CASA

Nemesio Rodriguez-Fernandez (IRAM)

Outline

- **Status and Measurement Set**
- **Structure**
- **Tools**
- **Tasks**
- **Scripts**
- **CASA and GILDAS**
- **Documentation**

CASA status

- **CASA = *Common Astronomy Software Applications***
- **Data reduction package for ALMA and EVLA**
- **CASA is under active development (by NRAO with collaboration of ESO, NAOJ, ASTRON, ATNF).**
  - **Current version is 3.2**
- **Supported platforms (binaries): RHEL, Fedora, SuSE, Ubuntu, Mac OS X 10.5 (32bit proc) and 10.6 (64 bits proc)**
- **Has the intention to be a general software package to reduce both interferometer and single dish data**
  - **Measurement Equation formalism (Hamaker et al. 1996, Hamaker 2000, CASA Cookbook Appendix D)**
  - **CASA data format : Measurement Set (MS). Wieringa & Cornwell 1996, Kemball & Wierenga 2000**

CASA Measurement Sets, ASDMs, and uvfits

The MS

- relational database system with fixed structure made from *CASA Tables*

- consists of a main *table* with 15 required *sub-tables* + several optional ones

- uses OS directory structure (need to copy with `cp -R`, remove with `rm -r`)

- visibilities stored in the MAIN table

- no compression

- manipulate an MS with the `ms` and the `tb` tool or with `browsetable()`

- during processing, CASA may add "scratch columns" to the MS main table

CASA design and implementation

Overall architecture:

1) A data structure

Tables: Images, Caltables, and the *Measurement Set* (MS)

2) A set of data import/export facilities

the so-called *fillers*: (ASDM, UVFITS, FITS-IDI, VLA archive) → MS, FITS → Image

3) A set of tools for data access, display, and editing

tools to load/write data into/from casacore data types,

Qt-based table browser, viewer, and (beta) x/y plotter, *matplotlib*-based x/y plotter

4) A set of tools for science analysis

built around the *Measurement Equation* (Hamaker et al. 1996),

a toolkit for radio astronomical calibration, imaging, and simulation

5) A set of high-level analysis procedures ("tasks")

user-friendly implementations of the solutions for all common analysis problems

6) A programmable command line interface with scripting

Python (augmented by *IPython*) gives a MATLAB-like interactive language

7) Documentation

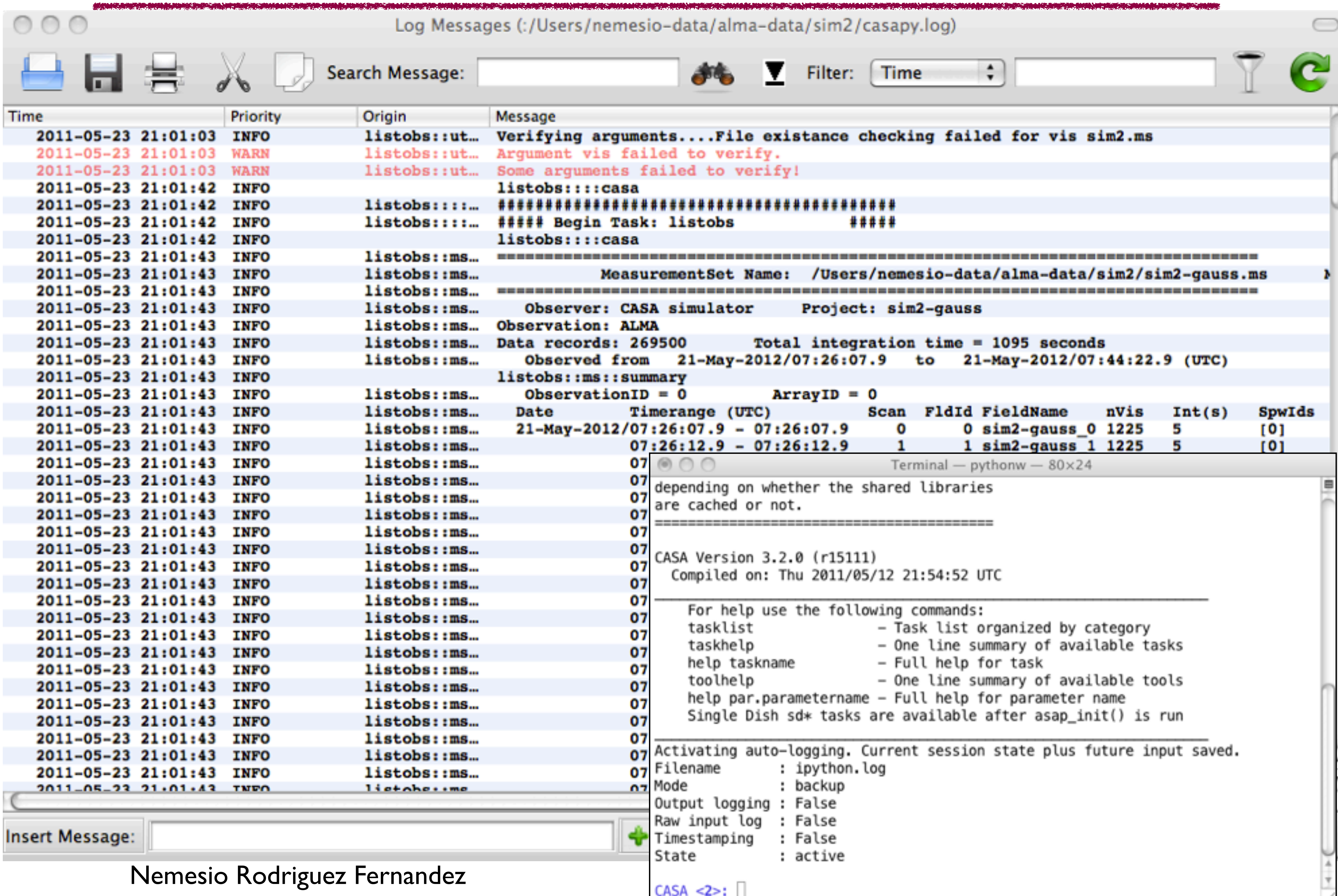
an extensive cookbook (500 pages) + documentation through help commands

(help, ?, pdoc) + online help pages, See **http://casa.nrao.edu/**

CASAPY

- To start a CASA session type: `casapy`
- To end a CASA session type: `Exit, Quit or Ctrl+D`

CASA prompt & logger



Nemesio Rodriguez Fernandez

martes 24 de mayo de 2011

CASAPY

- To start a CASA session type: `casapy`
- To end a CASA session type: `Exit, Quit or Ctrl+D`
- CASA uses IPython, which is an enhanced, interactive shell to Python which provides many features for efficient command line interaction
  - input/output history, avoid typing parentheses to call the tasks,...
- Shell access in interactive mode : `! cp file1 file2`
  - Commands like `ls, pwd, less, rm`... do not need the "!"
- Systems commands in scripts
  - `import os  + os.system('command')`
  - `Some exceptions as os.chdir()`

CASA stand-alone applications

- Other applications that can be run directly from your prompt are:

`casapyinfo`	.....	returns info about how CASA was built
`casabrowser`	.....	== browsetable() task within casapy
`casalogger`	.....	the logger started by default with casapy
`casaplotms`	.....	will be == plotms() task within casapy
`casaviewer`	.....	== viewer() task within casapy
`asdm2MS`	.....	the ASDM to MS converter, importasdm in casapy
`buildmytasks`	.....	integrates user-provided tasks into casapy

CASA GUIs: casabrowser/browsetable

Table Browser

otf.ms

table data | table keywords | field keywords

	UVW	FLAG	FLAG_CATEGORY	WEIGHT	SIGMA	ANTENNA1	ANTENNA2	ARRAY_ID	DATA_DESC_ID	EXPOSURE	FEED1	FEED2	FIELD_ID
0	[0, 0, 0]	[2, 3840] B...	[0, 0, 0] Bo...	[1, 1]	[1, 1]	0	0	0	0	1.152	0	0	0
1	[0, 0, 0]	[2, 3840] B...	[0, 0, 0] Bo...	[1, 1]	[1, 1]	1	1	0	0	1.152	0	0	0
2	[0, 0, 0]	[2, 3840] B...	[0, 0, 0] Bo...	[1, 1]	[1, 1]	2	2	0	0	1.152	0	0	0
3	[0, 0, 0]	[2, 3840] B...	[0, 0, 0] Bo...	[1, 1]	[1, 1]	3	3	0	0	1.152	0	0	0
4	[0, 0, 0]	[2, 3840] B...	[0, 0, 0] Bo...	[1, 1]	[1, 1]	0	0	0	0	1.152	0	0	0
5	[0, 0, 0]	[2, 3840] B...	[0, 0, 0] Bo...	[1, 1]	[1, 1]	1	1	0	0	1.152	0	0	0
6	[0, 0, 0]	[2, 3840] B...	[0, 0, 0] Bo...	[1, 1]	[1, 1]	2	2	0	0	1.152	0	0	0
7	[0, 0, 0]	[2, 3840] B...	[0, 0, 0] Bo...	[1, 1]	[1, 1]	3	3	0	0	1.152	0	0	0
8	[0, 0, 0]	[2, 3840] B...	[0, 0, 0] Bo...	[1, 1]	[1, 1]	0	0	0	0	1.152	0	0	0
9	[0, 0, 0]	[2, 3840] B...	[0, 0, 0] Bo...	[1, 1]	[1, 1]	1	1	0	0	1.152	0	0	0
10	[0, 0, 0]	[2, 3840] B...	[0, 0, 0] Bo...	[1, 1]	[1, 1]	2	2	0	0	1.152	0	0	0
11	[0, 0, 0]	[2, 3840] B...	[0, 0, 0] Bo...	[1, 1]	[1, 1]	3	3	0	0	1.152	0	0	0
12	[0, 0, 0]	[2, 3840] B...	[0, 0, 0] Bo...	[1, 1]	[1, 1]	0	0	0	0	1.152	0	0	0

Restore Columns | Resize Headers

PAGE NAVIGATION First << [1 / 61] >> Last 1 Go Loading 1000 rows.

Browsing table: otf.ms

The *CASA* user interface

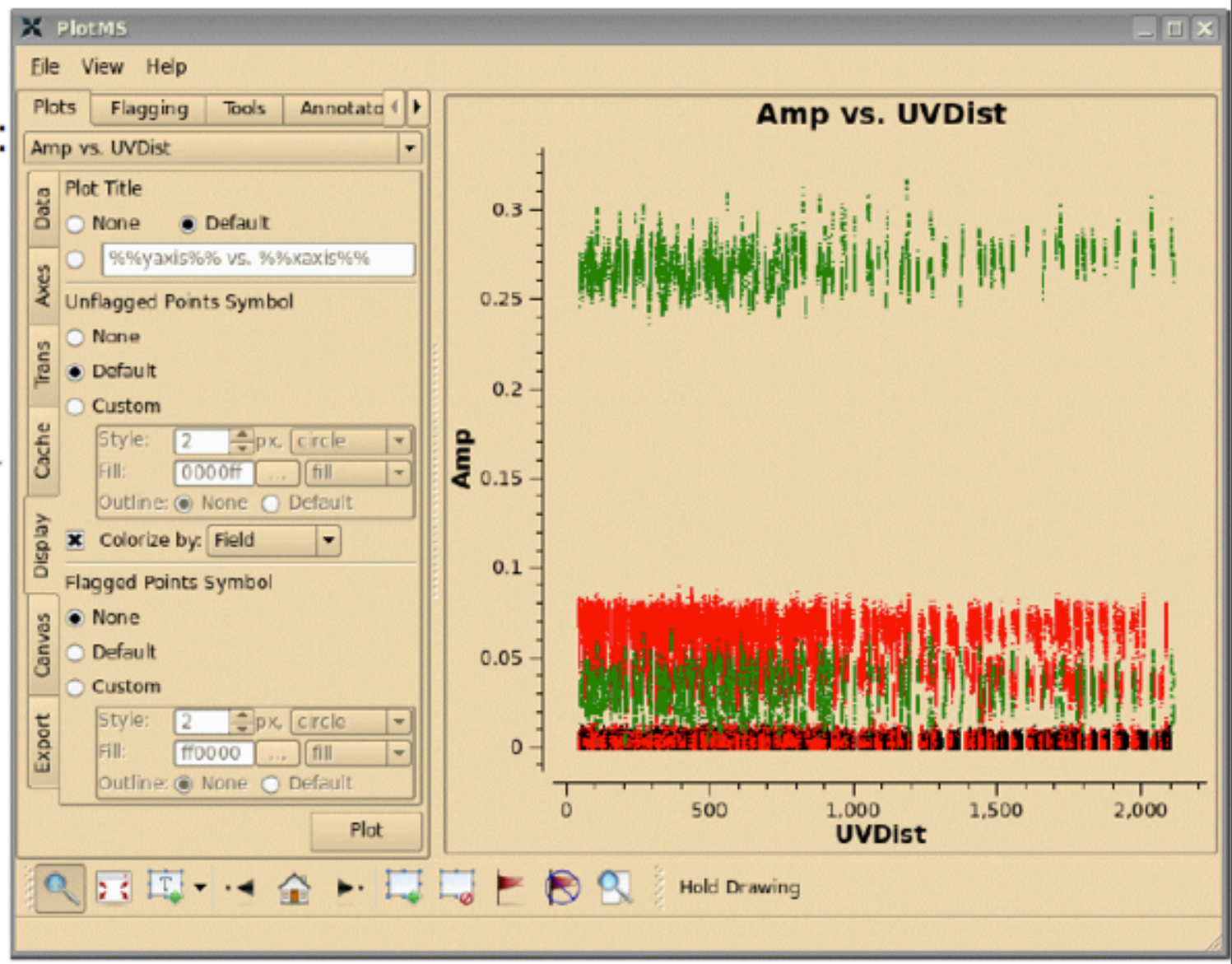
A typical analysis session

3) where needed, tools have GUIs:

plotxy, plotcal, browsetable, viewer, clean, plotms

(started in separate threads)

plotms is going to replace plotxy. Release 3.1 contains (advanced) beta version.



CASA Tools

Terminal — pythonw — 83×28

```
CASA <46>: toolhelp
---------> toolhelp()

Available tools:

 at : Juan Pardo ATM library
 cb : Calibration utilities
 cp : Cal solution plotting utilities
 fg : Flagging/Flag management utilities
 ia : Image analysis utilities
 im : Imaging utilities
 me : Measures utilities
 ms : MeasurementSet (MS) utilties
 mp : MS plotting (data (amp/phase) versus other quantities)
 pm : PlotMS utilities
 tb : Table utilities (selection, extraction, etc)
 tp : Table plotting utilities
 qa : Quanta utilities
 sl : Spectral line import and search
 sm : Simulation utilities
 vp : Voltage pattern/primary beam utilties
 ---
 pl : pylab functions (e.g., pl.title, etc)
 sd : (after running asap_init()) Single dish utilities
 ---

CASA <47>:
```

CASA Tools: help(toolname)

Terminal — less — 87×39

```
CASA <35>: help sm
---------> help(sm)
Help on simulator object:

class simulator(__builtin__.object)
 |  simulator object
 |
 |  Methods defined here:
 |
 |  __init__(...)
 |      x.__init__(...) initializes x; see x.__class__.__doc__ for signature
 |
 |  __str__(...)
 |      x.__str__() <==> str(x)
 |
 |  close(...)
 |      Close the newsimulator tool
 |      ----------------------------------------
 |
 |  corrupt(...)
 |      Corrupt the data with visibility errors
 |      ----------------------------------------
 |
 |  done(...)
 |      Close the newsimulator tool
 |      ----------------------------------------
 |
 |  name(...)
 |      Provide the name of the attached MeasurementSet
 |      ----------------------------------------
 |
 |  observe(...)
 |      Observe a given configuration
 |      --- --- --- --- --- --- Parameters  --- --- --- --- --- ---
 |        sourcename:  Name of source or field (must be specified) None
 |        spwname:  Unique user-supplied name for this spectral window None
```

CASA Tools: help(toolname.method)

Terminal — less — 86×39

```
CASA <38>:

CASA <39>: help sm.setconfig
---------> help(sm.setconfig)
Help on built-in function setconfig:

setconfig(...)
    Set the antenna configuration
    --- --- --- --- --- --- Parameters  --- --- --- --- --- ---
      telescopename:  Name of the telescope we are simulating (determines VP) VLA 'VLA
'
      x:  Vector of x values of all antennas [currently m] 0 []
      y:  Vector of y values of all antennas [currently m] 0 []
      z:  Vector of z values of all antennas [currently m] 0 []
      dishdiameter:  Vector of diameters of all antennas [currently m] 0 []
      offset:  Vector of offset of all antennas [currently m] 0 []
      mount:  Vector of mount types of all antennas (recognized mounts are 'ALT-AZ', '
EQUATORIAL', 'X-Y', 'ORBITING', 'BIZARRE' ALT-AZ []
      antname:  Vector of names of all antennas A []
      padname:  Vector of names of pads or stations P []
      coordsystem:  Coordinate system of antenna positions [x,y,z], possibilities are
'global', 'local' , 'longlat' global 'global'
      referencelocation:  Reference location [required for local coords] Position Meas
ure of Coordinates of array location. E.g me.position('ITRF', '30.5deg', -20.2deg', 60
00km') or me.observatory('ALMA') ALMA position measure
    --- --- --- --- --- --- --- --- --- --- --- --- --- --- ---
      telescopename     = VLA
      x = [ 0 ]
      y = [ 0 ]
      z = [ 0 ]
      dishdiameter      = [ 0 ]
      offset    = [ 0 ]
      mount     = [ ALT-AZ ]
      antname   = [ A ]
      padname   = [ P ]
      coordsystem       = global
      referencelocation
    ----------------------------------------
(END)
```

CASA tools in use

- not optimised for interactive use, behave just like Python objects

⇒ user calls methods of the tools:

<toolname>.<methodname>(<parameters>)

e.g., `ms.open('mydata.ms')` - open an MS read-only with the MS tool

```
# Other wise for more adventurous the toolkit ..this is how i ran for the one AL
im.open('serpens_otf.ms')
im.selectvis(field='2', spw='0')
im.defineimage(nx=1800, cellx='0.5arcsec', phasecenter=2 , spw=0, mode='mfs')
#, step=20, start=2450, nchan=25)
im.setoptions(ftmachine='mosaic')
im.setmfcontrol(fluxscale=imname+'.flux')
im.make(imname+'.model')
im.clean(algorithm='mfhogbom', niter=1000, model=imname+'.model',
image=imname+'.image', residual=imname+'.residual',
psfimage=imname+'.psf', interactive=False)
```

CASAPY : tasks

- Common analysis functionality for standard users. `Python scripts using tools`

Example: the task *flagautocorr(vis)* - *flag the rows with autocorrelation data in an MS*

```
import os
from taskinit import *
def flagautocorr(vis=None):
        casalog.origin('flagautocorr')
        try:
                fg.clearflagselection(0)
                if ((type(vis)==str) & (os.path.exists(vis))):
                        fg.open(vis)
                else:
                        raise Exception, 'Visibility data set not found'
                fg.setdata()
                fg.setmanualflags(autocorrelation=True)
                fg.run()
                fg.done()
                ms.open(vis,nomodify=False)
                ms.writehistory(message='flagautocorr',origin='flagautocorr')
                ms.close()
        except Exception, instance:
                fg.done()
                print '*** Error ***',instance
```

CASAPY : tasks

- Some useful commands
  - default(taskname)
  - inp(taskname)
  - go #executes active task
  - saveinputs(taskname, file)
  - tget(taskname, file)
- Help
  - tasklist
  - taskhelp
  - help taskname

CASA Tasks :

Terminal — Python — 76×39

```
CASA <4>: tasklist
--------> tasklist()
Available tasks, organized by category (experimental tasks in parenthesis):

Import/Export       Information   Data Editing  Display/Plotting
-------------       -----------   ------------  ----------------
importvla           imhead        concat        clearplot
importfits          imstat        fixvis        plotants
importuvfits        listcal       flagautocorr  plotcal
exportfits          listhistory   flagdata      plotms
exportuvfits        listobs       flagmanager   plotxy
(importasdm)        listvis       plotms        imview
(importgmrt)        vishead       plotxy        msview
                    visstat

Data Manipulation   Calibration   Imaging       Modelling
-----------------   -----------   -------       ---------
concat              accum         clean         setjy
cvel                applycal      deconvolve    uvcontsub
fixvis              bandpass      feather       uvmodelfit
hanningsmooth       blcal         ft            uvsub
split               calstat       csvclean      (uvcontsub2)
uvcontsub           clearcal      widefield
uvsub               cvel          (boxit )
(uvcontsub2)        fluxscale     (autoclean)
(msmoments)         fixvis
                    gaincal
                    gencal
                    listcal
                    polcal
                    setjy
                    smoothcal
                    (fringecal)
                    (peel)
```

CASA Tasks :

Terminal — Python — 76×39

```
CASA <4>: tasklist
--------> tasklist()
Available tasks, organized by category (experimental tasks in parenthesis):

Import/Export       Information   Data Editing  Display/Plotting
-------------       -----------   ------------  ----------------
importvla           imhead        concat        clearplot
importfits          imstat        fixvis        plotants
```

Terminal — Python — 76×29

```
Image Analysis  Simulation  Utilities           Single Dish
--------------  ----------  ---------           (after running asap_init())
imcontsub       simdata     browsetable         ---------------------------
imhead          (simdata2)  casalogger          sdaverage
imfit                       clearplot           sdbaseline
immath                      clearstat           sdcal
immoments                   csvclean            sdcoadd
imregrid                    filecatalog         sdfit
imsmooth                    find                sdflag
imstat                      help par.parameter  sdflagmanager
imval                       help task           sdimaging
(specfit)                   rmtables            sdimprocess
                            startup             sdlist
                            taskhelp            sdmath
                            tasklist            sdplot
                            toolhelp            sdsave
                                                sdscale
                                                sdsmooth
                                                sdstat
                                                sdtpimaging
                                                (sdsim)
                                                (msmoments)

User defined tasks
------------------

CASA <5>:
```

CASA Tasks : help taskname

Terminal — less — 87×54

```
Help on simdata task:

mosaic simulation task:
mosaic simulation task:

    This task simulates interferometric observations (currently
    only ALMA can be done easily).  New functionality is actively
    being added, so if you have changed versions of CASA, check
    the inputs carefully.
    Please contact CASA experts with any questions, especially
    about features noted below as *experimental*
    -----------------------------
    project -- root filename for all output files.
    -----------------------------
    modifymodel -- change the coordinate system of the model sky image?
       * if graphics selected, display the rescaled model image
    skymodel -- if modifymodel=False, use this as the sky model.
       * if modifyimage=True, use this as the starting point, modify it
         write the output to a different image (default $project.skymodel)
         and use that new image as the sky model
    inbright -- peak brightness in Jy/pixel, or "" for unchanged
       * NOTE: "unchanged" will take the numerical values in your image
         and assume they are in Jy/pixel, even if it says some other unit
         in the header.
    indirection -- central direction, or "" for unchanged
    incell -- spatial pixel size, or "" for unchanged
    incenter -- frequency of center channel e.g. "89GHz", or "" for unchanged
    inwidth -- width of channels, or "" for unchanged - this should be a
         string representing a quantity with units e.g. "10MHz"
       * NOTE: only works reliably with frequencies, not velocities
```

CASA Tasks : inp(taskname)

Terminal — pythonw — 87×54

```
NameError: name 'asdf' is not defined

CASA <45>: inp(simdata)
#  simdata :: mosaic simulation task:
project             = 'sim2-gauss'      #  root prefix for output file names
modifymodel         =       True        #  modify model image
     skymodel       = 'test-gauss.fits' #  model image to observe or modify
     inbright       =         ''        #  scale surface brightness of brightest pixel
                                        #   e.g. "1.2Jy/pixel" or ""
     indirection    = 'J2000 19h00m00 -40d00m00' #  "J2000 19h00m00 -40d00m00" or ""
     incell         =         ''        #  cell/pixel size e.g. "0.1arcsec" or ""
     incenter       =         ''        #  frequency of center channel e.g. "89GHz" or
                                        #   ""
     inwidth        =         ''        #  channel width e.g. "10MHz" or ""

setpointings        =       True
     integration    =        '5s'       #  integration (sampling) time
     direction      = 'J2000 19h00m00 -40d00m00' #  "J2000 19h00m00 -40d00m00" or "" to
                                        #   center on model
     mapsize        = ['2arcmin', '1arcmin'] #  angular size of map or "" to cover
                                        #   model
     maptype        =   'square'        #  hexagonal, square, etc
     pointingspacing =   '0.25PB'       #  spacing in between pointings or "0.25PB" or
                                        #   "" for 0.5 PB
```

CASA Tasks : inp(taskname)

```
predict                 =        True       #  calculate visibilites using ptgfile
     complist           =          ''       #  optional componentlist to observe with
                                            #   skymodel
     compwidth          =      '2GHz'       #  optional bandwidth if simulating from
                                            #   components only
     antennalist        = './array-config/alma.out01.cfg' #  antenna position file or ""
                                            #   for no interferometric MS
     refdate            = '2012/05/21/06:05:00' #  time/date of observation *see help
     totaltime          =    '1500s'        #  total time of observation
     caldirection       =          ''       #  pt source calibrator [experimental]
     calflux            =      '1Jy'
     sdantlist          =          ''       #  single dish antenna position file or "" for
                                            #   no total power MS
     sdant              =           0       #  single dish antenna index in file

thermalnoise            =          ''       #  add thermal noise: [tsys-atm|tsys-manual|""]
leakage                 =         0.0       #  cross polarization
image                   =       False       #  (re)image $project.ms to $project.image
analyze                 =       False       #  (only first 6 selected outputs will be
                                            #   displayed)
graphics                =    'screen'       #  display graphics at each stage to
                                            #   [screen|file|both|none]
verbose                 =       False
overwrite               =        True       #  overwrite files starting with $project
async                   =       False       #  If true the taskname must be started using
                                            #   simdata(...)
```

CASAPY scripts : simulating data

```
default(simdata)
inp(simdata)
#
project         = 'sim2-gauss'
modifymodel     = T
skymodel        = 'gauss-source.fits'
indirection     = "J2000 19h00m00 -40d00m00"
setpointings    = True
integration     = '5s'
direction       = "J2000 19h00m00 -40d00m00"
mapsize         = ['2arcmin','1arcmin']
maptype         = "square"
pointingspacing = "0.25PB"    # 12"
predict         = True
antennalist     = './array-config/alma.out01.cfg'  #resol approx 5"
refdate         = '2012/05/21/06:05:00'  # Elevation of 50 deg approx
totaltime       = "1500s"
image           = F
analyze         = F
showarray       = F
verbose         = F
overwrite       = T
#
inp(simdata)
simdata()
```

To execute:

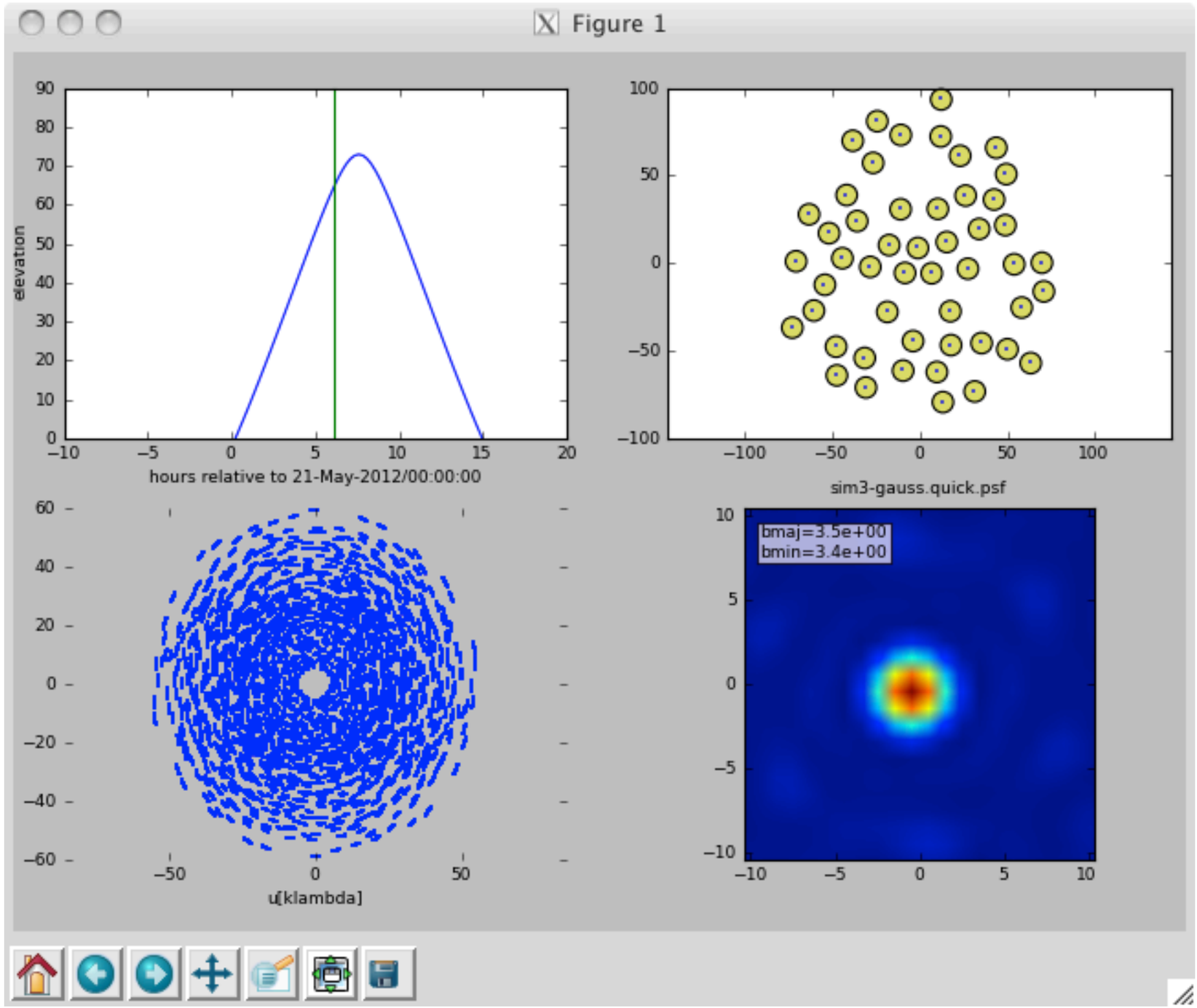
```
execfile('myscript.py')
```

Available in CASA 3.2 sources or in casaguides wiki (see last slide)

do not do the imaging

do not compare the result wrt model

Plot of simulator setup



CASAPY scripts : imaging

```
#
# execfile('test.image.py')
#
default(clean)
niter = 0
vis = 'sim2-gauss.ms'
imagename = 'sim2'  #comments
cell = ['1arcsec']
imsize = [1024,1024]
inp(clean)
clean()

viewer('sim2.image')
```

To produce only Dirty Image

The *CASA* user interface

Pictures from a typical analysis session

3) where needed, tools have GUIs:

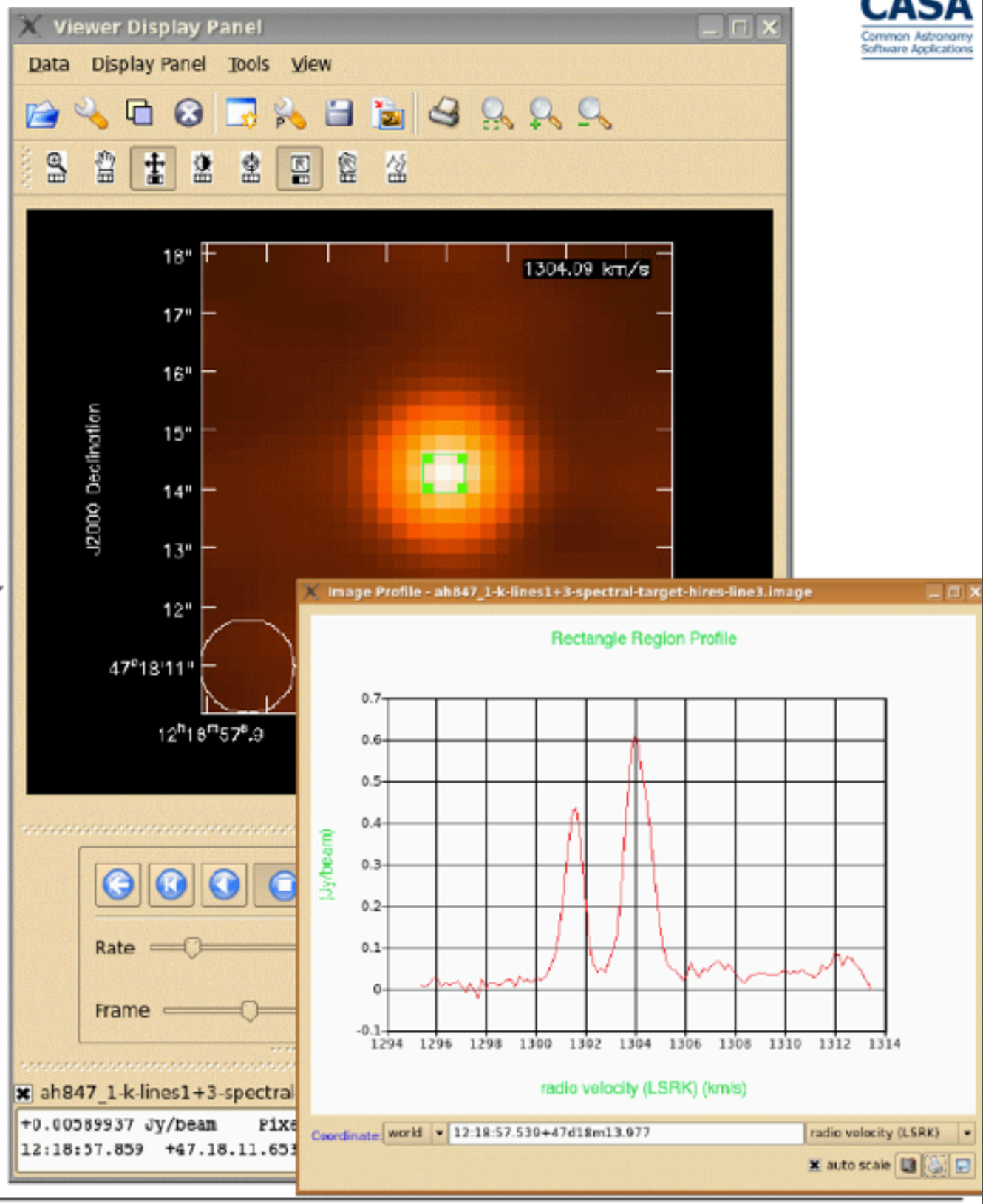
plotxy, plotcal, browsetable,
viewer, clean, plotms

(started in separate threads)

The ***viewer*** is a powerful multi-function tool for data selection and visualization.

Uses Qt widget set
(but 80% independent)

Rendering based on pgplot



CASA add-ons

- A plugin system to provide add-ons will be developed by NRAO
- There are two projects at IRAM dealing with CASA
  - TelCal-CASA interface: JC Roche, D Brogière
  - WideField Synthesis (On-the-Fly): JC Roche, M Lonjaret, N Rodriguez-Fernandez, F Gueth, J Pety

CASA and GILDAS

- Calibration
  - Data calibration should be done with the software package dedicated to the instrument (CASA for ALMA data, GILDAS for PdB data)
  - Raw data: “filler” exists … but it is not user-friendly to be used by standard users
- Imaging/simulations
  - Images / FITS / Simulations, model sources to aid deconvolution …
    - CASA : exportfits, importfits
    - GILDAS: gildas_fits, fits_gildas
    - CASA complains with some FITS keywords written by GILDAS
  - Visibility tables / UVFITS /
    - GILDAS: gildas_fits (format `aipsfits`)

Documentation

- CASA cookbook: http://casa.nrao.edu/Doc/Cookbook/casa_cookbook.pdf
- Task reference web page http://casa.nrao.edu/docs/taskref/TaskRef.html
- Toolkit manual web page http://casa.nrao.edu/docs/casaref/CasaRef.html
- CASAGuides wiki http://casaguides.nrao.edu/
  - Getting_Started_in_CASA
  - Simulations
- CASA tutorials (data reduction scripts)
  - http://www.alma.inaf.it/ (Meetings & Talks)
  - http://www.astro.uni-bonn.de/ARC/casascripts.shtml
  - http://www.alma.ac.uk/events/support/casa/using-casa