1 A toolbox for automated video analysis of rodents engaged in string-pulling: Phenotyping motor
2 behavior of mice for sensory, whole-body and bimanual skilled hand function

3 Samssoon Inayat¹, Surjeet Singh¹, Arashk Ghasroddashti¹, Qandeel¹, Pramuka Egodage¹, Ian Q.
4 Whishaw¹, and Majid H. Mohajerani^{1,4}

5 1. Canadian Centre for Behavioural Neuroscience, University of Lethbridge, Lethbridge, Alberta,
6 Canada

7 2. Corresponding authors: samssoon.inayat@gmail.com, mohajerani@uleth.ca

8

9

Abstract:

String-pulling in rodents (rats and mice) is a task in which animals make hand-over-hand movements to spontaneously reel in a string with or without a food reward attached to its end. The task elicits bilateral skilled hand movements for which rodents require little training. The task is suitable for phenotyping physiology and pathophysiology of sensorimotor integration in rodent models of neurological and motor disorders. Because a rodent stands in the same location and its movements are repetitive, the task lends itself to quantification of topographical and kinematic parameters for on-line tactile tracking of the string, skilled hand movements for grasping, and rhythmical bilateral forearm movements to advance the string. Here we describe a Matlab® based software with a graphical user interface to assist researchers in analyzing the video record of string pulling. The software allows global characterization of position and motion using optical flow estimation, descriptive statistics, principal component, and independent component analyses as well as temporal measures of Fano factor, entropy, and Higuchi fractal dimension. Based on image segmentation and object tracking heuristic algorithms, the software also allows independent tracking of the body, ears, nose, and forehands for estimation of kinematic parameters such as body length, body angle, head roll, head yaw, head pitch, movement paths and speed of hand movement. The utility of the task and that of the software is presented by describing mouse strain characteristics in string-pulling behavior of two strains of mice, C57BL/6 and Swiss Webster. Postural and skilled hand kinematic differences that characterize the strains highlight the utility of the task and assessment methods for phenotypic and neurological analysis of healthy and rodent models of diseases such as Parkinson's , Huntington's, Alzheimer's and other neurological and motor disorders.

Significance statement:

Mouse models are used to investigate the physiology and pathophysiology of motor deficits observed in human neurological conditions, for testing substances for therapeutic drug development, and to investigate the role of neural systems and their genetic basis in the expression of behavior. Behavioral tasks involving unconditioned and natural behavior can provide rich insights into motor performance in animal models and analyses can be aided by the automated processing of video data for reliable quantification and high throughput.

Keywords: String-pulling, Biomechanics, Sensorimotor integration, Hand Kinematics, Matlab Toolbox, Swiss Webster, C57BL/6

1. INTRODUCTION

String-pulling is a proto-tool behavior in which an animal pulls on a string to obtain a reward. Variations of the behavior are observed in over 160 species of animals including birds, bees, rodents and primates and humans (Jacobs and Osvath 2015, Singh, Mandziak et al. 2019). The task has been used to assess cognitive function, motivation, and work/reward relations in animals and as a therapy and exercise tool in humans. If animals are presented with an overhanging string, they adopt a standing or sitting posture and use hand-over-hand movements to reel in the string (Blackwell, Banovetz et al. 2018, Blackwell, Köppen et al. 2018, Blackwell, Widick et al. 2018). Mice will even spontaneously reel in a string, but the behavior becomes more reliable when reinforced by a food reward attached to the string (Laidre 2008). The movement is an on-line act, guided by sensory information from snout vibrissae, and features four hand shaping movements for grasping and releasing the string, and five arm movements for retrieving and advancing the string that are similar in mice, rats, and humans (Blackwell, Banovetz et al. 2018, Singh, Mandziak et al. 2019).

The reaching/grasping movements of string-pulling resemble the movements of reach-to-eat movements displayed by rodents in other tasks, including the staircase (Dunnett and Brooks, 2018) and single-pellet reaching tasks (Whishaw, 1996) but additionally both hands can be assessed concurrently and many reach movements can be collected rapidly. The organized structure of string pulling makes the task useful for investigating sensory, motor, cognitive and social learning, as well as studying pathophysiology of motor deficits in animal models of motor disorders (Blackwell, Widick et al. 2018). The string-pulling task for mice has promise as an unconditioned motor task for phenotyping the growing number of transgenic mouse models of neurological and other diseases but requires an analytical method for its systematic analysis. Although most of the work of advancing the string by a mouse is done with the arms and hands, the movements rely on online sensory control and appropriate postural support. Associated with strain differences or neurological impairment, any of these aspects of behavior may be impaired and compensated for by a variety of sensorimotor adjustments. Thus, the analysis of complex changes in behavior require sophisticated whole-body behavioral analysis in addition to the analysis of skilled hand movements. Currently, most of the analysis is carried out with manual annotation of frames leading to low yield of experiments. Automated analysis of mouse string-pulling is useful way for obtaining a kinematic assessment of movements, for identifying the contributions of different body parts to string-advancement, and for gaining insights into inter hand coordination (Whishaw 1996, Dunnett and

69 Brooks 2018). The objective of the present paper is to describe a state-of-the-art method for analyzing
70 mouse string-pulling.

71 We present a Matlab based toolbox to facilitate the analyses of video data of different strains of
72 intact or neurologically compromised mice engaged in string pulling. The toolbox features a graphical
73 user interface for video data analysis and follows the analysis framework with two major components, a
74 global assessment of whole-body position and speed and fine kinematic assessment of movements of
75 body, ears, nose, and hands (Fig. 1.1). As a first step, the user can run optical flow analysis on video data
76 (image sequence) to estimate velocity vector fields and speed frames. Next using descriptive statistics and
77 temporal measures of Fano factor, entropy, and Higuchi fractal dimension, the user can characterize
78 whole-body position, speeds, and their temporal variability. With principal and independent component
79 analyses of the image sequence and speed frames, the user can assess global position and speed patterns
80 to observe regions of significant motion. Additional spatial measures of entropy, sharpness, and Hausdorff
81 fractal dimension allow the user to quantify randomness, edges, and geometrical complexity respectively
82 and use these quantifications to statistically compare groups of animals. We demonstrate that with the
83 whole-body measures mentioned above, we can detect overall differences in position and motion patterns
84 of two different strains of mice, C57BL/6 (Black) and Swiss Webster Albino (White). Furthermore, using
85 color-based image segmentation and heuristic algorithms, the software allows detailed automatic tracking
86 of body, ears, nose, and hands for kinematic analyses. The software also allows the user to manually curate
87 and correct automatic tagging of body parts. Finally, the user can either utilize Matlab scripts provided
88 with the software for comparison of groups of mice or export data to Microsoft Excel for further flexible
89 data analyses. A modular approach has been used to build the toolbox, allowing users to add additional
90 analytical probes.

Figure 1.1. String-pulling video analysis framework. Characterization of whole-body position and speed and their temporal variability is assessed with descriptive statistics, Fano factor, entropy, and Higuchi fractal dimension. Speed is estimated with optical flow estimation. Principal and independent component analysis of frames of image sequence and speed frames provide overall assessment of position and speed patterns. Measures of spatial entropy, sharpness, and Hausdorff fractal dimension on frames obtained in earlier steps are used to statistically compare global position and motion patterns of groups of animals. For fine kinematic assessment of the motion of body, ears, nose, and hands, they are tracked using image segmentation and heuristic algorithms. Shaded boxes represent methods.

2. METHODS

2.1 Experimental animals

Adult mice 3-6 months old, 5 male C57 Bl/6J (Black) and 5 male Swiss Webster Albino (White), were used in this study. They were kept in a controlled temperature (22°C), humidity, and light with a 12:12 light/dark cycle. All testing and training were performed during the light phase of the cycle. All experiments were performed in accordance with the Canadian Council of Animal Care and were approved by the University of Lethbridge Animal Welfare Committee.

2.2 String-pulling task

The apparatus and training procedure were as described previously (Blackwell, Banovetz et al. 2018). Briefly, the test box consisted of a clear Plexiglas cage (20 cm long, 9 cm wide, and 20 cm high) in which a string was thrown over the center of the front wall so that its end could be reached by the mouse. A video camera was placed in front of the cage and behavior was filmed with a Panasonic camcorder at 60 f/sec with a 1/1,000 shutter speed and frame resolution of 1920x1080 pixels. For three White mice, the data was collected with another camera, a Sony Camcorder with 1/750 shutter speed and 60 f/sec. Mice would spontaneously pull the string into the cage but were also lightly food restricted so that they were motivated to pull the string to obtain a food reward (pieces of a cashew nut or cheerios) attached to the end of the string.

2.3 Analysis platform and preliminary processing

Matlab R2016 and Microsoft Windows platform was used for the development of the toolbox and analysis of the video record. From the videos, a subset of frames representing string-pulling epoch was selected. The start of the epoch was when the mouse touched the string and started pulling it down while the end of epoch was when the mouse stopped the bout of string-pulling. Furthermore, a subregion of frames was selected for analysis to reduce computation time. In the toolbox, this subregion is called “auto zoom window” and is found automatically but is also subject to adjustment.

In order to do a fair comparison between the performance of Black and White mice with whole-body analysis, the image frames of Black mice were complemented using Matlab’s “imcomplement” function. To accelerate the processing of whole-body analyses, red, green, and blue (RGB) frames were

first resized to one-fourth their original size using Matlab's "imresize" function and then converted to gray scale frames using Matlab's "rgb2gray" function. The user can, however, set the image resize value using the provided graphical user interface (see below). For kinematic assessment of movements, body, ears, nose, and hands were tracked in the original frames using color-based image segmentation and heuristic algorithms.

2.4 Optical flow analysis of image sequence for estimation of velocity vector fields and speed frames

The Combined Local-Global (CLG) method was used to estimate optical flow in string pulling videos (Bruhn, Weickert et al. 2005, Afrashteh, Inayat et al. 2017). The Matlab implementation of the CLG method provided by (Liu 2009) was used with the following default parameters which the user can modify if required (alpha = 0.01, ratio = 0.5, minWidth = 20, nOuterFPIterations = 7, nInnerFPIterations = 1, nSORIterations = 30). This method was chosen because it provides a better optical flow estimate compared to Horn-Schunck and Temporospatial methods (Afrashteh, Inayat et al. 2017). The chosen parameter values faithfully estimated optical flow, as was validated with visual observation of moving objects in frames. To reduce computational time, image frames in videos were reduced by four times but resizing of image frames did not affect the quality of images or the calculations of optical flow as confirmed with visual observation of velocity vector fields. As the optical flow computes one velocity vector field frame from two video frames, the number of optical flow frames i.e., velocity vector fields, is one less than that of video frames.

2.5 Descriptive statistics

The mean, median, mode, standard deviation, skewness, and kurtosis were calculated for image sequence and speed frames by using respective Matlab functions. For example, the mean frame was determined by finding the mean of the time series of all individual pixels in a frame sequence i.e., $\mu_{x,y} = \frac{1}{N} \sum_{t=1}^N I(x, y, t)$ where $\mu_{x,y}$ denotes mean value of intensity at x and y location, t is frame number, N is the total number of frames, and I is the average gray scale intensity determined from red, green, and blue frame sequences. Other parameters were similarly calculated.

2.5.1 Finding a mask for mean mouse position from mean frame: A mask was generated to estimate the average position of a mouse by using a threshold equal to average of all pixels in the mean frame. Matlab's

“imbinarize” function was used to generate the mask, which makes all pixels zero whose value is below the threshold and sets other pixels to a value of one.

2.6 Temporal analysis of image sequence with Fano factor, entropy, and Higuchi fractal dimension

To assess the temporal changes in the position and speed from image sequence and speed frames respectively, three measures, Fano factor, entropy, and Higuchi fractal dimension (HiFD) were used. A Fano factor frame for the image sequence was calculated to find the Fano factor value for each pixel from its time series using the following equation; $Fano\ Factor = \frac{\sigma^2}{\mu}$, where σ is the standard deviation and μ is the mean of time series. Similarly, an entropy frame for the image sequence was calculated by finding the entropy of the intensity time series of individual pixels by using the “entropy” function of Matlab which uses the following relationship; $E_{x,y} = -\sum p_{x,y} \log_2(p_{x,y})$ where $E_{x,y}$ is the entropy of the time series at x,y location and $p_{x,y}$ is the normalized histogram counts of intensity values for the same location returned by the “imhist” Matlab function. Prior to finding this measure, “mat2gray” Matlab function was applied on time series to scale maximum and minimum values as 1 and 0 respectively. The Higuchi fractal dimension (Higuchi 1988) frame was also calculated from the time series of individual pixels in a similar way using a function downloaded from

<https://www.mathworks.com/matlabcentral/fileexchange/50290-higuchi-and-katz-fractal-dimension-measures>. The value of Kmax parameter used to find HiFD was arbitrarily chosen to be 10 (Spasic, Kalauzi et al. 2005). The measures of Fano factor, entropy, and HiFD were also calculated in the same way for speed frames.

To statistically compare the values of the above-mentioned parameters between groups of Black and White mice, values of parameters were extracted from respective frames using the mean position mask (section 2.5.1). For example, for comparing entropy, values of entropy were selected from the entropy frame from pixels that had a corresponding value of 1 in the mean position mask. The mean of all these entropy values was chosen as a representative value for the animal. These means were statistically compared between groups using Student’s t-test.

2.7 Principal component analysis of image sequence and speed frames

In Matlab, the “pca” function was used to determine principal components (PCs) of the image sequence and motion profile (speed frames) determined with optical flow analysis. For a frame sequence, the 3D matrix $H \times W \times N$ (H , W , and N being height, width, and number of frames) was converted to a 2D matrix with the number of rows equal to $H \times W$ and columns equal to N . The function `pca` was then executed to determine N principal components. The individual component scores were then reshaped from 1D ($H \times W$) to 2D H rows and W columns. PCs of speed frames were similarly calculated.

2.8 Independent component analysis of image sequence and speed frames

A fast independent component analysis (fastICA) algorithm (Hyvarinen and Oja 2000) was used to find independent components (ICs) of the image sequence as well as speed frames. The function was downloaded from <https://www.mathworks.com/matlabcentral/fileexchange/38300-pca-and-ica-package>. To apply “fastICA” on the image sequence or speed frames, the 3D matrix of frames was converted to a 2D matrix with the number of rows equal to $H \times W$ and columns equal to N where H and W are the number of height and width pixels while N is the number of frames. After applying the fastICA function, individual component frames were determined by reshaping from 1D scores ($H \times W$) to H rows and W columns. Since, the algorithm to find ICs is iterative, the minimum tolerance set to stop iterations was $5e-5$ and the maximum number of iterations was set to 500. ICs were found before reaching the maximum number of iterations i.e., the error went below tolerance.

2.9 Calculation of spatial measures of entropy, sharpness, and Hausdorff fractal dimension

For various frames obtained in the previous sets of whole-body analyses; e.g., mean frame of image sequence or the first principal component frame, spatial entropy, sharpness, and Hausdorff fractal dimension (HFD) were calculated. The same entropy function described earlier was used for finding spatial entropy after scaling the frame values between 0 and 1 with “`mat2gray`” Matlab function. Sharpness was calculated using the following equation;

$$S_n = \left[\sum_{x,y} \sqrt{|\nabla I_{x,y}|} \right] / (H \times W),$$

where $\nabla I_{x,y}$ represents gradient of intensity image and H and W denote height and width of the image respectively. Function to calculate sharpness was downloaded from <https://www.mathworks.com/matlabcentral/fileexchange/32397-sharpness-estimation-from-image-gradients>.

The Hausdorff fractal dimension was obtained using the “boxCountfracDim” function downloaded from <https://www.mathworks.com/matlabcentral/fileexchange/58148-hausdorff-box-counting-fractal-dimension-with-multi-resolution-calculation>. This fractal dimension uses the following relationship; $HFD = -\log(N(\epsilon))/\log(\epsilon)$ where N is the number of pixels above zeros in a box and ϵ is the box size. In the box counting method, an image is first binarized using an appropriate threshold and then non-overlapping boxes of different sizes (different ϵ values) are overlaid on the image and the number of pixels above zero value are counted to obtain $N(\epsilon)$. A line is then fitted for a scatter plot of $\log(N(\epsilon))$ and $\log(\epsilon)$ and the slope of that line provides the fractal dimension (FD). Here, the threshold used to binarize an image was the mean value of all its pixels. For the Black mice, the frames were complemented before the calculation of HFD for comparison with White mice.

2.10 Synthetic data for testing and validation of algorithms

To test the efficacy of the software, two different types of data sets were generated, 1) empirical: based on experimental values of mouse body position and those of hands, nose, and ears, and 2) modelled: where the movement of hands were calculated by a sinusoidal function. For both data sets, color of fur/body, ears, nose, and hand were chosen from the range of observed values to create elliptical patches of body parts on a rectangular background patch. For empirical data, animal movements were created using the centroid, major- and minor- axis of the ellipse calculated from the real experiments. For modelled data, we observed that x- and y- motion of hands can be modelled using two 180° out of phase sinusoidal oscillators, and body tilt is in-phase with the reaching-hand. In general, we observed that one complete reach-grasp-release cycle was completed in ~12-frames (0.2 secs). Thus, the 360° x- and y- motion was divided into 12 parts to mimic the hand movements. Further, hand movements were linked to the body centre to mimic the changes in hand movements associated with body movements.

2.11 Color-based image segmentation for gross motion assessment and tracking of body, ears, nose, and hands

In computer vision, image segmentation is a basic process in which an image is divided into regions based on distinguishing features, such as color and texture. In the current video data, we utilized color-based segmentation of frames for finding regions containing fur, ears, nose, hands, and the string. To define a color, a user first selects a rectangular region around the object of interest (Fig. 2.1) e.g., hands. K-means clustering is then used to separate colors in the rectangular region into 3 clusters. With

trial and error, it was found that three clusters were sufficient to separate colors of interest for body and hands while seven clusters were sufficient for separating colors of interest for the string and nose. The number of clusters can be changed with the user interface, however. The user is then allowed to choose the cluster that represents the object of interest e.g., cyan cluster in Fig. 2.1 A3 for hands. Using these colors, a mask is then generated and shown to the user for visual inspection. Through the supplied interface, the user can then add or subtract colors until a reasonable mask is generated. The user can then move to the next frame and repeat the process for the new frame. This way, with five to ten frames within an epoch chosen randomly, an appropriate selection of colors for an object of interest is made. All color values in a cluster are stored and later used to find masks. In a similar fashion, the user selects colors for other objects of interest.

Figure 2.1. Image segmentation based on object colors for finding object masks and gross assessment from descriptive statistics of masks. (A) Color definitions in original frames are semi-automatically made with the user's assistance for fur, ears, nose, hands, and string. The user first selects a rectangular region around the object of interest (A1). K-means clustering is then applied to segregate different colors within the selected region (A2). The user then selects a color that corresponds to the object of interest (A3, example shown for selecting color for the hands). Using color selections for the fur and ears, quick segmentation is used to identify the fur and ears in every third frame in an epoch of interest and projected onto a single frame after edge detection (A4). A rectangular window around the blob is

drawn to define a subregion of frames for which masks are then found (A5, representative masks). Note that masks are not always clean and include regions other than objects of interest e.g., hand masks include regions identified in ears and feet because of color similarity.

Once colors are defined, masks are generated in which colors matching the color of interest are found (Fig. 2.1 A5). Three different methods can be used to generate masks; 1) K-Nearest Neighbor (Matlab's KNNsearch function), 2) a Range search algorithm (Matlab's rangearch function), and 3) a custom algorithm which divides a frame into a grid and uses rangearch for individual grid blocks. To save computational time and memory storage, masks are not found for the whole frame, rather for a subset of each frame. The region of a frame for which masks are determined is found such that it includes the mouse body in all frames of an epoch. First, the range of motion of the whole mouse body within an epoch is estimated by finding the edges of a mouse's body in every third frame of an epoch and plotting them together (Fig. 2.1 A4). The user then draws a window around the blob of plotted edges, which is then stored as a parameter in "auto zoom window" and used to find masks. On a computer with Intel® Core™ i7-4820K with two CPUs (each 3.7GHz) and 32GB RAM, it takes ~1 hour to process 477 frames (each frame being 457x895 pixels) for finding 5 sets of masks for fur, ears, nose, hands, and string. With an option to reduce image size by a factor adjustable with the user interface, the calculation of masks can be accelerated by several orders of magnitude.

2.12 Statistical Analysis

Statistical analysis was done using Matlab R2016b. Data are presented as mean $\pm$ the standard error of mean (Mean $\pm$ SEM). Two-sample Student's t-test and Kolmogorov-Smirnov (ks) tests were used for statistical comparisons and an alpha value of <0.05 was used to determine significance.

3. RESULTS

3.1 Characterization of position with central-tendency descriptive statistics of image sequence shows larger number of distinct positions for White compared to Black mice

Position characterization of a mouse during the string-pulling behavior (representative frames shown in Fig. 3.1 A and C for Black and White mice respectively) can be grossly done using central-tendency descriptive statistics i.e., mean, median, and mode, of the image sequence. By observing the

mean, median, and mode frames (Fig. 3.1 B and D), one can infer the average to most frequent positions of the mouse during string-pulling. These frames of the representative Black and White mice show sitting and standing positions respectively. For statistical comparison of these frames between groups of Black (N=5) and White (N=5) mice, spatial entropy, sharpness, and Hausdorff fractal dimension (HFD) measures were calculated and compared using Student's t-tests. Spatial entropy is measured from 1D projection of the frames while sharpness and HFD operates directly on a 2D frame. Spatial entropy, sharpness, and HFD quantify randomness, edges, and complexity of a frame respectively. The measures were similar for both groups except that the spatial entropy of the mean frames was larger for White compared to Black mice ($p = 0.0092$, Fig. 3.1 E). This suggests that the mean frame of White mice had more random values as compared to Black mice owing to larger number of distinct positions.

Figure 3.1. Characterization of whole-body position using central-tendency descriptive statistics. (A) Representative subset of image frames of a string-pulling epoch of a Black mouse. (B) Central-tendency descriptive statistics: mean, median, and mode of string-pulling image sequence for a Black mouse. Mean, median, and mode frames represent average to most frequent position of mouse. Shown below each frame are the respective values of measures of spatial entropy, sharpness, and Hausdorff fractal dimension separated by hyphens. The black line in the mean frame shows the boundary of a mask generated by selecting all pixels with values above the average value of the mean frame. (C) and (D) same as (A) and (B) respectively but for a White mouse. Descriptive statistics of representative Black mouse show sitting as most frequent posture while that of the White mouse is upright standing. (E) Mean $\pm$ SEM spatial entropy, sharpness, and Hausdorff fractal dimension shown for frames of descriptive statistics from 5 Black and 5 White mice.

3.2 Characterization of temporal changes in position with variability descriptive statistics, Fano factor, entropy, and Higuchi fractal dimension of image sequence depicts larger variability of positions of White compared to Black mice

To assess the temporal changes in the position of the mouse and the string, we found measures of the time series of individual pixels including variability descriptive statistics (standard deviation, skewness, and kurtosis), Fano factor, temporal entropy, and Higuchi fractal dimension (Fig. 3.2). The standard deviation frame shows a summary of spatial variation of the mouse position over time e.g., how the body of the representative Black mouse was in a sitting as well as standing position (Fig. 3.2A). Conversely, the White mouse remained in a standing position and moved its head left and right i.e., yaw motion (Fig. 3.2B). The skewness and kurtosis frames conveyed similar information and captured rare events such as positions of string in relation to the mouse body. Both skewness and kurtosis had large values around the mouse body. Like the standard deviation frame, the Fano factor frame also showed change in position over time but with respect to the mean position. Fano factor frame highlighted the changes more than the standard deviation frame; e.g., for both representative Black and White mice, there is large variation at the top of the head region where the ears are. The temporal entropy of the image sequence quantified randomness of the time series of individual pixels and hence could also be translated to the change in position i.e., quantity of motion at the individual pixel location. For the representative Black mouse, it is largest around the boundary of the mouse (note the thin red layer near body edge in the Entropy frame). For the lower half of the body, the thickness of the red band, signifying value, is thin whereas it is large in the upper half of the body where it captured the change in mouse's sitting to standing posture (Fig. 3.2D). For the White mouse, the temporal entropy value was quite large within the mouse's body depicting large motion of the body. In the upper half of the body, it also revealed a yaw motion of the head. The Higuchi fractal dimension (HiFD) quantified the complexity of the time series of individual pixels in the image sequence. For the representative Black and White mice, HiFD captured changing positions of the string with larger values (hot regions) outside the mouse body. Within the mouse body, it also captured the position of hands which were colocalized with the string in most of the frames (at least one of the hands). The smaller HiFD values (colder regions) for the Black mouse depict the gradual transition in mouse positions from sitting to standing while for the White mouse, they depict its head's yaw motion.

Figure 3.2. Characterization of changes in position with variability descriptive statistics, Fano factor, entropy, and Higuchi fractal dimension. (A) Representative variability descriptive statistics – standard deviation (Std. Dev.), skewness, and kurtosis, Fano factor, Entropy, and Higuchi fractal dimension of string-pulling image sequence of a Black Mouse. Standard deviation depicts summary of spatial variation over time (mouse posture from sitting to standing). Skewness and kurtosis capture regions of rare events around the mouse body such as position of string. Shown below each frame are the respective values of spatial entropy (SE), sharpness (S), and Hausdorff fractal dimension (HFD) separated with a hyphen. (B) same as (A) but for White mouse. The White mouse has greater yaw and pitch head motion as seen in the standard deviation. (C) Mean±SEM cumulative distribution function (CDF) of the standard deviation values within a mask obtained for each individual mouse from their respective mean frames. For the representative Black and White mice, outline of these masks is shown in Fig. 3.2 B and D respectively. (D-H) Same as (C) but for other parameters. (F) Mean±SEM SE, S, and HFD shown for frames of parameters mentioned above.

To assess temporal changes in positions of Black versus White mice, we compared the distributions of values of the above-mentioned parameters. The values of parameters were obtained from their respective frames (e.g., values of HiFD from the HiFD frame) using a mask obtained from the mean frame (see Methods section 2.5.1). The outline of this mask for the representative Black mouse is shown in the mean frame in Fig. 3.1B (for White mouse in Fig. 3.1D). For each animal the probability density function (PDF) of the values of a parameter were estimated using Gaussian Kernel density estimation. The cumulative density function (CDF) was then determined from PDF. The Mean $\pm$ SEM CDFs from 5 Black and 5 White mice are shown in Fig. 3.2 C-H: standard deviation, skewness, kurtosis, Fano factor, temporal entropy, and HiFD respectively. Alongside the CDF curves, means over animals are also shown that were obtained by first finding a mean (single value for each animal) of the distribution of a parameter. The means of skewness, temporal entropy, and HiFD were significantly larger for White compared to Black mice (individual Student's t-tests, $p < 0.001$, $p = 0.008$, and $p = 0.007$ respectively). The opposite was the case for Kurtosis values ($p = 0.009$).

We also calculated the spatial measures of entropy, sharpness, and Hausdorff fractal dimension for all the individual frames depicting variability and then used individual Student's t-tests (Fig. 3.2 I). The measures of spatial entropy and sharpness for the standard deviation frames were significantly larger for the White compared to Black mice ($p = 0.023$ and $p = 0.017$ respectively, Fig. 3.2 I). The spatial entropy of HiFD frame was significantly larger for the Black compared to White mice ($p = 0.023$). Finally, the sharpness of the Fano factor, temporal entropy, and HiFD frames was significantly larger for the Black compared to White mice ($p = 0.007$, $p < 0.001$, and $p = 0.007$ respectively). Combined these results suggest larger changes in positions of White compared to Black mice.

3.3 Characterization of speeds with central-tendency descriptive statistics reveals larger speeds of White compared to Black mice

By using the CLG optical flow algorithm, velocity vector fields were determined for each consecutive pair of image frames and then speed frames were obtained by finding the magnitude of velocity vectors. The algorithm faithfully captured the motion dynamics of video frames (Fig. 3.3 A and C). The mean, median, and mode speed frames exhibited the spatial distribution of average to most frequent speeds (Fig. 3.3 B and D). Observation of the respective frames for representative Black and White mice show clear similarities as well as differences. For example, the peak values of mean and

median frames for the White mouse are larger than those for the Black mouse but peak location is similar for both, and close to the center of mouse's body where the hands release the string. Similar conclusions can be derived from the median and the mode frames.

Figure 3.3. Characterization of speed with central-tendency descriptive statistics reveals White mice have larger speeds compared to Black mice. (A) Output of the optical flow analysis overlaid on three representative image frames. Velocity vector is shown for every fifth pixel (along both dimensions). For a pixel, the magnitude of the velocity vector shows instantaneous speed (cm/s). For the same image frames, respective speed frames are shown on the right. (B) Descriptive statistics of the speed frames. Shown below each frame are the respective values of spatial entropy (SE), sharpness (S), and Hausdorff fractal dimension (HFD) separated with a hyphen. The unit of color code is cm/sec. (C) and (D) same as (A) and (B) respectively but for White mouse. (E) Average cumulative distributions of speeds from the mean speed frames of Black (N=5) and White (N=5) mice. Shaded regions show SEM. Bar graph show Mean±SEM over animals of mean speeds. (F) Mean±SEM SE, S, and HFD shown for frames of descriptive statistics from 5 Black and 5 White mice.

To compare the distributions of speeds for Black and White mice, speed values were chosen from the mean speed frames using a mask (depicting the average position of mouse, see outline in the mean frames in Fig. 3.1). The mean speed frame was used because the number of frames in a string-pulling epoch can be different from mouse to mouse. Therefore, normalization is done for comparison purpose. The mask was used to get a better estimate of mouse speeds rather than those of spurious speeds of

background pixels. The average cumulative distribution of speeds for White and Black mice is shown in Fig. 3.3 E. The mean speed for White mice was significantly larger than that for Black mice (Student's t-test, $p = 0.003$). Comparison of White and Black mice was also done with measures of spatial entropy, sharpness, and Hausdorff fractal dimension for the mean, median, and mode frames. These measures were not different for groups of Black and White mice except that the spatial entropy of mode frames was significant larger for Black mice compared to White, Student's t-test, $p = 0.017$. These results suggest that White mice had faster movements than Black mice.

3.4 Characterization of temporal changes in speed with variability descriptive statistics, Fano factor, entropy, and Higuchi fractal dimension of speed frames shows larger variability of speeds of White compared to Black mice

Changes in speed were characterized by using variability descriptive statistics, Fano factor, entropy, and Higuchi fractal dimension (HiFD) of speed Frames (Fig. 3.4). For both representative Black and White mice, the standard deviation frame showed the largest variation in speeds around the center of body where the hands release the string. For the Black mouse, variation in speeds can also be seen in the upper left corner where the string appeared and moved in a subset of frames. For the White mouse, large variation can also be seen around the head depicting its yaw motion. The skewness and kurtosis frames show large values around the mouse body showing the distribution of speeds is more non-normal around the mouse (same as for the image sequence). The Fano factor frames measured dispersion of speeds in time relative to the mean. For both White and Black mice, the dispersion appears to be greater around the mouse body. The temporal entropy was greater within the mouse body as compared to outside and vice versa for HiFD.

Figure 3.4. Characterization of changes in speed with variability descriptive statistics, Fano factor, entropy, and Higuchi fractal dimension. (A) Representative variability descriptive statistics – standard deviation (Std. Dev.), skewness, and kurtosis, Fano factor, Entropy, and Higuchi fractal dimension of speed frames of a Black Mouse. Shown below each frame are the respective values of spatial entropy (SE), sharpness (S), and Hausdorff fractal dimension (HFD) separated with a hyphen. (B) Same as (A) but for a White mouse. (C) Mean±SEM cumulative distributions (CDF) of standard deviation values within a mask obtained for each individual mouse from their respective mean frames. For the representative Black and White mice, outline of these masks is shown in Fig. 3.1 B and D respectively. (D-H) Same as (C) but for other parameters. (F) Mean±SEM SE, S, and HFD shown for frames of parameters mentioned above.

To assess the temporal changes in speed for groups of Black and White mice, we compared the values of above-mentioned parameters using distributions of values obtained from each animal using its

respective mask calculated from the mean frame of image sequence (see outline in mean frame in Fig. 3.1 B and D for representative Black and White mice). The Mean $\pm$ SEM cumulative distributions are shown in Fig. 3.4 C-H along with Mean $\pm$ SEM over animals for standard deviation, skewness, kurtosis, Fano factor, temporal entropy, and HiFD. The mean skewness and Fano factor were larger for Black compared to White mice ($p = 0.028$ and $p = 0.023$ respectively). The opposite was true for temporal entropy and HiFD measures ($p = 0.003$ and $p = 0.008$ respectively). The spatial measures of entropy (spatial), sharpness, and Hausdorff fractal dimension for the frames of above-mentioned parameters were not significantly different between the Black and the White mice (Fig. 3.4 I). Taken together, these results suggest larger variability, temporal randomness, and temporal complexity of speeds within the mouse body for White vs Black mice.

3.5 Principal component analysis of image sequence and speed frames for assessment of spatial patterns of position and speed

Principal component analysis of the image sequence and speed frames revealed orthogonal components representing regions with the largest variance in intensity and speeds respectively (Fig. 3.5). The first principal component (PC1) of the image sequence showed the most dominant position of the mouse and was similar to frames obtained by first-order descriptive statistics. For the Black mouse, PC2 mostly captured the difference between sitting and standing posture of the mouse. Higher order components captured other dominant features of images e.g., positions of head and string. PC1 of the speed frames mostly represented motion of the hands, which constituted significant motion. Motion of the string body, head, and ears appeared in higher principal components. For the Black mouse, PC1 of speed also captured dominant speed of the string in the upper left-hand corner. Since, PC1 captures the largest variance, we compared the characteristics of PC1 for groups of Black and White mice. For PC1s of both image sequence and speed frames, the spatial entropy, sharpness, and Hausdorff fractal dimensions of Black and White mice were similar (Student's t-test, Fig. 3.5E), except the spatial entropy of PC1 of image sequence was larger for White compared to Black mice ($p = 0.011$). However, the mean explained variance of PC1 of the image sequence for Black mice was significantly larger than that of White mice compared with a Student's t-test ($p < 0.0001$), whereas the explained variance of PC1 of the speed frames was smaller for Black mice compared to White mice ($p = 0.032$). These findings corroborate results reported above that White mice have larger variability of positions and higher speeds compared to Black mice.

Figure 3.5. Principal component analysis of image sequence and speed frames for assessment of patterns of positions and speeds. (A) The first four principal components (PC) of the image sequence of the representative Black mouse. The number in parentheses indicates the value of explained variance. Shown below each frame are the respective values of spatial entropy (SE), sharpness (S), and Hausdorff fractal dimension (HFD) separated with a hyphen. (B) First four PCs of speed frames for the Black mouse. (C) and (D) same as (A) and (B) respectively but for the representative White mouse. (E) Mean±SEM values of explained variance (Var.), SE, S, and HFD for PC1 of positions and speeds.

3.6 Independent component analysis of image sequence and speed frames for detecting events with high spatiotemporal contrast

Independent component analysis (ICA) allows extraction of statistically independent information within a data set by decomposing it into non-Gaussian components. For the string-pulling motion sequence, the most dynamic movements are those of the hands and the string and they contribute to the non-gaussianity of the data set. When applied directly on the image sequence, ICA extracts events that have high spatiotemporal contrast i.e., where there are sudden changes in an image. Sudden changes include the appearance and disappearance of a hand or string in a frame. Fig. 3.6 A and C show representative independent components (ICs) of image sequences for Black and White mice respectively.

485 The heat maps show small regions with red and dark blue colors identifying regions of high spatiotemporal
 486 contrast (red and blue colors for positive and negative values respectively indicate appearance and
 487 disappearance of objects in frames). The ICs of speed frames were similar in appearance to the ICs of
 488 image sequence as they picked events with high spatiotemporal changes in speeds i.e., mostly movements
 489 of the hands and the string. In order to obtain a global picture of the independent components over time,
 490 maximum (Max) and minimum (Min) projections were determined (Fig. 3.6 B and D) for ICs of both
 491 image sequence and speed frames. For both White and Black mice, Max and Min projections of
 492 independent components show regions where the hands and string were located. In order to validate these
 493 observations, ICA was applied to a synthetic mouse data set in which only the hands moved, and this
 494 revealed the prominent appearance of hand locations in Max and Min projections of independent
 495 components (Fig. 3.6 E and F). ICA was also done on the principal components of image sequence and
 496 speed frames obtained earlier. These analyses yielded results and conclusions similar to those for ICA on
 497 the image sequence and speed frames. For the Max and Min projections of ICs, spatial entropy, sharpness,
 498 and Hausdorff fractal dimension measures were determined and compared for groups of Black and White
 499 mice. In comparisons where differences were significant are shown in Fig. 3.6G. The spatial entropy as
 500 well as sharpness was larger for White mice compared to Black supporting results reported above.

Figure 3.6. Independent component analysis captures regions with high spatiotemporal contrast. (A) Representative independent components for Black mouse. Individual independent components mostly represent snapshots of position of the hands and strings, as they are most dynamic in the image sequence. (B) Maximum (Max) and minimum (Min) value projections of all independent components of image sequence and speed frames. Max and Min projections grossly represent locations of sudden changes happening in an image sequence e.g., hand appearing or disappearing at a new location. (C) and (D) similar to (A) and (B) respectively but for White mouse. (E) Representative image frames of synthetic mouse data in which only hands moved. (F) Similar to (B) but for synthetic data. (G) Mean $\pm$ SEM of spatial entropy and sharpness of Max projections of ICs for image sequence, speed frames and their respective PCs.

3.7 Descriptive statistics on image sequences of masks of body, ears, nose, and hands

Descriptive statistics on the masks' image sequence can reveal useful information about the extent of motion of individual body objects, e.g., the standard deviation of body, ears, nose, and hand masks as shown in Fig. 3.7 A and B for representative Black and White mice provides the variability in the position of respective objects. One can easily see individual differences between the representative Black and White mice e.g., the body positions of a Black mouse are distinct as seen by crisp standard deviation frame compared to a White mouse which has diffuse appearance. Comparing the spatial measures between groups of Black and White mice revealed significantly higher spatial entropy for mean, standard deviation, skewness, and kurtosis frames for White mice ($p = 0.006$, $p = 0.008$, $p = 0.006$, $p = 0.009$ respectively, Fig. 3.7C). The sharpness of skewness was also larger for White mice ($p = 0.022$). The spatial measures were not different for ears, nose, and hands.

Figure 3.7. Image segmentation based on object colors for finding object masks and gross assessment from descriptive statistics of masks. (A-B) Standard deviation of the masks image sequence of body, ears, nose, and hands for overall assessment of respective motion patterns of Black and White mice respectively. Shown below each frame are the respective values of spatial entropy (SE), sharpness (S), and Hausdorff fractal dimension (HFD) separated with a hyphen. **(C)** Mean±SEM SE, S, and HFD shown for Body masks

3.8 Identification of fur, ears, and nose for quantification of body and head posture of Black mice

Body length, body tilt, head roll, head yaw, and head pitch are estimated from the masks for fur, ears, and nose (Fig. 3.8). From the mask for fur, the mouse's body centroid and body fit are estimated. Since in the fur mask, segregated regions may be segmented because of an overlapping object on the body

such as the string in Fig. 3.8 A1, a convex hull (Matlab function `bwconvhull`) operation is applied to combine regions and find a cumulative area representing fur (Fig. 3.8 A2, also see Supplementary Fig. S1). Matlab function “`regionprops`” is then used to identify regions in the cumulative mask and their properties e.g. area, centroid, orientation, major and minor axis length of an ellipse fitted on the region (Fig. 3.8 A7, green). Major axis length and orientation of the ellipse are then used as estimates of body length and tilt respectively. In the example string pulling epoch in the sample video (lasting about 8 secs), the starting body length is about 40mm and increases by ~1.5% as the mouse stands and elongates its body to virtually walk up on the string (Fig. 3.8 B). The body tilt however remains vertical (between 80°-90°) throughout the epoch (Fig. 3.8 C).

Figure 3.8. Quantification of body and head posture. **A)** Tagging of body, ears, and nose. From masks found with fur color (A1), body region (A2) is identified and fitted with an ellipse (A7, green). From the ears’ mask (A3), ear regions are identified by using body centroid position found earlier (A4, cyan and magenta dots in A7). Nose mask (A5) is used to identify the nose region (A6), which is fitted with an ellipse to identify nose position (A7, yellow dot). **B)** Body length vs time. Body length is the length of major axis of the body fitted ellipse. **C)** Body Angle vs time. Body angle is that of the major axis of the body fitted ellipse from horizontal. **D)** Shortest distance of the nose from the string vs time. **E)** Head Yaw vs time. Head yaw is estimated by finding the log of the ratio of distance of right ear from the nose to the

distance of left ear from the nose (ratio of lengths of White lines in A7). A value of 0 indicates 0 yaw angle. **F)** Head Roll Angle vs time. Head roll is the angle of the line joining ear centroids from horizontal using the right ear as the origin. **G)** Head Pitch vs time. Head pitch is estimated as the perpendicular distance between the nose and a straight line joining the ears (small red line in A7). Positive and negative values correspond to the nose above or below the line respectively.

To estimate head yaw, roll, and pitch, regions representing the right and left ears and the nose are first identified using the respective masks and the elliptical fit for the body found earlier. An elliptical mask is generated for the body using the elliptical fit and the intersection of an inverse of the elliptical mask (converting 0's to 1's and vice versa) with that of ears to provide regions that are outside the ellipse for body (Fig. 3.8. A4). If segmented regions are close, they are combined (Supplementary Fig. S1) and very small regions are ignored. In data for the Black mouse, only two regions remained after the end of this procedure and they were easily labeled as the right and left ears by finding their centroid location relative to body centroid i.e., regions left and right to the body centroid would be right and left ears respectively. However, if multiple regions remain after the intersection procedure, regions overlapping (or closest) to ear regions in a previous frame are chosen. Spatial clustering is also applied to combine fragmented regions that are close together. The centroids of the identified ear regions are used to represent their location (Fig. 3.8 A7, cyan and magenta dots). Next the nose is identified from the nose mask (Fig. 3.8 A5). After identifying spatial regions, clustering is first used to combine regions that are close together (Supplementary Fig. S1) and then the region closest to the top of the body ellipse is labeled as the nose region (Fig. 3.8 A6). An ellipse is fitted onto the nose region and the top of this ellipse is used as a single point to represent nose location (Fig. 3.8 A7, yellow dot). Head yaw is then estimated as the log of the ratio of distance between the nose and the right ear to the distance between nose and left ear. A head yaw value of 0 indicates 0 head yaw angle which means that the face of mouse is facing the camera. In the representative string-pulling epoch of the Black mouse, head yaw values are initially greater than 0 and gradually reduce to values less than 0, indicating that the mouse is moving its head from left to right (Fig. 3.8 E). Head roll angle is estimated by finding the angle of line joining the centroids of the regions representing the right and left ears from the horizontal (Fig. 3.8 A7, long red line). While the mouse pulls the string, the head rolls at a frequency of about 3 rolls/sec (Fig. 3.8 F) in sample video. However, there seems to be a bias for this mouse to keep its head titled such that the right ear is slightly higher than the left ear. Head pitch is estimated as the perpendicular distance between the nose and line joining the two ear centroids (Fig. 3.8 A7, small red line). Positive and negative values of head pitch correspond to nose

position above or below the line joining the ears respectively. In the sample video, the head pitch stays positive most of the time, as the mouse is moving its nose up for sniffing or using its whiskers to track the string (Fig. 3.8G). Furthermore, as has been previously shown (Blackwell, Banovetz et al. 2018), the nose stays close to the string (Fig. 3.6D).

3.9 Identification of right and left hands and kinematic measures of hand movement of Black mice

Since both hands move rapidly and have similar colors, it is a challenge to distinctly identify the right and left hands without using a heuristic-based approach. To make automatic identification simpler, we let the user initialize the location of hands in the starting frame, the first frame of an epoch, where a user manually marks two regions representing right and left hands respectively. In the subsequent frames, automatic identification of the hands in a frame is achieved by using information of the hand location in the previous frame. Automatic identification uses masks for the hands and finds regions using “regionprops” function of Matlab. As a first step, the number of regions is reduced such that regions that mostly likely would represent the hands are; kept e.g., regions with very small areas (less than 20mm²) are neglected (Fig. 3.9 A2 represent selected regions after extra regions were neglected). Close by regions are also combined using spatial clustering (Supplementary Fig. S1). Once a set of reduced number of regions is obtained, the algorithm (Supplementary Fig. S2) identifies the left and right hands. If only one region remains, the algorithm assumes that the hands are touching each other. It divides the region into two equal regions and labels the subregions as left and right hands based on their distances from left- and right-hand regions in the previous frame (Supplementary Fig. S2, Fig. 3.9 A3 and A4). If two distinct regions remain, the same algorithm is used. If more than two regions are identified, regions are further reduced based on their distances from (and overlaps between) left- and right-hand regions in the previous frame and the regions are fed to the same algorithm in a recursive manner.

The regions representing hands were also determined by first finding maximally stable extremal regions (MSER) in the original frame (Obdrzalek D. 2009). Using the Matlab function “detectMSERFeatures” MSERs were found (Fig. 3.9 B1). Color definition for the hands was then used to find MSERs that contained hand colors and a mask was thus generated (Fig. 3.9 B2). For the representative Black mouse video, hand masks generated this way were in general better than those created with the “rangeseach” function (see above); i.e., they had lesser number of spurious regions. These masks were then treated as described above to find regions representing the left and right hands (Fig. 3.9 B3 and B4).

Both methods of finding hand regions are made available in the graphical user interface using choices that can be made with check boxes.

617

Figure 3.9. Identification of hands and kinematic measures. **A)** Representative frame (127) with overlaid regions in hand masks (A1). Regions in A1 are reduced to find regions that would most likely represent the hands (A2). Note that it includes a region in the nose area. Using the position of hands in the previous frame (blue and red outlines in A3), the hands are identified in the current frame (A4). **B)** same as in A but masks are first found by identifying maximally stable extremal regions in the original image (B1). **C), D)** Vertical (Vert) and Horizontal (Horz) positions vs time respectively of right (blue) and left (red) hands. Positions are measured from the lower left corner of frame as the origin. Where horizontal movements are synchronized, vertical movements of right and left hands are alternating. **E), F)** Vertical and horizontal speed vs time of right and left hands. Maximum vertical speed is about three times larger than maximum horizontal speed.

628

Since the automatic algorithm is not deterministic but rather makes estimates using heuristics, it is possible that the algorithm incorrectly labels regions as either the right or left hand. In this case, the user can stop the automatic process and manually choose the appropriate identification in an individual frame.

631

632 The automatic identification of hands in a frame depends upon the position of hands in the previous frame,
633 therefore the user might have to reprocess all subsequent frames.

634 Once the positions of the hands are identified in all frames in behavioral epochs of interest i.e. the
635 x and y coordinates of centroids are found. They can be used to calculate kinematic measures. For
636 example, paths of hand motions can be visualized or the vertical and horizontal positions of each hand
637 from an origin point (lower left corner of the original frame) can be determined and plotted (Fig. 3.9 C
638 and D). In the representative epoch of string pulling for the Black mouse, the stroke length of pulling is
639 initially shorter and gradually grows for both right and left hands. With respect to the common origin, for
640 the vertical motion, the two hands are alternating, whereas for the horizontal motion, the two hands move
641 to the right and left together. From the position plots, the speed of the hands can be found by finding a
642 time derivative. The vertical speed of the hands is about 2-3 times faster than the horizontal speed. During
643 the whole epoch, there are two bouts of rapid pulling each about 2 secs long (from 2-4 secs and from 6-8
644 secs). During these bouts, the path shapes of strokes of one hand were similar to each other and to the
645 strokes of the other hand (Fig. 3.9G). For each stroke the rise time i.e., motion from point of release of the
646 string (near the bottom) to the point of the grasp (near the top) is smaller than the fall time which is the
647 time mouse is pulling the string downwards (Fig. 3.9H). For the horizontal motion however, the strokes
648 are bimodal (Fig. 3.9I), and the rise time is larger than the fall time (Fig. 3.9J).

649 **3.10 Quantification of body and head posture and kinematic analyses of White mice and comparison** 650 **with Black mice**

651 Distinguishing between the fur and other body parts (hands and ears) was difficult for the White
652 mice, resulting in the generation of poor-quality masks. Therefore, using sharpie markers, we painted the
653 hands and ears of the mice with blue and green colors respectively (Fig. 3.10 A1). After painting, image
654 segmentation for identifying the ears and the hands was easier (Fig. 3.10 A3 and A6 respectively). All
655 topographical and kinematic parameters for head posture and motion of the hands were determined (Fig.
656 3.10 B-E) in a similar fashion as described above. Whereas the Black mouse gradually stood up, increasing
657 its body length while pulling, the White mouse moved the whole body in a rhythmic fashion up and down
658 (Fig. 3.10B). The White mice occasionally used its mouth to hold the string (Online Video, V2).

659 There was no significant difference between maximum body length, minimum body length, and
660 change in body length (max – min) between White and Black mice, Student's t-test, $p > 0.05$. However,

body linear and angular speeds were significantly larger for White compared to Black mice ($p = 0.0198$ and $p = 0.0199$ respectively, Fig. 3.10F). The dominant frequency in body length oscillations (determined with Fourier transform of the time series of body length) was not different between Black and White mice. Similarly, the dominant frequency of motion for hands was not different and so was velocity. However, the amplitude of right-hand's vertical movement was larger for Black compared to White mice ($p < 0.001$, ks-test).

Figure 3.10. Identification of body parts and quantification of kinematic measures in White mice. **A)** Procedure for tagging of body, ears, nose, and hands was like that used for Black mice except that the ears and hands were colored green and blue respectively. From masks found using fur color (A2), the body region is identified and fitted with an ellipse. Ear and nose regions (A3 and A4 respectively) are identified and used to measure head posture (A5). Hand regions (A6) are identified based on the position of the hands in the previous frame (A6, red and blue boundaries), the position of the hands in the current frame are identified (A7). **B)** Body length vs time. **C)** Head Pitch vs time. In frames where both ears are not visible, y-distance between nose and ear is measured as head pitch. **D)** Vertical position of the hands (from bottom of frame) vs time. **E)** Shortest distance of the nose from the string vs time. **F)** Mean $\pm$ SEM of body linear and angular speeds from 5 Black and 5 White mice. **G)** Distributions of the amplitudes of all reach cycles (lift-advance-grasp-pull-push-release) of right-hand pooled from 5 Black and 5 White mice and corresponding mean speed illustrating strain-related differences in pulling movements.

3.11 Benchmarking of algorithms for finding body parts

To determine the accuracy of heuristic algorithms presented here, the percentage of frames in which body parts were detected automatically were calculated (Table 3.1). The overall average accuracy for all body parts is ~96%, with higher accuracies for fur and ears compared to those for nose and hands. This is because the nose is sometimes hidden behind the string, making detection difficult for the algorithm while the hands are moving at a rapid rate and the algorithm cannot differentiate between regions in the mask that belong to hands versus those that belong to hand-like body parts e.g., nose.

Table 3.1. Accuracy of algorithms. Percentage of frames in which a body part was detected automatically. N indicates the number of frames of the string-pulling epoch that was analyzed. (N/A indicates not applicable and values were not used for calculating the overall accuracy. For these situations, either color of a body part was not separable in masks or due to limitation of the camera, hands were blurry in most of the frames)

Mouse / Body Part	Body	Ears	Nose	Hands
Black Mouse 1 (N = 477 of 630)	100	100	98.32	98.74
Black Mouse 2 (N = 132 of 245)	100	100	96.12	87.12
Black Mouse 3 (N = 351 of 1435)	100	97.14	92.88	81.48
Black Mouse 4 (N = 191 of 629)	100	97.38	100	89.53
Black Mouse 5 (N = 171 of 808)	100	100	98.83	93.57
Simulated Black Mouse (N = 300 of 477)	100	100	100	99.67
Mean (Black Mice)	100	99.08	97.17	91.70
White Mouse 1 (N = 275 of 990)	100	96.36	82.91	94.91
White Mouse 2 (N = 300 of 594)	99.67	99	100	91
White Mouse 3 (N = 251 of 453)	100	94.02	N/A	60.16 (N/A)
White Mouse 4 (N = 166 of 421)	99.40	90.96	N/A	30.12 (N/A)
White Mouse 5 (N = 216 of 930)	100	N/A	N/A	0 (N/A)
Mean (White Mice)	99.80	95.10	91.45	92.95

3.12 Neural networks for identifying ears, nose, and hands

We also trained neural networks to analyze string-pulling videos to detect ears, nose, and hands in Black mice. We used the Python based frame work of Deeplabcut toolbox (Mathis, Mamidanna et al. 2018, Nath, Mathis et al. 2018) to train the ResNet50 network for identifying ears, nose, and hands (He, Zhang et al. 2016). Four separate networks were trained; three for the individual recognition of the ears, nose, and hands and one for the combined recognition of all three in a single step. Details of training the network and using it to analyze videos can be found in (Mathis, Mamidanna et al. 2018). For training each neuronal network, 54 frames were labelled (18 from three different videos) and used to create the training dataset. For each network, training took about 36 hours on a GPU (NVIDIA GeForce GTX 1080 with 8GB memory) and on a 64-bit Windows 10 computer with two processors (Intel® Xeon® CPU E5-2620 v4 @ 2.10 Gz) and 256GB RAM. Training iterations were 1030000 and the train and test errors were as follows in pixels. For network trained to identify; 1) only hands, errors were 3.3 and 3.94, 2) only nose, errors were 4.19 and 8.74, 3) only ears, errors were 4.76 and 11.85, and 4) all objects, errors were 5.07 and 10.46.

For the epoch shown in previous figures for the Black mouse, the networks (separate for ears, nose, and hands) faithfully identified the ears, nose, and hands, although with some errors (Fig. 3.11). Errors were mostly negligible for the identified position of ears and nose (Fig. 3.11 B, C, and F). However, errors were up to ~100 pixels (pix) (10mm) for the identification of hands in frames where the speed was large (Fig. 3.11 A, D, and E). These errors can be reduced if further training of previously trained networks is done with more frames. For now, we offer an option to correct for these errors using the provided graphical user interface in which Deeplabcut results can be loaded and manually curated. Our method can provide ground truth data for validating the results of neural networks-based tagging of body parts.

Figure 3.11. Comparison of identification of ears, nose, and hands with heuristic algorithms and machine learning based neural networks. A) Representative frames highlighting the differences in position of ears, nose, and hands identified with the two methods. Dots for heuristics and plus for neural networks. B-F) Differences in x and y coordinates of ears, nose, and hands identified with the two methods.

3.13 Graphical user interface of toolbox and procedure for processing video data

The details of the graphical user interface of the toolbox and procedure for processing video data are presented in the Supplementary material (see supplementary Fig. S3). The software can be downloaded from the following GitHub link along with two sample videos from Black and White mice whose analysis is presented in this paper. https://github.com/samsoon-inayat/string_pulling_mouse_matlab

The tagged video for the Black mouse is shown in supplementary video.

The above-mentioned videos along with other videos and processed data can be downloaded from Open Science Framework https://osf.io/gmk9y/?view_only=cdf229798f2d40e08b98aeadd927f1c2

4. Discussion

The description of behavior is central to understanding neural organization and function and accordingly behavior is frequently fractionated to describe the contribution of different body effectors. Fraction often misses the allied contributions of other body parts. For example, in using a hand to reach,

detailed descriptions of hand movement often neglect the contribution of postural adjustments of the body. The present procedure provides an integrated view of a prototool use behavior in which skilled hand movements are described along with the contributions of body movement, posture, and orientation. The procedure captures strain differences in the two exemplar strains of mice that are examined, which shows that the procedures can be used for phenotyping strain differences more generally. In addition, the behavior of string-pulling is homologous to string-pulling described in many other animal species including humans, suggesting that it can be exploited for species comparisons and for developing animal models of human neurological conditions (Ryait, Bermudez-Contreras et al. 2019).

We demonstrate phenotyping motor behavior of C57BL/6 (Black) and Swiss Webster (White) mice and present findings that White mice have larger variation in positions of the body, larger body speed but smaller amplitude of string-pulling motion as compared to Black mice. These findings corroborate earlier findings regarding the motion comparison of these two strains after complete transection of the spinal cord that White mice showed larger mobility post-transection compared to Black mice (Noristani, They et al. 2018). Differences between these two strains have also been reported for other physiological processes e.g., fatty acid uptake and trafficking (Seeger and Murphy 2016).

We present a Matlab based toolbox for seamlessly processing mouse string-pulling video data to perform whole-body assessment of position and speed as well as automatically identify the body, ears, nose, and hands in individual video frames. Quantification of the body and head postures is presented using estimates of body length and tilt, and head yaw, roll, and pitch angles while kinematic measures are determined for the hands. Kinematic measures for body and head can also be calculated from the quantifications provided. This toolbox is provided under the GNU General Public Licence v3.0, and is open source, free, and any modifications are allowed as long as this paper is cited when publishing or distributing a modified version. Here we discuss characteristics and limitations of the toolbox as well as comparison with other similar software.

4.1 Data compression for efficient storage of masks

While writing this software, data compression techniques were used to minimize storage of calculated masks in addition to the provision provided to select a zoom window (see section 3.4) which uses a subset of frames to process. Since each pixel of a mask is either 1 or a 0, it can occupy a single bit in a 1-byte number. This way individual bits of a byte represent a single pixel of an individual mask while

the whole byte can be used to store one pixel for eight masks. Using this methodology, 5 masks are stored in an array with each element 1 byte long.

4.2 Limitations of the software and data acquisition strategy to improve accuracy

Automatic identification of body parts is attempted using a heuristics approach which requires consistent input data, i.e., the videos be recorded in a consistent manner. The number of pixels in a frame, frame rate, lighting conditions, and overall position of the mouse in video frames, if consistent from video to video, will make processing effective. If these parameters are changed, thresholds that are set in the software can be readjusted. These thresholds are mentioned in the “selectAppropriateRegions” function in the toolbox and in individual functions to identify body parts. Due to the use of a heuristic approach, sometimes the algorithm fails to correctly identify regions for hands. In cases where a Matlab error occurs while identifying the hands, the user is given the option for manual tagging. Nevertheless, in case of no Matlab error, the algorithm can still incorrectly label a region as a nose rather than a hand. In such a case, the user has to manually observe frames for accurate detection and in case of erroneous tagging, use the manual tagging option provided in the GUI. Furthermore, the erroneous tagging in current frame can continue into subsequent frames because identification of the hands in the current frame depends on correct identification in a previous frame. Therefore, the recommendation is to process the 20 frames that are displayed along with GUI together. In the sample data of the Black mouse, less than 2% of frames (6/477) were incorrectly labelled for hands and had to be tagged manually. The other case in which erroneous tagging might happen includes videos in which a mouse is pulling a string when turned sideways. In this case only one ear is visible. Then, when both ears appear again, the algorithm still detects one ear. The user has to again manually initialize the detection of the other non-detected ear for the software to automatically detect the ear in subsequent frames.

Since, the heuristic algorithms rely on high-contrast image sequence, the choice of colors for background and string can affect the outcome e.g., if for a White mouse, white background or string is chosen, the software would not be able to distinguish between string and body (fur of the animal). Similarly, if shutter speed of the camera is not high enough, frames in the image sequence will have blurry objects particularly hands as they have high speed motion. The software will then have difficulty in resolving colors of hands from blurry images. Lighting conditions should also be such to enhance the contrast of acquired images. We recommend using front facing cold light (on the same side as the camera),

i.e., light is in front of the animal. High resolution of the video is also preferred for better pixel resolution for resolving body parts (HD 1920 x 1080 is recommended). Note that these requirements are important for identifying and tracking body parts in videos but whole-body assessment can still be done on low to medium quality videos.

4.3 Comparison with artificially intelligent machine learning based software

Machine learning (ML) based software for object recognition in videos can provide a seamless work flow, but they are computationally intensive initially when training is required (Mathis, Mamidanna et al. 2018, Arac, Zhao et al. 2019). For identifying a new/different object, or for improving the accuracy of detection (Fig. 3.11), usually retraining must be done. The algorithms we present here are simpler, require smaller computations, and can be implemented on any computer with reasonable computing power and temporary storage e.g. 2GHz processor and 32GB of RAM. In comparison, ML based software require high end computers ideally with graphical processing units.

4.4 Possible extensions of the toolbox and the string-pulling behavioral task

Additional functionality can be added to the toolbox regarding detection of phases of string-pulling movements e.g., when the hands grasp and release the string. String masks are available in the toolbox to add this functionality. The heuristic algorithms can also be easily modified by changing the appropriate functions. The string-pulling task can also be enriched with sensory, motor, and behavioral complexity. For example, sensory complexity would involve strings with different cross-sections, texture, or coated with odors. Motor complexity would involve adding tension or variable load to the string. Behavioral complexity would involve a paradigm in which the animal has to learn to associate string-pulling with a reward e.g., water. Yet another possibility could be to make the animal learn a go-no-go task with different lengths of the string. Social competition or social altruism can also be studied with string-pulling. For example, in a competitive strategy, two animals will pull the string from two ends to obtain reward attached in the middle. The scenario could be presented to study motivation before and after social defeat or stress. In experiments to study altruism, string pulling of one animal could give the reward to another animal. Motor memory formation and the role of sleep in its consolidation could also be studied (Inayat, Nazariahangarkolaee et al. 2019). We anticipate that this toolbox holds promise for facilitating high-throughput platform for the behavioral screening of drugs that may help in the treatment of Parkinson,

Huntington, Alzheimer and similar diseases in addition to brain injuries (i.e., ischemic stroke, spinal cord injury, etc.).

825

826 **FUNDING**

This study was supported by research fund provided by the Canadian Institutes of Health Research (CIHR) Grant# 390930, Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant #40352, Alberta Innovates (CAIP Chair) Grant #43568, Alberta Alzheimer Research Program Grant # PAZ15010 and PAZ17010, and Alzheimer Society of Canada Grant# 43674 to MHM.

831 **ACKNOWLEDGEMENTS**

We thank Di Shao for animal husbandry and Behroo Mirzagha for her help in collecting video data.

833

834 **AUTHOR CONTRIBUTIONS**

SI, SS, IQW, and MHM designed the study. Q, AG, and PE trained mice for string pulling and collected video data. SI developed the analyses framework and Matlab toolbox, AG and SS participated in the development and performed quality control. SI, AG, and SS performed the analyses of video data. SS generated synthetic data. SI trained neural networks. SI and IQW wrote the manuscript, which all authors commented on and edited. IQW and MHM supervised the study.

840

841 **SUBMISSION & CORRESPONDENCE INFORMATION**

Dr. Majid Mohajerani, mohajerani@uleth.ca

843

844 **WORK PERFORMED**

All the work was performed at the Canadian Center for Behavioral Neuroscience, University of

846 Lethbridge, Alberta, Canada

847

848 DISCLOSURE STATEMENT

849 None.

850

851 References

- 852 Afrashteh, N., S. Inayat, M. Mohsenvand and M. H. Mohajerani (2017). "Optical-flow analysis toolbox for
853 characterization of spatiotemporal dynamics in mesoscale optical imaging of brain activity." Neuroimage
854 **153**: 58-74.
- 855 Arac, A., P. Zhao, B. H. Dobkin, S. T. Carmichael and P. Golshani (2019). "DeepBehavior: A Deep Learning
856 Toolbox for Automated Analysis of Animal and Human Behavior Imaging Data." Frontiers in Systems
857 Neuroscience **13**(20).
- 858 Blackwell, A. A., M. T. Banovetz, Qandeel, I. Q. Whishaw and D. G. Wallace (2018). "The structure of arm and
859 hand movements in a spontaneous and food rewarded on-line string-pulling task by the mouse." Behav Brain
860 Res **345**: 49-58.
- 861 Blackwell, A. A., J. R. Köppen, I. Q. Whishaw and D. G. Wallace (2018). "String-pulling for food by the rat:
862 Assessment of movement, topography and kinematics of a bilaterally skilled forelimb act." Learning and
863 Motivation **61**: 63-73.
- 864 Blackwell, A. A., W. L. Widick, J. L. Cheatwood, I. Q. Whishaw and D. G. Wallace (2018). "Unilateral forelimb
865 sensorimotor cortex devascularization disrupts the topographic and kinematic characteristics of hand
866 movements while string-pulling for food in the rat." Behav Brain Res **338**: 88-100.
- 867 Bruhn, A., J. Weickert and C. Schnorr (2005). "Lucas/Kanade meets Horn/Schunck: Combining local and
868 global optic flow methods." International Journal of Computer Vision **61**(3): 211-231.
- 869 Dunnett, S. B. and S. P. Brooks (2018). "Motor Assessment in Huntington's Disease Mice." Methods Mol Biol
870 **1780**: 121-141.
- 871 He, K., X. Zhang, S. Ren and J. Sun (2016). Deep Residual Learning for Image Recognition. 2016 IEEE
872 Conference on Computer Vision and Pattern Recognition (CVPR).
- 873 Higuchi, T. (1988). "Approach to an irregular time series on the basis of the fractal theory." Phys. D **31**(2):
874 277-283.
- 875 Hyvarinen, A. and E. Oja (2000). "Independent component analysis: algorithms and applications." Neural
876 Netw **13**(4-5): 411-430.
- 877 Inayat, S., M. Nazariahangarkolaee, S. Singh, B. L. McNaughton, I. Q. Whishaw and M. H. Mohajerani (2019).
878 "Low acetylcholine during early sleep is important for motor memory consolidation." Sleep.
- 879 Jacobs, I. F. and M. Osvath (2015). "The string-pulling paradigm in comparative psychology." J Comp Psychol
880 **129**(2): 89-120.
- 881 Laidre, M. E. (2008). "Spontaneous performance of wild baboons on three novel food-access puzzles." Anim
882 Cogn **11**(2): 223-230.
- 883 Liu, C. (2009). Beyond Pixels: Exploring New Representations and Applications for Motion Analysis. PhD,
884 Massachusetts Institute of Technology.
- 885 Mathis, A., P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis and M. Bethge (2018). "DeepLabCut:
886 markerless pose estimation of user-defined body parts with deep learning." Nat Neurosci **21**(9): 1281-1289.

887 Nath, T., A. Mathis, A. C. Chen, A. Patel, M. Bethge and M. W. Mathis (2018). "Using DeepLabCut for 3D
888 markerless pose estimation across species and behaviors." bioRxiv: 476531.

889 Noristani, H. N., L. They and F. E. Perrin (2018). "C57BL/6 and Swiss Webster Mice Display Differences in
890 Mobility, Gliosis, Microcavity Formation and Lesion Volume After Severe Spinal Cord Injury." Front Cell
891 Neurosci **12**.

892 Obdrzalek D., S. B., L. Mach, and A. Mikulik. (2009). "Detecting Scene Elements Using Maximally Stable Colour
893 Regions." Communications in Computer and Information Science **82**: 107-115.

894 Ryait, H., E. Bermudez-Contreras, M. Harvey, J. Faraji, B. Mirza Agha, A. Gomez-Palacio Schjetnan, A. Gruber,
895 J. Doan, M. Mohajerani, G. A. S. Metz, I. Q. Whishaw and A. Luczak (2019). "Data-driven analyses of motor
896 impairments in animal models of neurological disorders." PLoS Biol **17**(11): e3000516.

897 Seeger, D. R. and E. J. Murphy (2016). "Mouse Strain Impacts Fatty Acid Uptake and Trafficking in Liver, Heart,
898 and Brain: A Comparison of C57BL/6 and Swiss Webster Mice." Lipids **51**(5): 549-560.

899 Singh, S., A. Mandziak, K. Barr, A. A. Blackwell, M. H. Mohajerani, D. G. Wallace and I. Q. Whishaw (2019).
900 "Human string-pulling with and without a string: movement, sensory control, and memory." Exp Brain Res
901 **237**(12): 3431-3447.

902 Spasic, S., A. Kalauzi, M. Culic, G. Grbic and L. Martac (2005). "Estimation of parameter kmax in fractal analysis
903 of rat brain activity." Ann N Y Acad Sci **1048**: 427-429.

904 Whishaw, I. Q. (1996). "An endpoint, descriptive, and kinematic comparison of skilled reaching in mice (*Mus*
905 *musculus*) with rats (*Rattus norvegicus*)." Behav Brain Res **78**(2): 101-111.

906

907

908

909

Supplementary Material

A toolbox for automated video analysis of rodents engaged in string-pulling: Phenotyping motor behavior of mice for sensory, whole-body and bimanual skilled hand function

Samsoon Inayat¹, Surjeet Singh¹, Arashk Ghasroddashti¹, Qandeel¹, Pramuka Egodage¹, Ian Q. Whishaw¹, and Majid H. Mohajerani^{1,4}

3. Canadian Centre for Behavioural Neuroscience, University of Lethbridge, Lethbridge, Alberta, Canada

4. Corresponding authors: samsoon.inayat@gmail.com, mohajerani@uleth.ca

923 The following figures provide flow charts of algorithms for spatial clustering of regions and finding left
924 and right hands from two given regions.

925

926 **Figure S1. Algorithm to reduce the number of regions by spatial clustering.** Close by regions are
927 combined provided that the combined region's area is less than the threshold and the length of the major
928 axis of fitted ellipse is less than 7mm. For ears and nose, only the length of major axis of fitted ellipse
929 should be less than 9 and 8 mm respectively.

930

Figure S2. Algorithm to label right- and left-hand regions from two unlabelled regions. The algorithm primarily uses distances of unlabelled regions from the right- and left-hand regions in the previous frame to label them as right and left hands.

Graphical user interface of toolbox and procedure for processing video data

We present a simple graphical user interface (GUI) which is self explanatory in terms of usage. To initiate the software, the user runs “string_pulling_behavior_analytics.m” file in Matlab which opens the GUI window. Clicking the open file button allows the user to select a video file for processing. It also lists 10 recently used files to make the selection easier for reopening a formerly processed or partially processed file. Once a file selection is successful, the file is loaded and detailed GUI options become visible (Fig. S3). The GUI window has been divided into four panels to run processing steps and one panel for post-processing steps i.e., for looking at results and plots. A display window showing the first 20 frames also appears. Using the panel “Control Frame Display”, the user can browse through the video. The number of

frames in the display window can be changed by putting the number of rows and columns in the provided edit boxes. The analysis can be done in four steps:

947

Figure S3. Graphical user interface of the toolbox. Individual steps for processing data are separated into panel blocks. The video frames can be viewed in the display window using the “Control Frame Display” panel. Step 0, the user has to define a zoom window for digital magnification of the video and scale for finding pixels/mm. Step 1, the user specifies epochs in which the mouse pulls the string. Step 2, the user defines colors. Step 3, the user finds zoom window, a sub region of the frames for which masks are calculated. Step 4, the user processes the epoch for whole-body analysis through “Gross Motion Assessment” panel and also finds masks and later objects of interest. Finally, the user can generate plots using the “Post-Processing” panel.

956

957 *Step 0 – Defining zoom window and measurement scale.* A user defines a zoom window to select
 958 four corner points for choosing subsets of frames to view in the display window. Since the captured frames
 959 might have additional areas surrounding the container in which mouse pulls the string, a zoom window
 960 allows view regions in which there is a mouse. The zoom window can be reset and set as required. Next,
 961 the user defines a scale by drawing a line in a frame on an object of known length and specifying its length
 962 in mm. For the current video data, the thickness of the Plexiglas of the cage, which is 5mm, is used to
 963 define a scale in pixels/mm.

964 *Step 1 – Defining epochs.* The user defines starting and ending frames of epochs in which, the
 965 mouse pulls the string. The table in the GUI allows multiple epochs of interest. After entering frame
 966 numbers in the table, clicking in the epochs table displays that frame as the starting frame in the display
 967 window. Epoch definitions are necessary to proceed further in the analysis.

968 *Step 2 – Defining colors.* As described in the main text, the user defines colors for the fur (body),
 969 ears, nose, hands, and string. An epoch should be chosen to define colors.

970 *Step 3 – Defining the zoom window for finding and storing masks.* To reduce computational time
 971 and save disk space, only a subset of frames is used to find masks and perform further analyses. By
 972 pressing the “Find Body Box” button, the software automatically finds the region in frames where the
 973 mouse is present and uses this region for finding and storing masks. The user can also find head boxes
 974 around the head. The margins on the right of in this panel allows the user to adjust how much space around
 975 the mouse body or head should be kept.

976 *Step 4.* In this step, the user can do whole-body analysis using optical flow analysis, descriptive
 977 statistics, principal and independent component analysis. Furthermore, in this step one can find masks and
 978 objects (body, ears, nose, and hands) in frames of choice. The best way to process the data for finding
 979 objects is 20 frames at a time, which is the number shown in the display window. After processing the
 980 user can manually verify for the accuracy of automatic identification and if an incorrect tagging is
 981 observed, the user can then manually correct it by selecting the frame and using the “Find Manually”
 982 button.

983 *Post-processing.* Finally, the user can post-process data using the “Post-Processing” panel e.g. find
 984 the shortest distance between the nose and string and generate plots for selected frames. The data can be

exported to a Microsoft Excel sheet. If one has used the DeepLabCut toolbox for identifying the position of the ears, hands, and nose, they can load the data here and manually assess the results. A labelled video of selected frames can also be made.

Matlab dependencies

Our String-Pulling Behavioral Analytics toolbox requires the following Matlab Toolboxes

- 1) Signal processing
- 2) Image processing
- 3) Statistics and Machine Learning
- 4) Computer vision system