Imaging techniques:
## Image analysis, moments, spectral line analysis

Adam Avison

Outline

- Analysing images with the CASA viewer

- CASA command line tasks and tools for image analysis

The CASA viewer

- During the tutorials we have see the CASA viewer.

- Beyond simply allowing us to view images it can be used to preform image analysis.

- Can be started within CASA with the call **viewer()** or outside of CASA with **casaviewer** on the command line.
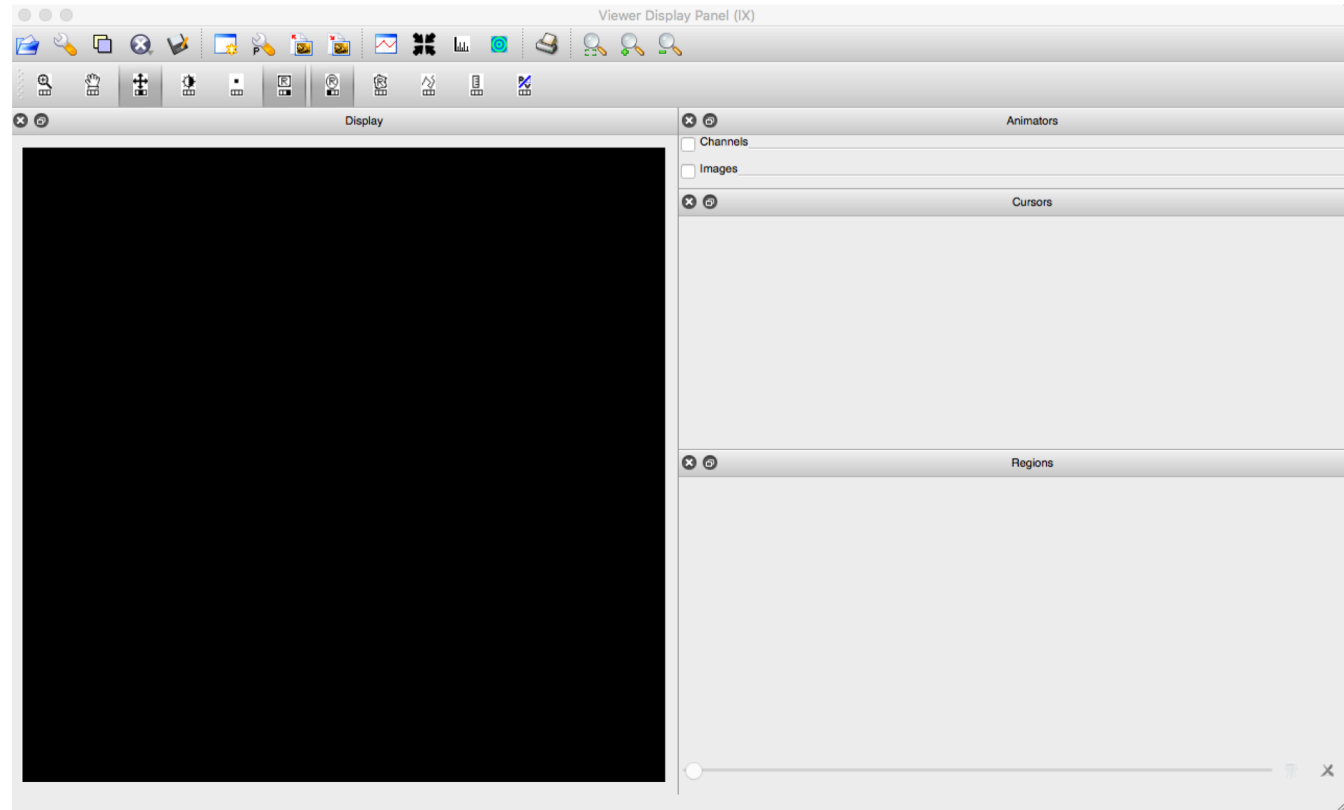
IMPORTANT INFORMATION:

In the future (potentially as early as CASA 5.5), the CASA Viewer will be replaced by a new "Viewer" known as CARTA.

The bad news: CARTA may not behave exactly the same as Viewer.

The good news: The baseline requirement for CARTA to replace Viewer is that it offer the same functionality.

Spectral line fit

- We can inspect a spectral line seen in this example data using the spectrum viewer.

- Highlight the area of emission with a circular region.

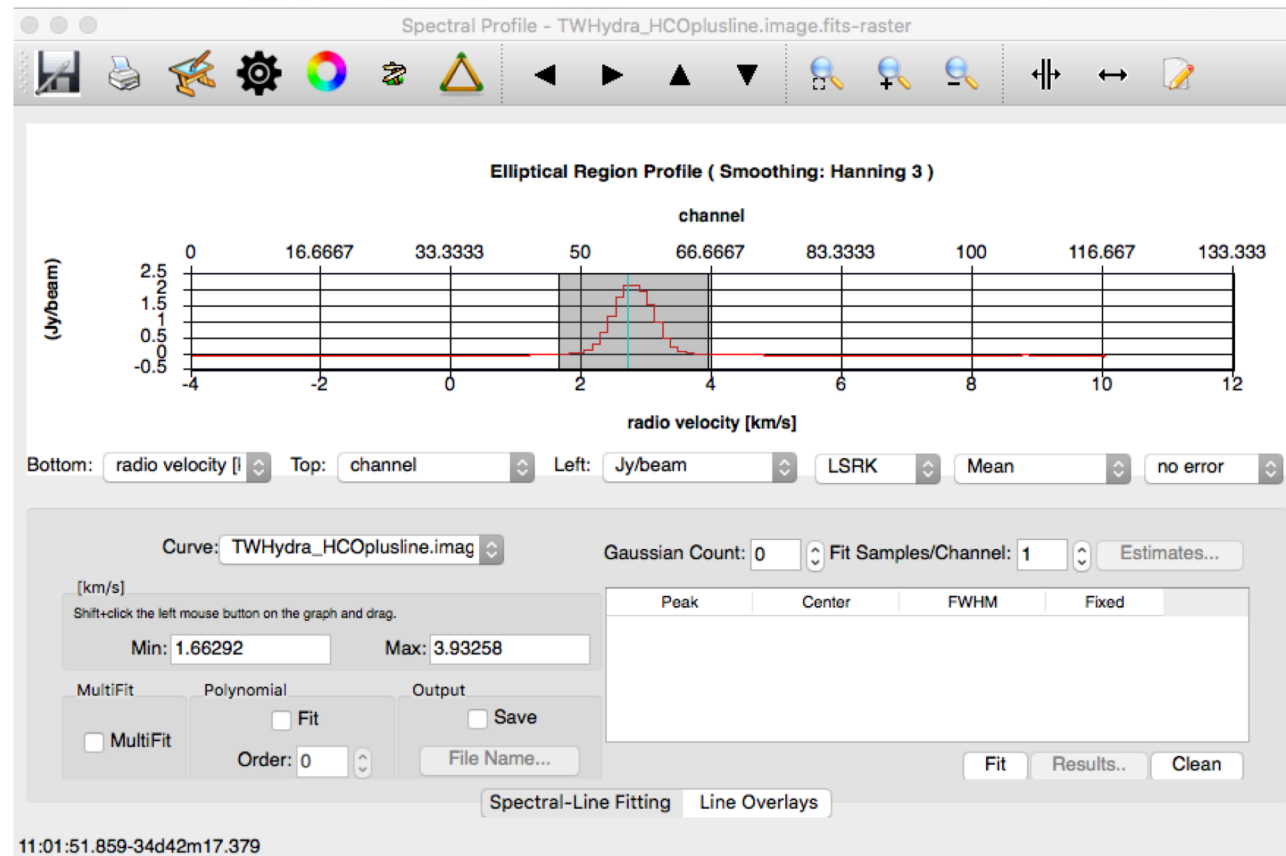- Then click the spectrum viewer button
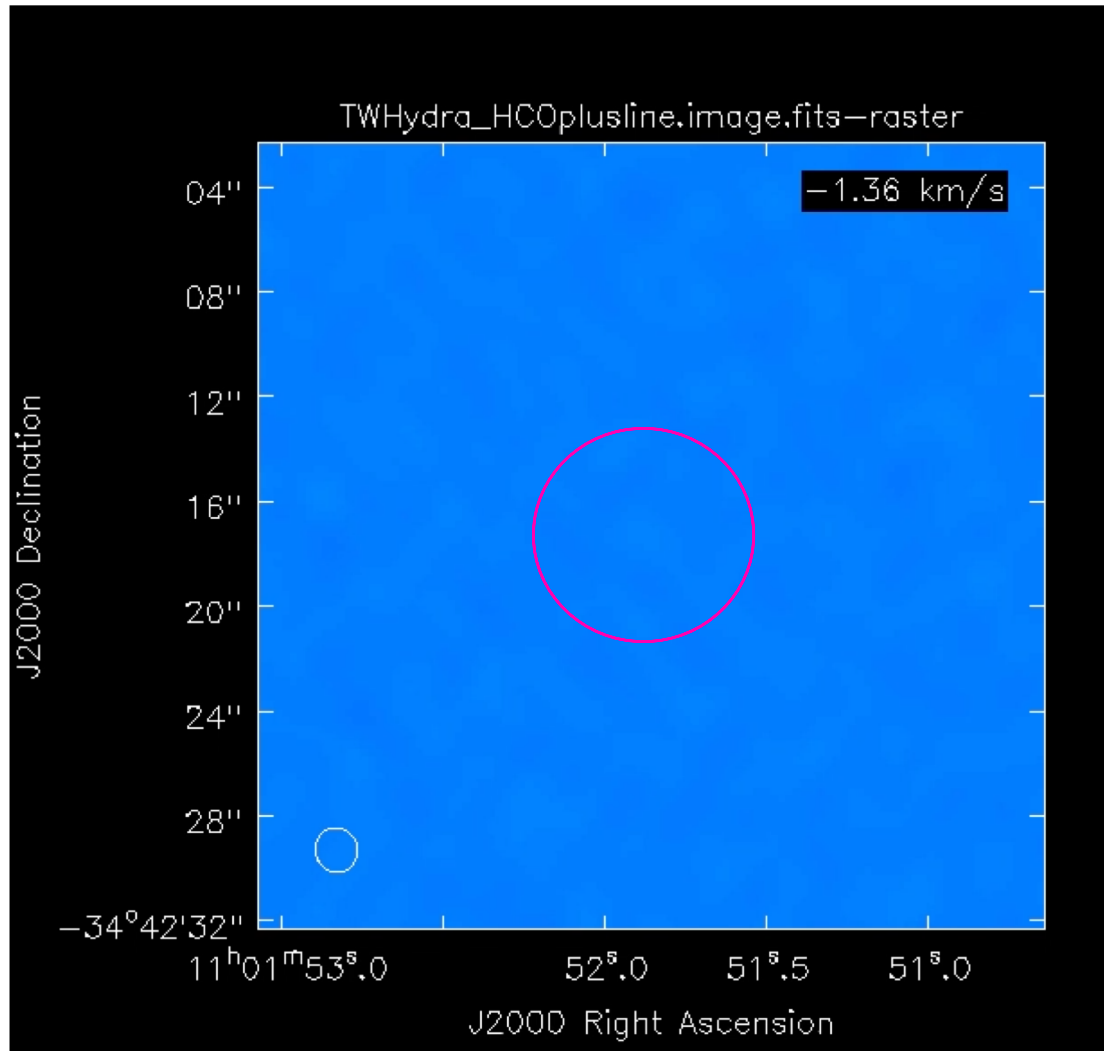
Here we are using a CASA example dataset from TW-Hydra
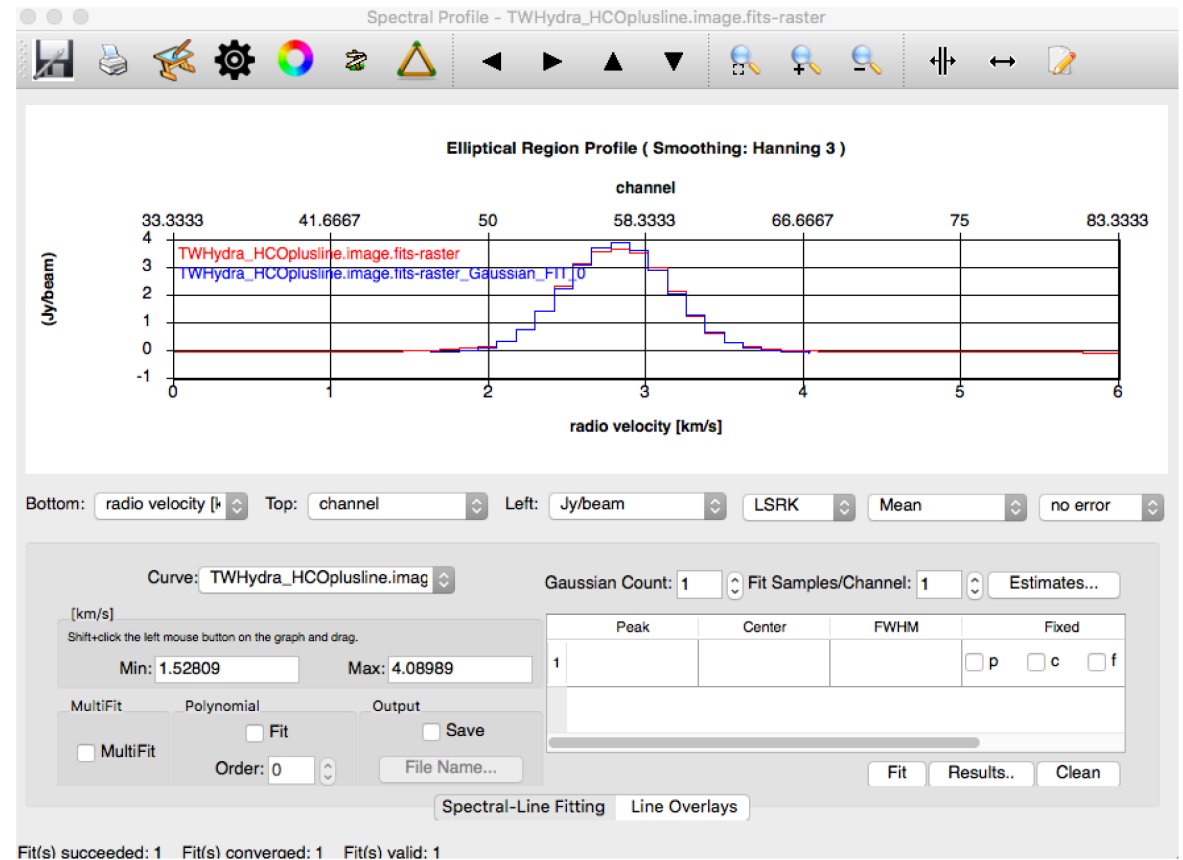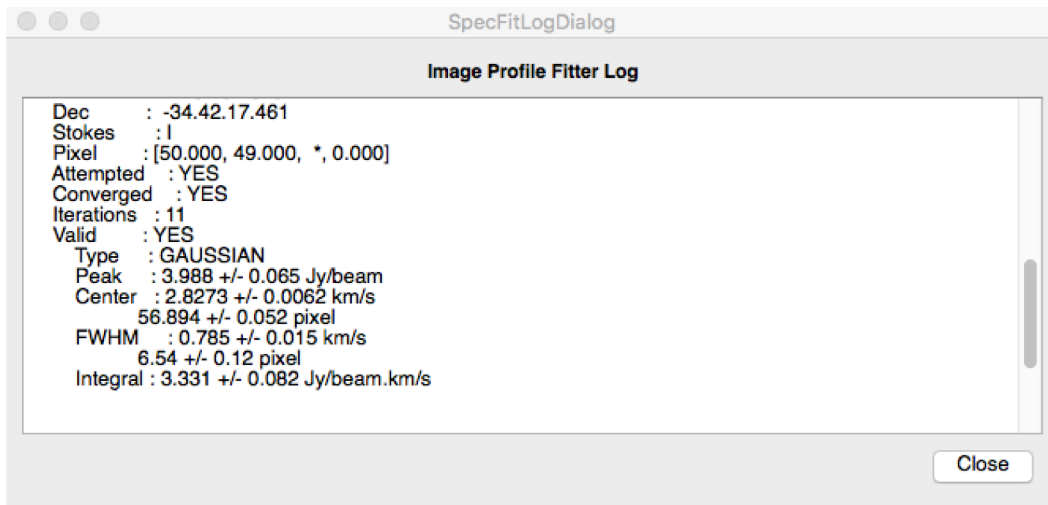Observed in HCO+ in ALMA Band 7
Data available from: https://casaguides.nrao.edu/index.php/TWHydraBand7

Spectral line fitting

- We can highlight the line with the ⫼ button.

- Set Gaussian Count to 1

- Then hit the Fit button

Spectral line fitting

- This will plot the fitted line over the data and pop-up a Spectral Fit dialogue box with the fit parameters in it.
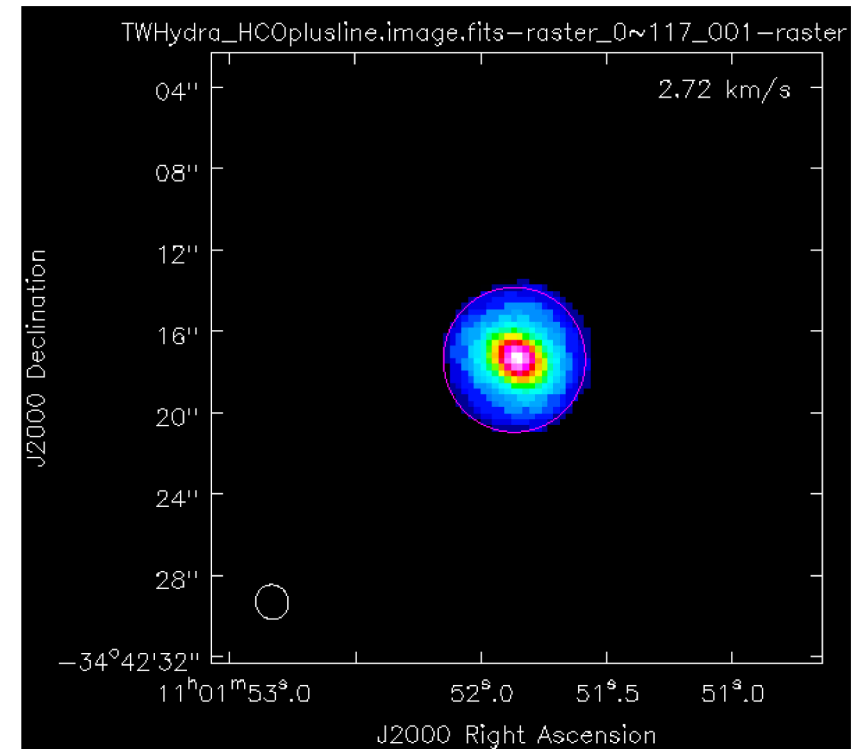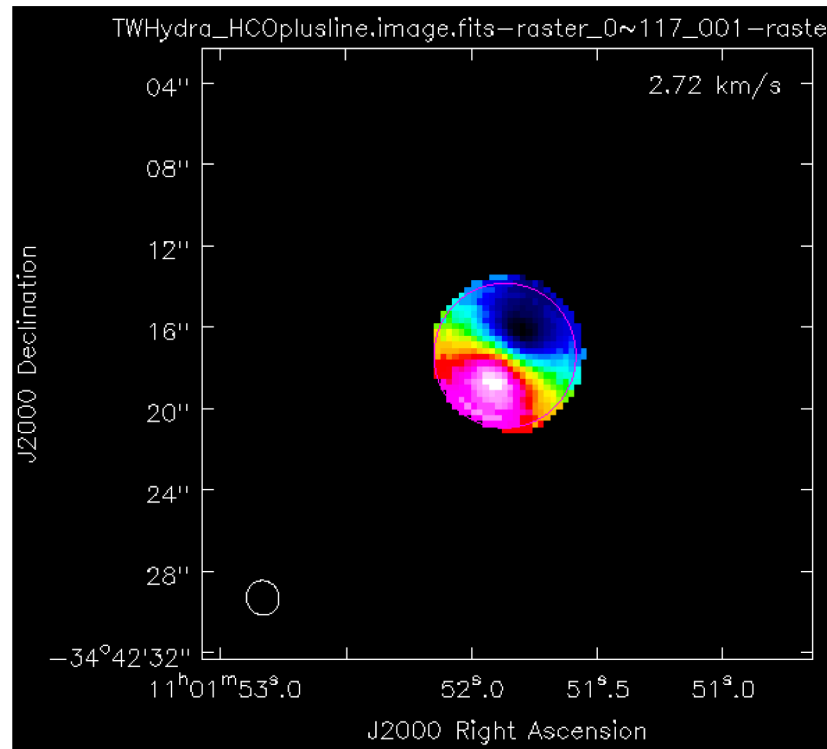
Collapse (image moments)

- GUI version of the CASA `immoments()` task used in the tutorial.

- Again highlight region of emission but we can access the collapse functionality with the button.

- Highlight desired moments

- Set include/excludepix values and hit Collapse.
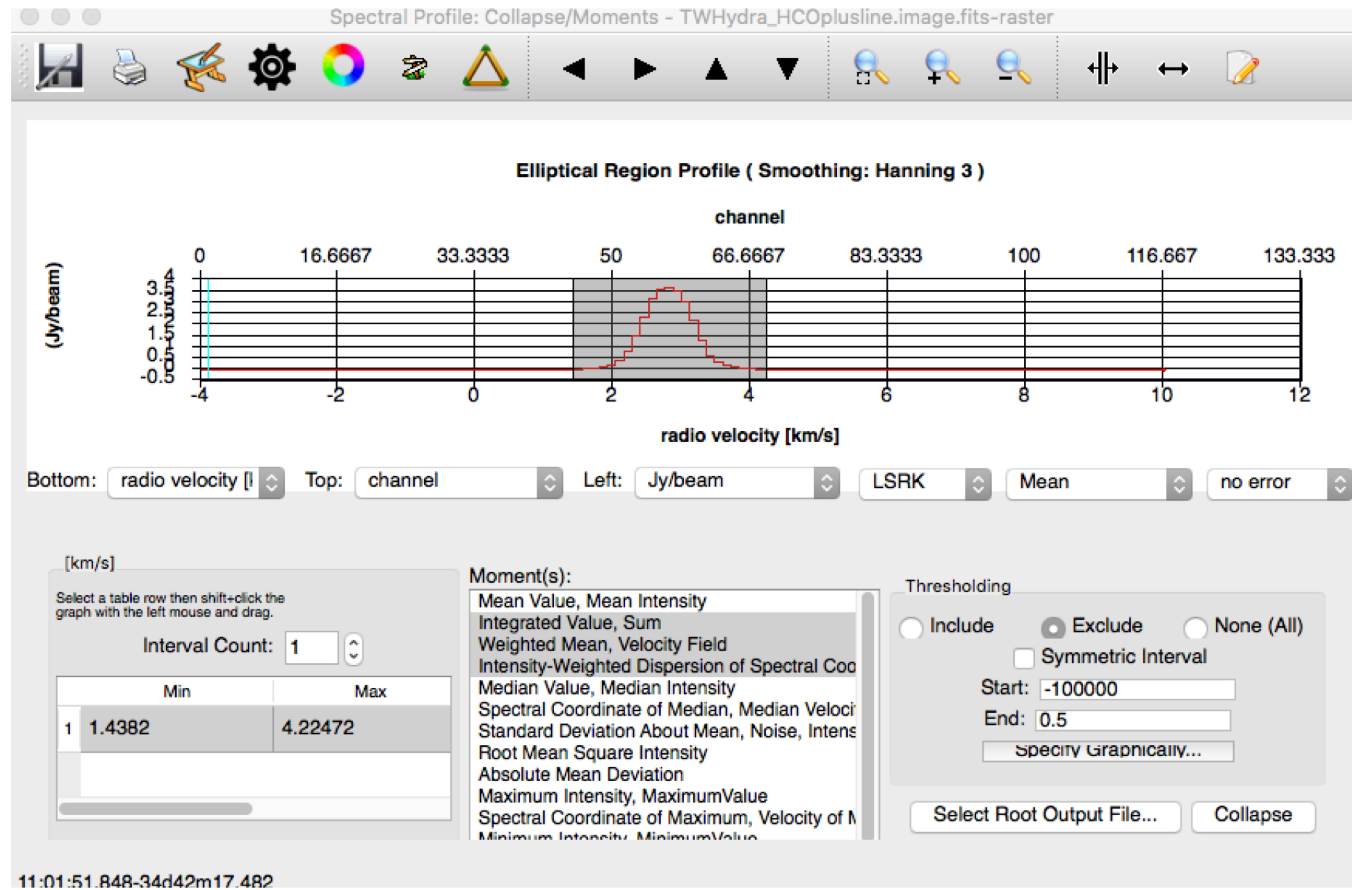
Integrated intensity, mom = 0

Velocity field, mom = 1

Velocity dispersion, mom = 2
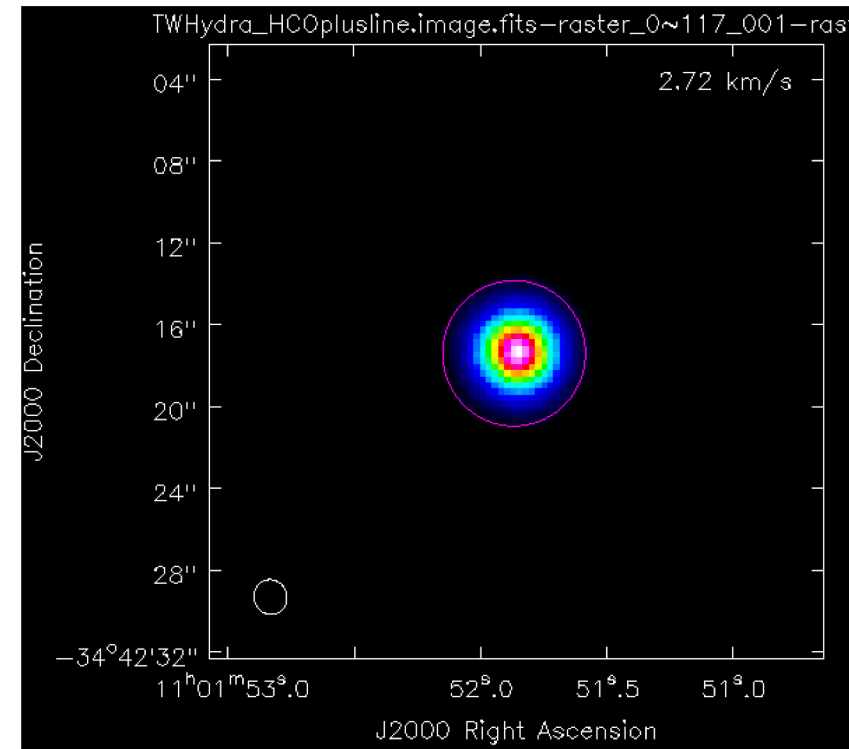
Position/Velocity analysis

- To investigate velocity structure we can use the PV analysis too.

- Select this using the ___ button and draw a line along the direction you wish to inspect.

- Then press 'Generate P/V'.

Position/Velocity analysis
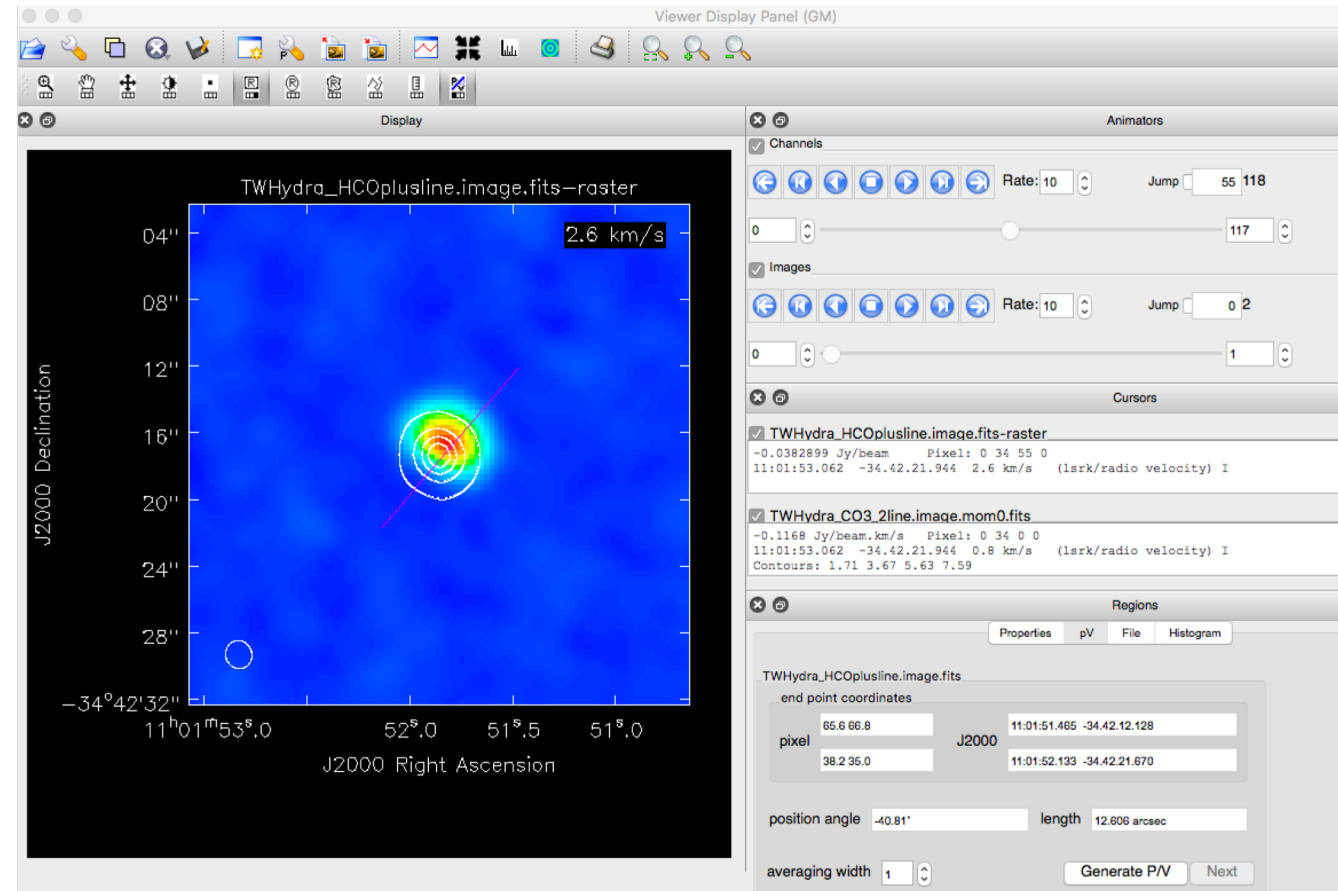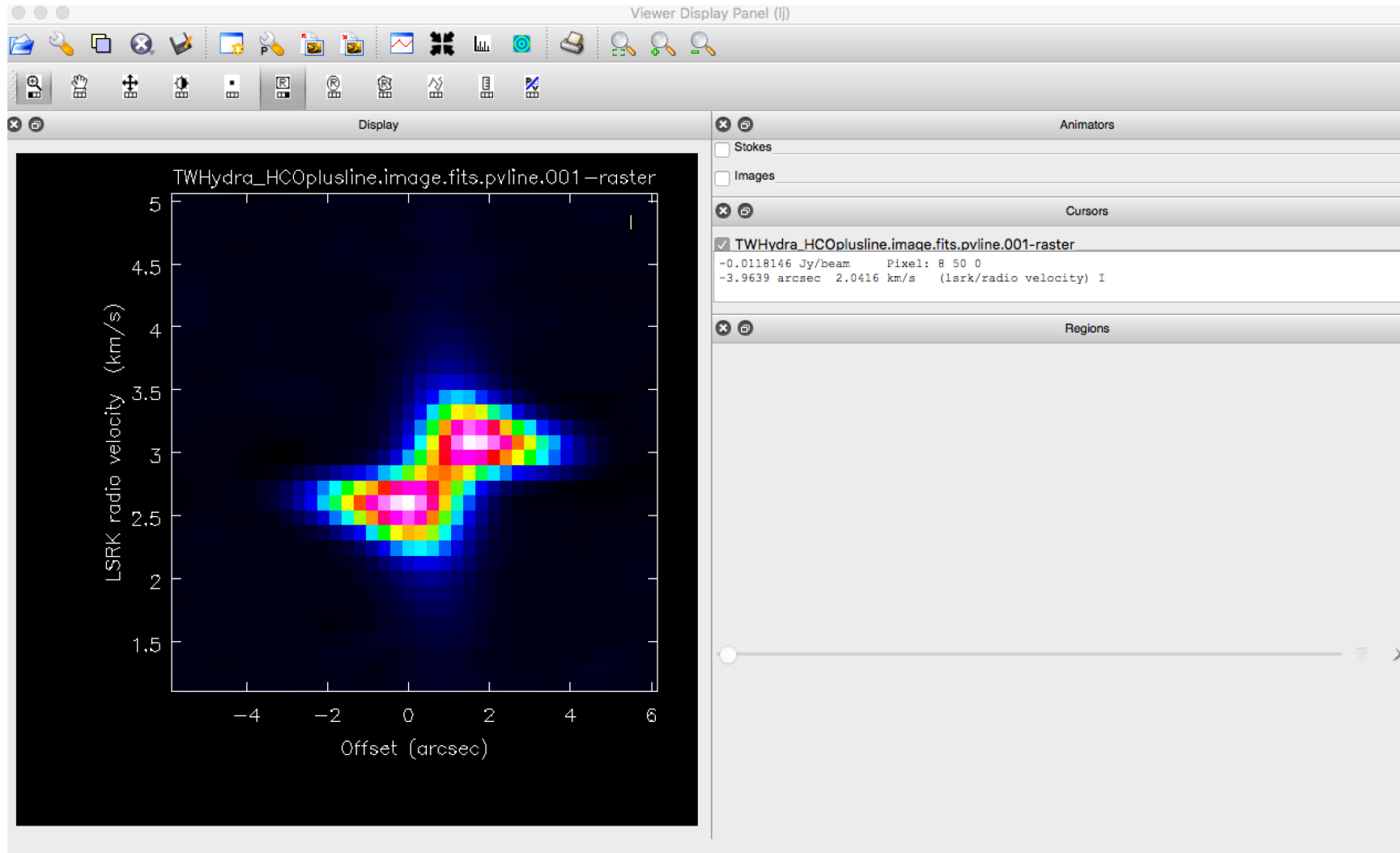
CASA Tasks and Tools for image analysis

TASKS:
Front end, user friend command line functions for data reduction, manipulation in CASA.

Built upon the TOOL kit functions available in CASA.

Typically have a bit more functionality than functions available in GUIs e.g. viewer

VS

TOOLS:
'Under the hood' basic functions upon which tasks are built.

Preform simple tasks but can be useful in image manipulation.

A non-exhaustive list of useful image analysis tasks available in CASA

immoments()
Compute moments from an image

imhead()
List, get and put image header parameters

imsubimage()
Create a (sub)image from a region of the image

specfit ()
Fit 1-dimensional gaussians and/or polynomial models to an image or image region

imfit()
Fit one or more elliptical Gaussian components on an image region

imval()
Get the data value(s) and/or mask value in an image

impv ()
Construct a position-velocity image by choosing two points in the direction plane

immath()
Perform mathematic operations on images

imstat()
Displays statistical information from an image or image region

Using the imaging toolkit functions

- The Imaging toolkit is more object oriented and Pythonic than using CASA tasks

- You will need to use multiple tools to achieve a single functions as you have to open and close the target image before doing anything to it.

- An example of a simple sequence of calls is given on the right.

```
CASA <2>: ia.open('myImage.image')
Out[2]: True

CASA <3>: ia.maxfit()   #-- Find and fit max pixel in image
Out[3]:
{'component0': {'flux': {'error': array([ 0.,  0.,  0.,  0.]),
  'polarisation': 'Stokes',
  'unit': 'Jy',
  ... }}

CASA <4>: ia.close()
Out[4]: True

CASA <5>: ia.done()
Out[5]: True
```

A non-exhaustive list of useful image analysis toolkit functions available in CASA

ia.open()
Open a new image file with this image tool

ia.close()
Close the image tool

ia.done()
Destroy this image tool

ia.getchunk ()
Get the pixel values from a regular region of the image into an array

ia.coordsys()
Get the Coordinate System of the image

ia.findsources()
Find point sources in the sky

ia.maxfit ()
Find maximum and do parabolic fit in the sky

ia.newimagefromarray()
Construct a casa image from an array

ia.convolve2D()
Convolve image by a 2D kernel