ZebraZoom Documentation

Table of Contents

Visit our website zebrazoom.org to learn more about ZebraZoom.

I - Easy Installation and Usage through the GUI:

A – Download and Installation:

See instructions here: https://zebrazoom.org/index.html?#download

B – Using ZebraZoom with the GUI:

NB: GUI means “Graphical User Interface”

To be able to track animals in videos you need to create a configuration file for each “type” of video you want to track. A “type” of video is defined by light intensity, number and shape of wells, number of animals per well, number of pixels per animal, the type of animal in your video, etc...

Start by testing that ZebraZoom is working on your machine. To do that, download the test videos (see: https://zebrazoom.org/testVideos.html) and try to run the tracking on those: in the GUI's main menu, click on “Run ZebraZoom on a video”, choose the video you want to track and then the configuration file which will have the same name as the video you want to track. Once the tracking is done, go back to the main menu and click on “Visualize ZebraZoom's output”, then on the video you just tracked. If the tracking worked well, you should be able to visualize the output produced by ZebraZoom (by clicking on “View video for well 0” for example).

II – Adjusting ZebraZoom hyperparameters

A – Adjusting hyperparameters through the GUI:

In order to track videos other than the ones provided on ZebraZoom's website, you might need to create your own configuration files. In order to do that, you can click on "Prepare configuration file for tracking" and follow the steps described to create a configuration file adapted to the videos you want to track. Please note that this procedure isn't complete and may not work on all videos. If you don't manage to create a configuration file on your own, you can contact us at info@zebrazoom.org and we will try to make one for you.

Tip: once you've created a configuration file for some videos and launched the tracking on those videos using that configuration file, check the quality of the tracking and bouts extraction by clicking on "Visualize ZebraZoom's output". If you are unsatisfied with the results, you can refine the configuration file you created by clicking on "Prepare configuration file for tracking" in the main menu and then by clicking on the box "Click here to start from a configuration file previously created (instead of from scratch)": this will allow you to reload and refine the configuration file you already created. You can then save that refined configuration file and use it to re-tracked your videos.

B – Further adjustments of tail angle smoothing and bouts and bends detection:

If you are tracking the tail of zebrafish larva, then you might need to further refine the parameters controlling the smoothing of the tail angle and the identification of bouts and bends. To do this, start by clicking on "Visualize ZebraZoom's output" and then on the name of the video you just tracked. Then look at some of the bouts and click on the button "Change Visualization" to compare the smoothed tail angle from which the bends are extracted with the raw un-smoothed tail angle. If the smoothing of the tail angle or the bouts and bends detection is not good enough, you can refine the configuration file to adjust the parameters controlling the smoothing and the bouts and bends detection. To do this, open your configuration file in a text editor (your configuration file should be in the folder ZebraZoom/configuration), and add/change the parameters listed below. You can then relaunch the tracking with that updated configuration file. When you relaunch the tracking, check the box "I ran the tracking already, I only want to redo the extraction of parameters.".

Post-processing of bouts initially detected: parameters below control the removal of "outlier bouts":

detectBoutMinNbFrames : default: 2:
minimum number of frames a bout must have to be detected

detectBoutMinDist : default: 4:
minimum distance traveled during the bout (between beginning and finish) for the bout to be detected

detectBoutMinAngleDiff : default: -1:
minimum variation of the angle (max(angle)-min(angle)) for the bout to be detected

minNbPeaksForBoutDetect: default: 2:
minimum required number of bends in a bout for the bout to be detected

noChecksForBoutSelectionInExtractParams: default: 0:
If set to 1, none of the checks described below will happen

Parameters related to the smoothing of the tail angle:

tailAngleSmoothingFactor : default: 0.001:
Smoothing factor applied on the tail angle. Higher values lead to more smoothing.

tailAngleMedianFilter : default: 3:
Window of the median filter applied to the tail angle (before smoothing).

Parameters related to the detection of bends:

These two first parameters control the initial detection of the bend through the “find_peaks” function of scipy (https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html).

windowForLocalBendMinMaxFind : default: 1:

minProminenceForBendsDetect : default: 0.01:

For time t, if the angle is a local minimum/maximum for the values between t-windowForLocalBendMinMaxFind and t+windowForLocalBendMinMaxFind, and if the “depth” of that maximum/minimum is at least minProminenceForBendsDetect, then a bend is detected at time t. If minProminenceForBendsDetect is equal to -1, then minProminenceForBendsDetect is set to minProminenceForBendsDetect = maxDiffPeakToPeak / 10, maxDiffPeakToPeak being the difference between the maximum and the minimum values of the tail angle over the entire bout.

The parameters below control the post processing of the peaks previously found (they control the removal of “outlier bends”):

minDiffBetweenSubsequentBendAmp : default: 0.02:
if the bend “n” has a value X, then the bend “n+1” must have a value Y for which absoluteValue(X-Y) > minDiffBetweenSubsequentBendAmp. If the bend “n+1” doesn't satisfy that condition, then the bend is not detected.

minFirstBendValue : default: -1:
minimum value required for the first bend (so by default all bends are accepted)

doubleCheckBendMinMaxStatus : default: 1:
if doubleCheckBendMinMaxStatus is equal to 1, then only keeps bends for which:
bend(n-1) > bend(n) and bend(n) < bend(n+1)
bend(n-1) < bend(n) and bend(n) > bend(n+1)

removeFirstSmallBend : default: 0:
if removeFirstSmallBend is different than 0 (so not by default), then removes the first bend if:
abs(TailAngle_smoothed[firstBend]) < abs(TailAngle_smoothed[secondBend]) / hyperparameters["removeFirstSmallBend"]

III – Analyzing the results with Python:

A result folder will be created for each video you launch ZebraZoom on inside the ZebraZoomPy/ZZoutput folder.

If you have launch ZebraZoom on a video named “video”, you can load the results in Python with the following code:

```
import json

with open('ZZoutput/video/results_video.txt') as f:
  supstruct = json.load(f)
```

Then, you can see the data for the well numWell, the fish numFish, and the bout numBout using the following command:
supstruct['wellPoissMouv'][numWell][numFish][numBout]

For example, if you want to look at the data for the first bout of the “fish 1” in the third well, you can type:

```
supstruct['wellPoissMouv'][2][0][0]
```

You can then, for example, plot the tail angle with the following command:

```
import matplotlib.pyplot as plt
plt.plot(supstruct['wellPoissMouv'][2][0][0]["TailAngle_smoothed"])
plt.show()
```

IV – Launching ZebraZoom on several videos at the same time:

A – Easy method through the GUI:

You can very easily launch ZebraZoom on several videos just as you would for just one video through the GUI. However, this feature is only available with the premium version of ZebraZoom (this additional feature is the only difference between the normal and premium version of ZebraZoom). Please consider purchasing this premium version of ZebraZoom as it will help us continue to develop our free software (contact us at *info@zebrazoom.org* to know more). If you do not want to help us continue to develop our free software, you can use the more complex method described below.

B – More complicated method through the command line:

To launch ZebraZoomPy on a video, open a terminal, navigate to ZebraZoomPy/ZebraZoomPy and launch ZebraZoomPy with the following parameters:
1: path to video
2: video name
3: video extension
4: path to configuration file

For example:

```
python ZebraZoom.py ../videosForZZPy/ video avi configuration/config1-1.json
```

Make sure you put a "/" at the end of the videos path

If you create an .sh file, put that command line inside it, and launch that .sh file, it will launch the tracking as well. Instead of putting only one line inside of the file launch.sh, you can put as many lines as you want inside it. For example, you could put inside launch.sh:

```
python ZebraZoom.py folderPath1 fileName1 videoExtension config.txt
python ZebraZoom.py folderPath2 fileName2 videoExtension config.txt
python ZebraZoom.py folderPath3 fileName3 videoExtension config.txt
python ZebraZoom.py folderPath4 fileName4 videoExtension config.txt
python ZebraZoom.py folderPath5 fileName5 videoExtension config.txt
```

each line representing an execution of ZebraZoom on one video. You can then run all of those videos at once by executing that script launch.sh in a terminal.

V – Further analysis of the data extracted:

We have some code to further analyze data, for example to compare different populations or to cluster bouts of movement. Contact us at info@zebrazoom.org to know more and visit https://zebrazoom.org/index.html?#features to see how we might be able to further help you.