



Analysis Tools

Alex Hygate

(hygate@strw.leidenuniv.nl)



EUROPEAN ARC

ALMA Regional Centre || Allegro



Prepare for the exercises

In your analysis folder

- `mkdir analysis_tools`

Copy the data

- `cp -r ../../archive/DRT2023/TW_hydra/twhya_n2hp.image analysis_tools/`
- `cp -r ../../archive/DRT2023/TW_hydra/sis14_twhya_cont.image analysis_tools/`

Copy scripts from the 'scripts' folder to your own folder

- `> cp ../../scripts/analysis_tools_script.py analysis_tools/`

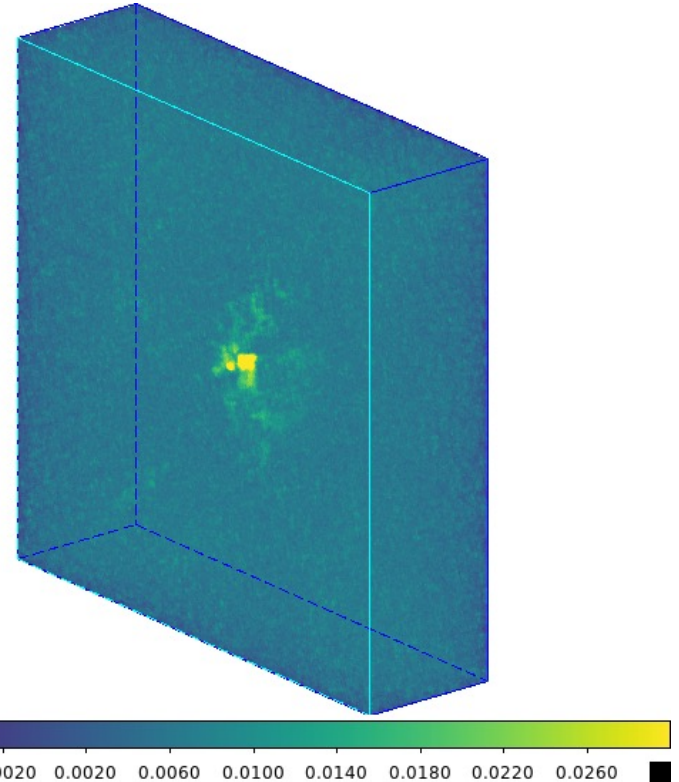
Start CASA

- `nice +10 env -u PYTHONPATH -u LD_LIBRARY_PATH casapy-660`

- **If you already did all this:**
 - `cd ../analysis_tools`
 - **Start CASA**

Datacubes

- 3D data product:
 - 2 spatial dimensions (RA, DEC)
 - 1 velocity dimension
- We will look at a few datacube analysis methods





Analysis tools

- We will look at analysis in CASA
- CASA - https://casa.nrao.edu/casa_obtaining.shtml
- Many others tools are available! We have many of these available on our computers
- Contact us at alma@strw.leidenuniv.nl with questions about or requests for assistance with analysing your alma data

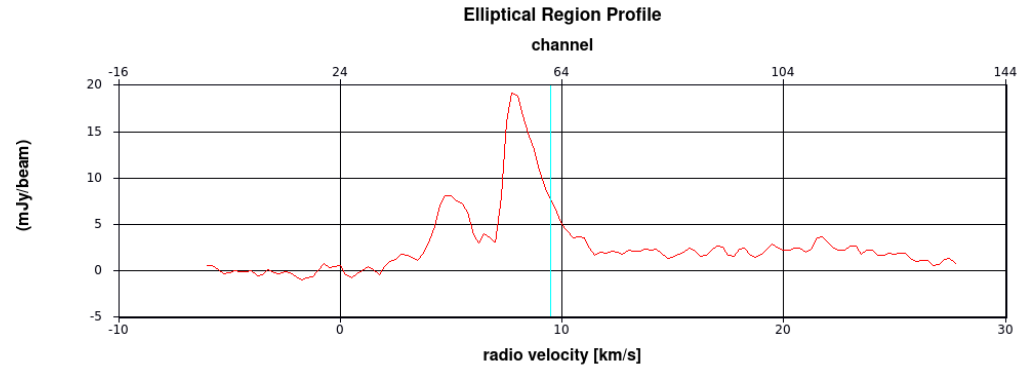


Analysis Techniques

- To visualize the information we usually make 1-D or 2-D projections providing different visualisation/analysis methods:
 - Spectral profiles (1-D slices along the velocity axis)
 - Channel maps (2-D slices along the velocity axis)
 - Moment maps (integration along the vel. axis)
 - Position-velocity plots (slices along spatial dimension)
 - Movies (series of 2-D slices along velocity axis)

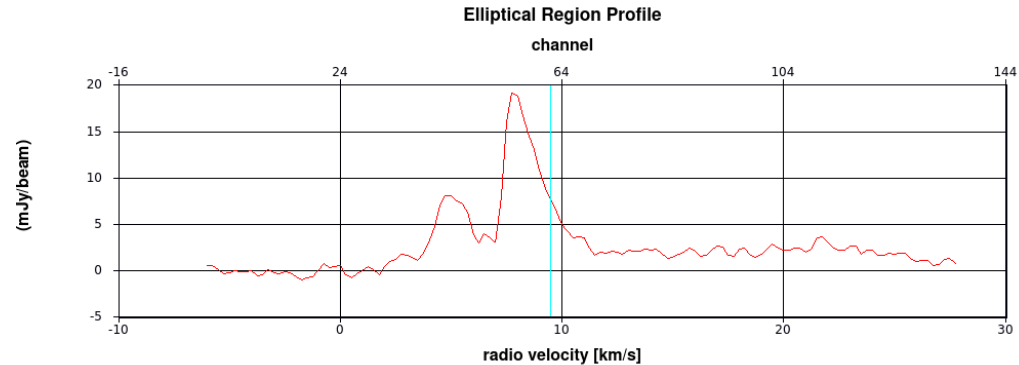
Spectral profiles

- Spectral profiles are 1-D slices along the velocity axis
- They can be created with the CASA viewer



Spectral profiles

- Spectral profiles are 1-D slices along the velocity axis
- They can be created with the CASA viewer





Exercise – start up CASA

- Use the “nice” command to prevent taking up all the CPU power for example for CASA
 - In (ba)sh shell: `> nice -n 10 env -u PYTHONPATH -u LD_LIBRARY_PATH casapy-660`
 - In (t)csch shell: `> nice +10 env -u PYTHONPATH -u LD_LIBRARY_PATH casapy-660`

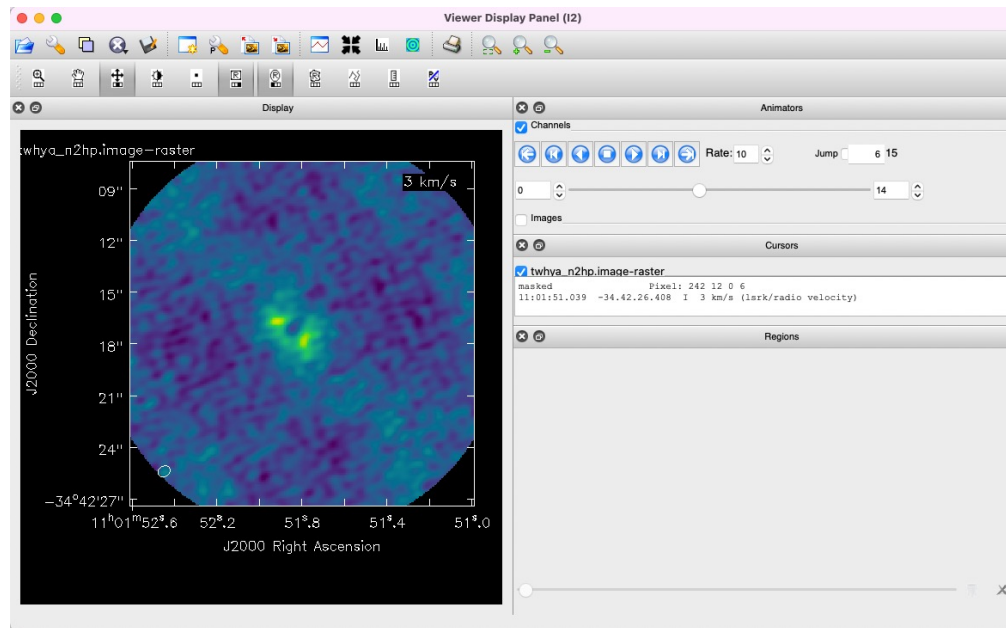
Spectral profiles

Open the casa viewer:

- **CASA <1>:**
`imview('twhya_n2hp.image')`

NB: Also works with .fits images

- **CASA <1>:** `imview('my_image.fits')`





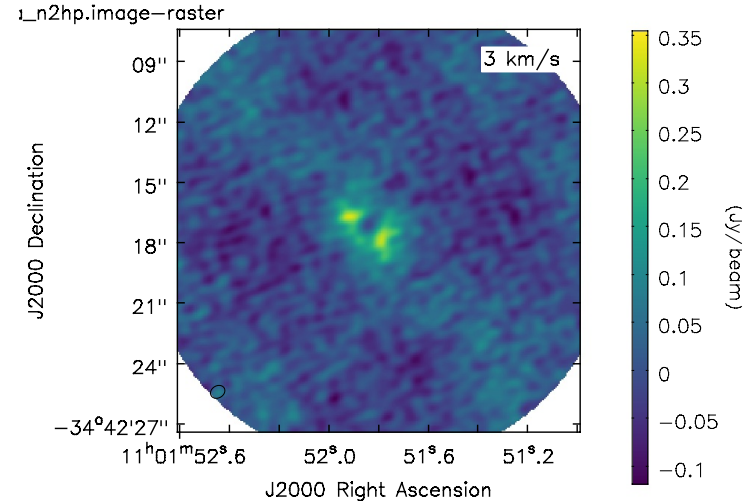
Exercise – Run the script

Open the casa viewer:


- **CASA <1>**: `mysteps = [1]`
- **CASA <2>**: `execfile('analysis_tools_script.py')`

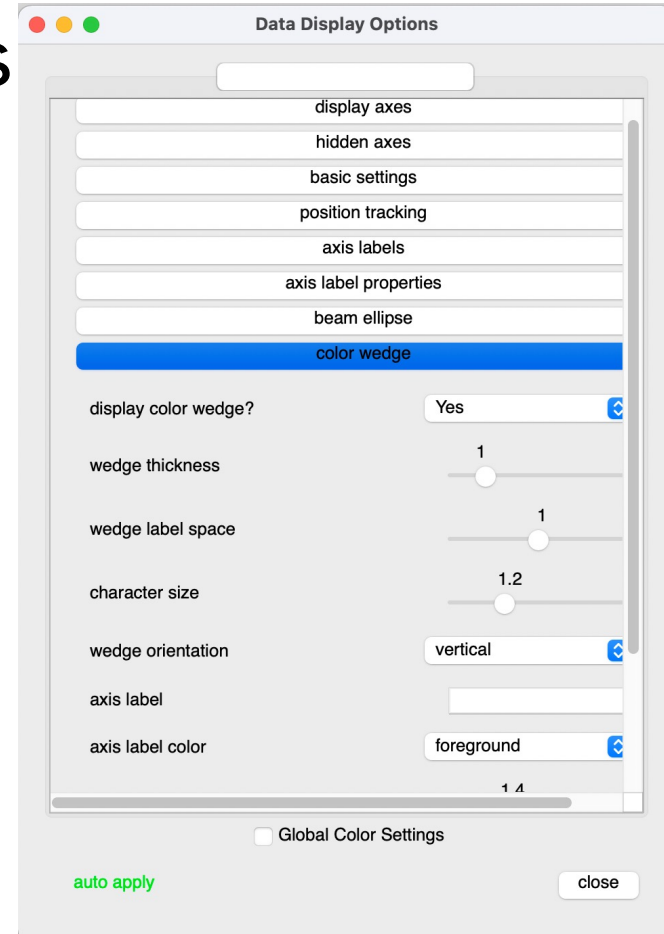
Channel maps

- Channels are 2D slices along the velocity axis
- Channel maps and other plots can be created in the CASA viewer
- First open the image in the viewer and select the channel you want to plot
 - `imview('twhya_n2hp.image')`

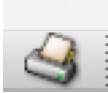


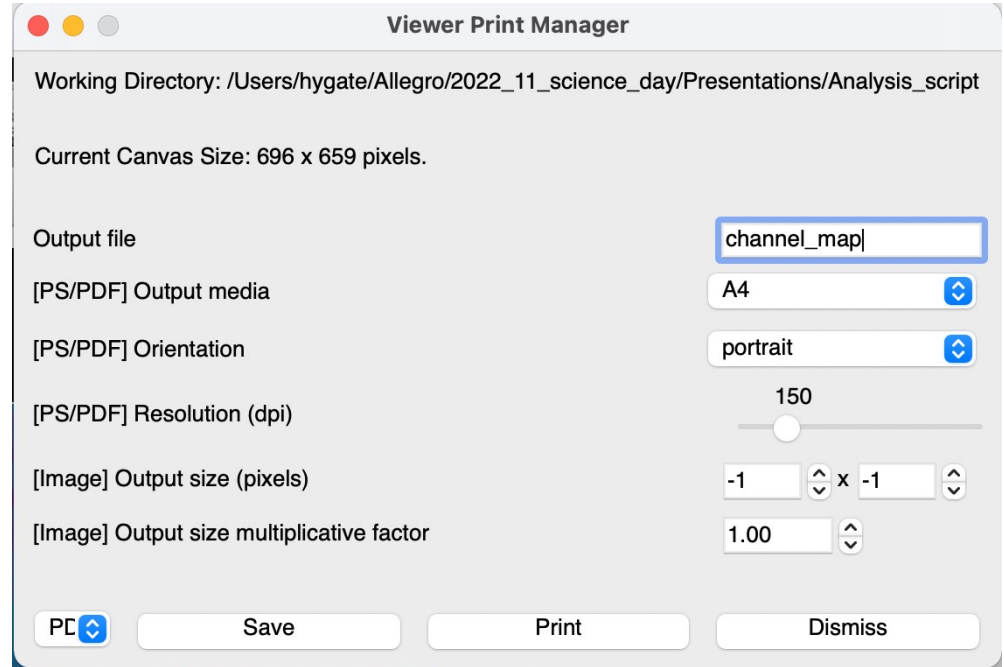
Channel maps

- Click the spanner button 
- Select options to customise your plot
 - e.g. set “display color wedge” to true



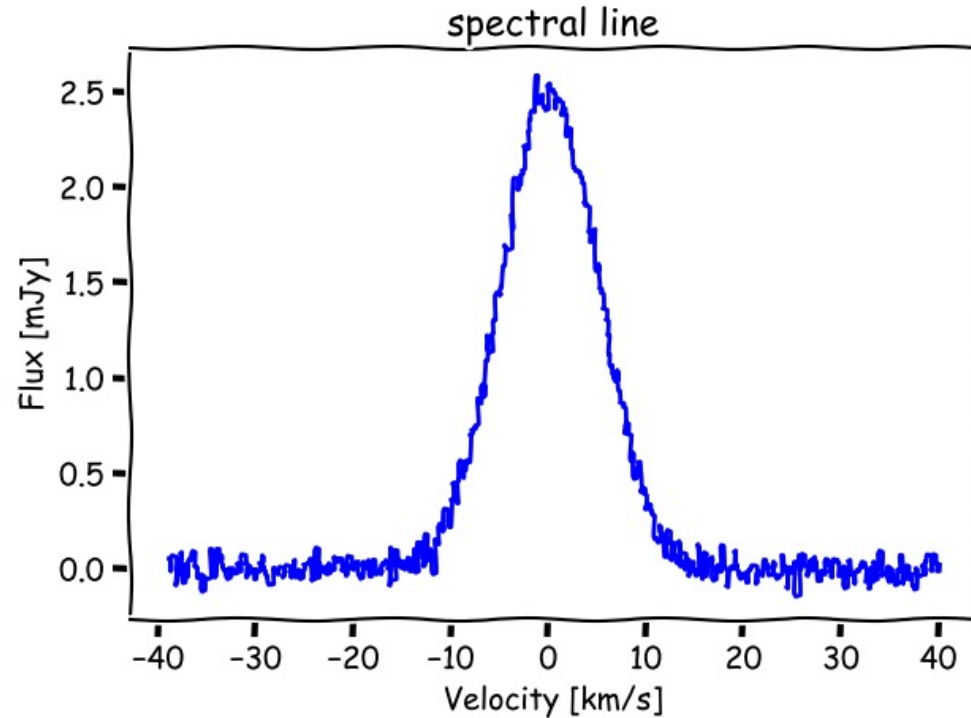
Channel maps

- Click the printer button 
- Select “pdf” (or your favourite format) from the drop down menu in the bottom left
- The same process works for any image in casa
- But you can make nicer plots with e.g. matplotlib!



Moment analysis

- Moments allow us to characterise a spectral line





Moment analysis

- **-1:** the mean value of the spectrum
- **0:** the integrated value of the spectrum
- **1:** the intensity weighted coordinate – used for velocity fields
- **2:** the intensity weighted dispersion of the coordinate – used for velocity dispersion fields
- **3:** the median of the spectrum
- **4:** the median velocity
- **5:** the standard deviation about the mean of the spectrum
- **6:** the root mean square of the spectrum
- **7:** the absolute mean deviation of the spectrum
- **8:** the maximum value of the spectrum
- **9:** the coordinate of the maximum value of the spectrum
- **10:** the minimum value of the spectrum
- **11:** the coordinate of the minimum value of the spectrum

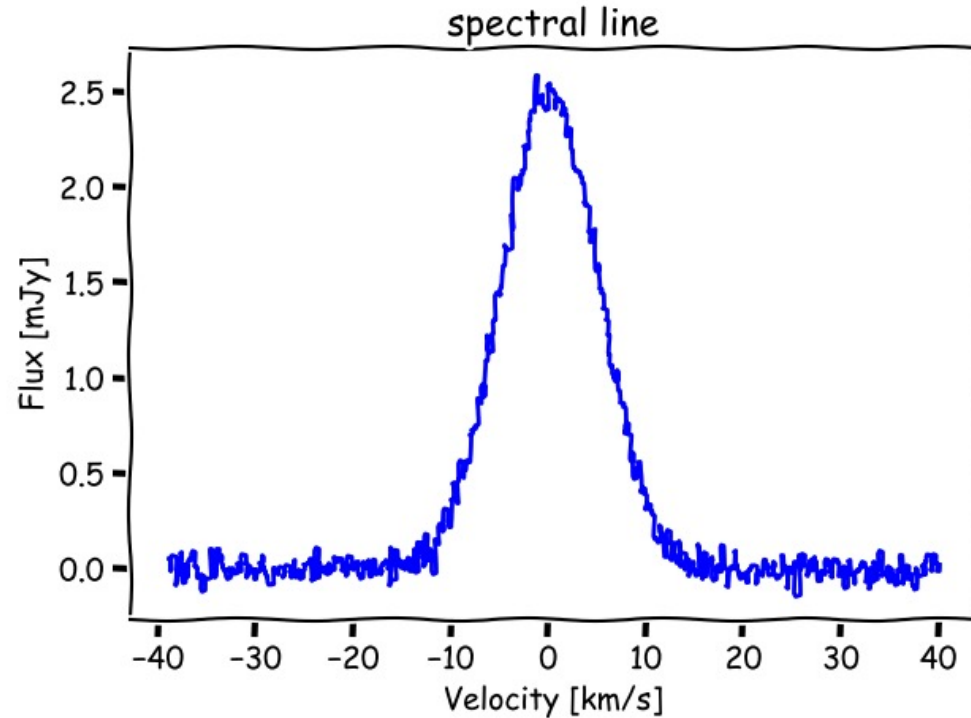


Moment analysis

- **-1:** the mean value of the spectrum
- **0:** the **integrated value** of the spectrum
- **1:** the intensity weighted coordinate – used for “**velocity**” fields
- **2:** the intensity weighted dispersion of the coordinate – used for “**velocity dispersion**” fields
- **3:** the median of the spectrum
- **4:** the median velocity
- **5:** the standard deviation about the mean of the spectrum
- **6:** the root mean square of the spectrum
- **7:** the absolute mean deviation of the spectrum
- **8:** the maximum value of the spectrum
- **9:** the coordinate of the maximum value of the spectrum
- **10:** the minimum value of the spectrum
- **11:** the coordinate of the minimum value of the spectrum

Moment analysis

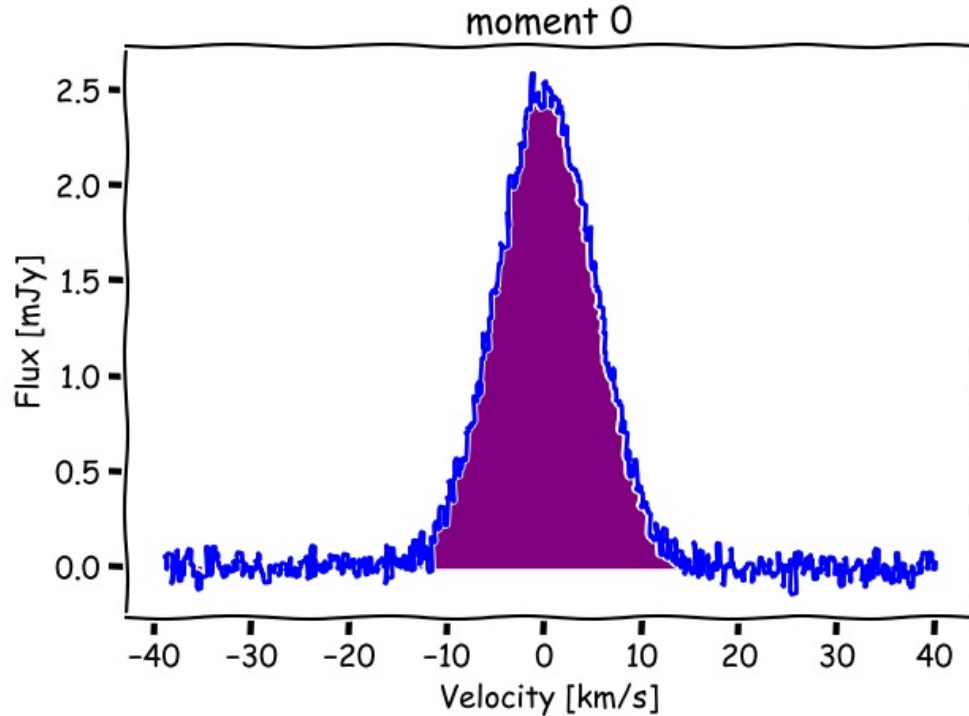
- immoments task in CASA:
https://casa.nrao.edu/casadocs/latest/global-task-list/task_immoments/about
- Generally no threshold for moment 0
- Exclude noise with threshold from moment 1 or higher maps



Moment 0 – integrated flux

- `CASA <1>`: `immoments(axis='spec', imagename='my_image', moments=[0], outfile='my_mom0')`
- Normally no threshold

$$M_0 = \Delta v \sum I(v)$$



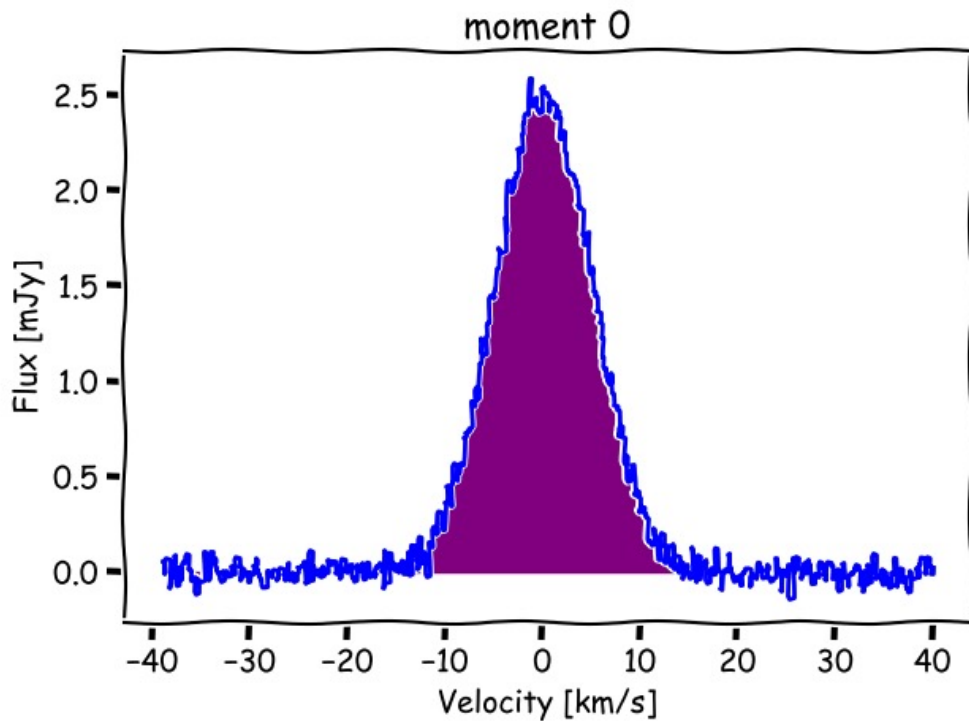
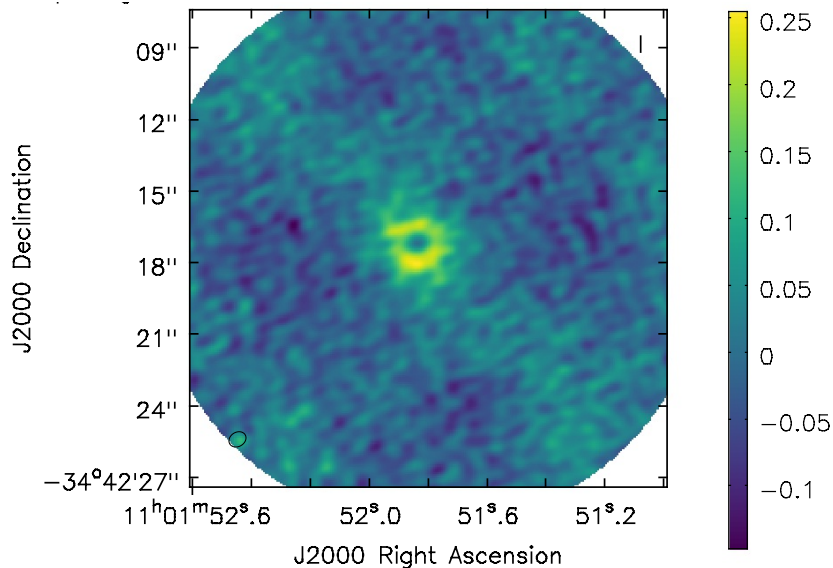


Exercise – Run the script

- CASA <1>: mysteps = [2]
- CASA <2>: `execfile('analysis_tools_script.py')`

- CASA <1>: mysteps = [3]
- CASA <2>: `execfile('analysis_tools_script.py')`

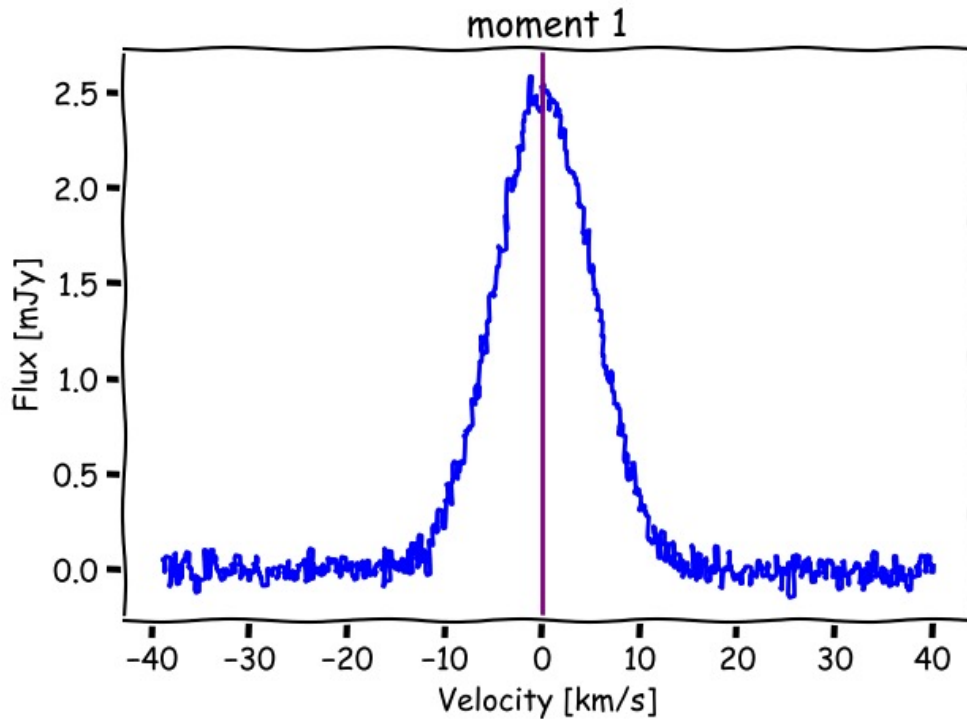
Moment 0 - integrated flux



Moment 1 - Velocity

- `CASA <1>`: `immoments(axis='spec', imagename='my_image', includepix= [3 * sigma, maxpix] moments=[1], outfile='my_mom1')`
- use “includepix” for thresholding. E.g. from 3σ to the maximum pixel value

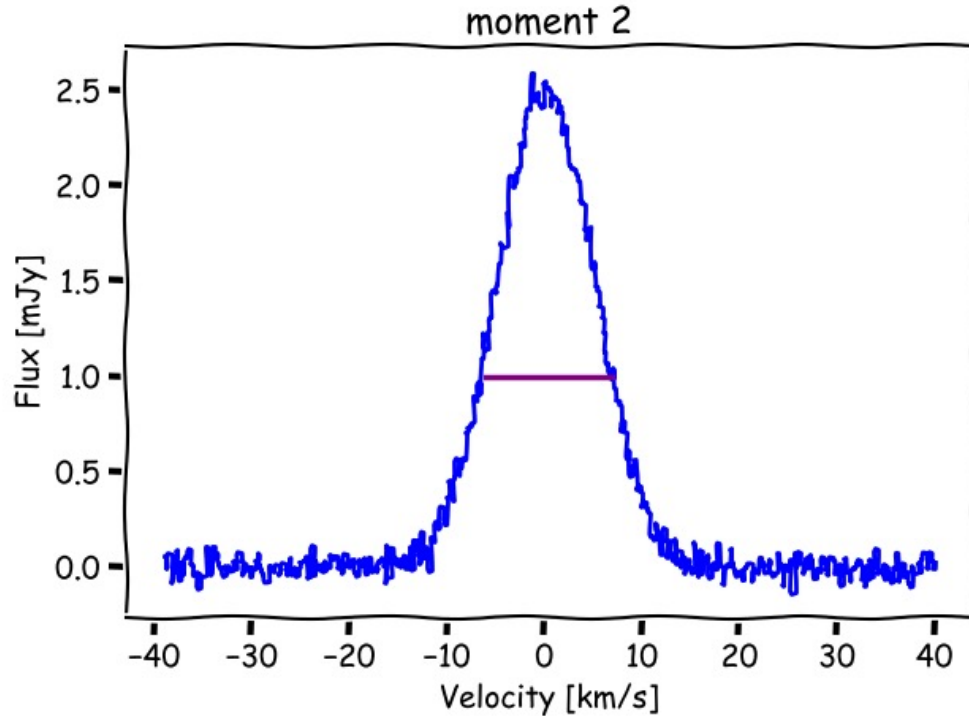
$$M_1 = \frac{\sum v I(v)}{\sum I(v)}$$



Moment 2 - Velocity dispersion

- `CASA <1>`: `immoments(axis='spec',
imagenam='my_image',
includepix= [3 * sigma, maxpix]
moments=[2], outfile='my_mom2')`
- use “includepix” for thresholding. E.g. from 3σ to the maximum pixel value

$$M_2 = \sqrt{\frac{\sum (v - M_1)^2 I(v)}{\sum I(v)}}$$





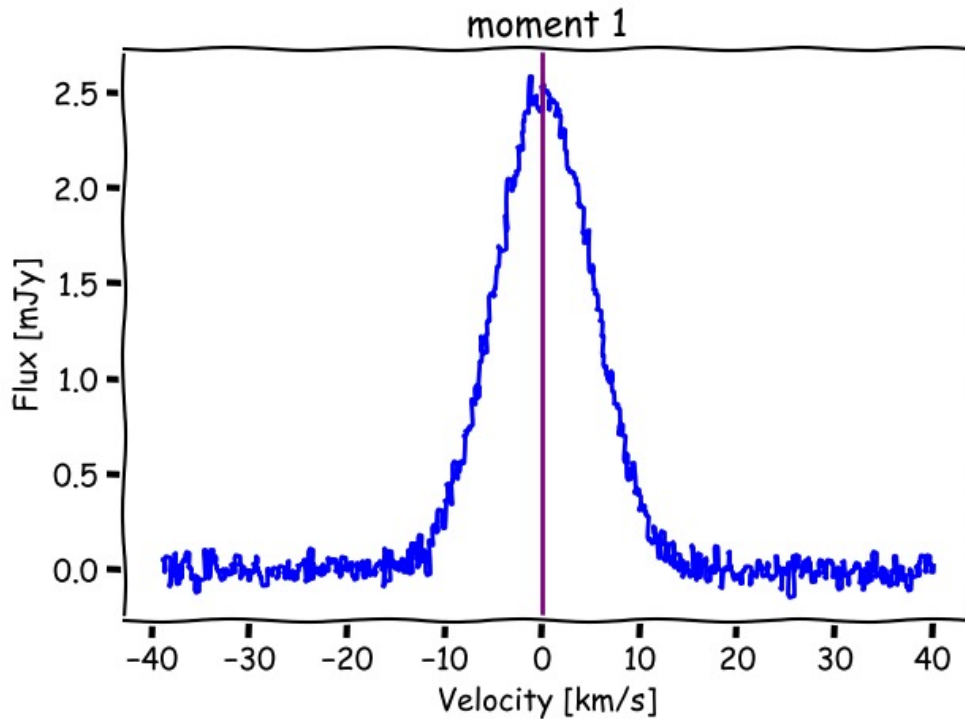
Exercise – Run the script

- **CASA <1>**: `mysteps = [4]`
- **CASA <2>**: `execfile('analysis_tools_script.py')`

Moment 1 - Velocity

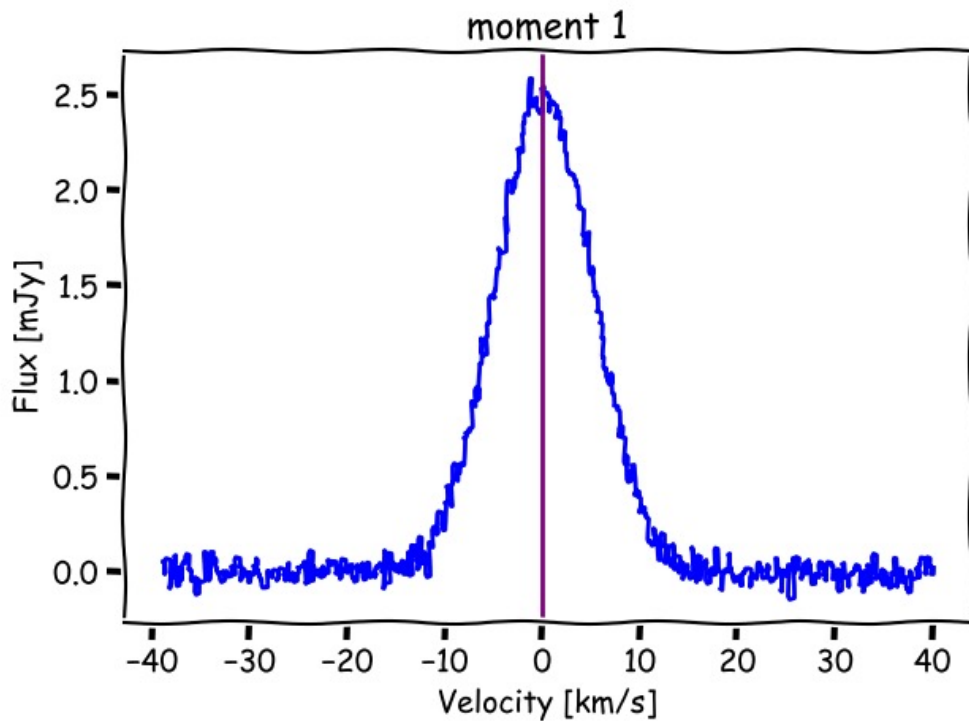
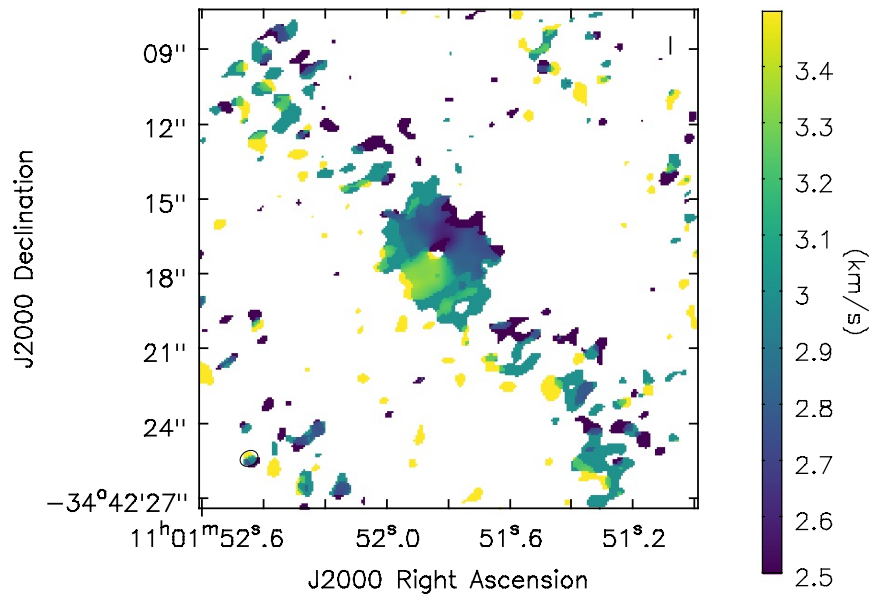
- `CASA <1>`: `immoments(axis='spec',
imagenam='my_image',
includepix= [3 * sigma, maxpix]
moments=[1], outfile='my_mom1')`
- use “includepix” for thresholding. E.g. from 3σ to the maximum pixel value

$$M_1 = \frac{\sum v I(v)}{\sum I(v)}$$





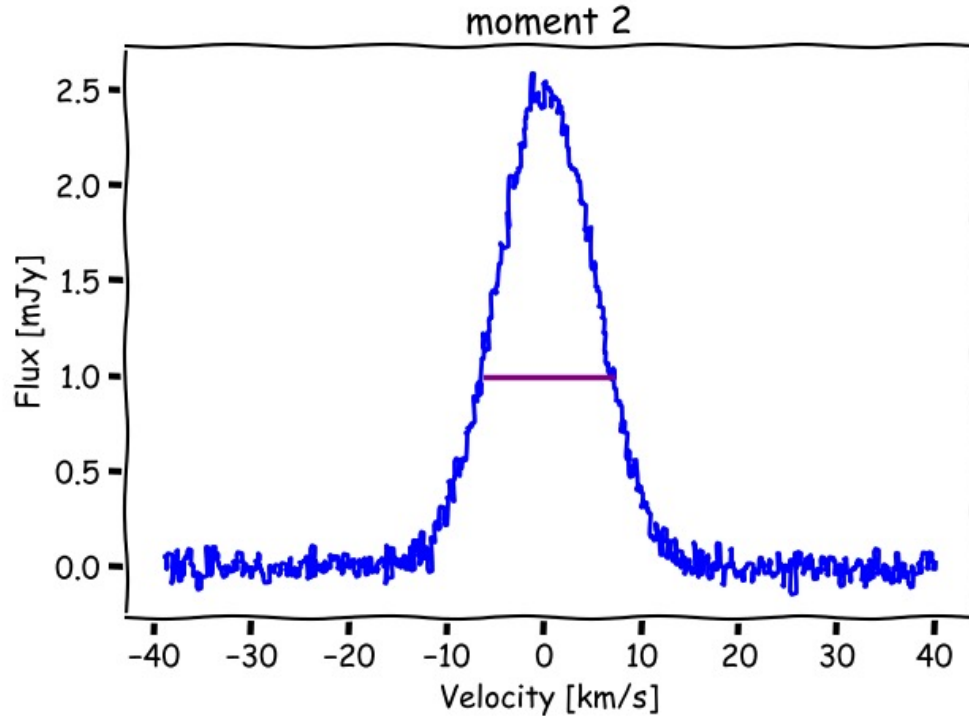
Moment 1 - Velocity



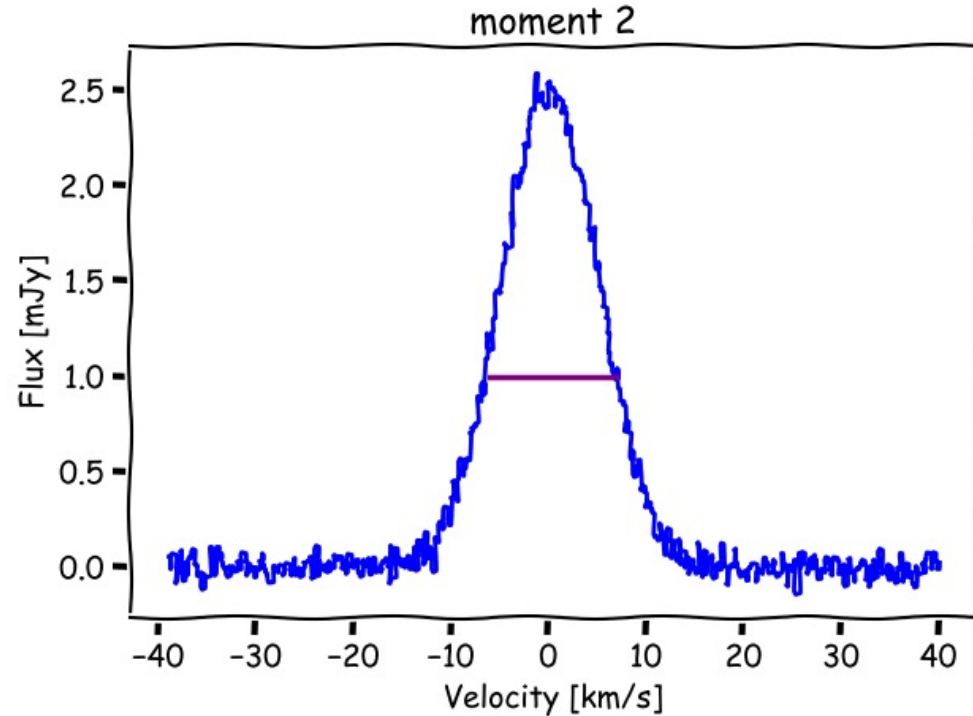
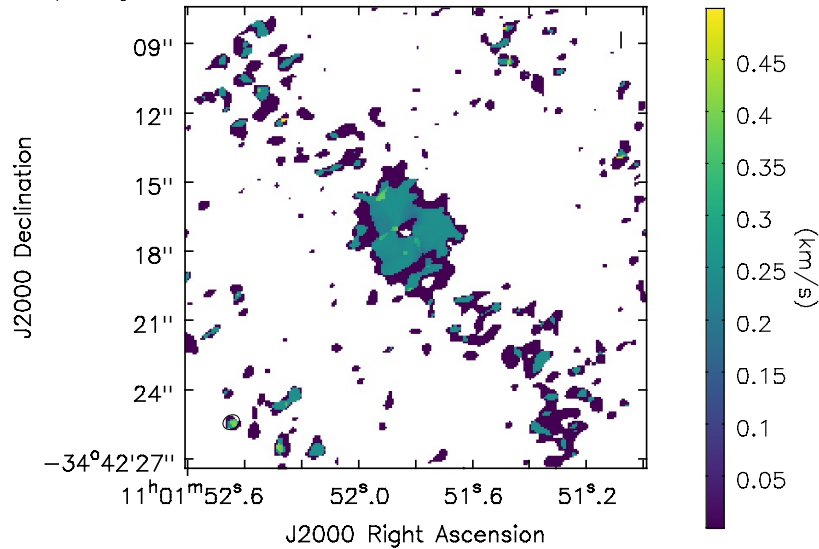
Moment 2 - Velocity dispersion

- `CASA <1>`: `immoments(axis='spec', imagename='my_image', includepix=[3 * sigma, maxpix] moments=[2], outfile='my_mom2')`
- use “includepix” for thresholding. E.g. from 3σ to the maximum pixel value

$$M_2 = \sqrt{\frac{\sum (v - M_1)^2 I(v)}{\sum I(v)}}$$



Moment 2 - Velocity dispersion





Exportfits

- Export your image to a .fits file for use with other software (e.g. ds9, python...)
- **CASA <1>**: `exportfits(imagename='twhya_n2hp.image',`
- `...: fitsimage='twhya_n2hp.image' + '.fits')`



Exercise – Run the script

- **CASA <1>**: `mysteps = [5]`
- **CASA <2>**: `execfile('analysis_tools_script.py')`



Modelling - imfit

- CASA <15>: `imfit(imagename="sis14_twhya_cont.image",`
- `...` `region="imfit_region.crtf", logfile="contin_fit.log",`
- `...` `model="sis14_twhya_cont.image.imfit",`
- `...` `residual="sis14_twhya_cont.image.fitresid")`

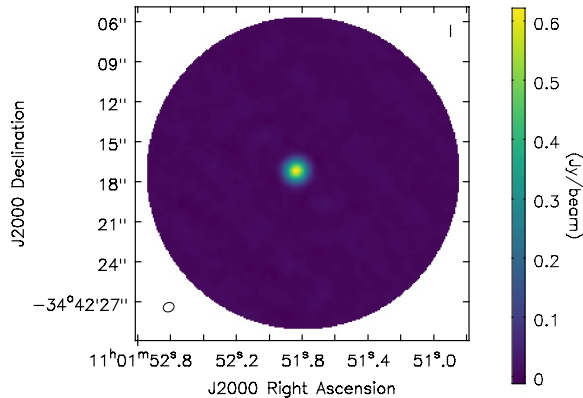


Exercise – Run the script

- **CASA <1>**: `mysteps = [6]`
- **CASA <2>**: `execfile('analysis_tools_script.py')`

Modelling – assessing results

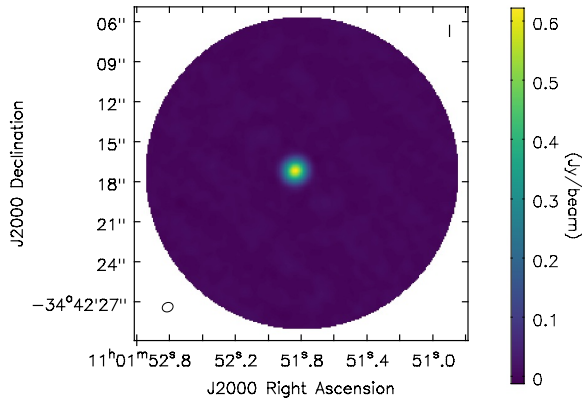
Image



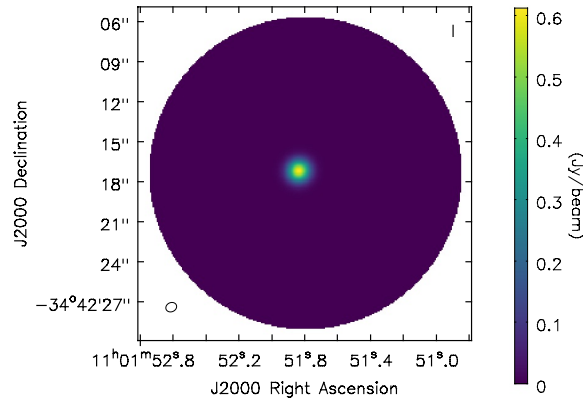
Let's check our model

Modelling – assessing results

Image



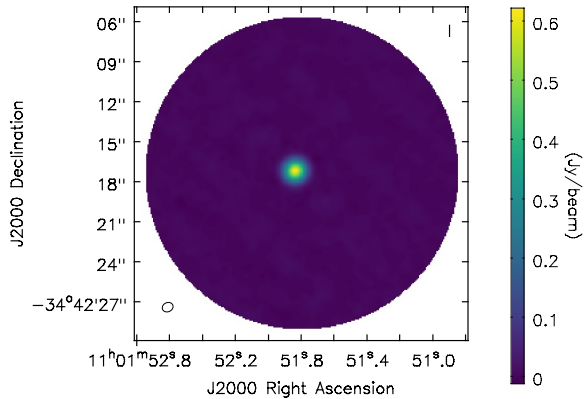
Model



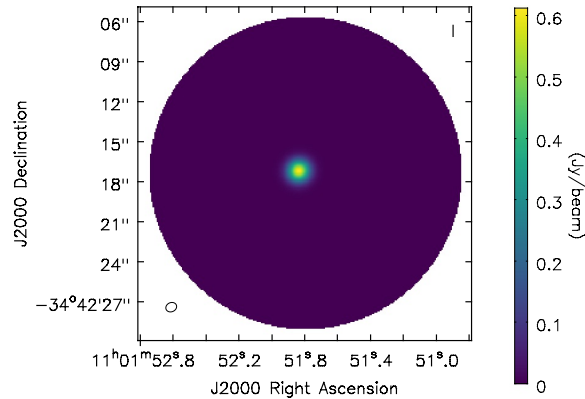
Looks like a reasonable fit... but let's check the residuals...

Modelling - assessing results

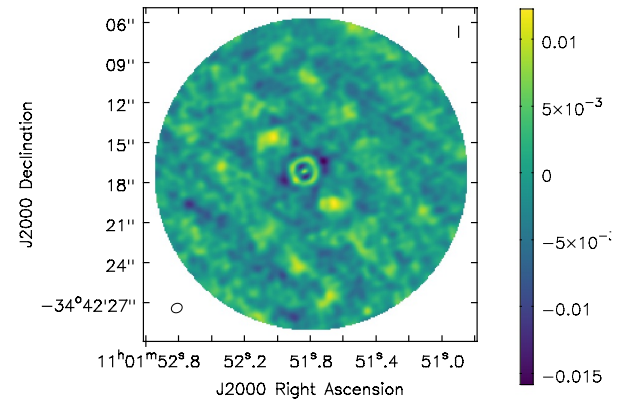
Image



Model



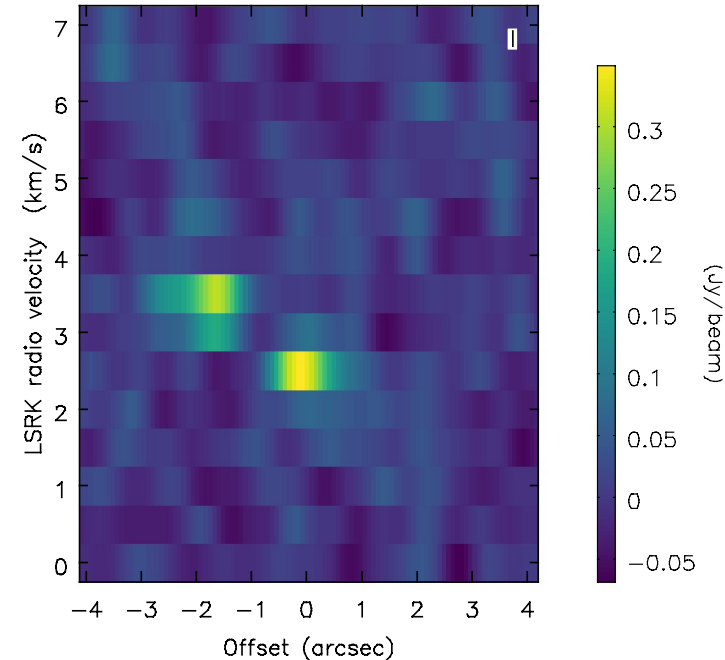
Residual




Position-Velocity (P-V) Diagrams – Interactive

- CASA <1>: mysteps = [7]
- CASA <2>:
execfile('analysis_tools_script.py')

- Click the pv button





Position-Velocity (P-V) Diagrams – automatic


- **CASA <1>**: `mysteps = [8]`
- **CASA <2>**: `execfile('analysis_tools_script.py')`



Re-gridding images – run the script

- **CASA <1>**: `mysteps = [9]`
- **CASA <2>**: `execfile('analysis_tools_script.py')`

- `imregrid(imagename='twhya_n2hp.image',`
- `template='GALACTIC',`
- `output='twhya_n2hp.image' + '.galactic')`



Bonus - Movies

- `ds9 my_cube.fits -cmap viridis -scale limits min max -movie slice gif movie.gif -exit`
- <http://ds9.si.edu/doc/ref/command.html#movie>
- If this fails update DS9, or for a worse-looking movie try:
 - `-cmap bb`
 - `-movie slice movie.mpeg`



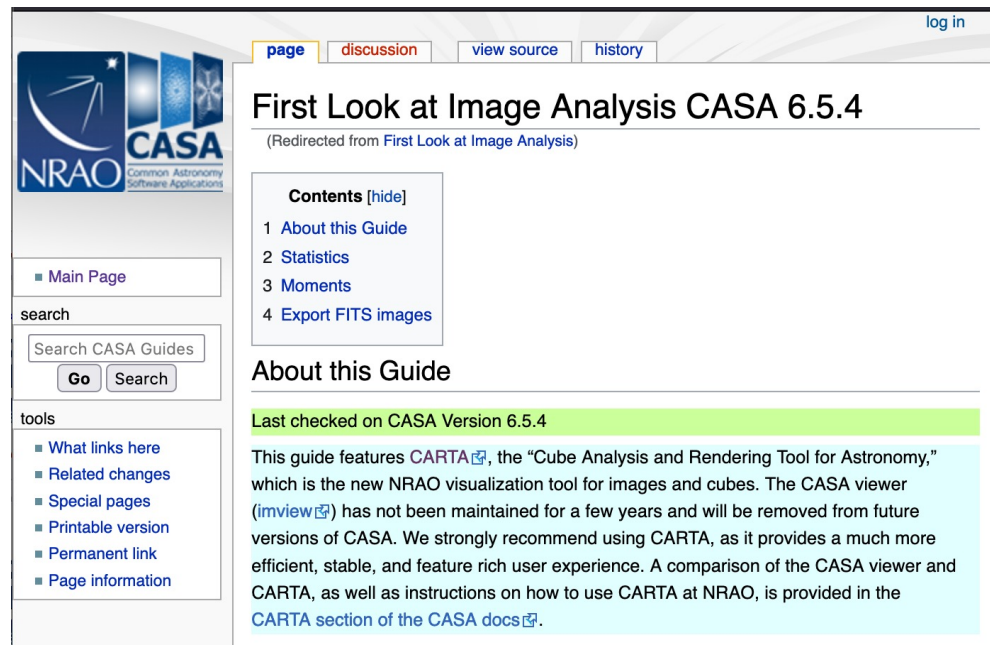
Exercise – Run the script

- **CASA <1>**: `mysteps = [10]`
- **CASA <2>**: `execfile('analysis_tools_script.py')`

Where next? CASA guides

https://casaguides.nrao.edu/index.php?title=First_Look_at_Image_Analysis_CASA_6.5.4

- Short tutorial on image analysis in CASA



The screenshot shows the web interface for the CASA 6.5.4 guide. At the top right, there is a 'log in' link. Below it are navigation tabs for 'page', 'discussion', 'view source', and 'history'. The main heading is 'First Look at Image Analysis CASA 6.5.4', with a sub-note '(Redirected from First Look at Image Analysis)'. A 'Contents [hide]' section lists four items: '1 About this Guide', '2 Statistics', '3 Moments', and '4 Export FITS images'. Below this is the 'About this Guide' section, which includes a green banner stating 'Last checked on CASA Version 6.5.4'. The text explains that the guide features CARTA, the 'Cube Analysis and Rendering Tool for Astronomy', and mentions that the 'imview' viewer is being phased out in favor of CARTA. A comparison of the two viewers and instructions for using CARTA are provided in the 'CARTA section of the CASA docs'.

log in

page discussion view source history

First Look at Image Analysis CASA 6.5.4

(Redirected from [First Look at Image Analysis](#))

Contents [hide]

- 1 [About this Guide](#)
- 2 [Statistics](#)
- 3 [Moments](#)
- 4 [Export FITS images](#)

About this Guide

Last checked on **CASA Version 6.5.4**

This guide features [CARTA](#), the “Cube Analysis and Rendering Tool for Astronomy,” which is the new NRAO visualization tool for images and cubes. The CASA viewer ([imview](#)) has not been maintained for a few years and will be removed from future versions of CASA. We strongly recommend using CARTA, as it provides a much more efficient, stable, and feature rich user experience. A comparison of the CASA viewer and CARTA, as well as instructions on how to use CARTA at NRAO, is provided in the [CARTA section of the CASA docs](#).

NRAO CASA
Common Astronomy Software Applications

■ Main Page

search

Search CASA Guides

Go Search

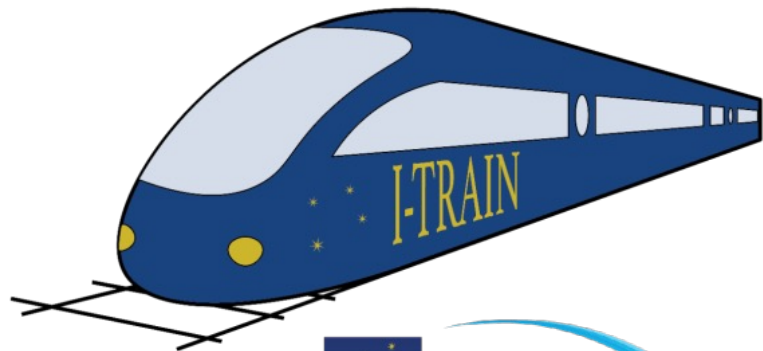
tools

- [What links here](#)
- [Related changes](#)
- [Special pages](#)
- [Printable version](#)
- [Permanent link](#)
- [Page information](#)

Where next? Itrain

<https://almascience.eso.org/tools/eu-arc-network/i-train>

- Video tutorials for some packages, for example:
 - I-TRAIN #3: UVMultiFit
 - I-TRAIN #9: Stacking spectra in the image domain with LineStacker



EUROPEAN ARC
ALMA Regional Centre



EUROPEAN ARC
ALMA Regional Centre || Allegro



Where next? Allegro software list

- List of ALMA-related software tools hosted by allegro.
- Will be announced soon – sign up to our newsletter to stay informed
- Newsletter: <http://bit.ly/AllegroNews>





Resources

- CASA documentation: <https://casa.nrao.edu/casadocs/latest/>
- CASA guides: https://casaguides.nrao.edu/index.php?title=First_Look_at_Image_Analysis
- CASA cookbook: https://casa.nrao.edu/Doc/Cookbook/casa_cookbook.pdf
- Download CASA: https://casa.nrao.edu/casa_obtaining.shtml
- Download DS9: <https://sites.google.com/cfa.harvard.edu/saoimageds9>
- EU arc network tools: <https://almascience.eso.org/tools/eu-arc-network-tools>

- Allegro maintains a number of analysis tools on our computers
- **Contact us at alma@strw.leidenuniv.nl with questions or requests for assistance with your alma data**