

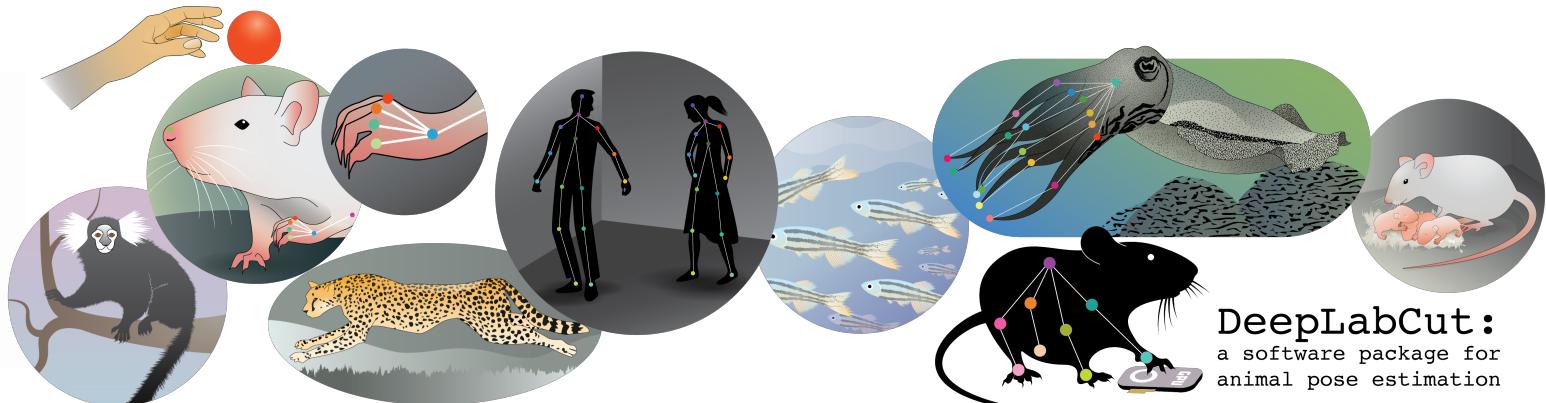
DeepLabCut workshop

Session 3: network evaluation & video analysis

16th and 17th of January 2020

Rowland Institute, Harvard University

Mackenzie Mathis
Alexander Mathis
Jessy Lauer



Chan
Zuckerberg
Initiative 

HARVARD
UNIVERSITY



DeepLabCut 2.0 workflow

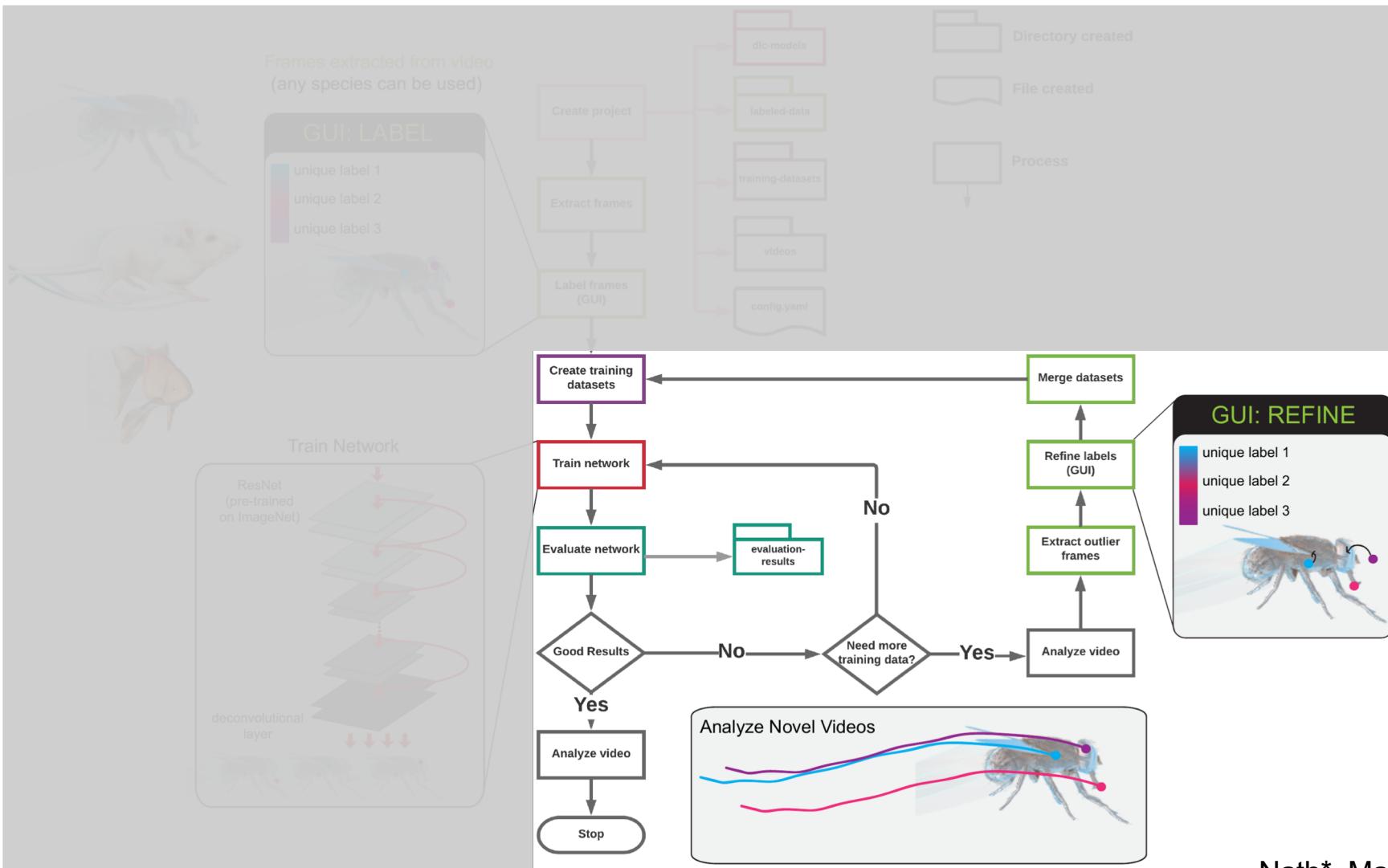


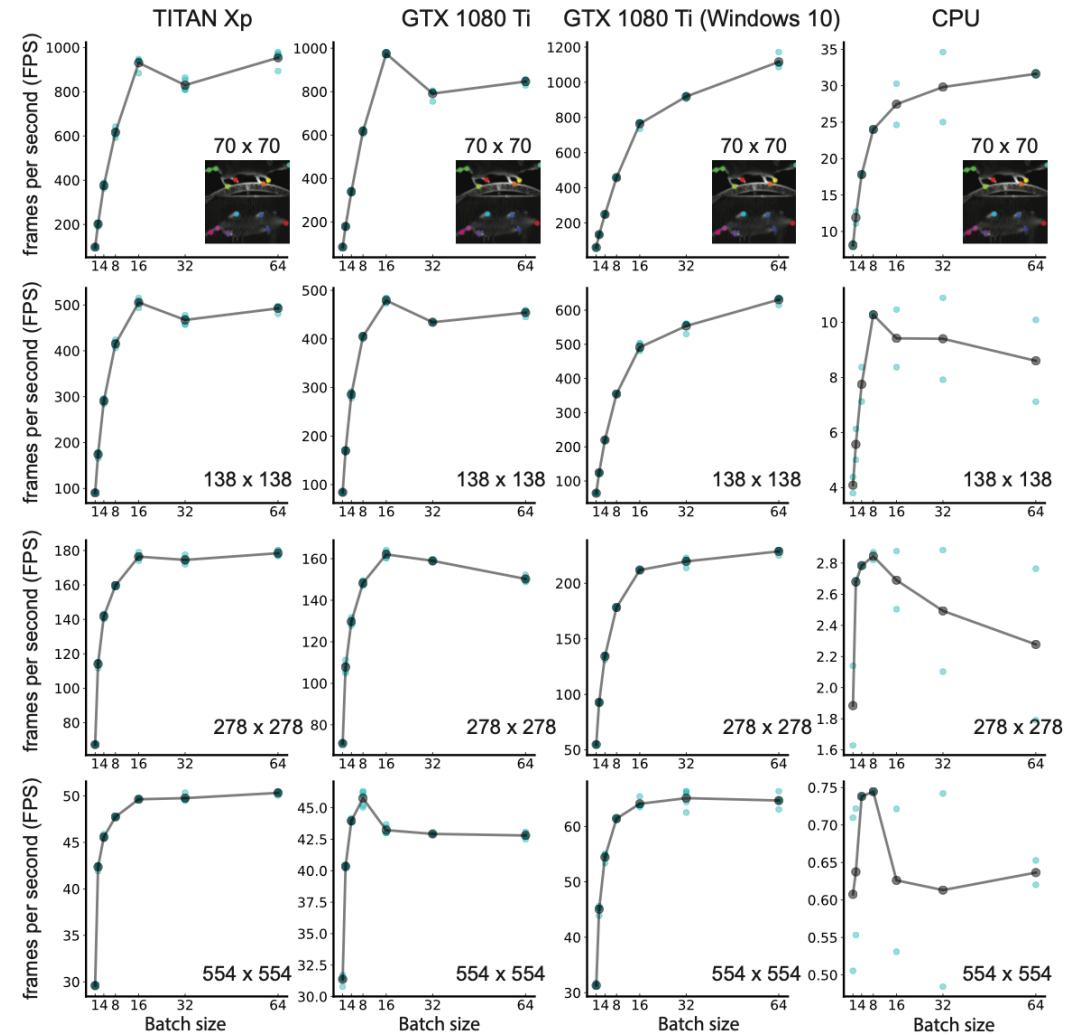
Table 1 | Summary of commands

Evaluate the trained network (Step 11)	<code>deeplabcut.evaluate_network(config_path)</code>
Video analysis and plotting results (Step 11)	<code>deeplabcut.analyze_videos(config_path, ['path of video 1 or folder', 'path of video2', ...])</code>
Video analysis and plotting results (Step 12)	<code>deeplabcut.plot_trajectories(config_path, ['path of video 1', 'path of video2', ...])</code>
Video analysis and plotting results (Step 13)	<code>deeplabcut.create_labeled_video(config_path, ['path of video 1', 'path of video2', ...])</code>
Refinement: extract outlier frames (Step 14)	<code>deeplabcut.extract_outlier_frames(config_path, ['path of video 1', 'path of video 2'])</code>
Refine labels (Step 15)	<code>deeplabcut.refine_labels(config_path)</code>
Combine datasets (Step 16)	<code>deeplabcut.merge_datasets(config_path)</code>

Video analysis

New videos need **not** be added to the config.yaml!

Set *batch_size* in the config.yaml!

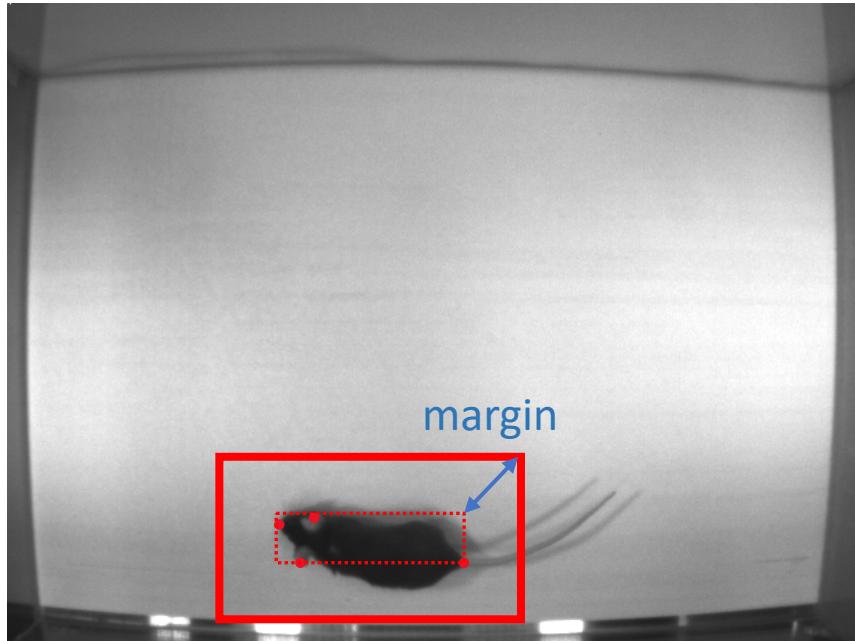


Dynamic cropping

Whether to perform dynamic cropping or not
state = True or False

Extent by which the cropping window
is expanded around the animal
margin (integer, in pixels)

```
deeplabcut.analyze_videos(path_config_file,[video],save_as_csv=True, dynamic=(True,.1,30))
```



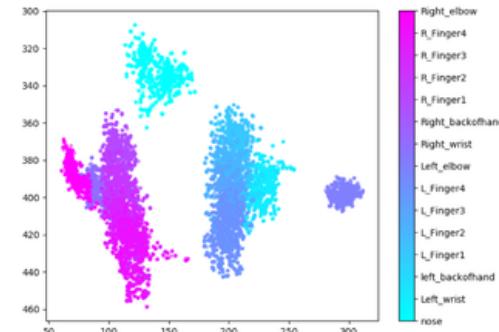
Threshold (probability) above which
body parts are confidently detected
 $0 < \text{detectionthreshold} \leq 1$

Trajectory visualization

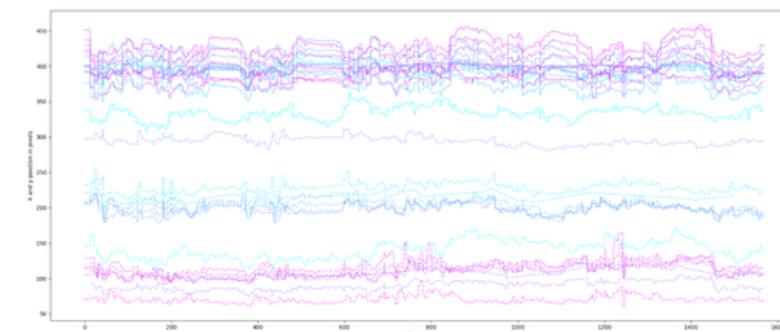
```
deeplabcut.plot_trajectories(config_path,['fullpath/analysis/project/videos/reachingvideo1.avi'])
```



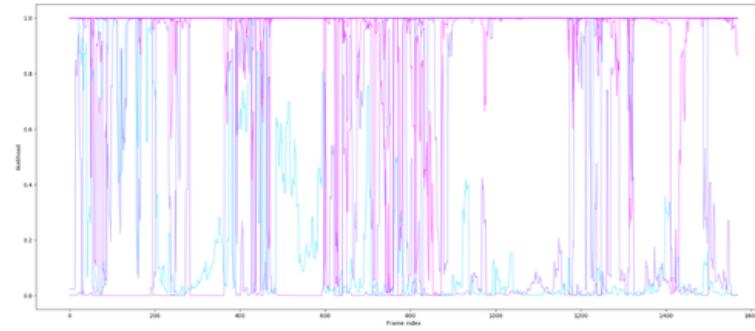
Body parts plotted in space
(over all the frames)



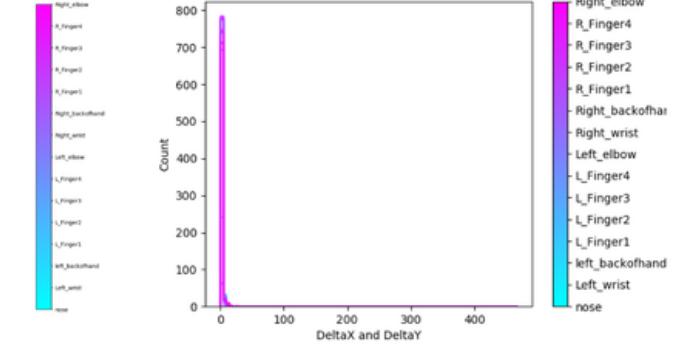
All body parts across time (frames).
solid lines are Y and dashed lines are X



Every body part likelihood over time
(over all the frames)



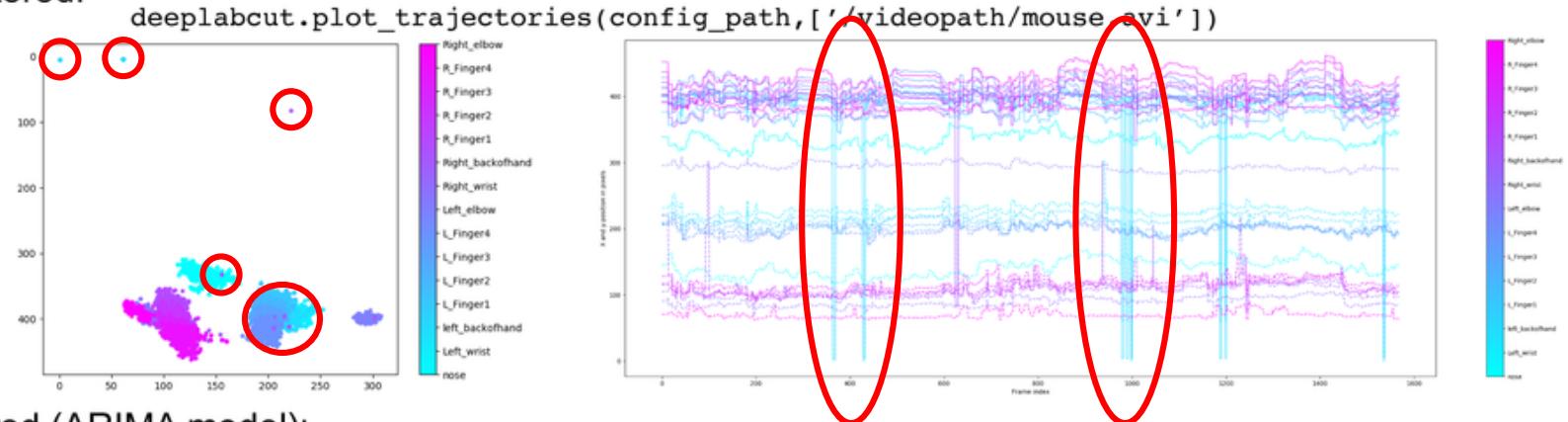
Consecutive coordinate differences
(low values = minimal jumps across frames)



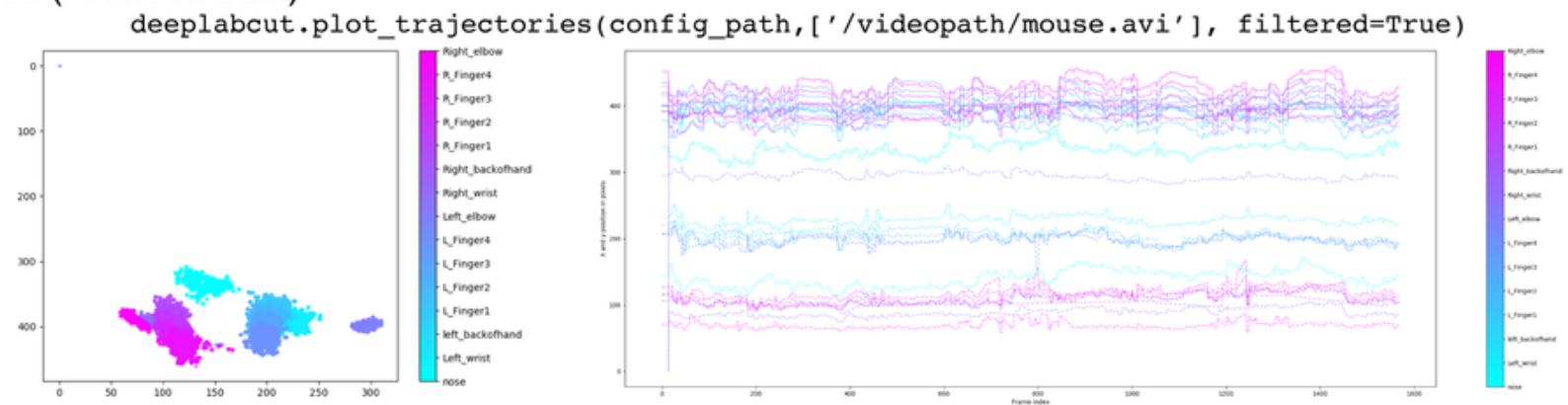
Trajectory filtering

```
deeplabcut.filterpredictions(config_path,['/videopath/mouse.avi'], p_bound=0.1, ARdegree=5, MAdegree=1)
```

Unfiltered:



Filtered (ARIMA model):



Labeled video

```
deeplabcut.create_labeled_video(config_path,['fullpath/afolderofvideos'], videotype='.mp4', filtered=True)
```

```
deeplabcut.create_labeled_video(config_path,['fullpath/afolderofvideos'], videotype='.mp4', draw_skeleton=True)
```

```
deeplabcut.create_labeled_video(config_path,['fullpath/afolderofvideos'], videotype='.mp4', trailpoints=10)
```

```
deeplabcut.create_labeled_video(config_path,['fullpath/afolderofvideos'], save_frames=True/False)
```

Active learning

- If there are errors in analyzed videos, you can correct those manually and re-train!
- Note you do not need to correct all errors (presumably some are fixed due to generalization)

Active learning

```
deeplabcut.extract_outlier_frames(config_path, ['videofile_path'])
```

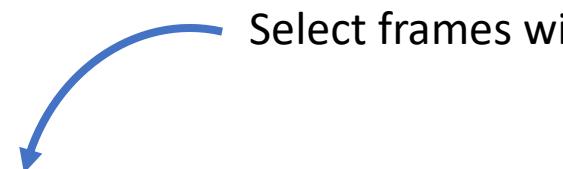
- Key arguments

`outlieralgorithm: 'fitting', 'jump', or 'uncertain'`

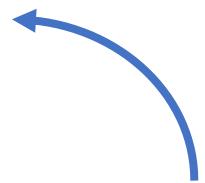


Select frames deviating from
a statistical model fit to the data

Select frames with jumps between consecutive frames $> \epsilon_{pixel}$



Select frames with confidences $< p_{bound}$



Random sampling

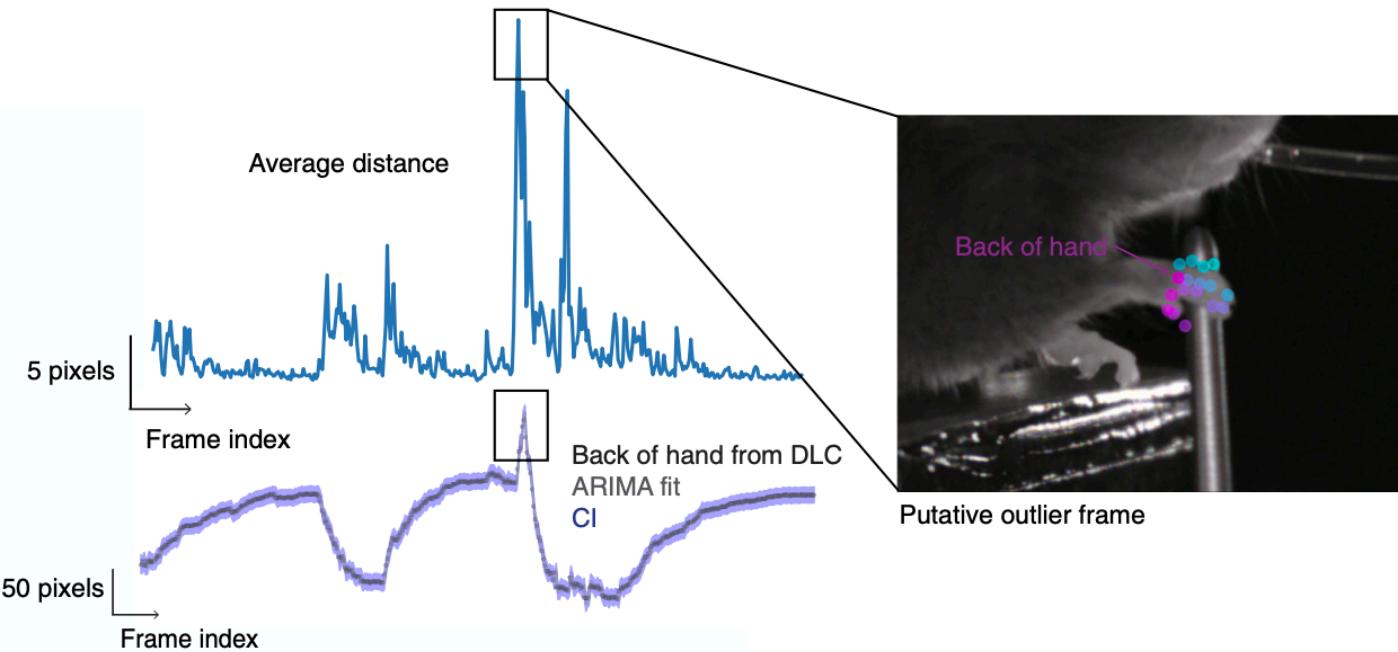
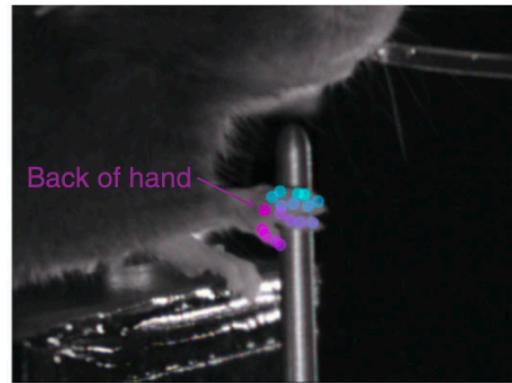
`extractionalgorithm='uniform'`

Clustering

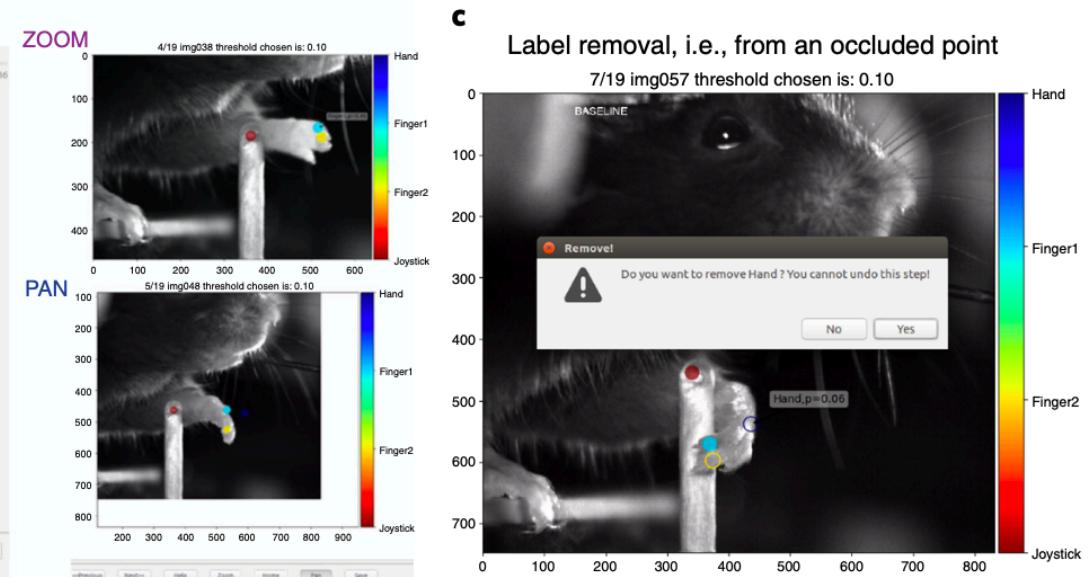
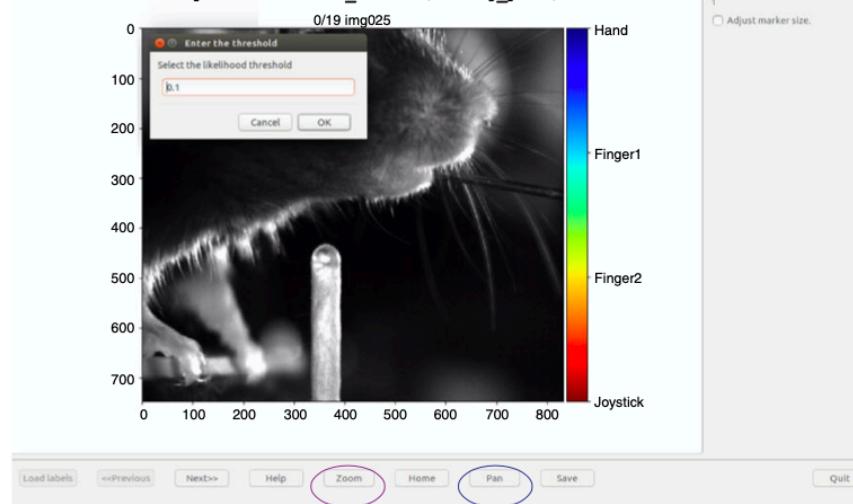
`extractionalgorithm='k-means'`

a

Identification of outlier frames

**b**

Refinement of DeepLabCut-applied label location(s)

`deeplabcut.refine_labels(config_path)`

Next...

- Merge the data sets and train again!

```
deeplabcut.merge_datasets(config_path)
```

- The weights will be reinitialized from ImageNet weights! (because w.l.o.g. e.g. # of body parts could have changed).
- You can manually link the *init_weights* in *pose_cfg.yaml* to the snapshot from previous iteration before starting to train

Scaling workflow

```
9  import os
10
11 import deeplabcut
12
13 def getsubfolders(folder):
14     ''' returns list of subfolders '''
15     return [os.path.join(folder,p) for p in os.listdir(folder) if os.path.isdir(os.path.join(folder,p))]
16
17 project='ComplexWheelD3-12-Fumi-2019-01-28'
18
19 shuffle=1
20
21 prefix='/home/alex/DLC-workshopRowland'
22
23 projectpath=os.path.join(prefix,project)
24 config=os.path.join(projectpath,'config.yaml')
25
26 basepath='/home/alex/BenchmarkingExperimentsJan2019' #data'
27
28 ...
29
30 Imagine that the data (here: videos of 3 different types) are in subfolders:
31     /January/January29 ..
32     /February/February1
33     /February/February2
34
35     etc.
36
37 ...
38
39 subfolders=getsubfolders(basepath)
40 for subfolder in subfolders: #this would be January, February etc. in the upper example
41     print("Starting analyze data in:", subfolder)
42     subsubfolders=getsubfolders(subfolder)
43     for subsubfolder in subsubfolders: #this would be Febuary1, etc. in the upper example...
44         print("Starting analyze data in:", subsubfolder)
45         for vtype in ['.mp4','.m4v','mpg']:
46             deeplabcut.analyze_videos(config,[subsubfolder],shuffle=shuffle,videotype=vtype,save_as_csv=True)
47
48
```