# **DANCE Analysis Workflow - README**

## **Overview**

This document provides step-by-step instructions for processing and analyzing JAABA classifier score files, extracting behavioral data, and visualizing results.

## **Prerequisites**

### **Install Required Packages**

1. Download and install Python (recommended: 3.8+); Use Visual Studio Code (VS Code) or another preferred IDE for execution
2. Install dependencies using:
   * pip install pandas numpy openpyxl matplotlib

For Conda users:

* + conda install pandas numpy openpyxl matplotlib

1. Download and install MATLAB
2. Janelia Animal Automatic Behavior Annotator (JAABA)
   * Download and install: <https://sourceforge.net/projects/jaaba/files/>
   * JAABA Git repository: <https://github.com/kristinbranson/JAABA>
   * JAABA documentation: <https://jaaba.sourceforge.net/>
3. FlyTracker
   * GitHub repository: <https://github.com/kristinbranson/FlyTracker>
   * Download and install all the necessary files.
4. After installing JAABA and FlyTracker, open MATLAB, go to **Home > Set Path > Add with Subfolders, select both folders, click Save, and then Close.’**

### **II. Data Organization**

#### **Before Analysis**

The raw data should be structured as follows:

project\_directory/

│-- DD/MM/YYYY/ # Folder named with the recording date

│ │-- video1.mp4 # Original experimental movie 1

│ │-- video2.mp4 # Original experimental movie 2

│ │-- video3.mp4 # Original experimental movie 3

#### **After Analysis**

Once analysis is completed, the data structure will be:

project\_directory/

│-- DD/MM/YYYY/ # Folder named with the recording date

│ │-- video1/ # Original video folder

│ │ │-- video1\_0/ # Cropped version 1

-- video1\_0\_JAABA/ # Tracker folder for each cropped video

│ │ │ │ │-- perframe/ # Per-frame analysis

│ │ │ │ │-- trx.mat # Tracking data

│ │ │ │ │-- score\_files/ # Post-tracking score files

│ │ │-- video1\_1/ # Cropped version 2

│ │ │-- video1\_2/ # Cropped version 3

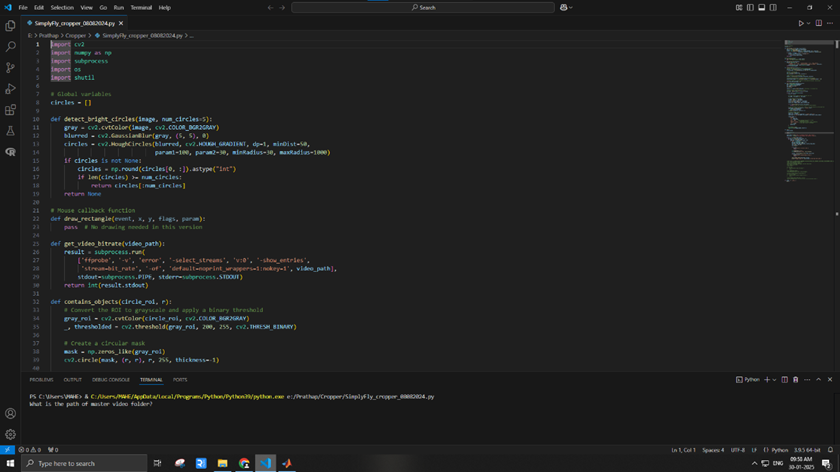
**Step-by-step instructions to quantify behavior:**

**1. Video recording**

· Record videos in either .mp4 or .avi format.

**2. Video segmentation using cropper script (see README.txt for SimplyFly\_cropper\_v4.py)**

* Double-click on **cropper.py**. This will open the script in your Python IDE (we used Visual Studio code as our IDE).



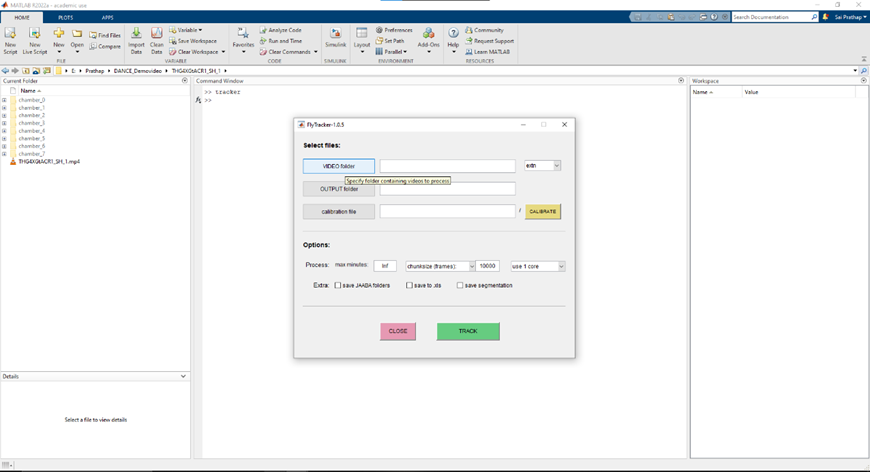
* In **Visual Studio Code**, click on the **Play** button at the top left to run the Python file.
* The Visual Studio Code terminal will ask: **"What is the path of the master video folder?"**
* Copy and paste the master video folder path into the terminal and press **Enter**.
* The terminal will ask: **"How many chambers are there per video?"**
* Enter the number of chambers in the arena and press **enter**.
* The script will process and segment the video into individual folders (as FlyTracker requires video folders, not individual video files).
* When segmentation is complete, the terminal will display: **"Cropped videos for X circles have been saved."** This confirms successful segmentation.

### **Notes**

* Ensure the **master video folder** contains the necessary input video files before running the script.
* The **output folder** will be structured as required for FlyTracker.
* If errors occur, verify that the correct folder path and number of chambers were provided.

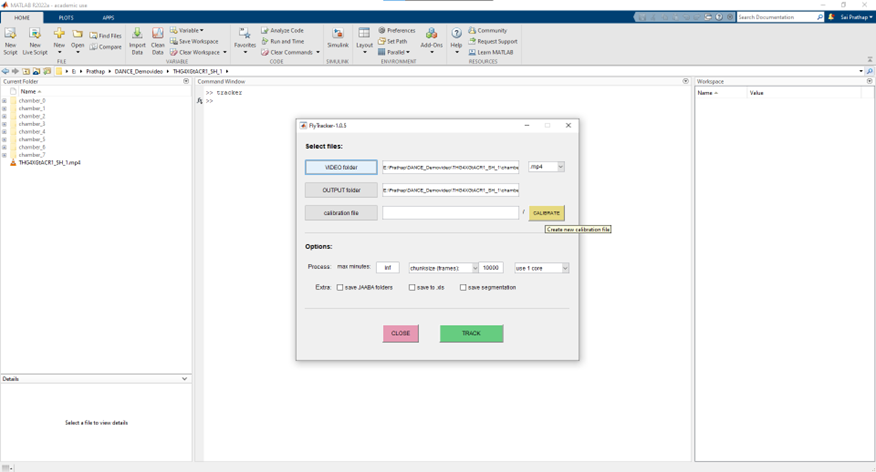
**3. Fly tracking using MATLAB and FlyTracker**

* Open **MATLAB**.
* Click on **Set Path** > **Add with Subfolders** to include both **FlyTracker** and **JAABA** folders.
* In the MATLAB command prompt, type: ‘tracker’
* A dialog box will appear. Click on **Video Folder**, select the appropriate folder, and click **Select Folder**.

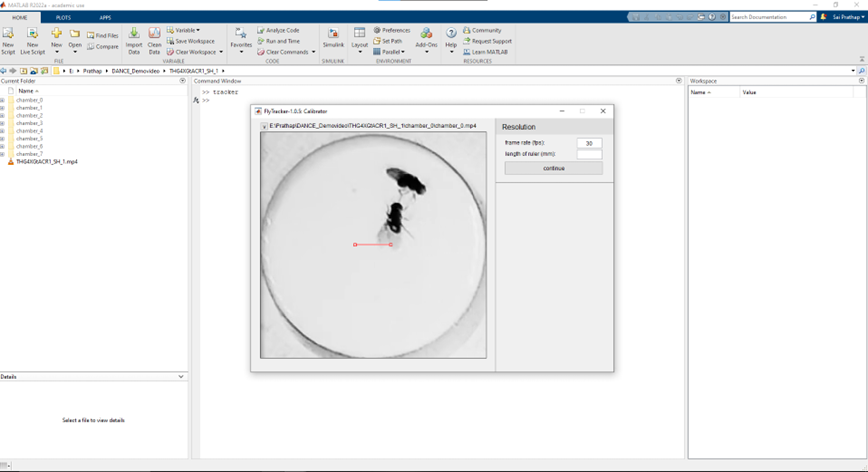


**4. Calibration in FlyTracker (see FlyTracker: https://kristinbranson.github.io/FlyTracker/)**

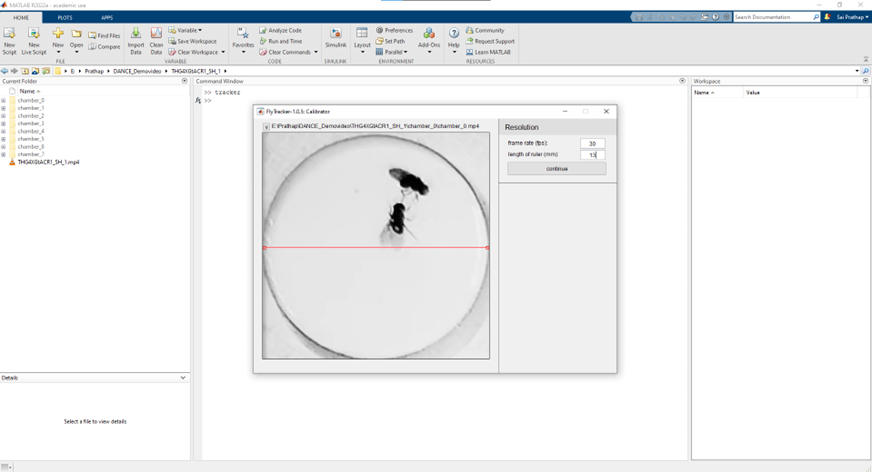
* Click **Calibrate**. A dialog box titled **"Loading Calibrator"** will appear.



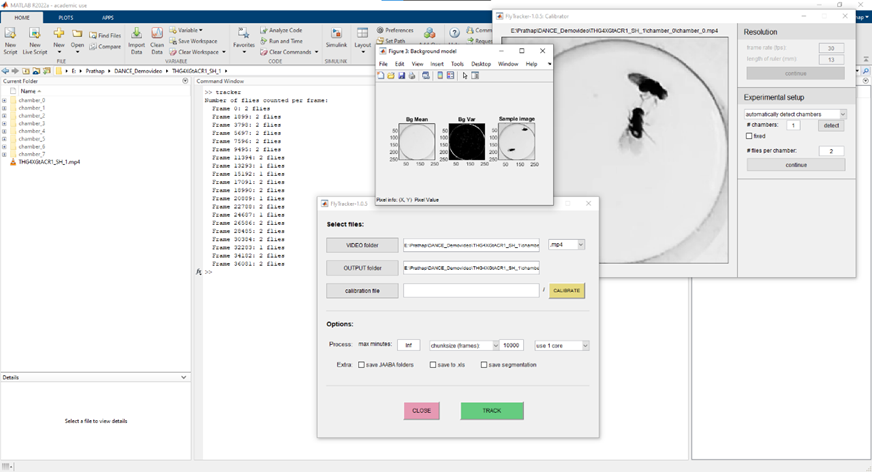
* A chamber outline with red markers will be displayed.



* Click and drag the red markers to align with the chamber’s right and left boundaries.
* In the left panel under **Resolution**, enter 13 in the **Length of the ruler (mm)** field (adjust as per the chamber size).
* Click **Continue**.



* A **Progress** dialog box will appear, displaying **"Computing background image"**.
* MATLAB will indicate progress by displaying frame numbers (e.g., 20 minutes × 30 fps = 36,000 frames).
* Once completed, a **Background** model with Bg Mean, Bg Var, and a Sample Image will be shown.



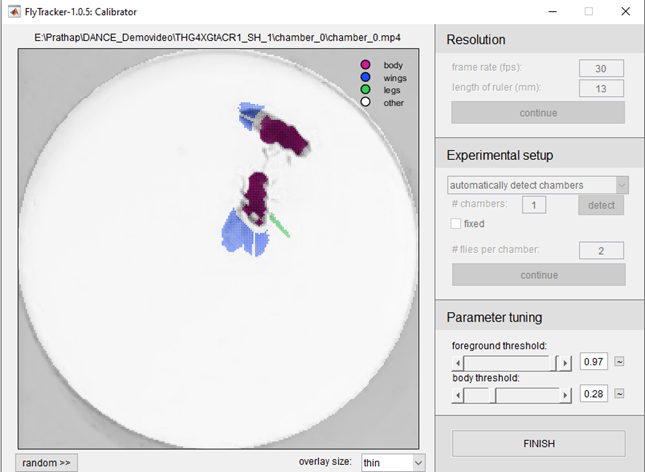
**5. Experimental setup in FlyTracker**

* Under **Experimental Setup**, specify the number of chambers and the number of flies per chamber.
* Enter the number of chambers (e.g., 1 or 12) and click **Detect**.
* A **Progress** dialog box will appear, showing **"Detecting chambers"**, with blue circles marking detected chambers.
* If chambers are not properly detected: Click on the **Drop-down menu** under **"Automatically Detect Chamber"**.
* Select **"Manually Set Chambers"** and adjust chamber boundaries accordingly.
* The system will automatically detect **# Flies Per Chamber**.
* Click **Continue**.

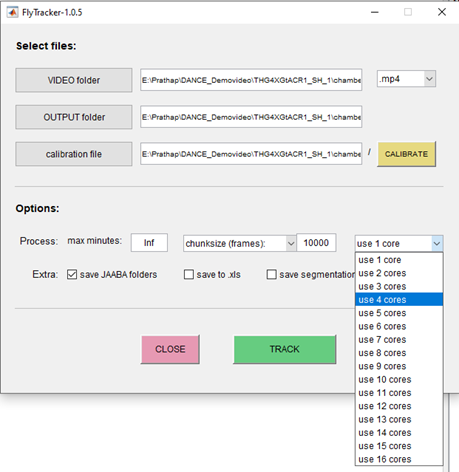


**6. Parameter Tuning and Tracking**

* Adjust the **Foreground** and **Body Threshold** to accurately identify fly body, legs, and wings using the color representation at the chamber’s corner.
* Click **Finish**.

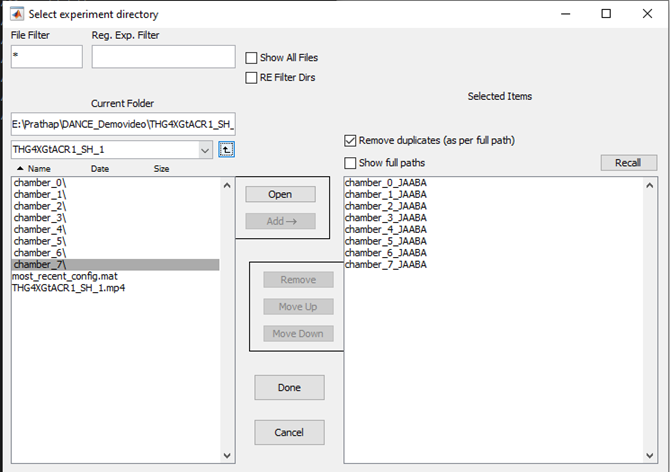


* A dialog box will appear: **"Finalizing Calibration"**.
* Enter the number of CPU cores for computation.
* Click **Save JAABA Folders**.
* Click **TRACK** to begin tracking.
* Once completed, a JAABA folder will be created inside the master folder, containing a perframe folder with perframe features and a trx.mat file.

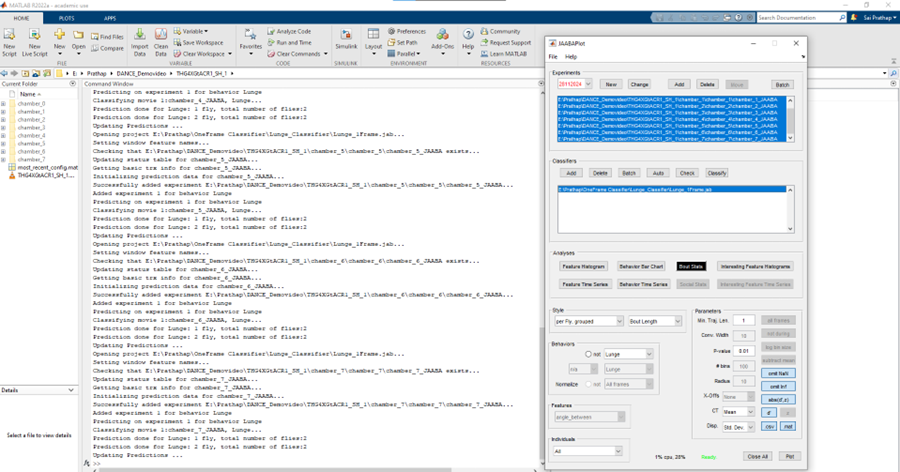


**7.** **Generating score files using JAABA**

* Open MATLAB and type the following command in the command prompt: JAABAPlot
* A dialog box will appear. Under the **Experiments** section, click **New**, assign a file name (e.g., use the date of the video folder, such as 28112024), and click **OK**.
* A color selection dialog box will appear. Choose color-blind-friendly colors and press **OK**.
* Click **Add**, and a dialog box will appear for selecting the experiment directory.
* Copy and paste the tracked video path into the folder selection window.
* Double-click on the folder, add the **JAABA** subfolder and press **Done**. This step ensures that all JAABA folders are added.



* Under the **Classifiers** section, click **Add** and navigate to the **lunge** or **courtship** classifiers to quantify behaviors.
* Click **Classify**. Once processing is complete, it will display **Ready**, confirming the successful generation of score files. These files will be located in the **JAABA** folder within the master video directory, alongside the perframe folder and trx.mat file.



**8. Lunge behavior quantification**

* Locate the **Lunge\_Analyzer\_indiC.py** script (see README.txt).
* Double-click on Lunge\_Analyzer\_indiC.py to open the script in **Visual Studio Code**.
* In Visual Studio Code, click the **Play** button at the top left to execute the script.
* The VS Code terminal will ask: What is the complete path of the parent folder containing all videos and JAABA folders?
* Copy and paste the full path of the parent directory and press **Enter**.
* The script processes the score files and generates compiled .csv files containing behavior quantification data.

**Results and data output**

* The final output consists of **compiled score files** in .csv format, which contain quantification of the analyzed behaviors. These results facilitate downstream statistical analysis and visualization.

**9. Courtship behavior quantification**

Compiles CSV files based on specific classifiers, aggregating courtship behavior data for analysis. IIdentifies male flies by assuming the male’s wing extension counts are higher than the female’s, then assigns that identity to the respective fly for all other behaviors. Generates new CSV files summarizing the data.

* Locate **SheetMaker\_postprocessed.m**, add it to the **MATLAB path** using **Set Path > Add with Subfolders**, then click **Save** and **Close**.

Next, open **coutship\_bout\_num\_extractor.py** in **Visual Studio Code**, run the script, and when prompted, enter the complete path of the parent folder containing both **video** and **JAABA** folders. **Results and data outputs:**

* **A folder named "score\_files" contains renamed and organized files.**
* **A compiled CSV file for each classifier.**
* **Extracted courtship behavior frames in terms of bouts.**

**Alternative: Processing Entire Videos Without Segmentation**

You can process the entire video without segmenting chambers by following the same steps for FlyTracker and JAABA analysis.

To quantify courtship bouts in unsegmented videos, use:

**coutship\_bout\_num\_extractor\_unsegmented.py** – This script works just like the previous one and follows the same steps.