CASA
Common Astronomy Software Applications

Introduction to CASA

Jürgen Ott (NRAO Socorro)

6th VLA Data Reduction Workshop
Socorro NM
23 October 2017

Atacama Large Millimeter/submillimeter Array
Expanded Very Large Array

NRAO

Overview of this talk

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

General description

- CASA: Common Astronomy Software Applications
  - Post-processing package for next gen facilities like ALMA and VLA, and both interferometric and single dish
  - Data from other telescopes also usually work
  - Developed at NRAO (lead), ESO, NAOJ, CSIRO/ATNF and ASTRON
  - Active in community since October 2007

- Code is C++ (fast) underneath iPython interface (easy access & scripting)

- Latest CASA release is **version 5.1.0/5.1.1**
  - Many tasks and a lot of tools
  - Contains automated calibration pipeline
  - Uses iPython 5.1.0 for Linux, 5.3.0 for Mac

http://casa.nrao.edu

CASA releases

- New releases about every 6 months (typically May and November)
  - Also "prerelease" versions that are markers on path to next release with more functionality, but likely contain unfinished developments, less tested code, and less up-to-date documentation

- Latest version 5.1.0/5.1.1 runs on:
  - Red Hat Linux 6 and 7 (64-bit)
  - Mac OS X 10.11 (El Capitan) and 10.12 (Sierra)
  - May also work on other systems, see website for details
  - 5.3.0 (spring 2018) is planned to run on Mac OS X 10.13 (High Sierra), drop 10.11

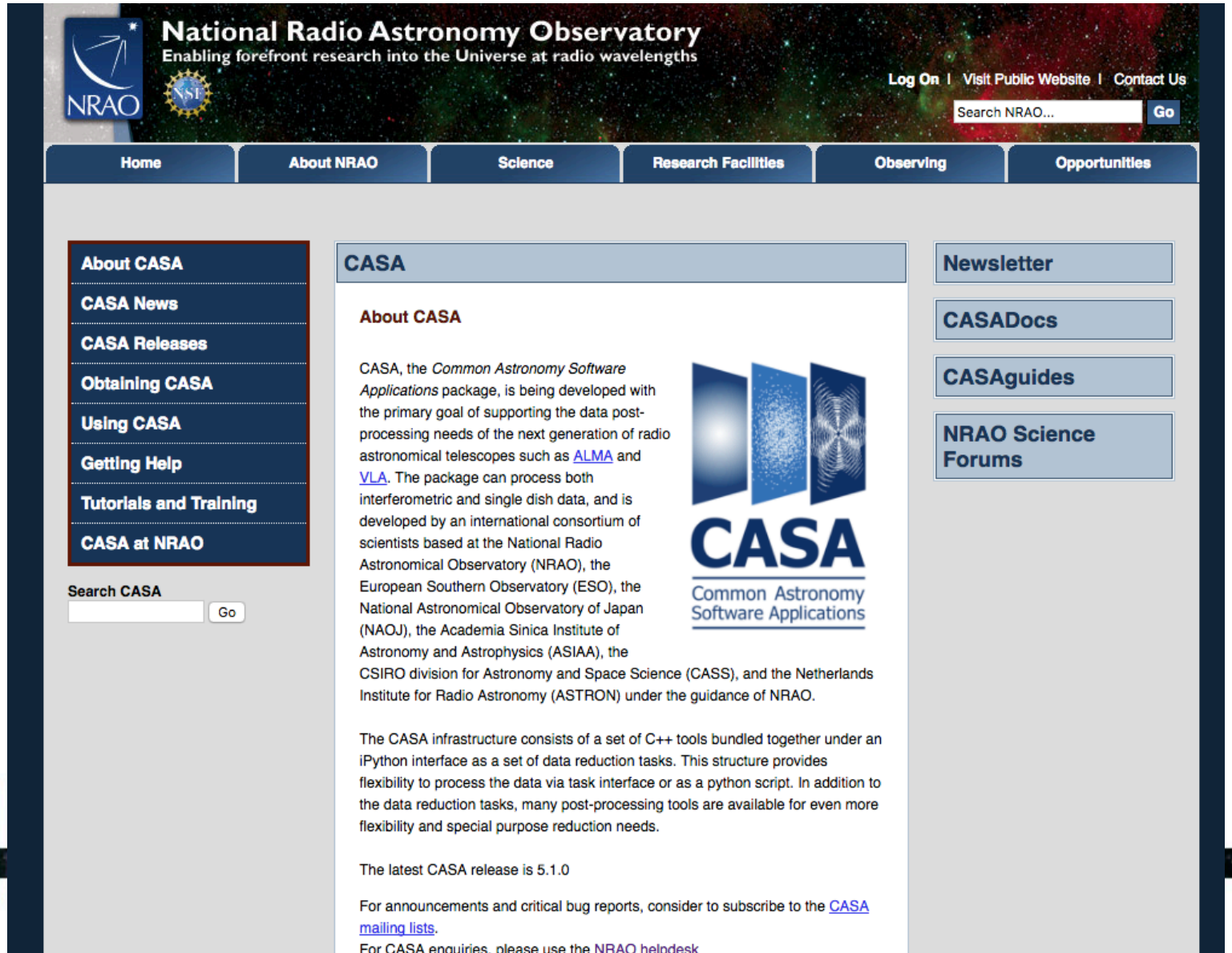
Why use CASA?

- Import data, inspect, edit, calibrate, image, view, analyze
  - CASA has some of the most sophisticated imaging algorithms (multi-scale clean, Taylor term expansion for wide bandwidths, W-term projection, OTF mosaicking, etc.)

- We have an active Algorithm Research Group, so expect more features in future versions, → various talks in this workshop, see also CASA newsletter

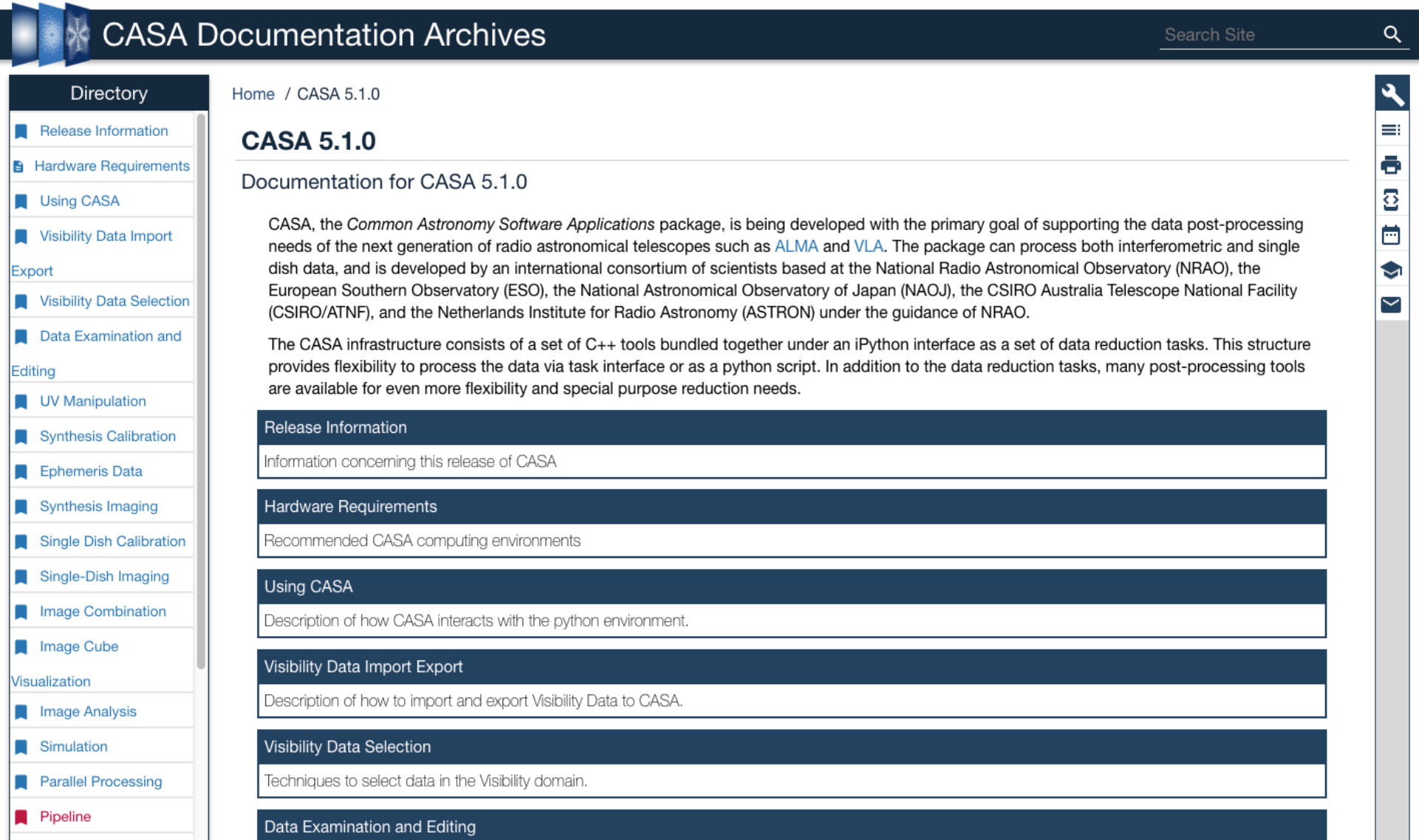
CASA documentation and web resources

CASA Homepage http://casa.nrao.edu



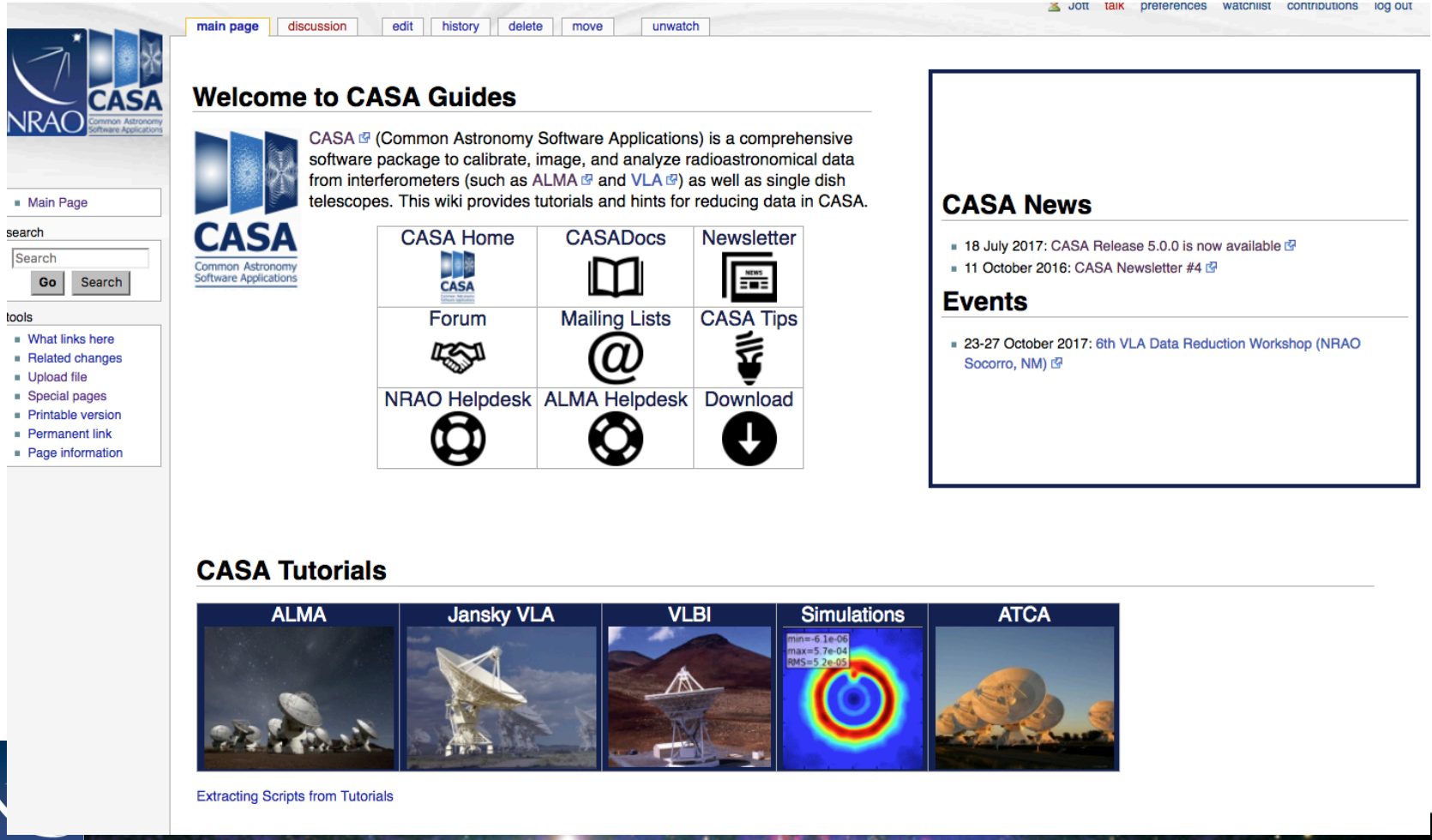
CASA documentation and web resources

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



CASA documentation and web resources

CASAguides https://casaguides.nrao.edu Tutorials



Starting CASA

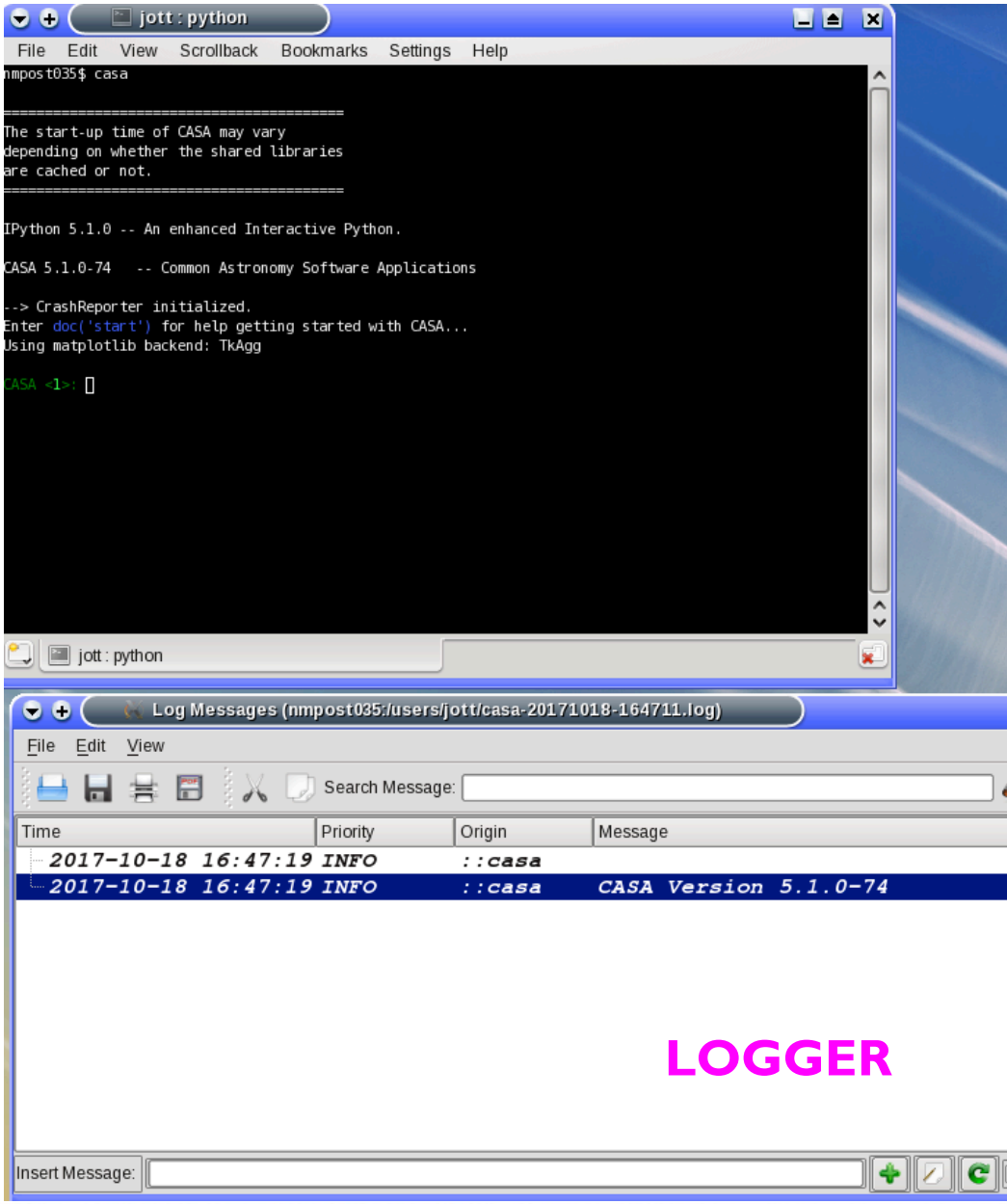
- Start CASA from the UNIX shell:

  % casa

- Session logging:
  - **ipython-TIMESTAMP.log**

    iPython command history
  - **casapy-TIMESTAMP.log**

    CASA logger messages



CASA interactive interface

- iPython interface (ipython.org) provides:
  - Numbered input/output
  - Shell access with leading exclamation mark, e.g. **!pwd** (or **os.system**)
  - Tab auto-completion
  - Auto-parenthesis
  - Command history (up-arrow or **hist [-n]**)
  - History/searching (start typing then use up-arrow, or use Ctrl-r)

- Some python tips:
  - Indentation matters, used for loops & conditions (beware copy paste)
  - Indices start from 0 and run to n-1

CASA tasks, tools, and applications

TASKS

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

TOOLS

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

APPLICATIONS

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

Find the right task

To see an organized list, type:

> tasklist

Find the right task

To see short summaries, type:

> taskhelp

Terminal <3>

File Edit View Search Terminal Help

```
CASA <12>: taskhelp
---------> taskhelp()
Available tasks:

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

Task interface

Inspect task inputs:

> inp clean

Black: default value

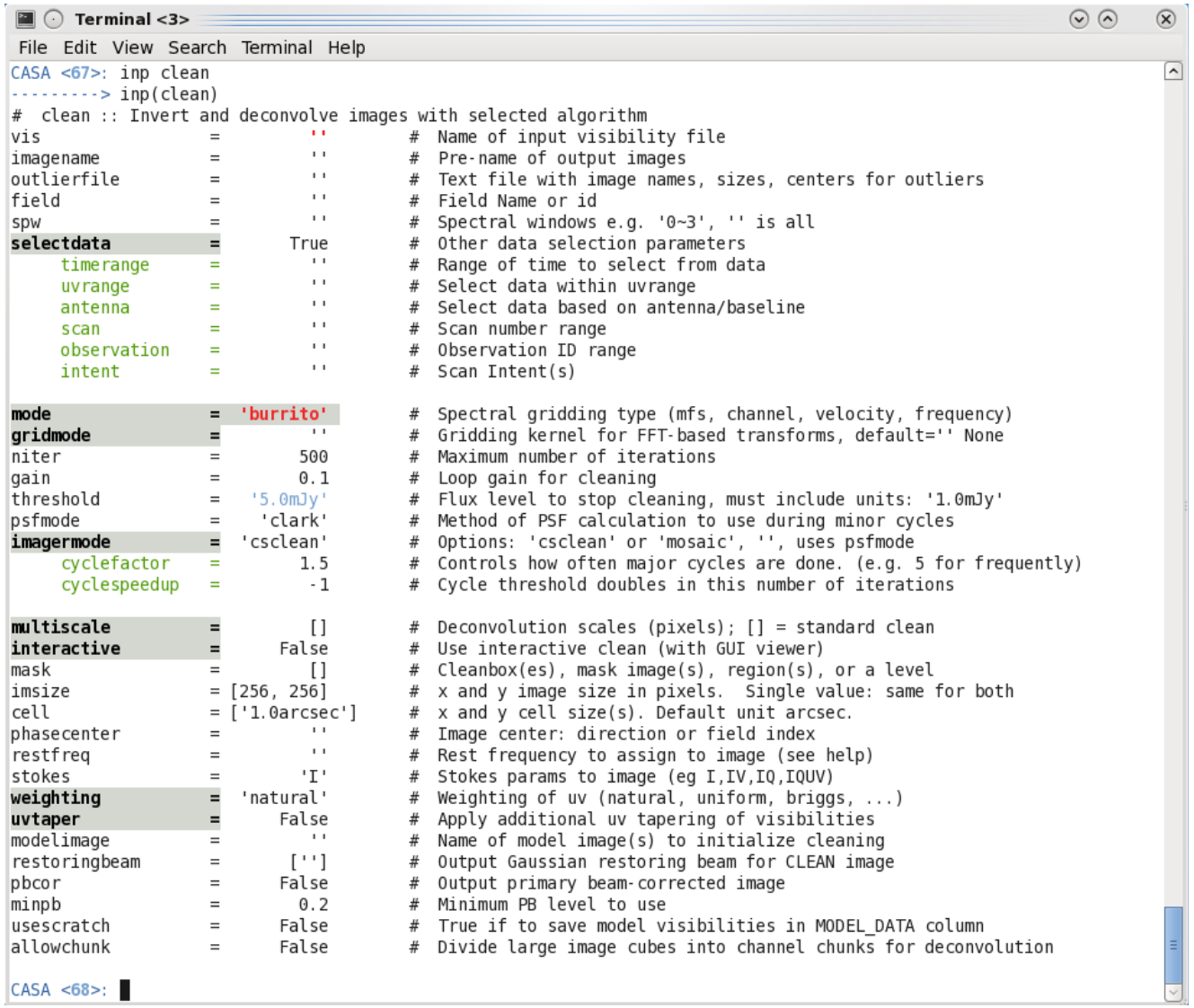
Red: invalid value

Blue: non-default value

Reset defaults:

> default clean

Grey: expandable

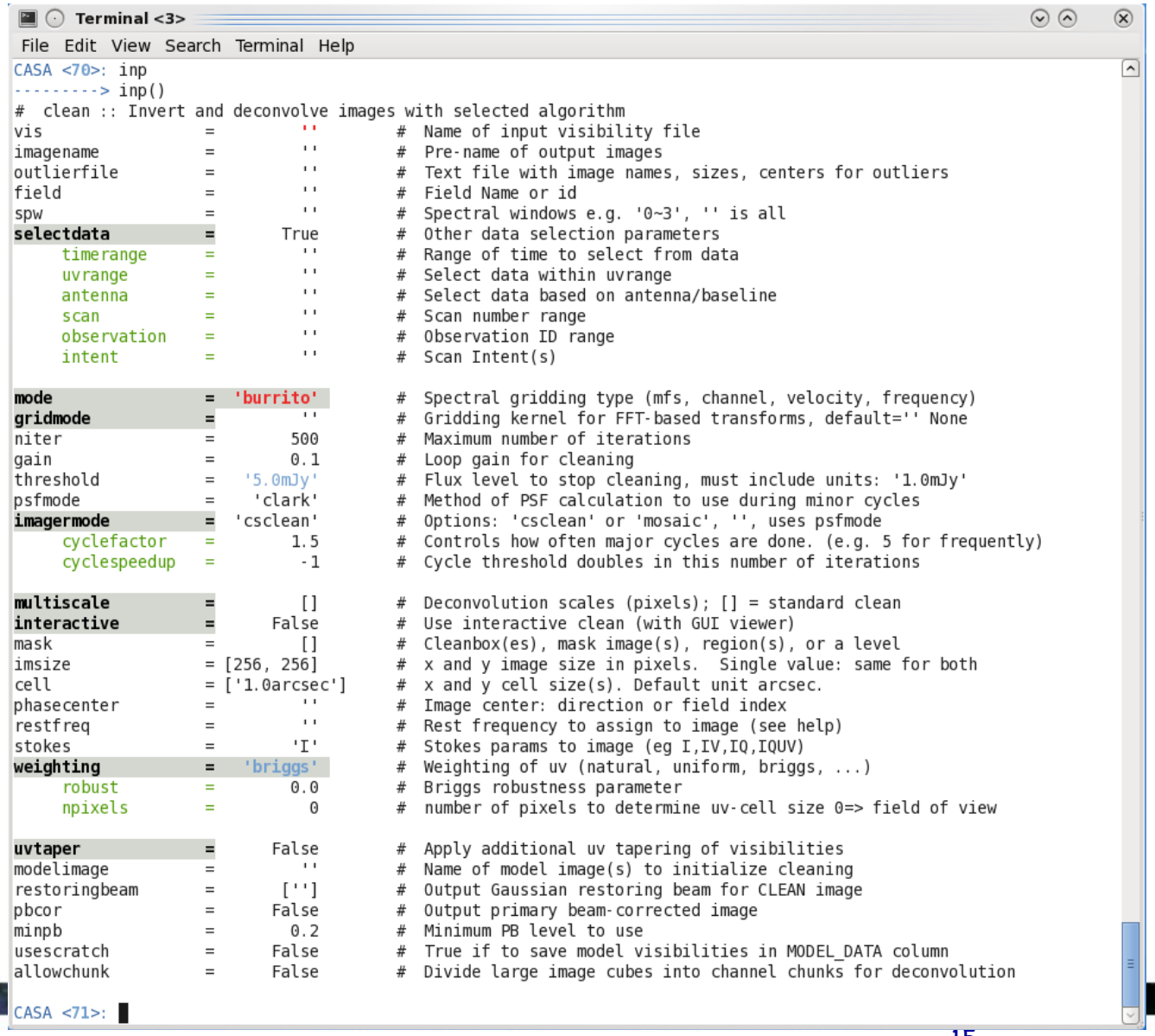


Task interface

Grey: expandable

Green: sub-parameter

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



Task help

Type:

> help clean

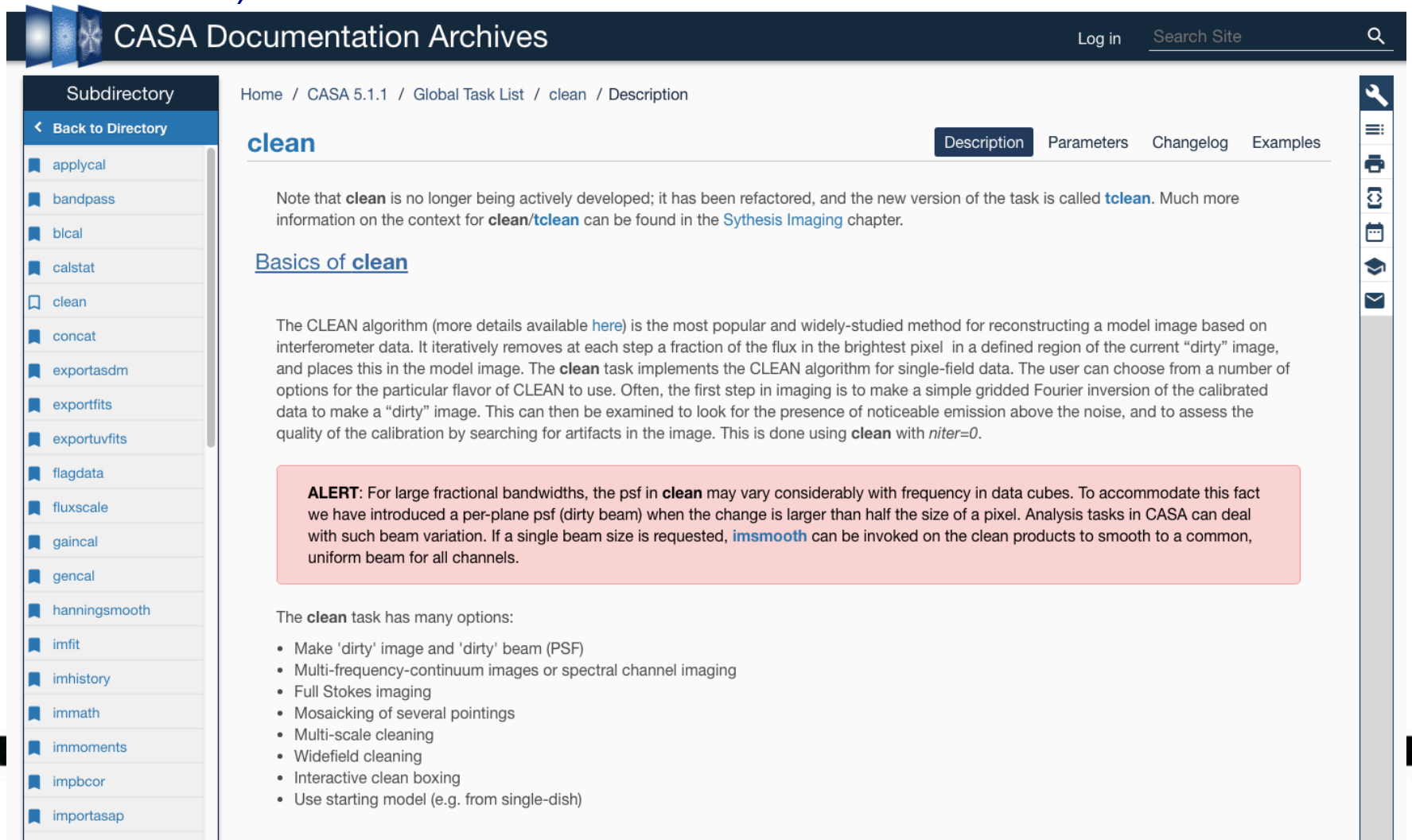
Terminal <3>

File Edit View Search Terminal Help

```
 Example :

The clean task has many options:

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


vis -- Name(s) of input visibility file(s)
        default: none;
        example: vis='ngc5921.ms'
                 vis=['ngc5921a.ms','ngc5921b.ms']; multiple MSes
imagename -- Pre-name of output images:
        default: none; example: imagename='m2'
        output images are:
          m2.image; cleaned and restored image
                 With or without primary beam correction
          m2.psf; point-spread function (dirty beam)
          m2.flux;  relative sky sensitivity over field
          m2.flux.pbcoverage;  relative pb coverage over field
                               (gets created only for ft='mosaic')
          m2.model; image of clean components
          m2.residual; image of residuals
          m2.interactive.mask; image containing clean regions
        To include outlier fields:
          imagename=['n5921','outlier1','outlier2']
outlierfile --- Text file name which contains image names, sizes, field
                centers (See 'HINTS ON CLEAN WITH FLANKING FIELDS' below
                for the format of this outlier file.)
field -- Select fields to image or mosaic.  Use field id(s) or name(s).
           ['go listobs' to obtain the list id's or names]
        default: ''= all fields
          If field string is a non-negative integer, it is assumed to
          be a field index otherwise, it is assumed to be a
          field name
          field='0~2'; field ids 0,1,2
          field='0,4,5~7'; field ids 0,4,5,6,7
          field='3C286,3C295'; field named 3C286 and 3C295
          field = '3,4C*'; field id 3, all names starting with 4C
          For multiple MS input, a list of field strings can be used:
          field = ['0~2','0~4']; field ids 0-2 for the first MS and 0-4
lines 301-348
```

Task help

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



How to run a task

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

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

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

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

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

Some things to note about running tasks

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

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

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

  - Exception is if global parameter is explicitly referenced

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

Tools

- What if there's no task?

  → Use CASA tools!

- Tools (and their methods) are the building blocks of tasks
  - Contain full functionality of CASA
  - Used internally by tasks
  - E.g. imaging utilities (im), table utilities (tb), …

Using tools and their methods

- Tools contain a number of methods
  - Access using **tool.method()**
  - Use tab-completion to see listing

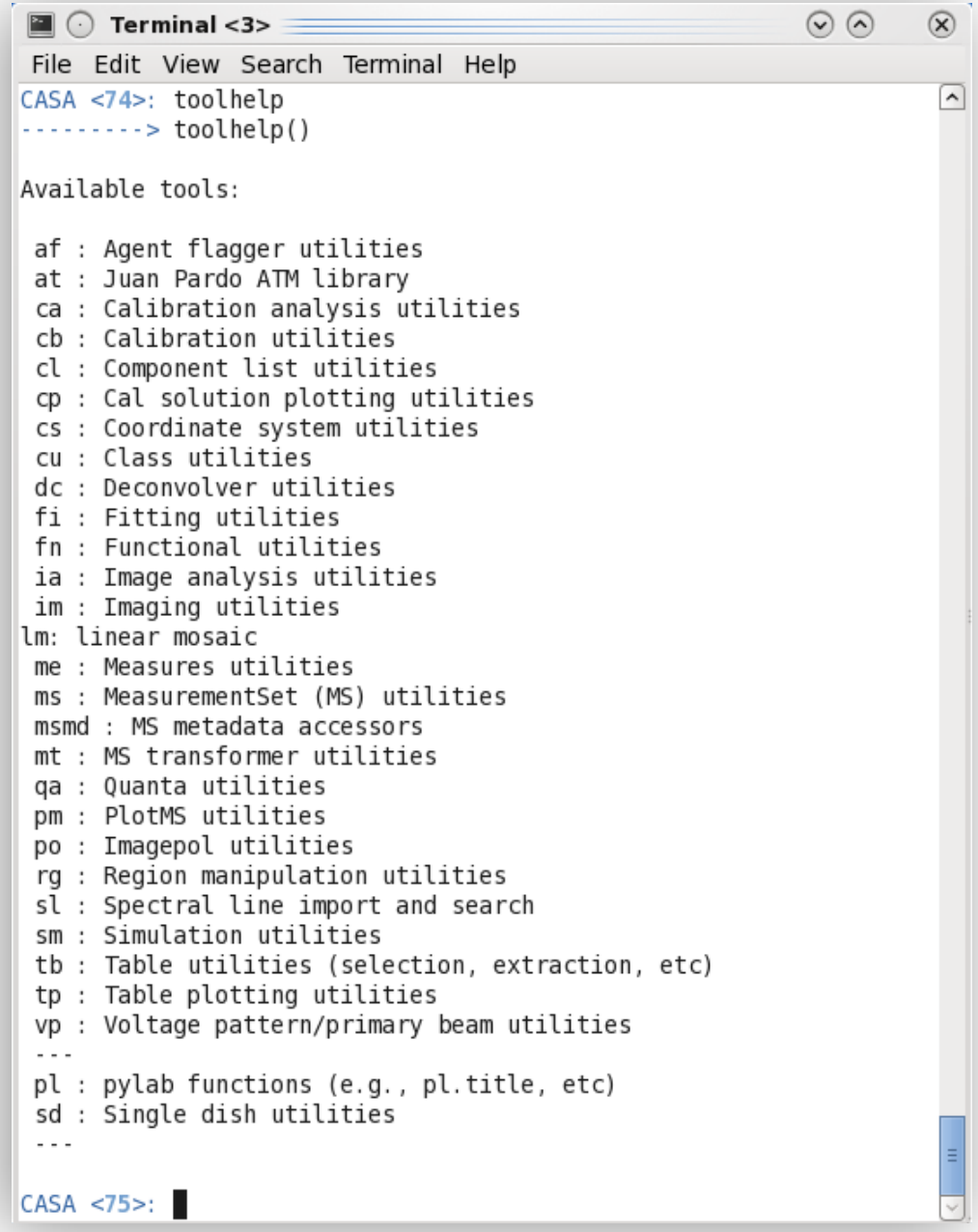
- Typically, data must be opened and closed (unlike tasks)
  - Failure to close may block other tasks and clutter memory

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

Find the right tool

To see short summaries, type:

> toolhelp



CASA toolkit reference manual

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

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

  - → see Steve Myer's talk

CASA toolkit reference manual

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

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

- → see Steve Myer's talk

Transition to CASAdocs underway

Still searching for functionality?

- Look through contributed scripts and tasks at:

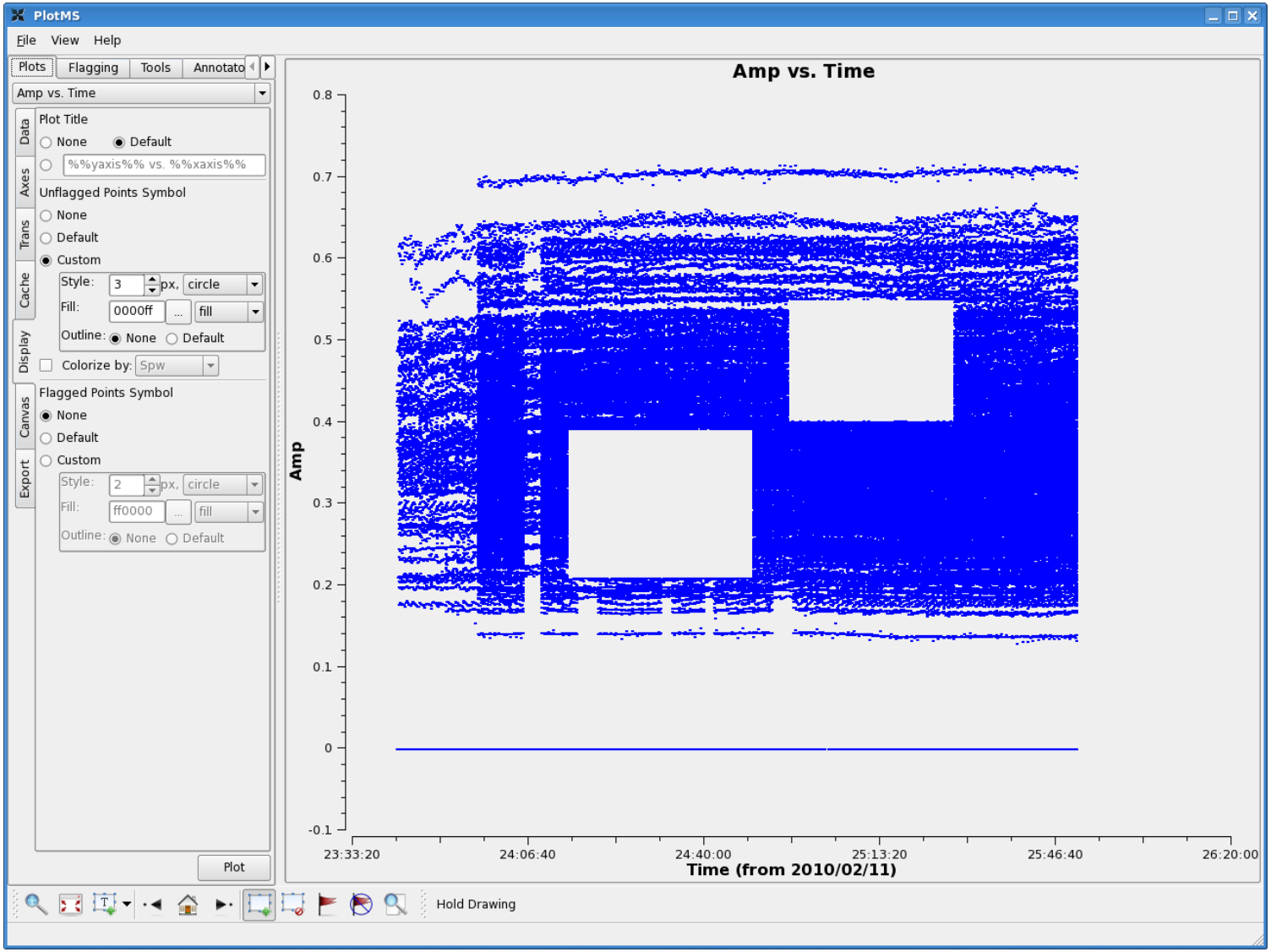
  http://casaguides.nrao.edu/

- If you still can't find what you need, write your own task!
  - Combination of Python plus CASA toolkit is very powerful
  - We encourage you to submit your own scripts to us through the helpdesk

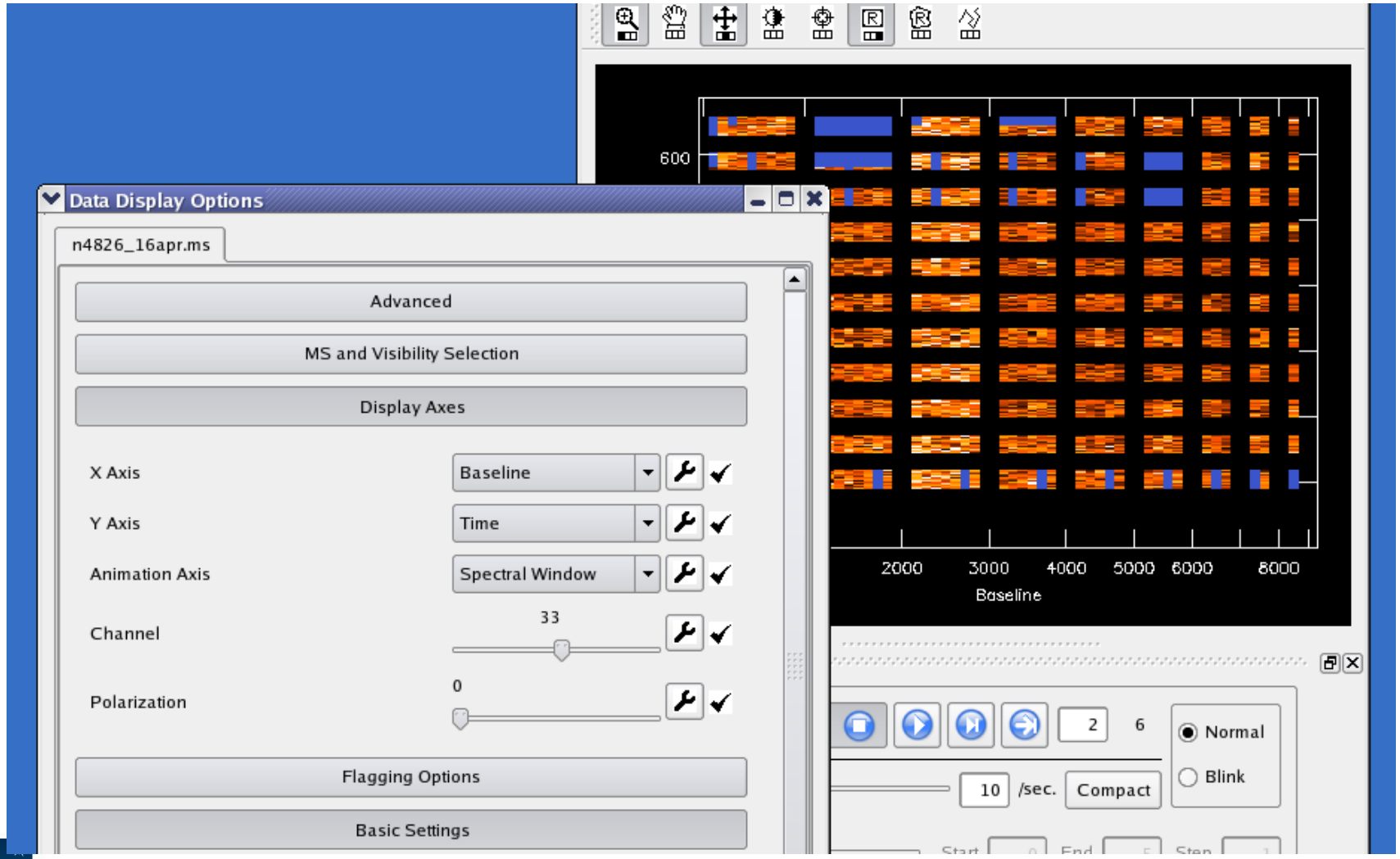
Applications

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

PlotMS


PlotMS
File View Help
Plots Flagging Tools Annotato
Amp vs. Time
Data
Axes
Trans
Cache
Display
Canvas
Export
Plot Title
None Default
%%yaxis%% vs. %%xaxis%%
Unflagged Points Symbol
None
Default
Custom
Style: 3 px, circle
Fill: 0000ff ... fill
Outline: None Default
Colorize by: Spw
Flagged Points Symbol
None
Default
Custom
Style: 2 px, circle
Fill: ff0000 ... fill
Outline: None Default
Plot
Amp vs. Time
Amp
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
23:33:20
24:06:40
24:40:00
25:13:20
25:46:40
26:20:00
Time (from 2010/02/11)
Hold Drawing

MSView


Data Display Options
n4826_16apr.ms
Advanced
MS and Visibility Selection
Display Axes
X Axis
Baseline
Y Axis
Time
Animation Axis
Spectral Window
Channel
33
Polarization
0
Flagging Options
Basic Settings
600
2000
3000
4000
5000
6000
8000
Baseline
2
6
Normal
Blink
10
/sec.
Compact

Plotcal

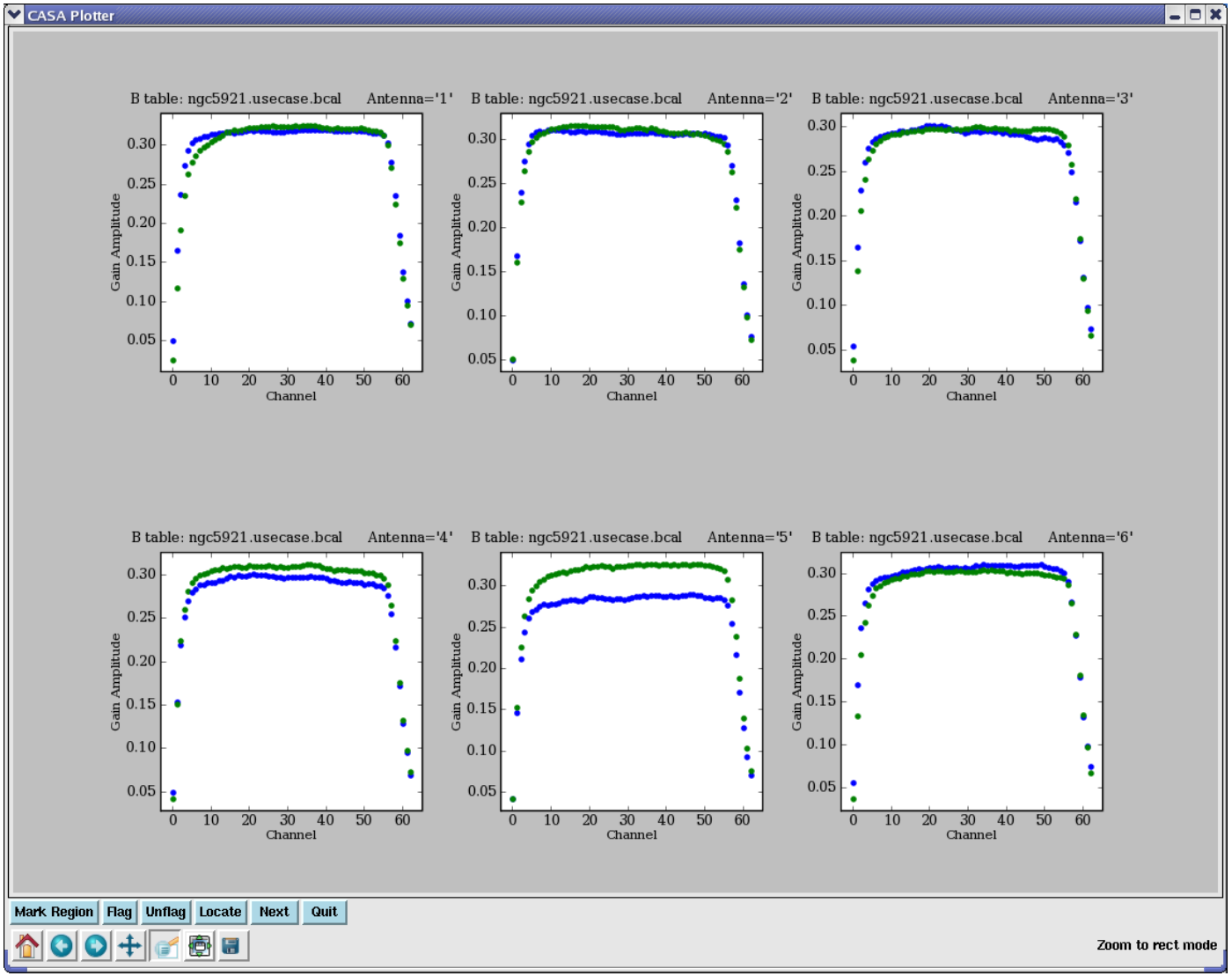

CASA Plotter
B table: ngc5921.usecase.bcal Antenna='1'
B table: ngc5921.usecase.bcal Antenna='2'
B table: ngc5921.usecase.bcal Antenna='3'
B table: ngc5921.usecase.bcal Antenna='4'
B table: ngc5921.usecase.bcal Antenna='5'
B table: ngc5921.usecase.bcal Antenna='6'
Gain Amplitude
Channel
Mark Region
Flag
Unflag
Locate
Next
Quit
Zoom to rect mode

Image Viewer

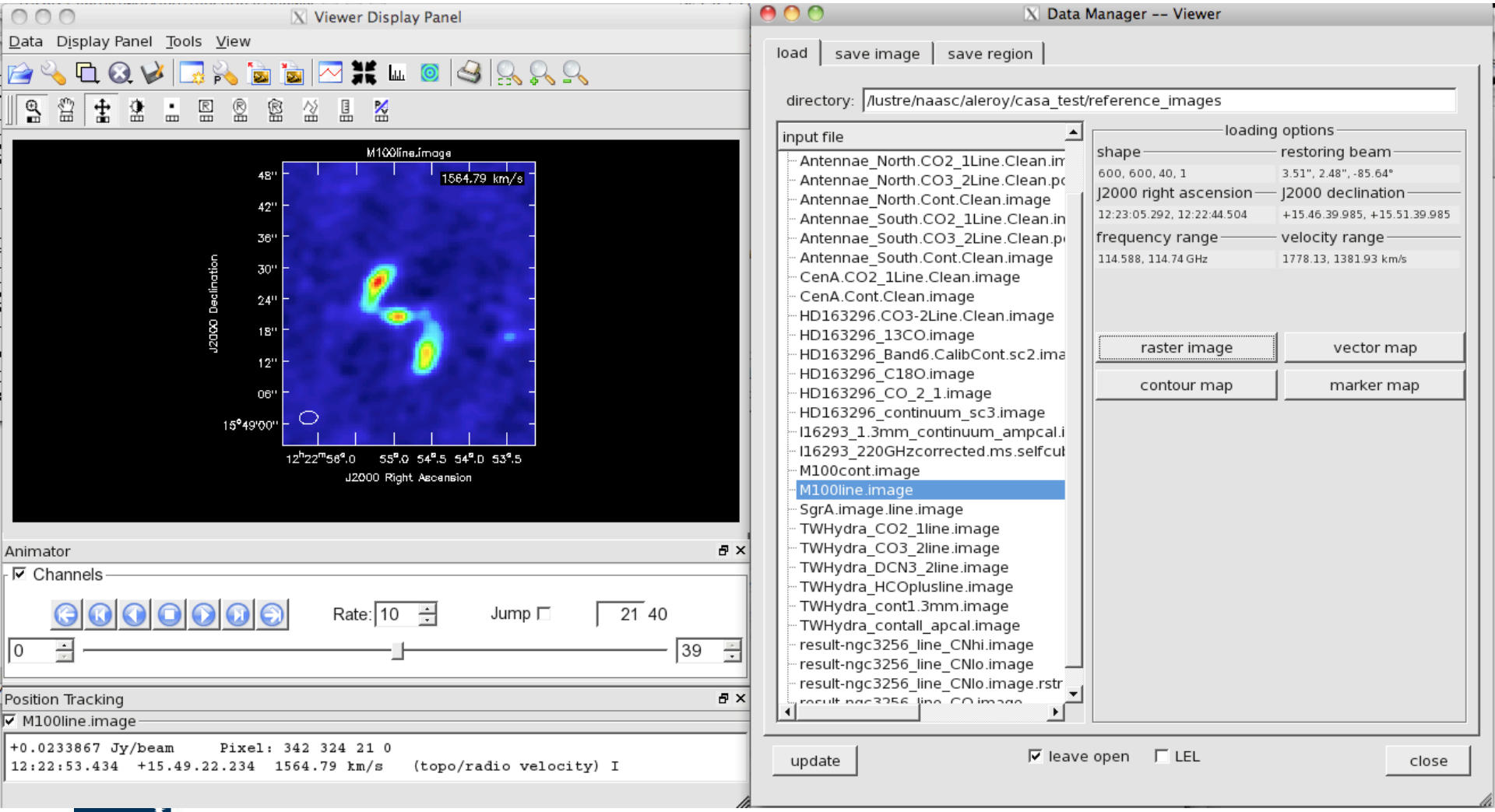
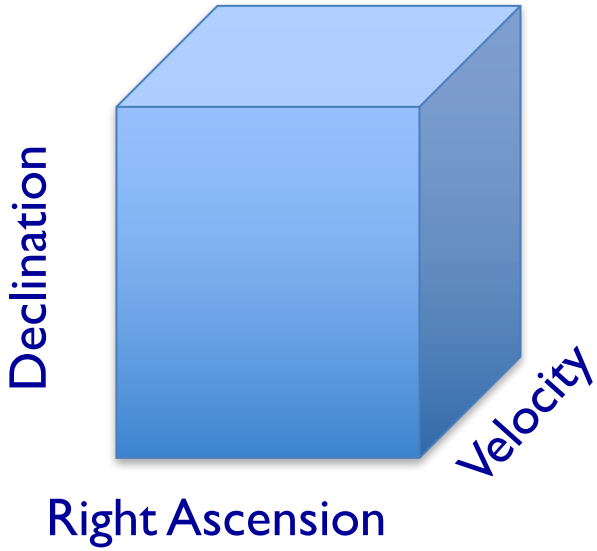


Image Viewer

- Cubes
- Movies
- Channel maps



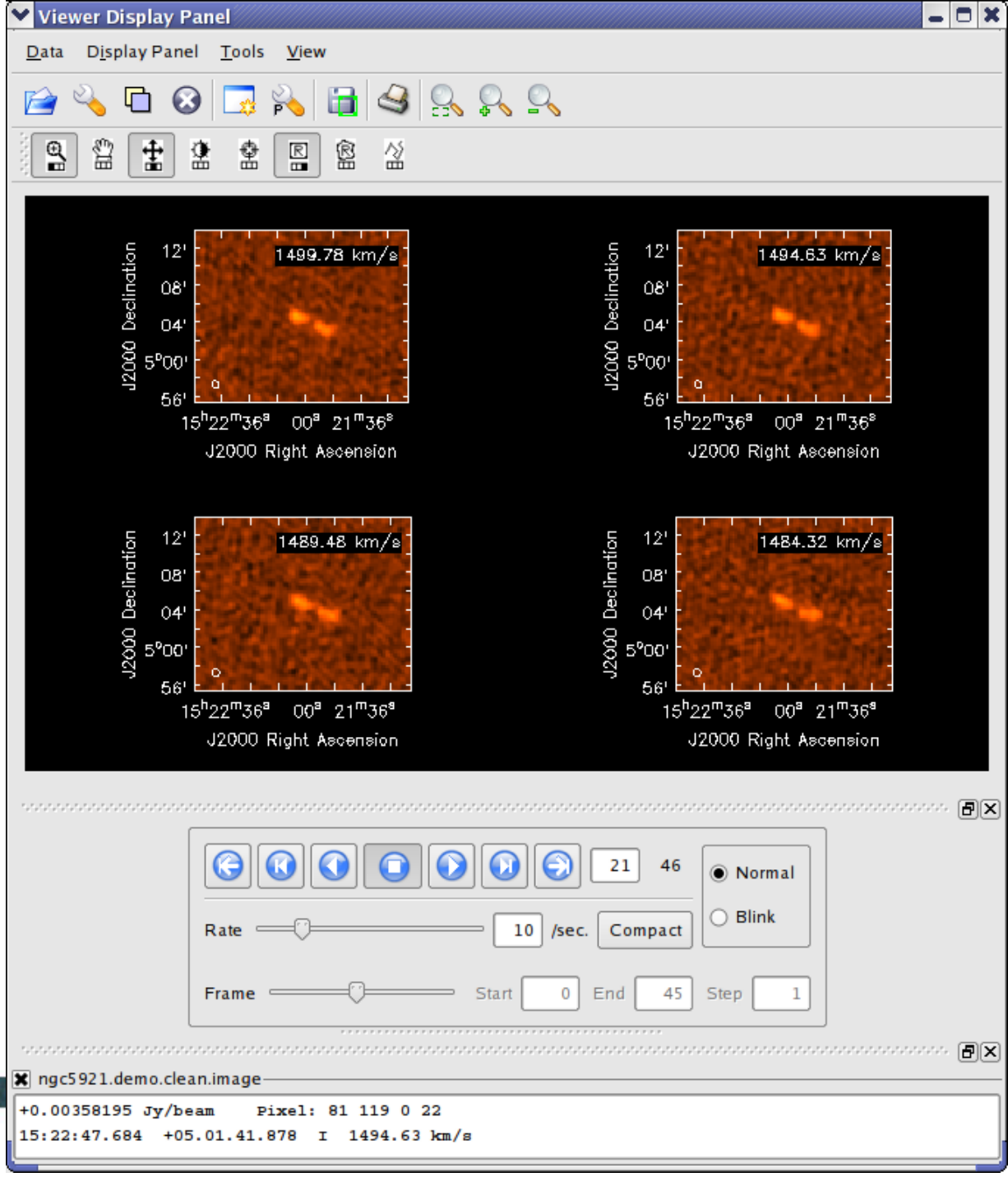
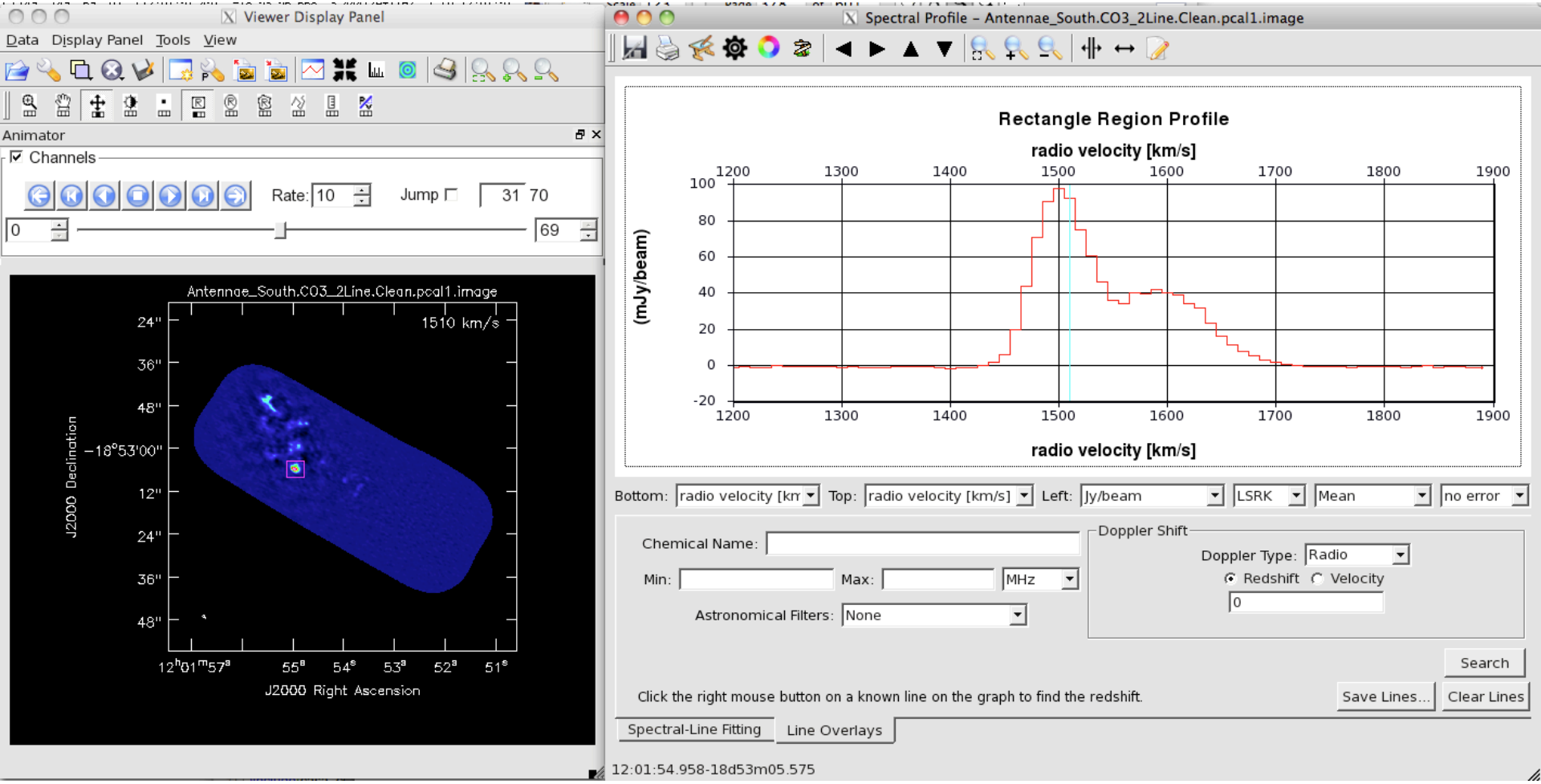
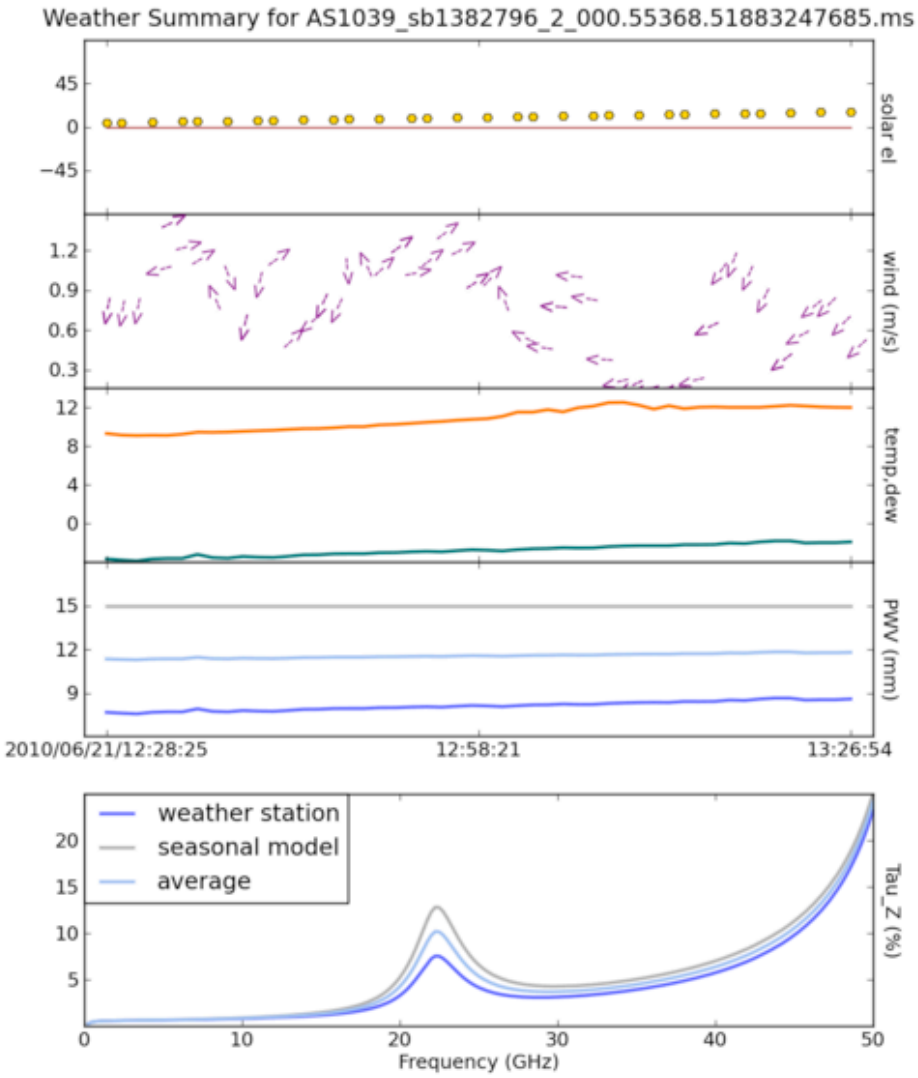


Image Viewer



Plot anything - matplotlib



Data structures

- Observatory data stored in (A)SDM format, CASA uses measurement sets
  - Use **importasdm** for ALMA, **importasdm** or **importevla** for EVLA/JVLA, **importvla** for old VLA, etc.

- Measurement set contains visibilities

  Calibration information stored in calibration tables (not in ms)

  Images are in CASA image format

- All of these are *directories* which contain the necessary information
  - Copying requires recursive option (**!cp –r**)

- Delete tables using **rmtables('mydata.ms')**

  !rm –rf or **os.system('rm –rf mydata.ms')** may also work,

  but could leave traces in the cache

Inspect a measurement set (ms)

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

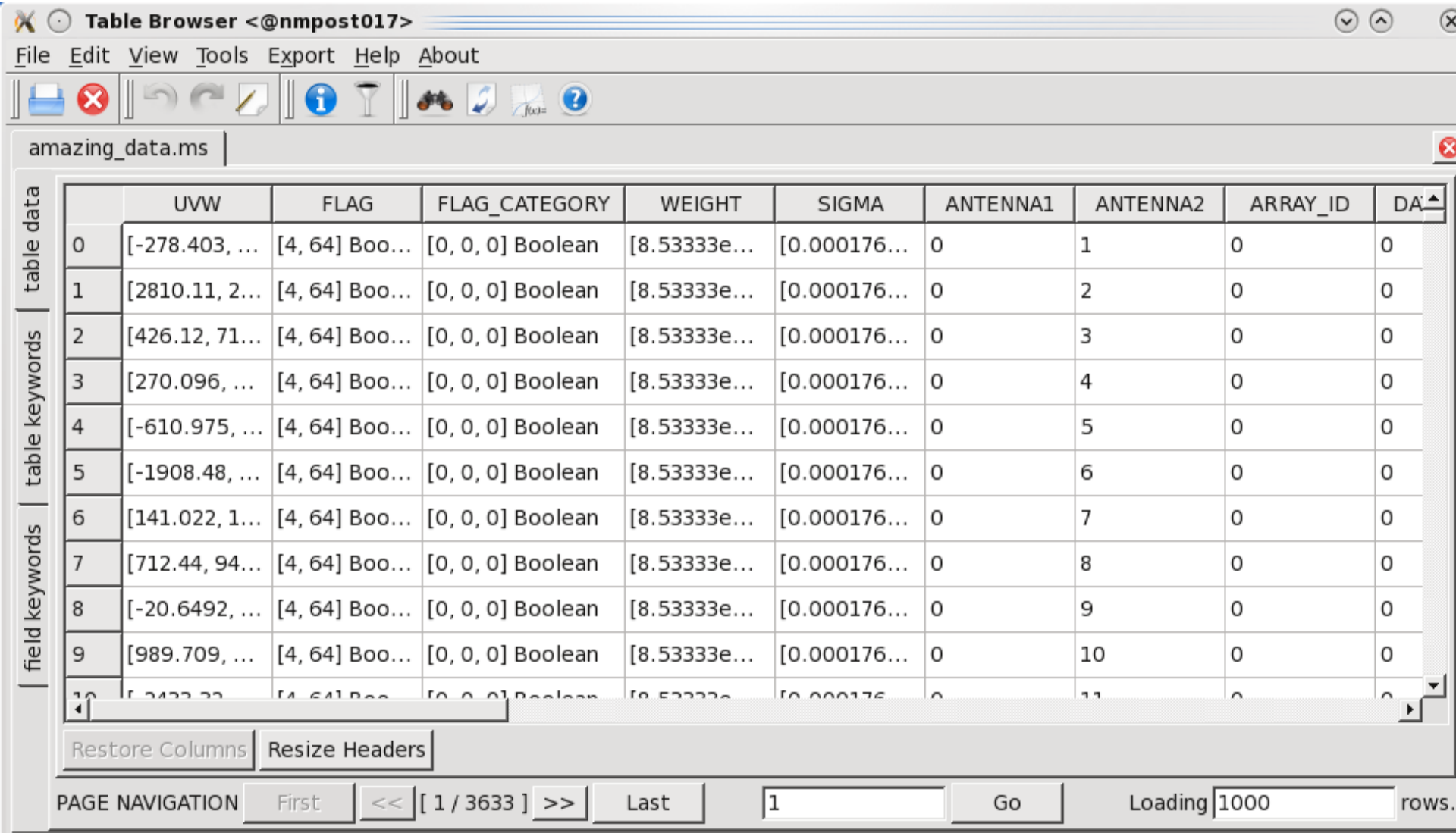
```
CASA <80>: !ls amazing_data.ms
ANTENNA           POINTING         SYSPOWER   table.f15       table.f20_TSM0  table.f24_TSM1  table.f8
CALDEVICE         POLARIZATION     table.dat  table.f16       table.f21       table.f25       table.f9
DATA_DESCRIPTION  PROCESSOR        table.f1   table.f17    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

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

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

MS MAIN table contents

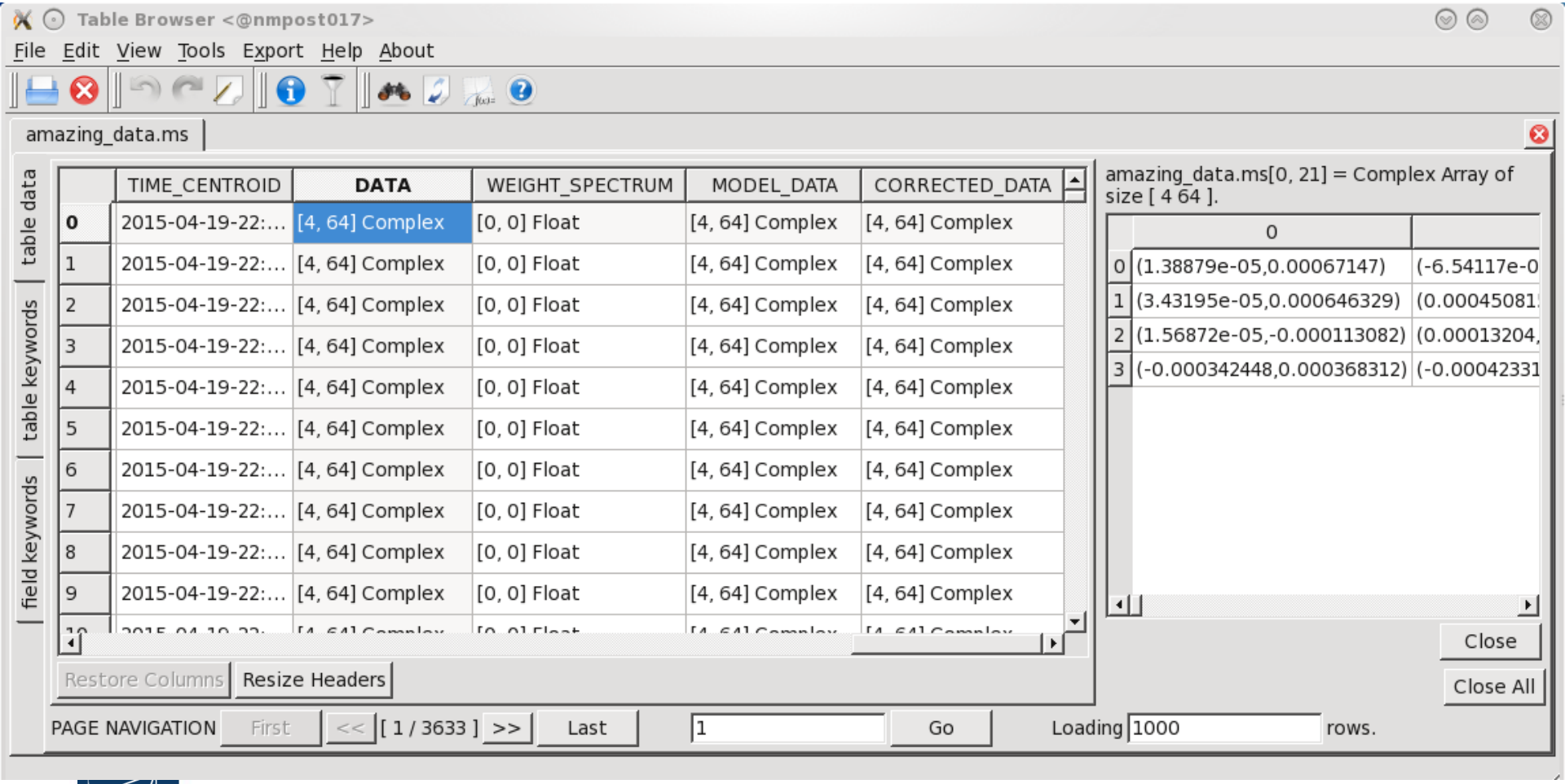
Inspect with task **browsetable** (or standalone **% casabrowser**)



	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

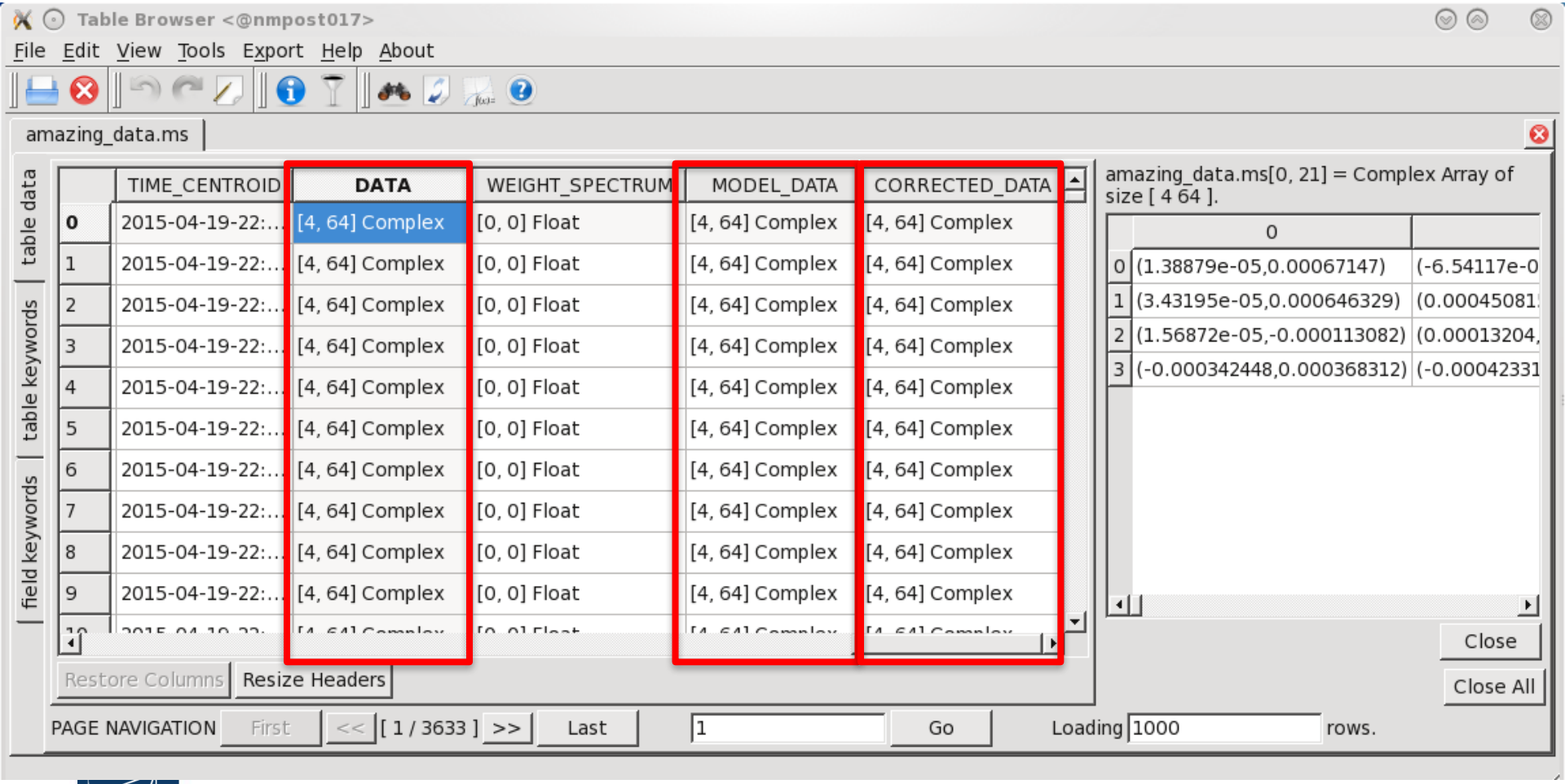
MS MAIN table contents

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

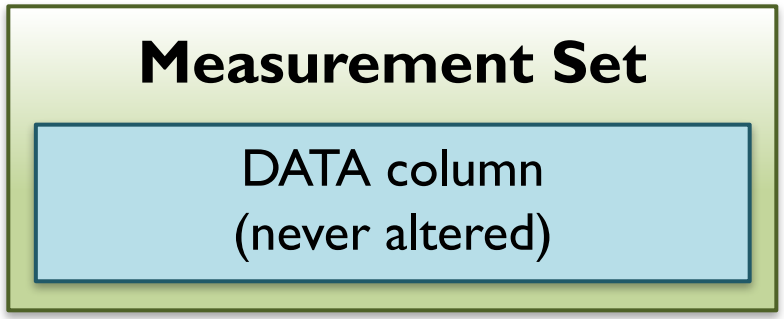


MS MAIN table contents

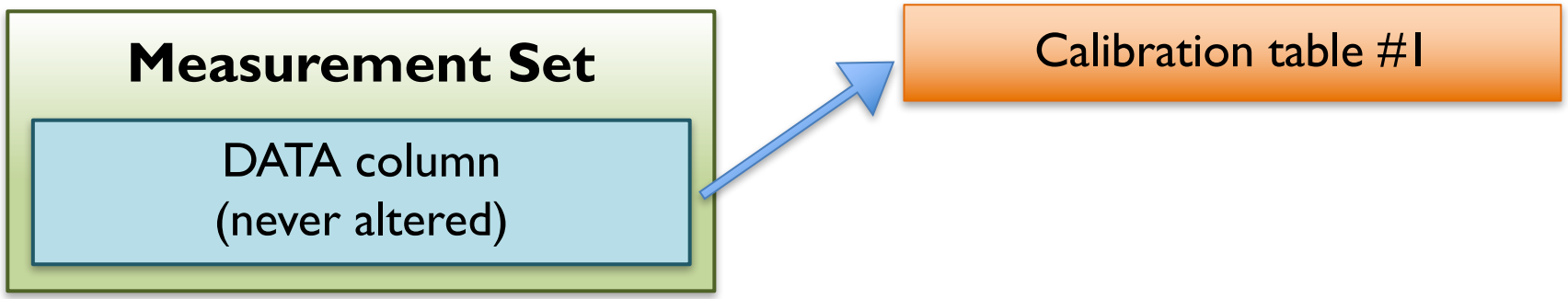
Inspect with task **browsetable** (or standalone **% casabrowser**)



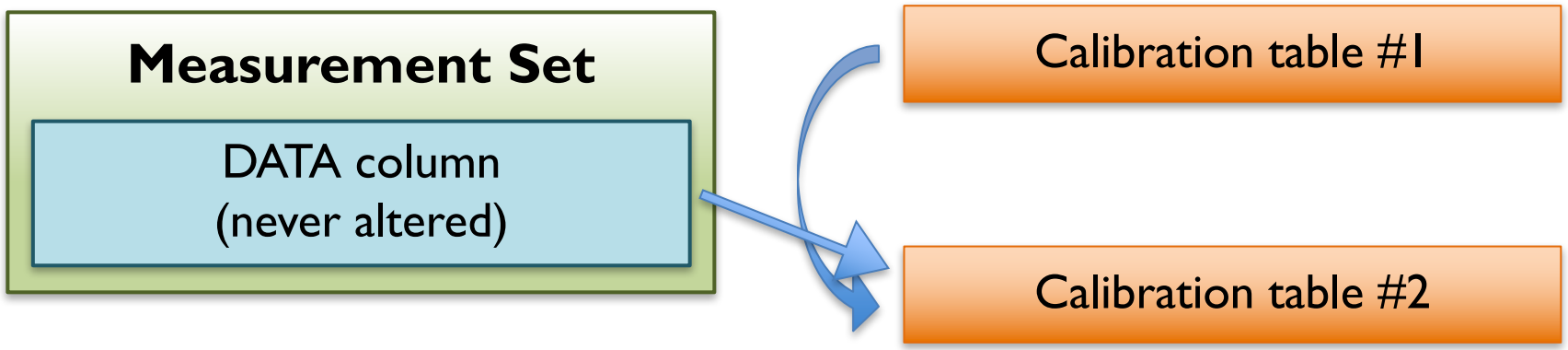
MS columns & calibration tables



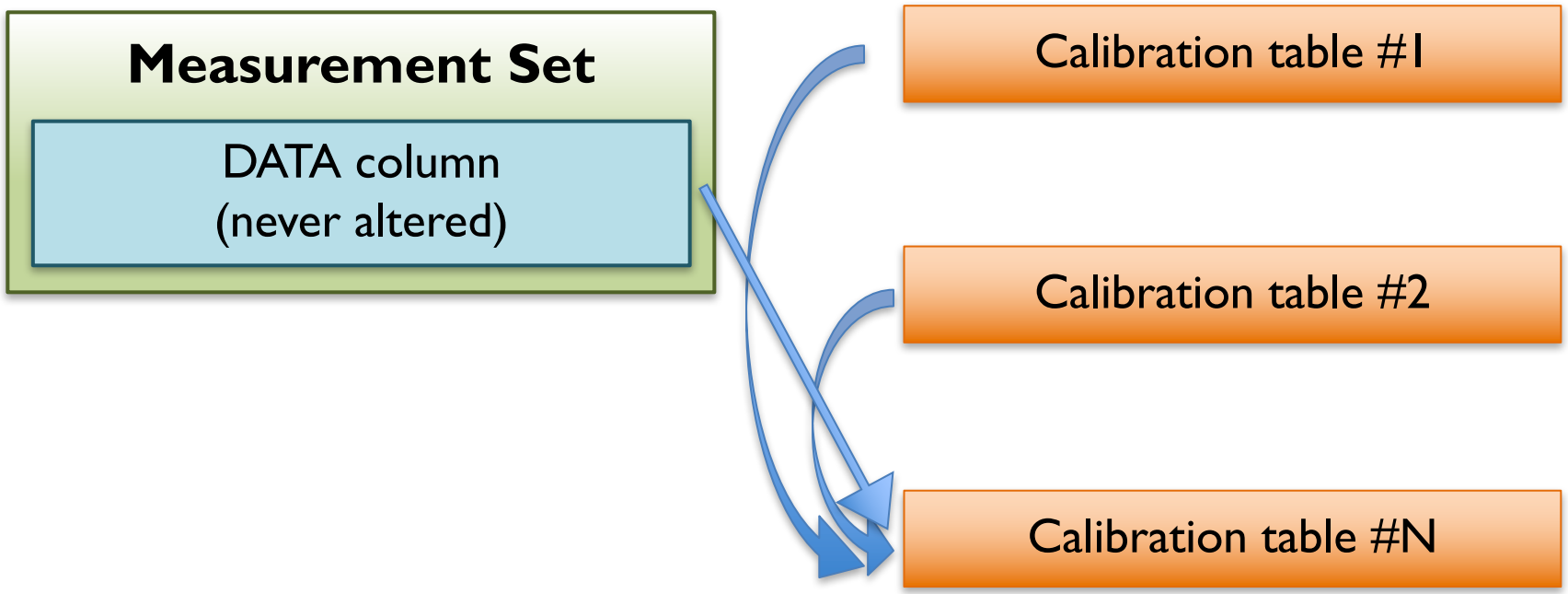
MS columns & calibration tables



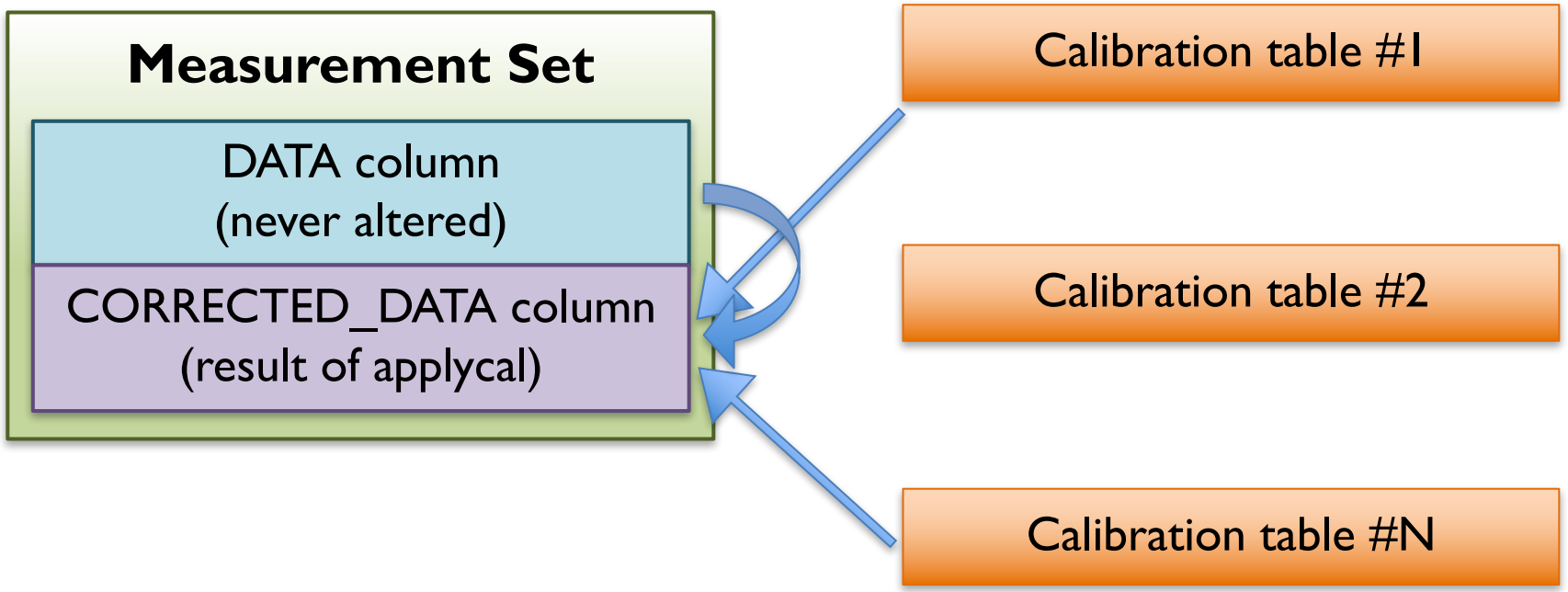
MS columns & calibration tables



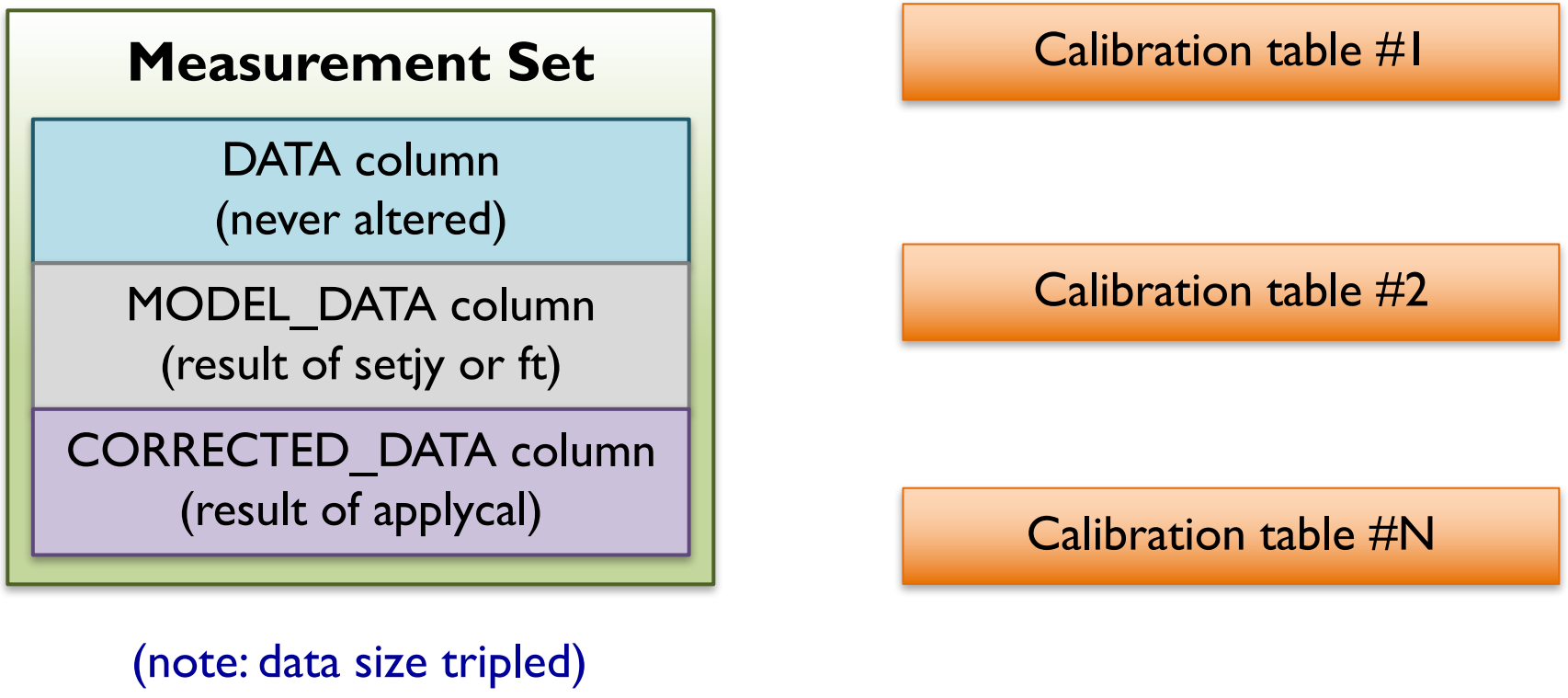
MS columns & calibration tables



MS columns & calibration tables



MS columns & calibration tables



(note: data size tripled)

MS data selection syntax

- You can select subset of visibilities to perform actions on:
  - Antennas, baselines, frequencies, time, polarization, etc.

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

For full syntax see CASAdocs

MS data selection syntax: Examples

- **field** (*spatial*)
  - String with source name or field ID (checks former 1st)
    - Beware field names that are integers
  - Examples: **field='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'

MS data selection syntax: Examples

- **timerange** (*spatial*)
  - String with date/time range in format T0~T1
  - Can give T0+dT, where missing parts of T1 default to T0
  - Example: **timerange = '2014/10/21/01:00:00~06:30:00'**

- **antenna**
  - String with antenna name or ID (checks former 1st)
    - Beware VLA name 1-27, these have ID's 0-26
  - & = CC only , && = CC+AC , &&& = AC only
  - Examples: **antenna = '1~5,8'** ; **antenna='!ea01&ea10'** ; **antenna='ea05&&&'**

MS data selection syntax: Examples

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

Ongoing CASA Developments

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

Need a person to help?

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

CASA documentation and web resources

http://casa.nrao.edu

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