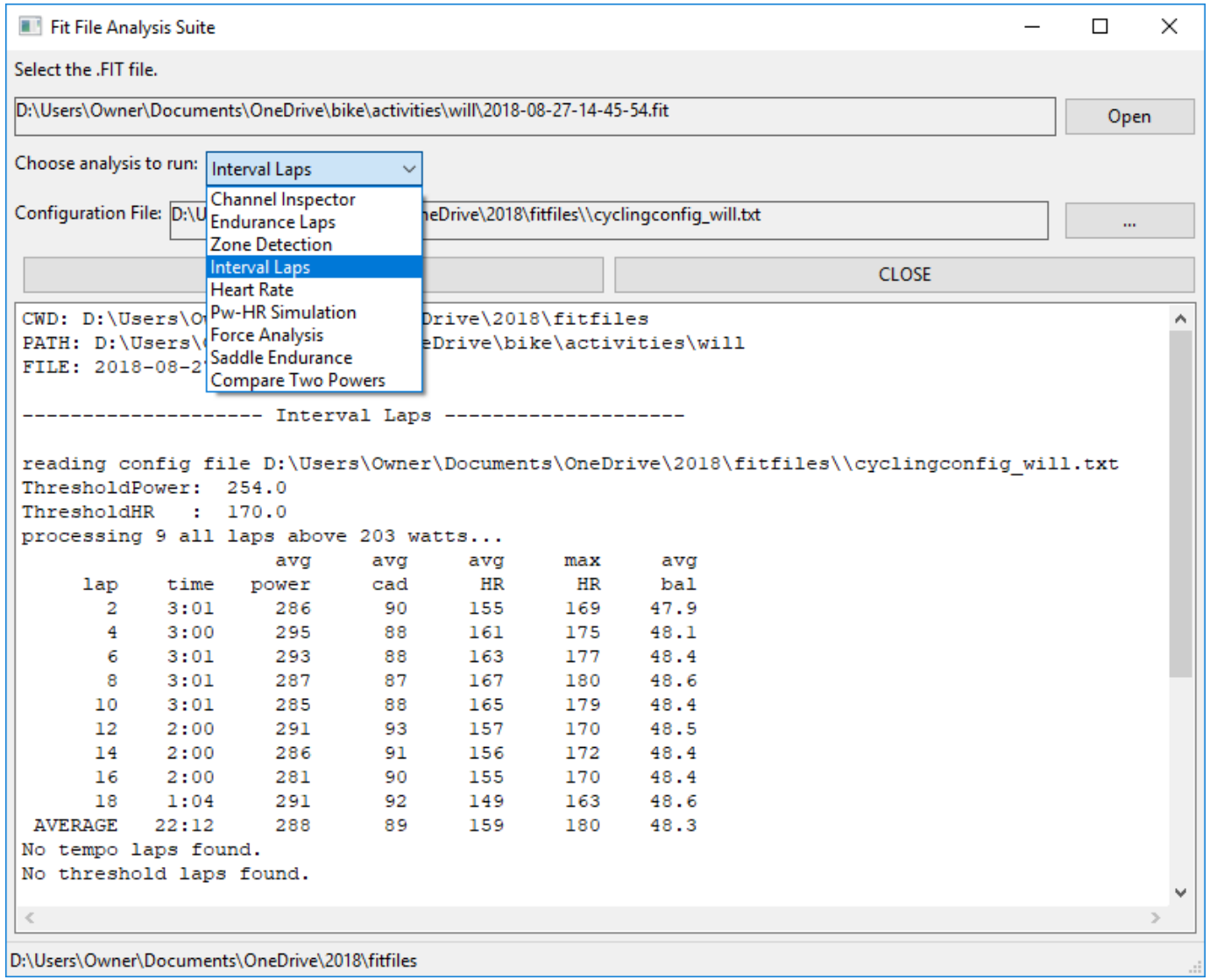
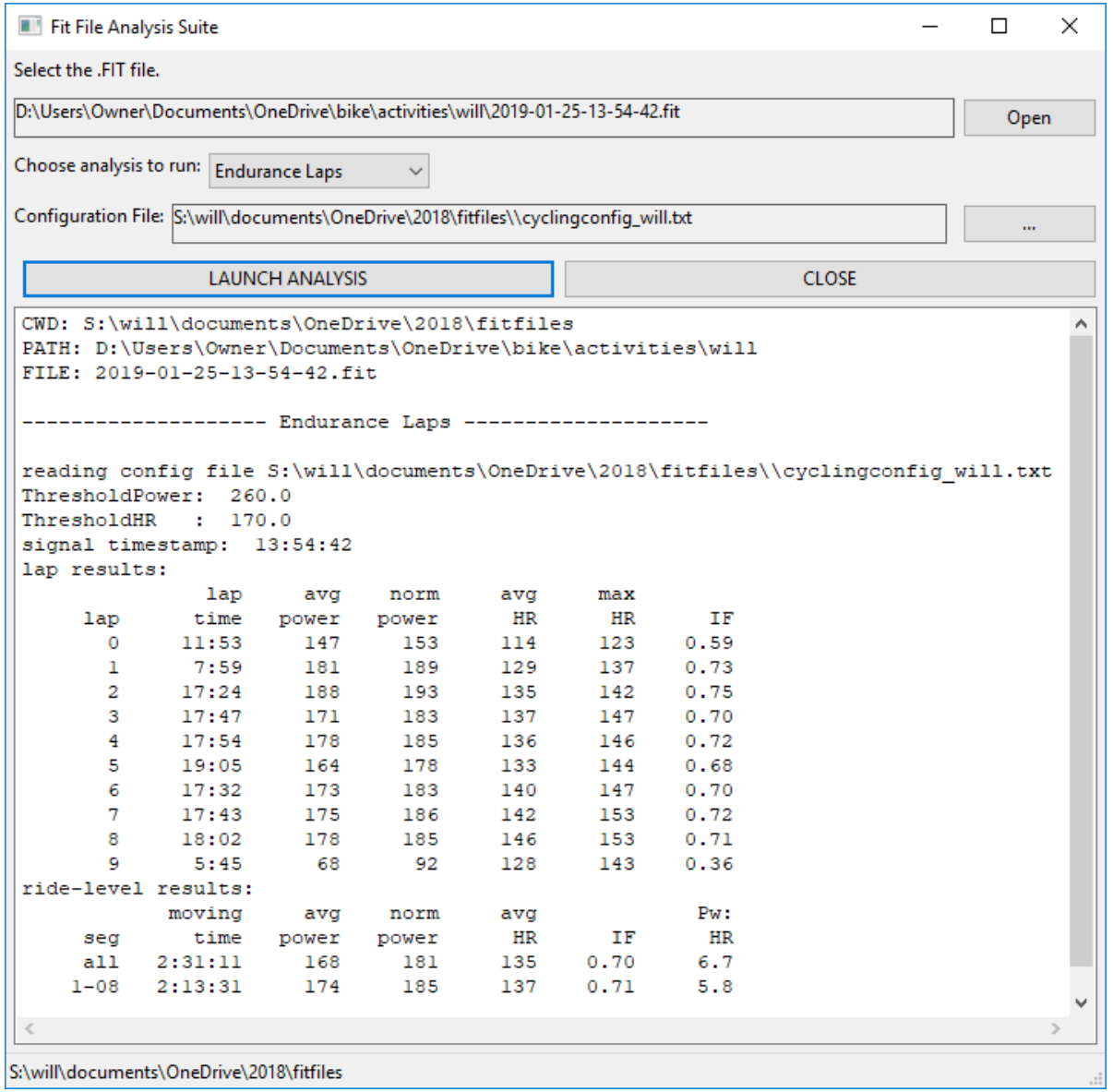
FitFiles Project

- Fast access to a suite of custom tools for analyzing cycling workouts.
- Drag-drop support for .FIT files.
- Formatted text output for copy-pasting into workout diary.
- Support for user-configuration files.
- Available on GitHub (user: guitarsenall)



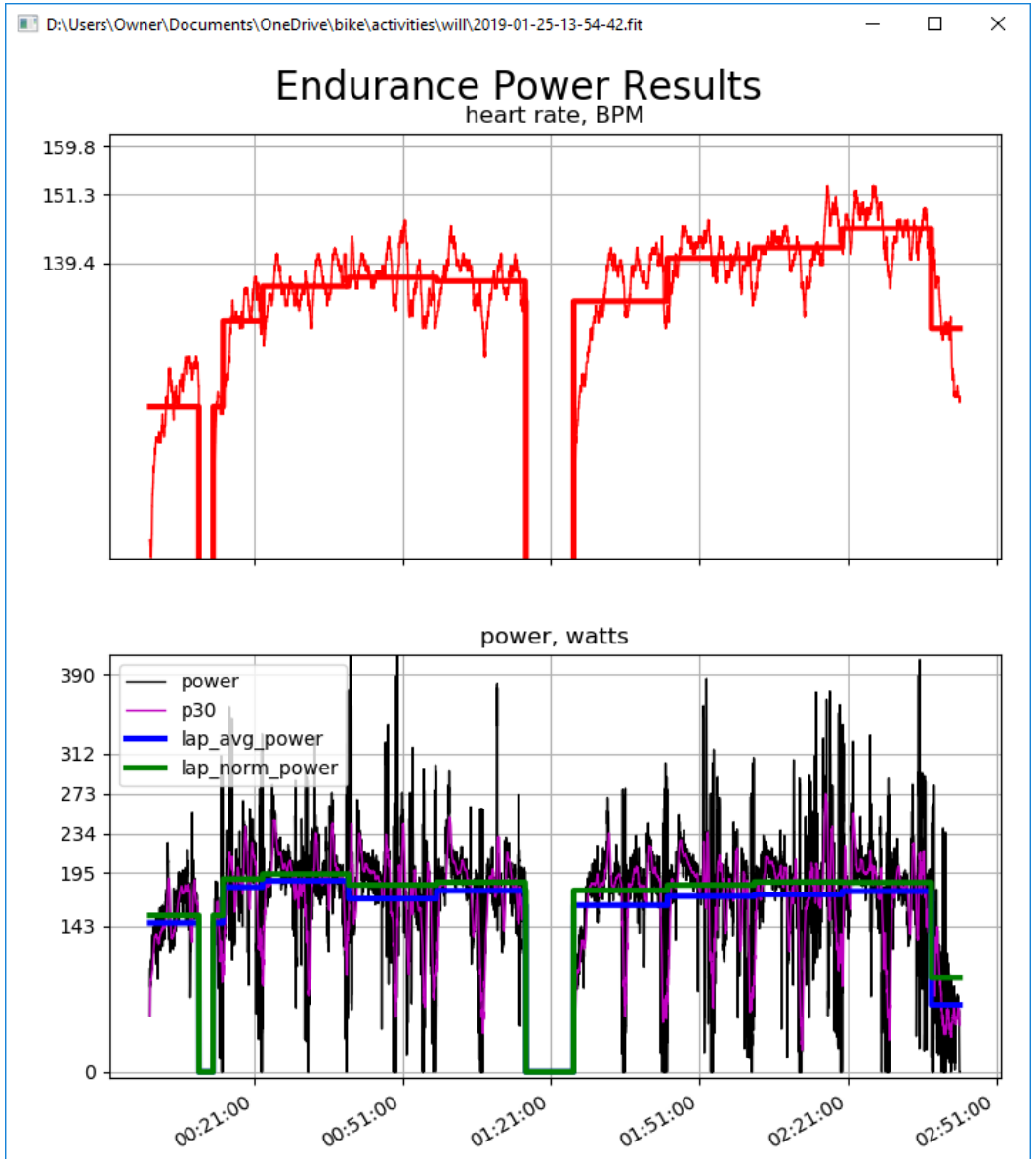
FitFiles: Endurance Summary

- Print lap statistics to show progress.
- Print results for whole ride and inner laps (assume first and last lap are warmup and cooldown).



FitFiles: Endurance Summary

- Plot heart rate with grids at zone boundaries and lap averages.
- Plot raw and 30-second-average power with lap averages and grids at zone boundaries.



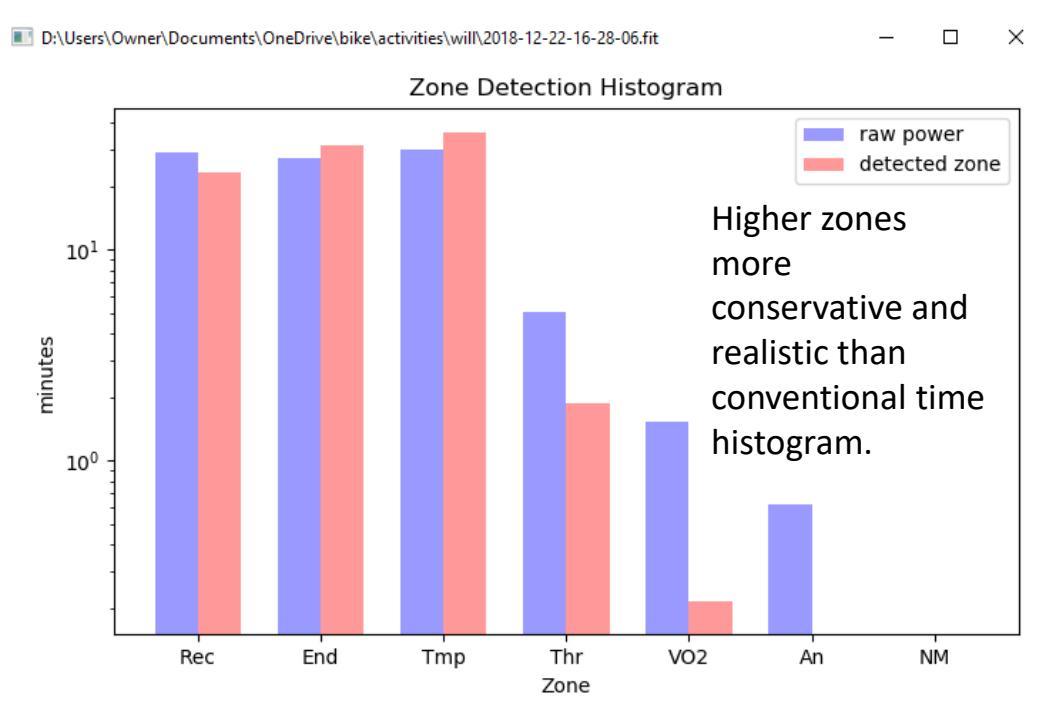
FitFiles: Zone Detection Analysis

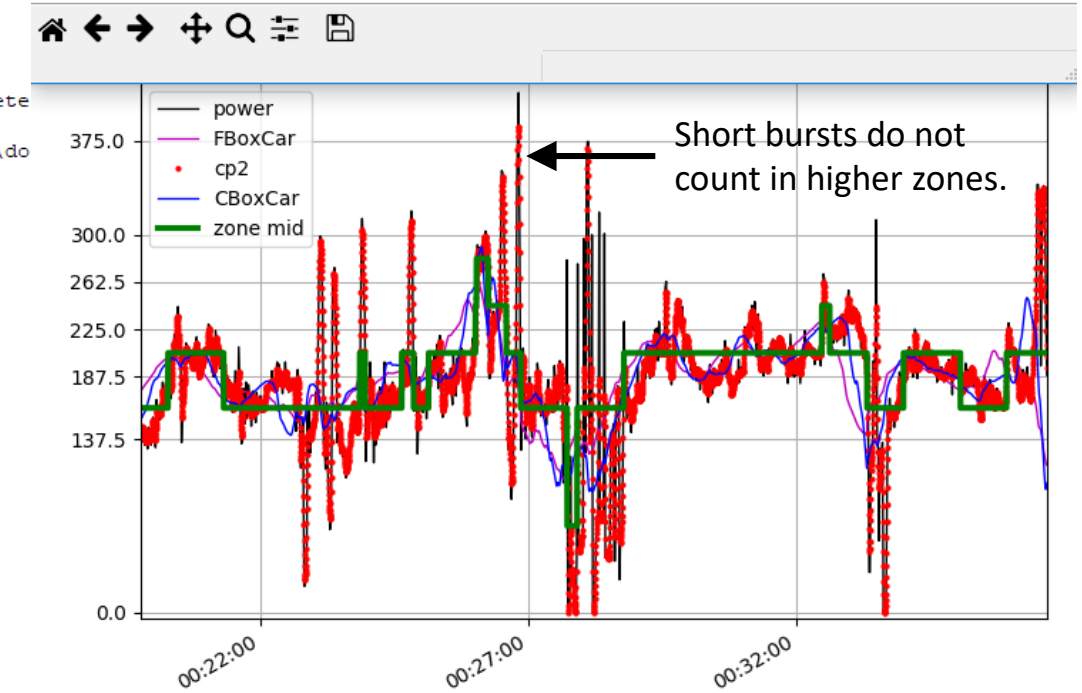
A better way of measuring time spent in power zones.
Innovation: To my knowledge, no one else offers this feature.

- Attaining a power zone should require more than spending one second at that power. Model as a state machine with transitions based on rules regarding time spent in new zone.
- Formatted print of histogram for training diary.

```
-------------------- Zone Dete

reading config file S:\will\do
ThresholdPower:  260.0
ThresholdHR   :  170.0
Power Zone Histogram:
    Zone 1:  0:26:58 (28%)
    Zone 2:  0:39:57 (42%)
    Zone 3:  0:25:53 (27%)
    Zone 4:  0:00:33 ( 0%)
    Zone 5:  0:00:00 ( 0%)
    Zone 6:  0:00:00 ( 0%)
    Zone 7:  0:00:00 ( 0%)
     total:  1:33:23
```

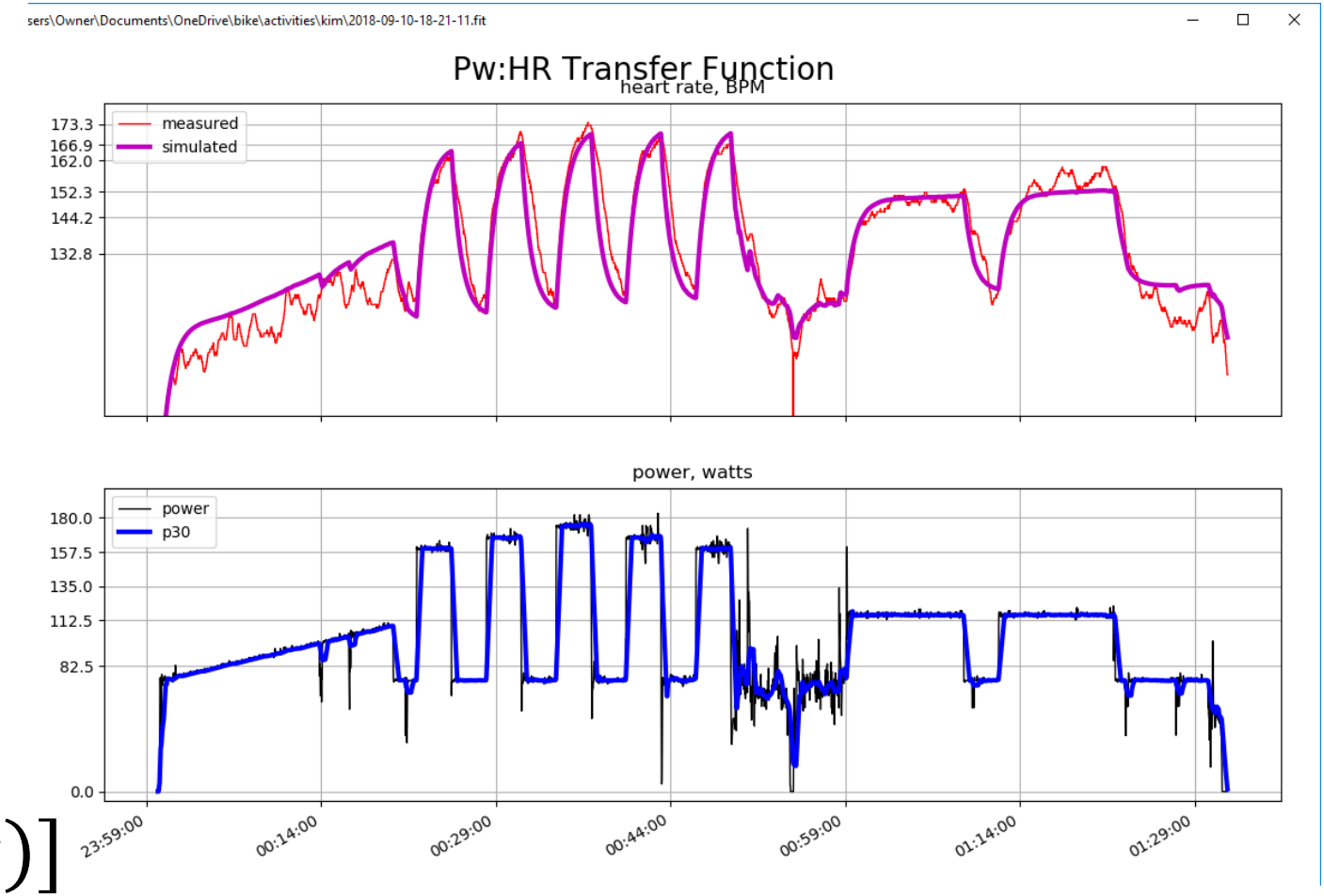




Heartrate Simulation

Predict heartrate from power based on a first-order, linear ODE. Four parameters:

- Time constant: how quickly HR rises in response to power.
- Functional Threshold Power (FTP): Max power sustainable for one hour.
- Functional Threshold Heartrate (FTHR): Average heartrate during FTP.
- Drift rate. Cardiac drift proportional to fatigue (BPM/TSS).



$$\frac{dH}{dt} = \frac{1}{\tau}\left[H_T(p) + H_D F - H(t)\right]$$

- $\frac{dH}{dt}$: Heartrate derivative at t
- τ: Time constant
- $H_T(p)$: Heartrate target: function of power at t
- $H_D F$: Heartrate drift-rate times fatigue (TSS at t)
- $H(t)$: Heartrate at t

Inspired by treadmill-control research:

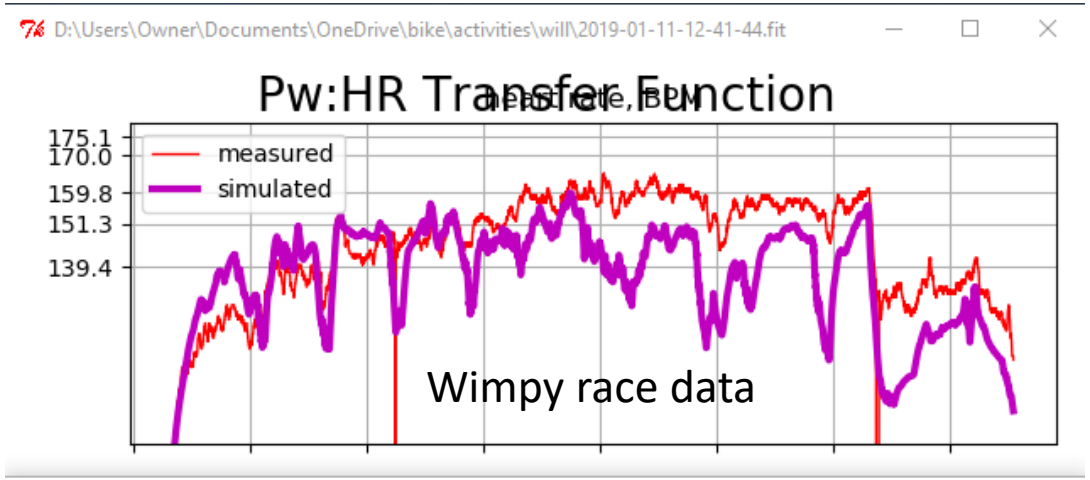
$$H(s) = \frac{k}{\tau s + 1} P(s)$$

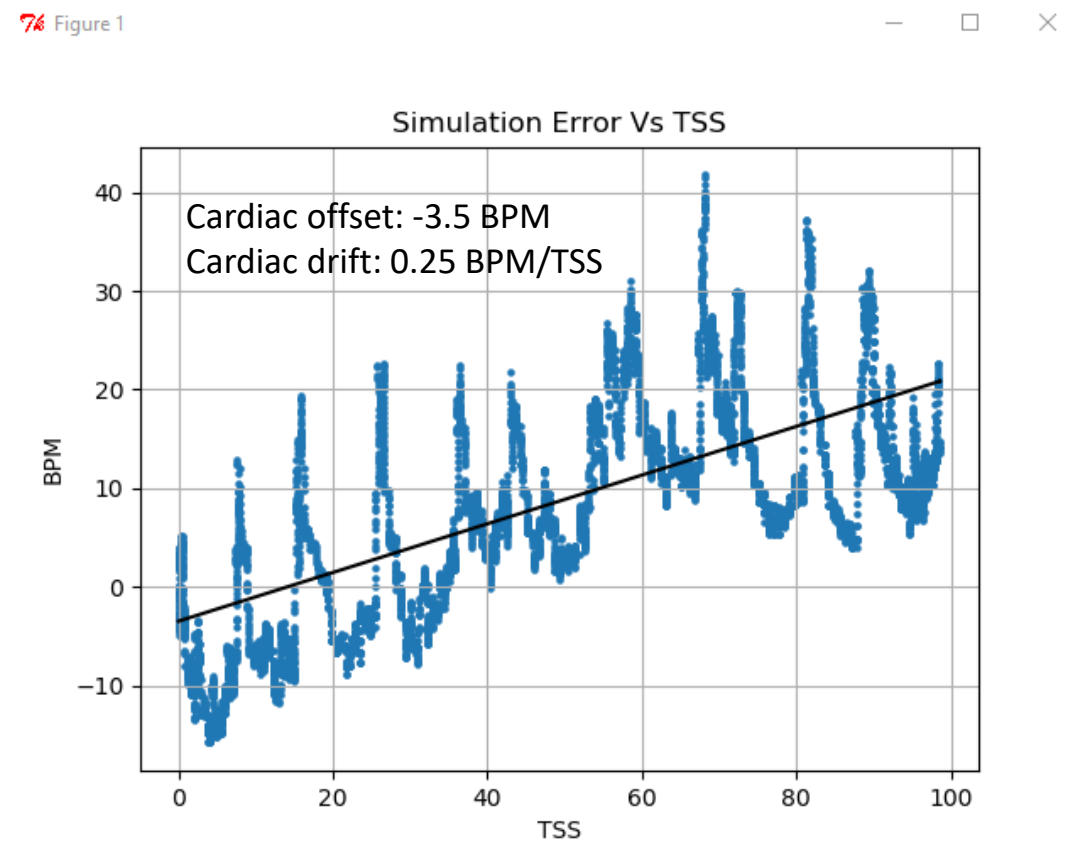
Heartrate Simulation

Provides a better estimate for two useful metrics which used to require a structured workout:

- Threshold Estimate:
  - Provides estimate of Functional Threshold Heart Rate (FTHR) for ANY workout (e.g., racing); structured endurance workout not necessary.
  - If FTHR is well known, it answers the question, “how much higher or lower than normal is my heartrate for this workout?”
- Cardiac drift:
  - Now modeled as proportional to fatigue (BPM/TSS) rather than a percentage over duration.
  - Available for ANY workout (e.g., racing); structured endurance workout not necessary.

Innovation: To my knowledge, no one else offers this feature.



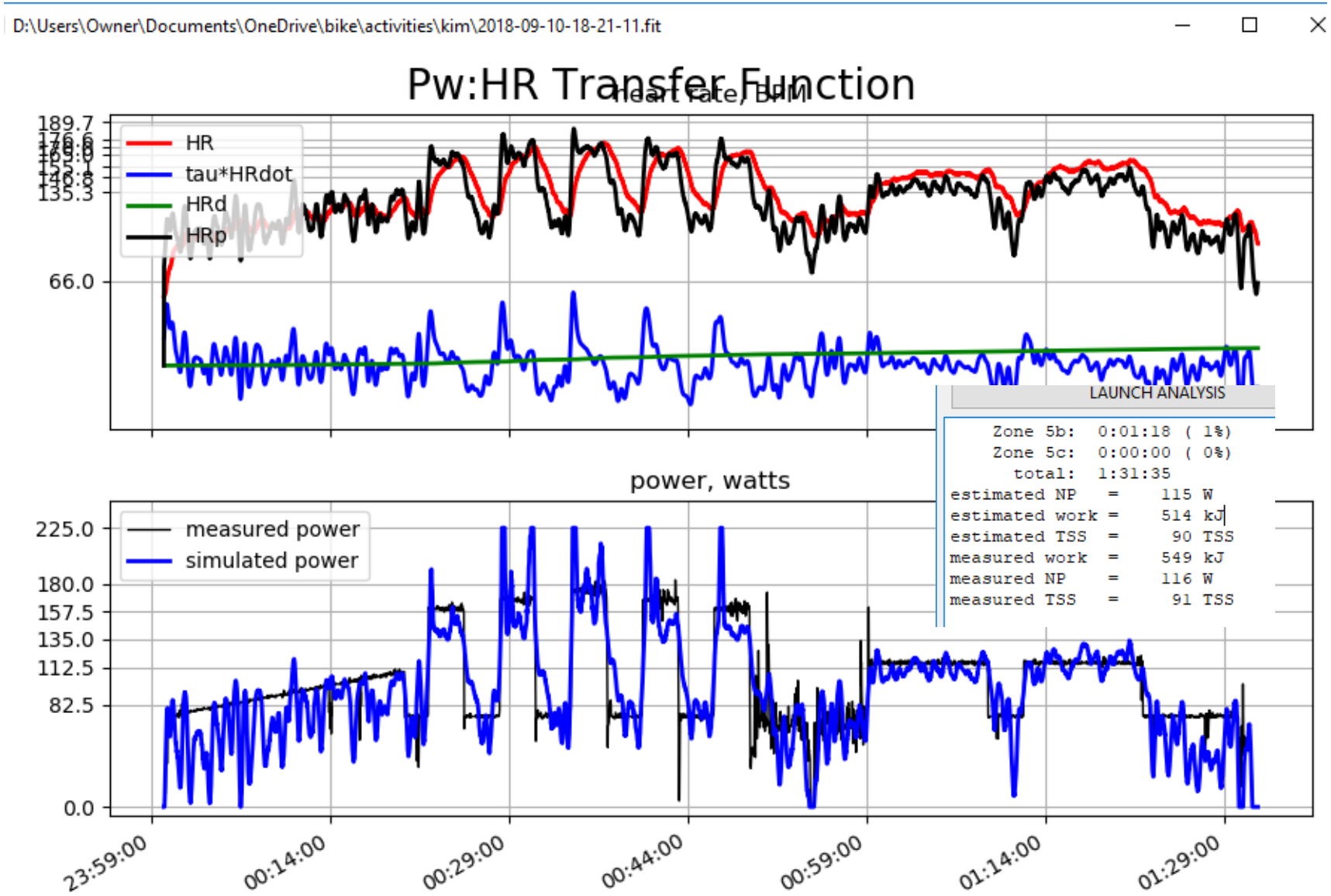


Heartrate Simulation

Also provides for going the other way—simulating power from heart rate!

- Estimate work, intensity, and TSS for workouts that lack a power signal.

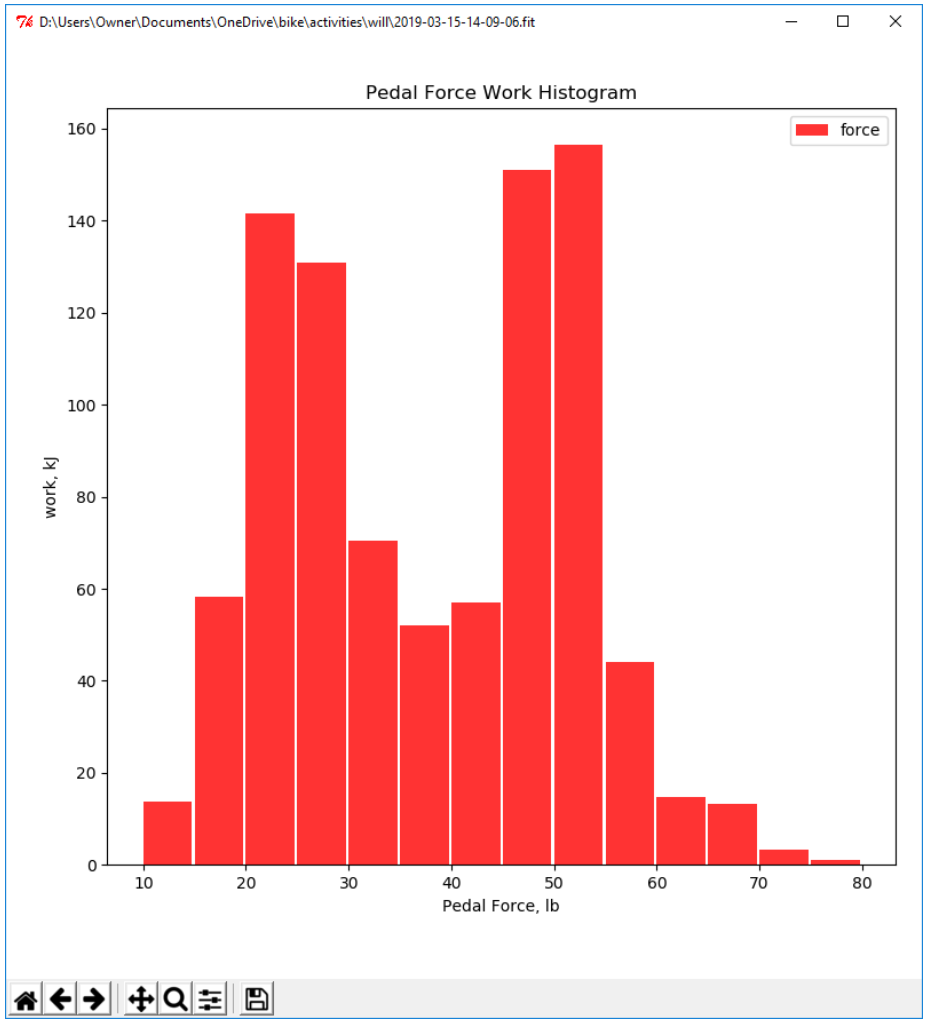
Innovation: To my knowledge, no one else offers this feature.



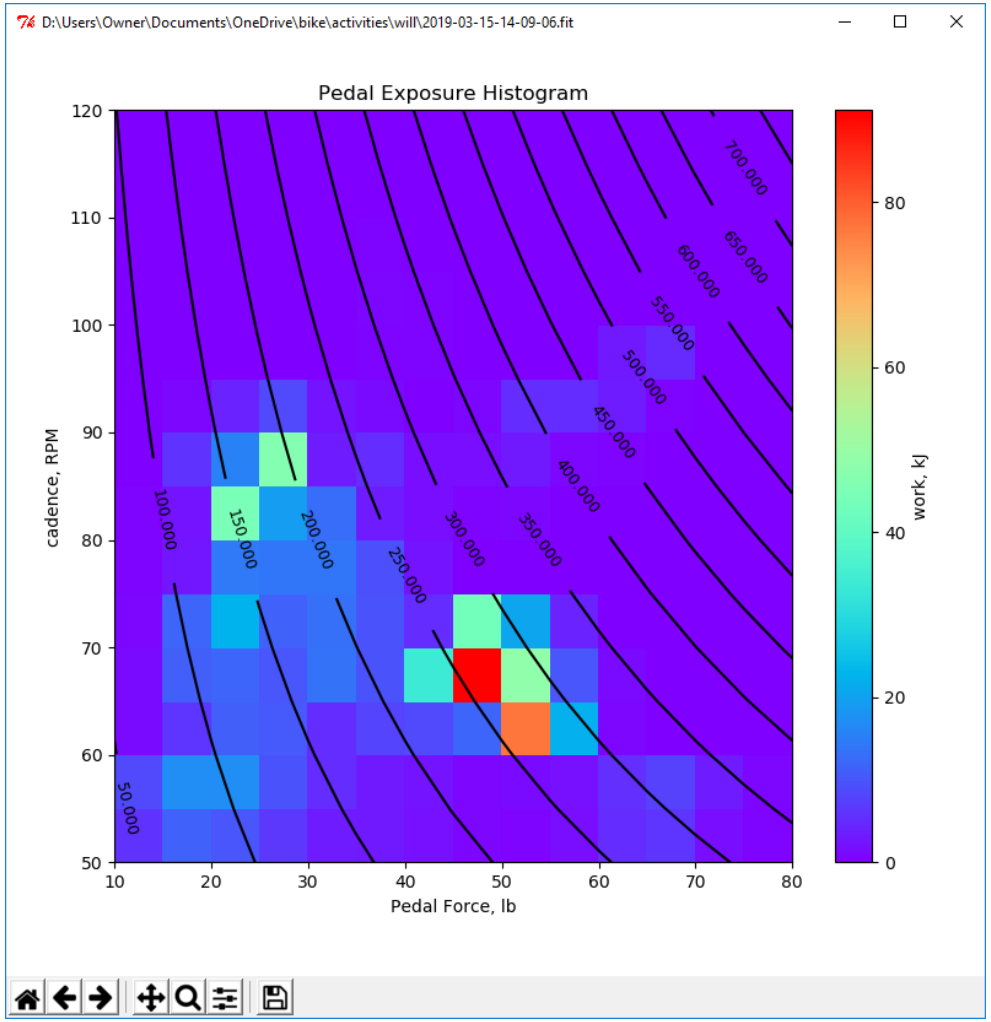
Force Analysis

Useful for analyzing force workouts (e.g. hill repeats) and fast interval sessions to verify targets are met.

One-way histogram shows work (kJ) done at various levels of pedal force.

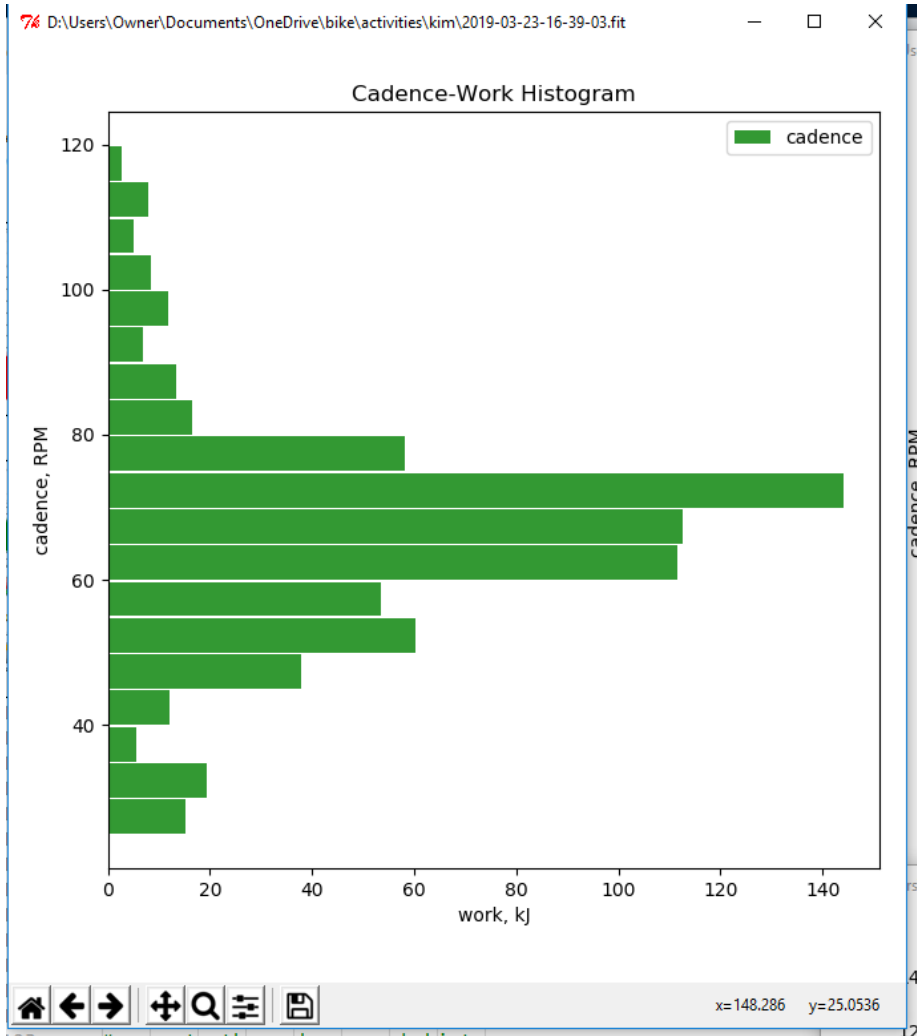


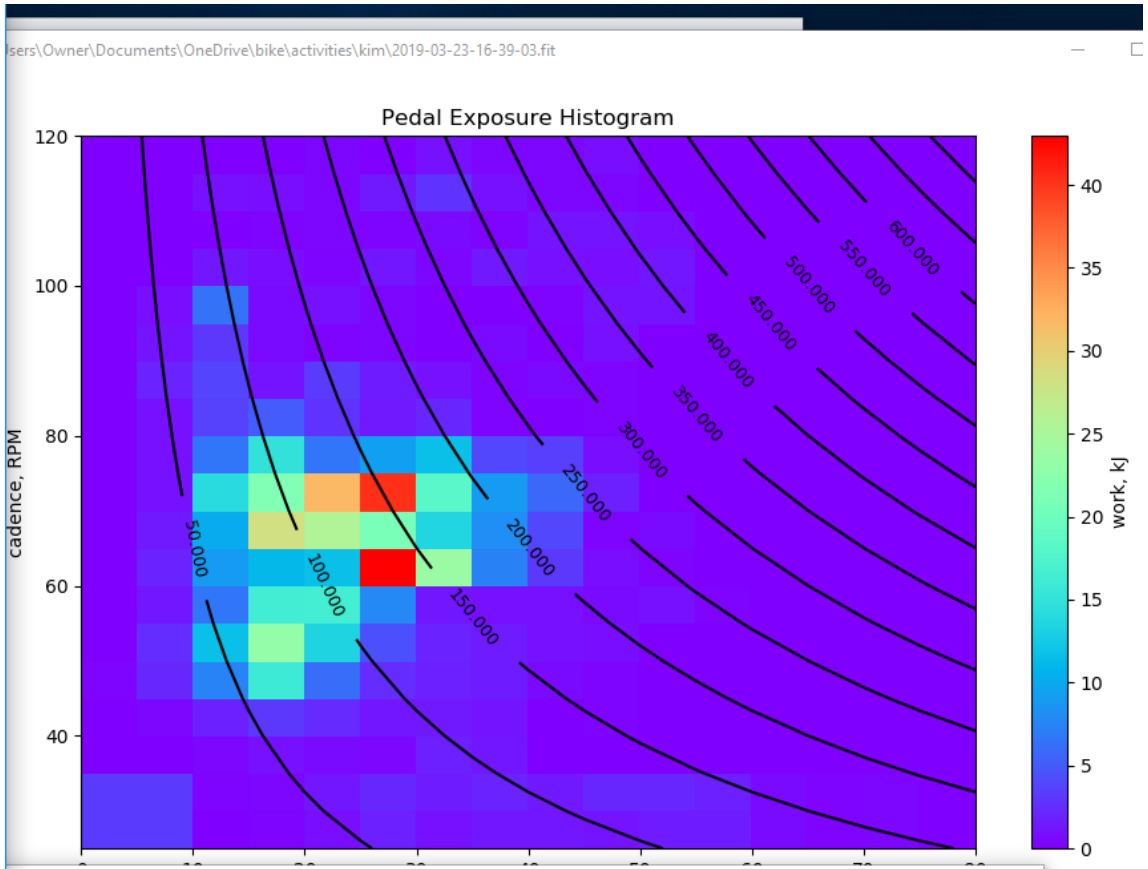
Two-way histogram includes cadence with contour lines indicating power.



Force Analysis

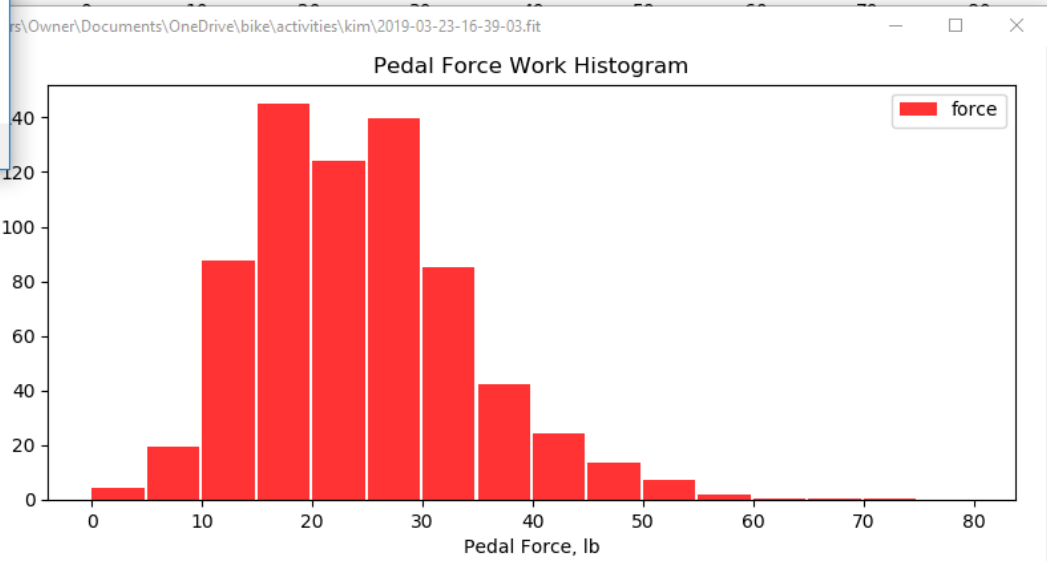
A horizontal cadence-work histogram captures work done at various cadences, just as the force-work histogram captures work done as a function of pedal force.





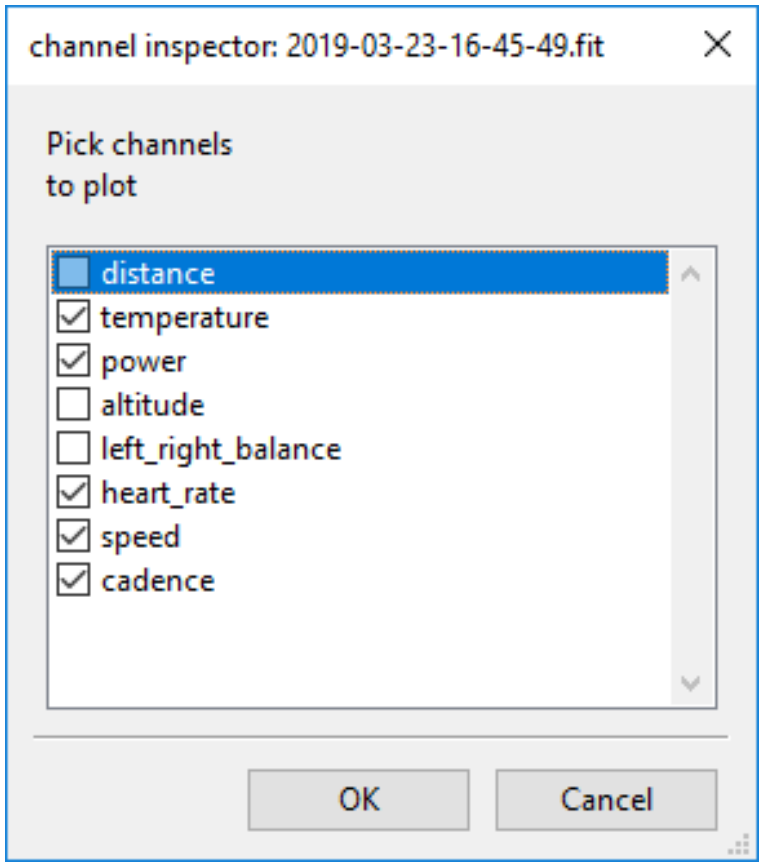

```
# compute the cadence-work histogram
nCBins          = len(cadence_bins)-1
cadence_width   = (cadence_bins[1:nCBins+1] - cadence_bins[0
cadence_work    = np.zeros(nCBins)
for i in range(nCBins):
    CBinLo  = cadence_bins[i] if i>1 else leg_force.min()
    CBinHi  = cadence_bins[i+1] if i<nCBins else cadence.max
    ii = np.nonzero( np.logical_and(
                cadence >= CBinLo,
                cadence <  CBinHi ) )[0]
    cadence_work[i] = sum(power[ii]) / SampleRate   \
                    / 1000.0                        # J -> k
```

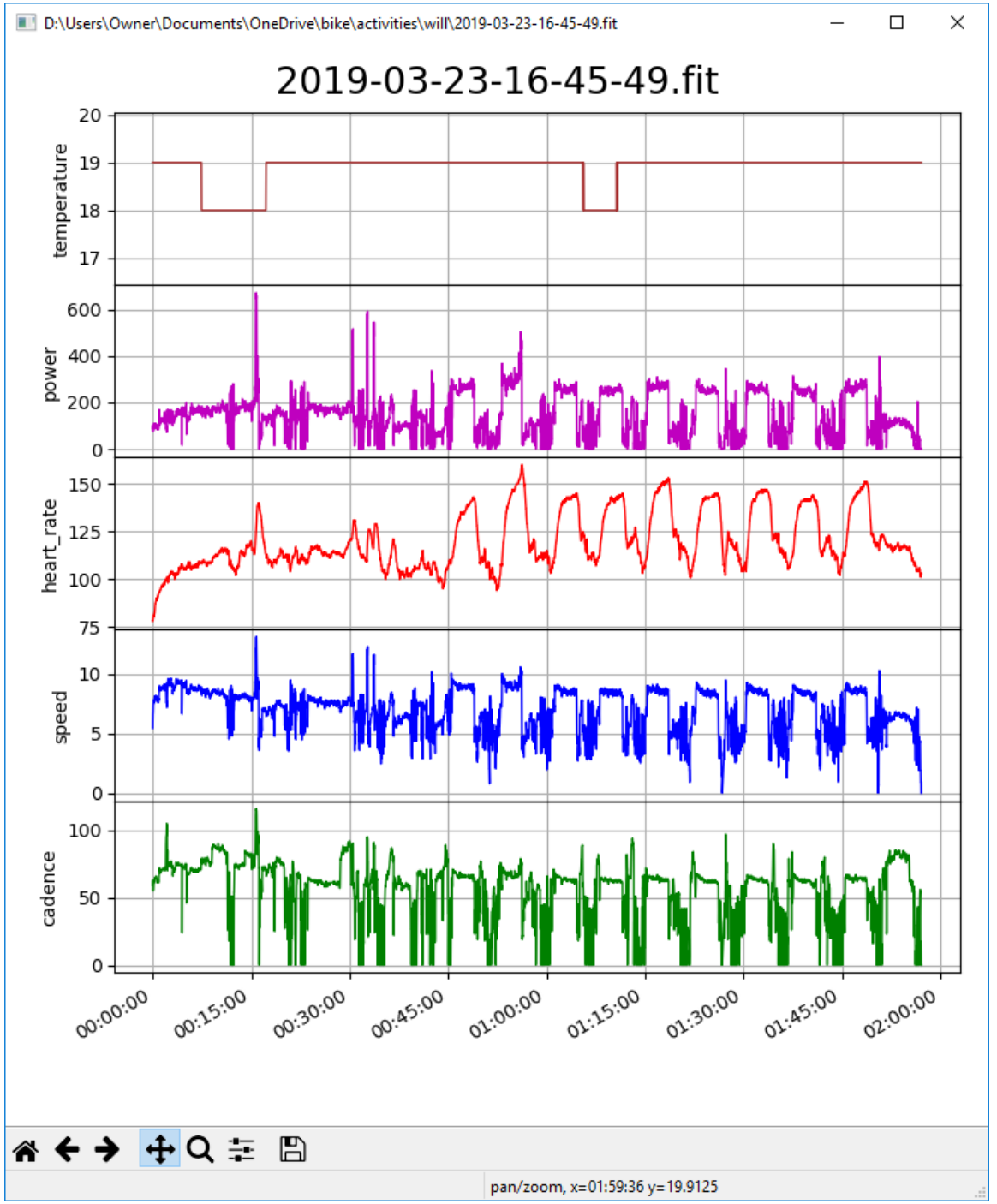
analysis_notes.txt | force_analysis.py



Channel Inspector

Nothing revolutionary here—just quick access to see what a .FIT file contains and quickly plot channels of interest.





Saddle Endurance

A sophisticated state-machine algorithm senses when I am seated or standing (to relieve saddle fatigue). I generally need to stand every 10 minutes and do so by pedaling three strokes, resting a few seconds, pedaling another three strokes, and so on. But I am trying to improve my saddle endurance, so I want to measure it.

Choose analysis to run: Saddle Endurance

Configuration File: D:\Users\Owner\Documents\OneDrive\2018\fitfile

LAUNCH ANALYSIS

```
-------------------- Saddle Endurance --------

total time      :  2:05:02
seated time     :  1:44:15 (83%))
standing time   :  0:20:47 (16%))
standing segments:
     segment      start       stop  duration
           0    0:00:00    0:02:20      2:20
           1    0:02:28    0:09:18      6:50
           2    0:10:18    0:12:46      2:28
           3    0:13:04    0:22:03      8:59
           4    0:22:07    0:23:32      1:25
           5    0:24:18    0:34:50     10:32
           6    0:35:26    0:45:21      9:55
           7    0:46:22    0:54:09      7:47
           8    0:54:27    0:59:23      4:56
           9    1:04:54    1:05:03      0:09
          10    1:05:09    1:09:21      4:12
          11    1:09:58    1:14:46      4:48
          12    1:15:27    1:19:44      4:17
          13    1:20:08    1:22:51      2:43
          14    1:23:16    1:23:28      0:12
          15    1:23:33    1:30:30      6:57
          16    1:31:14    1:33:14      2:00
          17    1:33:41    1:35:40      1:59
          18    1:35:44    1:36:01      0:17
          19    1:36:07    1:38:22      2:15
          20    1:38:54    1:39:28      0:34
          21    1:39:48    1:46:33      6:45
          22    1:47:19    1:49:32      2:13
          23    1:49:57    1:54:43      4:46
          24    1:56:41    1:57:28      0:47
          25    1:57:32    2:00:41      3:09
          26    2:02:19    2:02:41      0:22
          27    2:04:09    2:04:19      0:10
          28    2:04:29    2:04:56      0:27
best-hour segments:
     segment      start       stop  duration
           5    0:24:18    0:34:50     10:32
           6    0:35:26    0:45:21      9:55
           3    0:13:04    0:22:03      8:59
           7    0:46:22    0:54:09      7:47
          15    1:23:33    1:30:30      6:57
           1    0:02:28    0:09:18      6:50
          21    1:39:48    1:46:33      6:45
           8    0:54:27    0:59:23      4:56
        BEST ONE-HOUR AVERAGE:          7:50
```