This library allows you to:

- Define and write input files for Cloudy code. As you can have it in a code, you may generate automatically sets of input files, changing parameters from one to the other; easy grid of models.
- Read the Cloudy output files and play with the data: you will be able to plot line emissivity ratio vs. the radius of the nebula, the electron temperature, or any Cloudy output.
- Build pseudo-3D models, a la Cloudy_3D. This means: run a set of models, changing parameters (e.g. inner radius, density) following angular laws, read the outputs of the set of models and interpolate the results \( T_e, n_e, \) line emissivities in a 3D cube.
- If you have a multi-core computer, distinct models can be run at the same time on distinct cores (hard-way parallelisation).

Once the 3D cube of emissivities is obtained, any rotation and projection on the sky plane can be obtained. False colors images: RGB can be obtained, mixing any monochromatic images obtained by selecting emission lines.

Monochromatic images can be generated, as well as line ratio maps, cut within the volume of the nebula from the 3D cells or the image pixels, plots of any variable can be easily obtained. Matplotlib graphic libraries allow the user to obtain very illustrative figures;

Once 3D emissivity cubes have been obtained, one can define a velocity field and compute line profiles through any aperture. Py-diagrams can also be generated through any site.

Grid of models can easily be obtain. Example: building a grid of Cloudy models to separate pure star forming galaxies in the BPT diagram.

PyCloudy has ability to work with: PyNeb library /Luridiana, Morisset & Shaw, 2015/ using T_e, N_e and X_e, PyNeb is used to re-compute the line emissivities according to any atomic data available from its large library. Energetic balance is obviously slightly broken, but it still may be very useful to check effects of changing atomic data (not that easy from within Cloudy).

Once emission line images have been obtained, it is simple to put a mask on it to only extract the intensity observed through a given slit.

The PyCloudy package provides easy ways to obtain some Cloudy outputs in many different units. No more questions about erg/s/cm2 vs. \( \mu \text{Jy} \).

https://sites.google.com/site/pycloudy/