The manual of *LabGym* (v2.2)

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/).

      Not to download the latest version of Python3 as some Python libraries used in *LabGym*

      might not be updated yet to be compatible with the latest version of Python3.

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

      python3 -m pip install LabGym

   c. After *LabGym* is installed, if you want to use its Detectors, 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), activate Python3 interactive

      shell by typing:

      python3

   b. After the Python3 interactive shell is activated, type:

      from LabGym import gui

   c. After the importing is done, then type:

1   gui.gui()

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

3

3. Extended guide on using the GUI of *LabGym*

Put your mouse cursor above each button to see a detailed description for it.

6

3.1. '*Preprocessing Module*'

Preprocess your videos to make them more appropriate for analysis.

9

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.

13

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

When trimming videos, if you want to set different time windows for different videos, use code '_stt_' and '_edt_' in the filename of a video, replacing the last 't' with the start ('st') time point and end ('ed') time point. For example, 'XX_st10_ed20.avi' (the '_' is unnecessary if it is at the end of file name) means the trimming will start at the 10th second and end at the 20th second. You can set multiple time windows for a video, just use multiple '_stt_' and '_edt_' following the time sequence.

21

3.2. '*Training Module*'

The first 4 buttons in this module can be used to train and test 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 distinguishing 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. This 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 identities are critical.

The last 4 buttons can be used to train and test Categorizers, which is the way to teach *LabGym* to recognize behaviors that are defined by you.

3.2.1. '*Generate Image Examples*'

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

To train a Detector with high accuracy, 50~150 annotated images that covers different poses / states / appearance of the animals / objects are enough for a typical scenario. If you use 'Interactive advanced' mode, the annotation should be focused on images in which animals / objects have close body contact. Annotate their outlines well in the boundary regions. To reduce labor, you may only include a few, or no images in which the animals / objects are well-separated. The more colliding scenarios the Detector sees during training, the higher accuracy of the detection, and the higher chance of avoiding identity-switching during analysis. If the Detector sees all the colliding scenarios during training, the identity-switching during analysis can be eliminated. Moreover, although too many training iterations may cause overfitting, more iterations typically yield higher accuracy. During training, pay attention to the 'total_loss', which should be ideally less than 0.2 at the end of training. If the 'total_loss' is too high, increase the training iterations.

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

It determines the speed-accuracy trade-off of Detector performance. You can try large size

such as 960 or above GPUs are used. If you don't use a GPU, start with small size such as 256

and increase the size if the accuracy is not ideal. 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. For general

scenarios, size that is smaller than 192 or larger than 1600 is not recommended.

3.2.4. '*Test Detectors*'

To test a trained Detector, you need to prepare a ground-truth testing image dataset, just like

what you do for preparing the training image dataset, i.e., annotate the images with the same

object / animal names as those in the Detector to test. The mean average precision (mAP) of

the selected Detector will be printed out at the end of testing.

3.2.5. '*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 and sort them. **DO NOT** change the file names for the generated behavior examples

since they contain the key information for later training steps and the function of 'Sort Behavior

Examples' will also be based on these file names.

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

There are 2 modes of interactive behaviors that *LabGym* can generate examples and

analyze. "Interactive basic" is faster in analysis and is useful in the scenario when you don't

care about which individual does what in an interaction, or all the individuals do the same

behavior in an interaction. It analyzes the behaviors of all animals / objects in an interactive pair

/ group as one and does not distinguish individuals. The behavior examples for this mode

contain all the animals in a frame. The behavioral quantifications in this mode are behavior

count, behavior latency and behavior duration.

The "Interactive advanced" is slower in analysis than "basic", but it distinguishes individuals

during close body contact. It can tell which animal does what in an interaction. The behavior

examples for this mode contain all "characters" within a "social distance" defined by you, with a

spotlight on the main character. The social distance should be an integer, for example, if enter

"2", all animals / objects whose distance from others is within "2 folds of the square root of an

averaged animal area" will be included in one behavior example. You can do a rough

estimation. If enter "0", the social distance is set to be infinity and all the animals in a frame will

be included. When annotate such examples, you annotate the behavior of the main character.

The behavioral quantifications in this mode are behavior count, behavior latency, behavior

duration, 3 length parameters, 3 areal parameters, and 4 locomotion parameters.

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

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

3   '*Subtract background*' is the first choice for videos in which the background is static, the

4   illumination is stable overtime, and the total behavior events are more important than animal

5   IDs, because this method is fast and accurate in such videos but cannot distinguish entangled

6   animals and the IDs might switch after they re-separate. When using this method, you need to

7   specify the scenarios of their experiments: '*Animal brighter than background*', '*Animal darker*

8   *than background*', or '*Hard to tell*'. A pop-up option '(Optional) load existing background?' will

9   need to be specified. *LabGym* will output the extracted backgrounds (as images) for each video

10   processed. Therefore, this pop-up option can be used to save the step of background extraction

11   if the background for this video has already been extracted and output. You also need to specify

12   whether the illumination in videos is unstable. If the illumination is very stable overtime, choose

13   'No' to increase the processing speed. Finally, you need to specify a time window for

14   background extraction. An appropriate time window for background extraction should be a

15   period (typically 10~60 seconds) during which the animals move around. This time window

16   should be as short as possible for increasing processing speed. There are 3 options to specify

17   this time window: (1) '*Use the entire duration of a video*': not recommended if processing speed

18   is critical. (2) '*Decode from filenames: _xst_ and _xet_*': can be used if multiple videos are

19   selected and the time window for background extraction is different for each video. Two code

20   tags '_xst_ and _xet_' ('xs' stands for 'extraction start time', 'xe' stands for 'extraction end time',

21   and 't' should be an integer) need to be added into the file name of each video to let *LabGym*

22   decode the time window. For example, if a video name is 'XX.avi', to set the time window for

23   background extraction from the 25th second to the 47th second, rename the video to

24   'XX_xs25_xe47.avi' (the '_' is unnecessary if it is at the end of file name). (3) '*Enter two time*

25   *points*': can used if only one video is selected, or multiple videos share the same time window

26   for background extraction.

1   '*Use trained Detectors*' is useful in any kind of videos or experimental settings. It is also

2   useful for differentiate individual animals when they entangle. The only caveat of this method is

3   slow.

4

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

6   •   (Pop-up option) 'Illumination shifts?'

7   This option is only for background subtraction-based detection method. It is useful when

8   there are sudden bright-to-dark or dark-to-bright illumination transitions in videos. If choose

9   'Yes', there will be 3 options to specify the beginning time: '*Automatic (for light on and off)*',

10   '*Decode from filenames: _bt_*', and '*Enter a time point*'. If choose 'No', only the latter two options

11   can be chosen.

12   '*Automatic (for light on and off)*' is for videos involving lighting on and off (e.g., optogenetics)

13   and the first time point of illumination change will be automatically detected and used as the

14   beginning time to generate behavior examples.

15   '*Decode from filenames: _bt_*' can be used if multiple videos are selected and the beginning

16   time for each video is different. You need to add a code tag '_bt_' ('b' stands for 'beginning time'

17   and 't' should be number) into the file name of each video to let *LabGym* decode the beginning

18   time. For example, if a video name is 'XX.avi', to set the beginning time at the 12.35 second,

19   rename the video to 'XX_b12.35.avi' (the '_' is unnecessary if it is at the end of file name).

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

21   same beginning time.

22

23   (Button) 'Specify the number of animals in a video'

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

25   '*Decode from filenames: _nn_*' can be used if multiple videos are selected and the number

26   of animals in each video is different. A code tag '_nn_' (the first 'n' stands for 'number of

1   animals' and the second 'n' should be an integer) needs to be added into the file name to let

2   *LabGym* decode the animal number. For example, if a video name is 'XX.avi', to set the number

3   of animals in this video as 8, rename the video to 'XX_n8.avi' (the last '_' is unnecessary if it is

4   at the end of file name). If there are multiple kinds of animals / objects in the videos to detect,

5   you can put in multiple '_nn_' code. The order should be consistent with that of the 'animal /

6   object name' in the Detector to use (when you select a Detector, the animal / object name will

7   display in order, and you can also specify which ones to detect).

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

9   share the same animal number.

10

11   (Button) 'Specify the number of frames for an animation / pattern image'

12   The animations and their paired pattern images in the behavior examples spans a duration

13   (the number of frames, an integer) defined by you, which should approximate the duration of a

14   behavior episode. This duration needs to be the same across all the behavior examples that are

15   used to train one Categorizer. If the duration of different behavior episode is different, use the

16   longest one.

17

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

19   examples'

20   *LabGym* generates a pair of behavior example which spans a user-defined duration at a

21   frame. If two consecutively generated examples are too close in time, say, one generated at the

22   10th frame and the other at 12th frame, suppose their duration is 10 frames, they will have 8

23   overlapping frames. These two behavior examples are too similar and will impair the training

24   efficiency when training a Categorizer. Generating too many such similar examples will also

25   make the example sorting laborious. Therefore, you can choose to skip certain frames between

26   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 equal to the duration you set for a behavior episode.

(Button) 'Start to generate behavior examples'

• (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.

3.2.6. '*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. You can

select a number of behavior examples and move them into a folder, or use this button to watch

each generated behavior example one by one and sort them using shortcut keys. The more

behavior examples for each behavior type are included, the higher accuracy of the trained

Categorizer; the more diverse (let the Categorizer see as many variations of each behavior type

as possible), the higher accuracy of the trained Categorizer. For general scenarios, 100~200

pairs of behavior examples for each behavior type are enough for training a Categorizer with

high accuracy.

Importantly, when you define behavior categories, make the list of categories as complete

as possible. For example, there are 'A', 'B', 'C', and 'D', 4 types of behaviors showing frequently

in all experimental sessions, and if you only sort behavior 'A' and 'C' because they are the

behaviors of your interest. After training, the Categorizer can only recognize behavior 'A' and 'C'

and when it sees type 'B' during analysis, it will forcibly categorize the 'B' into either 'A' or 'C',

based on how similar 'B' is to 'A' or 'C'. If a behavior type happens in a low frequency in all

experimental sessions, then you may not need to include it since it will not affect the

categorization accuracy too much.

The behavior examples within the same behavior category should share similar features or

patterns that can be identified by watching them with your naked eyes. Likewise, the behavior

examples belonging to different behavior categories should possess different features or

patterns that can be identified by watching them with your naked eyes. *LabGym* distinguishes

behavior examples based on visually identifiable features or patterns in them without knowing

additional information from other sources. So, if you cannot identify the shared features or

patterns of behavior examples within the same behavior category by just watching them,

LabGym will not be able to learn this category well. Likewise, if you cannot distinguish the

behavior examples of different behavior categories by just watching them, *LabGym* will not be

able distinguish them either.

3.2.7. '*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. The complexity of Categorizer is customizable.

It takes some trial and errors to find the best setting that perfectly fits your behavioral data.

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

- (Pop-up option) 'Categorizer types'

1   If the processing speed is critical to you, choose the Categorizer with only Pattern

2 Recognizer. If you want to maximize the analysis accuracy and care less about the analysis

3 speed, choose the Categorizer with both Animation Analyzer and Pattern Recognizer.

4 • (Pop-up option) 'Complexity level'

5   There are 7 complexity levels (1~7, from simpler to more complex) of either Animation

6 Analyzer or Pattern Recognizer. You can start with simpler levels like 1 ~ 3 and go more

7 complex if the accuracy is not ideal. If the Animation Analyzer is gray scale (animal / object color

8 is behavior irrelevant), you can make the complexity levels of Pattern Recognizer a little higher

9 than that of Animation Analyzer because the Pattern Recognizer analyzes pattern images

10 containing additional color information, which is indicates the temporal sequence of a behavior.

11 • (Pop-up option) Behavior mode'

12   You need to indicate which mode of behavior examples are used for training the

13 Categorizer. For "Interact basic" and "Interact advanced" behavior examples, you can find the

14 mode information in the filenames of the behavior examples. "_itbs" is "Interact basic" and

15 "_itadv" is "Interact advanced". If the behavior mode is "Interact advanced", you also need to

16 specify a social distance, which can also be found in the filenames of the behavior examples

17 with "_scdtXX_" where "XX" is the number you need to enter for the social distance.

18

19 <u>(Button) 'Specify the input shape for Animation Analyzer / Pattern Recognizer'</u>

20   You can start with the smaller input frame / image sizes such as 16 or 32. If there are many

21 detailed appearance features in the frames that are behavior relevant, go larger size such as 64

22 or 96 or even 128. And always go deeper (increase complexity level) first, rather than go wider

23 (increase input frame / image size) first.

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

25   Choose 'Yes' if the color of animals is behavior irrelevant.

26

1 (Button) 'Specify the methods for data augmentation'

2 Briefly, '*rotation*' will rotate the animal blobs; '*flipping*' will flip the animal blobs; '*brightening*'

3 and '*dimming*' will increase and decrease the brightness of the animal blobs; '*shearing*' will

4 distort the animal blobs; '*rescaling*' will change the width / height ratio of the animal blobs;

5 '*deletion*' will delete one or two frames in the animations (mimic the scenario in which animals

6 are not detected in one or two frames during analysis).

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

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

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

10 If the total number of behavior example pairs used for training a Categorizer is less than

11 1,000 before augmentation, choose "Yes" for this option.

12

13 **3.2.8. '*Test Categorizers*'**

14 To test a trained Categorizer, you need to sort a ground-truth testing folder, in which different

15 behavior examples are stored in different subfolders that are named after the behavior name.

16 The subfolder names (behavior names) should match the behavior names in the selected

17 Categorizer. The precision, recall, and f1 score of each behavior category, as well as the overall

18 accuracy will be printed out at the end of testing.

19

20 **3.3.   '*Analysis Module*'**

21 Automatically track animals / objects of your interest, identify and quantify their behaviors in

22 videos, and automatically mine the analysis results to display the data details that show

23 statistically significant differences among groups of your selection.

24

25 **3.3.1. '*Analyze Behaviors*'**

1   Will output a raster plot for all behavior events for all videos, an annotated video copy for each

2   video, various spreadsheets storing quantification results for each selected behavior parameter.

3

4   <u>(Button) 'Select a Categorizer for behavior classification'</u>

5   Choose a Categorizer for behavior classification in analysis, or just let *LabGym* track the

6   animals and calculate their motion parameters and body kinematics. If the former, you need to

7   specify a 'Uncertainty level' for the Categorizer. This number in percentage determines the

8   threshold for the Categorizer to output an 'NA' for behavioral classification. For example, when

9   the probabilities of behavior A, B, and C are 60%, 10%, and 30%, respectively. If the uncertain

10   level is set to be 31, the behavior classification will be 'NA', as the uncertainty level exceeds the

11   difference between probability of the highest-likely behavior (A, 60%) and the second highest-

12   likely behavior (C, 30%), which is 30%. Setting uncertainty level can reduce the possible false

13   positives in behavior classification since ambiguous classification will be output as an 'NA'.

14   Higher uncertainty level, higher precision but more false negatives at the same time. If choose

15   to not classify behaviors, you need to specify a time window for calculating the motion

16   parameters and body kinematics.

17

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

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

20   Downsizing the frame will exponentially increase the analysis speed. This is **highly**

21   **recommended** especially when using Detector-based detection method and GPUs are not

22   available. The analysis accuracy will not decline if the animal size after downsizing is still larger

23   than the input size of the Categorizer used for analysis. For example, suppose the original size

24   of a video frame is 1000 x 500, the size of an animal is approximately 1/4 to that of a frame, and

25   input size of the Categorizer is 48 x 48. After downsizing the video frames to 500 x 250, the

26   animal size is roughly 125 x 63 and still is larger than the input size of the Categorizer (48 x 48).

1   In this scenario, the analysis accuracy will not decline since the animal blob will be downsized to

2   48 x 48 anyway when input to the Categorizer for analysis.

3

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

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

6   '*Subtract background*' is the first choice for videos in which the background is static, the

7   illumination is stable overtime, and the total behavior events are more important than animal

8   IDs, because this method is fast and accurate in such videos but cannot distinguish entangled

9   animals and the IDs might switch after they re-separate. When using this method, you need to

10   specify the scenarios of their experiments: '*Animal brighter than background*', '*Animal darker*

11   *than background*', or '*Hard to tell*'. A pop-up option '(Optional) load existing background?' will

12   need to be specified. *LabGym* will output the extracted backgrounds (as images) for each video

13   processed. Therefore, this pop-up option can be used to save the step of background extraction

14   if the background for this video has already been extracted and output. You also need to specify

15   whether the illumination in videos is unstable. If the illumination is very stable overtime, choose

16   'No' to increase the processing speed. Finally, you need to specify a time window for

17   background extraction. An appropriate time window for background extraction should be a

18   period (typically 10~60 seconds) during which the animals move around. This time window

19   should be as short as possible for increasing processing speed. There are 3 options to specify

20   this time window: (1) '*Use the entire duration of a video*': not recommended if processing speed

21   is critical. (2) '*Decode from filenames: _xst_ and _xet_*': can be used if multiple videos are

22   selected and the time window for background extraction is different for each video. Two code

23   tags '_xst_ and _xet_' ('xs' stands for 'extraction start time', 'xe' stands for 'extraction end time',

24   and 't' should be an integer) need to be added into the file name of each video to let *LabGym*

25   decode the time window. For example, if a video name is 'XX.avi', to set the time window for

26   background extraction from the 25th second to the 47th second, rename the video file to

1 ‘XX_xs25_xe47.avi’ (the ‘_’ is unnecessary if it is at the end of file name). (3) '*Enter two time*

2 *points*': can used if only one video is selected, or multiple videos share the same time window

3 for background extraction.

4   ‘*Use trained Detectors*’ is useful in any kind of videos or experimental settings. It is also

5 useful for differentiate individual animals when they entangle. The only caveat of this method is

6 slow.

7

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

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

10   Specify whether there are sudden bright-to-dark or dark-to-bright illumination transitions. If

11 choose ‘Yes’, there will be 3 options to specify the beginning time: '*Automatic (for light on and*

12 *off)*', '*Decode from filenames: _bt_*', and '*Enter a time point*'. If choose ‘No’, only the latter two

13 options can be chosen.

14   '*Automatic (for light on and off)*' is for videos involving lighting on and off (e.g., optogenetics)

15 and the first time point of illumination change will be automatically detected and used as the

16 beginning time to generate behavior examples.

17   '*Decode from filenames: _bt_*' can be used if multiple videos are selected and the beginning

18 time for each video is different. You need to add a code tag ‘_bt_’ (‘b’ stands for ‘beginning time’

19 and ‘t’ should be number) into the file name of each video to let *LabGym* decode the beginning

20 time. For example, if a video name is ‘XX.avi’, to set the beginning time at the 12.35 second,

21 rename the video file to ‘XX_b12.35.avi’ (the ‘_’ is unnecessary if it is at the end of file name).

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

23 same beginning time.

24

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

26

1   (Button) 'Specify the number of animals in a video'

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

3   '*Decode from filenames: _nn_*' can be used if multiple videos are selected and the number

4   of animals in each video is different. A code tag '_nn_' (the first 'n' stands for 'number of

5   animals' and the second 'n' should be an integer) needs to be added into the file name to let

6   *LabGym* decode the animal number. For example, if a video name is 'XX.avi', to set the number

7   of animals as 8, rename the video to 'XX_n8.avi' (the last '_' is unnecessary if it is at the end of

8   file name). If there are multiple kinds of animals / objects in the videos to detect, you can put in

9   multiple '_nn_' code. The order should be consistent with that of the 'animal / object name' in

10   the Detector to use (when you select a Detector, the animal / object name will display in order,

11   and you can also specify which ones to detect).

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

13   share the same animal number.

14

15   (Button) 'Select the behaviors for annotations and plots'

16   • (Pop-up option) Specify individual-specific behaviors?'

17   If the selected Categorizer is for "Interactive advanced" mode, you can select individual

18   specific behaviors to initiate the "behavior-guided" identity correction. For example, behavior A

19   only occurs in a specific individual whose initial ID is "0", if its ID switches with another individual

20   (e.g., ID "1") because of false detection in a frame, after that, when behavior A occurs, its ID will

21   switch back to "0" because behavior A only occurs in this specific individual "0". In this way, this

22   individual "0" can correctly maintain its ID during the entire analysis period.

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

24   Specify a color to represent a behavior category in the annotated videos and the raster plot

25   for behavior events. In the annotated videos, the value of % confidence of behavior

26   categorization will be shown; in the raster plot, the color intensity indicates the value of %

1   confidence, from 0% of the color intensity (clear) indicating 0% of the confidence, to 100% of the

2   color intensity indicating 100% of the confidence. If choose not to specify the colors, *LabGym*

3   will use the default colors to represent the behaviors.

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

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

6

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

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

9   ◊   The *count* is the summary of the behavioral frequencies, which is the occurrence

10   number of a behavior within the entire duration of analysis. Consecutive single

11   occurrences (at a single frame) of the same behavior are considered as one count.

12   ◊   The *latency* is the summary of how soon a behavior starts, which is the time starting

13   from the beginning of the analysis to the time point that the behavior occurs for the first

14   time.

15   ◊   The *duration* is the summary of how persistent a behavior is, which is the total time of a

16   behavior within the entire duration of analysis.

17   ◊   The *speed* is the summary of how fast the animal moves when performing a behavior,

18   which is the total distance traveled (can be back and forth) (d) (between the two centers

19   of mass of the animal) during the time window (t_w) for categorizing the behavior divided

20   by t_w.

21   ◊   The *velocity* is the summary of how efficient the animal's movement is when performing

22   a behavior, which is the shortest distance between the start and the end positions (dt)

23   (between the two centers of mass of the animal) divided by the time (t) that such

24   displacement takes place.

25   ◊   The *acceleration / velocity reduction* is the summary of how fast the animal's velocity

26   changes while performing a behavior, which is the difference between maximum velocity

27   (v_{max}) and minimum velocity (v_{min}) divided by the time (t_v) that such velocity change takes

28   place.

29   ◊   The *distance* is the total distance traveled of the animal by performing a behavior within

30   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).

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.

1   The Shapiro test will be first performed to assess the normality of data distribution. For

2   normally distributed data, if unpaired, unpaired t-test for 2 groups, ANOVA for more than 2

3   groups, with either Tukey (comparing each pair) or Dunnett's (comparing all groups against the

4   control group) posthoc comparison; if paired, paired t-test for 2 groups, ANOVA for more than 2

5   groups, with either Tukey (comparing each pair) or Dunnett's (comparing all groups against the

6   control group) posthoc comparison. For data that is not normally distributed, if unpaired, Mann

7   Whitney U test for 2 groups, Kruskal Wallis for more than 2 groups, with Dunn's posthoc

8   comparison for both comparison of all groups and against control; if paired, Wilcoxon test for 2

9   groups, Friedman for more than 2 groups with Dunn's posthoc. The selections of the tests are

10 consistent with those in GraphPad Prism 9.

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.