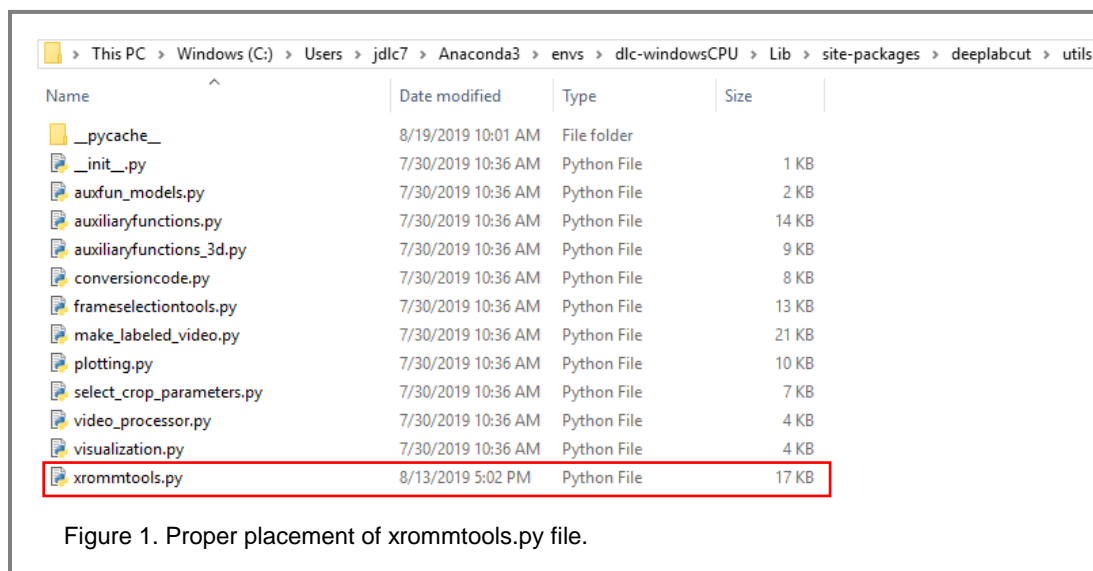


DeepLabCut XMALab Workflow Instructions. Updated 9/10/2019.

## Part 1: Install

1. Ensure latest version of [XMALab](#) is installed.
2. Follow [instructions](#) to install Anaconda + Jupyter Notebooks implementation of [DeepLabCut](#).
3. Download the [XROMMTools for DeepLabCut](#) supplemental files
4. Copy xrommtools.py file into the utils folder where the DLC environment is located (Fig. 1). On Windows, this is probably:

`user\Anaconda3\envs\dlc-windowsCPU(or GPU)\Lib\site-packages\deeplabcut\utils`



## Part 2: Prepare data tracked in XMALab

1. Track 200-600 frames across multiple trials of a single individual in XMALab  
Ensure tracking is as accurate as possible, and that if a marker disappears or is not visible, it is not tracked. DLC is robust to markers going on and offscreen. Marker identities and order must be consistent across trials, otherwise it will not work. For this reason, we recommend using descriptive names (AnteriorMandible and PosteriorMandible instead of Marker 1 and Marker 2).

*N.B.: The appropriate number of frames for the initial training set depends on the number of markers as well as the degree of variation in the full dataset. Deep learning is essentially pattern recognition, so if there are novel postures in the data that aren't in the training dataset, performance will suffer. Augmentation of the training dataset to deal with these problems areas is built into the workflow, however.*

2. Export the 2D points into trial folders. The function xma2dlc randomly selects frames from what is provided, until the preset number of frames is reached. Thus, if you don't want to include some tracked frames in the training dataset, either delete tracked data from those frames in XMA Lab, or replace the rows with NaNs in the exported 2D points file.
3. Create folder structure (Fig. 3). There should be a directory for a given individual, and subfolders for each trial which contain the respective trial images (.avi, .mp4 or image stack) and the exported 2D points file.

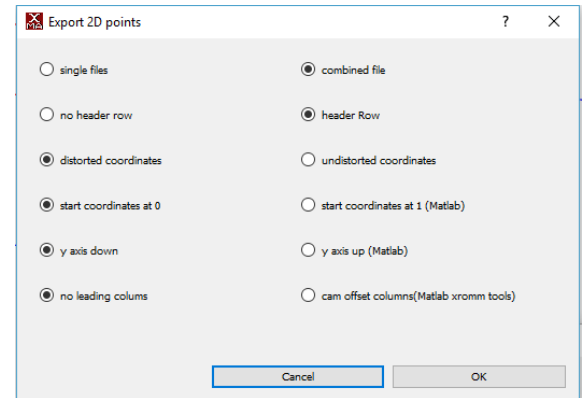


Figure 2. Export settings for 2D points.

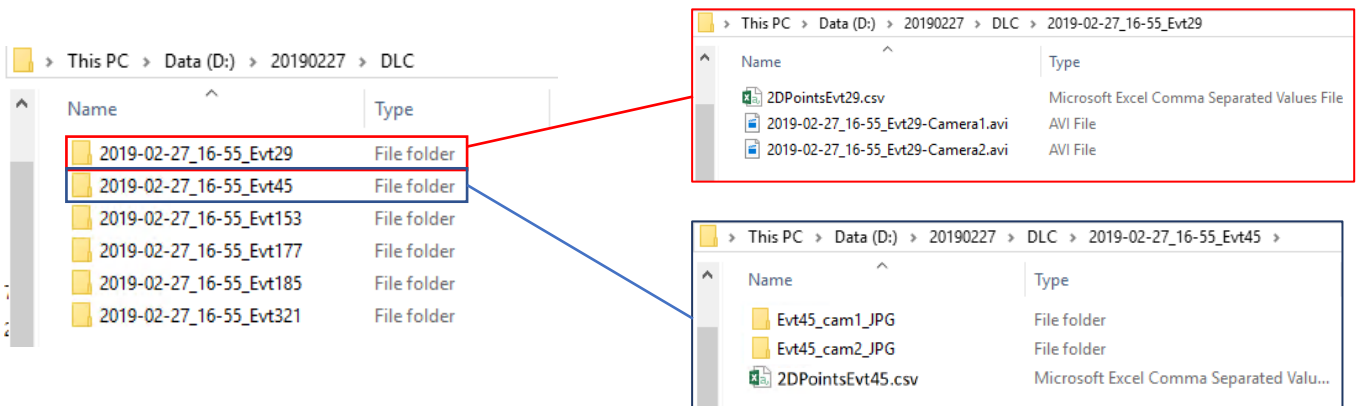


Figure 3. Folders structure for xma2dlc function. Each trial you wish to include in the training dataset should have its own subfolder, which contains the exported 2D points data, and the trial images.

### Part 3: Create projects, training datasets, and train and deploy first iteration of networks

At this point, you can transition to the Jupyter Notebook which contains instructions for the following:

- Create DeepLabCut projects for camera 1 and camera 2. Be sure to update config.yaml files with project paths and point names
- Create training dataset with xma2dlc function.
- Create DeepLabCut training dataset with create\_training\_dataset function.
- Train neural networks.

To open the Jupyter Notebook in Anaconda Navigator (Fig. 4):

1. Open Anaconda Navigator
2. Load DeepLabCut environment
3. Open Jupyter Notebooks

4. Navigate to XROMM\_Pipeline\_Demo.ipynb file (wherever the download was saved).

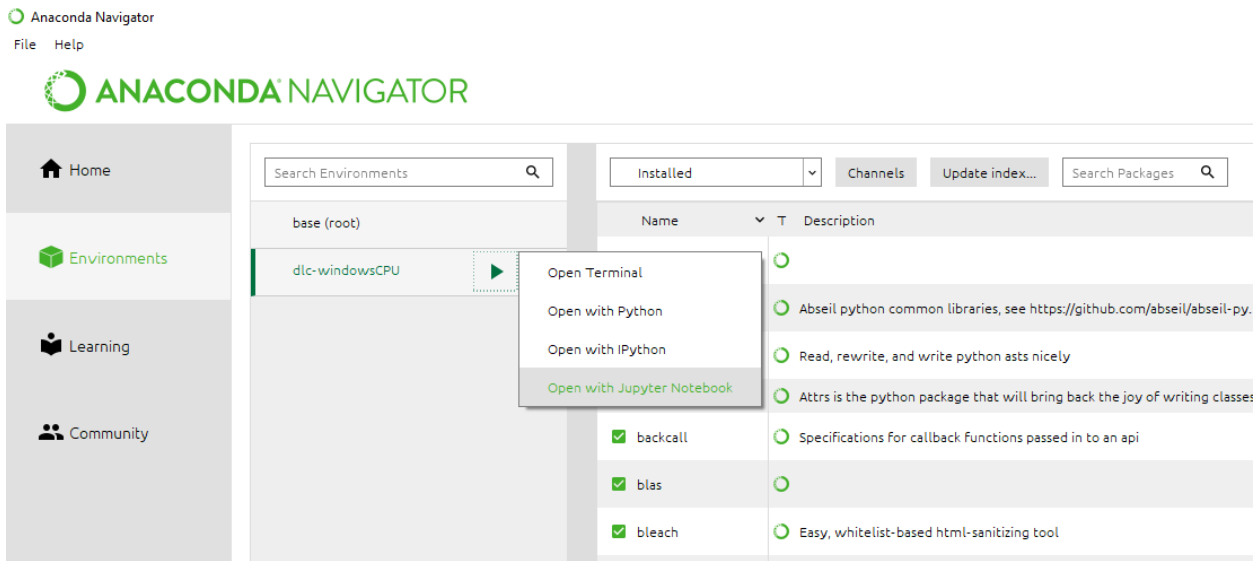


Figure 4. How to open DeepLabCut via Jupyter Notebook in Anaconda.

#### Part 4: Evaluate performance and correct select frames from novel trials

1. Import predicted 2D points (output of analyze\_xromm\_videos) into XMALab.
2. Evaluate networks' performance using built in XMALab tools (reprojection error, rigid body error, etc.).
3. Correct select frames where error is too high/performance is poor. **Important: Keep track of the specific frames you correct in each trial, and be sure to correct all markers in those frames.**
4. Create a corrected frame index csv file (Fig. 5; filename doesn't matter) in which column 1 is trial folder name, and columns 2+ are frame #'s of corrected frames.
5. Export 2D points for trials where frames were corrected, and **be sure that "corrected" is somewhere in the filename.**

#### Part 5: Augment training dataset and retrain

Once corrected points are exported and the frameindex.csv file is generated, you can proceed in the Jupyter notebook. It will take you through augmenting the training dataset (add\_frames), and retraining. You can iteratively perform these steps until performance is acceptable.

	A	B	C	D	E	F	G
1	trial01	111	222	333	220		
2	trial2	112	115	114			
3	trial04	47	223	559	17	59	
4							
5							
6							

Figure 5. Example contents and format of corrected frame index csv file.