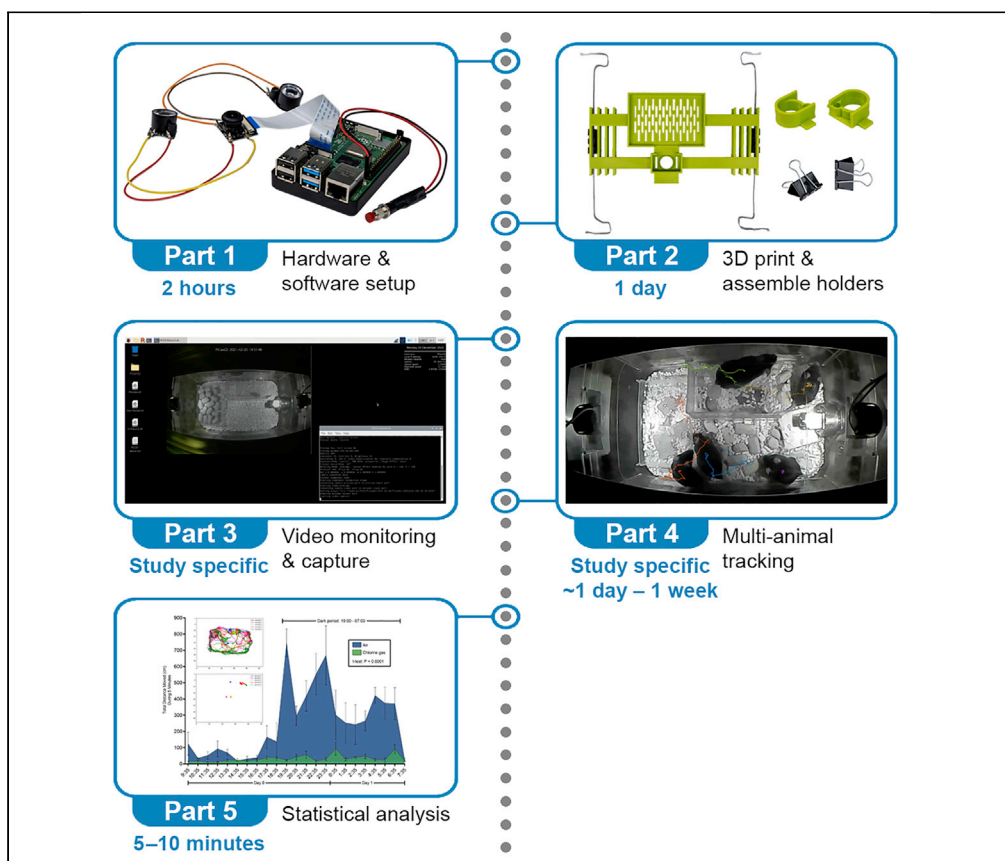


## Protocol

# Protocol for non-invasive assessment of spontaneous movements of group-housed animals using remote video monitoring



Alan David Marcus,  
Satyanarayana  
Achanta, Sven-Eric  
Jordt

alan.david.marcus@gmail.com

### Highlights

Secure, low-cost, non-invasive, remote infrared video monitoring of animals

Tracks group-housed, unmarked mice to quantify behavior and assess drug efficacy

Extensible and requires no advanced technological or programming skills

Frequent monitoring of laboratory animals is critical for ensuring animal welfare and experimental data collection. To minimize the adverse and confounding effects caused by current monitoring protocols and human presence, we developed a low-cost, non-invasive, remotely accessible, extensible infrared video monitoring system. This protocol describes the construction and operation of the system, followed by applying deep-learning neural networks to track group-housed, unmarked mice for objective behavioral quantification. This system can be adapted to a variety of home-cage environments and species.

Marcus et al., STAR Protocols  
3, 101326  
June 17, 2022 © 2022 The  
Author(s).  
[https://doi.org/10.1016/  
j.xpro.2022.101326](https://doi.org/10.1016/j.xpro.2022.101326)



## Protocol

## Protocol for non-invasive assessment of spontaneous movements of group-housed animals using remote video monitoring

Alan David Marcus,<sup>1,4,5,\*</sup> Satyanarayana Achanta,<sup>1</sup> and Sven-Eric Jordt<sup>1,2,3</sup><sup>1</sup>Department of Anesthesiology, Duke University School of Medicine, Durham, NC 27710, USA<sup>2</sup>Department of Pharmacology and Cancer Biology, Duke University School of Medicine, Durham, NC 27710, USA<sup>3</sup>Integrated Toxicology & Environmental Health Program, Duke University, Durham, NC 27710, USA<sup>4</sup>Technical contact<sup>5</sup>Lead contact\*Correspondence: [alan.david.marcus@gmail.com](mailto:alan.david.marcus@gmail.com)<https://doi.org/10.1016/j.xpro.2022.101326>

## SUMMARY

Frequent monitoring of laboratory animals is critical for ensuring animal welfare and experimental data collection. To minimize the adverse and confounding effects caused by current monitoring protocols and human presence, we developed a low-cost, non-invasive, remotely accessible, extensible infrared video monitoring system. This protocol describes the construction and operation of the system, followed by applying deep-learning neural networks to track group-housed, unmarked mice for objective behavioral quantification. This system can be adapted to a variety of home-cage environments and species.

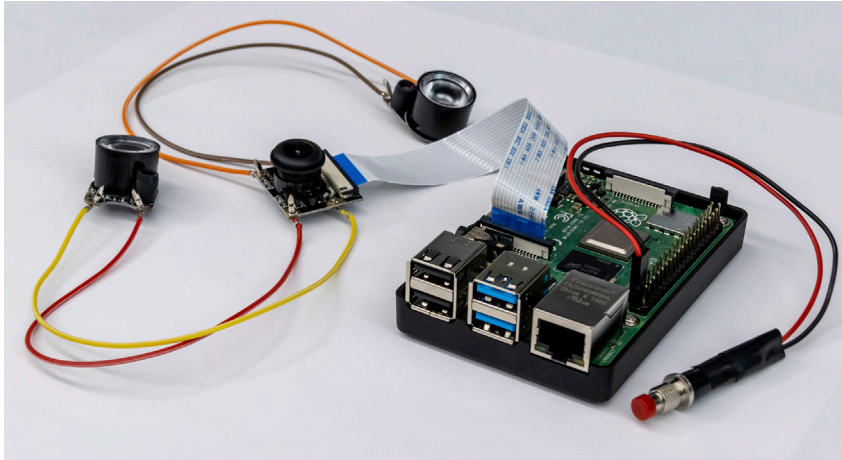
## BEFORE YOU BEGIN

Commercial options for video monitoring and tracking of laboratory animals are relatively expensive, may require a proprietary housing environment, and – due to their various technological limitations – impose severe restrictions on the number of individuals that can be concurrently housed (e.g., see review by [Voikar and Gaburro, 2020](#)). To overcome these limitations, we have developed the Home-Cage Camera System (HCCS), a low-cost and enclosure-agnostic solution for which the limitation on the number of individuals that can be concurrently monitored and tracked is bound only by practical and species-specific welfare considerations.

The HCCS offers secure remote real-time video monitoring with complete system control and the ability to enable automatic or delayed recording on startup. By offering encrypted access to the desktop environment, our system differs from other low-cost solutions (e.g., [Singh et al., 2019](#)) that provide only limited control via a web interface running on a bespoke local Apache web server. This detailed protocol guides you through the construction and operation of the HCCS. No advanced technological or programming skills are required. The essential components of our system are widely available and affordable; we constructed four video systems at an approximate cost of \$140 each. Although beyond the scope of this protocol, the system may also be readily extended to incorporate features such as automated radio frequency identification (RFID), weighing, individualized dosing, and operant conditioning tasks (e.g., [Noorshams et al., 2017](#); [Woodard et al., 2020](#)).

This protocol was developed and tested with male and female C57BL/6 and BALB/c mice group-housed (n = 5) in standard laboratory cages. However, our methods are readily transferable to other strains and species of rodents, and with only slight modifications may be implemented in different





**Figure 1. The core hardware of the Home-Cage Camera System**

Shown is a Raspberry Pi 4 with a connected infrared camera and two infrared LEDs with extended connections. Also shown is the optional shutdown button attached to the GPIO pins. The sides and top of the protective case have been removed.

environments with other species for monitoring, tracking, and quantification of additional behavioral measures.

Before implementing this protocol, obtain ethical approvals from your institutional animal care and use committee. Additional institutional policies pertaining to animal video recording may also apply.

This protocol details the preparatory steps 1) video hardware and software setup; 2) 3D printing and mounting of custom holders; 3) analysis software setup; and then the implementation of this protocol involving 4) video monitoring and capture; 5) data preprocessing; 6) multi-animal tracking with SLEAP for movement analysis; and 7) statistical analysis. We demonstrate the expected outcomes of this protocol using an acute chlorine exposure model with group-housed male C57BL/6 mice. Customized software is available from this protocol's GitHub repository: <https://github.com/Alan-Marcus/HCCS>. Whilst we used a Windows 10 computer and Android mobile device for some of the steps in this protocol, equivalent functionality is readily achieved by users with access to only macOS, iOS, or Linux systems.

### Video hardware and software setup

⌚ Timing: 1 h

The core hardware of the HCCS consists of a single-board Linux-based Raspberry Pi 4 computer (RPI) with a 256 GB microSD card, connected to a wide-angle (130°) infrared-sensitive camera and two infrared LEDs (see the [key resources table](#) for component details; [Figure 1](#)). In this section, we describe the hardware and software setup of our system. Additional documentation regarding general and advanced use of the Raspberry Pi is available: <https://raspberrypi.com/documentation>.

1. Extend the LED-camera connection.
  - a. Note, this step may be optional depending on your particular cage environment setup or if external supplementary infrared lighting is used in place of the integrated LEDs. This step may also be skipped and undertaken at a later time.
  - b. Extend the LED-camera connection with soldered jumper cables so that the LEDs can abut the cage lid to avoid reflections, whilst enabling the camera to be elevated to facilitate imaging

the entire cage area. Before soldering the connections, you may choose to test the setup by simply bending and securing the terminal jumper cable connectors through the screw holes on the LEDs and camera.

2. Connect the camera ribbon cable to the RPi.
  - a. Note, the included camera cable is approximately 15 cm in length; longer cables are available if required for your particular setup (e.g., we have successfully used cables up to 200 cm in length: <https://amazon.com/dp/B0863NBKY5>).

**Optional:** Install a real-time clock. The RPi does not maintain the time when powered off, rather it relies on an internet connection to sync with a time server. As an alternative, a coin-battery-powered real-time clock module can be readily installed using the General-Purpose Input/Output (GPIO) pins to maintain the date and time. Compatible real-time clocks are widely available and affordable (e.g., see further instructions and some module options here: <https://learn.adafruit.com/adding-a-real-time-clock-to-raspberry-pi>).

**Optional:** Install a physical shutdown button. As for many electronic devices, the RPi has a shutdown procedure that should be followed before removing the power. When using the graphical desktop environment, this can be achieved by selecting “shutdown” in the power options from the main menu (the raspberry icon in the top left of the taskbar) or sending a shutdown request via the command line. As an alternative, a push button connected to pins 37 (GPIO 26) and 39 (ground) via jumper cables can be installed to initiate the shutdown procedure. Regardless of the approach taken to shut down the RPi, wait until the green LED stops flashing before unplugging the power cable.

3. Fit the RPi into the protective case.
  - a. Whilst some cases include heatsinks to attach to the RPi and fans to power via the GPIO pins, we found them to be unnecessary for this protocol and chose to avoid the potential noise/vibration disturbance of the fan.
  - b. The CPU temperature is reported on the RPi’s taskbar and can indicate if cooling strategies are required in your particular environment. See [troubleshooting 1](#).
4. Install the RPi operating system.
  - a. Flash our preconfigured disk image (*HCCS\_2021\_12\_06.img.xz*; 1.11 GB) to the microSD card using Raspberry Pi Imager (or a similar disk image flashing program). This image includes our preconfigured scripts and additional customizations to improve functionality. It is based on the 2021-12-02 Buster legacy version release of Raspberry Pi OS.
    - i. Click “Choose OS” and scroll down to click “Use Custom”. Select our downloaded disk image from your computer.
    - ii. Click “Choose Storage” and select the microSD card to flash.
    - iii. Click “Write” and confirm that you want to continue. Wait for the software to complete writing and verifying the image.
  - b. Insert the microSD card into the RPi and connect a screen (using the micro HDMI cable inserted into HDMI0 – the port nearest the USBC power port) and a USB keyboard and mouse.
  - c. Insert the RPi’s power cable and wait for the system to complete installation. This process will take approximately 1 min and the RPi may reboot several times. A welcome message will display in a terminal window on the desktop environment to indicate when the system is ready for you to continue with the setup instructions below.
5. Change the RPi’s password (optional, but recommended).
  - a. In the terminal window, enter:

```
>passwd
```

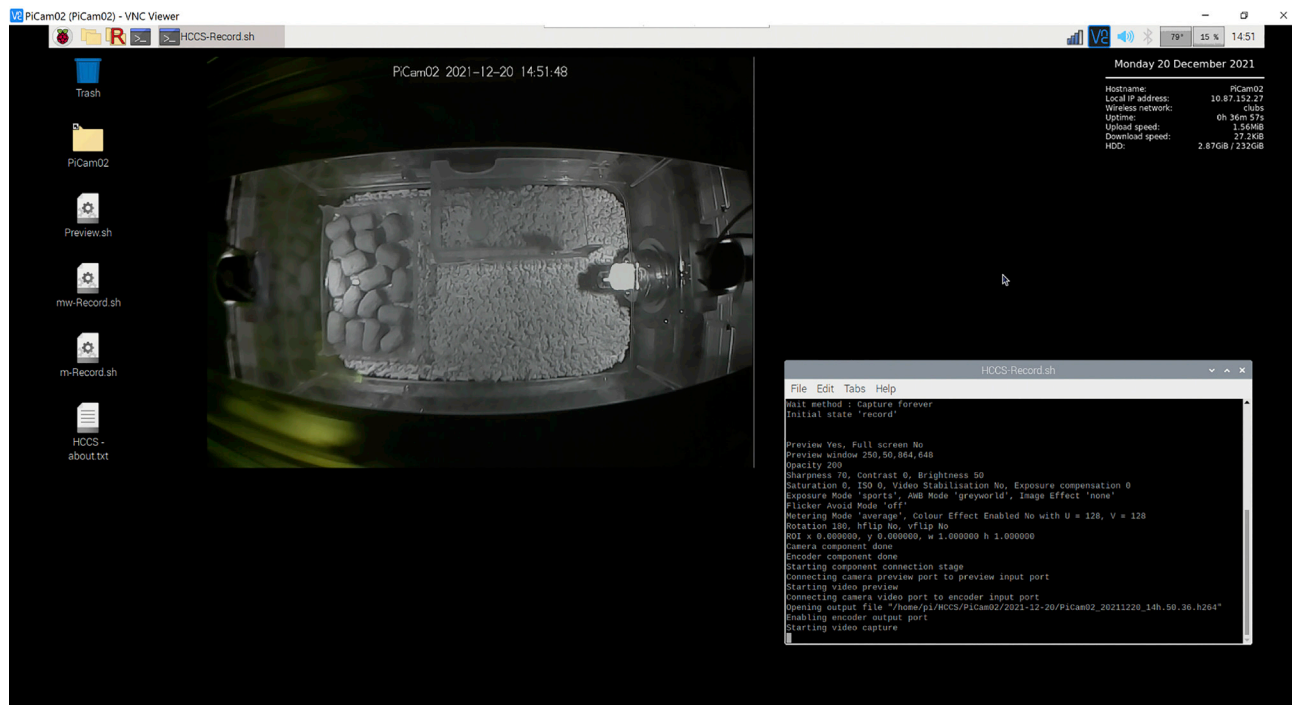
- b. You will be prompted to enter the current password “RemoteCamera” and then your new password.
    - c. Alternatively, you may change your password using the Raspberry Pi Configuration software, available under “Preferences” in the main menu. Here, you may also change the RPi’s hostname, language, timezone, and other settings, as necessary.
  6. Connect to your network.
    - a. Plug in an Ethernet cable or left-click the Network Manager icon in the taskbar (computer screens icon in the top right) and select your wireless network.
    - b. Right-clicking Network Manager provides additional options, including the ability to edit the connection details and prioritize network connection preferences.
    - c. Note, the RPi’s hostname, IP address, and connected network name are displayed on the desktop, along with other useful system information such as available space on the microSD card.

**Optional:** Connect to a VPN. Given the multitude of institutional network configurations, it is beyond the scope of this protocol to provide detailed networking advice that will satisfactorily cover all users, however, we do note that VPN connections are readily implemented if desired. For example, with the preinstalled plugins, we were able to connect to our VPN via the Network Manager “VPN Connections” option, requiring only the correct gateway address, username, and password.

7. Establish remote access to the RPi.
  - a. Download and install VNC Viewer on your remote device(s). Multiple devices and users may connect to the same RPi simultaneously.
  - b. Create a free RealVNC account: <https://manage.realvnc.com> (for non-commercial use, you can activate the free “Home” subscription; you do not need to sign up for the Professional or Enterprise options). Sign in with this account in VNC Server on the RPi (VNC icon in the top right of the taskbar) and in VNC Viewer on your other device(s). Note, if your other device(s) will always be on the same local network as the RPi, a RealVNC account is optional.
  - c. View and control the RPi via VNC Viewer on another device.
    - i. When on the same local network as the RPi, you may connect by entering the RPi’s IP address, username (“pi”), and password (“RemoteCamera”, unless changed).
    - ii. Alternatively, when signed in with your RealVNC account, irrespective of whether you are on the same network, your available RPi(s) will be listed in your “Team”. Select “PiCam01” and enter the RPi’s username and password to connect.
    - iii. You will see the same desktop view in VNC Viewer as you do on the connected HDMI screen.

**Optional:** You can now unplug the screen, keyboard, and mouse, and operate the RPi as a “headless” device remotely using VNC. See [troubleshooting 2](#).

8. Using the camera.
  - a. The camera is controlled by scripts that access the underlying raspivid software. We have provided three preconfigured scripts on the RPi’s desktop that can be manually started by double-clicking on them or they can be configured to automatically start when the RPi is turned on.
  - b. Each of the scripts opens a terminal window that displays the camera commands and a live feed from the camera is displayed as a preview window that is fixed in position ([Figure 2](#)). The translucent nature of the preview enables you to access interface elements behind the preview; the terminal window should be moved to another part of the desktop if obscured or minimized to the taskbar. The camera and scripts are stopped by closing the terminal window or pressing “Ctrl+c” in the terminal window.



**Figure 2. The Home-Cage Camera System desktop with a live camera preview during day-time of an empty mouse home-cage, as seen remotely using VNC**

Present in the home-cage is an infrared transmittable shelter and dry extruded food pellets inside a plastic feeding tray.

- c. Note, only one script can control the camera at a time. If another script is started (or the same script started more than once), it will be denied access to the camera and close.
- d. Preview script (*Preview.sh*): shows the live feed from the camera without recording to a file. This is useful for testing your setup and for monitoring, when recorded video files are not required.
  - i. This script can be auto-started by renaming it to *a-Preview.sh*.
  - ii. To modify the operation of this script (see below), edit *HCCS-Preview.sh* located in `"/home/pi/HCCS/Scripts"`.
- e. Recording scripts (*m-Record.sh* and *mw-Record.sh*): these scripts show the live feed from the camera and save the video to a file in a new folder based on the date when the script was initiated (if recording continues until the next day, all recorded files will be in the same folder). These record scripts are identical, however, *mw-Record.sh* will show a countdown from 1 min before executing. This is primarily useful when auto-starting the recording and if a real-time clock is not installed, to give the RPi a chance to sync its time. Even if the time sync fails (e.g., internet access is not available), the script still runs so that data is not lost (although the videos will have the incorrect date and time).
  - i. These scripts can be auto-started by renaming them to *a-Record.sh* or *aw-Record.sh*, respectively.
  - ii. Both scripts control the camera using *HCCS-Record.sh*, located in `"/home/pi/HCCS/Scripts"`.
  - iii. The duration that *mw-Record.sh* will wait before recording can be adjusted by specifying the hours, minutes, and/or seconds in *HCCS-Wait.sh*, located in `"/home/pi/HCCS/Scripts"`.
- f. See [troubleshooting 3–5](#).
9. Adjusting camera settings.
  - a. Additional documentation on the use of the camera and all available command-line options is available: <https://raspberrypi.com/documentation/accessories/camera.html>.

- b. Our recording script (*HCCS-Record.sh*) is configured to continuously save video in 5-min segments until stopped. The videos are encoded using H264 with a completely variable bitrate, captured at 30 frames per second with 1296 × 972 resolution, use the “sports” exposure mode (to ensure a fast shutter speed), and “greyworld” auto white balance (to correct for the color shift that occurs without an infrared filter). Additionally, we have increased the sharpness setting to a value of 70 (default value 0; nominal scale from -100 to 100) and set the quantisation parameter to 35 (can be varied from 10 to 40; higher values reduce quality and file size). The video is annotated with the RPi’s hostname and the date and time in YYYY-MM-DD HH:MM:SS format and each file is similarly named. The preview window is displayed at 864 × 648 resolution (two-thirds the recorded resolution) and with an opacity value of 200 (0 = transparent and 255 = opaque). All of these settings can be readily changed:
  - i. Locate *HCCS-Record.sh* in “/home/pi/HCCS/Scripts”.
  - ii. Right-click the script and select ‘Mousepad’.
  - iii. Save the file once you have made the necessary changes.
  - iv. Alternatively, the script can be edited using a terminal-based editor:

```
>nano /home/pi/HCCS/Scripts/HCCS-Record.sh
```

#### 10. File management.

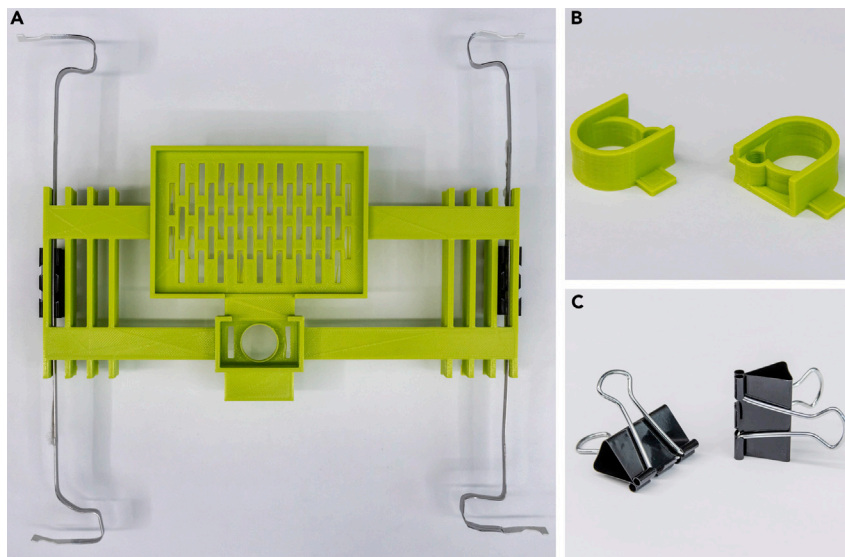
- a. Video files are saved in a dated folder within “/home/pi/HCCS/PiCam01”. This may be accessed using the shortcut on the desktop or via File Manager. We have also provided Root File Manager which has elevated permissions (the red “R” icon in the top left of the taskbar or available under “Other” in the main menu).
  - i. You must use Root File Manager to delete any video files. This adds an extra layer of protection against accidental data loss.
  - ii. Deleted files are sent to “Root Trash” (located at “/root/.local/share/Trash” and bookmarked in Root File Manager).
  - iii. “Root Trash” can only be emptied via the terminal:

```
>sudo trash-empty
```

- b. Video files may be played back on the RPi using VLC Media Player (see step 17 in [analysis software setup](#) for details about converting H264 video files to MP4 files for more convenient playback and use on other devices).
- c. Files may be transferred from the RPi via USB, to a network drive, via a file-sharing service (e.g., Dropbox), or using VNC (right-click the VNC icon and select “File Transfer”).

#### 11. Reboot the RPi to confirm that the camera script starts automatically (if so configured), that the internet connects, and that you can access the RPi via VNC.

**Optional:** To set up more than one RPi camera system, follow the steps above, however, each RPi should have a unique hostname (e.g., “PiCam01”, “PiCam02”, etc.). After step 5 (changing the default password), change the hostname using Raspberry Pi Configuration. Edit the hostname in *HCCS-Record.sh* (there are four instances of the hostname in this file to change). This will ensure that video files are saved with the correct identification. Optionally, edit the one instance of the hostname in *HCCS-Preview.sh*. *Reboot the RPi and continue with the rest of the setup.*



**Figure 3. 3D printed custom holders**

(A) The RPi-camera holder with attached aluminium support struts.

(B) The LED holders.

(C) Binder clips may be used to secure the RPi-camera holder and supports to the mouse rack.

### 3D printing and mounting of custom holders

⌚ Timing: 1 day

The RPi and camera can be mounted in a single cage space of a standard mouse rack (Allentown, NJ, USA) using our 3D-printed polyester holder and hand-crafted aluminium support frames (Figures 3 and 4). The LEDs can be held in place with our 3D-printed holders and affixed to the cage lid using tape, facilitating quick removal of the LEDs from the holders (or both together) for removal of the cage from the rack. We designed the holders using Blender (<https://blender.org>) and prepared the meshes for printing using Microsoft 3D Builder (<https://microsoft.com/en-us/p/3d-builder/9wzdncrfj3t6>); our designs can be readily modified using standard 3D design software to suit other environments as required.

#### 12. 3D print the holders.

- a. We printed the holders (*HCCS\_RPi-camera\_holder.stl* and  $2 \times$  *HCCS\_LED\_holder.stl*) using PLA material on an Ultimaker S3; other models of extrusion-based 3D printers and materials may be used. Our print settings included the use of a brim and supports, layer height of 0.2 mm, wall thickness of 0.8 mm, and infill density of 5%. The expected printing time is approximately 8.5–9.5 h, depending on the part layout and whether they are printed together or individually.
  - i. Remove the brim and support material.
  - ii. Light sanding of the camera lens hole may be necessary to ensure a smooth surface and unobstructed fit.

- b. Mount the RPi-camera into the holder.

#### 13. Construct supporting frame.

- a. To mount the printed RPi-camera holder in the mouse rack, support struts can be fashioned by bending aluminium strips into shape (we sourced these from Esselte Pendaflex file folders, however, other materials may be readily used as a replacement). We positioned the supporting frame approximately 12 cm above the cage lid; this height will vary with the size of the cage and the camera's field of view.





**Figure 4. The Home-Cage Camera System hardware mounted in a standard mouse rack**  
The camera preview shown in [Figure 2](#) was recorded with the equipment in the position shown here.

- i. Bend two pieces of aluminium to wrap around the vertical frames of the mouse rack at the back and the front of the cage space.
- ii. Secure the aluminium frame in place using binder clips.
14. Secure the RPi-camera/holder on the supporting frame.
  - a. Use the preview script and view the live feed through a mobile or nearby device to help position the holder on the frame.
    - i. Adjust the position of the holder forwards/backwards in the rack (left/right on the video preview) by changing which slot sits over the aluminium frame.
    - ii. Adjust the position of the holder left/right in the rack (down/up on the video preview) by sliding along the aluminium frame.
    - iii. Once the optimal position has been obtained, secure the holder to the aluminium frame using binder clips.
15. Secure the LED holders onto the cage lid.
  - a. Position the LED holders on top of the cage lid at the front and back. Secure in place with tape.
  - b. Insert the LEDs into the holders and ensure that the cage is adequately illuminated when the room lights are on and off.

16. Check the camera focus.
  - a. The camera focus can be adjusted by rotating the lens in either direction.
  - b. Shut down the RPi and unplug the power cable. Your HCCS is ready for use.

### Analysis software setup

⌚ Timing: 1 h

The following section describes the software used for the processing and analysis of videos recorded using this protocol. Whilst we used a Windows 10 computer, the software packages are also available for other operating systems or there are freely available equivalents.

17. FFmpeg.
  - a. Purpose/Use: a multimedia command-line tool that we use to convert and concatenate the RPi's H264 video files to MP4 files. The process we implemented is fast and lossless as it does not involve any encoding – it simply repackages (remuxes) the video data. We have provided two scripts for use:
    - i. *FFmpeg\_convert-to-mp4.bat*: drag-and-drop any number of H264 files onto this file and they will be individually converted to MP4 files.
    - ii. *FFmpeg\_concatenate-convert-to-mp4.bat*: drag-and-drop any number of H264 files onto this file and they will be concatenated and converted into a single MP4. Note, the files will be concatenated in ascending order, but the file that is clicked to drag onto the script will be the first file in the new video (e.g., with these files selected – one.h264, two.h264, three.h264, four.h264 – clicking and dragging with 'three.h264' will create an MP4 file concatenating the files as three-one-two-four, whereas clicking and dragging with 'one.h264' will create an MP4 file ordered as one-two-three-four). For this script to function, the folder path to the H264 files must not contain a space.
  - b. Links to precompiled binaries or the source code for compilation are available from <https://ffmpeg.org>; alternatively, we have provided an older build (*ffmpeg-20200828-ccc7120-win64-static*) in our GitHub repository.
  - c. To install, extract the archive and add the executable directory to the Windows Environment Path, for example:
    - i. Click the Windows button and type "path", then select "Edit environment variables for your account".
    - ii. Select the "Path" variable and click "Edit".
    - iii. Click "New" and add the path to the executable directory (e.g., "C:\ffmpeg\bin").
    - iv. Click "OK" to close the windows.
18. LosslessCut.
  - a. Purpose/Use: a graphical interface to FFmpeg that we use to quickly and losslessly trim video files.
  - b. In settings, change "Keyframe cut mode" to "Keyframe cut" instead of "Normal cut".
19. PowerRename (part of Microsoft's PowerToys utilities).
  - a. Purpose/Use: a bulk file renaming tool that we use to rename the concatenated and trimmed video files.
20. SLEAP.
  - a. Purpose/Use: a multi-animal pose estimation framework that we use to train a deep-learning neural network to track the individual animals in our video files.
  - b. Requires installation of Miniconda or Anaconda; additional documentation, installation guide, and tutorials are available at <https://sleap.ai>.
  - c. Note, training and inference with SLEAP requires a GPU, either on your local computer or remotely (e.g., an institutional/commercial compute cluster or a free cloud service like Google Colaboratory (Colab) – <https://colab.research.google.com>).

- d. We have provided example Colab notebooks for use with our protocol and two scripts for generating inputs to the notebooks:
  - i. *HCCS\_SLEAP-create-training-videos-list.bat*: place this file in your videos folder and run it to generate a text document containing a list of MP4 files.
  - ii. *HCCS\_SLEAP-create-analysis-h5-list.bat*: place this file in your videos folder and run it to generate a text document containing a list of H5 files.
21. Spreadsheet software.
  - a. Purpose/Use: a spreadsheet program is recommended for data collection and compilation. We used Microsoft Excel but any general-purpose spreadsheet program may be used as an alternative.
    - i. We have provided an example worksheet (*HCCS\_example\_project\_analysis\_template.xlsx*) for use with our protocol that demonstrates the expected outcomes.
22. Statistical software.
  - a. Purpose/Use: a statistical analysis program is recommended for data analysis and significance testing. We used GraphPad Prism but many other programs may be used as an alternative.
    - i. We have provided an example Prism project file (*HCCS\_example\_project\_AUC\_template.pzfx*) for use with our protocol that demonstrates the expected outcomes.

## KEY RESOURCES TABLE

| REAGENT or RESOURCE  | SOURCE                               | IDENTIFIER  |
|--|--------------------------------------|---|
| <b>Software and algorithms</b>   |                                      |   |
| Home-Cage Camera System (HCCS; includes customized Raspberry Pi OS, scripts and notebooks for data processing and analysis, and 3D holder designs) | This paper; GitHub                   | <a href="https://github.com/Alan-Marcus/HCCS">https://github.com/Alan-Marcus/HCCS</a>   |
| Raspberry Pi Imager  | Raspberry Pi                         | <a href="https://raspberrypi.com/software">https://raspberrypi.com/software</a>   |
| VNC Viewer   | RealVNC                              | <a href="https://realvnc.com/en/connect/download/viewer">https://realvnc.com/en/connect/download/viewer</a>   |
| FFmpeg   | FFmpeg                               | <a href="https://ffmpeg.org">https://ffmpeg.org</a> or <a href="https://github.com/Alan-Marcus/HCCS">https://github.com/Alan-Marcus/HCCS</a>  |
| LosslessCut  | GitHub                               | <a href="https://github.com/mifi/lossless-cut">https://github.com/mifi/lossless-cut</a>   |
| PowerRename (PowerToys)  | GitHub                               | <a href="https://github.com/microsoft/PowerToys">https://github.com/microsoft/PowerToys</a>   |
| SLEAP  | <a href="#">Pereira et al., 2022</a> | <a href="https://github.com/murthylab/sleap">https://github.com/murthylab/sleap</a>   |
| <b>Experimental models: Organisms/strains</b>  |                                      |   |
| Mouse: C57BL/6   | Charles River Laboratories           | Strain code: 027  |
| <b>Other</b>   |                                      |   |
| Raspberry Pi 4B Computer (4 GB RAM; the 2 and 8 GB RAM models are also expected to be compatible)  | Raspberry Pi                         | <a href="https://raspberrypi.com/products/raspberry-pi-4-model-b/?variant=raspberrypi-4-model-b-4gb">https://raspberrypi.com/products/raspberry-pi-4-model-b/?variant=raspberrypi-4-model-b-4gb</a> |
| CanaKit 3.5A Raspberry Pi 4 Power Supply (USB-C)   | CanaKit                              | <a href="https://canakit.com/raspberry-pi-4-power-supply.html">https://canakit.com/raspberry-pi-4-power-supply.html</a>   |
| Lexar 256 GB UHS-I MicroSDXC Card (recommended minimum size)   | Amazon                               | <a href="https://amazon.com/dp/B08GGCMT2Z">https://amazon.com/dp/B08GGCMT2Z</a>   |
| NoIR (Night Vision) Camera using a 5 MP OV5647 Sensor with a 130° Lens and two Infrared LEDs   | Aliexpress                           | <a href="https://aliexpress.com/i/32935079559.html">https://aliexpress.com/i/32935079559.html</a>   |
| Raspberry Pi 4 Case  | Amazon                               | <a href="https://amazon.com/dp/B07D3S4KBK/">https://amazon.com/dp/B07D3S4KBK/</a>   |
| Jumper Wire Cables (120 pack)  | Amazon                               | <a href="https://amazon.com/dp/B01EV70C78">https://amazon.com/dp/B01EV70C78</a>   |
| Push Button Momentary Switch (12 pack)   | Amazon                               | <a href="https://amazon.com/dp/B07DF78VJ7">https://amazon.com/dp/B07DF78VJ7</a>   |
| CanaKit Raspberry Pi 4 Micro HDMI Cable  | CanaKit                              | <a href="https://canakit.com/raspberry-4-mico-hdmi-cable.html">https://canakit.com/raspberry-4-mico-hdmi-cable.html</a>   |
| Esselte Pendaflex File Folders #59203 (25 pack)  | Amazon                               | <a href="https://amazon.com/dp/B001P4RAGM">https://amazon.com/dp/B001P4RAGM</a>   |
| Binder Clips (24 pack)   | Amazon                               | <a href="https://amazon.com/dp/B07BPJRCFS">https://amazon.com/dp/B07BPJRCFS</a>   |

**Note:** With the exception of the RPi, all hardware components may be swapped out with an alternative brand/model. Many alternative cameras are available (including options for a different field of view and a newer 8 MP IMX219 sensor). In our experience, hardware components were available from multiple suppliers and inclusion of a specific link in the table above is intended to be representative of one available option at the time of publication, rather than a strong recommendation for sourcing.

### STEP-BY-STEP METHOD DETAILS

Here, we describe the implementation of this protocol for remote video monitoring and non-invasive assessment of spontaneous movement, and demonstrate its use with the example of an acute chlorine exposure model. All procedures were approved by Duke University's Institutional Animal Care and Use Committee and Division of Laboratory Animal Resources.

Our research group has a long-standing interest in the development of countermeasures for treatment of chemical inhalation injuries, including pulmonary injuries caused by exposures to chlorine gas. Chlorine gas inhalation represents a major chemical threat in domestic, occupational, and transportation accidents. Chlorine gas has also been used as a chemical weapon beginning in World War I and was most recently used by the Syrian regime in the country's civil war (Schneider and Lütkefend, 2019). Chlorine gas also represents a terrorism threat due to its wide availability and ease of manufacture. Exposure to chlorine gas causes acute respiratory distress syndrome and mortality, depending on the exposure conditions (i.e., concentration and duration of exposure). No specific antidote is available for treating chlorine inhalation injuries and symptomatic treatment is considered the standard of care (Achanta and Jordt, 2021; Achanta and Jordt, 2020). Rodent models are widely used for studying the pathophysiology of chlorine inhalation injuries and for developing medical countermeasures (Achanta and Jordt, 2021; Balakrishna et al., 2014). Chlorine exposure of rodents reduces their subsequent mobility (Filippidis et al., 2012) and results in decreased feed and water intake. Increased mobility in response to administration of an experimental chlorine countermeasure may be interpreted as evidence for efficacy, deserving more detailed analysis of the countermeasure response. This protocol was developed to remotely and accurately monitor the mobility of cohorts of chlorine-exposed mice and air-controls, to be used in the future to assess the efficacy of countermeasures. Remote monitoring of animals enables investigators to collect experimental data without the adverse and confounding effects of human presence, whilst ensuring animal welfare and facilitating timely intervention before any individuals reach their *a priori* humane endpoints.

In our example study, 7-week-old male C57BL/6 mice were housed in two groups of five in cages on our HCCS-equipped rack and maintained in climate-controlled facilities under a 12:12 h light:dark cycle ('daytime': 07:00–19:00). Food (PicoLab Rodent Diet 20, #5053, LabDiet, MO, USA) was provided on the cage floor, instead of the standard overhead feeding baskets, so that the camera's field of view was not obstructed. Water was available *ad lib* via nipple drinkers and supplemental hydrogel (ClearH<sub>2</sub>O, ME, USA) was provided in trays. Nesting material and infrared transmittable shelters (Safe Harbor Mouse Retreat, Bio-Serv, NJ, USA) were provided as additional enrichment items.

Following acclimatization to their home-cage environment, the mice were exposed to either 400 ppm chlorine gas or medical-grade air for 30 min in a whole-body exposure chamber within a chemical fume hood, as per our previously published model with some minor modifications (Balakrishna et al., 2014). Mice were then transferred to an open-topped enclosure within the fume hood for 5 min, before being returned to their group home-cages. At 24 h post-exposure, mice were terminally anesthetized for sample collection for studying other injury parameters, as previously described (Balakrishna et al., 2014).

**Note:** Although the mice were individually identified for longitudinal body weight monitoring using tail ink marks (VWR Lab Markers, Avantor, PA, USA), as the ink used transmits infrared

light, marks were not resolved by the camera under the infrared illumination and hence could not be used to identify specific mice in the recorded videos. If the identification of specific individuals within a group is important for your video analysis, ensure that they are marked in such a way so as to be distinctly identifiable in the recorded videos (e.g., with unique fur cuts or marking using non-infrared-transmitting dyes/inks).

## Video monitoring and capture

### ⌚ Timing: Specific to the intended duration of the experiment

In this section, we describe the use of the HCCS to monitor and record animals during an experiment. During the pre-experimental acclimatization period, the HCCS should be turned on to acclimatize the animals to the LED lights. Animals may also be monitored during this period using either the preview or the recording scripts.

1. Check the camera focus.
  - a. Before each experiment, the camera's position and focus should be checked and adjusted as required.
  - b. Verify that the RPi has the correct date and time.
2. Start recording.
  - a. Start the video recording script manually, if not already initiated with an automatic script.
  - b. Monitor the animals and cage environment remotely via VNC.
3. Cage removal and experimental interventions.
  - a. Recordings can be stopped by terminating the camera script or simply by shutting down the RPi.
    - i. Unless the cage is to be removed for an extended period of time, we tend to continue the recording and discard/disregard the intervening videos.
  - b. Remove the LEDs and remove the cage.
  - c. After the experimental intervention and cage replacement, insert the LEDs into their holders and, if stopped, restart the video recording manually or automatically by turning the RPi back on. Verify and adjust the camera's position if necessary.
4. End recording.
  - a. After your experiment, terminate the camera script and transfer the video files to another device for storage, backup, and analysis.
  - b. Delete the video files from the RPi's memory card and shut down the RPi.

## Data preprocessing

### ⌚ Timing: <1 day

Although it is possible to analyze the entirety of the recorded experimental period, this may be unnecessary for many scientific applications and impractical if implementing manual proofreading as a data quality assurance step. In this section, we describe the preparation of a representative sample of recorded videos for input into the SLEAP framework.

5. Choose the time points of interest.
  - a. The duration and number of time points to analyze will be dependent upon the outcome of interest and design of each specific experiment. In our example study, we chose to analyze a 5-min period every hour, starting 5 min after the mice were returned to their cage and replaced on the rack. For the analysis of spontaneous movement, it is important to consider and minimize potential sources of disturbance within the animal housing area. On the other hand, the response to a stimulus challenge may reveal informative differences between groups.
6. Concatenate time point H264 files and convert to MP4.

- a. Use *FFmpeg\_concatenate-convert-to-mp4.bat* to concatenate and convert the two (or more) files encompassing the selected time points. If your time point duration is shorter than the length of the recorded segment, and entirely encompassed within one recording, then use *FFmpeg\_convert-to-mp4.bat* to just convert the files to MP4.
7. Trim concatenated files.
  - a. Use LosslessCut to trim each MP4 file to the period of interest plus at least one keyframe before and after. Whilst optional, trimming the files will reduce the required processing time and storage space.
8. Rename trimmed files.
  - a. Use PowerRename to batch rename the trimmed files. For example, in our study with recordings from two groups (air- and chlorine-exposed) with the time point of interest at 35 min past the hour, every hour, we renamed our files to fit the format "RPIHostname-Group\_YYYYMMDD\_HH-35.mp4":
    - i. Select all video files, right-click, and select "PowerRename".
    - ii. Tick "Use regular expressions".
    - iii. Search for "h.\*" and replace with "-35.mp4". Click "Apply".
    - iv. Search for "Cam01" and replace with "Cam01-Air". Click "Apply".
    - v. Search for "Cam02" and replace with "Cam02-Cl2". Click "Apply and close".
    - vi. Result: PiCam01-Air\_20210122\_09-35.mp4, PiCam01-Air\_20210122\_10-35.mp4, etc.; and PiCam02-Cl2\_20210122\_09-35.mp4, PiCam02-Cl2\_20210122\_10-35.mp4, etc.
  - b. Move the renamed video files to a project folder called "videos".

### Multi-animal tracking with SLEAP for movement analysis

⌚ Timing: Varies with the number of video files to analyze and number/type of GPUs available

In this section, we describe the use of SLEAP to train a deep-learning neural network to track the movements of individual animals in the videos. Whilst we strongly recommend the use of a GPU computing cluster to considerably reduce the amount of time required by taking advantage of parallel processing, in order to maximize the accessibility of this protocol, we provide directions here for use with Colab (a freely accessible cloud notebook environment with a single GPU). The steps in this section entail iteratively training a model with prediction-assisted labeling, using the model to predict (infer) the position of all animals across all videos, proofreading the model predictions, and then calculating the "Total Distance Moved" and "Time Mobile" for each animal at each time point. Note, in this protocol we demonstrate the tracking and analysis of only a single point on each mouse – the centroid – however, more complex pose estimation is easily implemented with SLEAP and can facilitate more complex analysis and feed into other analytical pipelines (e.g., SIMBA; Nilsson et al., 2020). We recommend reviewing the tutorial available at <https://sleap.ai> for more information and advice about using the SLEAP interface, beyond our specifications and brief description here.

9. Initial labeling.
  - a. Open SLEAP and save a new project (File > Save As; e.g., *HCCS\_example\_project.slp*). Intermittently save the project as you are working.
  - b. In the "Skeleton" panel, click "New Node" and rename it "centroid".
  - c. In the "Video" panel, click "Add Videos" and select your recorded videos.
  - d. Select the first video. In the "Instances" panel, click "New Instance". A labeled dot will appear in the video – move this "centroid" to the appropriate position on one of your animals.
  - e. Add more instances until you have labeled all of the animals in the frame.
    - i. If the centroid position of an individual is not visible, but its position can be reasonably inferred, position the centroid as appropriate – this helps the network to learn about the likely positions of occluded parts.
    - ii. If you cannot reasonably determine the position of an occluded centroid, leave it unlabeled.

- f. Label approximately 10–20 frames across several different videos, ideally representing individuals in different positions and postures.
  - i. Move to other frames by using the keyboard shortcuts or clicking along the timeline at the bottom of the screen.
  - ii. You can also use the “Labeling Suggestions” panel to generate a list of frames to label.
10. Specify training parameters and export labeled frames.
  - a. Open the training pipeline dialog (Predict > Run Training) and select “multi-animal top-down” as the pipeline type.
  - b. Select “centroid” as the anchor part.
  - c. Change “Predict On” to “nothing”.
  - d. Customize the hyperparameters to suit your data set in the centroid and centered instance model configuration tabs, as necessary.
    - i. Note, although the default hyperparameter values may be appropriate, particular for initial training, it is recommended to adjust the receptive field size to suit the size of your animals and turn on contrast and brightness augmentation; see <https://sleap.ai/guides/choosing-models.html> for more details.
  - e. Click “Save configuration files” and select an output folder (e.g., create a folder called “config”).
    - i. This will export the model specification files (*centered\_instance.json* and *centroid.json*) as well as two scripts (*inference-script.sh* and *train-script.sh*); you can delete the inference and training scripts, which are not used in this protocol.
    - ii. Click “Cancel” to close the dialog box. Note, clicking “Run” will attempt to train the model locally.
  - f. Export the labeled frames to the config folder (Predict > Export Labels Package > Labeled frames; e.g., *HCCS\_example\_project.pkg.slp*).
11. Upload files to Google drive to train and use an initial model.
  - a. Create a Google Drive account or use an existing account (<https://google.com/drive>).
    - i. Drive is used to store and transfer the data used for training and inference with Colab; any data present within Colab is otherwise lost at the end of each session.
  - b. Create a new folder “SLEAP” and within this folder create new folders called “analysis”, “config”, “models”, and “videos”. Within the “analysis” folder, create the folder “plots”.
  - c. Upload your configuration and video files to their respective folders.
  - d. Open our Colab notebook *SLEAP\_Training\_&\_Inference* and follow the instructions within.
    - i. This notebook will guide you through training an initial top-down model to identify and track the centroids on all recorded animals. No programming skills are required; you simply need to input details where prompted and click the “play” buttons on each cell (each section of code) to run the code.
    - ii. Only one cell will run at a time; by clicking on each cell in order whilst a prior one is running, you can queue them to play automatically.
    - iii. Use *HCCS\_SLEAP-create-training-videos-list.bat* to create the list of videos for input into step 3 of the notebook.
    - iv. The time to train and infer is dependent upon the type of GPU allocated to you, the size of your training file, and the number of videos and frames selected for inference.
    - v. Now is a good time to enjoy a cup of tea.

**Optional:** Depending on your study design and the characteristics of your videos, you may want to edit some of the settings for inference. In our example study, we culled the number of individuals to track to a maximum of five, used the “simple” tracker with Hungarian matching, a tracking window of 10, and connected short breaks in tracking. The culling and tracking window settings may be readily adjusted using the notebook inputs. Other settings may be changed by editing the notebook code; a reference of options is available: <https://sleap.ai/guides/cli.html>.

12. Assess initial model and use predictions for further labeling.
  - a. Unzip the downloaded predictions into your videos project folder.

- b. Open each predictions file in SLEAP (File > Open Project) and scroll through the video, taking note of the “Score” value in the “Instances” panel for each predicted instance.
  - c. Correct the predictions.
    - i. Make the predictions editable for the current frame (Labels > Add Instances from All Predictions on Current Frame).
    - ii. Move, add, and delete instances as required.
    - iii. Repeat for as many frames as desired, particularly for egregious predictions.
    - iv. When done, delete the remaining predictions (Labels > Delete All Predictions), and save the file.
  - d. Open the main project file and add in the new labeled frames (File > Merge into Project; select all corrected prediction files).
    - i. A dialog box will provide a summary of merged changes and options to manage any conflicts.
  - e. Export the labeled frames file again and upload to Drive. You do not need to change the model specification configuration files (although you can, if desired).
  - f. Retrain your model using the new labeled frames file to generate a (hopefully) better model.
  - g. Continue this iterative prediction-assisted labeling and training process until you are satisfied with the performance of your model. Metrics are available to assist with model assessment (Predict > Evaluation Metrics for Trained Models).
  - h. Once you have decided on a “final” model, perform inference across all frames for all of your videos.
13. Proofread predictions/tracks and export data.
- a. Open in SLEAP the predictions file from your “final” model for the first video and “Add Instances from all Predictions on Current Frame” for the first and last frames in the video. Adjust the instance positions if necessary.

**Optional:** Remove all predictions with a low score (Labels > Delete Predictions with Low Score) to potentially improve the quality of your tracking. We deleted all predictions with a score below 0.8, however, the nature of your videos and the performance of your trained model will influence what is an appropriate threshold for your study.

- b. Scroll through the video to review and correct the positions of the predicted instances and their temporal association into individual tracks (Figure 5).
  - i. Color each individual (View > Color Predicted Instances) and adjust the trail length to suit the speed of review and nature of the animal movements (View > Trail Length).
  - ii. Correct or delete erroneous predicted instances and delete, combine, or correct the identities of tracks, as necessary. There should be one track for each animal present in the video.
  - iii. Temporal gaps in the tracks are okay (see analysis below) although missing data should be minimized where possible. Files with many deleted points may indicate that further labeling and training is required.
- c. Save the corrected prediction file with a new name (e.g., we replace the suffix “predictions” with “tracked”).
- d. Export the analysis file (File > Export Analysis HDF5).
  - i. Verify using the terminal output that the correct number of tracks are present.
- e. Record the following details in the “video\_details” tab of *HCCS\_example\_project\_analysis\_template.xlsx* (this is required for the movement analysis step below):
  - i. h5\_filename (you can use *HCCS\_SLEAP-create-analysis-h5-list.bat* to create a list of all H5 files to copy into this column, rather than copying each manually).
  - ii. date (of the recorded video; e.g., 22-Jan-2021).
  - iii. time\_point (e.g., 9:35).
  - iv. group (e.g., Air).
  - v. start\_frame (this is the frame number that first shows the embedded time annotation for the start of the time point; e.g., first frame with time 09:35:00 is frame 8. Note, SLEAP shows the frame number in the bottom left of the screen).





**Figure 5. A cropped screenshot from SLEAP showing a representative night-time infrared-illuminated camera view with five mice and their centroid tracks**

- vi. end\_frame (this is the frame number that last shows the time for the end of the time point; e.g., last frame with time 09:39:59 is frame 9012).
  - vii. pixels\_per\_cm (see below for calculation).
  - viii. fps (e.g., 30).
  - f. Repeat this step for all predictions files.
14. Calculate the real-world video scale (pixels\_per\_cm).
- a. To determine the scale for each camera, use a video recording of an object of known size in the same position in each cage, e.g., we used the width of the feeding tray in the first recorded video for each group. Alternatively, you may print and use a checkerboard pattern of known size (e.g., *HCCS\_1cm\_checkerboard.png*; [Figure 6](#)). This pattern may also be used for assessing and correcting image distortion (see discussion in the [limitations](#) section).
    - i. You only need to determine this scale once for each camera, unless the position of the camera changes during the study.
  - b. Open SLEAP and save a new project (e.g., *HCCS\_example\_project-scale.slp*).
  - c. In the "Skeleton" panel, click "New Node" and rename it "scale".
  - d. Add the videos for measurement.
  - e. Add two instances and position them at either end of your object of known size.
  - f. Add a separate track to each instance (select instance; Tracks > Set Instance Track > New Track) and give them unique names (e.g., PiCam01\_S1; PiCam01\_S2).
  - g. Repeat for all cameras used in the study.
  - h. Export the analysis file and upload it to the "analysis" folder in Drive.
  - i. Open our Colab notebook *SLEAP\_Scale* and follow the instructions within.
    - i. This notebook will extract the instance coordinates from an H5 analysis file and save them in a CSV file for input into the analysis template.
  - j. Open the downloaded coordinates file and copy the data into the respective fields in the "scale" tab of *HCCS\_example\_project\_analysis\_template.xlsx*.
  - k. Copy the pixels\_per\_cm value for each camera to the appropriate rows in the "video\_details" tab.
    - i. Ensure that you paste the value and not the underlying formula.



**Figure 6.** A frame from a day-time video showing a 1 cm checkerboard pattern on the bottom of an empty home-cage. *HCCS\_1cm\_checkerboard.png* was printed, trimmed to size, and taped in position. This pattern may be used to determine the real-world scale of your video and assess the need for image distortion correction.

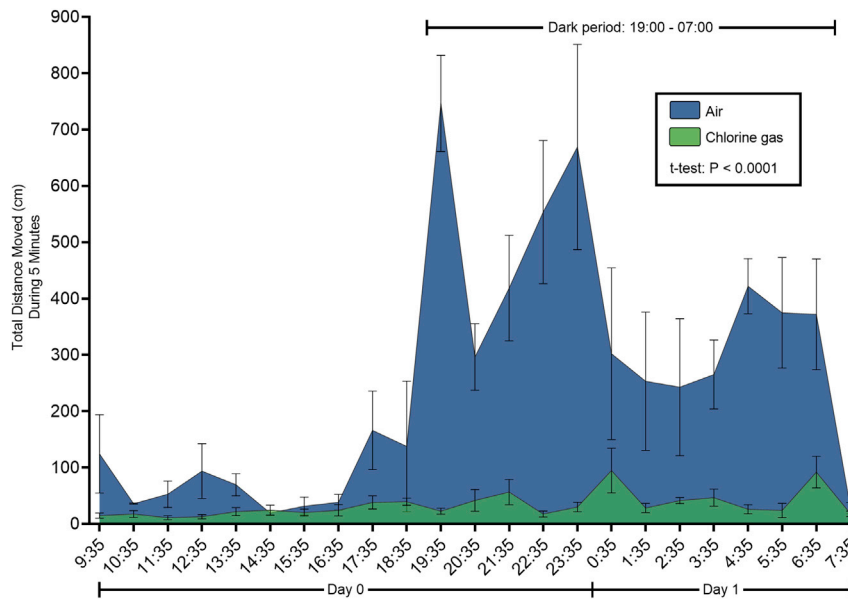
15. Calculate movement parameters.
  - a. Export the “video\_details” tab as a CSV file and upload it along with the video H5 analysis files to the “analysis” folder in Drive.
  - b. Open our Colab notebook *SLEAP\_Movement\_Analysis* and follow the instructions within.
    - i. This notebook will calculate the Total Distance Moved and Time Mobile for each animal at each time point and save them in a CSV file. The raw centroid movements of each animal can also be plotted.
    - ii. The method of missing data imputation can be chosen (in our example study, we used the previous available value to more accurately estimate the Total Distance Moved, however, this will underestimate the Time Mobile).
    - iii. To reduce the impact of any variability in frame-to-frame tracking, you can apply a movement threshold that zeros movements equal to or less than the set value (for our dataset, we used a value of 0.2 cm).

### Statistical analysis

⌚ Timing: 5–10 min

In this section, we describe how to calculate the area under the curve (AUC) for the movement parameters and statistically compare between two groups. We have provided an example GraphPad Prism project file (*HCCS\_example\_project\_AUC\_template.pzfx*) to help demonstrate the analysis.

16. Import the Total Distance Moved data into GraphPad Prism.
  - a. Create a grouped table with replicate subcolumns for the number of animals tracked.
17. Perform an AUC analysis and compare differences in mean AUCs with an unpaired t-test.
  - a. Run an AUC analysis (XY analyses).
  - b. Copy the “Total Peak Area” and the “Std. Error” values into a new grouped table formatted for entry of “Mean, SEM, N”.



**Figure 7. Spontaneous movement is significantly reduced in mice after exposure to chlorine gas**

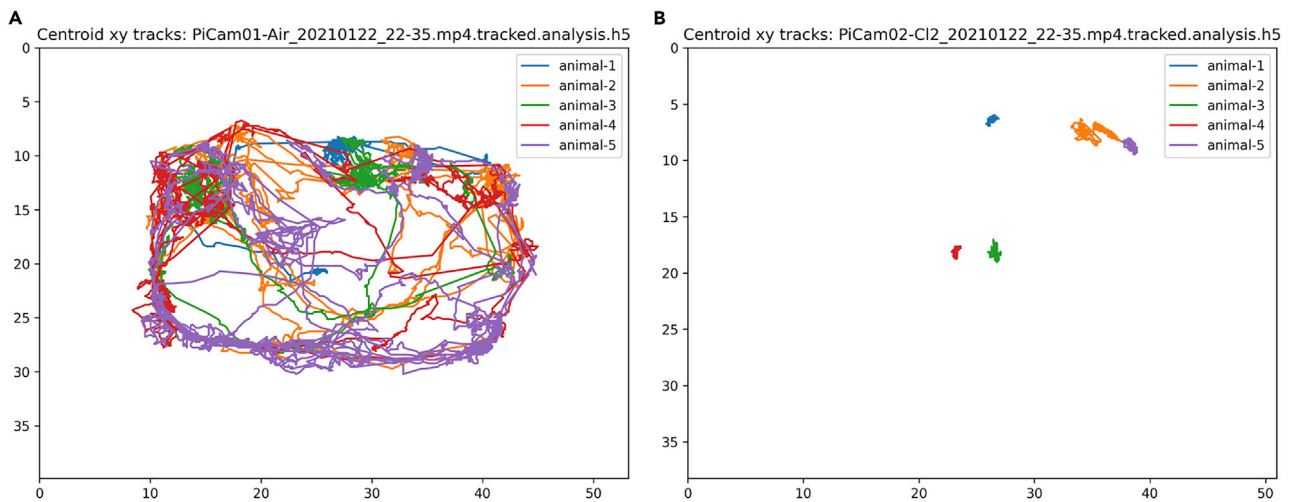
The area under the curve graph displays the mean ( $\pm$  SEM) Total Distance Moved during 5 min at each time point after a 30 min exposure to either 400 ppm chlorine gas or air. Control mice demonstrated a normal diurnal pattern of activity.

- c. The “N” column is calculated for each group as the degrees of freedom plus one (calculated as the number of data points minus the number of time points plus 1). In our example study, the “N” values were 73 and 87 for the Air and Chlorine groups, respectively.
  - d. Run an unpaired t-test (Column analyses).
18. Create an AUC graph using the data entered in step 16.
    - a. Create an XY graph with area fill, plotting the mean and SEM.
  19. Repeat the analysis with the Time Mobile data.

## EXPECTED OUTCOMES

Successful completion of this protocol will produce one or more real-time video monitoring systems with secure remote access that enables complete system control and video preview of the home-cage environment. By processing and analyzing the recorded videos as outlined in this protocol, you can obtain non-invasive temporal tracking data for multiple group-housed unmarked animals. This data may then be processed to objectively describe the spontaneous movement (or other behavioral characteristics) of each individual at each time point. Differences between individuals within a cage or between groups can be statistically tested and provide additional experimental insights.

With the HCCS, we successfully monitored and recorded video data for two groups of mice in an acute chlorine exposure study. In addition to ensuring the expected welfare of the mice was maintained via non-invasive and non-intrusive monitoring, analysis of the recorded data demonstrated that spontaneous movement is significantly reduced in mice after exposure to chlorine gas (see [Figures 7 and 8](#), and additional graphs in [HCCS\\_example\\_project\\_analysis\\_template.xlsx](#)). Whilst the normal diurnal pattern of activity was demonstrated by control mice, it was essentially abolished in chlorine-exposed mice. Our findings are similar to the decreased movement observed in chlorine-exposed mice when intermittently removed from their home-cage and individually recorded in an empty testing cage ([Filippidis et al., 2012](#)), however, our system offers the advantage of removing the adverse and potentially confounding impact of human physical presence and handling on the mice. In doing so, exploration of additional hypotheses and conditions are possible (e.g., differences between day-time and genuine night-time lighting conditions; a greater number of time points) and provides the flexibility to *post hoc* assess and accommodate unexpected occurrences without necessarily having to repeat experiments. In



**Figure 8. Raw centroid movement plots produced by our Colab notebook *SLEAP\_Movement\_Analysis***  
Shown are xy track plots of (A) air- and (B) chlorine-exposed mice during a representative 5 min night-time time point.

ongoing investigations, we are using this system to quantify the severity of models and the efficacy of therapeutic interventions, as a supplement to other experimental outcomes.

## LIMITATIONS

Whilst our system offers several distinct advantages over currently available commercial cage-monitoring systems, in particular, there is no inherent limitation to the number of individuals that can be video-tracked simultaneously, there are some potential limitations that should be considered.

The single top-down field of view is subject to vertical occlusions and there may be gaps in tracking or it might not even be possible to track some individuals at some time points, depending on the enclosure design (e.g., in our presented study, primarily during daytime periods, two mice sleeping on top of the shelter would occlude the visibility of mice within the shelter below). The need to replace the commonly-used overhead food basket with a floor or wall-mounted receptacle may also not be practical for all enclosures. As a possible solution, the camera could be positioned to obtain an oblique or side view, or an additional synchronized camera (or several) could be incorporated into the system. The development of flexible 3D tracking systems is an area of active research with several emerging technologies (Marshall et al., 2021).

In this protocol, we did not adjust for any image distortion introduced by the wide-angle nature of the camera lens used. This is a possible non-trivial preprocessing step that requires specific calibration for each camera/lens combination and may be undertaken prior to analyzing the videos with SLEAP. The degree of image distortion may be qualitatively and quantitatively assessed by observing the distortion of a checkerboard pattern (Figure 6). Videos may be re-encoded with the distortion computationally corrected, or instance coordinates may be remapped to their “correct” position (e.g., see [https://docs.opencv.org/4.5.5/dc/dbb/tutorial\\_py\\_calibration.html](https://docs.opencv.org/4.5.5/dc/dbb/tutorial_py_calibration.html)). However, we found this step to be unnecessary with our system for two key reasons. Firstly, by measuring the horizontal and vertical pixel distance of each 1 cm square in 3 cm × 3 cm grids positioned within each corner and the center of the home-cage, we identified the maximum difference in mean pixel distance between areas to be only 2.9 pixels. This is equivalent to 0.11–0.12 cm, depending on which area was used as the “ground-truth”. Secondly, because we are measuring frame-to-frame movements, the distance any instance can move between frames is relatively short and all movements may be considered to occur within a manifold, that is, all movements occur within a local neighborhood of Euclidean space. Hence, the small amount of image distortion across the home-cage has a negligible impact on the calculated movement parameters.

## TROUBLESHOOTING

### Problem 1

(Before you begin, [video hardware and software setup](#), step 3).

The CPU temperature is colored orange or red and the system may be performing slowly.

#### Potential solution

The RPi will throttle (slow down) the CPU and GPU when its temperature approaches or exceeds 85°C. In addition to coloring the CPU temperature, thermometer icons may also be displayed. The throttling is designed to prevent overheating from causing damage to the RPi and no action is necessarily required. However, performance and functionality of the system may be impacted. Adding heatsinks to the RPi, installing a fan, and/or removing the protective case can help to better dissipate heat.

### Problem 2

(Before you begin, [video hardware and software setup](#), step 7).

When connecting to the RPi using VNC, you see a black background with the message “Cannot currently show the desktop”.

#### Potential solution

This is considered normal behavior for the message to appear briefly. If your desktop does not display, you may need to change the display resolution setting using either the terminal with an attached screen or via SSH:

```
>sudo raspi-config
```

### Problem 3

(Before you begin, [video hardware and software setup](#), step 8).

You have no camera preview on the attached monitor.

#### Potential solution

Check that the camera cable is inserted the correct way for both the RPi and camera connections.

### Problem 4

(Before you begin, [video hardware and software setup](#), step 8).

You have no camera preview via VNC but you do on the attached monitor.

#### Potential solution

Enable “direct capture mode” in VNC Server on the RPi (Options > troubleshooting).

### Problem 5

(Before you begin, [video hardware and software setup](#), step 8).

There are horizontal interference lines in the camera preview/recording.

#### Potential solution

Check the connections between the LEDs and the camera board. Insulate the wires to reduce interference between themselves and other equipment.

## RESOURCE AVAILABILITY

### Lead contact

Further information and requests for resources should be directed to and will be fulfilled by the lead contact, Alan David Marcus ([alan.david.marcus@gmail.com](mailto:alan.david.marcus@gmail.com)).

### Materials availability

This study did not generate new unique reagents.

### Data and code availability

Customized software and code is available from this protocol's GitHub repository: <https://github.com/Alan-Marcus/HCCS>. This repository has been archived at Zenodo: <https://10.5281/zenodo.6377089>.

There are restrictions to the availability of our animal video data due to institutional policies.

## ACKNOWLEDGMENTS

This work was supported by U01ES030672 (S.E.J. and S.A.), 1R21ES030331-01A1 (S.A. and S.E.J.), and 1R21ES033020 (S.A.) of the NIH Countermeasures Against Chemical Threats (CounterACT) program. The content is solely the responsibility of the authors and does not necessarily represent the views of the NIH. We thank Jesse DeGraff and Duke Laboratory Animal Resources for their assistance with the animal housing equipment and management of animal husbandry.

## AUTHOR CONTRIBUTIONS

Conceptualization, A.D.M., S.A., and S.E.J.; methodology, A.D.M. and S.A.; software, A.D.M.; formal analysis, A.D.M.; investigation, A.D.M. and S.A.; writing – original draft, A.D.M.; writing – review & editing, A.D.M., S.A., and S.E.J.; supervision, S.A. and S.E.J.; funding acquisition, S.A. and S.E.J.

## DECLARATION OF INTERESTS

The authors declare no competing interests.

## REFERENCES

- Achanta, S., and Jordt, S.E. (2021). Toxic effects of chlorine gas and potential treatments: a literature review. *Toxicol. Mech. Methods* *31*, 244–256.
- Achanta, S., and Jordt, S.E. (2020). Transient receptor potential channels in pulmonary chemical injuries and as countermeasure targets. *Ann. N. Y. Acad. Sci.* *1480*, 73–103.
- Balakrishna, S., Song, W., Achanta, S., Doran, S.F., Liu, B., Kaelberer, M.M., Yu, Z., Sui, A., Cheung, M., Leishman, E., and Eidam, H.S. (2014). TRPV4 inhibition counteracts edema and inflammation and improves pulmonary function and oxygen saturation in chemically induced acute lung injury. *Am. J. Physiol. Lung Cell Mol. Physiol.* *307*, L158–L172.
- Filippidis, A.S., Zarogiannis, S.G., Randich, A., Ness, T.J., and Matalon, S. (2012). Assessment of locomotion in chlorine exposed mice by computer vision and neural networks. *J. Appl. Physiol.* *112*, 1064–1072.
- Marshall, J.D., Li, T., Wu, J.H., and Dunn, T.W. (2021). Leaving flatland: advances in 3D behavioral measurement. Preprint at arXiv. <https://doi.org/10.48550/arXiv.2112.01987>.
- Nilsson, S.R., Goodwin, N.L., Choong, J.J., Hwang, S., Wright, H.R., Norville, Z.C., Tong, X., Lin, D., Bentzley, B.S., Eshel, N., and McLaughlin, R.J. (2020). Simple behavioral analysis (SimBA)—an open source toolkit for computer classification of complex social behaviors in experimental animals. Preprint at BioRxiv. <https://doi.org/10.1101/2020.04.19.049452>.
- Noorshams, O., Boyd, J.D., and Murphy, T.H. (2017). Automating mouse weighing in group homecages with Raspberry Pi micro-computers. *J. Neurosci. Methods* *285*, 1–5.
- Pereira, T.D., Tabris, N., Matsliah, A., Turner, D.M., Li, J., Ravindranath, S., Papadoyannis, E.S., Normand, E., Deutsch, D.S., Wang, Z.Y., McKenzie-Smith, G.C., Mitelut, C.C., Castro, M.D., D'Uva, J., Kislin, M., Sanes, D.H., Kocher, S.D., Wang, S.S.-H., Falkner, A.L., Shaevitz, J.W., and Murphy, M. (2022). Sleep: A deep learning system for multi-animal pose tracking. *Nature Methods*. <https://doi.org/10.1038/s41592-022-01426-1>.
- Schneider, T., and Lütkefend, T. (2019). Nowhere to Hide: The Logic of Chemical Weapons Use in Syria (Global Public Policy Institute).
- Singh, S., Bermudez-Contreras, E., Nazari, M., Sutherland, R.J., and Mohajerani, M.H. (2019). Low-cost solution for rodent home-cage behaviour monitoring. *PLoS One* *14*, e0220751.
- Voikar, V., and Gaburro, S. (2020). Three pillars of automated home-cage phenotyping of mice: novel findings, refinement, and reproducibility based on literature and experience. *Front. Behav. Neurosci.* *14*, 575434.
- Woodard, C.L., Nasrallah, W.B., Samiei, B.V., Murphy, T.H., and Raymond, L.A. (2020). PiDose: an open-source system for accurate and automated oral drug administration to group-housed mice. *Sci. Rep.* *10*, 1–10.