TOOLS FOR BEHAVIORAL PHENOTYPING OF C. ELEGANS

A Dissertation
Presented to
The Academic Faculty

by

Kathleen Bates

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Bioengineering

Georgia Institute of Technology
August 2020

TOOLS FOR BEHAVIORAL PHENOTYPING OF C. ELEGANS

Approved by:

Dr. Hang Lu, Advisor
School of Chemical & Biomolecular Engineering
Georgia Institute of Technology

Dr. Patrick McGrath
School of Biology
Georgia Institute of Technology

Dr. Gordon Berman
Department of Biology
Emory University

Dr. Mark Styczynski
School of Chemical & Biomolecular Engineering
Georgia Institute of Technology

Dr. Daniel Goldman
School of Physics
Georgia Institute of Technology

Date Approved: March 11, 2020

ACKNOWLEDGEMENTS

Many people have played important roles in the creation of this document, but I owe my advisor, Hang Lu, the most gratitude. Hang’s scientific advice and tolerance have been both necessary and appreciated. To my other committee members I also owe sincere thanks for their advice and expertise.

To the many labmates and friends who have provided pep talks when I needed them, and distracted me from failed experiments, I deeply appreciate your support. I owe special thanks to those who have directly contributed to this work, in particular Lucinda Peng, Jimmy Ding and Caden Jiang, who have variously helped with experiments, proofreading, and interface design. You have all been great collaborators and have always lent a helping hand when I needed it. Finally, Wei, thank you for making me feel that I belonged.

Above all, I thank my family. Cameron, without seeing you test the doctoral waters first, I wouldn’t have considered it. Dad, I have always admired your quiet and kind intelligence and creativity; this is proof of my attempts to emulate you. Mom, I am so thankful to have a feisty engineer as a mom. I could hardly have found a better role model than you. Thank you endlessly for your love, support, patience, inspiration, and everything else under the sun.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

C. elegans	*Caenorhabditis elegans*
E. coli	*Escherichia coli*
ATR	All-trans retinol
ChR2	Channel rhodopsin 2
GFP	Green Fluorescent Protein
M9	Buffer suitable for suspending *C. elegans*
FoV	Field of View
OP50	Strain of *E. coli* used as feedstock for *C. elegans*
PDMS	Polydimethylsiloxane, used for fabrication of microfluidic devices
t-SNE	t-distributed stochastic neighbor embedding
PDF	Probability density function
2D	2-dimensional
3D	3-dimensional

SUMMARY

Animal behavior is critical to survival and provides a window into how the brain makes decisions and integrates sensory information. However, behavior is also a result of complex interactions between genes, neural function, anatomy, and the physical environment. A simple model organism that allows researchers to more precisely interrogate the relationships between behavior and the brain is the nematode *C. elegans.* Despite its small nervous system, the worm demonstrates complex behaviors, and has been used extensively to link genes to function of the nervous system. However, current phenotyping tools have technical limitations that make observing, intervening in, and quantifying behavior in diverse settings difficult.

This thesis aims to develop enabling technological systems to resolve these challenges. To address scaling issues in observation and intervention in long-term behavior, I develop a platform for long-term continuous imaging, online behavior quantification, and online behavior-conditional intervention. I show that this tool is easy to build and use and can operate in an automated fashion for days at a time. I demonstrate that it can be used for behavioral phenotyping of individual animals from larval through adult stages. I then use this platform to understand the consequences of quiescence deprivation to *C. elegans* health. This tool can enable real-time processing and behavior data compression that will both enable novel behavior-conditional perturbation experiments at scale and ease the bottleneck of behavior data processing significantly. It may also be readily adapted for other model systems and many other types of automated behavior-based interventions.

To quantify complex animal postures, I develop an app to enable fast, versatile and quantitative annotation and demonstrate that it is both ~ 130-fold faster and, in some cases, less error-prone than state-of-the-art computational methods. This app is agnostic to image content and allows freehand annotation of curves and other complex and non-uniform shapes. It enables faster annotation both through ease of use and automated distribution of image annotation tasks to many users at once. In addition to annotation of *C. elegans* posture, we demonstrate applications in annotating plant and stem cell aggregate morphology. This tool may be used to generate ground truth sets for testing or creating automated algorithms.

Finally, I quantify *C. elegans* behavior using an automated quantitative analysis to identify behaviors and map the worm's behavioral repertoire across multiple physical environments that more closely mimic *C. elegans'* natural environment. From this analysis, I identified subtle behaviors that are not easily distinguishable by eye and built a tool that allows others to explore our video dataset and behaviors in a facile way. I also use this analysis to examine the richness of *C. elegans* behavior across selected environments and find that behavior diversity is not uniform across environments. This has important implications for choice of media for behavioral phenotyping, as it suggests that the appropriate media choice may increase our ability to distinguish behavioral phenotypes in *C. elegans*. This tool may be useful in phenotyping *C. elegans* behavior through aging and development or large-scale phenotyping of genetic mutants that may exhibit subtle behavioral phenotypes.

Together, these tools enable novel behavior experiments at a larger scale and with more nuanced phenotyping compared to currently available tools.

CHAPTER 1. INTRODUCTION

Behavior is what all animals do. It is a feature of living systems ranging from bacteria to humans that allows animals to avoid predation, find nutrients, and reproduce. Animal behavior provides a window into the brain, including decision making, sensory integration, and learning. Understanding these nervous system functions is an important goal of neuroscience. In human medical sciences, behavior is often used to diagnose and understand disease [1], and while many psychiatric disorders are understood to be influenced by genetics, how genetics and behavior are connected is poorly understood [2]. Behavior is made of a complex web of interactions between genes, neural function, anatomy, and the physical environment that presents many challenges. Chief among these challenges are both ethical and technical constraints on how we observe, intervene in, and quantify behavior.

As a result, model organisms such as mice, fruit flies, zebrafish, and worms have assumed a prominent role in our understanding of the relationships between behavior and the systems that cause it. In these model organisms, many ethical and technical barriers are drastically reduced. The small roundworm *C. elegans* is particularly amenable to understanding behavior. Its compact nervous system consists of just 302 neurons and the connections between all neurons have been fully mapped [3]. In addition, the ease with which the worm may be genetically manipulated has allowed us to better understand neural function and behavior and manipulate neural function from the inside out. Importantly, *C.*

elegans is also straightforward to culture isogenically in large numbers, providing statistical power that is difficult to achieve even in many other model organisms.

Despite the advantages of *C. elegans* as a model organism, technical constraints still exist that limit our ability to observe, intervene in, and quantify behavior even in these simple animals. First, their physical size can pose a challenge. At about 1 mm in length in adulthood, and barely 250 μm at hatching, microscopy methods are required to observe their behavior. While advances in efficiently monitoring large populations of animals have made it possible to better understand unperturbed animal behavior, intervening on animal behavior, particularly in a conditional way, is still highly inefficient [4–7]. Second, an important part of describing animal behavior is to be able to describe their instantaneous posture. While most *C. elegans* posture is straightforward to interpret and describe from 2-dimensional video data, more complex postures, particularly those where the worm is self-occluding, remain challenging to interpret from 2D data [8–10]. Lastly, the application of computer vision and machine learning techniques have drastically increased our ability to quantify and describe behavior [11,12]. However, classifying as well as quantifying behavior without applying anthropocentric heuristics remains a challenge, especially when comparing behavior in different physical environments. In the following section, I will treat each of these limitations in greater detail.

In this thesis, I develop and demonstrate a variety of tools designed to improve our ability to observe, intervene in, and quantify *C. elegans* behavior and applications of these technologies to answer specific biological questions. In the remainder of this chapter, I provide a brief review of existing behavior technologies and their limitations, followed by the objectives of this thesis. Each chapter of this thesis will provide detailed introduction specific to that chapter.

1.1 General challenges with conventional behavioral tools

Scientific advances that have already been made through behavior observation and analysis demonstrate that there is significant value in systematically characterizing behavior[13–15]. Traditional approaches to characterizing behavior have stemmed from both ethological and psychological roots and rely heavily on animal observation, behavior classification and description, and frequently intervening on the animals' environment either in nature or in the lab[16,17]. While these methods have prompted many fundamental theories about animal behavior and revealed how important and intricate behavior is even to the most outwardly simple animals, observing and describing animal behavior is an incredibly labor-intensive task. It is also easily biased by human involvement in description and discrimination of behaviors.

Since digital imaging has taken hold in consumer markets, observation can now frequently be performed by a camera rather than a person, with quantitative description and classification of the behavior post-hoc[18]. With these technologies, longer periods of continuous observation have become more tractable. However, most scientific cameras still easily cost several thousand dollars, and may only be able to observe one or a few animals at a time. This presents a particular problem for small model organisms. Equally problematic is the ability to quantify much more massive volumes of data produced by long-term behavior video[18]. Many researchers approach this conundrum by either limiting the temporal frequency of behavior observation or limiting the overall timescale of behavior observation. While these approaches may be appropriate for understanding behaviors either over very long timescales or very short ones, animal behavior itself is continuous.

Another key component of behavior studies is disrupting the animals' environment and studying their response. For example, an impactful early ethological study of gull behavior by Niko Tinbergen demonstrated that the gulls preferred to incubate fake, supernormally-sized eggs over real eggs [16]. Today, our methods of intervention have advanced significantly, enabling researchers to apply stimuli in highly automated, and even behavior-responsive ways. Virtual reality systems built to either mimic or distort an animal's perception of the world can provide insight into animal's conditional responses to visually perceived stimuli [19–22]. While systems like these enable new types of experiments that were previously impossible, technology for behavior-responsive intervention is expensive and difficult to scale.

Significant progress has been made in automating behavior analysis, but the development of automated methods requires validating them on hand-annotated data. Even hand-annotating a small amount of a video data set can require a prohibitive amount of time. Tools for hand-annotating images face trade-offs between speed, accuracy, and content of the annotation [23–26]. Applications requiring free-hand annotation of complex features are very difficult (i.e. slow), as most annotation tools are designed for fast annotation of single points or regions of interest.

Another challenge in ethology is understanding behavior in its natural contexts. However, observing animals in their natural habitat is often intractable. Striking the right balance between laboratory-derived settings and the natural context of the animals' behavior can yield greater understanding of how animals process information to make decisions. The range of environments in which animals behave can vary widely in their natural habitats, yet the tools we have for comparing behavior in different environments often presume very different classifications of behavior [27,28]. Comparing behavior between environments that mimic the naturalistic range that animals may find themselves in is therefore easily biased.

While these challenges are common to behavior paradigms across animal models, other common barriers such as the high level of individual-to-individual variability in behavior, difficulty in culturing sufficient animals to achieve statistical power, and paucity of genetic and neuroscience tools can be resolved through a judicious choice of model organism.

1.2 *C. elegans* as a model system for behavioral neuroscience

When Sydney Brenner popularized the use of a small (~1mm in length), transparent, nematode as a model organism in the late 1960s, some of his first studies of *Caenorhabditis elegans* (*C. elegans*) were forward genetic screens identifying mutant worms based on their distinctive behavioral and morphological phenotypes [29]. The genes causing these strong phenotypes were soon identified, demonstrating that *C. elegans* could be used to make connections between genes and behavior. Several features of the worm's life cycle also play an important role in making it a useful model organism. As behavior is highly multigenic, the ability to control for genetic changes is an important characteristic for behavior models to have. *C. elegans* is simple to culture isogenically at a large scale, as each individual hermaphrodite produces approximately 300 nearly isogenic progeny after only a short, three-day development period. This short developmental timescale also lends itself to nuanced understanding of development. Despite the small number of neurons in adult animals, they exhibit complex sensory capabilities and navigational behaviors, including chemotaxis, thigmotaxis and aversive olfactory learning [30–34]. In addition, evolutionary conservation of genes and pathways identified in *C. elegans* has been demonstrated in higher-order organisms and mammals, including humans. Pathways that regulate programmed cell death, aging, developmental timing, and stress in the worm all have closely related homologs in humans[35–38]. These key traits allow a level of tractability that other model systems are hard-pressed to approach balanced with biological relevance to higher animals.

A diverse collection of resources and tools have also played important roles in making *C. elegans* an excellent model system for behavioral neuroscience. The wiring of the *C. elegans* nervous system has been fully described and *C. elegans* was the first organism to have its genome fully sequenced [3,39]. Fluorescent proteins such as GFP and RFP can be used to visualize where genes are expressed [40]. More recently, genetic tools such as CRISPR and optogenetic tools and techniques have allowed *C. elegans* researchers to edit genes more efficiently and read and write to neurons in behaving animals [41,42]. The development of microfluidics to manipulate *C. elegans* has also proven incredibly useful in understanding how its nervous system functions [28,34,43–45]. These technological advances have all contributed to a powerful ability to link genes, the nervous system, and behavior together in *C. elegans*.

Despite this powerful ability, understanding *C. elegans'* behavior remains challenging, in great part due to the same challenges that are present in conventional behavorial paradigms.

1.2.1 Challenges for C. elegans paradigms

In typical lab culture conditions, *C. elegans'* behavior indeed looks very simple. It primarily moves in a sinusoidal manner on agarose plates, with occasional reorientations it accomplishes by exaggerating its sinusoidal wave pattern. Despite this seeming simplicity, as the field's ability to systematically quantify behavior has grown, so has the identification of genetic and environmental factors that influence increasingly subtle behaviors of the worm. For example, large-scale phenotyping studies of animal behavior on agarose have identified behavioral phenotypes for many mutants with no previously described phenotype, and comparing behavior between mutant strains has implicated genetic and protein pathway relationships [9,46]. The use of microfluidics to deliver spatially and temporally defined stimuli while tracking *C. elegans* behavior revealed genes involved in components of olfactory response [28]. These examples demonstrate that there is significant scientific advantage to be gained by the ability to systematically quantify behavior of *C. elegans*.

These important advances belie a remaining behavioral phenotyping gap. 85% of *C. elegans* genes have no reported phenotypic effect when knocked down with RNAi although most knockdowns detectably reduce fitness. Subtle behaviors have only recently begun to be studied. This is in part because of a lack of statistical power due to individuality and stochasticity in animal behavior.

Yet another challenge is the significant variation in *C. elegans'* behavior. The intrinsic underlying stochasticity in behavior combined with variations in extrinsically imposed environmental conditions creates a complex landscape of potential behavior response even in a very well-controlled system. It is therefore important to have a large population of animals that are as similar to one another as possible, both genetically and experientially.

1.2.1.1 Behavior collection and intervention

For small organisms that require a microscope to see, constant observation of behavior is at best expensive and at worst infeasible at the scale needed. Although it is easy to culture worms isogenically *en masse,* recording the behavior of many animals at scale is limited by the cost of microscopes and video collection equipment, which often amount to more than $10k per system for very simple setups. Thus, many behavior databases for *C. elegans* have been constructed from relatively short-term observation over the course of minutes despite the timescales of known *C. elegans* behaviors like quiescence [9] and the hours-long timescale of developmental processes.

Several more recent approaches have demonstrated an ability to monitor many worms in more cost-efficient ways, in some cases even across the entire lifespan [4,6,7,47]. Although these tools scale more efficiently, they are also lower content compared to conventional methods in terms of temporal resolution and image resolution.

While interventions in *C. elegans* behavior is straightforward in comparison with other organisms because of the ability to manipulate the worms' perception precisely using microfluidics and optogenetics, systems that are behavior responsive must be applied to one individual at a time[6,48]. Because of the high equipment costs for performing behavior responsive assays with current technologies, this class of assays are both expensive and low-throughput.

1.2.1.2 Behavior annotation

C. elegans behavior is often deceptively simple. The majority of the time, it is a great advantage that the worm's body plan is a pliable cylinder. Overall, this simplifies quantitative descriptions of worm posture. However, the extreme flexibility of *C. elegans* also leads to self-collision and often self-occlusion, where the animal's posture is much more difficult to infer. Several methods have been described to address this complication, including heuristic-based approaches and more general generative approaches [8,10]. These attempts to fill the gaps in worm behavior dynamics remain both slow and inaccurate, especially as their performance in non-standard situations is poor. In order to develop automated methods for quantifying more complex behaviors in complex environments, we first need a way to annotate the posture of the animal in many video frames. However, while some tools exist that enable community annotation of images, annotation tools that allow freehand annotations are slow, often custom-designed for specific purposes, and make it difficult to distribute annotations.

1.2.1.3 Behavior quantification

Conventional methods for quantifying *C. elegans* behavior have been based upon identifying and describing discrete behavior states in the worm, such as forward movement, reversals, and reorientation behaviors. With the rise of advanced computer vision techniques, a plethora of automated worm trackers have become available that segment animals, track them, and quantify behavior to varying degrees of robustness[27,49–51]. At the same time, new methods have evolved to describe continuous features of behavior beyond speed and acceleration, most notably *C. elegans* posture [10,12,52]. The most prevalent current method of describing *C. elegans* posture is to express the curvature of the animal in worm-centric coordinates. It has been shown that close to 95% of the animals posture can be described by the linear combination of just five eigenvectors calculated from the worm-centric curvature.

Describing posture in this low-dimensional way has been useful in not only quantifying posture but also in beginning to define behavior independently from an anthropomorphized point of view. Several machine learning based techniques have been demonstrated that attempt to define behavioral motifs in *C. elegans* without reliance on external definitions or heuristics [46,53]. One such approach that has been extensively demonstrated in fly behavior and more recently applied to worm behavior is t-distributed stochastic neighbour embedding (t-SNE) of postural frequency data [11,53–55]. Applying this technique to fruit flies has provided a human-independent quantitative method to classify and compare fly behavior and has enabled the behavioral dissection of fly motor control using optogenetics [55]. The application of these methods to classify and compare animal behavior in disparate physical environments has the potential to help us better phenotype animals and understand how they adapt to changing environments.

1.3 Thesis rationale

Despite the many technical advances that have enhanced our ability to collect, intervene in, and analyze behavior data, limitations in scalability presents a behavioral phenotyping problem. To overcome these limitations, in this thesis I develop platforms that integrate microscopy, computer vision, and machine learning techniques to more efficiently scale behavior data collection, intervention, and analysis of *C. elegans*. To summarize, there are three major challenges in behavioral genetics research in *C. elegans*:

1. Collecting large amounts of continuous behavioral data cheaply, automatically, and with feedback control
2. Annotating complex images to describe complex posture features
3. Interpretation of behavior data, particularly in variable environmental conditions where dynamics occur at different time scales.

1.4 Thesis outline

In this thesis, I address each of the technical challenges laid out above and develop improved tools for behavioral phenotyping of *C. elegans*. It consists of five chapters. Chapter two describes the development of a scalable method for long-term continuous imaging and online behavior-conditional intervention. I demonstrate an application of this platform to understanding the consequences of quiescence deprivation to *C. elegans* health.

Chapter three presents a scalable smartphone-based method for distributing annotation of complex image characteristics. We then use this tool to annotate complex postures of *C. elegans* that are difficult to quantify with existing error-prone and time-consuming methods. In chapter four, I adapt machine learning methods to quantify and compare worm behavior in a variety of environments designed to span the range of environments worms may encounter in their natural habitat. We use this to examine the subtlety and richness of *C. elegans* behaviors in different physical environments and as a consequence at different time scales. The final chapter provides conclusions and a discussion of future work stemming from the results presented.

CHAPTER 2. A SCALABLE MICROSCOPY SYSTEM FOR ONLINE BEHAVIOR MANIPULATION

2.1 Introduction

In this chapter, I demonstrate a scalable microscopy system that enables online behavior detection and conditional intervention in the environment of individual animals. I show that it can be used to collect long-term continuous behavior data for multiple larval stages and the adult stage of *C. elegans*, and that this data is of sufficient quality for postural phenotyping. I then demonstrate a variety of computer vision and machine learning techniques that I use on the system for online behavioral phenotyping and data compression. Finally, I use the system to create an extreme sleep-deprivation model in *C. elegans* and evaluate the health effects on the model.

2.2 Background

Systematic intervention into animal life is a mainstay across biomedical research. It allows us to understand how complex biological systems interact with external factors and is critical for understanding how diseases are caused and how to treat them. Just as medical doctors may diagnose disease based on behavior, evaluating animal behavior serves as a high-level proxy for health. Pairing behavior monitoring with systematic intervention is therefore an important tool across biomedical research and is heavily used in fields ranging from drug discovery and aging to neuroscience[56–61].

However, despite its prevalence, coupling behavior monitoring and external intervention is a labor-intensive and expensive process, even in model organisms. In many cases, we desire intervention only under certain conditions, such as dosing a drug only when it is needed. In these conditions, constant behavior monitoring may be required so that both short- and long-timescale behaviors can be effectively observed. This compounds with a need to compensate for the great variability in animal behavior by observing and intervening in the lives of many animals over long time periods. Under these circumstances, human monitoring quickly becomes intractable.

A rich variety of automated laboratory tools offer the ability to systematize exposure to varying environmental conditions [28,34,44,53,62–64]. However, systems that can respond to the behavior of individual animals in real time are costly to implement and require specialized tools and knowledge to build. Subsequent higher-depth phenotyping analysis of any behavior data collected is further time-limiting. These challenges force researchers to limit behavior feedback experiments to short timescales, small numbers of animals, or both.

This chapter covers a DIY framework (mi-pi, or microscopy-pi) using off-the-shelf components that allows users to continuously collect, analyze, and respond to behavior in real-time while scaling efficiently to high animal volumes. Our system is at least an order of magnitude less expensive than equivalent systems suited for high-content long-term behavior analysis. Despite its DIY nature, it is simple to assemble, install, and use, and is scalable.

I first demonstrate that our system enables high-depth behavioral phenotyping of the small roundworm *Caenorhabditis elegans*, including postural dynamics. We show that we can collect continuous behavior data over days-long timescales for much smaller developing animals in addition to adults. We then develop several methods for measuring animal motion online including neural nets for counting and locating animals. The first of these neural networks is well-suited for applications where high accuracy is needed, while the second enables animal detection at rates faster than video frame collection. These features enable us to track and disrupt developmental quiescence and track health affects in *C. elegans*.

While here we use *C. elegans* to demonstrate our system, we expect that it could easily be adapted for use with other model organisms, such as fruit flies or their larvae, and augmented with hardware for controlling exposure to alternative external sensory cues or drugs.

2.3 Materials and Methods

2.3.1 C. elegans maintenance

C. elegans strains were maintained under standard conditions at 20°C unless otherwise noted [65]. Strains used in this work include N2, CB101[*unc-9*(*e101*)], DA1814 [*ser-1*(*ok345*)], and GT323 [*lite-1*(*ok530*)].

2.3.2 Plate assays

To prevent animals from leaving the microscope field of view (FoV), we prepared special plates. Palmitic acid has been demonstrated as an effective barrier for worms in behavior experiments[66]. It is typically applied as a solution in ethanol to a standard plate and the ethanol is allowed to evaporate off. However, it is hard to deposit in a controlled way due to the palmitic acid solution wetting the agar. We used an ethanol-sterilized piece of PDMS as a negative to prevent a 10 mg/mL palmitic acid in ethanol solution from wetting the center of a 5cm NGM plate, allowing the ethanol to evaporate for at least 30 minutes before removing the PDMS with tweezers. These plates were subsequently seeded with 10 ul of OP50 and incubated at room temperature for about 24 hours to allow a thin lawn to formed. Plates were stored at 4°C until an hour before use. For short-term assays on adult animals, animals were picked onto plates about an hour before experiments started after the plates had warmed to room temperature. For developmental assays, adult animals were bleached to obtain eggs. Eggs were allowed to hatch and larvae allowed to reach L1 arrest by agitating eggs overnight in M9 buffer. L1s were then pipetted onto an unseeded NGM plate and single animals were pipetted onto the prepared seeded palmitic acid plates. These plates were then parafilmed and incubated at 20°C until animals reached L3 stage (20 hours after plating), when each plate was placed on a mi-pi system. Developmental experiments lasted 44 hours, at which point worms have typically reached sexual maturity and plates were removed from mi-pi systems. At the 48 hour time point, the number of larvae and eggs on each plate were counted. Data from any plates where the original animal could not be found, or where any contamination or potential starvation was identified were censored.

2.3.3 Microfluidic Experiment

We cultured animals under standard conditions until they reached day 1 adulthood. We then washed animals off of plates with M9 buffer and suspended animals in ~1mL of M9 buffer with 0.1% Triton-X, a surfactant used to prevent worms from sticking to one another. We then loaded animals into microfluidic devices as previously described[43]. Animal behavior was collected with our microscope over the course of an hour.

2.3.4 Microscope hardware

The microscope is built from off-the-shelf parts and requires minimal specialized tools or skills to build. Almost all of the microscope can be built with hands and screwdrivers, with the exception of the LED driver used to drive the bright blue LED string, which requires soldering 6 joints. The housing is made from building blocks, which are easily reconfigurable (**Figure 2-1**). Once each microscope is built, it can be operated without a keyboard or mouse. The full list of materials and costs at time of writing can be found in **Appendix A.1 Bill of Materials** and at https://github.com/lu-lab/mi-pi. The total cost for a single system is ~$400. Extensive step-by-step documentation on building mi-pi can be found at https://github.com/lu-lab/mi-pi and in **Appendix A.2 Hardware setup**.

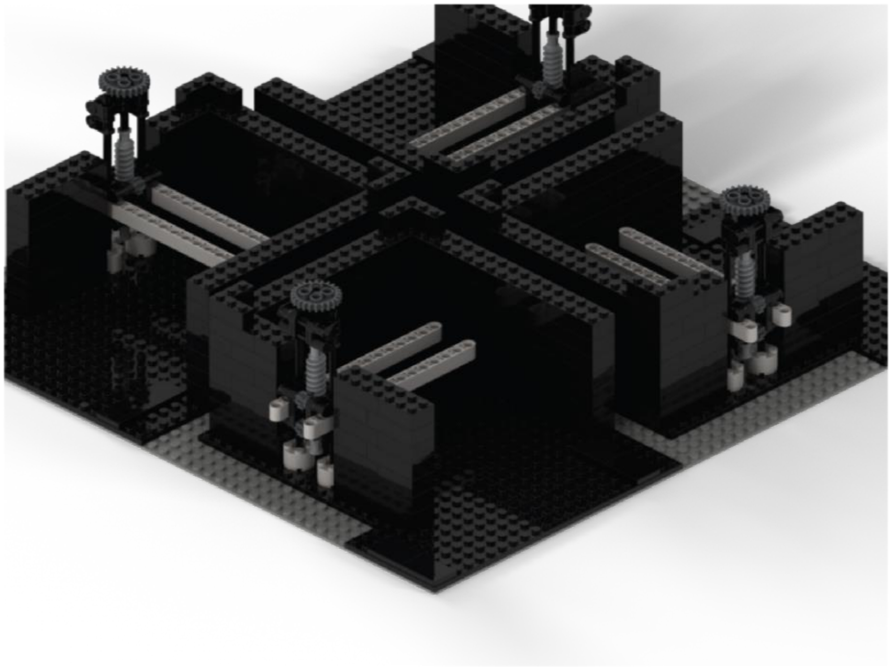


Figure 2-1 Building block microscope housing. Four mi-pi enclosures rendered with Stud.io 2.0. All parts are standard Lego pieces.

2.3.5 Microscope control software

Installing the appropriate software packages is often a barrier to using a ‘DIY’ system, so we have provided an image of an operating system with all packages installed, including the mi-pi software. On the system desktop, there is a shortcut icon that is used to start mi-pi. On initial start-up, the user is guided through set-up of rclone (https://rclone.org/), a cloud storage sync program, and obtaining an authentication key for Google Spreadsheets associated with a Google account, which we use to control experiments (**Figure 2-2**). When the mi-pi user interface opens, users can adjust settings that control most aspects of the mi-pi system, including the type of image processing, resolution of video, whether to stream video to YouTube, as well as annotate experimental metadata such as animal strain or age. The full list of settings with descriptions can be found in **Appendix A.3 Software setup**, along with software setup instructions.

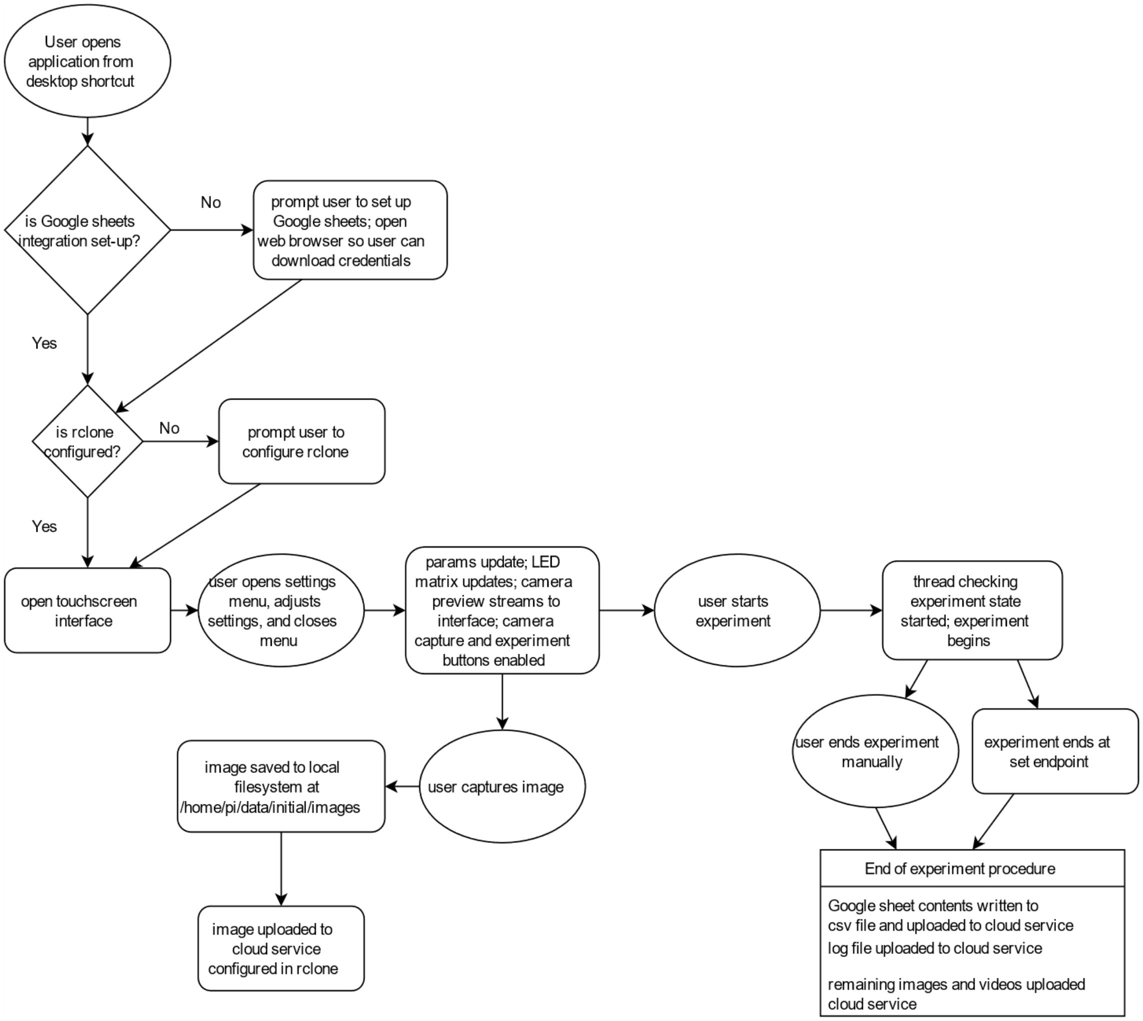

User opens application from desktop shortcut
is Google sheets integration set-up?
No
prompt user to set up Google sheets; open web browser so user can download credentials
Yes
is rclone configured?
No
prompt user to configure rclone
Yes
open touchscreen interface
user opens settings menu, adjusts settings, and closes menu
params update; LED matrix updates; camera preview streams to interface; camera capture and experiment buttons enabled
user starts experiment
thread checking experiment state started; experiment begins
user captures image
image saved to local filesystem at /home/pi/data/initial/images
image uploaded to cloud service configured in rclone
user ends experiment manually
experiment ends at set endpoint
End of experiment procedure
Google sheet contents written to csv file and uploaded to cloud service
log file uploaded to cloud service
remaining images and videos uploaded cloud service

Figure 2-2. Mi-pi start-up user decision tree

There are essentially three functions of the software: (1) to set up experiment parameters and metadata and ensure animals are in focus through a touch-based user interface (2) collect video data and process it and (3) read and write experiment parameters and data to Google Sheets and use these to update hardware illumination parameters. The software is written in Python, using Kivy, an open-source, cross platform Python library for developing touch applications, which allows us to avoid requiring mouse and keyboard for the operation of each microscope. To process images, we use a combination of OpenCV, Tensorflow, and PIL (Python Image Library), and we interface directly with the Google Sheets API. A high-level software architecture diagram can be found in **Figure 2-3**.

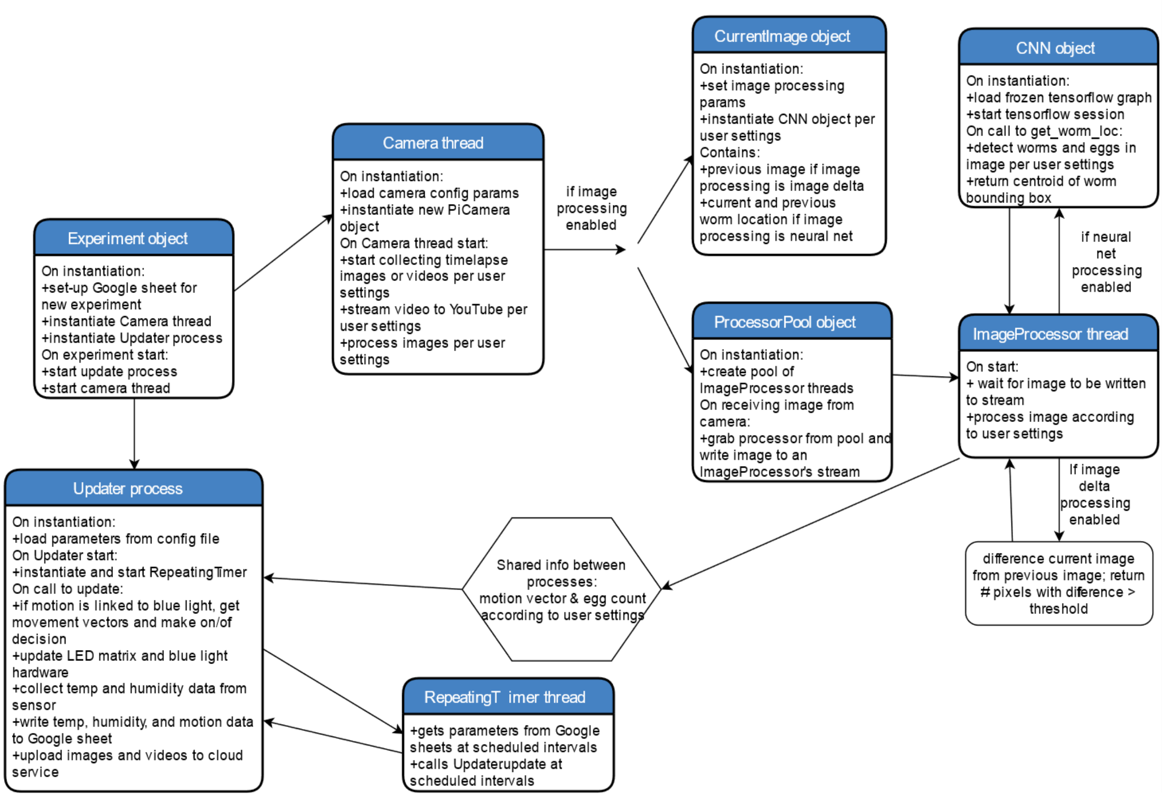

Experiment object
On instantiation:
+set-up Google sheet for new experiment
+instantiate Camera thread
+instantiate Updater process
On experiment start:
+start update process
+start camera thread
Camera thread
On instantiation:
+load camera config params
+instantiate new PiCamera object
On Camera thread start:
+start collecting timelapse images or videos per user settings
+stream video to YouTube per user settings
+process images per user settings
if image processing enabled
CurrentImage object
On instantiation:
+set image processing params
+instantiate CNN object per user settings
Contains:
+previous image if image processing is image delta
+current and previous worm location if image processing is neural net
CNN object
On instantiation:
+load frozen tensorflow graph
+start tensorflow session
On call to get_worm_loc:
+detect worms and eggs in image per user settings
+return centroid of worm bounding box
if neural net processing enabled
ProcessorPool object
On instantiation:
+create pool of ImageProcessor threads
On receiving image from camera:
+grab processor from pool and write image to an ImageProcessor's stream
ImageProcessor thread
On start:
+ wait for image to be written to stream
+process image according to user settings
If image delta processing enabled
difference current image from previous image; return # pixels with difference &gt; threshold
Shared info between processes:
motion vector &amp; egg count according to user settings
Updater process
On instantiation:
+load parameters from config file
On Updater start:
+instantiate and start RepeatingTimer
On call to update:
+if motion is linked to blue light, get movement vectors and make on/of decision
+update LED matrix and blue light hardware
+collect temp and humidity data from sensor
+write temp, humidity, and motion data to Google sheet
+upload images and videos to cloud service
RepeatingT imer thread
+gets parameters from Google sheets at scheduled intervals
+calls Updaterupdate at scheduled intervals

Figure 2-3 High-level software architecture of mi-pi.

We use Google Sheets to allow experimenters to easily update experimental parameters remotely. These parameters include the updating interval of the system, as well as LED matrix imaging modes (for example darkfield and brightfield), LED matrix color, and the state of the bright blue LEDs used for animal stimulation (**Figure 2-4**). At each update interval, the microscope queries the Google Sheet and relays these parameters to the microcontroller. At the update interval, we also read humidity and temperature from the sensor connected to the microcontroller. We chose to use Google Sheets to communicate between remote users and the Raspberry Pis because the Google Sheet interface is easy-to-use, is easily scaled up for many microscope systems, and freely usable. We use the command-line program rclone to upload data to any of a variety of cloud services, deleting data from the Raspberry Pis once we verify it has been uploaded to the remote to reduce the requisite size of the microSD card the Raspberry Pi uses as operating system storage. All of these features are easy to configure and parameters are adjustable in the microscope interface.

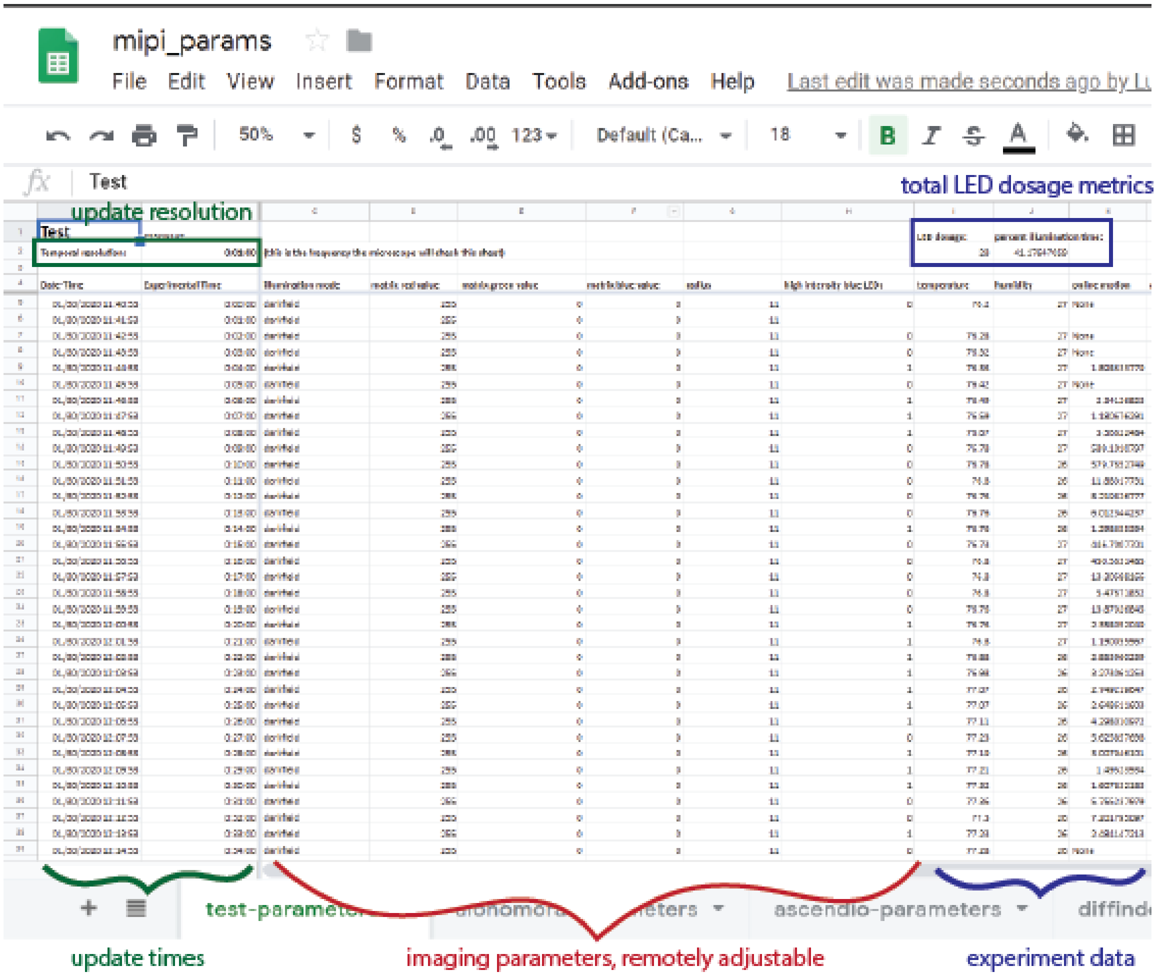


Figure 2-4 Google Sheet structure

2.3.6 Microscope characterization

We determined the resolution of the microscope using a 1951 USAF Target (ThorLabs). The resolving power is estimated by using mi-pi to image the target, which consists of series of lines at decreasing spacings. The minimum line spacing for which discrete lines were visible was 22.1 µm. The microscope's field of view is approximately 1.5cm x 2cm and the magnification is 0.56x.

2.3.7 Post-hoc behavior tracking for system validation

We used Tierpsy-Tracker (https://github.com/ver228/tierpsy-tracker) for post-hoc quantification of speed and posture for mutant and wild-type (N2) animals. To simplify processing for long-term experiments, we first concatenated movies into 10 minutes movies from the original 20s movie length. After processing data with Tierpsy-Tracker, we manually linked tracks from individual animals together over the length of the movie and exported velocity (in pixels/s) and posture (represented with 'eigenworms'[12]) for individual WT and mutant animals. From velocity vectors we took only positive velocity values (forward speed) to compare across strains.

2.3.8 Online motion analysis and feedback

We used mi-pi to process image frames as they were being collected, while simultaneously capturing continuous video to disk. We used several approaches to estimate motion between frames. The simplest approach uses OpenCV, an open-source image processing software package, to convert the incoming image to grayscale and subtract it from the previous image. On this difference image, we perform a morphological opening, and then count the total number of pixels above the intensity threshold of image noise, which was identified on an individual experiment basis by the experimenter. This gave a reasonable estimation of the amount of motion between two frames under uniform lighting conditions for adults but performed poorly for much smaller larval animals. We addressed this by developing two convolutional neural network models using the Tensorflow Python package, one a slower but more accurate model using a Faster R-CNN object detection architecture, and one using a much faster Mobilenet v2 object detection architecture (characterized below). We annotated images randomly subsampled from development experiments on 8 of our systems with labelImg (https://github.com/tzutalin/labelImg). Our Faster R-CNN model was trained on a set of 1,008 images that was validated with an independent annotated test set of 114 images, while our Mobilenet model was trained on a set of 5,109 images and validated with an independent test set of 517 images.

To estimate motion using these object detectors, we computed the centroid of the bounding box of the detected worm with the highest confidence score every minute and calculated the Euclidean norm between consecutive centroids. When no worms are detected above a confidence threshold of 80%, no centroid is calculated and the next frame where a worm is detected is used to estimate motion from the last frame with a detection.

Once we had quantified motion between two frames, we used the amount of motion to determine whether to stimulate the animals with bright blue light. For quiescence experiments we used the Faster R-CNN object detector to estimate motion and considered the animal in quiescence if its centroid moved less than 5 pixels (about 50 μm). For motion-coupled systems, at every minute interval, if the animal’s motion did not exceed this threshold, it had a chance of being dosed with blue light for 10 seconds out of that interval. This chance is adjustable in mi-pi’s settings. In quiescence experiments described here, this chance ranged from 30% to 100%. As a control, animals from the same population were subjected to the same light dosage uncoupled from their motion. For these systems, the animal’s motion is quantified online, however, it is not used to make stimulus decisions.

Instead, the total stimulus time from a paired motion-coupled system is used to update the dosage estimate every 6 hours.

2.4 Results

2.4.1 Scalable DIY systems for behavior monitoring

To enable scalable long-term, high-content behavior phenotyping, we developed a low-cost, DIY system from off-the-shelf components (**Figure 2-5**). The system is easy to construct (~ 2 hr total build time) and use (guided set-up in about 5 minutes). The main components include a Raspberry Pi and Raspberry Pi Camera for video acquisition, processing, and transfer, an LED matrix to enable high-contrast imaging, a high-intensity LED string for external intervention, and a housing made of building blocks (**Figure 2-5a & c**). The LED matrix, high-intensity LED string, and a humidity and temperature sensor are interfaced with the Pi via a serial connection with a microcontroller. We provide a detailed guide for constructing each system as well as operating system images for simple installation on the Raspberry Pi on our lab GitHub (https://github.com/lu-lab/mi-pi). Its building block construction enables a highly modular design that can be assembled or modified in minutes while maintaining structural uniformity among systems that exceeds that of most hobby-level 3d printers (about 10 micron building block tolerance vs. 200 micron printer resolution) (**Figure 2-5** and **Appendix A.2 Hardware setup**). The independent nature of individual systems enables uninterrupted video monitoring of multiple animal populations (strains, biological replicates, or controls) or individual animals, concurrently. Each system costs about $400 to assemble and represents a greater than 10-fold reduction in cost over commercially available systems and traditional microscope systems (**A.1 Bill of Materials**). From these readily available and non-specialist components, individual systems can run independently and collect high-content longitudinal behavior of animal populations continuously over days.

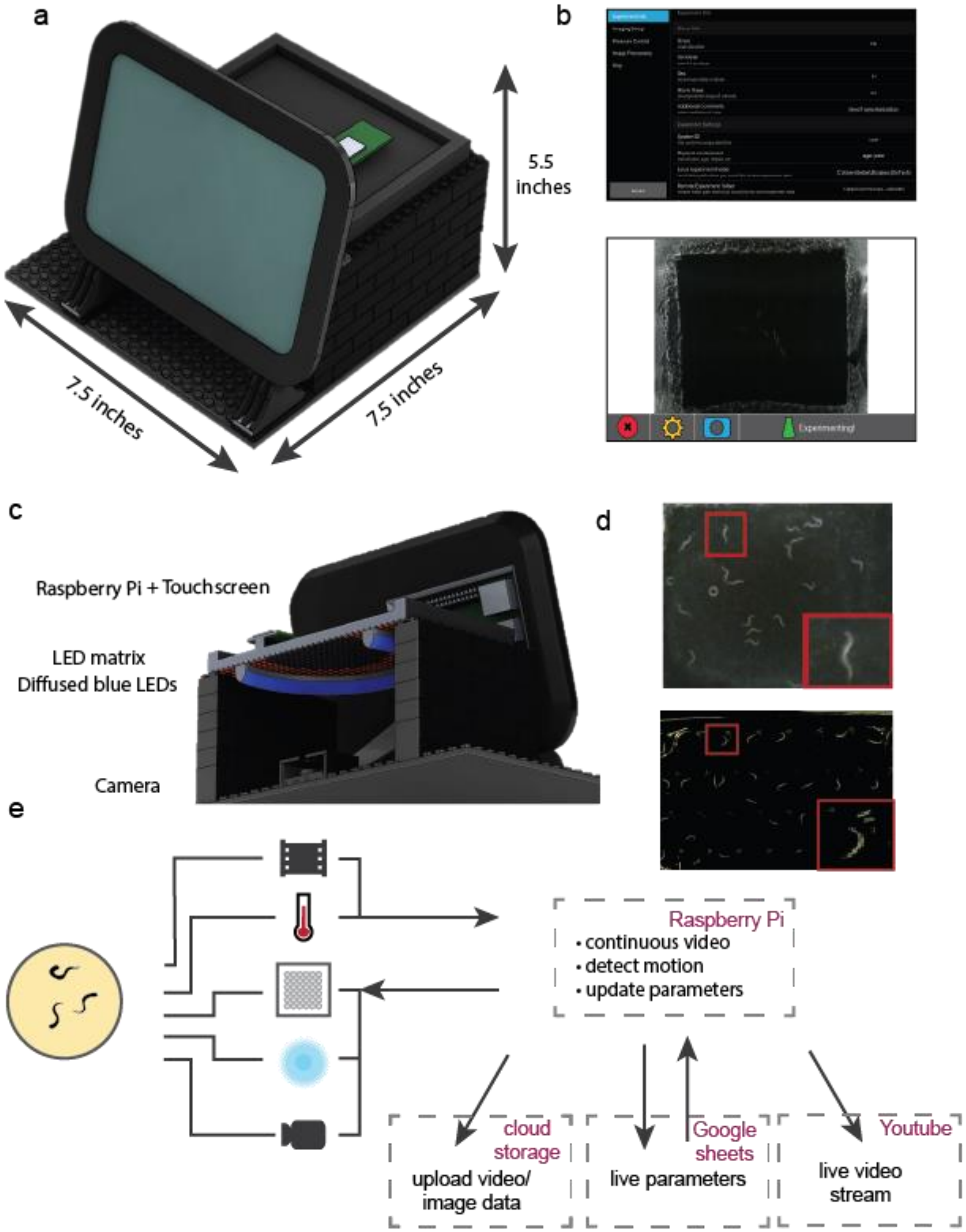


Figure 2-5 A scalable DIY system for online behavior-state feedback. a) External view of system with dimensions. b) Touchscreen interface screenshots. Upper screenshot is from settings page, and lower screenshot is from main page after an experiment is started. c) Cross-section of a mi-pi system. d) Example images from a plate and a microfluidic

experiment on the system, with inset of highlighted animals. e) System information flow during experiments. Raspberry Pi controls LED matrix, bright blue LEDs, and video capture, and collects and saves video, temperature, and humidity readings. Simultaneously, the Raspberry Pi computes motion from image data, uploads video and image data to remote cloud services, updates Google Sheets with motion, temperature, and humidity levels, and can stream live video to YouTube.

Our microscope system software is easy-to-use and fully automates operation during experiments (**Figure 2-2**). Experimental set-up is guided by a touchscreen app that captures experimental metadata and initializes experimental conditions (**Figure 2-2** and **Figure 2-5**). Once experiments are started, the experimenter can allow the system to run completely automated or control experimental parameters such as light exposure remotely via a Google Sheet during the experiment (**Figure 2-4**). As experiments run, data is recorded to each system's Google Sheet, including temperature, humidity, and if desired, animal motion. Additionally, experimenters can optionally stream live video from their experiments to a YouTube live channel. Videos and images collected during experiments can be uploaded to most cloud storage platforms as the experiment is on-going, enabling remote high-depth phenotyping as videos are uploaded, in addition to online analysis capabilities discussed in more detail later. The mi-pi software is written in Python while the microcontroller (Teensy 3.6) runs an Arduino script that can communicate bi-directionally with mi-pi. While here we demonstrate an optical intervention, other hardware could be integrated through unused microcontroller pins to intervene with other sensory modalities, for example chemo- and mechano-sensation, both of which can be minutely controlled through the use of microfluidic systems.

Each system's field of view is approximately 1.5cm x 2cm, with a resolution of 22 μm and magnification of about 0.5x, making it suitable for many small model organisms. The system field of view and magnification can also be easily modified to observe larger organisms by adjusting the lens with a simple hand tool. Observing smaller organisms could be accommodated by switching to a higher magnification lens. At the time of writing, we have built 12 independent systems that have been actively used for a period over 6 months to observe and respond to *C. elegans* behavior, collecting over 9 TB of video and image data.

2.4.2 High-content behavioral phenotyping at population and individual levels

We sought to show that our system has sufficient image quality and resolution to provide both individual and population-level behavioral data at a high phenotyping depth (e.g. ranging from discrete categories of movement to dynamic postural information). To do this, we selected several mutants with known behavioral phenotypes that would require posture and location tracking at the level of individuals. The two mutants we selected were *ser-1,* which encodes a serotonin receptor that is also an ortholog of human HTR2B, and *unc-9,* which encodes a protein involved in ion transmembrane transport[67,68]. The mutant *ser-1*(ok345) is known to have an increased amplitude of sinusoidal movement, increased body posture wavelength, and increased speed. Conversely, the mutant *unc-9*(e101) is known to have a decreased amplitude of sinusoidal movement and no recorded speed phenotype. We tracked populations of adult wild-type (N2) animals, as well as populations of several mutant strains with known behavioral traits for 20 minutes on standard agar plates to replicate these known behavioral phenotypes (**Figure 2-6**).

From these recordings, we extracted speed and posture of individual animals from each strain (n=12 animals, 117,400 total tracked frames for N2, n=14 animals, 79,370 total tracked frames for *ser-1*, n=12 animals, 135,480 total tracked frames for *unc-9*) and found that previously identified subtle behavioral phenotypes of tracked mutants were replicated. In our tracked dataset, we found that *ser-1's* increased speed phenotype compared to WT animals was replicated (Wilcoxon rank-sum test with Bonferroni correction, p=2.9x10^{-22}), as well as increased amplitude of sinusoidal movement (Wilcoxon rank-sum test with Bonferroni correction, p = 1.6x10$^{-28)}$ (**Figure 2-6 a-b**). Projections of *ser-1* posture into the 'eigenworm' space are also consistent with this finding (**Figure 2-6** and **Figure 2-7** Posture projections into eigenworm space.). We also examined *unc-9* mutants. We found that as expected, *unc-9* animals had a significant amplitude difference from N2 (Wilcoxon rank-sum test with Bonferroni correction, p=0.011) (**Figure 2-6**). These data show that our low-cost system can capture high-content phenotyping data and replicate known behavioral phenotypes.

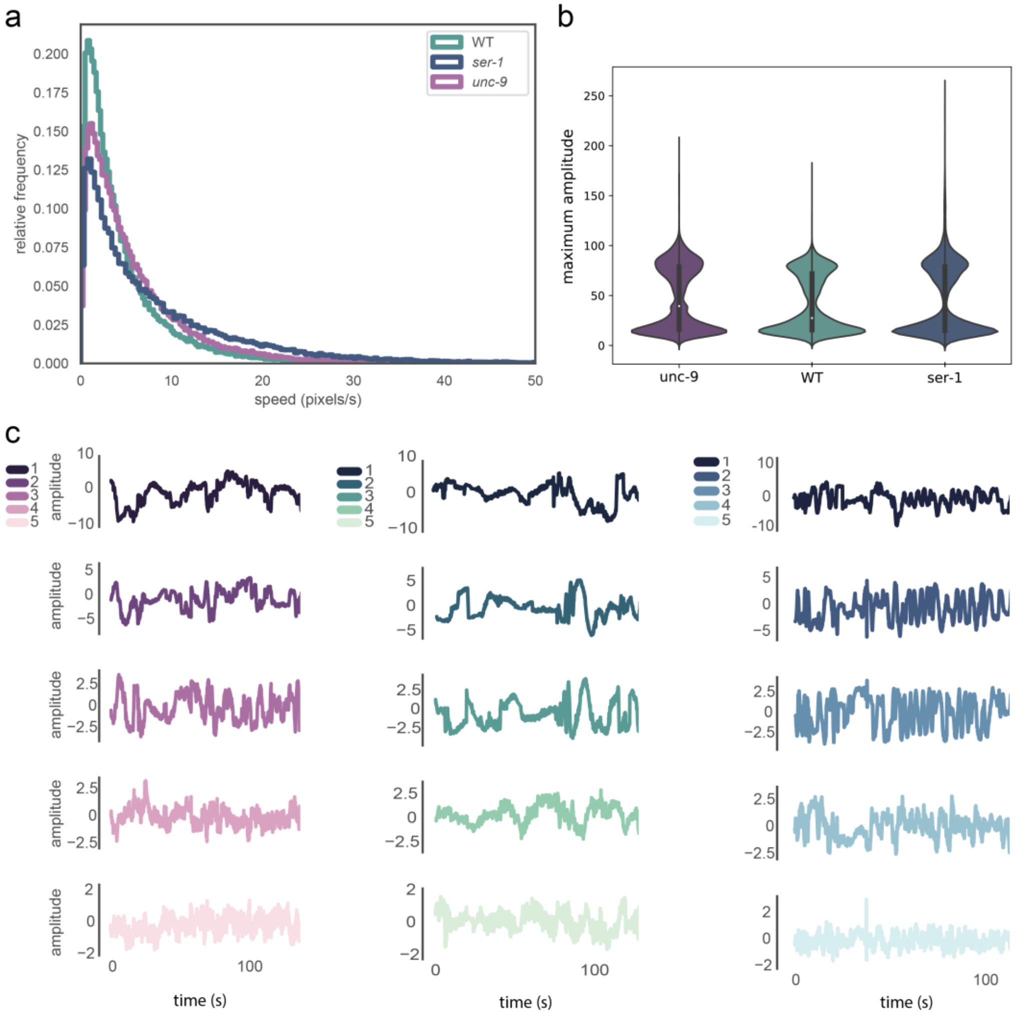


Figure 2-6 Population and individual phenotyping. a) Speed histograms of wild-type N2 (WT), *unc-9* mutants, and *ser-1* mutants animals. b) Maximum bend amplitude (measured in pixels) distribution for N2 (WT), *unc-9* and *ser-1* mutants. c) Example eigenworm amplitude traces for N2 (WT), unc-9 and ser-1 mutants. The combination of the amplitude of the first five eigenworms has been demonstrated to capture about 95% of the variance in *C. elegans* posture [12].

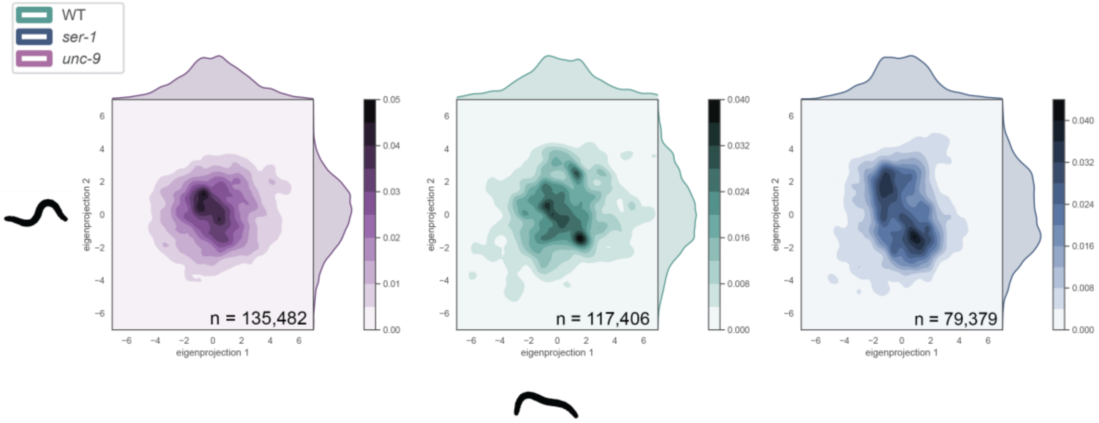


Figure 2-7 Posture projections into eigenworm space. Color indicates density of projections into the space of the first and second projection. Worm shapes show the 'eigenworm' associated with each axis (the same eigenworms are used for all data projections).

To track individuals over extended timescales, we used agar plates with palmitic acid barriers to retain individuals in the field of view. When animals encounter the resulting crystalline structures, they reverse back into the arena. Agar plates were then seeded with OP50 bacteria to prevent animals from starving. We imaged a population of animals starting at the second larval stage (L2) and recorded their behavior over the course of 30 hours, through to adulthood, tracking their behavior *post-hoc* through developmental stages (**Figure 2-8**). As expected, locomotion of animals fluctuates over time, with larva frequently moving at faster rates relative to their body length. As animals reached adulthood, we found population velocity relative to length dropped drastically as worms spent most of their time feeding.

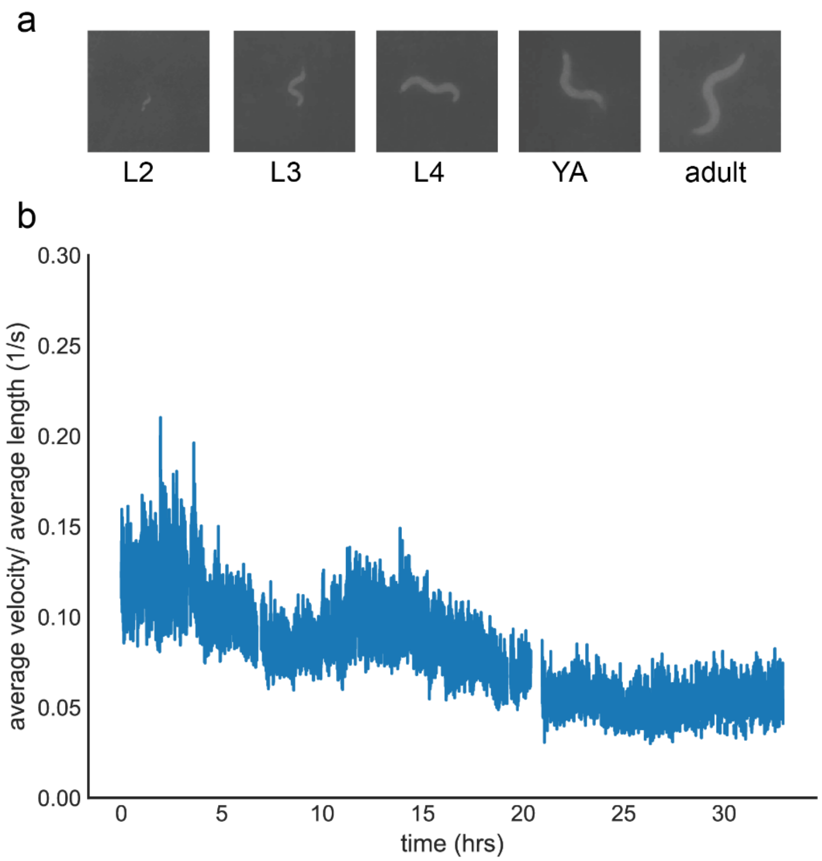


Figure 2-8 Longitudinal larval imaging a) Example images of animals in second larval stage to adult. b) Post-hoc quantification of population motion levels throughout development, measured as average velocity of animals normalized to average length of animals through development. Breaks in velocity are a result of interruptions in internet connection (using an early version of mi-pi). N = 30

These experiments demonstrate that (1) both adult and larval behavior can be captured on these systems, (2) image quality is high enough to capture and quantify posture and (3) the systems can capture behavior continuously over extended timescales.

2.4.3 Longitudinal on-line behavior tracking through animal development

To intervene in each animal's life depending on that individual's behavior, we first require a way to track behavior over long time periods in a timely way so that the intervention is concordant with the behavior's onset. As a first target, we chose to identify quiescence bouts within the lethargus developmental state in *C. elegans* development. These periodic bouts are akin to human sleep and under extreme circumstances deprivation can be lethal to worms [69]. Each lethargus state typically lasts 2-3 hrs and these are spaced throughout the animal's days-long development [70,71]. During lethargus, worms are less active, and when animals are deprived of quiescence by external stimulation, they face an increasing pressure to enter a quiescent state [70]. Quiescence behavior spans development, while switching between quiescent and active states occurs on the timescale of seconds to minutes, demanding a need for timely behavior tracking over a period of days in order to consistently perturb each individual animal's quiescence state [71].

In order to intervene on quiescence behavior, we first developed methods for detecting quiescence at relevant timescales. A simple method that is often used to detect movement, image subtraction, can be used to quickly estimate whether animals are quiescent. In short, greyscale images are subtracted from the adjacent frame, and a threshold is used to count how many pixels change more than a threshold amount. A second threshold for how many changed pixels constitutes animal movement can be imposed to estimate the degree of animal movement (**Figure 2-9 a**). This method has been used previously to monitor *C. elegans* healthspan and quiescence [7,70–72]. We used the image processing package OpenCV to process pairs of images collected at time intervals that can be adjusted in mi-pi's interface. Using this motion detection scheme, we processed images at a rate of 1.16 ± 0.32 seconds (mean and standard deviation from 1267 instances) (**Figure 2-9**).

2.4.4 High-accuracy near-online animal motion measurement

While the image subtraction method had more than sufficient temporal resolution to meet quiescence detection needs, it was very sensitive to illumination conditions and animal contrast from background (particularly bacterial lawn artifacts), making it less robust at measuring movement of animals at younger larval stages. When imaging more than one animal, this method requires normalization to animal size or alternatively physical isolation of each animal. Lastly, if animals are highly confined, as is common when tracking many individuals simultaneously over long timescales, higher temporal resolution may be required to detect animal movement, as the animal is more likely to return to its original position after moving.

To address these shortcomings, we trained a convolutional neural network to identify worms and track their location. We chose a Faster R-CNN, a model architecture shown to have high accuracy on the COCO image dataset [73–75]. We used transfer learning to re-train the model's final layer with images collected on our fleet of systems of animals throughout development (1,008 annotated images randomly selected). The average precision of this detector on our test dataset of 114 representative images was 0.919, and the recall (a measure of the false negative rate) was nearly 1 (indicating no false negatives) for precision values of 0.94. (**Figure 2-9 e** and **Figure 2-10**). When integrated into the mi-pi system, the average speed of inferencing was 82.23 ± 35.05 seconds (mean and standard deviation for 22, 280 observations on 8 mi-pis) built with Raspberry Pi 3Bs (**Figure 2-9**). On a Raspberry Pi 4 (4 GB RAM) system, the average speed of detection was 14.58 ± 1.55 seconds (mean and standard deviation for 710 observations on 1 mi-pi) (**Figure 2-9 d**) .

While Faster R-CNN object detection was significantly slower than the image differencing method, it was much more robust at detecting movement of younger worms, and still met speed requirements based on the timescale of quiescence bouts during *C. elegans* lethargus, which occur on the order of minutes in *C. elegans* [71]. We also found it to be remarkably resilient to changing and non-uniform lighting, as shown in **Figure 2-9 c**. This method also provides other information useful in understanding the social contexts of behavior, including the number of worms detected in the image and the physical location of those animals, while requiring no parameter setting on the part of the user.

2.4.5 Realtime animal motion measurement

While the Faster R-CNN model was sufficient for detecting quiescence bouts and other longer period behavior, many behaviors occur on a much shorter timescale. In addition, our continuous recording scheme generated raw video data at a very high rate – with 10 systems operating, this easily generated 7 GB of video an hour (>300 GB for each set of 10 experiments over 44 hrs). Any higher-depth *post-hoc* behavior quantification would often take the length of the experiment again to complete. This inefficiency is wide-spread throughout behavior imaging and leads to 'unmined' behavior data, where despite high-quality long-term imaging, only the simplest metrics are applied to describe the animal's behavior[18]. If the worm's location could be determined faster than the video collection rate, however, we could compress data significantly by saving video data sparsely while at the same time enabling more efficient post-hoc analysis, and have more biological applications by virtue of a shorter detection latency.

To see if truly online detection was possible on a low-cost system like mi-pi, we used transfer learning to re-train a Mobilenet v2 object detection model, a model commonly used on smartphone processors for real-time analysis[76]. Our training data once again consisted of images collected on mi-pis collected throughout animal development, this time with a training set of 5,109 total images. On our test set of 518 images, this model performed with an average precision of 0.428 (**Figure 2-9**). While the average precision of this model is lower than our Faster R-CNN model, we qualitatively saw that this lower precision resulted primarily from poor detection of larval animals, likely due to the lower resolution input images required for the Mobilenet model. Similarly, we also saw substantially lower recall, indicating a much higher false negative rate that is likely also due to smaller worms and lower resolution images (**Figure 2-10**).We first deployed this on a Raspberry Pi 4 (4 GB RAM) mi-pi system and found that detections took 1.16 ± 0.18 seconds (467 observations on 1 mi-pi) (**Figure 2-9 e**). Although not yet at the video framerate, the speed of this method was comparable with the image differencing method we first applied to detect quiescence, and thus sufficient for many applications while also providing higher quality information. We then deployed the model on the same system with a USB inferencing accelerator (Coral Edge TPU). On this system, we found we were able to inference at a speed of 48 frames per second (0.0206 ± 0.0076 seconds per frame, from 12,053 observations on 1 mi-pi), almost twice the maximum framerate of the camera

(**Figure 2-9**). This is more than sufficient for online speed measurement, and could be used to detect instantaneous reversals that start at a sub-second time scale.

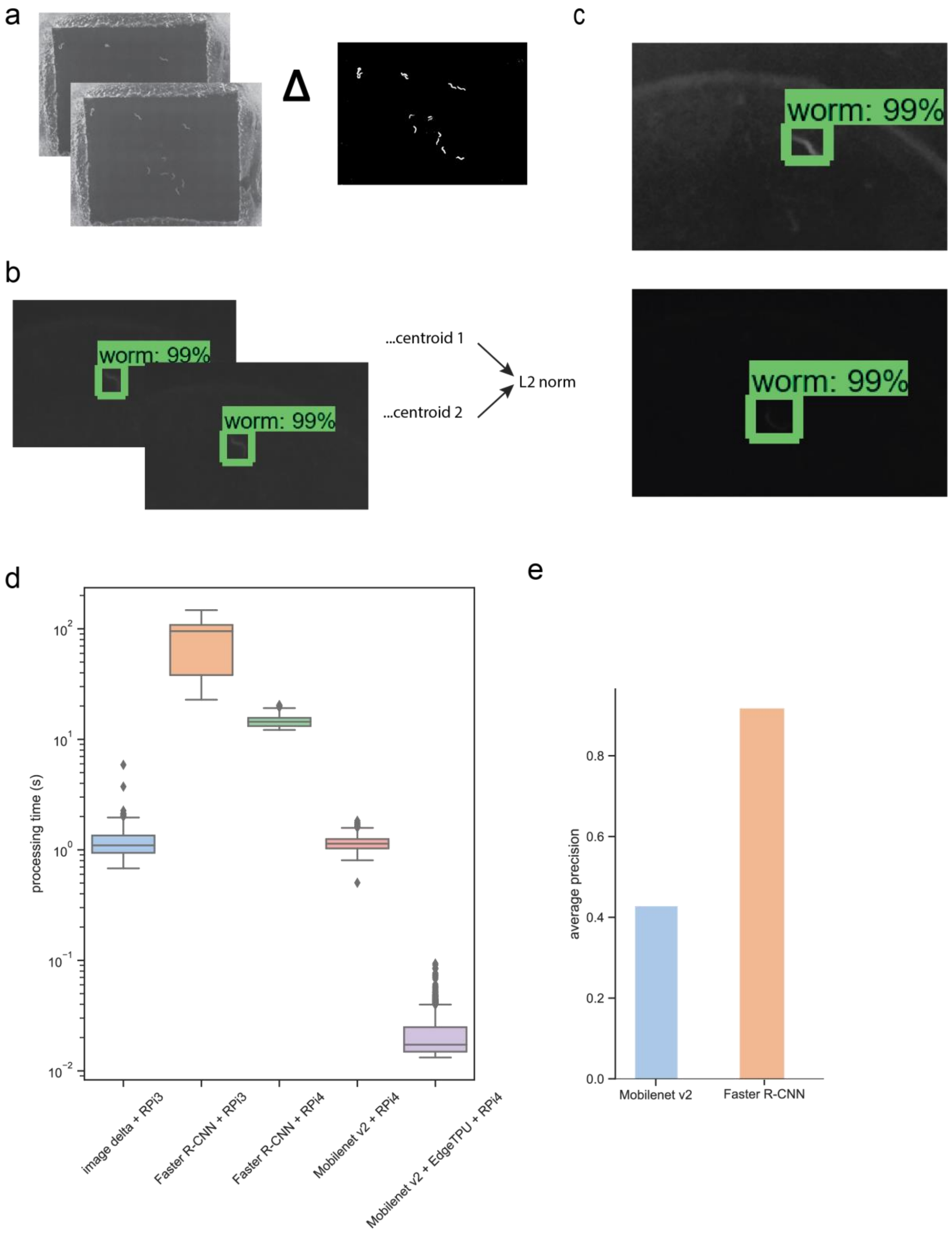

a
Δ
c
worm: 99%
worm: 99%
b
worm: 99%
worm: 99%
...centroid 1
...centroid 2
L2 norm
d
processing time (s)
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
image delta + RPi3
Faster R-CNN + RPi3
Faster R-CNN + RPi4
Mobilenet v2 + RPi4
Mobilenet v2 + EdgeTPU + RPi4
e
average precision
0.0
0.2
0.4
0.6
0.8
Mobilenet v2
Faster R-CNN

Figure 2-9 Neural net and computer vision image processing schemes for online behavior measurement. a) image differencing motion scheme and example with L4 worms. b) Convolutional neural net motion scheme and example with L3 worms. c) Faster-RCNN success in varied illumination conditions. d) Image processing speed for different behavior measurement schemes and system configurations. Box indicates quartiles, with whiskers extending to 1.5 times the inter-quartile range. Gray diamonds indicate points outside of this range.

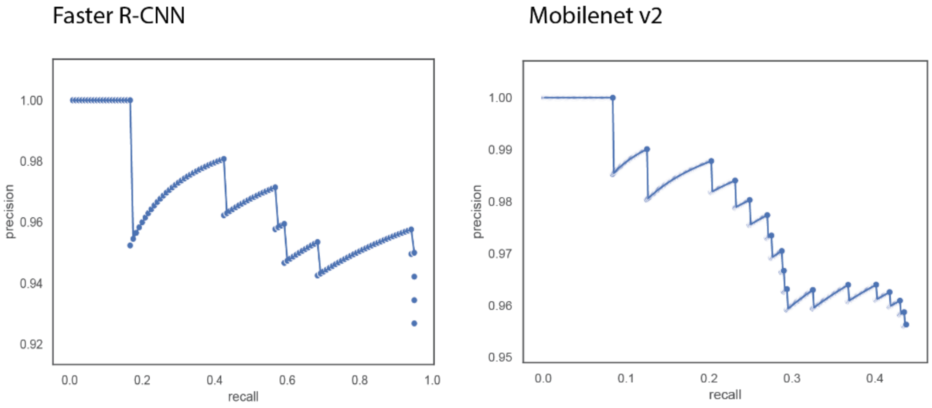


Figure 2-10. Precision- recall curves for Faster R-CNN and Mobilenet v2 CNN.

The variety of motion detection modalities we have developed for this system are well-suited for detecting a wide range of *C. elegans'* behavioral states online. For example, foraging and dwelling states persist for minutes and could be detected using this system [77]. We also demonstrated online tracking capabilities that could enable tracking of instantaneous features of behavior such as changes in velocity and acceleration. In addition, online animal tracking is important in reducing the time burden of *post hoc* behavior analysis. Implementing this in a scalable system is particularly important and represents an important technological advancement. In the remainder of this chapter, we will again consider the example of *C. elegans* quiescence to demonstrate an important biological question that is addressable through online behavior-based interventions.

2.4.6 Quiescence perturbation through scalable online motion analysis

Sleep is a highly conserved behavior across phyla in the animal kingdom, suggesting that it is very important for animal survival [78]. Several conserved characteristics of sleep include increased sleep pressure when animals are prevented from sleeping and higher response thresholds to external stimuli during sleep. These sleep characteristics have also been identified in *C. elegans* lethargus. Because responsiveness is inextricable from sleep quality, interrupting sleep regardless of the animal's present motion state ignores the contextual neural state that prompted a particular behavioral response. Perturbing quiescence based on motion level, in contrast, would allow us to push animals to their sleep deprivation limit and to examine the consequences to animal health. We chose to use intense blue light, a strong aversive stimulus and stressor for *C. elegans*, to interrupt quiescence [79,80].

While there have been previously reported systems that enable quiescence state dependent stimuli, they are poorly scalable [7,72,81]. Several of these use conventional fluorescence microscopes paired with conventional computers to gain optogenetic and calcium imaging activity, making them very costly and low-throughput [72,81]. Another system that uses a conventional computer and scientific camera has been previously shown capable of imaging 100s of worms at once to perform high-throughput *C. elegans* lifespan monitoring [6]. However, while it can detect and interrupt quiescence based on image differencing methods, this online perturbation is still limited to a throughput of a single animal at a time [7]. This limitation makes it about an order of magnitude less cost efficient compared to mi-pi.

A challenge that arises from behavior-dependent intervention is constructing the appropriate controls. On initial examination it may seem sufficient to provide pairs of animals with inverse stimulus conditions, for example one animal receives a stimulus when it performs a behavior, while a paired animal receives a stimulus when it does not perform that behavior. However, under these conditions, if the behavior does not occur exactly half the time, one animal will experience a higher dosage than the other. This control scheme neglects the potential effect of the differences in stimulus dosage rather than the specific temporal delivery of the dosage. A second complicating factor is the well-documented effects of rearing conditions on animal behavioral biases. This suggests that the best controls would be isogenic animals reared under the same conditions (i.e. from the same parent with synchronized development), then split such that one receives stimulus dependent on behavior while the control animal receives an equivalent dosage of the stimulus uncoupled from its behavior.

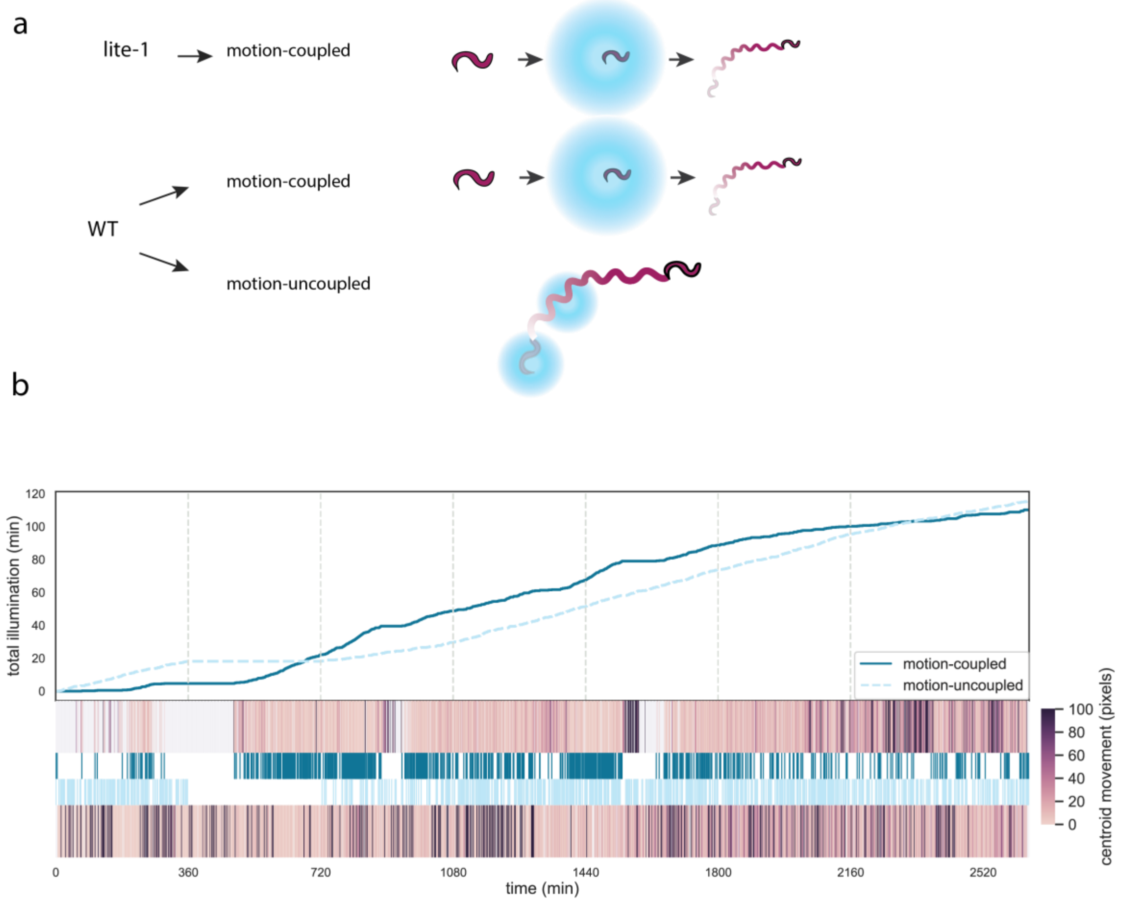


Figure 2-11 Dosage-matching scheme. a) Experimental scheme for quiescence perturbation experiments. b) Example of a paired system. Upper plot shows the total illumination time for the motion-coupled and motion-uncoupled paired system as a function of experimental time. At each gray line, the motion-uncoupled system updates its estimation of total dosage based on the dosage of the motion-coupled system. Lower heat map plots show online motion measurements (capped at 100 pixels moved) for paired systems, with the actual illumination profile for each system between the heatmaps. The motion-coupled system in shown in dark blue and motion-uncoupled system is shown in light blue. Gray sections in heatmap motion plots indicate times where no worm detection exceeded the confidence threshold of 80%.

To implement this, we developed a proportional control scheme to ensure that pairs of experimental and control animals reared together experience the same stimulus dosage despite experiencing different temporal distributions of the dosage (**Figure 2-11 b**). We accomplish this by estimating the percent of time animals will spend in quiescence, and randomly deciding at each short time interval (here, 1 minute) whether to expose the control animal to light with a probability matching our estimation of quiescence proportion. At defined intervals (here 6 hours to ensure that stereotypical lethargus time scales are well exceeded), the probability of light exposure is updated proportionally to the actual exposure of the experimental animal.

We then evaluated the effect of blue light exposure during *C. elegans* quiescence on the number of progeny (both eggs and larvae) 50 hours after the start of the blue light exposure regimen (about 6 hours after blue light regimen was stopped and about 24 hours after animals reach maturity). We dynamically perturbed *C. elegans* sleep by dosing experimental animals with intense blue light only when animals were inactive, using the Faster R-CNN to measure animal centroid movement at 1 minute intervals. Animals whose bounding box centroids moved less than 5 pixels over each interval were deemed to be quiescent. To ensure that animals were not dosed at such a high level as to be lethal, we built in both a cap on total light exposure via a total percentage of time animals may be dosed, i.e. 0% means they are never dosed, even in quiescence, and 100% means that every time they meet the quiescence criteria they are dosed. A random number generated at each update interval was compared to the total allowable light exposure percentage in order to determine the actual blue LED state. In contrast, on systems where animal motion was uncoupled from light exposure, the total light exposure measured on the motion-coupled system was compared to a randomly-generated number at each update interval to determine illumination state.

Individual L1 animals were placed on seeded agar plates with barriers to prevent animals from leaving the field of view, and grown until L3 stage, at which point each plate was aligned on a mi-pi system and experiments were started. In addition to WT animals experiencing motion-coupled and motion-uncoupled illumination, we also used animals with the light-sensing mutation *lite-1* to determine whether any effects on number of progeny were related to effects of light exposure other than light sensation, for example UV damage [82]. This experimental scheme is laid out in **Figure 2-11**.

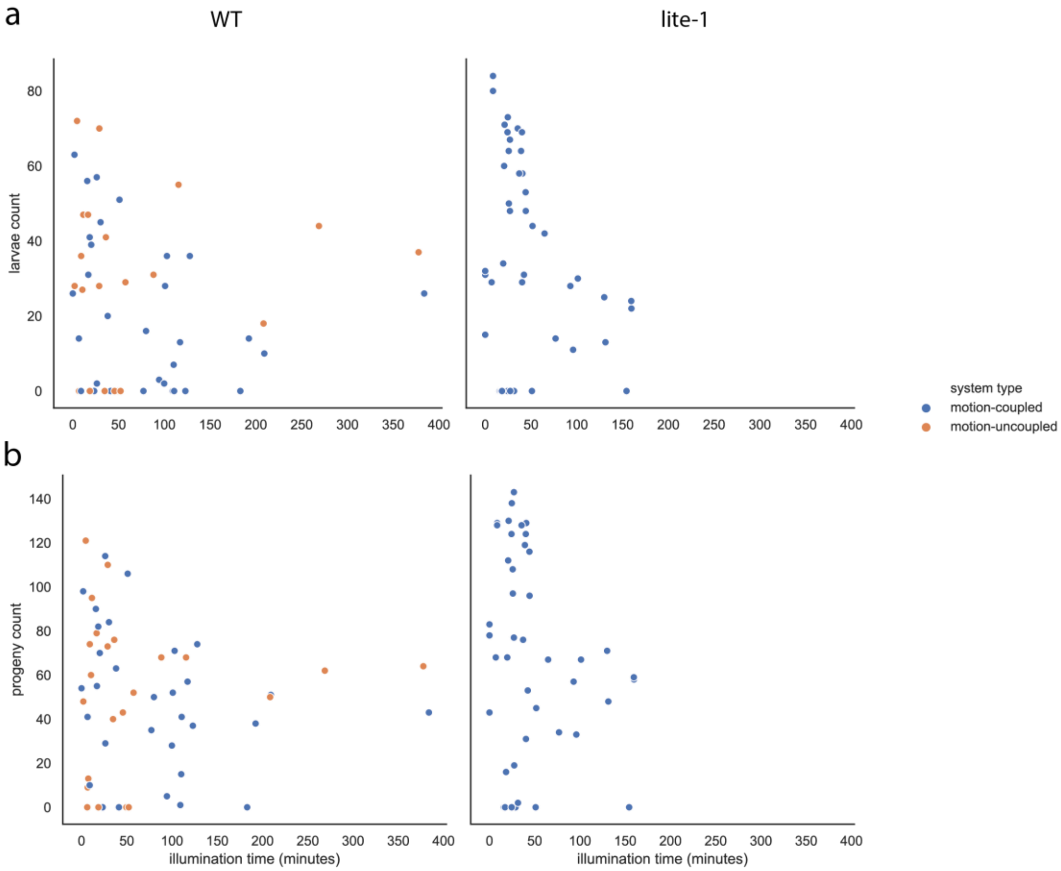


Figure 2-12. Progeny counts for sleep-deprivation model. a) Number of larvae counted for individual animals as a function of the total dosage of illumination. b) Number of total

progeny (eggs and larvae) counted for individual animals as a function of the total dosage of illumination. Orange dots represent animals on motion-uncoupled systems and blue dots represent worms on motion-coupled systems. Left column is WT animals and right is *lite-1* light insensitive mutants. WT motion-coupled n = 30, WT motion-uncoupled n = 23, *lite-1* motion-coupled N = 43.

At low light dosages (< 50 minutes total distributed over 44 hrs), we found that progeny counts of animals whose light exposure was motion-coupled, motion-uncoupled, and light-insensitive were not distinguishable, suggesting some tolerance of blue light (**Figure 2-12**). In light of this, we increased the maximum allowable exposure such that every time animals move less than the threshold distance, they are exposed to light. The resulting broad variation in larvae and progeny is expected as behavior, including egg-laying is both stochastic and individualistic. In addition, we note that motion-uncoupled animals may still experience quiescence interruption, since their light dosage is semi-random. Further, motion-uncoupled animals exposed to light dosages over about 50 minutes consistently bore more progeny by the endpoint time we used. *lite-1* animals in motion-coupled stimulation also had a strong decrease in larvae and progeny with increasing light dosage. This suggests that the sensation of light may not be the only way that our blue light stimulus affects animals. It may instead suggest that the blue light is providing a stressor during quiescent periods, potentially in addition to eliciting an aversive response from the light stimulus. While these results are not yet conclusive, greater numbers of animals exposed at these higher dosages will provide greater statistical power upon which to build.

2.5 Discussion

In this work, we have built a scalable system that enables conditional intervention into animal life. While intervention into animal life is commonplace in biomedical fields, conditional intervention is substantially more challenging, as it requires either human judgement or a high level of automation to intervene at the correct point over potentially very long time scales. Here, we showed that we can intervene on animal life conditional upon their behavior over the time scale of days.

Our results intervening in *C. elegans* quiescence so far indicate that quiescence interruption impacts animal development. While we are aware of one other system that is theoretically capable of closed-loop sleep deprivation, experimental results from this system have not been published, because this system scales poorly without a way to illuminate single animals selectively [7]. Other limitations of the system are their use of image subtraction as the sole online metric they can measure and collect timelapse data instead of continuous video data, likely because quiescence behavior occurs on second to minute timescales. In contrast, we have demonstrated that we can track animals online using a neural net. Thus, our system has potential to identify (and respond to) other behavior that occurs on the order of less than a second.

While here we demonstrate tracking of worms online using an object detection neural net, neural nets have also been effectively demonstrated for segmentation of images [83]. In particular, MobileNet DeepLab architectures allow fast semantic segmentation that would likely reach real-time or almost real time processing with the use of a USB accelerators such as the one we use here. It is possible to run both neural nets in sequence on each image, possibly also in real-time or close to it on a USB accelerator. This has important implications for potential future behavior tracking, as segmentation at high speeds could provide a way to track and respond to the worm's posture. Tracking of animal posture for animals with articulated joints has been successfully demonstrated with DeepLabCut and could also be adapted for use on the mi-pi system for animals such as bees or ants [84,85]. For *C. elegans*, it may also be very useful to track not only where worms are, but where other relevant landmarks are, including where eggs are laid, and where the bacterial lawn is. We have done several preliminary experiments tracking eggs that suggest that under slightly higher contrast or higher magnification conditions, egg counting may be feasible (**Figure 2-13**).

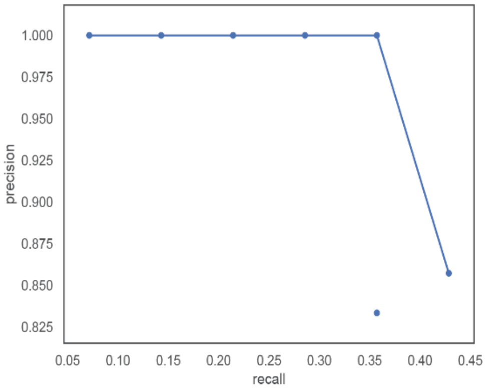


Figure 2-13 Faster R-CNN egg detection precision-recall curve. A Faster R-CNN object detection network was trained to detect *C. elegans* eggs from images captured on conventional dissecting microscopes.

In this work we have limited our intervention to a purely light-based stimulus. However, other methods of intervention could be incorporated in a straightforward way. As the Lu lab has significant expertise in microfluidics and microfluidic control, a natural next step might be to use mi-pi to sort animals based on their behavior in droplet microfluidics by using the microcontroller to control a pressure source [62], or to drug animals only under certain behavioral conditions. One could also imagine complex tests of *C. elegans* learning, wherein animals are given a choice of two food sources and given aversive stimuli only in association with one food source. These applications are currently only possible on very complex and expensive systems where they can only be achieved on a small scale.

This is a very powerful technical tool that changes the scope and scale of conditional intervention experiments. We provide extensive documentation on its construction and usage and believe that many researchers, including those outside of the *C. elegans* space, will be able to use or adapt it to their needs.

CHAPTER 3. FAST, VERSATILE AND QUANTITATIVE ANNOTATION OF COMPLEX IMAGES

3.1 Introduction

The accelerating ease of collecting very large image data sets (terabytes to petabytes) has led to a shift in scientific bottlenecks from image collection to image analysis across many disciplines, including connectomics[24,86,87], cell lineage tracing[88], and ethology[89–92]. Although highly specialized computational pipelines are emerging to address this new bottleneck, these pipelines require significant effort to develop, are computationally expensive and not error-free, and may still rely on human image annotation to establish ground truths. The widespread dependence on human image annotation or correction is likely to continue, and yet tools for image annotation, especially at large scales, often do not meet the needs of researchers.

Specifically, tools for quantitative annotation of images are hindered by a trade-off between speed, accuracy, and versatility. Some automated tools require extensive tuning or parameter optimization prior to annotation to enhance accuracy, and many image processing pipelines are not well-suited for heterogeneous image sets. In addition, many tools for human annotation limit the way users can define image features of interest, for example, via rectangles, polygons, or circles[25]. Annotation speed is limited by the complexity of annotation software, and, ultimately, how quickly annotators can mark phenotypes accurately[93]. Equally critical for efficient annotation of large datasets is ease in distributing annotation tasks, as well as broadness in settings or locations where users can annotate. To serve the greatest number of researchers effectively, tools for large scale image annotation should be generalizable, fast, and accurate.

Here we report a highly versatile, fast, and quantitative method for image annotation. Features of interest of an arbitrary image can be annotated simply from user's finger- or stylus-tracings. We demonstrate the use of a simple and intuitive smartphone- and tablet-based app to annotate complex body postures in *Caenorhabditis elegans*, morphology of stem cell aggregates, and root growth of *Oryza sativa* (rice) and *Zea mays* (corn). We crowd-sourced annotations of over 16,000 nematode images, 500 stem cell aggregate images, and 900 root images, with a total of over 30,000 user annotations (**Figure 3-1a-e**).

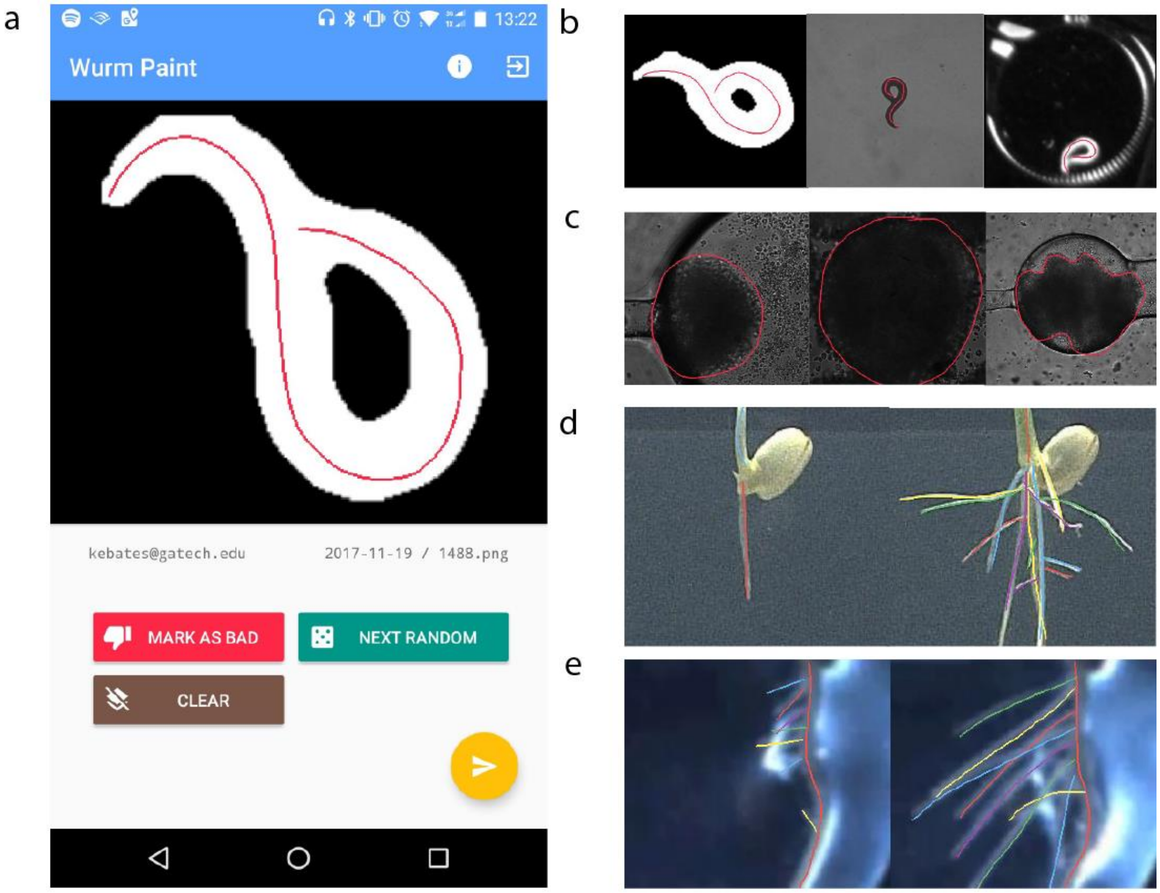
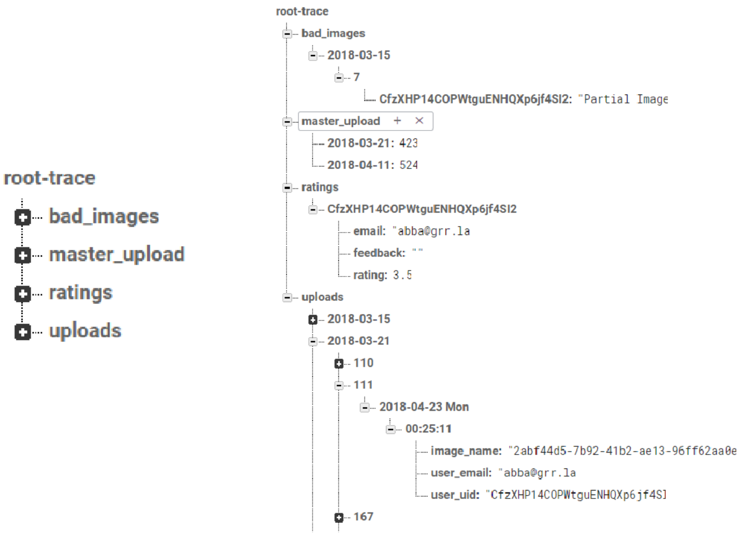


Figure 3-1 (a) Screen capture of Android interface of worm tracing app, ‘Wurm Paint’. See Supplemental Movie 1 for a video of the app in use. (b) User annotations of worm posture in binary image, grayscale brightfield image, and grayscale darkfield image. (c) User annotations of stem cell aggregate morphology using app with same source code as Wurm Paint. (d) User annotations of rice root structure. Left-hand images temporally precede right-hand images. (e) User annotations of corn root structure. Left-hand images temporally precede right-hand images.

3.2 Materials and Methods

3.2.1 Development of image annotation app

The app is written in Java, utilizing Android Studio to package the app for Android phones. Briefly, the app loads images from an online database managed on Google's Firebase to the user's Android device, on which users draw their best annotation. The structure of our database is shown in **Figure 3-2**. The user then has the option to clear their annotation and try annotating again before uploading, to report the image as something they are unable to annotate, or to load a new random image from the cloud database. Once the user is satisfied with their annotation, they upload their annotated image (as well as vectors of the annotation's trajectory) and are immediately presented with another image from the image set. The app conforms to material design and focuses on clean user interfaces for better usability and a smoother drawing experience.



```
root-trace
  bad_images
  master_upload
  ratings
  uploads
```

```
root-trace
  bad_images
    2018-03-15
      7
        CfzXHP14COPWtguENHQXp6jf4SI2: "Partial Image
  master_upload
    2018-03-21: 423
    2018-04-11: 524
  ratings
    CfzXHP14COPWtguENHQXp6jf4SI2
      email: "abba@grr.la
      feedback: ""
      rating: 3.5
  uploads
    2018-03-15
    2018-03-21
      110
      111
        2018-04-23 Mon
          00:25:11
            image_name: "2abf44d5-7b92-41b2-ae13-96ff62aa0e
            user_email: "abba@grr.la
            user_uid: "CfzXHP14COPWtguENHQXp6jf4SI
      167
```

Figure 3-2 Top-level (left) and expanded (right) structure of the root tracing app database. All apps have similarly structured databases. 'Bad_images' contains mapping to user-reported images. 'Master_upload' defines which source image sets are live on the app, as well as the number of images in each source set. User feedback is stored in the 'ratings' structure. Finally, 'uploads' maps user annotations (with user id, image name, and date and time of annotation) to the source image. In newer app versions available on our Github, we

also save line trajectories at the bottom of the 'uploads' structure. To initialize the app, only the 'master_upload' structure is needed.

3.2.2 Beta-testing of WurmPaint example application

During beta-testing, we recruited 6 users, 4 of whom work with *C. elegans* on a daily basis, with the other 2 having some general knowledge of the worm. Although we collected more annotations from other users, including children, only the annotations from these 6 users were used in our behavioral analysis of *C. elegans*, and make up >90% of the total annotations collected.

Several Morningside Elementary School (Atlanta, GA) students tested the usability of the app by drawing worm shapes with their finger. Combined, students annotated approximately 30 worm images, none of which were used in worm posture analysis (see below). All children who annotated using the app did so with the verbal consent of their parents, and no demographic or other information was collected from them.

Annotations from users of our example app published on Google Play are not included in this study, but we inform users on our Google Play site, on our app information site (https://sites.google.com/view/wurm/app-privacy-policy), and within the app itself what information we collect: user emails so that they may establish an account; annotations they produce; timestamps of when each annotation is updated. No demographic information is collected from users and we do not contact users via their email or share their email addresses.

To familiarize non-expert users with typical worm movement and shapes, we assembled a brief tutorial https://sites.google.com/view/wurm/tutorial. As general guidelines, we asked users to draw a continuous contour along the midline of the worm, starting at one end of the worm to the other end, so that the contour did not contain sharp corners, rather smooth bends along the length of the worm.

3.2.3 Measurement of annotation speed

We collect timestamps when users upload images and drawing vectors with a resolution of 1s, based on the user's device's time. To determine a conservative average user annotation speed, we grouped all annotations by user and computed the time between each upload for that user. All inter-upload times were pooled. Because inter-upload times could range from a few seconds to days depending on the user's usage frequency, we imposed an upper threshold of 30s for worm image annotations and an upper threshold of 90s for root image annotations to determine the average user annotation speed.

3.2.4 Worm tracking

We built upon an existing worm tracker[10] for our initial image analysis and to identify movie frames where worms were partially self-occluded (i.e. ambiguous). A subset of these frames was uploaded to our database for annotation. Using the generative algorithm included in the existing worm tracker to predict worm posture for occluded shapes, we optimized parameters for our data set and found predicted worm postures for several full videos from which we had drawn ambiguous postures for our database. The worm tracker uses MATLAB software (we used MATLAB version 2017a). To evaluate the time required to process individual frames using this worm tracker, we used MATLAB to measure how long the point-swarm (PS) optimization (generation of alternative posture predictions) required for each ambiguous frame. This step took an average of 776.2 ± 5.2 s/ ambiguous frame (95% CI, n = 66) with parallel processing (a local pool consisting of 4 cores). After this generative step, a progressive optimizing interpolation (POI) step evaluates the alternative posture predictions to determine which makes sense in the context of the worm postures in the surrounding frames. For this step, we timed the total time until a solution was generated. For a movie with 444 ambiguous frames, this step required 155.5 s/ ambiguous frame (equivalent to the time required for > 20 human annotations). Combined, the PS and POI tracking steps required on average 931.7 s/ ambiguous frame, or the equivalent of 133 human annotations. The computers used were Dell Precision Tower model 5810 with 32 GB RAM and Intel Xeon CPU (model E5-1620 v4, 3.5 GHz).

3.2.5 Post-processing of annotated worm images

Although the current version of the app allows us to upload the coordinate trajectories of user annotations, the initial version that much of the data presented here originates from only the annotation superimposed on the source image. Thus, to extract annotations and reconstruct trajectories from uploaded images, some post-processing of annotated images was required. Briefly, to identify annotations, we found non-grey pixels in each image. We then binarized the annotation alone and skeletonized the image, followed by removal of branch points if branch points existed. We then checked the curvature of each line segment to ensure it fell in a reasonable range – if it did not, we broke the segment at its point of maximum curvature. Using the resulting line segments, we attempted to reconnect them to each other using both the proximity of segment endpoints and local segment slope. Once segments had been reconnected, the worm's midline was reconstructed using the projections onto the first five eigenvectors as described previously[12]. Average speed of this post-processing was 0.0597 s/ frame (n = 1000). This process is illustrated in **Figure 3-3a**, and code for these steps is available in our GitHub repository. However, we emphasize that other app users need not perform any post-processing of images. Instead, coordinate trajectories can be accessed by parsing JSON files that are downloadable from our Firebase database.

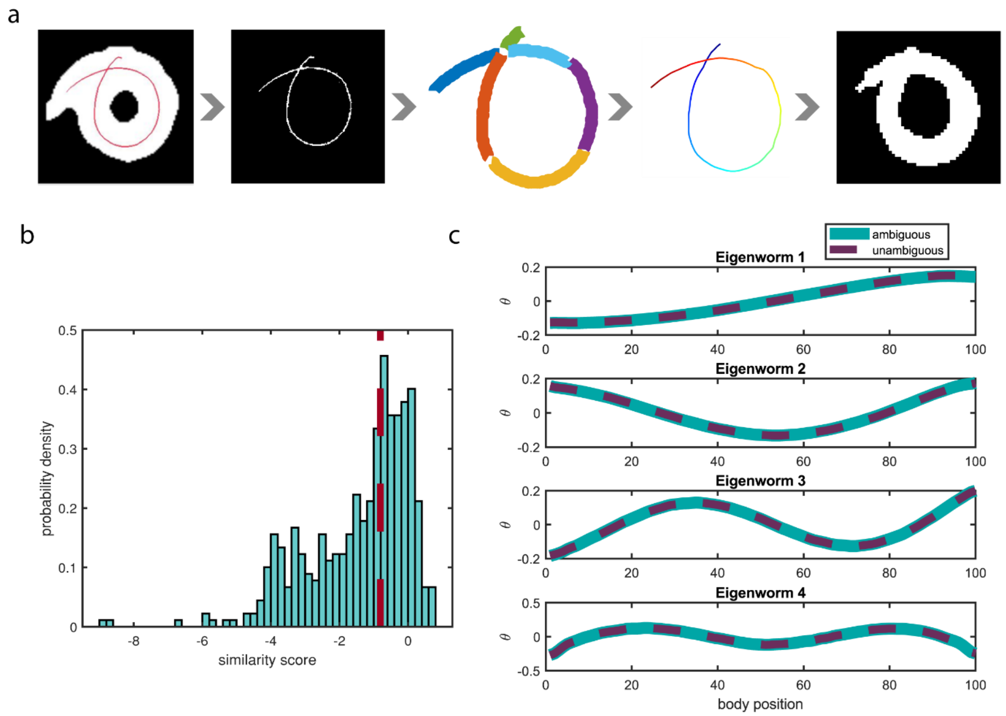


Figure 3-3 Worm annotation characterization (a) Reconstruction pipeline for worm midlines. Worm backbone annotations used in the main text were collected as images superimposed on the source image, so midlines must be reconstructed. Newer versions of the app save the drawn trajectory directly, so reconstruction is unnecessary. We found non-gray pixels (annotations) within the annotated images and binarized the annotation,

followed by breaking the annotation at points of intersection or extreme curvature. Then we used local curvature and distance metrics to predict which line segments were connected and reconstructed the midline and image. (b) Probability density of similarity scores for ambiguous posture predictions compared to ambiguous posture consensus annotations. Red dashed line is threshold used for consensus generation. The broader similarity score distribution compared to unambiguous annotation to unambiguous ground truth comparison is caused partially by user variability and lower user accuracy and partially by the predictive nature of the state-of-the-art algorithm that sometimes leads to incorrect solutions. N = 449 (c) First four eigenvectors ('eigenworms') of the *C. elegans* posture space computed from four annotated videos (>37,000 frames). Computing eigenworms from only unambiguous postures or both ambiguous and unambiguous postures resulted in little difference, as reported in other work. Compared to eigenworms reported in other work, ours are similar, but with different eigenworms capturing a greater fraction of the total postural variability.

3.2.6 Similarity score calculation

In order to compare two worm annotations, or a worm annotation to ground truth, we matched 100 points between two worm midlines and computed the Euclidean distance between each pair, summing all of these distances and normalizing the distance by 75% of the width of that particular worm at each of the 100 matched points. Mathematically,

$$1 - \sum_{i=1}^{100} \frac{||midline1_i - midline2_i||_2}{0.75 * worm\ thickness_i} \tag{1}$$

This metric is equal to one when both worm midlines match exactly at each point. The greater the deviation from one, the worse the midline equivalency.

3.2.7 Consensus contour generation

To construct consensus midlines from user annotations, we noted that even for pairs of reconstructed midlines that were below a zero similarity score, users were making essentially the same annotation. To identify a threshold similarity score below which we could consider two annotations to be from distinct groups, we modeled the distribution of similarity scores from user-user comparisons (**Figure 3-4d**) as a mixture of gaussians. The primary mode was centered at -0.068 and the secondary mode was centered at -3.260. To ensure that most generally similar annotations were grouped together, we computed a threshold two standard deviations below the primary mode, a similarity score value of -0.809. We found that several other methods of identifying this similarity score threshold identified thresholds that ranged from slightly positive to slightly negative. These methods included the Otsu thresholding method on user-user similarity scores and searching for the lowest threshold of the user-user similarity scores for which the Wilcoxon rank-sum test failed to reject the null hypothesis that the user-user similarity scores and user-ground truth similarity scores were drawn from the same distribution at the 5% significance level.

Having identified a reasonable threshold, we generated consensus contours. During this process, we used the projections of worm backbones into the space of the first five eigenvectors. We identified and removed annotations whose eigenvector projections were outside of the range of *C. elegans* posture space. Then, for each source image, we identified all annotations of the source image and removed any remaining outlier annotations of that image, where an outlier is a value more than three absolute deviations away from the median. We computed similarity scores for all pairs of annotations and used our previously identified threshold to identify pairs of images that were very similar to one another. We further grouped these pairs into larger groups of similar annotations and identified the group of similar annotations with the largest number of members. For example, if image pairs (1, 2), (2, 3) and (5,6) all have similarity scores above our threshold, we take the union of all pairs that contain images 1, 2 and 3 and, separately, the union of all pairs that contain images 5 and 6. If more images belong to the first union set than the second, we use the first set to calculate a consensus contour by finding the centroid of this group of contours in the five-dimensional space of posture projections.

3.3 Results

3.3.1 General operation and versatility

Our app is indiscriminate to the nature of images or annotations. Worm images on our database were derived from brightfield and darkfield microscope configurations, solid and liquid imaging environments, and included both processed, binarized images as well as unprocessed frames from raw videos (**Figure 3-1b**). Stem cell aggregate images on our database were derived from phase images of both live and fixed aggregates grown in tissue culture plates as well as aggregates grown in microfluidic devices [94] (**Figure 3-1c**). For both nematode and stem cell aggregate applications, users are presented with randomized images from the full dataset and draw a single contour. This generic annotation scheme could also be used to trace individual cells or features of developing organisms (such as *Drosophila melanogaster*, *Xenopus*, or zebrafish), to name a few. To allow users to annotate video frames in a pre-defined order (e.g. when temporal context is critical to annotation) and in cases where an image contains multiple features of interest, we created a second version of the app that presents uploaded images in order and allows users to draw as many contours as needed. We used this app version to annotate rice and corn root systems (**Figure 3-1d-e**). We expect that these two versions of the app could serve many other image annotation problems equally well with little to no changes of the source code.

3.3.2 Simplicity and Speed

The app is extremely easy for annotators to use. By using smartphones as the basis for our image annotation system, users need only draw with a finger or stylus, as compared to the greater difficulty of drawing with a computer mouse, or, as in ImageJ, drawing piecewise lines. The interface itself is simple and intuitive compared to popular image annotation and analysis tools. We had 7-12 year-olds use the worm tracing app, and found that it was simple enough for them to use without help after a brief explanation (**Figure 3-4a**). Although the quality of children's annotations was far more variable than annotations by adults, many of the children's annotations were of indistinguishable quality compared to those of adults and annotations inconsistent with other user's annotations were easy to identify.

We sought to demonstrate that our app enabled fast annotation. For two of our applications, we quantified the time between image uploads of single users as a conservative estimate of time per annotation. For worm tracing, which always required a single user-drawn contour, the average annotation time was 7 ± 0 s/image (95% CI), and for root tracing, which often required multiple contours per image, the average annotation time was 14 ± 1 s/image (95% CI). To benchmark user annotation speed in our app, we annotated worm images using ImageJ[23], which routinely required more time. In addition to the importance of individual users' speed, overall speed is dependent on how many users can annotate in parallel. Smartphone-based annotation not only allows us to easily distribute image annotation tasks as narrowly (a single expert) or broadly (general public) as desired, it also expands geospatial locations and settings where users can annotate[95].

3.3.3 Annotation accuracy

We assessed the ability of users to trace known shapes accurately. We did this by comparing averaged hand-drawn worm postures to computationally generated ground truth postures. For worms with unambiguous postures, we matched points along the averaged hand-drawn worm midlines with points along the corresponding ground truth midline and summed the Euclidean norm between all point pairs. To determine an overall similarity between any two worm midlines, we reasoned that an acceptably similar midline should lie within the center three-quarters of the worm's total width at any given point. We therefore normalized similarity scores so that a score of one indicated identical midlines, any positive score indicated that the midlines were on average less than three-quarters of the worm width apart, and negative similarity scores indicated that midlines were further than three-quarters of the worm width apart (**Figure 3-4b**). Most averaged annotations of unambiguous postures had similarity scores above zero when compared to their corresponding ground truth midline, including data collected from non-expert annotators (**Figure 3-4c**). We concluded that the annotation accuracy was sufficiently high for tracing worms.

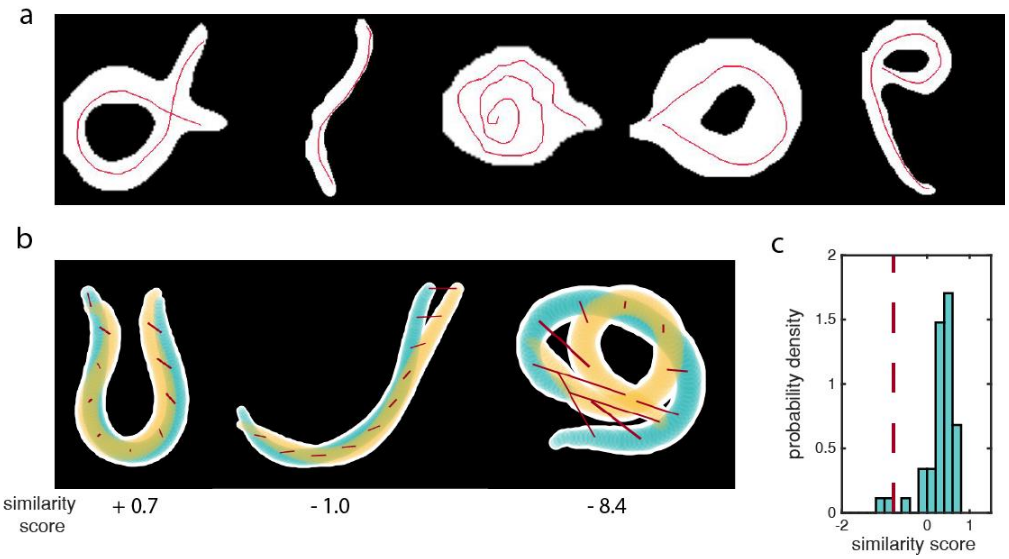


Figure 3-4 (a) Annotations by 7-12 year-olds. (b) Sketch of similarity score calculations. For each panel, two worm contours (white overlaid with yellow or blue) are reconstructed based on ground truth or annotated midlines. 100 points along the midline are matched, and the Euclidean distance between each pair is computed. Here we show this at 10 points along the backbone (red lines). Yellow and blue highlight the center three-quarters of the worm's width. (c) Probability density of similarity scores for unambiguous posture solutions compared to averaged user annotations of the same unambiguous postures. The dashed red line indicates the threshold we use to calculate consensus contours. N = 44.

3.3.4 Consensus across annotators

To further demonstrate a practical application of our app, we focused on using annotations of ambiguous *C. elegans* postures to reconstruct the dynamics of worm behavior. Ambiguous postures result from segmentation errors, or more frequently, the worm partially occluding itself, for example during stereotyped Ω- or δ-turns. A major advantage of using human annotators is the ability to quickly generate varied predictions for images that humans and algorithms alike struggle to find a ground truth for. *C. elegans* postures are often simplistic and sinusoidal, but ~7% of the worms' behavior results in postures that are impossible to segment using current tools. One approach relies on computationally expensive optimization to attempt a quantitative posture description[10,50]. Although accurate in most instances, this state-of-the-art strategy for predicting ambiguous nematode posture requires on average 931.7s (n = 66) per video frame. Based on our average worm annotation time, users can make predictions about 130-fold faster than this computational strategy. User predictions for individual ambiguous images varied, but could typically be grouped into several distinct shapes, indicating that there were often only a few reasonable predictions for each ambiguous posture (**Figure 3-5a**). To characterize this variability quantitatively, we calculated pairwise similarity scores comparing different annotations of the same image for more than 500 source images and found that similarity scores peaked between zero and one and had a left-skewed distribution with a significant tail (**Figure 3-5b**). This is consistent with our observation that although there is significant variability in user annotations, users are frequently in agreement with one another, suggesting the utility of a consensus-based approach in identifying a best solution. The ease and speed of generating viable predictions based on human intelligence with the app gives it particular advantage in analyzing images where a single ‘correct’ solution is non-existent and several solutions have high likelihood.

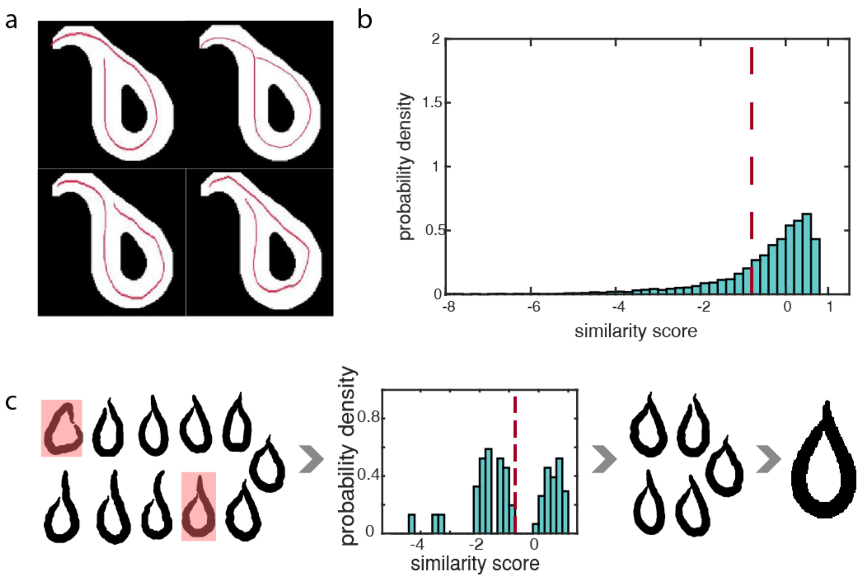


Figure 3-5 (a) Example set of ambiguous images. (b) Probability density of similarity scores for comparison between different user annotations of the same ambiguous posture. The dotted red line indicates the threshold we use to calculate consensus contours. The threshold was determined by modeling probability density as a mixture of two Gaussians (see Methods). N =26,098. (c) Illustration of consensus generation scheme.

To resolve the ambiguities in our postural data set, we used annotations to create a consensus prediction for ambiguous images (**Figure 3-5c)**. For each source image, we first eliminated annotations that were outliers or that created shapes outside of *C. elegans* postural space, then used pairwise similarity scores to identify groups of similar annotations. We chose the group containing the most individual annotations, and averaged annotations in this group to come to a consensus contour. We compared these disambiguated annotations to predictions generated by the state-of-the-art computational method and found that the mode of the similarity score distribution was -1, indicating that although consensus contours had somewhat reduced accuracy, they overall agreed well with computational predictions **(Figure 3-3b**). Further, for frames where initial segmentation failed, users could correctly annotate grayscale source images, while computational predictions were erroneous.

3.3.5 Reconstructing C. elegans behavioral dynamics

C. elegans is a powerful model organism with a large suite of tools for genetic manipulation[41,96]. These tools, along with a fully mapped nervous system[3], have enabled researchers to identify molecular mechanisms and individual genes associated with behavioral phenotypes[97–99]. However, quantitative analysis of some of the most complex behaviors, large-angle turns that commonly include ambiguous postures, remains difficult, and gaps in quantifiable behavior prevent dynamic posture analysis altogether. Using our consensus worm contours, we recreated the postural repertoire and behavioral dynamics of *C. elegans*. We sought to answer how significantly complex worm postures affect the overall shape space of *C. elegans*. To answer this, we calculated the first four principle components of *C. elegans'* shape space[12] ('eigenworms') using either unambiguous results alone or both unambiguous results and consensus contours (**Figure 3-6a, Figure 3-3c**). Consistent with prior reports, we found that the first four principle components were very similar with or without ambiguous postures[10]. Interestingly, the fractional variance of the worm's posture space captured by these eigenworms is greater when ambiguous postures are included **(Figure 3-6b)**. Lastly, we recreated complete timeseries of the first four eigenworm amplitudes for individual worms using the consensus contours (**Figure 3-6c**). These traces fill in the gaps left by ambiguous shapes and outperform the computational prediction in some cases where the worm is tightly coiled. In addition to adding to our knowledge of *C. elegans* behavioral dynamics purely through image annotation, this app can help improve existing posture prediction algorithms by using these results.

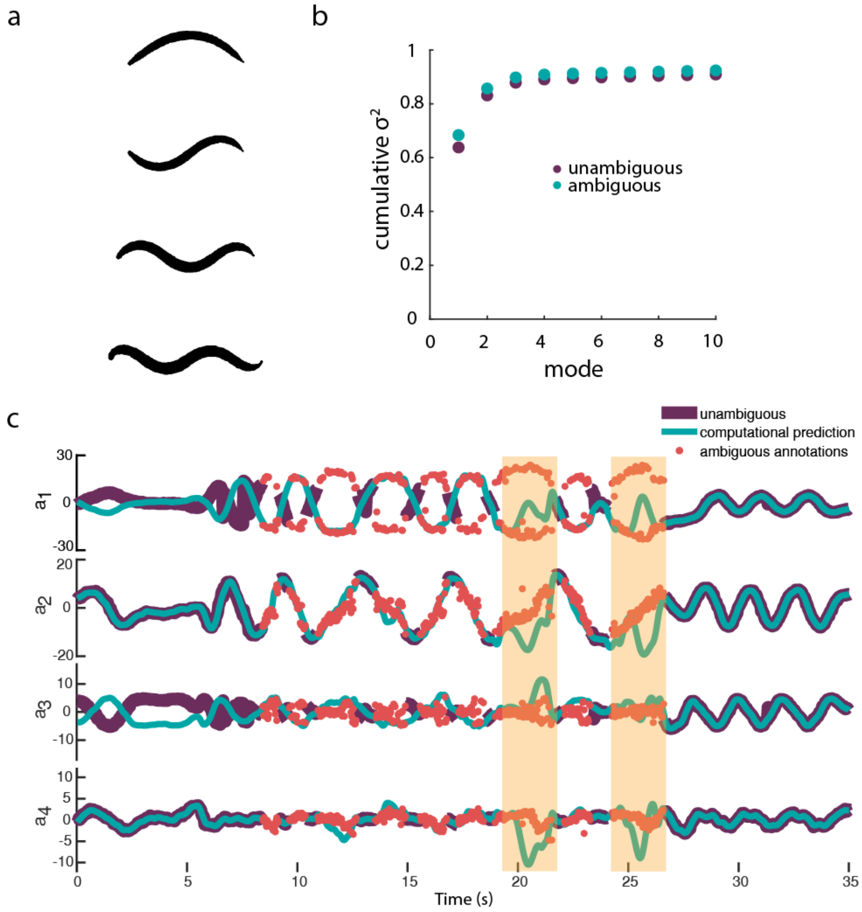


Figure 3-6 Reconstruction of continuous behavior dynamics from annotations. (a) Representation of first four principle components of worm posture (‘eigenworms’) including both unambiguous and ambiguous postures from four annotated videos in our dataset (37,784 frames). As reported in other works, our eigenworms both with and without ambiguous postures were very similar (see **Figure 3-3**). (b) Cumulative variance captured by each additional eigenworm (‘mode’) for both unambiguous images only and both unambiguous and ambiguous images together. (c) Traces of amplitude of first four eigenworms in time for an individual worm. Dark purple lines are amplitudes calculated for unambiguous postures via image processing. Gaps in purple lines correspond with frames containing ambiguous postures that are usually associated with reorientation of the worm. Blue lines are computational predictions for the full video, including ambiguous postures. Red dots represent consensus contours for individual frames found using app user

annotations. Yellow highlighted regions are time points where computational predictions do not match consensus predictions. The top inset image is the unphysical computational prediction at the timepoint corresponding to the grey dashed line. The bottom inset image is the consensus contour prediction generated from the app at the same timepoint. For the amplitudes of the first eigenworm in particular, the red dots follow two opposing sinusoidal contours simultaneously, one contour representing the opposite head orientation of the worm compared to the other contour.

3.4 Discussion

Our app-based annotation scheme allows researchers from any field to quickly and easily annotate complex images in quantitative ways. Here, we demonstrated its flexibility and speed in annotating rice root growth and structure, stem cell aggregate morphology, and complex worm postures, where we showed that the app is ~ 130-fold faster than state-of-the-art posture optimization techniques. We expect that the app will be useful as an alternative to creating complex and bespoke computational image processing pipelines, as a way to complement and augment existing computational pipelines, and as a simple way to generate consensus ground truths towards improving machine learning algorithms for image processing.

As image datasets become larger and the demand for more nuanced analysis of complex image features becomes commonplace, human annotation will remain important in establishing ground truths and correcting outputs from automated analysis pipelines. We will continue to need tools for large-scale human annotation to accomplish these tasks. Here, we demonstrated an efficient method for collecting complex annotations; the time needed for an individual annotation is shorter than a comparable annotation using other software. Thus, this is a useful tool even when the total number of annotations needed is modestly scaled. For applications that require much greater scale, recruitment of more annotators may be necessary. Several methods for recruiting annotators include paying annotators a small amount of money for each annotation by integrating with Amazon Turk[25], gamifying the app[100], or creating a citizen science effort[86]. We envision that this type of accurately human-curated images will support (by providing ground truth) and complement (in rare and unanticipated scenarios) the machine learning approaches as they become dominant in image-based analyses for many fields of scientific inquiry.

CHAPTER 4. MAPPING THE BEHAVIOR SPACE OF C. ELEGANS

4.1 Introduction

In this chapter, I adapt a machine learning method for classifying, quantifying and comparing *C. elegans* behavior and use it to map *C. elegans'* behavior space. I apply this method to a large set of video behavior data collected across a selection of environments that span a range of viscosities similar to the worm's natural environment. From this behavioral phenotyping, we identified subtle stereotyped behaviors that are not easily distinguishable by eye. In addition, we built a tool for interactive visualization and exploration of our video data set that can be repurposed for other datasets where points are associated with video data. Finally, we found that the behavioral diversity of *C. elegans* was not uniform across the environments we sampled, suggesting that media choice substantially impacts the ability to distinguish between phenotypes.

4.2 Background

Many animal species demonstrate behavioral flexibility that enables them to adapt to acute shifts in their physical surroundings [101]. For example, many mammals and amphibians split their time between aqueous and land environments, and the environments of invertebrates may fall across a wide range under different seasonal conditions, including broad changes in material properties of habitats. How animals adapt their behavior to their environment is governed by physics, neural circuits, and musculature systems that make understanding and comparing behavior dynamics in different environments challenging.

One invertebrate that is naturally found in complex environs is *Caenorhabditis elegans*. These ~1mm long roundworms are a well-studied model system that are cultured in the lab on agarose plates with bacterial lawns. Their behavior on agar plates has been consistently studied and used to understand the genetic inputs to behavior for over half a century [29,65]. *C. elegans* can also be cultured in bulk liquid, suggesting that these worms are capable of behavioral plasticity exceeding what is typically perceived in their normal lab culture [65,102]. This is further borne out by the isolation points of *C. elegans*, primarily rotting fruit and plant matter [103,104]. Studies of *C. elegans* population density in rotting fruit show that later stages of decomposition are correlated with greater population size [105]. While this variability in proliferation can be at least partially explained by the availability of nutritious bacteria, it is not well understood whether the physical features of decomposing plant matter contribute to advantageous behavioral adaptations on the part of the worm [106]. Recent evidence has shown that *C. elegans* navigate preferentially to stiffer and tighter spaces (durotaxis and thigmotaxis, respectively), suggesting that some advantage may be conferred by migration to more resistive substrates [30,107].

Based on these observations, we wondered how or whether the *C. elegans* behavioral repertoire might change to suit its physical surroundings. While significant effort has been devoted to understanding whether gait changes when moving across different viscous media are primarily governed by mechanics or neural processes, there is limited comparison of other aspects of *C. elegans* behavior in these media [108–113]. For example, during taxis *C. elegans* uses both sharp reorientations and gradual steering to navigate sensory gradients [32,33]. It has also recently been demonstrated that *C. elegans* uses head swings to integrate sensory information for navigation, suggesting that head movements are particularly important to the worms' behavioral capabilities [114]. Indeed, several recent studies demonstrate that the full breadth of *C. elegans* behavior is not necessarily well known. New behavioral descriptions include a stereotyped large-angle reorientation behavior termed a δ-turn, and so-called rolling behaviors that enable the worm to reorient in 3-dimensional environments [10,115].

While it is not possible at present to observe *C. elegans* in their natural habitat for practical reasons, many microfluidic tools have been developed that allow researchers to manipulate the environment of small model organisms. Microfluidics are compatible with many optical microscopy methods and have been demonstrated as useful tools for constructing chemosensory gradients and understanding chemotactic behaviors, as a method for applying well-defined amount of force for understanding mechanosensation, and to understand thigmotactic behaviors [28,30,34,44]. 'Soil-like' microfluidic devices have been used to help worms behavior in microfluidics conform to similar behaviors to those described on agarose media, including forward movement, reversals, and stereotyped high-angle turning behaviors like Ω- and δ-turns[28]. To the other extreme, microwell-like microfluidic devices can be used to isolate individuals and monitor their behavior in liquid at long timescales [43]. In order to observe the behavior of many age-synchronized animals simultaneously while restricting their movement to the 2-D plane, we chose to use these microchamber devices to isolate animals while immersing them in media of different viscosities.

While using microfluidics allowed us to scale up our behavior monitoring significantly, comparing animal behavior across environments is not trivial. Metrics commonly used to describe *C. elegans* behavior, such as velocity or thrashing frequency, are not sufficient to describe the potentially complex forms of *C. elegans* behavioral dynamics. With current methodology, it is difficult to categorize animal behavior without imposing biased heuristics, in particular across disparate environments.

Here we apply a machine-learning methodology, t-distributed Stochastic Neighbor Embedding (t-SNE), to a behavior dataset consisting of more than 12 million frames of data (110 hours of individual animals); we do so over a set of physical environments including aqueous media, solid media, and a range of viscous media. This machine-learning technique has previously been applied to mapping the stereotyped behavior of fruit flies and to understand the behavioral effects of optogenetic stimuli in both fruit flies and very recently to *C. elegans* [11,53–55]. We chose this technique because it relies solely on the postural dynamics of the animal to identify stereotyped behaviors, rather than relying on heuristic behavioral definitions based on human inspection. From this set of behavior data, we examined the dimensionality of *C. elegans* posture and behavior in both conventional lab environments as well as more complex environments. Our data suggest that the worm's postural dimensionality remains low-dimensional, and that behavior dimensionality is dependent on the environment *C. elegans* is behaving in. From the environments sampled, we found that behavioral diversity was greatest at intermediate viscosities that enabled us to identify subtle changes in head movements. This implies that the ability to discriminate *C. elegans* behavioral phenotypes could be enhanced by phenotyping worms in the appropriate media. Considering the importance of behavioral phenotyping in *C. elegans* as a tool for understanding how genetics relates to behavior, this result should inform the design of behavioral phenotyping studies.

4.3 Materials and Methods

4.3.1 C. elegans maintenance

Unless otherwise noted, *C. elegans* were cultured under standard conditions at 20°C. Strains used in this work include N2 and AQ2334: *lite-1(ce314)*; *ljIs123[pmec-4::ChR2*; *punc-122::rfp*].

4.3.2 Plate behavior assays

For optogenetic experiments, OP50 bacterial lawns were supplemented with the ChR2 cofactor ATR. Stock ATR solution of 50mM was diluted to 200 μM in OP50 suspension and used to seed 5cm NGM culture plates (100uL per plate). In parallel, a population of animals from the same strain was cultured on plates seeded with 100uL of OP50 suspension without ATR from the same batch. Animals were cultured for at least 2 generations on either OP50 or OP50 supplemented with ATR before being used in experiments.

Before all optogenetic experiments, unseeded plates were warmed to room temperature and animals were picked onto these plates and allowed to starve for an hour. At the end of the hour, animals were picked onto fresh, room temperature plates and allowed to crawl freely.

To stimulate the animals, we used a microscope and illumination system described previously[48]. Briefly, a motorized stage and online image processing is used to keep the animal in the center of the microscope’s field of view, while a modified projector is used to selectively illuminate individual animals with blue light. Videos of animal behavior were collected from this system at 30 Hz and a magnification of 10x.

4.3.3 Microfluidic behavior assays

For all microfluidic behavior assays, animals were synchronized via hatch-off and cultured on plate until they reached day 1 adulthood. These synchronized populations were then washed off of culture plates with M9 buffer. Unless otherwise noted, video data was collected on a dissecting microscope (Leica MZ16) using a CMOS camera (Thorlabs DCC3240M), with a framerate of 30 Hz and a magnification of 1.2x.

4.3.3.1 In buffer

Microfluidic devices were fabricated as described previously [43]. After fabrication of microfluidic devices, devices and tubing and connectors were autoclaved. Animals were loaded into devices as described previously after washing worms off plates with M9 buffer. Behavior was then recorded over the course of about an hour. Importantly, cavities in which worms are loaded in these devices are only slightly greater depth than the width of an adult worm, which restricts worms to the focal plane of the microscope and almost entirely 2-dimensional behavior. Animals spent a maximum of 2 hours total in buffer without food. **Figure 4-3** describes the number of animals, percent of frames originating from each environment, and total frames analyzed.

4.3.3.2 In methylcellulose

Methylcellulose solutions were prepared at concentrations of 0.5%, 1%, 2%, and 3% weight in volume of M9 buffer. For a 50mL total volume of each concentration, we first heated 20 mL of M9 on hot plate to > 80°C to aid in dissolving the high concentrations of methylcellulose. The methylcellulose was added and stirred with a magnetic stir bar for ~ 10 min until methylcellulose was wetted and mixed reasonably well. We added the remaining 30mL of M9, chilled to 4°C, and removed from the hot plate. The solution was moved to a 4°C fridge and put on a stir plate to continue to stir for an hour, at which point the solution was no longer cloudy.

We measured the viscosity of the methylcellulose solutions with a rheometer and confirmed that as with previous studies, the methylcellulose solutions are primarily Newtonian over the range of shear rates typical of *C. elegans* in these conditions (**Figure 4-1**)[108].

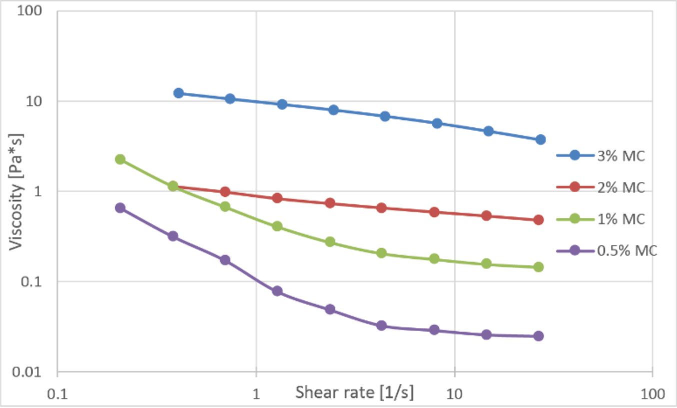


Figure 4-1 Viscosity of methylcellulose solutions as a function of shear rate. Shear rates typical of *C. elegans* movement are between 5 and 15 Hz. MC stands for methylcellulose.

The extremely viscous nature of the methylcellulose solutions prevented us from loading animals into bonded microfluidic devices; however, we wanted to maintain similar conditions compared to buffer and agarose environments, especially confining the animals to an almost entirely 2-D plane of behavior. To ensure that single animals could be isolated in single chambers of the unbonded microchamber microfluidic device, we first picked animals onto a room-temperature, unseeded plate. To ensure that animals were fully immersed in methylcellulose mixture, we used a glass pipet to aspirate a small amount of methylcellulose solution, and then aspirated animals from the unseeded plate one at a time into the methylcellulose solution. Then, single animals surrounded by methylcellulose mixture were pipetted into individual chambers of an unbonded PDMS chamber device. The device could then be flipped over onto a sterile 10cm Petri dish and gently pressed down until the individual chamber walls came into contact with the Petri dish, preventing animals from leaving their chambers. Animals were then imaged in devices for about 20 minutes.

4.3.4 Posture tracking and analysis

For initial posture tracking we used EigenwormTracker code described in Broekmans et al., 2016, to track the midline of worms while in non-occluded, simple postures [10]. We modified this code to allow us to better segment animals in our darkfield conditions in microfluidic devices. Due to the size of our behavioral dataset and time required for predicting each occluded frame (close to 30 minutes per frame), we found that using their generative tracking algorithm was computationally infeasible. We thus decided to drop frames in which worms were self-occluded in our dataset. Frames in which segmentation errors could be inferred from inconsistency in the shape and size of the worm were also dropped from the dataset. After elimination of these frames, about 88% of total frames remained.

After identifying midlines of animals throughout time, we can then express the posture of worms in 'worm-centric' coordinates by taking the angle between consecutive, equally-spaced points along the worm's midline. Then we used principle component analysis (PCA) to identify a low-dimensional vector space in which the worm's posture can be expressed, following methods previously described [12]. For each environment considered here (agarose, 3% methylcellulose, 2% methylcellulose, 1% methylcellulose, 0.5% methylcellulose, and buffer), we first found the PCA vector space for each environment separately, and calculated residuals, defined as $1-\sigma$, where σ is the cumulative variance captured by each additional eigenvector. When embedding all methylcellulose conditions and the buffer condition together using t-SNE, we used the eigenvectors computed from all methylcellulose and buffer conditions together. The eigenvectors computed from these conditions separately were very similar (see **Figure 4-3**), and the eigenvectors resulting from all the methylcellulose and buffer conditions together were nearly identical to the eigenvectors computed from these conditions separately.

Once these eigenvectors (also referred to as 'eigenworms') were calculated, we expressed all worm postures as a linear combination of the first 10 eigenworms. We chose the first 10 eigenworms as a conservative cut-off for how many eigenworms represented data signal compared to noise. When we shuffled our midline angle data so that correlation between position of each angle along the length of the worm's body was destroyed, we found that for each environment at most 6 eigenvectors contributed to signal. For all t-SNE clustering that considers only one environment, the animals behavior is expressed as a linear combination of the eigenworms calculated for that environment alone. For t-SNE clustering that considers more than one environment, the animals behavior is expressed as a linear combination of the eigenworms calculated for that combination of environments.

4.3.5 T-SNE clustering

To characterize posture dynamics from eigenworm posture descriptions, we followed the methodology laid out in Berman et al., 2014 [11] (**Figure 4-2**). Spectrograms were generated from eigenworm posture descriptions by applying a Morlet continuous wavelet transform at 25 frequencies dyadically spaced between 0.2 Hz and 30 Hz. As the video framerate for all movies was 30 Hz, the Nyquist frequency of the data is 15 Hz, and is therefore the meaningful upper bound of the frequency spectra. We therefore do not show any mode amplitude maps corresponding to these high frequencies. While the spectrogram does not encode the directionality of the animal's movement (i.e. whether the animal is moving forward or backward), in low-viscosity fluids such as buffer there is no clear evidence that *C. elegans* can actively navigate. In addition, microchambers used to house worms have diameters only slightly longer than the length scale of the worm itself (adult worms are ~ 1mm in length, while the microchamber diameter is 1.1mm). We thus decided to ignore the directionality of the movement.

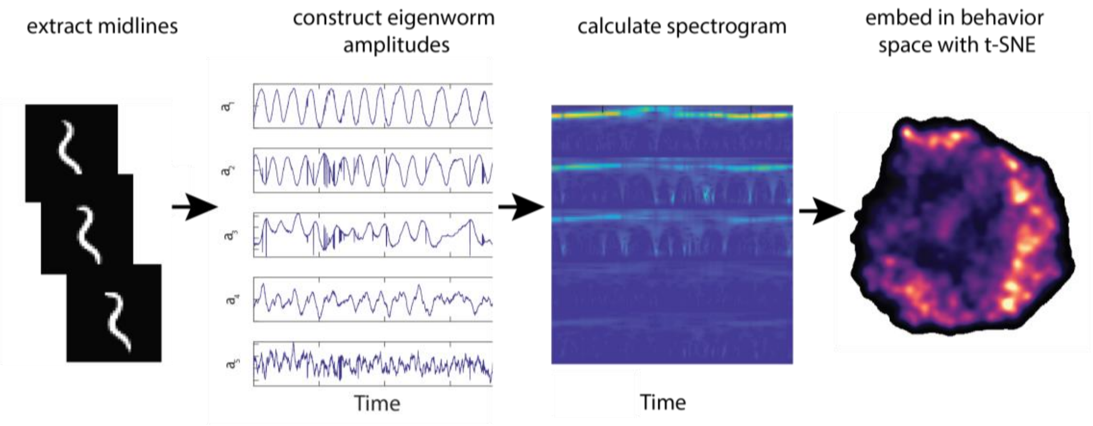


Figure 4-2. Embedding of behavioral dynamics using t-SNE. First, animals are segmented and posture is extracted. Posture is then represented as a timeseries of amplitudes by projecting into the eigenworm space. Next, temporal feature vectors are generated by creating spectrograms from postural timeseries. Finally, the t-SNE clustering algorithm is applied to embed timepoints into clusters that represent stereotyped behavior. A PDF of an embedded dataset is shown.

To cluster and define behaviors independently of our external definitions, we used t-distributed stochastic neighbour embedding (t-SNE) to embed our wavelet data into a 2-dimensional space in which local distances between feature vectors (here wavelets) are conserved while distant points may be distorted [116]. Our dataset was sufficiently large to make it infeasible to embed all data at once, so we used the subsampling method described in Berman et al., 2014 to select data for a training embedding into which the remainder of our dataset could be projected. Since our movement dataset was not fully continuous due to segmentation errors or self-occlusion, we only selected training data from stretches of uninterrupted frames longer than 5s. Our full training sets consisted of wavelets from 35,000 timepoints. For embeddings of individual environments, all training wavelets were exclusively from the dataset for that environment. For embedding of behavior from all methycellulose and buffer environments together, we sampled equally from wavelets originating from each environment (i.e. 7,000 timepoint feature vectors from each of the 5 environments in the embedding).

4.3.6 Defining behavior states and a behavior map

Within t-SNE embeddings, clusters of points close together indicate a well-stereotyped behavior, so we use probability density (PDF) maps of the 2-dimensional space to visualize where such behaviors exist in each map. We segmented each map by applying a watershed function to the corresponding PDF, after smoothing the PDF using an isotropic Gaussian filter with sigma equal to 8 to reduce over-segmentation. After segmentation, we used video data as well as eigenworm representation across each map to visually confirm behaviors and provide descriptions for map regions.

In order to visualize how each eigenworm was represented across each behavior map, for each frequency and eigenworm we plotted the average amplitude (linear multiplier) for the corresponding eigenworm at each point in the behavior map.

4.3.7 Significance testing for behavior maps

In order to test which regions in two maps were significantly different from one another, we used the hierarchical bootstrapping method described in Saravanan et al., 2019 [117]. The video behavior data collected has several layers of hierarchy, including different days of data collection and different animals for each of our environment conditions. While under traditional pooled statistical tests, independence between different days or different animals is assumed, the hierarchical bootstrapping method allows us to avoid making that assumption. For each environment, we first sampled from the different days on which experiments were performed with replacement such that the overall distribution of days remained the same; then we sampled with replacement from the individual animals on each day; finally we sampled with replacement from all embedded points for that animal on that day. This hierarchical bootstrap was repeated 1000 times for each environment. From these bootstrapped embedding distributions, we calculated 1000 new PDFs.

At each point in the PDF space for each environment, we then had a distribution of densities. We modelled the density distributions as a gaussian mixture model with between one and four components (whichever fit best as determined by the AIC metric), and used these to compute joint probability distributions for each point in the PDF space for pairs of environments. As the null hypothesis that both distributions are the same implies that the joint probability distribution would be a circle lying on y=x, integrating the joint probability distribution to one side of the line allows us to test the hypothesis. A volume greater than 1-α/2 implies the first group is significantly greater than or equal to the second and a volume less than α/2 implies the second group is significantly greater than or equal to the first. We applied a Bonferroni correction to correct for multiple testing.

4.3.8 Data visualization tool

To visualize our very large set of behavior data, we developed a web-based tool written in Java using the D3 visualization library (**Figure 4-5**). The tool allows interaction with the embedded data points that enables intuitive exploration of the data. The full dataset of over 12 million timepoints was randomly subsampled to 3 million points for faster website loading times. From this data, a further subsample of 350 embedded points is displayed so that the point density allows users to click on individual points. This selection of points can be resampled if desired. Clicking on a point displays metadata including the environment of the animal and the proportion of animals captured by any selected filters. An embedded YouTube link to the original worm video allows users to view the raw data starting from a timepoint within about one second of the embedded point. To relate these points to the probability density that is indicative of a stereotyped behavior, users can switch between display of the sampled points and display of the entire dataset as a heatmap to inspect the density of point embeddings. As of Feb 2020, this web-based tool is available at https://wurmvis.ebb.gatech.edu/.

4.3.9 Behavior transitions

At each timepoint in our data, we identified the corresponding discrete behavior using the behavior maps generated by watershedding embedding PDFs. Since frames in our dataset have been dropped, we filled stretches less than or equal to 15 frames (half a second) of missing data with the same discrete behavior as the last known embedding point. Missing data in stretches longer than 15 frames were removed and the resulting behavior vectors considered as not contiguous when computing transition behavior. Once discrete behaviors were assigned to timepoints, we counted the frequency of each behavior transitioning to any behavior other than itself in contiguous portions of data, normalizing the resulting matrix of behaviors so that they represent a probability of any behavior transitioning to any other. The first transition ($\tau = 1$) represents the first behavior transition after the initial, the second transition ($\tau = 2$) represents two transitions after the initial behavior and so forth. As $\tau \rightarrow \infty$, the behavior transition matrix will converge to the probability of being in any particular behavior state.

4.4 Results

4.4.1 Posture space across environments

C. elegans posture is conventionally understood to be low-dimensional. Work by Greg Stephens and colleagues originally demonstrated that the posture of the worm requires only five eigenvectors to capture more than 95% of its postural variance [12]. This general result of postural low-dimensionality has been replicated many times for animals behaving on solid media, both for wild-type animals as well as in behavioral mutants [50]. Examination of *C. elegans* posture in three dimensions has also revealed strikingly similar results, with just four 3D eigenworms capturing 95% of the animals postural variance [118]. However, the postural dimensionality of *C. elegans* over a range of environments with varied mechanical properties is unknown. We asked whether the dimensionality of *C. elegans'* posture would increase in physical environments more closely resembling their natural habitat.

To address this question, we first imaged worm behavior across five environments with varying mechanical properties. These physical environments included conventional aqueous buffer, 0.5%, 1%, 2%, and 3% methylcellulose, and conventional agarose media [65]. Animal behavior was recorded at 30 Hz in microfluidic devices (for aqueous and methylcellulose media) or on typical agarose-filled Petri dishes. For each environment, the behavior of up to 190 individuals was recorded for between 4 and 30 minutes. The combined data contains more than 12 million video frames and 100 hours of posture data for individual animals.

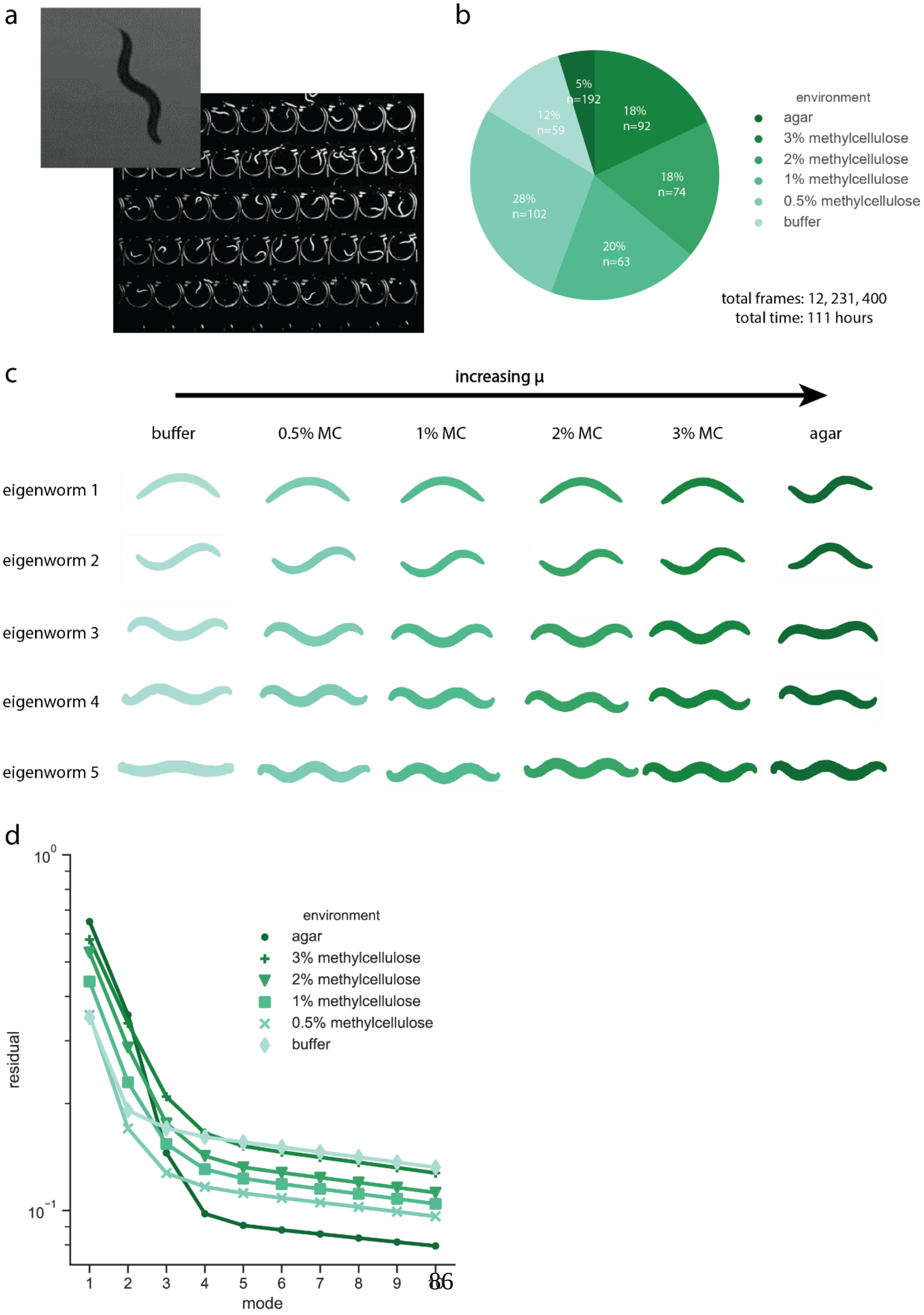

a
b
5%
n=192
18%
n=92
18%
n=74
20%
n=63
28%
n=102
12%
n=59
environment
● agar
● 3% methylcellulose
● 2% methylcellulose
● 1% methylcellulose
● 0.5% methylcellulose
● buffer
total frames: 12, 231, 400
total time: 111 hours
c
increasing μ
buffer
0.5% MC
1% MC
2% MC
3% MC
agar
eigenworm 1
eigenworm 2
eigenworm 3
eigenworm 4
eigenworm 5
d
residual
10^0
10^{-1}
mode
1
2
3
4
5
6
7
8
9
environment
agar
3% methylcellulose
2% methylcellulose
1% methylcellulose
0.5% methylcellulose
buffer

Figure 4-3 Postural space across varied environments. a) Example frames from behavior recording on agarose (front) and in microfluidic devices (back) b) Data proportion from each environment, with n numbers indicating the number of animals assayed in each environment. Total frames indicate total segmented frames. c) Pictorial representation of the first five eigenworms as calculated from each environment independently from lowest viscosity to highest viscosity. d) The remaining unexplained postural variance (residual) as a function of each additional eigenworm mode, as calculated for each environment independently for the first 10 modes.

We found that despite the wide range of physical environments, the posture space of *C. elegans* remained relatively small, with four eigenworms capturing over 80% of postural variance in all cases (**Figure 4-3 d**). While the eigenworms we derived from behavior on agarose media were similar to those described previously, we found that eigenworms derived from behavior in buffer and methylcellulose environments were different from our agarose-derived eigenworms as well as eigenworms derived in other works (**Figure 4-3 c**). Typical thrashing or swimming behavior in buffer frequently produces a single large amplitude bend over the body of the worm, which the first eigenworm in buffer environment captures. However, while this change was intuitive, more surprising was the similarity between eigenworms for animals behaving in buffer and all concentrations of methylcellulose (**Figure 4-3 c**). In addition, we saw that the postural variability that each additional eigenworm captured was not uniform across all environments. In fact, while the first four eigenworms of animals behaving on agar captured about 90% of the postural variability for animals crawling on agar, the first four eigenworms of animals behaving in buffer captured 83% of the postural variability for animals swimming in buffer. This may suggest that the physical properties of agar limits the postural diversity of *C. elegans*, perhaps due to reduction in the animal's overall degrees of freedom.

Interestingly, we saw a trend indicating that the postural variability captured by the first eigenworm in each environment decreased in decreasing viscosity media (**Figure 4-3 d**). For each environment, the slope of the residual changes drastically, after which each additional mode only incrementally improves the explanation of postural variability. Animals behaving in buffer reached this slope change the fastest after just 2 modes, while animals behaving in any methylcellulose media reached this slope change after 3 modes, and animals behaving on agar reached this slope change after 4 modes. This observation in combination with the observation that overall postural variability captured by 'buffer only' eigenworms was much lower seemed to suggest that animals behaving in buffer may have fewer primary postures with potentially more postural extremes compared to animals behaving on agarose.

The implications of this for behavioral phenotyping are important. Studies that use *C. elegans* to link genes to behavior rely on comparing behavioral phenotypes of mutants. Our comparison of postural diversity suggests that selecting an environment to perform postural phenotyping, and potentially behavioral phenotyping in is non-trivial and could affect the ability to effectively link genes to behavior. In order to better understand whether not just postural diversity but behavioral diversity might change in our sampled environments, we needed to compare behavioral repertoire between our distinct environments.

4.4.2 Behavioral space comparison between environments

While our postural analysis suggested that animals behaving in aqueous and methylcellulose solutions had a lower postural complexity than animals behaving on agarose, this does not necessarily imply a change in behavioral complexity across environments. To assess whether behavioral complexity was similar across environments, we used a machine-learning technique to cluster similar behaviors together [11]. Briefly, each animals posture can be projected into the eigenworm space at each timepoint, and then the spectral features of a timeseries of projections is used as the input for the t-SNE clustering technique [11,116]. This technique has been previously demonstrated with flies and worms and used to dissect neural circuits and understand stereotypy and hierarchy in animal behavior [53–55,119]. Previous application of this technique to *C. elegans* behavior focused on linking behavioral responses to optogenetic stimuli for animals behaving on agarose media. Under the assay conditions used in that work, clustering *C. elegans'* behavior resulted in a low-dimensional representation that primarily encoded for the worms' velocity. However, whether this would hold in more diverse environments was not known.

To initially compare the behavioral complexity of *C. elegans* in our 6 environments, we clustered data from each environment into a behavior map for each environment separately, using watershedding as described above (81) to segment the behavior map into discrete, stereotyped behaviors. As an initial measure of complexity in each environment, we simply counted the number of discrete behaviors identified through watershedding. On agarose, we identified 32 discrete behavior regions (**Figure 4-4 a**), on 3% methylcellulose 28 discrete behavior regions (**Figure 4-4 b**), on 2% methylcellulose 27 discrete behavior regions (**Figure 4-4 c**), on 1% methylcellulose 39 discrete behavior regions (**Figure 4-4 d**), on 0.05% methylcellulose 35 discrete behavior regions (**Figure 4-4 e**), and in aqueous media we identified 34 discrete behavior regions (**Figure 4-4 f**). This suggested some level of similarity in overall complexity of behavior. However, the behavior density maps indicated that in some environments just a few stereotyped behaviors dominated. This suggested that some environments might enable greater breadth of behavior. However, comparing individual maps to each other with this technique prevents us from directly comparing behaviors between environments and examining whether or where their behavior space overlaps.

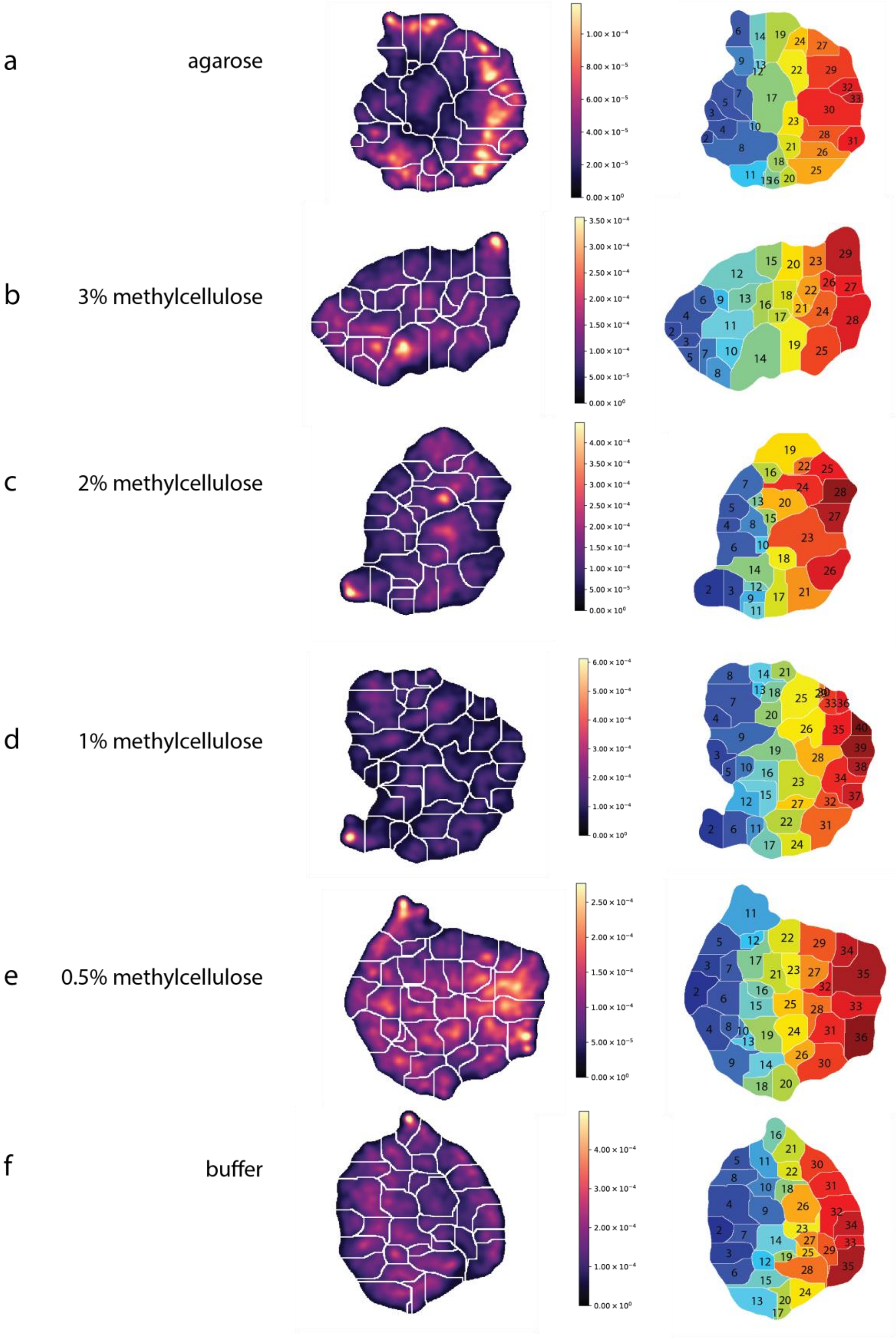

a
agarose
1.00×10^{-4}
8.00×10^{-5}
6.00×10^{-5}
4.00×10^{-5}
2.00×10^{-5}
0.00×10^{0}
b
3% methylcellulose
3.50×10^{-4}
3.00×10^{-4}
2.50×10^{-4}
2.00×10^{-4}
1.50×10^{-4}
1.00×10^{-4}
5.00×10^{-5}
0.00×10^{0}
c
2% methylcellulose
4.00×10^{-4}
3.50×10^{-4}
3.00×10^{-4}
2.50×10^{-4}
2.00×10^{-4}
1.50×10^{-4}
1.00×10^{-4}
5.00×10^{-5}
0.00×10^{0}
d
1% methylcellulose
6.00×10^{-4}
5.00×10^{-4}
4.00×10^{-4}
3.00×10^{-4}
2.00×10^{-4}
1.00×10^{-4}
0.00×10^{0}
e
0.5% methylcellulose
2.50×10^{-4}
2.00×10^{-4}
1.50×10^{-4}
1.00×10^{-4}
5.00×10^{-5}
0.00×10^{0}
f
buffer
4.00×10^{-4}
3.00×10^{-4}
2.00×10^{-4}
1.00×10^{-4}
0.00×10^{0}

Figure 4-4 Behavior maps for each environment PDF with behavior regions overlaid (first column) and discretized behavior map (second column) for all animals behaving on or in the named media. a) agarose, b) 3% methylcellulose, c) 2% methylcellulose, d) 1% methylcellulose, e) 0.5% methylcellulose, and f) buffer.

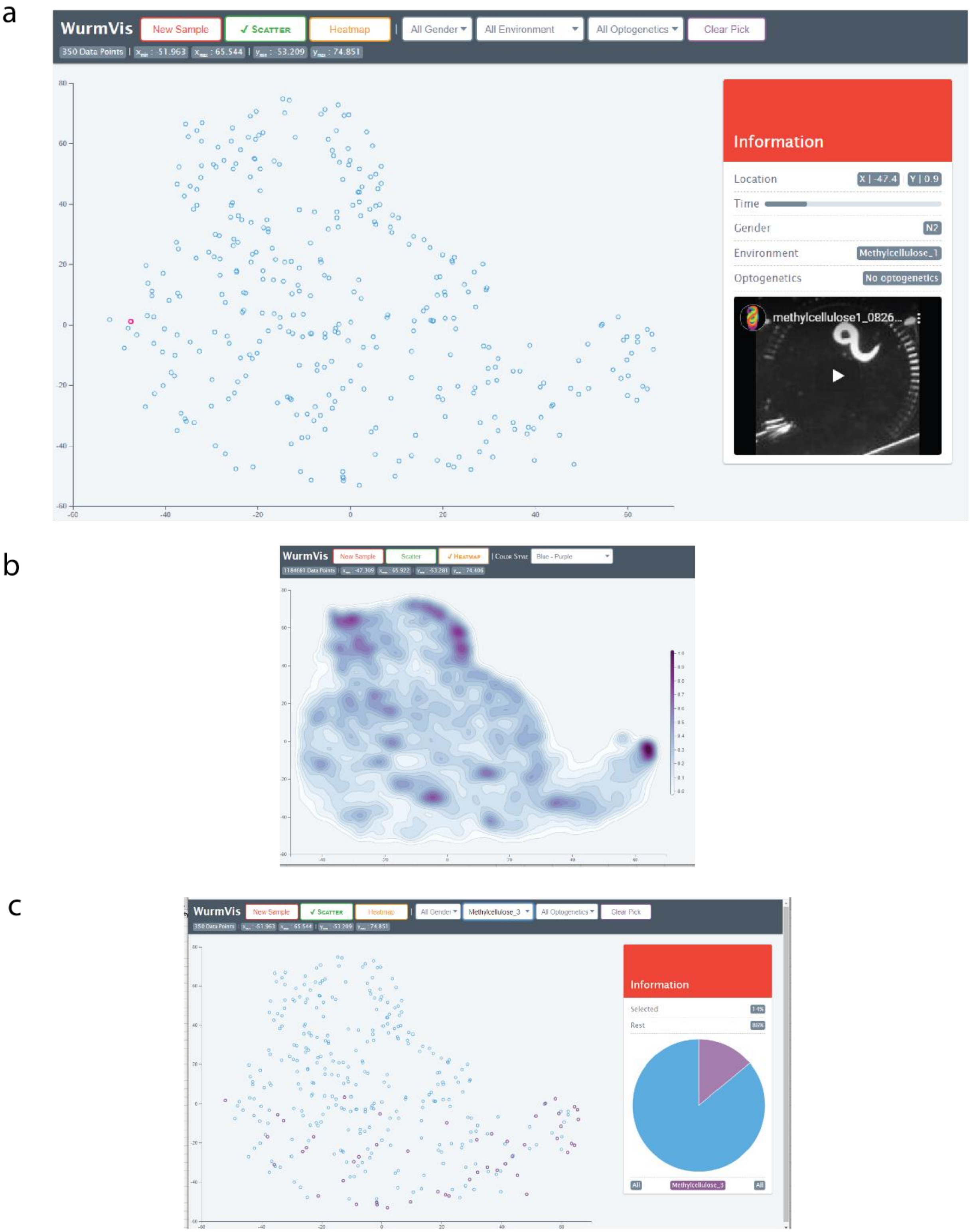
To further probe whether animals in all environments had access to the same breadth of behaviors, we examined how each of the first five eigenworms was represented across the breadth of the frequency spectrum analysed.

4.4.3 Tool for visualization and interaction with video data

A critical challenge of describing behavior data with machine learning is verifying the output has meaning. This can be particularly challenging when working with video data, which is difficult to explore and interact with, especially at large volumes. In order to aid in our own and others understanding of our video data, we built an interactive tool that allows users to view the raw video data associated with each timepoint embedded into our behavior map (**Figure 4-5**). As of Feb 2020, this tool is available on the web at https://wurmvis.ebb.gatech.edu/.

The main screen presents a sparse set of points from the united behavior map described below. Clicking on a point highlights the point and brings up information about the point, including an embedded YouTube video of the raw data starting from within a few seconds of the selected point (**Figure 4-5** a). Users can also filter the data by environment and generate a new sample of sparse points from the larger dataset (**Figure 4-5 c**). By switching to the heatmap tab, users can view an estimate of the PDF of the full dataset (**Figure 4-5 b**).

This tool makes it easier to verify a meaningful output from behavior maps. In addition, it makes available all our raw data for other researchers to reuse in a straightforward way via YouTube. Finally, it can be modified and used to interact with any video data that is associated with a 2D point, not just this method of behavioral embedding.



Location X | -47.4 Y | 0.9
Time
Gender N2
Environment Methylcellulose_1
Optogenetics No optogenetics

Selected 14%
Rest 86%

Figure 4-5 WurmVis interactive data exploration. a) Main screen interface with one point selected (pink). Right-hand sidebar contains metadata and embedded raw video. b) Heatmap that approximates PDF contours of all embedded points. c) Interface filtering feature highlights points within a given condition. Purple highlighted points are timepoints from animals behaving in 3% methylcellulose. Right-hand sidebar shows proportion of total datapoints that the filter applies to.

4.4.4 Unified behavior map

We also wanted to examine how much overlap or uniqueness there was in behavior across the environments we sampled. In order to examine this, we created a unified behavior map of all animals behaving in methylcellulose and aqueous conditions. We excluded animals crawling on agarose as the agarose-associated eigenworms were quite different from those of animals in our other environments. To make sure each environment was equally represented, we subsampled our data as described above to obtain a training embedding, and then embedded the remainder of our dataset into the same space. In our unified behavior map we identified 25 potentially stereotyped behaviors (**Figure 4-6**).

PDF maps of each environment within this unified behavior space depicted sharp separation between environments, suggesting that there were indeed distinctive behavioral shifts between each (**Figure 4-6**). The simplest explanation for these behavioral shifts would be a change in frequency of a universal behavior. To examine whether this was the case, we first used the wavelet feature vector of eigenworm amplitudes across frequencies to determine what eigenworms were represented to what extent at each frequency (**Figure 4-7**). We found that the first eigenworm was highly represented across the map at low frequencies, and at higher frequencies highly represented to the left side of the map. This eigenworm is likely to be associated with both deep bends and reorientation behaviors, as these often involve large turning angles. Eigenworms two and four are both well-represented at the bottom of the map at low frequencies and the top of the map at high frequencies, and their shape in addition to their representation pattern suggests that they may be involved in crawling behavior. Similar to the first eigenworm, the third eigenworm is best represented at high frequencies to the right side of the map, and finally the fifth eigenworm is best represented to the bottom right at low frequency. These patterns gave some indication of what general behaviors might be present where on the map, and matched well with our high-level intuition of dynamics of those behaviors. For example, animals behaving in buffer move much faster than those in methylcellulose, and the primary measured behavior of worms in buffer in literature is a simple thrashing motion that could be well-represented by the first eigenworm. At the highest frequencies, the amplitude of the first eigenworm is high near the top of the map, which matches where our buffer behaviors lie. Despite these clues as to what behaviors might be where, we could not explore whether more nuanced behaviors were also represented without examining the video data itself for patterns.

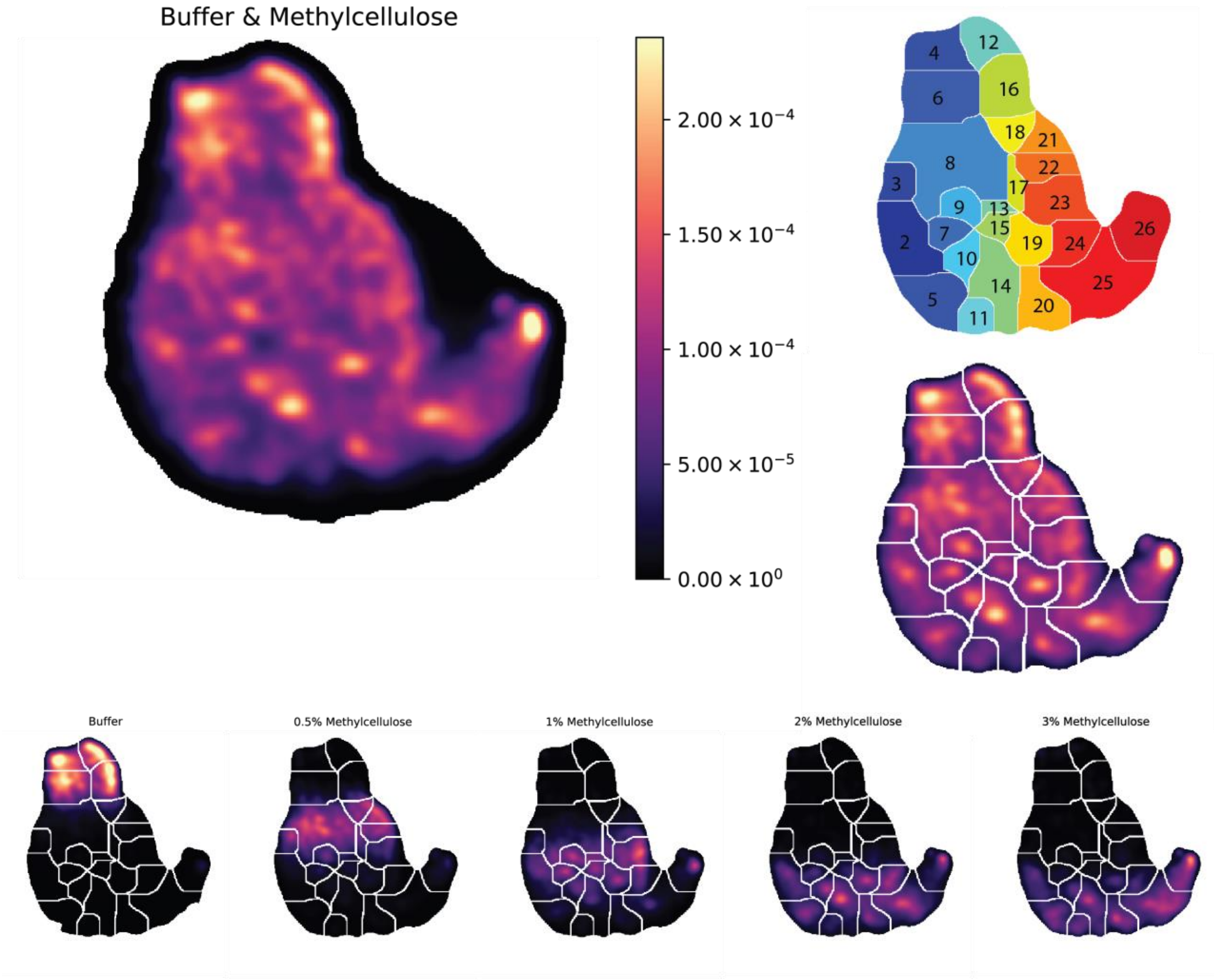


Figure 4-6 Unified behavior map a) t-SNE embedding of all methylcellulose and aqueous data into a single unified map. b) Color-coded and numbered discrete behavior regions, top. PDF overlaid with behavior regions, bottom. c) PDF for animals behaving in the environment named.

To confirm this, we composed short videos of individual animals whose behavior fell within each region, and from these were able to annotate broad regions of the unified behavior map (**Figure 4-8, Figure 4-9**). Some sections of this map did represent behaviors that are considered to be a continuum of a single behavior, such as the transition from ‘thrashing’, the typical descriptor for *C. elegans* behavior in aqueous media to slower thrashing, to crawling to slow crawling that unfolds from the top of the unified map to the bottom. We also noted that some regions of the behavioral space described subtler aspects of *C. elegans* behavior, such as regions that were characterized by extreme bends, particularly near the worms head (e.g. regions 2 and 3), or short wavelength bends near the head combined with long wavelength bends at the tail of the worm (e.g. region 9).

While there was little behavioral overlap between all environments, we noted that all idle states regardless of environment shared a region (region 26). Behavioral overlap between sets of environments was overall small. Buffer and 0.5% methylcellulose overlap only at a narrow band between regions, 0.5% methylcellulose and 1% methylcellulose overlap most notably in region 8 (slow thrashing) and 9. The overlap between 1%, 2% and 3% methylcellulose, however, is more easily discernable, suggesting that some limits of behavior have been reached due to the physical characteristics of the environment and the energetic output of the worm. In these more viscous media we observed the emergence of much deeper bends, especially near the head of the worm, that overlapped across viscosities. To evaluate more rigorously what behaviors were similar between each pair of environments, we computed where each pair was significantly different (**Figure 4-10**). Most notably, 2% and 3% methylcellulose were the most similar, with the prevalence of regions 2 (a slow, deep bend) and 25 (small head movements with hardly any forward movement) similar enough as to not be significant.

Using the unified behavior map, we were also able to compare the apparent complexity of *C. elegans* behavior in each viscosity range by counting the number of regions represented in each environment. Animal behavior in buffer was almost completely captured in just 4 regions that were all identified to be animals thrashing at varying frequencies. Animal behavior in 0.5% methylcellulose was primarily captured in 8 regions composed of slower thrashing behaviors, with notably greater amplitude bending compared to animals in buffer. Animal behavior in 1% methylcellulose spanned 15 regions, ranging from behaviors similar to crawling to very deep, localized bends at the head. In 2% and 3% methylcellulose, 10 regions dominate that range from slow crawling and idle animals to high amplitude head bends. The relative breadth of behavior in each environment, both by visual inspection and the number of regions defined, suggests that 1% methylcellulose provides a richer behavioral environment compared to the other environments sampled.

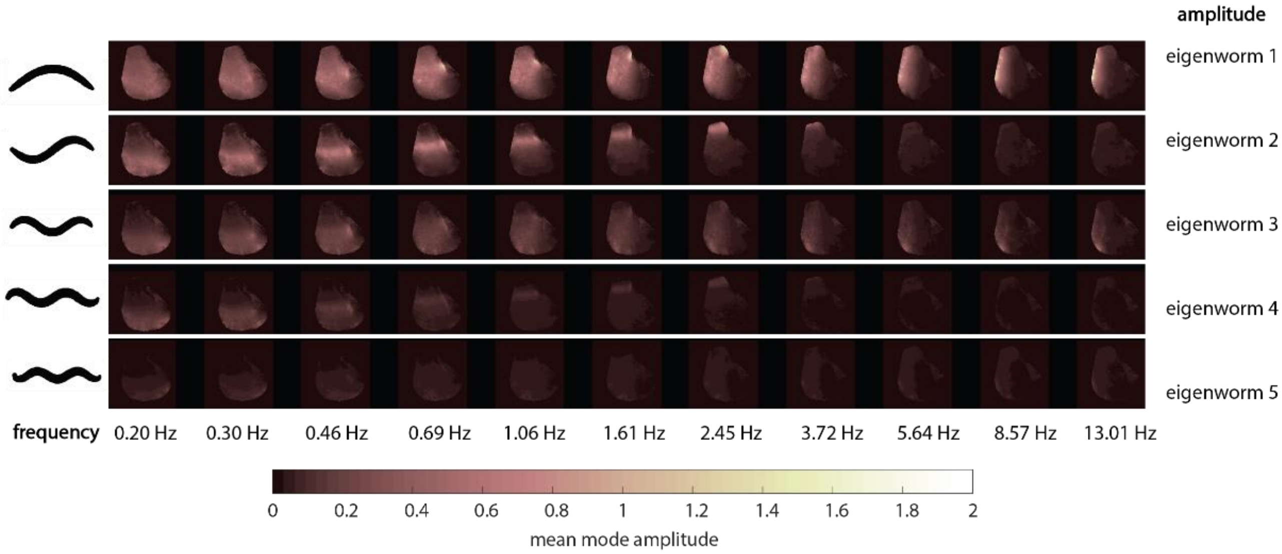


Figure 4-7 Mode activity across united behavior map. The mean amplitude of each eigenworm at each point in the behavior map across 11 frequencies. Pictorial representations of the eigenworms used are to the left.

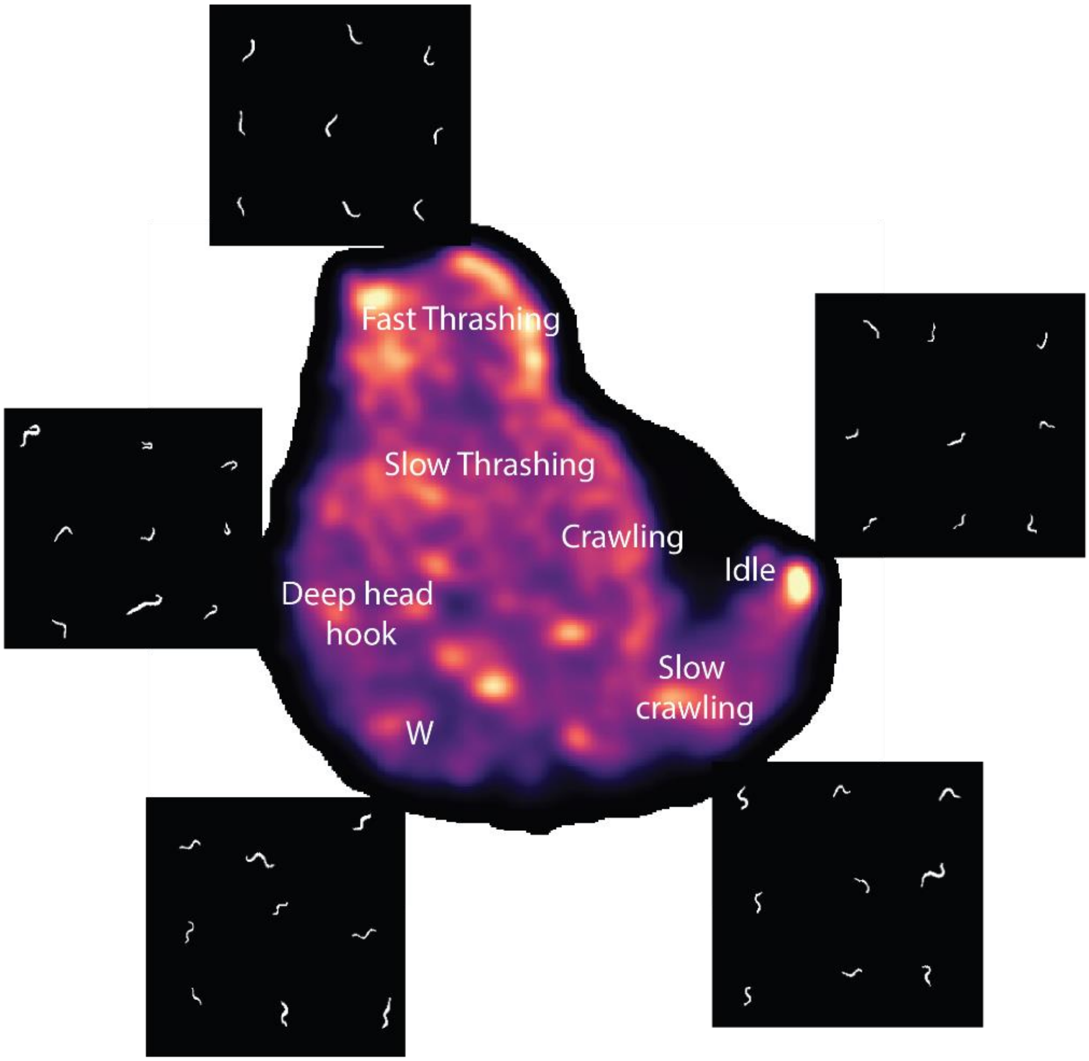


Figure 4-8 Behavior map interpretation. PDF annotated with descriptors for several regions. Single segmented frames of various animals as they are behaving in each region are located near annotations.

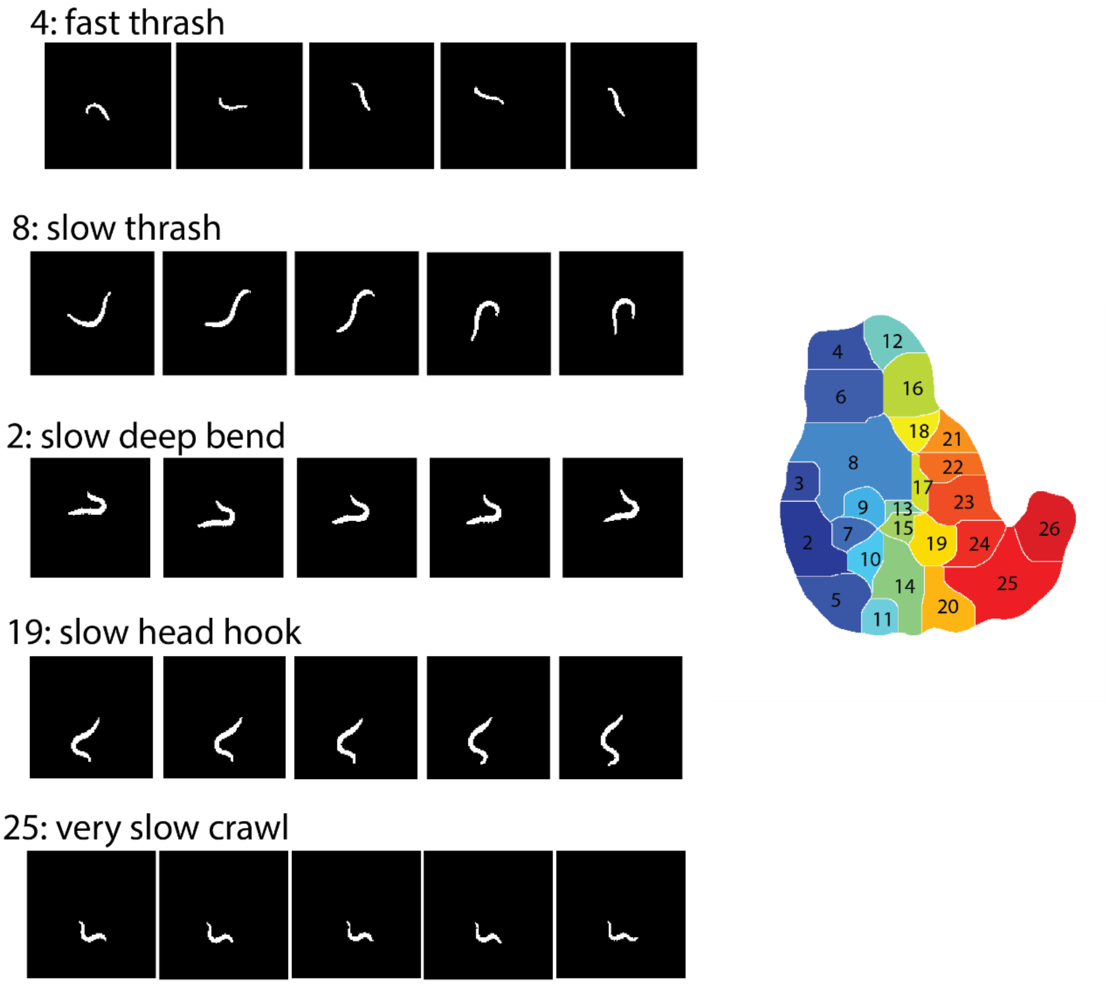


Figure 4-9 Example animal behaviors from select behavior regions. Frames from individuals in selected regions over 0.66 seconds.

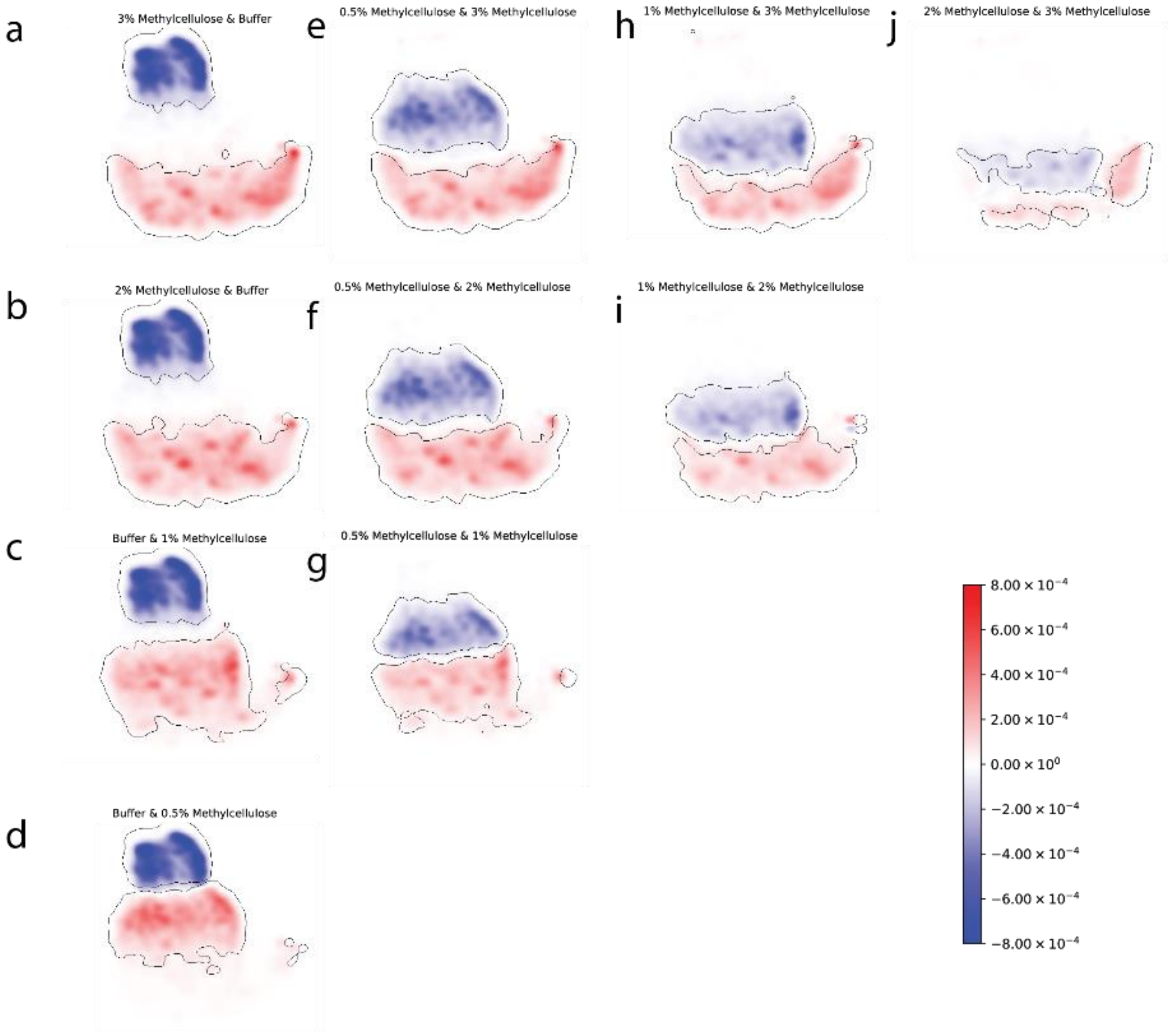


Figure 4-10 Behavior map environmental differences. Each map compares two environments, as noted above each map. The colormap indicates the density difference between each pair of environments, where the less viscous environment is always in blue. Overlaid in black are contours of regions determined to be significant ($p < 0.05$) via the hierarchical bootstrapping method described above.

4.4.5 Transitional behavior

Finally, we wondered whether transitions between behavior regions would present interesting patterns, and over what scale these patterns might decay to the underlying probability of being in a particular end state given the start state. We found where transitions occurred between behavior regions and determined the probability of transitioning from any behavior to any other behavior at increasing transition (τ) steps (**Figure 4-11**). The strong diagonal structure of the even transition matrices shows that most animals consistently transition back to their starting behavior after transitions out of that behavior. This strong preference decays somewhat at higher tau, although several behaviors remain very dominant. At 20 transitions from the original behavior, the transition probabilities have reached a steady state, and along the diagonal regions 4, 6, 8, and 14 remain dominant. Regions 4 and 6 are both thrashing behavior in buffer media, 8 is a slower thrashing in 0.5% methylcellulose, and 14 is a very slow crawl in 2% and 3% methylcellulose. This supports our hypothesis that *C. elegans* behavior is richer in 1% methylcellulose.

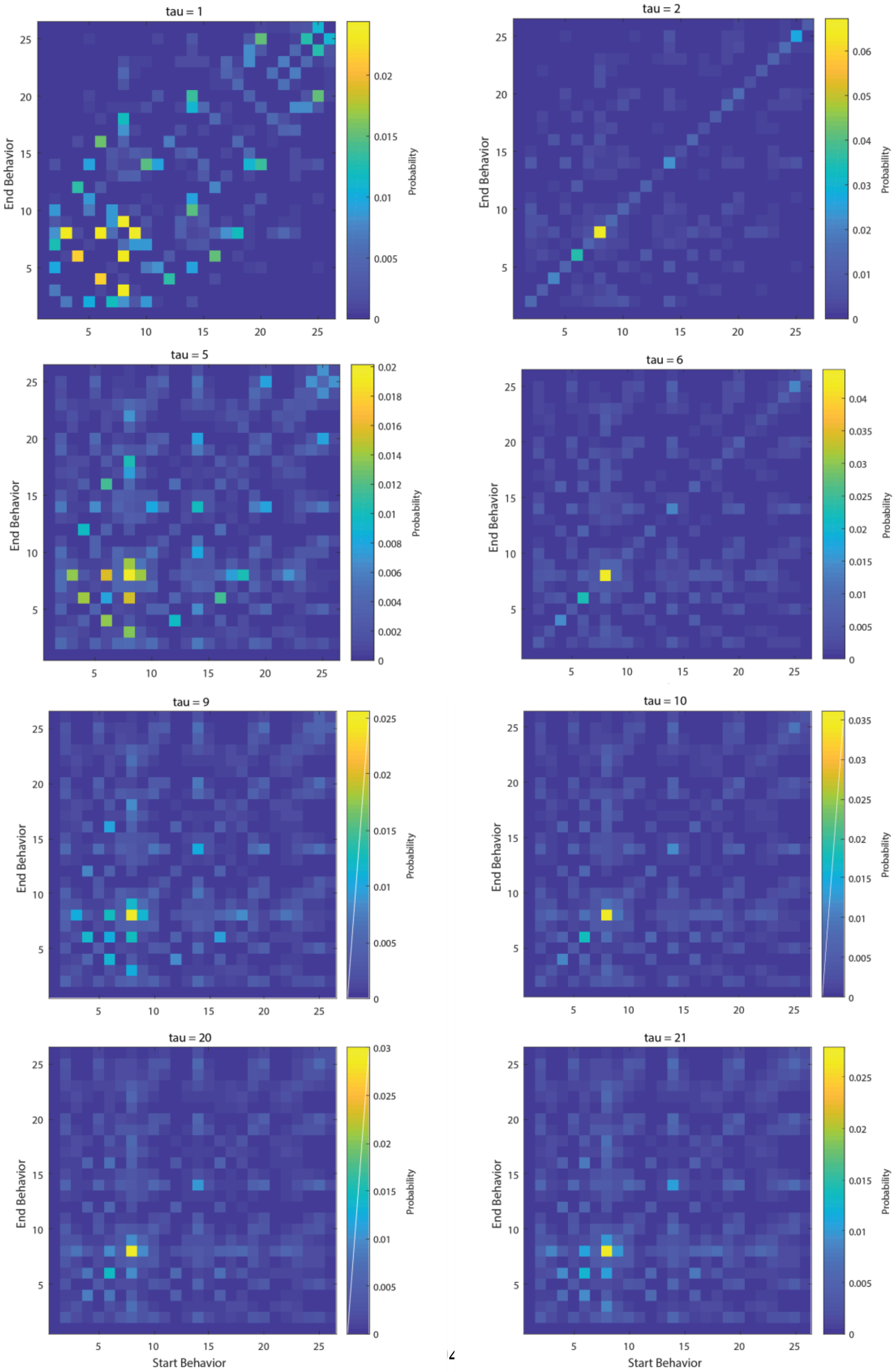

tau = 1
tau = 2
tau = 5
tau = 6
tau = 9
tau = 10
tau = 20
tau = 21
End Behavior
Start Behavior
Probability
5
10
15
20
25
0
0.005
0.01
0.015
0.02
0.025
0.03
0.035
0.04
0.05
0.06
0.002
0.004
0.006
0.008
0.012
0.014
0.016
0.018

Figure 4-11 Transitions between unified behaviors. For selected transitions between 1 and 21, the probability of any end behavior following a given start behavior.

4.5 Discussion

Our results here suggest that some physical environments enhance the diversity of *C. elegans* behavior. This finding has important consequences for understanding the role of genetics in behavior. It is well-known that there is a behavioral phenotyping gap for *C. elegans* – 85% of genes knocked down by RNAi have no observable behavioral phenotype, yet most reduce fitness over generations [120]. Despite increasingly high-content and precise methods for describing behavior in *C. elegans,* this gap, although slightly reduced, remains. The results we show here suggest that one possible cause of this gap is the ability to effectively elicit differential behavioral phenotypes.

4.5.1 Transitional environments

While here we only consider behavioral diversity of animals exposed to a single environment over the entire experiment, understanding how animal behavior preferences change as they navigate through transitions in their physical environment. Previous studies on thigmotaxis and durotaxis in *C. elegans* indicate that a navigation strategy is likely. While it is not so straightforward to construct a viscosity gradient in a highly defined way, a microfluidic approach could be used to alter the density of pillar spacing in a 'soil-like' microfluidic device to examine how worm navigation strategy and behavior preferences change at transition points (for example, microfluidic devices such as those in **Figure 4-12)**. The use of microfluidic devices to construct hydrogels with gradients in mechanical properties such as elastic modulus has also been demonstrated as a tool for understanding durotaxis in cells [121]. These tools could be useful for understanding *C. elegans* behavior as they move through gradients like those they may experience in their natural habitats.

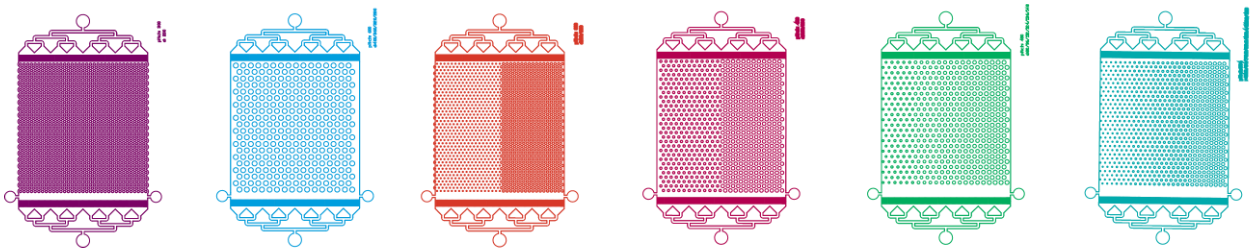


Figure 4-12 Transition and gradient microfluidic devices.

4.5.2 Occluded posture prediction from annotated grayscale images

While here we concentrated on understanding the behavior of worms without considering self-occluding postures, our own data as well as data from other large-scale behavior assays suggest that 7% of the worms total behavior (in terms of time) is composed of turns. While the timescale necessary to apply the consensus annotation method discussed in Chapter 3 was intractable to apply to all frames in this dataset, there are several modifications to the annotation approach that may allow us to elucidate posture for self-occluding frames. Instead of a consensus approach, where a single contour is eventually produced from many annotations from potentially non-expert users, we could instead have only experts annotate. This somewhat negates the power of being able to distribute annotation effectively, however, sampling more sparsely from continuous stretches of self-occluding frames may be sufficient to interpolate worm posture between each annotated frame. This would likely still be faster and less error-prone than applying either the alternative generative method described in Broekmans et al., 2016 or using heuristic-based methods previously described for animals behaving on agarose media [8,10,53].

CHAPTER 5. CONCLUSIONS AND FUTURE WORK

5.1 Thesis Contributions

The objective of this thesis was to build platforms that enable scalable behavioral phenotyping of *C. elegans*. I aimed to use computer vision, machine learning, and microscopy tools to create a combination of systems that would enable new experimental capabilities. This advances the field of *C. elegans* neuroethology by making previously intractable biological questions possible to address through new types of experiments, new annotation tools, and methodology. Several of these tools may be translatable to other organismal systems.

In chapter two, I developed a scalable microscopy platform for collecting long-term behavior data and for conditionally intervening in *C. elegans* development. This tool is easy to build and use, with a touch-based interface and days-long automated operation. I demonstrate that it can be used for behavioral phenotyping of individual animals from larval stages through adult stage, and that it can track motion online using computer vision techniques. Finally, I demonstrated that it can be used to intervene on animals conditional upon their behavior. This allowed us to create an extreme sleep deprivation model in *C. elegans* and test the health outcomes of sleep-deprived animals. Mi-pi enables a wide range of biologically impactful *C. elegans* experiments. For example, it could be used to better understand the role of neuropeptides in lethargus homeostasis and behavioral changes throughout aging or development. This tool could also be adapted for use with other small model systems by exchanging imaging lenses to monitor larger or smaller animals and adapting the neural networks used for detecting animals online. It may also be useful for monitoring social interactions both intra- and interspecies.

In chapter three, I developed a fast and reliable tool for annotating complex images. Annotating images is critical for developing ground truths that can be used for validating automated methods, but few freehand annotation tools exist. We built a smartphone app to more easily distribute annotations and enable freehand drawing of annotations. We demonstrated annotation of stem cell and plant morphology as well as annotation of complex *C. elegans* posture. By crowd-sourcing posture annotations, we were able to produce consensus annotations of *C. elegans* posture that outperformed the state-of-the-art algorithm both in terms of time (about 130-fold improvement) and in cases of rare or hard-to-predict postures. This tool is agnostic to the content of the images, and it is especially well-suited for freehand annotation of ambiguous images. We believe it could be used to equal effect for building ground truth datasets for any number of applications in addition to the applications of posture, plant morphology, and stem cell aggregate morphology we demonstrated in chapter three.

In chapter four, I develop machine learning tools for describing *C. elegans* behavior and use them to compare *C. elegans* behavior in a variety of media. I used this approach to generate a map of *C. elegans* behavior in a range of environments that are diverse in material properties, similar to what *C. elegans* may encounter in its natural environment. From this analysis, we identified subtle behaviors that are not easily distinguishable by eye.

We also built an interactive data exploration tool to facilitate understanding of our results that is publicly accessible. Finally, we found that the diversity of *C. elegans* behavior was not uniform across the environments we sampled, indicating that there is an optimum media that may increase our ability to distinguish behavioral phenotypes in *C. elegans*. The machine learning tool adapted in this work may be useful for phenotyping *C. elegans* behavior throughout aging and developmental processes, where subtle behavioral shifts may not be apparent using conventional techniques. It may also be useful for understanding the behavioral consequences of complex behavior interventions, such as those that the mi-pi system enables.

The behavioral phenotyping tools developed in this thesis were designed to enable novel biological experiments and answer specific biological questions about the complexity of *C. elegans'* behavior and how behavioral outputs can be influenced by external factors. These platforms have provided insight into how behavior and health is influenced by perturbations in animal environment, and we foresee the application of these tools in a broad range of behavioral questions, both within the *C. elegans* research community and outside of it.

5.2 Future Directions

5.2.1 Scalable microscopy for optogenetics and online segmentation

The scalable microscopy platform I developed was used in this thesis to perturb animal environment through light. There are several natural extensions to this platform that would allow other types of interventions contingent on animal behavior. High-powered LEDs have been demonstrated in other works to scale optogenetic perturbation [122]. This could be used to more directly interfere with worm quiescence by activating neurons that are sleep-active. An optogenetic perturbation method in combination with online object detection could also enable novel and complex learning paradigms that may otherwise be intractably low-throughput. For example, food avoidance as a result of pathogenic learning has been demonstrated in *C. elegans,* but whether *C. elegans* could learn the same avoidance based on other stimuli it encounters proximate to the food is not known[31]. Using the mi-pi platform with optogenetics, it would be possible to test whether *C. elegans* can learn food avoidance from alternative stimuli by tracking the worms location in relationship to food location, which can be annotated using the touchscreen app. If *C. elegans* can learn food avoidance as a result of aversive stimuli applied only when animals are on a specific food source, it would suggest a degree of sensory integration previously unknown in *C. elegans*. We are currently working on developing such an assay.

I demonstrated the potential for real-time interventions using this platform through object detection neural networks. This level of real-time processing could be furthered in several ways that could be useful both for online behavior classification as well as online data compression. One important advancement would be semantic segmentation of worms and bacterial lawn online. At present, we are working on implementing this and think it reasonably probable that fast semantic segmentation methods may be implemented online or almost online in series with object detection methods. With segmentation capabilities in addition to object detection, we could potentially track individual’s velocity online and classify behavior into a few broad categories of behavior based on characteristics of the segmented body, for example turns, forward or backward movement, or pauses. Performing these computations online would not only enable new and more complex types of experiments, it would also ease the significant burden of video processing that remains a bottleneck in analysis.

5.2.2 Behavior mapping for occluded postures, and navigational behavioral phenotypes

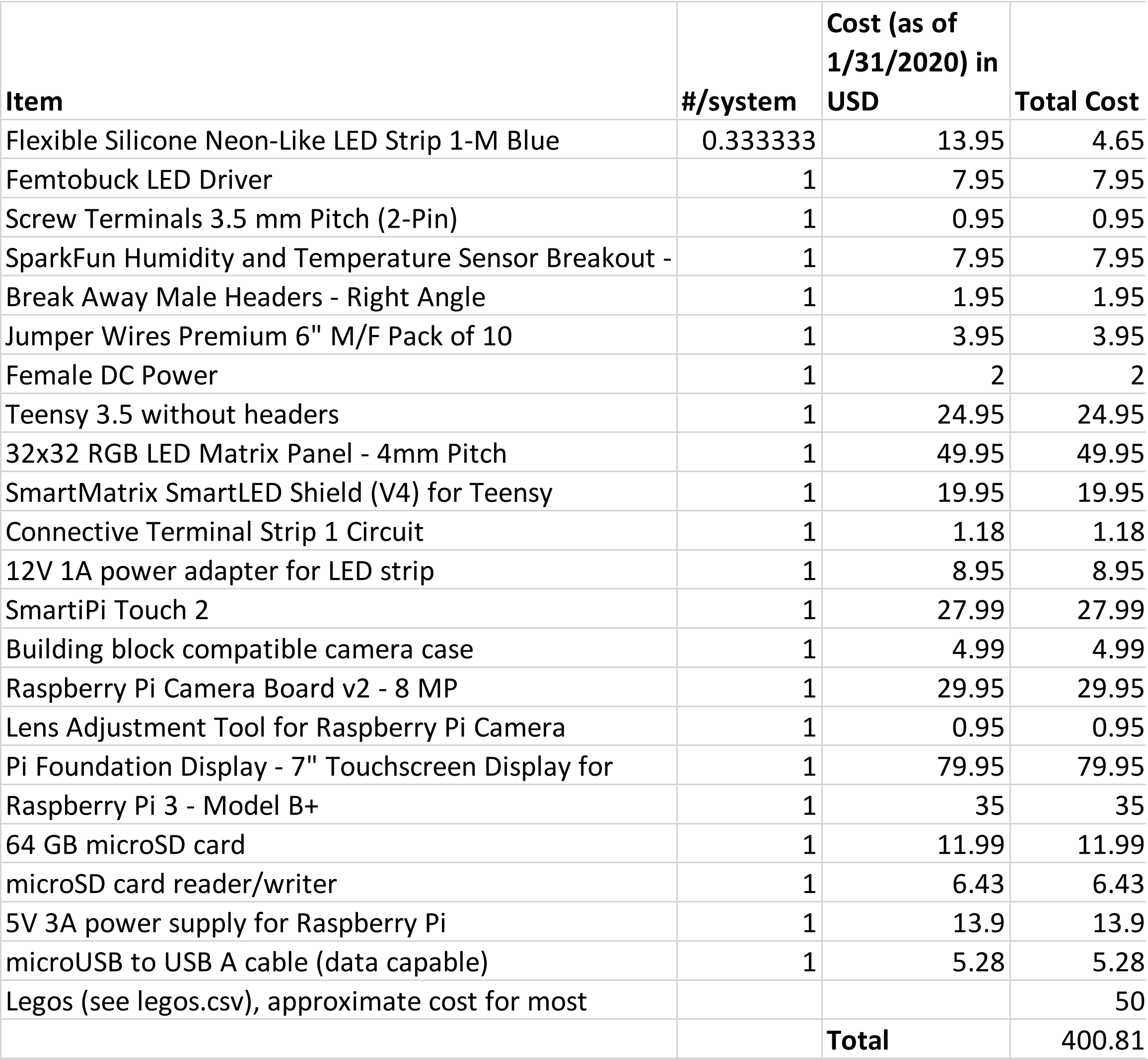
In this thesis, I used an machine-learning approach to compare non-occluded *C. elegans* behavior in environments with different mechanical properties. Combining analysis of complex postures in this dataset alongside non-occluded postures could answer questions about the stereotypy and prevalence of turning and coiling behaviors across environments. At present, a small portion of our occluded behavior dataset from agarose and 0.5% methylcellulose behavior has been annotated, and we intend to combine additional annotation of the rest of our dataset with other predictive techniques to better understand the diversity in dynamics of these complex behaviors and improve phenotyping. In addition, we have collected some data from a more expansive range of microfluidic environments in which worms can explore well-defined physical gradients like pillar density. We plan to use this to study navigational tactics through physical environments in *C. elegans* and compare these tactics to well-described navigational tactics in response to thermal and chemical gradients.

APPENDIX A. SUPPLEMENTARY MI-PI DOCUMENTATION

This appendix describes hardware and software setup of mi-pi microscopy systems.

A.1 Bill of Materials

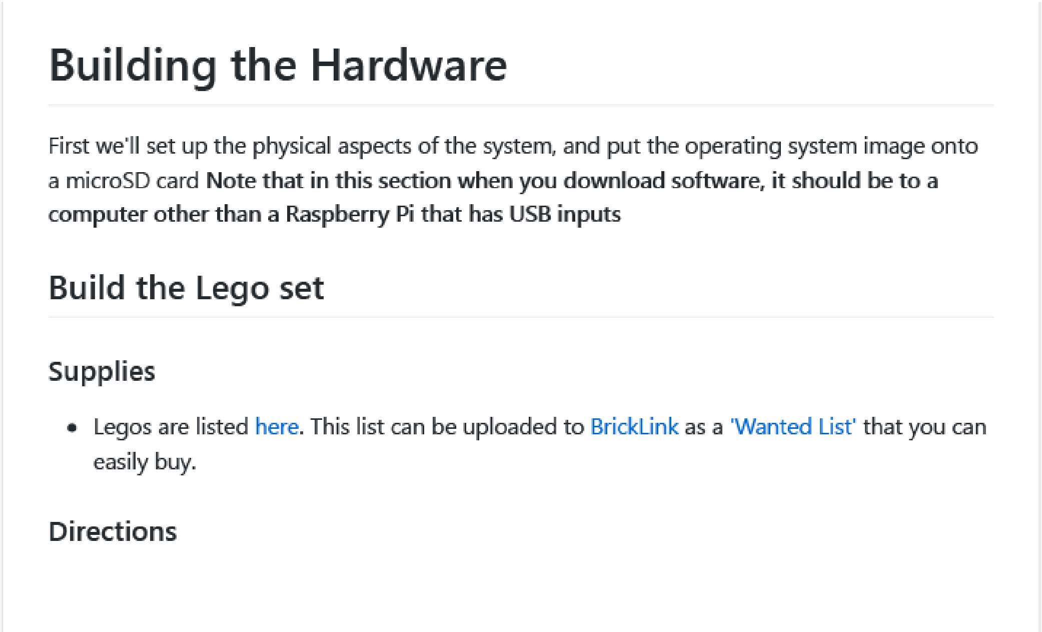
Table 1. Mi-pi Bill of Materials



Item	#/system	Cost (as of 1/31/2020) in USD	Total Cost
Flexible Silicone Neon-Like LED Strip 1-M Blue	0.333333	13.95	4.65
Femtobuck LED Driver	1	7.95	7.95
Screw Terminals 3.5 mm Pitch (2-Pin)	1	0.95	0.95
SparkFun Humidity and Temperature Sensor Breakout -	1	7.95	7.95
Break Away Male Headers - Right Angle	1	1.95	1.95
Jumper Wires Premium 6" M/F Pack of 10	1	3.95	3.95
Female DC Power	1	2	2
Teensy 3.5 without headers	1	24.95	24.95
32x32 RGB LED Matrix Panel - 4mm Pitch	1	49.95	49.95
SmartMatrix SmartLED Shield (V4) for Teensy	1	19.95	19.95
Connective Terminal Strip 1 Circuit	1	1.18	1.18
12V 1A power adapter for LED strip	1	8.95	8.95
SmartiPi Touch 2	1	27.99	27.99
Building block compatible camera case	1	4.99	4.99
Raspberry Pi Camera Board v2 - 8 MP	1	29.95	29.95
Lens Adjustment Tool for Raspberry Pi Camera	1	0.95	0.95
Pi Foundation Display - 7" Touchscreen Display for	1	79.95	79.95
Raspberry Pi 3 - Model B+	1	35	35
64 GB microSD card	1	11.99	11.99
microSD card reader/writer	1	6.43	6.43
5V 3A power supply for Raspberry Pi	1	13.9	13.9
microUSB to USB A cable (data capable)	1	5.28	5.28
Legos (see legos.csv), approximate cost for most			50
		Total	400.81

A.2 Hardware setup

This section describes setup of the hardware, and may also be found at https://github.com/lu-lab/mi-pi/blob/master/docs/hardware.md



Building the Hardware

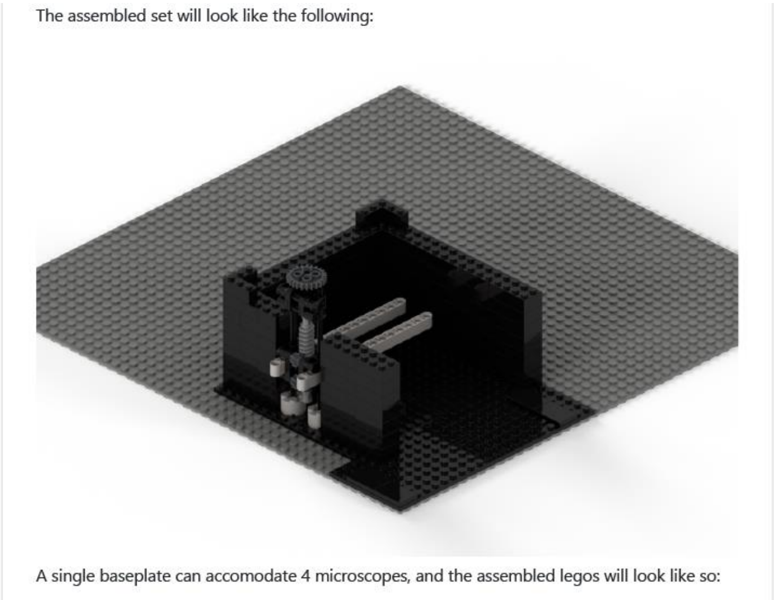
First we'll set up the physical aspects of the system, and put the operating system image onto a microSD card **Note that in this section when you download software, it should be to a computer other than a Raspberry Pi that has USB inputs**

Build the Lego set

Supplies

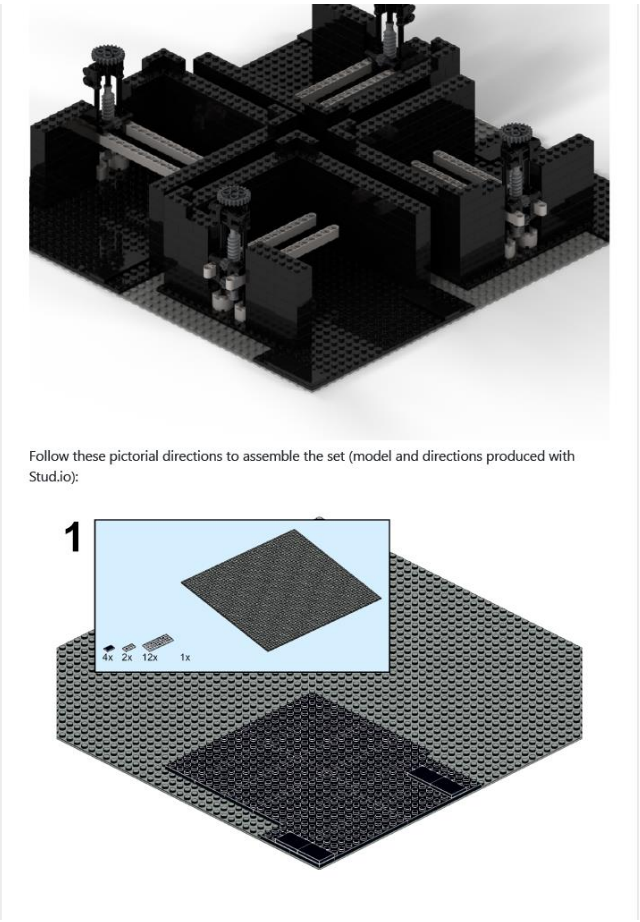
- Legos are listed here. This list can be uploaded to BrickLink as a 'Wanted List' that you can easily buy.

Directions



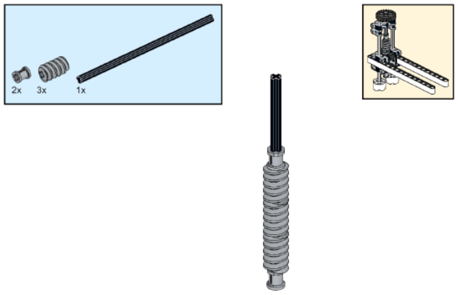
The assembled set will look like the following:

A single baseplate can accomodate 4 microscopes, and the assembled legos will look like so:

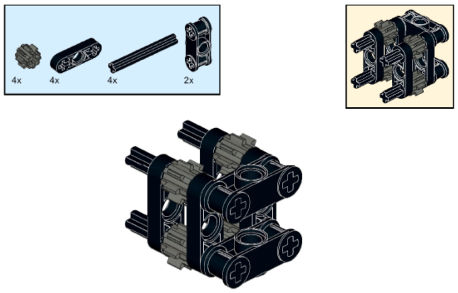


Follow these pictorial directions to assemble the set (model and directions produced with Stud.io):

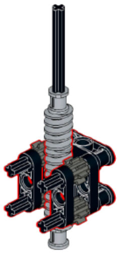
2



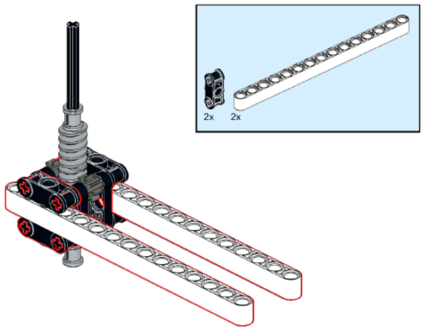
3



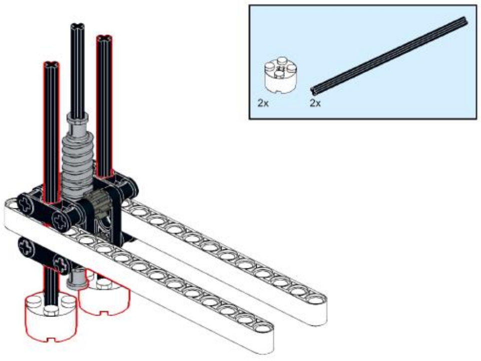
4



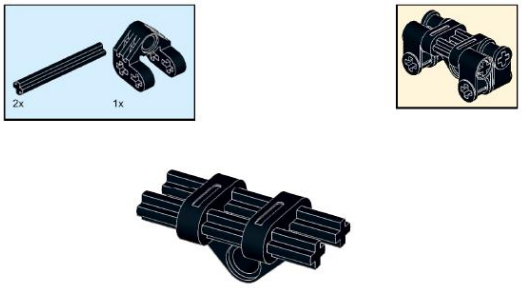
5



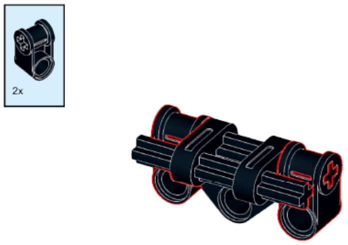
6



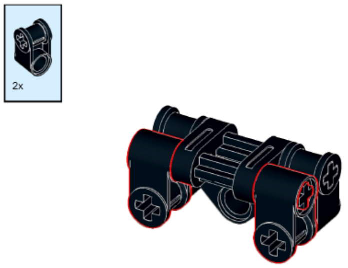
7



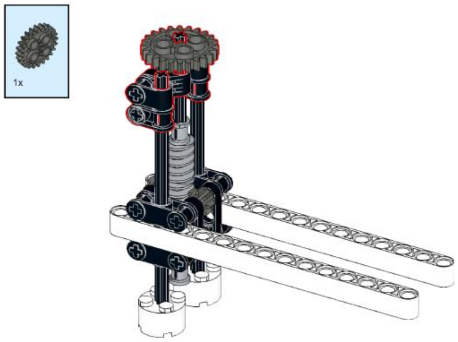
8



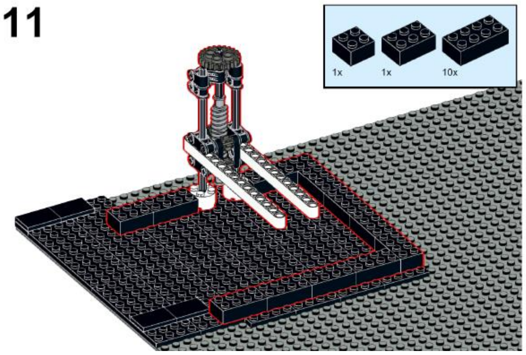
9



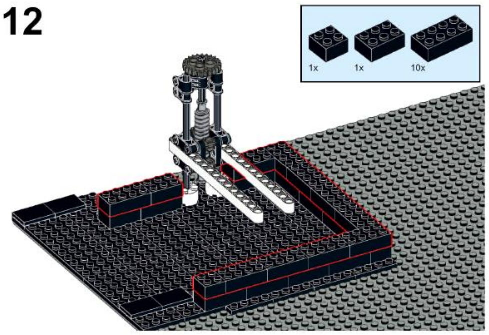
10



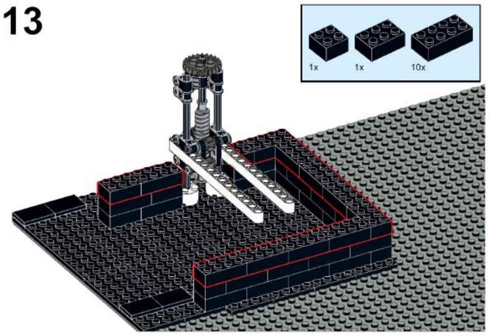
11



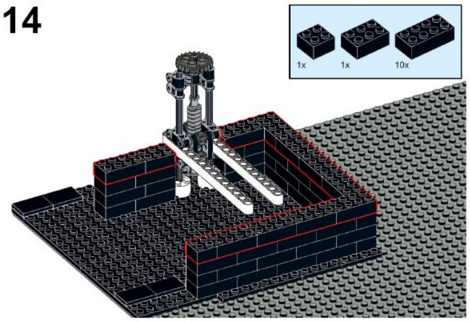
12



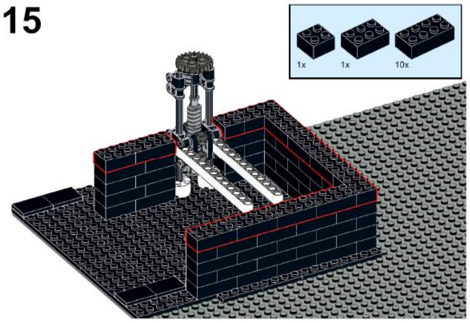
13



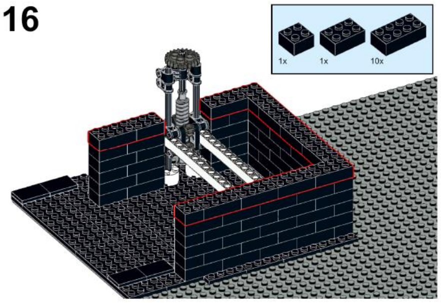
14



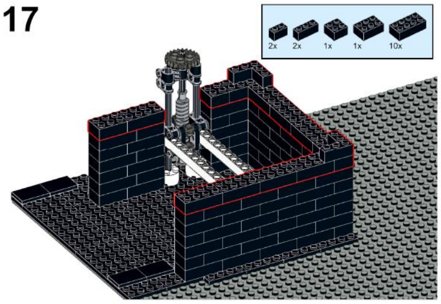
15



16

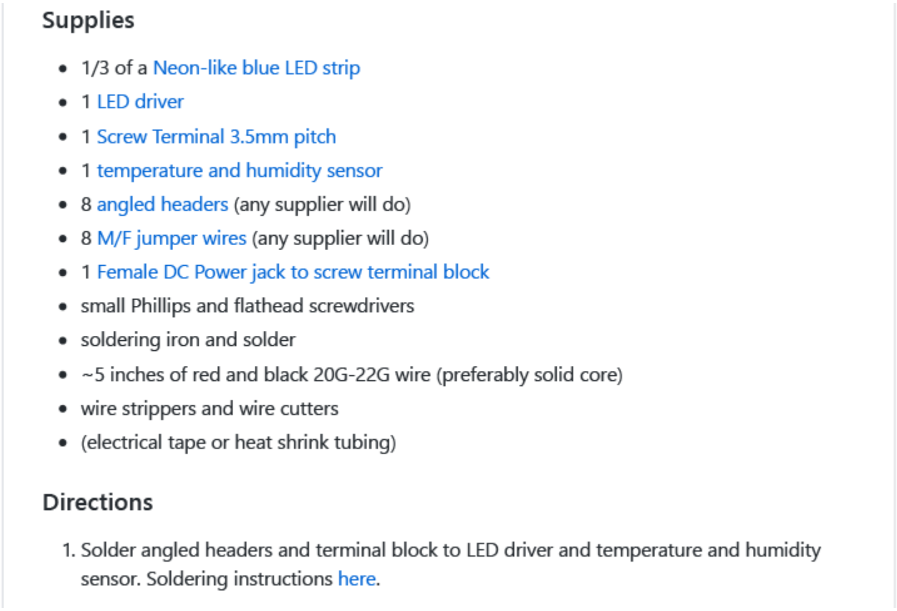


17



\

Assemble the temperature sensor and bright blue LED strip

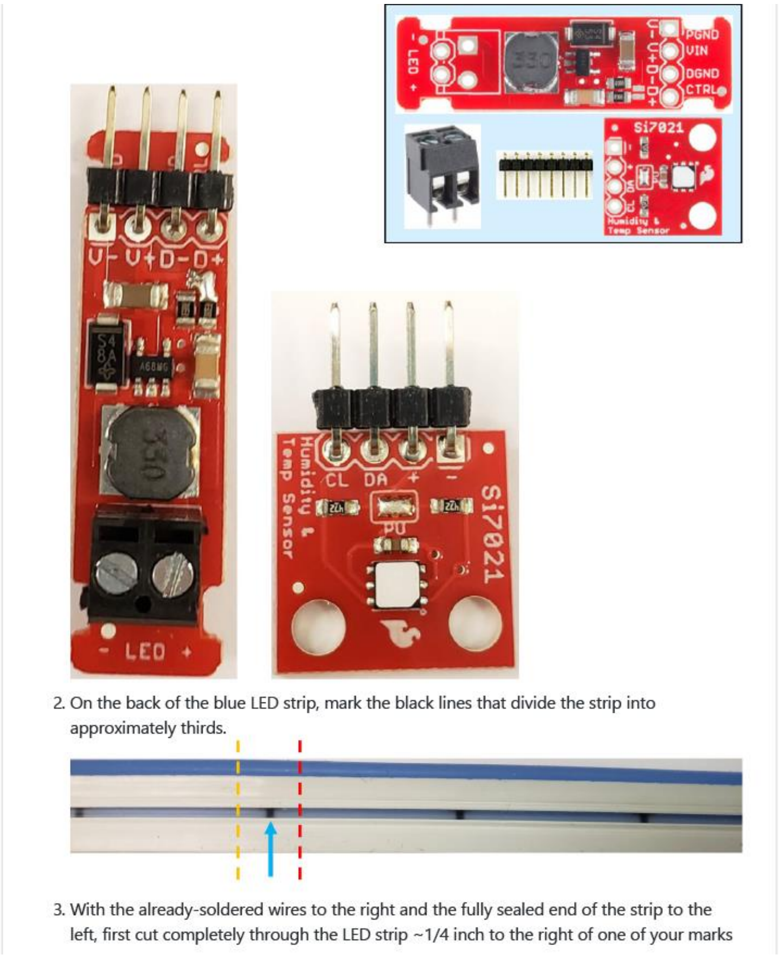


Supplies

- 1/3 of a Neon-like blue LED strip
- 1 LED driver
- 1 Screw Terminal 3.5mm pitch
- 1 temperature and humidity sensor
- 8 angled headers (any supplier will do)
- 8 M/F jumper wires (any supplier will do)
- 1 Female DC Power jack to screw terminal block
- small Phillips and flathead screwdrivers
- soldering iron and solder
- ~5 inches of red and black 20G-22G wire (preferably solid core)
- wire strippers and wire cutters
- (electrical tape or heat shrink tubing)

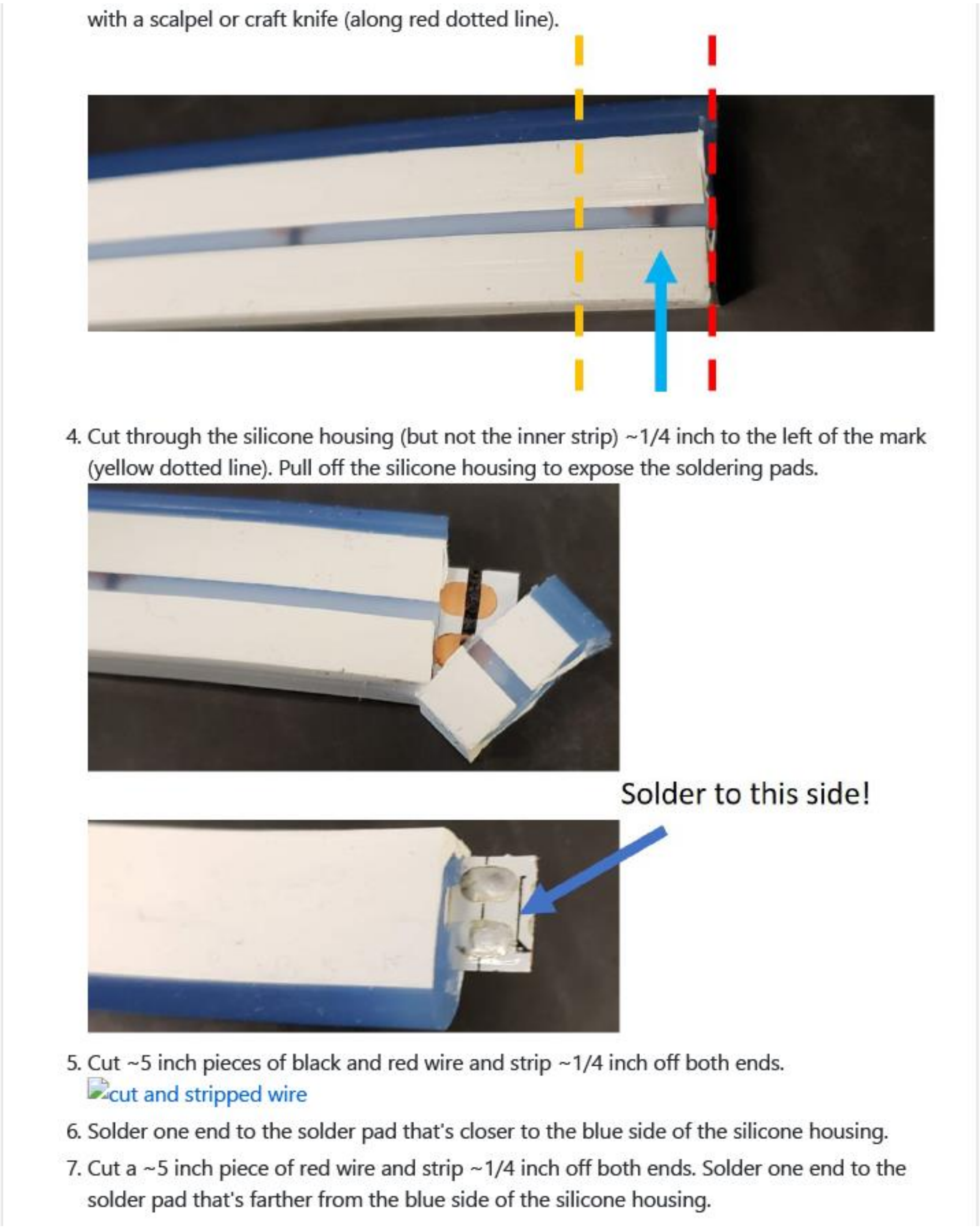
Directions

1. Solder angled headers and terminal block to LED driver and temperature and humidity sensor. Soldering instructions here.



2. On the back of the blue LED strip, mark the black lines that divide the strip into approximately thirds.

3. With the already-soldered wires to the right and the fully sealed end of the strip to the left, first cut completely through the LED strip ~1/4 inch to the right of one of your marks

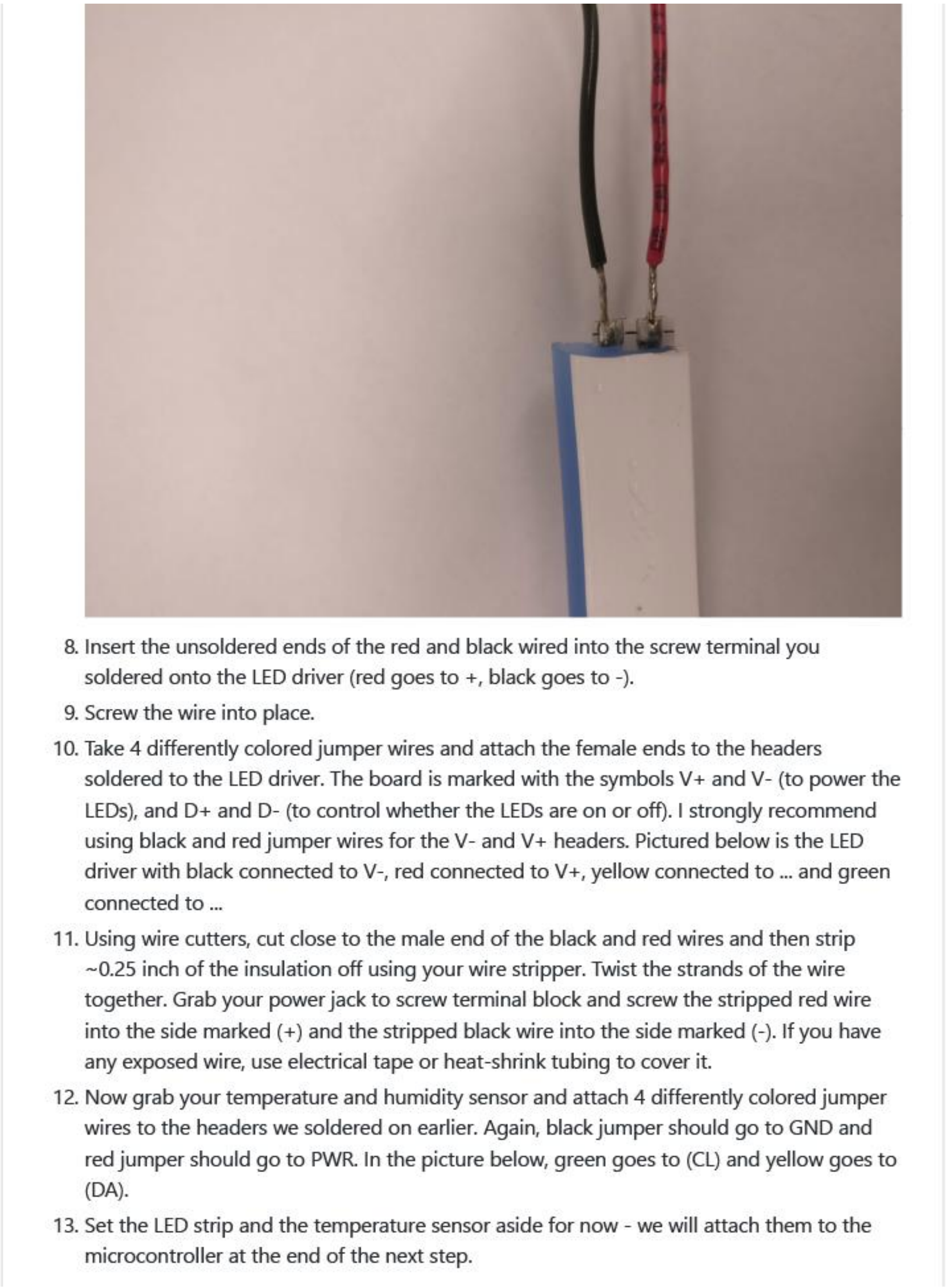


with a scalpel or craft knife (along red dotted line).

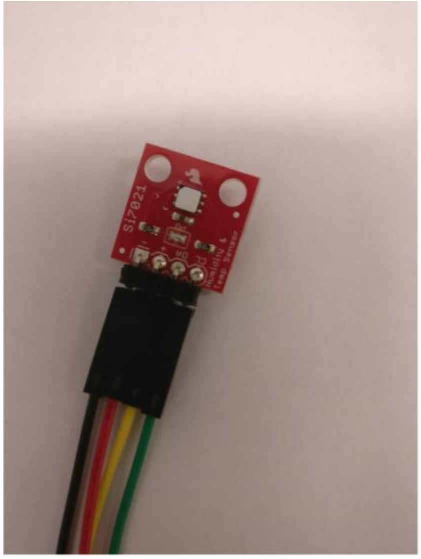
4. Cut through the silicone housing (but not the inner strip) ~1/4 inch to the left of the mark (yellow dotted line). Pull off the silicone housing to expose the soldering pads.

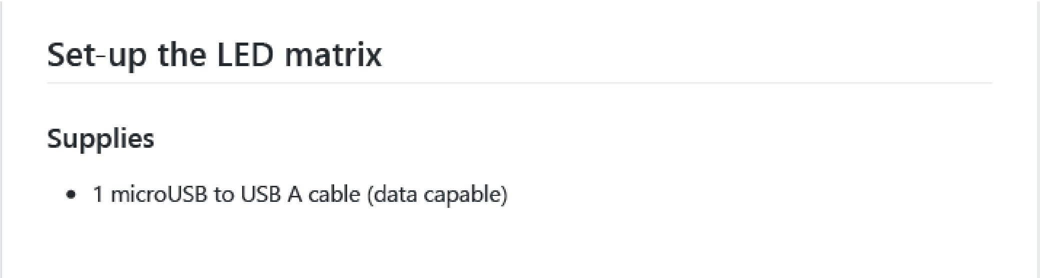
Solder to this side!

5. Cut ~5 inch pieces of black and red wire and strip ~1/4 inch off both ends.
   cut and stripped wire
6. Solder one end to the solder pad that's closer to the blue side of the silicone housing.
7. Cut a ~5 inch piece of red wire and strip ~1/4 inch off both ends. Solder one end to the solder pad that's farther from the blue side of the silicone housing.



8. Insert the unsoldered ends of the red and black wired into the screw terminal you soldered onto the LED driver (red goes to +, black goes to -).
9. Screw the wire into place.
10. Take 4 differently colored jumper wires and attach the female ends to the headers soldered to the LED driver. The board is marked with the symbols V+ and V- (to power the LEDs), and D+ and D- (to control whether the LEDs are on or off). I strongly recommend using black and red jumper wires for the V- and V+ headers. Pictured below is the LED driver with black connected to V-, red connected to V+, yellow connected to ... and green connected to ...
11. Using wire cutters, cut close to the male end of the black and red wires and then strip ~0.25 inch of the insulation off using your wire stripper. Twist the strands of the wire together. Grab your power jack to screw terminal block and screw the stripped red wire into the side marked (+) and the stripped black wire into the side marked (-). If you have any exposed wire, use electrical tape or heat-shrink tubing to cover it.
12. Now grab your temperature and humidity sensor and attach 4 differently colored jumper wires to the headers we soldered on earlier. Again, black jumper should go to GND and red jumper should go to PWR. In the picture below, green goes to (CL) and yellow goes to (DA).
13. Set the LED strip and the temperature sensor aside for now - we will attach them to the microcontroller at the end of the next step.

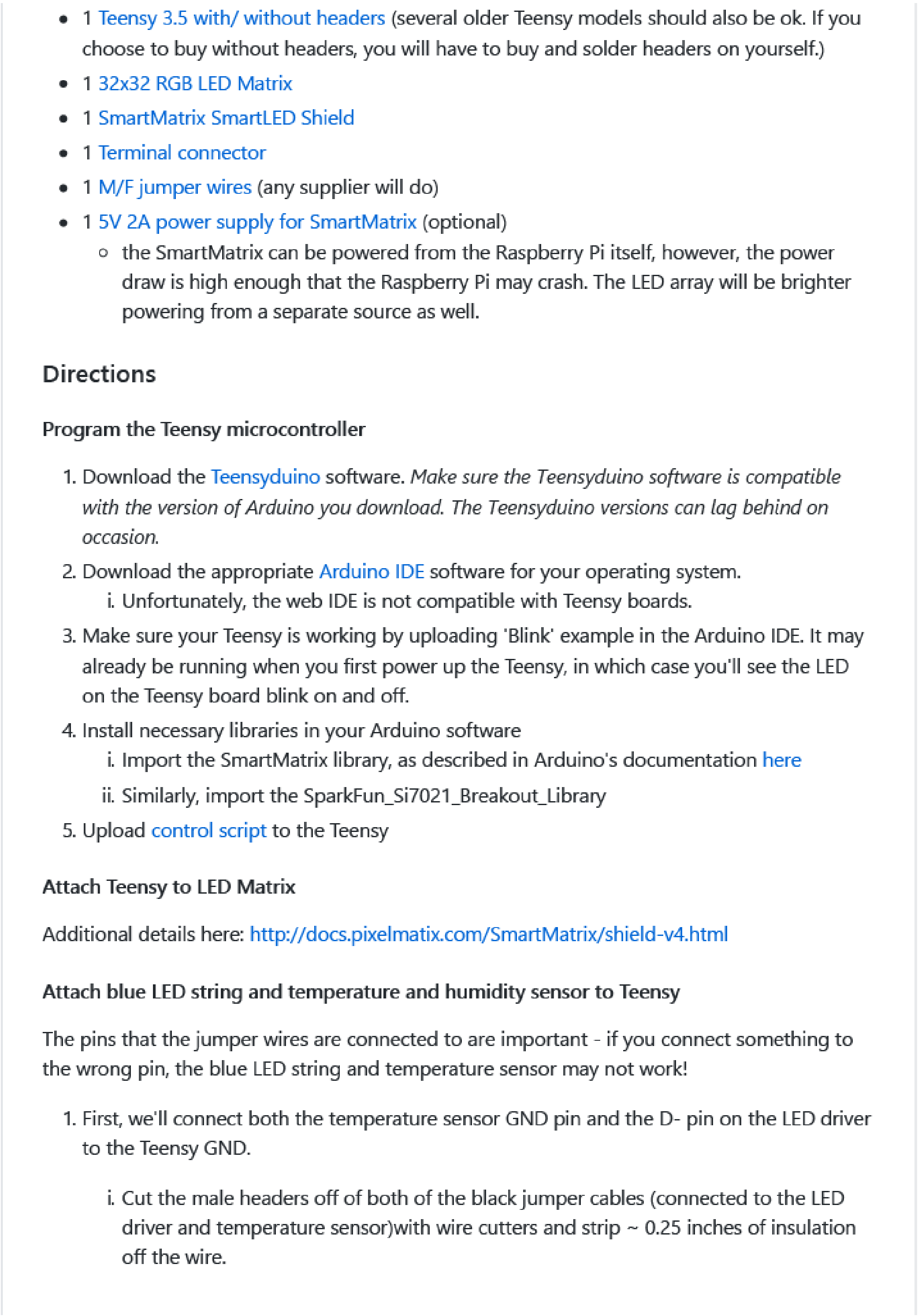




Set-up the LED matrix

Supplies

- 1 microUSB to USB A cable (data capable)



- 1 Teensy 3.5 with/ without headers (several older Teensy models should also be ok. If you choose to buy without headers, you will have to buy and solder headers on yourself.)
- 1 32x32 RGB LED Matrix
- 1 SmartMatrix SmartLED Shield
- 1 Terminal connector
- 1 M/F jumper wires (any supplier will do)
- 1 5V 2A power supply for SmartMatrix (optional)
  - the SmartMatrix can be powered from the Raspberry Pi itself, however, the power draw is high enough that the Raspberry Pi may crash. The LED array will be brighter powering from a separate source as well.

Directions

Program the Teensy microcontroller

1. Download the Teensyduino software. *Make sure the Teensyduino software is compatible with the version of Arduino you download. The Teensyduino versions can lag behind on occasion.*
2. Download the appropriate Arduino IDE software for your operating system.
   i. Unfortunately, the web IDE is not compatible with Teensy boards.
3. Make sure your Teensy is working by uploading 'Blink' example in the Arduino IDE. It may already be running when you first power up the Teensy, in which case you'll see the LED on the Teensy board blink on and off.
4. Install necessary libraries in your Arduino software
   i. Import the SmartMatrix library, as described in Arduino's documentation here
   ii. Similarly, import the SparkFun_Si7021_Breakout_Library
5. Upload control script to the Teensy

Attach Teensy to LED Matrix

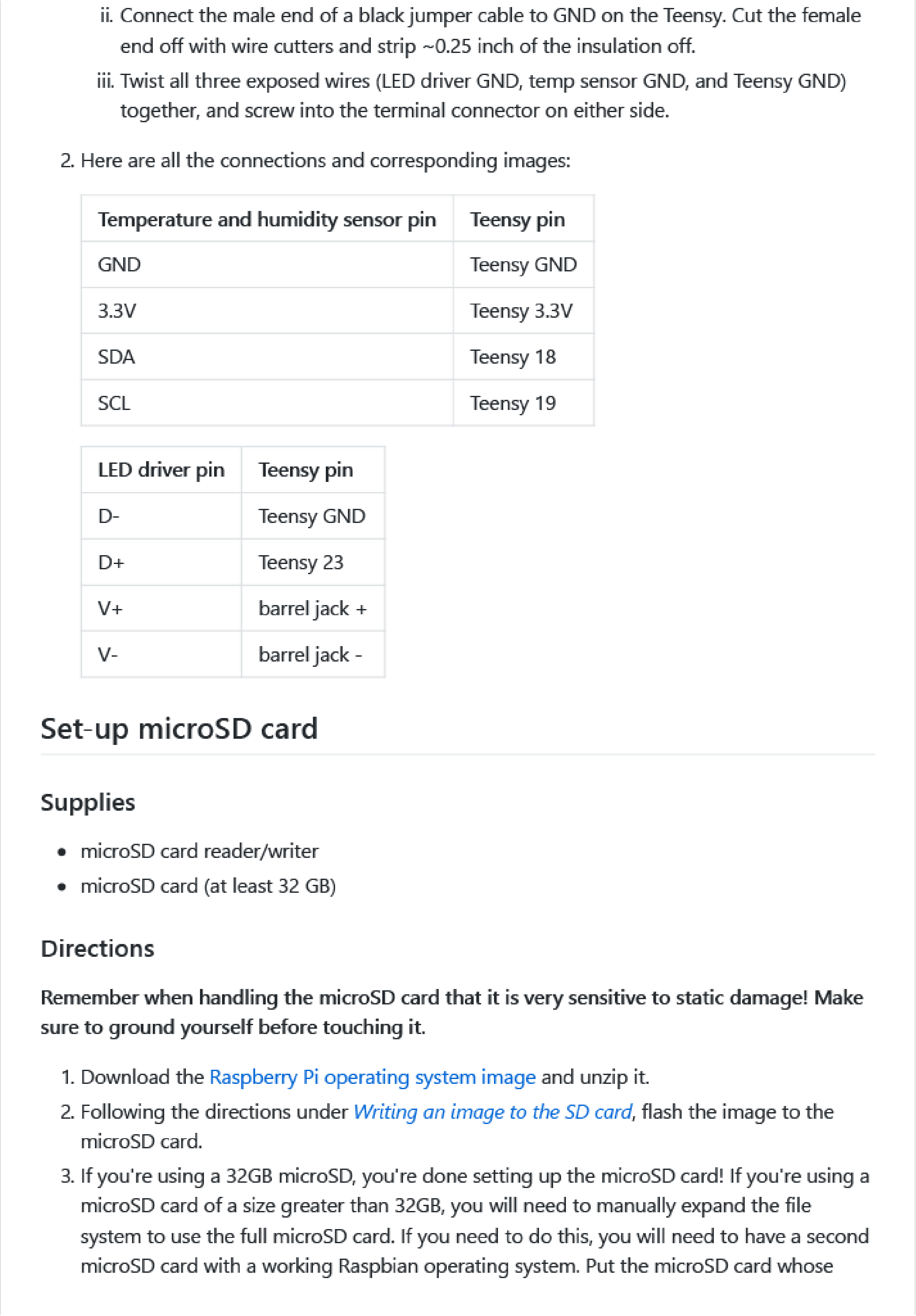
Additional details here: http://docs.pixelmatix.com/SmartMatrix/shield-v4.html

Attach blue LED string and temperature and humidity sensor to Teensy

The pins that the jumper wires are connected to are important - if you connect something to the wrong pin, the blue LED string and temperature sensor may not work!

1. First, we'll connect both the temperature sensor GND pin and the D- pin on the LED driver to the Teensy GND.

   i. Cut the male headers off of both of the black jumper cables (connected to the LED driver and temperature sensor)with wire cutters and strip ~ 0.25 inches of insulation off the wire.



   ii. Connect the male end of a black jumper cable to GND on the Teensy. Cut the female end off with wire cutters and strip ~0.25 inch of the insulation off.

   iii. Twist all three exposed wires (LED driver GND, temp sensor GND, and Teensy GND) together, and screw into the terminal connector on either side.

2. Here are all the connections and corresponding images:

Temperature and humidity sensor pin	Teensy pin
GND	Teensy GND
3.3V	Teensy 3.3V
SDA	Teensy 18
SCL	Teensy 19

LED driver pin	Teensy pin
D-	Teensy GND
D+	Teensy 23
V+	barrel jack +
V-	barrel jack -

Set-up microSD card

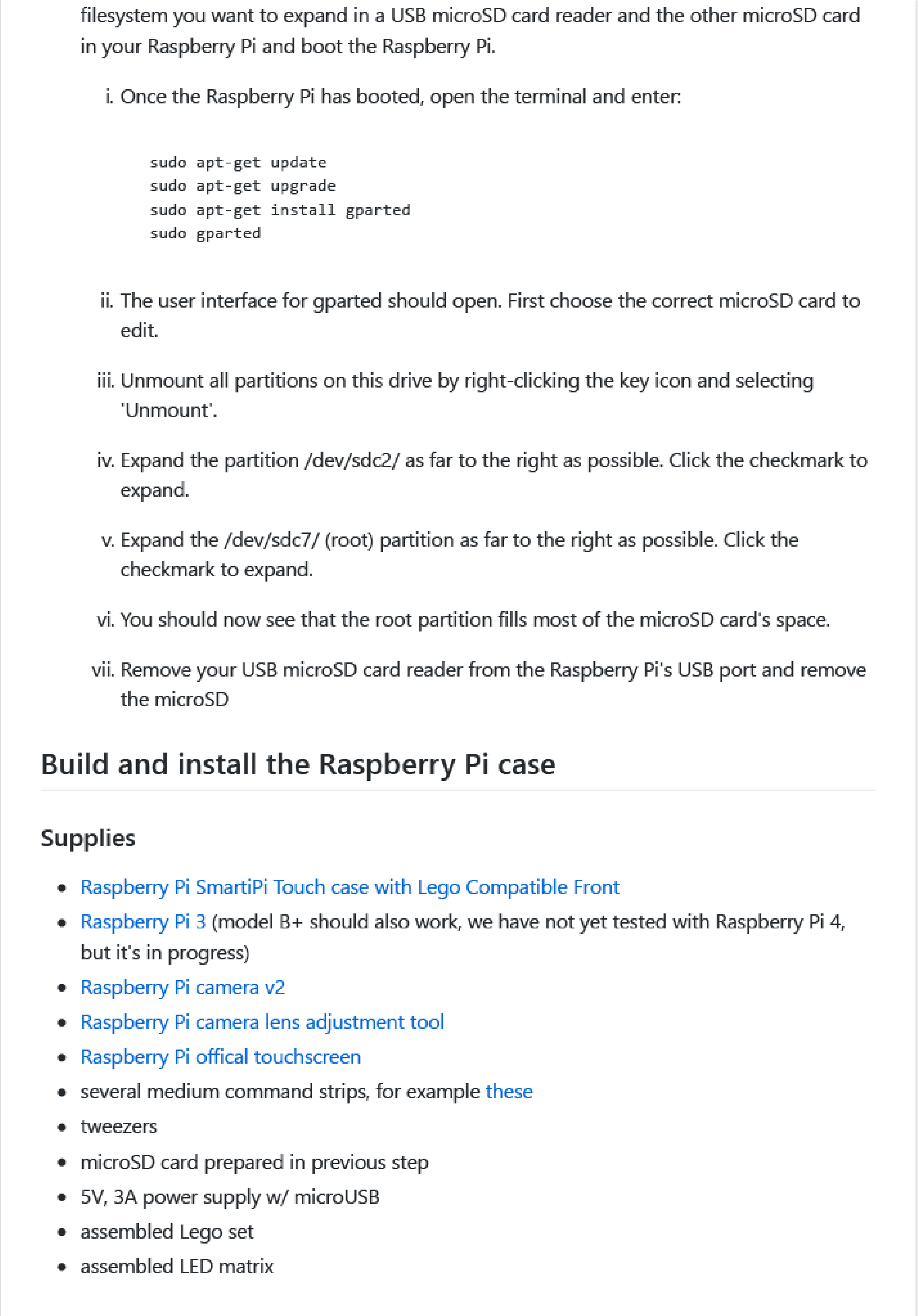
Supplies

- microSD card reader/writer
- microSD card (at least 32 GB)

Directions

Remember when handling the microSD card that it is very sensitive to static damage! Make sure to ground yourself before touching it.

1. Download the Raspberry Pi operating system image and unzip it.
2. Following the directions under *Writing an image to the SD card*, flash the image to the microSD card.
3. If you're using a 32GB microSD, you're done setting up the microSD card! If you're using a microSD card of a size greater than 32GB, you will need to manually expand the file system to use the full microSD card. If you need to do this, you will need to have a second microSD card with a working Raspbian operating system. Put the microSD card whose



filesystem you want to expand in a USB microSD card reader and the other microSD card in your Raspberry Pi and boot the Raspberry Pi.

i. Once the Raspberry Pi has booted, open the terminal and enter:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install gparted
sudo gparted
```

ii. The user interface for gparted should open. First choose the correct microSD card to edit.

iii. Unmount all partitions on this drive by right-clicking the key icon and selecting 'Unmount'.

iv. Expand the partition /dev/sdc2/ as far to the right as possible. Click the checkmark to expand.

v. Expand the /dev/sdc7/ (root) partition as far to the right as possible. Click the checkmark to expand.

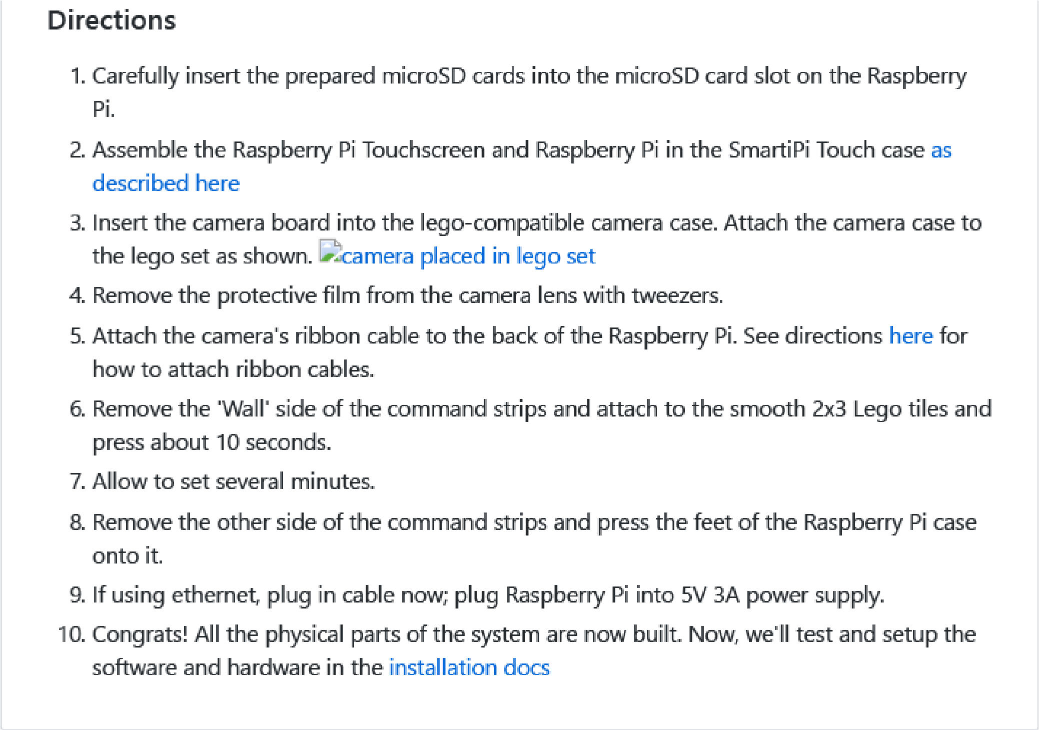
vi. You should now see that the root partition fills most of the microSD card's space.

vii. Remove your USB microSD card reader from the Raspberry Pi's USB port and remove the microSD

Build and install the Raspberry Pi case

Supplies

- Raspberry Pi SmartiPi Touch case with Lego Compatible Front
- Raspberry Pi 3 (model B+ should also work, we have not yet tested with Raspberry Pi 4, but it's in progress)
- Raspberry Pi camera v2
- Raspberry Pi camera lens adjustment tool
- Raspberry Pi offical touchscreen
- several medium command strips, for example these
- tweezers
- microSD card prepared in previous step
- 5V, 3A power supply w/ microUSB
- assembled Lego set
- assembled LED matrix

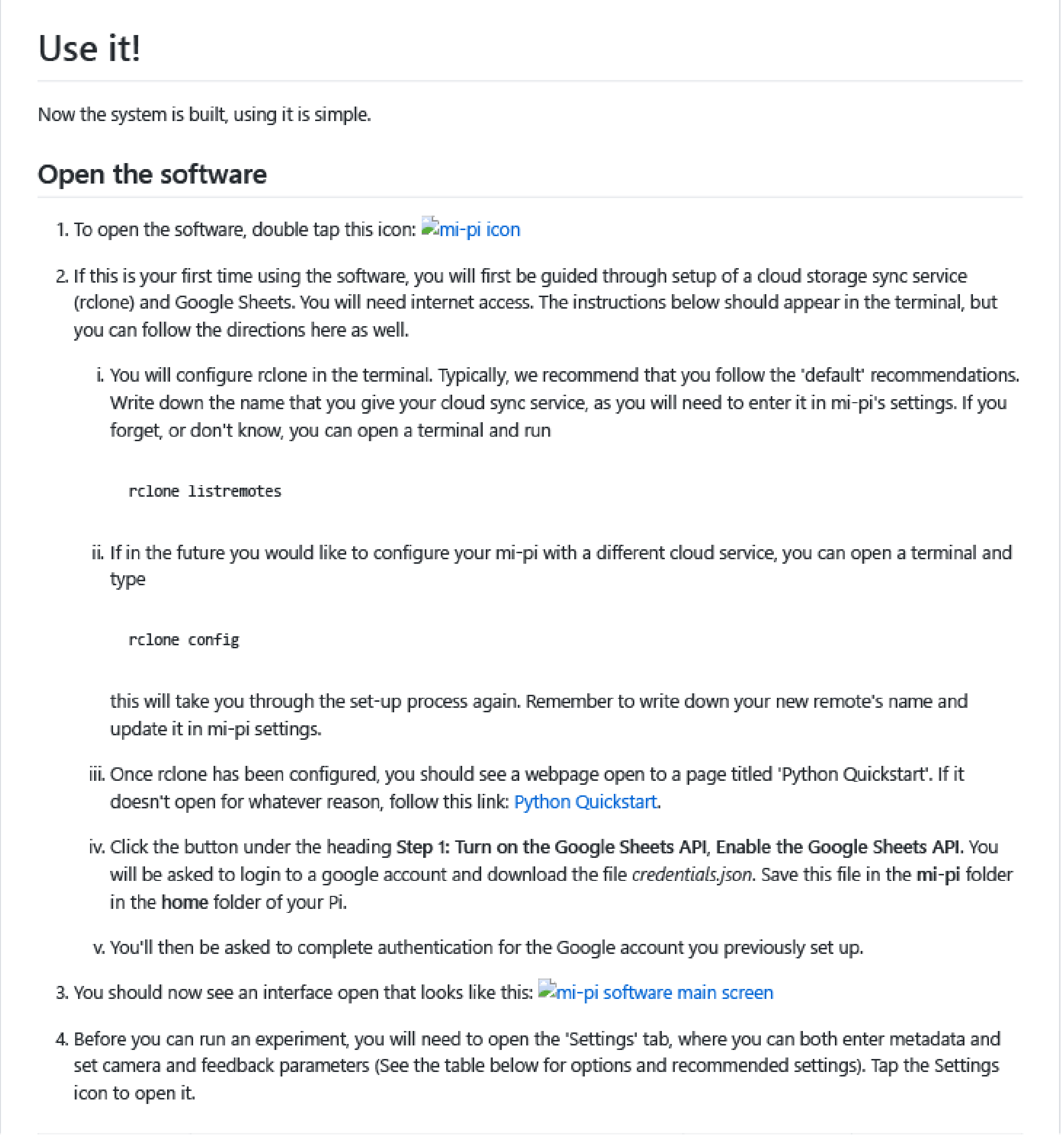


Directions

1. Carefully insert the prepared microSD cards into the microSD card slot on the Raspberry Pi.
2. Assemble the Raspberry Pi Touchscreen and Raspberry Pi in the SmartiPi Touch case as described here
3. Insert the camera board into the lego-compatible camera case. Attach the camera case to the lego set as shown. camera placed in lego set
4. Remove the protective film from the camera lens with tweezers.
5. Attach the camera's ribbon cable to the back of the Raspberry Pi. See directions here for how to attach ribbon cables.
6. Remove the 'Wall' side of the command strips and attach to the smooth 2x3 Lego tiles and press about 10 seconds.
7. Allow to set several minutes.
8. Remove the other side of the command strips and press the feet of the Raspberry Pi case onto it.
9. If using ethernet, plug in cable now; plug Raspberry Pi into 5V 3A power supply.
10. Congrats! All the physical parts of the system are now built. Now, we'll test and setup the software and hardware in the installation docs

A.3 Software setup

This section details how to use the system once all hardware is assembled. This documentation can also be found at https://github.com/lu-lab/mi-pi/blob/master/docs/useit.md



Use it!

Now the system is built, using it is simple.

Open the software

1. To open the software, double tap this icon: mi-pi icon
2. If this is your first time using the software, you will first be guided through setup of a cloud storage sync service (rclone) and Google Sheets. You will need internet access. The instructions below should appear in the terminal, but you can follow the directions here as well.
   i. You will configure rclone in the terminal. Typically, we recommend that you follow the 'default' recommendations. Write down the name that you give your cloud sync service, as you will need to enter it in mi-pi's settings. If you forget, or don't know, you can open a terminal and run

   ```
   rclone listremotes
   ```

   ii. If in the future you would like to configure your mi-pi with a different cloud service, you can open a terminal and type

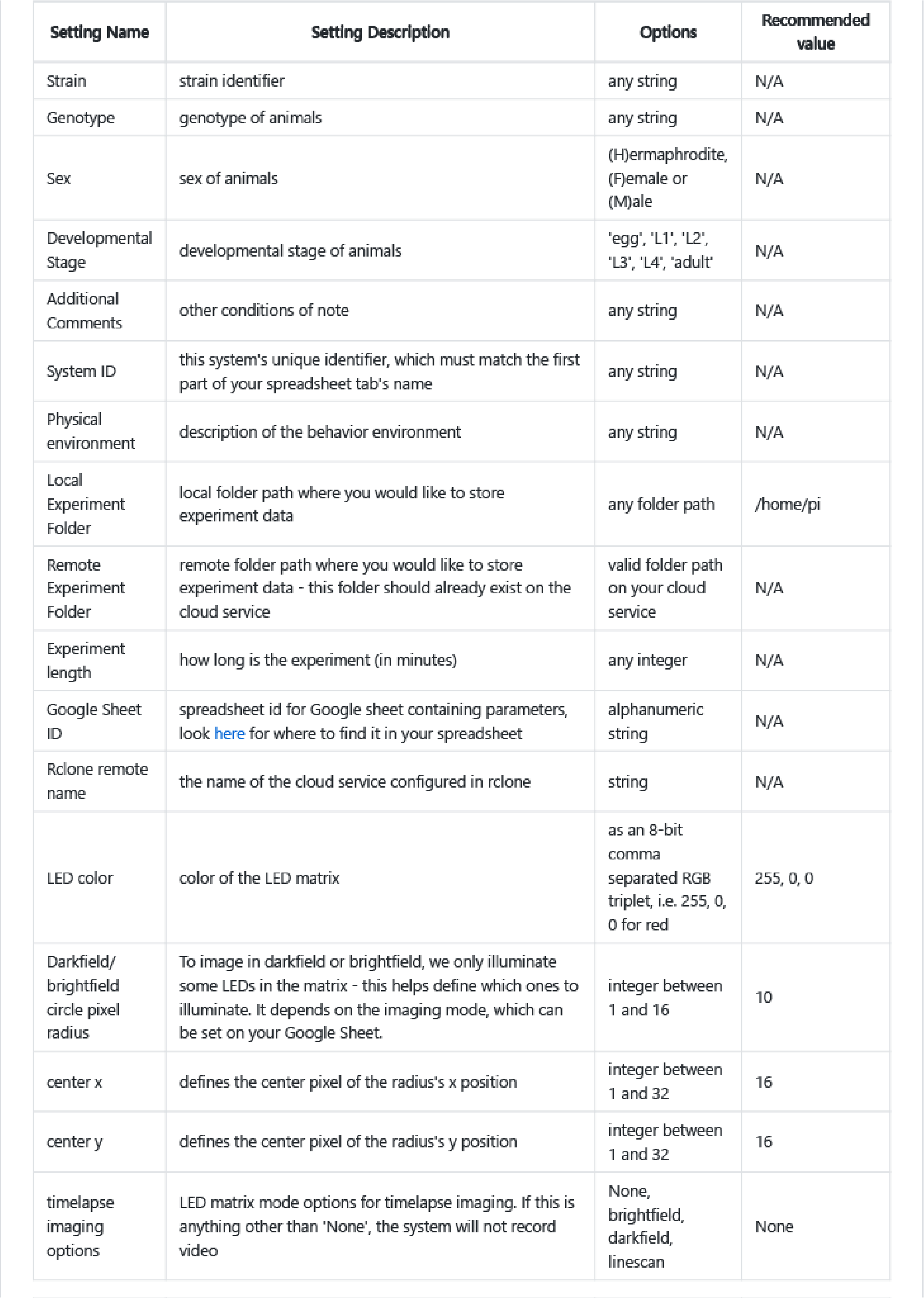
   ```
   rclone config
   ```

   this will take you through the set-up process again. Remember to write down your new remote's name and update it in mi-pi settings.

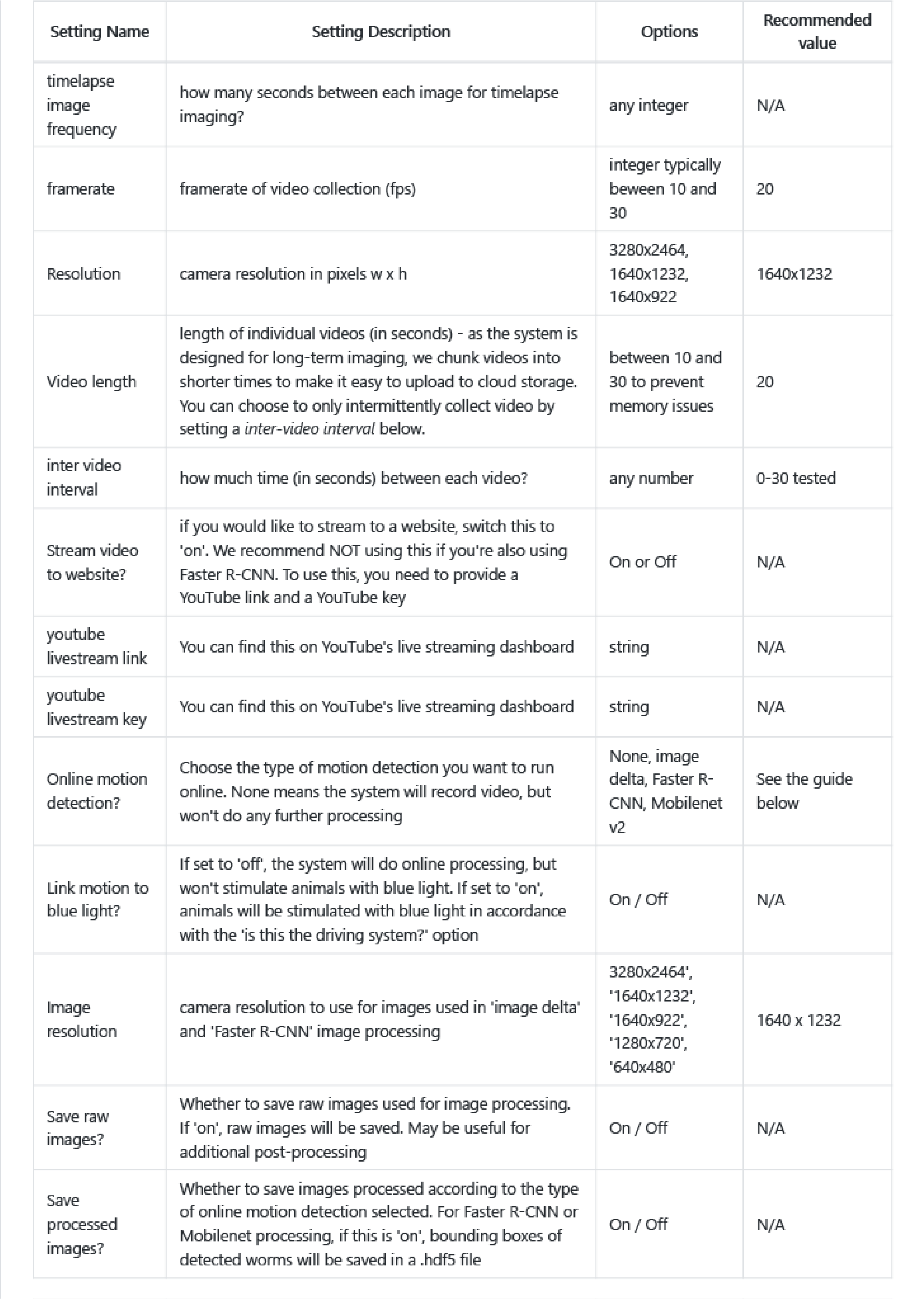
   iii. Once rclone has been configured, you should see a webpage open to a page titled 'Python Quickstart'. If it doesn't open for whatever reason, follow this link: Python Quickstart.

   iv. Click the button under the heading **Step 1: Turn on the Google Sheets API**, **Enable the Google Sheets API**. You will be asked to login to a google account and download the file *credentials.json*. Save this file in the **mi-pi** folder in the **home** folder of your Pi.

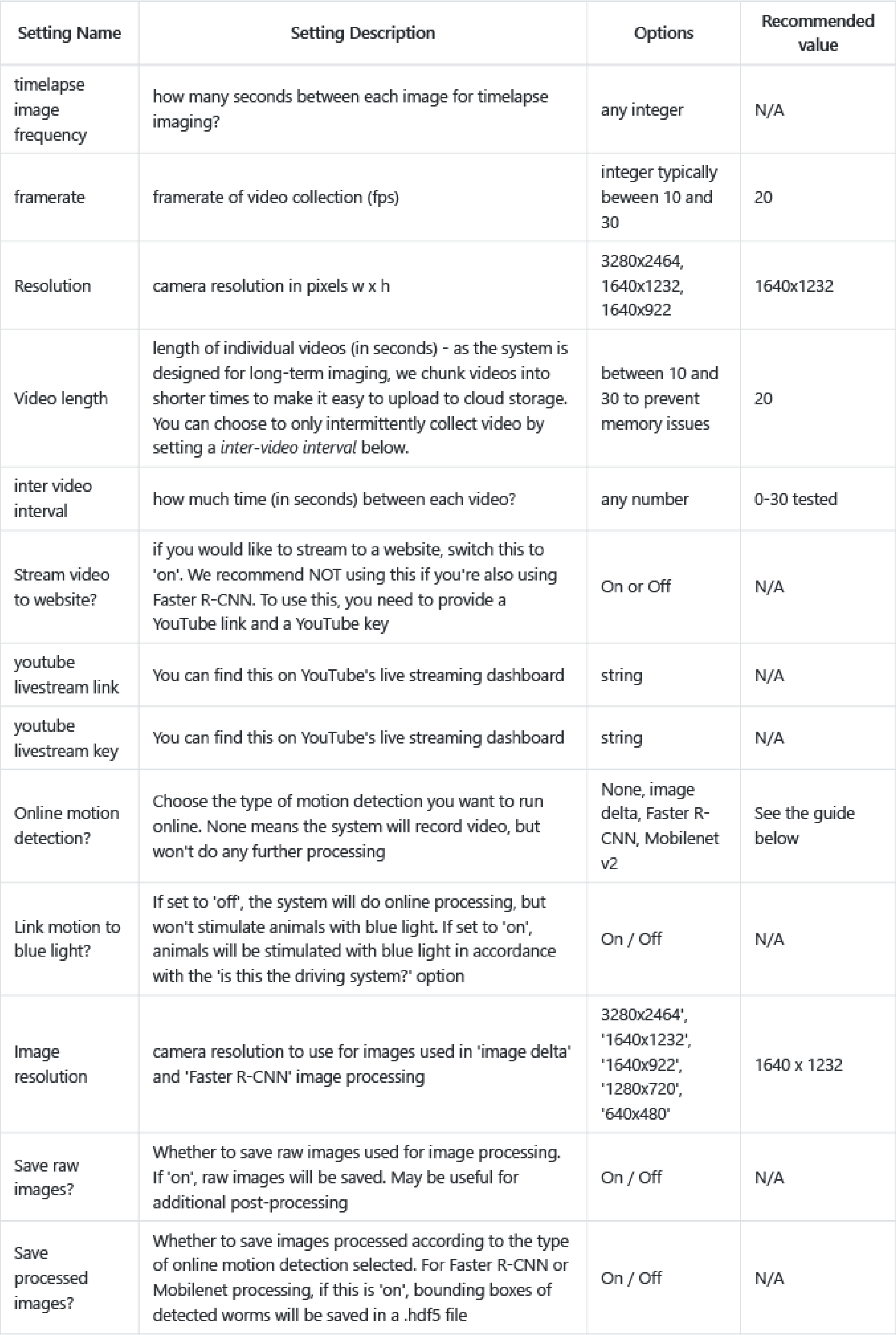
   v. You'll then be asked to complete authentication for the Google account you previously set up.
3. You should now see an interface open that looks like this: mi-pi software main screen
4. Before you can run an experiment, you will need to open the 'Settings' tab, where you can both enter metadata and set camera and feedback parameters (See the table below for options and recommended settings). Tap the Settings icon to open it.



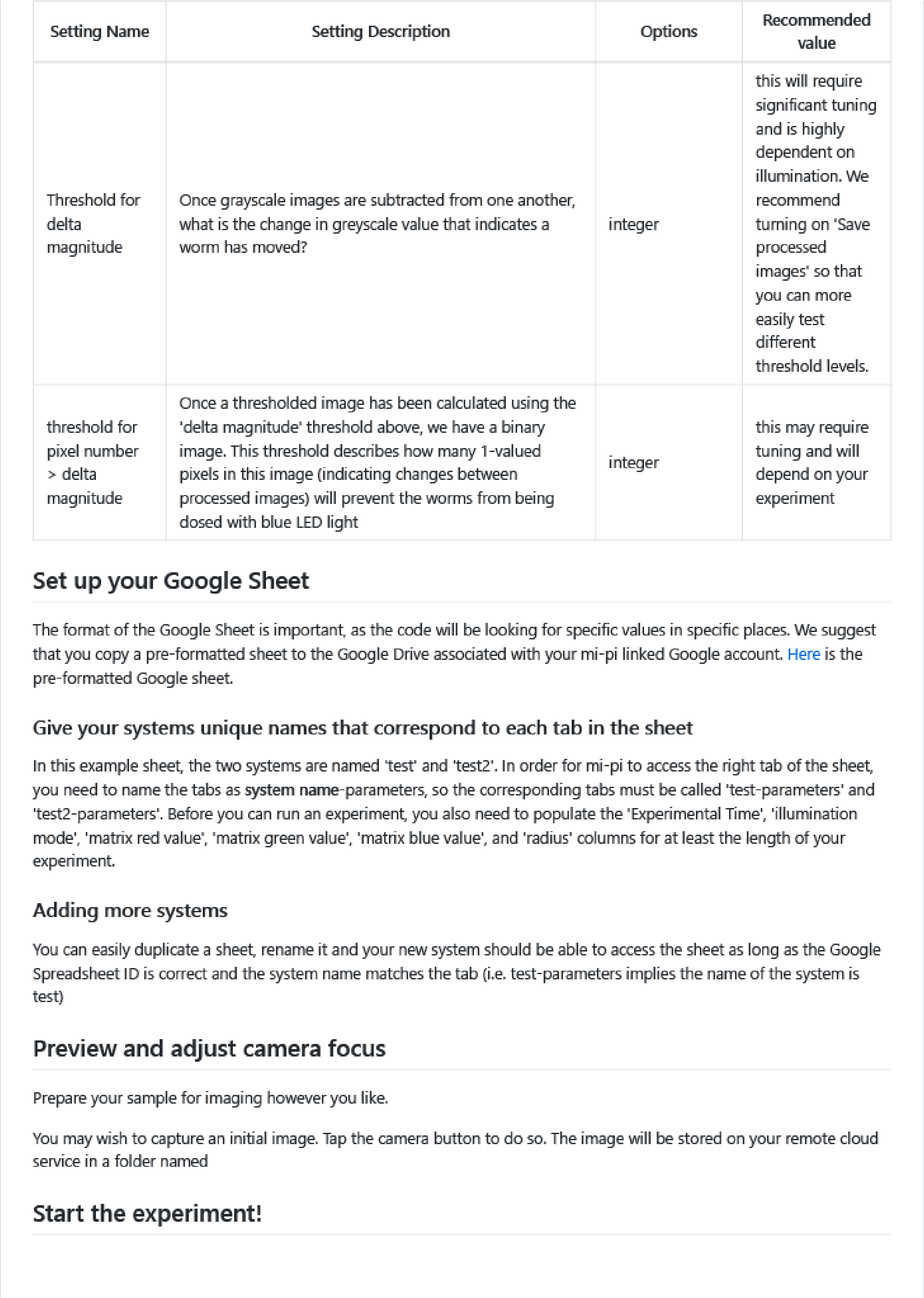
Setting Name	Setting Description	Options	Recommended value
Strain	strain identifier	any string	N/A
Genotype	genotype of animals	any string	N/A
Sex	sex of animals	(H)ermaphrodite, (F)emale or (M)ale	N/A
Developmental Stage	developmental stage of animals	'egg', 'L1', 'L2', 'L3', 'L4', 'adult'	N/A
Additional Comments	other conditions of note	any string	N/A
System ID	this system's unique identifier, which must match the first part of your spreadsheet tab's name	any string	N/A
Physical environment	description of the behavior environment	any string	N/A
Local Experiment Folder	local folder path where you would like to store experiment data	any folder path	/home/pi
Remote Experiment Folder	remote folder path where you would like to store experiment data - this folder should already exist on the cloud service	valid folder path on your cloud service	N/A
Experiment length	how long is the experiment (in minutes)	any integer	N/A
Google Sheet ID	spreadsheet id for Google sheet containing parameters, look here for where to find it in your spreadsheet	alphanumeric string	N/A
Rclone remote name	the name of the cloud service configured in rclone	string	N/A
LED color	color of the LED matrix	as an 8-bit comma separated RGB triplet, i.e. 255, 0, 0 for red	255, 0, 0
Darkfield/ brightfield circle pixel radius	To image in darkfield or brightfield, we only illuminate some LEDs in the matrix - this helps define which ones to illuminate. It depends on the imaging mode, which can be set on your Google Sheet.	integer between 1 and 16	10
center x	defines the center pixel of the radius's x position	integer between 1 and 32	16
center y	defines the center pixel of the radius's y position	integer between 1 and 32	16
timelapse imaging options	LED matrix mode options for timelapse imaging. If this is anything other than 'None', the system will not record video	None, brightfield, darkfield, linescan	None



Setting Name	Setting Description	Options	Recommended value
timelapse image frequency	how many seconds between each image for timelapse imaging?	any integer	N/A
framerate	framerate of video collection (fps)	integer typically beween 10 and 30	20
Resolution	camera resolution in pixels w x h	3280x2464, 1640x1232, 1640x922	1640x1232
Video length	length of individual videos (in seconds) - as the system is designed for long-term imaging, we chunk videos into shorter times to make it easy to upload to cloud storage. You can choose to only intermittently collect video by setting a *inter-video interval* below.	between 10 and 30 to prevent memory issues	20
inter video interval	how much time (in seconds) between each video?	any number	0-30 tested
Stream video to website?	if you would like to stream to a website, switch this to 'on'. We recommend NOT using this if you're also using Faster R-CNN. To use this, you need to provide a YouTube link and a YouTube key	On or Off	N/A
youtube livestream link	You can find this on YouTube's live streaming dashboard	string	N/A
youtube livestream key	You can find this on YouTube's live streaming dashboard	string	N/A
Online motion detection?	Choose the type of motion detection you want to run online. None means the system will record video, but won't do any further processing	None, image delta, Faster R-CNN, Mobilenet v2	See the guide below
Link motion to blue light?	If set to 'off', the system will do online processing, but won't stimulate animals with blue light. If set to 'on', animals will be stimulated with blue light in accordance with the 'is this the driving system?' option	On / Off	N/A
Image resolution	camera resolution to use for images used in 'image delta' and 'Faster R-CNN' image processing	3280x2464', '1640x1232', '1640x922', '1280x720', '640x480'	1640 x 1232
Save raw images?	Whether to save raw images used for image processing. If 'on', raw images will be saved. May be useful for additional post-processing	On / Off	N/A
Save processed images?	Whether to save images processed according to the type of online motion detection selected. For Faster R-CNN or Mobilenet processing, if this is 'on', bounding boxes of detected worms will be saved in a .hdf5 file	On / Off	N/A



Setting Name	Setting Description	Options	Recommended value
timelapse image frequency	how many seconds between each image for timelapse imaging?	any integer	N/A
framerate	framerate of video collection (fps)	integer typically beween 10 and 30	20
Resolution	camera resolution in pixels w x h	3280x2464, 1640x1232, 1640x922	1640x1232
Video length	length of individual videos (in seconds) - as the system is designed for long-term imaging, we chunk videos into shorter times to make it easy to upload to cloud storage. You can choose to only intermittently collect video by setting a *inter-video interval* below.	between 10 and 30 to prevent memory issues	20
inter video interval	how much time (in seconds) between each video?	any number	0-30 tested
Stream video to website?	if you would like to stream to a website, switch this to 'on'. We recommend NOT using this if you're also using Faster R-CNN. To use this, you need to provide a YouTube link and a YouTube key	On or Off	N/A
youtube livestream link	You can find this on YouTube's live streaming dashboard	string	N/A
youtube livestream key	You can find this on YouTube's live streaming dashboard	string	N/A
Online motion detection?	Choose the type of motion detection you want to run online. None means the system will record video, but won't do any further processing	None, image delta, Faster R-CNN, Mobilenet v2	See the guide below
Link motion to blue light?	If set to 'off', the system will do online processing, but won't stimulate animals with blue light. If set to 'on', animals will be stimulated with blue light in accordance with the 'is this the driving system?' option	On / Off	N/A
Image resolution	camera resolution to use for images used in 'image delta' and 'Faster R-CNN' image processing	3280x2464', '1640x1232', '1640x922', '1280x720', '640x480'	1640 x 1232
Save raw images?	Whether to save raw images used for image processing. If 'on', raw images will be saved. May be useful for additional post-processing	On / Off	N/A
Save processed images?	Whether to save images processed according to the type of online motion detection selected. For Faster R-CNN or Mobilenet processing, if this is 'on', bounding boxes of detected worms will be saved in a .hdf5 file	On / Off	N/A



Setting Name	Setting Description	Options	Recommended value
Threshold for delta magnitude	Once grayscale images are subtracted from one another, what is the change in greyscale value that indicates a worm has moved?	integer	this will require significant tuning and is highly dependent on illumination. We recommend turning on 'Save processed images' so that you can more easily test different threshold levels.
threshold for pixel number > delta magnitude	Once a thresholded image has been calculated using the 'delta magnitude' threshold above, we have a binary image. This threshold describes how many 1-valued pixels in this image (indicating changes between processed images) will prevent the worms from being dosed with blue LED light	integer	this may require tuning and will depend on your experiment

Set up your Google Sheet

The format of the Google Sheet is important, as the code will be looking for specific values in specific places. We suggest that you copy a pre-formatted sheet to the Google Drive associated with your mi-pi linked Google account. Here is the pre-formatted Google sheet.

Give your systems unique names that correspond to each tab in the sheet

In this example sheet, the two systems are named 'test' and 'test2'. In order for mi-pi to access the right tab of the sheet, you need to name the tabs as **system name**-parameters, so the corresponding tabs must be called 'test-parameters' and 'test2-parameters'. Before you can run an experiment, you also need to populate the 'Experimental Time', 'illumination mode', 'matrix red value', 'matrix green value', 'matrix blue value', and 'radius' columns for at least the length of your experiment.

Adding more systems

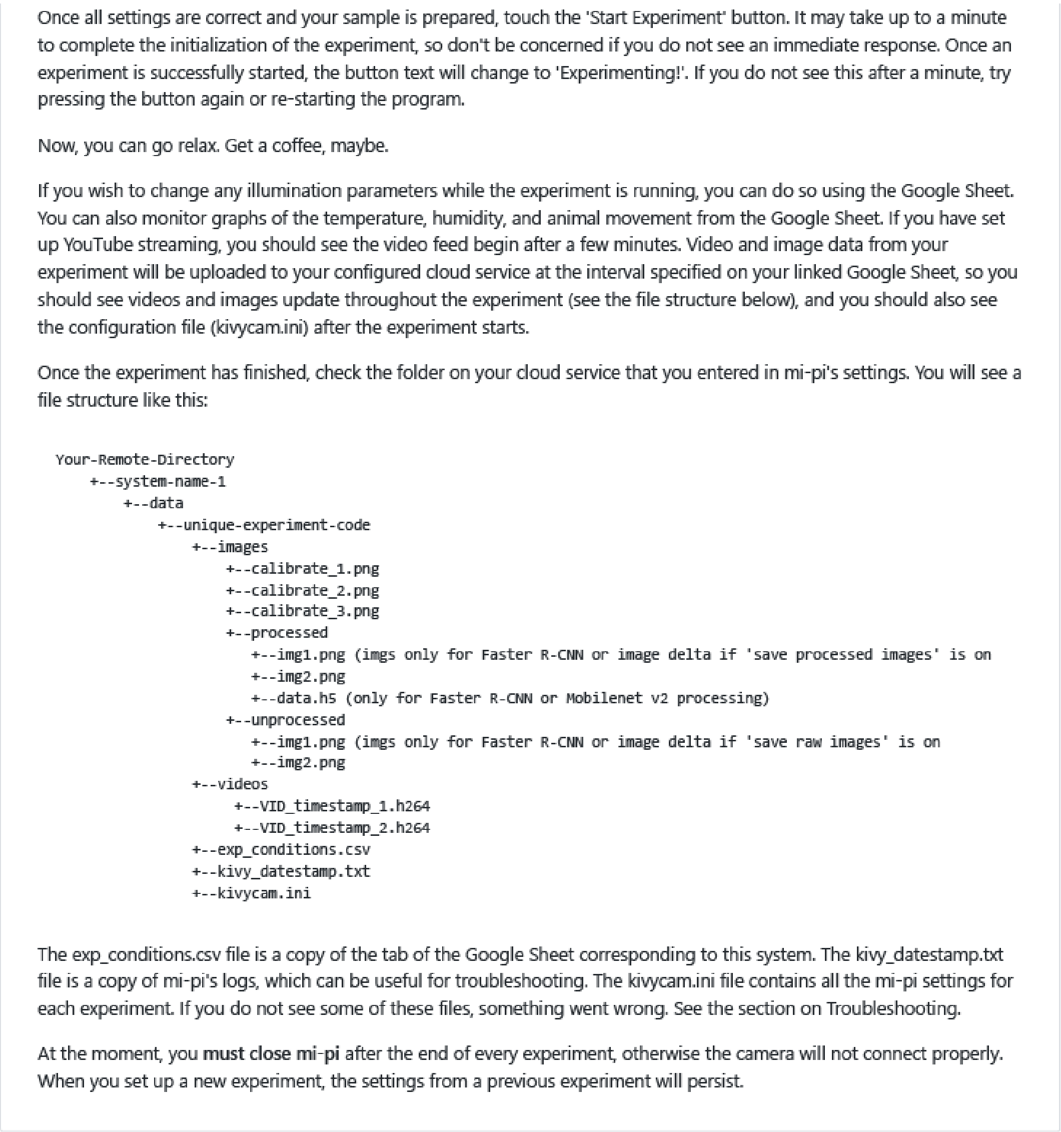
You can easily duplicate a sheet, rename it and your new system should be able to access the sheet as long as the Google Spreadsheet ID is correct and the system name matches the tab (i.e. test-parameters implies the name of the system is test)

Preview and adjust camera focus

Prepare your sample for imaging however you like.

You may wish to capture an initial image. Tap the camera button to do so. The image will be stored on your remote cloud service in a folder named

Start the experiment!



Once all settings are correct and your sample is prepared, touch the 'Start Experiment' button. It may take up to a minute to complete the initialization of the experiment, so don't be concerned if you do not see an immediate response. Once an experiment is successfully started, the button text will change to 'Experimenting!'. If you do not see this after a minute, try pressing the button again or re-starting the program.

Now, you can go relax. Get a coffee, maybe.

If you wish to change any illumination parameters while the experiment is running, you can do so using the Google Sheet. You can also monitor graphs of the temperature, humidity, and animal movement from the Google Sheet. If you have set up YouTube streaming, you should see the video feed begin after a few minutes. Video and image data from your experiment will be uploaded to your configured cloud service at the interval specified on your linked Google Sheet, so you should see videos and images update throughout the experiment (see the file structure below), and you should also see the configuration file (kivycam.ini) after the experiment starts.

Once the experiment has finished, check the folder on your cloud service that you entered in mi-pi's settings. You will see a file structure like this:

```
Your-Remote-Directory
    +--system-name-1
        +--data
            +--unique-experiment-code
                +--images
                    +--calibrate_1.png
                    +--calibrate_2.png
                    +--calibrate_3.png
                    +--processed
                       +--img1.png (imgs only for Faster R-CNN or image delta if 'save processed images' is on
                       +--img2.png
                       +--data.h5 (only for Faster R-CNN or Mobilenet v2 processing)
                    +--unprocessed
                       +--img1.png (imgs only for Faster R-CNN or image delta if 'save raw images' is on
                       +--img2.png
                +--videos
                     +--VID_timestamp_1.h264
                     +--VID_timestamp_2.h264
                +--exp_conditions.csv
                +--kivy_datestamp.txt
                +--kivycam.ini
```

The exp_conditions.csv file is a copy of the tab of the Google Sheet corresponding to this system. The kivy_datestamp.txt file is a copy of mi-pi's logs, which can be useful for troubleshooting. The kivycam.ini file contains all the mi-pi settings for each experiment. If you do not see some of these files, something went wrong. See the section on Troubleshooting.

At the moment, you **must close mi-pi** after the end of every experiment, otherwise the camera will not connect properly. When you set up a new experiment, the settings from a previous experiment will persist.

REFERENCES

1. Kanfer, F. H. & Saslow, G. Behavioral Analysis: An Alternative to Diagnostic Classification. *Arch. Gen. Psychiatry* **12**, 529–538 (1965).

2. Geschwind, D. H. & Flint, J. Genetics and genomics of psychiatric disease. *Science* **349**, 1489–1494 (2015).

3. White, J. G., Southgate, E., Thomson, J. N. & Brenner, S. The structure of the nervous system of the nematode Caenorhabditis elegans. **314**, 1–340 (1986).

4. Stroustrup, N. *et al.* The Caenorhabditis elegans Lifespan Machine. *Nat. Methods* **10**, 665–70 (2013).

5. Zhang, W. B. *et al.* Extended Twilight among Isogenic C. elegans Causes a Disproportionate Scaling between Lifespan and Health. *Cell Syst.* **3**, 333-345.e4 (2016).

6. Churgin, M. A. *et al.* Longitudinal imaging of *Caenorhabditis elegans* in a microfabricated device reveals variation in behavioral decline during aging. *Elife* **6**, e26652 (2017).

7. Churgin, M. A. *et al.* Quantitative imaging of sleep behavior in Caenorhabditis elegans and larval Drosophila melanogaster. *Nat. Protoc.* doi:10.1038/s41596-019-0146-6

8. Huang, K.-M., Cosman, P. & Schafer, W. R. Machine vision based detection of omega bends and reversals in C. elegans. *J. Neurosci. Methods* **158**, 323–336 (2006).

9. Yemini, E., Jucikas, T., Grundy, L. J., Brown, A. E. X. & Schafer, W. R. A database of Caenorhabditis elegans behavioral phenotypes. *Nat. Methods* **10**, 877–879 (2013).

10. Broekmans, O. D., Rodgers, J. B., Ryu, W. S. & Stephens, G. J. Resolving coiled shapes reveals new reorientation behaviors in C. elegans. *Elife* **5**, 1077–1084 (2016).

11. Berman, G. J., Choi, D. M., Bialek, W. & Shaevitz, J. W. Mapping the stereotyped behaviour of freely moving fruit flies. *J. R. Soc. Interface* **11**, 20140672–20140672 (2014).

12. Stephens, G. J., Johnson-Kerner, B., Bialek, W. & Ryu, W. S. Dimensionality and dynamics in the behavior of C. elegans. *PLoS Comput. Biol.* **4**, e1000028 (2008).

13. Tinbergen, N. *The study of instinct*. (Clarendon Press, 1989).

14. von Frisch, K. *The dancing bees: An account of the life and senses of the honey bee*. (Harcourt, Brace, 1953).

15. Lorenz, K. *Evolution and modification of behavior*. (University of Chicago Press, 1986).

16. Tinbergen, N. *The herring gull's world : a study of the social behaviour of birds*. (Lyons & Burford, 1989).

17. Skinner, B. *The Behavior of Organisms: An Experimental Analysis.* (Appleton-Century, 1938).

18. Gomez-Marin, A., Paton, J. J., Kampff, A. R., Costa, R. M. & Mainen, Z. F. Big behavioral data: Psychology, ethology and the foundations of neuroscience. *Nature Neuroscience* **17**, 1455–1462 (2014).

19. Reiser, M. B. & Dickinson, M. H. A modular display system for insect behavioral neuroscience. *J. Neurosci. Methods* **167**, 127–139 (2008).

20. Stowers, J. R. *et al.* Virtual reality for freely moving animals. *Nat. Methods* **14**, 995–1002 (2017).

21. Huang, K. H. *et al.* A virtual reality system to analyze neural activity and behavior in adult zebrafish. *Nat. Methods* **17**, 343–351 (2020).

22. Hölscher, C., Schnee, A., Dahmen, H., Setia, L. & Mallot, H. A. Rats are able to navigate in virtual environments. *J. Exp. Biol.* **208**, 561–569 (2005).

23. Schneider, C. A., Rasband, W. S. & Eliceiri, K. W. NIH Image to ImageJ: 25 years of image analysis. *Nat. Methods* **9**, 671–675 (2012).

24. Boergens, K. M. *et al.* webKnossos: efficient online 3D data annotation for connectomics. *Nat. Methods* **14**, 691–694 (2017).

25. Hughes, A. J. *et al.* Quanti.us: a tool for rapid, flexible, crowd-based annotation of images. *Nat. Methods* **15**, 587–590 (2018).

26. Berg, S. *et al.* ilastik: interactive machine learning for (bio)image analysis. *Nat. Methods* **16**, 1226–1232 (2019).

27. Restif, C. *et al.* CeleST: Computer Vision Software for Quantitative Analysis of C. elegans Swim Behavior Reveals Novel Features of Locomotion. *PLoS Comput. Biol.* **10**, e1003702 (2014).

28. Albrecht, D. R. & Bargmann, C. I. High-content behavioral analysis of Caenorhabditis elegans in precise spatiotemporal chemical environments. *Nat. Methods* **8**, 599–605 (2011).

29. Brenner, S. The Genetics of Caenorhabditis elegans. *Genetics* **77**, 71–94 (1974).

30. Han, B. *et al.* Dopamine signaling tunes spatial pattern selectivity in C. elegans. *Elife* **6**, (2017).

31. Zhang, Y., Lu, H. & Bargmann, C. I. Pathogenic bacteria induce aversive olfactory learning in Caenorhabditis elegans. *Nature* **438**, 179–184 (2005).

32. Suzuki, H. *et al.* Functional asymmetry in Caenorhabditis elegans taste neurons and its computational role in chemotaxis. *Nature* **454**, 114–117 (2008).

33. Bargmann, C. I. Chemosensation in C. elegans. *WormBook* 1–29 (2006). doi:10.1895/wormbook.1.123.1

34. Chalasani, S. H. *et al.* Dissecting a circuit for olfactory behaviour in Caenorhabditis elegans. *Nature* **450**, 63–70 (2007).

35. Jeon, M. *et al.* Similarity of the C. elegans developmental timing protein LIN-42 to circadian rhythm proteins. **286**, (1999).

36. Ellis, H. & Horvitz, H. R. Genetic control of programmed cell death in the nematode C. elegans. *Cell* **44**, 817–829 (1986).

37. Hsu, A. L., Murphy, C. T. & Kenyon, C. Regulation of aging and age-related disease by DAF-16 and heat-shock factor. *Science (80-.).* **300**, 1142–1145 (2003).

38. Rodriguez, M., Basten Snoek, L., De Bono, M. & Kammenga, J. E. Worms under stress: C. elegans stress response and its relevance to complex human disease and aging. *Trends in Genetics* **29**, 367–374 (2013).

39. Consortium*, T. C. elegans S. Genome sequence of the nematode C. elegans: A platform for investigating biology. *Science* **282**, 2012–2018 (1998).

40. Chalfie, M., Tu, Y., Euskirchen, G., Ward, W. W. & Prasher, D. C. Green fluorescent protein as a marker for gene expression. *Science* **263**, 802–5 (1994).

41. Friedland, A. E. *et al.* Heritable genome editing in C. elegans via a CRISPR-Cas9 system. *Nat. Methods* **10**, 741–743 (2013).

42. Stirman, J. N. *et al.* Real-time multimodal optical control of neurons and muscles in freely behaving Caenorhabditis elegans. *Nat. Methods* **8**, 153–8 (2011).

43. Chung, K. *et al.* Microfluidic chamber arrays for whole-organism behavior-based chemical screening. *Lab Chip* **11**, 3689 (2011).

44. Cho, Y. *et al.* Automated and controlled mechanical stimulation and functional imaging: In vivo in C. elegans. *Lab Chip* **17**, 2609–2618 (2017).

45. Crane, M. M. *et al.* Autonomous screening of C. elegans identifies genes implicated in synaptogenesis. *Nat. Methods* **9**, 977–980 (2012).

46. Brown, A. E. X., Yemini, E. I., Grundy, L. J., Jucikas, T. & Schafer, W. R. A dictionary of behavioral motifs reveals clusters of genes affecting Caenorhabditis

elegans locomotion. *Proc. Natl. Acad. Sci. U. S. A.* **110**, 791–796 (2013).

47. Stern, S., Kirst, C. & Bargmann, C. I. Neuromodulatory Control of Long-Term Behavioral Patterns and Individuality across Development. *Cell* **171**, 1649-1662.e10 (2017).

48. Porto, D. A., Giblin, J., Zhao, Y. & Lu, H. Reverse-Correlation Analysis of the Mechanosensation Circuit and Behavior in C. elegans Reveals Temporal and Spatial Encoding. *Sci. Rep.* **9**, 1–14 (2019).

49. Husson, S. J., Costa, W. S., Schmitt, C. & Gottschalk, A. Keeping track of worm trackers. *WormBook* 1–17 (2012). doi:10.1895/wormbook.1.156.1

50. Yemini, E., Jucikas, T., Grundy, L. J., Brown, A. E. X. & Schafer, W. R. A database of C. elegans behavioral phenotypes. *Nat. Methods* **10**, 877–879 (2013).

51. Javer, A. *et al.* An open-source platform for analyzing and sharing worm-behavior data. *Nature Methods* **15**, (2018).

52. Padmanabhan, V. *et al.* Locomotion of C. elegans: A Piecewise-Harmonic Curvature Representation of Nematode Behavior. *PLoS One* **7**, e40121 (2012).

53. Liu, M., Sharma, A. K., Shaevitz, J. W. & Leifer, A. M. Temporal processing and context dependency in caenorhabditis elegans response to mechanosensation. *Elife* **7**, (2018).

54. Berman, G. J., Bialek, W. & Shaevitz, J. W. Predictability and hierarchy in Drosophila behavior. *Proc. Natl. Acad. Sci. U. S. A.* **113**, 11943–11948 (2016).

55. Cande, J. *et al.* Optogenetic dissection of descending behavioral control in Drosophila. *Elife* **7**, (2018).

56. Silverman, J. L., Yang, M., Lord, C. & Crawley, J. N. Behavioural phenotyping assays for mouse models of autism. *Nature Reviews Neuroscience* **11**, 490–502 (2010).

57. Patterson, P. H. Immune involvement in schizophrenia and autism: Etiology, pathology and animal models. *Behav. Brain Res.* **204**, 313–321 (2009).

58. Nestler, E. J. & Hyman, S. E. Animal models of neuropsychiatric disorders. *Nature Neuroscience* **13**, 1161–1169 (2010).

59. Götz, J. & Ittner, L. M. Animal models of Alzheimer's disease and frontotemporal dementia. *Nature Reviews Neuroscience* **9**, 532–544 (2008).

60. Dillin, A. *et al.* Rates of behavior and aging specified by mitochondrial function during development. *Science (80-.).* **298**, 2398–2401 (2002).

61. Felkai, S. *et al.* CLK-1 Controls Respiration, Behavior and Aging in the Nematode Caenorhabditis Elegans - PubMed. *EMBO J* **18**, 1783–1792 (1999).

62. Aubry, G. & Lu, H. Droplet array for screening acute behaviour response to chemicals in: Caenorhabditis elegans. *Lab Chip* **17**, 4303–4311 (2017).

63. Winter, Y. & Schaefers, A. T. U. A sorting system with automated gates permits individual operant experiments with mice from a social home cage. *J. Neurosci. Methods* **196**, 276–280 (2011).

64. Wong, B. G., Mancuso, C. P., Kiriakov, S., Bashor, C. J. & Khalil, A. S. Precise, automated control of conditions for high-throughput growth of yeast and bacteria with eVOLVER. *Nat. Biotechnol.* **36**, 614–623 (2018).

65. Stiernagle, T. Maintenance of C. elegans. *WormBook : the online review of C. elegans biology* 1–11 (2006). doi:10.1895/wormbook.1.101.1

66. Fletcher, M. & Kim, D. H. Age-Dependent Neuroendocrine Signaling from Sensory Neurons Modulates the Effect of Dietary Restriction on Longevity of Caenorhabditis elegans. *PLOS Genet.* **13**, e1006544 (2017).

67. Starich, T. A., Xu, J., Skerrett, I. M., Nicholson, B. J. & Shaw, J. E. Interactions between innexins UNC-7 and UNC-9 mediate electrical synapse specificity in the Caenorhabditis elegans locomotory nervous system. *Neural Dev.* **4**, 1–28 (2009).

68. Dernovici, S., Starc, T., Dent, J. A. & Ribeiro, P. The serotonin receptor SER-1 (5HT2ce) contributes to the regulation of locomotion in Caenorhabditis elegans. *Dev. Neurobiol.* **67**, 189–204 (2007).

69. Driver, R. J. J., Lamb, A. L. L., Wyner, A. J. J. & Raizen, D. M. M. DAF-16/FOXO Regulates Homeostasis of Essential Sleep-like Behavior during Larval Transitions in C. elegans. *Curr. Biol.* **23**, 501–506 (2013).

70. Raizen, D. M. *et al.* Lethargus is a Caenorhabditis elegans sleep-like state. *Nature* **451**, 569–72 (2008).

71. Iwanir, S. *et al.* The Microarchitecture of C. elegans Behavior during Lethargus: Homeostatic Bout Dynamics, a Typical Body Posture, and Regulation by a Central Neuron. *Sleep* **36**, 385–395 (2013).

72. Spies, J. & Bringmann, H. Automated detection and manipulation of sleep in C. Elegans reveals depolarization of a sleep-active neuron during mechanical stimulation-induced sleep deprivation. *Sci. Rep.* **8**, (2018).

73. Huang, J. *et al.* Speed/accuracy trade-offs for modern convolutional object detectors. in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* **2017-January**, 3296–3305 (Institute of Electrical and Electronics Engineers Inc., 2017).

74. Lin, T. Y. *et al.* Microsoft COCO: Common objects in context. in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **8693 LNCS**, 740–755 (Springer Verlag, 2014).

75. Ren, S., He, K., Girshick, R. & Sun, J. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*.

76. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. & Chen, L. C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 4510–4520 (IEEE Computer Society, 2018). doi:10.1109/CVPR.2018.00474

77. Flavell, S. W. *et al.* Serotonin and the neuropeptide PDF initiate and extend opposing behavioral states in C. Elegans. *Cell* **154**, 1023–1035 (2013).

78. Cirelli, C. The genetic and molecular regulation of sleep: from fruit flies to humans. *Nat. Rev. Neurosci.* **10**, 549–560 (2009).

79. Edwards, S. L. *et al.* A Novel Molecular Solution for Ultraviolet Light Detection in Caenorhabditis elegans. *PLoS Biol.* **6**, e198 (2008).

80. Ward, A., Liu, J., Feng, Z. & Xu, X. Z. S. Light-sensitive neurons and channels mediate phototaxis in C. elegans. *Nat. Neurosci.* **11**, 916–922 (2008).

81. Lawler, D. E. *et al.* Automated analysis of sleep in adult C. elegans with closed-loop assessment of state-dependent neural activity. *bioRxiv* 791764 (2019). doi:10.1101/791764

82. Gong, J. *et al.* The C. elegans taste receptor homolog LITE-1 is a photoreceptor. *Cell* **167**, 1252-1263.e10 (2016).

83. Guo, Y., Liu, Y., Georgiou, T. & Lew, M. S. A review of semantic segmentation using deep neural networks. *Int. J. Multimed. Inf. Retr.* **7**, 87–93 (2018).

84. Mathis, A. *et al.* DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nat. Neurosci.* **21**, (2018).

85. Nath, T. *et al.* Using DeepLabCut for 3D markerless pose estimation across species and behaviors. *Nat. Protoc.* **14**, 2152–2176 (2019).

86. Kim, J. S. *et al.* Space–time wiring specificity supports direction selectivity in the retina. *Nature* **509**, 331–336 (2014).

87. Helmstaedter, M. Cellular-resolution connectomics: challenges of dense neural circuit reconstruction. *Nat. Methods* **10**, 501–507 (2013).

88. Amat, F. *et al.* Fast, accurate reconstruction of cell lineages from large-scale fluorescence microscopy data. *Nat. Methods* **11**, 951–958 (2014).

89. de Chaumont, F. *et al.* Computerized video analysis of social interactions in mice. *Nat. Methods* **9**, 410–417 (2012).

90. Kabra, M., Robie, A. A., Rivera-Alba, M., Branson, S. & Branson, K. JAABA: interactive machine learning for automatic annotation of animal behavior. *Nat. Methods* **10**, 64–67 (2013).

91. Tungtur, S. K., Nishimune, N., Radel, J. & Nishimune, H. Mouse Behavior Tracker: An economical method for tracking behavior in home cages. *Biotechniques* **63**, (2017).

92. Pottash, A. E., McKay, R., Virgile, C. R., Ueda, H. & Bentley, W. E. TumbleScore: Run and tumble analysis for low frame-rate motility videos. *Biotechniques* **62**, (2017).

93. Carpenter, A. E., Kamentsky, L. & Eliceiri, K. W. A call for bioimaging software usability. *Nat. Methods* **9**, 666–70 (2012).

94. Jackson-Holmes, E. L., McDevitt, T. C. & Lu, H. A microfluidic trap array for longitudinal monitoring and multi-modal phenotypic analysis of individual stem cell aggregates. *Lab Chip* **17**, 3634–3642 (2017).

95. Rainie, L. & Zickuhr, K. *Americans' Views on Mobile Etiquette*. (2015).

96. Fire, A. *et al.* Potent and specific genetic interference by double-stranded RNA in Caenorhabditis elegans. *Nature* **391**, 806–11 (1998).

97. Nelson, L. S. *et al.* Disruption of a neuropeptide gene, flp-1, causes multiple behavioral defects in Caenorhabditis elegans. *Science* **281**, 1686–90 (1998).

98. Mendel, J. *et al.* Participation of the protein Go in multiple aspects of behavior in C. elegans. *Science (80-.).* **267**, 1652–1655 (1995).

99. de Bono, M. & Bargmann, C. I. Natural Variation in a Neuropeptide Y Receptor Homolog Modifies Social Behavior and Food Response in C. elegans. *Cell* **94**, 679–689 (1998).

100. Cooper, S. *et al.* Predicting protein structures with a multiplayer online game. *Nature* **466**, 756–760 (2010).

101. Wong, B. B. M. & Candolin, U. Behavioral responses to changing environments. *Behavioral Ecology* **26**, 665–673 (2015).

102. Lewis, J. A. & Fleming, J. T. Chapter 1: Basic Culture Methods. *Methods Cell Biol.* **48**, 3–29 (1995).

103. Frézal, L. & Félix, M.-A. C. elegans outside the Petri dish. *Elife* **4**, e05849 (2015).

104. Schulenburg, H. & Félix, M.-A. The Natural Biotic Environment of Caenorhabditis elegans. *Genetics* **206**, 55–86 (2017).

105. Samuel, B. S., Rowedder, H., Braendle, C., Félix, M. A. & Ruvkun, G. Caenorhabditis elegans responses to bacteria from its natural habitats. *Proc. Natl. Acad. Sci. U. S. A.* **113**, E3941–E3949 (2016).

106. Golden, J. W. & Riddle, D. L. A Caenorhabditis elegans dauer-inducing pheromone and an antagonistic component of the food supply. *J. Chem. Ecol.* **10**, 1265–1280 (1984).

107. Parida, L. & Padmanabhan, V. Durotaxis in Nematode Caenorhabditis elegans. *Biophys. J.* **111**, 666–674 (2016).

108. Korta, J., Clark, D. A., Gabel, C. V., Mahadevan, L. & Samuel, A. D. T. Mechanosensation and mechanical load modulate the locomotory gait of swimming C. elegans. *J. Exp. Biol.* **210**, 2383–2389 (2007).

109. Backholm, M., Kasper, A. K. S., Schulman, R. D., Ryu, W. S. & Dalnoki-Veress, K. The effects of viscosity on the undulatory swimming dynamics of C. elegans. *Phys. Fluids* **27**, 091901 (2015).

110. Boyle, J. H., Berri, S., Tassieri, M., Hope, I. A. & Cohen, N. Gait Modulation in C. Elegans: It's Not a Choice, It's a Reflex! *Front. Behav. Neurosci.* **5**, 10 (2011).

111. Berri, S., Boyle, J. H., Tassieri, M., Hope, I. A. & Cohen, N. Forward locomotion of the nematode C. elegans is achieved through modulation of a single gait. *HFSP J.* **3**, 186–193 (2009).

112. Boyle, J. H., Berri, S. & Cohen, N. Gait Modulation in C. elegans: An Integrated Neuromechanical Model. *Front. Comput. Neurosci.* **6**, 10 (2012).

113. Fang-Yen, C. *et al.* Biomechanical analysis of gait adaptation in the nematode Caenorhabditis elegans. *Proc. Natl. Acad. Sci. U. S. A.* **107**, 20323–20328 (2010).

114. Liu, H. *et al.* Cholinergic Sensorimotor Integration Regulates Olfactory Steering Highlights d Sensorimotor integration regulates goal-directed behavioral tasks d Two cholinergic signals encode sensory and motor information d Sensory and motor cholinergic signals interact to generate integration d Experience-dependent changes in sensorimotor integration. *Neuron* **97**, 390-405.e3 (2018).

115. Bilbao, A., Patel, A. K., Rahman, M., Vanapalli, S. A. & Blawzdziewicz, J. Roll maneuvers are essential for active reorientation ofCaenorhabditis elegansin 3D media. *Proc. Natl. Acad. Sci. U. S. A.* 201706754 (2018). doi:10.1073/pnas.1706754115

116. Maaten, L. van der & Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).

117. Saravanan, V., Berman, G. J. & Sober, S. J. Application of the hierarchical bootstrap to multi-level data in neuroscience. *bioRxiv* 819334 (2019). doi:10.1101/819334

118. Shaw, M. *et al.* Three-dimensional behavioural phenotyping of freely moving C. elegans using quantitative light field microscopy. *PLoS One* **13**, e0200108 (2018).

119. Klibaite, U., Berman, G. J., Cande, J., Stern, D. L. & Shaevitz, J. W. An unsupervised method for quantifying the behavior of paired animals. *Phys. Biol.* **14**, (2017).

120. Ramani, A. K. *et al.* The majority of animal genes are required for wild-type fitness. *Cell* **148**, 792–802 (2012).

121. Zaari, N., Rajagopalan, P., Kim, S. K., Engler, A. J. & Wong, J. Y. Photopolymerization in Microfluidic Gradient Generators: Microscale Control of Substrate Compliance to Manipulate Cell Response. *Adv. Mater.* **16**, 2133–2137 (2004).

122. Busack, I., Jordan, F., Sapir, P. & Bringmann, H. The OptoGenBox - a device for long-term optogenetics in C. elegans. *bioRxiv* 2020.01.13.903948 (2020). doi:10.1101/2020.01.13.903948