Data Selection Process

The script *calibrate_day_att* in *utilities/calib_functions.py* is used to process the volume scans to estimate a Z calibration bias. The methodology follows that of Gourley et al. 2009: https://doi.org/10.1175/2008JTECHA1152.1

The steps in the code are described as follows:

1. Calculate height of every range gate, “beam_height”, in metres above sea level

```
beam_height[(360*j)+i,:] = radh/1000 + np.sin(np.deg2rad(el[j]))*rg + np.sqrt(rg**2 +
    (6371*4/3.0)**2) - (6371*4/3.0);
```

2. Extract melting layer height and ZDR bias from the files produced during the ZDR processing

```
mlh, zdr_bias = extract_ml_zdr(time, ml_zdr)
```

3. Where the beam is above the melting layer, set values to nan

```
zind = beam_height > mlh
zdr[zind==True] = np.nan
```

4. Create a list of azimuths and elevations that you want to exclude from the processing e.g. due to known areas of blockage. These are set up in the SETTINGS.py file.

```
exclusions = SETTINGS.EXCLUSIONS
exclude_radials = np.any([np.all([rad.elevation['data']>=ele[0],
                                  rad.elevation['data']<ele[1],
                                  rad.azimuth['data']>=azi[0],
                                  rad.azimuth['data']<azi[1]],axis=0) for ele, azi in
exclusions],axis=0)
az_index = np.where(~exclude_radials)[0]
```

5. Loop through the rays that aren’t excluded, extracting each ray in turn. Use a moving window to find the first 10 valid values of phidp and use this as the starting point of the ray.

```
for i in az_index:
        data=phidp[i,:]
        [phase1,r1] = identify_first_phase_ray(data, data.mask, 0, 10, 5, len(data),
missing_points=0)
```

6. Adjust ZDR for the bias found from the vertical scans (retrieved in step 2)

```
zdr_f = zdr[i,r1:] - zdr_bias
```

7. Reset phidp ray to start at 0

```
phidp_att = phidp_f - phidp_f[0]
```

8. Apply an attenuation correction (can also be run without a correction)

```
PA = np.maximum.accumulate(phidp_att)
uzh_att = uzh_f + 0.28*PA
zdr_att = zdr_f + 0.04*PA
```

9. Find indices where PhiDP has increased by between 4 and 6 degrees.
   ```
   ind = np.where(np.logical_and(phidp_att > 4, phidp_att < 6))[0]
   ```

10. Find the index, ***ib***, of the maximum value of PhiDP between 4 and 6
    ```
    ib = np.where(phidp_f==max(phidp_f[ind]))[0][0]
    ```

11. Assign the maximum value of PhiDP at the end of the path
    ```
    pdpmax = phidp_att[ib]
    ```

12. Calculate the length of the path and exclude paths less than 15km
    ```
    path_len = rg_f[ib]-rg[r1]
    ```

13. Discard rays with any ZDR > ZDRmax (2dB)
    ```
    ind = zdr_att[phi1:ib+1] > ZDRmax
    ```

14. Discard rays where 3 or more values of RhoHV are less than 0.98
    ```
    ind = np.logical_and(rhohv_f > 0.0, rhohv_f < 0.98)
    ```

If we get as far as this point, then we have a good ray!

We take the observed value of PhiDP at the end of the ray (at ib) and compare it with the calculated value estimated from ZDR and Z. Any offset is an indication of the bias in Z.

phiobs = pdpmax
phiest = f(Z, ZDR)
bias = (phiest - phiobs) / phiobs

SCRIPT AND OUTPUT SUMMARY

STEP 1

- RUN ***process_volume_scans_time_series.py***
  - CALLS ***process_volume_scans_day.py***
    - CALLS ***calibrate_day_att*** in ***calib_functions.py***

Produces the following files:
phiobs_all_att_YYYYMMDD.npy - observed phidp
phiest_all_att_YYYYMMDD.npy - calculated phidp
startphi_all_YYYYMMDD.npy - calculated values for initial phidp at the start of each ray (used to track how it changes over time)

STEP 2

plot_Zcalib.py or plot_Zcalib.ipynb is used to plot the results

Calculating initial phidp of each ray

In the *calibrate_day_att* there is a line that calculates and saves the starting phidp in each of the selected good rays.

The code takes the first 10 values from the starting point of the ray (calculated above) and calculates the median value.

```
startphi[i] = np.nanmedian(uphidp_f[phi1:phi1+10])
```

These values are saved for each file in the same way phiobs and phiest are saved.

A python notebook can be used to plot up the results
plot_initial_phase.ipynb