Astroinformatics School 2009

ASAP Component on Friday 17 April 2009

Tutorial 3 - Spectral-line gaussian component fitting and rms noise levels

Prepared by: Shari Breen

File Information:

tute3-slb.txt: List of commands
mon.asap: Data file

Data Log:

ON-OFF Position switching mode with Hobart
4 scans (2 OFF and 2 ON)
Dual pol, 1 IF

Instructions

- Work through the list of commands given in the text file to calibrate the data and fit gaussian components to the emission. The commands should be typed into ASAP line-by-line until you get to the ADVANCED FITTING step which will require you to experiment with the parameters.
- ADDITIONAL exercise: Determine the rms noise in the spectrum using the create_mask function you learned in the first section and the stat command specified in the text file.

Fitting gaussians: An example.

Load the scantable for the source into ASAP

```
ASAP> mon=scantable('source')
ASAP> mon.summary()
```

Select the on-source scans

```
source=mon.get_scan([1,3])
```

Select the reference scans

```
ASAP> ref=mon.get_scan([0,2])
```

Make the quotient spectra and plot it

```
ASAP> quot=quotient(source,ref)
ASAP> plotter.plot(quot)
```

Remove the baseline and plot

```
ASAP> quot.auto_poly_baseline()
ASAP> plotter.plot(quot)
```

Average the 2 scans together

```
ASAP> av=quot.average_time(align=True)
ASAP> plotter.plot(av)
```

Check the the rest frequency is set correctly and define unit as km/s

```
ASAP> av.get_restfreqs()
ASAP> av.set_unit('km/s') ASAP> plotter.plot(av)
```

Scale the two polarizations separately

```
ASAP> sel=selector()
ASAP> sel.set_polarisations(0)
ASAP> av.set_selection(sel)
ASAP> av.scale(50.6)
ASAP> sel.set_polarisations(1)
ASAP> av.set_selection(sel)
ASAP> av.scale(32.5)
```

Average the polarizations together

```
ASAP> avpol=av.average_pol()
```

Plot the data

```
ASAP> plotter.plot(avpol)
```

Create a mask that contains the emission

```
ASAP> msk=avpol.create_mask([-10,30])
```

Set up some fitting parameters

```
ASAP> f=fitter()
ASAP> f.set_scan(avpol,msk)
ASAP> f.set_function(gauss=2)
```

Fit a gaussian to the emission, plot the emission and fit

```
ASAP> f.fit()
ASAP> f.plot()
```

Save the plot

```
ASAP> plotter.save('monr2.eps')
```

Get the fit parameters

```
ASAP> f.get_parameters()
```

Plot the fit residuals

```
ASAP> f.plot(residual=True)
```

Plot each of the fitted gaussians separately, with or without the fit parameters overlaid

```
ASAP> f.plot(components=0,plotparms=True)
ASAP> f.plot(components=1)
```

ADVANCED FITTING: Sometimes it is necessary to force some of the fitting parameters such as peak or FWHM. For example, if you want to fix the peak of the first gaussian component to 39 Jy you can use the following command,

```
ASAP> f.set_gauss_parameters(39, 10.2, 1, peakfixed=1, component=0)
```

You can now experiment with forcing the values of the different parameters. To look at the help file on this function type

```
ASAP> help fitter.set_gauss_parameters
```

Additional Exercise: Determine the rms noise in the spectrum.

1) Create a mask that excludes the emission (if you leave it in the rms noise will be significantly inflated), using the same command as before except this time you will have to specify two ranges of values either side of the emission.

2) Find the rms noise of the spectrum

```
ASAP> avpol.stats(stat='rms',mask=*your mask*)
```

The stats function can be used to extract many other statistics from the data such as the maximum and minimum values, the median value and many more. Use help to find out how to extract these statistics from the spectrum.