

<sup>1</sup> ***pi\_tailtrack: A compact, inexpensive, and open-source behaviour-tracking system for head-restrained zebrafish***

<sup>4</sup> **Owen Randlett<sup>1</sup>**

<sup>\*Correspondence:</sup>

<sup>5</sup> <sup>1</sup> Laboratoire MeLiS, UCBL - CNRS UMR5284 - Inserm U1314, Institut NeuroMyoGène, Faculté de Médecine et de Pharmacie, 8 avenue Rockefeller,  
<sup>univ-lyon1.fr</sup> (OR) <sup>6</sup> 69008, Lyon, France

<sup>7</sup> **Abstract**

<sup>8</sup> Quantifying animal behavior during microscopy is crucial to associate optically recorded neural activity with  
<sup>9</sup> behavioural outputs and states. Here I describe an imaging and tracking system for head-restrained larval  
<sup>10</sup> zebrafish compatible with functional microscopy. This system is based on the Raspberry Pi computer, Pi NoIR  
<sup>11</sup> camera, and open-source software for the real-time tail segmentation and skeletonization of the zebrafish tail at  
<sup>12</sup> over 100hz. This allows for precise and long-term analyses of swimming behaviour, that can be related to  
<sup>13</sup> functional signals recorded in individual neurons. This system offers a simple but performant solution for  
<sup>14</sup> quantifying the behavior of head-restrained larval zebrafish, which can be built for 340€.  
<sup>15</sup>

<sup>16</sup>

<sup>17</sup> **Introduction**

<sup>18</sup> A chief application of the larval zebrafish for neuroscience is to image the activity of neurons in the intact and  
<sup>19</sup> behaving animal using microscopy. This is facilitated by its translucent and small brain, measuring approximately  
<sup>20</sup> 0.1 mm<sup>3</sup>. By expressing genetically encoded indicators, such as the GCaMP Ca<sup>2+</sup> sensors (*Akerboom et al., 2012*;  
<sup>21</sup> *Chen et al., 2013*), signals related to the activity of practically any or all neurons can be recorded from the larval  
<sup>22</sup> zebrafish brain (*Ahrens et al., 2012; Portugues et al., 2014*).

<sup>23</sup> Ca<sup>2+</sup> imaging can be performed with standard microscopes, but such systems are not equipped for monitoring  
<sup>24</sup> the behaviour of the animal. Therefore, any analyses directly relating neural activity to behaviour will require the  
<sup>25</sup> integration of a behavioural recording apparatus. Behavioural recording is typically done in the context of custom-  
<sup>26</sup> built microscopes, which can be designed explicitly with this behaviour-monitoring goal in mind. However, many  
<sup>27</sup> groups (including my own) have neither the financial or technical means to implement such a complete system.  
<sup>28</sup> We rely on microscope equipment in a shared core facility. Such microscopes generally cannot be substantially or  
<sup>29</sup> permanently modified, and often present physical and optical constraints that make installing a behaviour imaging  
<sup>30</sup> system challenging.

<sup>31</sup> Here I present a solution for this problem based on the Raspberry Pi computer, that I call *pi\_tailtrack*. The system  
<sup>32</sup> includes illumination, camera, computer and software, yielding a complete setup that is compact, inexpensive, and  
<sup>33</sup> self-contained. The *pi\_tailtrack* system can reliably track larval zebrafish behaviour at over 100hz while performing  
<sup>34</sup> functional imaging experiments.

## 35 Results and Discussion

### 36 Design goals

37 I wanted to track the swimming behaviour of head-restrained larval zebrafish while performing  $\text{Ca}^{2+}$  imaging. There  
38 are many ways that this might be accomplished, but I wanted a system that was:

- 39 1. Able to identify and characterize individual swimming events while we are imaging the brain using 2-photon  
40 microscopy.
- 41 2. Compact and self contained, so that it can be easily and rapidly installed and removed for our imaging sessions  
42 on a shared microscope.
- 43 3. Made using low-cost and open source hardware and software to facilitate re-use in other contexts, and be-  
44 cause I am a ~~cheap~~ financially responsible researcher.

### 45 Using a Raspberry Pi camera to image the larval zebrafish tail

46 The Raspberry Pi is a very inexpensive,  
47 credit-card-sized computer that plugs into  
48 a standard monitor, keyboard, and mouse.  
49 The Raspberry Pi's open-source nature  
50 and large user community, and its abil-  
51 ity to control and interface with a vari-  
52 ety of devices and sensors make it a pow-  
53 erful and accessible platform for devel-  
54 oping and sharing custom neuroscience  
55 and behavioural research tools. Indeed  
56 many such systems have been developed  
57 in recent years based around the Rasp-  
58 berry Pi and the Pi Camera, and especially  
59 the IR-sensitive Pi NoIR camera, as an ac-  
60 quisition device (Geissmann *et al.*, 2017;  
61 Maia Chagas *et al.*, 2017; Saunders *et al.*,  
62 2019; Tadres and Louis, 2020; Broussard  
63 *et al.*, 2022).

64 However, obtaining sufficient resolu-  
65 tion and contrast to resolve the larval ze-  
66 brafish tail is challenging since the tail  
67 is very narrow ( $\approx 0.25\text{mm}$  diameter), and  
68 nearly transparent. This is especially  
69 true in de-pigmented animals that are  
70 generally used for brain imaging due to  
71 their lack of melanin pigment over the  
72 brain (e.g. *mifta*/Nacre mutants, or larvae  
73 treated with N-Phenylthiourea). This also  
74 removes melanin pigment from the tail, increasing its transparency and making it harder to image and track. Thus,  
75 it was not clear if the 26€ Pi NoIR Camera would be up to this task.

76 The stock lens configuration on the Pi Camera is also not designed for macro photography, and has a minimum  
77 focus distance of 50cm. But, extension tubes are a well-known macro-photography hack that work by increasing the  
78 distance between the lens and the camera (Wikipedia contributors, 2022). Increasing this distance acts to decreases  
79 the focus distance of the optical system, increasing the maximal magnification. By unscrewing the lens of the Pi  
80 NoIR camera until just before it falls off, it is possible to focus on objects at a 2 cm distance, allowing for sufficient  
81 magnification to observe the and track the tail of *mifta* mutant zebrafish (Figure 1, Figure 3).

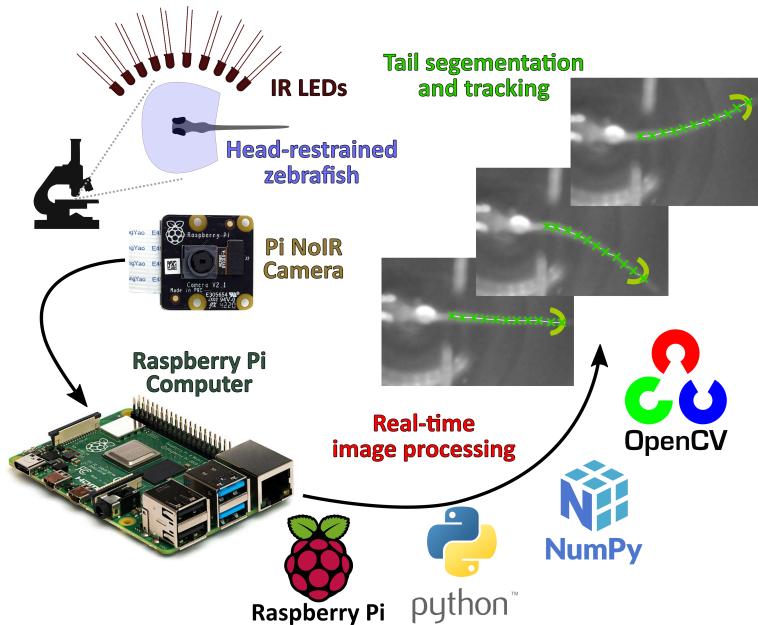


Figure 1. Schematic for the *pi\_tailtrack* apparatus.

The zebrafish larvae being imaged under the microscope is illuminated with infra-red (IR) LEDs, and imaged with the IR-sensitive Raspberry Pi NoIR Camera. Image acquisition and processing is done with a Raspberry Pi Computer and open-source Python packages. The zebrafish tail is identified and segmented in real-time as a sequence of 10 tail segments (green X's).

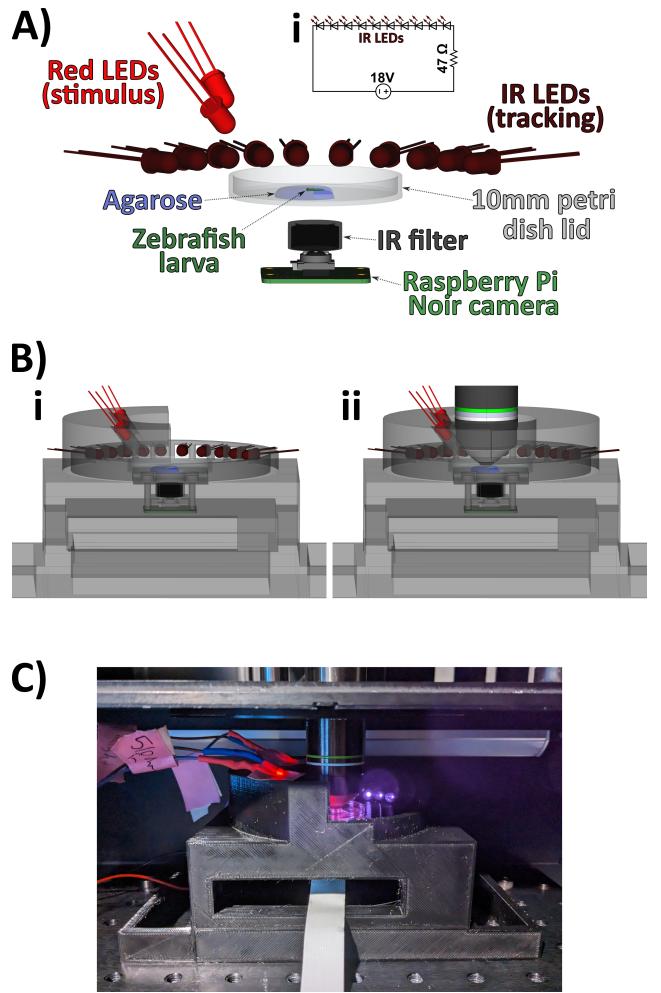
82 A second challenge is that larval zebrafish move  
 83 their tails very rapidly, with a tail-beat frequency of  
 84 between 20-40 hz for normal swimming, which can in-  
 85 crease to 100 hz during burst/escape swimming (*Bu-*  
 86 *dick and O'Malley, 2000; Muller, 2004; Severi et al.,*  
 87 *2014*). The V2.1 camera documentation indicates  
 88 maximum frame rate of 30hz, which is insufficient  
 89 for imaging tail dynamics. However, by adopting a  
 90 cropped sensor configuration, and by omitting the  
 91 JPG compression step in image processing, the cam-  
 92 era can be pushed to image at up to 1000hz (*Elder,*  
 93 *2019*). I adopted a configuration where I image with  
 94 a cropped sensor of 128x128 pixels, which gives suf-  
 95 ficient spatial resolution to observe and track the tail  
 96 of the fish, and most importantly, minimal CPU load.  
 97 This frees the limited CPU resources on the Raspberry  
 98 Pi to be used for real-time image processing and tail  
 99 tracking.

## 100 Hardware Setup

101 The short 2 cm focal distance between the animal and  
 102 the camera allowed for a compact and direct imaging  
 103 setup, where the camera is mounted directly below  
 104 the larva (*Figure 2A*). This avoids the need for any mir-  
 105 rrors, and frees the space above the animal for the mi-  
 106 croscope objective, and any stimulus apparatus nec-  
 107 essary. In our case we use red LEDs to provide visual  
 108 stimuli to the larvae (*Lamiré et al., 2022*).

109 To illuminate the larvae and visualize the tail, I  
 110 used 890nm IR LEDs. Using the IR LEDs as an oblique  
 111 illumination source generated a nicely resolved im-  
 112 age of the *mifta* mutant zebrafish tail that was suffi-  
 113 cient for reliable identification and tracking (*Figure 3*).  
 114 IR LEDs were wired in a simple circuit, with 10 LEDs  
 115 in a series, powered by a 18V DC power supply and  
 116 a 47ohm current limiting resistor (*Figure 2Ai*). Us-  
 117 ing these exact Voltage/Resistance configurations is  
 118 not important, provided a relevant power supply and  
 119 resistor are chosen to match the LED characteris-  
 120 tics (forward voltage =1.4V, current = 100mA, for our  
 121 890nm LEDs: see for example [amplifiedparts.com:  
LED Parallel/Series Calculator](http://amplifiedparts.com/LED Parallel/Series Calculator)).

123 We used an 880 nm bandpass filter in front of the  
 124 Raspberry Pi NoIR camera module to selectively pass the IR LED light. This filter is essential to block the intense  
 125 microscope laser light, which will obscure the image of the fish by saturating (and likely damaging) the camera  
 126 sensor. Notably, this filter it is the most expensive part in the setup, costing more than the computer and camera,  
 127 combined (*Table 1*). With our typical 2-photon GFP/GCaMP imaging settings and the laser tuned to 930nm, laser  
 128 light is not visible in the camera image. Using such a bandpass filter in the 880 nm range should allow this system  
 129 to be compatible with many other imaging modalities (confocal, epifluorescence, brightfield, etc), provided that  
 130 the excitation wavelengths are not in the ≈870-900nm range, and the microscope system effectively filters out the  
 131 890nm light from the LEDs. If necessary, these wavelength characteristics can be adapted using different LED and



**Figure 2. *pi\_tailtrack* hardware.**

A) CAD drawing of the main components of the apparatus. IR leds illuminate the zebrafish larvae that is head-restrained in agarose in a 35mm diameter petri dish lid. An IR filter blocks the visible stimulus lights (Red LEDs), and the microscope laser from reaching the Raspberry Pi NoIR camera suspended below the fish. ii) Wiring diagram for powering the IR LEDs.

B) same as (A), but with i) the 3D printed mount visible, and ii) the microscope objective and semicircle shape component to encircle the objective.

C) Photo of the apparatus within the microscope, without the semicircle shape component. IR LED light is visible as the indigo colour.

132 filter components.

133 To house the system components I used a 3D printed mount (*Figure 2B,C*). This was designed using FreeCAD  
134 ([freecad.org](http://freecad.org), [FreeCAD file](#)), and 3D printed in black PETG using and Creality Ender 3 Pro 3D printer. It consists of the  
135 main body shape that holds the the camera, IR filter, red stimulus LEDs above the fish, and IR LEDs in the oblique  
136 illumination configuration ([Main Shape](#)). An insert is placed into the depression above the IR filter, forming the plat-  
137 form onto which the zebrafish dish is placed ([Depression Insert](#)). The final 3D printed component is a semicircular  
138 shape that completes the encirclement of the objective, and helps minimize light scattering (*Figure 2Bi*, [Semicircle](#)  
139 [STL file](#)).

140 I would note that I built up the size of the platform of the mount to match with the relatively spacious config-  
141 uration of the microscope I was using (*Figure 2C*). A much more compact configuration is possible, since we only  
142 require  $\approx$ 26 mm of clearance from the fish to the bottom of the  $\approx$ 6 mm thick camera. The base design could be  
143 easily adapted to match different microscope stage configurations. For example the entire system could be inverted  
144 to accommodate an inverted microscope (though performing behavioural experiments on inverted zebrafish may  
145 be sub-optimal). Or, if stimuli need to be bottom-projected, a small 45-degree hot mirror could be used to divert  
146 the image to the camera and free the space directly beneath the animal for stimuli.

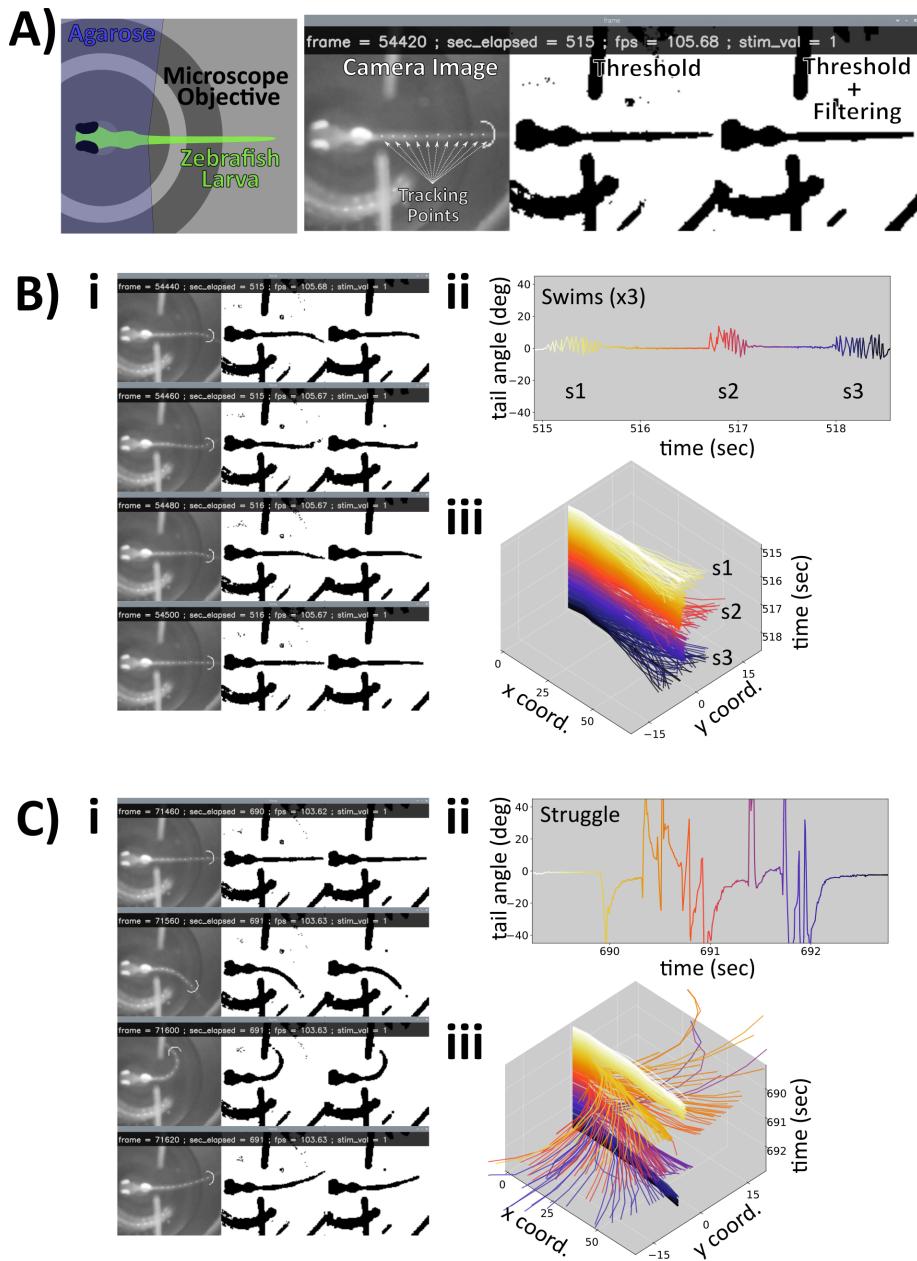
#### 147 **Software and tail tracking strategy**

148 Tracking objects in images and videos has undergone a revolution with deep learning and neural network frame-  
149 works, where the tracking and reconstruction of complex animal postures is possible after training networks on only  
150 a few example images ([Mathis et al., 2018](#); [Pereira et al., 2022](#)). However, such approaches are computationally  
151 intensive and generally require dedicated and GPU hardware beyond the capabilities of the standard Raspberry Pi,  
152 making them incompatible with our project goals. In contexts where the image background is predictable and stable,  
153 classical computer vision methods like background subtraction, filtering and thresholding may still be preferable to  
154 network-based object identification, especially when speed or computational resources are priorities. Here I have  
155 used the *Numpy* ([Harris et al., 2020](#)) and *OpenCV* ([Bradski, 2000](#)) libraries to handle the image data and computer  
156 vision tasks (*Figure 1*).

157 Image frames are acquired directly from the camera buffer as an 8-bit Numpy array, and thresholded using  
158 Adaptive Thresholding (`cv2.adaptiveThreshold`) to identify bright objects in the image (*Figure 3*, "Threshold"). This  
159 binary image is then filtered using a morphological Opening and Closing operation (`cv2.morphologyEx`). This combi-  
160 nation generally results in a nicely segmented fish blob in the final binary image (*Figure 3A*, "Threshold + Filtering").  
161 However, this method identifies all large bright objects in the image, including borders of the agarose block and  
162 reflections on the microscope objective, and therefore we need a method to identify the fish object among these  
163 various segmented blobs.

164 The fish object is identified with a pre-defined coordinate that acts as the first tracking point of the fish. The fish  
165 object is then skeletonized into up to 10 tail segements (*Figure 3A*, 'Tracking Pts'), which can be used to reconstruct  
166 the posture of the tail to identify swimming events (*Figure 3B,C*). To do this skeletonization, the tracking points are  
167 iteratively identified based on the intersection of a semicicle and the fish object, offset 7 pixels from the previous  
168 tracking point, and oriented in the direction of the previous segment (similar to [Štih et al. \(2019\)](#); [Randlett et al.  
\(2019\)](#)). For the first search, this direction is toward the right of the image. Therefore, this strategy relies on the  
169 zebrafish larvae being oriented with its tail pointed towards the right, and being placed in the same position such  
170 that the exit point of the tail from the agarose block intersects with the fist tracking point. It also requires that  
171 no other bright objects intersect with the fish object after binarization. Therefore, it is critical to avoid distracting  
172 objects in the imaging scene, such as scratches in the dish or stray pieces of agarose.

174 This computationally lean segmentation and skeletonization strategy takes less than 10 ms on the Raspberry  
175 Pi CPU. The imaging frame rate when using the *picamera* python package will adjust based on the throughput of  
176 the analysis system, which can change with the complexity of the binary images that are processed or external  
177 CPU demands, but runs at approximately 104hz (*Figure 3*, 'fps ='). This is sufficient to clearly distinguish different  
178 types of movement events, such as swims from escapes/struggles *Figure 3B* vs C), and where individual tail beats  
179 during swimming events are resolvable. However, this will not be true during rapid/burst swimming, in which tail  
180 beat frequency will be exceed our frame rate. If such temporal resolution is required our setup will be insufficient,  
181 and we will only reliably track tail half-beat frequencies of  $\leq$ 50hz. Therefore, this system is not capable of



**Figure 3. Larval zebrafish trail tracking examples.**

**A)** Screenshot of a single frame of a tracking image, showing the image from the camera ("Camera Image") with the resultant tracking points overlaid as white dots. The final tracking point is shown as a white semicircle, which is used in the coordinate search algorithm. "Threshold" shows the result of the Adaptive Thresholding operation, and "Threshold + Filtering" the result of the morphological Opening and Closing operations. Displayed along the top are the: frame (current frame number of the experiment), sec\_elapsed (number of seconds elapsed in the experiment), fps (current frame rate, frames per second), stim\_val (the current value read on the stimulus recording pin: GPIO Pin 4). A schematic of the image field, depicting the agarose mounting medium, the position of the zebrafish, and the microscope objective visible in the background is shown in the left panel.

**B)** i) Example frames during a swimming event. ii) Tail angle deflections during 3 distinct swim events. iii) 3D plot of tail coordinates through the same time period as (ii), drawn in the same time color code.

**C)** Same as (B), but for a period in which the larvae executes a struggle/escape maneuver and associated high amplitude tail deflections.

**Figure 3—video 1.** Screen recording of the tail tracking example, [download](#)

182 comprehensive behavioural characterization, but is impressively capable considering this is done with 26€ camera  
183 and a 95€ computer.

184 During the experiment the software provides a visual display, as shown in the screenshots in (*Figure 3*), and  
185 screen capture video (*Figure 3-video 1*). Results of the thresholding, filtering, and skeleton tracking are visible  
186 and updated in real-time. This can be used to optimize the position of the zebrafish, the Adaptive Thresholding  
187 parameters (neighborhood, threshold) using the 'w/a/s/d' keys, and the position of the first tracking point using the  
188 arrow keys.

## 189 **Output data format**

190 The tail tracking data are saved in a comma-separated text file '*\*\_coords.txt*', the 10 pairs of "X" and "Y" coordinates  
191 for each tail point are saved as rows, and thus there are two rows with 10 columns for every tracked frame. 'NaN'  
192 values represent instances where a tail point is not identified.

193 The timing of the data is saved in a separate text file '*\*\_tstamps.txt*', which also has two rows for each frame. The  
194 first value is the "timestamp" reflecting the time elapsed since the beginning of the tracking experiment. This is  
195 used to relate the tail posture and behavioural events to specific points in time. This is important for experiments  
196 in which precise timing of behavioural events is important, because the frame rate is not fixed and can fluctuate  
197 during the experiment (see above).

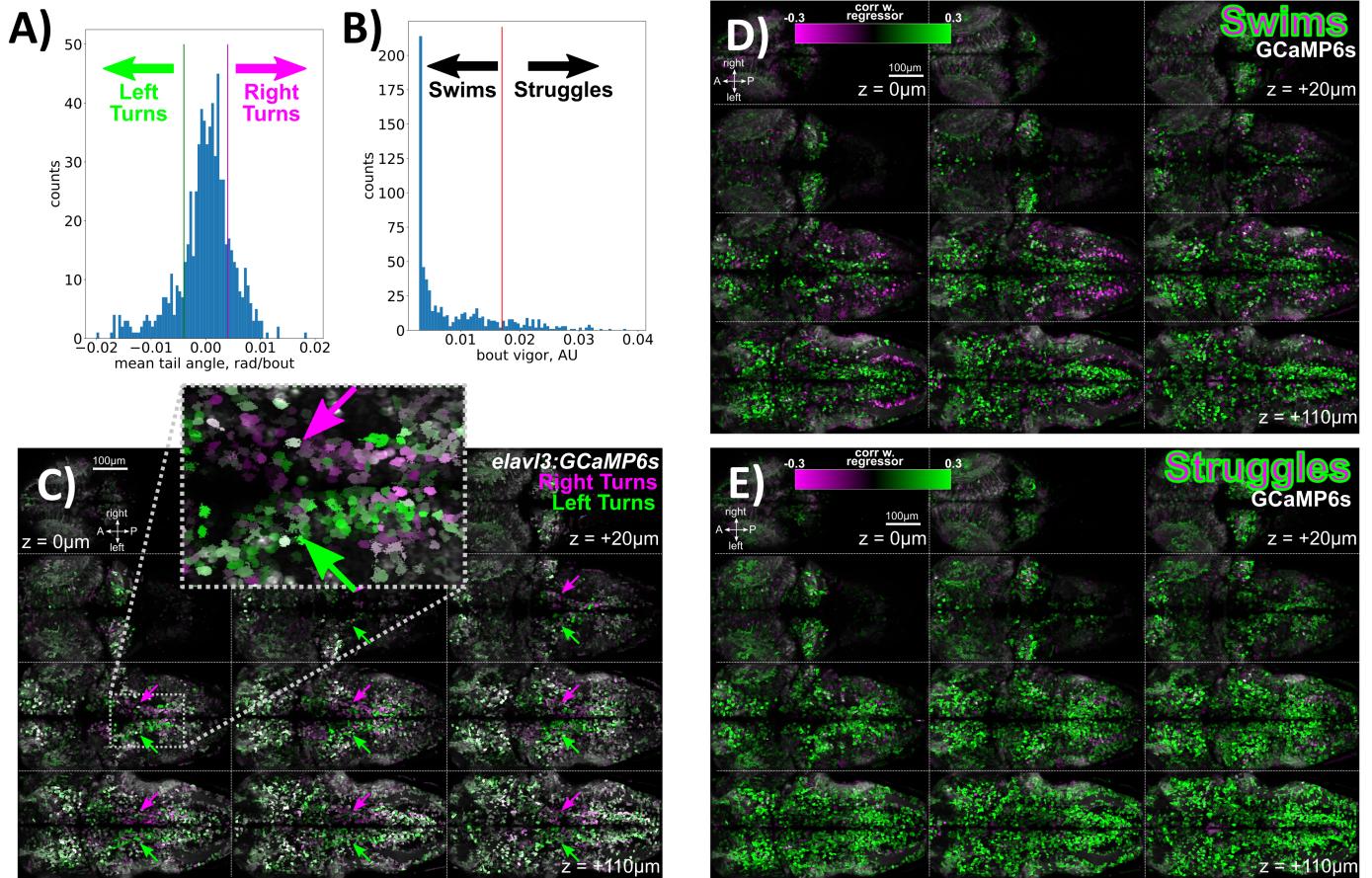
198 The second value in the '*\*\_tstamps.txt*' file is the value recorded on one of the GPIO pin 4 of the Raspberry Pi.  
199 This value will read either "low"=0 for a voltage less than 1.8V, or "high"=1 for 1.8-3.3V. I use these recordings to  
200 synchronize the behavioural recordings with the frames recorded on the microscope. In our typical setup we are  
201 using an analog output pin from the DAQ board on the microscope to control the red stimulus LEDs (*Figure 2A*), and  
202 we also connect this output of the DAQ board to GPIO pin 4 on the microscope. In this way, we can synchronize the  
203 stimuli, microscope imaging frames, and the behavioural recordings.

204 These datasets can be read into python for analysis using, for example:

```
205     1 import numpy as np
206
207     2
208     3 # load tracking coordinates
209     4 data = np.loadtxt('*_coords.txt', delimiter=',')
210
211     5
212     6 # separate 'x' and 'y' coordinates of tracking points
213     7 x_coords = data[:,0]
214     8 y_coords = data[:,1]
215
216     9
217    10 # load timing data
218    11 t_stamps = np.loadtxt('*_tstamps.txt', delimiter=',')
219
220    12
221    13 # separate 'timestamps' and 'stimulus state' recordings
222    14 time = t_stamps[:,0]
223    15 stim = t_stamps[:,1]
```

## 220 **Behavioural analysis of Ca<sup>2+</sup> imaging data**

221 To test the performance of the *pi\_tailtrack* system, I analyzed Ca<sup>2+</sup> imaging data from an 80 minute-long volumetric  
222 recording covering a large proportion of the brain (as in *Lamiré et al. (2022)*). To identify neurons tuned to be-  
223 havioural parameters I used "regressors" derived from the *pi\_tailtrack* recordings reflecting different motor states  
224 convolved with the GCaMP response kernel (as in *Miri et al. (2011)*). Zebrafish swim bouts can be classified as either  
225 forward swims or turns, and an area within the anterior hindbrain is associated with turning direction. This area is  
226 known as the Anterior Rhombencephalic Turning Region (ARTR: *Dunn et al. (2016)*, also called the HBO: *Ahrens et al.*  
227 (*2013*); *Wolf et al. (2017)*), and shows a conspicuous activity pattern with stripes of neurons tuned to the ipsilateral  
228 turning direction. By looking at correlations to regressors reflecting right and left turns, I identified these stripes  
229 of neurons in the ARTR-region, indicating that I can successfully identify the ARTR using *pi\_tailtrack* (*Figure 4A,C*).



**Figure 4. Identification of behaviour-associated neurons in a larval zebrafish brain via 2-photon  $\text{Ca}^{2+}$  imaging.**

**A)** Histogram for the mean tail angle during individual movement bouts for a single larva over an 80 minute imaging session. Bouts are classified as left or right turns based on a threshold value of 0.004 radians/bout.

**B)** Histogram for the bout vigor, quantified using a rolling standard deviation of absolute tail angle. Movements are classified as "swims" or "struggles" based on a threshold value of 0.017 (AU: arbitrary units).

**C)** Tuning of  $\text{Ca}^{2+}$  traces in ROIs to turns to the left (green) or right (magenta), as classified in (A). Images are the Pearson correlation coefficient to each behavioral regressor (left or right turns), scaled from 0.0 to 0.3. *Tg2(elavl3:GCaMP6s)* expression pattern is shown in grey. Arrows highlight the Anterior Rhombencaphalic Turning Region (ARTR): with ipsilateral tuning to turning direction. A = Anterior, P = Posterior

**D, E)** Tuning of neurons to swims (**D**), and struggles (**E**), as classified in (B).

230 A similar analysis looking at "swims" vs "struggles", with "struggles" reflecting high-amplitude tail flicking events  
231 (**Figure 3C**, **Figure 4B**), identified differential neuronal activation in the context of these two movement categories  
232 (**Figure 4B,D,E**), with the presence of lateral hindbrain populations of neurons that were negatively correlated with  
233 "swims", and a broader and more positively correlated population with "struggles".

### 234 **Future developments**

235 Here I have used the *pi\_tailtrack* system to simply record the behaviour of the animal independent of the microscopy  
236 or any stimulus delivery. Therefore, the timing of microscope image acquisition is controlled by the microscope  
237 computer and is independent of *pi\_tailtrack*. These separate experimental clocks (microscope frames vs Pi Camera  
238 frames) must be synchronized, and in our case I have used the GPIO input pin on the Raspberry Pi to record the  
239 timing of the stimuli delivered by the microscope relative to the Pi Camera frames. An alternative solution would  
240 be to use the Raspberry Pi to deliver the stimuli, perhaps by integrating a video projector system to allow for the  
241 delivery of arbitrary and complex visual stimuli. This would also open up possibilities for performing "virtual reality"  
242 experiments, where the behaviour of the animal dictates the stimulus in closed-loop. In some microscope systems  
243 it should also be possible to use the Raspberry Pi GPIO to trigger microscope acquisitions. This may be preferable  
244 if the synchronization between imaging and behaviour frames is critical.

245 It is also important to note that hardware in this micro-computer/Raspberry Pi space is rapidly evolving. Indeed,  
246 a new suite of Raspberry Pi V3 Cameras have just been released, offering increased resolution, dynamic range, and  
247 frame rate. Using these cameras, we may be able to increase the frame rate of tracking into the multiple-hundreds of  
248 hz, which would allow us to more reliably resolve individual tail half-beats. The Raspberry Pi "Global Shutter" Camera  
249 has also recently been released, which is likely also going to be very interesting for behavioural neuroscience, as  
250 the use of a global shutter avoids rolling shutter artifacts that distort images along the frame during rapid motion.

### 251 **Conclusion**

252 Here I described our system for tracking the tail of the larval zebrafish during microscopy. Many of the practical  
253 considerations of this setup may be specific to our application, and therefore may need modification for use in  
254 other experiments in other labs. However, I feel that the core and simple idea of using an IR-sensitive Raspberry  
255 Pi Camera, a simple Python script, coupled with IR LEDs and and IR filter, provides an approachable and flexible  
256 solution that may be widely useful for observing and tracking the behaviour of zebrafish (or perhaps other animals)  
257 while performing imaging experiments.

258 Even if this project is not directly useful to you or your research, I hope it can serve as an example of how  
259 by combining rudimentary knowledge of electronics hardware and Python scripting (supplemented with extensive  
260 use of Google and StackOverflow), it is possible to construct a very inexpensive but capable system. This system's  
261 attributes may also make it an ideal tool for community engagement activities such as school outreach programs. It  
262 could serve as a platform for learning about microelectronics, behavioural analyses, machine vision, and hardware  
263 design and construction.

## 264 **Methods**

### 265 **Animals**

266 All experiments were performed on larval zebrafish at 5 days post fertilization (dpf), raised at a density of  $\approx$ 1 lar-  
267 vae/mL of E3 media in a 14:10h light/dark cycle at 28-29°C. Adult zebrafish were housed, cared for, and bred at the  
268 Lyon PRECI zebrafish facility. *mitfa*/Nacre mutant animals (ZDB-ALT-990423-22) were used to prevent pigmentation.

269 Larval zebrafish were mounted and head restrained for 2-photon imaging and behavioural analysis by placing  
270 them in a very small drop of E3 in the lid of a 35mm petri dish (Greiner bio-one, 627102). Molten ( $\approx$ 42°C) 2% low  
271 melting point agarose (Sigma A9414) in E3 Medium was added to the dish in an approximately 10mm-diameter  
272 droplet around the fish, and the zebrafish was repositioned within the solidifying agarose using a gel-loading pipette  
273 tip, such that it was oriented symmetrically for imaging with the dorsal surface of the head at the surface of the  
274 agarose. After the agarose had solidified ( $\approx$ 10 minutes), E3 was added to the dish, and then the agarose around  
275 the tail was cut away. This was done using an scalpel in two strokes emanating laterally from just below the swim

**Table 1.** Bill of Materials

Component	Manufacturer	Cat. Number	≈Price (€)	Supplier/Link
Raspberry Pi Computer	Raspberry Pi Found.	4B Rev 1.4 8gb	95	<a href="#">kubii</a>
Pi NoIR Camera	Raspberry Pi Found.	NoIR v2.1	26	<a href="#">kubii</a>
24" Pi Camera Cable	Samtec	FJ-15-D-24.00-4	18	<a href="#">farnell</a>
880nm IR Bandpass filter	Edmund Optics	65-122	177	<a href="#">Edmund Optics</a>
890nm LEDs	Vishay Semiconductor	TSHF5410	$0.35 \times 10 = 4$	<a href="#">RS Components</a>
18V DC power supply <sup>1</sup>	generic, for IR LEDs	min ≈200mA	15	e.g. <a href="#">amazon.fr</a>
Current Limiting Resistor <sup>1</sup>	generic, for IR LEDs	minimum 1W power	1	e.g. <a href="#">amazon</a>
3D printed parts <sup>2</sup>	Black/opaque, generic	PETG <sup>3</sup>	1	<a href="#">github:pi_tailtrack</a>
M3 screws and nuts	generic	to secure camera	1	
Computer screen, keyboard, mouse	generic	recycle/borrow/steal		

**Total: ≈338€**

<sup>1</sup>These particular specs are not required, but the power supply and resistor must be matched appropriately. See [amplifiedparts.com: LED Parallel/Series Calculator](#)

<sup>2</sup>Parts were printed on an [Ender 3 Pro](#) 3D printer: Price ≈200€.

<sup>3</sup>This is the material I had on hand, but likely anything will work (PLA, ABS, Resin, etc)

276 bladder (illustrated in [Figure 3A](#)). It is critical to not scratch the dish in the vicinity of the freed tail, which can interfere  
277 with tail-tracking.

## 278 Software

279 Software was written in Python, using the `picamera` library for camera control ([Raspberry Pi Foundation, 2023](#)). Trail  
280 tracking was performed using `OpenCV` (cv2 version 4.5.5) ([Bradski, 2000](#)), and `Numpy` (version 1.19.5) ([Harris et al.,  
281 2020](#)). All code is provided in the file `record_tail.py`. The main method for tail extraction was Adaptive Thresholding  
282 (cv2.adaptiveThreshold), using a threshold of -10 and a 33 pixel neighborhood. These parameters can be adjusted  
283 in real-time using the w/s and a/d keys. The starting coordinate for the tail tracking can be adjusted using the arrow  
284 keys.

285 Code for generating the figure panels in [Figure 3](#) can be found in: `plot_tail.ipynb`

## 286 Hardware

287 I used a Raspberry Pi 4 Model B Rev 1.4 computer, running Raspbian GNU/Linux 11 (bullseye). [Table 1](#) contains  
288 the details of the hardware components that I used, their approximate price, and an option for supplier (keeping in  
289 mind that these later two are subject to change and will rapidly become inaccurate).

## 290 Ca<sup>2+</sup> imaging and analysis

291 2-photon Ca<sup>2+</sup> imaging was performed and analyzed as described in ([Lamiré et al., 2022](#)). Briefly, a 5dpf *Tg2(elavl3:GCaMP6s)*  
292 (ZDB-ALT-180502-2, [Dunn et al. \(2016\)](#)) larvae was imaged using a 20x 1.0NA water dipping objective (Olympus) on  
293 a Bruker Ultima microscope at the CQLC imaging platform (Lyon, LYMIC). Frames were acquired using a resonant  
294 scanner over a rectangular region of 1024×512 pixels (0.6μm x/y resolution) and piezo objective to scan 12 planes  
295 separated at 10μm steps, with a repeat rate of 1.98 hz. ROIs were identified and fluorescence timeseries extracted  
296 using suite2p ([Pachitariu et al., 2016](#)). The zebrafish was stimulated with 60 "dark flash" stimuli at 60 second ISI  
297 ([Lamiré et al., 2022](#)), though responses to these stimuli were not incorporated into the analyses presented here,  
298 other than to synchronize the behavioural tracking with the microscope acquisition timing.

299 Code for generating the figure panels in [Figure 4](#) can be found in: `gcamp_corr_swimming.ipynb`. Images output  
300 from the analysis were adjusted for brightness/contrast and LUT using FIJI/ImageJ ([Schindelin et al., 2012](#)).

301 **Acknowledgements**

302 This work was supported by funding from the ATIP-Avenir program of the CNRS and Inserm, a Fondation Fyssen  
303 research grant, and the IDEX-Impulsion initiative of the University of Lyon.

304 **References**

- 305 **Ahrens MB**, Li JM, Orger MB, Robson DN, Schier AF, Engert F, Portugues R. Brain-wide neuronal dynamics during motor adaptation  
306 in zebrafish. *Nature*. 2012 May; 485(7399):471–477. <https://doi.org/10.1038/nature11057>, doi: 10.1038/nature11057.
- 307 **Ahrens MB**, Orger MB, Robson DN, Li JM, Keller PJ. Whole-brain functional imaging at cellular resolution using light-sheet mi-  
308 croscopy. *Nat Methods*. 2013 May; 10(5):413–420.
- 309 **Akerboom J**, Chen TW, Wardill TJ, Tian L, Marvin JS, Mutlu S, Calderón NC, Esposti F, Borghuis BG, Sun XR, Gordus A, Orger MB,  
310 Portugues R, Engert F, Macklin JJ, Filosa A, Aggarwal A, Kerr RA, Takagi R, Kracun S, et al. Optimization of a GCaMP Calcium  
311 Indicator for Neural Activity Imaging. *The Journal of Neuroscience*. 2012 Oct; 32(40):13819–13840. <https://doi.org/10.1523/jneurosci.2601-12.2012>, doi: 10.1523/jneurosci.2601-12.2012.
- 313 **Bradski G**. The OpenCV Library. *Dr Dobb's Journal of Software Tools*. 2000; .
- 314 **Broussard GJ**, Kislin M, Jung C, Wang SSH. A flexible platform for monitoring cerebellum-dependent sensory associative learning.  
315 *J Vis Exp*. 2022 Jan; (179).
- 316 **Budick SA**, O'Malley DM. Locomotor repertoire of the larval zebrafish: swimming, turning and prey capture. *J Exp Biol*. 2000 Sep;  
317 203(Pt 17):2565–2579.
- 318 **Chen TW**, Wardill TJ, Sun Y, Pulver SR, Renninger SL, Baohan A, Schreiter ER, Kerr RA, Orger MB, Jayaraman V, Looger LL, Svoboda  
319 K, Kim DS. Ultrasensitive fluorescent proteins for imaging neuronal activity. *Nature*. 2013 Jul; 499(7458):295–300. <https://doi.org/10.1038/nature12354>, doi: 10.1038/nature12354.
- 321 **Dunn TW**, Mu Y, Narayan S, Randlett O, Naumann EA, Yang CT, Schier AF, Freeman J, Engert F, Ahrens MB. Brain-wide mapping of  
322 neural activity controlling zebrafish exploratory locomotion. *Elife*. 2016 Mar; 5:e12741.
- 323 **Elder R**, A Guide to Recording 660FPS Video On A 6 dollar Raspberry Pi Camera; 2019. <https://blog.robertelder.org/recording-660-fps-on-raspberry-pi-camera/>, [Online; accessed 5-May-2023].
- 325 **Geissmann Q**, Garcia Rodriguez L, Beckwith EJ, French AS, Jamasb AR, Gilestro GF. Ethoscopes: An open platform for high-  
326 throughput ethomics. *PLoS Biol*. 2017 Oct; 15(10):e2003026.
- 327 **Harris CR**, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, Wieser E, Taylor J, Berg S, Smith NJ, Kern R, Picus M,  
328 Hoyer S, van Kerkwijk MH, Brett M, Haldane A, del Rio JF, Wiebe M, Peterson P, Gérard-Marchant P, et al. Array programming with  
329 NumPy. *Nature*. 2020 Sep; 585(7825):357–362. <https://doi.org/10.1038/s41586-020-2649-2>, doi: 10.1038/s41586-020-2649-2.
- 330 **Lamiré LA**, Haesemeyer M, Engert F, Granato M, Randlett O. Functional and pharmacological analyses of visual habituation  
331 learning in larval zebrafish; 2022.
- 332 **Maia Chagas A**, Prieto-Godino LL, Arrenberg AB, Baden T. The €100 lab: A 3D-printable open-source platform for fluorescence  
333 microscopy, optogenetics, and accurate temperature control during behaviour of zebrafish, *Drosophila*, and *Caenorhabditis elegans*. *PLoS Biol*. 2017 Jul; 15(7):e2002702.
- 335 **Mathis A**, Mamidanna P, Cury KM, Abe T, Murthy VN, Mathis MW, Bethge M. DeepLabCut: markerless pose estimation of user-  
336 defined body parts with deep learning. *Nat Neurosci*. 2018 Sep; 21(9):1281–1289.
- 337 **Miri A**, Daie K, Burdine RD, Aksay E, Tank DW. Regression-based identification of behavior-encoding neurons during large-scale  
338 optical imaging of neural activity at cellular resolution. *J Neurophysiol*. 2011 Feb; 105(2):964–980.
- 339 **Muller UK**. Swimming of larval zebrafish: ontogeny of body waves and implications for locomotory development. *J Exp Biol*. 2004  
340 Feb; 207(5):853–868.
- 341 **Pachitariu M**, Stringer C, Dipoppa M, Schröder S, Rossi LF, Dalgleish H, Carandini M, Harris KD. Suite2p: beyond 10,000 neurons  
342 with standard two-photon microscopy; 2016.
- 343 **Pereira TD**, Tabris N, Matsliah A, Turner DM, Li J, Ravindranath S, Papadoyannis ES, Normand E, Deutsch DS, Wang ZY, McKenzie-  
344 Smith GC, Mitelut CC, Castro MD, D'Uva J, Kislin M, Sanes DH, Kocher SD, Wang SSH, Falkner AL, Shaevitz JW, et al. Publisher  
345 Correction: SLEAP: A deep learning system for multi-animal pose tracking. *Nat Methods*. 2022 May; 19(5):628.

- 346 **Portugues R**, Feierstein CE, Engert F, Orger MB. Whole-Brain Activity Maps Reveal Stereotyped, Distributed Networks  
347 for Visuomotor Behavior. *Neuron*. 2014 Mar; 81(6):1328–1343. <https://doi.org/10.1016/j.neuron.2014.01.019>, doi:  
348 [10.1016/j.neuron.2014.01.019](https://doi.org/10.1016/j.neuron.2014.01.019).
- 349 **Randlett O**, Haesemeyer M, Forkin G, Shoenhard H, Schier AF, Engert F, Granato M. Distributed plasticity drives visual habituation  
350 learning in larval zebrafish. *Curr Biol*. 2019 Apr; 29(8):1337–1345.e4.
- 351 **Raspberry Pi Foundation**, picamera; 2023. <https://picamera.readthedocs.io/>, [Online; accessed 02-May-2023].
- 352 **Saunders JL**, Ott LA, Wehr M. AUTOPILOT: Automating experiments with lots of Raspberry Pis; 2019.
- 353 **Schindelin J**, Arganda-Carreras I, Frise E, Kaynig V, Longair M, Pietzsch T, Preibisch S, Rueden C, Saalfeld S, Schmid B, Tinevez JY,  
354 White DJ, Hartenstein V, Eliceiri K, Tomancak P, Cardona A. Fiji: an open-source platform for biological-image analysis. *Nat  
355 Methods*. 2012 Jun; 9(7):676–682.
- 356 **Severi KE**, Portugues R, Marques JC, O’Malley DM, Orger MB, Engert F. Neural control and modulation of swimming speed in the  
357 larval zebrafish. *Neuron*. 2014 Aug; 83(3):692–707.
- 358 **Štih V**, Petrucco L, Kist AM, Portugues R. Stytra: An open-source, integrated system for stimulation, tracking and closed-loop  
359 behavioral experiments. *PLoS Comput Biol*. 2019 Apr; 15(4):e1006699.
- 360 **Tadres D**, Louis M. PiVR: An affordable and versatile closed-loop platform to study unrestrained sensorimotor behavior. *PLoS  
361 Biol*. 2020 Jul; 18(7):e3000712.
- 362 **Wikipedia contributors**, Extension tube — Wikipedia, The Free Encyclopedia; 2022. [https://en.wikipedia.org/w/index.php?title=Extension\\_tube&oldid=1118116052](https://en.wikipedia.org/w/index.php?title=Extension_tube&oldid=1118116052), [Online; accessed 28-April-2023].
- 364 **Wolf S**, Dubreuil AM, Bertoni T, Böhm UL, Bormuth V, Candelier R, Karpenko S, Hildebrand DGC, Bianco IH, Monasson R, Debrégeas  
365 G. Sensorimotor computation underlying phototaxis in zebrafish. *Nat Commun*. 2017 Sep; 8(1).