

Astroinformatics School 2009

ASAP Component on Friday 17 April 2009

Tutorial 2 - Data Reduction for multiple spectra

Prepared by: Kate Brooks

File Information:

tut2-kjb.txt	List of commands
2006-07-09_1431-M168.rpf	Data file (63 Mb)
3mm.txt	Emission line catalogue
QuickData.py	Data Reduction GUI (by Cormac Purcell)

Data Log:

ON-OFF-OFF-ON Position switching mode with Mopra

4 Scans (2 OFF and 2 ON)

4 x 2-GH bands (IF 0, 1, 2, 3)

Instructions:

1. Work through the list of commands given in the text file to calibrate data taken with the Mopra telescope and identify the emission lines. Commands should be typed line-by-line into ASAP. Seek help from the tutors if there are any commands you don't understand.
2. The commands given in the text file are for 1 Mopra frequency band only (IF0). Now work on the other 3 bands and make a final plot showing all four frequency bands stitched together.
3. Now repeat the calibration steps using the Mopra Data Reduction GUI. This GUI has been created via a python script (QuickData.py).

Estimated time to complete ~ 30 mins

```

# Start ASAP
localhost> asap

# Load data into memory and display short description
s = scantable('2006-07-09_1431-M168.rpf')
print s

# Set the plotting mode
plotter.set_mode(stacking='i', panelling='t')
plotter.set_histogram()
plotter.set_colours('black')
plotter.set_linestyles('solid')

# Plot all the raw data
plotter.plot(s)

# Form the quotient spectra
q = s.auto_quotient()

# Average all scans in time, weighting according to Tsys value
av = q.average_time(weight='tsys')

# Average the two polarisations together, weighting according to
Tsys value
iav= av.average_pol(weight='tsys')

### Work on IF0 only

# Define a selector
sell = selector()

# Select the first IF for scantable 'iav'
sell.set_ifs(0)
iav.set_selection(sell)

# Plot the first IF with x-axis in channel number
iav.set_unit('channel')
plotter.plot(iav)

# Flag out the end channels
msk1 = iav.create_mask([0,120])
iav.flag(msk1)
msk1 = iav.create_mask([7900,8200])
iav.flag(msk1)

# Redo the plot
plotter.plot(iav)

# Identify emission lines
iav.set_unit('GHz')
plotter.plot(iav)

# Load in catalogue of emission lines and overlay on plot
lc3 = linecatalog('3mm.txt')
plotter.plot_lines(lc3,location='top',rotate=90,doppler=-45)

```