Protocol to use Smart-Mouse

1. Login with NX Client to the cluster
2. Open a X-terminal
3. <cd SmartMouse/Code/CurrentVerion>
4. <./runme &>
5. At thoint, matlab will start and you will see the repository screen
6. Select : File->Add Video Sequences
7. Add the movies with individual mice
8. This might take some time since the first time movies are being loaded a lookup table for each frame is generated.
9. Select all videos
10. Select: Pre-processing->Track Single Mouse
11. Answer “YES” for the question “Submit to cluster”
12. At this point, each video sequence will be analyzed by a single computer in the cluster, this might take some time (~1/2-1 hour). You can check whether the jobs that were submitted to the cluster are done by typing <qstat> in the x-terminal window
13. [Optional step] Once jobs are finished, you can verify the tracking results by selecting a single sequence and select Pre-Processing->Verify tracking on single mouse
14. Make sure all video sequences are selected in the repository and select Pre-Processing ->Train classifiers on selected videos. Answering “no” for the Cross-validation will make things faster. Once you get a dialog box which asks where to save a file, select “Classifiers.mat” (or any other name you want).
15. Load the long video sequence to the repository (this can take a long time….)
16. Select the long sequence and select Pre-Processing->Learn Background model
17. In the GUI, select Algorithms->Learn Background.
18. Once the background is learned select Algorithms->Save, and select a file name like “Background.mat” (or any other name you want).
19. Close the background GUI
20. Select Pre-Processing->Create Setup
21. Select the classifiers file and then the background file you previously generated. Save the result under /Data/Setups/setup.mat
22. At this point you should have a new “Setup” in the right list box in the repository screen
23. Select the long video sequence and the corresponding setup file and select Processing->Submit
24. Press “OK” (assuming all parameters are correct)
25. A dialog will appear, asking whether you compiled the application. If this is the first time you are using a new Smart-Mouse version, you should select “No, but compile and continue”. Otherwise, select “Yes”
26. The “Reliable GUI” will pop up. Select “Find Reliable Frames”. This will take ~2-5 minutes. At the end, it will display the detected key frames of the long sequence.
27. Go over each of the key frames and make sure the initial mouse position is correct. The head/tail orientation is not important. Precise position is also not important.
28. [Optional Step] Select “File->Save” and save the key frames into a file.
29. Select “Submit to Cluster”
30. Wait until all jobs are finished (qstat to check what is going on)
31. Go to the repository GUI and select the long sequence, and then Post-Processing->Collect Jobs Results
32. Go to the Jobs manager GUI and select “Refresh”. Then, select all jobs (ctrl+A) and press “Merge Results”
33. Select Algorithms->Fix Identities
34. Select Video->Save Fixed sequence. The output file will contain the positional information for each mouse.
35. Load the output file into a new matlab session
36. The variable astrctTrackers is an array of structure, each one contains the following fields:
    1. m\_afX, m\_afY: size 1xN, containing the X and Y position of the mouse (relative to the top left corner)
    2. m\_afA, m\_afB, size 1xN, containing the major and minor axis length of the fitted elipse
    3. m\_afTheta, size 1xN, containing the ellipse orientation (relative to the X axis) in radians.
37. [Optional Step, GUI and algorithms are still in development stage] Select the long video sequence in the GUI and select Post-Processing->Annotation GUI
    1. Select File->Load Tracking Results And load the file saved in step 34
    2. Run the behavior detectors.
    3. Select File->Save Annotation