LabGym user guide (v1.9)

1. Installation of *LabGym* (https://github.com/umyelab/LabGym)[1]

a. Download Python3 [2] (version >= 3.9.7) from its official website (https://www.python.org/downloads/).

   Note: recommend not to download the latest version of Python3 as some Python libraries used in *LabGym* might be updated yet to be compatible with the latest version of Python3.

b. Open the Terminal (Mac) or Command Prompt (Windows)

c. To install *LabGym*, in the Terminal / Command Prompt, type:

   ```
   python3 -m pip install LabGym
   ```

d. After *LabGym* is installed, if you want to use its Detector module, the Detectron2 [3] (https://github.com/facebookresearch/detectron2) needs to be installed separately. To do so, type:

   ```
   python3 -m pip install 'git+https://github.com/facebookresearch/detectron2.git'
   ```

   for Windows user, you may type:

   ```
   python3 -m pip install git+https://github.com/facebookresearch/detectron2.git
   ```

   If encounter issues when installing Detectron2, please refer to its documentation page (https://detectron2.readthedocs.io/en/latest/tutorials/install.html).

2. Initiate the graphical user interface (GUI) of *LabGym* for each use

a. Open the Terminal (Mac) or Command Prompt (Windows)

b. To activate Python3 interactive shell, in the Terminal / Command Prompt, type:

   ```
   python3
   ```

c. After the Python3 interactive shell is activated, in the Terminal / Command Prompt, type:

```
from LabGym import gui
```

d. Then in the Terminal / Command Prompt, type:

```
gui.gui()
```

Now the GUI of *LabGym* is initiated and is ready to use.

3. Tips on using the GUI of *LabGym*

The GUI of *LabGym* consists of 3 major modules: '***Preprocessing Module***', '***Training Module***', '***Analysis Module***'. *Put your mouse cursor above each button to see a detailed description for it.*

3.1. '***Preprocessing Module***'

Preprocess your videos to make them fit your analysis goal better.

3.1.1. '***Preprocess Videos***'

Use this to enhance contrast of the videos, crop frames to exclude unnecessary regions, or trim the videos to only keep necessary time windows.

(Button) 'Select the video(s) for preprocessing'

(Button) 'Select a folder to store the processed videos'

(Button) 'Specify whether to enter time windows to form a trimmed video'

(Button) 'Specify whether to crop the video frames'

(Button) 'Specify whether to enhance the contrast in videos'

(Button) 'Start to preprocess the videos'

3.2. '*Training Module*'

The first three buttons in this module can be used to train Detectron2-based (https://github.com/facebookresearch/detectron2)[3] Detectors, which is the way to teach *LabGym* to recognize animals / objects of your interest. There are two detection methods in *LabGym*. One is the Detector-based, which is slow but versatile in any recording settings. It is also useful for distinguish individuals when they have close body contact. The other detection method is background subtraction-based, which is much faster but requires the background to be static and illumination to be stable in a video. The background subtraction-based method is your first choice if the videos meet the criteria. Note that this method cannot distinguish individuals when they have body contact and may not be useful for analyzing interactive behaviors when the individual identify is critical.

The last three buttons can be used to train Categorizers, which is the way to teach *LabGym* to recognize behaviors that are defined by you.

3.2.1. '*Generate Image Examples*'

To train Detectors, you need to provide training images in which the outlines of animals / objects of your interest are precisely labeled. But first, you need to generate some image examples (extract some frames from videos).

(Button) 'Select the video(s) to generate image examples'

- (Pop-up option) '(Optional) resize the frames?'

  The resizing will keep the original "width / height" ratio.

(Button) 'Select a folder to store the generated image examples'

(Button) 'Specify when generating image examples should begin (unit: second)'

(Button) 'Specify how long generating examples should last (unit: second)'

(Button) 'Specify how many frames to skip when generating two consecutive images'

(Button) 'Start to generate image examples'

3.2.2. '*Annotate images with Roboflow*'

After images are generated, you need to annotate them to train a Detector. The annotation is to label the outline of the animals / objects of your interest. You can do this by using online annotation tools such as Roboflow (https://roboflow.com) or CVAT (https://www.cvat.ai) or VGG Image Annotator (https://www.robots.ox.ac.uk/~vgg/software/via/). Roboflow is recommended because it is easy to use. When annotated images, make sure to select "Instance Segmentation" for the annotation type. When export the annotation file, make sure to select "COCO instance segmentation" format, which will be a '*.json' file.

3.2.3. '*Train Detectors*'

Trained Detectors will be stored in the Detector list in *LabGym* and ready to use.

(Button) 'Select the folder containing all the training images'

(Button) 'Select the *.json annotation file'

(Button) 'Specify the inferencing framesize for the Detector to train'

It determines the speed-accuracy trade-off of Detector performance. You can start with small size such as 256 and increase it if the accuracy is not ideal. Size that is smaller than 192 or larger than 1024 is not recommended for general scenarios. If there are many animals / objects to detect in one frame, and they are relatively small compared with the entire frame, you may use larger inferencing size such as 640 or higher. If there is only one animal / object and it occupies large area of the frame, use smaller size such as 320 for faster processing speed.

(Button) 'Specify the iteration number for the Detector training'

(Button) 'Train the Detector'

(Button) 'Delete a Detector'

3.2.4. '*Generate Behavior Examples*'

To teach *LabGym* to recognize behaviors that are defined by you, you need to use sorted behavior examples to train a Categorizer. But first, you need to generate some behavior examples to sort them.

(Button) 'Specify the kind of behavior examples to generate'

There are 2 kinds of interactive behaviors that *LabGym* can generate examples and identify. "Interactive basic" is faster in analysis and is useful in the scenario when you only care about the interactive behavior events but not the behaviors of individuals. It classifies the behaviors of all animals / objects in a frame as an interactive pair / group and does not track individuals. The behavioral quantifications for this kind are behavior count, behavior latency and behavior duration. The "Interactive advance" is slower in analysis than "basic", but it distinguishes and tracks individuals all the time even during close body contact. The behavioral quantifications for this kind are behavior count, behavior latency, behavior duration, 3 length parameters, 3 areal parameters, and 4 locomotion parameters.

<u>(Button) ‘Select the video(s) to generate behavior examples’</u>

- (Pop-up option) ‘(Optional) resize the frames?’

  The resizing will keep the original “width / height” ratio.

<u>(Button) ‘Select a folder to store the generated behavior examples’</u>

<u>(Button) ‘Specify when generating behavior examples should begin (unit: second)’</u>

- (Pop-up option) ‘Illumination shifts?’

  This option is only for background subtraction-based detection method. It is useful when there are sudden bright-to-dark or dark-to-bright illumination transitions in videos. If choose ‘Yes’, there will be 3 options to specify the beginning time: '*Automatic (for light on and off)*', '*Decode from filenames: _bt_*', and '*Enter a time point*'. If choose ‘No’, only the latter two options can be chosen.

  '*Automatic (for light on and off)*' is for videos involving lighting on and off (e.g., optogenetics) and the first time point of illumination change will be automatically detected and used as the beginning time to generate behavior examples.

  '*Decode from filenames: _bt_*' can be used if multiple videos are selected and the beginning time for each video is different. You need to add a code tag ‘_bt_’ (‘b’ stands for ‘beginning time’ and ‘t’ should be number) into the file name of each video to let *LabGym* decode the beginning time. For example, suppose the video filename is ‘A.avi’ and you want the beginning time to be at the 12.35 second, you can rename the video file to ‘A_b12.35.avi’ (the ‘_’ is unnecessary if it is at the end of file name).

'*Enter a time point*' can be used if only one video is selected, or multiple videos share the same beginning time.

(Button) ‘Specify how long generating examples should last (unit: second)’

(Button) ‘Specify the number of animals in a video’

There are two options: ‘*Decode from filenames: _nn_*' and *'Enter the number of animals*'.

*'Decode from filenames: _nn_*' can be used if multiple videos are selected and the number of animals in each video is different. A code tag ‘_nn_’ (the first ‘n’ stands for ‘number of animals’ and the second ‘n’ should be an integer) needs to be added into the file name to let *LabGym* decode the animal number. For example, suppose the video file name is ‘A.avi’ and the number of animals in this video is 8, rename the video file to ‘A_n8.avi’ (the last ‘_’ is unnecessary if it is at the end of file name). If there are multiple kinds of animals / objects in the videos to detect, you can put in multiple ‘_nn_’ code. The order should be consistent with that of the ‘animal / object name’ in the Detector to use (when you select a Detector, the animal / object name will display in order, and you can also specify which ones to detect).

'*Enter the number of animals*' can be used if only one video is selected, or multiple videos share the same animal number.

(Button) ‘Specify the method to detect animals or objects’

There are two options: ‘*Subtract background*’ and ‘*Use trained Detectors*’

‘*Subtract background*’ is the first choice for videos in which the background is static, the illumination is stable overtime, and the total behavior events are more important than animal IDs, because this method is fast and accurate in such videos but cannot distinguish entangled animals and the IDs might switch after they re-separate. When using this method, you need to specify the scenarios of their experiments: '*Animal brighter than background*', '*Animal darker* *than background*', or '*Hard to tell*'. A pop-up option '(Optional) load existing background?' will need to be specified. *LabGym* will output the extracted backgrounds (as images) for each video processed. Therefore, this pop-up option can be used to save the step of background extraction if the background for this video has already been extracted and output. You also need to specify whether the illumination in videos is unstable. If the illumination is very stable overtime, choose 'No' to increase the processing speed. Finally, you need to specify a time window for background extraction. An appropriate time window for background extraction should be a period (typically 10~60 seconds) during which the animals move around. This time window should be as short as possible for increasing processing speed. There are 3 options to specify this time window: (1) '*Use the entire duration of a video*': not recommended if processing speed is critical. (2) '*Decode from filenames: _xst_ and _xet_*': can be used if multiple videos are selected and the time window for background extraction is different for each video. Two code tags '_xst_ and _xet_' ('xs' stands for 'extraction start time', 'xe' stands for 'extraction end time', and 't' should be an integer) need to be added into the file name of each video to let *LabGym* decode the time window. For example, suppose the video file name is 'A.avi' and the time window for background extraction is from the 25th second to the 47th second, rename the video file to 'A_xs25_xe47.avi' (the '_' is unnecessary if it is at the end of file name). (3) '*Enter two time points*': can used if only one video is selected, or multiple videos share the same time window for background extraction.

'*Use trained Detectors*' is useful in any kind of videos or experimental settings. It is also useful for differentiate individual animals when they entangle. The only caveat of this method is slow.

<u>(Button) 'Specify the number of frames for an animation / pattern image'</u>

The animations and their paired pattern images in the behavior examples spans a duration (the number of frames, an integer) defined by you, which should approximate the duration of a behavior episode. This duration needs to be the same across all the behavior examples that are used to train one Categorizer. If the duration of different behavior episode is different, use the longest one.

<u>(Button) ‘Specify how many frames to skip when generating two consecutive behavior examples’</u>

LabGym generates a pair of behavior example which spans a user-defined duration at a frame. If two consecutively generated examples are too close in time, say, one generated at the 10^{th} frame and the other at 12^{th} frame, suppose their duration is 10 frames, they will have 8 overlapping frames. These two behavior examples are too similar and will impair the training efficiency when training a Categorizer. Generating too many such similar examples will also make the example sorting laborious. Therefore, you can choose to skip certain frames between the two consecutively generated examples. A practical recommendation to achieve a balance between getting the perfect examples that just span the behavior episode and reducing the labor is to set this number as the half of the duration for a behavior episode.

<u>(Button) ‘Start to generate behavior examples’</u>

- (Pop-up option) 'Including background?'

  Choose ‘No’ if the background information is behavior irrelevant.

- (Pop-up option) 'Including body parts?'

  Choose ‘Yes’ if the motion pattern of individual body parts such as limbs or noses are critical for behavioral identification. If choose ‘Yes’, a ‘STD’ need to be entered, which should be an integer between 0 and 255. The STD value decides the threshold to show the how many ‘motion pixels’ of the body parts in the pattern images. Lager STD, less motion pixels will be shown.

3.2.5. '*Sort Behavior Examples*'

The behavior examples need to be sorted into different folders according to the behavior types they belong to. This is the way how you teach *LabGym* the behaviors defined by you. Use this to sort behavior examples in an easier way.

<u>(Button) 'Select the folder that stores unsorted behavior examples'</u>

<u>(Button) 'Select the folder to store the sorted behavior examples'</u>

<u>(Button) 'Enter the behavior names and corresponding shortcut keys'</u>

<u>(Button) 'Sort behavior examples'</u>

3.2.6. '*Train Categorizers*'

There are two parts: one is for preparing the training examples; the other is for using the prepared training examples to train a Categorizer. The Categorizers are the core of *LabGym*, which identifies behaviors defined by you during analysis. Trained Categorizers will be stored in the Categorizer list in *LabGym* and ready to use.

<u>(Button) 'Select the folder that stores the sorted behavior examples'</u>

<u>(Button) 'Select a new folder to store all the prepared behavior examples'</u>

- (Pop-up option) 'Resize the frames?'

During the training, the behavior examples will be resized to the input frame size of the Categorizer. You can choose to downsize the frame / image size at this step to increase the training speed. The targeted frame / image size should not be smaller than the targeted input size of the Categorizers to train.

- (Pop-up option) 'Background-free animations?'

  You need to specify whether the animations used for training include the background.

<u>(Button) ‘Start to prepare the training examples’</u>

<u>(Button) ‘Specify the type / complexity of the Categorizer to train’</u>

- (Pop-up option) 'Categorizer types'

  If the processing speed is critical to you, choose the Categorizer with only Pattern Recognizer. If you want to maximize the analysis accuracy and care less about the analysis speed, choose the Categorizer with both Animation Analyzer and Pattern Recognizer.

- (Pop-up option) 'Complexity level'

  There are 7 complexity levels (1~7, from simpler to more complex) of either Animation Analyzer or Pattern Recognizer. You can start with simpler levels and go more complex if the accuracy is not ideal. If the Animation Analyzer is gray scale (animal / object color is behavior irrelevant), you can make the complexity levels of Pattern Recognizer a little higher than that of Animation Analyzer because the Pattern Recognizer analyzes pattern images containing additional color information, which is indicates the temporal sequence of a behavior.

- (Pop-up option) Behavior kind'

  You need to indicate which kind of behavior examples are used for training the Categorizer. For “Interact basic” and “Interact advance” behavior examples, you can find the kind information in the filenames of the behavior examples. “_itbs_” is “Interact basic” and “itadv” is “Interact advance”.

(Button) ‘Specify the input shape for Animation Analyzer / Pattern Recognizer’

You can start with the smaller input frame / image sizes such as 16 or 32. If there are many detailed appearance features in the frames that are behavior relevant, go larger size such as 64 or 96 or even 128. And always go deeper (increase complexity level) first, rather than go wider (increase input frame / image size) first.

- (Pop-up option) 'Grayscale Animation Analyzer?'

  Choose ‘Yes’ if the color of animals is behavior irrelevant.

(Button) ‘Specify the number of frames for an animation / pattern image’

(Button) ‘Select the folder that stores all the prepared training examples’

- (Pop-up option) 'Background-free animations?'

  Specify whether the animations in the behavior examples include background.

- (Pop-up option) 'Body parts in pattern images?'

  Specify whether the pattern images in the behavior examples include body parts. If choose ‘Yes’, you need to enter the STD. The value of STD should match that of the generated patten images. This information can be found in the filenames of the generated pattern images (‘xxx_stdxx.jpg’) if you chose including body parts and specified a STD value when generating these behavior examples.

(Button) ‘Specify the methods for data augmentation’

Briefly, '*rotation*' will rotate the animal blobs; '*flipping*' will flip the animal blobs; '*brightening*' and '*dimming*' will increase and decrease the brightness of the animal blobs; '*shearing*' will distort the animal blobs; '*rescaling*' will change the width / height ratio of the animal blobs; '*deletion*' will delete one or two frames in the animations (mimic the scenario in which animals are not detected in one or two frames during analysis).

- (Pop-up option) 'Use default augmentation?'

  Default augmentation will use '*rotation*', '*flipping*', '*brightening*' and '*dimming*'.

- (Pop-up option) 'Augment validation data?'

  If the total number of behavior example pairs used for training a Categorizer is less than 1,000 before augmentation, choose "Yes" for this option.

(Button) 'Select a folder to export training reports'

(Button) 'Train the Categorizer'

(Button) 'Delete a Categorizer'

3.3. '*Analysis Module*'

Automatically track animals / objects of your interest, identify and quantify their behaviors in videos, and automatically mine the analysis results to display the data details that show statistically significant differences among groups of your selection.

3.3.1. '*Analyze Behaviors*'

Will output a raster plot for all behavior events for all videos, an annotated video copy for each video, various spreadsheets storing quantification results for each selected behavior parameter.

(Button) 'Select a Categorizer for behavior classification'

Choose a Categorizer for behavior classification in analysis, or just let *LabGym* track the animals and calculate their motion parameters and body kinematics. If the former, you need to specify a 'Uncertainty level' for the Categorizer. This number in percentage determines the threshold for the Categorizer to output an 'NA' for behavioral classification. For example, when the probabilities of behavior A, B, and C are 60%, 10%, and 30%, respectively. If the uncertain level is set to be 31, the behavior classification will be 'NA', as the uncertainty level exceeds the difference between probability of the highest-likely behavior (A, 60%) and the second highest-likely behavior (C, 30%), which is 30%. Setting uncertainty level can reduce the possible false positives in behavior classification since ambiguous classification will be output as an 'NA'. If choose to not classify behaviors, you need to specify a time window for calculating the motion parameters and body kinematics.

(Button) 'Select the video(s) for behavior analysis'

- (Pop-up option) '(Optional) resize the frames?'

Downsizing the frame is **highly recommended**, which will exponentially increase the analysis speed. The resizing will keep the original "width / height" ratio. The analysis accuracy will not decline if the animal size after downsizing is still larger than the input size of the Categorizer used for analysis. For example, suppose the original size of a video frame is 1000 x 500, the size of an animal is approximately 1/4 to that of a frame, and input size of the Categorizer is 48 x 48. After downsizing the video frames to 500 x 250, the animal size is roughly 125 x 63 and still is larger than the input size of the Categorizer (48 x 48). In this scenario, the analysis accuracy will not decline since the animal blob will be downsized to 48 x 48 anyway when input to the Categorizer for analysis.

<u>(Button) 'Select a folder to store the analysis results'</u>

<u>(Button) 'Specify the method to detect animals or objects'</u>

There are two options: '*Subtract background*' and '*Use trained Detectors*'

'*Subtract background*' is the first choice for videos in which the background is static, the illumination is stable overtime, and the total behavior events are more important than animal

IDs, because this method is fast and accurate in such videos but cannot distinguish entangled animals and the IDs might switch after they re-separate. When using this method, you need to specify the scenarios of their experiments: '*Animal brighter than background*', '*Animal darker than background*', or '*Hard to tell*'. A pop-up option '(Optional) load existing background?' will need to be specified. *LabGym* will output the extracted backgrounds (as images) for each video processed. Therefore, this pop-up option can be used to save the step of background extraction if the background for this video has already been extracted and output. You also need to specify whether the illumination in videos is unstable. If the illumination is very stable overtime, choose 'No' to increase the processing speed. Finally, you need to specify a time window for background extraction. An appropriate time window for background extraction should be a period (typically 10~60 seconds) during which the animals move around. This time window should be as short as possible for increasing processing speed. There are 3 options to specify this time window: (1) '*Use the entire duration of a video*': not recommended if processing speed is critical. (2) '*Decode from filenames: _xst_ and _xet_*': can be used if multiple videos are selected and the time window for background extraction is different for each video. Two code tags '_xst_ and _xet_' ('xs' stands for 'extraction start time', 'xe' stands for 'extraction end time', and 't' should be an integer) need to be added into the file name of each video to let *LabGym* decode the time window. For example, suppose the video file name is 'A.avi' and the time window for background extraction is from the 25^{th} second to the 47^{th} second, rename the video file to 'A_xs25_xe47.avi' (the '_' is unnecessary if it is at the end of file name). (3) '*Enter two time points*': can used if only one video is selected, or multiple videos share the same time window for background extraction.

'*Use trained Detectors*' is useful in any kind of videos or experimental settings. It is also useful for differentiate individual animals when they entangle. The only caveat of this method is slow.

(Button) ‘Specify when the analysis should begin (unit: second)’

- (Pop-up option) ‘Illumination shifts?’

Specify whether there are sudden bright-to-dark or dark-to-bright illumination transitions. If choose ‘Yes’, there will be 3 options to specify the beginning time: '*Automatic (for light on and off)*', '*Decode from filenames: _bt_*', and '*Enter a time point*'. If choose ‘No’, only the latter two options can be chosen.

'*Automatic (for light on and off)*' is for videos involving lighting on and off (e.g., optogenetics) and the first time point of illumination change will be automatically detected and used as the beginning time to generate behavior examples.

'*Decode from filenames: _bt_*' can be used if multiple videos are selected and the beginning time for each video is different. You need to add a code tag ‘_bt_’ (‘b’ stands for ‘beginning time’ and ‘t’ should be number) into the file name of each video to let *LabGym* decode the beginning time. For example, suppose the video filename is ‘A.avi’ and the user wants the beginning time to be at the 12.35 second, the user can rename the video file to ‘A_b12.35.avi’ (the ‘_’ is unnecessary if it is at the end of file name).

'*Enter a time point*' can be used if only one video is selected, or multiple videos share the same beginning time.

(Button) ‘Specify the analysis duration (unit: second)’

(Button) ‘Specify the number of animals in a video’

There are two options: ‘*Decode from filenames: _nn_*' and '*Enter the number of animals*'.

'*Decode from filenames: _nn_*' can be used if multiple videos are selected and the number of animals in each video is different. A code tag ‘_nn_’ (the first ‘n’ stands for ‘number of animals’ and the second ‘n’ should be an integer) needs to be added into the file name to let *LabGym* decode the animal number. For example, suppose the video file name is ‘A.avi’ and the number of animals in this video is 8, rename the video file to 'A_n8.avi' (the last '_' is unnecessary if it is at the end of file name). If there are multiple kinds of animals / objects in the videos to detect, you can put in multiple '_nn_' code. The order should be consistent with that of the 'animal / object name' in the Detector to use (when you select a Detector, the animal / object name will display in order, and you can also specify which ones to detect).

'*Enter the number of animals*' can be used if only one video is selected, or multiple videos share the same animal number.

<u>(Button) 'Select the behaviors for annotations and plots'</u>

- (Pop-up option) 'Specify colors for behaviors?'

Specify a color to represent a behavior category in the annotated videos and the raster plot for behavior events. In the annotated videos, the value of % confidence of behavior categorization will be shown; in the raster plot, the color intensity indicates the value of % confidence, from 0% of the color intensity (clear) indicating 0% of the confidence, to 100% of the color intensity indicating 100% of the confidence. If choose not to specify the colors, *LabGym* will use the default colors to represent the behaviors.

- (Pop-up option) 'Legend in video?'

Specify whether to show the legend of behavior names in the annotated videos.

<u>(Button) 'Select the quantitative measurements for each behavior'</u>

There are 13 quantitative measurements (parameters) for each behavior for you to choose:

◊ The *count* is the summary of the behavioral frequencies, which is the occurrence number of a behavior within the entire duration of analysis. Consecutive single occurrences (at a single frame) of the same behavior are considered as one count.

◊ The *latency* is the summary of how soon a behavior starts, which is the time starting from the beginning of the analysis to the time point that the behavior occurs for the first time.

◊ The *duration* is the summary of how persistent a behavior is, which is the total time of a behavior within the entire duration of analysis.

◊ The *speed* is the summary of how fast the animal moves when performing a behavior, which is the total distance traveled (can be back and forth) (d) (between the two centers of mass of the animal) during the time window (t_w) for categorizing the behavior divided by t_w.

◊ The *velocity* is the summary of how efficient the animal's movement is when performing a behavior, which is the shortest distance between the start and the end positions (dt) (between the two centers of mass of the animal) divided by the time (t) that such displacement takes place.

◊ The *acceleration* / *velocity reduction* is the summary of how fast the animal's velocity changes while performing a behavior, which is the difference between maximum velocity (v_{max}) and minimum velocity (v_{min}) divided by the time (t_v) that such velocity change takes place.

◊ The *distance* is the total distance traveled of the animal by performing a behavior within the entire duration of analysis.

◊ The *intensity (area)* / *intensity (length)* is the summary of how intense a behavior is, which is the accumulated proportional changes of the animal body area (a) / length (l) between frames divided by the time window for categorizing the behaviors (t_w) when performing a behavior.

◊ The *magnitude (area)* / *magnitude (length)* is the summary of the motion magnitude, which is the maximum proportional change in animal body area (a) or length (l) when performing a behavior.

◊ The *vigor (area)* / *vigor (length)* is the summary of how vigorous a behavior is, which is the magnitude (area) / magnitude (length) divided by the time (t_a or t_l) that such a change takes place.

Details on how they are calculated are in *LabGym* paper [1].

- (Pop-up option) 'Normalize the distances?'

If choose 'No', all the distances will be output in pixels. The unit of all the distance related measurements will be 'pixel' or pixel related. If choose 'Yes', all the distances (calculated in pixels) will be normalized to (divided by) the size of a single animal (also calculated in pixels). In this scenario, all distance related measurements will be normalized measurements (e.g., normalized speed) and will not have a unit. In this way, you do not need to worry about the ratio of pixel / actual size (length) across different videos with different recoding methods, if the size of animals used in these videos are consistent. The ratio of pixel / actual size (length) is not easy to obtain and is subject to change easily (e.g., when the zoom-in level changes). With the option of normalizing distances to the size of a single animal, you can compare the analysis results across different recordings or experimental sessions without worrying about the changes in the ratio of pixel / actual size (length).

<u>(Button) 'Start to analyze the behaviors'</u>

3.3.2. '*Mine Results*'

Automatically performs parametric / non-parametric statistical analysis among groups that you selected, according to the data distribution, to compare the mean / median of different groups and display the data details that show statistically significant difference.

<u>(Button) 'Select the folder that stores the data files'</u>

- (Pop-up option) 'Paired data?'

  Specify whether the data is paired. Different statistical analysis method will be applied accordingly.

<u>(Button) 'Select the control group'</u>

<u>(Button) 'Select the folder to store the data mining results'</u>

<u>(Button) 'Start to mine data'</u>

The Shapiro test will be first performed to assess the normality of data distribution. For normally distributed data, if unpaired, unpaired t-test for 2 groups, ANOVA for more than 2 groups, with either Tukey (comparing each pair) or Dunnett's (comparing all groups against the control group) posthoc comparison; if paired, paired t-test for 2 groups, ANOVA for more than 2 groups, with either Tukey (comparing each pair) or Dunnett's (comparing all groups against the control group) posthoc comparison. For data that is not normally distributed, if unpaired, Mann Whitney U test for 2 groups, Kruskal Wallis for more than 2 groups, with Dunn's posthoc comparison for both comparison of all groups and against control; if paired, Wilcoxon test for 2 groups, Friedman for more than 2 groups with Dunn’s posthoc. The selections of the tests are consistent with those in GraphPad Prism 9.

ACKNOWLEDGEMENTS

This research was supported by National Institutes of Health (NIH) to B.Y. (R01NS104299 and R01EB028159). The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

1. Hu, Y., Ferrario, C.R., Maitland, A.D., Ionides, R.B., Ghimire, A., Watson, B., Iwasaki, K., White, H., Xi, Y., Zhou, J., and Ye, B. (2023). LabGym: Quantification of user-defined animal behaviors using learning-based holistic assessment. Cell Rep Methods *3*, 100415. 10.1016/j.crmeth.2023.100415.
2. Van Rossum, G., and Drake, F.L. (2000). Python reference manual (iUniverse Indiana).
3. Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., and Girshick, R. (2019). Detectron2.