Spectroscopy with pyspec.

a tool written by Samuel George

Last updated October 10, 2009

1 Introduction

The aim of this manual is to guide you through the use of pyspec and general ideas of spectroscopy.

1.1 Installing dependencies of pyspec

To get the lastest version of pyspec and for installation instructions see:
`http://www.krioma.net/pyspec.php`

2 DS9

2.1 Background to DS9

DS9 is a very useful image display program that allows interactions with images. We will utilise some of its features during the image analysis and so it is important that you have a basic understanding of how to use it.

You might like to know that DS9 stands for Deep Space Nine (i.e. the Star Trek television program) and is the follow up to a package known as TNG (stands for The Next Generation – again Star Trek). For more information on DS9 see: `http://hea-www.harvard.edu/RD/ds9/`

2.2 Installing DS9

DS9 is not required for running the software but is used for visualisation of the FITS file such that the correct calibration can be made. DS9 is simple to install on any Linux distribution, here are some instructions:

wget http://hea-
www.harvard.edu/saord/download/ds9/linux/ds9.linux.5.6.tar.gz
tar -zxf ds9.linux.5.6.tar.gz
sudo mv ds9 /usr/local/bin
sudo chmod +x /usr/local/bin/ds9

3 Accessing DS9

To access DS9, at the terminal window type: **ds9 &**

This will open up ds9 but still allow you to use the terminal window. *(The "&" command is a general Unix command for running programs in the background, such that your command line is free for you.)*

4 Using DS9

The menu bar across the top of the program is pretty self explanatory, if you need any help with this though see help (top right). Below the menu bar you will see an information panel, this is where all the information about the image is displayed.

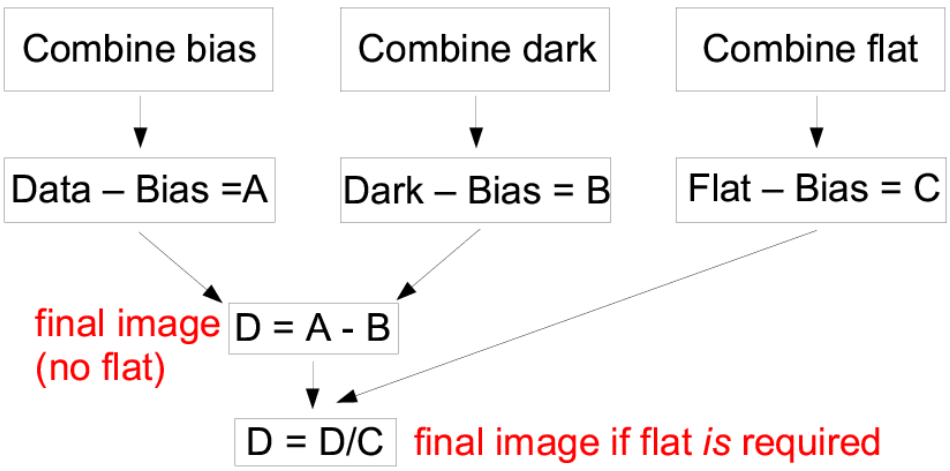
5 Running pyspec

This section provides a short overview of the steps completed within pyspec with a full run through of the package at the end.

5.1 Image Preparation

The first step in manipulating your data is to remove the bias from your frames. Firstly you must combine all of the bias frames into an average frame. Before you go about this it is worth taking a look at your images using DS9 to check for any CCD readout errors (it is quite likely you would have noticed these at the observatory, but just in case you have forgotten which images have problems). you should not combine data that is bad as this will, obviously, introduce huge errors in your work. The method is summarised nicely in figure 1.

Figure 1: Image preparation flow chart



This is the first step in pyspec. Bias files should be named bias*.fit where the * represents any postive interger. The same is said for your dark*.fit files.

5.2 Combining Images

If you have multiple images of the same object and none of the images are particularly bright (if there are very strong and obvious features in your images then you probably don't need to do this), i.e. counts in the image are not much above the background level, one can add the images together. For our data reduction the important thing is that we can simply add two images together. By doing this you get a factor of N increase in the signal but only $N^{0.5}$ for the noise and so can make a drammatic gain in faintly exposed spectra. Make sure you only add images together that are reasonable. If you simply run this task it will ask you which two images you want to use and how to combine them. When you run pyspec it will ask you if you wish to combine multiple target images, if you choose yes it will ask you to input the list of images.

5.3 Producing Spectra

In DS9 you should open your reference lamp spectra. Now find a line (absorption or emission) along the dispersion axis and note its position (this is the physical x value). This should be done for all the lines in the reference lamp. You should then using reference data for your lamp identify the lines. Some calibration data can be found at `http://www.sr.bham.ac.uk/observatory/cal_lamps.php`

You should record these values in a text file, called ref.txt - you can use pico or any other text editor todo this. The table should just be a table of pixel and wavelength values (see ref_example.txt for the structure).

5.4 Region to extract

You also need to look at your dataset (for both your calibration and source targets) and find the correct row that you would like to extra and the number of rows over you wish to sum, for weakly exposed spectra you might want to make this large (however, for spectra with rotation structure, e.g. Saturn you will want todo this over a narrow range).

5.5 Conversion between pixel and wavelength

When you have the made the calibration table for your reference lamp you should run pyspec. pyspec performs a linear least squares fit to your data and thus at least three peaks need to be identified (though eventually the intention is to offer other fitting) and will output:

cal_out.txt - a data table of pixel vs wavelenght for your final calibrated dataset, useful for reploting data / reading into fitting tools (you can read this data into IRAF or anything else really).

output.txt - a line of numbers, these represent the values determine of the median values of bias and the dark-bias, c and m from the linear regression. This is used if you choose not to repeat the whole process for another dataset.

fig_Ia.png - pixel / intensity plot for reference lamp.

fig_Ib.png - the wavlenght calibration regression, completed by python tool polyfit.

fig_Ic.png - the wavelength calibrated reference lamp.

fig_Id.png - the calibrated target dataset.

fig_out.png - the calibrated target dataset (if reapplying the calibrated dataset).