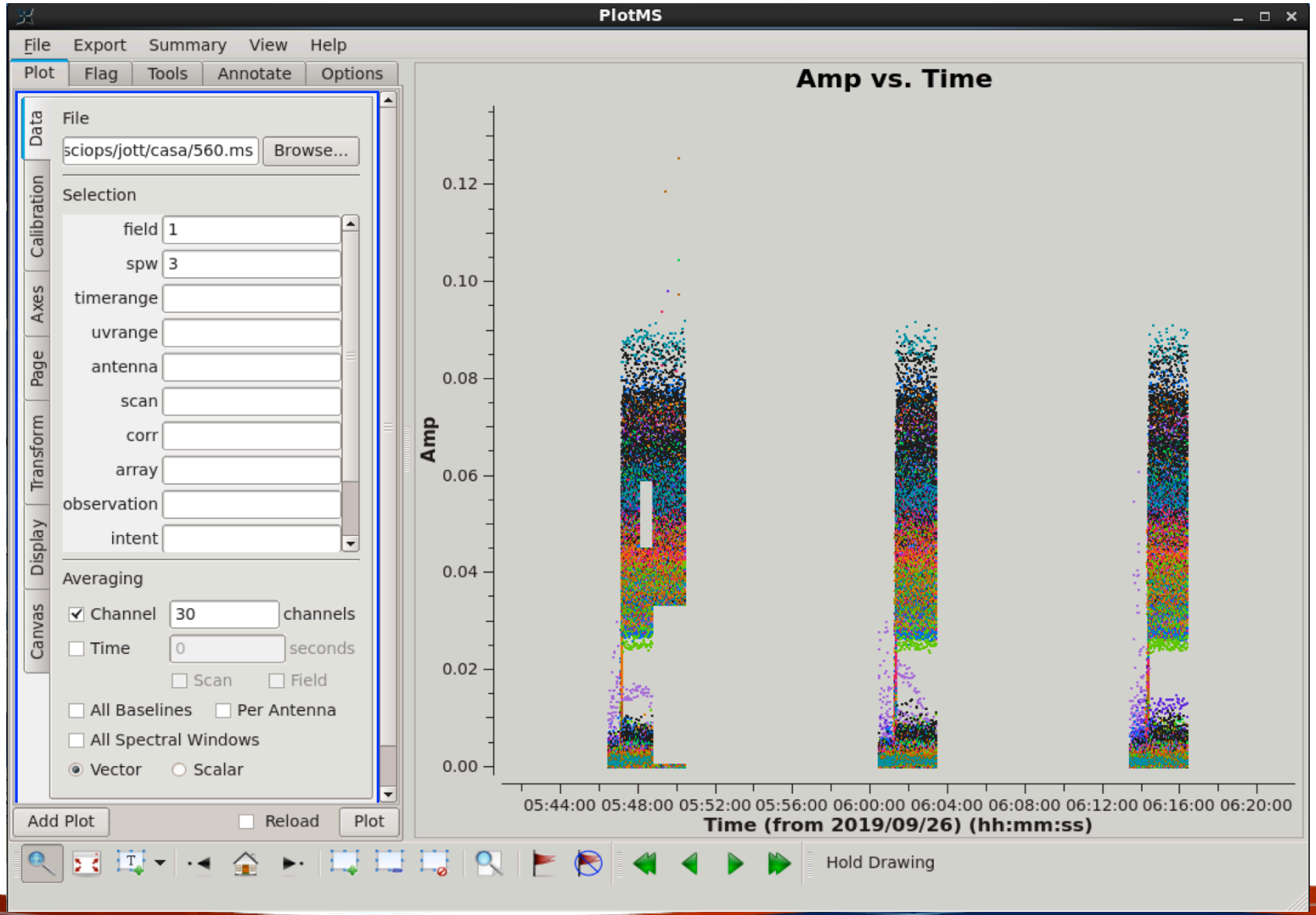
- If you still can't find what you need, write your own task!
  - Combination of Python plus CASA toolkit is very powerful

See also Josh Marvil's talk

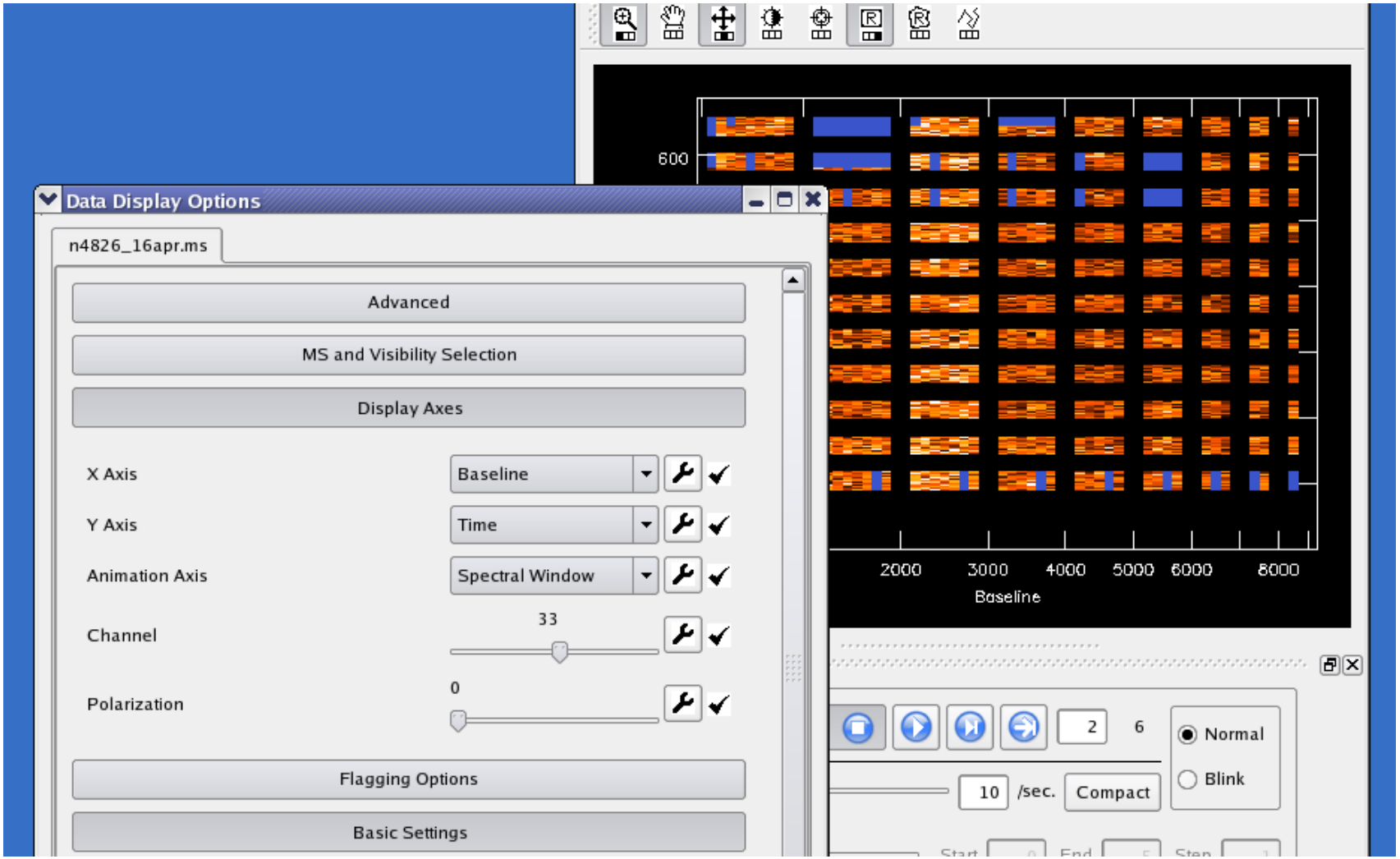
Applications

- Used to display and edit data (visibilities, calibration tables, images)
- Can be invoked inside CASA or as standalone programs from Linux shell
- Visibilities and calibration tables: plotms (from shell: casaplotms), msview, viewer, casafeather
- Any CASA (table) data: browsetable (from shell: casabrowser)
- Images: imview, viewer (from shell: casaviewer)
- Don't forget about full functionality of python! e.g. matplotlib, astropy, …

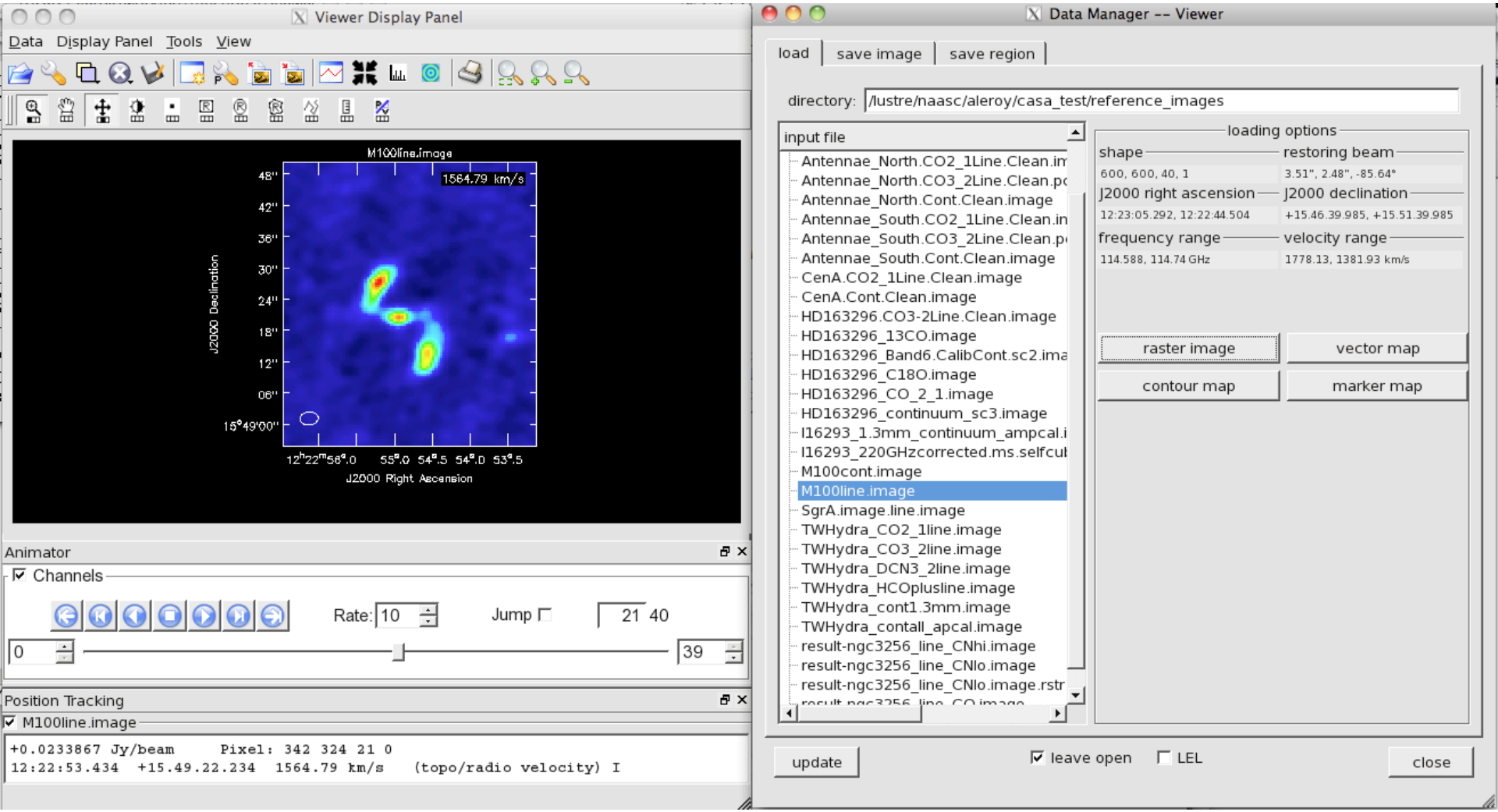
PlotMS


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

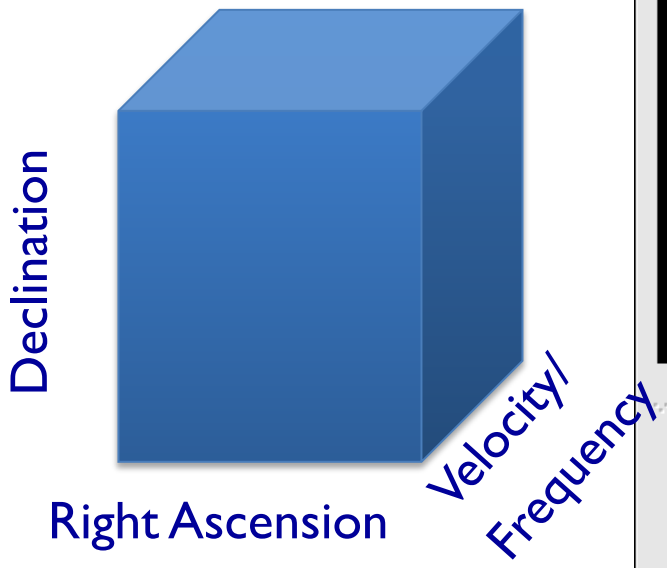
Viewer (msview)

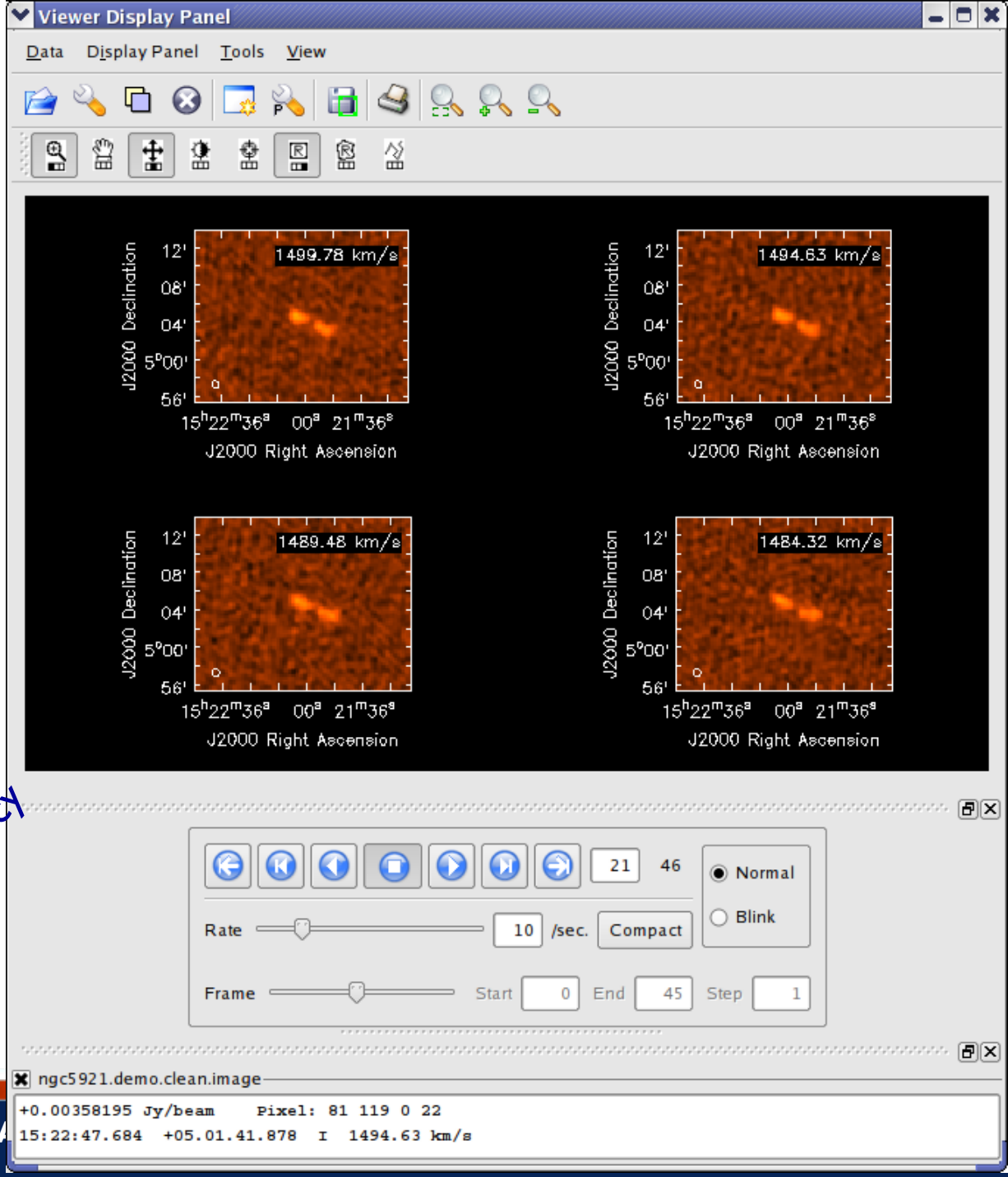


Viewer (imview)

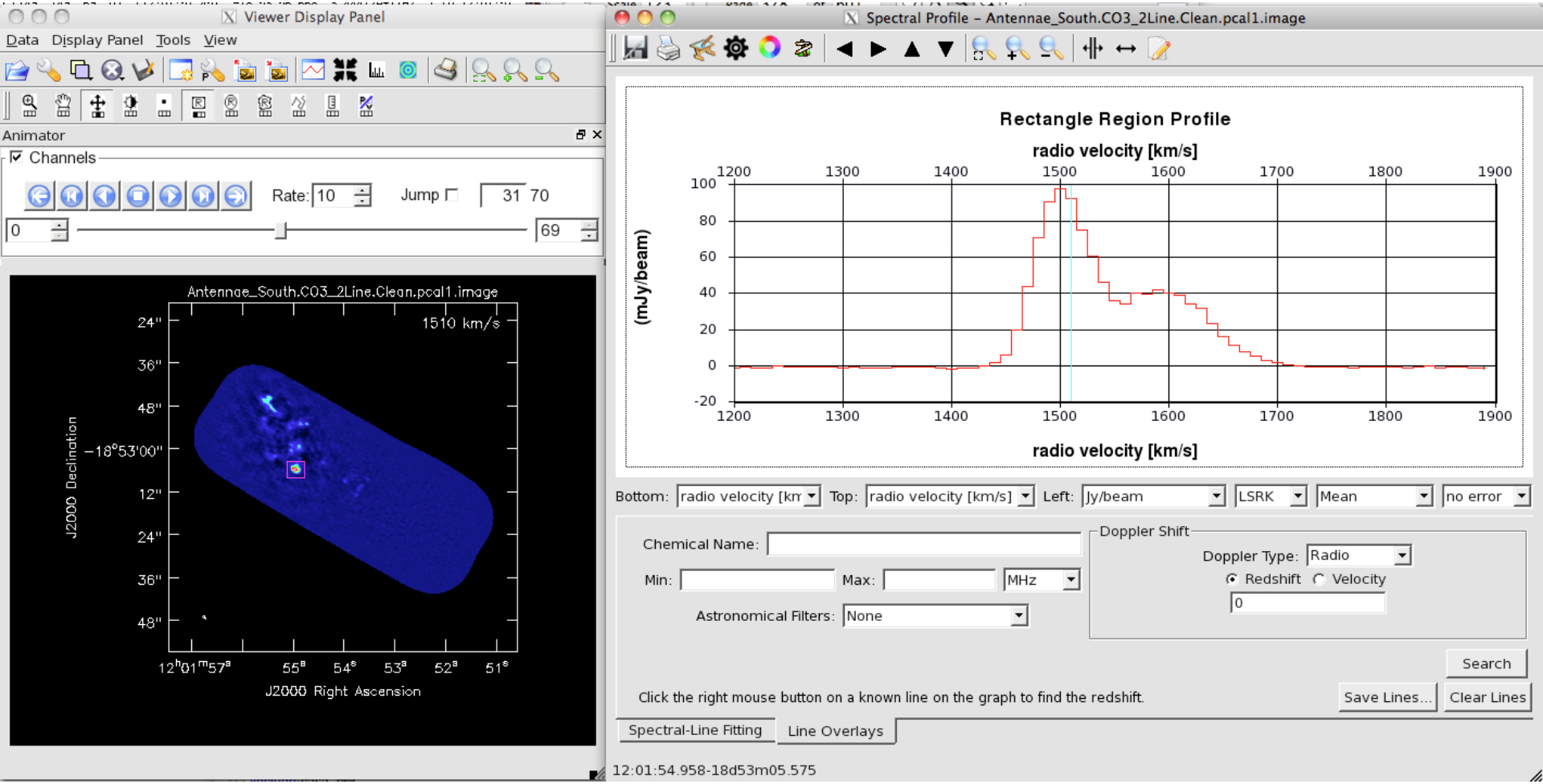


- Cubes
- Movies
- Channel maps

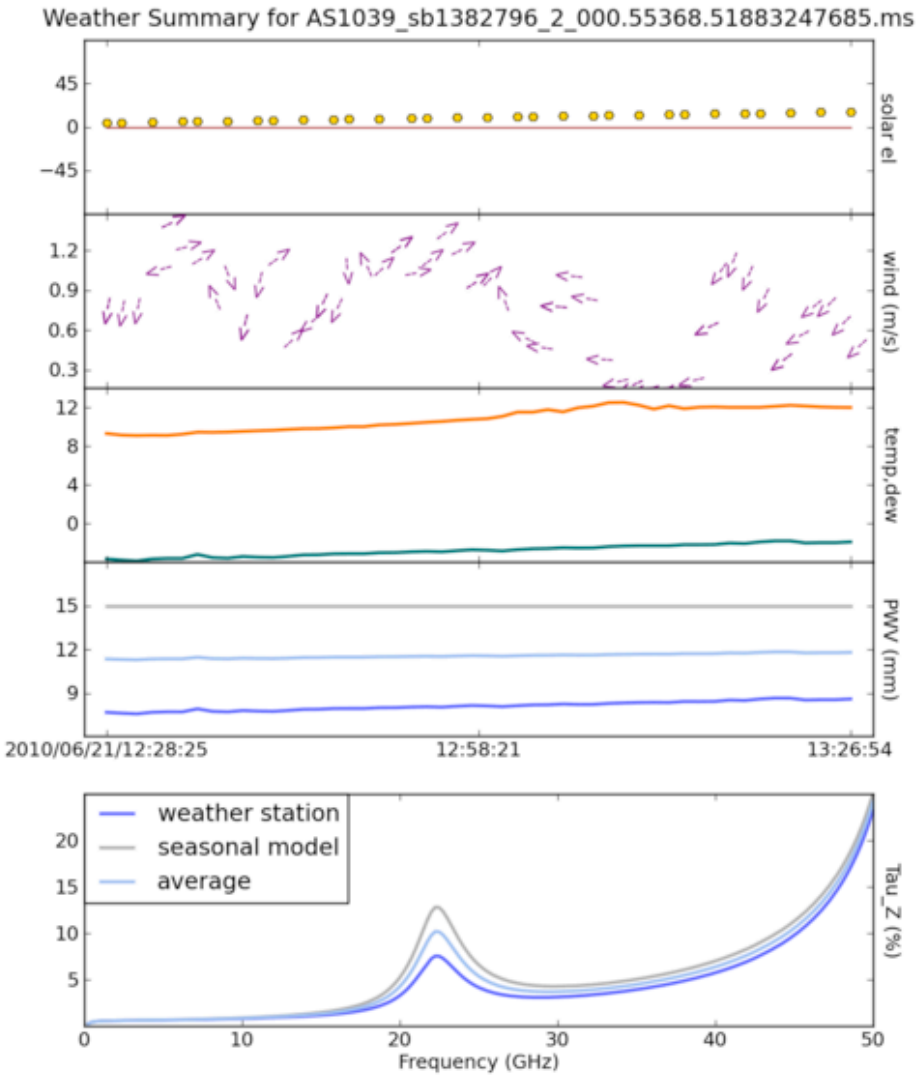




Viewer (imview)



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**

- 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

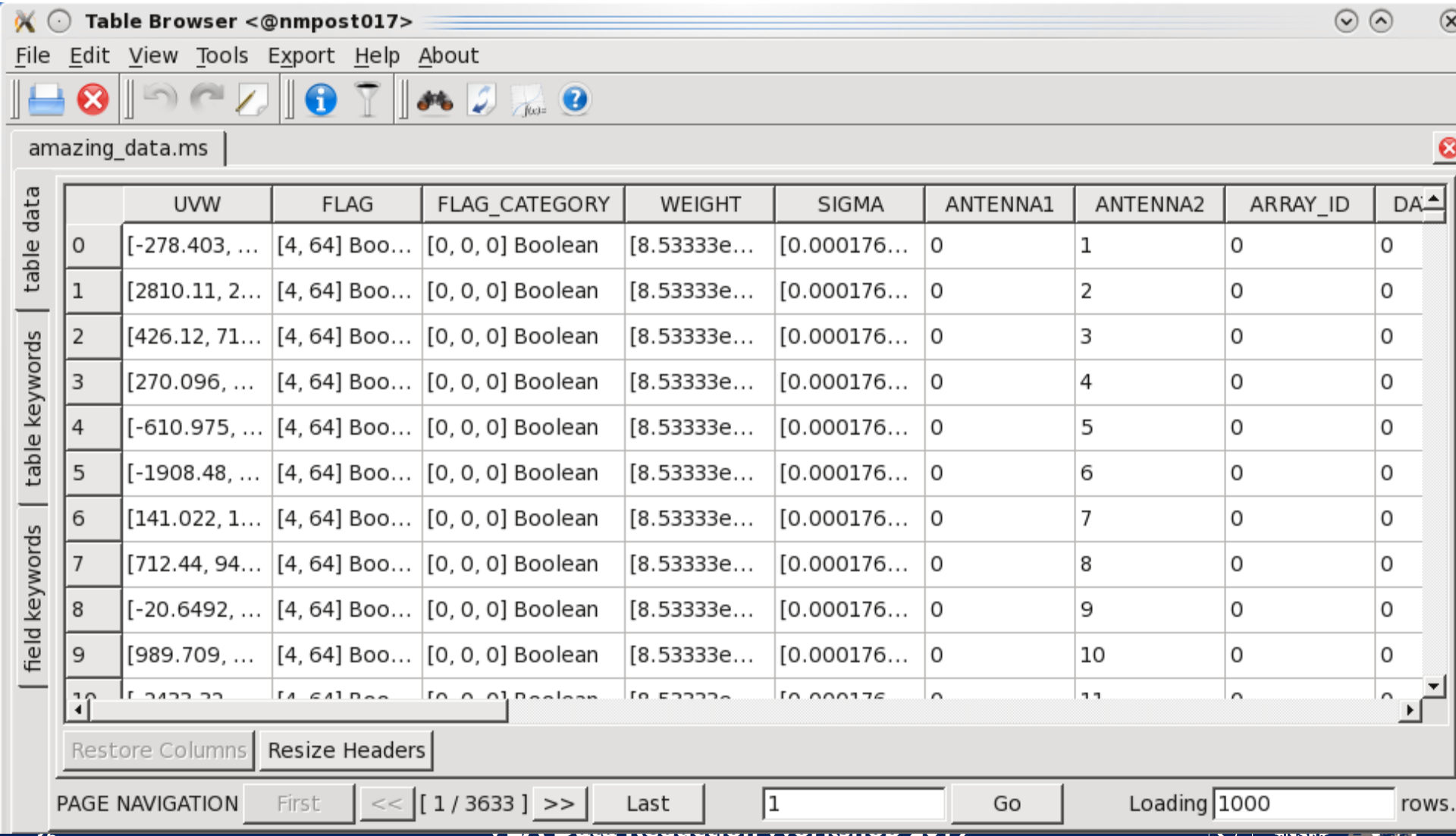
```
CASA <80>: !ls amazing_data.ms
ANTENNA           POINTING         SYSPOWER   table.f15       table.f20_TSM0  table.f24_TSM1  table.f8
CALDEVICE         POLARIZATION     table.dat  table.f16       table.f21       table.f25       table.f9
DATA_DESCRIPTION  PROCESSOR        table.f1   table.f17       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
```

MS MAIN table contents

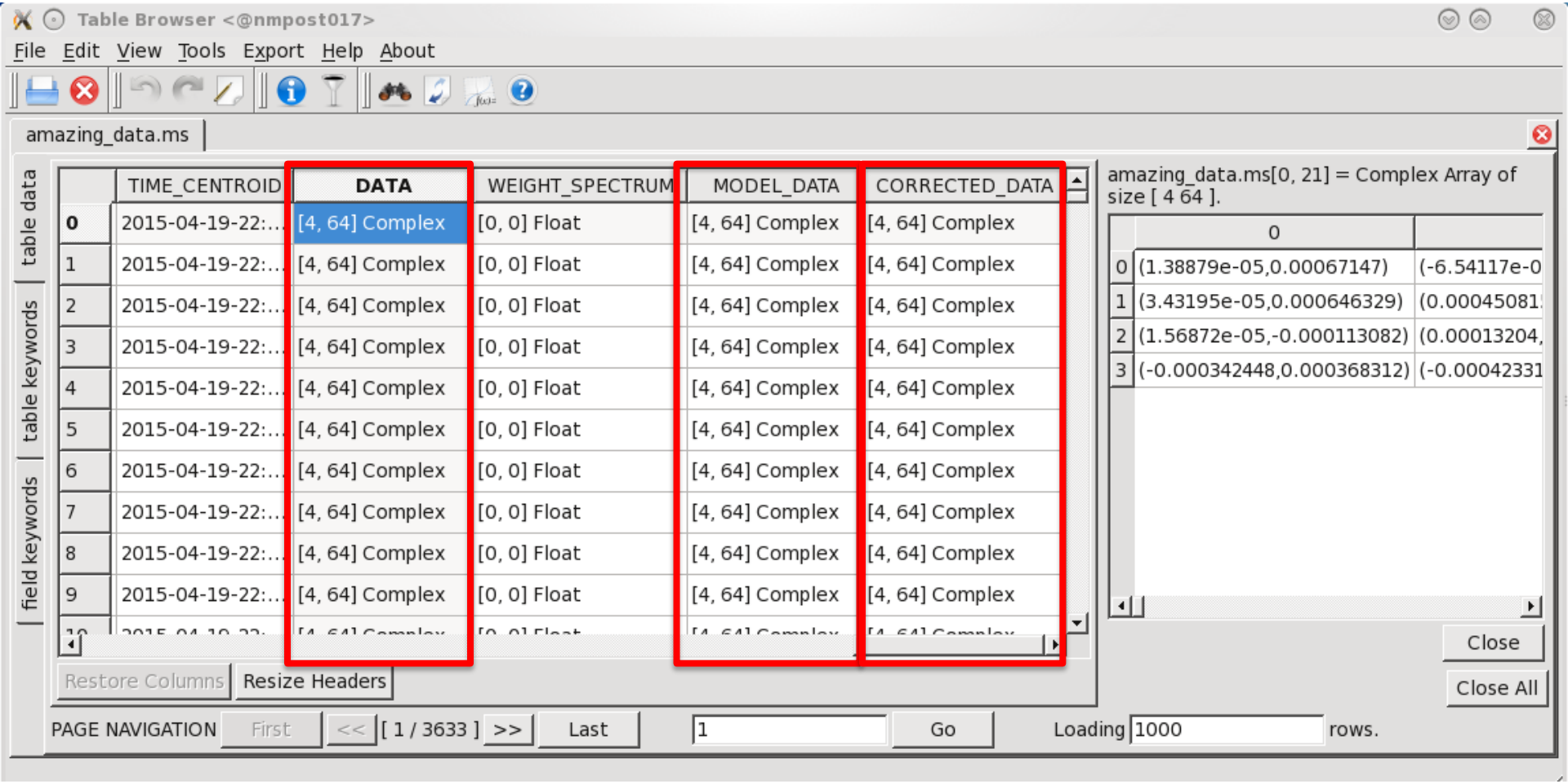
Inspect with task **browsetable** (from shell: **casabrowser**)



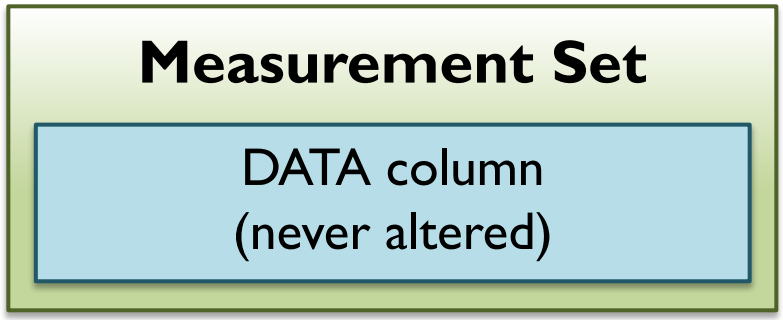
	UVW	FLAG	FLAG_CATEGORY	WEIGHT	SIGMA	ANTENNA1	ANTENNA2	ARRAY_ID	DA
0	[-278.403, ...	[4, 64] Boo...	[0, 0, 0] Boolean	[8.53333e...	[0.000176...	0	1	0	0
1	[2810.11, 2...	[4, 64] Boo...	[0, 0, 0] Boolean	[8.53333e...	[0.000176...	0	2	0	0
2	[426.12, 71...	[4, 64] Boo...	[0, 0, 0] Boolean	[8.53333e...	[0.000176...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

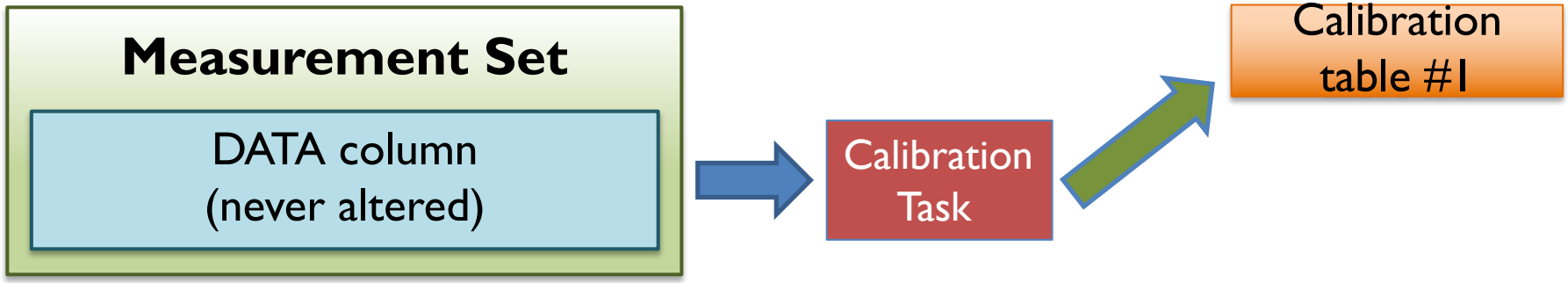
Inspect with task **browsetable** (from shell: **casabrowser**)



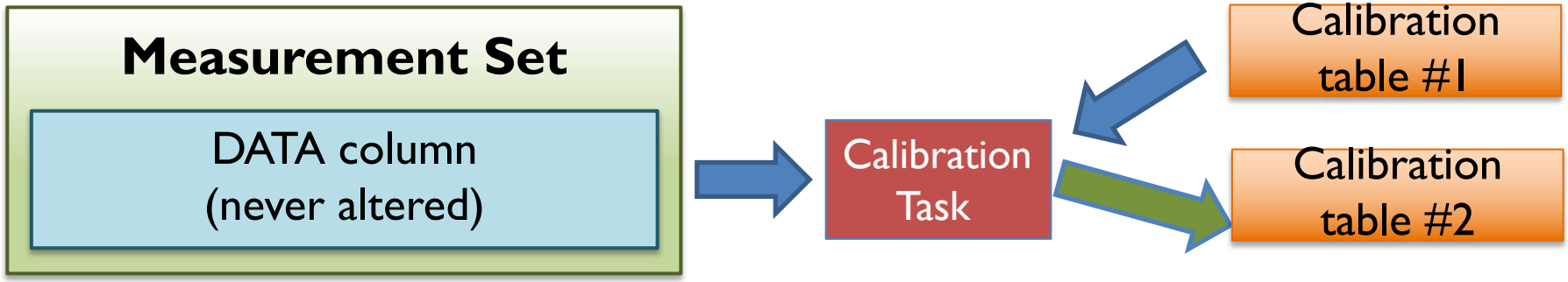
MS columns & calibration tables



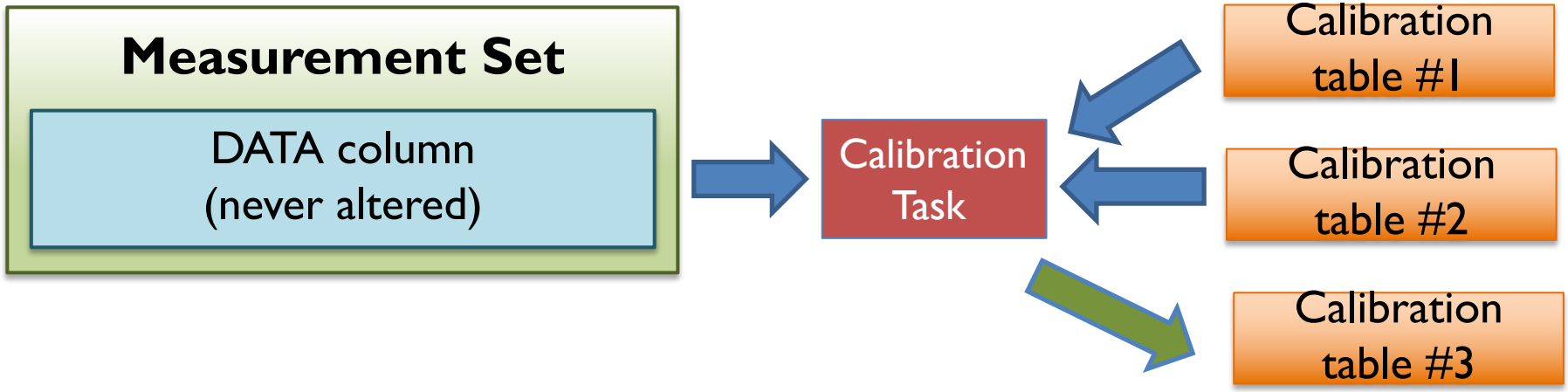
MS columns & calibration tables



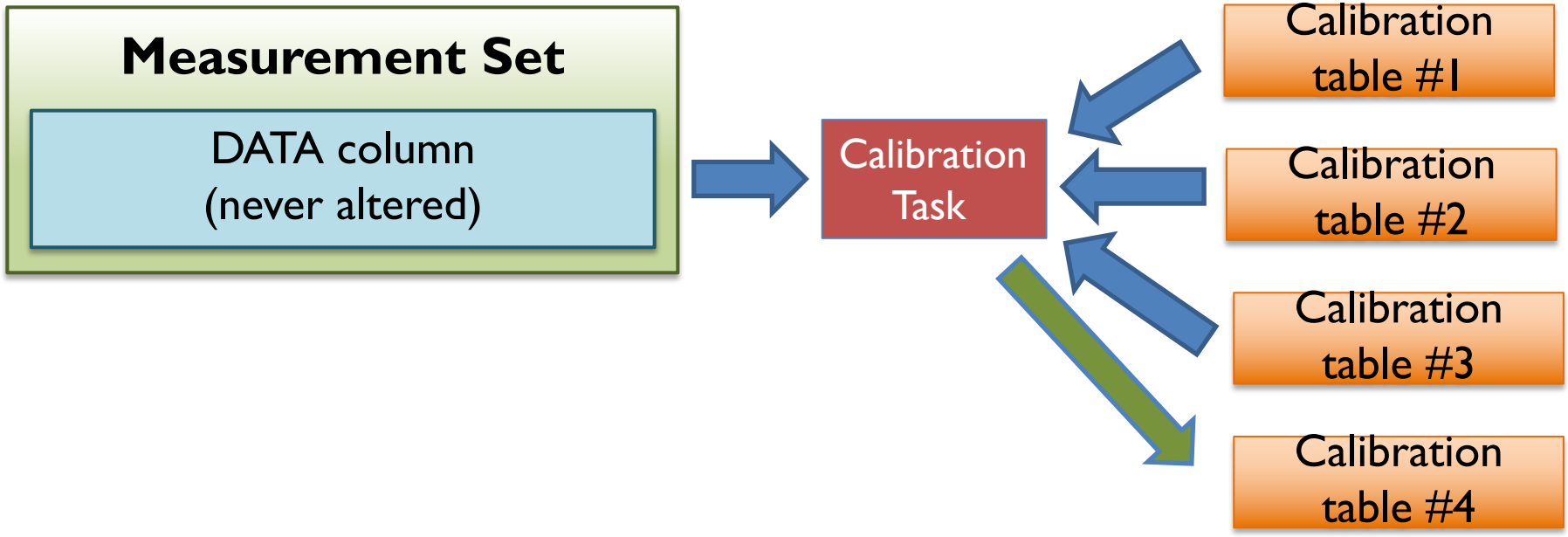
MS columns & calibration tables



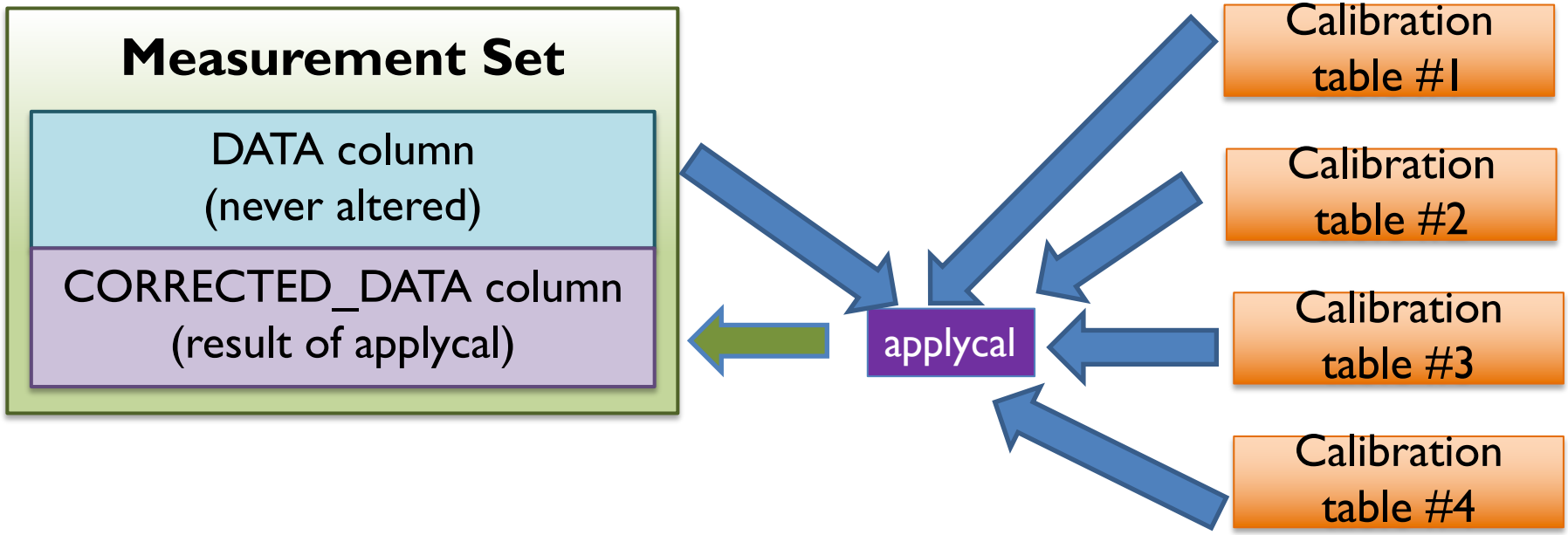
MS columns & calibration tables



MS columns & calibration tables

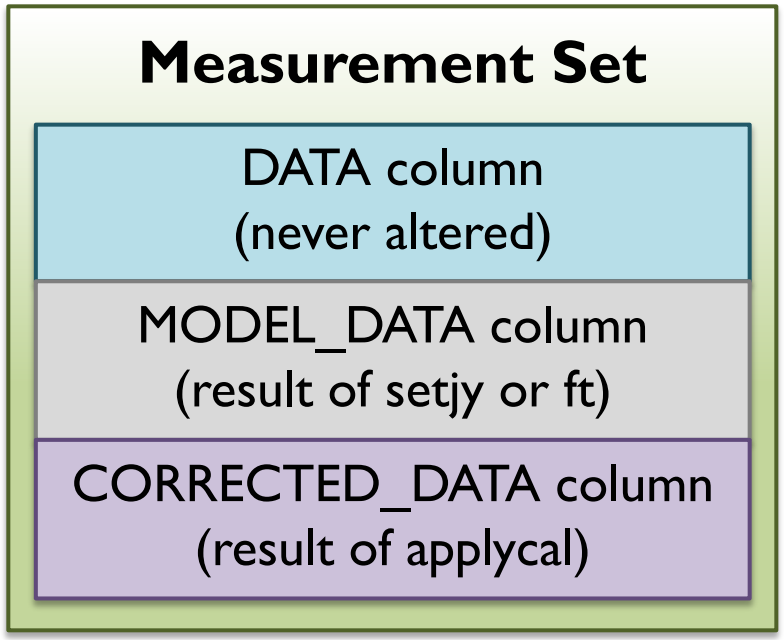


MS columns & calibration tables



If CORRECTED_DATA exists,
applycal will overwrite

MS columns & calibration tables

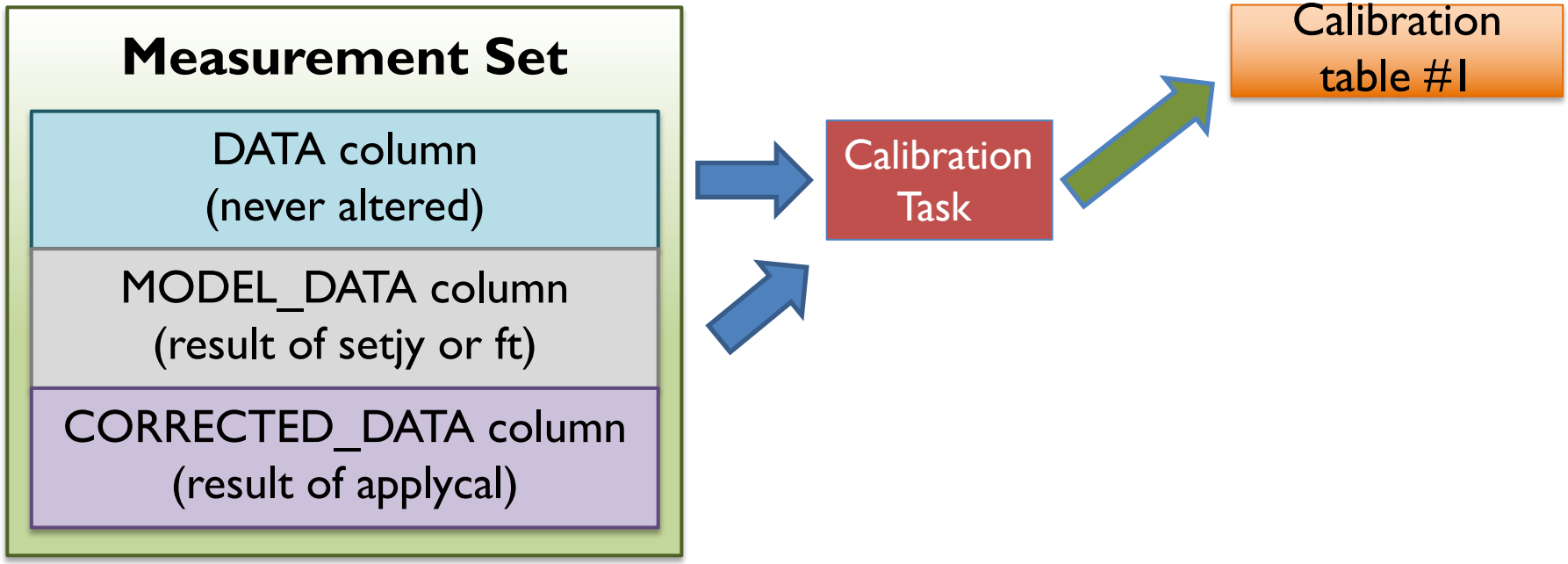


If a model is supplied in the MODEL_DATA column, the model will be used for the calibration tasks (otherwise a point source in the phase center is assumed)

Model supplied by
setjy
ft
tclean

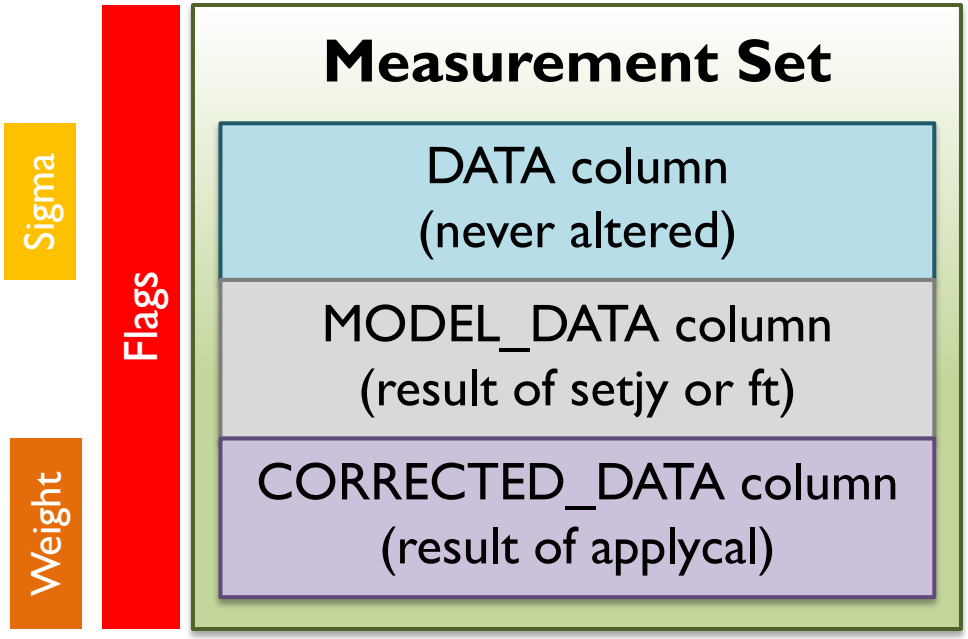
(note: data size tripled)

MS columns & calibration tables



(note: data size tripled)

MS columns & calibration tables



(note: data size tripled)

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)

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	310.9401	-194.1704	-2.6834	-1600888.922300	-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

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 image cube viewer
  (developed by NRAO, IDIA, ASIAA, U Alberta,
  1.3. release available on **http://cartavis.github.io/**)
- New Imaging Algorithms (incl. single dish + interferometric combination)
- Parallelization (mpicasa) for calibration and imaging
- VLA pipeline improvements (e.g. Imaging Pipeline for general VLA, polarization calibration, …; developed now for VLASS)
- Fringe fitting/VLBI (developed by JIVE and NRAO)
- CASAdocs transition for tools
- CASA 6 python 3, and modularization of CASA (**https://casa-pip.nrao.edu**)
- Nextgen CASA infrastructure for massive high performance computing applications

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.