

# analyze\_single\_anemotaxis

September 26, 2025

## 0.1 Analyze single `trx.mat` file to generate analyzed HDF5 file required for plotting

```
[1]: import sys
import os
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.gridspec import GridSpec

# Set matplotlib style
%matplotlib widget
%load_ext autoreload
%autoreload 2
plt.style.use('..../anemotaxis.mplstyle')
```

```
[2]: import core.data_loader as data_loader
import core.data_processor as data_processor
import viz.plot_data as plot_data
import utils.preprocessing as preprocessing
```

```
[3]: single_path = "/Users/sharbat/Projects/anemotaxis/data/
˓→FCF_attP2-40@UAS_TNT_2_0003/p_5gradient2_2s1x600s0s#n#n#n/20240223_112627/
˓→trx.mat"
trx_data = data_loader.load_single_trx_file(single_path, show_progress=True)
```

Processing file: /Users/sharbat/Projects/anemotaxis/data/FCF\_attP2-  
40@UAS\_TNT\_2\_0003/p\_5gradient2\_2s1x600s0s#n#n#n/20240223\_112627/trx.mat  
Number of larvae: 41

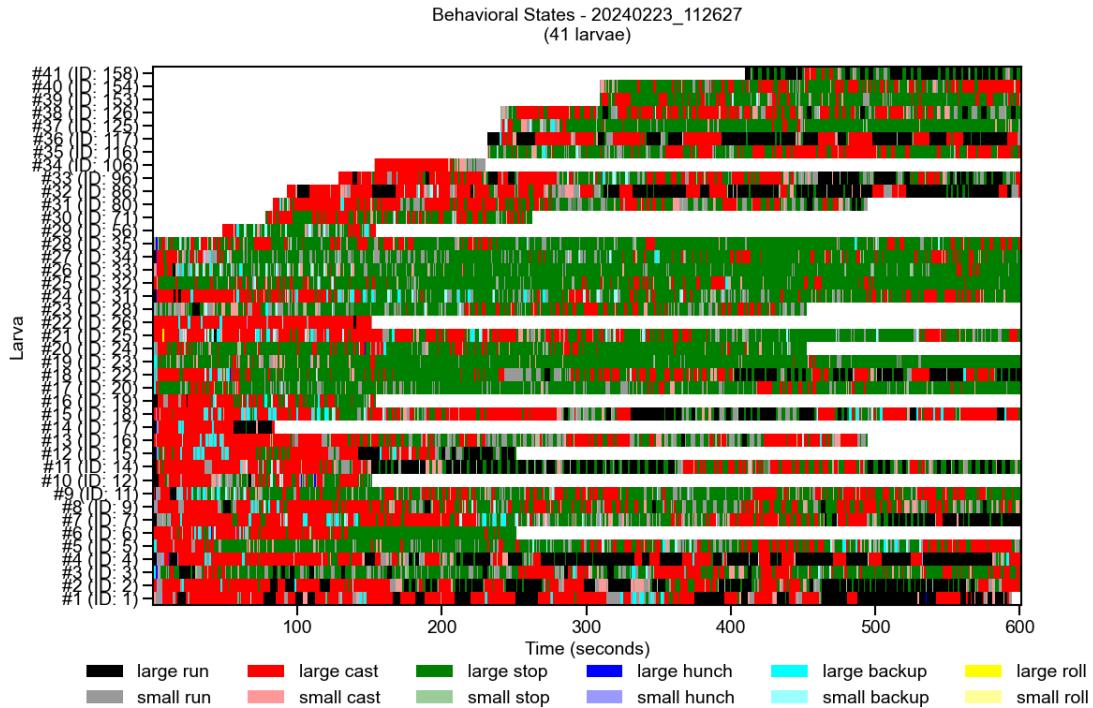
Processing larvae: 0% | 0/41 [00:00<?, ?it/s]

## 0.2 Ethogram (unfiltered)

1. Run - black
2. Cast - Red
3. Stop - Green
4. Hunch - Blue
5. Backup - Cyan

## 6. Roll - Yellow

```
[4]: results_behavior_matrix = plot_data.plot_global_behavior_matrix(trx_data)
```

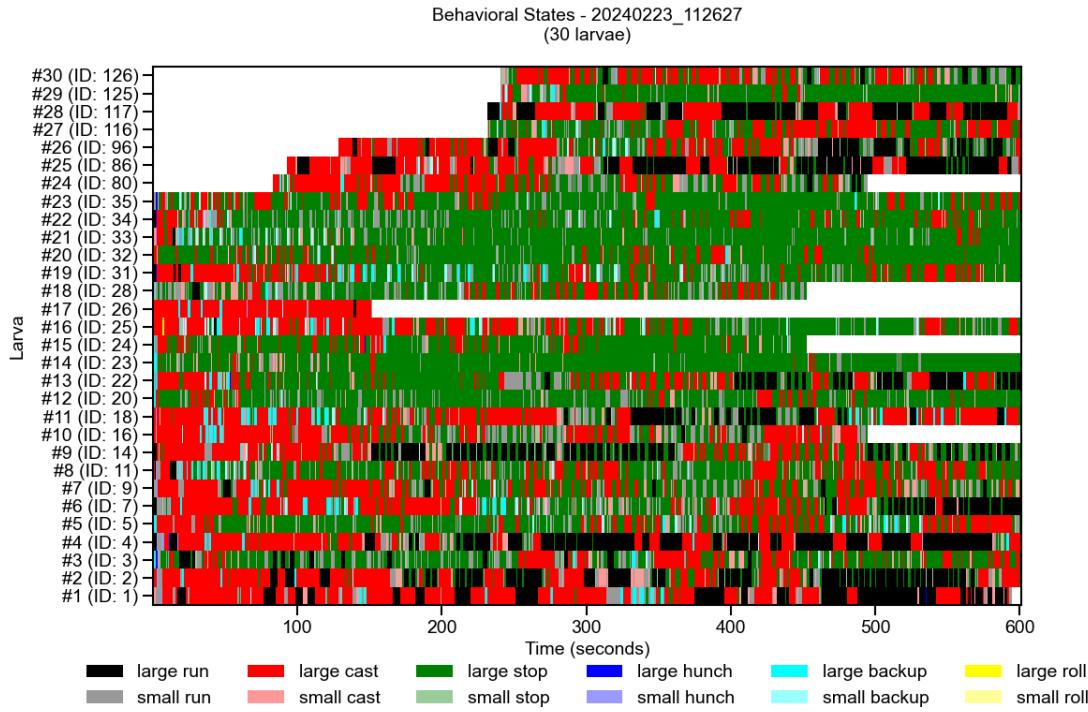


### 0.2.1 Filtering and preprocessing

```
[5]: trx_filtered_by_duration = preprocessing.filter_larvae_by_duration(trx_data,  
                     ↪min_total_duration=300)  
results_filtered_behavior_matrix = plot_data.  
                     ↪plot_global_behavior_matrix(trx_filtered_by_duration)
```

Duration filtering results (threshold: 300.0s):

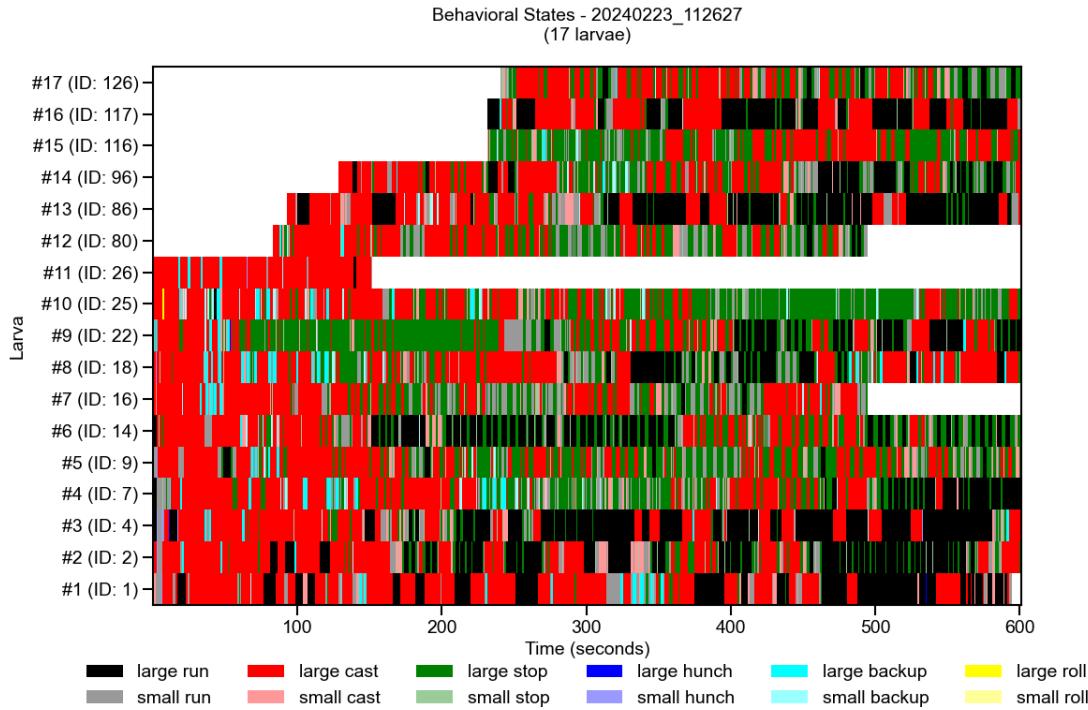
- Removed 11 larvae with <300.0s total duration
- 30 larvae remaining



```
[6]: trx_filtered_by_removing_stops = preprocessing.
    ↪filter_larvae_by_excess_stop_time(trx_filtered_by_duration, ↪
    ↪max_stop_percentage=0.5)
results_filtered_behavior_matrix = plot_data.
    ↪plot_global_behavior_matrix(trx_filtered_by_removing_stops)
```

Excess stop time filtering results (threshold: 50%):

- Removed 13 larvae with >50% time in stop state
- 17 larvae remaining

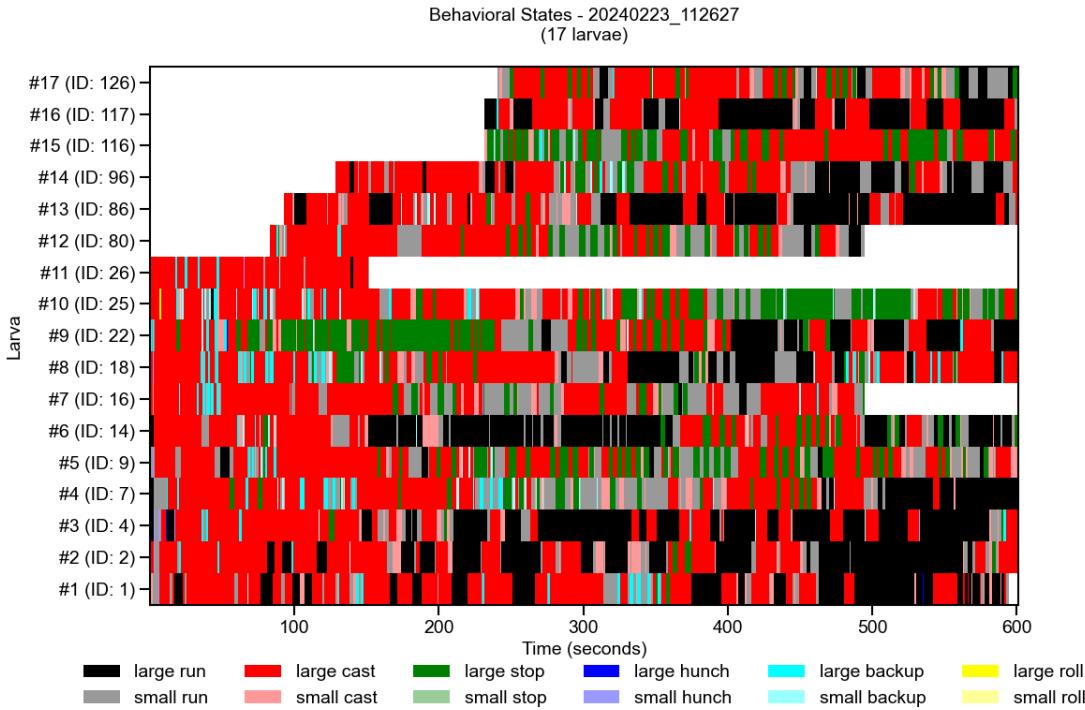


```
[7]: trx_filtered_by_merging = preprocessing.
    ↪merge_short_stop_sequences(trx_filtered_by_removing_stops,
    ↪min_stop_duration_cast=2.0, min_stop_duration_run=3.0)
results_merged_behavior_matrix = plot_data.
    ↪plot_global_behavior_matrix(trx_filtered_by_merging)
```

Merged 965 sequences with short stops:

- 379 cast-stop-cast sequences
- 509 run-stop-run sequences
- 77 mixed sequences (run-stop-cast or cast-stop-run)

Total duration saved: 997.62 seconds



### 0.3 Analysis of one single `trx.mat` file

```
[8]: # Create subplots with one regular and one polar
fig = plt.figure(figsize=(12, 6))

# Create regular subplot
ax1 = fig.add_subplot(1, 2, 1)

# Create polar subplot
ax2 = fig.add_subplot(1, 2, 2, projection='polar')

# Analyze and plot
run_results = data_processor.analyze_run_orientations(trx_filtered_by_merging,
                                                       bin_width=10, sigma=2)

# Regular plot
plot_data.plot_run_orientations(run_results, ax=ax1)

# Polar bars with error lines on top
plot_data.plot_run_orientations_polar(run_results, ax=ax2, bar_style=True)

# Or still use line style if needed
# plot_data.plot_run_orientations_polar(run_results, ax=ax2, bar_style=False)
```

```
# plt.tight_layout()
```

[8]: <PolarAxes: title={'center': 'Run Orientations (Polar)'>

