Simultaneous measurement of sleep and feeding in individual *Drosophila*

Keith R Murphy[1–3], Jin Hong Park[1,2], Robert Huber[4,5] & William W Ja[1,2]

[1]Department of Neuroscience, The Scripps Research Institute, Jupiter, Florida, USA. [2]Center on Aging, The Scripps Research Institute, Jupiter, Florida, USA. [3]Program in Integrative Biology and Neuroscience, Florida Atlantic University, Jupiter, Florida, USA. [4]Radcliffe Institute for Advanced Study, Harvard University, Cambridge, Massachusetts, USA. [5]J.P. Scott Center for Neuroscience, Mind and Behavior, Bowling Green State University, Bowling Green, Ohio, USA. Correspondence should be addressed to R.H. (lobsterman.bgsu@gmail.com) or W.W.J. (wja@scripps.edu).

Published online 12 October 2017; doi:10.1038/nprot.2017.096.

***Drosophila* is widely used for the dissection of genetic and neuronal mechanisms of behavior. Recently, flies have emerged as a model for investigating the regulation of feeding and sleep. Although typically studied in isolation, increasing evidence points to a fundamental connection between these behaviors. Thus, a system for measuring sleep and feeding simultaneously in a single integrated system is important for interpreting behavioral shifts of either state. Here, we describe the construction and use of the Activity Recording Capillary Feeder or CAFE (ARC), a machine-vision (automated image tracking)-based system for the integrated measurement of sleep and feeding in individual *Drosophila*. Flies feed on liquid food from a microcapillary, and consumption is measured by tracking the liquid meniscus over time. Sleep measurements are obtained from positional tracking of individual animals, and arousal threshold can be determined by vibrational stimulus response. Using this system, a single computer and experimenter can track diverse behaviors from up to 60 individual flies in a single integrated system. The ARC is efficiently assembled with minimal training, and each experiment can be run for up to ~7 d, with a total setup and breakdown time of ~2 h.**

INTRODUCTION

Increasing evidence points to a fundamental relationship between feeding and sleep. In various animal models, restricting food intake drives wakefulness[1–3] and manipulating available nutrients modifies sleep on both short and long time scales[4–9]. Beyond the physical act of feeding, changes in energy storage or metabolism stemming from feeding behavior may also be reflected in sleep. Sleep loss can increase appetite[10], and daytime sleepiness has been associated with obesity[11].

Drosophila has recently emerged as a model for studying the genetic, environmental, and neuronal basis of sleep—largely because of the ease of high-throughput studies at low spatial and monetary cost[12–14]. Analogous to its mammalian counterpart, fly sleep is characterized by a rise in arousal threshold, a postural shift, homeostatic regulation, and altered electrical activity in the brain[15–17]. Fly sleep is also necessary for memory formation and survival[18–20]. Because of this functional conservation and the powerful genetic tools available for manipulating flies, *Drosophila* will probably continue to have a large role in dissecting the fundamental relationships between sleep and feeding. However, sleep–feeding studies suffer from a major caveat—the two behaviors are typically not measured together in the same animals. Insights into the direct impact of environmental, genetic, and other manipulations on sleep and feeding—and their interaction—might only be revealed by characterizing both behaviors concurrently in individual animals.

Here, we describe a protocol for the simultaneous measurement of sleep and feeding in adult *Drosophila*. This method uses high-resolution machine vision to accurately track the position of individual flies and to follow the meniscus of liquid-food-containing capillaries that the animals feed from. Previously, use of the ARC was described for measuring the timing and size of discrete feeding events while simultaneously assessing sleep before and after each event. This allowed us to examine the rapid shifts in sleep that result from a meal[4]. The primary advantage of the ARC is the coupling of sleep, arousal, and feeding measurements into a single system to reveal direct behavioral correlates and increase experimental throughput. However, individual ARC modules for measuring feeding, sleep, or activity can also be run independently and offer advantages over other independent behavioral measurement systems, including ease of chamber customization, increased throughput, and real-time analysis capabilities. Although we previously published an iteration of the ARC that used separate cameras and computers for tracking fly movement and food intake[4], the upgraded system described here uses a single camera per chamber of 30 flies for real-time measurement of both sleep and feeding. A single computer can typically handle two cameras simultaneously for the study of 60 flies in two chambers.

Comparison with other available methods

The ARC monitors consumption continuously during an experiment by repeatedly sampling the liquid-food meniscus position rather than making a single measurement at the end[21]. The system therefore resolves discrete feeding events, facilitating the study of prandial behavior as well as the finer integration of feeding and spatially derived behaviors such as sleep[4]. In the described setup, the ARC makes measurements from 30 individuals per camera; yet the data quality is comparable with that from another recently described capillary-based tool, the Expresso, which is also capable of nanoliter resolution[22]. Replication of an Expresso experiment shows slightly lower absolute feeding in the ARC but similar relative changes in consumption across several diets (**Supplementary Fig. 1**). The Expresso measures food intake of five flies by using LED sensor arrays to measure capillary liquid food levels. Alternative systems for capturing high-resolution feeding behavior have focused on quantifying fly-food contact, a correlate of food intake, by using capacitance[23,24]. Similar to the Expresso, these systems require independent sensors for each fly, increasing the cost per animal. Sensors can also obscure the view of each fly, making it difficult to measure sleep. By assessing the feeding behavior of many flies from a single camera, the ARC substantially reduces cost and increases experimental throughput. It is also notable that the ARC can detect smaller feeding events by positioning the camera closer to a subset of capillaries.

The ARC implements object tracking by using JavaGrinders[4], an open-source, Java-based framework. Acquisition of frames and image operations are performed with OpenCV[25], a real-time computer-vision library. Although video tracking is notoriously processor intensive—requiring most object trackers to rely on analysis of prerecorded videos—the ARC alleviates this difficulty by using various computational optimizations that allow for real-time extraction of behavioral properties. Hence, information is saved to a coordinate output file during the experiment, eliminating the need to store large video files and analyze them post experiment. For most laboratories, this level of data reduction would allow experiments to span days or weeks, rather than restricting studies to briefer periods. The basic ARC does not implement computational methods to classify specific behaviors such as grooming or wing flicking. However, video (or images) can be optionally stored from the ARC data capture program and analyzed after the fact with machine-learning-based analytic systems such as the Janelia Automatic Animal Behavior Annotator (JAABA)[26]. The Java framework also allows for a custom class to override the built-in analysis procedure. This would potentially allow characterization of specific behaviors at run time rather than in post processing with a saved video file.

Although we describe implementation of the ARC for tracking sleep, feeding, and arousal threshold in individual animals, additional JavaGrinders modules are currently available that extend ARC capabilities. The ARC can be configured to track multiple animals in one arena, identify color-labeled objects, combine information from two cameras for 3D spatial localization, and control real-time robotic interfaces in response to animal behavior. One example of these capabilities would be to drive a light stimulus to activate optogenetic tools when flies enter a specific region of an arena, consume food, or perform a certain behavior. With support for input and output to various microcontrollers (e.g., Arduino, Phidget, Raspberry PI GPIO pins, Allen Bradley Programmable Logic Controllers, and Analog Devices SMT), JavaGrinders can communicate with relays, servos, and sensors over a range of communication protocols (e.g., USB, I2C/SPI, and TCP/IP). These features offer a widely adaptable system to explore animal behavior in an adaptive environment not previously accessible in open-source tools. Furthermore, the ARC can be run on single-board computers such as the Raspberry Pi 3 Model B or the Hardkernel Odroid. Instructions and fully installed images are available for download on the JavaGrinders ARC blog (http://www.javagrinders-arc.blogspot.com/2017/01/install-open-source-tools-for-running.html).

Experimental design

Using the ARC involves four major stages: construction and preparation of the ARC chamber and environment, configuring of the computational framework for the data-capture software (Equipment Setup), running of the ARC experiments (Steps 1–13), and analysis of ARC data (Steps 14–15). See **Supplementary Video 1** for an overview of the procedure.

Construction and preparation of the ARC. Although the protocol provided here is a rigid instruction set for building and operating the ARC, the method is highly flexible and can be customized for different uses and conditions. Beginning with its construction, the ARC requires only that animals be individually housed in a chamber that (i) allows full visibility of the animals by a camera and (ii) contains an overhead opening in which a capillary containing liquid food can be secured vertically, the bottom of which must be accessible by the animal. The design we provide is an optimization of these steps, meant to maximize throughput and behavioral measurement.

The ARC chamber (**Fig. 1a**) contains half-cylindrical-shaped wells to minimize edge dwelling, with a volume substantially larger than the commonly used *Drosophila* Activity Monitor (DAM) tubes[27], allowing the animals more room to explore and exhibit spatial tendencies. The size also allows a camera to track 30 flies, a number comparable to the capacity of a DAM system (32 animals), while still allowing a clear outline of each fly with a full high-definition (FHD) camera (1,080p resolution). The chambers are vertically oriented to allow visual access to both the flies and the food capillaries from a single camera (**Fig. 1b**). Previously, we found that the vertical orientation does not affect daily sleep patterning of individual flies[4]. Notably, one ARC controller can operate multiple independent setups; we include a program implementing two tracking cameras (60 flies) called ARCControllerMultiCam.java.

All parts can be printed in acrylonitrile butadiene styrene (ABS) or polylactic acid (PLA). The chamber material used should be consistent across experiments, as there may be differences in fly climbing ability or other behaviors across materials. The ARC chamber, along with the rest of the pieces shown, can be printed using the included .stl files (**Supplementary Data**). Once printed, the face of the chamber is mounted with a 2-mm-thick, clear plexiglass shield to allow visualization of each fly. Although plexiglass is sufficiently strong and transparent, other equivalent materials could be used.

At the bottom of each well, agar is dispensed before each experiment. This provides flies with *ad libitum* access to water, eliminating any experimental confounds that could arise from dehydration[28]. Beneath the chamber, a base plate is used to keep the chamber upright. The base plate should fit snugly to the bottom of the chamber. Above the wells, a gate containing regularly spaced 2.5-mm holes aligned with each well is inserted into the groove above the chamber and pushed into place (**Fig. 1c**). Each of the holes is fitted with a sleeve to hold 5-μl glass capillaries in place.

A glass capillary containing aqueous food is provided for each fly. To track changes in liquid levels, each capillary also contains a 2-mm oil layer immediately above the aqueous food. This organic phase is made up of dodecane and light mineral oil in a mixture that was determined empirically. Tracking of the oil layer is dependent on an IR-absorbent, oil-soluble dye that was previously identified by screening several candidates[4]. Behind the capillaries, a backsplash is attached using a piece of construction paper and double-sided tape (**Fig. 1a**). This allows the experimenter to keep the capillaries straight and to create a flat contrast between dye bands and the background.

To limit experimental variability, the ARC should be housed in an incubator with controlled light, temperature, and humidity.

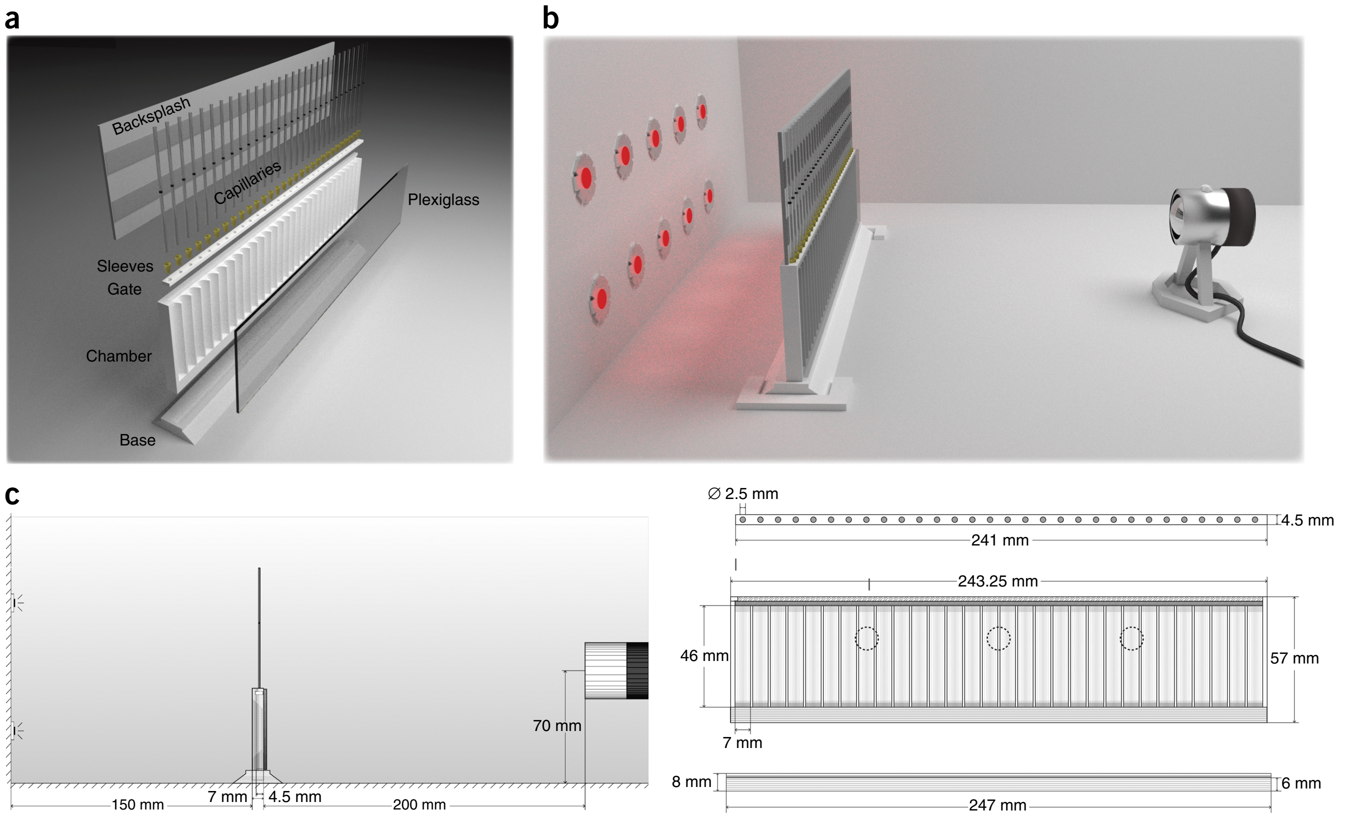


Figure 1 | ARC schematic. (**a**) Model of the disassembled ARC components includes the chamber, chamber base, front plexiglass panel, chamber gate, capillary holder sleeves, capillaries, and paper backsplash. (**b**) Model of the assembled ARC chamber within an incubator with a front-facing camera and IR LED backlighting. (**c**) Blueprints of the ARC setup, showing a side (left) and front (right) profile of the ARC, with corresponding measurements. The three dashed-line circles in the right-hand image represent positions for the placement of vibration motors on the back of the chamber.

Drosophila sleep experiments are often conducted at 22–25 °C at a relative humidity of ~60% under a 12/12-h light/dark cycle. ARC measurements are dependent only on IR light and are unaffected by any visible light source used to set the circadian rhythm of the flies. The ARC should be placed on a flat surface between a camera and IR backlighting. The camera should be facing the wells of the chamber, be ~200 mm away from the ARC, and have a full view of both the chamber wells and capillaries (**Fig. 1b**,**c**). The camera must have its filters replaced with an IR-pass filter. Alternatively, any USB camera that senses only IR light can be used.

To detect arousal threshold, an Arduino controller and motor attachment must be set up (**Fig. 2a**). This system allows the user to deliver an incremental series of vibrational stimuli to test the arousal threshold in addition to the detection of feeding and inactivity-based sleep. Notably, the stimulus delivery from the Arduino is pseudo-analog voltage, which can be used to control light, sound, or other stimuli. This allows flexibility in the type of arousal measured for various setups.

Assembly of the data capture software. The ARC controller can be run by most modern computers; older systems may exhibit lag or crashing. We recommend a Mac with ≥4 GB RAM. Use of the two-camera program generally requires ≥6 GB RAM. PCs are an acceptable option but must be running a Linux operating system. Performance, such as the rate of tracking or number of animals, scales with available hardware resources. Thus, increasing the number of cores and/or available RAM will increase experimental capabilities.

Support for the ARC requires installation of several open-source libraries as dependencies. The required steps for installing these packages are fairly simple and are listed in detail in the ARC installation blog (http://www.javagrinders-arc.blogspot.com/2017/01/install-open-source-tools-for-running.html). Specifically, the blog lists installation instructions for Java Standard Edition 8 SDK (https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html), Eclipse IDE 3.8+ for Java developers (http://www.eclipse.org/downloads/packages/), and MacPorts (https://www.macports.org/install.php), which is used to install OpenCV. The software versions listed have been tested with the current build. Although older versions may not function properly, the blog and provided software will be updated periodically to support future releases of Java and OpenCV.

The ARC acquires data from a video feed of the chamber and food capillaries (**Fig. 3a**). The image areas delineating the entire chamber and set of capillaries are defined by the user. These areas are divided into equally spaced tracks based on the user-defined number of flies or individual enclosures. For animal motion tracking, a reference image of the ARC is taken and used to background-subtract subsequent images in real time, providing an image of pixel differences. This difference should be largest in the image space containing the fly, assuming the animal has moved

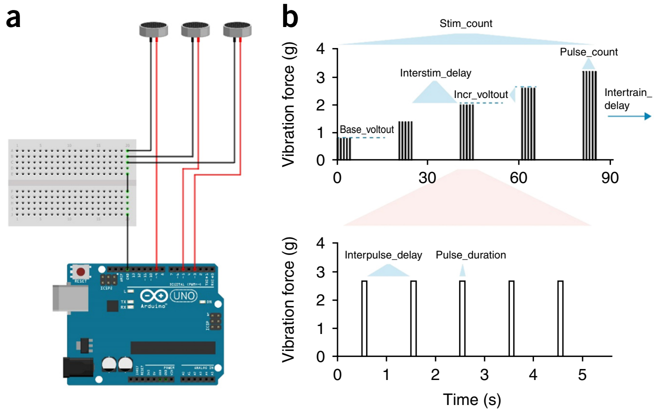
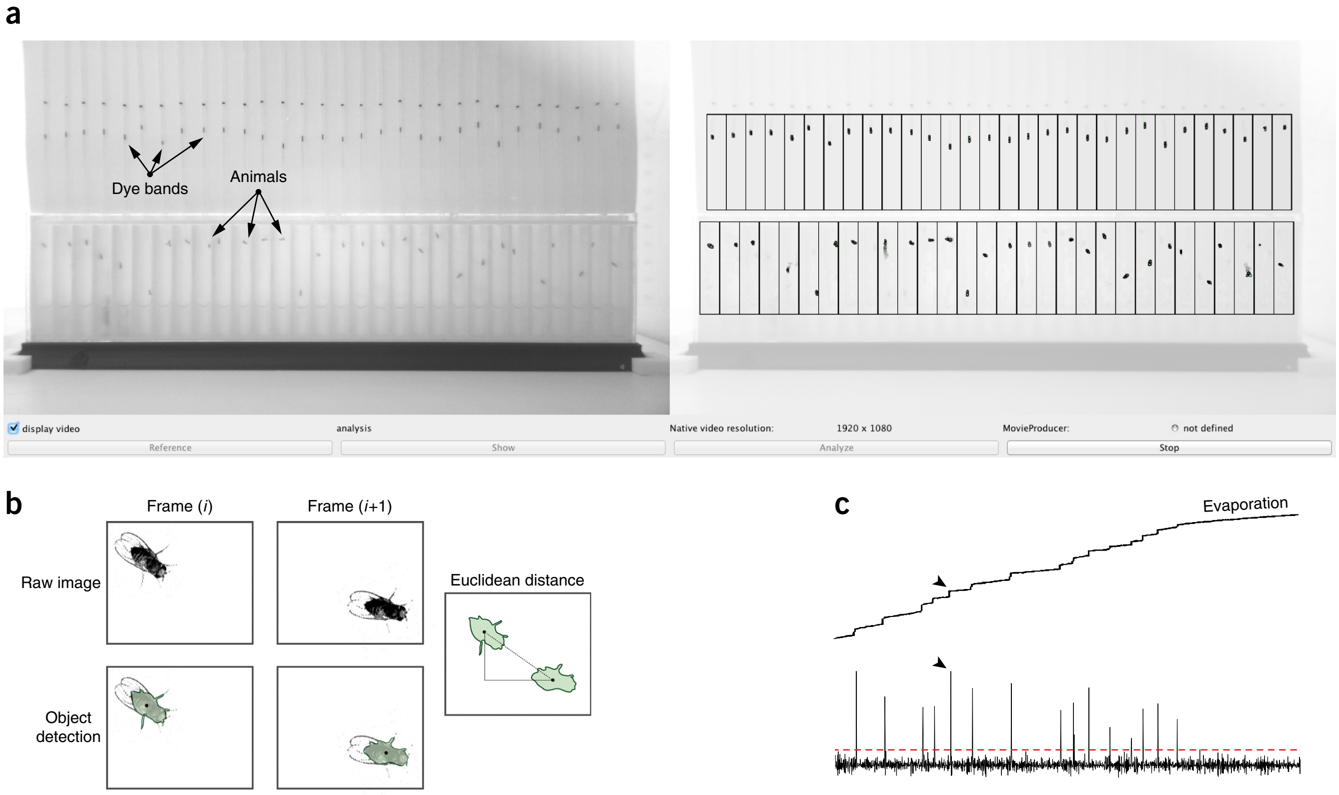


Figure 2 | Arousal threshold measurement components. (**a**) Wiring diagram of the shaftless vibration motor coupling to the Arduino microcontroller. (**b**) Plot of stimulus protocol, showing vibration force (*g*) delivered to the back of the ARC chamber over time. Variable names displayed can be altered in the ARCController.java document. **b** adapted with permission from ref. 31), Nature Publishing Group.

from its position in the reference image. The tracker then scans each fly track region for the point of greatest difference, and, if it is greater than the user-defined threshold, the scanner moves laterally one pixel at a time until a subthreshold pixel is reached. This threshold scanning continues in a clockwise direction until the outline returns to the starting coordinate (**Fig. 3a**,b). If the outlined polygon is roughly the size of a fly (also a user-defined parameter), its centroid coordinate is recorded to an output text file. During processing, animal motion is considered to be any Euclidean change in the coordinates that exceeds 50% of the animal's body length coordinate (**Fig. 3b**).

A similar algorithm is used for tracking the vertical position of the dye band marking the liquid-food meniscus. As the dye band does not necessarily move from its initial position in the reference image, meniscus tracking is performed using background estimation for each capillary track region. The tracker finds the darkest pixel for each track and then uses autodetection to outline the object. Any remaining pixels are averaged, with the average value serving as a flat background-subtraction value. The meniscus is then tracked using the same method used for animal tracking and is also recorded to the output file. To increase the signal-to-noise ratio, average dye-band positions are calculated from 1-min segments before analysis, although this period length can be changed for higher temporal resolution. In post-experiment data analysis (see below), feeding events are statistically identified as dye-band object motion events that are at least 4 s.d. greater than the estimated noise (**Fig. 3c**).

To track individual dye bands or animals, the parameters establishing the tracking regions within an image and the thresholds



☑ display video

analysis

Native video resolution: 1920 x 1080

MovieProducer: ○ not defined

Figure 3 | ARC data capture. (**a**) Image of the ARCController.java visual interface. On the left is a camera view of dye bands and animals. Black markings above dye bands are calibration marks and should be excluded from the tracking region. On the right is a visual overlay of user-defined tracking regions over the original image. Outlines of detected animals are displayed as black outlines with green fill. (**b**) Animal motion is calculated using Euclidean distance between coordinates of each frame. (**c**) 24-h trace of the dye band from one capillary using 1-min averages. Motion events are observed (black arrows mark an example) that exceed a statistical cutoff (dashed line), and are defined as meals. Evaporation is evident between meals as the slight continuous increase in baseline. **b** adapted with permission from ref. 4), eLife Sciences Publications (license: https://creativecommons.org/licenses/by/4.0/).

for determining an object must be set before the experiment. The tracking region for the animals is set to encompass the total image space in which any given animal can move. Because animal tracking uses a reference image of the chamber, it is relatively insensitive to dark edges or marks within the image. By contrast, the background estimation used in dye-band tracking requires that the dye be the only dark object in the tracking region. Calibrated 5-μl capillaries are prelabeled with a black mark near the middle of their length. Special attention must be given to loading the dye band below this mark and creating a tracking region that excludes the mark. Alternatively, the marks can be removed by rolling the capillaries in acetone on a hard surface or obtaining unmarked capillaries. Capillaries should be washed after mark removal or before each reuse, as described in the PROCEDURE.

Running the ARC. An important part of running a successful ARC experiment is controlling for environmental and developmental effects that might affect the behavior of the animals. General practices to develop a consistent experimental cohort of flies have been described previously[29]. Briefly, it is important that the animals develop without overcrowding, are housed in similar numbers after eclosion (we generally use seven to ten flies per vial), and are tested at the same age (we typically assess 5- to 9-d-old adults), and that different test groups are assigned to alternating (or random) compartments in the ARC chamber. This will control for possible effects of microenvironment, including uneven lighting or temperature. Both males and females can be used in the ARC. Egg laying and hatched larvae may be problematic in long-term experiments with mated females, however. Once animals are obtained, the experiments can be carried out by following the steps outlined in the PROCEDURE.

Diets can be composed of any water-soluble ingredients. A 2.5% (wt/vol) sucrose + 2.5% (wt/vol) yeast extract solution is often sufficient for 24 h of recording without running out of food[4,21,30]. Other components, including drugs, non-nutritive sweeteners, buffers, preservatives, or other nutritional ingredients can be used in the diet. Low-nutrient-concentration diets or other manipulations that result in increased food intake may require more frequent capillary changes. ARC runs should be stopped, capillaries replaced, and the ARC Controller program restarted at each food change. The ARC Controller program contains all the necessary parameters defined by the user for operating the ARC data acquisition, including the frame capture rate and the definition of the tracking regions (**Fig. 4**).

Before recording, flies are typically habituated to the ARC chamber for 20–24 h. The capillaries do not need to be loaded with the dye-labeled oil during habituation, as measurements are not typically made during this period. Habituation time can vary or be ignored to fit experimental needs. Depending on the incubator humidity, the agar at the bottom of each well dries out within 5–7 d, providing a practical time limit to long-term studies. If necessary, flies could be extracted from individual wells by mouth pipette and ARC measurements continued in another chamber with fresh agar.

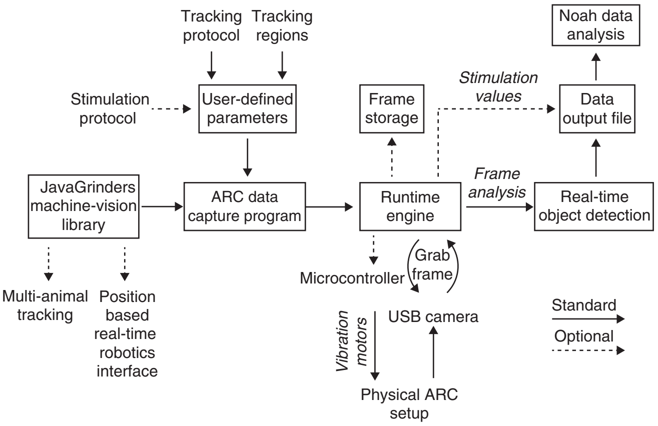


Figure 4 | Flow chart of ARC implementation of the JavaGrinders library. Illustration of instruction and data flow within the ARC. The ARC data capture program uses functions from the JavaGrinders machine-vision library to grab and analyze frames from a USB camera. Pixel coordinate data of animals and food menisci are recorded to a text file and can be analyzed for various behaviors using the Noah program. Vibrational stimuli can optionally be delivered and recorded to the text file.

Analyzing ARC data. Once the ARC is finished running, there will be a text file created by ARCController.java containing the coordinates of each tracked dye band and fly. The first column of the data set is a time stamp, followed by a *y* value for each dye band, and then finally an *x*–*y* pair for each animal track (**Supplementary Fig. 2**). If the `startArduinoStimulator();` is active, the output file will also contain a stimulus record as an additional column appended to the end of the file. This file can be directly loaded and processed by the Noah analysis program. After loading the file, many forms of analysis can be performed on the data set using intuitive user prompts and labeled output files. These include binning behaviors by a specified time, spatial mapping, statistical behavioral compilation, and arousal threshold analysis. Alternatively, the regular structure of this data output is well suited to the use of custom analysis software or porting into existing analysis software. A sample data set of 30 Canton-S males recorded over 2 d is included (**Supplementary Data**).

Level of expertise

Construction of the ARC is simple enough for most graduate or capable undergraduate students. Installation of the data capture and analysis software requires some familiarity with computers, particularly software installation, use of the terminal, and folder structures; the installation is straightforward when using the provided instructions. Executing an ARC experiment can be managed by an undergraduate student with minimal training.

MATERIALS

REAGENTS

- Flies (control strain or genotype of interest). We have used Canton-S for the data shown here. This strain was originally obtained from S. Benzer (California Institute of Technology) and has been maintained in our laboratory for >7 years.
- Dodecane ($CH_3(CH_2)_{10}CH_3$; Sigma-Aldrich, cat. no. 297879)
  ! CAUTION Dodecane can cause respiratory sensitization if inhaled.
- Light mineral oil (Sigma-Aldrich, cat. no. 330779)
- Copper (II) 5,9,14,18,23,27,32,36-octabutoxy-2,3-naphthalocyanine (Sigma-Aldrich, cat. no. 415286)

- Chloroform (Sigma-Aldrich, cat. no. C2432) **! CAUTION** Chloroform readily evaporates and can affect the CNS through inhalation, ingestion, or absorption through the skin. Use gloves when handling this substance. Work with this substance in a well-ventilated area or under a fume hood.
- Sucrose (Sigma-Aldrich, cat. no. 57-50-1)
- Bacto Yeast Extract (BD, cat. no. 212750)
- Agar (BD, cat. no. 214030)

EQUIPMENT

Software

- Python 3.4+ (https://www.python.org/downloads/)
- Noah.py (included in the **Supplementary Data**)
- Java Standard Edition 8 Development Kit (https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html)
- Eclipse IDE for Java (http://www.eclipse.org/downloads/packages/)

Computer

- Any modern Mac or PC (PC must be running Linux OS) with ≥4 GB RAM (≥6 GB RAM for two cameras) for data capture
- PC for data analysis
- Camera (Microsoft, model no. LifeCam 1080p (IR filter must be replaced with IR-pass filter) or any IR-sensitive USB camera)

3D print design files (all .stl files are included in the **Supplementary Data**)

- ARC_chamber.stl
- ARC_base.stl
- ARC_gate.stl
- ARC_cameramount.stl
- ARC_cornerbracket.stl
- ARC_chamberpiece1.stl and ARC_chamberpiece2.stl (if chamber is printed as halves)
- 5-μl Borosilicate glass capillaries (VWR, cat. no. 53432-706)
- Parafilm (Sigma-Aldrich, cat. no. P7793-1EA)
- Kimwipes (Fisher Scientific, cat. no. 06-666)
- Double-sided tape (Staples, cat. no. 504829)
- White construction paper (Staples, cat. no. 826292)
- 3D printer and filament (3D prints can alternatively be ordered online through various agents)
- IR 850-nm LED reel (Environmental Lights, product no. irrf850-390-reel)
- IR Polyester Pass Filter (B&H Photo, cat. no. LE87C44)
- 200-μl Pipette tips (VWR, cat. no. 53503-288)
- 0.2-μm Cellulose acetate syringe filter (VWR, cat. no. 28145-477)

Optional equipment for arousal threshold analysis

- Arduino Uno microcontroller (Adafruit Industries, product ID 50)
- Shaftless vibrating motors (Precision Microdrives, model no. 312-110)
- USB A/B-type cable (Staples, cat. no. 2094913)

REAGENT SETUP

Meniscus-labeling dye To make the meniscus-marking dye, prepare 4 ml of light mineral oil/dodecane solution at a ratio of 3:1. Add 10 mg of the copper dye reagent and vortex the solution thoroughly. Centrifuge the solution at 900*g* for 5 s at room temperature (22 °C) and collect the supernatant for use. Store the meniscus-labeling dye at room temperature away from light and heat for up to 2 years.

Liquid diet preparation For a standard liquid diet, dissolve 2.5 g of sucrose and 2.5 g of Bacto yeast extract in dH_2O to a final volume of 100 ml. A hot plate with a magnetic stirrer can be used. After cooling, filter-sterilize the solution using a 0.2-μm cellulose acetate syringe filter. Aliquots can be stored at –20 °C for at least 1 year. Preparation may vary depending on diet.

Agar preparation Combine 1 g of agar with 100 ml of dH_2O. Dissolve the agar completely by boiling it on a hot plate with a magnetic stirrer. After removing the agar from the hot plate, wait until the agar is warm to the touch (~50 °C) before adding it to the ARC chamber, as described below. Use freshly prepared agar for each experiment to avoid microbial contamination.

EQUIPMENT SETUP

Printing chamber Parts can be printed using either ABS or PLA filament. **▲ CRITICAL** Different materials may affect fly climbing or other behaviors. If flies have trouble climbing PLA, print chambers in ABS. If printing of the chamber consistently leads to edge-warping, print the two pieces separately (ARC_chamberpiece1.stl and ARC_chamberpiece2.stl in the **Supplementary Data**) and fuse post print using chloroform.

Construction of the chamber Cut a piece of 2-mm-thick acrylic plexiglass to 50 × 244 mm, or large enough to cover the well and gate region of the chamber while leaving the solid bottom of the chamber exposed for insertion into the base. Clamp the plexiglass to the chamber and apply a small amount of chloroform to all edges where the two materials meet. Be careful not to use too much chloroform. A proper amount will draw in between the surfaces, whereas an excess will spill onto the glass and create a cloudy view into the chamber. Allow the materials to fuse for at least 1 h in a well-ventilated area or fume hood. The gate should be sanded gently until it can be moved in and out of the chamber sliding port without excessive force. To make sleeves to hold the capillaries in the gate, cut the narrow end of a 200-μl pipette tip with a razor blade until a capillary can be fit through it using a small amount of force. Then cut the wide end so that the section of the pipette tip is ~5 mm in length. Capillaries should not be able to slide freely through the sleeve.

Preparation of the recording environment If the camera used is not already IR sensitive, it must be converted by opening the housing and replacing the IR filter with an IR-pass filter. The IR filter, which can be removed with a razor blade, is the small glass lens with a shiny green tint found close to the actual sensor. For the pass filter, a lens-shaped piece of polyester IR-pass sheet is cut and fit over the exterior glass lens of the camera housing.

The IR light source should be placed ~150 mm behind the chamber (**Fig. 1c**). Ideally, the lighting should be placed on the metal surface of the incubator wall or some other material that can serve as a heat sink. Alternatively, a fan can be placed perpendicular to the chamber to prevent local differences in temperature. Although the number and distribution of IR sources may vary depending on the brand of LEDs and camera used, IR backlighting should be evenly distributed across the chamber and sufficiently intense to easily distinguish flies and dye bands from their respective backgrounds. We suggest several parallel strips of 850-nm IR lighting reel (Environmental Lights, see 'Materials') of a width equal to that of the chamber. The camera should be mounted in the printed camera mount and secured to the incubator ~200 mm from the camera mount. If the lighting and filtering have been set up correctly, the camera should show a dark image under normal (visible) light when the IR lights are off.

Once the position of the camera and LED lighting are set, the chamber position must be determined. To check this, view a live image on the computer screen by using native webcam software or the ARCController software (described below). While viewing the live image, place the chamber between the camera and the IR lighting such that (i) the entire chamber and at least the liquid-filled section of the capillaries just fit in the camera view and (ii) the edges of the chamber are square in the image. Once these conditions are met, mark the edges of the chamber base. Remove the chamber and, in place of the markings, secure two corner brackets to ensure consistent chamber placement (**Fig. 1b**).

Construction of the stimulus delivery system To use the ARC for stimulus response measurements, a microcontroller setup must be included. Notably, this setup is optional and is not required for normal motion-based sleep or feeding measurements. For the default stimulus response setup, attach an Arduino Uno to the data capture computer using a USB A/B-type cable. On the Arduino, load the jArduino firmware (included with the **Supplementary Data** or on GitHub, https://www.github.com/SINTEF-9012/JArduino; instructions are found in the readme.txt file). Notably, the GitHub source contains a bug that may prevent loading to the board. The bug is fixed in the version provided in the **Supplementary Data**, use of which is recommended. Attach one wire of a Precision Microdrives shaftless vibration motor to a PWM pin on the board. These will be marked by a "~" symbol and are usually pins 3, 5, 6, 9, 10, and 11, but they may vary on different board layouts. Finally, connect the vibrator's ground wires to the Arduino's ground (GND) pin (**Fig. 2a**). The parameters passed to the Arduino for delivering stimuli (**Fig. 2b**) are listed in the ARC runtime program and are based on prior work[31].

Vibration motor attachment Although there are many ways in which the vibration motors can be temporarily attached to the chambers, we found that a small dot of hot glue applied by a glue gun worked well and was easily removed before washing the chambers. An example setup using three motors secured to the chamber is shown (**Fig. 1c**).

Software installation To install the necessary open-source libraries as dependencies, follow the instructions on the ARC blog (http://www.javagrinders-arc.blogspot.com/2017/01/install-open-source-tools-for-running.html). Once the necessary components are installed, the ARC data-capture program, ARCController.java, can be run within an integrated developer environment (IDE). The Eclipse IDE allows users to access and change the ARCController settings and optimize tracking conditions. To set up the IDE, open the Eclipse application and, when prompted, choose a default workspace (typically in the Documents folder). Next, copy the JavaGrinders_ARC folder (from the **Supplementary Data** or download and unzip it from the ARC installation blog) into the Eclipse workspace folder. In Eclipse, open the package explorer by selecting the menu item *Window* → *Show View Package Explorer*. From the *File* menu, select *New* → *Java Project*. In the *Java Project Wizard*, enter the *Project name* '`JavaGrinders_ARC`' and click *Finish*. If the ARC project is successfully imported into the package explorer, navigate to *JavaGrinders_ARC* → *src* →*_ARC* and select *ARCController.java*. Select the menu item *Navigate* → *Open* to access all of the necessary parameters for operating the ARC data acquisition program, including the frame capture rate, definitions for the tracking regions, and the stimulus protocol. Default parameters are provided and their optimized settings are described below. To test the camera, select the menu item *Run* → *Run*. If the program finds a video camera, it will start the tracker and open the main window.

Setting and optimizing tracking parameters To define tracking regions for the animals and the capillaries, coordinates must be determined. The program will take a given user-defined rectangle and subdivide it into an equal number of tracks, defined by the value of `kNAnimals` (30 for the standard ARC chamber). To obtain coordinates for each rectangle, first place an ARC chamber in the incubator within the mounting brackets. Next, run ARCController.java. After defining the output file name and location, a pane with two windows will appear. The left is the live image from the camera and the right is an analysis window showing object-tracking data overlaid on the image. Click *Reference* to capture a reference image of your chamber. The reference button should be used only to find appropriate parameters and not during a run, as it can modify the adaptive background determination performed for the food tracking. At the top of the *Tracker* window, select *Objects* → *Screen info* → *Rectangle*. A single rectangle can now be drawn by clicking and dragging on the live image. Once you have drawn the rectangle, which should define the outer edges of the entire area containing the fly enclosures, the following coordinates will be displayed in the Eclipse console window as *new Rectangle* (*ARCAnimalLeft*, *ARCAnimalTop*, *ARCAnimalWidth*, *ARCAnimalHeight*). These four parameters should be entered and saved into the ARCController.java program. Once these parameters are set, close the current tracker window and rerun ARCController.java to check that your defined tracking window aligns well with the chamber.

For tracking of the capillary dye bands, use the same method for defining the fly enclosures to define the capillary-tracking region. Importantly, this region should include as much of the food-containing segment of the capillaries as possible while excluding any existing markings on the capillaries and parts of the chamber that are darker than the dye band. An example of how the chamber and dye-tracking regions should look is shown (**Fig. 3a**). The parameters derived from the rectangle drawn around the capillaries will be displayed in the Eclipse console window as *new Rectangle* (*ARCFoodLeft*, *ARCFoodTop*, *ARCFoodWidth*, *ARCFoodHeight*). As for the fly tracking, these four parameters should be entered and saved into ARCController.java. Once these parameters are set, again close and rerun ARCController.java to check that the animal- and capillary-tracking windows align well with the chamber.

The variables used by the program to identify flies are `kAnimPxThresh`, which defines the threshold pixel difference for outlining objects; `kAnimObjSize`, which creates a range of acceptable object size; and `kAnimIsDkObj`, which tells the program whether the fly is a dark object on a light background (`true`) or vice versa (`false`). Lowering the `kAnimPxThresh` value will track objects that have less contrast from their background, but it may also increase false-positive tracks. If an animal is not within the acceptable `kAnimObjSize`, the tracker records a missed read. To optimize `kAnimPxThresh`, place an ARC loaded with animals into the brackets and run ARCController.java. Click *Objects* → *Get Pixel Value* and click on the darkest region of the background within your animal track region. Record the pixel value written to the console. Next, find a stationary animal and record the pixel value of its center. Subtracting the animal pixel value from the background pixel value will give you an approximation of `kAnimPxThresh`. Change the value and restart ARCController.java. Click *Reference* and then *Show*. If the red representations of the animals in the analysis window are smaller than the animals, decrease the `kAnimPxThresh` value, and vice versa. Repeat until the analysis window shows the correct size for the animals. To determine an appropriate animal size (`kAnimObjSize`), click *Objects* → *Screen Info* → *Rectangle*, draw a rectangle around a fly, and use the area output written to the console.

Once all parameters are set and saved, rerun ARCController.java and only click *Show*. If the flies and dye bands are faithfully represented in red color on the right panel of the tracking window, click *Analyze*. Data are now being saved to the output file in real time. The animals and dye bands should then be identified as green polygons in the analysis window (**Fig. 3a**). Notably, JavaGrinders uses background subtraction, so if a fly has not moved from its position in the reference image, then it will not show up immediately in the right panel. This is not an issue, as the Noah analysis software will backfill initial empty coordinates with the first identified position. The capillaries use adaptive thresholding and should all show up immediately. The program will then collect data until *Stop* is clicked or the user-defined recording duration is reached.

To measure arousal threshold, the stimulus protocol must be turned on. In ARCController.java, find the line containing the phrase '`//startArduinoStimulator();`'. Delete the '`//`' at the beginning of the line to activate the stimulus protocol. ARCController contains default stimulus protocol values that have been successfully used previously[4]. However, the specific stimuli delivery can be tailored to individual experimental setups. The default parameters deliver a stimulus protocol once every hour (**Fig. 2b**). To modify this protocol, edit the values in ARCController.java found immediately below the line that reads '`public static void runArduinoStimulator() {`'.

Although the ARC is designed to eliminate the need for video/image storage, an option for saving frames at a given rate is included. To use this, remove the '`//`' at the beginning of the following lines in ARCController.java.

```
  //ImgProducer theProducer = new
ImgProducer(null,theARCTracker.nativeWidth,theARCTr
acker.nativeHeight,60000);
  //theProducer.getItsRenderer().setPlotTargetStyle
(FrameRenderer.TargetStyle_RAWIMAGE);
  //theARCTracker.setVideoProducer(theProducer);
```

To save images of only the capillaries, also remove the '`//`' at the beginning of '`// theProducer.setSubRec(kFoodArea);`'.

The value of 60,000 (highlighted in gray above) denotes the time in milliseconds between the storage of images. Notably, storing these images is processor-intensive and can increase the risk of older machines crashing.

Installing data analysis software Download Python 3.x on a PC (currently Noah.py runs only on the PC version of Python) in order to run Noah.py. When running Noah for data analysis, it will prompt for the entry of several values in regard to the tracking conditions. First, it will request the number of pixels per centimeter in the tracking setup. To obtain this value, measure the width of the chamber in centimeters (using a ruler) and in pixels using the *Rectangle* tool in the ARCController.java program, as described above (the third parameter provides the user-drawn rectangle width in pixels). Use these numbers to derive the pixels per centimeter value. The length of a fly is assumed to be ~0.3 cm for filtering motion events less than half a body length, which has been found to be appropriate for video-based fly motion tracking[32]. A calibration preset in Noah determines the volume of food consumed by using a centimeter-to-microliter conversion based on the recommended 5-μl capillaries. Both the fly length and conversion factor can be changed in the constants section of Noah.py, which is found in the first 60 lines of the script. Python scripts can be edited by right-clicking the program icon and selecting *Edit with IDLE*. The conversion factor for capillaries can be determined by loading several capillaries with 5 μl of water, measuring the liquid height and calculating the average centimeter-to-microliter value.

PROCEDURE

Chamber setup ● TIMING ~20 min

▲ CRITICAL See **Supplementary Video 1** for further details on how to perform these steps.

1| Boil the agar (Reagent Setup) and allow it to cool until the flask is warm to the touch.

2| Using a 1,000-μl pipettor, pipette 300 μl of 1% (wt/vol) agar into each fly compartment. Angle the pipettor against the back wall (not the transparent front cover) during pipetting. Allow ≥2 h to cool and solidify. Just before the experiment, slide a chamber gate into place, using a small dot of hot glue to secure it if the gate feels loose.

Loading of animals for habituation ● TIMING ~40 min

▲ CRITICAL See **Supplementary Video 1** for further details on how to perform these steps.

3| Thaw a liquid diet and fill the appropriate number of capillaries (30 per ARC chamber) with habituation diet. Wipe away excess fluid on the exterior of each capillary. Capillaries can be filled and maintained on a 'capillary-holding box', which is made by wrapping a small plastic container, such as a 1,000-μl pipette tip container, with a strip of double-sided tape. Capillaries are filled approximately halfway to two-thirds full.

4| Using a mouth aspirator, gently aspirate a fly through the gate hole for each compartment. Quickly place the gate sleeve into the hole, followed by a habituation capillary to block the fly from exiting. Repeat until all wells are loaded.

5| Attach capillary backsplash to the capillaries. If the experiment requires arousal threshold measurement, attach shaftless vibration motors by applying a small dot of hot glue to the back of the motors and then pressing against the chamber (these can be easily removed with a flat metal object after the experiment).

6| Place the chamber in the recording incubator inside brackets and allow the flies to habituate before preparing dye-labeled capillaries. Flies are typically habituated under these conditions for 20–24 h, but this can be changed according to the experiment.

Preparation, loading, and changing of dye-labeled capillaries and running of the experiment ● TIMING ~45 min

▲ CRITICAL See **Supplementary Video 1** for further details on how to perform these steps.

7| Place ~40 μl of the marking dye on a small piece of Parafilm. Quickly touch the tip of a capillary to the dye solution, allowing a small amount to load. Gently blot the capillary tip on a Kimwipe until the dye band is ~2 mm in height. Quickly place the same capillary end into liquid diet and allow the dye band to rise until it is about halfway to two-thirds up the capillary or ~5 mm below any external marks that might obscure the meniscus.
▲ CRITICAL STEP If the dye band is visually split (e.g., because of an air bubble), discard the capillary. This split can cause major tracking issues. Maintain the capillaries on a holding box (Step 3) and repeat capillary filling until the needed number of capillaries are made.
? TROUBLESHOOTING

8| Take the chamber out of the recording incubator and swap out all habituation capillaries with dye-band-labeled capillaries. The backsplash should be kept on during this step.
▲ CRITICAL STEP Do not remove more than three capillaries at any time. This will ensure that the backsplash stays in place and that animals do not escape through the sleeves. Make sure the capillaries are relatively straight and at the same height. Return the chamber to the recording incubator.

9| Open ARCController.java in the Eclipse compiler. If you are not running an arousal threshold experiment, comment the line containing '`StartArduinoStimulator();`' by placing '`//`' at the start of the line. Run ARCController.java by selecting *Run* from the *Run* menu or by pressing the green *play* icon at the top of the eclipse window. Name your output file.
▲ CRITICAL STEP If you name an output file with a name already used in the environment, the file will be overwritten.
If you have not already set the tracking regions and optimized the tracking parameters, refer to the 'Setting and optimizing tracking parameters' section under Equipment Setup. If you have, proceed to the next step.
? TROUBLESHOOTING

10| Click on *Show*. If the tracking-analysis window outlines the animals and dye bands well, click on *Analyze*. If not, continue to optimize the tracking parameters until tracking is accurate. Stop the software when the experiment is over,

or stop and start again for capillary changes (typically once per day). Use a different output file name for each continuous period of data collection.
? TROUBLESHOOTING

11| When the experiment is done, disassemble the chamber. Remove the sleeves and soak them in 70% (vol/vol) ethanol for at least 1 h before rinsing with dH$_2$O and drying.

Cleaning of the chamber and capillaries ● TIMING ~10 min
12| Use a water pick to remove the agar from the wells. Scrub each well with a pipe cleaner using unscented soapy water, followed by a thorough rinse. Allow the chambers to dry and store them until the next experiment. If they are going to be reused, capillaries can be washed by flushing them under running dH$_2$O to clear the remaining liquid, boiling them in dH$_2$O for 30 min, rinsing with dH$_2$O, and then using a vacuum system to suck the capillaries dry. However, this may lead to dye streaking if done improperly. Chambers and capillaries can be stored indefinitely at room temperature in covered containers.

13| Rinse the sleeves with dH$_2$O and allow them to dry. Store them indefinitely in a clean container at room temperature until the next experiment.
■ PAUSE POINT Data can be processed at a later time point if preferred.

Processing of the ARC file ● TIMING ~2 min
14| Move the file generated by the ARC to a PC running Python. Run *Noah.py* and select option 1 by typing `1` and pressing enter. Click and drag the text file onto the *Noah.py* window. The address of the file will then appear. If it does not, type the address of the file manually (e.g., `C:\Users\UserName\Desktop\outputfile.txt`) and press enter.

15| Feeding, tracking, and stimulus delivery data are now ready to be used by Noah for various analyses. The software is prompt-based, in which the program indicates the desired input at each stage. To use any analysis feature, type the number given in the menu for the corresponding analysis feature. For instance, to obtain feeding data analyzed using specific temporal bins, select option 5. Input the desired bin size, following the prompt *Enter Bin Size(mins):*. You will then have the option to similarly analyze meal frequency, meal size, and satiety by typing ‘`y`’ in response to each request (e.g., *Add bins for frequency (y/n):*). The file will then automatically be written to the environment in which the program is in as a comma-separated values sheet. The output file name is automatically tied to the original file name (e.g., ‘outputfile.txt’ yields feedBins_binSize_outputfile.csv). All other analyses work in a similar manner.

? TROUBLESHOOTING
Troubleshooting advice can be found in **Table 1.**

TABLE 1 | Troubleshooting table.

Step	Problem	Possible reason	Solution

Equipment Setup	jArduino firmware will not load to the board and generates the following error message:
	error: no matching function for call to ’JArduino::init_JArduino()’
		The serial connection is not synchronized between the computer and Arduino	Go into the JArduinoFirmware.ino file and replace the following line
			‘{
			//initialize the JArduino protocol
			_JArduino.init_JArduino();
			}’
			with
			‘void setup(){
			//initialize the JArduino protocol
			uint16_t baud = 9600;
			_JArduino.init_JArduino(baud);
			}’

7	Meniscus-labeling dye bands are frequently splitting when loading capillaries	Excess time between loading the dye band and liquid food allows a bubble to form at the base of the capillary	Load capillary more quickly or increase the percentage of dodecane in the dye solution to prevent the dye from climbing walls of capillaries

(continued)

TABLE 1 | Troubleshooting table (continued).

Step	Problem	Possible reason	Solution
9	ARCTracker failed to connect to the USB camera at startup with error message: 'java.lang.Exception: FrameCapture Exception in OpenCVRecordProc: no valid frame during initialization'	The USB camera is being used by another program from OpenCV	Unplug the USB camera, reboot the computer, and plug the USB camera back in
10	Meniscus-labeling dye bands are streaking (or disappearing) during the experiment	There is not enough surfactant to keep the dye band from sticking to the wall of the capillary	Increase the percentage of dodecane in the dye solution and make sure the dye is centrifuged thoroughly to remove nondissolved particulates
	The liquid food evaporates too quickly	There is low humidity in the chamber or the gate does not seal the chamber well	Either create a tighter fit between the gate and the chamber by using a new gate that is slightly larger, increase the humidity in the incubator housing the ARC, or lower the incubator temperature to within an acceptable experimental range
	Liquid diet is totally consumed before 24 h	Some diets or manipulations will cause this to happen because the fly is too hungry or because the evaporation rate is too high	Stop ARCTracker and change the capillaries at mid-day, then restart ARCTracker. It may be best to do this switch at Zeitgeber time = 12 h, when the animals are already quite active, to limit disruption of any natural quiescence
	Animal tracking has a high error rate	This could result from several optical traits within the ARC system, but it is most likely caused by poor contrast between the chamber and the flies	Increase backlighting by using more IR LEDs. If this is not an option, the tracking parameters can be altered to limit the number of false tracks obtained. To do this, run the ARCTracker program with incremental increases or decreases in kAnimPxThresh. Using the *Show* button can verify the accuracy of your tracking parameters
	Dye-band tracking has a high error rate	Backlighting is insufficient or there is another object in the dye-band tracking space	Add lighting or change the USB camera settings until the lighting provides strong contrast between the dye bands and the background. Search the image carefully for any specks or discolorations on the capillary backsplash paper. Make sure that the capillary markings are outside of the defined track areas
	Selecting *Show* outlines objects clearly in the analysis window, but objects are not being tracked after starting the analysis	This is likely because objects are being identified, but are not close enough to the specified size of the object	Check the approximate size in pixels of the dye bands and animals. Change the kAnimObjSize or kFoodObjSize values to reflect the size of the animals or dye bands

● TIMING

Steps 1 and 2, chamber setup: ~20 min
Steps 3–6, loading of animals for habituation: ~40 min
Steps 7–11, preparation, loading, and changing of dye-labeled capillaries and running of the experiment: ~45 min
Steps 12 and 13, cleaning of the chamber and capillaries: ~10 min
Steps 14 and 15, processing of the ARC file: ~2 min

ANTICIPATED RESULTS

Data obtained from a standard ARC experiment can be compared with the sample data set that used 2.5% (wt/vol) sucrose + 2.5% (wt/vol) yeast extract food (**Fig. 5**). Circadian profiles should show heightened activity and reduced sleep near Zeitgeber times 0 and 12 h (when the visible light source turns on and off, respectively). Rhythmicity should be evident in individuals and in averaged cohort data (**Fig. 5a**,**b**). Feeding should also show rhythmicity[33]; however, the peak timing and intensity can vary

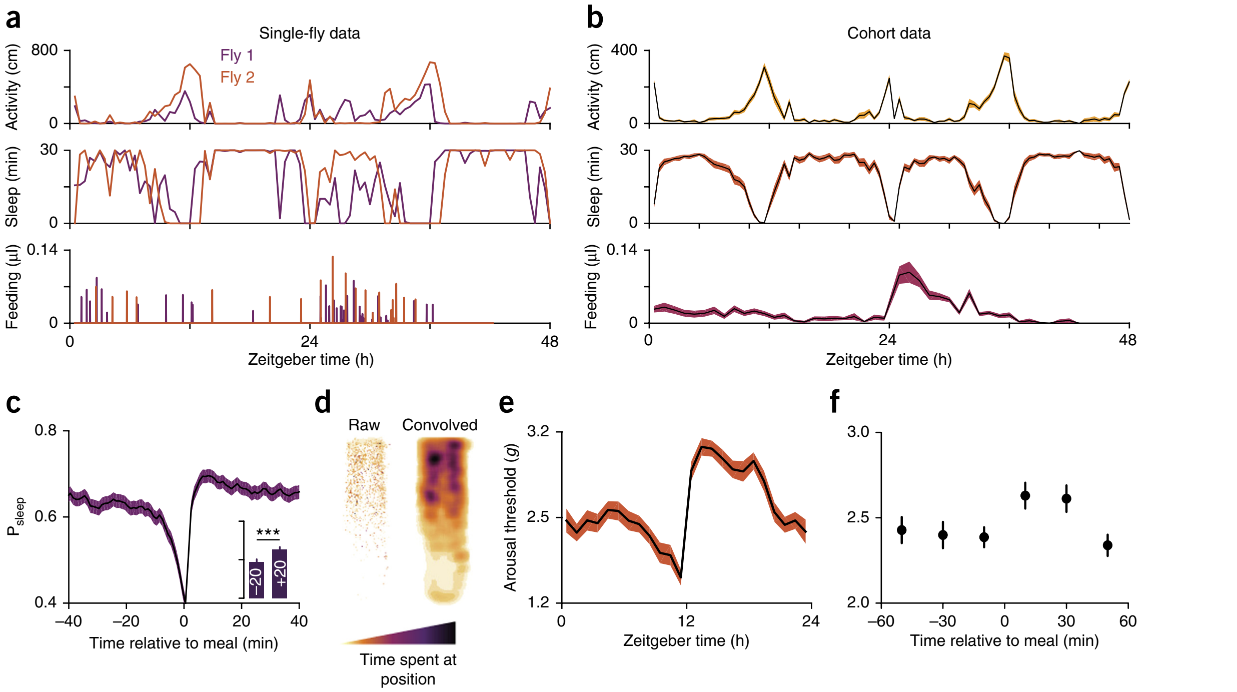


Figure 5 | Expected data from the ARC. (**a**) Individual activity, sleep, and feeding profiles for two individuals over a 48-h period. (**b**) Activity, sleep, and feeding averaged from a cohort (n = 30 flies; shaded line represents mean ± s.e.m.). (**c**) Postprandial sleep plot of individual meals from a cohort. Inset graph shows average P_{sleep} (± s.e.m.) in the 20 min before (−20) and after (+20) meals (axis scaled to the parent graph). n = 486 events from 30 flies; ***P < 0.001, Wilcoxon matched-pairs sign rank test. (**d**) Heat map of fly location over the 48-h period. n = 30 flies. (**e**) Circadian rhythm of arousal threshold has a similar profile to sleep. (**f**) Animal stimulus response at 0–20, 20–40, and 40–60 min before and after meals. Arousal events are filtered for prior time inactive to control for sleep depth (0–5 min shown; minimum 1 s inactivity). Circles represent mean ± s.e.m.; n = 316 meals from 60 flies. All flies were Canton-S males.

substantially depending on the diet, genotype, or the amount of time that the animals have been habituated in the chamber. Integration of sleep and feeding should show an increase in sleep after meals (**Fig. 5c**); however, the intensity of postprandial sleep is also dependent on the diet used[4]. Analysis of postprandial sleep typically uses individual meals rather than animals to capture the variability in meal size and nutrient consumption within individuals[4]. These data are usually not normally distributed. Inter-fly comparisons using individual animal averages are often normally distributed, however, gross trends are nearly indistinguishable when using individual meals or animals. Spatial analysis reveals that flies spend more time near the food capillary at the top of the chamber than the agar at the bottom (**Fig. 5d**). Unhealthy flies or animals under other manipulations may show deviations from this pattern. If the experiment contains arousal threshold measurements, these values should share a similar profile to that of sleep (**Fig. 5e**). Furthermore, arousal thresholds surrounding meals should show that consumption temporarily raises arousal threshold (**Fig. 5f**). Poor data quality or unhealthy animals may result in less obvious trends in circadian behavior, spatial tendency, or postprandial sleep; however, there is no consensus on how health affects these behaviors.

Note: Any Supplementary Information and Source Data files are available in the online version of the paper.

ACKNOWLEDGMENTS We thank J. Jacobs and A. Sehgal for comments on the manuscript. This work was funded by the National Institutes of Health (R21DK092735 to W.W.J.).

AUTHOR CONTRIBUTIONS K.R.M., J.H.P., R.H., and W.W.J. contributed to the conception, development, and testing of the ARC; K.R.M., R.H., and W.W.J. wrote the manuscript; K.R.M., J.H.P., R.H., and W.W.J. revised the manuscript.

COMPETING FINANCIAL INTERESTS The authors declare no competing financial interests.

1. MacFadyen, U.M., Oswald, I. & Lewis, S.A. Starvation and human slow-wave sleep. *J. Appl. Physiol.* **35**, 391–394 (1973).
2. Dario, A.J., Lopes, P.R., Freitas, C.G., Paschoalini, M.A. & Marino-Neto, J. Electrographic patterns of postprandial sleep after food deprivation or intraventricular adrenaline injections in pigeons. *Brain Res. Bull.* **39**, 249–254 (1996).
3. Keene, A.C. *et al.* Clock and cycle limit starvation-induced sleep loss in *Drosophila*. *Curr. Biol.* **20**, 1209–1215 (2010).
4. Murphy, K.R. *et al.* Postprandial sleep mechanics in *Drosophila*. *eLife* **5**, e19334 (2016).
5. Linford, N.J., Chan, T.P. & Pletcher, S.D. Re-patterning sleep architecture in *Drosophila* through gustatory perception and nutritional quality. *PLoS Genet.* **8**, e1002668 (2012).
6. Catterson, J.H. *et al.* Dietary modulation of *Drosophila* sleep-wake behaviour. *PLoS ONE* **5**, e12062 (2010).
7. Wells, A.S., Read, N.W., Uvnas-Moberg, K. & Alster, P. Influences of fat and carbohydrate on postprandial sleepiness, mood, and hormones. *Physiol. Behav.* **61**, 679–686 (1997).

8. Landstrom, U., Knutsson, A., Lennernas, M. & Soderberg, L. Laboratory studies of the effects of carbohydrate consumption on wakefulness. *Nutr. Health* **13**, 213–225 (2000).
9. Grandner, M.A., Kripke, D.F., Naidoo, N. & Langer, R.D. Relationships among dietary nutrients and subjective sleep, objective sleep, and napping in women. *Sleep Med.* **11**, 180–184 (2010).
10. Koban, M., Sita, L.V., Le, W.W. & Hoffman, G.E. Sleep deprivation of rats: the hyperphagic response is real. *Sleep* **31**, 927–933 (2008).
11. Spiegel, K., Tasali, E., Penev, P. & Van Cauter, E. Brief communication: sleep curtailment in healthy young men is associated with decreased leptin levels, elevated ghrelin levels, and increased hunger and appetite. *Ann. Intern. Med.* **141**, 846–850 (2004).
12. Cirelli, C. & Bushey, D. Sleep and wakefulness in *Drosophila melanogaster*. *Ann. N. Y. Acad. Sci.* **1129**, 323–329 (2008).
13. Artiushin, G. & Sehgal, A. The *Drosophila* circuitry of sleep-wake regulation. *Curr. Opin. Neurobiol.* **44**, 243–250 (2017).
14. Greenspan, R.J., Tononi, G., Cirelli, C. & Shaw, P.J. Sleep and the fruit fly. *Trends Neurosci.* **24**, 142–145 (2001).
15. Shaw, P.J., Cirelli, C., Greenspan, R.J. & Tononi, G. Correlates of sleep and waking in *Drosophila melanogaster*. *Science* **287**, 1834–1837 (2000).
16. Hendricks, J.C. *et al.* Rest in *Drosophila* is a sleep-like state. *Neuron* **25**, 129–138 (2000).
17. van Alphen, B., Yap, M.H., Kirszenblat, L., Kottler, B. & van Swinderen, B. A dynamic deep sleep stage in *Drosophila*. *J. Neurosci.* **33**, 6917–6927 (2013).
18. Donlea, J.M., Thimgan, M.S., Suzuki, Y., Gottschalk, L. & Shaw, P.J. Inducing sleep by remote control facilitates memory consolidation in *Drosophila*. *Science* **332**, 1571–1576 (2011).
19. Shaw, P.J., Tononi, G., Greenspan, R.J. & Robinson, D.F. Stress response genes protect against lethal effects of sleep deprivation in *Drosophila*. *Nature* **417**, 287–291 (2002).
20. Berry, J.A., Cervantes-Sandoval, I., Chakraborty, M. & Davis, R.L. Sleep facilitates memory by blocking dopamine neuron-mediated forgetting. *Cell* **161**, 1656–1667 (2015).
21. Deshpande, S.A. *et al.* Quantifying *Drosophila* food intake: comparative analysis of current methodology. *Nat. Methods* **11**, 535–540 (2014).
22. Yapici, N., Cohn, R., Schusterreiter, C., Ruta, V. & Vosshall, L.B. A taste circuit that regulates ingestion by integrating food and hunger signals. *Cell* **165**, 715–729 (2016).
23. Ro, J., Harvanek, Z.M. & Pletcher, S.D. FLIC: high-throughput, continuous analysis of feeding behaviors in *Drosophila*. *PLoS ONE* **9**, e101107 (2014).
24. Itskov, P.M. *et al.* Automated monitoring and quantitative analysis of feeding behaviour in *Drosophila*. *Nat. Commun.* **5**, 4560 (2014).
25. Bradski, G. The OpenCV library. *Doctor Dobbs J.* **25**, 120–126 (2000).
26. 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).
27. Pfeiffenberger, C., Lear, B.C., Keegan, K.P. & Allada, R. Locomotor activity level monitoring using the Drosophila Activity Monitoring (DAM) System. *Cold Spring Harb. Protoc.* http://dx.doi.org/10.1101/pdb.prot5518 (2010).
28. Ja, W.W. *et al.* Water- and nutrient-dependent effects of dietary restriction on *Drosophila* lifespan. *Proc. Natl. Acad. Sci. USA* **106**, 18633–18637 (2009).
29. Gilestro, G.F. Video tracking and analysis of sleep in *Drosophila melanogaster*. *Nat. Protoc.* **7**, 995–1007 (2012).
30. Ja, W.W. *et al.* Prandiology of *Drosophila* and the CAFE assay. *Proc. Natl. Acad. Sci. USA* **104**, 8253–8256 (2007).
31. Faville, R., Kottler, B., Goodhill, G.J., Shaw, P.J. & van Swinderen, B. How deeply does your mutant sleep? Probing arousal to better understand sleep defects in *Drosophila*. *Sci. Rep.* **5**, 8454 (2015).
32. Donelson, N.C. *et al.* High-resolution positional tracking for long-term analysis of *Drosophila* sleep and locomotion using the 'tracker' program. *PLoS ONE* **7**, e37250 (2012).
33. Xu, K., Zheng, X. & Sehgal, A. Regulation of feeding and metabolism by neuronal and peripheral clocks in *Drosophila*. *Cell Metab.* **8**, 289–300 (2008).