

split_files.py

Input Files

main_path = r"/Users/<username>/Desktop/Flight_scripts/"

ID	died?	chamber	set_number
9		A-1	6
218		A-2	6
419		A-3	6
262		A-4	6
250		B-1	6
72		B-2	6
2		B-3	6

data/datasheet.csv

TBF voltage reading event_happened,event_num date, time event_marker

```

"Samples per sec. = 400/1"
.000, 4.90234E+00, 4.93774E+00, 4.89136E+00, 4.94385E+00, 1,1,02-24-20,08:40:36
.010, 4.90234E+00, 4.93774E+00, 4.89136E+00, 4.94385E+00, 0,0,02-24-20,08:40:36
.020, 4.90234E+00, 4.93774E+00, 4.89136E+00, 4.94385E+00, 0,0,02-24-20,08:40:36
.030, 4.90234E+00, 4.93774E+00, 4.89136E+00, 4.94385E+00, 0,0,02-24-20,08:40:36
...
2301.250, 4.84497E+00, 4.93774E+00, 4.88037E+00, 4.94263E+00, 0,0,02-24-20,09:18:57
2301.260, 4.84497E+00, 4.93774E+00, 4.88037E+00, 4.94263E+00, 0,0,02-24-20,09:18:57
2301.270, 4.84497E+00, 4.93774E+00, 4.88037E+00, 4.94263E+00, 0,0,02-24-20,09:18:57
2301.280, 4.84497E+00, 4.93774E+00, 4.88037E+00, 4.94263E+00, 2,2,02-24-20,09:18:57,"158 start"
2301.290, 4.84497E+00, 4.93774E+00, 4.88037E+00, 4.94263E+00, 0,0,02-24-20,09:18:57
    
```

recordings/*.TXT

Process

functions

Output Files

complex data structure

define_dicts

Creates 2 dictionaries.

```

first_flight_dict = { (6, A, 1): 9,
                      (6, A, 2): 218,
                      (6, A, 3): 419,
                      (6, A, 4): 262,
                      (6, B, 1): 250,
                      (6, B, 2): 72,
                      (6, B, 3): 2
                    }

current_flight_dict = { (6, A, 9): 1,
                       (6, A, 218): 2,
                       (6, A, 419): 3,
                       (6, A, 262): 4,
                       (6, B, 250): 1,
                       (6, B, 72): 2,
                       (6, B, 2): 3
                     }
    
```

map_IDs

Uses dictionaries to map IDs from .TXT file event marker comments.

```

.000, 4.90234E+00, 4.93774E+00, 4.89136E+00, 4.94385E+00, 0,2020-02-24 08:40:36 250,72,2,403
.010, 4.90234E+00, 4.93774E+00, 4.89136E+00, 4.94385E+00, 0,2020-02-24 08:40:36 250,72,2,403
.020, 4.90234E+00, 4.93774E+00, 4.89136E+00, 4.94385E+00, 0,2020-02-24 08:40:36 250,72,2,403
.030, 4.90234E+00, 4.93774E+00, 4.89136E+00, 4.94385E+00, 0,2020-02-24 08:40:36 250,72,2,403
.040, 4.90234E+00, 4.93774E+00, 4.89136E+00, 4.94385E+00, 0,2020-02-24 08:40:36 250,72,2,403
    
```

Generates an intermediate file in the files2split folder that maps (by ID) when bugs come in and out of the flight mill for all 4 chambers.

split_files

Generates .TXT files that contain the TBF, voltage readings, and datetimes specific to each bug tested.

```

T1_set006-2-24-2020-B1_34.txt
T1_set006-2-24-2020-B1_47.txt
T1_set006-2-24-2020-B1_72.txt
T1_set006-2-24-2020-B1_124.txt
T1_set006-2-24-2020-B1_146.txt
T1_set006-2-24-2020-B1_158.txt
T1_set006-2-24-2020-B1_187.txt
T1_set006-2-24-2020-B1_205.txt
T1_set006-2-24-2020-B1_217.txt
    
```

ID is the key to ID_data

```

ID_data = { 34: [(TBF: 17927.440, voltage: 4.85962E+00, datetime: 2020-02-24 13:39:23),
                 (TBF: 17927.450, voltage: 4.85962E+00, datetime: 2020-02-24 13:39:23),
                 (TBF: 17927.460, voltage: 4.85962E+00, datetime: 2020-02-24 13:39:23),
                 ...]
            }
    
```

A list of dictionaries are the values to ID_data. Each dictionary becomes a row in the final .TXT file.

standardize_troughs.py

Input Files

main_path = r"/Users/<username>/Desktop/Flight_scripts/"

```

TBF voltage datetime
17927.440, 4.85962E+00, 2020-02-24 13:39:23
17927.450, 4.85962E+00, 2020-02-24 13:39:23
17927.460, 4.85962E+00, 2020-02-24 13:39:23
17927.470, 4.85962E+00, 2020-02-24 13:39:23
17927.480, 4.85962E+00, 2020-02-24 13:39:23
    
```

split_files/*.TXT

Process

functions

Output Files

trough_standardize

Identifies large dips in voltage as troughs.

Standardization interval equations and examples:

```

voltage_min = volt_mean - dev_min
voltage_max = volt_mean + dev_max
x = (voltage[TBF] - volt_min) / (volt_max - volt_min)
    
```

Not a trough:
4.78 = 4.79 - 0.01
4.80 = 4.79 + 0.01
-2 = (4.74 - 4.78) / (4.80 - 4.78)

Trough (x < -2):
-202 = (2.74 - 4.78) / (4.80 - 4.78)



Representative standardization interval - anything that falls outside it is identified as a trough.

map_diagnostics

Generates a heat map (.PNG) to diagnose the robustness of the recording file.

A larger standardization interval is more robust and will clearly identify large troughs. A smaller one is more sensitive and may let noise through.



write_to_file

Generates standardized .TXT files that contain the TBF and location of troughs (trough = 1).

```

27273.20, 0.00
27273.21, 0.00
27273.22, 1.00
27273.23, 0.00
27273.24, 0.00
    
```

Trough at TBF 27273.22 seconds.

flight_analysis.py

Input Files

main_path = r"/Users/<username>/Desktop/Flight_scripts/"

```

TBF troughs (0 = no trough; 1 = trough)
27273.20, 0.00
27273.21, 0.00
27273.22, 1.00
27273.23, 0.00
27273.24, 0.00
    
```

standardized_files/*.TXT

Process

functions

Output Files

time_list

Filters out only for time values where a trough occurs.

[2119.48, 2122.85, 2125.58, 2127.96, 2130.17, 2132.32, 2134.44, ...]

speed_list

Corrects for false troughs and calculates the speed between troughs.

Speed calculation depends on the circular flight path.

[0, 1.1186350148368316, 1.3808791208791118, 1.583949798318601, ...]

distance

Corrects for false troughs and calculates the distance flown and average speed.

recording_duration

Calculates the duration of the bug spent flying.

flying_bouts

Calculates bout durations and % spent flying.

graph

Cleans up speed and time columns for easy graphing.

```

ID: 318
CHAMBER A-2
Average speed (m/s) -> 1.34
Total flight time (s) -> 879.86
Distance (m) -> 610.71
Shortest flying bout (s) -> 60.28
Longest flying bout (s) -> 255.23
This individual spent 0.140 of its time flying with this composition:
600s-800s = 0.467 with 0 events
800s-900s = 0.000 with 0 events
900s-1000s = 0.000 with 0 events
1000s-1100s = 0.000 with 0 events
1100s-1200s = 0.000 with 0 events
1200s-1300s = 0.000 with 0 events
1300s-1400s = 0.000 with 0 events
1400s-1500s = 0.000 with 0 events
    
```

Between bouts, new time and speed data points are added to represent that the insect stopped flying (e.g. 2295.33 s, 0 m/s):

[2122.85, 2125.58, 2126.58, 2294.33, 2295.33, ...]

[1.1186350148368316, 1.3808791208791118, 0, 0, ...]

Generates a flight_stats_summary.csv with multiple flight statistics:

ID	filename	trial_type	chamber	channel_letter	channel_num	set_number	average_speed	total_flight_time	distance	shortest_flying_bout	longest_flying_bout	portion_flying	recording_duration	max_speed	
318	set001-2-17-2020-A2_318.txt	T1	A-2	A		2	1	1.34	610.71	60.28	255.23	0.14	6300.43	1.4	
371	set004-2-20-2020-A4_371.txt	T1	A-4	A		4	4	1.24	1834.37	2129.94	91.04	1323.21	0.74	2402.19	1.4
176	set004-2-20-2020-B3_176.txt	T1	B-3	B		3	4	1.17	32.04	22.62	0	254.14	0.01	2347.31	1.3