Comparing Raw Accelerometer data vs GGIR Output

1 Data Processing

- Load raw data built off the pygt3x module from: https://github.com/actigraph/pygt3x
- GGIR output found in /meta/basic/*Rdata
- GGIR output is: downsampled and calibrated.
    - Downsampling: Takes mean of all observations in user defined time window (typically 5 or 10s). Metrics calculated from those mean values.
    - Calibration: Method defined in: https://journals.physiology.org/doi/epdf/10.1152/japplphysiol.00421.2014
      In summary: find non-movement times, normalize total g-force/acceleration to $1g$. Scale and offset in each axis (xyz) computed as: $s_i(t) = a_i s^*(t) + b_i$, where i is the sensor axis, s^* is the measured signal, s is the calibrated signal, and a and b are the computed scaling and offset. Iterative process to minimize error/reach some tolerance between calibrated g force and unit sphere.
- Raw data is processed at each observation (30Hz), no down sampling and no calibration (note that the "calibration" defined in pygt3x is a conversion from the recorded 8-bit data to g force data).
- GGIR output has 'non-linear' time sampling. When sleep/idle mode is detected GGIR process removes those data points. This does not occur with the raw processing.
- Note: GGIR truncates ENMO (any negative value, even after calibration is set to 0)

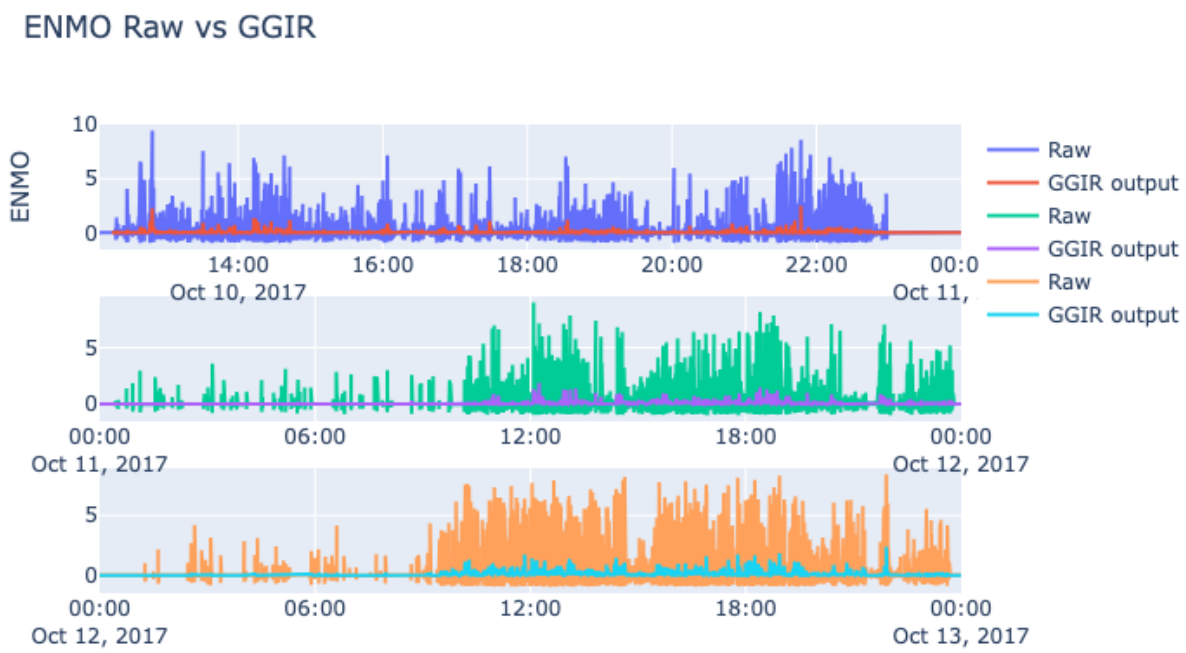
2 Code

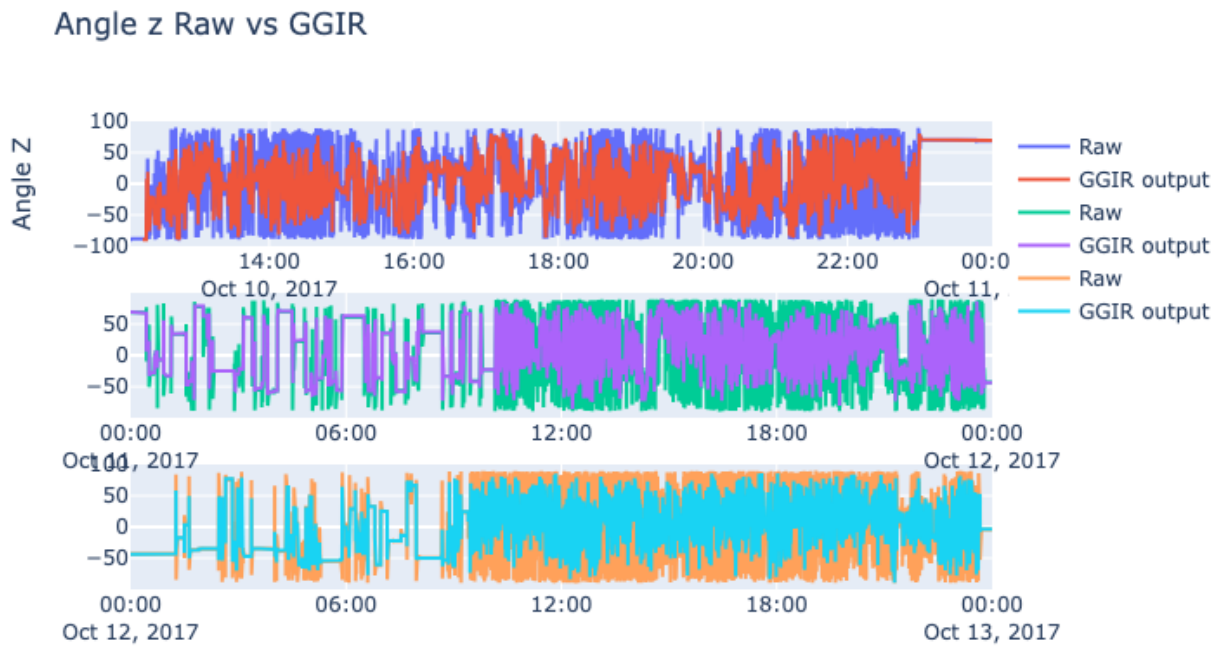
- Currently as Jupyter notebook
- User to define: directory with the .gt3x raw data (one file)
- Pygt3x does not include ms data, manually add ms data to timestamps based on sampling rate.
- Will compute ENMO and angle z on all raw data measurements points.
- Ignores timezone data
- Creates month, day, and day_idx columns for cross-comparison.
- User to define: directory with the *Rdata (should be /output_*/meta/basic/
- Loads ENMO, angle z, and time stamps that is given by GGIR processing
- Removes timezone information
- Create day index relative to Jan 1 as with raw data

3 Output

- User defines the number of days and the starting date (where day 0 is the first day of recording)
- Figures as plotly scatter plots (N rows, where N is the user defined number of days)
- Two figures, ENMO comparison and Angle Z
- Saves as .html
- Shows the two figures.

Example output, as static .png:





4 Concerns, Comments, and Issues

- Raw data is very large (150M observations, in the tested example, other datasets can be larger), this causes sections of the code to run very slowly (10+ minutes for day indexing and plotting) Maybe due to memory swapping?? Code needs to be optimized/improved. Processing Rdata runs much smoother/faster.
- Currently, plots compare a user defined number of days, with a user defined date start. No flag, exception checking is in place if user defines days outside range.
- Plotting and saving plots can also be time consuming. Saved plotly figures as .html for 3 days is ~400MB!
- Same issue for data loading/processing. Assumes user will point to single gt3x file and that *Rdata output is in the usual /meta/basic/. Error/exception checking to be added??
- Day indexing uses date relative to Jan 1. What happens when the data wraps around the calendar year? Data will be discontinuous. How to handle this exception??
- Script performance on various machines/PCs to be tested??
- Implement the same calibration that GGIR uses?
- Standardize each ENMO to [0 1]? (daily standardization? Or across entire sample set?)