Astroinformatics School 2009

ASAP Component on Friday 17 April 2009

Tutorial 4 – Data Reduction for Parkes Methanol Multibeam

Prepared by: Jimi Green

File Information:

MMB_MX_Tutorial.txt	List of commands
mmb-mx.rpf	Data file (9.4 Mb)

Data Log:

7 Spectra taken with Parkes Methanol Multibeam

Instructions:

1. Work through the list of commands given in the text file to calibrate data taken with the Parkes Methanol Multibeam. Commands should be typed line-by-line into ASAP. Seek help from the tutors if there are any commands you don't understand.

2. Write a python script to automate the calibration procedure for data taken with the Parkes Methanol Multibeam. Incorporate the list of commands used in step 1 as well as a routine to cycle through the 7 different beams.

   Note: Your python script should be executed in a terminal (and not within ASAP) with the following command:

   ```
   localhost> python –i myscript.py
   ```

Estimated time to complete ~ 40 mins

```
# Load data (with filename mmb-mx.rpf) into memory and display
data = scantable('mmb-mx.rpf')
print data

# Set the polarisation feed type
data.set_feedtype('circular')

# Select just the first IF (the data actually contains two, the
methanol transition at 6.7GHz and the excited-state OH transition
at 6GHz, but we will only look at methanol).
sel=selector()
sel.set_ifs(0)
data.set_selection(sel)

# Set the rest frequency
data.set_restfreqs(6.6685192e9)

# Set the cal values for the 7 beams, both polarisations.
calfact = (( 2.29, 2.28 ), ( 2.18, 1.93 ), ( 4.37, 4.37 ), (2.53,
3.20 ), ( 3.69, 3.89 ), ( 3.74, 3.51 ), ( 1.98, 1.70 ))
sel = selector()

# Apply cal factors to first beam, first polarisation
sel.reset()
sel.set_beams(0)
sel.set_polarisations(0)
data.set_selection(sel)
data.scale(calfact[0][0], insitu=True, tsys=True)

# Apply cal factors to first beam, second polarisation
sel.reset()
sel.set_beams(0)
sel.set_polarisations(1)
data.set_selection(sel)
data.scale(calfact[0][1], insitu=True, tsys=True)

# Now repeat above 10 steps for the other 6 beams

# Reset selection parameter
data.set_selection(oldsel)

# Set plotter output to show both polarisations on the same plot,
but each beam on a separate plot.
plotter.plot(data)
plotter.set_mode('p','b')

# Plot the first scan only
sel = selector()
sel.set_scans(1)
plotter.set_selection(sel)

# Average 'off-source' scans for each beam, then use as the
reference scan to form a quotient.
q = data.mx_quotient()
plotter.plot(q)

# Define the channel unit.
```

```
q.set_unit('km/s')

plotter.plot()
plotter.set_range(-60,-10)

# Average all the multiple beam data together to form a long
integration spectrum.
avb = q.average_beam()
plotter.plot(avb)
plotter.set_range()

# Average polarisations together
avp = avb.average_pol()
plotter.plot(avp)

# Fit a linear baseline (avoiding the maser feature)
msk=avp.create_mask([-110,-70],[10,40])
avp.poly_baseline(msk,order=1)
plotter.plot(avp)

# Make a nice file
plotter.set_colors('black')
plotter.set_legend(mode=-1)
plotter.set_title('G300.969+1.148')
plotter.save('G300p96.ps')
```