

Sep 02, 2024

Tardigrade videography, tracking, and step analysis

DOI

dx.doi.org/10.17504/protocols.io.kxygxy8nwl8j/v1

Ian Woods¹

¹Ithaca College



Ian Woods

Ithaca College

OPEN  ACCESS



DOI: dx.doi.org/10.17504/protocols.io.kxygxy8nwl8j/v1

Protocol Citation: Ian Woods 2024. Tardigrade videography, tracking, and step analysis. **protocols.io**
<https://dx.doi.org/10.17504/protocols.io.kxygxy8nwl8j/v1>

License: This is an open access protocol distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Protocol status: Working

We use this protocol and it's working

Created: September 01, 2024

Last Modified: September 02, 2024

Protocol Integer ID: 106808

Keywords: tardigrade, gait analysis, computer vision, automated tracking, videotracking, tardigrades,

Abstract

Protocols for tardigrade handling, preparation for videography, video acquisition, path tracking, step timing quantification, and analysis. Descriptions of Python scripts used in the analysis pipeline.

Gel-bottomed dishes

- 1 Tardigrades will not generally walk in glass or plastic Petri dishes, because they need something to grip with their claws. *Hypsibius exemplaris* individuals reliably walk if they are able to grasp the surface with all eight claws. We use plastic Petri dishes lined with 2% agar or agarose (in Poland Spring water). Agarose is preferable when using inverted microscopes, because agar tends to have more debris which can obscure the tardigrade and interfere with automated tracking.

Tardigrade Wrangling

- 2 Tardigrades are sensitive to heat; exposure to warm lights (e.g. in the base of dissecting microscopes) can be mitigated by placing the tardigrade dish on top of an inverted Petri dish or a lid. We transfer tardigrades via glass capillaries. Because tardigrades are so small, they are easily lost; when possible, the tardigrade should be watched under the microscope throughout the entire transfer procedure.
- 3 To capture a tardigrade, gently knock it off its feet, and apply gentle suction to the capillary.
- 4 Lift the capillary out of the water, hold it in place under the microscope, and position the new container below the capillary.
- 5 Expel the tardigrade onto the surface of the agar or agarose in a small volume of water.
- 6 Confirm that the tardigrade has been transferred successfully, and use the glass capillary to clean out debris and reduce the volume of water in the drop. As the volume of water decreases, the tardigrade will be able to grip the surface.
- 7 If necessary, the tardigrade can be gently rolled onto its feet with a small flexible wire or an "Irwin Loop" (**Schram and Davison 2012**). Once the tardigrade establishes a grip, it will begin walking.

Videos of tardigrade locomotion

- 8 Prepare 35 mm Petri dishes by adding approximately 500 μ L of molten 2% agarose (e.g. 200 mg agarose in 10 mL Poland Spring water) to the bottom of each dish.
- 9 Prepare your compound scope for video capture: launch the camera software, take a micrometer image under the 10X objective, and make a folder to save your videos. Locating

and following a tardigrade under the scope can be stressful; you want to minimize any other distractions and things to do.

- 10 From stock dishes, select a large tardigrade, preferably a green-bellied one without eggs. Under a dissecting microscope, transfer the tardigrade into the center of the agarose-bottomed Petri dish. To prevent loss of the tardigrade, it is highly recommended to keep the tardigrade in view under the microscope.
- 11 Remove algae or other debris, leaving the tardigrade in a small drop of water in the dish. The small droplet of water is helpful in locating the tardigrade when the dish is transferred to a compound scope. To encourage the tardigrade to grasp the surface of the agarose, it may be necessary to remove a bit of water or to gently drag the tardigrade to the edge of the droplet.
- 12 Transfer the dish with the tardigrade to a compound scope. We use an inverted microscope, though an upright microscope would also work. Locate the tardigrade under low power, and focus on the claws. Keeping the tardigrade in view, slowly add 1000 μ L of Poland Spring water to the dish. If water is added too quickly, the tardigrade may be knocked off its feet and start squirming. If the tardigrade loses contact with the agarose, start again with a new tardigrade, or remove most of the water from the dish, leaving the tardigrade in a small droplet.
- 13 Making sure to keep the tardigrade in view, switch to the 10X objective and focus on the claws. Start recording video. Adjust the stage position as needed to follow the tardigrade, but the goal is to keep the background stationary as much as possible. We typically record videos for 2 minutes. Save the video with a standardized filename (e.g. initials_date_treatment_tardigradeNN.mov)

Path tracking

- 14 Extract clips from the video, where the tardigrade is walking but the background is stationary. On a Mac, these clips can be extracted using the Trim tool in QuickTime; on a Windows PC, these trimmed clips can be generated via the Photos app. Save each clip with a standardized filename (e.g. initials_date_treatment_tardigradeNN_clipstart-clipend.mov).
- 15 To run the tracking scripts, you need to have Python installed, along with numpy, matplotlib, scipy, opencv, and their dependencies. The Anaconda distribution of Python (<https://www.anaconda.com/products/distribution>) has most of these; opencv needs to be installed after the Anaconda environment is set up (e.g. by pip3 install opencv-python). Download the tracking scripts from GitHub (<http://www.github.com/enwudz/gait>) and save the scripts in the same folder as your videos (or in a separate location, if you are comfortable providing the directory path when running the scripts).
- 16 After dividing each movie into clips with stationary backgrounds, run the scripts below. Navigate in your Terminal (in Mac, or the Anaconda Command Prompt in Windows) to the folder that contains your videos. To run a program, type python [program name] into the



Terminal. When you run each program, you will be asked to select which clip you would like to analyze.

- 17 Run `initializeClip.py`, which makes an excel file to record the data in each clip, and extracts information about the video (e.g. frame size, frame rate, duration).
- 18 Run `autoTracker.py`, which identifies the moving object in the video, and records the coordinates of the centroid of the object in each video frame. Adds this data to the excel file created by `initializeClip.py`.
- 19 Run `analyzeTrack.py`, which smoothes the coordinates of the tracked path, and calculates the speed and direction of travel of the moving object in each video frame. Identifies bouts of sustained forward walking, during which step kinematics and interleg coordination patterns can be measured (below). Adds this data to the excel file created by `initializeClip.py`.
- 20 [Optional]: run `plotClip.py`, which provides options for plotting aspects of the path of each tracked object.
- 21 [Optional]: run `demoTracking.py`, which takes the output of path tracking done by `autoTracker.py`, and makes a video of the tracked tardigrade labeled with the center of mass in each video frame, the path of the tardigrade, and the calls of stops and turns. This script is useful for validating the tracking scripts.

Step and gait tracking

- 22 After path tracking (above), step timing data can be collected from periods of sustained forward walking in selected clips. In general, we aimed to collect step timing data for 10-20 seconds for each tardigrade, and prioritized periods of sustained forward walking that were around 8 seconds (the typical time it takes for a tardigrade to walk across the field of view under a 10X objective).
- 23 To run a program, type `python [program name]` into the Terminal. When you run each program, you will be asked to select which clip you would like to analyze.
- 24 Run `cruiseSummaries.py`, which reads path tracking data for all the excel files associated with each individual, and summarizes the bouts of sustained forward walking. Output of this script is useful to determine which clips to analyze in `frameStepper.py`
- 25 Run `frameStepper.py`, which gives an option to create rotated, cropped frames, orienting the animal so that the direction of motion is up in each video frame, and keeping the center of mass in the center of each image frame. Asks which leg to track, and then allows frame-by-frame progression through each video. When the target leg initiates stance or swing, timing can be recorded by keystrokes. All legs can be processed in one session, or the existing leg data can be saved and finished later. Adds timing of leg swings and stances to the excel file produced by `initializeClip.py`.



- 26 Run analyzeSteps.py, which calculates step kinematic data (stance duration, swing duration, duty factor, step period) for each stride of each leg. Calculates averages for these values for each leg. Determines which legs are in swing phase in each frame, and assigns a hexapod ICP and a rear ICP to each frame. Calculates the tetrapod and tripod coordination strength for each bout where these ICPs are evident. Adds this data to the excel file created by initializeClip.py.
- 27 [Optional]: Run plotClip.py. Once step data is collected for a clip, plotClip.py script provides options for plotting aspects of step kinematics and interleg coordination for that clip.

Combining data from individual clips

- 28 Data from individual clips can be combined per individual for further analyses or for comparisons between groups (e.g. treatments).
- 29 Run combineClips.py, which combines data from all excel files within a folder, grouping each excel file by each individual, calculating averages across the data for that individual, and combines data into a single excel file.
- 30 [Optional] Run compareGroups.py: offers options to plot comparisons between groups within the spreadsheet produced by combineClips.py.

Producing the Figures in Anderson et al. 2024

- 31 Figures for this study were produced in jupyter notebook, and the notebooks are named according to the figures. Notebooks can be launched by typing 'jupyter notebook' into the terminal, after navigating to the directory with the notebooks. Jupyter is included in the Anaconda distribution of Python. Input for these figures is the excel file produced by combineClips.py above.
- 32 01_Tracking_Overview.ipynb produces Figure 1, S1 Figure, and the 'Exploratory Locomotion' columns in S1 Table.
- 33 02_Single_Clip_Quantification.ipynb produces Figure 2.
- 34 03_Step_Kinematics_lateral_rear.ipynb produces Figure 3 and the 'Step Kinematics' columns in S1 Table.
- 35 04_Phase_Offsets_ICPs.ipynb produces Figure 4, S3 Figure, and the 'Measurements of Interleg Coordination' columns in S1 Table.
- 36 05_Speed_Partitions.ipynb produces Figure 5.



- 37 06_Rear_leg_coordination.ipynb produces Figure 6.
- 38 07a_Adults_vs_Juveniles.ipynb produces the portions of Figure 7 and S2 Table relevant to comparisons between adults and juveniles.
- 39 07b_exemplaris_vs_Ramazzottius.ipynb produces the portions of Figure 7 and S2 Table relevant to comparisons between *H. exemplaris* and *Ramazzottius*.
- 40 07c_Control_vs_disulfiram.ipynb produces the portions of Figure 7 and S2 Table relevant to comparisons between controls and disulfiram-treated tardigrades.
- 41 0S2_Example_ICPs.ipynb produces S2 Figure.
- 42 0S4_Suppression_release_leg_swings.ipynb produces S4 Figure.

Other scripts in the 'gait' repository

- 43 Many of these scripts support the scripts above, are useful in validating output of the scripts above, or were not used in the primary analysis for the Anderson et al. 2024 manuscript.
- 44 batchRename.py renames selected file types in a directory.
- 45 boxWalker.py makes an animation of swinging and standing legs, in the form of a grid of boxes of different colors, based on step timing data produced by frameStepper.py.
- 46 clickerTracker.py has the same functionality as autoTracker.py, but requires the user to mark the center of mass in each video frame.
- 47 demoTracking.py takes the output of path tracking done by autoTracker.py, and makes a video of the tracked tardigrade labeled with the center of mass in each video frame, the path of the tardigrade, and the calls of stops and turns. This script is useful for validating the tracking scripts.



- 48 gaitFunctions.py contains tracking and analysis code that is used by several of the core analysis scripts.
- 49 legAnimator.py makes an animation of swinging and standing legs, in the form of a line-drawn tardigrade (or human, or cat), based on step timing data produced by frameStepper.py.
- 50 legacyStepTracking.py updates data collected in the early portion of this project to a format that is consistent with later data collection.
- 51 makeMovieFromImages.py creates an .mp4 movie from a folder full of individual image frames.
- 52 manualCritterMeasurement.py enables measurement of length and width of an animal from selected video frames. This script is useful in validating the measurements produced by autoTracker.py.
- 53 measureImage.py enables measurement of any distance between objects or landmarks on selected video frames. This script is useful in validating the measurements produced by autoTracker.py.
- 54 qcStepTracking.py compares timing of stance and swing onset calls from two runs of frameStepper.py. This script is useful in training individuals to do the analysis, and for quality control of step tracking data.
- 55 retrack.py enables retracking of a video clip, but leaves existing step timing data intact.
- 56 rotaZoomer.py is used by frameStepper.py to produce a new movie from a tracked clip, where the animal is continually oriented in the direction of motion, and the frame is cropped to only show the animal. This script was also used to produce the movies in S2 Movie.
- 57 runClip.py enables several of the tracking and step timing analysis steps to be done with one command.
- 58 walkaPalooza.py produces an animation simultaneously showing the stepping animal and the step timing data

Drug treatments



- 59 Make 5 mL of control and drug solutions in 1% DMSO in Poland Spring Water.
- 60 Make fresh agarose-bottomed (2% w/v, e.g. 200 mg in 10 mL Poland Spring water) 50 mm Petri dishes. From a single stock of tardigrades, add 10-15 tardigrades to each new experimental dish. Selected tardigrades should be consistent in size and walking ability. Avoid tardigrades that are filled with eggs; their bellies often appear brown rather than green.
- 61 Fill each dish with 5 mL of control or drug solutions. Add 5 drops of fresh, concentrated *Chlorococcus* algae to each dish.

Work Cited

- 62 **Schram, Mark D., and Paul G. Davison. 2012. "Irwin Loops — A History and Method of Constructing Homemade Loops." Transactions of the Kansas Academy of Science (1903-) 115 (1/2): 35–40.**