Radar Precipitation Feature/TRMM-identification Codes

Here is a list of codes that are used for the radar PF analysis and what they do.

`chivo_pf_id_v1_maxdbz.py`

What it is: Main code to identify features.
Function: `PF_finder_max`
Inputs function takes (in order):

- reflectivity
- x
- y
- z
- conv/strat
- HID
- Ice Water Path (IWP)
- Altitude for rain rate (in km)
- dBZ threshold for PF (default is 0 dBZ)
- dBZ threshold for convection/convective TRMM features (default is 40 dBZ)
- height of DCC (in km, default is 10 km)
- area threshold for WCC/DWCC (in km, default is 1000 km)
- area threshold for BSR (in km, default is 40,000 but we used 10,000 km)

Things that are hard coded in that you may want to look into:

- conv/strat indices
- composite reflectivity levels (3-5 km)
- HID classifications (from CSU_Radartools but will need to be updated if you are using NCAR PID)
- PF threshold set at > 4 km^2 area and > 3 km echotop (don't want small pixels that could be noise)
- Rain rate equation (from Chen et al. 2017, eq3)
- colors of TRMM ellipses (if you want to plot a visual)

`chivo_pf_id_submit_ppi.py`

What it is: Code that loops through all files and calls the `PF_finder_max` function
Functions: `read_files` and `write_output`

`read_files` function summary: extract data from the input file to be used in the `PF_finder_max` function. This includes the date, x, y, z, reflectivity, HID, IWP, and conv/strat. Two additional codes are called here to get the HID/IWP (`get_csu_hid` from `hid_iwp_chivo.py`) and conv/strat (`conv_stra_sep` from `hyu_convstrat.py`).

`write_output` function summary: create a dictionary to output statistics in a csv file. This is based on the output from the `PF_finder_max` function and everything is hard coded in. This means if you add/subtract/change variables that you output, you will have to make sure those changes are made here too.

Other info:

1) I decided to run the HID/IWP on the gridded data in this script as well. There is a section towards the bottom which requires sounding data to do this. This may have to be updated.
2) You will have to change the paths of the input data in the loops.

`hyu_convstrat.py`

What it is: Code by Hungjui Yu that is basically the Steiner et al. 1995 conv/strat partitioning algorithm written in python.
Function: `conv_stra_sep`
Inputs function takes (in order):

- reflectivity (must be a 2D grid)
- y (must be a 2D grid)
- x (must be a 2D grid)
- grid resolution (1 km)
- projection type ('C')
- which algorithm ('SHY')

We calculate c/s from 3-5 km, then composite over those levels (to match with composite reflectivity)

`hid_iwp_chivo.py`

What it is: Code to calculate HID and ice water path from CSU_Radartools on the gridded data.
Function: `get_csu_hid`
Inputs function takes (in order):

- radar file
- sounding times
- sounding files

This finds the closest sounding time to radar time and pulls out the temperature profile which is needed for the HID and IWP. Ice and water masses are calculated as well. IWP is the sum of ice mass above the melting level (in this case above 5 km).