

NRAO



CASA

Juergen Ott (NRAO)

National Radio Astronomy Observatory



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



NRAO



CASA

Juergen Ott (NRAO)

National Radio Astronomy Observatory



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)

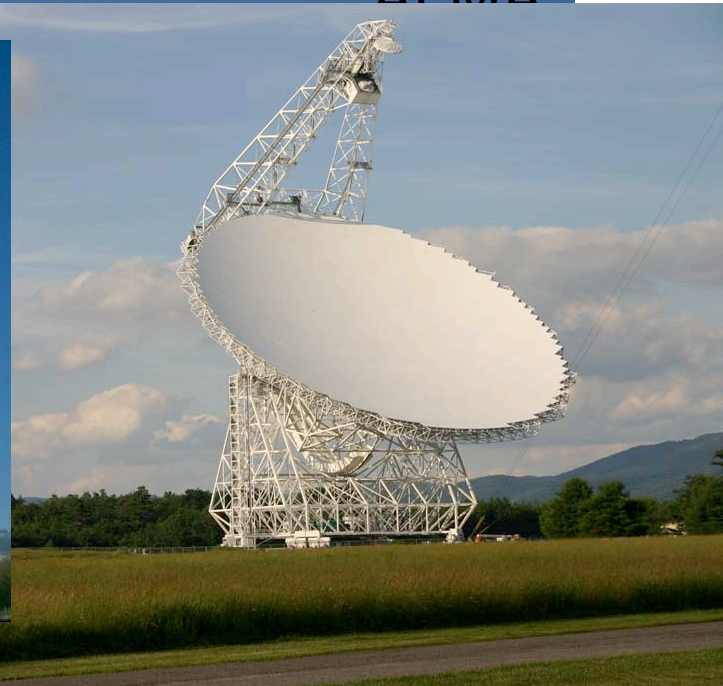


Introduction to CASA

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



ALMA



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

```
Default
New Info : Customize Bookmarks Close :
CASA <2>: 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            viewer
(importgmrt)       vishead        plotxy            (viewerconnection)
visstat

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

Image Analysis     Simulation      Utilities          Single Dish
-----
imcontsub          simdata        browsetable       (after running asap_init())
imhead            (simdata2)    casalogger
imfit             clearplot
immath            clearstat
immoments         csvclean
imregrid          filecatalog
imsmooth          find
imstat            help par.parameter
imval             help task
(specfit)         rmtables
                  startup
                  taskhelp
                  tasklist
                  toolhelp

sdaverage
sdbaseline
sdcal
sdcoadd
sdffit
sdflag
sdimaging
sdimprocess
sdlist
sdmath
sdplot
sdsave
sdscale
sdsmooth
sdstat
sdtpimaging
(sdsim)
(msmoments)

User defined tasks
-----
CASA <3>: |
```



Key Tasks

To see list of tasks with short help:

>taskhelp

```
Default
New Info : Customize Bookmarks Close :
CASA <15>: taskhelp
-----> taskhelp()
Available tasks:

accum          : Accumulate incremental calibration solutions into a calibration table
applycal       : Apply calibrations solutions(s) to data
autoclean      : CLEAN an image with automatically-chosen clean regions.
bandpass       : Calculates a bandpass calibration solution
blcal          : Calculate a baseline-based calibration solution (gain or bandpass)
boxit          : Box regions in image above given threshold value.
browsetable    : Browse a table (MS, calibration table, image)
calstat        : Displays statistical information on a calibration table
clean          : Invert and deconvolve images with selected algorithm
clearcal       : Re-initializes the calibration for a visibility data set
clearplot      : Clear the matplotlib plotter and all layers
clearstat      : Clear all autolock locks
concat         : Concatenate several visibility data sets.
conjugatevis   : Change the sign of the phases in all visibility columns.
csvclean       : This task does an invert of the visibilities and deconvolve in the image plane.
cvel          : regrid an MS to a new spectral window / channel structure or frame
deconvolve     : Image based deconvolver
exportasdm     : Convert a CASA visibility file (MS) into an ALMA Science Data Model
exportfits     : Convert a CASA image to a FITS file
exportuvfits   : Convert a CASA visibility data set to a UVFITS file:
feather        : Combine two images using their Fourier transforms
find           : Find string in tasks, task names, parameter names:
fixvis         : Recalculates or converts (u, v, w)
flagautocorr   : Flag autocorrelations
flagcmd        : Flagging task based on flagging commands
flagdata       : All purpose flagging task based on selections
flagdata2      : All purpose flagging task based on selections. It allows the combination of se
flagmanager    : Enable list, save, restore, delete and rename flag version files.
fluxscale      : Bootstrap the flux density scale from standard calibrators
ft             : Insert a source model into the MODEL_DATA column of a visibility set:
gaincal        : Determine temporal gains from calibrator observations
gencal         : Specify Calibration Values of Various Types
hanningsmooth  : Hanning smooth frequency channel data to remove Gibbs ringing
imcollapse     : Collapse image along one axis, aggregating pixel values along that axis.
imcontsub      : Subtracts specified continuum channels from a spectral line data set
imfit          : Fit one or more elliptical Gaussian components on an image region(s)
imhead        : List, get and put image header parameters
immath         : Perform math operations on images
immoments      : Compute moments from an image

Default      Default      Default      Default      Default
```



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>: |
```



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

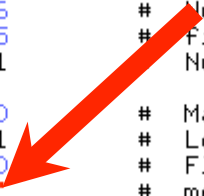
sanity checks of parameters in `inp` :

```
IPy:Jupiter
CASA <5>: psfmode='hogwarts'
CASA <6>: inp()
# clean :: Deconvolve an image with selected algorithm
vis = 'ngc5921.usecase.ms.cont' # Visibility file
imagename = 'ngc5921.usecase.clean' # Images
field = '0' # Field index
spw = '' # Spectral window index '' is all
selectdata = False # Other parameters
mode = 'channel' # Type of selection (ms, 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 = 'hogwarts' # 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
restoringbeam = [''] # Output Gaussian restoring beam for CLEAN image
```

erroneous values in red



Help on Tasks

 In-line help:

>help 'clean' OR >pdoc clean

```
IPy:Jupyter
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
```

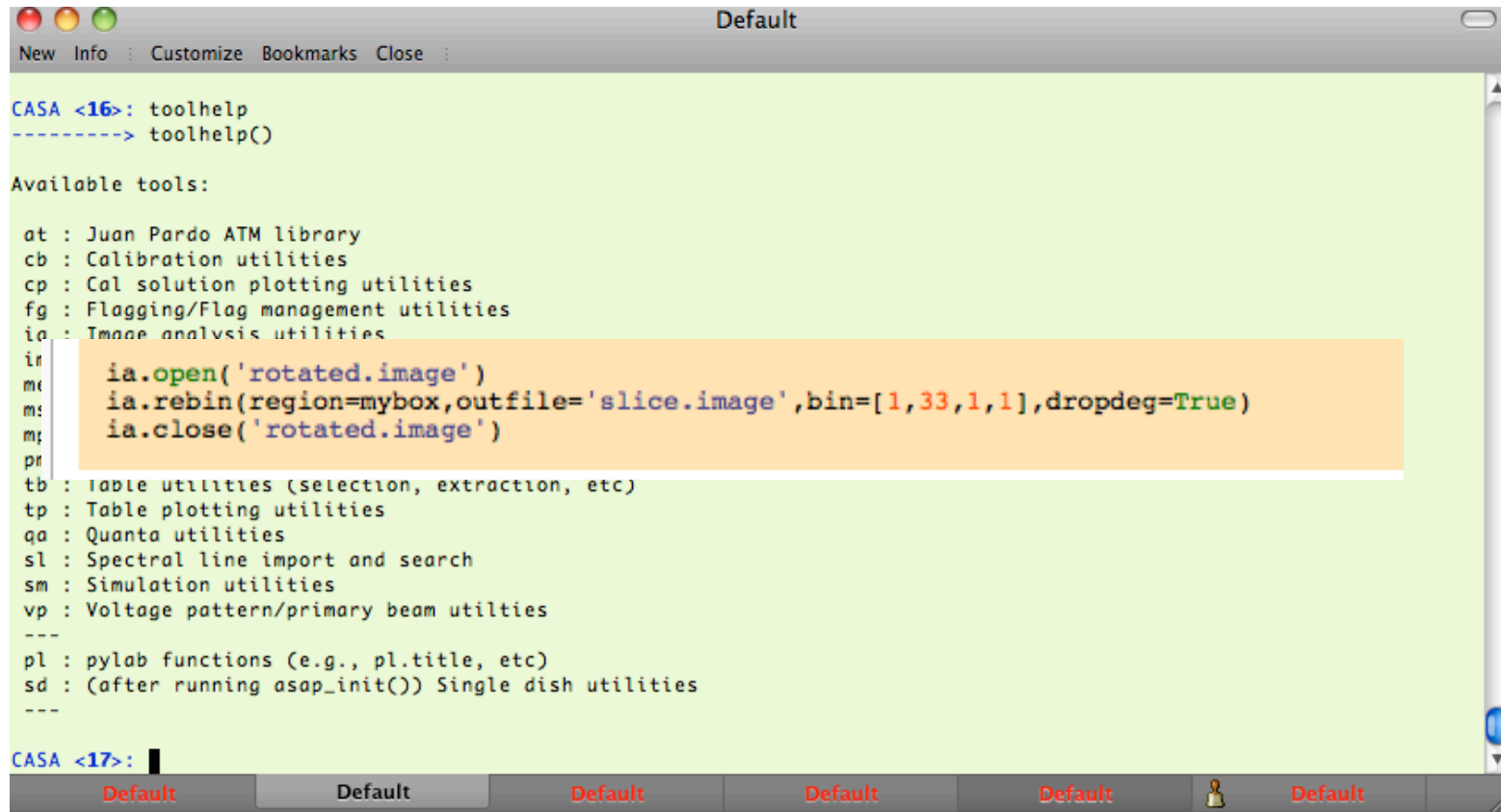


Tools in CASA

- 🕒 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) , calibrator (cb), ms (ms) , etc. (see `toolhelp`)

CASA Tool List

list of default tools from [toolhelp](#) :



```
Default
New Info : Customize Bookmarks Close :

CASA <16>: 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
in
me
ms
mp
pr
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 <17>:

```


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>

[What imager produces:](#)
[What imager does not do:](#)
[What improvement to imager are in the works:](#)
[Advanced use of imager:](#)
[Overview of imager tool functions:](#)

2.4.1 imager - Tool

[imager.imager - Function](#)
[imager.advise - Function](#)
[imager.approximatepsf - Function](#)
[imager.boxmask - Function](#)
[imager.calcuvw - Function](#)
[imager.clean - Function](#)
[imager.clipimage - Function](#)
[imager.clipvis - Function](#)
[imager.close - Function](#)
[imager.defineimage - Function](#)
[imager.done - Function](#)
[imager.drawmask - Function](#)
[imager.exprmask - Function](#)
[imager.feather - Function](#)
[imager.filter - Function](#)
[imager.fitpsf - Function](#)
[imager.fixvis - Function](#)
[imager.ft - Function](#)
[imager.linear mosaic - Function](#)
[imager.make - Function](#)
[imager.makeimage - Function](#)
[imager.makemodelfromsd - Function](#)
[imager.mask - Function](#)
[imager.mem - Function](#)
[imager.nnls - Function](#)
[imager.open - Function](#)
[imager.pb - Function](#)
[imager.plotsummary - Function](#)
[imager.plotuv - Function](#)
[imager.plotvis - Function](#)
[imager.plotweights - Function](#)
[imager.regionmask - Function](#)
[imager.regiontoimagemask - Function](#)
[imager.residual - Function](#)
[imager.restore - Function](#)
[imager.sensitivity - Function](#)

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:

Table Browser

ngc5921.usecase.ms

	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

Restore Columns Resize Headers

PAGE NAVIGATION First << [1 / 23] >> Last 1 Go Loading 1000 rows.

Browsing table: ngc5921.usecase.ms

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

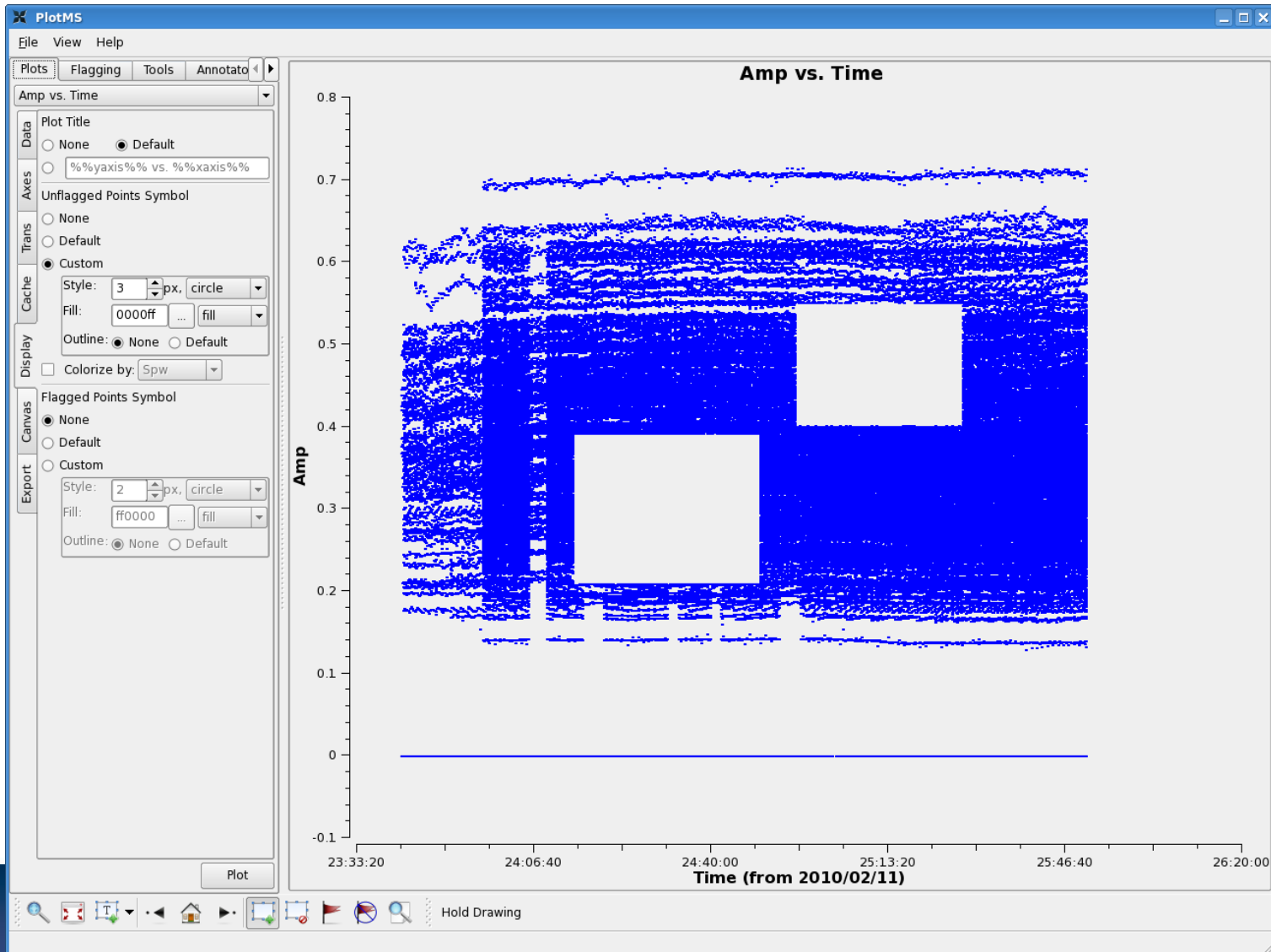


Image Viewer

The image shows a software interface for viewing astronomical data. The main window is titled "Viewer Display Panel" and contains a plot of a celestial object with overlaid contours. The plot axes are labeled "J2000 Declination" (y-axis, ranging from 58' to 10') and "J2000 Right Ascension" (x-axis, ranging from 15^h22^m18^s to 36^s). The object is a bright orange-yellow disk with several blue contour lines overlaid. Below the plot is a control panel with navigation buttons (back, forward, home, etc.), a "Rate" slider set to 10 /sec, and a "Frame" slider with "Start" and "End" set to 0 and "Step" set to 1. There are also radio buttons for "Normal" and "Blink" display modes.

To the right of the main window is a "Data Display Options" panel. It contains several sections for configuring the display of the data:

- Display axes**: A button to show/hide the axes.
- Hidden axes**: A button to show/hide hidden axes.
- Basic Settings**:
 - Aspect ratio: fixed world
 - Pixel treatment: edge
 - Resampling mode: bilinear
 - Relative Contour Levels: [0.2, 0.4, 0.6, 0.8]
 - Base Contour Level: 1381.3
 - Unit Contour Level: 1567.1
 - Line width: 0.5
 - Dash negative contours?: true
 - Dash positive contours?: false
 - Line color: blue
- Position tracking**: A button.
- Axis labels**: A button.
- Axis label properties**: A button.
- Beam Ellipse**: A button.
- Apply**: A button.
- Dismiss**: A button at the bottom right.

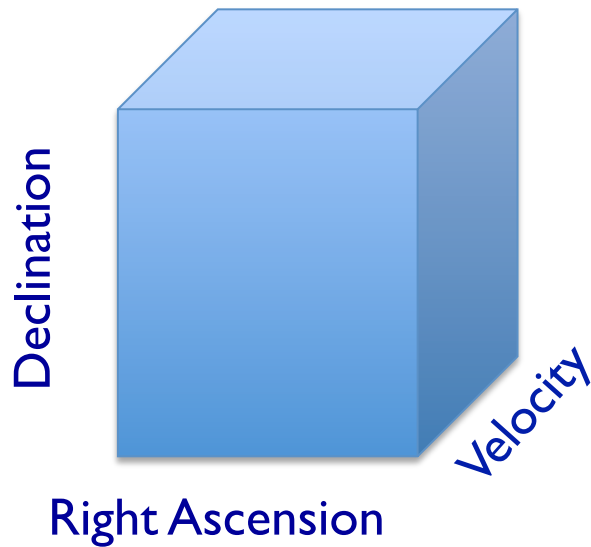
At the bottom of the main window, there are two panels showing data for the current image:

```
ngc5921.demo.moments.weighted_coord-contour
masked          Pixel: 155 120 0 0
15:21:32.830 +05.01:52.605 I 1607.99 km/s
Contours: 1418.5 1455.6 1492.8 1529.9

ngc5921.demo.moments.integrated
masked          Pixel: 155 120 0 0
15:21:32.830 +05.01:52.605 I 1607.99 km/s
```

Image Viewer

- Displaying cubes
- Movies
- Channel maps



Viewer Display Panel

Data Display Panel Tools View

1499.78 km/s

1494.63 km/s

1489.48 km/s

1484.32 km/s

J2000 Declination

J2000 Right Ascension

Rate 10 /sec. Compact

Frame Start 0 End 45 Step 1

ngc5921.demo.clean.image

+0.00358195 Jy/beam Pixel: 81 119 0 22
15:22:47.684 +05.01.41.878 I 1494.63 km/s

Viewer Display Panel

Data Display Panel Tools View

J2000 Declination

J2000 Right Ascension

Coordinate: world 15:22:07.927+05d01'47.92 velocity

Rate: 10 /sec. Compact

Frame: Start 0 End 45 Step 1

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

Display Options

ngc5921.usecase.clean.image ngc5921.usecase.clean.image-contour

Display axes

Hidden axes

Basic Settings

Image Profile - ngc5921.usecase.clean.image

Rectangle Region Profile

Flux Density (mJy)

velocity

Coordinate: world 15:22:07.927+05d01'47.92 velocity

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

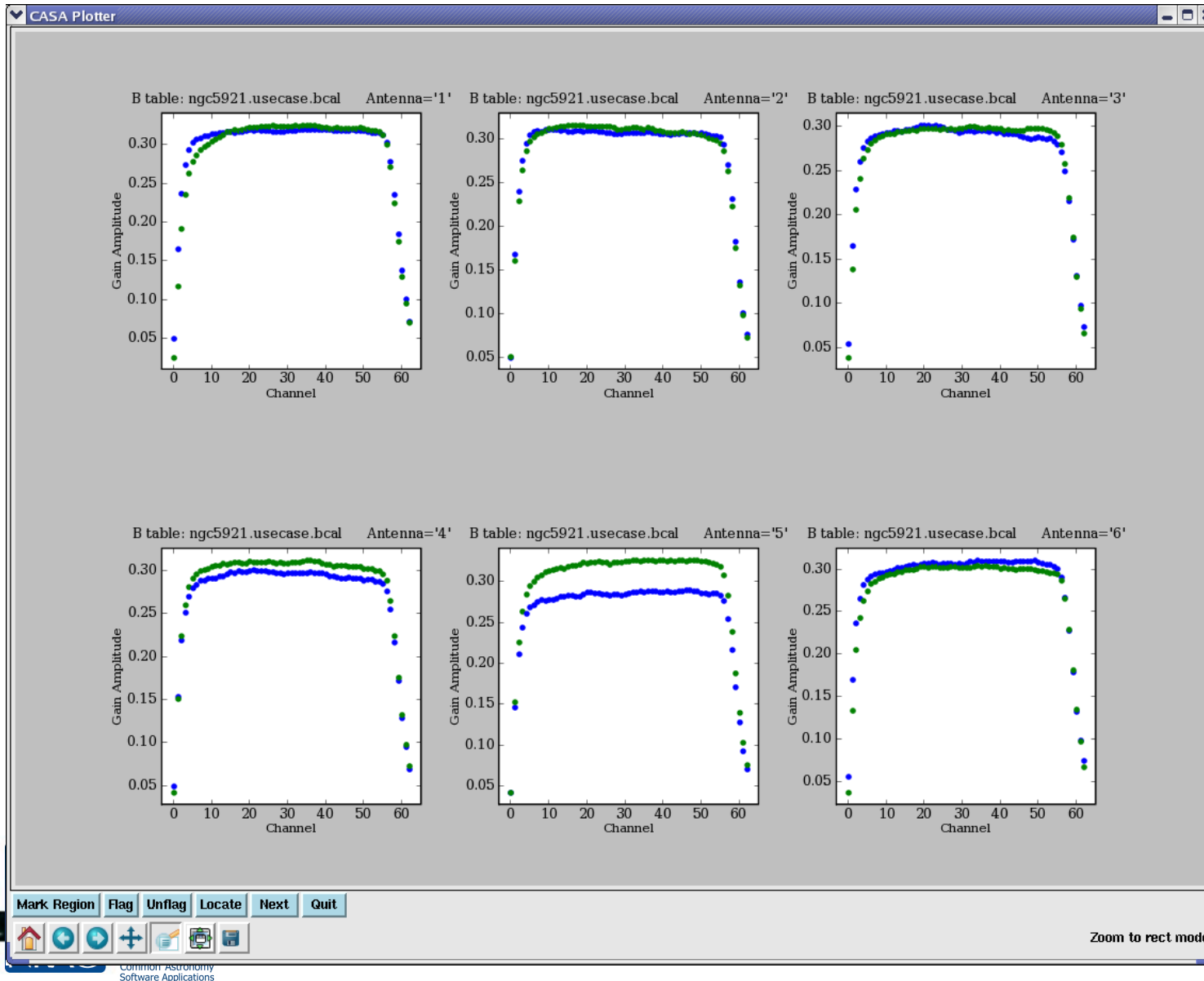
Marker Map

Update Leave Open Done

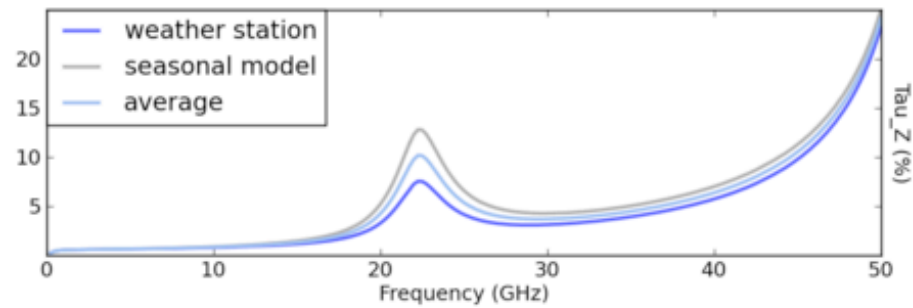
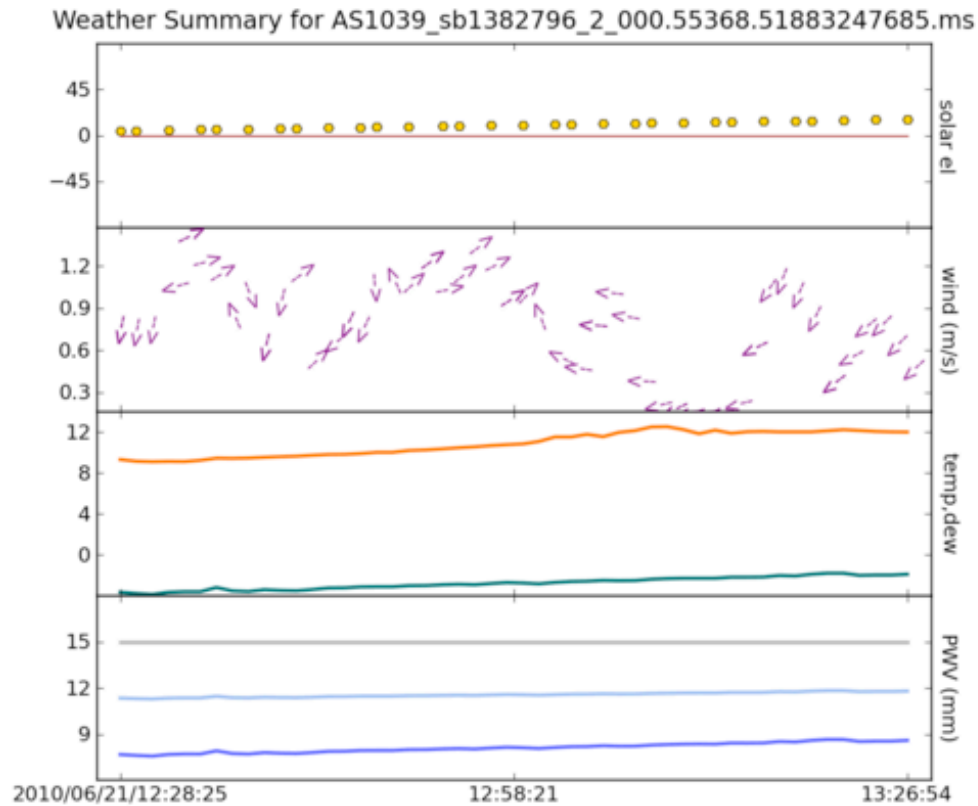
MSViewer

The screenshot displays the MSViewer software interface. A main window shows a grid of spectral data plots with a horizontal axis labeled 'Baseline' ranging from 2000 to 8000. A 'Data Display Options' dialog box is open, showing settings for the file 'n4826_16apr.ms'. The dialog has several sections: 'Advanced', 'MS and Visibility Selection', 'Display Axes', 'Flagging Options', and 'Basic Settings'. Under 'Display Axes', the X Axis is set to 'Baseline', Y Axis to 'Time', and Animation Axis to 'Spectral Window'. The Channel is set to 33 and Polarization to 0. The 'Flagging Options' section is currently collapsed. Below the main plot, there are playback controls including buttons for home, play, stop, and next, along with numerical input fields for frame numbers (2 and 6) and a speed setting of 10 /sec. The 'Compact' view is selected, and there are radio buttons for 'Normal' and 'Blink' display modes.

Plotcal



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 I-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!

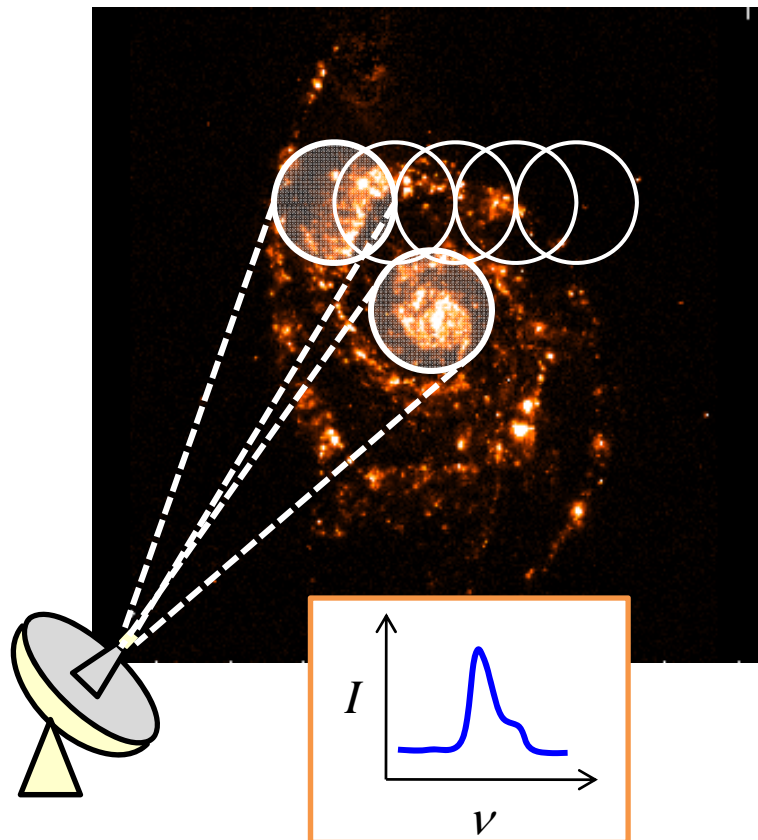
Contents

- ✓ 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

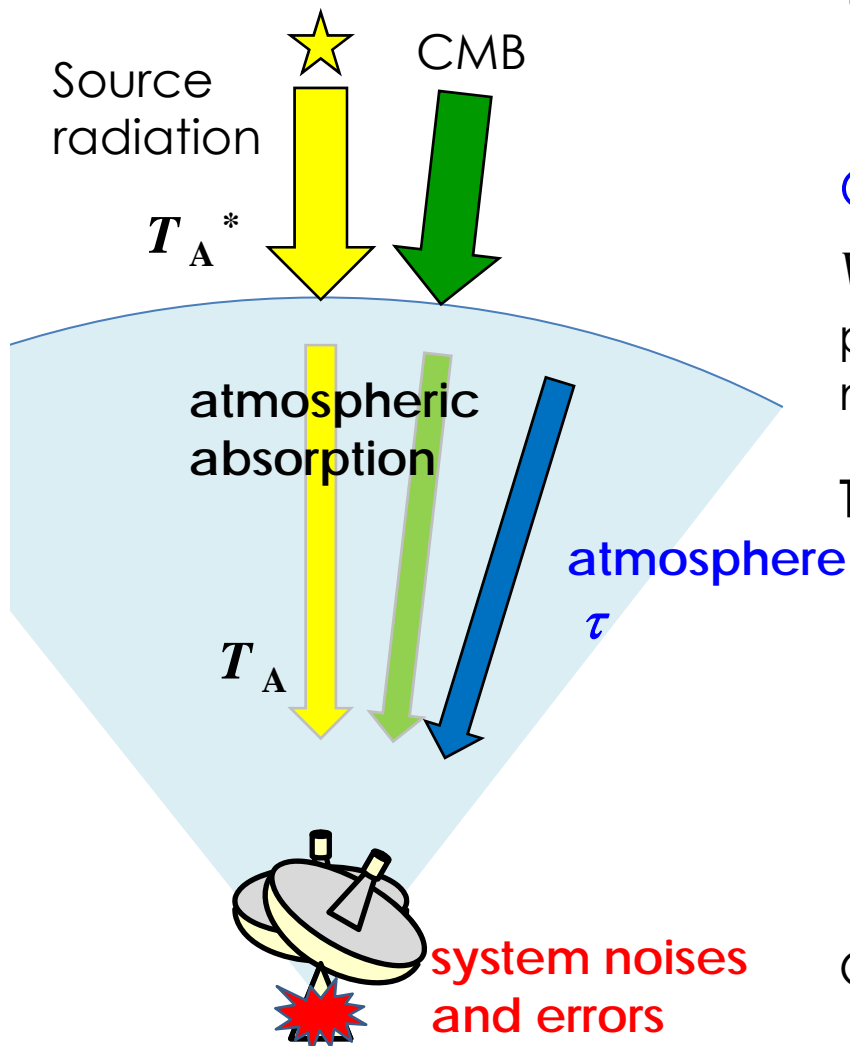
Imaging

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

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

OFF source signal

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

position switching, frequency switching, nutator switching, etc.

The brightness temperature of a source

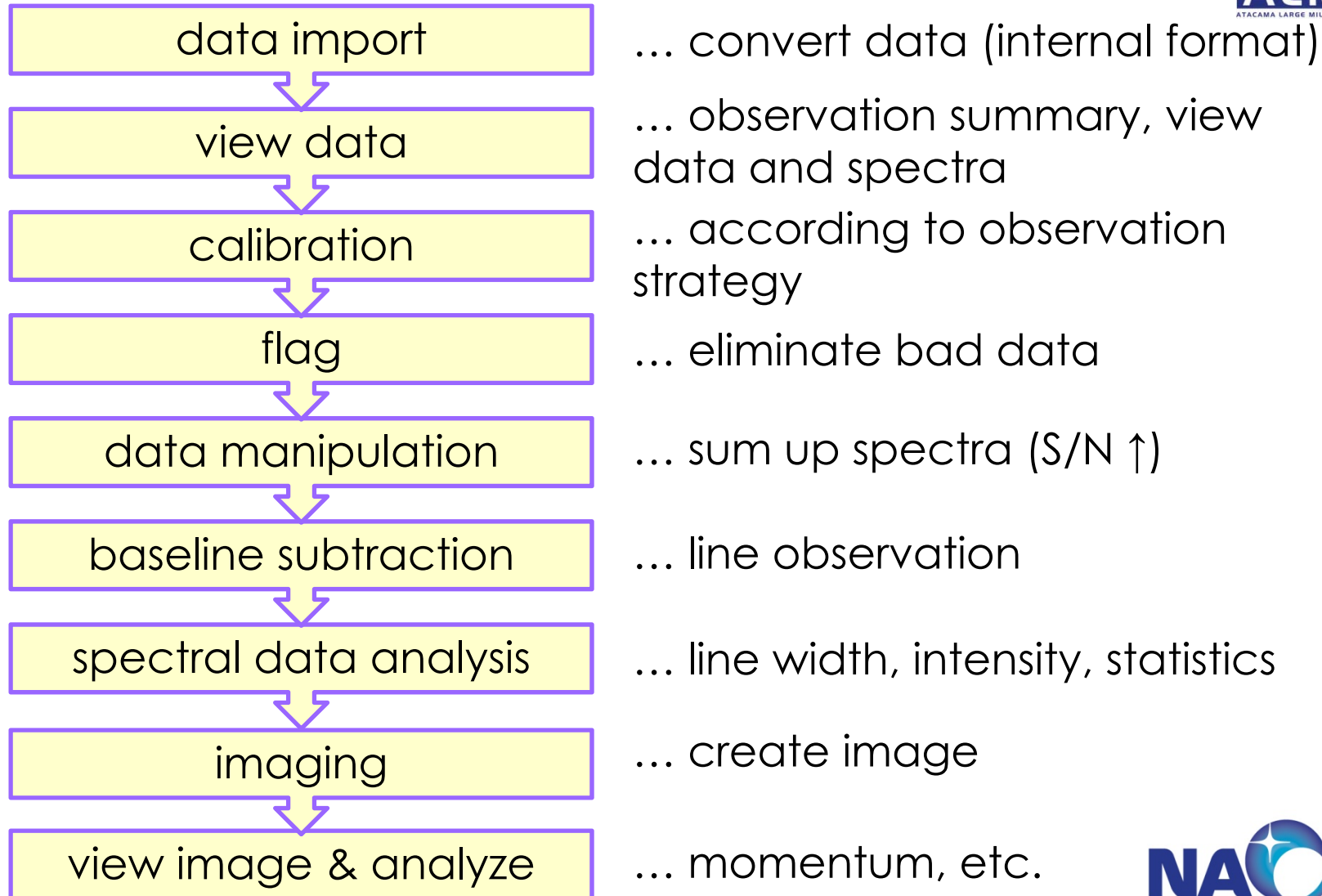
$$T_A = T_{\text{CAL}} \frac{V_{\text{ON}} - V_{\text{OFF}}}{V_{\text{CAL}}}$$

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

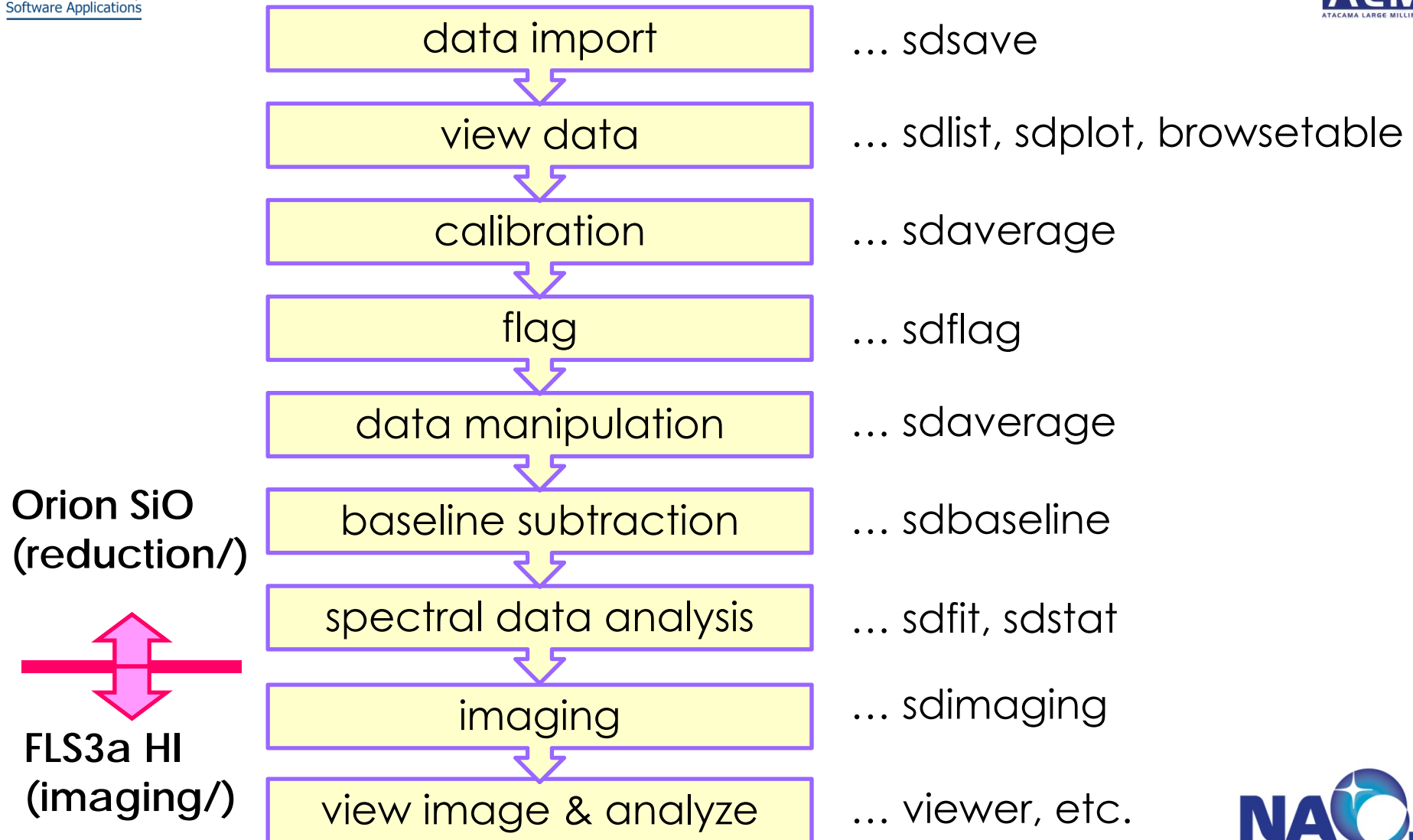
the conversion factor :
a signal strength → the
brightness temperature

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

Single dish data reduction steps example



Single dish data reduction with CASA



Tutorial data

Spectral line data reduction (reduction/)

- ✓ Data file: OrionS_rawACSm0d
- ✓ 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

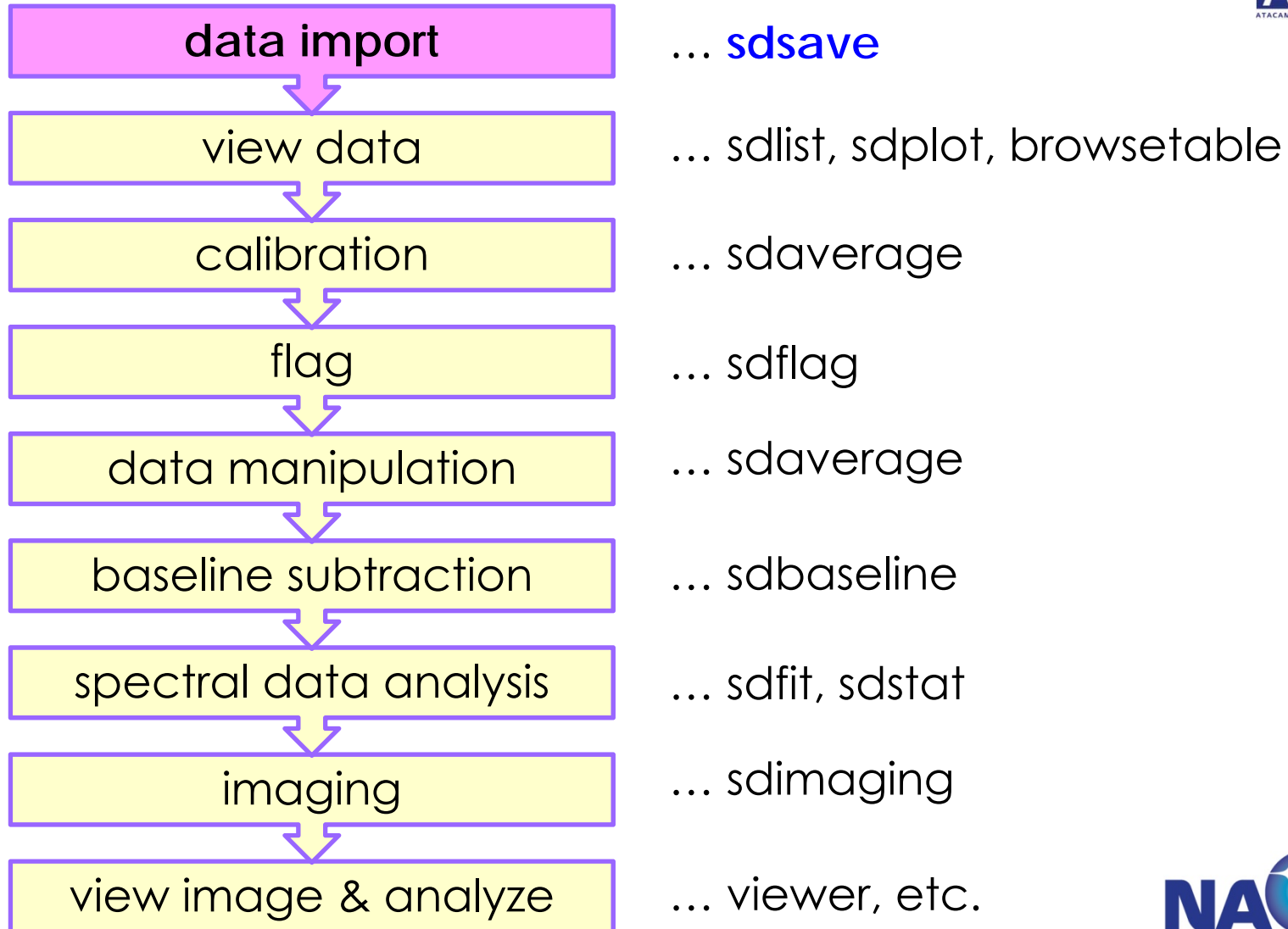
```
CASA<>:  asap_init()
```

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



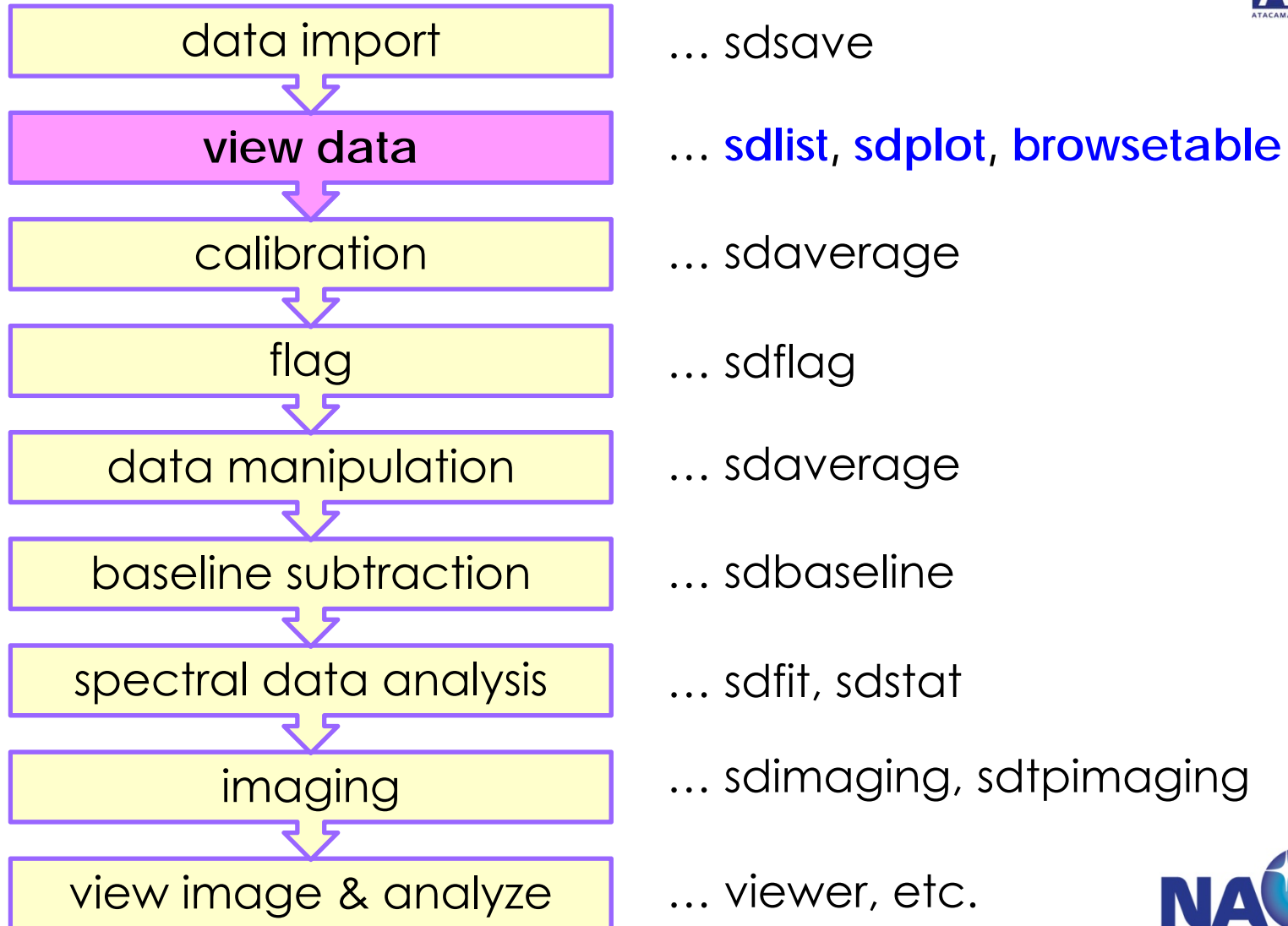
sdsave - import / export data

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 — 96x17
# 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_Si0.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(...)
```

Single dish data reduction with CASA



sdlist – list observation

View scantable summary

```
ターミナル — ssh — 96x9

# 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



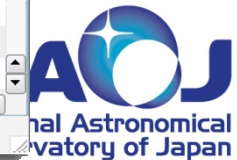
Log Messages (ana03:/home/sugimtkn/workcasa/singledish/reduction/casapy.log)

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..	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)

Table Browser

File Edit View Tools Export Help

orion_SiO.asap

table data

	SRCTYPE	FIELDNAME	SPECTRA	FLAGTRA	TSYS	DIRECTION	AZIM
0	1	OrionS	[28.1654, 0.185437, 0.185668, 0....	[0, 0, 0, 0, 0...	[1]	[1.37542, -0.0942875]	2.7994
1	1	OrionS	[6.16194, 0.30304, 0.302895, 0.3...	[0, 0, 0, 0, 0...	[1]	[1.37542, -0.0942875]	2.7994
2	1	OrionS	[31.302, 0.153334, 0.153913, 0.1...	[0, 0, 0, 0, 0...	[1]	[1.37542, -0.0942875]	2.7994
3	1	OrionS	[12.9472, 0.0422599, 0.0421586,...	[0, 0, 0, 0, 0...	[1]	[1.37542, -0.0942875]	2.7994
4	1	OrionS	[7.73297, 0.240423, 0.239547, 0...	[0, 0, 0, 0, 0...	[1]	[1.37542, -0.0942875]	2.7994
5	1	OrionS	[33.5171, 0.257387, 0.257224, 0....	[0, 0, 0, 0, 0...	[1]	[1.37542, -0.0942875]	2.7994
6	1	OrionS	[4.91839, 0.221327, 0.221545, 0....	[0, 0, 0, 0, 0...	[1]	[1.37542, -0.0942875]	2.7994
7	1	OrionS	[14.2414, 0.264974, 0.264911, 0....	[0, 0, 0, 0, 0...	[1]	[1.37542, -0.0942875]	2.7994
8	11	OrionS	[28.3547, 0.198971, 0.198763, 0....	[0, 0, 0, 0, 0...	[1]	[1.37542, -0.0942875]	2.7994
9	11	OrionS	[6.37802, 0.319975, 0.319337, 0....	[0, 0, 0, 0, 0...	[1]	[1.37542, -0.0942875]	2.7994

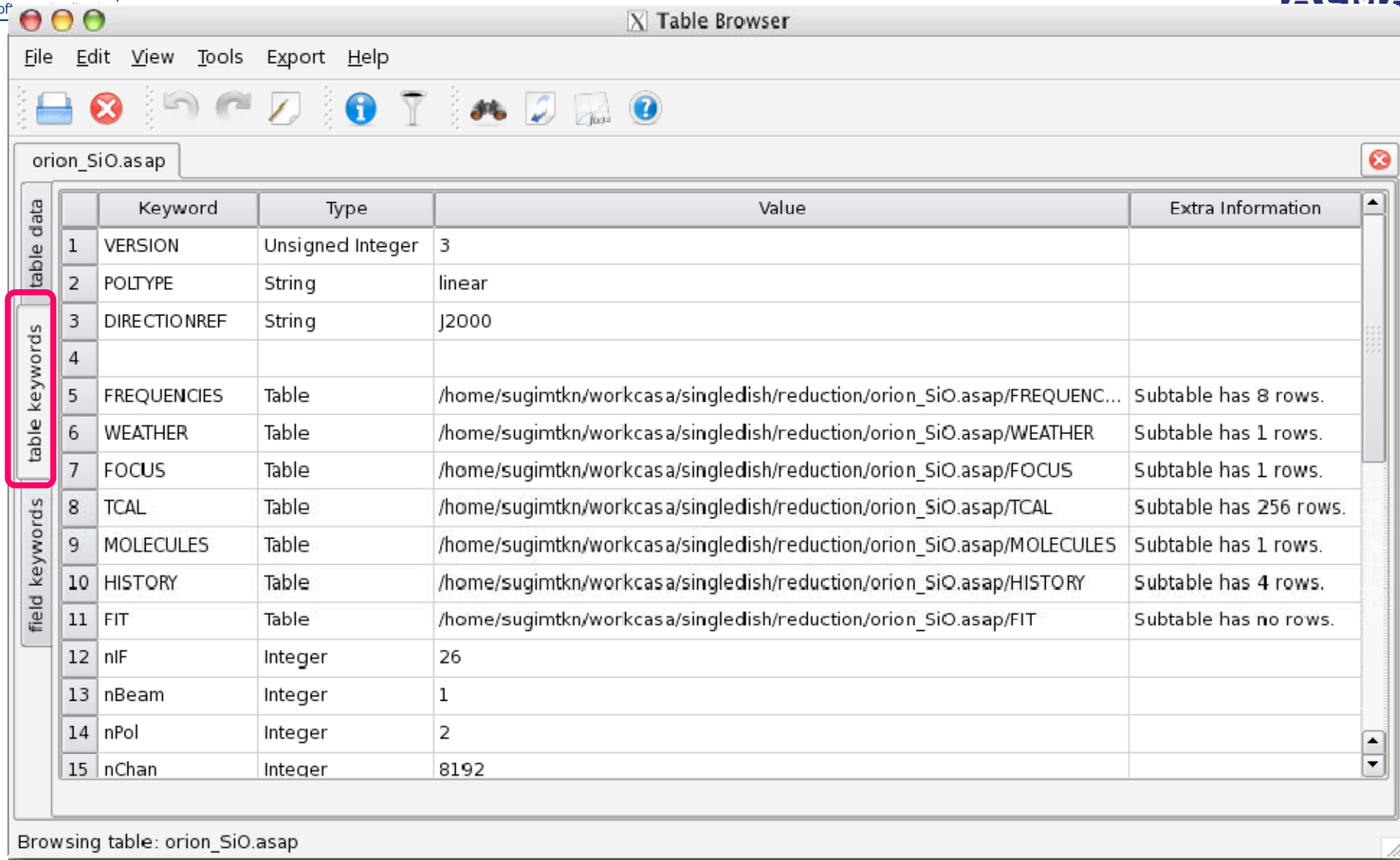
table keywords

field keywords

Restore Columns Resize Headers

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

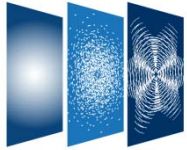
browsetable – view scantable (table data)



The screenshot shows the 'Table Browser' application window. The title bar reads 'Table Browser'. The menu bar includes 'File', 'Edit', 'View', 'Tools', 'Export', and 'Help'. Below the menu bar is a toolbar with various icons. The main window displays a table with the following columns: 'Keyword', 'Type', 'Value', and 'Extra Information'. The table is titled 'orion_SiO.asap'. On the left side of the table, there are two vertical labels: 'table data' and 'table keywords'. The 'table keywords' label is highlighted with a red box. The table contains 15 rows of data.

	Keyword	Type	Value	Extra Information
1	VERSION	Unsigned Integer	3	
2	POLTYPE	String	linear	
3	DIRECTIONREF	String	J2000	
4				
5	FREQUENCIES	Table	/home/sugimtkn/workcasa/singledish/reduction/orion_SiO.asap/FREQUENC...	Subtable has 8 rows.
6	WEATHER	Table	/home/sugimtkn/workcasa/singledish/reduction/orion_SiO.asap/WEATHER	Subtable has 1 rows.
7	FOCUS	Table	/home/sugimtkn/workcasa/singledish/reduction/orion_SiO.asap/FOCUS	Subtable has 1 rows.
8	TCAL	Table	/home/sugimtkn/workcasa/singledish/reduction/orion_SiO.asap/TCAL	Subtable has 256 rows.
9	MOLECULES	Table	/home/sugimtkn/workcasa/singledish/reduction/orion_SiO.asap/MOLECULES	Subtable has 1 rows.
10	HISTORY	Table	/home/sugimtkn/workcasa/singledish/reduction/orion_SiO.asap/HISTORY	Subtable has 4 rows.
11	FIT	Table	/home/sugimtkn/workcasa/singledish/reduction/orion_SiO.asap/FIT	Subtable has no rows.
12	nIF	Integer	26	
13	nBeam	Integer	1	
14	nPol	Integer	2	
15	nChan	Integer	8192	

Browsing table: orion_SiO.asap



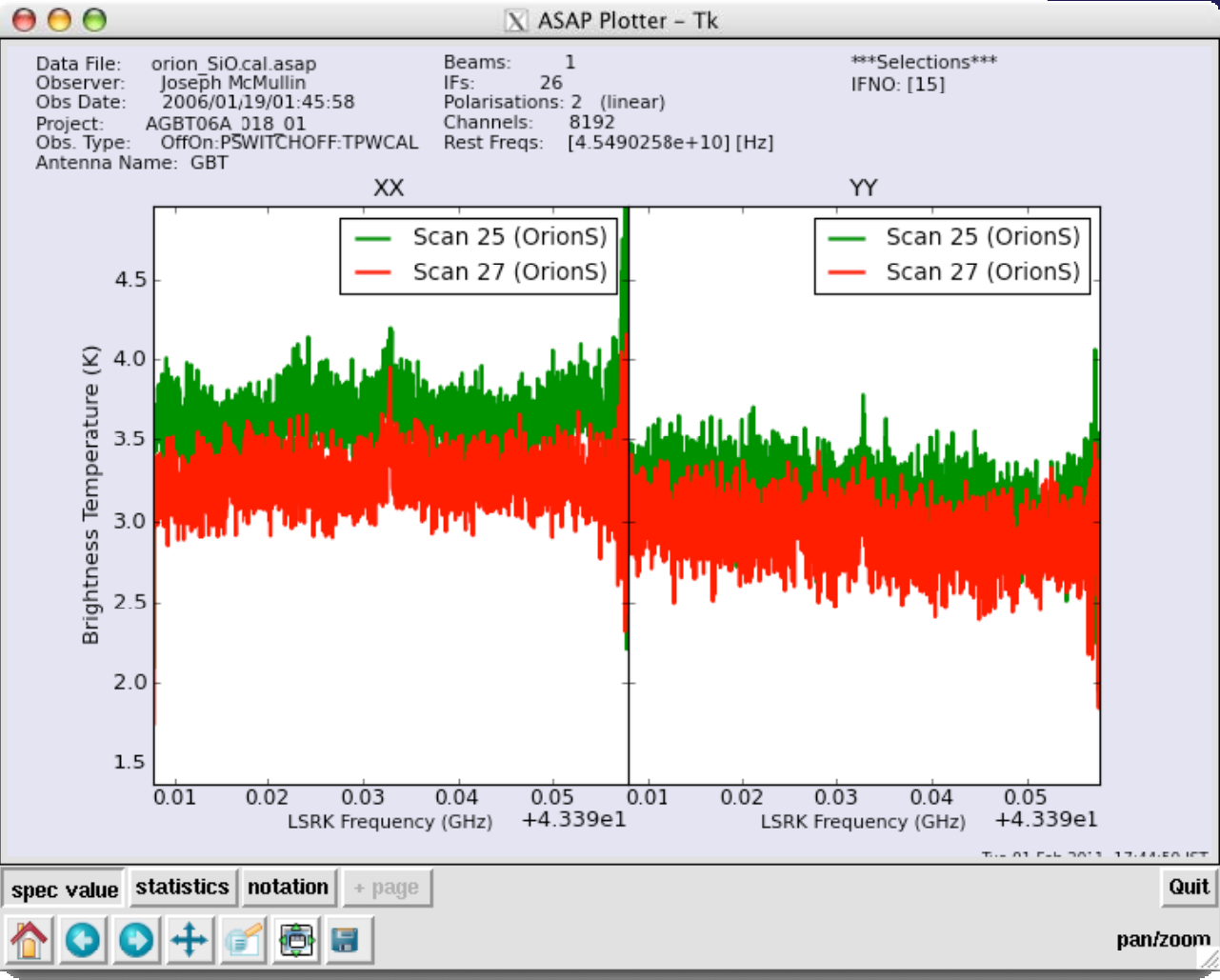
Orion spectra (calibrated)



```

# sdplot :: ASAP SD plotting task
Cor sdfile = 'orion_SiO.cal.asap' # name of the SD file
Sof antenna = 0 # antenna number
fluxunit = '' # units
specunit = 'GHz' # units
restfreq = '' # rest frequency
frame = '' # frequency frame
doppler = '' # doppler correction
scanlist = [] # list of scan numbers
field = '' # string
iflist = [15] # list of IF numbers
pollist = [] # list of poll numbers
scanaverage = True # average over scans
timeaverage = False # average over time
polaverage = False # average over polarizations
kernel = 'none' # type of kernel
plottype = 'spectra' # type of plot
stack = 's' # code for stacking
panel = 'p' # code for panel
flrange = [] # range of frequencies
sprange = [] # range of scan numbers
linecat = 'none' # control line catalog
linedop = 0.0 # doppler correction
colormap = 'none' # the colormap
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(...)

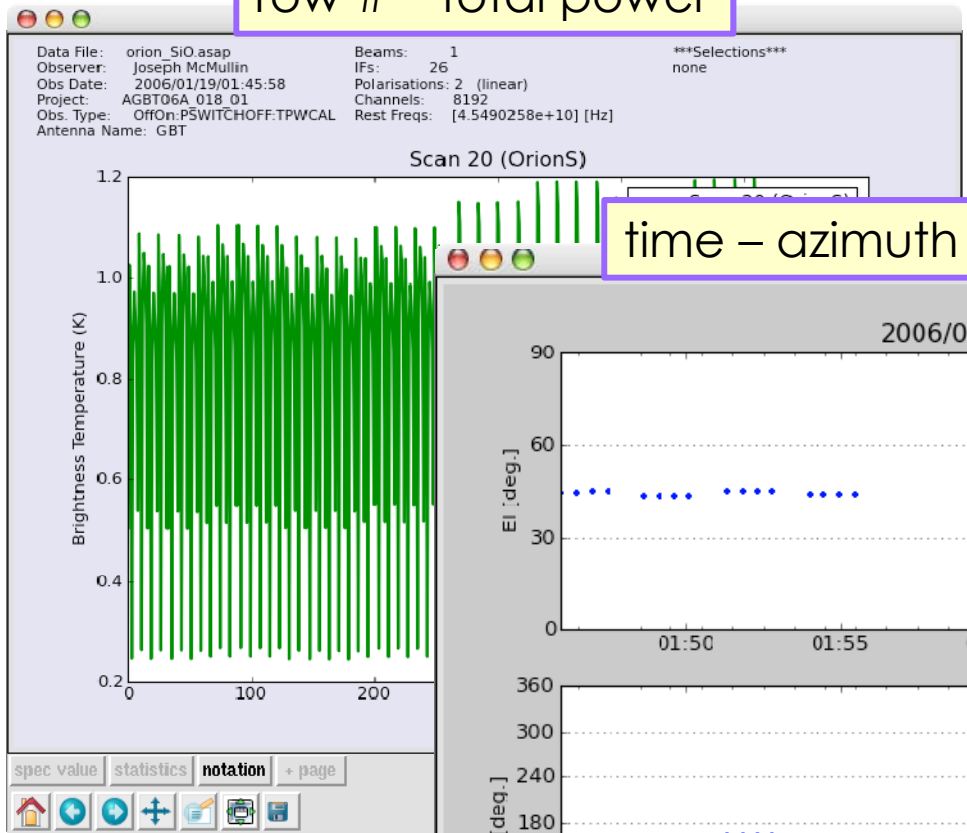
```



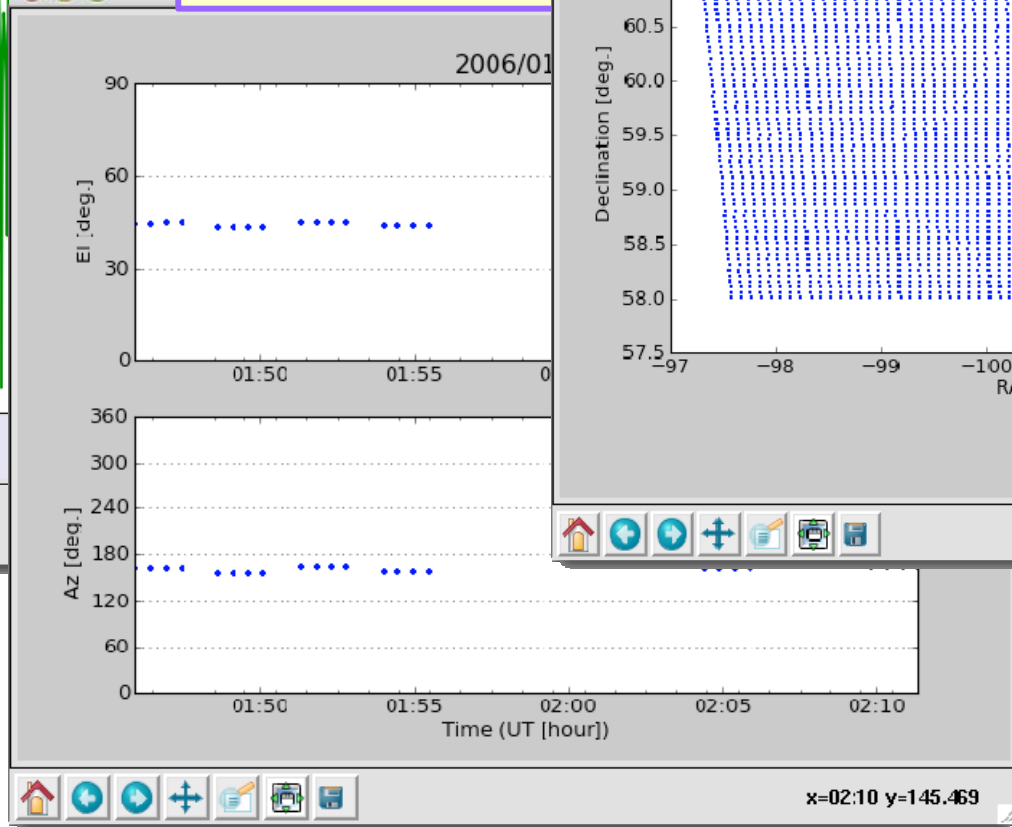
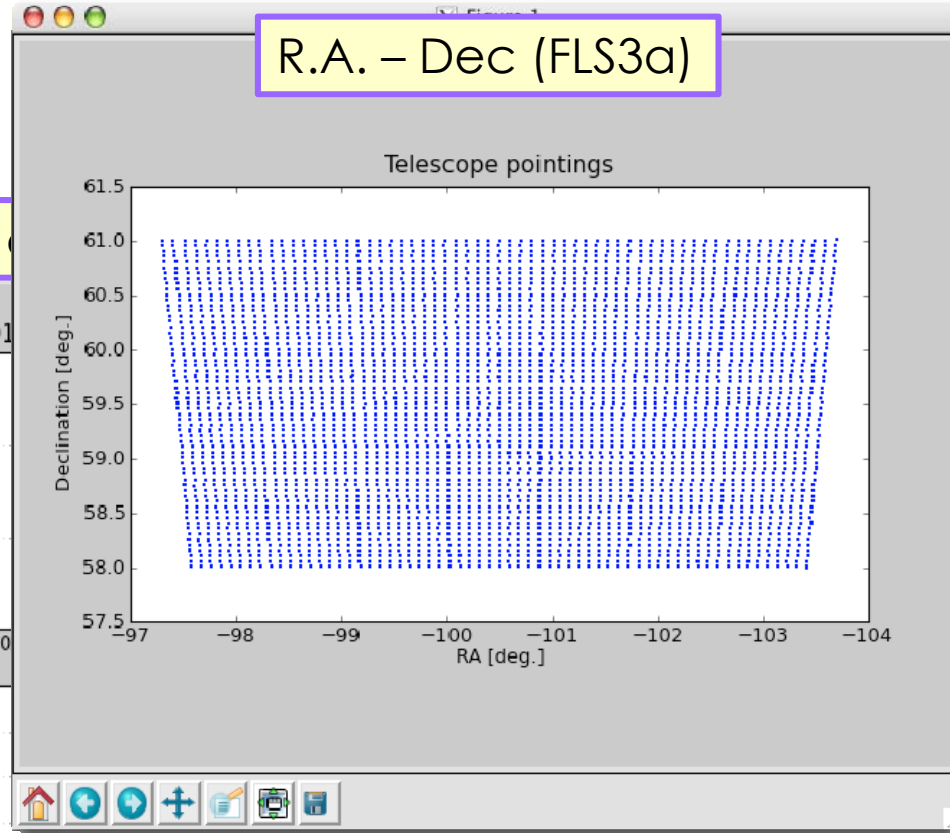
sdplot – plot scantable

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

row # – total power

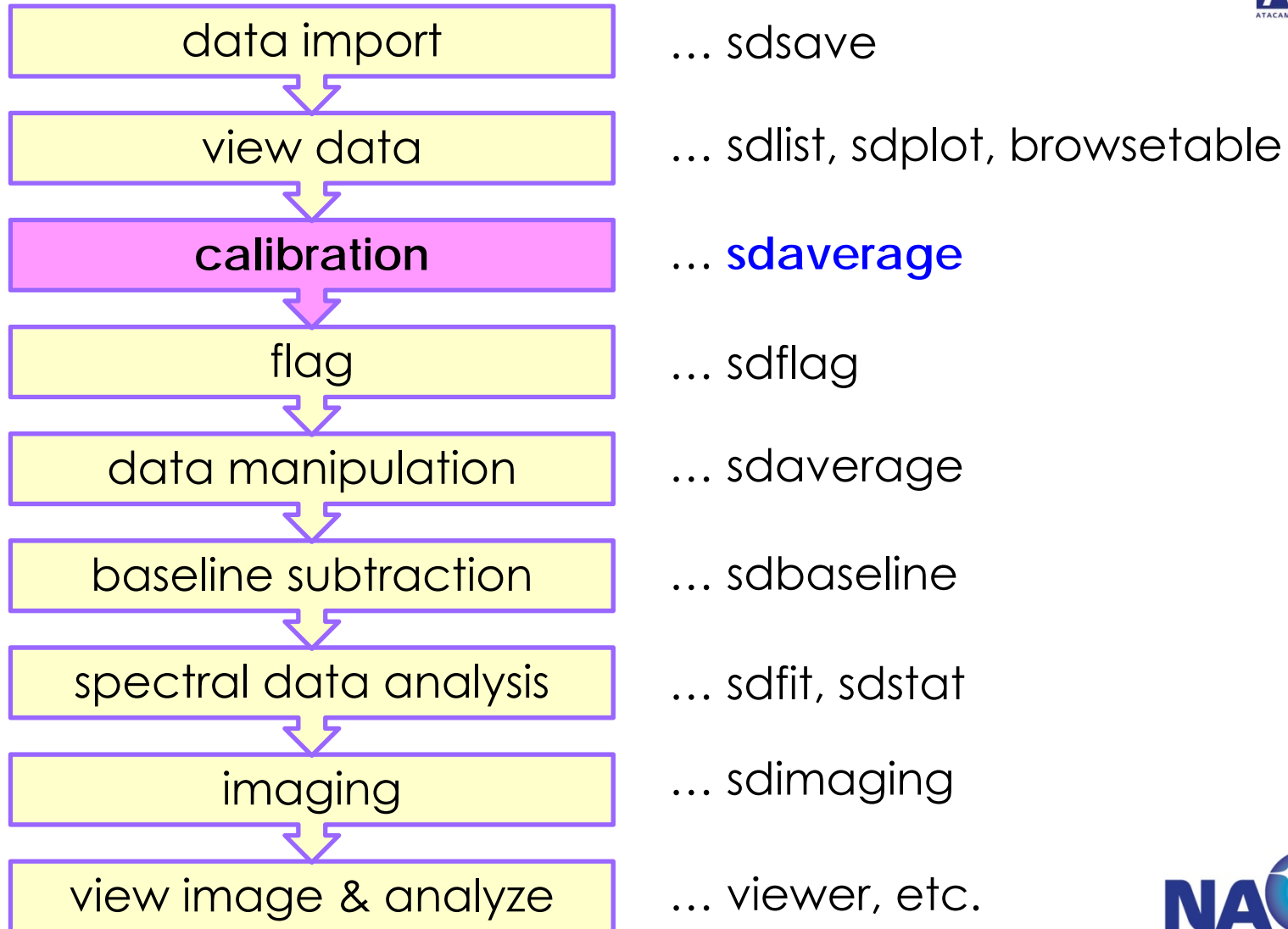


R.A. – Dec (FLS3a)



x=02:10 y=145.469

Single dish data reduction with CASA



sdaverage – calibration

ターミナル — ssh — 94x27

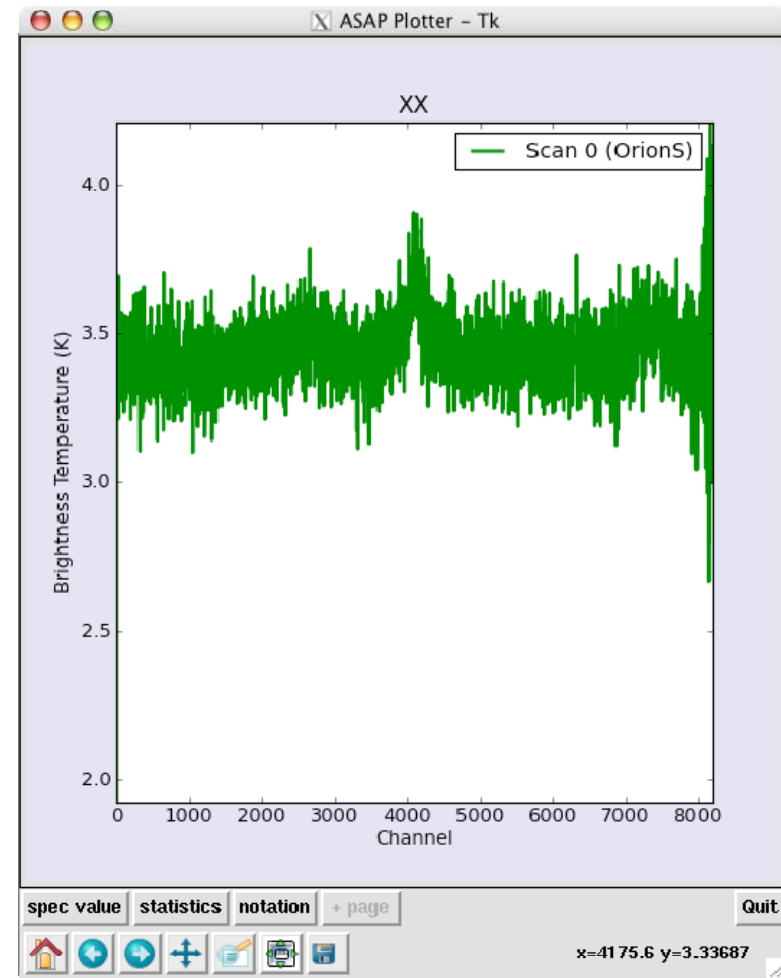
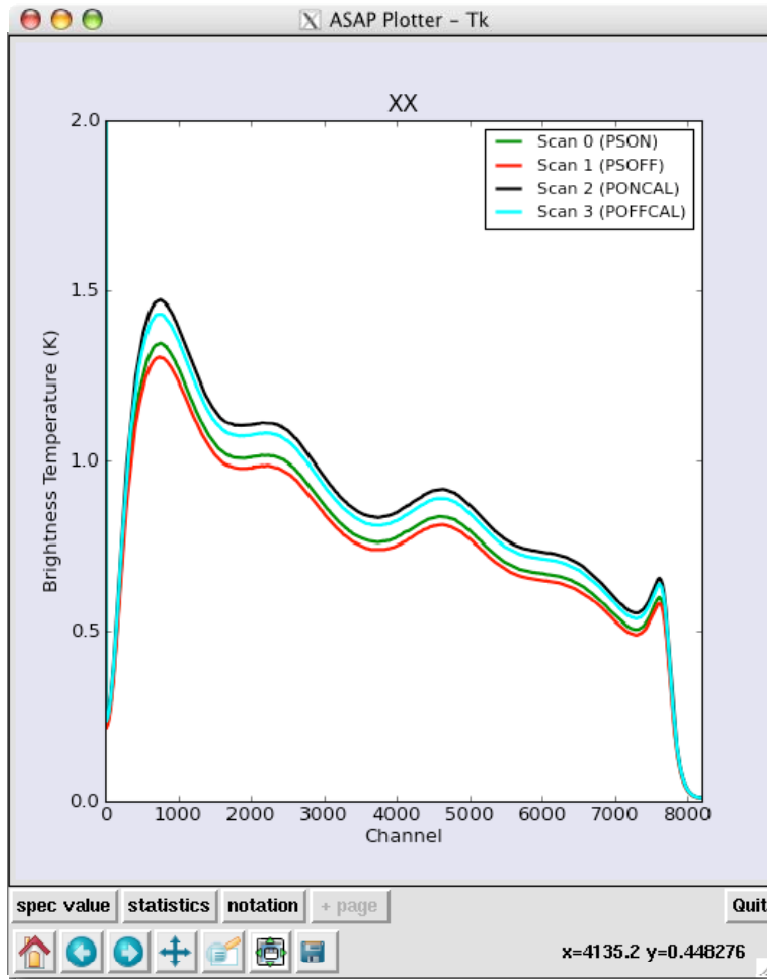
```
# sdaverage :: ASAP SD task: do data selection, calibration, and averaging
sdfile          = 'orion_Si0.asap' # name of input SD dataset
antenna         =          0       # antenna name or id (only effective for MS input)
fluxunit        = 'K'             # units for line flux (K,Jy) (=current)
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_Si0.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(...)
```

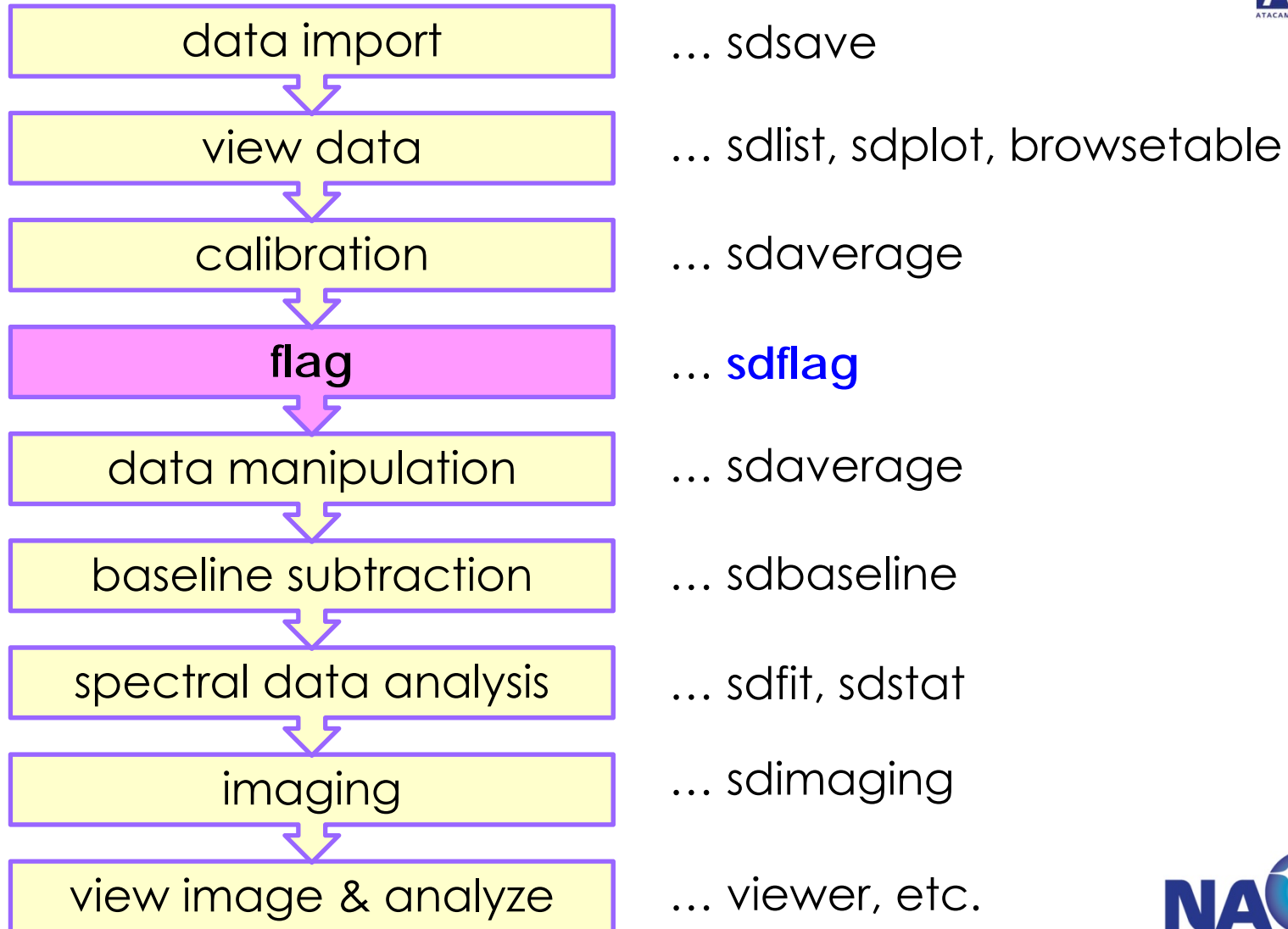
sdaverage – calibration

Observation

Calibrated



Single dish data reduction with CASA



sdflag – flag out bad data

```

ターミナル — ssh — 94x20

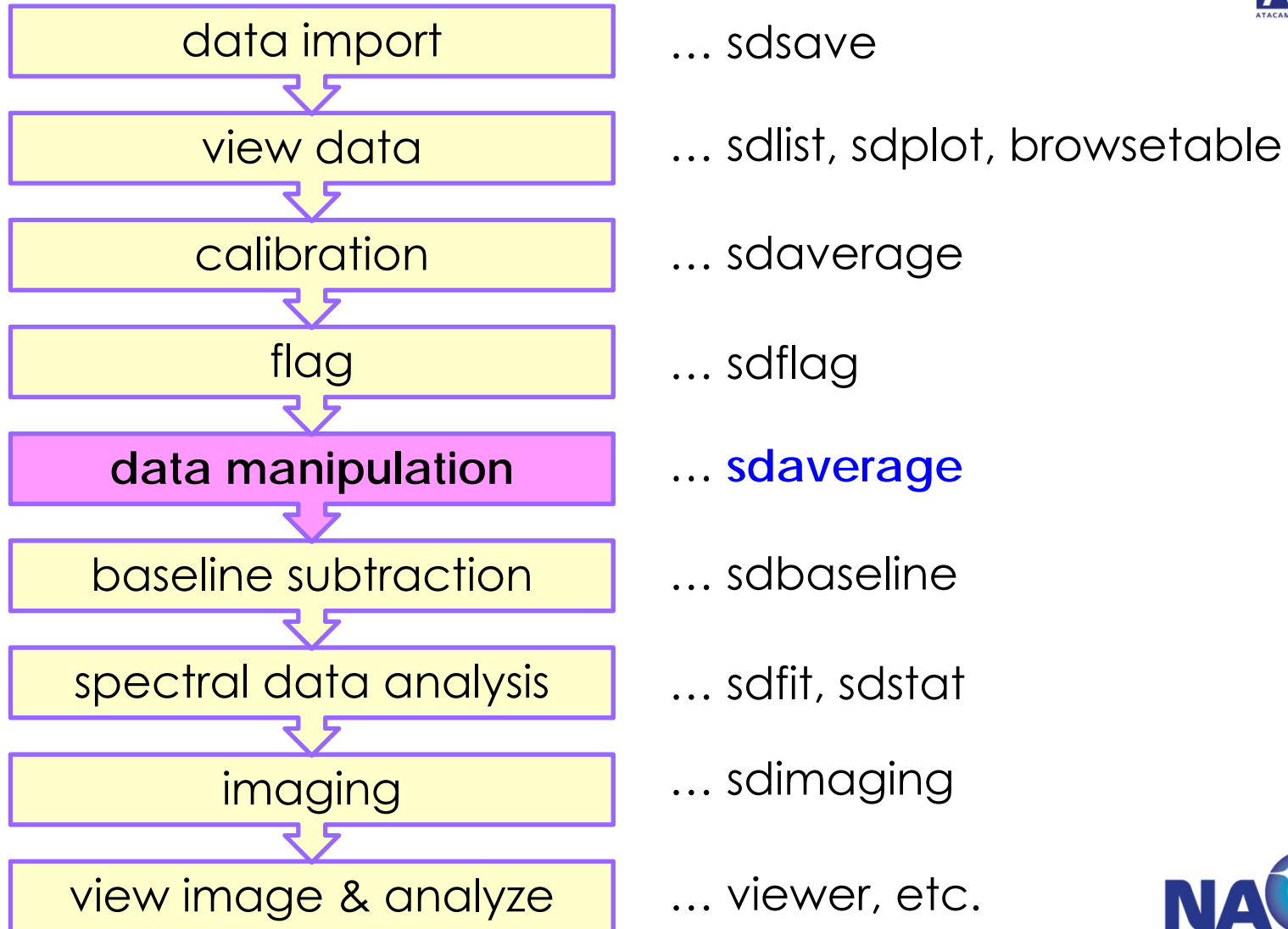
# sdflag :: ASAP SD spectral flagging task
sdfile           = 'orion_Si0.cal.asap' # name of input SD dataset
antenna          = 0                    # antenna name or id (only effective for MS input)
scanlist         = []                  # list of scans to use (e.g. [1,2,3,4])
field            = ''                  # string for selection by source name
iflist           = []                  # list of IF ids to select (e.g. [0,1])
pollist          = []                  # list of poller names to select
# (e.g. [0,1])
maskflaa        = [[0, 10], [8000, 8191]] # list of mask regions to flag/unflag
flagrow          = []                  # list of row IDs to apply row-based
clip             = False               # clip values outside specified range
flagmode         = 'flag'              # flag mode (flag,unflag)
outfile          = 'orion_Si0.fl.asap' # name of output file
outform          = 'ASAP'              # output file format (ASCII,MS,SDFITS,ASAP)
overwrite        = True                # overwrite the output file if already exists
plotlevel        = 0                   # control for plotting of results
async            = False               # If true the taskname must be started using
# sdflag(...)
  
```

by channel

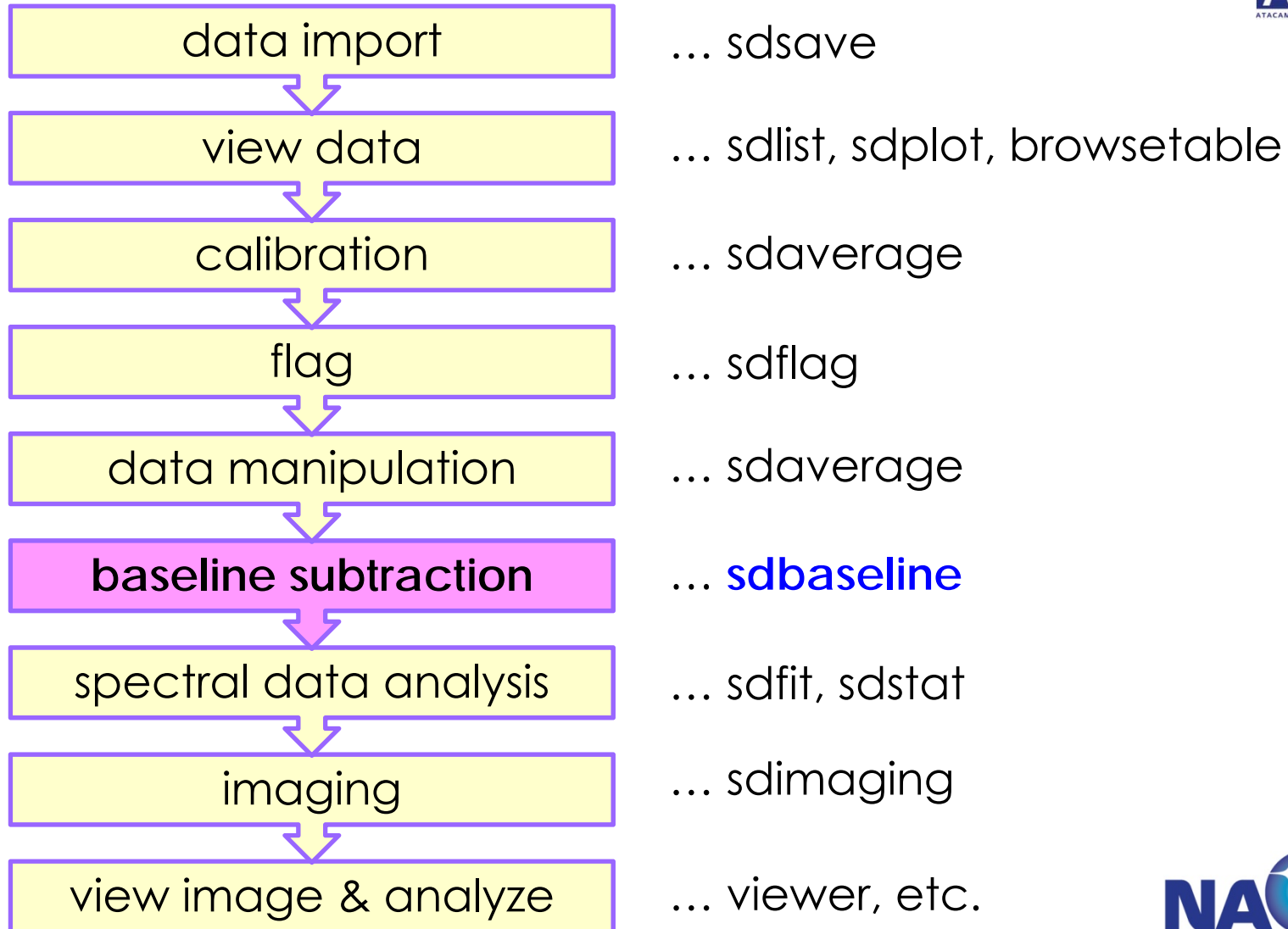
by row ID

by flux

Single dish data reduction with CASA



Single dish data reduction with CASA



sdbaseline – baseline subtraction

polynomial baseline fitting and subtraction

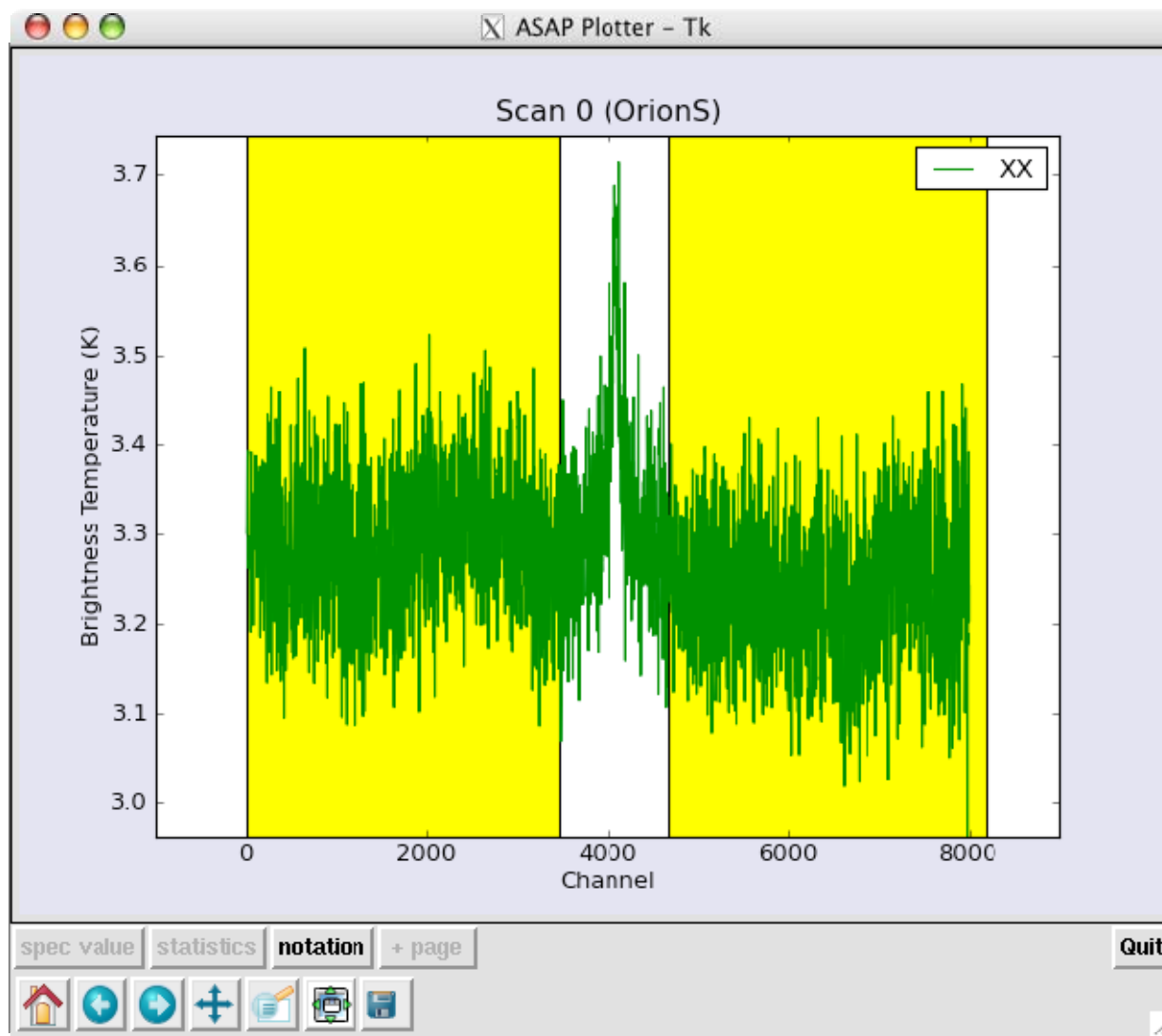
```
ターミナル — ssh — 85x27
# sdbaseline :: ASAP SD task: fit/remove a spectral baseline
sdfile          = 'orion_Si0.ave.asap' # name of input SD dataset
antenna         = 0                    # antenna name or id (only effective for MS
                                     # input)
fluxunit        = ''                  # 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          = []                  # F ids to select (e.g. [0,1])
pollist         = []                  # polarization ids to select (e.g.
tau             = 0.0                 # atmospheric optical depth for correction
blmode          = 'interact'          # mode for baseline fitting
blpoly          = 5                   # order of baseline polynomial
verify          = False               # verify the results of baseline fitting
masklist        = []                  # list of mask regions to INCLUDE in
                                     # BASELINE fit
outfile         = 'orion_Si0.bl.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
                                     # sdbaseline(...)
```

'auto', 'list',
or 'interact'

blmode = 'interact'
blpoly = 5

sdbaseline – baseline subtraction

interactive mask selection



draw rectangle

- **add** mask chans:

LEFT-mouse

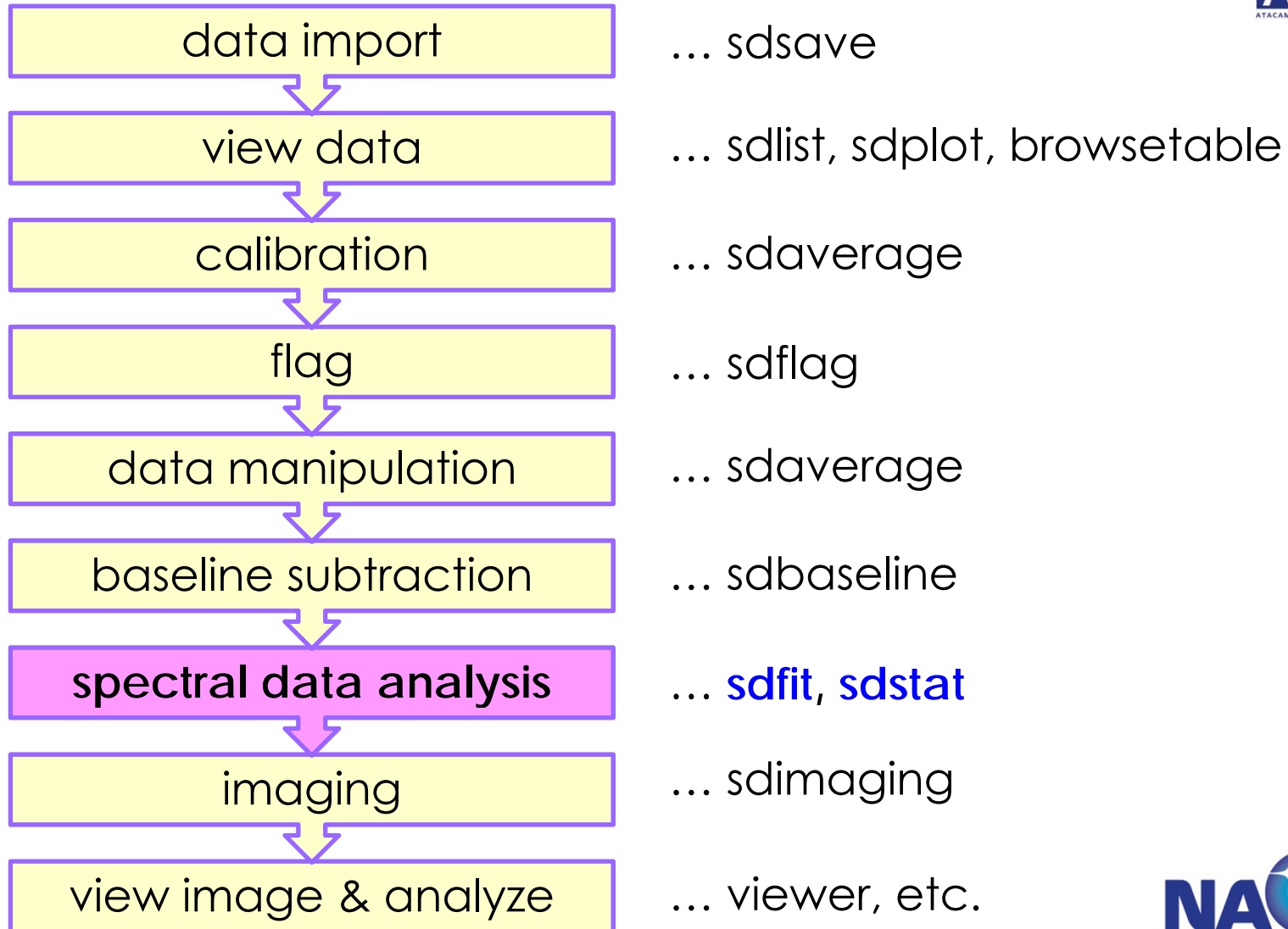
- **delete** mask chans:

RIGHT-mouse

current mask in
yellow

press "Enter" to finish
selection

Single dish data reduction with CASA



sdfit – fit spectral line

Gaussian or Lorentzian fitting

```

ターミナル — ssh — 85x27
# sdfit :: ASAP SD task: fit a spectral line
sdfile          = 'orion_Si0.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           = ''                  # field selection by source name
iflist          = []                  # ifids to select (e.g. [0,1])
pollist         = []                  # polarization ids to select (e.g.
                                     # [1])
fitfunc         = 'gauss'             # function for fitting
fitmode         = 'list'              # mode for fitting
maskline        = [3000, 5000]       # list of mask regions to INCLUDE in LINE
                                     # fitting
invertmask      = False               # invert mask (EXCLUDE masklist instead)
nfit            = [1]                 # list of number of gaussian/lorentzian
                                     # lines to fit in in maskline region
fitfile         = 'orion_Si0.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(...)

```

'auto', 'list',
or 'interact'

fitmode = 'list'
maskline = [3000, 5000]
invertmask = False
nfit = [1]

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

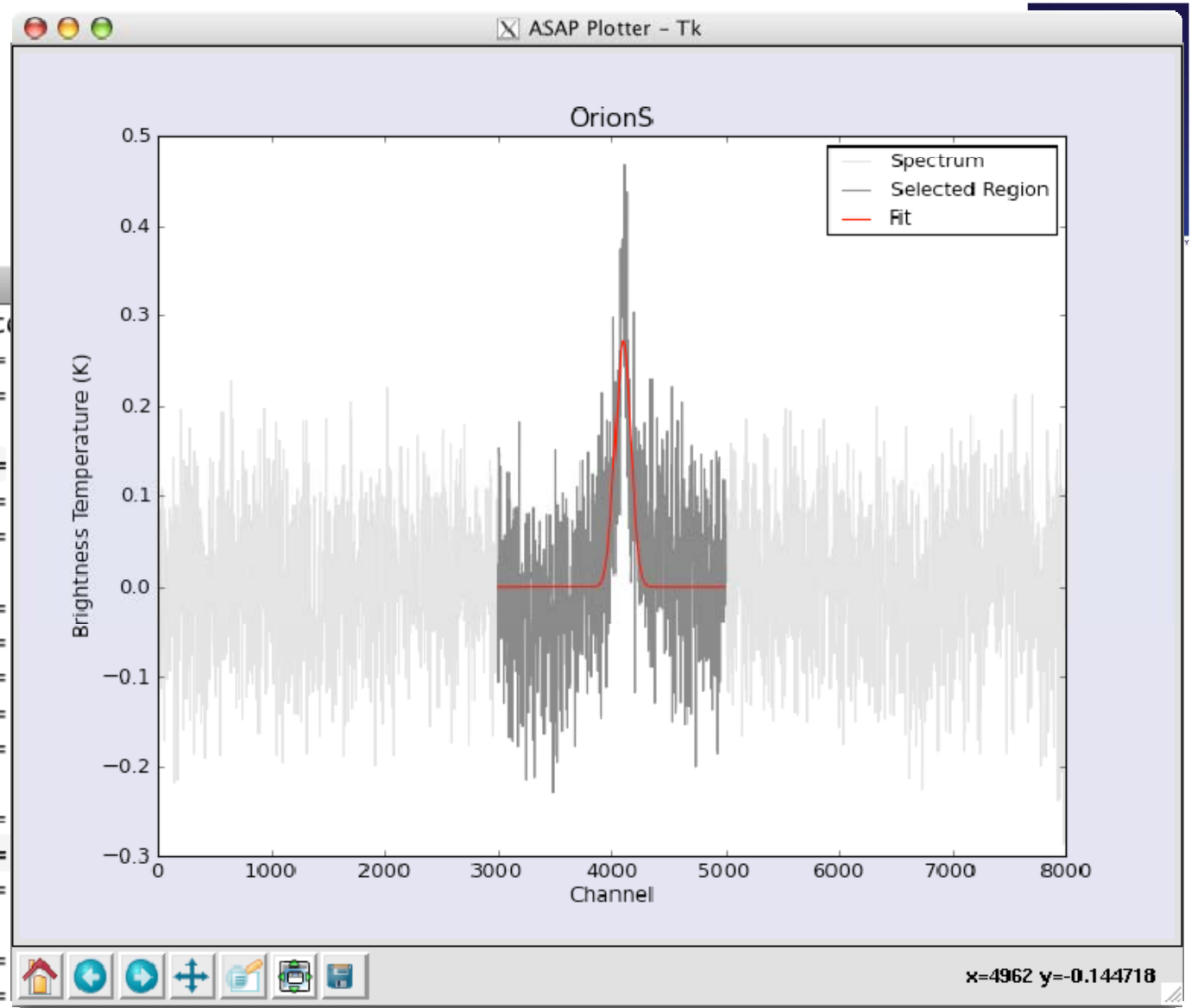
fluxunit =
specunit =
frame =

doppler =
scanlist =
field =
iflist =
pollist =

fitfunc =
fitmode =
maskline =

invertmask =
nfit =
```

```
fitfile = 'orion_Si0.fit.txt'
overwrite = True
plotlevel = 1
async = False
```



```
# lines to fit in in maskline region
# name of output file for fit results
# overwrite the fitfile if already exists
# control for plotting of results
# If true the taskname must be started using
# sdfit(...)
```

sdstat – calculate statistics

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

```
ターミナル - ssh - 91x19
# sdstat :: ASAP SD task: list statistics of spectral region
sdfile           = 'orion_Si0.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_Si0.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

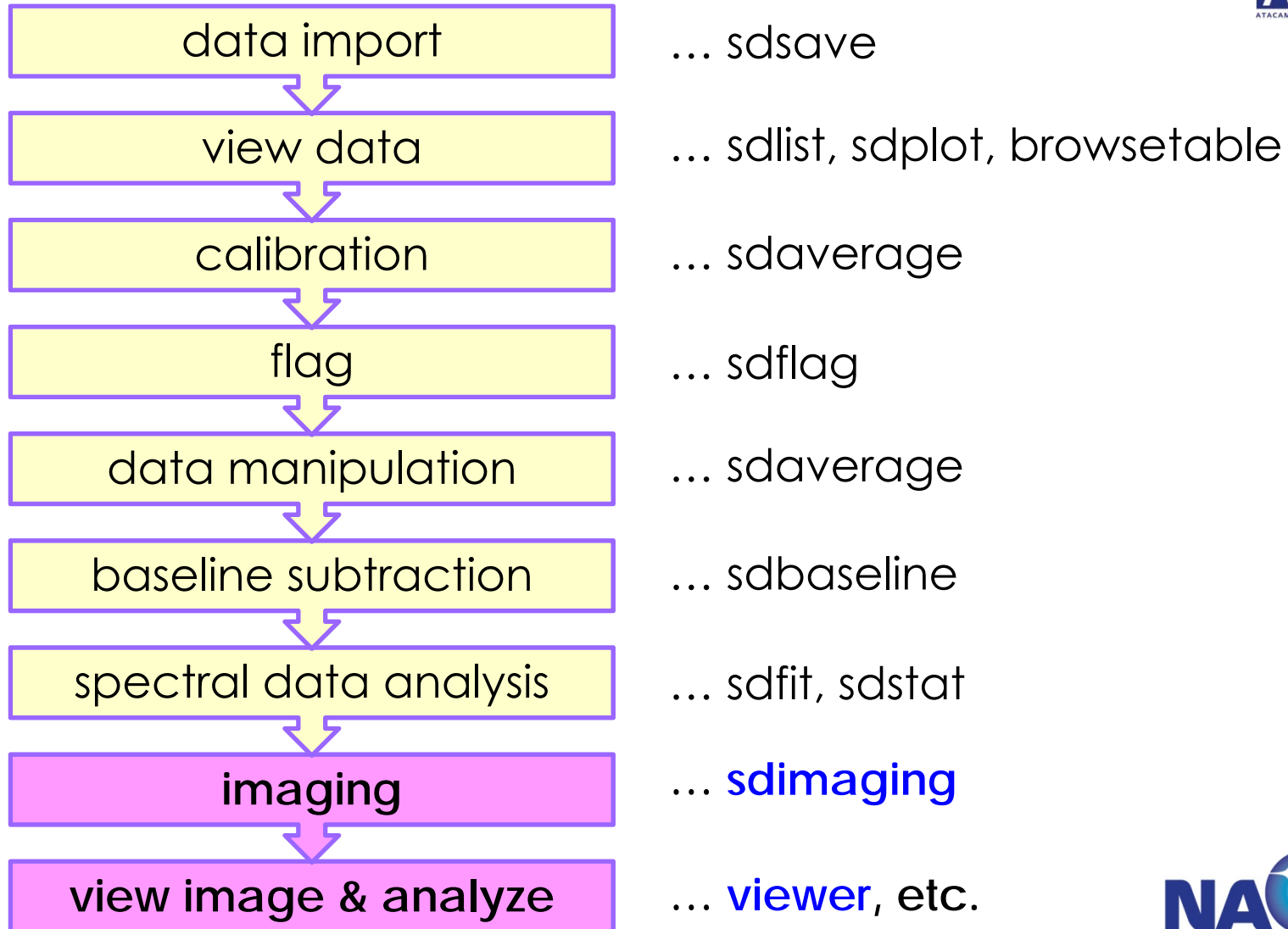
```

Log Messages (ana03:/home/sugimtkn/workcasa/singledish/reduction/casapy.log)
File Edit View
Search Message:
Filter: Time

Priority  Origin  Message
INFO     sdstat... #####
INFO     sdstat... ##### Begin Task: sdstat          #####
INFO     sdstat... sdstat:::casa
INFO     sdstat... Current fluxunit = K
INFO     sdstat... No need to convert fluxunits
INFO     sdstat... Using current frequency frame
INFO     sdstat... Using current doppler convention
INFO     sdstat... Using full region
INFO     sdstat... .....
INFO     sdstat... max
INFO     sdstat... .....
INFO     sdstat... Scan[0] (OrionS) Time[2006/01/19/02:07:54]:
INFO     sdstat... IF[15] = 0.469 (@ 4109.000 [channel])
INFO     sdstat... .....
INFO     sdstat... .....
INFO     sdstat... min
INFO     sdstat... .....
INFO     sdstat... Scan[0] (OrionS) Time[2006/01/19/02:07:54]:
INFO     sdstat... IF[15] = -0.286 (@ 7980.000 [channel])
INFO     sdstat... .....
  
```

Insert Message: Lock scroll

Single dish data reduction with CASA



sdsave (again) – convert data format

Important!

Convert data to **MeasurementSet** before imaging.

```
ターミナル - ssh - 91x16
# 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

Table Browser

File Edit View Tools Export Help

fls3a_HI.ms

	OBSERVATION_ID	PROCESSOR_ID	SCAN_NUMBER	STATE_ID	TIME	TIME_CENTROID	CALFCTR	FOAT_DATA
0	0	0	1	0	2003-11-12-...	2003-11-12-...	[0, 0]	[2, 1024] Float
1	0	0	1	0	2003-11-12-...	2003-11-12-...	[0, 0]	[2, 1024] Float
2	0	0	1	0	2003-11-12-...	2003-11-12-...	[0, 0]	[2, 1024] Float
3	0	0	1	0	2003-11-12-...	2003-11-12-...	[0, 0]	[2, 1024] Float
4	0	0	1	0	2003-11-12-...	2003-11-12-...	[0, 0]	[2, 1024] Float
5	0	0	1	0	2003-11-12-...	2003-11-12-...	[0, 0]	[2, 1024] Float
6	0	0	1	0	2003-11-12-...	2003-11-12-...	[0, 0]	[2, 1024] Float
7	0	0	1	0	2003-11-12-...	2003-11-12-...	[0, 0]	[2, 1024] Float
8	0	0	1	0	2003-11-12-...	2003-11-12-...	[0, 0]	[2, 1024] Float
9	0	0	1	0	2003-11-12-...	2003-11-12-...	[0, 0]	[2, 1024] Float

Restore Columns Resize Headers

PAGE NAVIGATION First << [1 / 16] >> Last 1 Go Loading 1000 rows.

Browsing table: fls3a_HI.ms

sdimaging – single dish imaging

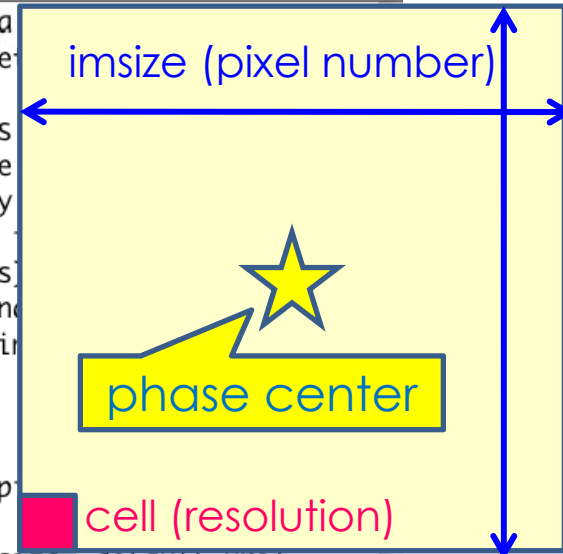
Input: MeasurementSet Output: **CASA image**

```

ターミナル — ssh — 88x31
# 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



The screenshot displays the CASA viewer interface. On the left, the 'Viewer Display Panel' shows a spectral line image with a color scale from red to white. The axes are labeled 'J2000 Declination' (ranging from 58°00' to 61°00') and 'J2000 Right Ascension' (ranging from 17h 32m to 08m). A green rectangle highlights a region of interest, and a velocity label '-1.56216 km/s' is shown. Below the image are control panels for navigation, zoom (64 to 100), and frame selection. At the bottom, a status bar shows the current image file 'fls3a_HI.full.im' and its parameters: '+0 Jy/beam', 'Pixel: 140 42 0 64', and '17:04:30.471 +58.38.18.525 I -1.56216 km/s (lsrk/radio)'. The right window, titled 'Image Profile - fls3a_HI.full.im', shows a 'Rectangle Region Profile' plot. The y-axis is '(Jy/beam)' from 0 to 7, and the x-axis is 'radio velocity (km/s)' from -300 to 200. A blue curve shows the profile, with a sharp peak at approximately 0 km/s. The plot includes a legend for 'fls3a_HI.full.im' and 'fls3a_HI.full.im-contour'. Below the plot, the coordinate is set to 'world' with values '17:29:03.33+58d00m51.056', and the velocity scale is 'LSRK'. Control options include 'Overlay', 'Relative', 'X auto scale', and 'Y auto scale'.

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>