Introduction to CASA

Martin Zwaan
ALMA Regional Centre, ESO Garching

What is *CASA*

* **C**ommon **A**stronomy **S**oftware **A**pplications

* developed by a collaboration led by NRAO

* primary goal: *"supporting the data post-processing needs of the next generation of radio astronomical telescopes such as the ALMA and EVLA projects"*

* suite of C++ application libraries packaged through a Python scripting interface

* fully scriptable

CASA beta

* This really is a Beta Release!

  * still missing functionality

  * there are known failure modes

  * you will find new problems (use HelpDesk)

  * improvements in upcoming patches

  * move towards handling full ALMA/EVLA "use cases"

  * better handling of calibration

CASA beta

* The current release is 2.2

* Beta release patch 3 target release date: October 15

* Open to the public through **my.nrao.edu**

CASA on the Web

* http://casa.nrao.edu -- CASA Home Page

    * Main resource for end users

* http://my.nrao.edu -- NRAO Services (incl. registration/download of CASA)

    * Help Desk/Installation Front; manned by scientists (ALMA, ARC, NAUG, etc) to handle front-line user support.

CASA Documentation

✳ CASA Analysis cookbook: http://casa.nrao.edu/Doc/ Cookbook/casa_cookbook.pdf

✳ CASA User Reference Manual: http://casa.nrao.edu/ docs/casaref/CasaRef.html

✳ Python: http://python.org/doc (e.g., see Tutorial for novices)

✳ IPython: http://ipython.scipy.org/moin/Documentation

✳ matplotlib: http://matplotlib.sourceforge.net/

CASA development

* Development team of 16 people (14 fte) spread over NA (11), Europe (2) and East Asia (3).

    * three of them are in the audience

* Scientific oversight: CASA project scientist + large group of scientists, ASAC, etc + CASA Advisory Group
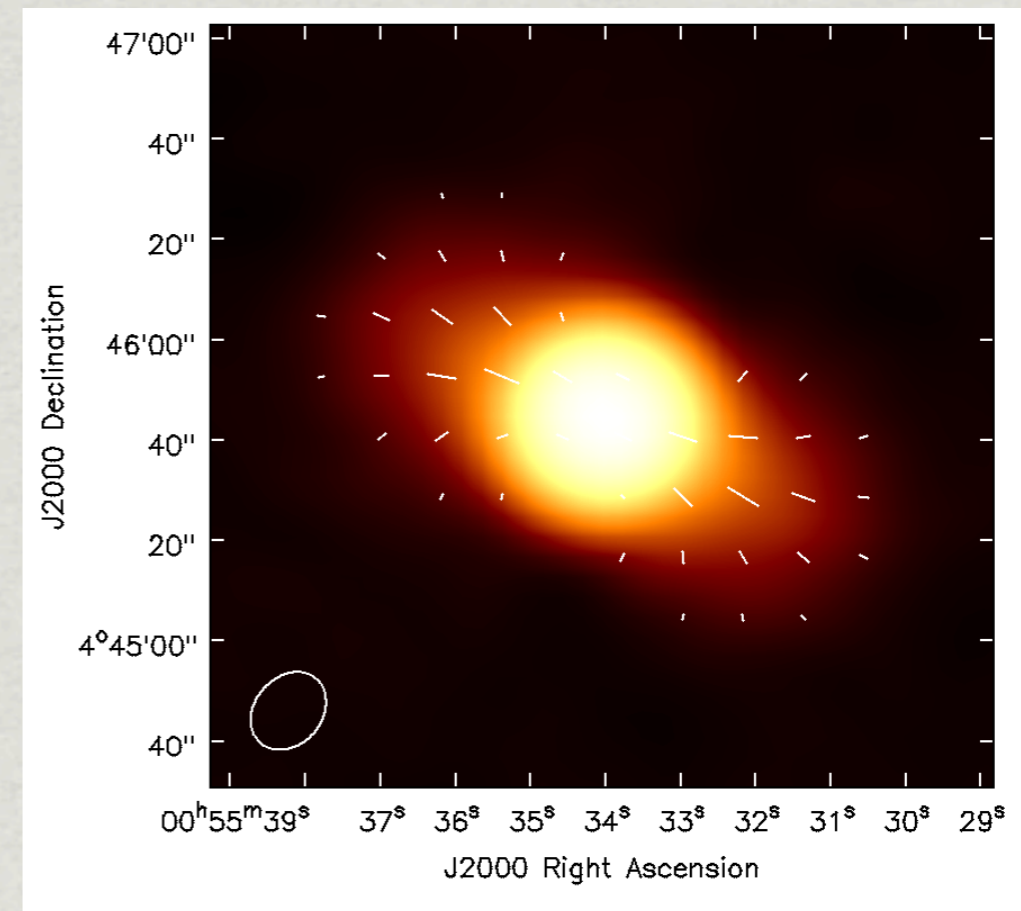
CASA training

* There have been training sessions in the past for people involved in ALMA/EVLA

* Are you interested in a CASA training session in Europe? Contact your ARC!
  (Send me email: `mzwaan@eso.org`)

Some CASA results
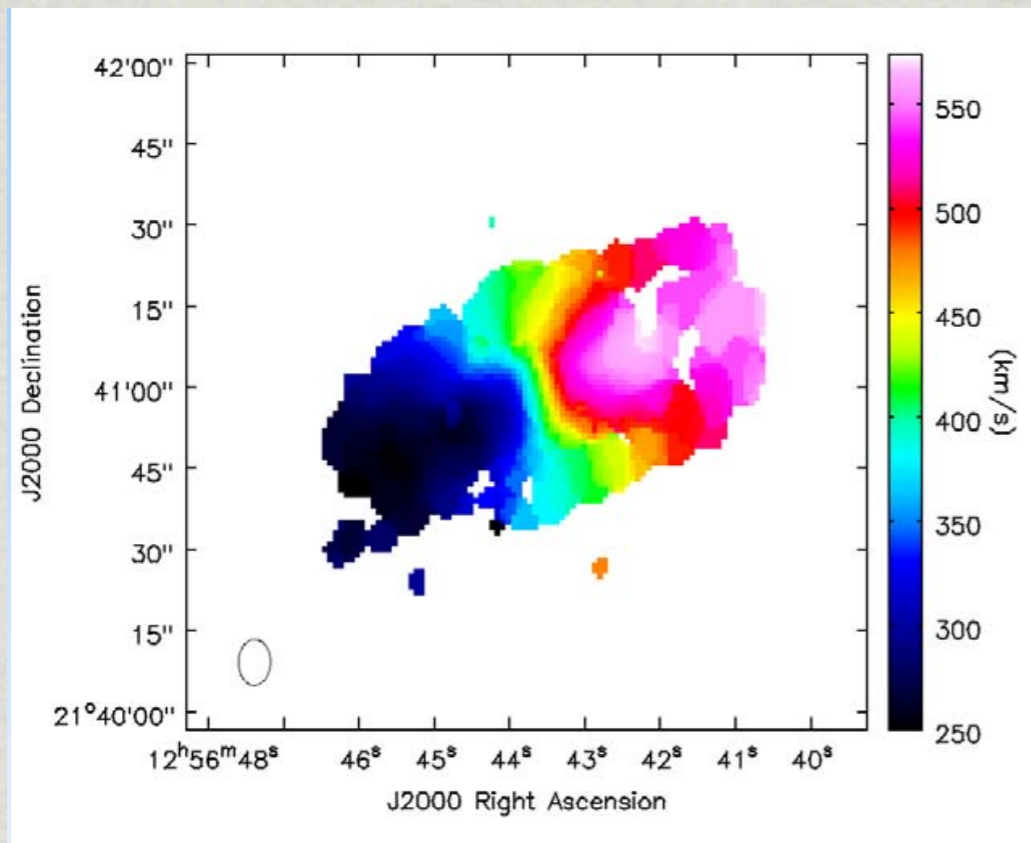
CO(1—0) kinematics (moment 1) of the galaxy NGC 4826 from the BIMA SONG survey (data originally published in Helfer, Thornley, Regan et al. 2003)
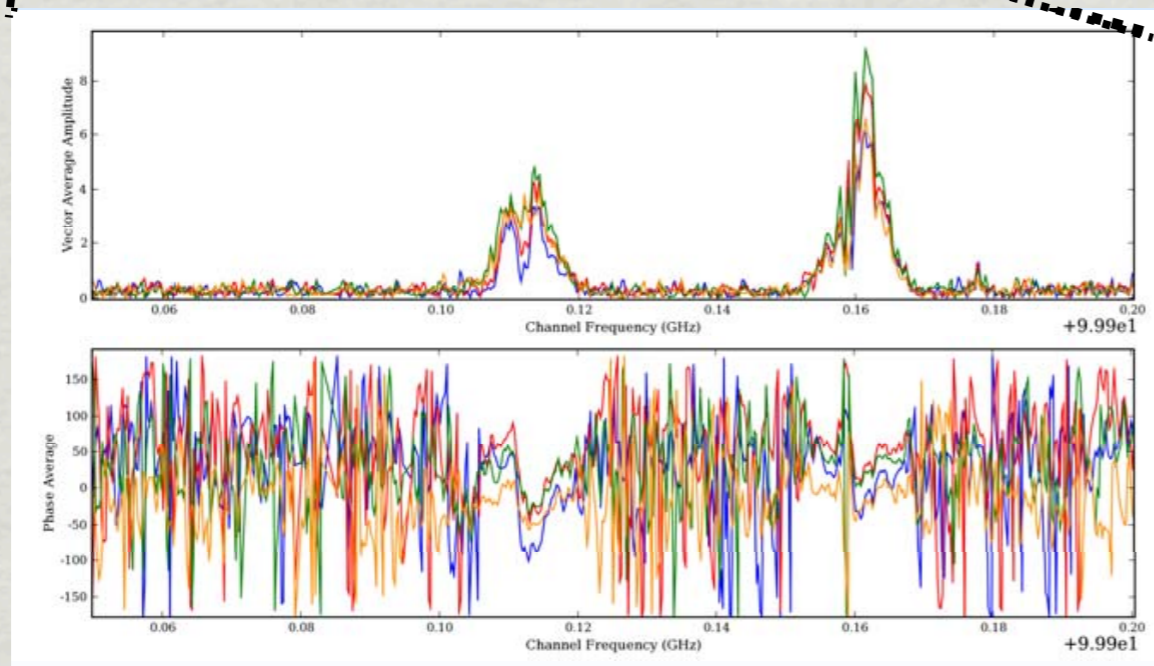
E-field vectors in Jupiter magnetosphere. Archival VLA 6 cm D-configuration full Stokes polarization data.

Data Calibrated and Imaged in CASA Tutorials at NRAO Synthesis Imaging Workshop June 2008

ALMA simulations
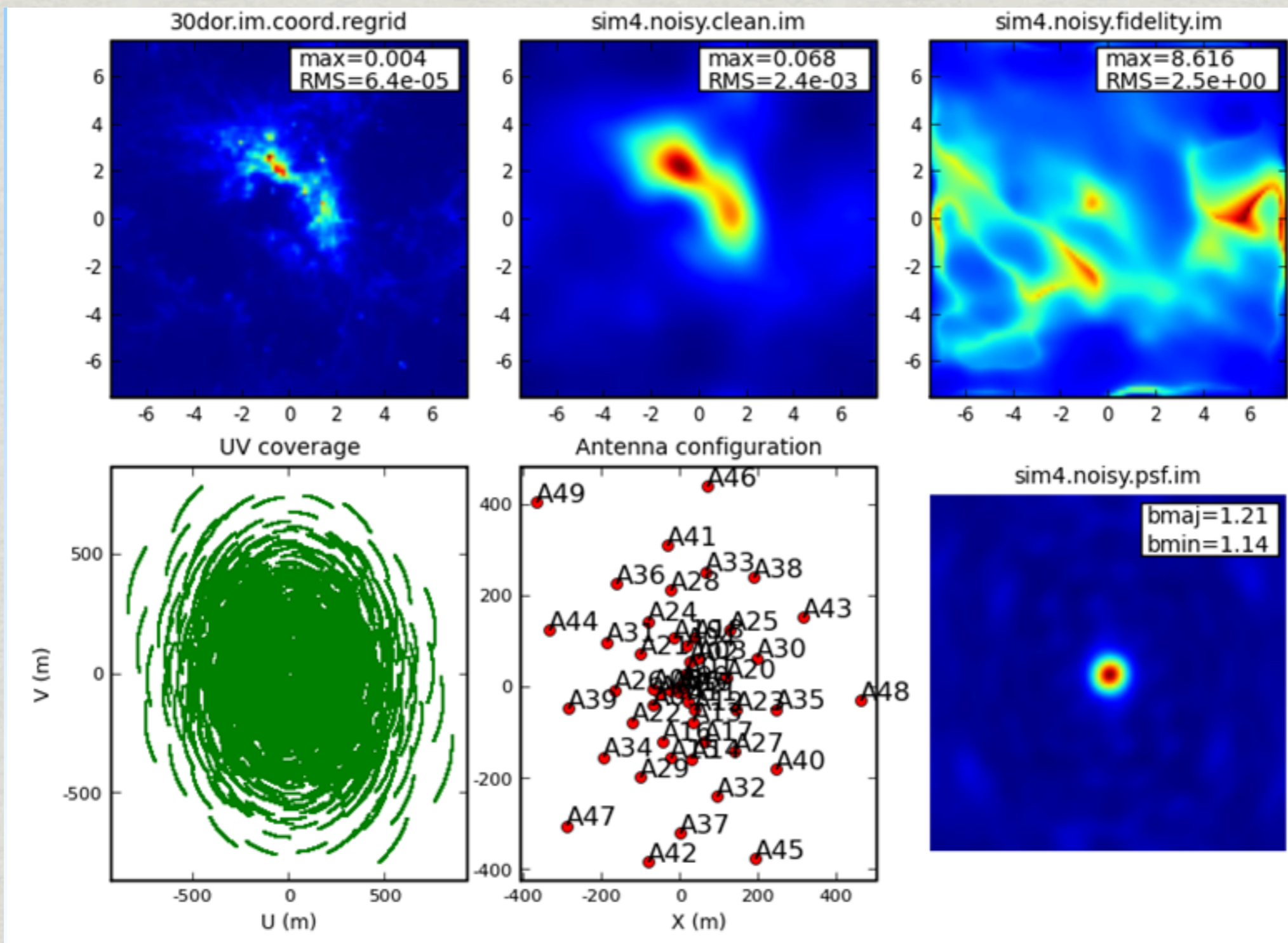
CASA used at ATF
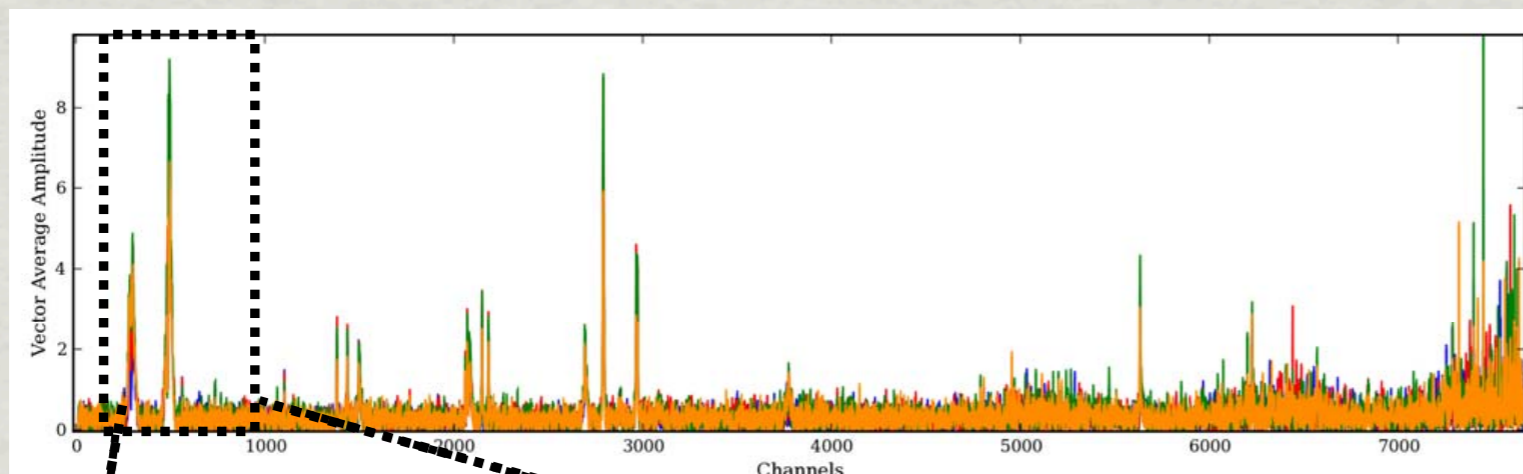
ALMA data format ASDM to CASA filler completed

CASA routinely being used to reduce data at the ALMA Test Facility at VLA

CASA Installation

* http://casa.nrao.edu

* Supports:

  * Fedora Core 6, Fedora 7 (32 and 64 bit)

  * RH Linux 4, 5 (32 and 64 bit)

  * Ubuntu 8.0.4 (32 and 64 bit)

  * openSUSE 10.0 and 11.0 (32 and 64 bit)

  * Debian 4.0 (32 bit and 64 bit)

  * Mac OSX Intel (10.4.7 and later, 32 bit only)

National Radio Astronomy Observatory

Tuesday 07 October 2008

Username: mzwaan

Password: ••••••

Forgot Password?

Create Account   Login »

* System inactivity timeout is 60 mins

MY.NRAO is compatible with the international World Wide Web Consortium (W3C) standards for site design,
It is therefore best viewed using the FireFox web browser.

Download Firefox

Proposal Help Desk

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

https://my.nrao.edu/nrao-2.0/secure/Home.htm

Q▾ Google

NRC   astro-ph   ESO   ALMA ▾   Apple (98) ▾   Astro ▾   News (2885) ▾   Tools ▾   Divers ▾   Amazon   eBay   Yahoo!

NRAO   http://casa.nrao.edu/Tu...   Sixth IRAM Millimeter In...   NRAO: Casa Issue Tracker

National Radio Astronomy Observatory

Dashboard   **Proposals**   **CASA Beta Release**   **Profile**   **Administration**

Hi, Martin | Sign Out

Tuesday 07 October 2008

Help

DASHBOARD

Options

- **Dashboard**
  - 📁 News & General Information
    - 📄 Telescope News
    - 📄 Information for Astronomers
    - 📄 Documentation
    - 📄 Release Notes
    - 📄 Policies
  - 📁 My Information
    - 📄 My Data
    - 📄 My Collaborators
    - 📄 My Observing Schedules
  - 📄 Feedback

The Dashboard is a management tool to help you keep track of news from the NRAO telescopes, latest developments, your data, your collaborators, and the status of dynamic scheduling.

You will also be able to keep track of the various requests for feedback that NRAO has open at any given time. This utility will be continually evolved over the next one to two years, and new items will be added based on feedback from the user community, so please be sure to tell us if there is something you would like to see on the Dashboard that would make working with NRAO easier.

Proposal Help Desk

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

○○○ NRAO

https://my.nrao.edu/nrao-2.0/secure/Casa.htm?   ○ Q▾ Google

NRC   astro-ph   ESO   ALMA ▾   Apple (98) ▾   Astro ▾   News (2885) ▾   Tools ▾   Divers ▾   Amazon   eBay   Yahoo!

⊗   NRAO   ⊗ http://casa.nrao.edu/Tu...   ⊗ Sixth IRAM Millimeter In...   ⊗ NRAO: Casa Issue Tracker

National Radio Astronomy Observatory

| Dashboard | Proposals | **CASA Beta Release** | Profile | Administration |

Hi, Martin | Sign Out

General   Helpdesk   Setup

Options

- **CASA**
  - General Information

CASA (Common Astronomy Software Applications)

CASA HOME PAGE

CASA is the suite of C++ application libraries for the reduction and analysis of radioastronomical data, packaged through a Python scripting interface.
CASA will be the primary off-line data reduction package for both ALMA and the EVLA.

Patch 2 of the Beta release is now available for a trial period of open download (see below). Please submit all questions, bugs, and enhancement requests through the "helpdesk tab" in the top left of this page.

Obtaining CASA

- A Beta version of CASA is available for a trial period of open download.
- Download CASA
- Release Notes

User Documentation

- Analysis Cookbook
- Developer's Toolkit Reference Manual
- CASA Memos

Demos and Tutorials

- CASA Tutorial
- CASA Viewer Demo
- Python Tutorial
- Python Reference Manual

CASA Helpdesk

- Access the CASA helpdesk via the "Helpdesk" tab above
- This page allows you to view the status of your submitted helpdesk tickets
- Submit new helpdesk tickets by selecting the "Submit New Helpdesk Ticket" button at the top of the Helpdesk page

✳ Installation on a Mac:
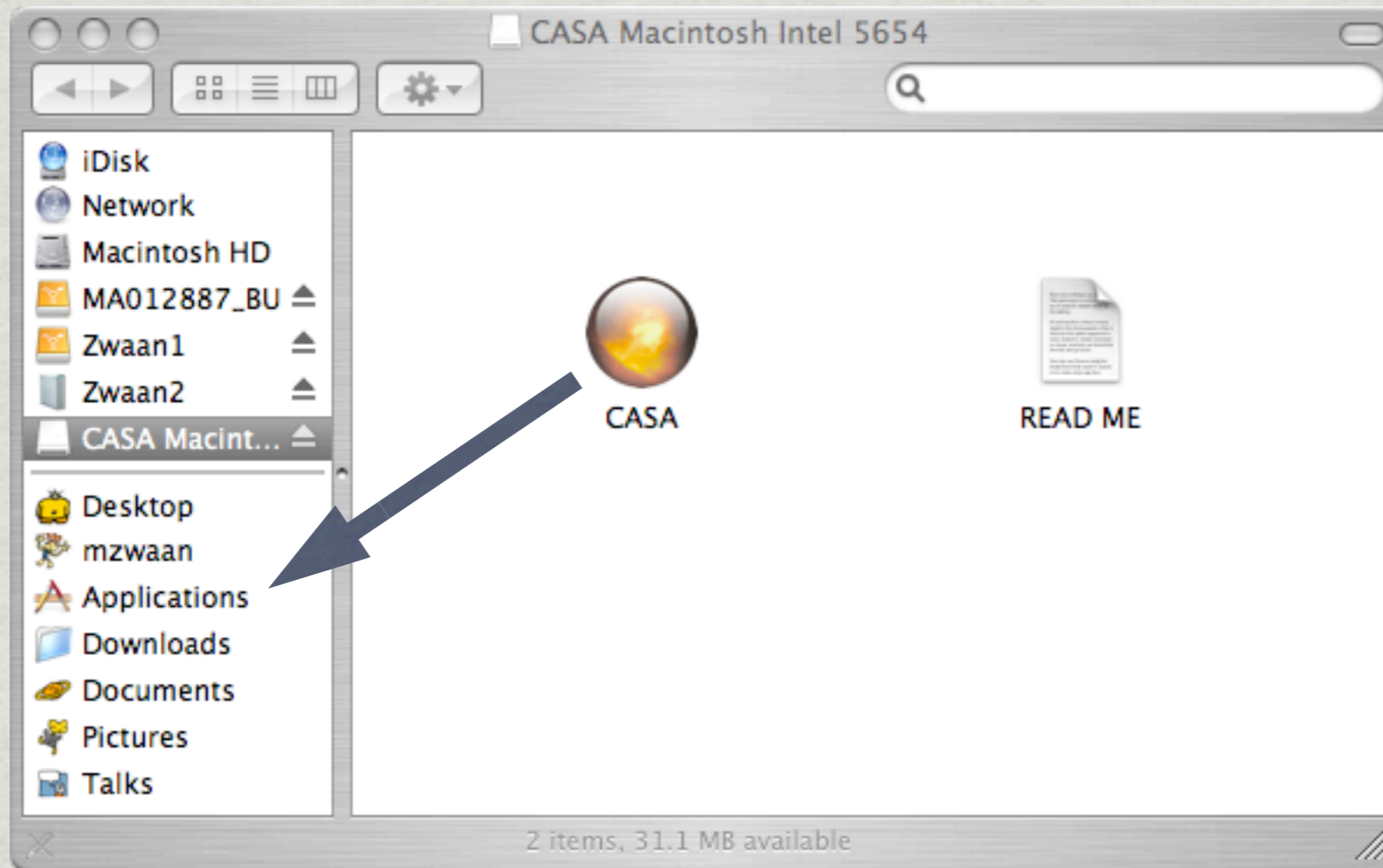
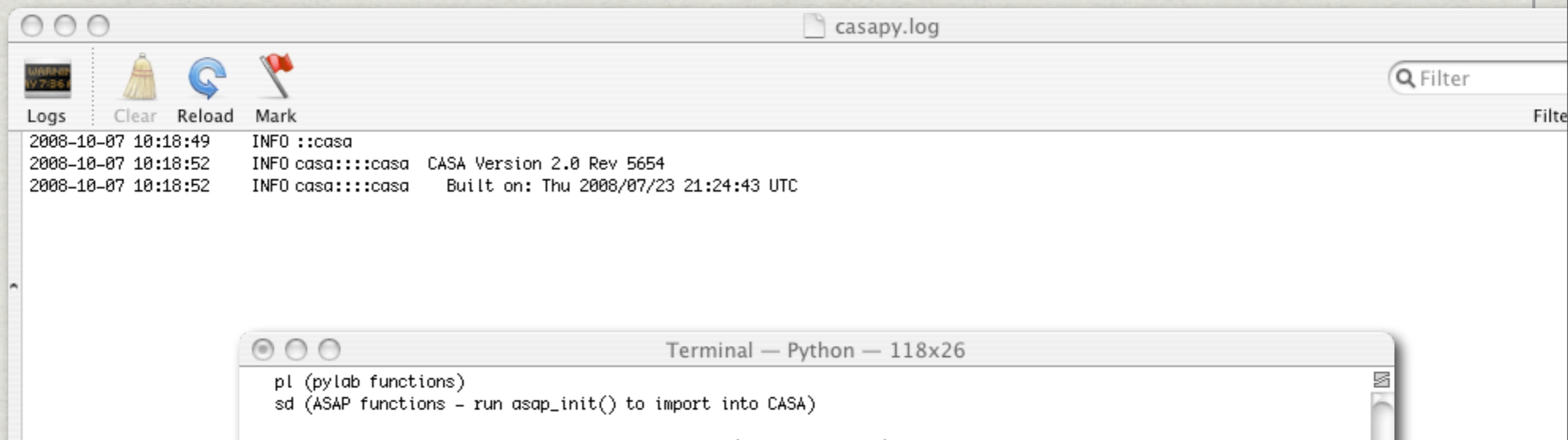

✳ Installation on linux:

✳ tar -zxf casapy-20.0.*.tar.gz

User support

* First read cookbook and help within CASA

* Use helpdesk at my.nrao.edu

  * tickets will be answered by (European) user support specialists

run CASA...   ✳ type "casapy"
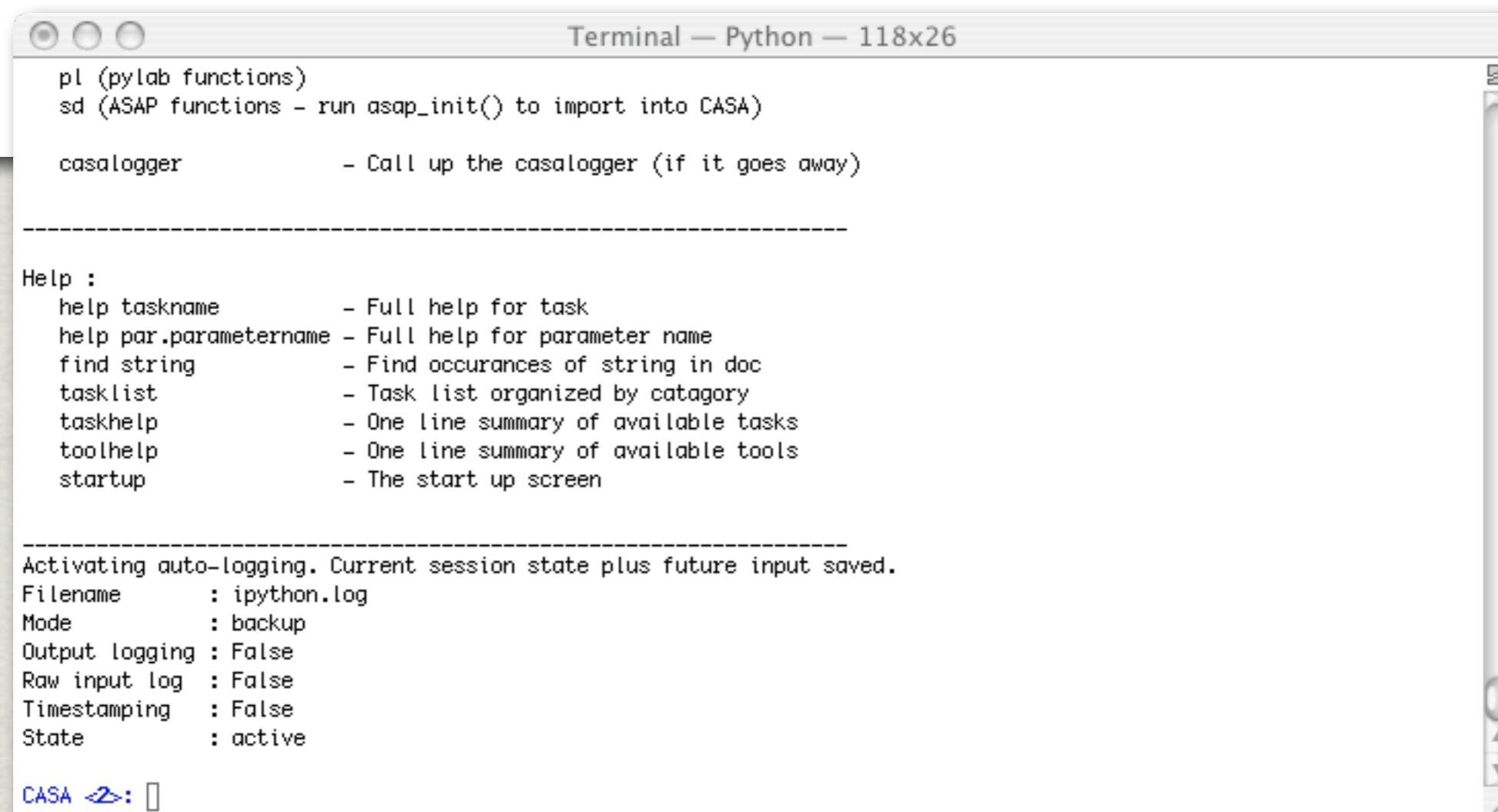

casapy.log

Logs   Clear   Reload   Mark   Q Filter   Filte

2008-10-07 10:18:49   INFO ::casa
2008-10-07 10:18:52   INFO casa::::casa   CASA Version 2.0 Rev 5654
2008-10-07 10:18:52   INFO casa::::casa   Built on: Thu 2008/07/23 21:24:43 UTC

Terminal — Python — 118x26

```
    pl (pylab functions)
    sd (ASAP functions - run asap_init() to import into CASA)

    casalogger               - Call up the casalogger (if it goes away)


    ----------------------------------------------------------------

Help :
    help taskname            - Full help for task
    help par.parametername   - Full help for parameter name
    find string              - Find occurances of string in doc
    tasklist                 - Task list organized by catagory
    taskhelp                 - One line summary of available tasks
    toolhelp                 - One line summary of available tools
    startup                  - The start up screen


    ----------------------------------------------------------------
Activating auto-logging. Current session state plus future input saved.
Filename       : ipython.log
Mode           : backup
Output logging : False
Raw input log  : False
Timestamping   : False
State          : active

CASA <2>: []
```

Summary of Current Capabilities

* **Data Import**

  * VLA (EVLA) archive

  * External ALMA and EVLA fillers complete

  * UVFITS from "any" other telescope

* **Flagging**

  * UV-plot based including time, channel averaging

  * Viewer flagging

  * Manual flagging

* **Calibration**

  * Polarization

  * Standard flux density calibrator images

  * Spline fitting and polynomial bandpass determination

  * Flexible combination of multiple spectral windows

Summary of Current Capabilities (continued)

* **Imaging**

  * Single field and mosaics

  * Widefield (low-frequency) imaging

  * Heterogeneous imaging

  * Interferometer plus single dish combination (3 methods)

  * Multi-scale clean (somewhat experimental)

  * Interactive clean boxing

  * Analysis includes image math, statistics, image plane fitting

CASA Interface

* IPython
    * shell access
    * autoparenthesis (autocall)
    * command history
    * macros
    * session logging (ipython.log, casapy.log)
    * numbered input/output
    * history/searching

Tasks in CASA

- ✸ tasks: high (user) level functionality

  - ✸ call from Python as functions

  - ✸ standard tasking interface

    - ✸ parameter manipulation using inp, default, saveinputs, tget

  - ✸ arguments are parameters

    - ✸ these are global Python variables

    - ✸ see Chapter 1.3 in Cookbook

Task Interface

✳ examine task parameters with <u>inp</u> :

```
xterm <2>

CASA <23>: default 'clean'
--------> default('clean')

CASA <24>: inp
--------> inp()
#  clean :: Calculates a deconvolved image with a selected clean algorithm

vis               =           ''          #   Name of input visibility file
imagename         =           ''          #   Pre-name of output images
mode              =          'mfs'        #   Type of selection (mfs, channel, velocity, frequency)
alg               =         'clark'       #   Algorithm to use (hogbom, clark, csclean, multiscale)
niter             =          500          #   Number of iterations
gain              =          0.1          #   Loop gain for cleaning
threshold         =          0.0          #   Flux level to stop cleaning (mJy)
mask              =          ['']         #   Name of mask image used in cleaning
cleanbox          =          []           #   clean box regions or file name or 'interactive'
imsize            = [256, 256]            #   Image size in pixels [nx,ny]; symmetric for single value
cell              = ['1.0arcsec', '1.0arcsec']     #   Cell size in arcseconds [x,y]
stokes            =          'I'          #   Stokes parameter to image (I,IV,IQU,IQUV)
field             =          'O'          #   Field name
phasecenter       =          ''           #   Field Identifier or direction of the image phase center
spw               =          ''           #   spectral window:channels: ''=>all
weighting         =       'natural'       #   Weighting to apply to visibilities
uvfilter          =        False          #   Apply additional filtering/uv tapering of the visibilities
timerange         =          ''           #   range of time to select from data
restfreq          =          ''           #   restfrequency to use in image
async             =        False          #   if True run in the background, prompt is freed

CASA <25>: █
```
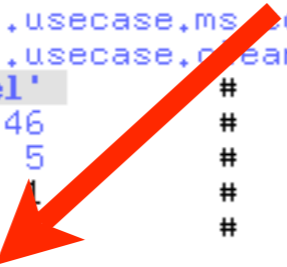
Expandable Parameters

* **boldface** parameter are expandable


```
xterm <2>                                                                    _

CASA <29>: tget clean
--------> tget(clean)
Restored parameters from file clean.last

CASA <30>: inp
--------> inp()
#  clean :: Calculates a deconvolved image with a selected clean algorithm

vis             = 'ngc5921.usecase.ms.contsub'    #   Name of input visibility file
imagename       = 'ngc5921.usecase.clean'   #   Pre-name of output images
mode            =      'channel'        #   Type of selection (mfs, channel, velocity, frequency)
    nchan       =           46       #   Number of channels to select
    start       =            5       #   Start channel
    step        =            1       #   Increment between channels/velocity
    width       =            1       #   Channel width (value > 1 indicates channel averaging)

alg             =       'clark'      #   Algorithm to use (hogbom, clark, csclean, multiscale)
niter           =         6000       #   Number of iterations
gain            =          0.1       #   Loop gain for cleaning
threshold       =          8.0       #   Flux level to stop cleaning (mJy)
mask            =           ''       #   Name of mask image used in cleaning
cleanbox        =           []       #   clean box regions or file name or 'interactive'
imsize          =    [256, 256]      #   Image size in pixels [nx,ny]; symmetric for single value
cell            =    [15.0, 15.0]    #   Cell size in arcseconds [x,y]
stokes          =          'I'       #   Stokes parameter to image (I,IV,IQU,IQUV)
field           =          '0'       #   Field name
phasecenter     =           ''       #   Field Identifier or direction of the image phase center
spw             =           ''       #   spectral window:channels: ''=>all
weighting       =       'briggs'     #   Weighting to apply to visibilities
    rmode       =       'norm'       #   Robustness mode (for Briggs weighting)
    robust      =          0.5       #   Briggs robustness parameter
    noise       =       '0.0Jy'      #   noise parameter for briggs weighting when rmode='abs'
    npixels     =            0       #   number of pixels to determine uv-cell size 0=> field of view

uvfilter        =        False       #   Apply additional filtering/uv tapering of the visibilities
timerange       =           ''       #   range of time to select from data
```

Parameter Checking

* sanity checks of parameters in <u>inp</u> :

Parameter Checking

✳ sanity checks of parameters in <u>inp</u> :

Help on Tasks

※ In-line help <taskname> command:

```
CASA <1>: help clean
--------> help(clean)
Help on function clean in module clean:

clean(vis=None, imagename=None, mode=None, alg=None, niter=None, gain=None, thresh
p=None, imsize=None, cell=None, stokes=None, field=None, spw=None, weighting=None,
ne, uvfilterbpa=None, timerange=None, restfreq=None, noise=None, npixels=None, sca
    Deconvolves image with a selected clean algorithm.

    Three types of point-source deconvolution, as well as multi-scale
    deconvolution are available.  Cleaning boxes can be made and
    adjusted interactively, including the ability to do channel
    independent boxing.

    The clean task automatically uses the "corrected" datacolumn
    which will be different from the "data" column after application
    of the task applycal. To revert the contents of the "corrected"
    datacolumn to the "data" column run the task clearcal (for
    example, to start over after a previous self-calibration process).

    Keyword arguments:
    vis -- Name of input visibility file (MS):
            default: none; example: vis='n5921.ms'
    imagename -- Pre-name of output images:
            default: none; example: imagename='m2'
```

Tools in CASA

* CASA Toolkit underneath tasks

    * core AIPS++ code (mostly in C++)

* tools are functions

    * call from casapy as `<tool>.<method>()`

    * default tool objects are pre-constructed

   * e.g. imager (im) , calibrater (cb), ms (ms) , etc. (see toolhelp)

Data in CASA

* Measurement sets (ms): directory in your working directory

* MAIN table: contains main data, in columns:

  * **DATA**: original visibility data

  * **CORRECTED_DATA**: calibrated data

  * **MODEL_DATA**: Fourier transform of model

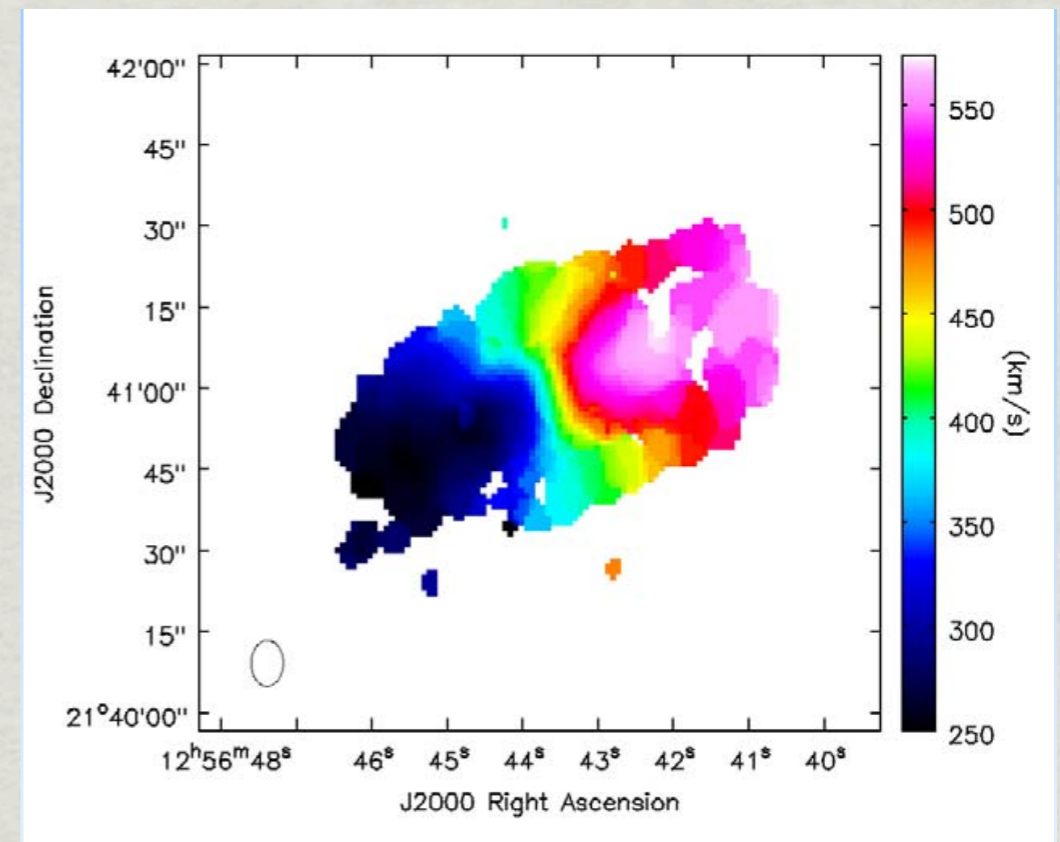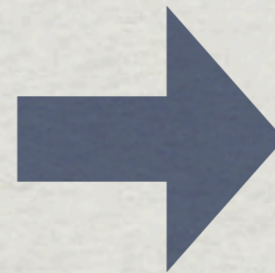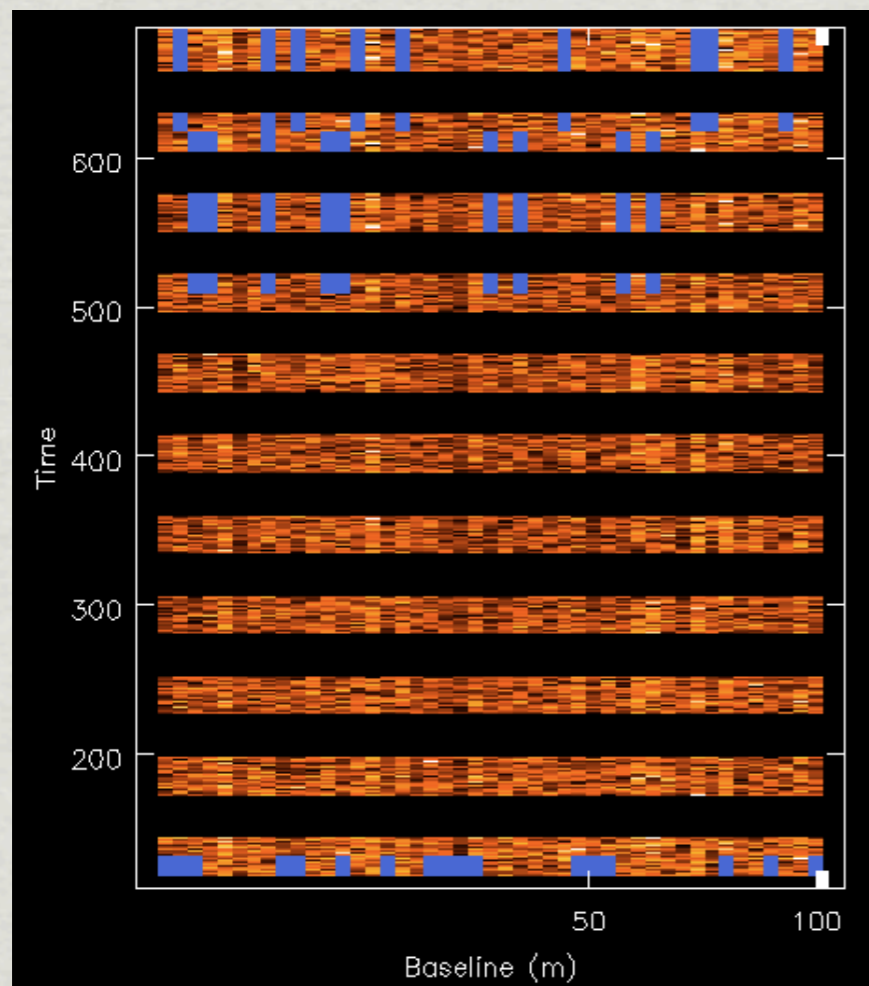 * **IMAGING_WEIGHT**: weights for imaging

getting your data into CASA

* UVFITS: **importuvfits** or **exportuvfits**

* VLA archive data: **importvla**

* ALMA and EVLA Science data model: **importasdm**

Data reduction in CASA
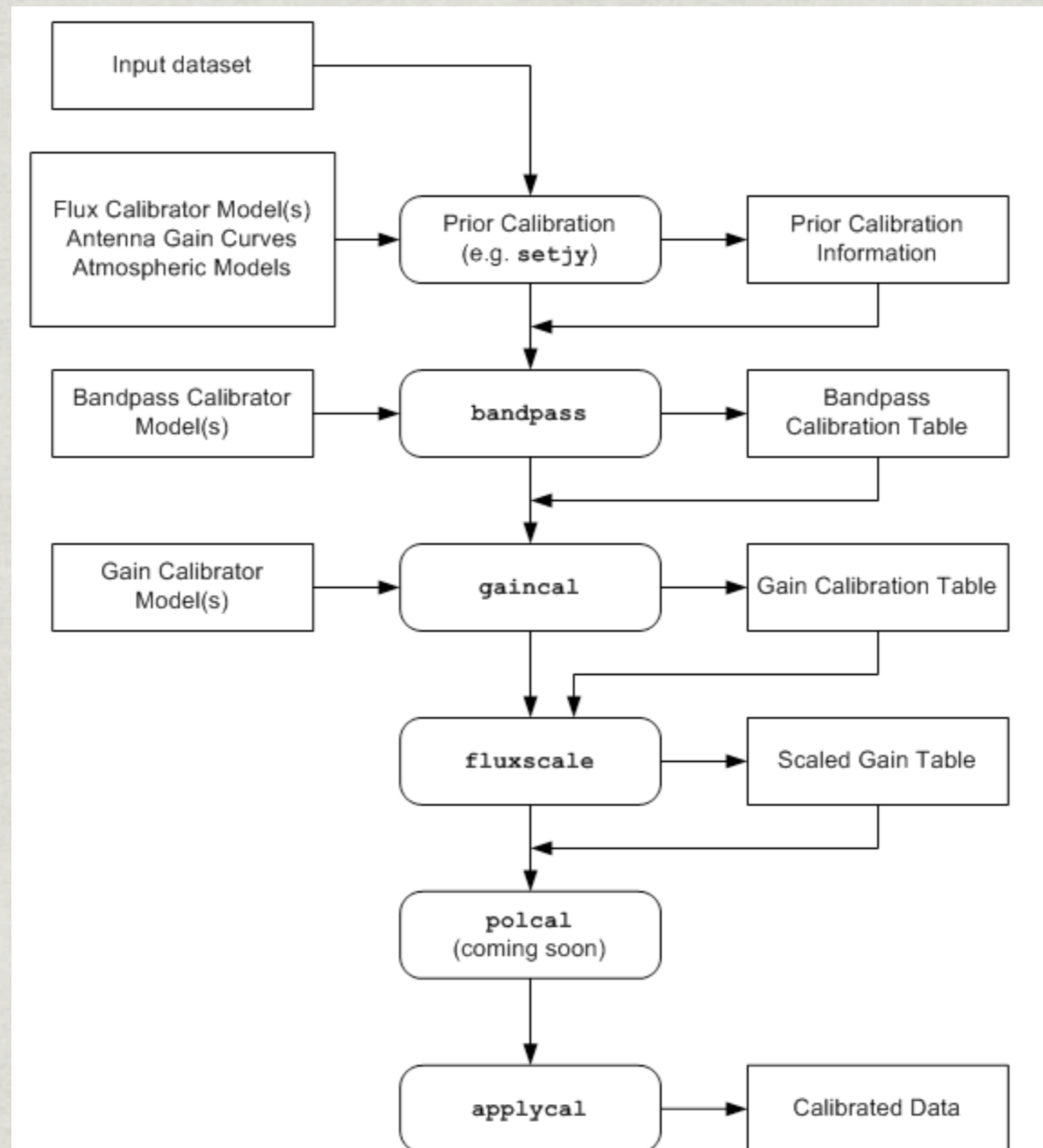
CASA demo

✳ Data reduction of NGC 4826 from the BIMA SONG survey (Helfer, Thornley, Regan et al. 2003)

CASA demonstration