NRAO

CASA

National Radio Astronomy Observatory

Juergen Ott (NRAO)

Atacama Large Millimeter/submillimeter Array
Expanded Very Large Array
Robert C. Byrd Green Bank Telescope
Very Long Baseline Array

NRAO

CASA

National Radio Astronomy Observatory

Juergen Ott (NRAO)

Atacama Large Millimeter/submillimeter Array
Expanded Very Large Array
Robert C. Byrd Green Bank Telescope
Very Long Baseline Array

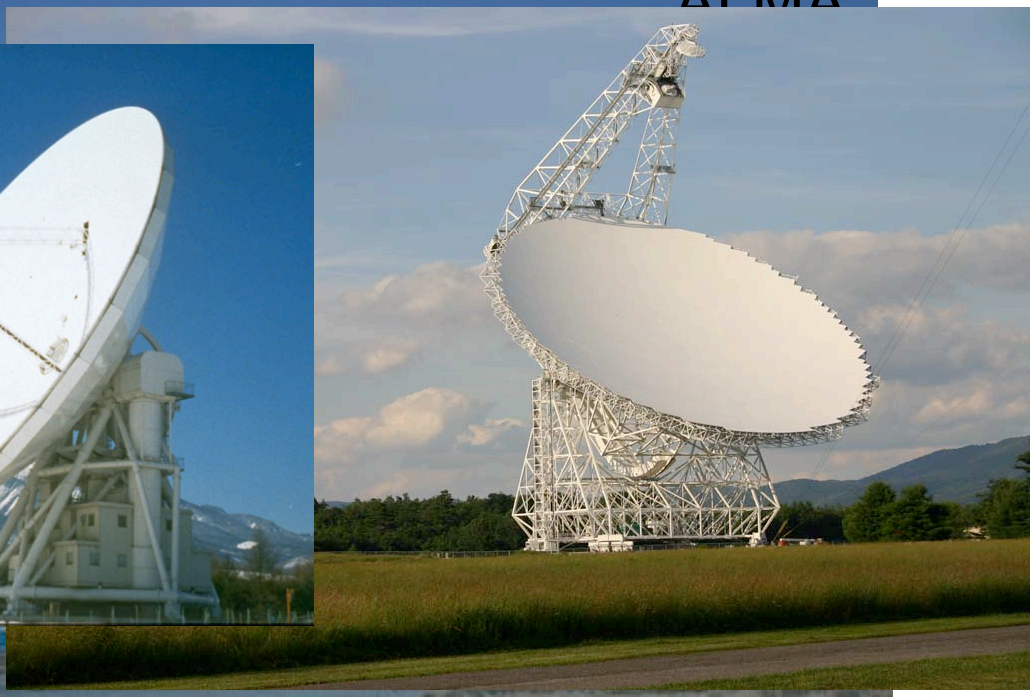
Introduction to CASA

Juergen Ott (CASA project scientist)
Crystal Brogan (CASA ALMA subsystem scientist)
Miriam Krauss (CASA EVLA subsystem scientist)
Jeff Kern (CASA manager)

Introduction to CASA

Juergen Ott (CASA project scientist)
Crystal Brogan (CASA ALMA subsystem scientist)
Miriam Krauss (CASA EVLA subsystem scientist)
Jeff Kern (CASA manager)

CASA
Common Astronomy Software Applications



NRAO CASA Common Astronomy Software Applications

CASA (Common Astronomy Software Applications)

- CASA is the offline data reduction package for ALMA and the EVLA (data from other telescopes usually work, too, but not primary goal of CASA)
- Import/export data, edit, calibrate, image, analyze
- Also supports single dish! (based on ASAP)
- CASA has many tasks and a LOT of tools
- It is very versatile on how you run it
- Easy to write scripts and we would like to build up repository for contributed scripts and tasks
- We have a lot of documentation, reduction tutorials, helpdesk
- CASA has some of the most sophisticated algorithms implemented (multi-scale clean, Taylor term expansion for wide bands, W-term projection, OTF mosaicing, etc.)
- We have a active Algorithm Research Group, so more coolness to come

CASA (Common Astronomy Software Applications)

- Current version: 3.2.1

  New releases about every 6 months.

  For download: casa.nrao.edu Linux, Mac OS X

  "release", "test" and "stable" versions available at NRAO/ESO/ALMA

  > casapy - latest release: underwent lots of testing, updated documentation

  > casapy-test - cutting edge capabilities, no documentation, bugs

  > casapy-stable - less bugs but also less features, could be a release

Outline

- IPython & Python
- CASA help
- CASA task interface
- MS and data selection
- Documentation

CASA Interface

- IPython
  - shell access
  - autoparenthesis (autocall)
  - command history
  - session logging
    - ipython.log – ipython command history
    - casapy.log – casa messages
  - numbered input/output
  - history/searching

Python Pointers

- to run a .py script:

  `execfile('<scriptname>')`

  example: `execfile('ngc5921_demo.py')`
- indentation matters!
  - be careful when doing cut-and-paste to Python
  - cut a few (4-6) lines at a time
- Python counts from 0 to n-1!
- variables are global when using task interface
- tasknames are objects (not variables)

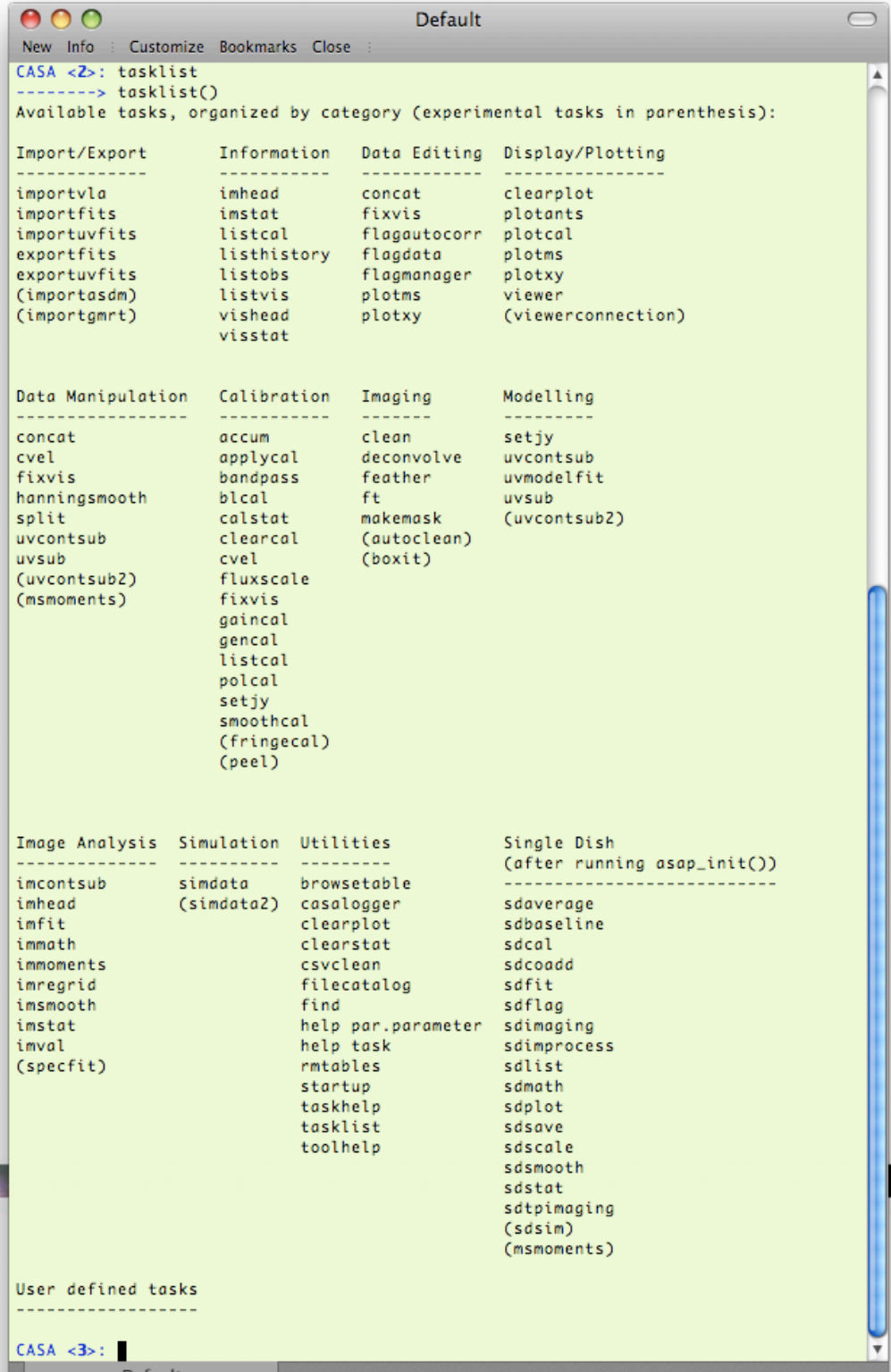
Tasks and tools in CASA

- Tasks - high-level functionality
  - function call or parameter handling interface
  - these are what you should use in tutorial
- Tools - complete functionality
  - tool.method calls, used by tasks
  - sometimes shown in tutorial scripts

Key Tasks

- To see list of tasks organized by type:

```
>tasklist
```

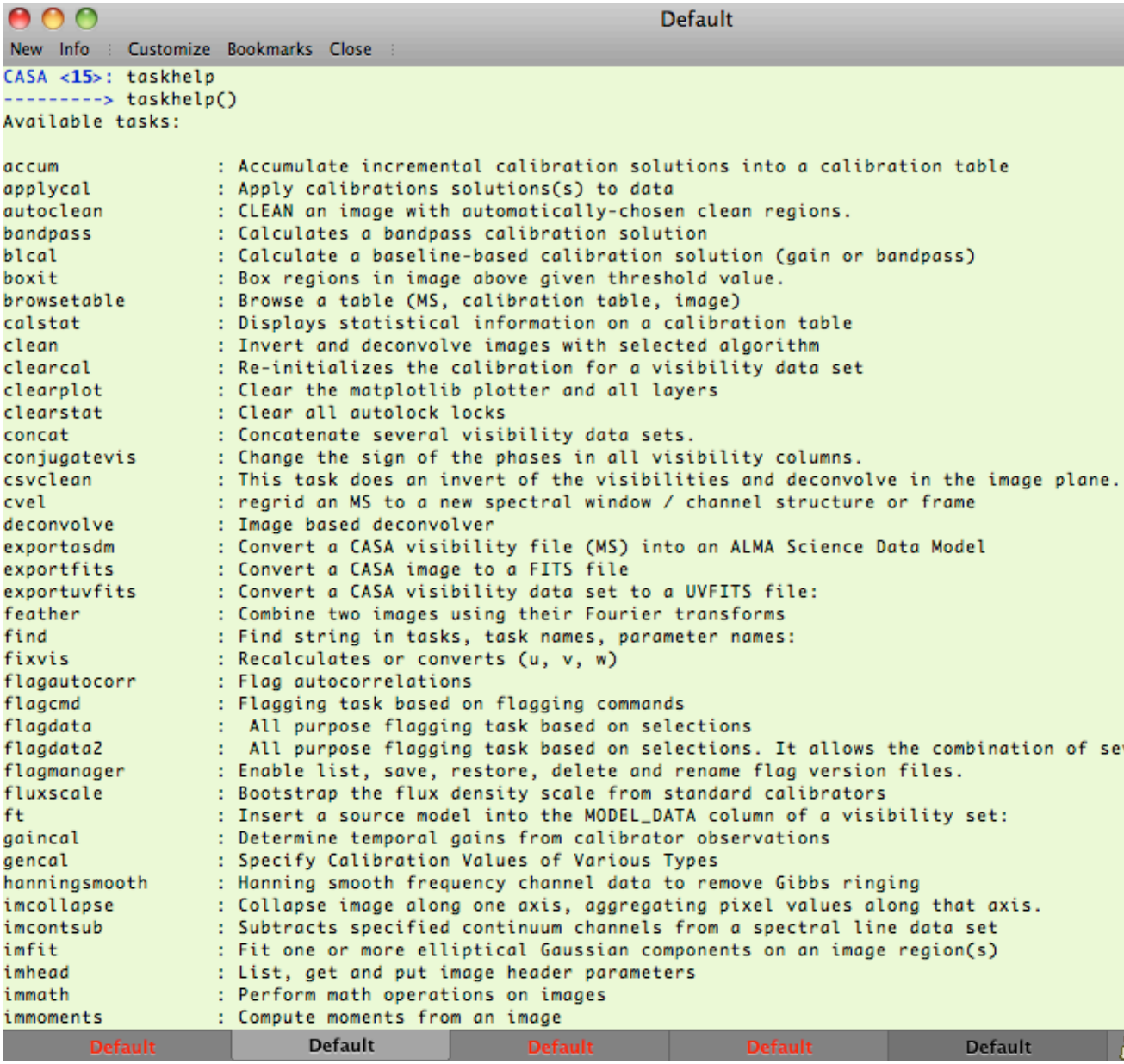


NRAO CASA Common Astronomy Software Applications

Key Tasks

- To see list of tasks with short help:

```
>taskhelp
```



Task Interface

- examine task parameters with inp :

Default

New Info Customize Bookmarks Close

```
CASA <12>: inp
---------> inp()
#  clean :: Invert and deconvolve images with selected algorithm
vis                 =         ''        #  Name of input visibility file
imagename           =         ''        #  Pre-name of output images
outlierfile         =         ''        #  Text file with image names, sizes, centers for outliers
field               =         ''        #  Field Name or id
spw                 =         ''        #  Spectral windows e.g. '0~3', '' is all
selectdata          =      False        #  Other data selection parameters
mode                =  'channel'        #  Spectral gridding type (mfs, channel, velocity, frequency)
     nchan          =         -1        #  Number of channels (planes) in output image; -1 = all
     start          =          0        #  Begin the output cube at the frequency of this channel in the MS
     width          =          1        #  Width of output channel relative to MS channel (# to average)
     interpolation  =   'linear'        #  Spectral interpolation (nearest, linear, cubic). Use nearest for
                                        #   mode=channel
     chaniter       =      False        #  Clean each channel to completion (True), or all channels each cycle (False)
     outframe       =         ''        #  velocity frame of output image

gridmode            =         ''        #  Gridding kernel for FFT-based transforms, default='' None
niter               =        500        #  Maximum number of iterations
gain                =        0.1        #  Loop gain for cleaning
threshold           =   '0.0mJy'        #  Flux level to stop cleaning, must include units: '1.0mJy'
psfmode             =    'clark'        #  Method of PSF calculation to use during minor cycles
imagermode          =         ''        #  Options: 'csclean' or 'mosaic', '', uses psfmode
multiscale          =         []        #  Deconvolution scales (pixels); [] = standard clean
interactive         =      False        #  Use interactive clean (with GUI viewer)
mask                =         []        #  Cleanbox(es), mask image(s), region(s), or a level
imsize              = [256, 256]        #  x and y image size in pixels. Single value: same for both
cell                = ['1.0arcsec']     #  x and y cell size(s). Default unit arcsec.
phasecenter         =         ''        #  Image center: direction or field index
restfreq            =         ''        #  Rest frequency to assign to image (see help)
stokes              =        'I'        #  Stokes params to image (eg I,IV,IQ,IQUV)
weighting           =  'natural'        #  Weighting of uv (natural, uniform, briggs, ...)
uvtaper             =      False        #  Apply additional uv tapering of visibilities
modelimage          =         ''        #  Name of model image(s) to initialize cleaning
restoringbeam       =       ['']        #  Output Gaussian restoring beam for CLEAN image
pbcor               =      False        #  Output primary beam-corrected image
minpb               =        0.2        #  Minimum PB level to use
calready            =       True        #  True required for self-calibration
async               =      False        #  If true the taskname must be started using clean(...)

CASA <13>:
```

Default | Default | Default | Default | Default

Task Interface

- standard tasking interface
  - use parameters set as global Python variables

    (set) `<param> = <value>`

    (e.g. `vis = 'ngc5921.demo.ms'`)
  - parameter manipulation command

    using inp , default , saveinputs , tget, tput
  - execute

    `<taskname>` or `go` (e.g. `clean()`)
  - return values

    some tasks return Python dictionaries, e.g. `myval=imval()`

Task Execution

- two ways to invoke:
  - call from Python as functions with arguments

    taskname(arg1=val1, arg2=val2, ...), like

    Clean(vis='input.ms', imagename='galaxy',selectvis=T, robust=0.5, imsize=[200,200])

    unspecified parameters will be defaulted (globals not used)
  - use standard tasking interface

    use global variables for task parameters
  - see Chapter 1.3 in Cookbook

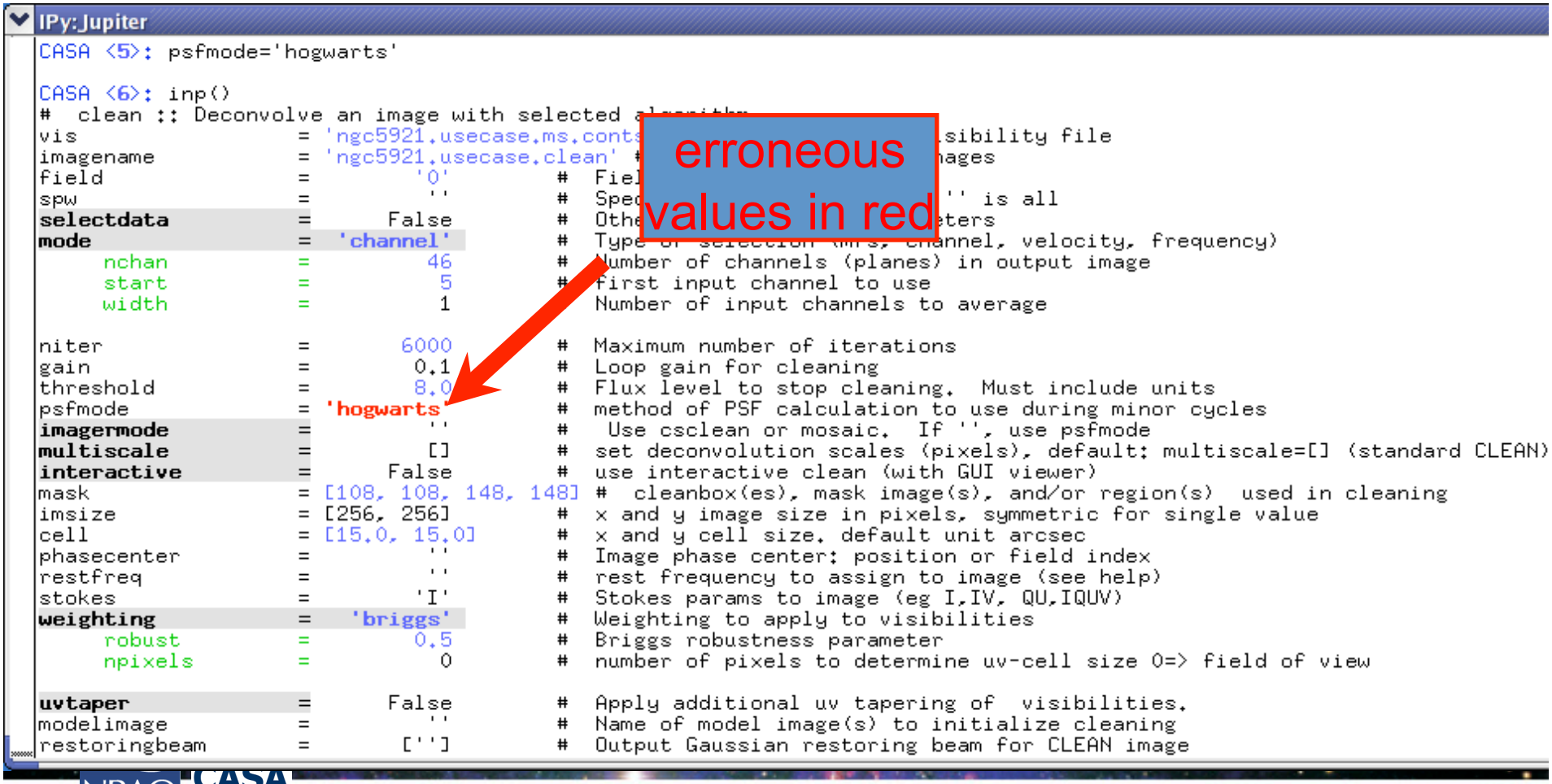
Expandable Parameters

IPy:Jupiter

```
CASA <3>: tget('clean')
Restored parameters from file clean.last

CASA <4>: inp()
#  clean :: Deconvolve an image with selected algorithm
vis                 = 'ngc5921.usecase.ms.contsub' #  name of input visibility file
imagename           = 'ngc5921.usecase.clean' #  Pre-name of output images
field               =        '0'           #  Field Name
spw                 =         ''           #  Spectral windows:channels: '' is all
selectdata          =      False           #  Other data selection parameters
mode                =  'channel'           #  Type of selection (mfs, channel, velocity, frequency)
     nchan          =         46           #  Number of channels (planes) in output image
     start          =          5           #  first input channel to use
     width          =          1           #  Number of input channels to average

niter               =       6000           #  Maximum number of iterations
gain                =        0.1           #  Loop gain for cleaning
threshold           =        8.0           #  Flux level to stop cleaning,  Must include units
psfmode             =    'clark'           #  method of PSF calculation to use during minor cycles
imagermode          =         ''           #   Use csclean or mosaic,  If '', use psfmode
multiscale          =         []           #  set deconvolution scales (pixels), default: multiscale=[] (standard CLEAN)
interactive         =      False           #  use interactive clean (with GUI viewer)
mask                = [108, 108, 148, 148] #  cleanbox(es), mask image(s), and/or region(s)  used in cleaning
imsize              = [256, 256]           #  x and y image size in pixels, symmetric for single value
cell                = [15.0, 15.0]         #  x and y cell size, default unit arcsec
phasecenter         =         ''           #  Image phase center: position or field index
restfreq            =         ''           #  rest frequency to assign to image (see help)
stokes              =        'I'           #  Stokes params to image (eg I,IV, QU,IQUV)
weighting           =   'briggs'           #  Weighting to apply to visibilities
     robust         =        0.5           #  Briggs robustness parameter
     npixels        =          0           #  number of pixels to determine uv-cell size 0=> field of view

uvtaper             =      False           #  Apply additional uv tapering of  visibilities,
modelimage          =         ''           #  Name of model image(s) to initialize cleaning
```

Parameter Checking

sanity checks of parameters in inp :



Help on Tasks

- In-line help:

>help ‘clean’ OR >pdoc clean

IPy:Jupiter

```
CASA <7>: help('clean')
Help on module clean:

NAME
    clean

FILE
    /usr/lib/casapy/20.0.5444test-001/lib/python2.5/clean.py

DESCRIPTION
    # This file was generated using xslt from its XML file
    #
    # Copyright 2007, Associated Universities Inc., Washington DC
    #

FUNCTIONS
    clean_imp(vis=None, imagename=None, field=None, spw=None, selectdata=
 gain=None, threshold=None, psfmode=None, imagermode=None, ftmachine=None
one, mask=None, nchan=None, start=None, width=None, imsize=None, cell=Non
er=None, outertaper=None, innertaper=None, modelimage=None, restoringbeam
r=None, cyclespeedup=None, async=None)
        Deconvolve an image with selected algorithm

             The main clean deconvolution task.  It contains many functio

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


             vis -- Name of input visibility file
                     default: none; example: vis='ngc5921.ms'
             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.model: image of clean components
                       m2.residual: image of residuals
```

Tools in CASA

- What if there's no task?

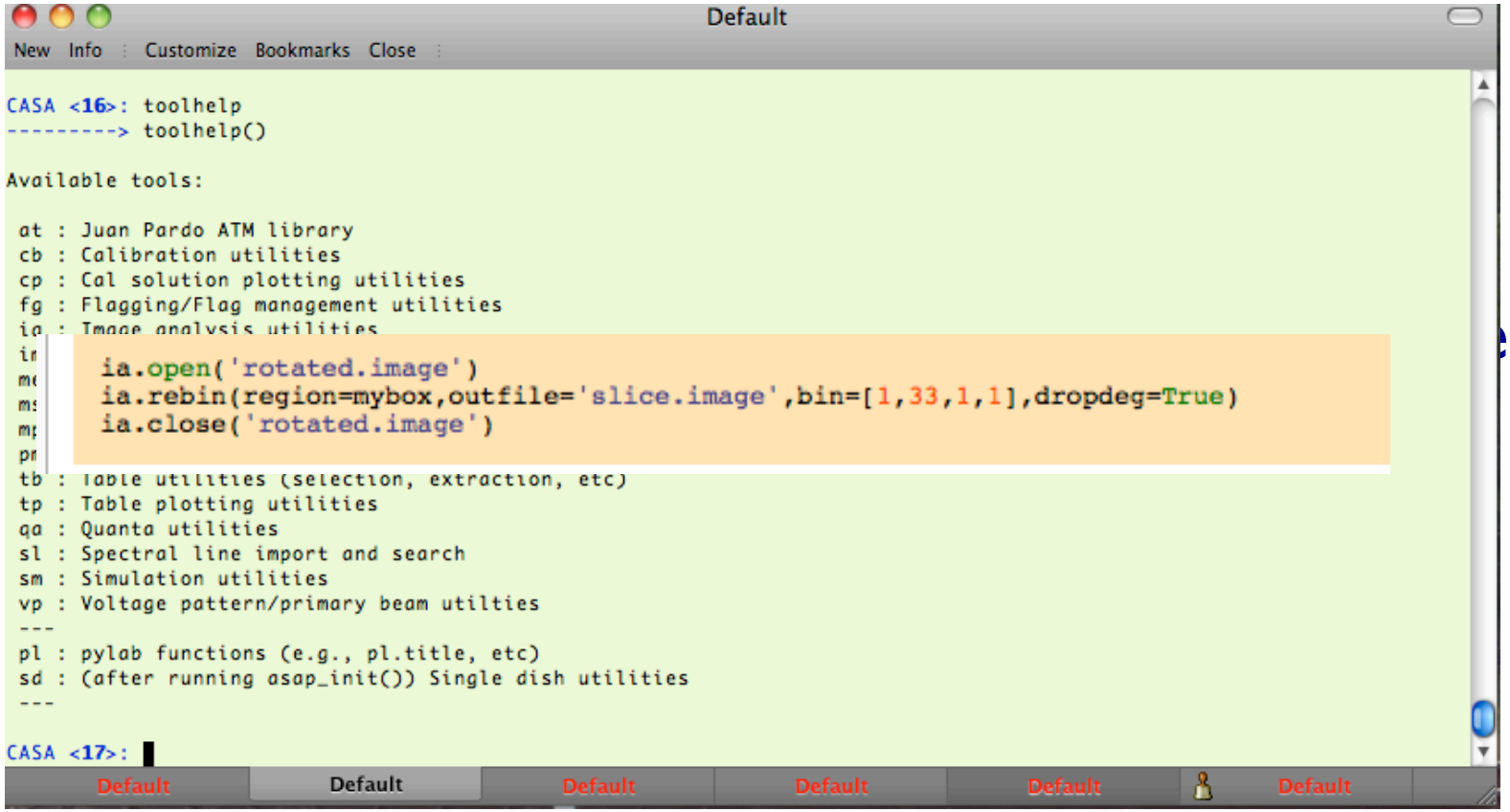
→ use CASA tools! (tasks are built upon tools)

- CASA Toolkit underneath tasks
- core AIPS++ code (mostly in C++)

- tools are functions/methods
  - call from casapy as `<tool>.<method>()`
  - default tool objects are pre-constructed
    - e.g. imager (im) , calibrater (cb), ms (ms) , etc. (see toolhelp)

CASA Tool List

- list of default tools from toolhelp :



CASA Tool List

- There's a good chance that your problem can be solved on the tool level, don't be afraid!
- ~1000 tools available!

- tools described in the CASA Toolkit Reference:
  - http://casa.nrao.edu/docs/CasaRef/CasaRef.html

The Measurement Set

- the MS is a directory on disk
  - the MAIN table in `table.*` files
  - also contains sub-tables
    - e.g. FIELD, SOURCE, ANTENNA, etc.
  - sub-tables are sub-directories
  - to copy must `cp -rf` to get contents
  - Best to remove ms with `rmtables('filename')`
    - WARNING: moving a MS can break cal-table dependencies

Example MS

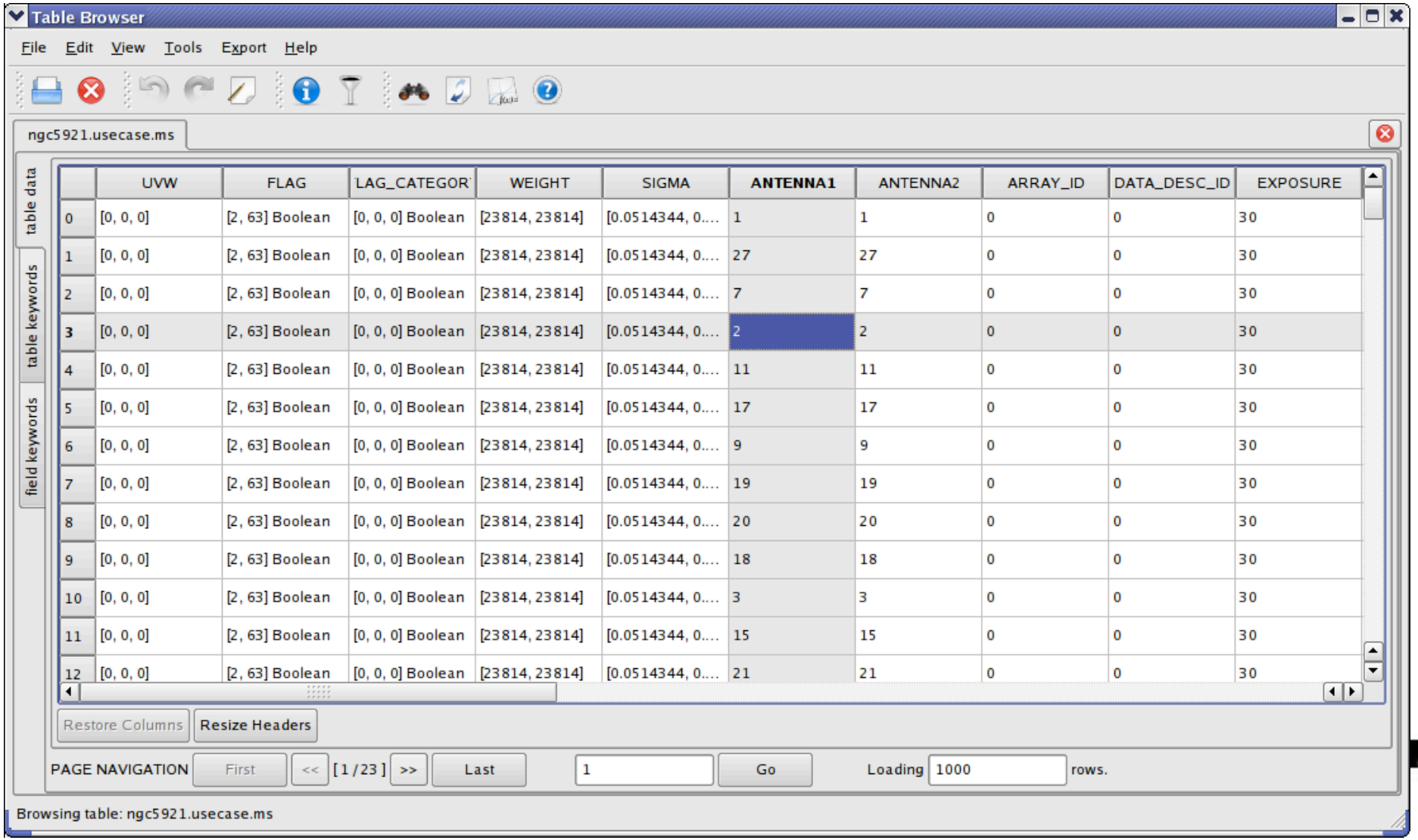
- **Example:** `ls ngc5921.usecase.ms`

```
smyers@olorin ~/CASA/Test $ ls ngc5921.usecase.ms
ANTENNA                 POLARIZATION            table.f1                table.f3_TSM1           table.f8
DATA_DESCRIPTION        PROCESSOR               table.f10               table.f4                table.f8_TSM1
FEED                    SORTED_TABLE            table.f10_TSM1          table.f5                table.f9
FIELD                   SOURCE                  table.f11               table.f5_TSM1           table.f9_TSM1
FLAG_CMD                SPECTRAL_WINDOW         table.f11_TSM1          table.f6                table.info
HISTORY                 STATE                   table.f2                table.f6_TSM0           table.lock
OBSERVATION             table.dat               table.f2_TSM1           table.f7
POINTING                table.f0                table.f3                table.f7_TSM1
```

```
smyers@olorin ~/CASA/Test $ ls ngc5921.usecase.ms/FIELD
table.dat       table.f0        table.f0i       table.info      table.lock
```

MAIN Table Contents

- Example using task browsetable:



	UVW	FLAG	LAG_CATEGOR	WEIGHT	SIGMA	ANTENNA1	ANTENNA2	ARRAY_ID	DATA_DESC_ID	EXPOSURE
0	[0, 0, 0]	[2, 63] Boolean	[0, 0, 0] Boolean	[23814, 23814]	[0.0514344, 0....	1	1	0	0	30
1	[0, 0, 0]	[2, 63] Boolean	[0, 0, 0] Boolean	[23814, 23814]	[0.0514344, 0....	27	27	0	0	30
2	[0, 0, 0]	[2, 63] Boolean	[0, 0, 0] Boolean	[23814, 23814]	[0.0514344, 0....	7	7	0	0	30
3	[0, 0, 0]	[2, 63] Boolean	[0, 0, 0] Boolean	[23814, 23814]	[0.0514344, 0....	2	2	0	0	30
4	[0, 0, 0]	[2, 63] Boolean	[0, 0, 0] Boolean	[23814, 23814]	[0.0514344, 0....	11	11	0	0	30
5	[0, 0, 0]	[2, 63] Boolean	[0, 0, 0] Boolean	[23814, 23814]	[0.0514344, 0....	17	17	0	0	30
6	[0, 0, 0]	[2, 63] Boolean	[0, 0, 0] Boolean	[23814, 23814]	[0.0514344, 0....	9	9	0	0	30
7	[0, 0, 0]	[2, 63] Boolean	[0, 0, 0] Boolean	[23814, 23814]	[0.0514344, 0....	19	19	0	0	30
8	[0, 0, 0]	[2, 63] Boolean	[0, 0, 0] Boolean	[23814, 23814]	[0.0514344, 0....	20	20	0	0	30
9	[0, 0, 0]	[2, 63] Boolean	[0, 0, 0] Boolean	[23814, 23814]	[0.0514344, 0....	18	18	0	0	30
10	[0, 0, 0]	[2, 63] Boolean	[0, 0, 0] Boolean	[23814, 23814]	[0.0514344, 0....	3	3	0	0	30
11	[0, 0, 0]	[2, 63] Boolean	[0, 0, 0] Boolean	[23814, 23814]	[0.0514344, 0....	15	15	0	0	30
12	[0, 0, 0]	[2, 63] Boolean	[0, 0, 0] Boolean	[23814, 23814]	[0.0514344, 0....	21	21	0	0	30

Visualization Tools

- Data needs to be displayed to understand it!
- Visibilities: plotms, msview
- Images: viewer, imview
- Calibration tables: plotcal (plotms)
- Any table values: browsetable
- Single dish: sdplot

- Plot anything: use python's matplotlib

PlotMS

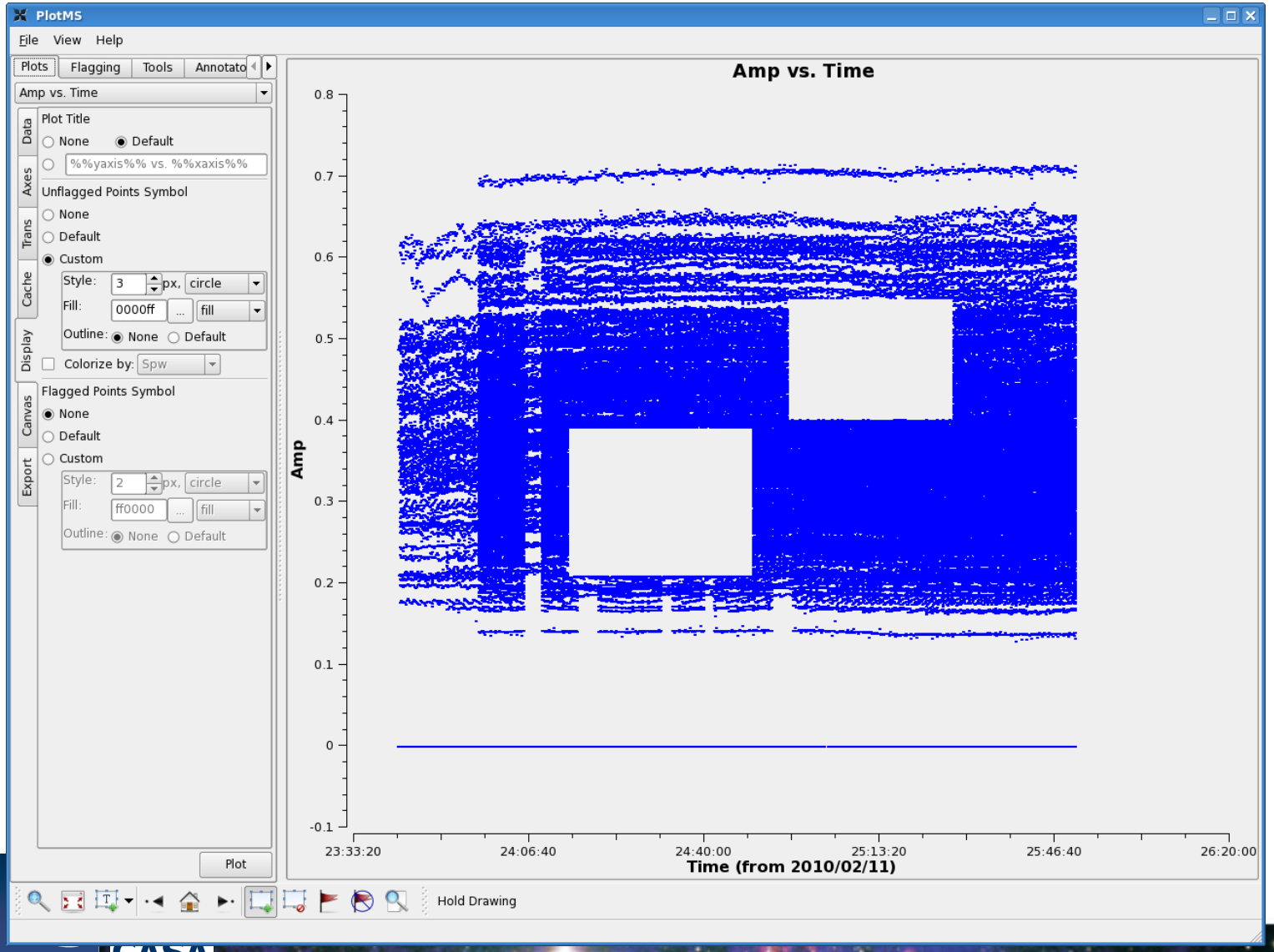

PlotMS
File View Help
Plots Flagging Tools Annotato
Amp vs. Time
Data Axes Trans Cache Display Canvas Export
Plot Title
None Default
%%yaxis%% vs. %%xaxis%%
Unflagged Points Symbol
None
Default
Custom
Style: 3 px, circle
Fill: 0000ff ... fill
Outline: None Default
Colorize by: Spw
Flagged Points Symbol
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

Image Viewer

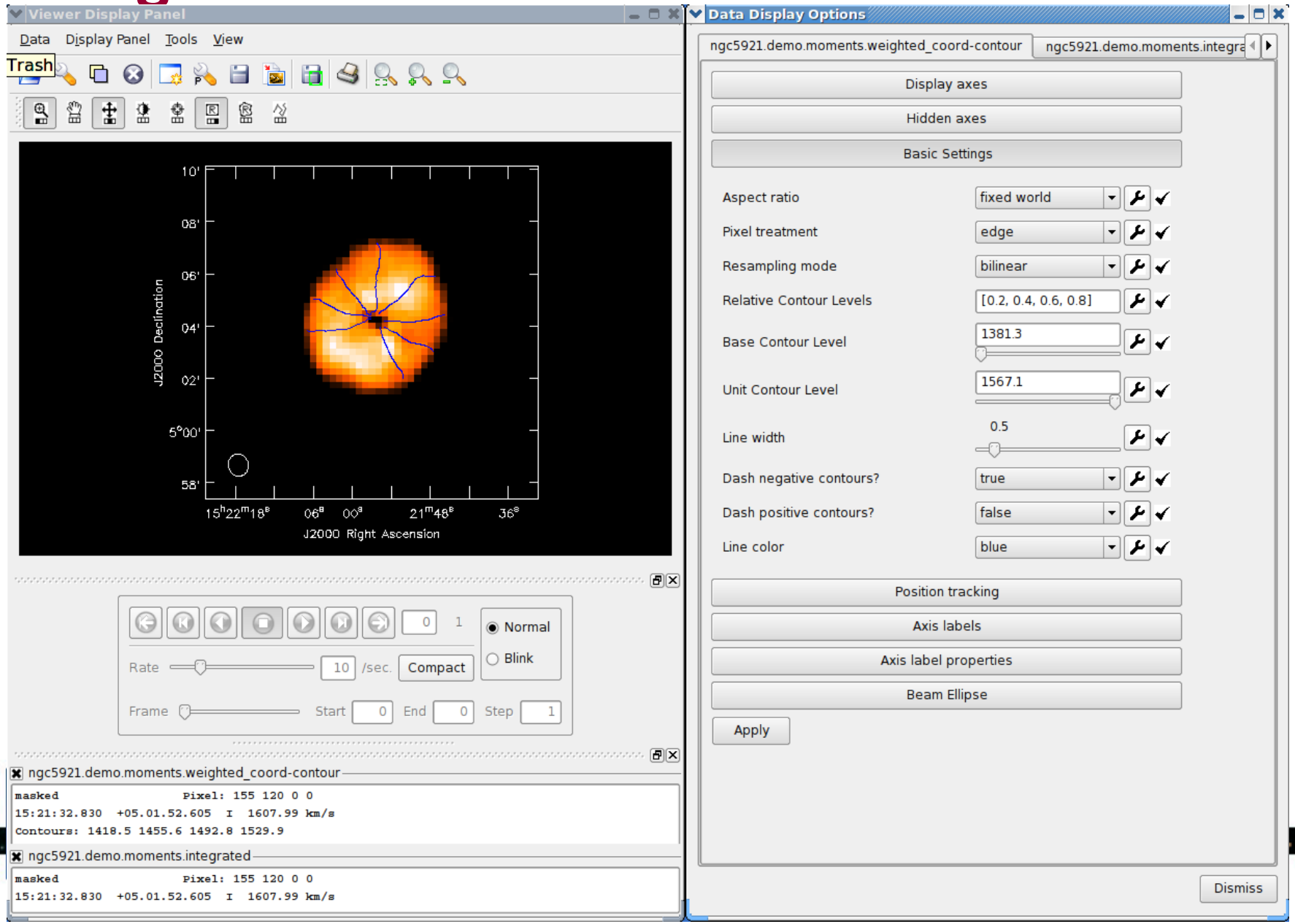
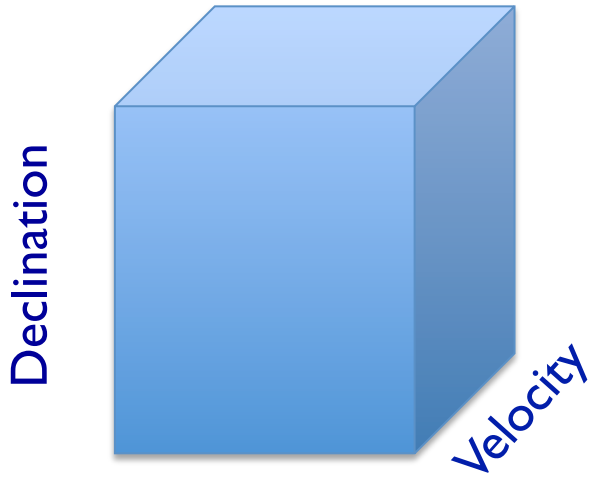
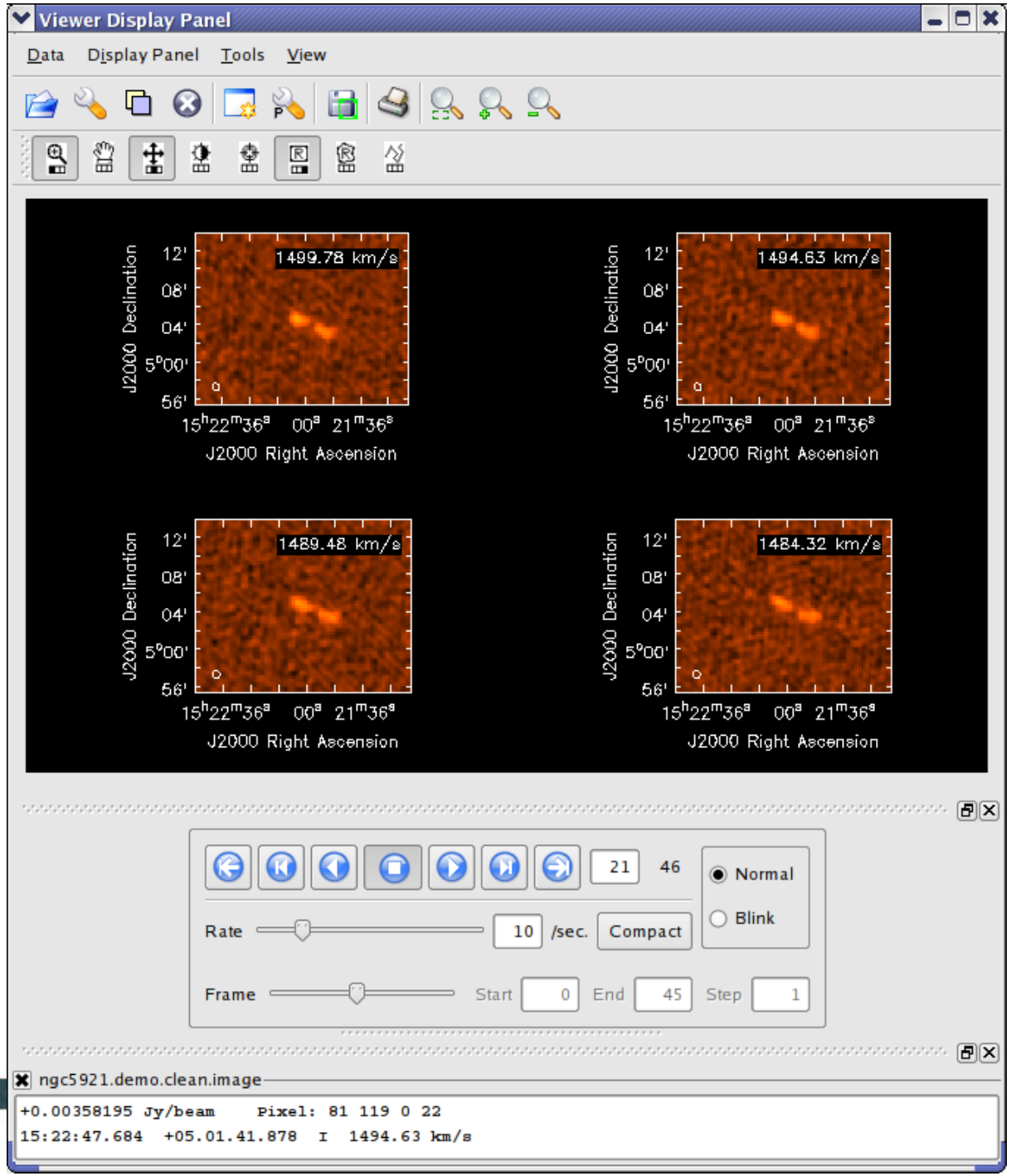


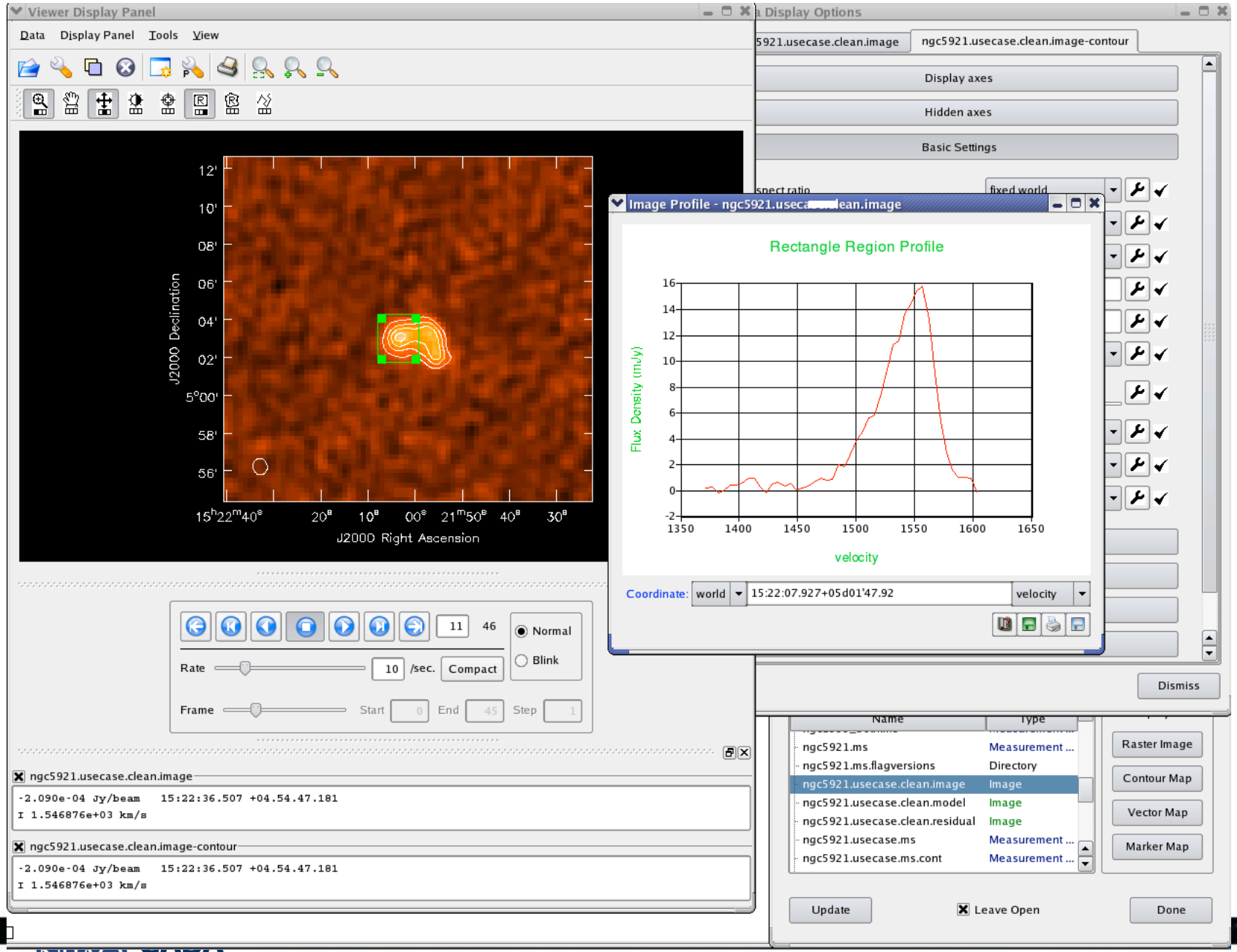
Image Viewer

- Displaying cubes
- Movies
- Channel maps

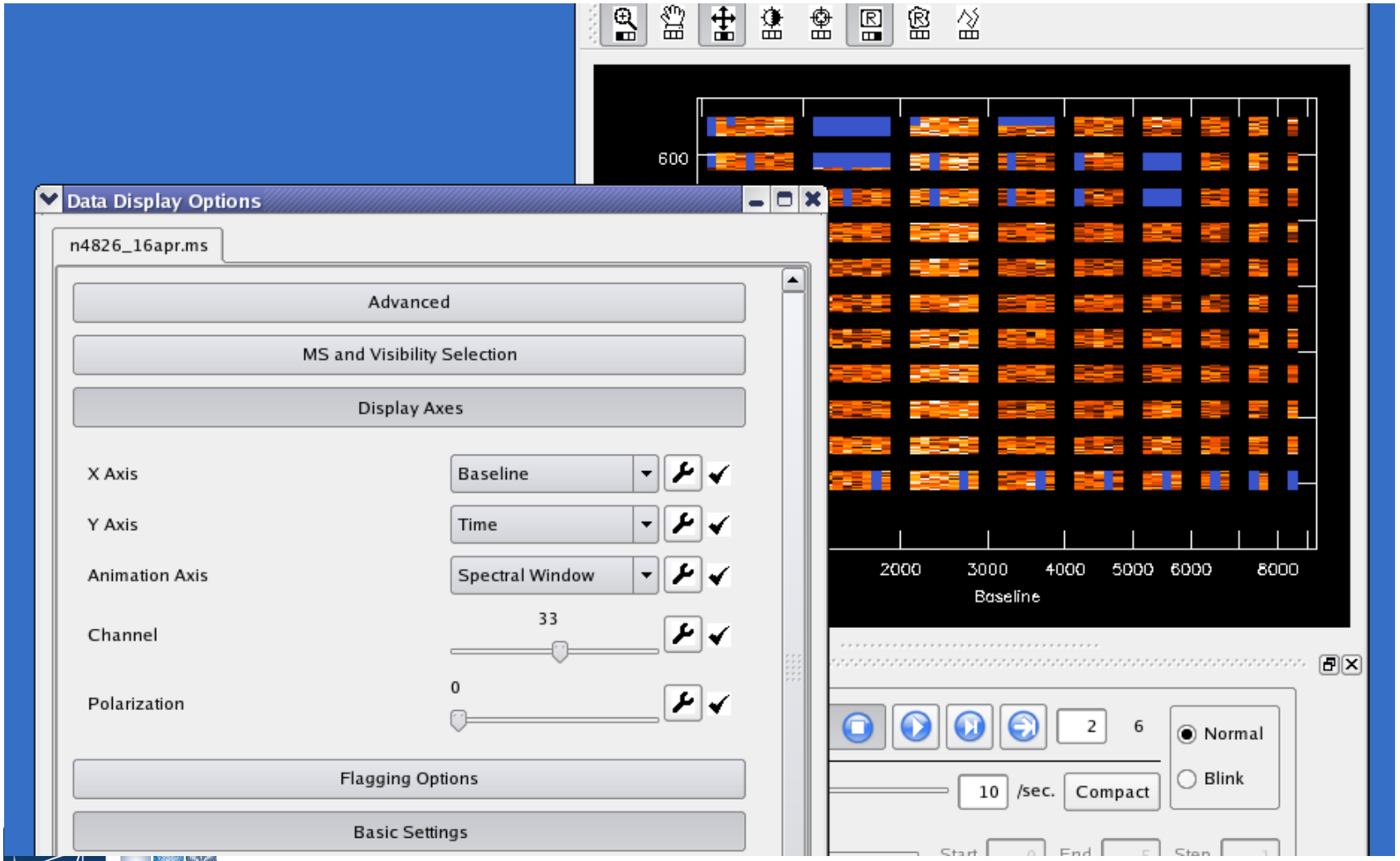




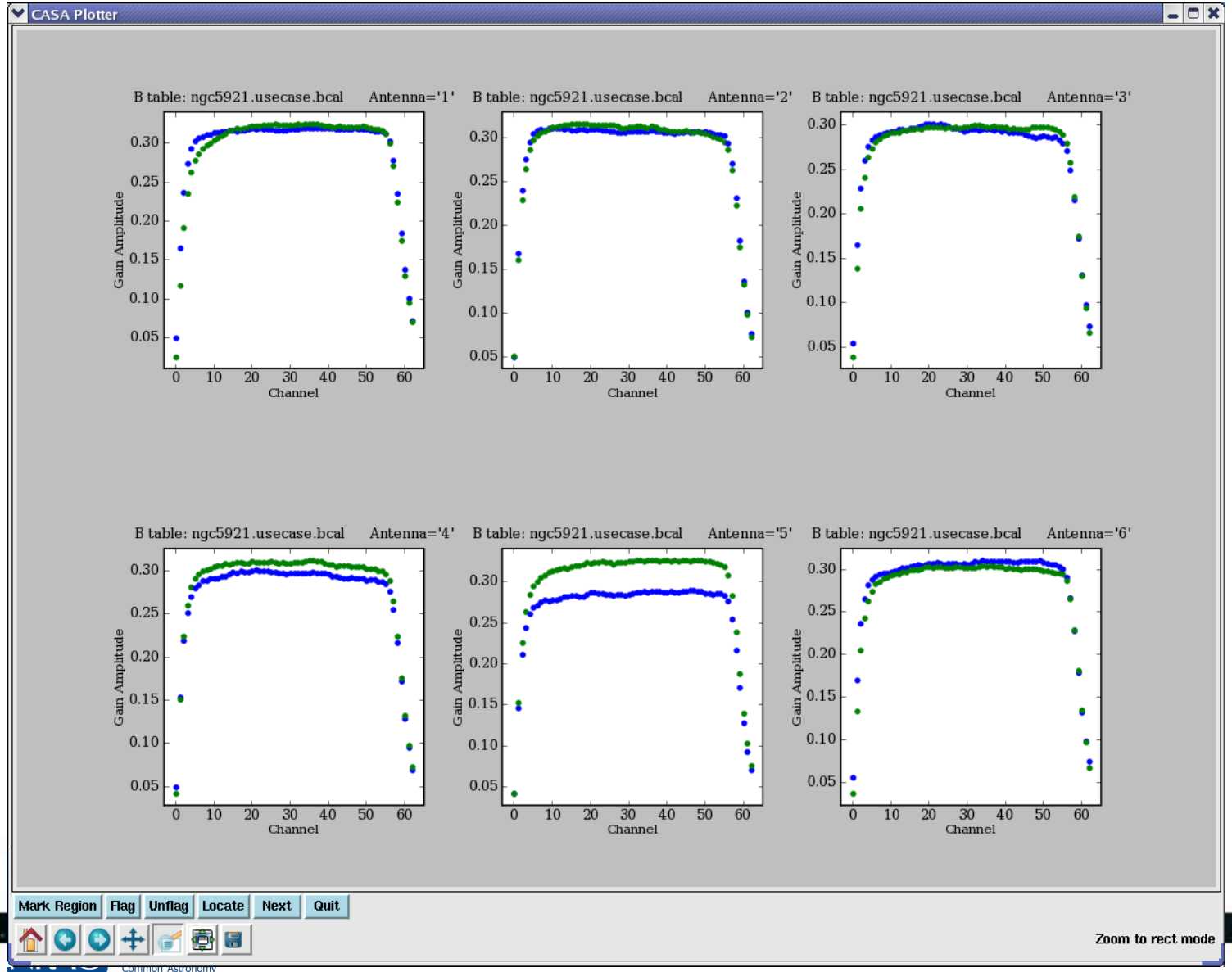
NRAO CASA Common Astronomy Software Applications


Viewer Display Panel
Data
Display Panel
Tools
View
J2000 Declination
J2000 Right Ascension
Image Profile - ngc5921.usecase.clean.image
Rectangle Region Profile
Flux Density (mJy)
velocity
Coordinate:
world
15:22:07.927+05d01'47.92
velocity
Display Options
ngc5921.usecase.clean.image
ngc5921.usecase.clean.image-contour
Display axes
Hidden axes
Basic Settings
fixed world
Normal
Blink
Rate
/sec.
Compact
Frame
Start
End
Step
ngc5921.usecase.clean.image
-2.090e-04 Jy/beam 15:22:36.507 +04.54.47.181
I 1.546876e+03 km/s
ngc5921.usecase.clean.image-contour
-2.090e-04 Jy/beam 15:22:36.507 +04.54.47.181
I 1.546876e+03 km/s
Name
Type
ngc5921.ms
Measurement ...
ngc5921.ms.flagversions
Directory
ngc5921.usecase.clean.image
Image
ngc5921.usecase.clean.model
Image
ngc5921.usecase.clean.residual
Image
ngc5921.usecase.ms
Measurement ...
ngc5921.usecase.ms.cont
Measurement ...
Raster Image
Contour Map
Vector Map
Marker Map
Dismiss
Update
Leave Open
Done

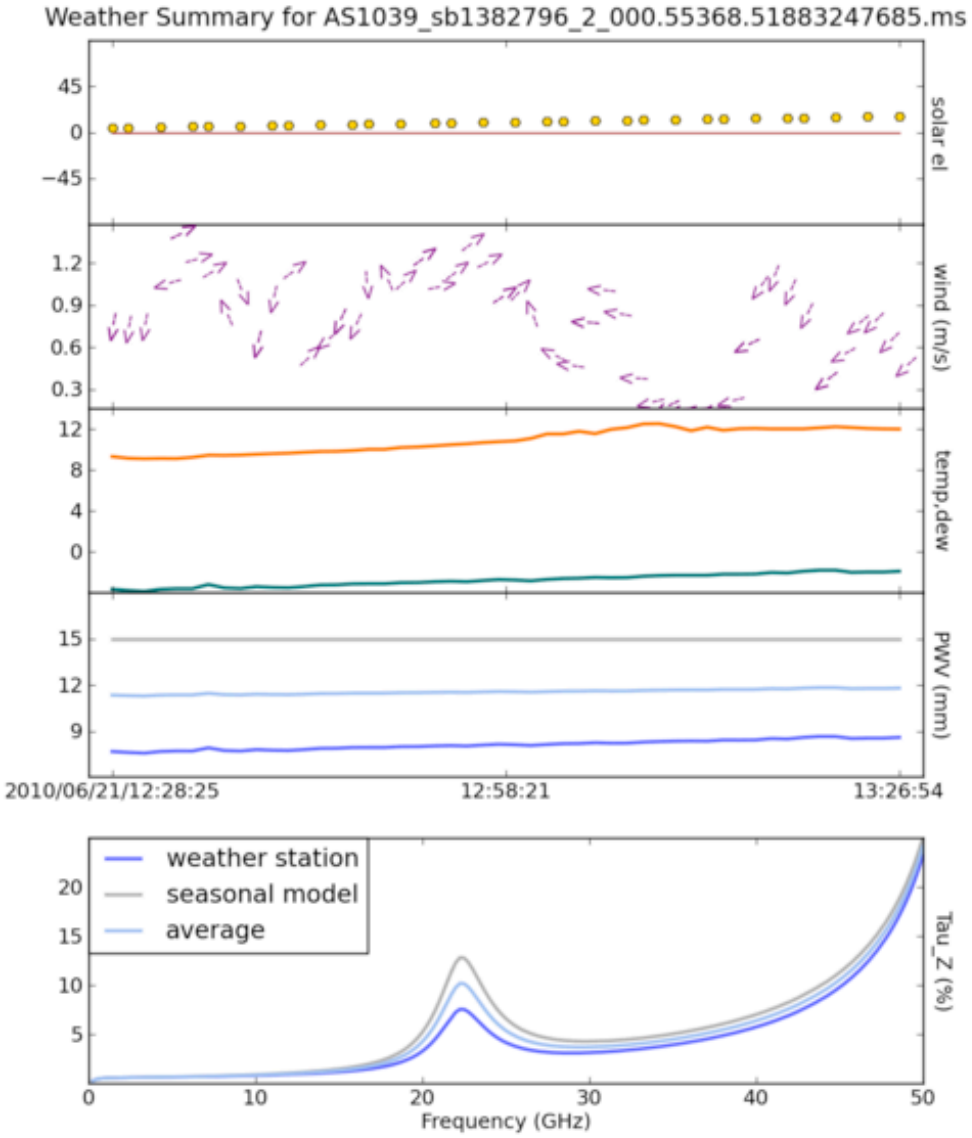
MSViewer


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

Anything - matplotlib



Data Selection Example

- standard selection parameters
  - e.g. for task gaincal:

```
CASA <14>: inp
---------> inp()
#  gaincal :: Determine temporal gains from calibrator observations:

vis                 = 'ngc5921.ms'      #   Name of input visibility file
caltable            = 'ngc5921.gcal'    #   Name of output calibration table
field               =       '0,1'       #   field names or index of calibrators ''==>all
spw                 =     '0:2~56'      #   spectral window:channels: ''==>all
selectdata          =        True       #   Other data selection parameters
     timerange      =          ''       #   time range: ''==>all
     uvrange        =          ''       #   uv range''=all
     antenna        =          ''       #   antenna/baselines: ''==>all
     scan           =          ''       #   scan numbers
     msselect       =          ''       #   Optional data selection (Specialized. but see help)
```

Data Selection Syntax

- see Chapter 2.5 of Cookbook
  - field - string with source name or field ID
    - can use '*' as wildcard, first checks for name, then ID
    - example: field = '1331+305' ; field = '3C*' ; field = '0,1,4~5'
  - spw - string with specwindow ID plus channels
    - use ':' as separator of spw from optional channelization
    - use '^' as separator of channels from step/width
    - example: spw = '0~2' ; spw = '1:10~30' ; spw = '2~5:5~54^5'

Selection Syntax

- see Chapter 2.5 of Cookbook
  - antenna - string with antenna name or ID
    - first check for name, then ID (beware VLA name 1-27, ID 0-26)
    - example: antenna = '1~5,11' ; antenna = 'EA*', '!VA'
    - Baselines: 'EA01&EA10'
  - timerange - string with date/time range
    - specify 'T0~T1' , missing parts of T1 default to T0, can give 'T0+dT'
    - example: timerange = '2007/10/16/01:00:00~06:30:00'

Calibration

- Data structure: 3 columns (scratch columns):
- DATA column (raw data)
- MODEL (source model, e.g. extended flux calibrators, selfcal model)
- CORRECTED_DATA calibrated data
- Columns are created when needed, this may take some time
- Calibration is with calibration tables, e.g. bandpass, gain, pol, antenna offset etc., and they are applied multiplicative (apply all previous calibration tables to create new one or to be applied to data)

Getting User Support

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

  Cookbook, online reference, download, example scripts

  - CASAguides.nrao.edu

  For data reduction tutorials, tips, tricks, …

  - "Helpdesk" at help.nrao.edu

    Submit questions, suggestions, bugs (needs my.nrao.edu registration)

  - CASA mailing lists: casa-announce, casa-users

  - User's forum in the future

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
- CASAguides Wiki:
  - http://casaguides.nrao.edu
- Python:
  - http://python.org/doc (e.g., see Tutorial for novices)
- IPython:
  - http://ipython.scipy.org/moin/Documentation
- matplotlib:
  - http://matplotlib.sourceforge.net/

Single Dish Data Reduction with CASA

Kana Sugimoto (ALMA project, NAOJ)

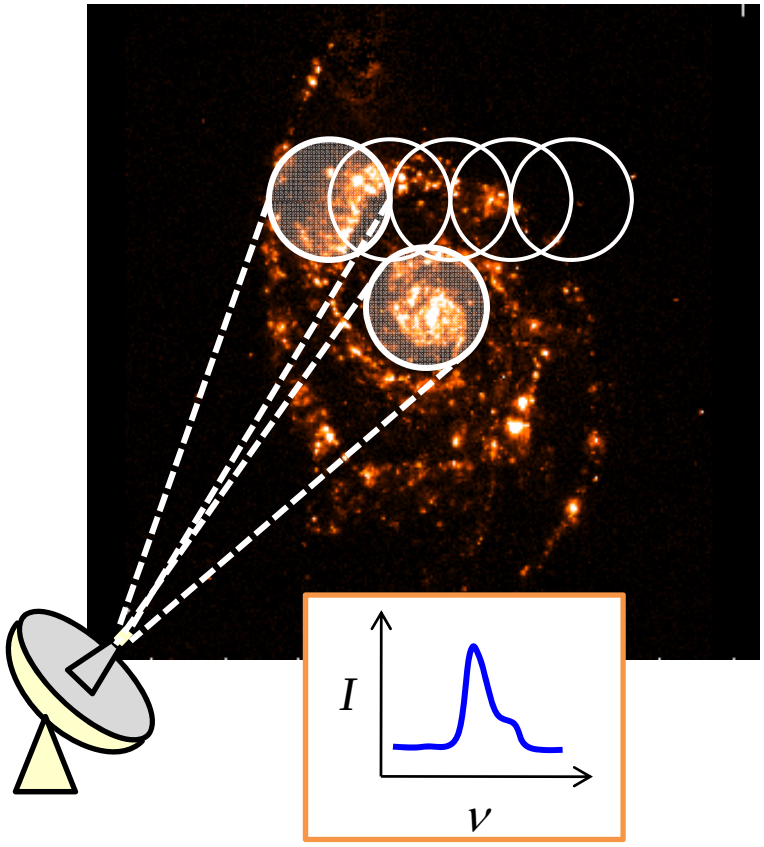
Give a first touch to CASA single dish data reduction package!

Contents

- ✓ Brief summary of single dish observation and data reduction
- ✓ Single dish data reduction with CASA

Brief summary of single dish observations

Single dish telescopes measure a total flux in their beam for each observation point of sky.



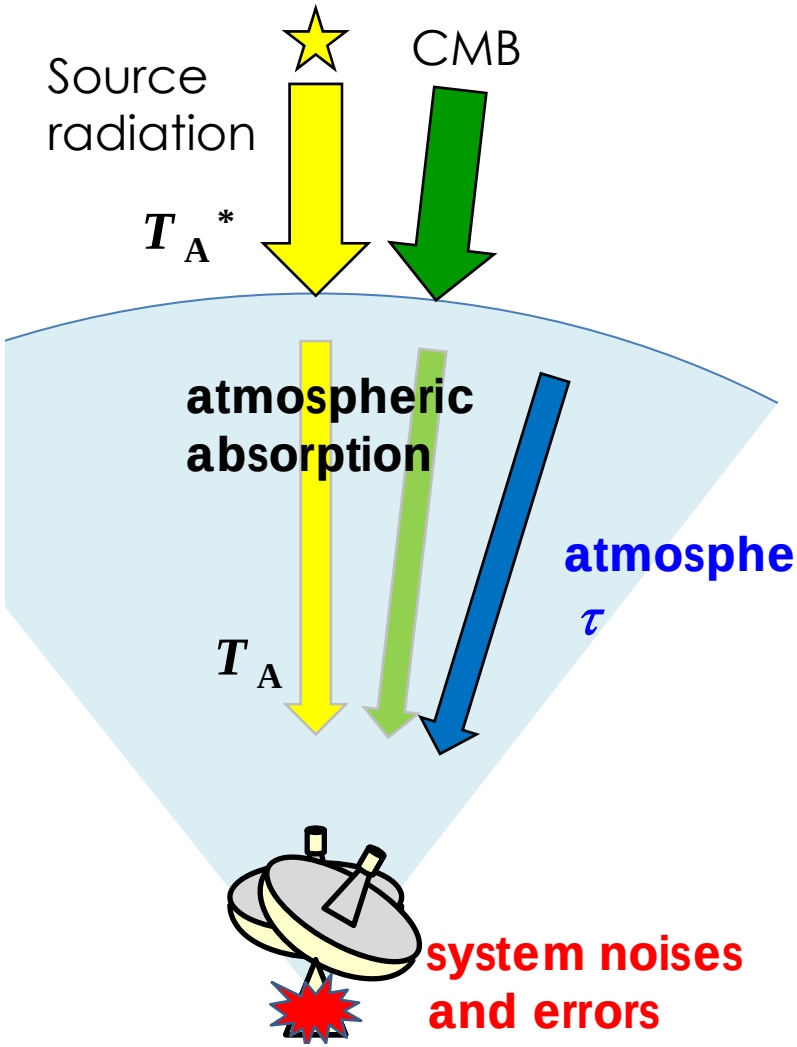
- continuum observation
- spectral observation
- polarization

<u>Imaging</u>
change antenna directions over the map region and measure flux of each direction … pointing observation, on-the-fly observation

Single dish Calibration Basics

Radiation don't come only form sources (atmosphere, CMB, system…) → need **calibration**

Example: position switching



ON source signal

$$V_{ON} = V_{source}\, e^{-\tau} + V_{CMB}\, e^{-\tau} + V_{atm}(1 - e^{-\tau}) + V_{sys}$$

OFF source signal

$$V_{OFF} = V_{CMB}\, e^{-\tau} + V_{atm}(1 - e^{-\tau}) + V_{sys}$$

position switching, frequency switching, nutator switching, etc.

The brightness temperature of a source

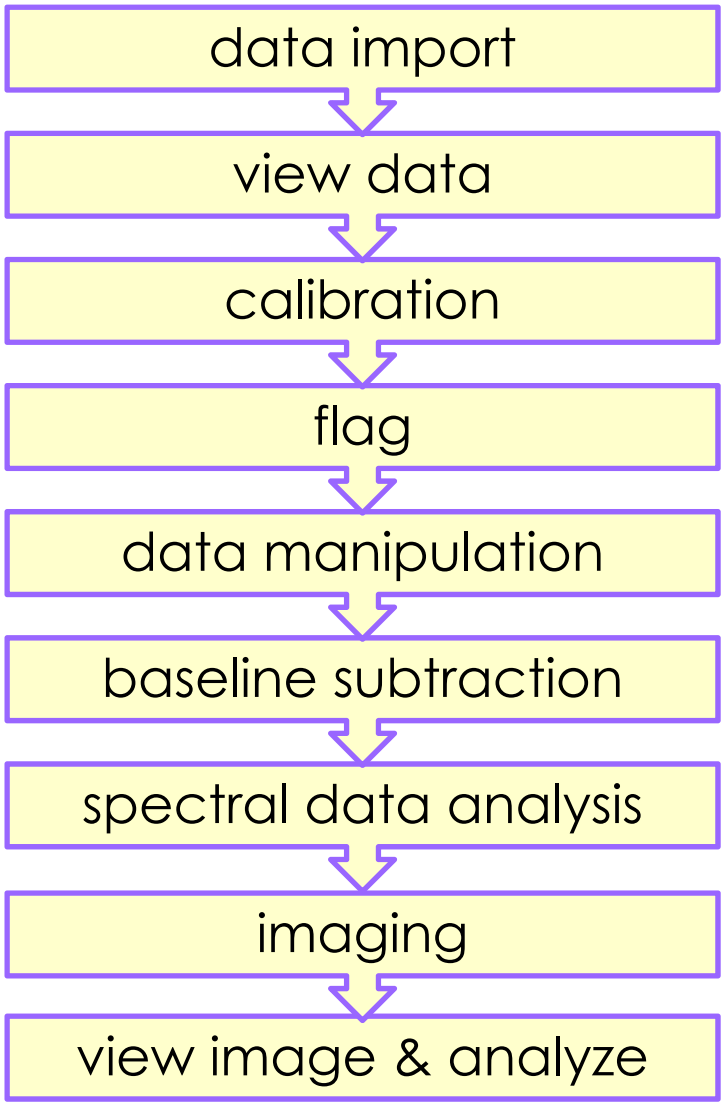
$$T_A = T_{CAL} \frac{V_{ON} - V_{OFF}}{V_{CAL}}$$

the conversion factor : a signal strength → the brightness temperature

$$T_A^* = \frac{T_A}{\eta\, e^{-\tau}}$$

GBT: $T_{cal} / V_{cal} = T_{sys} / V_{OFF}$

Single dish data reduction steps example



… convert data (internal format)

… observation summary, view data and spectra

… according to observation strategy

… eliminate bad data

… sum up spectra (S/N ↑)

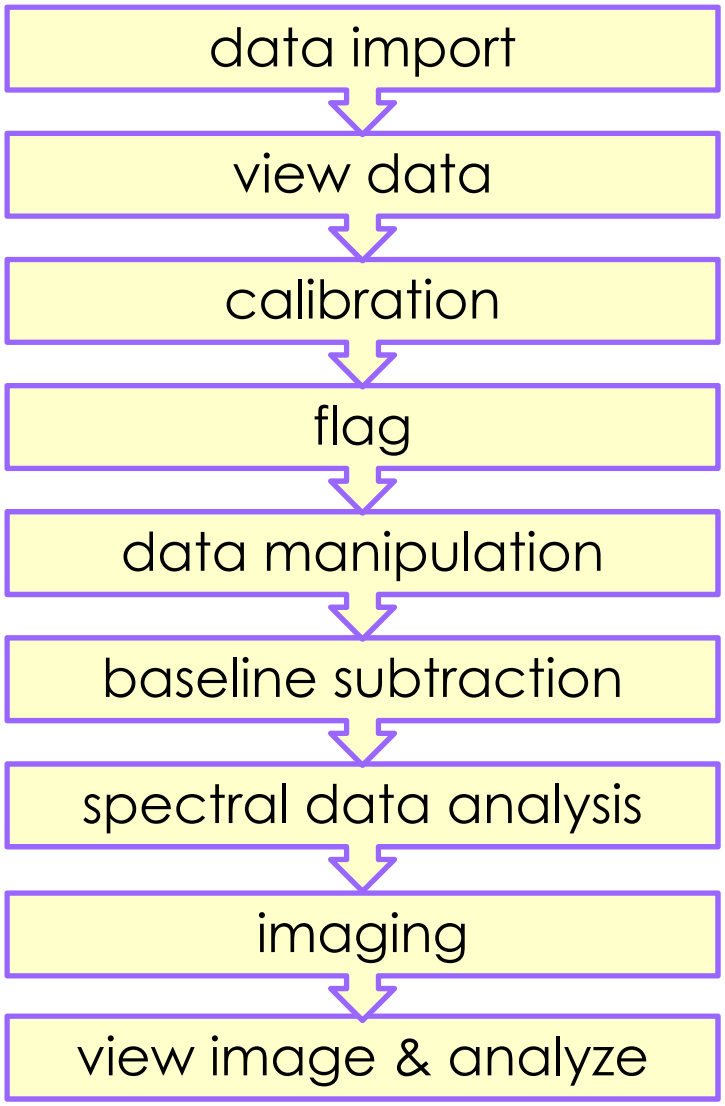
… line observation

… line width, intensity, statistics

… create image

… momentum, etc.

Single dish data reduction with CASA



… sdsave

… sdlist, sdplot, browsetable

… sdaverage

… sdflag

… sdaverage

… sdbaseline

… sdfit, sdstat

… sdimaging

… viewer, etc.

Orion SiO (reduction/)

FLS3a HI (imaging/)

Tutorial data

Spectral line data reduction (reduction/)

✓ Data file: OrionS_rawACSmod
✓ Target: Orion S
✓ GBT: 45GHz (Spws=8, 8192channels, XX YY)
✓ Observation: position-switching (4 scans/spw: ON-OFF-ON-OFF)
✓ Analysis script: orion_SiO.py (reduce IF15 = SiO line)

Spectral Imaging (imaging/)

✓ Data file: fls3a_HI.asap (calibrated data)
✓ Field: FLS3a (Galactic HI mapping)
✓ GBT: HI line @1.4GHz (Spws=2, 1024channels, XX YY)
✓ Observation: 6 x 3deg (grid : 0.1x 0.05 deg) mapping
✓ Analysis script: fls3a.py (generate image of IF0)

Step 0! Import single dish package

First of all, import **ASAP**, the single dish reduction package in CASA

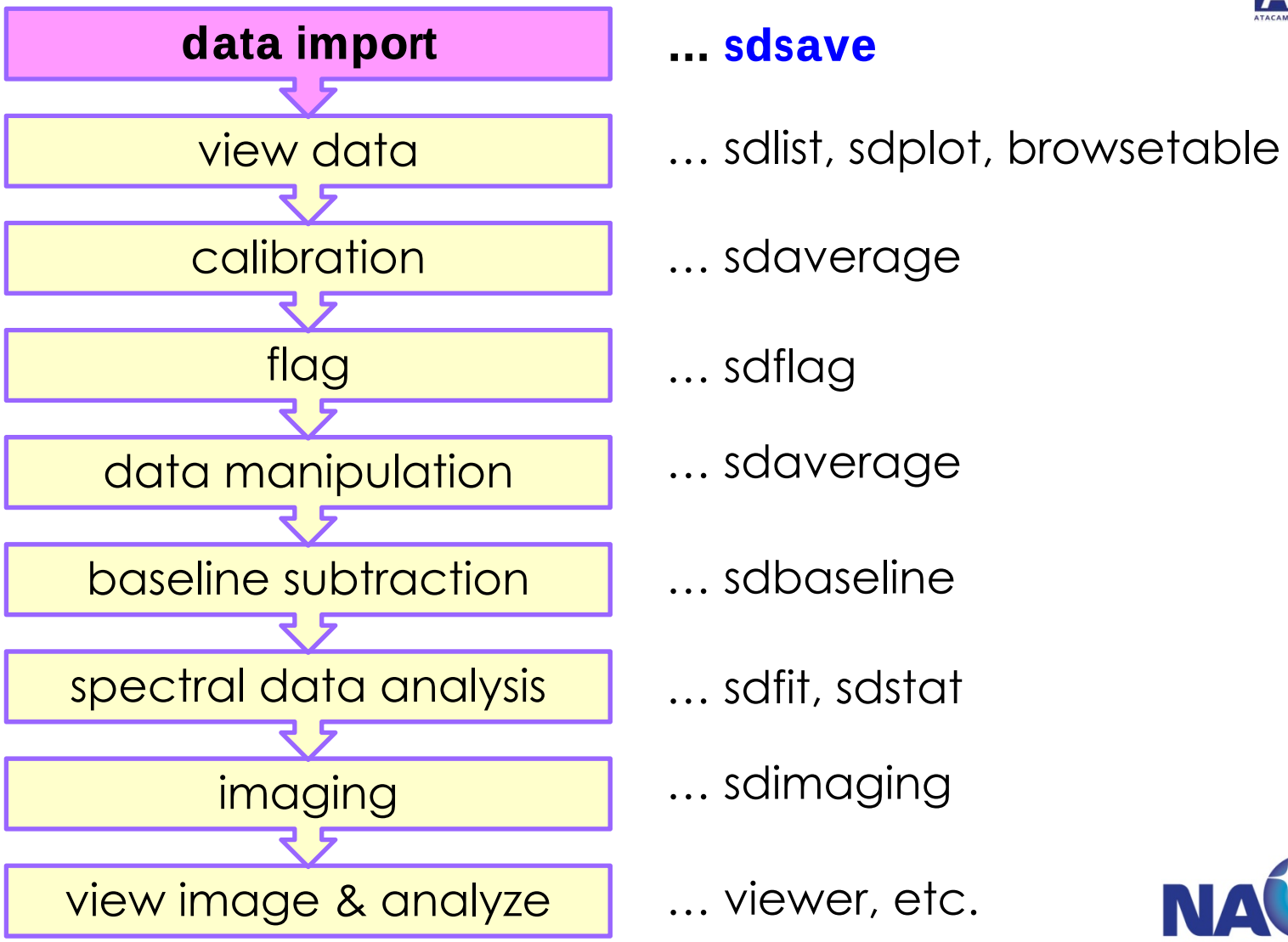
```
CASA<>:  asap_init()
```

Single dish spectral analysis functions in CASA are based on ASAP (ATNF Spectral Analysis Package) originally developed in Australia Telescope National Facility.

+ Many enhancements to analyze data from **ALMA** and the other telescopes such as **NRO 45m**, **ASTE**, **GBT**, and **ATNF**.

So far, **you should explicitly import the package to CASA**.

Single dish data reduction with CASA



sdsave - import / export data

Transform data format

- input: SDFITS, rpfits, NOSTAR, NEWSTAR , MS, scantable
- output: SDFITS, ASCII , MS, scantable

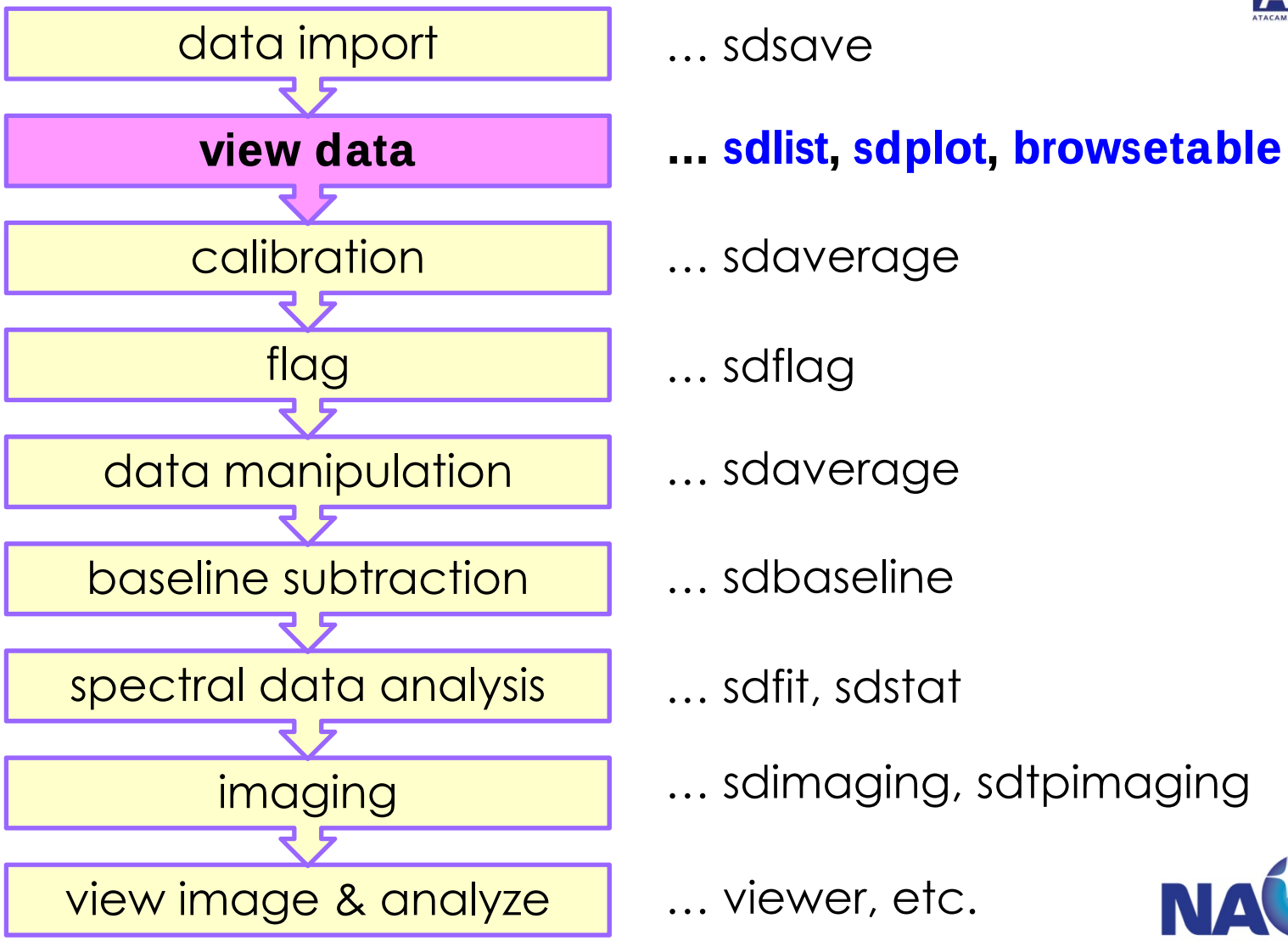
scantable … internal data format of ASAP

ターミナル — ssh — 96×17

```
#  sdsave :: ASAP SD task: save the sd spectra in various format
sdfile              = 'OrionS_rawACSmod' #  name of input SD dataset
antenna             =          0         #  antenna name or id (only effective for MS
                                         #   input)
rowlist             =         []         #  list of row numbers to process (e.g. [0,2,4,6])
scanlist            =         []         #  list of scan to process (e.g. [20,21,22,23])
field               =         ''         #  string for selection by source name
iflist              =         []         #  list of IF ids to select (e.g. [0,1])
pollist             =         []         #  list of polarization ids to select (e.g. [0,1])
scanaverage         =      False         #  average integs within scans (True,False)
timeaverage         =      False         #  average scans over time (True,False)
polaverage          =      False         #  average over polarizations (True,False)
outfile             = 'orion_SiO.asap'   #  output file name
outform             =     'ASAP'         #  output file format (ASCII,MS2,SDFITS,ASAP)
overwrite           =       True         #  overwrite the output file if already exists
async               =      False         #  If true the taskname must be started using
                                         #   sdsave(...)
```

National Astronomical Observatory of Japan

Single dish data reduction with CASA



sdlist – list observation

View scantable summary

ターミナル — ssh — 96×9

```
#  sdlist :: ASAP SD task: list summary of single dish data
sdfile                = 'orion_Si0.asap'  #  name of input SD dataset
antenna               =          0        #  antenna name or id (only effective for MS input)
scanaverage           =      False        #  average integs within scans (True,False)
listfile              =         ''        #  Name of output file for summary list
overwrite             =      False        #  overwrite the output file if already exists
async                 =      False        #  If true the taskname must be started using
                                          #   sdlist(...)
```

Output → CASA logger

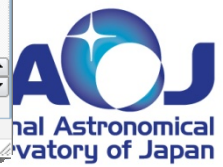
Log Messages (ana03:/home/sugimtkn/workcasa/singledish/reduction/casapy.log)

File Edit View

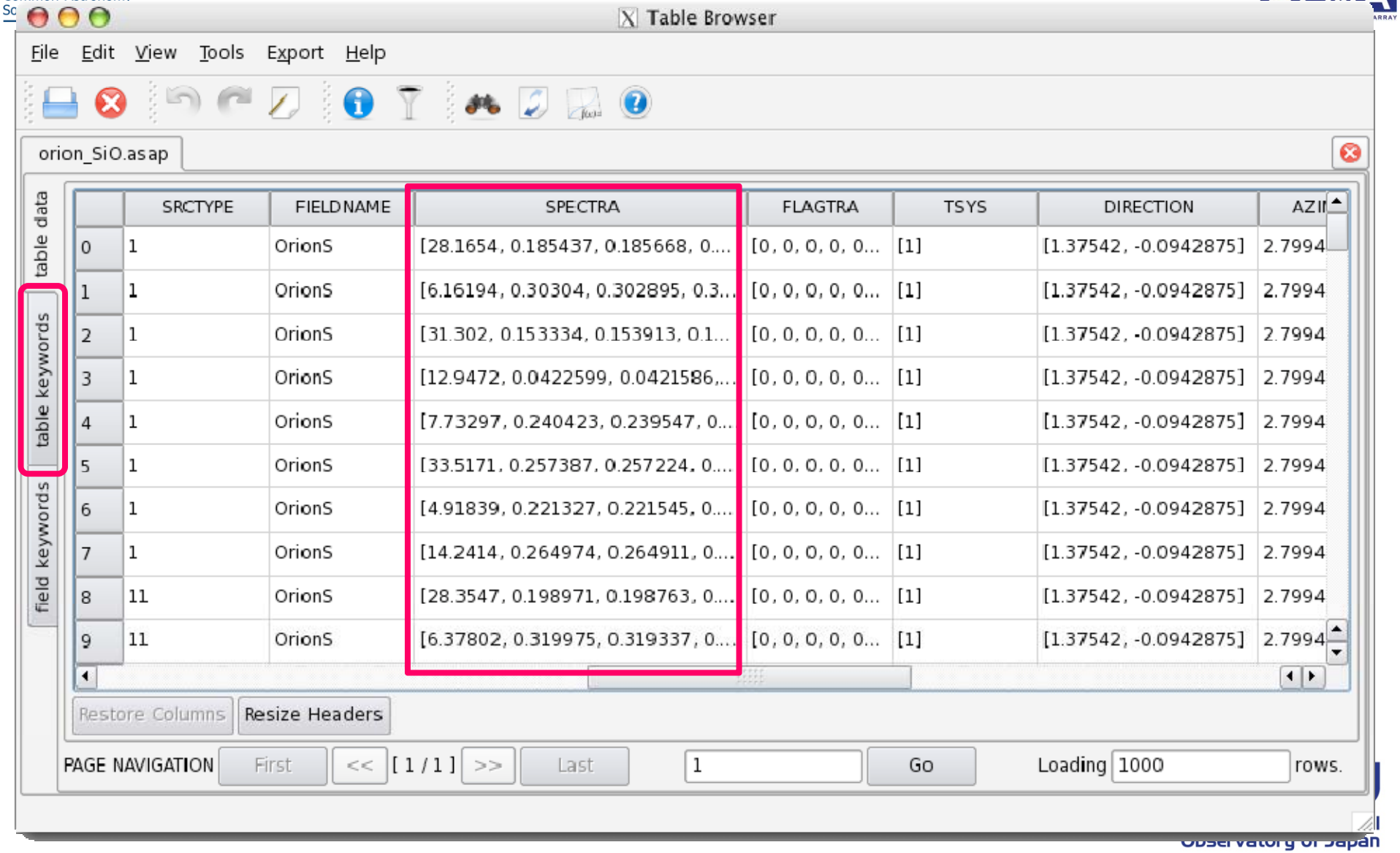
Search Message: Filter: Time

Priority	Origin	Message
INFO	sdlist…	----------
INFO	sdlist…	Scan Table Summary
INFO	sdlist…	----------
INFO	sdlist…	Beams: 1
INFO	sdlist…	IFs: 26
INFO	sdlist…	Polarisations: 2 (linear)
INFO	sdlist…	Channels: 8192
INFO	sdlist…	Observer: Joseph McMullin
INFO	sdlist…	Obs Date: 2006/01/19/01:45:58
INFO	sdlist…	Project: AGBT06A_018_01
INFO	sdlist…	Obs. Type: OffOn:PSWITCHOFF:TPWCAL
INFO	sdlist…	Antenna Name: GBT
INFO	sdlist…	Flux Unit:
INFO	sdlist…	Rest Freqs: [4.5490258e+10] [Hz]
INFO	sdlist…	Abcissa: Channel
INFO	sdlist…	Selection: none
INFO		sdlist::::casa+
INFO	sdlist…	Scan Source Time Integration
INFO	sdlist…	Beam Position (J2000)
INFO	sdlist…	IF Frame RefVal RefPix Increment Channels
INFO	sdlist…	----------
INFO	sdlist…	20 OrionS 01:45:58 4 x 30.0s
INFO	sdlist…	0 05:15:13.5 -05.24.08.2
INFO	sdlist…	0 LSRK 4.5489354e+10 4096 6104.233 8192
INFO	sdlist…	1 LSRK 4.5300785e+10 4096 6104.233 8192
INFO	sdlist…	2 LSRK 4.4074929e+10 4096 6104.233 8192
INFO	sdlist…	3 LSRK 4.4166215e+10 4096 6104.233 8192
INFO	sdlist…	21 OrionS 01:48:38 4 x 30.0s
INFO	sdlist…	0 05:35:13.5 -05.24.08.2
INFO	sdlist…	0 LSRK 4.5489354e+10 4096 6104.233 8192
INFO	sdlist…	1 LSRK 4.5300785e+10 4096 6104.233 8192
INFO	sdlist…	2 LSRK 4.4074929e+10 4096 6104.233 8192
INFO	sdlist…	3 LSRK 4.4166215e+10 4096 6104.233 8192
INFO	sdlist…	22 OrionS 01:51:21 4 x 30.0s
INFO	sdlist…	0 05:15:13.5 -05.24.08.2
INFO	sdlist…	0 LSRK 4.5489354e+10 4096 6104.233 8192
INFO	sdlist…	1 LSRK 4.5300785e+10 4096 6104.233 8192
INFO	sdlist…	2 LSRK 4.4074929e+10 4096 6104.233 8192

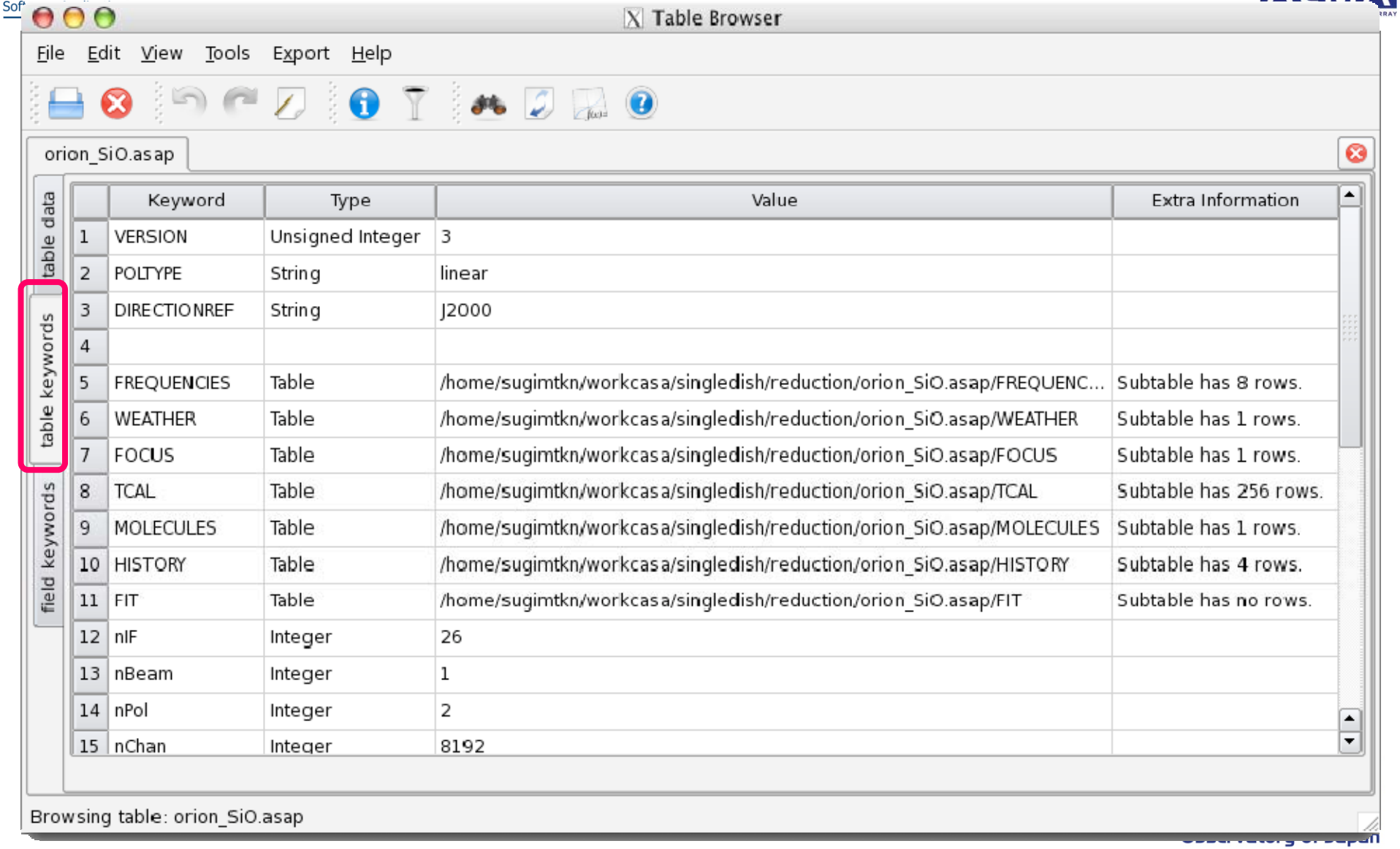
Insert Message: Lock scroll



browsetable – view scantable (table data)

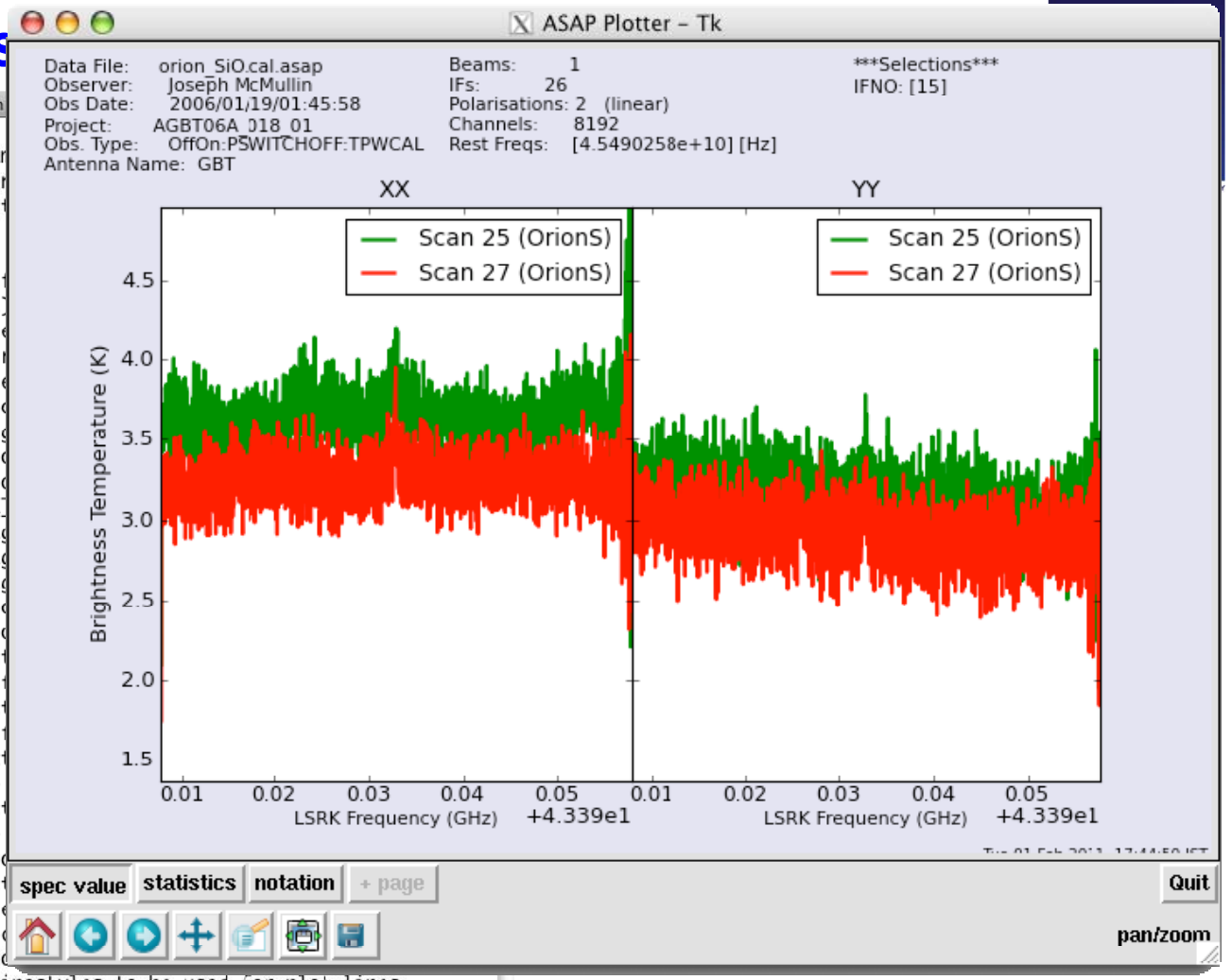


browsetable – view scantable (table data)



Orion spectra (calibrated)

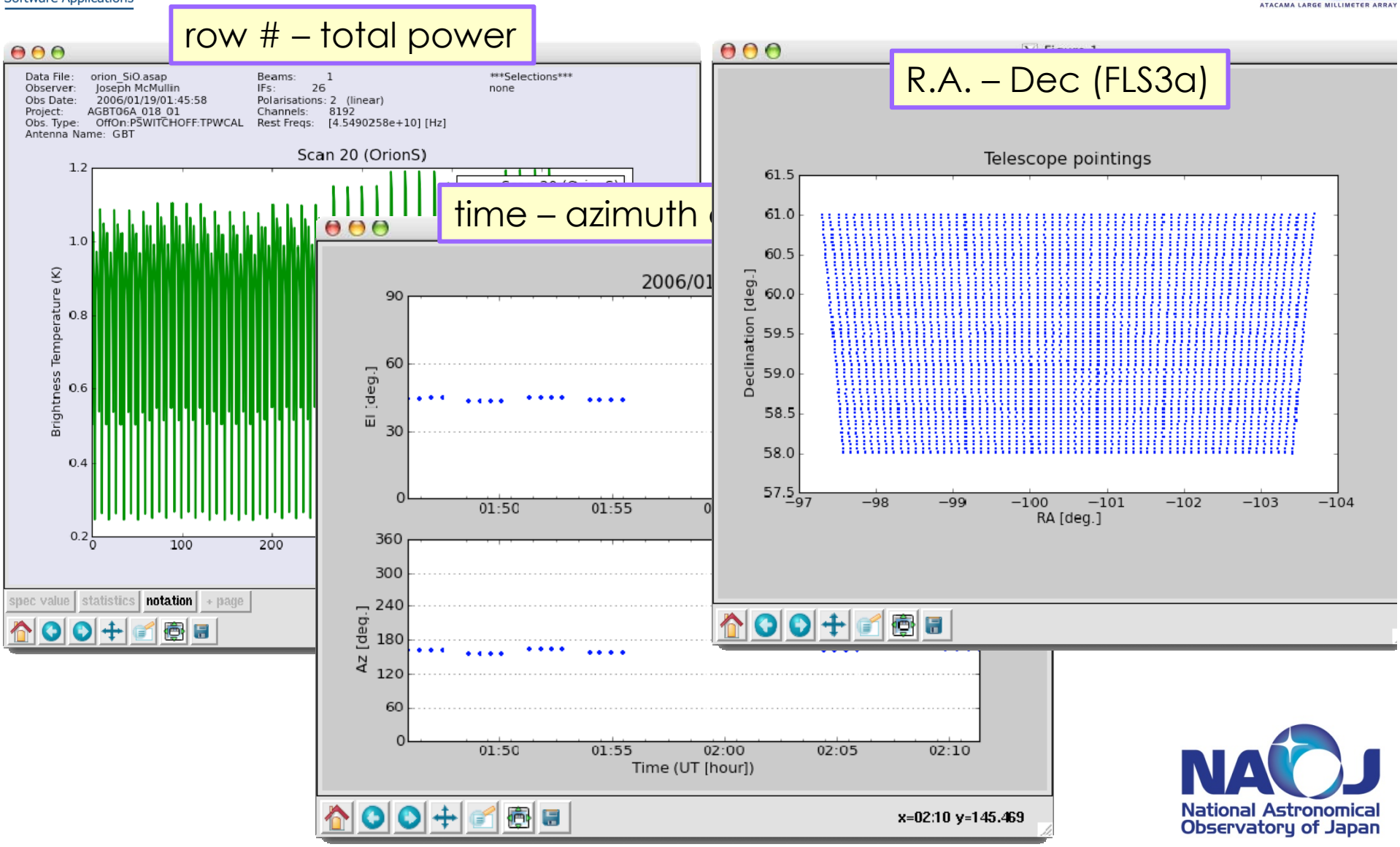
```
#  sdplot :: ASAP SD plotting task
sdfile          = 'orion_SiO.cal.asap' #  nam
antenna         =          0        #  antenn
                                    #   input
fluxunit        =         ''        #  units
specunit        =       'GHz'       #  units
restfreq        =         ''        #  rest f
                                    #   unit)
frame           =         ''        #  freque
                                    #   (=cur
doppler         =         ''        #  dopple
scanlist        =         []        #  list o
field           =         ''        #  string
iflist          =       [15]        #  list o
pollist         =         []        #  list o
                                    #   [0,1]
scanaverage     =       True        #  averag
timeaverage     =      False        #  averag
polaverage      =      False        #  averag
kernel          =     'none'        #  type o
plottype        =  'spectra'        #  type o
                                    #   point
     stack      =        's'        #  code t
                                    #   plott
     panel      =        'p'        #  code t
                                    #   spect
     frrange    =         []        #  range
                                    #   plott
     sprange    =         []        #  range
     linecat    =     'none'        #  contro
                                    #   plott
     linedop    =        0.0        #  dopple
                                    #   (spec
     colormap   =     'none'        #  the co
     linestyles =     'none'        #  the linestyles to be used for plot lines
     linewidth  =          2        #  width of plotted lines
     histogram  =      False        #  plot histogram

header          =       True        #  print header information on the plot
     headsize   =         10        #  header fontsize

plotstyle       =      False        #  customise plot settings
plotfile        =         ''        #  file name for hardcopy output
overwrite       =      False        #  overwrite the output file if already exists
async           =      False        #  If true the taskname must be started using
                                    #   sdplot(...)
```



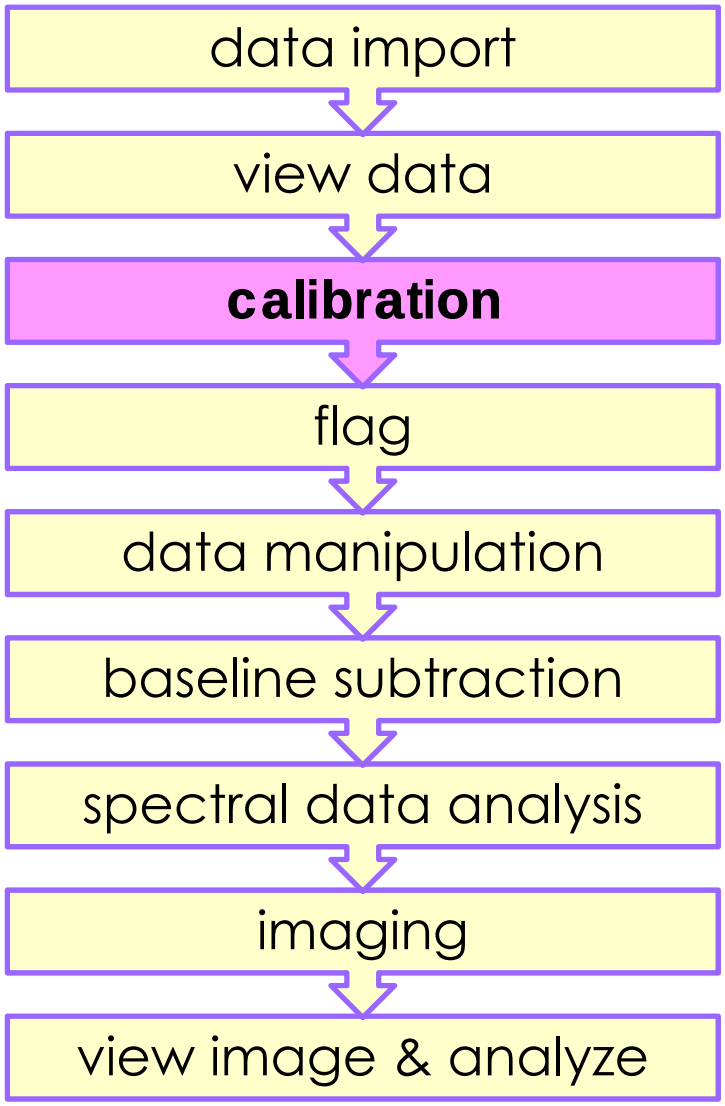
National Astronomical Observatory of Japan

sdplot – plot scantable

plottype = 'totalpower', 'azel', 'pointing'



Single dish data reduction with CASA



… sdsave

… sdlist, sdplot, browsetable

… sdaverage

… sdflag

… sdaverage

… sdbaseline

… sdfit, sdstat

… sdimaging

… viewer, etc.

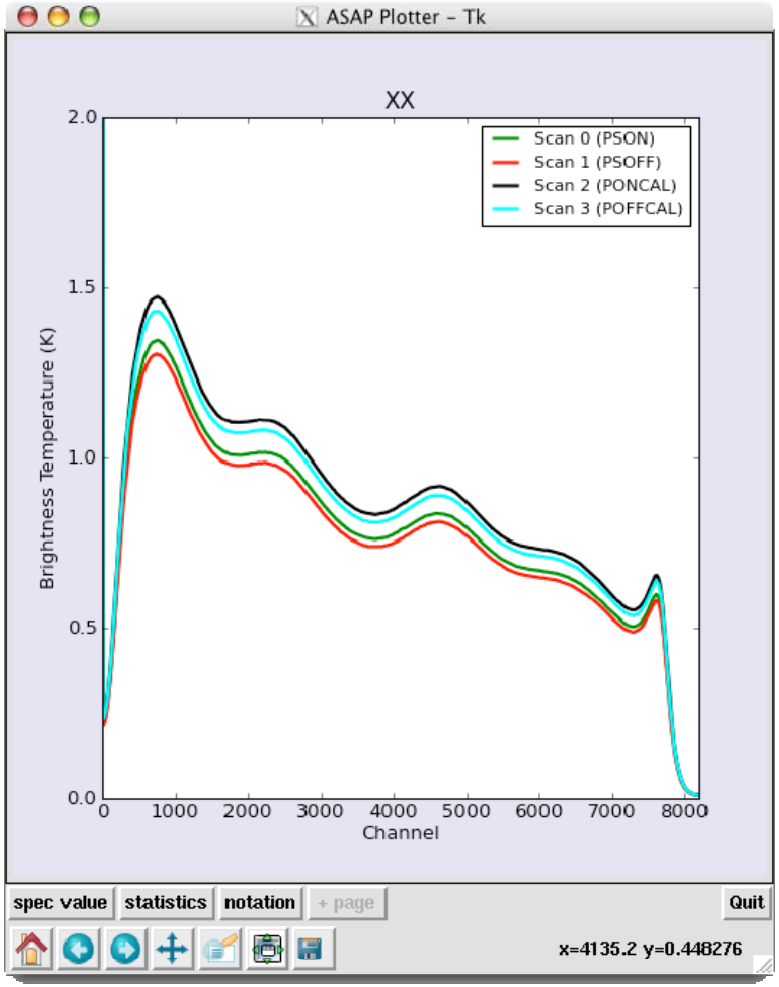
sdaverage – calibration

ターミナル — ssh — 94×27

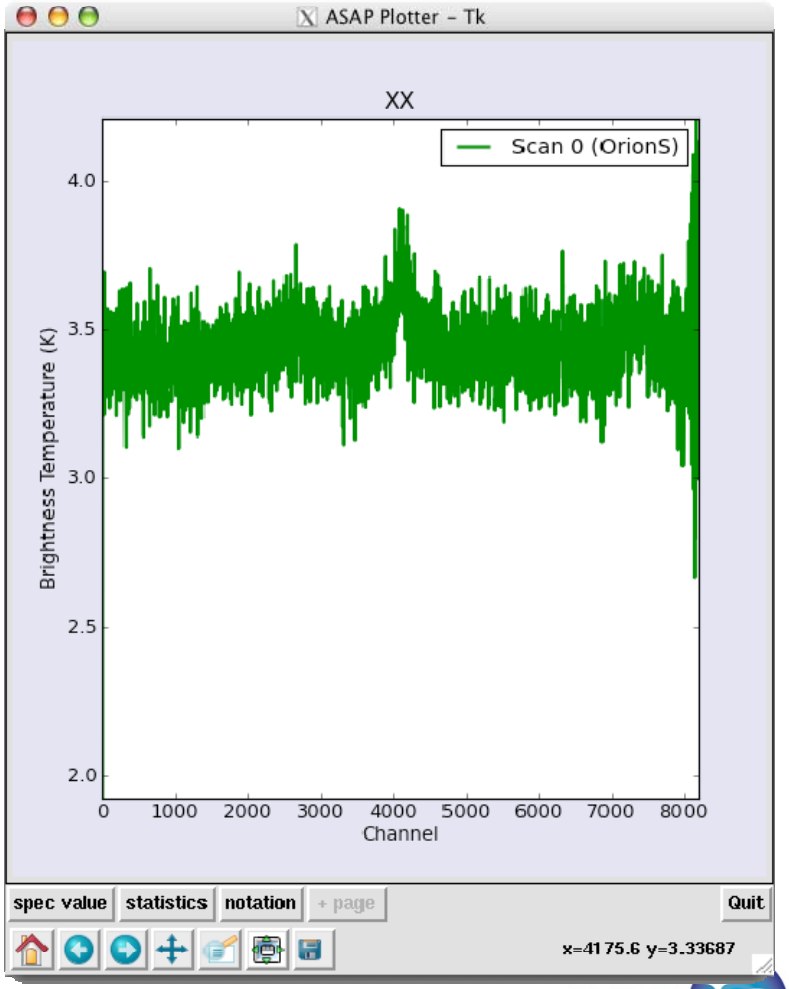
```
#  sdaverage :: ASAP SD task: do data selection, calibration, and averaging
sdfile              = 'orion_SiO.asap'  #  name of input SD dataset
antenna             =          0        #  antenna name or id (only effective for MS input)
fluxunit            =        'K'        #  units for line flux (K,Jy) (=current)
     telescopeparm  =         ''        #  param of telescope for flux conversion

specunit            =         ''        #  units for spectral axis (channel,km/s,GHz,=current)
frame               =         ''        #  frequency reference frame, e.g. LSRK (=current)
doppler             =         ''        #  doppler convention, e.g. RADIO (=current)
calmode             =       'ps'        #  SD calibration mode (ps,nod,fs,fsotf,none)
scanlist            = [24, 25, 26, 27]  #  list of scans to use (e.g. [1,2,3,4])
field               =         ''        #  string for selection by source name
iflist              =       [15]        #  list of IF ids to select (e.g. [0,1])
pollist             =         []        #  list of polarization ids to select (e.g. [0,1])
channelrange        =         []        #  channel range selection (e.g. [0,5000])
scanaverage         =      False        #  average integs within scans (True,False)
timeaverage         =      False        #  average scans over time (True,False)
polaverage          =      False        #  average over polarizations (True,False)
tau                 =       0.09        #  atmospheric optical depth for correction
verify              =      False        #  verify the results of calibration
outfile             = 'orion_SiO.cal.asap' #  output file name
outform             =    'ASAP'         #  output file format (ASCII,MS,SDFITS,ASAP)
overwrite           =       True        #  overwrite the output file if already exists
plotlevel           =          0        #  plot results (0=none,1+=some,<0=hardcopy)
async               =      False        #  If true the taskname must be started using
                                        #   sdaverage(...)
```

sdaverage – calibration

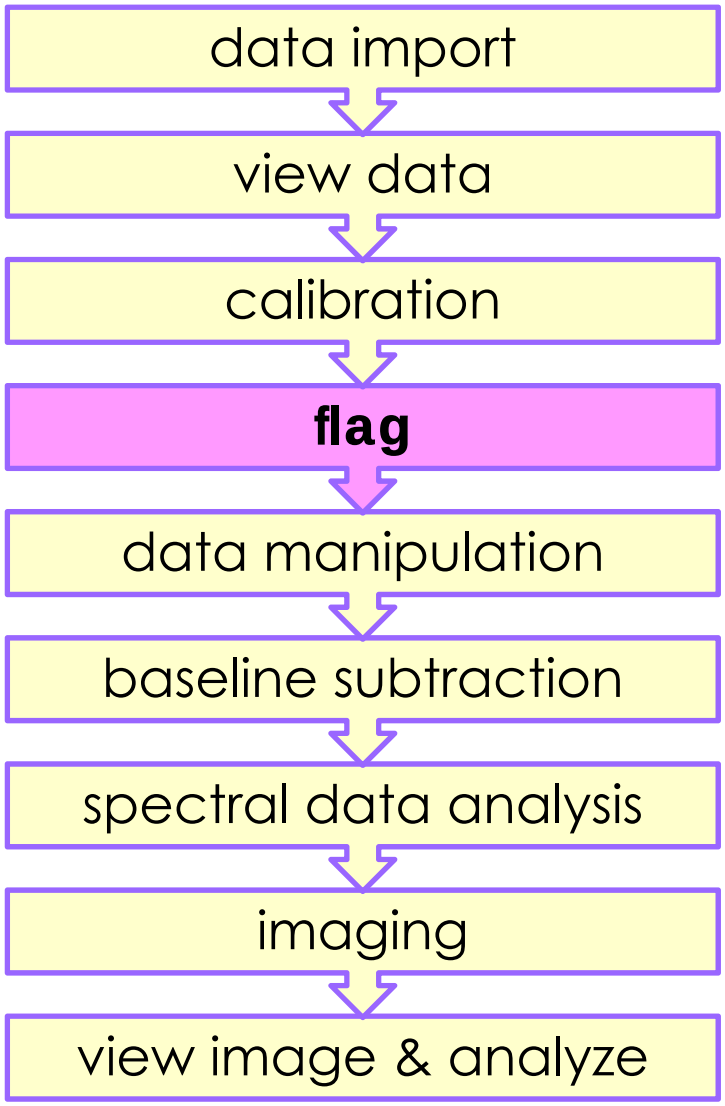
Observation



Calibrated



Single dish data reduction with CASA



… sdsave

… sdlist, sdplot, browsetable

… sdaverage

… sdflag

… sdaverage

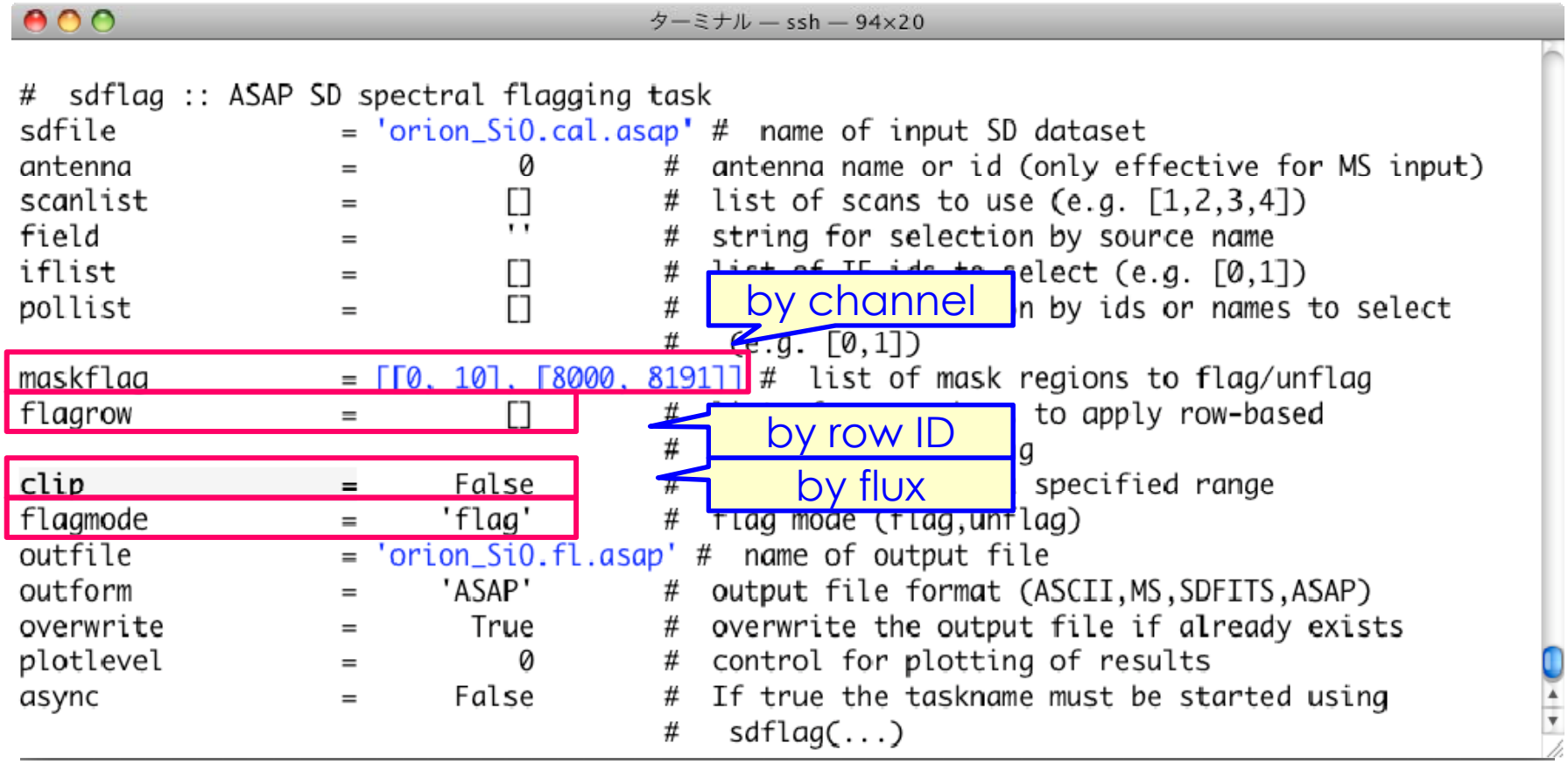
… sdbaseline

… sdfit, sdstat

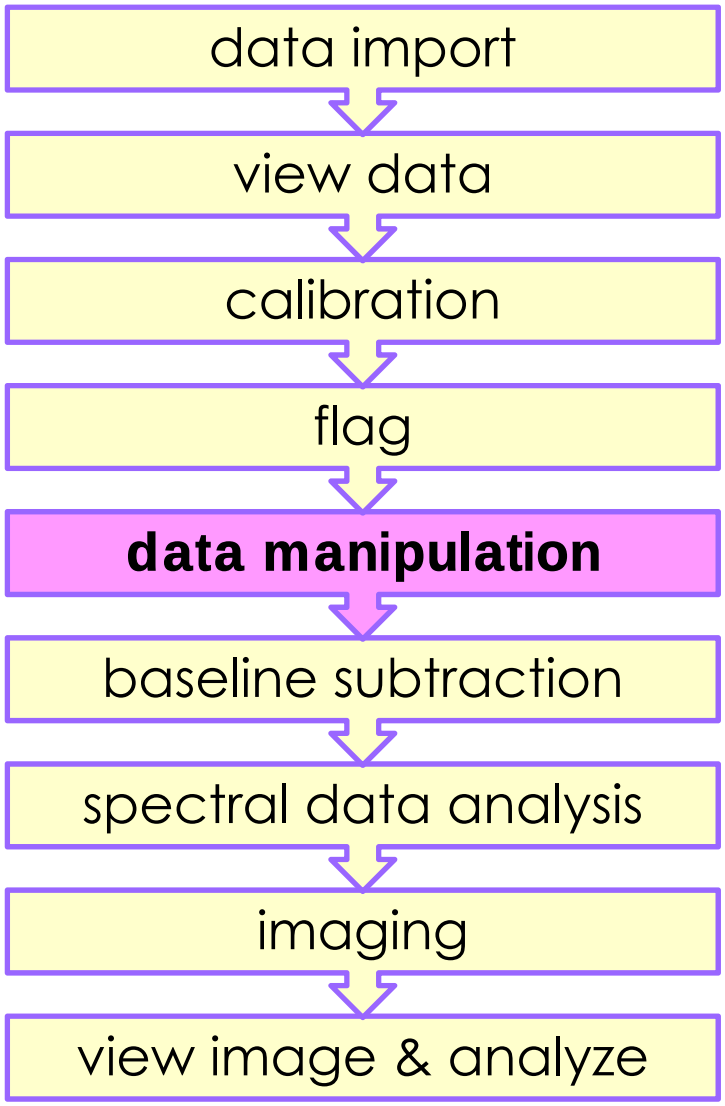
… sdimaging

… viewer, etc.

sdflag – flag out bad data



Single dish data reduction with CASA



… sdsave

… sdlist, sdplot, browsetable

… sdaverage

… sdflag

… sdaverage

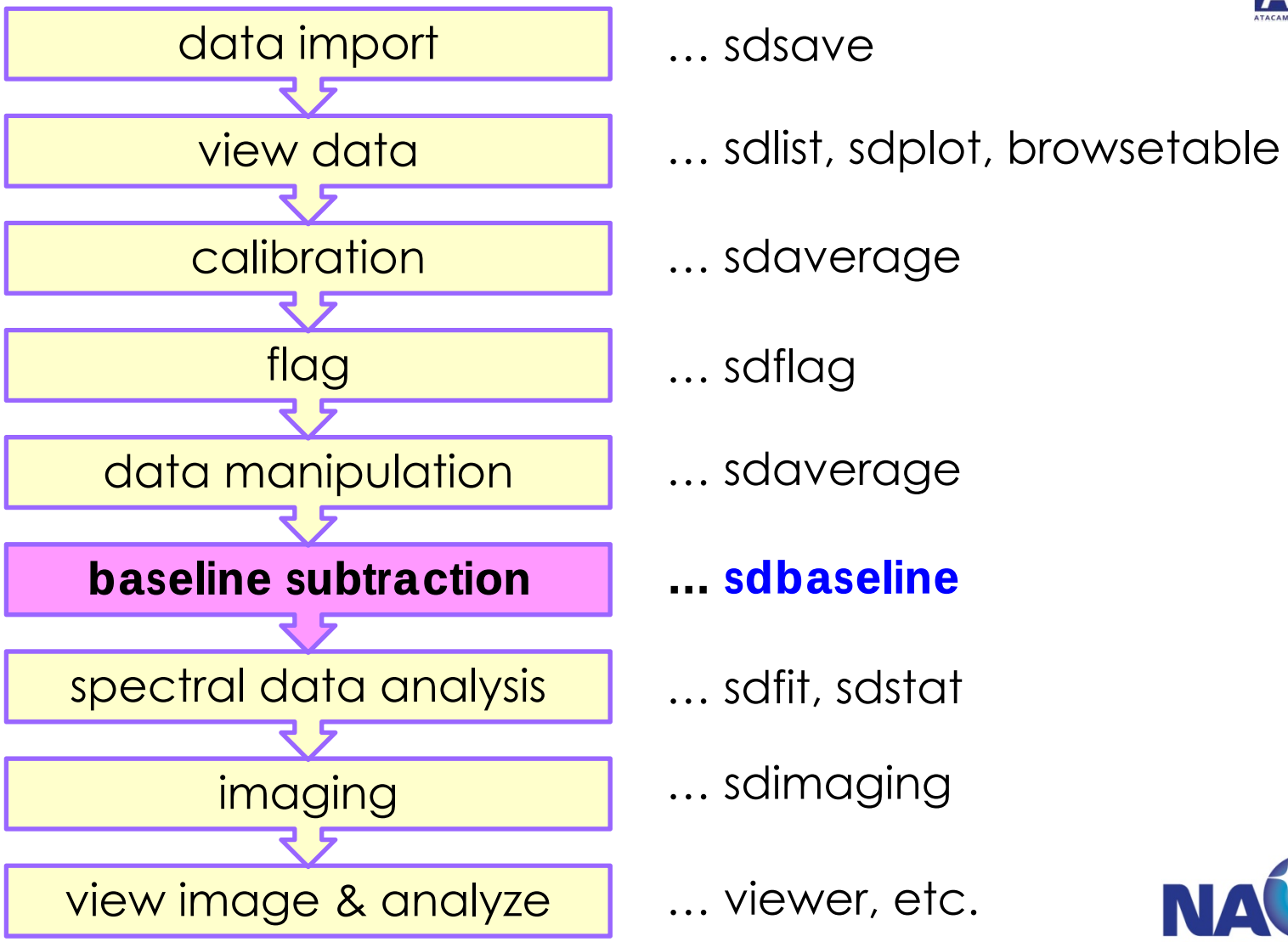
… sdbaseline

… sdfit, sdstat

… sdimaging

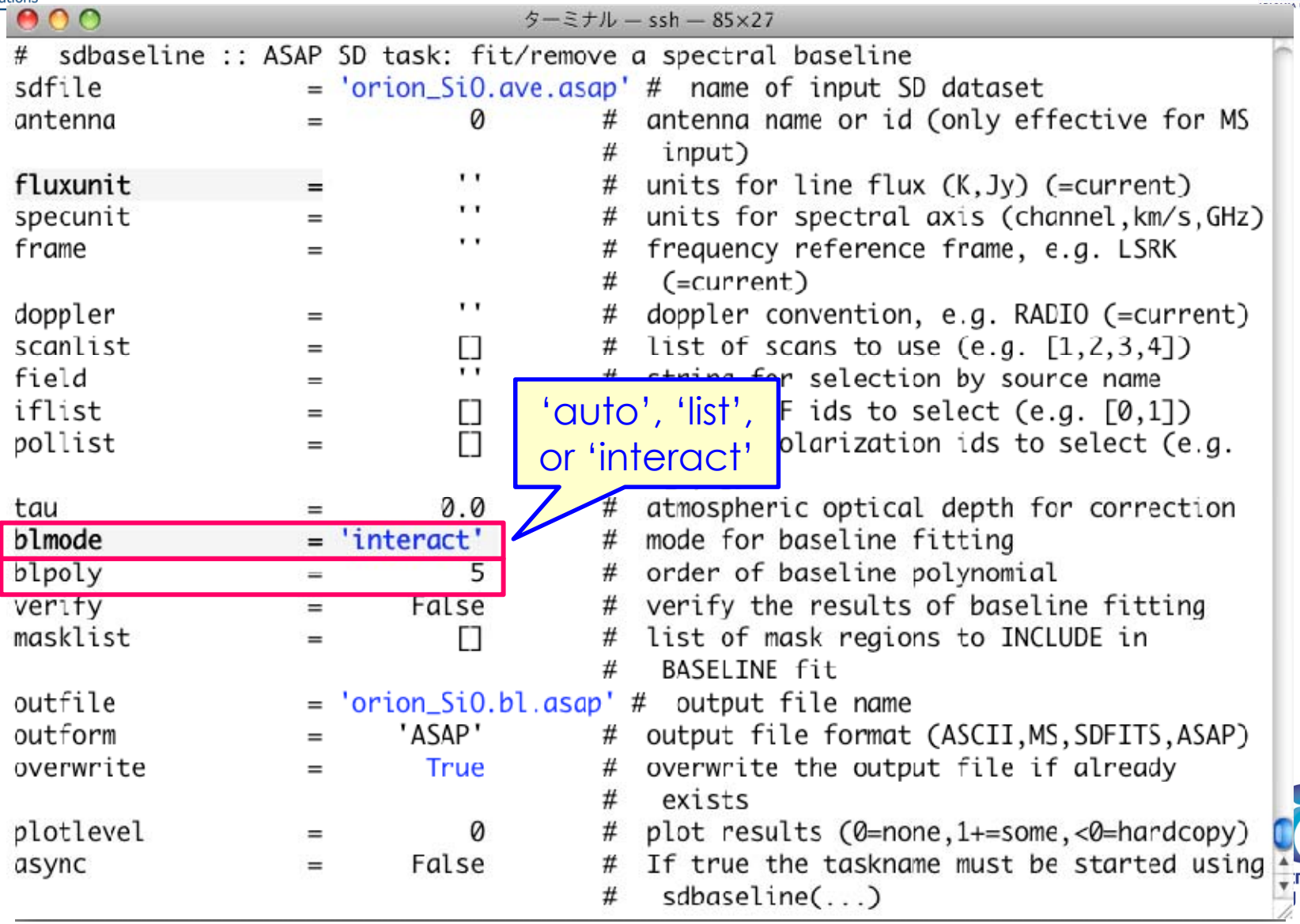
… viewer, etc.

Single dish data reduction with CASA



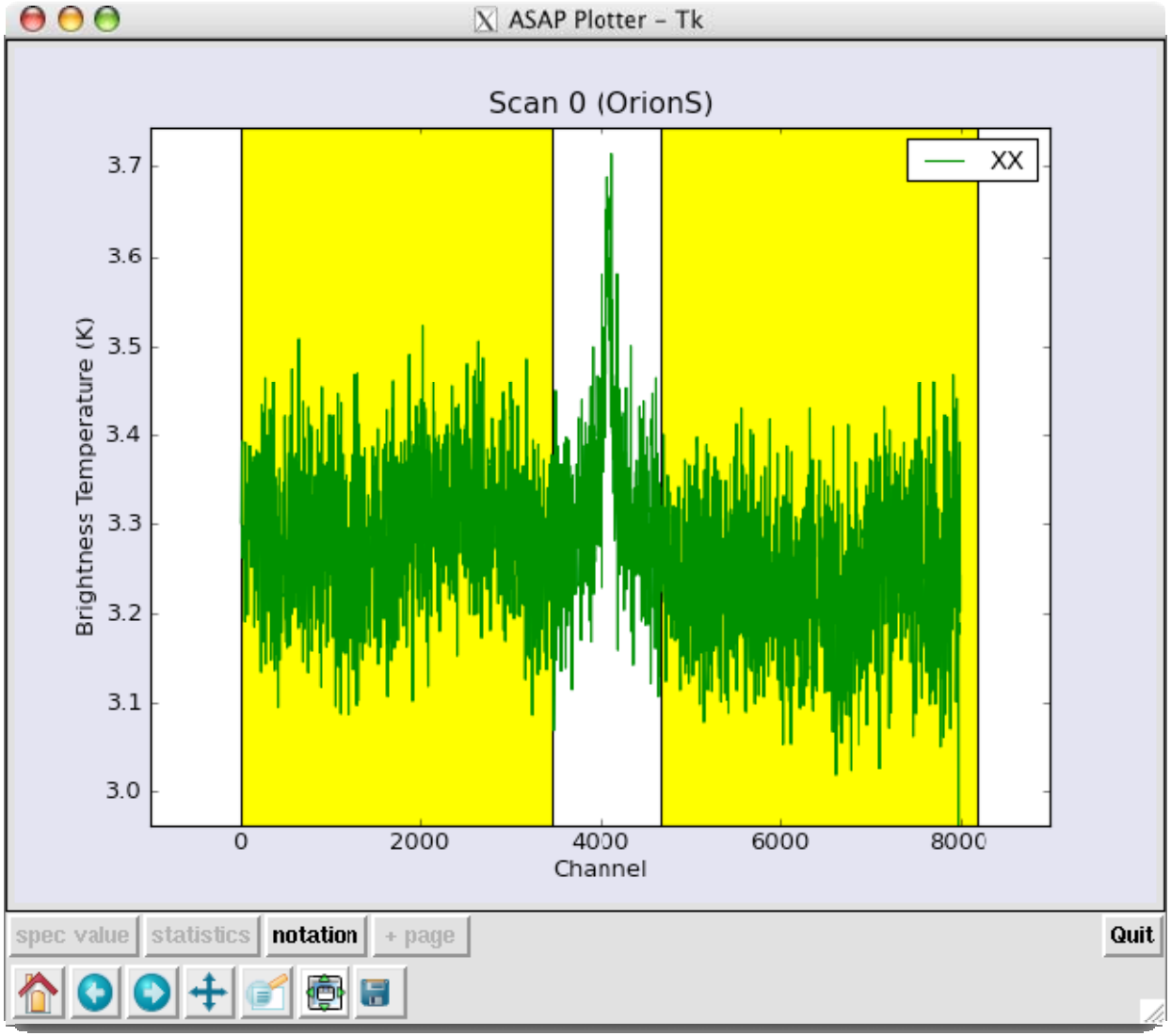
sdbaseline – baseline subtraction

polynominal baseline fitting and subtraction



sdbaseline – baseline subtraction

interactive mask selection



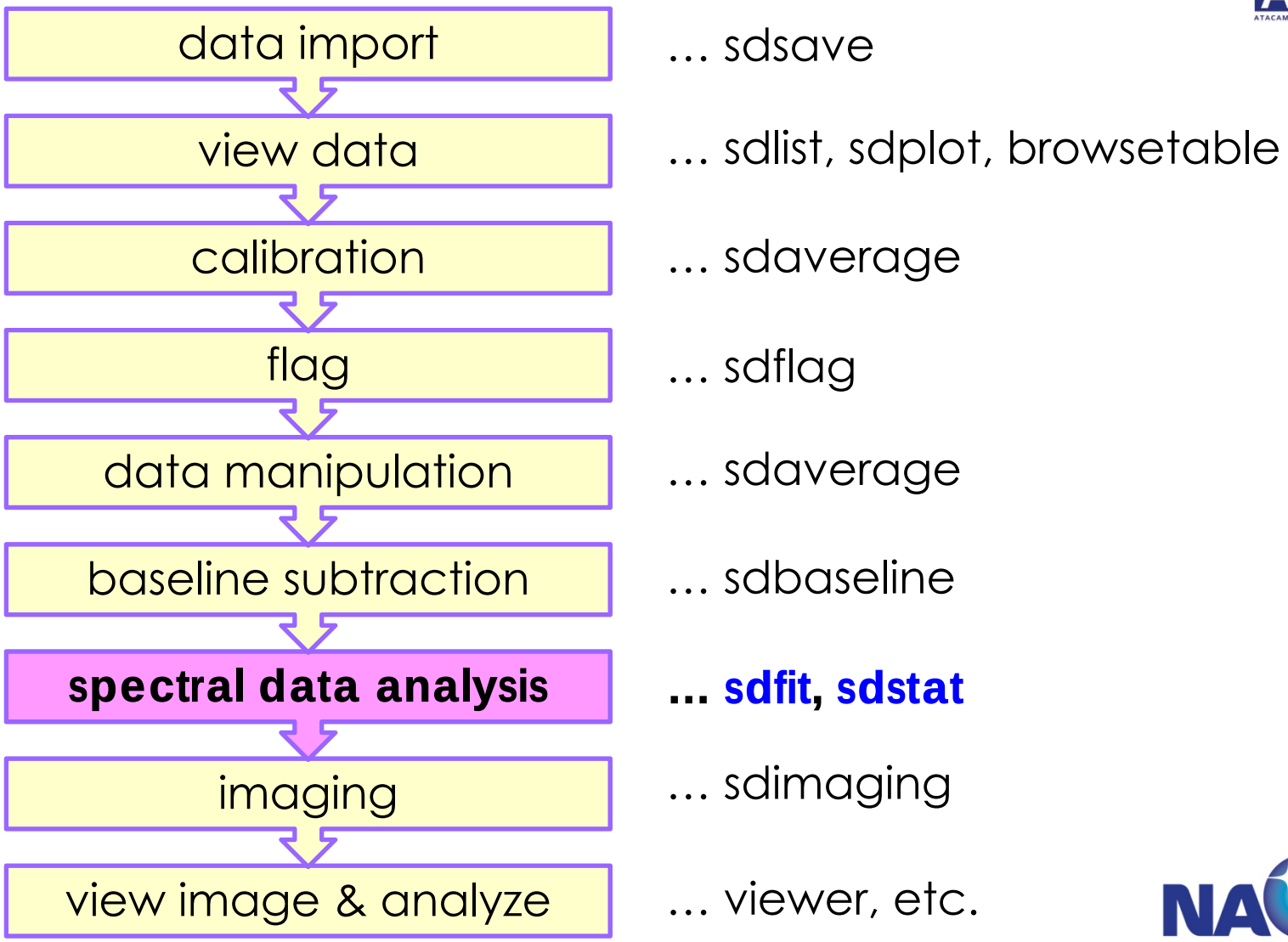
draw rectangle

- **add** mask chans:
  LEFT-mouse
- **delete** mask chans:
  RIGHT-mouse

current mask in yellow

press "Enter" to finish selection

Single dish data reduction with CASA



… sdsave

… sdlist, sdplot, browsetable

… sdaverage

… sdflag

… sdaverage

… sdbaseline

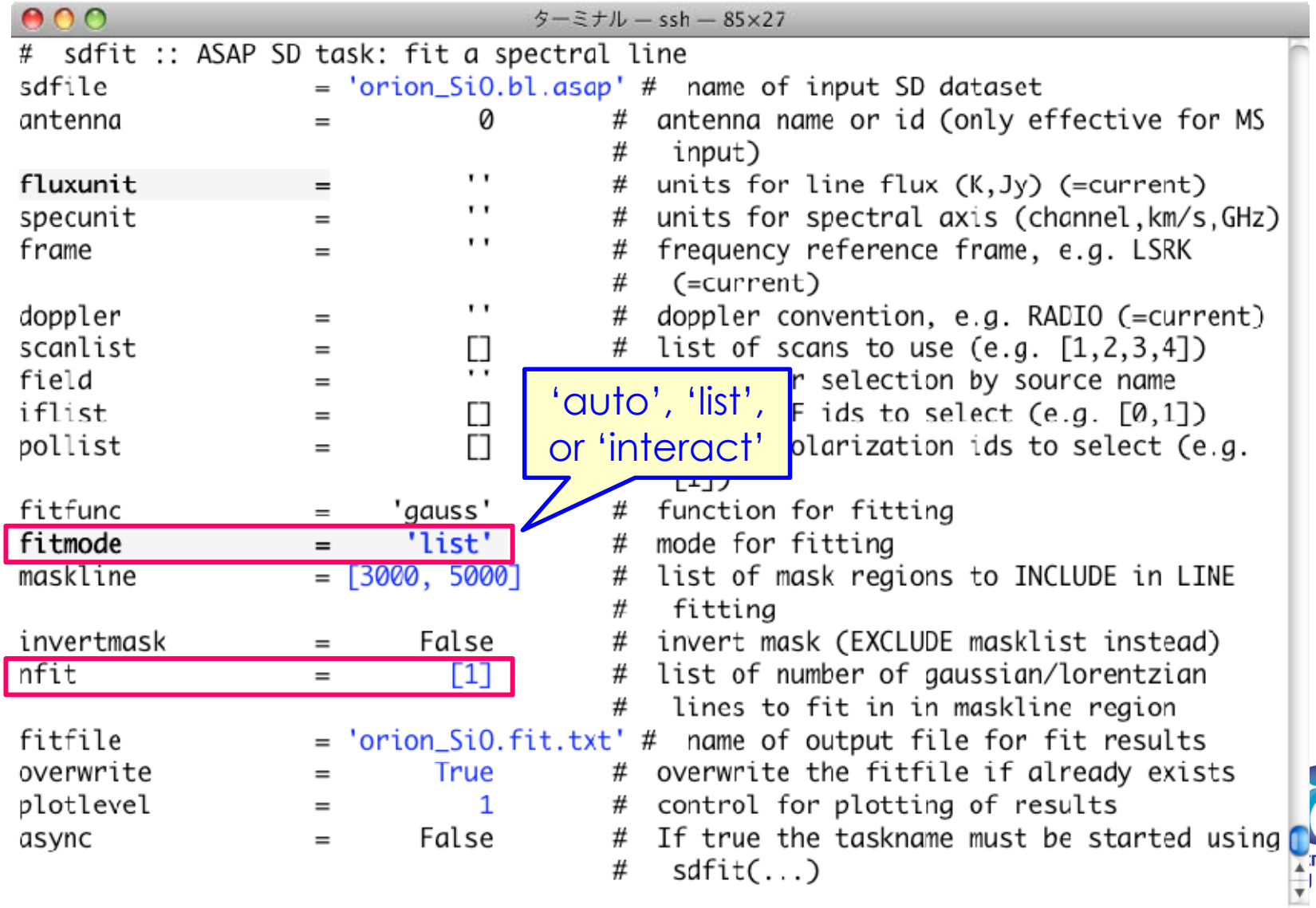
… sdfit, sdstat

… sdimaging

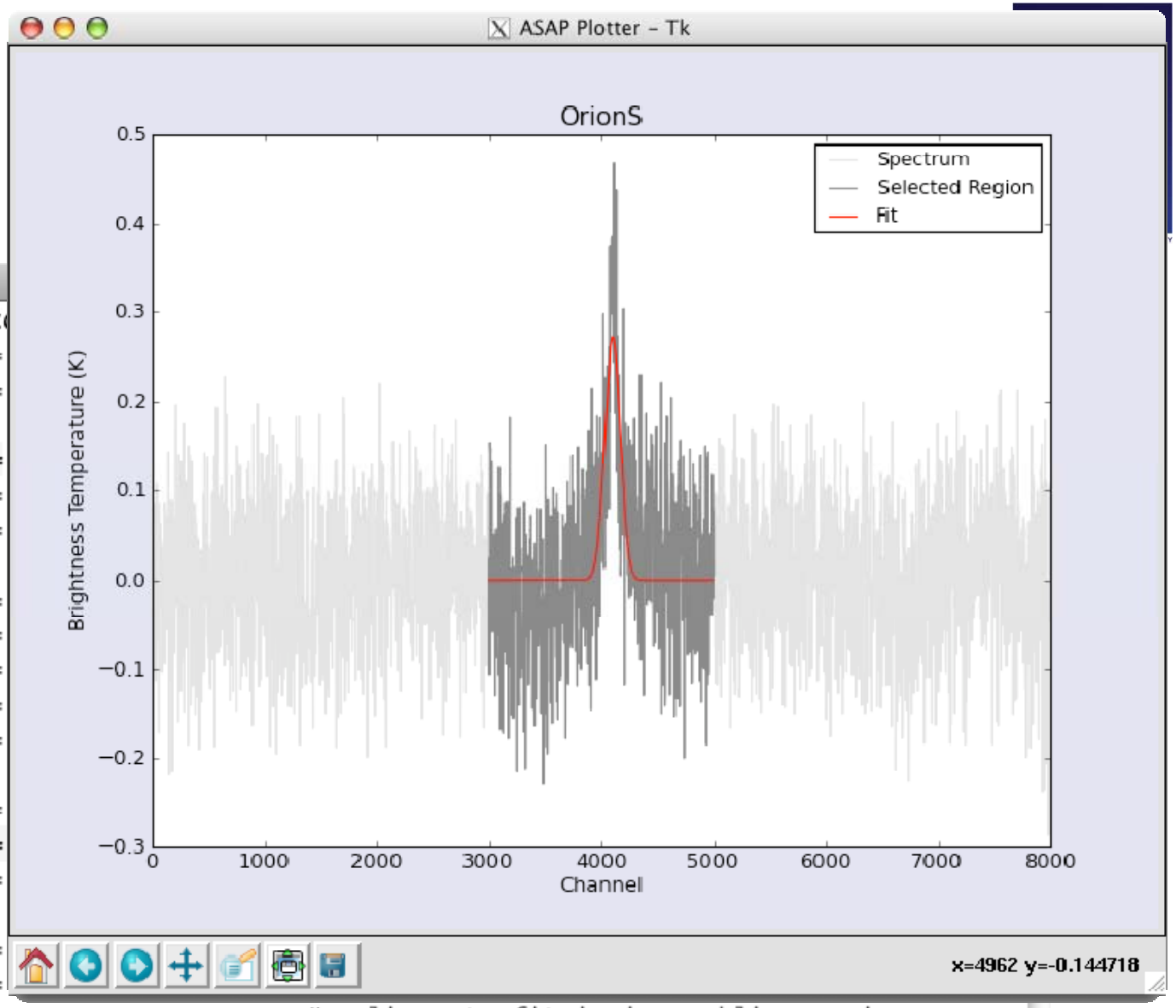
… viewer, etc.

sdfit – fit spectral line

Gaussian or Lorentzian fitting



CASA
Common Astronomy Software Applications




```
#  sdfit :: ASAP SD t
sdfile                =
antenna               =

fluxunit              =
specunit              =
frame                 =

doppler               =
scanlist              =
field                 =
iflist                =
pollist               =

fitfunc               =
fitmode               =
maskline              =

invertmask            =
nfit                  =

                                #   lines to fit in in maskline region
fitfile               = 'orion_SiO.fit.txt' #  name of output file for fit results
overwrite             =       True          #  overwrite the fitfile if already exists
plotlevel             =          1          #  control for plotting of results
async                 =      False          #  If true the taskname must be started using
                                #   sdfit(...)
```

ronomical
of Japan

sdstat – calculate statistics

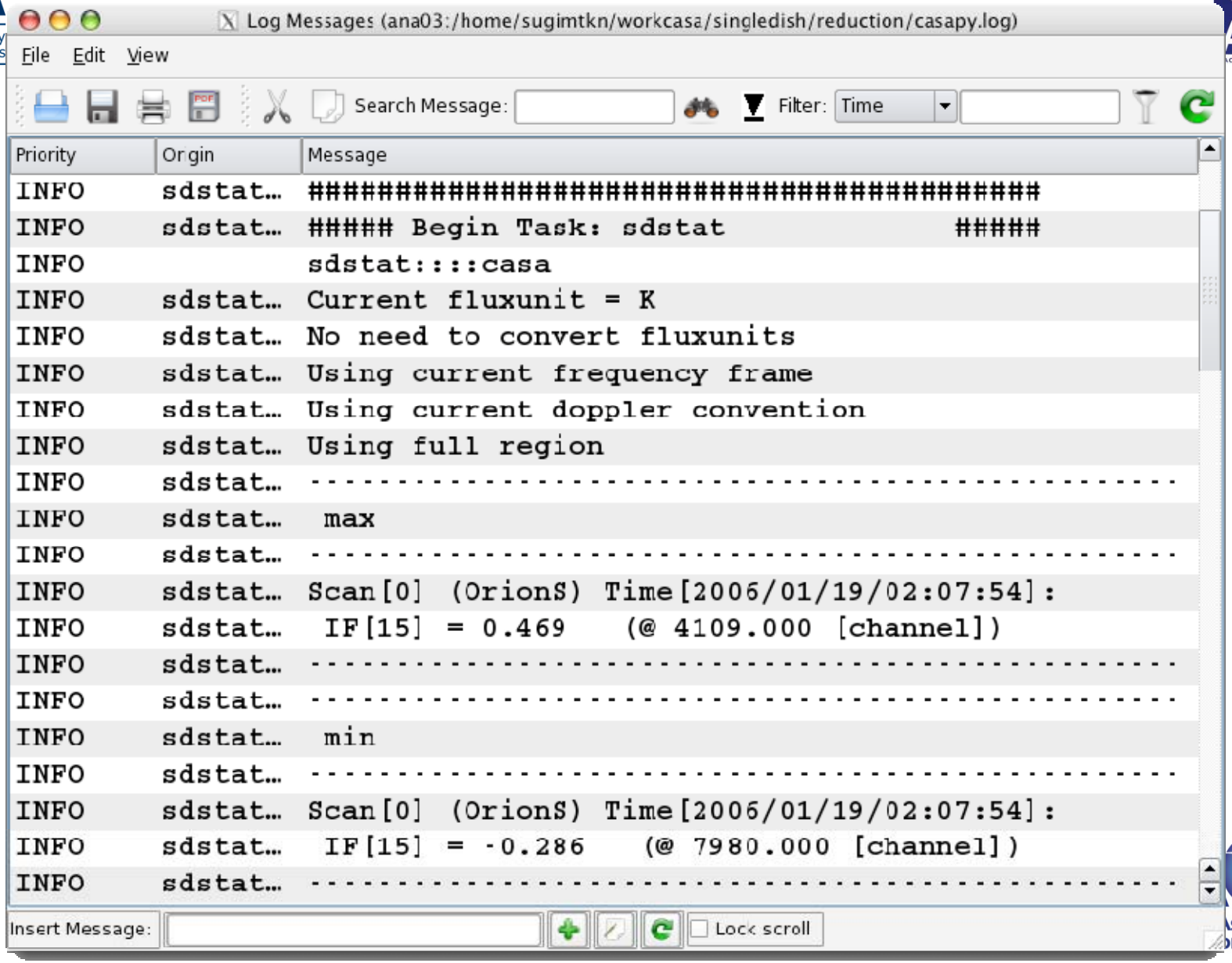
Min (channel), max (channel), mean, median, sum, rms, stddev, integrated intensity, and equivalent width

ターミナル — ssh — 91×19

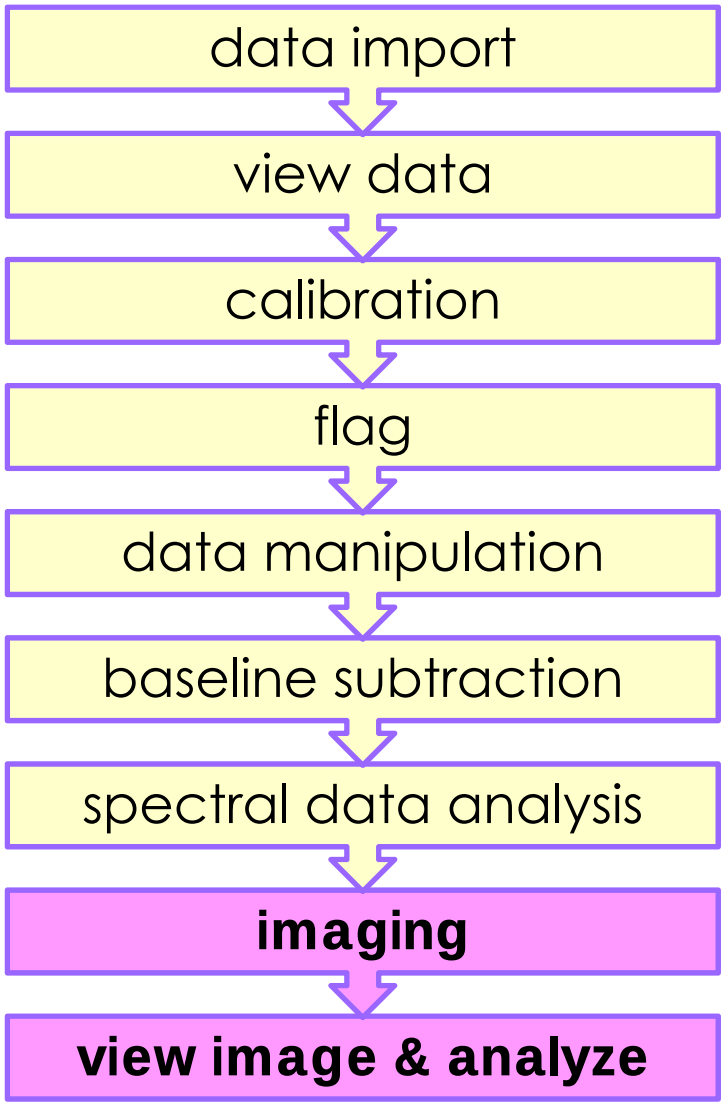
```
#  sdstat :: ASAP SD task: list statistics of spectral region
sdfile                 = 'orion_SiO.bl.asap' #  name of input SD dataset
antenna                =          0          #  antenna name or id (only effective for MS input)
fluxunit               =         ''          #  units for line flux (K,Jy) (=current)
specunit               =         ''          #  units for spectral axis (channel,km/s,GHz)
frame                  =         ''          #  frequency reference frame, e.g. LSRK (=current)
doppler                =         ''          #  doppler convention, e.g. RADIO (=current)
scanlist               =         []          #  list of scans to use (e.g. [1,2,3,4])
field                  =         ''          #  string for selection by source name
iflist                 =         []          #  list of IF ids to select (e.g. [0,1])
pollist                =         []          #  list of polarization ids to select (e.g. [0,1])
masklist               =         []          #  list of mask regions to INCLUDE in stats
invertmask             =      False          #  invert mask (EXCLUDE masklist instead)
interactive            =      False          #  determines interactive masking
statfile               = 'orion_SiO.stat.txt' #  name of output file for line statistics
format                 =     '3.3f'          #  format string to print statistic values
overwrite              =       True          #  overwrite the statistics file if already exists
async                  =      False          #  If true the taskname must be started using
                                             #   sdstat(...)
```

Output → CASA logger

sdstat – calculate statistics



Single dish data reduction with CASA



… sdsave

… sdlist, sdplot, browsetable

… sdaverage

… sdflag

… sdaverage

… sdbaseline

… sdfit, sdstat

… sdimaging

… viewer, etc.

sdsave (again) – convert data format

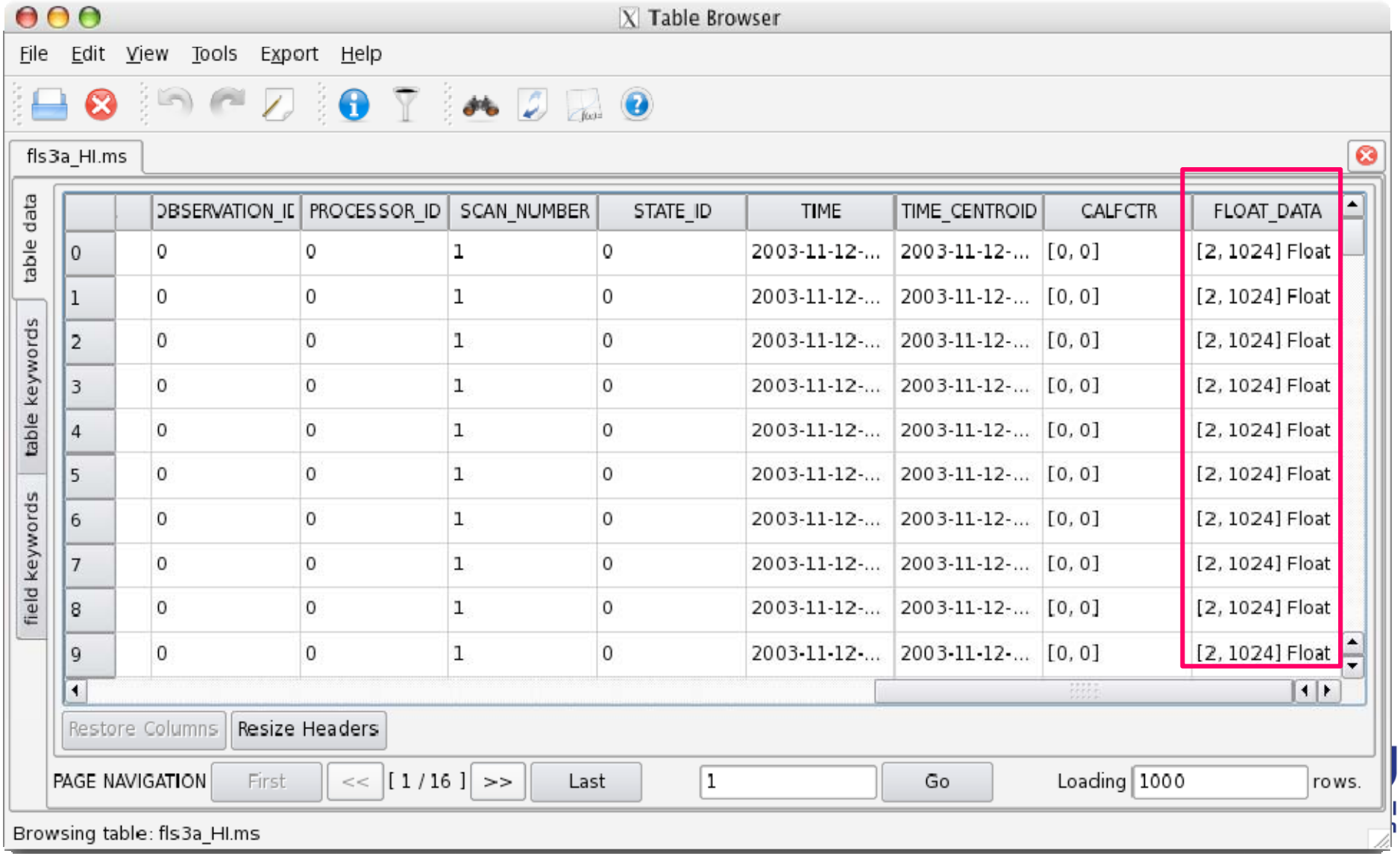
Important!
Convert data to **MeasurementSet** before imaging.

ターミナル — ssh — 91×16

```
#  sdsave :: ASAP SD task: save the sd spectra in various format
sdfile                 = 'fls3a_HI.asap'   #  name of input SD dataset
antenna                =          0        #  antenna name or id (only effective for MS input)
rowlist                =         []        #  list of row numbers to process (e.g. [0,2,4,6])
scanlist               =         []        #  list of scan to process (e.g. [20,21,22,23])
field                  =         ''        #  string for selection by source name
iflist                 =         []        #  list of IF ids to select (e.g. [0,1])
pollist                =         []        #  list of polarization ids to select (e.g. [0,1])
scanaverage            =      False        #  average integs within scans (True,False)
timeaverage            =      False        #  average scans over time (True,False)
polaverage             =      False        #  average over polarizations (True,False)
outfile                = 'fls3a_HI.ms'     #  output file name
outform                =       'MS2'       #  output file format (ASCII,MS2,SDFITS,ASAP)
overwrite              =       True        #  overwrite the output file if already exists
async                  =      False        #  If true the taskname must be started using
                                           #   sdsave(...)
```

single dish MS

Single dish data is stored in **FOAT_DATA** or **DATA** column

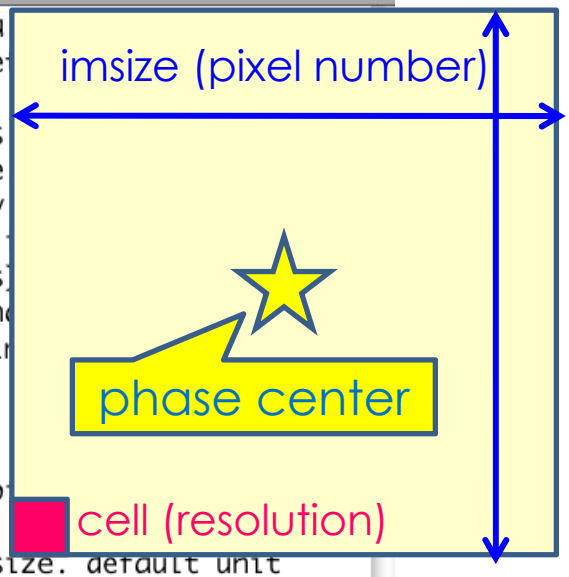


sdimaging – single dish imaging

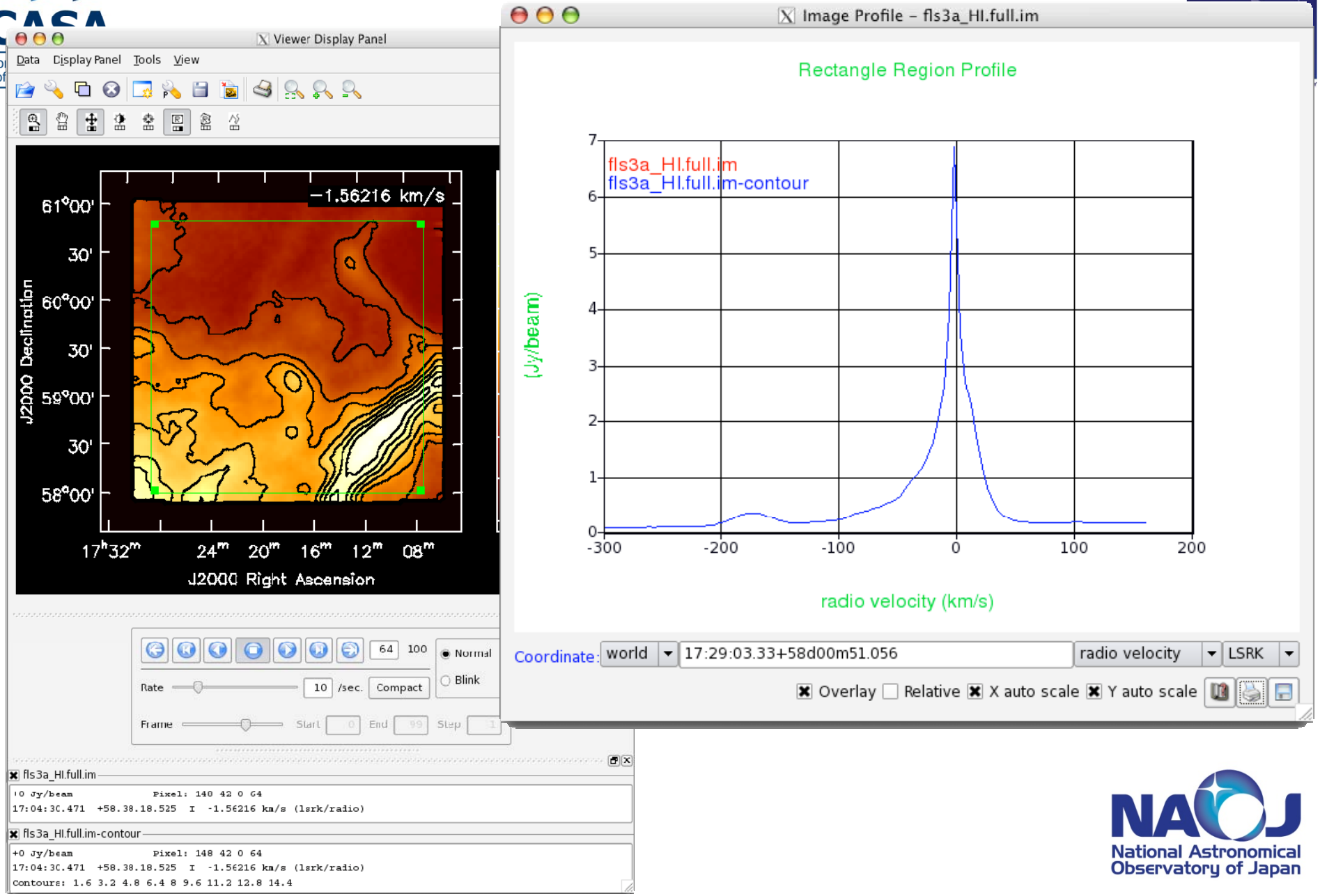
Input: **MeasurementSet** Output: **CASA image**

ターミナル — ssh — 88×31

```
#  sdimaging :: SD task: imaging for total power and spectral data
sdfile                 = 'fls3a_HI.ms'      #  name of input SD dataset
                                            #   for this task)
specunit               =  'channel'         #  units for spectral axis
scanlist               =          []        #  list of scans to use (e
field                  =           0        #  string for selection by
spw                    =           0        #  spectral window id for
antenna                =           0        #  antenna name(s) or id(s)
stokes                 =         'I'        #  stokes or correlation n
gridfunction           =        'PB'        #  gridding function for i
                                            #   ("BOX","SF","PB")
imagename              = 'fls3a_HI.full.im' #  output image name
overwrite              =        True        #  overwrite option
imsize                 = [150, 150]         #  x and y image size in p
                                            #   single value
cell                   = ['1.5arcmin', '1.5arcmin'] #  x and y cell size. default unit
                                            #   arcmin
dochannelmap           =        True        #  True for channel map, False for total power
                                            #   imaging
     nchan             =         100        #  number of spectral channel for created image
     start             =          30        #  reference value of start channel (in units
                                            #   of specunit)
     step              =           9        #  width of each spectral channel (in units of
                                            #   specunit)

phasecenter            = 'J2000 17:18:29 +59.31.23' #  Image phase center: position or
                                            #   field index
ephemsrcname           =          ''        #  ephemeris source name
pointingcolumn         = 'direction'        #  pointing data column to use
async                  =      False        #  If true the taskname must be started using
                                            #   sdimaging(...)
```



viewer – view CASA image



National Astronomical Observatory of Japan

The other single dish tasks

- ✓ **sdsmooth** … channel average spectra
- ✓ **sdcal** = **sdaverage** + **sdbaseline** + **sdsmooth**
- ✓ **sdflagmanager** … save, restore, and list flag versions
- ✓ **sdtpimaging** … total power imaging (incl. scan noise subtraction)
- ✓ **sdimprocess** … subtract scan noise
- ✓ **sdscale** … scale spectra
- ✓ **sdmath** … various arithmetic (+, -, *, /)operations on spectra
- ✓ **sdcoadd** … merge scantables

♪ Enjoy! ♪

Step 1: run the reduction scripts (orion_SiO.py & fls3a.py) by execfile , e.g.,

```
CASA<>:  execfile('orion_SiO.py')
```

to understand single dish data reduction steps

Step 2: open the scripts to see how parameters are set. Copy and paste each line to set parameters and run tasks. You can also change parameters as you like to see the changes

✓ CASA task reference
http://casa.nrao.edu/docs/TaskRef/TaskRef.html
✓ CASA cookbook
http://casa.nrao.edu/docs/UserMan/UserMan.html