Writing CASA Tasks using the CASA Toolkit

Miriam Krauss (NRAO)

Caltech CASA Radio Analysis Workshop
19-20 January 2012

What is a CASA Task?

- CASA offers over 100 built-in tasks
- Each task is written in Python (v.2.6)
- Most underlying functionality is from CASA Toolkit
- Task interface includes help and inp functionality

```
CASA <6>: tasklist
--------> tasklist()
Available tasks, organized by category (experimental tasks in parens ()
  deprecated tasks in curly brackets {}).
  Single Dish sd* tasks are available after asap_init() is run.


Import/export       Information         Editing             Manipulation
------------------  ------------------  ------------------  ------------------
exportfits          imhead              fixplanets          concat
exportuvfits        imstat              fixvis              conjugatevis
importaipscaltable  imval               flagautocorr        cvel
importasdm          listcal             flagcmd             fixvis
importfits          listhistory         flagdata            hanningsmooth
importfitsidi       listobs             flagmanager         imhead
importuvfits        listvis             msview              msmoments
importvla           plotms              plotms              plotms
(exportasdm)        plotuv              plotxy              plotxy
(importevla)        plotxy              (flagdata2)         split
(importgmrt)        vishead             (testautoflag)      testconcat
{importoldasdm}     visstat                                 uvcontsub
                    (listsdm)                               vishead
                                                            {uvcontsub2}
```


What is a CASA Task?

- CASA offers over 100 built-in tasks
- Each task is written in Python (v.2.6)
- Most underlying functionality is from CASA Toolkit
- Task interface includes help and inp functionality

```
CASA <6>: tasklist
--------> tasklist()
Available tasks, organized by category (experimental tasks in parens ()
  deprecated tasks in curly brackets {}).
  Single Dish sd* tasks are available after asap_init() is run.


Import/export        Information          Editing              Manipulation
------------------   ------------------   ------------------   ------------------
exportfits           imhead               fixplanets           concat
exportuvfits         imstat               fixvis               conjugatevis
importaipscaltable   imval                flagautocorr         cvel
importasdm           listcal              flagcmd              fixvis
importfits           listhistory          flagdata             hanningsmooth
importfitsidi        listobs              flagmanager          imhead
importuvfits         listvis              msview               msmoments
importvla            plotms               plotms               plotms
(exportasdm)         plotuv               plotxy               plotxy
(importevla)         plotxy               (flagdata2)          split
(importgmrt)         vishead              (testautoflag)       testconcat
{importoldasdm}      visstat                                   uvcontsub
                     (listsdm)                                 vishead
                                                               {uvcontsub2}
```

```
CASA <8>: help exportfits
--------> help(exportfits)
Help on exportfits task:

Convert a CASA image to a FITS file

        CASA-produced images can be written to disk for transporting
        to other software packages.  No subimaging of the fits image
        can be made with this task.

        Keyword arguments:
        imagename -- Name of input CASA image
                default: none; example: imagename='3C273XC1.image'
        fitsimage -- Name of ouput image FITS file
                default: none; example='3C273XC1.fits'
        velocity -- Use velocity (rather than frequency) as spectral axis
                default: False
        optical -- Use the optical (rather than radio) velocity convention
                default: False;
        bitpix -- Bits per pixel
                default: -32 (floating point)
                <Options: -32 (floating point), 16 (integer)>
        minpix -- Minimum pixel value
                default: 0 = autoscale
        maxpix -- Maximum pixel value
                default: 0 = autoscale
        overwrite -- Overwrite pre-existing imagename
                default=False; example: overwrite=True
        dropstokes -- Drop Stokes axis?
                default: False; example: dropstokes=True
        stokeslast -- Put Stokes axis last in header?
                default: True; example: stokeslast=False
        async --  Run asynchronously
                default = False;
```


What is a CASA Task?

- CASA offers over 100 built-in tasks
- Each task is written in Python (v.2.6)
- Most underlying functionality is from CASA Toolkit
- Task interface includes help and inp functionality

```
CASA <9>: inp exportfits
--------> inp(exportfits)
#  exportfits :: Convert a CASA image to a FITS file
imagename       =        ''       #  Name of input CASA image
fitsimage       =        ''       #  Name of output image FITS file
velocity        =     False       #  Use velocity (rather than frequency) as spectral axis
optical         =     False       #  Use the optical (rather than radio) velocity convention
bitpix          =       -32       #  Bits per pixel
minpix          =         0       #  Minimum pixel value (if minpix > maxpix, value is automatically determin
maxpix          =        -1       #  Maximum pixel value (if minpix > maxpix, value is automatically determin
overwrite       =     False       #  Overwrite pre-existing imagename
dropstokes      =     False       #  Drop the Stokes axis?
stokeslast      =      True       #  Put Stokes axis last in header?
async           =     False       #  If true the taskname must be started using exportfits(...)
```

```
CASA <6>: tasklist
--------> tasklist()
Available tasks, organized by category (experimental tasks in parens ()
  deprecated tasks in curly brackets {})
  Single Dish sd* tasks are available after asap_init() is run.

Import/export        Information        Editing            Manipulation
-----------------    -----------------  -----------------  -----------------
exportfits           imhead             fixplanets         concat
exportuvfits         imstat             fixvis             conjugatevis
importaipscaltable   imval              flagautocorr       cvel
importasdm           listcal            flagcmd            fixvis
importfits           listhistory        flagdata           hanningsmooth
importfitsidi        listobs            flagmanager        imhead
importuvfits         listvis            msview             msmoments
importvla            plotms             plotms             plotms
(exportasdm)         plotuv             plotxy             plotxy
(importevla)         plotxy             (flagdata2)        split
(importgmrt)         vishead            (testautoflag)     testconcat
{importoldasdm}      visstat                               uvcontsub
                     (listsdm)                             vishead
                                                           {uvcontsub2}
```

```
CASA <8>: help exportfits
--------> help(exportfits)
Help on exportfits task:

Convert a CASA image to a FITS file

        CASA-produced images can be written to disk for transporting
        to other software packages.  No subimaging of the fits image
        can be made with this task.

        Keyword arguments:
        imagename -- Name of input CASA image
                default: none; example: imagename='3C273XC1.image'
        fitsimage -- Name of ouput image FITS file
                default: none; example='3C273XC1.fits'
        velocity -- Use velocity (rather than frequency) as spectral axis
                default: False
        optical -- Use the optical (rather than radio) velocity convention
                default: False;
        bitpix -- Bits per pixel
                default: -32 (floating point)
                <Options: -32 (floating point), 16 (integer)>
        minpix -- Minimum pixel value
                default: 0 = autoscale
        maxpix -- Maximum pixel value
                default: 0 = autoscale
        overwrite -- Overwrite pre-existing imagename
                default=False; example: overwrite=True
        dropstokes -- Drop Stokes axis?
                default: False; example: dropstokes=True
        stokeslast -- Put Stokes axis last in header?
                default: True; example: stokeslast=False
        async --  Run asynchronously
                default = False;
```


What is a CASA Task?

- Tasks comprise two components:
  - Python script (task_<name>.py)
  - XML file for input and help information (<name>.xml)
- Looking at existing tasks a great way to learn!

task_imstat.py

File Edit Options Buffers Tools IM-Python Python Help

```python
from taskinit import *

def imstat(
    imagename=None, axes=None, region=None, box=None, chans=None,
    stokes=None, listit=None, verbose=None
):
    _myia = iatool.create()
    try:
        casalog.origin('imstat')
        _myia.open(imagename)
        mycsys = _myia.coordsys()
        reg = rg.frombcs(
            mycsys.torecord(), _myia.shape(),
            box, chans, stokes, "a", region
        )
        retValue = _myia.statistics(
            axes=axes, region=reg, list=listit,
            verbose=verbose, robust=True
        )
        _myia.done()
        return retValue
    except Exception, instance:
        _myia.done()
        casalog.post( '*** Error ***'+str(instance), 'SEVERE' )
        return {}
```

imstat.xml

File Edit Options Buffers Tools XML Help

```xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" ?>

<casaxml xmlns="http://casa.nrao.edu/schema/psetTypes.html"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://casa.nrao.edu/schema/casa.xsd
file:///opt/casa/code/xmlcasa/xml/casa.xsd">


<!-- This is the image statitics task                    -->

<task type="function" name="imstat" category="analysis, information">
<shortdescription>Displays statistical information from an image or image region</sho
rtdescription>

<input>
    <param type="string" name="imagename" kind="image" mustexist="true" >
        <description>Name of the input image</description>
        <value></value>
        <example>imagename=ngc5921_task.image</example>
    </param>
    <param type="any" name="axes">
        <any type="variant"/>
        <description>List of axes to e aluate statistics over. Default is all axes.</
description>
        <value>-1</value>
        <example>[0, 1]</example>
    </param>
    <param type="string" name="region">
        <description>Image Region or name. Use Viewer</description>
        <value></value>
    </param>

    <param type="string" name="box">
        <description>Select one or more box regions</description>
        <value></value>
    </param>

    <param type="string" name="chans" >
        <description>Select the channel(spectral) range</description>
        <value></value>
        <example>1.4126GHz~1.428GHz  # between 2 frequency</example>
        <example>1.4km/s             # above the given velocity</example>
        <example>!1~5                # Select all but the ange of channel numbers</ex
ample>
```

Caltech CASA Radio Analysis Workshop
19-20 January 2012

The CASA Toolkit

- Tools (mostly) written in C++
- Toolkit grouped into "packages" (General, Synthesis, Utility, Third Party, and Single-dish)
- Each package is subdivided into modules; e..g, General contains Images, Coordsys, MeasurementSet, Measures, and Quanta
- Each module one or more tools, which have associated functions
- Over 1000 tool functions available
- Documentation online: http://casa.nrao.edu/docs/CasaRef/CasaRef.html

1 Package General

1.1 images - Module

1.1.1 image - Tool

image.newimage - Function
image.newimagefromfile - Function
image.imagecalc - Function
image.collapse - Function
image.imageconcat - Function
image.fromarray - Function
image.fromascii - Function
image.fromfits - Function
image.fromimage - Function
image.fromshape - Function
image.maketestimage - Function
image.adddegaxes - Function
image.addnoise - Function
image.convolve - Function
image.boundingbox - Function
image.brightnessunit - Function
image.calc - Function
image.calcmask - Function
image.close - Function

The CASA Toolkit

- Tool names are abbreviated within CASA; e.g. "image" is ia
- Tool functions are called using this abbreviation with the function name: ia.open('myimage')
- In some cases, function may return a value (list, dictionary, etc.; in this case, a list of long integers):

```
CASA <15>: ia.fromshape(shape=[10,20,30])
  Out[15]: True

CASA <16>: imshape = ia.shape()

CASA <17>: imshape
  Out[17]: [10L, 20L, 30L]
```

Toolkit tool	CASA abbreviation
image	ia
calibrater	cb
componentlist	cl
coordsys	cs
flagger	fg
imagepol	po
imager	im
measures	me
ms	ms
msplot	mp
quanta	qa
regionmanager	rg
simulator	sm
table	tb
vpmanager	vp

The CASA Toolkit

- Two places to get Toolkit help
- Online help contains examples
- If missing or out-of-date, please notify the Helpdesk (http://casa.nrao.edu/help_desk_all.shtml)

```
CASA <13>: help ms.open
---------> help(ms.open)
Help on built-in function open:

open(...)
    Attach the ms tool to a measurement set table
    --- --- --- --- --- --- Parameters  --- --- --- --- --- ---
      thems:  Name of the measurement set table to open
      nomodify:  prevent changes to the measurement set true
      lock:  lock the table for exclusive use by this tool false
    --- --- --- --- --- --- --- --- --- --- --- --- --- --- ---
          thems
          nomodify  = true
          lock      = false
        ----------------------------------------


CASA <14>:
```

ms.open - Function

1.2.1 Attach the ms tool to a measurement set table

Description

Use this function when you have detached (using the close function) the ms tool from a measurement set table and wish to reattach to another measurement set table.

Arguments

Inputs		
thems	Name of the measurement set table to open	
	allowed:	string
	Default:	
nomodify	prevent changes to the measurement set	
	allowed:	bool
	Default:	true
lock	lock the table for exclusive use by this tool	
	allowed:	bool
	Default:	false

Returns
bool

Example

```
ms.open('3C273XC1.MS')
ms.close()
ms.open("anotherms", nomodify=False, lock=False)
```

Caltech CASA Radio Analysis Workshop
19-20 January 2012

The CASA Toolkit

- Good idea to create an instance of a tool before using
- Do this by declaring:

```
CASA <53>: myimage = iatool.create()

CASA <54>: myimage
  Out[54]: <casac.image object at 0x11edaf210>

CASA <55>: myimage.
Display all 110 possibilities? (y or n)
myimage.__class__                myimage.fitpolynomial            myimage.open
myimage.__delattr__              myimage.fitprofile               myimage.outputvariant
myimage.__doc__                  myimage.fromarray                myimage.pbcor
myimage.__format__               myimage.fromascii                myimage.pixelvalue
myimage.__getattribute__         myimage.fromfits                 myimage.putchunk
myimage.__hash__                 myimage.fromimage                myimage.putregion
myimage.__init__                 myimage.fromrecord               myimage.rebin
myimage.__new__                  myimage.fromshape                myimage.regrid
myimage.__reduce__               myimage.getchunk                 myimage.remove
myimage.__reduce_ex__            myimage.getregion                myimage.removefile
myimage.__repr__                 myimage.getslice                 myimage.rename
myimage.__setattr__              myimage.hanning                  myimage.reorder
myimage.__sizeof__               myimage.haslock                  myimage.replacemaskedpixels
```

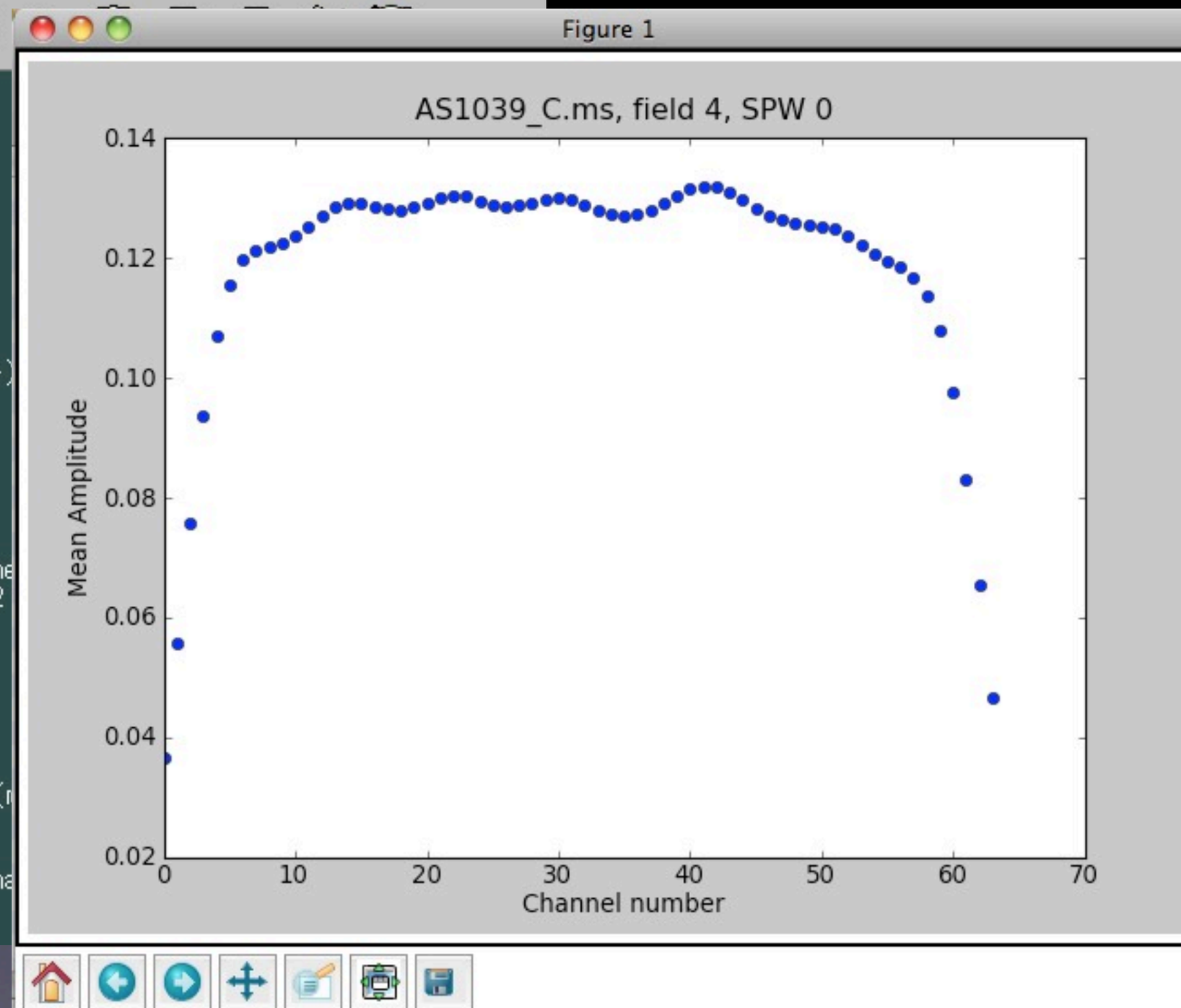
```
CASA <56>: myimage.fromshape('newim',[1,2,3])
  Out[56]: True
```


CASA Toolkit Example:

```
msName = 'AS1039_C.ms'
fieldID = 4

import numpy as np

myMS = mstool.create()
myMS.open(msName)
myMS.selectinit(datadescid = 2)
myMS.selectchannel(64,0,1,1)
myMS.select({'field_id':[fieldID]})

data = myMS.getdata('DATA')
flag = myMS.getdata('FLAG')

RRamp = []
for i in range(0, 64):
    RR = data['data'][0,i,:][np.where(flag['flag'][0,i,:] == False)[0]]
    RRampt = (np.sqrt((RR.real)**2 + (RR.imag)**2)).mean()
    RRamp.append(RRampt)

myMS.close()

pl.clf()
pl.plot(range(0,64), RRamp, 'bo')
pl.title('%s, field %i, SPW 0' % (msName, fieldID))
pl.xlabel('Channel number')
pl.ylabel('Mean Amplitude')
pl.savefig('field4.spw0.png', format='png')
pl.show
```


CASA Toolkit Example:

```
msName = 'AS1039_C.ms'
fieldID = 4

import numpy as np

myMS = mstool.create()
myMS.open(msName)
myMS.selectinit(datadescid = 2)
myMS.selectchannel(64,0,1,1)
myMS.select({'field_id':[fieldID]})

data = myMS.getdata('DATA')
flag = myMS.getdata('FLAG')

RRamp = []
for i in range(0, 64):
    RR = data['data'][0,i,:][np.whe
    RRampt = (np.sqrt((RR.real)**2
    RRamp.append(RRampt)

myMS.close()

pl.clf()
pl.plot(range(0,64), RRamp, 'bo')
pl.title('%s, field %i, SPW 0' % (r
pl.xlabel('Channel number')
pl.ylabel('Mean Amplitude')
pl.savefig('field4.spw0.png', forma
pl.show
```



Building & Importing Tasks

- Once you have a new task written (task_<name>.py and <name>.xml), simple to build and import
- Script buildmytasks does the work
- Run from command line or within CASA itself (be sure you're in right directory)
- buildmytasks creates mytasks.py
- Import this in CASA: execfile('mytasks.py')
- tasklist will show new tasks under "User defined tasks"
- inp and help will work as with other CASA tasks

```
CASA <49>: !buildmytasks

CASA <50>: execfile('mytasks.py')
```

```
User defined tasks
------------------
applycal
applycal2
findsources
findzeros
flagcmdtest
flagdata
getmodelimage
importevla3
listsdm
listsdm2
mkmodelimage
newtask
opacitycalc
peel2
plotwx
wbpipeline
widarcheck
```


Building & Importing Tasks

- Also possible to auto-load homegrown tasks on startup
- Create a file called "init.py" in your ~/.casa directory
- In this file, add the same execfile command you used to import "mytasks.py"
- When CASA starts, will execute commands in init.py
- Could load tasks from several different directories
- Caution: if loading fails, CASA will abort on startup

- NOTE: once a task has been loaded, if changes are made, CASA must be restarted for these changes to be seen

Example 1: findsources

- Creates a file that can be used to make "outlier" field images
- Queries VizieR catalogs based on either RA or Dec, or field center of given MS
- Uses the "vizquery" tool, available at http://cdsarc.u-strasbg.fr/doc/cdsclient.html
- Can use any VizieR catalog (1306 are radio-related!); able to input a flux limit
- This kind of functionality can't easily be added to CASA because of reliance on external package
- But also not likely to add the functionality of vizquery to CASA!
- Perfect for a self-written task...

Example 1: findsources

```
CASA <2>: inp findsources
--------> inp(findsources)
#  findsources :: Find sources using VizieR catalogs and create an outlierfile for input to clean
outlierfile        =         ''        #  Name of output outlierfile to produce
querycenter        =  'byfield'        #  Central position for catalog query (byfield, bycoord)
     vis           =         ''        #  Name of input visibility file
     field         =         ''        #  Field name or index

catalog            =     'NVSS'        #  Catalog name (see http://vizier.u-strasbg.fr/viz-bin/VizieR)
fluxparam          =     'S1.4'        #  VizieR catalog flux density key
fluxlim            =       0.0         #  VizieR catalog flux density limit
minradius          =       0.0         #  Inner radius for catalog search (arcminutes)
maxradius          =      10.0         #  Outer radius for catalog search (arcminutes)
imsize             = [256, 256]        #  x and y image sizes in pixels
clobber            =     False         #  Clobber/overwrite existing files?
async              =     False         #  If true the taskname must be started using findsources(...)

CASA <3>: querycenter='bycoord'

CASA <4>: ra=134.1523

CASA <5>: dec=55.3141

CASA <6>: outlierfile='testout.reg'

CASA <7>: go
--------> go()
Executing:  findsources()

CASA <8>: !cat testout.reg
C 0 256 256 08 56 48.19 +55 18 09.9
C 1 256 256 08 56 44.47 +55 28 32.2

CASA <9>:
```


Example 1: findsources

Log Messages (gygax:/users/mkrauss/casapy.log)

File Edit View

Search Message: Filter: Time

Time	Priority	Origin	Message
2012-01-19 04:19:34	INFO	::casa	
2012-01-19 04:19:53	INFO	::casa	
2012-01-19 04:19:57	INFO	casa::::casa	---
2012-01-19 04:19:57	INFO	casa::::casa	CASA Version 3.3.0 (release r16856)
2012-01-19 04:19:57	INFO	casa::::casa	Tagged on: Wed, 02 Nov 2011
2012-01-19 04:20:39	INFO	findsources:…	
2012-01-19 04:20:39	INFO	findsources:…	##
2012-01-19 04:20:39	INFO	findsources:…	##### Begin Task: findsources #####
2012-01-19 04:20:39	INFO	findsources:…	
2012-01-19 04:20:39	INFO	findsources:…	Input parameters:
2012-01-19 04:20:39	INFO	findsources:…	ra = 134.152300
2012-01-19 04:20:39	INFO	findsources:…	dec = 55.314100
2012-01-19 04:20:39	INFO	findsources:…	outlierfile = testout.reg
2012-01-19 04:20:39	INFO	findsources:…	catalog = NVSS
2012-01-19 04:20:39	INFO	findsources:…	fluxparam = S1.4
2012-01-19 04:20:39	INFO	findsources:…	fluxlim = 0.000
2012-01-19 04:20:39	INFO	findsources:…	minradius = 0.00 arcmin
2012-01-19 04:20:39	INFO	findsources:…	maxradius = 10.00 arcmin
2012-01-19 04:20:39	INFO	findsources:…	imsize = [256, 256]
2012-01-19 04:20:39	INFO	findsources:…	clobber = False
2012-01-19 04:20:39	INFO	findsources:…	Querying VizieR catalog NVSS around position RA = 08 56 36.55, Dec = +055 18 50.760
2012-01-19 04:20:39	INFO	findsources:…	Writing VizieR query file to VizieR_query-2012.1.18-21h20m38s.txt and
2012-01-19 04:20:39	INFO	findsources:…	VizieR output to VizieR_return-2012.1.18-21h20m38s.txt
2012-01-19 04:20:44	INFO	findsources:…	2 sources written to outlierfile testout.reg
2012-01-19 04:20:44	INFO	findsources:…	
2012-01-19 04:20:44	INFO	findsources:…	##### End Task: findsources #####
2012-01-19 04:20:44	INFO	findsources:…	##

Example 2: mkpipeline

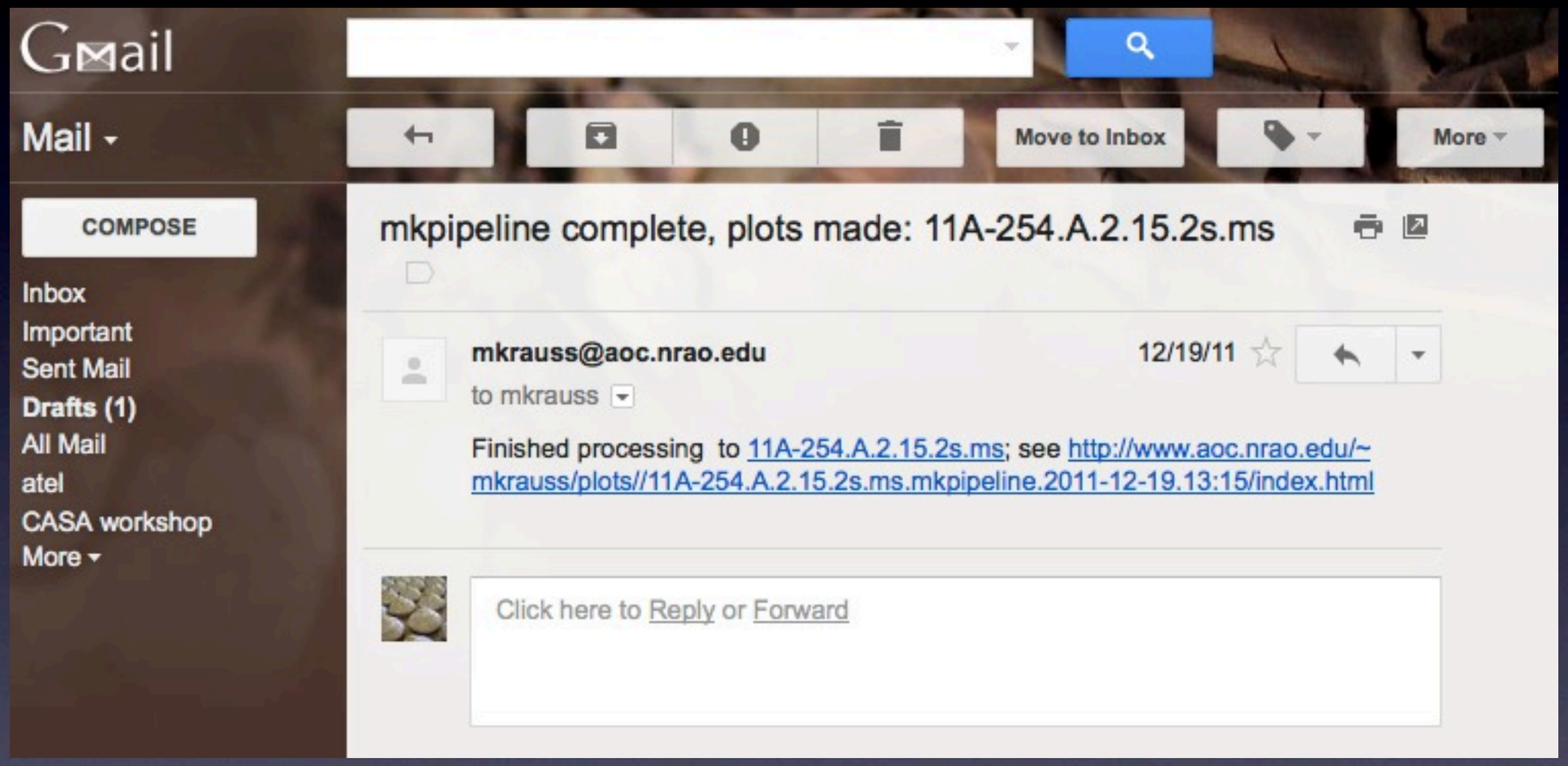
- Basic continuum pipeline for EVLA data
- End-to-end first-cut processing & imaging
- Creates web-page output and sends email notification when complete
- Calls tasks as well as tools
- Over 2500 lines long and includes a number of member functions for better organization
- Writes to a separate, dedicated log file
- May be run for "initial processing" (starting from raw SDM data) or "reprocessing" (an MS which has had basic flags applied)

Example 2: mkpipeline

```
CASA <3>: inp mkpipeline
--------> inp(mkpipeline)
#  mkpipeline :: Task for checking and processing EVLA continuum data.

mode            = 'initproc'      #  Processing mode (initproc, reproc)
     sdmname    =         ''      #  Name of input SDM directory for processing
     rootname   =         ''      #  Root name for output files
     band       =         ''      #  Band name (L, S, C, X, Ku, K, Ka, Q)
     dummy      =      False      #  Are the first two scans within this band dummies, to be ignored?
     checkzeros =       True      #  Check data for presence of zeros?
     timeave    =       True      #  Average data in time?

flaglist        =         ''      #  List of flagging commands (flagcmd formatted)
doplot          =       True      #  Plot raw data and calibration products?
calimage        =       True      #  Image the calibration sources?
email           =         ''      #  Email address to send notification when pipeline has completed
webdir          =         ''      #  Web space to which output HTML pages and plots will be copied
http            =         ''      #  Root http address where plots will appear
async           =      False      #  If true the taskname must be started using mkpipeline(...)

CASA <4>: sdmfile='AS1039_sb1382796_2_000.55368.51883247685'

CASA <5>: rootname='AS1039_C'

CASA <6>: band='C'

CASA <7>: checkzeros=False

CASA <8>: email='mkrauss@nrao.edu'

CASA <9>: dummy=True

CASA <10>: go
---------> go()
Executing:  mkpipeline()
```

```
2012-01-19 04:34:22  INFO   mkpipeline::…  ##########################################
2012-01-19 04:34:22  INFO   mkpipeline::…  ##### Begin Task: mkpipeline        #####
2012-01-19 04:34:22  INFO   mkpipeline::…
2012-01-19 04:34:22  INFO   mkpipeline::…  Running mkpipeline v 1.01, 6 November 2011
2012-01-19 04:34:22  INFO   mkpipeline::…  See AS1039_C.mkpipeline.log for ongoing log output
```


Example 2: mkpipeline



Example 2: mkpipeline

Plots and log files for 11A-277.K.30nov11.10s.ms:

Observation metadata:

- Antenna position diagram
- Weather data
- Online flags

Raw data:

- Raw amplitudes and phases vs. frequency, SPW 0~7
- Raw amplitudes and phases vs. frequency, SPW 8~15

Calibration:

- Initial phase solution plots
- Bandpass plots: SPW 0~7
- Antenna-based gain calibration plots: SPW 0~7
- Flux-scaled gain calibration plots: SPW 0~7
- Bandpass plots: SPW 8~15
- Antenna-based gain calibration plots: SPW 8~15
- Flux-scaled gain calibration plots: SPW 8~15
- Calibrated data: field 0
- Calibrated data: field 3
- Calibrated data: field 5

Imaging:

- Cleaned images

Log files:

- Select information from log file (also see below)
- Complete log file

```
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Task inputs:
2011-12-01 18:18:11     INFO    mkpipeline::::casa      mode = initproc
2011-12-01 18:18:11     INFO    mkpipeline::::casa      rootname = 11A-277.K.30nov11
2011-12-01 18:18:11     INFO    mkpipeline::::casa      sdmname = 11B-234.sb5969333.eb5978214.55895.501138101856
2011-12-01 18:18:11     INFO    mkpipeline::::casa      band = K
2011-12-01 18:18:11     INFO    mkpipeline::::casa      dummy = False
2011-12-01 18:18:11     INFO    mkpipeline::::casa      checkzeros = False
2011-12-01 18:18:11     INFO    mkpipeline::::casa      timeave = True
2011-12-01 18:18:11     INFO    mkpipeline::::casa      doplot = True
2011-12-01 18:18:11     INFO    mkpipeline::::casa      calimage = False
2011-12-01 18:18:11     INFO    mkpipeline::::casa      email = mkrauss@nrao.edu
2011-12-01 18:18:11     INFO    mkpipeline::::casa      webdir = /home/www.aoc.nrao.edu/homes/mkrauss/plots
2011-12-01 18:18:11     INFO    mkpipeline::::casa      http = http://www.aoc.nrao.edu/~mkrauss/plots/
2011-12-01 18:18:11     INFO    mkpipeline::::casa
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Bandpass calibrator fields: [3]
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Gain calibrator fields: [0, 5]
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Flux calibrator fields: [3]
2011-12-01 18:18:11     INFO    mkpipeline::::casa      All calibrator fields: [0, 3, 5]
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Science target fields: [6]
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Fields to image: [6]
2011-12-01 18:18:11     INFO    mkpipeline::::casa
```

Example 2: mkpipeline

Plots and log files for 11A-277.K.30nov11.10s.ms:

Observation metadata:

- Antenna position diagram
- Weather data
- Online flags

Raw data:

- Raw amplitudes and phases vs. frequency, SPW 0~7
- Raw amplitudes and phases vs. frequency, SPW 8~15

Calibration:

- Initial phase solution plots
- Bandpass plots: SPW 0~7
- Antenna-based gain calibration plots: SPW 0~7
- Flux-scaled gain calibration plots: SPW 0~7
- Bandpass plots: SPW 8~15
- Antenna-based gain calibration plots: SPW 8~15
- Flux-scaled gain calibration plots: SPW 8~15
- Calibrated data: field 0
- Calibrated data: field 3
- Calibrated data: field 5

Imaging:

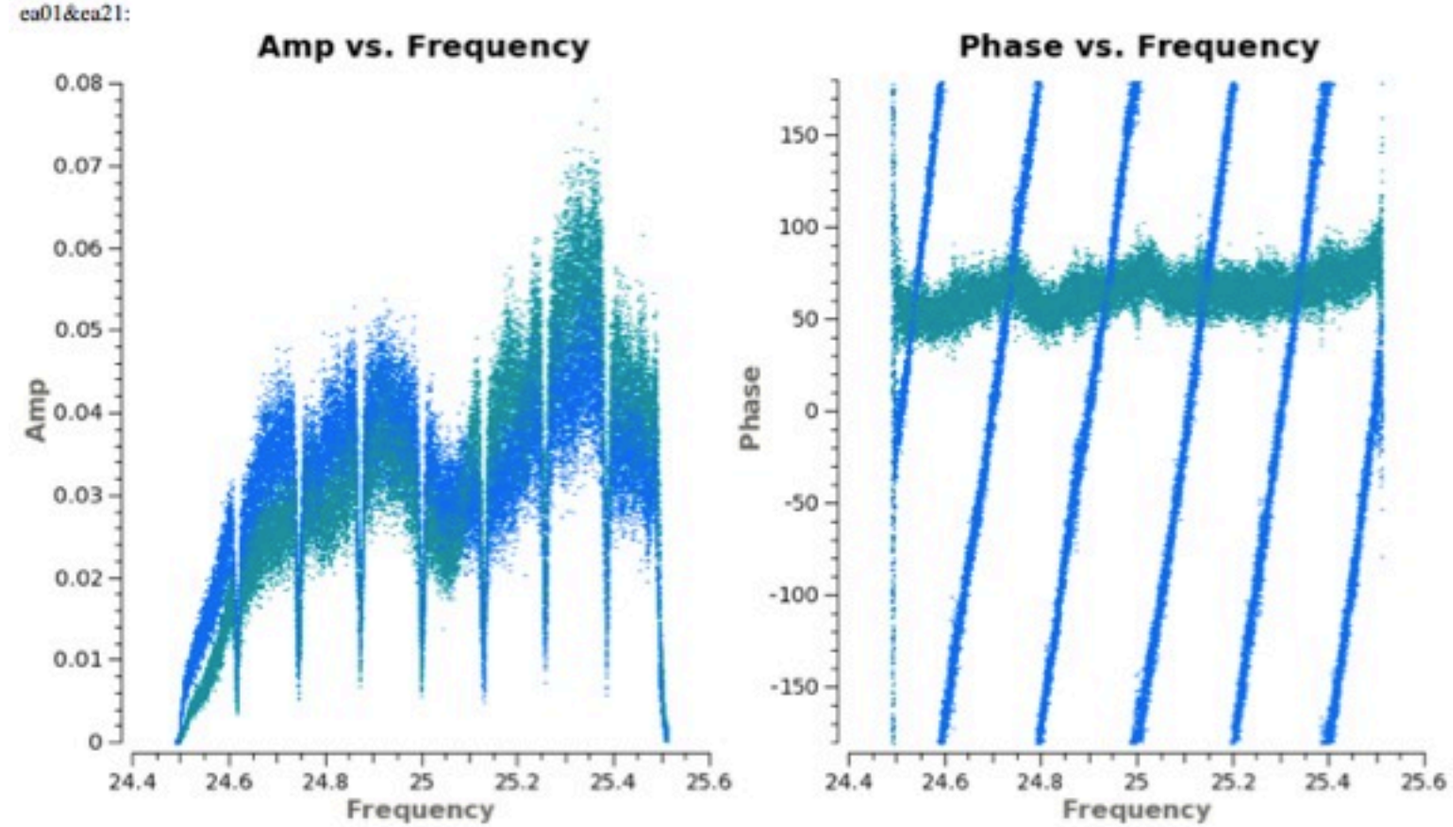
- Cleaned images

Log files:

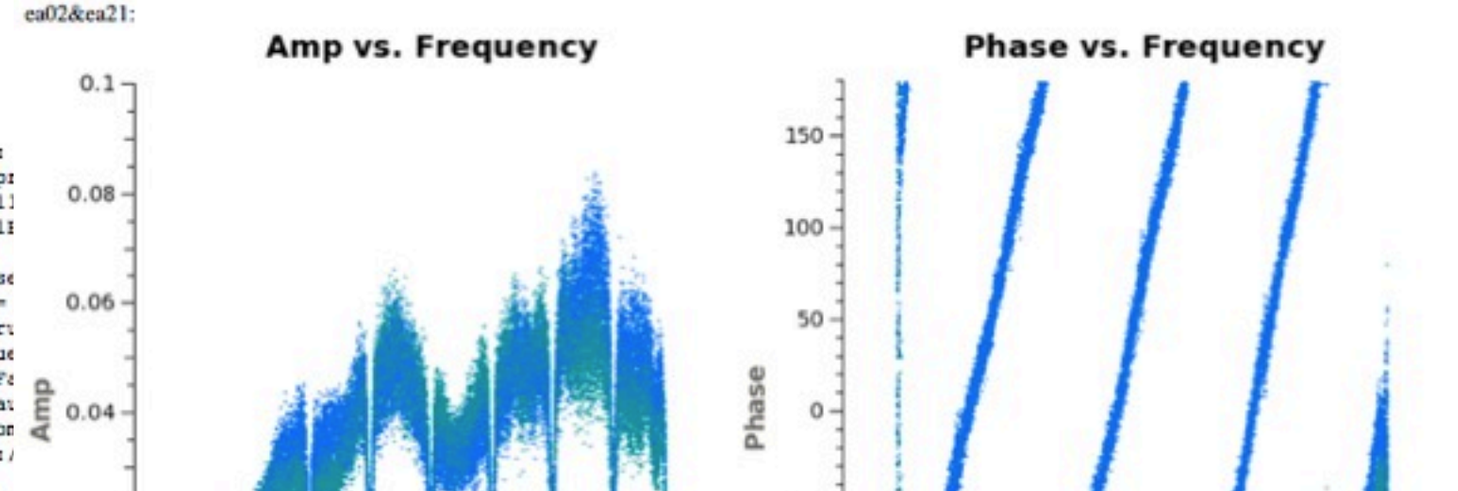
- Select information from log file (also see below)
- Complete log file

```
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Task inputs:
2011-12-01 18:18:11     INFO    mkpipeline::::casa      mode = initp
2011-12-01 18:18:11     INFO    mkpipeline::::casa      rootname = 1
2011-12-01 18:18:11     INFO    mkpipeline::::casa      sdmname = 11
2011-12-01 18:18:11     INFO    mkpipeline::::casa      band = K
2011-12-01 18:18:11     INFO    mkpipeline::::casa      dummy = Fals
2011-12-01 18:18:11     INFO    mkpipeline::::casa      checkzeros =
2011-12-01 18:18:11     INFO    mkpipeline::::casa      timeave = Tr
2011-12-01 18:18:11     INFO    mkpipeline::::casa      doplot = Tru
2011-12-01 18:18:11     INFO    mkpipeline::::casa      calimage = F
2011-12-01 18:18:11     INFO    mkpipeline::::casa      email = mkra
2011-12-01 18:18:11     INFO    mkpipeline::::casa      webdir = /ho
2011-12-01 18:18:11     INFO    mkpipeline::::casa      http = http:/
2011-12-01 18:18:11     INFO    mkpipeline::::casa
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Bandpass calibrator fields: [3]
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Gain calibrator fields: [0, 5]
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Flux calibrator fields: [3]
2011-12-01 18:18:11     INFO    mkpipeline::::casa      All calibrator fields: [0, 3, 5]
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Science target fields: [6]
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Fields to image: [6]
```

ea01&ea21:



ea02&ea21:



Example 2: mkpipeline

Plots and log files for 11A-277.K.30nov11.10s.ms:

Observation metadata:

- Antenna position diagram
- Weather data
- Online flags

Raw data:

- Raw amplitudes and phases vs. frequency, SPW 0~7
- Raw amplitudes and phases vs. frequency, SPW 8~15

Calibration:

- Initial phase solution plots
- Bandpass plots: SPW 0~7
- Antenna-based gain calibration plots: SPW 0~7
- Flux-scaled gain calibration plots: SPW 0~7
- Bandpass plots: SPW 8~15
- Antenna-based gain calibration plots: SPW 8~15
- Flux-scaled gain calibration plots: SPW 8~15
- Calibrated data: field 0
- Calibrated data: field 3
- Calibrated data: field 5

Imaging:

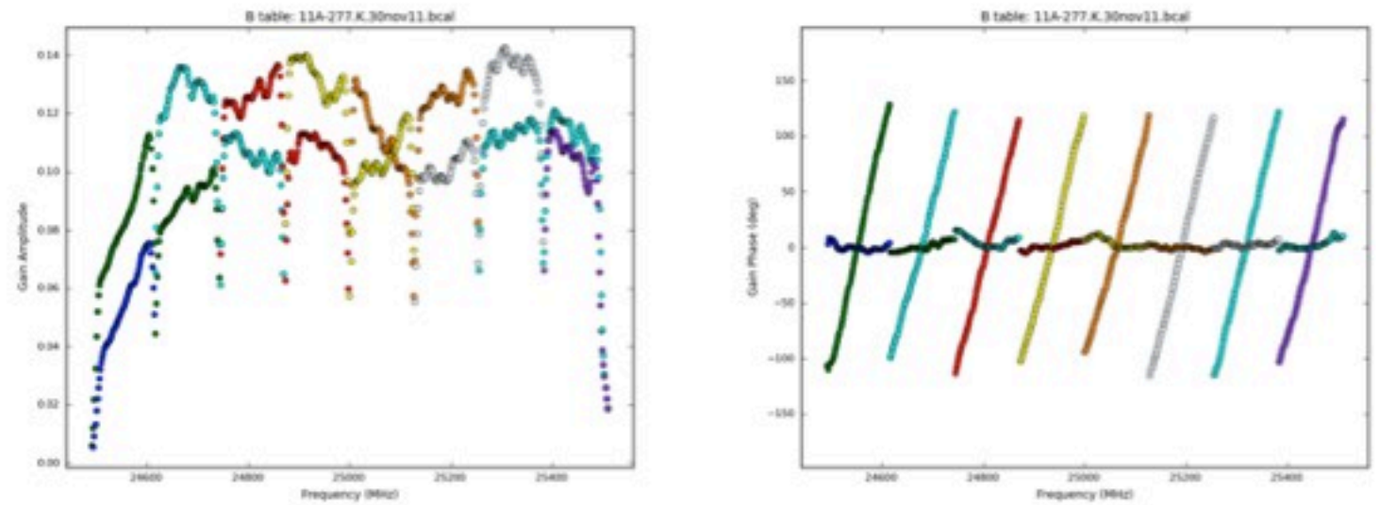
- Cleaned images

Log files:

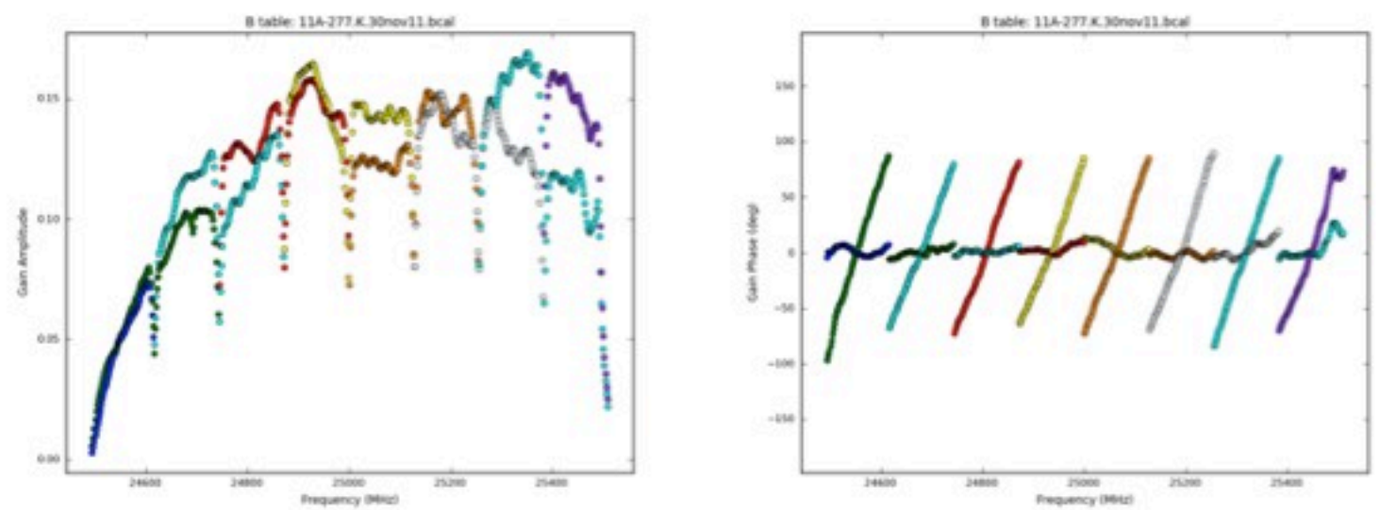
- Select information from log file (also see below)
- Complete log file

```
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Task inputs:
2011-12-01 18:18:11     INFO    mkpipeline::::casa      mode = initproc
2011-12-01 18:18:11     INFO    mkpipeline::::casa      rootname = 11A-2
2011-12-01 18:18:11     INFO    mkpipeline::::casa      sdmname = 11B-23
2011-12-01 18:18:11     INFO    mkpipeline::::casa      band = K
2011-12-01 18:18:11     INFO    mkpipeline::::casa      dummy = False
2011-12-01 18:18:11     INFO    mkpipeline::::casa      checkzeros = Fal
2011-12-01 18:18:11     INFO    mkpipeline::::casa      timeave = True
2011-12-01 18:18:11     INFO    mkpipeline::::casa      doplot = True
2011-12-01 18:18:11     INFO    mkpipeline::::casa      calimage = False
2011-12-01 18:18:11     INFO    mkpipeline::::casa      email = mkrauss@
2011-12-01 18:18:11     INFO    mkpipeline::::casa      webdir = /home/w
2011-12-01 18:18:11     INFO    mkpipeline::::casa      http = http://ww
2011-12-01 18:18:11     INFO    mkpipeline::::casa
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Bandpass calibra
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Gain calibrator
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Flux calibrator fields: [3]
2011-12-01 18:18:11     INFO    mkpipeline::::casa      All calibrator fields: [0, 3, 5]
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Science target fields: [6]
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Fields to image: [6]
```

ea01:



ea02:



ea03:

Example 2: mkpipeline

Plots and log files for 11A-277.K.30nov11.10s.ms:

Observation metadata:

- Antenna position diagram
- Weather data
- Online flags

Raw data:

- Raw amplitudes and phases vs. frequency, SPW 0~7
- Raw amplitudes and phases vs. frequency, SPW 8~15

Calibration:

- Initial phase solution plots
- Bandpass plots: SPW 0~7
- Antenna-based gain calibration plots: SPW 0~7
- Flux-scaled gain calibration plots: SPW 0~7
- Bandpass plots: SPW 8~15
- Antenna-based gain calibration plots: SPW 8~15
- Flux-scaled gain calibration plots: SPW 8~15
- Calibrated data: field 0
- Calibrated data: field 3
- Calibrated data: field 5

Imaging:

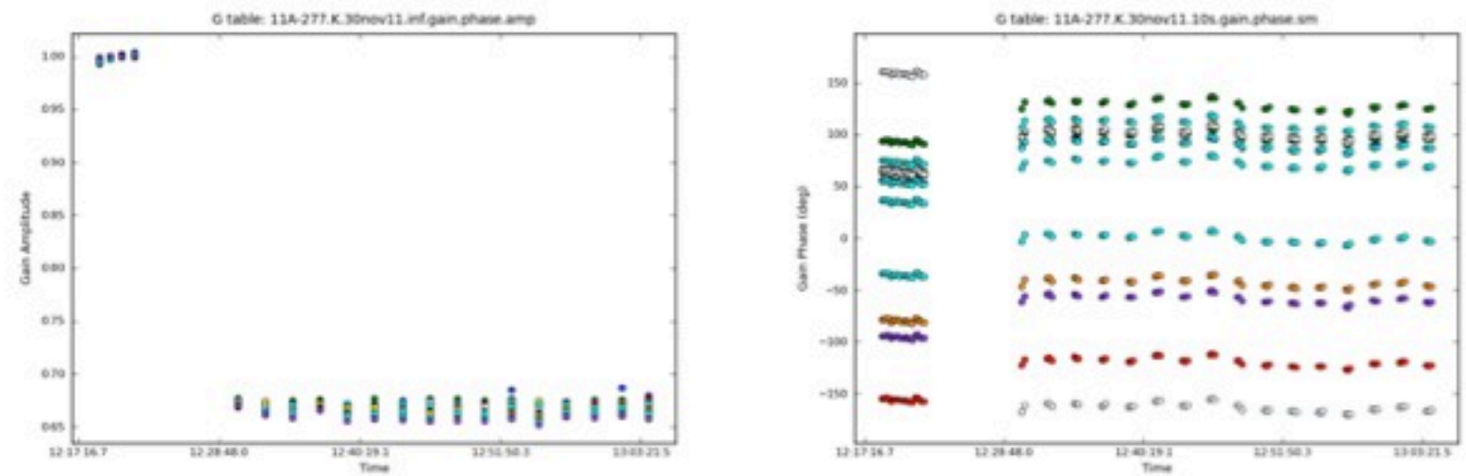
- Cleaned images

Log files:

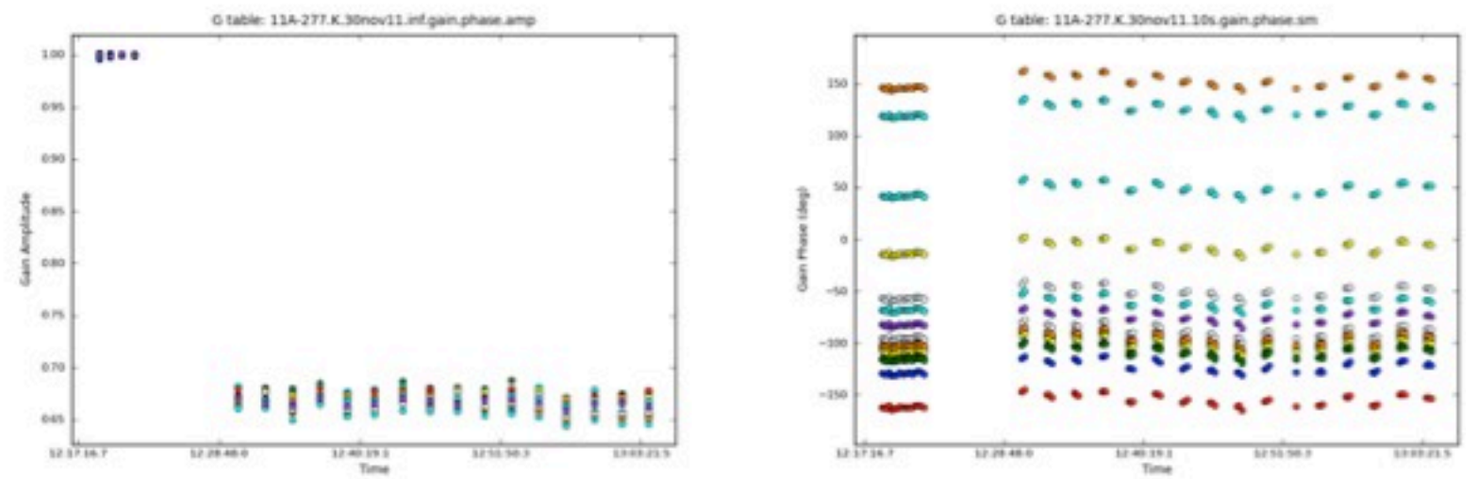
- Select information from log file (also see below)
- Complete log file

```
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Task inp
2011-12-01 18:18:11     INFO    mkpipeline::::casa      mode = i
2011-12-01 18:18:11     INFO    mkpipeline::::casa      rootname
2011-12-01 18:18:11     INFO    mkpipeline::::casa      sdmname
2011-12-01 18:18:11     INFO    mkpipeline::::casa      band = K
2011-12-01 18:18:11     INFO    mkpipeline::::casa      dummy =
2011-12-01 18:18:11     INFO    mkpipeline::::casa      checkzer
2011-12-01 18:18:11     INFO    mkpipeline::::casa      timeave
2011-12-01 18:18:11     INFO    mkpipeline::::casa      doplot =
2011-12-01 18:18:11     INFO    mkpipeline::::casa      calimage
2011-12-01 18:18:11     INFO    mkpipeline::::casa      email =
2011-12-01 18:18:11     INFO    mkpipeline::::casa      webdir =
2011-12-01 18:18:11     INFO    mkpipeline::::casa      http = h
2011-12-01 18:18:11     INFO    mkpipeline::::casa
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Bandpass
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Gain cal
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Flux cal
2011-12-01 18:18:11     INFO    mkpipeline::::casa      All cali
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Science
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Fields to image: [6]
2011-12-01 18:18:11     INFO    mkpipeline::::casa
```

ea01:



ea02:



ea03:

Example 2: mkpipeline

Plots and log files for 11A-277.K.30nov11.10s.ms:

Observation metadata:

- Antenna position diagram
- Weather data
- Online flags

Raw data:

- Raw amplitudes and phases vs. frequency, SPW 0~7
- Raw amplitudes and phases vs. frequency, SPW 8~15

Calibration:

- Initial phase solution plots
- Bandpass plots: SPW 0~7
- Antenna-based gain calibration plots: SPW 0~7
- Flux-scaled gain calibration plots: SPW 0~7
- Bandpass plots: SPW 8~15
- Antenna-based gain calibration plots: SPW 8~15
- Flux-scaled gain calibration plots: SPW 8~15
- Calibrated data: field 0
- Calibrated data: field 3
- Calibrated data: field 5

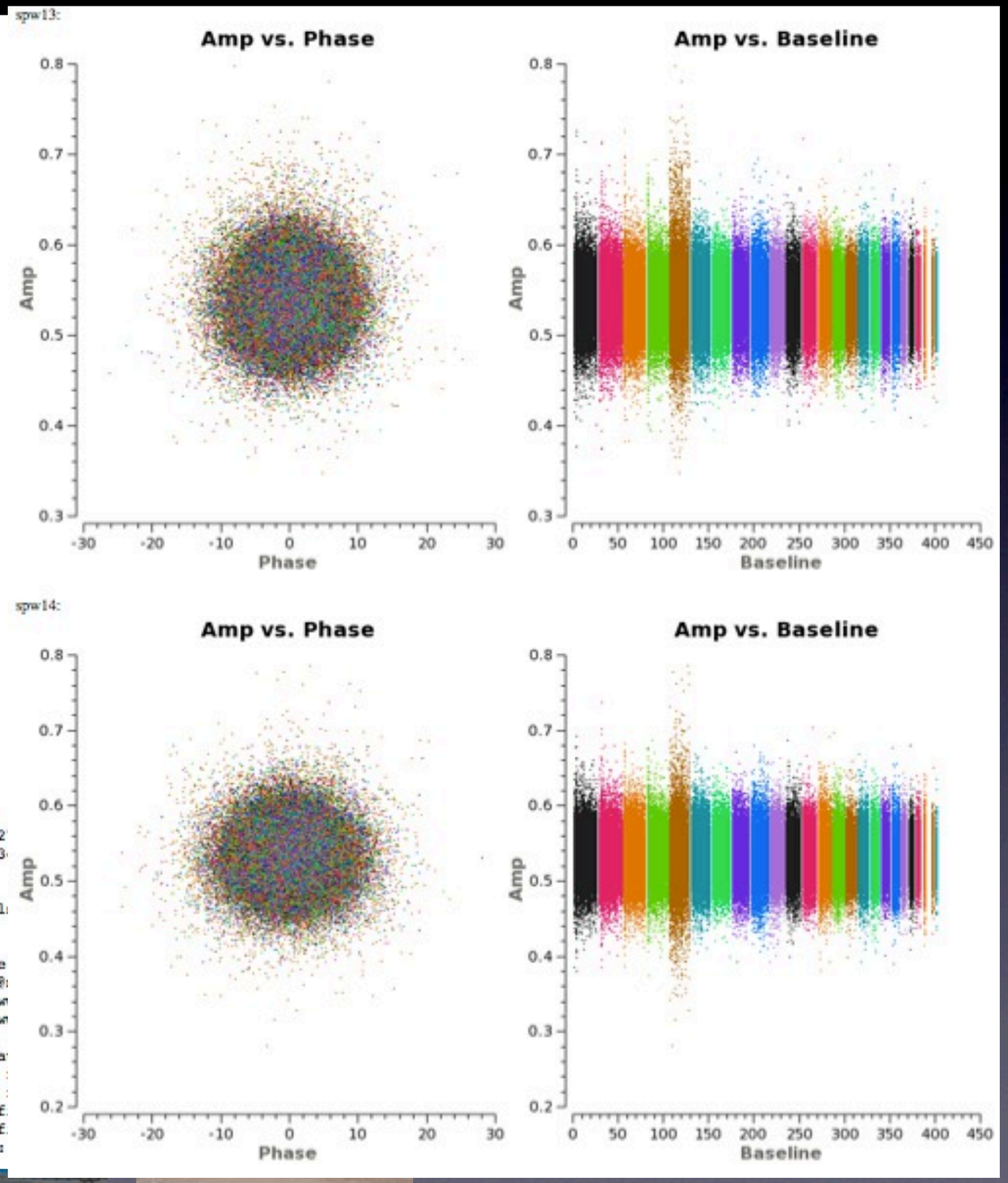
Imaging:

- Cleaned images

Log files:

- Select information from log file (also see below)
- Complete log file

```
2011-12-01 18:18:11    INFO    mkpipeline::::casa    Task inputs:
2011-12-01 18:18:11    INFO    mkpipeline::::casa    mode = initproc
2011-12-01 18:18:11    INFO    mkpipeline::::casa    rootname = 11A-2
2011-12-01 18:18:11    INFO    mkpipeline::::casa    sdmname = 11B-23
2011-12-01 18:18:11    INFO    mkpipeline::::casa    band = K
2011-12-01 18:18:11    INFO    mkpipeline::::casa    dummy = False
2011-12-01 18:18:11    INFO    mkpipeline::::casa    checkzeros = Fal
2011-12-01 18:18:11    INFO    mkpipeline::::casa    timeave = True
2011-12-01 18:18:11    INFO    mkpipeline::::casa    doplot = True
2011-12-01 18:18:11    INFO    mkpipeline::::casa    calimage = False
2011-12-01 18:18:11    INFO    mkpipeline::::casa    email = mkrauss@
2011-12-01 18:18:11    INFO    mkpipeline::::casa    webdir = /home/w
2011-12-01 18:18:11    INFO    mkpipeline::::casa    http = http://ww
2011-12-01 18:18:11    INFO    mkpipeline::::casa
2011-12-01 18:18:11    INFO    mkpipeline::::casa    Bandpass calibra
2011-12-01 18:18:11    INFO    mkpipeline::::casa    Gain calibrator
2011-12-01 18:18:11    INFO    mkpipeline::::casa    Flux calibrator
2011-12-01 18:18:11    INFO    mkpipeline::::casa    All calibrator f
2011-12-01 18:18:11    INFO    mkpipeline::::casa    Science target f
2011-12-01 18:18:11    INFO    mkpipeline::::casa    Fields to image:
```



Example 2: mkpipeline

Plots and log files for 11A-277.K.30nov11.10s.ms:

Observation metadata:

- Antenna position diagram
- Weather data
- Online flags

Raw data:

- Raw amplitudes and phases vs. frequency, SPW 0~7
- Raw amplitudes and phases vs. frequency, SPW 8~15

Calibration:

- Initial phase solution plots
- Bandpass plots: SPW 0~7
- Antenna-based gain calibration plots: SPW 0~7
- Flux-scaled gain calibration plots: SPW 0~7
- Bandpass plots: SPW 8~15
- Antenna-based gain calibration plots: SPW 8~15
- Flux-scaled gain calibration plots: SPW 8~15
- Calibrated data: field 0
- Calibrated data: field 3
- Calibrated data: field 5

Imaging:

- Cleaned images

Log files:

- Select information from log file (also see below)
- Complete log file

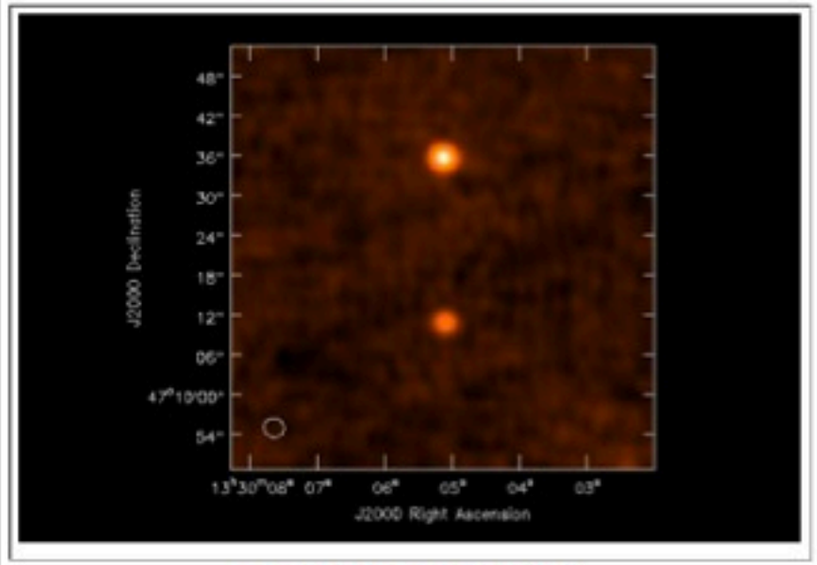
```
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Task inputs:
2011-12-01 18:18:11     INFO    mkpipeline::::casa      mode = initproc
2011-12-01 18:18:11     INFO    mkpipeline::::casa      rootname = 11A-277.K.30
2011-12-01 18:18:11     INFO    mkpipeline::::casa      sdmname = 11B-234.sb596
2011-12-01 18:18:11     INFO    mkpipeline::::casa      band = K
2011-12-01 18:18:11     INFO    mkpipeline::::casa      dummy = False
2011-12-01 18:18:11     INFO    mkpipeline::::casa      checkzeros = False
2011-12-01 18:18:11     INFO    mkpipeline::::casa      timeave = True
2011-12-01 18:18:11     INFO    mkpipeline::::casa      doplot = True
2011-12-01 18:18:11     INFO    mkpipeline::::casa      calimage = False
2011-12-01 18:18:11     INFO    mkpipeline::::casa      email = mkrauss@nrao.ed
2011-12-01 18:18:11     INFO    mkpipeline::::casa      webdir = /home/www.aoc.
2011-12-01 18:18:11     INFO    mkpipeline::::casa      http = http://www.aoc.n
2011-12-01 18:18:11     INFO    mkpipeline::::casa
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Bandpass calibrator fie
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Gain calibrator fields:
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Flux calibrator fields:
2011-12-01 18:18:11     INFO    mkpipeline::::casa      All calibrator fields:
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Science target fields: [6]
2011-12-01 18:18:11     INFO    mkpipeline::::casa      Fields to image: [6]
```

Image median RMS values and dynamic ranges:

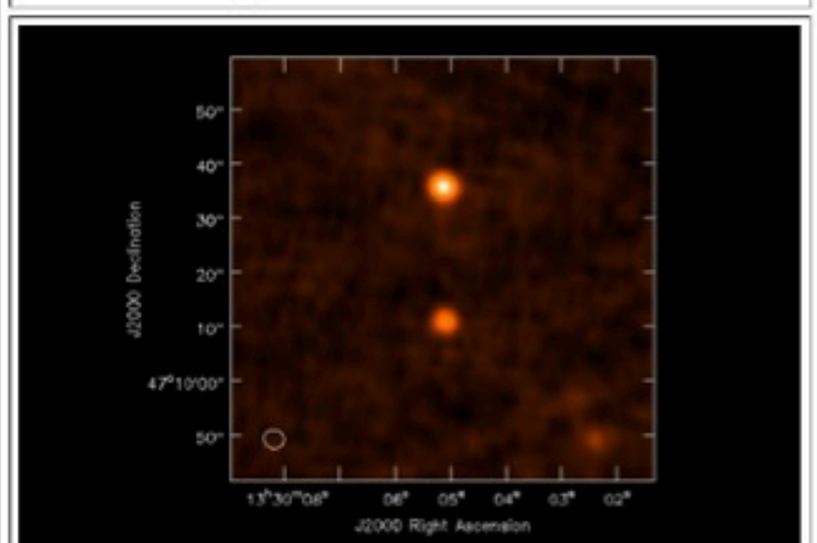
Source (field ID):	SPW (frequency):	Median image RMS:	Source flux:	Dynamic range:
SN2011DH (6)	spw 0~7, 25.00 GHz	20 uJy / beam	1.334 mJy / beam	65
SN2011DH (6)	spw 8~15, 20.50 GHz	20 uJy / beam	1.546 mJy / beam	74

Images:

spw0~7, SN2011DH (field ID 6):



spw8~15, SN2011DH (field ID 6):



Writing CASA Tasks using the CASA Toolkit

- Tasks provide a user-friendly, flexible interface for Python scripting of Toolkit and other functionality
- Tutorial to help step through process of writing a task: http://casaguides.nrao.edu/index.php?title=Writing_a_CASA_Task
- Anyone can contribute / find contributed tasks on the NRAO Science Forums (https://science.nrao.edu/forums/)
- User-contributed tasks may be posted on CASA Guides website