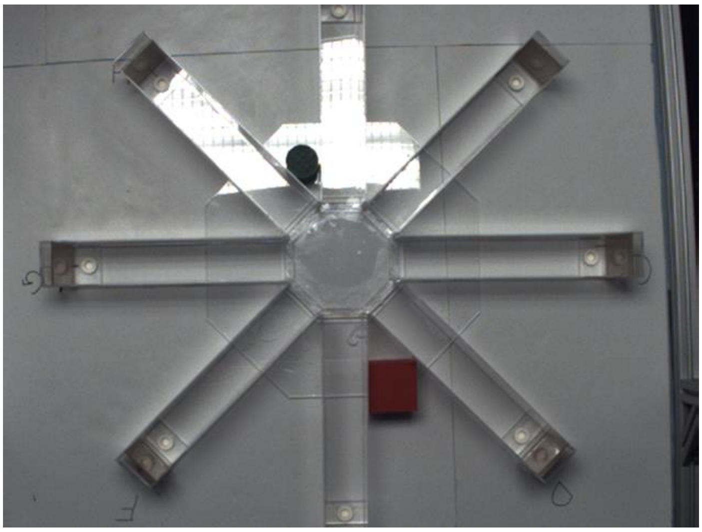
Mannual for Radial Arm Maze Experiments

ezTrack-new serves as a toolkit to populate and fix experimental dataset in Radial Maze Experiment. It is based on ezTrack, an open-source video analysis pipeline for the investigation of animal behavior.

Procedure

Part I: Prepare the maze at the beginning of each training day

1. At the beginning of each training day, thoroughly clean the inside and outside of the maze with ethanol. The goal is to remove any visual spots from the acrylic panels of the maze.
2. At the beginning of each training day, begin by rotating the maze by 90 degrees clockwise. This rotation does not need to be repeated during experiments conducted on the same day. This will prevent the mouse from using defects in the maze as a visual cue for which arm to travel down. This will also account for any other variables that may be influencing a mouse to choose a particular arm of the maze.
   a) It is important to remember to replace the external visual cues in the same relative position to the maze. Note that the green cylinder is between arms A and H, and that the red cube is between arm D and E, with the side of the cube aligned along arm E. Each day the setup on the camera view should be as close to the one pictured below as possible.



   b) It is also important to exchange the fruit loops at the end of each maze at least once per week. These fruit loops act as an odor control in order to assure that a similar smell can be found in every arm.
3. When arranging the configuration for the maze it is also important to only use the same four bait cups, no matter the testing configuration. This means that if the bait is contained in arms A,C,D,F

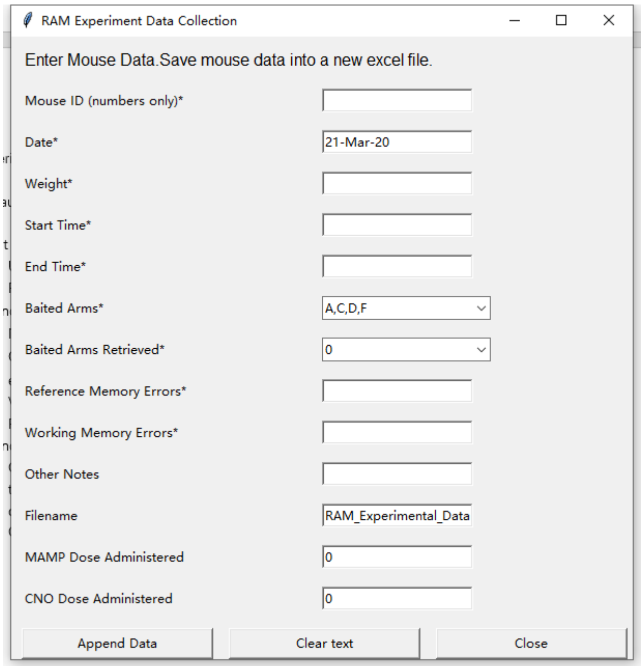
and you need to switch to a B,D,F,H configuration – take the bait cups out of A,C,D,F and switch them to B,D,F,H accordingly. This will prevent you from having to clean the cups between each trial and will ensure that when mice retrieve the wrong arm they get absolutely nothing.

4. Cut a fruit loop into approximately 1/8" x 1/8" pieces and place one into each of the four designated arms for your specific configuration.
5. Place the cardboard shield besides the maze properly, make sure it will not fall down during the experiment.

Part II: Launch all of the required software

---

1. Start the computer.
   a) User: NEC_Lab01
   b) Password: Neur0c0ntr0ls
2. Launch camera software.
   a) Navigate to C:\Users\NEC Lab Maze01\Documents. Select the 'RAM Camera Configuration' and click 'Open' to launch IC Capture 2.4, the camera software. This step ensures that the videos we record have consistent parameters such as fps and resolution.
   b) Verify that the image appears on the screen and that the maze is oriented appropriately. Reference Image in 2a above.
3. Launch jupyter notebook.
   a) Open Anaconda Prompt from the task bar, which is supposed to be in *(base)* environment, type in *jupyter notebook* and hit enter. This should open a window of directories in the default browser. Navigate to /Documents/ezTrack-new.
   b) Open 00_DataCollectionGUI.ipynb, run it to get a GUI like this:



   c) Open LocationTracking_Individual.ipynb, which is the pipeline for individual video processing and analysis.

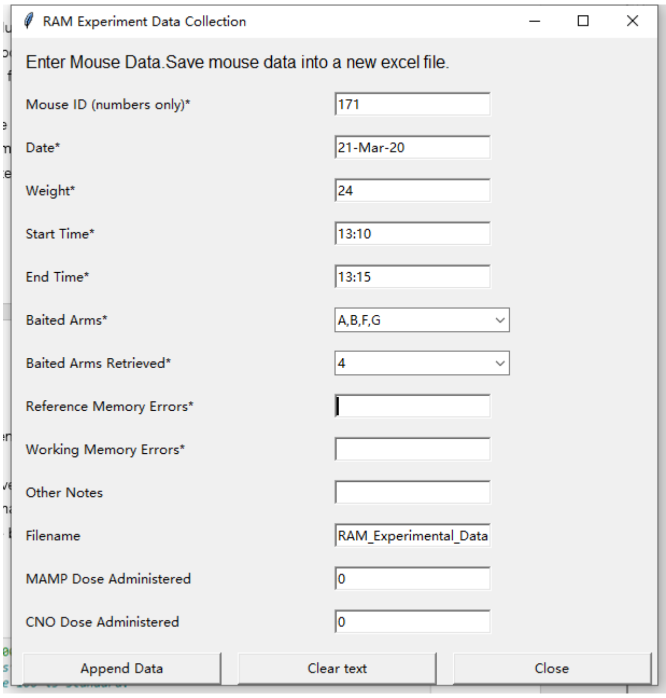
Part III: Record a Reference video

1. Return to IC Capture 2.4, navigate to Capture > Toggle Recording Info Dialog.
   a) A dialog prompt window appears every time you change the filename or the time limit, you can also select the location where the recorded video will be saved when changing the filename. Don't forget to click on 'apply' and 'okay' after changing it.
   b) At the bottom of that window, you can click on the red circle to start recording, this should create an empty file in your targeted folder.
   c) Hit on the black square when you want to end the video, or it will end itself automatically when it reaches the time limit. Upon the end of the recording, the empty video file gets rewritten, and we can check the end time by right click on the .avi file to view its properties (to see the timestamp of modified).
2. Create a reference video for that day's configuration. This must be done at the beginning of every experiment day **and every time the maze is moved from its original position.**
   a) Click to change the filename and create a new folder for that day's experiments, in the 'Radial Arm Maze Experiments' folder.
   b) Name the reference box according to this format: 2020_01_01_EmptyBox1, then click 'Save' and 'Apply.' If you are recording a new reference video on the same training day, make sure to increase the index by 1, for example 2020_01_01_EmptyBox2.
   c) **For reference videos, set the time limit to 10 seconds**. Set it to 00:00:10 and click record, click apply and okay. Click the red circle to start and let it end automatically. Verify it recorded correctly.
3. **Don't forget to change the time limit back to 10 seconds – 00:10:00.**

Part IV: Record Video for Individual

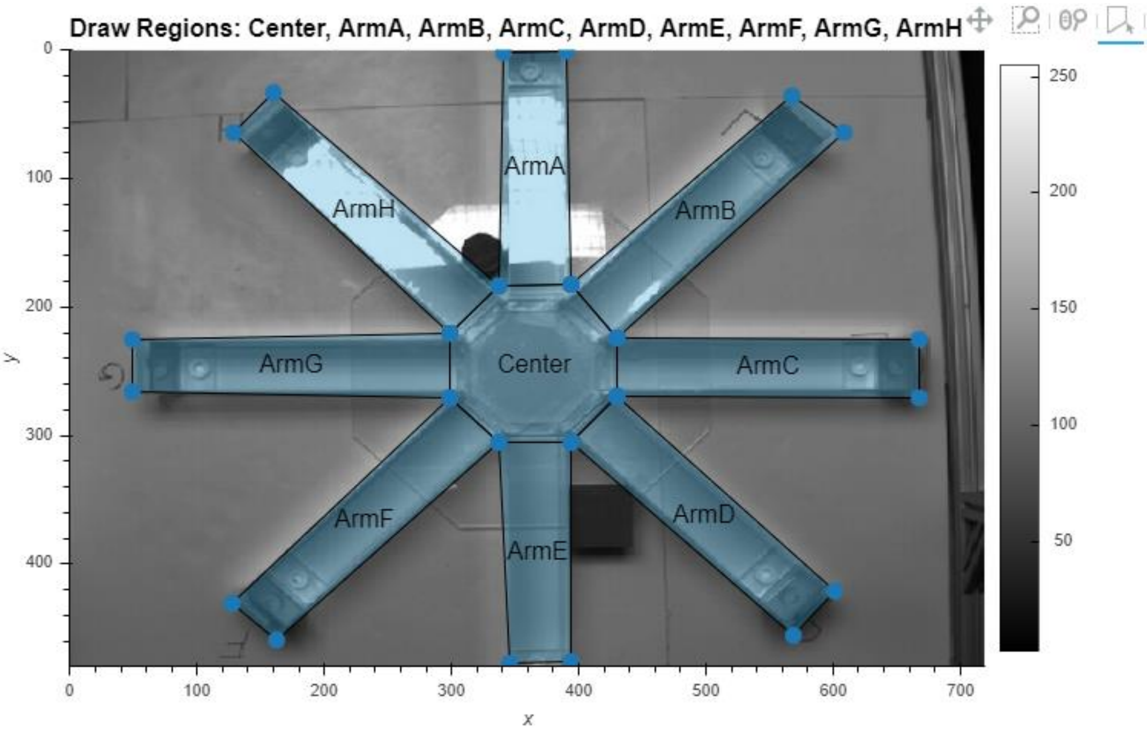
1. Select the mouse you would like to test and then change the video filename to this format:
   2020_01_16_N171_ABFG_EB1_Trial1,
   Where it is the :
   YYYY_MM_DD_MouseNumber_ArmConfig_ReferenceVideo_TrialNumberForThatDay
   Make sure to use the corresponding Emptybox if you have recorded a new one.
2. Remove the selected mouse from its cage and record its ID, weight and baited arms in the data collection GUI.
3. Place the mouse in the maze and click Record on the Recording Info Dialog Box. Make note of the starting time in 24hr format.
4. Observe the mouse and write down each arm where it successfully retrieves a fruit loop treat.
5. After all four baits have been retrieved, stop the recording and take the mouse out of the maze immediately. Make note of the end time. If the mouse does not retrieve all four treats, the recording automatically ends after 10 minutes.

6. Indicate how many arms the bait was successfully retrieved from. If any exception occurred or any interesting observations, please put it into the ‘Other Notes’ entry.

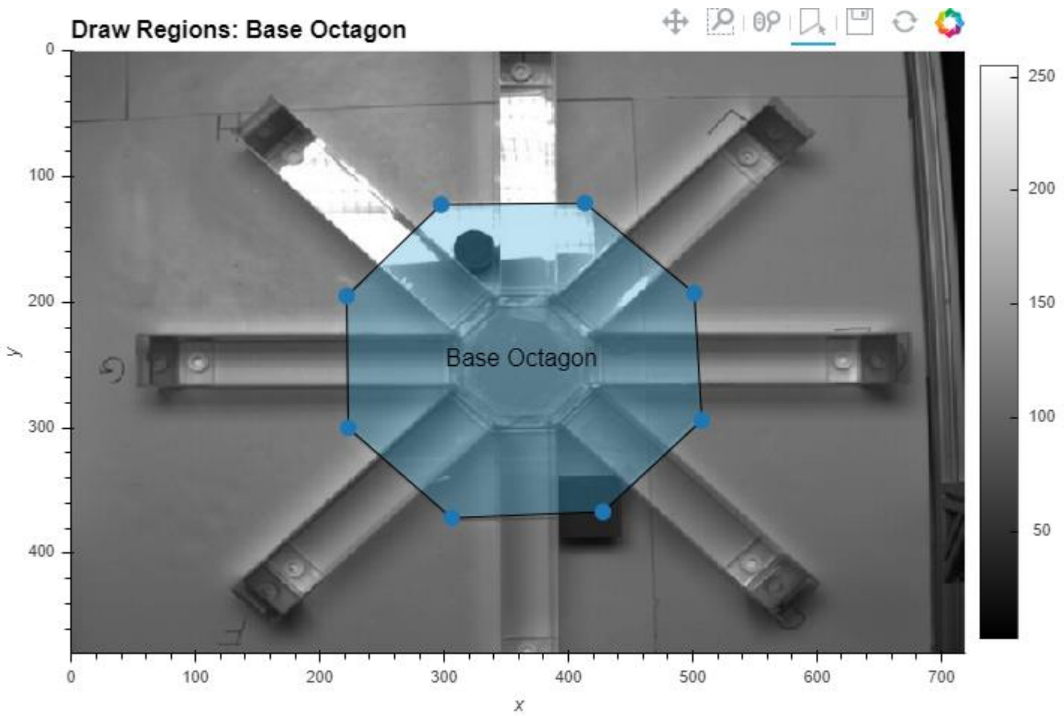


Part V: Run Individual Pipeline

1. Open ‘00_LocationTracking_Individual.ipynb’.
2. Run Cell 1 to load packages.
3. Change the directories and configurations in Cell 2, then run it.
4. Run Cell 3.
5. Run Cell 4 [Option 1] to create a reference from. If there is any shady spot, run it again until there is none. Or you can run Cell4 [Option 2] to get rid of strong stains.
6. Run Cell 5a and use the interactive tools to draw the nine regions of interest. Always begin with the Center and then draw Arm A, Arm B, Arm C, etc.



7. Run Cell 5b to set scale.

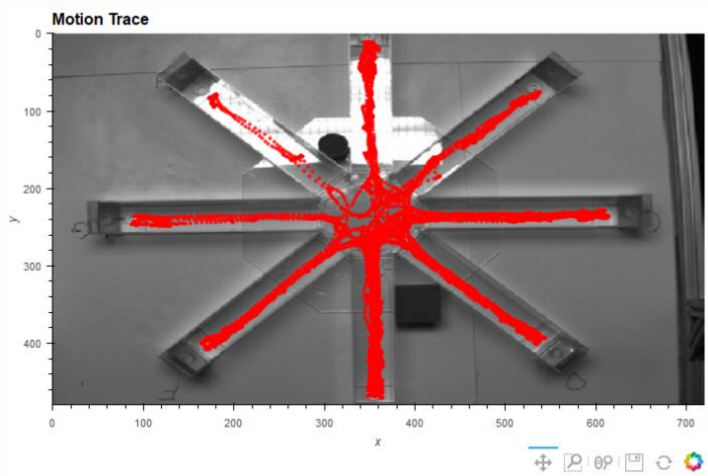


Then run the cell below to convert the metric. The value should be between 0.161 to 0.165 cm per pixel.

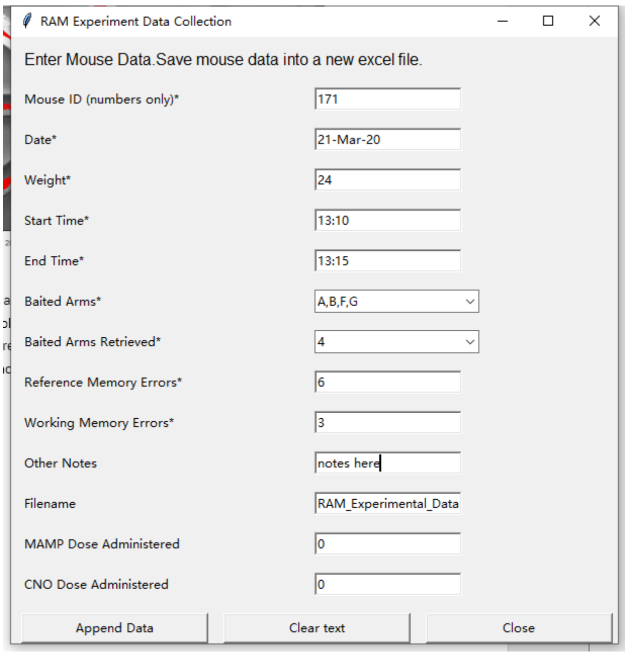
```
In [15]:   cm_per_pixel = va.set_scale(poly_stream_1)

           Conversion:
            0.16435494245790125 cm/pixel
```

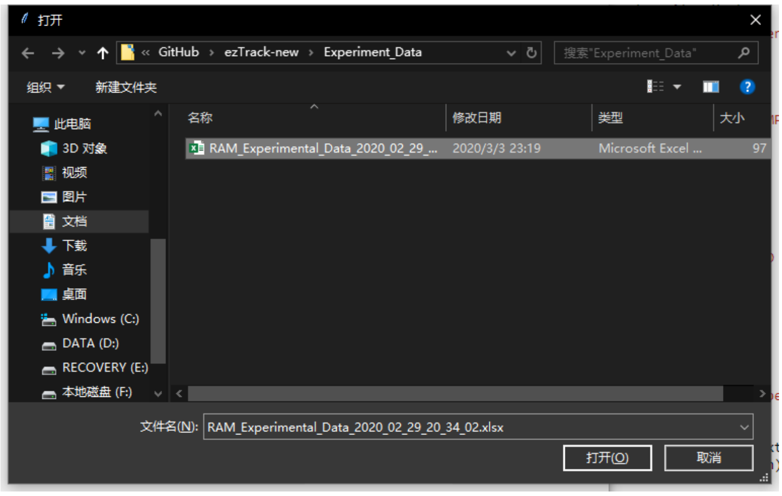
8. Run Cell 6a to create tracking parameters.
9. Run Cell 6b to generate random samples of the tracking algorithm. Verify that the red cross always appears on top of the mouse. Or you can change the parameters in 6a (smaller window_size and higher window_weight) then try again.
10. Run Cell 6c to generate an excel file of the location of the mouse across the entire trial.
11. Run Cell 6d to turn the location data into an overlay of mouse position in the maze.
12. Verify in the folder for that day that the route of the mouse looks appropriate. Here is an example of a good result:



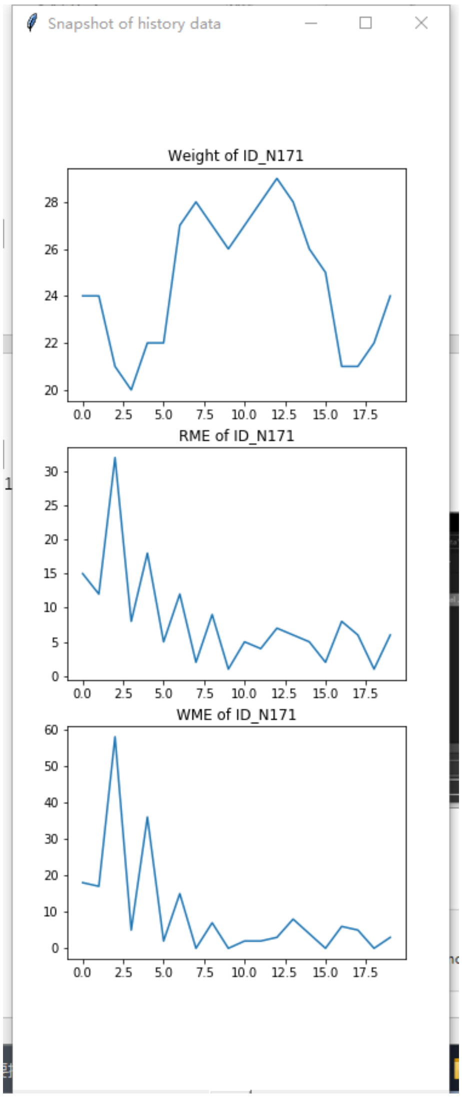
13. Run Cell 7 to create a summary of the trial.
14. Run Cell 8a to get a scatter plot of velocity distribution.
15. Run Cell 8b to calculate the reference memory errors and working memory errors for that trial.
16. Enter the number of RME and WME into the data collection GUI.



17. Hit the ‘Append Data’ button, choose the latest RAM data file to append to.



18. Open it, a new RAM file with time stamp will be created under the same folder, and you can see a snapshot plot of the history of this individual. Close the Snapshot window and hit 'Clear text' button to erase inputs, then do back to Part IV and get prepared for the next mouse.

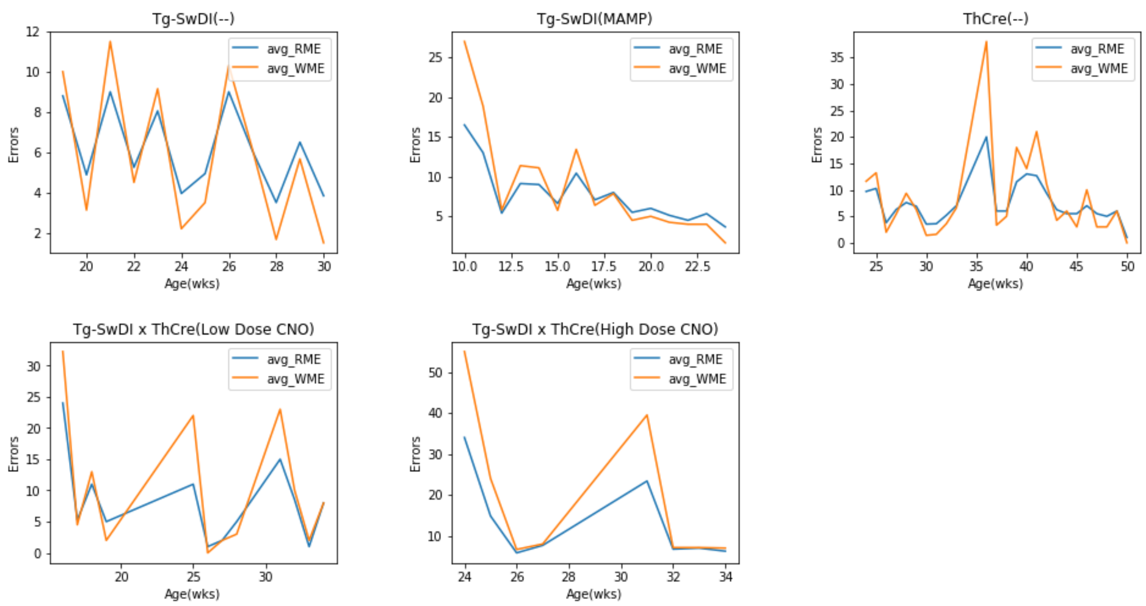


Part VI: Data Integration and Group Analysis

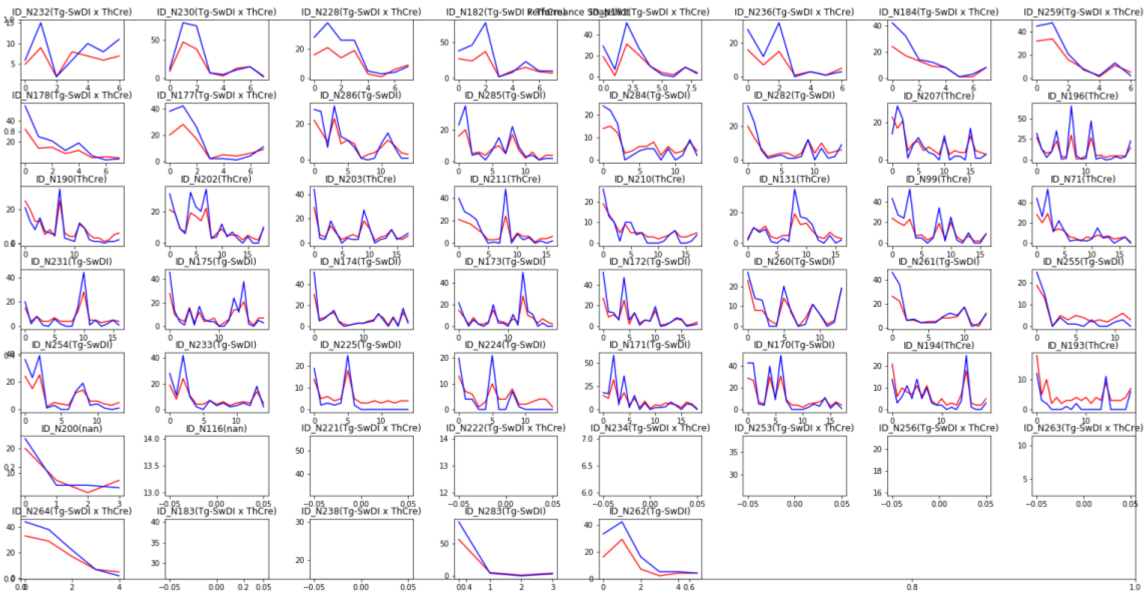
1. By far, you are supposed to have:
   a) A YYYY_MM_DD folder under Radial Arm Maze Experiment Folder, which contains YYYY_MM_DD_EmptyBoxN.avi, as well as following files for each individual trial.

      2019_11_21_N171_ABFG_EB1_Trial1.avi
      2019_11_21_N171_ABFG_EB1_Trial1.png
      2019_11_21_N171_ABFG_EB1_Trial1_ArmRetrieve.png
      2019_11_21_N171_ABFG_EB1_Trial1_LocationOutput.csv
      2019_11_21_N171_ABFG_EB1_Trial1_SummaryStats.csv
      2019_11_21_N171_ABFG_EB1_Trial1_VelocityAndArmRetrieval.csv

   b) History .xlsx files with timestamp under Radial Arm Maze Experiment Folder/Experiment_Data.
2. Now move all YYYY_MM_DD folders into E:/Backup Radial Arm Maze Experiment Folder, which may take a few minutes.
3. Then, open 00_UpdateTables.ipynb and run through it as prompted, which may take a few minutes.
4. Open 00_PerformanceAnalysis.ipynb and run through it to get group performance snapshot since 2000-01-01.



5. Open 00_PerformanceSnapshot_Individual.ipynb and run through it, choose the latest RAM file in the prompt dialog window to get a snapshot of history performance for all individuals.



Part Optional: Batch Process

1. Weight all the mice, keep note of their weight. Train all mice and get videos recorded. No need to keep notes of start or end time if you are going to run 00_PerformanceAnalysis.ipynb afterwards.
2. Run 00_Pipeline_BatchProcess.ipynb, leave it running.

Run Batch Process

```
In [16]:  bp.Batch_Process(video_dict,tracking_params,bin_dict,region_names,stretch,crop,poly_stream,poly_stream_1,scale=cm_per_pixel)
```

```
Processing File: 2019_11_21_N171_ABFG_EB1_Trial1.avi C:\Users\hp\Desktop\cy_test_2020_02_29\2019_11_21\2019_11_21_N171_ABFG_EB1
_Trial1.avi
total frames processed: 3735
Save results to:C:\Users\hp\Desktop\cy_test_2020_02_29\2019_11_21\2019_11_21_N171_ABFG_EB1_Trial1_VelocityAndArmRetrieval.csv
The number of reference memory errors was 10
The number of working memory errors was 7
--------------------------
```

3. When it's done, append memory errors manually using the data collection GUI. As we do not have autofill functions.