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
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

---

D. Petry, Bologna CASA Workshop, April 2010
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-ID, 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/

D. Petry, ALMA Community Days, ESO, April 2011
To start a CASA session type: casapy

To end a CASA session type: Exit, Quit or Ctrl+D
CASAS prompt & logger
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` .......................... `== 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

![CASA GUIs: casabrowser/browsetable](image)

#### Table Browser

<table>
<thead>
<tr>
<th>UVW</th>
<th>FLAG</th>
<th>FLAG_CATEGORY</th>
<th>WEIGHT</th>
<th>SIGMA</th>
<th>ANTENNA1</th>
<th>ANTENNA2</th>
<th>ARRAY_ID</th>
<th>DATA_DESC_ID</th>
<th>EXPOSURE</th>
<th>FEED1</th>
<th>FEED2</th>
<th>FIELD_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>[0, 0, 0]</td>
<td>[2, 3840] B...</td>
<td>[0, 0, 0] Bo...</td>
<td>[1, 1]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.152</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>[0, 0, 0]</td>
<td>[2, 3840] B...</td>
<td>[0, 0, 0] Bo...</td>
<td>[1, 1]</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1.152</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>[0, 0, 0]</td>
<td>[2, 3840] B...</td>
<td>[0, 0, 0] Bo...</td>
<td>[1, 1]</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1.152</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>[0, 0, 0]</td>
<td>[2, 3840] B...</td>
<td>[0, 0, 0] Bo...</td>
<td>[1, 1]</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1.152</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>[0, 0, 0]</td>
<td>[2, 3840] B...</td>
<td>[0, 0, 0] Bo...</td>
<td>[1, 1]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.152</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>[0, 0, 0]</td>
<td>[2, 3840] B...</td>
<td>[0, 0, 0] Bo...</td>
<td>[1, 1]</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1.152</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>[0, 0, 0]</td>
<td>[2, 3840] B...</td>
<td>[0, 0, 0] Bo...</td>
<td>[1, 1]</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1.152</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>[0, 0, 0]</td>
<td>[2, 3840] B...</td>
<td>[0, 0, 0] Bo...</td>
<td>[1, 1]</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1.152</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>[0, 0, 0]</td>
<td>[2, 3840] B...</td>
<td>[0, 0, 0] Bo...</td>
<td>[1, 1]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.152</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>[0, 0, 0]</td>
<td>[2, 3840] B...</td>
<td>[0, 0, 0] Bo...</td>
<td>[1, 1]</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1.152</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>[0, 0, 0]</td>
<td>[2, 3840] B...</td>
<td>[0, 0, 0] Bo...</td>
<td>[1, 1]</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1.152</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>[0, 0, 0]</td>
<td>[2, 3840] B...</td>
<td>[0, 0, 0] Bo...</td>
<td>[1, 1]</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1.152</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>[0, 0, 0]</td>
<td>[2, 3840] B...</td>
<td>[0, 0, 0] Bo...</td>
<td>[1, 1]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.152</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

---

**Nemesio Rodriguez Fernandez**  
**9**  
**ALMA ES Proposal prep. May 2011**  

**martes 24 de mayo de 2011**
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 <46>: 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) utilities
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 utilities

pl : pylab functions (e.g., pl.title, etc)
sd : (after running asap_init()) Single dish utilities
CASA Tools: help(toolname)

```
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)

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', 'ORBITE', '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 'local', 'longlat' global 'global'
  referencelocation: Reference location [required for local coords] Position Measure of Coordinates of array location. E.g me.position('ITRF', '30.5deg', -20.2deg', 60 k00km') 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
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

# Otherwise for more adventurous the toolkit .. this is how I ran for the one ALMA
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.setmcontrol(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

```python
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:

<table>
<thead>
<tr>
<th>Import/Export</th>
<th>Information</th>
<th>Data Editing</th>
<th>Display/Plotting</th>
</tr>
</thead>
<tbody>
<tr>
<td>importvla</td>
<td>imhead</td>
<td>concat</td>
<td>clearplot</td>
</tr>
<tr>
<td>importfits</td>
<td>imstat</td>
<td>fixvis</td>
<td>plotsants</td>
</tr>
<tr>
<td>importuvfits</td>
<td>listcal</td>
<td>flagautocorr</td>
<td>plotants</td>
</tr>
<tr>
<td>exportfits</td>
<td>listhistory</td>
<td>flagdata</td>
<td>plotcal</td>
</tr>
<tr>
<td>exportuvfits</td>
<td>listobs</td>
<td>flagmanager</td>
<td>plotms</td>
</tr>
<tr>
<td>(importasdm)</td>
<td>listvis</td>
<td>plotxy</td>
<td>imview</td>
</tr>
<tr>
<td>(importgmrt)</td>
<td>vishead</td>
<td></td>
<td>msview</td>
</tr>
<tr>
<td></td>
<td>visstat</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Manipulation</td>
<td>Calibration</td>
<td>Imaging</td>
<td>Modelling</td>
</tr>
<tr>
<td>concat</td>
<td>accum</td>
<td>clean</td>
<td>setjy</td>
</tr>
<tr>
<td>cvel</td>
<td>applycal</td>
<td>deconvolve</td>
<td>uvcontsub</td>
</tr>
<tr>
<td>fixvis</td>
<td>bandpass</td>
<td>feather</td>
<td>uvcontsub2</td>
</tr>
<tr>
<td>hanningsmooth</td>
<td>blcal</td>
<td>ft</td>
<td>(uvcontsub2)</td>
</tr>
<tr>
<td>split</td>
<td>calstat</td>
<td>csvclean</td>
<td></td>
</tr>
<tr>
<td>uvcontsub</td>
<td>clearcal</td>
<td>widefield</td>
<td>(boxit)</td>
</tr>
<tr>
<td>uvsub</td>
<td>cvel</td>
<td></td>
<td>(autoclean)</td>
</tr>
<tr>
<td>(uvcontsub2)</td>
<td>fluxscale</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(msmoments)</td>
<td>fixvis</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>gaincal</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>gencal</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>listcal</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>polcal</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>setjy</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>smoothcal</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(fringecal)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(peel)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CASA Tasks:

```
CASA <4>: tasklist
------------
tasklist()
```

Available tasks, organized by category (experimental tasks in parenthesis):

<table>
<thead>
<tr>
<th>Category</th>
<th>Import/Export</th>
<th>Information</th>
<th>Data Editing</th>
<th>Display/Plotting</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>importvla</td>
<td>imhead</td>
<td>concat</td>
<td>clearplot</td>
</tr>
<tr>
<td></td>
<td>importfits</td>
<td>imstat</td>
<td>figvis</td>
<td>plotants</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Category</th>
<th>Image Analysis</th>
<th>Simulation</th>
<th>Utilities</th>
<th>Single Dish</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>incontsub</td>
<td>simdata</td>
<td>browsetable</td>
<td>(after running asap_init())</td>
</tr>
<tr>
<td></td>
<td>imhead</td>
<td>(simdata2)</td>
<td>casalogger</td>
<td></td>
</tr>
<tr>
<td></td>
<td>imfit</td>
<td></td>
<td>clearplot</td>
<td></td>
</tr>
<tr>
<td></td>
<td>immath</td>
<td></td>
<td>clearstat</td>
<td></td>
</tr>
<tr>
<td></td>
<td>immoments</td>
<td></td>
<td>csvclean</td>
<td></td>
</tr>
<tr>
<td></td>
<td>imregrid</td>
<td></td>
<td>filecatalog</td>
<td></td>
</tr>
<tr>
<td></td>
<td>imsmooth</td>
<td></td>
<td>find</td>
<td></td>
</tr>
<tr>
<td></td>
<td>imstat</td>
<td></td>
<td>help par.parameter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>imval</td>
<td></td>
<td>help task</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(specfit)</td>
<td></td>
<td>rmtables</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>startup</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>taskhelp</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>tasklist</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>toolhelp</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

User defined tasks

```
CASA <5>: 
```
CASA Tasks: help taskname

```
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
```

martes 24 de mayo de 2011
CASA Tasks: `inp(taskname)`

```
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 ""
inell = '' # 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 (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
pointingsspacing = '0.25PB' # spacing in between pointings or "0.25PB" or
# "" for 0.5 PB
```
CASA Tasks: inp(taskname)

```
predict = True  # calculate visibilities using ptgfile
complist = ''  # optional componentlist to observe with
compwidth = '2GHz'  # optional bandwidth if simulating from
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'  # single dish antenna position file or """ for
sdantlist = ''  # 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
# only first 6 selected outputs will be displayed
analyze = False  # display graphics at each stage to
# [screen|file|both|none]
graphics = 'screen'  # overwrite files starting with $project
# If true the taskname must be started using
# simdata(...)"
To execute:

```python
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

```python
# 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

D. Petry, ALMA Community Days, ESO, April 2011
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
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