fishproviz: Fish Tracking Data Processing and Visualization Module

Project 21: Developing Exploration Behavior

Holds the scripts to visualize the molly's trajectory. To make use of the links to the mp4 and csv-files – connect to the server `loopbio_data`. Currently the links work for MacOS systems and best with Adobe Reader.

Requirements

- `python3, gcc, latex`
- Install dependencies:
  - `pip3 install -r requirements.txt`

Build

- To compile the *Cython* code and creating you own `fishproviz/config.env`, run:
  - `python3 setup.py build_ext --inplace`
- To install the package and import functions elsewhere:
  - `python3 -m pip install .`

Run the main.py

```
usage: python3 main.py [-h] [-ti TIME_INTERVAL] [-fid FISH_ID] [--include_median]
                       {trajectory,feeding,trial_times,activity,turning_angle,
                       abs_angle,tortuosity,entropy,wall_distance,all,clear}

This program computes metrics and visualizations for fish trajectories,
the results are saved in the directory:
'DIR_CSV_LOCAL'

positional arguments:
  {trajectory,feeding,trial_times,activity,turning_angle,abs_angle,
  tortuosity,entropy,wall_distance,all,clear}
                        Select the program you want to execute

options:
  -h, --help            show this help message and exit
  -ti TIME_INTERVAL, --time_interval TIME_INTERVAL
                        Choose a time interval in second to compute
                        averages of metrics. Also possible [day, hour].
  -fid FISH_ID, --fish_id FISH_ID
                        Fish ID to run can be set by 'camera_position'
                        or index, default is all fish_ids
  --include_median      Include median or not only for activity

Example of use: python3 main.py trajectory -fid 0
```

File Structure

The variable *path_csv_local* in fishproviz/config.env is the root of the project and the place where all generated data is stored. In addition to the front an back directory where all the tracking data is stored you will find the following directories after the corresponding program executes. - visualizations/trajectory (pdf) - visualizations/feeding (pdf) - visualizations/plots (single plots) - config_data - where we store feeding times, area coordinates, calibration, etc. - results

1. Trajectory Visualization PDFs

fishproviz/config.env contains the paths to the trajectory data. One can configure these to point to the correct location of the data. Reading the data directly from the server `loopbio_data` results in long running times. It is recommended to use a external hard drive. If you path uses spaces, for example the name of the hard drive, rename it to underscores – `_`.
Accessing the data from the server is very slow.

1.1 Generate the trajectory visualizations, ***run***:

- Trajectories: `python3 main.py trajectory`
- Feeding Trajectories: `python3 main.py feeding` The CSV-file for time spend feeding and number of visits are stored at `results/feeding`.
  - Requirements: Provide a csv-file (;-separated) with start and end time for the feeding measures with columns in the following format:

day	time_in_start	time_in_stop	time_out_start	time_out_stop
dd.mm.yy	hh:mm	hh:mm	hh:mm	hh:mm

An example template can be found at data/recordings_feeding_times_template.csv

Then run the `bash-script`:

- `bash scripts/build-trajectories.sh`
- Optional argument:
  - `--feeding` or `-f` for the feeding trajectories.
  - `--test`, `-t` is used to test the script, to generate only the fist pdf.
  - `--local`, `-l` to use the paths of the local hard drive to link the csv file in the pdf.

  - `--cam-id, -cam` followed by `cameraID_position`, to create only the pdf for the given camera.

Remark: For the bash-script you can not build feeding and non feeding trajectories in parallel as they use the same files.

2. Data File and Path Validation

The python script `fishproviz/path_validation.py` is used to validate the filenames and paths of the data files. It logs all error messages into `log-path-validation.txt`.

2.1 Run the script:

- `python -m fishproviz.path_validation path="path/to/root"` – where you would find the directories for front and back position.
- for example: `python -m fishproviz.path_validation path="/Volumes/Extreme_SSD/FE_tracks"`
- Optional arguments:
  - `delete=1` – to delete duplicated filenames
  - `n_files=<<number of files>>` – to change the expected number of files in a folder for a day. The default is *15* for feeding use *8*.

3. Trajectory Analysis

- run: `python3 main.py <<metric>>`
- For **metric** use one keyword out of:
  - `activity, turning_angle, tortuosity, entropy, abs_angle, wall_distance`.
- run `python3 main.py <<metric>> --time_interval <<hour/day>>` to record mean and standard derivation per fish per hour/day in one csv-file.

3.1 Metrics:

- step length is the length of the vector drawn between to consecutive data frames.
- the mean and standard derivation illustrated in the visualization is computed from filtered data frames, removing obvious error point and normed by the distance between data frame when erroneous data point where removed.
- The number of spikes is defined by the threshold of ‘ $> \mu + 3\sigma$‘
- For the sum of angles we take each angle between consecutive steps anticlockwise ‘$\alpha \in [-\pi, \pi]$‘.
- For the average angle each angle ‘$\alpha > 0$‘

4 DATA Visualizations

4.1 Entropy Density

- run: `python3 -m src.visualizations.entropy_plots` *plotly needs to be installed*
- run: `bach scripts/entropy_density.sh` The PDFs with show in tex/entropy_density

4.1 Metrics over 4 Weeks

- run: `python3 main.py program=all time_interval="day"` to calculate all metrics by day an save them to a csv
- run: `python3 -m src.visualizations.activity_plotting` to plot the data of the csv-files.
- run: `bash scripts/metrics.sh` to create the summery PDF.

Further documentation will follow here…