If on a Mac: make sure you have XQuartz7 to enable the Graphic User Interface (GUI)

If on a PC: Use MobaXterm to enable the GUI

#*The following commands open the HPC Interactive server. Can also use ‘foxcroft.bowdoin.edu’ or ‘pauling.bowdoin.edu.’ My inputs are highlighted in yellow.*

(base) Druck125-03:~ jeschole$ ssh -Y jeschole@dover.bowdoin.edu

jeschole@dover.bowdoin.edu's password: \*insert your password here\*

+----------------------------------------------------------------------+

| Welcome to Bowdoin College |

|======================================================================|

| For information about our Linux HPC environment, please visit |

| http://hpc.bowdoin.edu/ |

+----------------------------------------------------------------------+

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Please be mindful that this machine is a shared resource. If you need

to run very intensive computational programs, please consider running

them on the HPC Grid instead. For more info, please peek at:

http://hpc.bowdoin.edu/

Thank you!

#*The following command opens the conditions the virtual environment that Dj has created for DeepLabCut on the HPC with the proper software packages updated to the proper software iteration*

[jeschole@dover ~]$ source /mnt/local/python-venv/dlc/bin/activate

#*This command opens ipython interface (shown by prompts ‘In [#]:’)*

(dlc) [jeschole@dover ~]$ ipython

Python 3.6.3 (default, Mar 20 2018, 13:50:41)

Type 'copyright', 'credits' or 'license' for more information

IPython 6.0.0 -- An enhanced Interactive Python. Type '?' for help.

#*open deeplabcut software in ipython*

In [**1**]: **import** **deeplabcut**

*#create a new project in DLC. In parentheses: (‘Project name’, ‘project user’ [‘video path file’], working\_directory=’folder path’)*

In [**2**]: deeplabcut.create\_new\_project('Example', 'jeschole', ['/mnt/research/hhorch/jeschole/videos\_for\_training/example\_vid.m4v'], working\_directory='/mnt/research/hhorch/jeschole/')

Created "/mnt/research/hhorch/jeschole/Example-jeschole-2019-07-10/videos"

Created "/mnt/research/hhorch/jeschole/Example-jeschole-2019-07-10/labeled-data"

Created "/mnt/research/hhorch/jeschole/Example-jeschole-2019-07-10/training-datasets"

Created "/mnt/research/hhorch/jeschole/Example-jeschole-2019-07-10/dlc-models"

Creating the symbolic link of the video

Created the symlink of /mnt/research/hhorch/jeschole/videos\_for\_training/example\_vid.m4v to /mnt/research/hhorch/jeschole/Example-jeschole-2019-07-10/videos/example\_vid.m4v

/mnt/research/hhorch/jeschole/Example-jeschole-2019-07-10/videos/example\_vid.m4v

Generated "/mnt/research/hhorch/jeschole/Example-jeschole-2019-07-10/config.yaml"

A new project with name Example-jeschole-2019-07-10 is created at /mnt/research/hhorch/jeschole and a configurable file (config.yaml) is stored there. Change the parameters in this file to adapt to your project's needs.

Once you have changed the configuration file, use the function 'extract\_frames' to select frames for labeling.

. [OPTIONAL] Use the function 'add\_new\_videos' to add new videos to your project (at any stage).

Out[**2**]: '/mnt/research/hhorch/jeschole/Example-jeschole-2019-07-10/config.yaml'

In [**3**]: config\_path = '/mnt/research/hhorch/jeschole/Example-jeschole-2019-07-10/config.yaml'

In [**4**]: deeplabcut.extract\_frames(config\_path)

Config file read successfully.

Do you want to extract (perhaps additional) frames for video: /mnt/research/hhorch/jeschole/videos\_for\_training/example\_vid.m4v ?

yes/noyes

Extracting frames based on kmeans ...

Kmeans-quantization based extracting of frames from 0.0 seconds to 97.06 seconds.

Extracting and downsampling... 3880 frames from the video.

3880it [00:08, 431.39it/s]

Kmeans clustering ... (this might take a while)

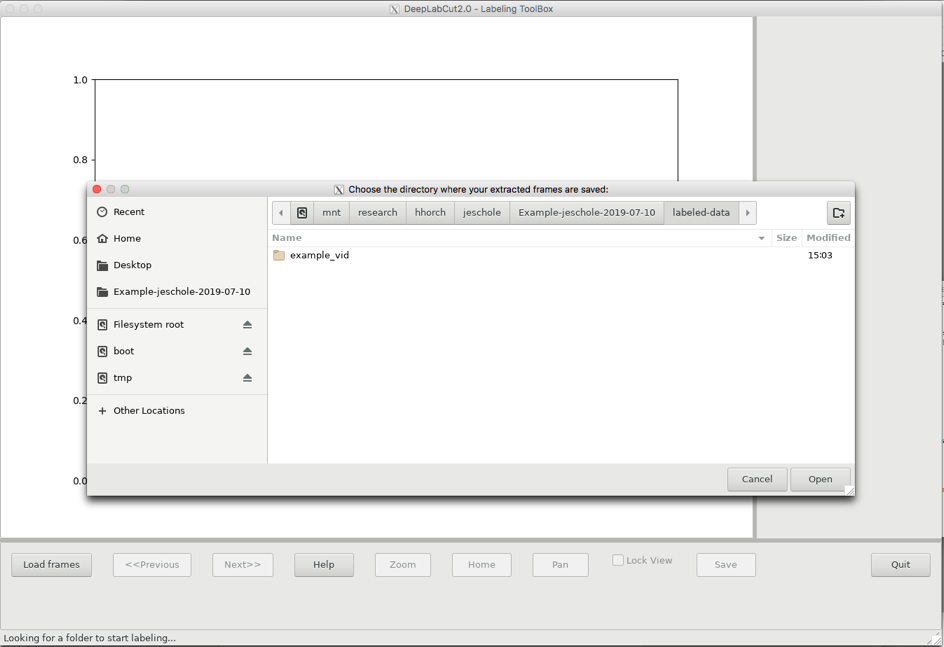
Frames were selected.

You can now label the frames using the function 'label\_frames' (if you extracted enough frames for all videos)

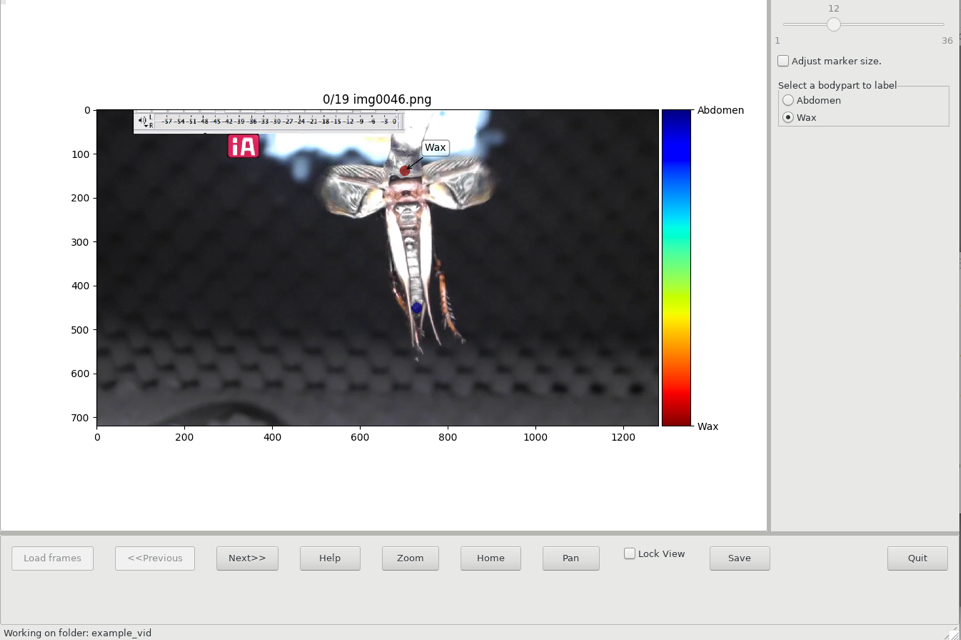
In [**5**]: deeplabcut.label\_frames(config\_path)

You can now check the labels, using 'check\_labels' before proceeding. Then, you can use the function 'create\_training\_dataset' to create the training dataset.

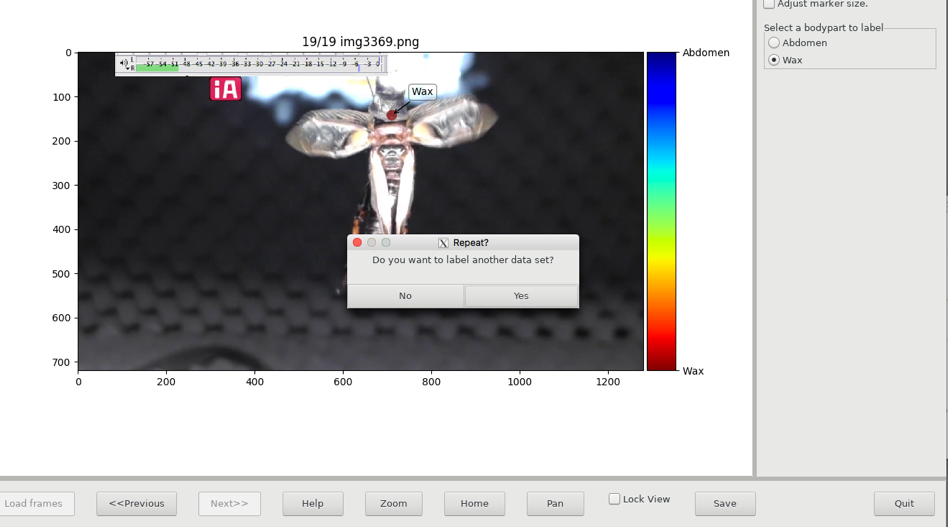
(Graphic User Interface (GUI) opens)

**

*Figure 1. Load frames from desired video*

**

*Figure 2. Assign labels to areas of interest*

**

*Figure 3. Save video and label next video or close GUI*

(Graphic User Interface (GUI) closes)

In [**6**]: deeplabcut.create\_training\_dataset(config\_path)

The training dataset is successfully created. Use the function 'train\_network' to start training. Happy training!

In [**8**]: quit

\*OPEN NEW TERMINAL WINDOW\*

(base) Druck125-03:~ jeschole$ ssh -Y jeschole@moosehead.bowdoin.edu

jeschole@moosehead.bowdoin.edu's password: \*insert your password here\*

Last login: Wed Jul 10 14:03:32 2019 from druck122c-01.campus.bowdoin.edu

+----------------------------------------------------------------------+

| Welcome to Bowdoin College |

|======================================================================|

| For information about our Linux HPC environment, please visit |

| http://hpc.bowdoin.edu/ |

+----------------------------------------------------------------------+

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Moosehead is setup to submit batch jobs to the HPC Grid.

For interactive jobs, please use dover, foxcroft, or pauling.

For more info, please peek at:

http://hpc.bowdoin.edu/

#*Change directories to the folder containing your scripts to submit to the HPC Grid*

[jeschole@moosehead ~]$ cd /mnt/research/hhorch/jeschole/Example-jeschole-2019-07-10/HPC\_Scripts

#*Change the modification of the shell script so it can be run by the HPC grid*

[jeschole@moosehead /mnt/research/hhorch/jeschole/Example-jeschole-2019-07-10/HPC\_Scripts]$ chmod u+x training\_script.sh

#*Submit the training script to the highest power gpu (gpu2080 or gpup100). Training should take 12-24 hours depending on how many iterations you ask it to complete*

[jeschole@moosehead /mnt/research/hhorch/jeschole/Example-jeschole-2019-07-10/HPC\_Scripts]$ qsub -l gpu2080=1 training\_script.sh

Your job 592272 ("training\_script.sh") has been submitted

#*Check the status of the project (if ‘r’ it is running, if ‘qw’ it is queued to be run in the grid)*

[jeschole@moosehead /mnt/research/hhorch/jeschole/Example-jeschole-2019-07-10/HPC\_Scripts]$ qstat

job-ID prior name user state submit/start at queue slots ja-task-ID

-----------------------------------------------------------------------------------------------------------------

592272 0.50500 training\_s jeschole qw 07/10/2019 15:24:39 1

[jeschole@moosehead /mnt/research/hhorch/jeschole/Example-jeschole-2019-07-10/HPC\_Scripts]$ chmod u+x evaluate\_script.sh

[jeschole@moosehead /mnt/research/hhorch/jeschole/Example-jeschole-2019-07-10/HPC\_Scripts]$ qsub -l gpu2080=1 evaluate\_script.sh

Your job 592272 ("evaluate\_script.sh") has been submitted

[jeschole@moosehead /mnt/research/hhorch/jeschole/Example-jeschole-2019-07-10/HPC\_Scripts]$ chmod u+x analyze\_script.sh

[jeschole@moosehead /mnt/research/hhorch/jeschole/Example-jeschole-2019-07-10/HPC\_Scripts]$ qsub -l gpu2080=1 analyze\_script.sh

Your job 592272 ("analyze\_script.sh") has been submitted

**Option1: Visualize returned analysis**

\*RETURN TO INTERACTIVE COMMAND WINDOW\*

[jeschole@dover ~]$ scl enable rh-python36 /bin/bash

[jeschole@dover ~]$ export IPYTHONDIR=/tmp

[jeschole@dover ~]$ xeyes

[jeschole@dover ~]$ ipython

Python 3.6.3 (default, Mar 20 2018, 13:50:41)

Type 'copyright', 'credits' or 'license' for more information

IPython 6.0.0 -- An enhanced Interactive Python. Type '?' for help.

In [**1**]: **import** **deeplabcut**

In [**2**]: config\_path= 'mnt/research/hhorch/jeschole/Example-jeschole-2019-07-10/config.yaml'

#*Create a video with the labeled plotted points to track and visualize labels across a video:*

In [**3**]: deeplabcut.create\_labeled\_video(config\_path, ['/mnt/research/hhorch/jeschole/Example-jeschole-2019-07-10/videos/example\_vid.m4v'], videotype='.m4v')

Analyzing all the videos in the directory

Starting % . ['/mnt/research/hhorch/jeschole/190710-Julie-2019-07-10/videos/']

Loading example\_vid.m4v and data.

False 0 1280 0 720

3955

Duration of video [s]: 97.51 , recorded with 39.98 fps!

Overall # of frames: 3898 with cropped frame dimensions: 1280 720

Generating frames and creating video.

100%|███████████████████████████████████████| 3898/3898 [00:41<00:00, 93.21it/s]

#*Filter the xy-coordinates and overall data from the analysis using an ARIMA model*

In [**4**]: deeplabcut.filterpredictions(config\_path, ['/mnt/research/hhorch/jeschole/Example-jeschole-2019-07-10/videos/example\_vid.m4v'])

Filtering with median model /mnt/research/hhorch/jeschole/Example-jeschole-2019-07-10/videos/example\_vid.m4v

#*Create plots of the trajectories of the labels from the analyzed video and plots of the coordinates over time*

In [**5**]: deeplabcut.plot\_trajectories(config\_path, ['/mnt/research/hhorch/jeschole/Example-jeschole-2019-07-10/videos/example\_vid.m4v'])

Starting % /mnt/research/hhorch/jeschole/Example-jeschole-2019-07-10/videos/example\_vid.m4v

Loading /mnt/research/hhorch/jeschole/Example-jeschole-2019-07-10/videos/example\_vid.m4v and data.

Plots created! Please check the directory "plot-poses" within the video directory

**Option2: Refine the label placement and networking training to increase labeling accuracy**

*#Extract frames that don’t fit in the p-value cutoff for accurate data and use kmeans clustering to select some frames to relabel*

In [**6**]: deeplabcut.extract\_outlier\_frames(config\_path, ['/mnt/research/hhorch/jeschole/Example-jeschole-2019-07-08/videos/example\_vid.m4v'])

network parameters: DeepCut\_resnet50\_190710Jul10shuffle1\_200000

Method jump found 3135 putative outlier frames.

Do you want to proceed with extracting 50 of those?

If this list is very large, perhaps consider changing the paramters (start, stop, epsilon, comparisonbodyparts) or use a different method.

yes/noyes

Frames from video example\_vid already extracted (more will be added)!

Loading video...

Duration of video [s]: 104.10731505095994 , recorded @ 39.97797847310465 fps!

Overall # of frames: 4162 with (cropped) frame dimensions:

Kmeans-quantization based extracting of frames from 0.0 seconds to 105.53 seconds.

Extracting and downsampling... 3135 frames from the video.

3135it [09:27, 5.01it/s]

Kmeans clustering ... (this might take a while)

Let's select frames indices: [4052, 4152, 40, 1915, 3560, 1558, 426, 1254, 2348, 3224, 2696, 1449, 1814, 635, 3272, 2149, 3093, 1598, 2243, 2900, 1568, 3211, 3514, 1408, 157, 1969, 1619, 1238, 1497, 2432, 675, 3837, 1062, 1481, 57, 2480, 3438, 495, 1798, 3775, 3069, 819, 2843, 31, 3733, 1657, 1854, 771, 3891, 1988]

New video was added to the project! Use the function 'extract\_frames' to select frames for labeling.

The outlier frames are extracted. They are stored in the subdirectory labeled-data\example\_vid.

Once you extracted frames for all videos, use 'refine\_labels' to manually correct the labels.

#*Open a GUI to move the labels that have been selected in the ‘extract\_frames’ command*

In [**7**]: deeplabcut.refine\_labels(config\_path)

Linux

Checking labels if they are outside the image

Closing... The refined labels are stored in a subdirectory under labeled-data. Use the function 'merge\_datasets' to augment the training dataset, and then re-train a network using create\_training\_dataset followed by train\_network!

#*Merge the newly labeled frames with the dataset from the initial set of labeled data*

In [**8**]: deeplabcut.merge\_datasets(config\_path)

Merged data sets and updated refinement iteration to 1.

Now you can create a new training set for the expanded annotated images (use create\_training\_dataset).

Repeat training steps until satisfied with the labeled outputs