Margaret Haswell
PHYS265
5/18/2025

pyspeckit: Analyzing Astronomical Spectra

1. What is the name of the package? Describe what is the basic aim of what the package does or solves?

This package is called “pyspeckit”. The purpose of the package is to be a tool for assisting astronomers with fitting theoretical distributions to astronomical spectra, and performing baseline and continuum fits. It is mostly intended for fitting spectra in the visible and radio ranges to gaussian and voigt profile distributions, but it can be used for other parts of the electromagnetic spectrum as well. Additionally, it can be used on data collected using a number of different methods and instruments. And it also has many different methods for plotting the spectra and fits.

2. Why/how did you select this package?

I selected this package from the list provided because I am an astronomy major, and I want to learn more about tools for astronomers. In my current astronomy lab we have been working with spectra of stars and galaxies for various purposes, such as creating a rotation curve of the Milky Way and finding Hubble’s Constant. But in that class, we only use very basic python, and we usually have to analyze those spectra by eye. So I was curious to learn more about how python can be used to analyze spectra in more accurate ways.

3. How old is the package? Does it have a genealogy, i.e. what related codes came before or after. Are there other codes you can find that solve the same problem? Can you figure out which version you installed?

The package was first copyrighted in 2010, so the package is around fifteen years old. Before that, the community mostly relied on the Image Reduction and Analysis Facility (IRAF), but recently that has fallen out of use, which is why the authors created pyspeckit.

Pyspeckit used both IRAF and another library called XSPEC, which is meant for fitting x-ray spectra, as a base for the fitting and plotting components of the code. Now, there are many other libraries which can perform similar functions to pyspeckit. For example Specutils, Sherpa, and Pyplatefit are all useful for fitting spectra.

I used version 1.0.4.dev of pyspeckit.

4. is it still maintained, and by the original author(s)? Are there instructions how to contribute to this project.

It appears to still be maintained by the main author, Dr. Adam Ginsburg from the University of Florida, who occasionally updates the github repository. The README provides instructions on how to contact the authors to provide use cases, report issues, and contribute to the project.

[5, 6]. evaluate how easy it was to install and use. What commands did you use to install? Does it install via the "standard" pip/conda, or is it more complex?

Pyspeckit was very easy to install. I used the following commands to install the package:

```
!pip install pyspeckit
```

```
import pyspeckit as psk
```

It was also pretty easy to use, since their github and website have plenty of examples of how to perform different tasks.

7. is the source code available? For example, "pip install galpy" may get it to you, but where can you inspect the code?

The source code is available at this github link.

8. Is the code used by other packages (if so, give one or two examples). ASCL codes have citations via their ADS link. See also 22.

Yes, according to this paper (https://ui.adsabs.harvard.edu/abs/2024A%26A...685A.164J/abstract) there are several other packages that use code from pyspeckit, including astroclover and SCOUSEPY, which were both created for analyzing spectral data.

9. How is the code used. Is it commandline, python script, or a jupyter notebook, or even a web interface?

The source code and all of the examples in the github are python script files, and it can be used in jupyter notebook as well.

10. Provide examples using the code. if you prefer to use a jupyter notebook instead of a python script, that's ok. See also 12.

The following code imports the data of a spectra of an Hbeta line, finds the error in the data, and then creates a spectrum object.

```
data = np.genfromtxt("ngc253_hb.tab")
wavelength = data[:,0]
intensity = data[:,1]

# find noise/standard deviation:

n1 = intensity[:205]
n2 = intensity[216:]
noise = np.concatenate((n1, n2))
std = np.std(noise)
error = std*np.ones_like(intensity)

sp = psk.Spectrum(data = intensity, error = error, xarr = wavelength, xarrkwargs={'unit':'AA'}, unit='Flux')
```

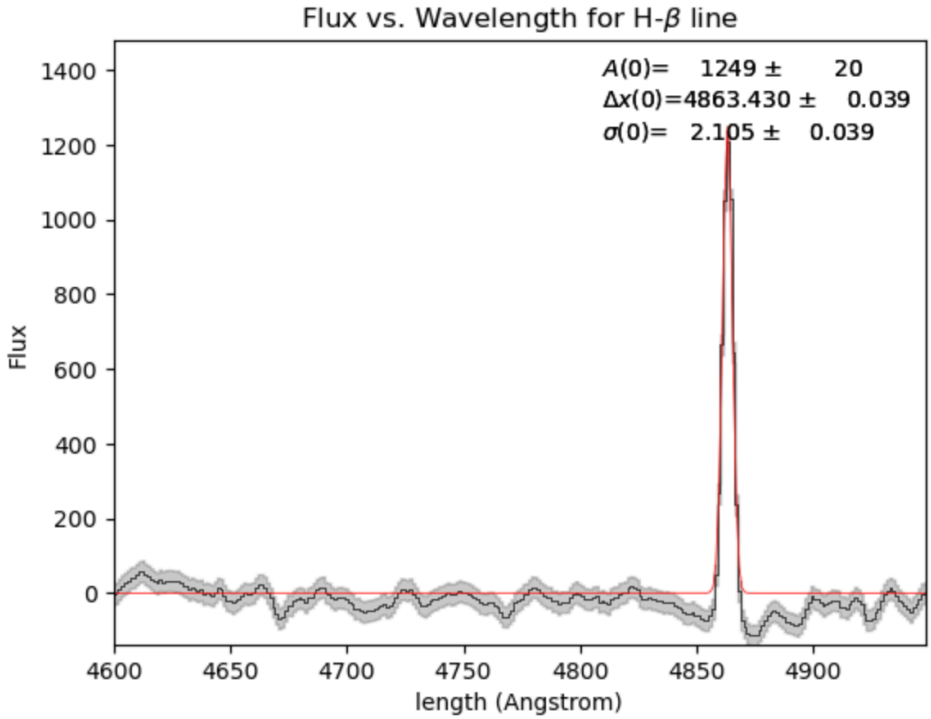
The following code fits the data to a gaussian distribution, and plots the data and the fitted theory.

```
sp.plotter()
sp.baseline()

amplitude_guess = intensity.max()
center_guess = 4863
width_guess = 5 # FWHM
guesses = [amplitude_guess, center_guess, width_guess]

sp.specfit(fittype='gaussian', guesses=guesses)

sp.plotter(errstyle='fill')
sp.specfit.plot_fit()
```

11. Does the package produce figures, or are you on your own? Is matplotlib used?

The package does produce figures of the fit and data. It uses matplotlib to create the plots.

12. Your code and report should show at least one figure, and create a nice figure caption explaining what it shows. Your notebook should show how the figure was made (i.e. be reproducible). Second figure is optional, but only use it when you need to illustrate something extra.



Here is the figure of the H-β emission line data and fit from NGC-253, corresponding to the code above. The black line is the data, and the gray region represents the error in the data. There is a clear emission line at about 4860 Å, which we fit to a gaussian distribution, which is shown in red. The annotations in the top right corner include the height, mean, and standard deviation of the gaussian.

13. Is the package pure python? or does it need accompanying C/C++/Fortran code?

The package only uses python.

14. What is the input to the package? Just parameters, or dataset(s), or can they be generated from scratch?

You can input your own datasets, which is the primary purpose of the package, and then fit and plot them.

15. What is the output of the package? Just parameters, or dataset(s)?, or just a screen output you would need to capture

The package will output the fit parameters and a plot of the data and fit. For instance, in the Gaussian fit example above, it produced the plot, and then the height, mean, and standard deviation of the fit as an annotation on the plot.

[16, 17]. Does the code provide any unit tests, regression or benchmarking? How can you feel confident the code produce a reliable result? (see also previous question)

Yes, there are tests in place to check that the code works for different versions of python. They can be see under the “Actions” header in github, and they were run by one of the authors.

18: What (main) python package(s) does it use or depend on (e.g. numpy, curve_fit, solve_ivp) - how did you find this out?

This package requires numpy, matplotlib, and astropy. Mpfit, a package used for solving non-linear least-squares regression, is included in the package. Scipy is listed as optional, and is only a requirement if you need to use certain types of optimization. It also says you will probably want at least one from the following list: astropy, atpy, or hdf5, which are all packages designed for data processing and organization. Astropy and atpy are specific to astronomy.

This information is listed in the pyspeckit github repository in a doc titled “Installation and Requirements”, and in the README.

19. What kind of documentation does the package provide? Was it sufficient for you?

Pyspeckit has a lot of documentation on their website detailing the classes, methods, fitting and plotting guides, and examples of how to use the package. The github repository containing the source code also had lots of examples of usage. The examples were particularly helpful for this project.

20. If you use this code in a paper, do they give a preferred citation method?

Their preferred citations are:
https://ui.adsabs.harvard.edu/abs/2022AJ....163..291G/abstract and
http://adsabs.harvard.edu/abs/2011ascl.soft09001G.

21. Provide any other references you used in your report.

Pyspeckit website: https://pyspeckit.readthedocs.io/en/latest/index.html

Specutlis: https://specutils.readthedocs.io/en/stable/

SHERPA: https://parameter-sherpa.readthedocs.io/en/latest/

Pyplatefit: https://pyplatefit.readthedocs.io/en/latest/tutorial.html

xspec: http://astroa.physics.metu.edu.tr/MANUALS/xspec12_html/XspecIntroduction.html

astroclover: https://arxiv.org/abs/1909.08727

sousepy: https://scousepy.readthedocs.io/en/latest/index.html

IRAF: https://ui.adsabs.harvard.edu/abs/1986SPIE..627..733T/abstract

Fast fitting of spectral lines with gaussian and hyperfine structure models:
https://www.aanda.org/articles/aa/full_html/2024/05/aa49044-23/aa49044-23.html

22. can you find two other papers that used this package. E.g. use ADS citations for ASCL based code. See also 8.

ADS has 62 papers that cite pyspeckit:
https://ui.adsabs.harvard.edu/abs/2022AJ....163..291G/citations.

Two examples are: https://ui.adsabs.harvard.edu/abs/2024MNRAS.533.1938C/abstract and
https://ui.adsabs.harvard.edu/abs/2025ApJ...983...49F/abstract

23. Did you have to learn new python methods to use this package? Or was the class good enough to get you through this project.

I did not have to learn any new methods, the class was sufficient.

24. Final Disclaimer: you need to state if you have prior experience in using the package or the data, or this is all new to you. In addition, if you collaborated in a group, as long as this is your work.

I do not have any prior experience with this package or data.

I worked in a group with Jasmin Mohammadi, Zya Woodfork, and Debika Biswas.