Stuff I tried to do behavior analysis: 1. MATLAB scripts

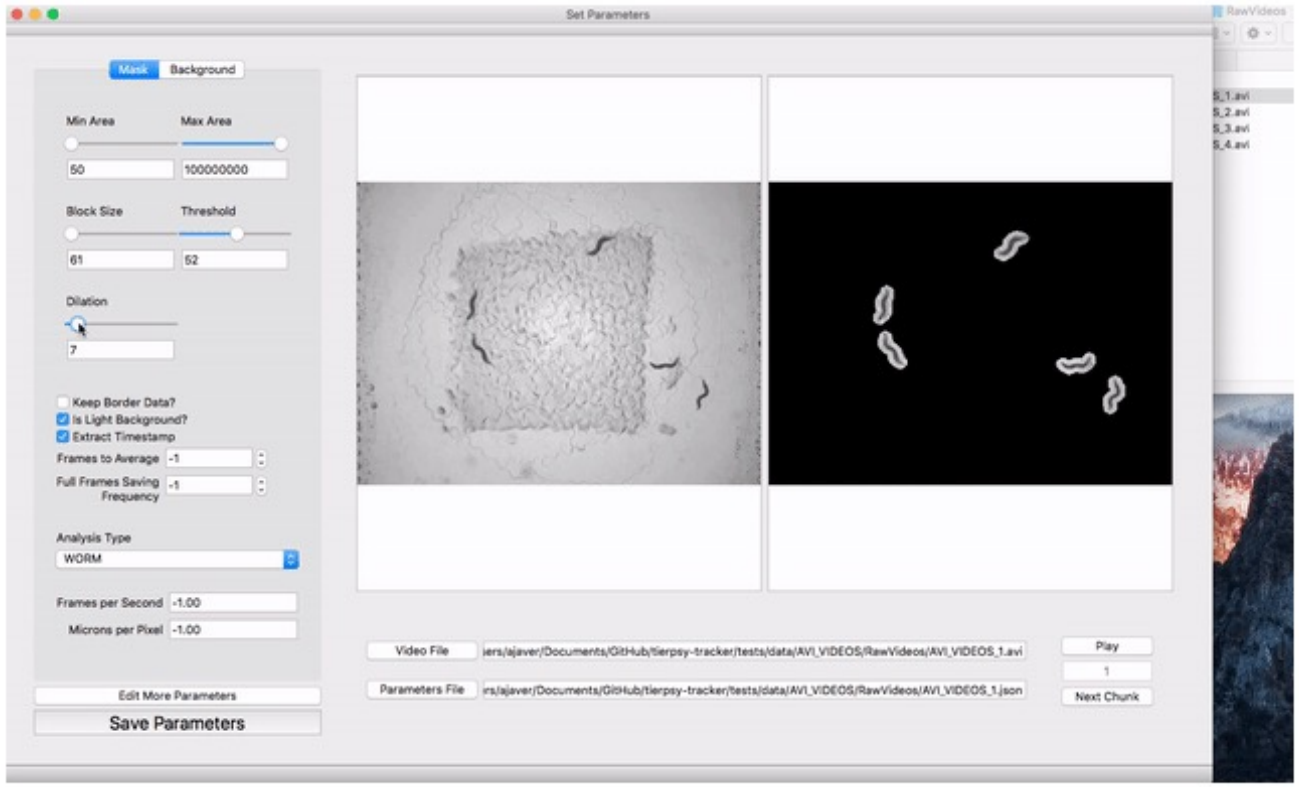
```
AngSpeed
Curvature
Direction
Eccentricity
FrameRate
Frames
Height
MajorAxes
Name
NumFrames
PixelSize
Reorientations
RingDistance
Size
SmoothX
SmoothY
Speed
State
Time
Width
Wormlength
body_contour
bound_box_corner
mvt_init
numActiveFrames
original_track_indicies
stimulus_vector
```

```
procFrame = [];
if(Prefs.NumCPU<=1 ||  NumFrames < 2*Prefs.TrackProcessChunkSize) % just do it in one process
    procFrame = master_process_movie_frames(MovieName, localpath, FilePrefix, Ring, stimulusfile, startFrame, endFrame, target_numworms);
    dummystring = sprintf('%s%s.%d_%d.procFrame.mat',localpath, FilePrefix, startFrame, endFrame);
    mv(dummystring, procFrame_file);
    clear('dummystring');
else % fork_process_movie_frames to process chunks in parallel

    if(isempty(Ring.NumWorms) || isempty(Ring.DefaultThresh) || isempty(Ring.meanWormSize))
        [~, ~, ~, Ring] = default_worm_threshold_level(MovieName, calculate_background(MovieName), procFrame, target_numworms, Ring);
        save(sprintf('%s%s.Ring.mat',localpath, FilePrefix), 'Ring');
    end
```

- Would have loved to use this firstly of course
- Very hard for me to understand and setup (might actually just be incompatible with a Mac)
- Some functions seem very dated (reading individual video frames at a time)

2. “Tierpsy tracker”



Software Explanation

This page explains all the steps executed in the analysis of a video file. See the Output Files section for a description of each of the files created by Tierpsy Tracker.

Create Trajectories

TRAJ_CREATE

The first step is to identify possible particles. We divide the image into regions of non-zero connected pixels. For each candidate region we calculate a simple threshold and create a binary mask. We identify each individual object and calculate their centroids, areas and bounding boxes. Only objects with features within user-defined ranges are kept. This information is stored in the /plate_worms table.

TRAJ_JOIN

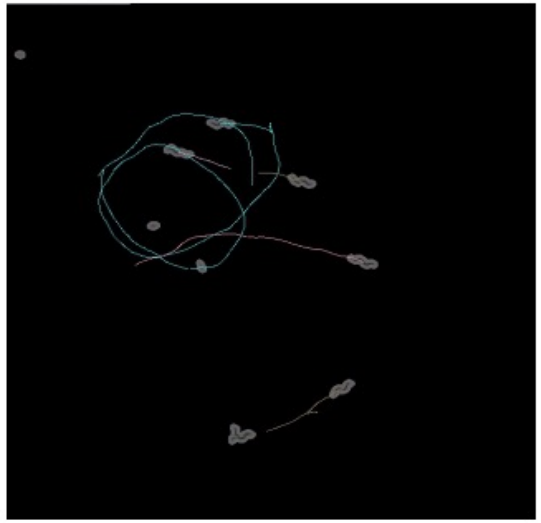
The second step is to join the identified particles into trajectories. The algorithm changes depending if the analysis is selected for a single-worm or for multi-worm. See the instructions of the widget Set Parameters for more information.

In the **single-worm**, the algorithm the particles are first filter by area. The filter threshold is calculated with the assumption that in most of the frames the worm is the largest object. Only one trajectory is linked using the closest neighbors in consecutive frames.

In the **multi-worm** case, we link the particles' trajectories by using their nearest neighbor in consecutive frames. The nearest neighbor must be less than `max_allowed_dist` away and the fractional change in area must be less than `area_ratio_lim`. One particle can only be joined to a single particle in consecutive frames, no split or merged trajectories are allowed. If these conditions are not satisfied it means that there was a problem in the trajectory e.g. two worms collided and therefore in the next frame the closest object is twice the area, or a worm disapeared from the field of view. If there is any conflict, the trajectory will be broken and a new label will be assigned to any unassigned particle.

In a subsequent step, Tierpsy Tracker tries to join trajectories that have a small time gap between them i.e. the worm was lost for a few frames. Additionally we will remove any spurious trajectories shorter than `min_track_size`.

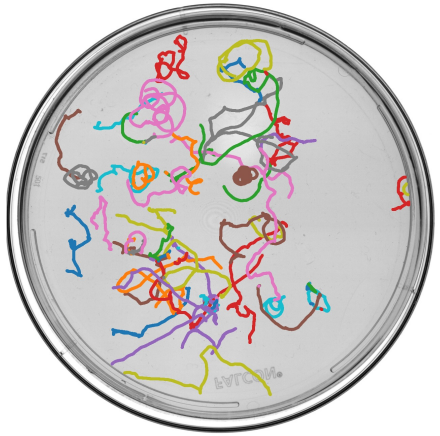
Below there is an example of how the trajectories look after tracking.

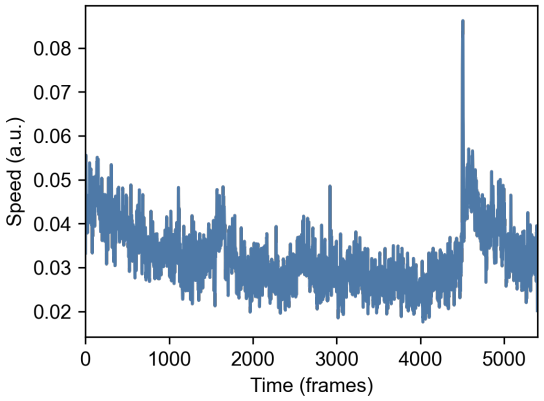


- Maybe 30 minutes a video, downstream fitting of worm to a skeleton is very rigorous
- Would also have loved to use this, just couldn’t get it to work with smaller worms

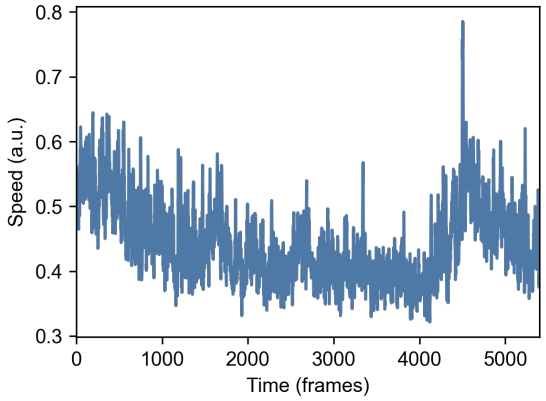
3. Python script

MATLAB





Python




```
def main():
    make_tif_dir()
    get_thresh_and_mask()
    make_unlinked_csv()
    seperate_unlinked_csv()
    make_linked_csv()
    trim_csv()
```

```
Done with tif conversion! 29 seconds
Done with getting threshold and mask! 0 seconds
Done with identifying blobs! 59 seconds
Done with separating initial csv! 1 seconds
Done with linking blobs! 3 seconds
Done with trimming! 0 seconds
```