

A system for the high-throughput analysis of acute thermal avoidance and adaptation in *C. elegans*

Andrei-Stefan Lia, Dominique A. Glauser*

Supplementary text

Text S1. Background illumination ring: plastic support assembly.

Due to our 3D printer's model size constraints, we had to design the illumination ring as four separate interlocking parts (Fig. S5A), two containing a 20x20 cm mounting hole each, two without. To produce the support, follow these steps:

- 1) 3D print the INFERNO_LED_ring_segmented.stl model.
- 2) To assemble the ring, connect one element with a mounting hole to one without (Fig. S5B).
- 3) Repeat this step for the remaining two elements.
- 4) Assemble the two resulting parts so that the two mounting holes are diametrically opposed to each other (Fig. S5C).

Troubleshooting: Some variability in ease of insertion is expected, as the printing quality and fidelity varies across 3D printer models and materials used. If the parts do not smoothly interlock, you might consider slightly filing down the insertion pegs with some sandpaper. If they don't fit snugly, glue can be used.

Text S2. Background illumination ring: LED circuit assembly.

Caution: always have a professional electrician verifying your circuit before applying any current, and comply with any local regulations. These instructions are only broad guidelines. Performing these actions may result in material damage, injury or death. Proceed at your own risk.

- 1) Insert 40 blue-light LEDs into the indicated LED slots of the 3D printed ring (Fig. 3A), creating four 10-LED stretches. In terms of orientation, ensure that each individual anode LED leg is oriented in the same direction (i.e. left/clockwise), with the cathode LED leg oriented in the other direction (i.e. right/counter-clockwise). Initial insertion can be done by hand, wedging a LED into the square LED housing.
- 2) With a screwdriver, push the LED through the circular mounting slot by pressing the flat side (from which the metal legs emerge), until the LED will not go down further.
- 3) For each 10-LED series, bend the first anode leg in the series towards the top side of the ring, and the final cathode towards the bottom side of the ring.
- 4) Bend the remaining anode and cathode legs so that the first cathode overlaps the second anode, the second cathode overlaps the third anode, etc., thus forming a series circuit (Fig. S6A).
- 5) Individually solder the 9 resulting junctions.
- 6) Optional: Trim down excess metal tips from the anode/cathode junctions
- 7) To create a parallel circuit (Fig. S6B), wire and solder the first anodes of all 4 series together, then wire and solder the last cathodes of all 4 series together.
- 8) To power the circuit, affix the positive/negative wires of a power supply to the positive/negative sides of the parallel wire of the circuit. *TIP: Initial testing can be performed with crocodile clips, but it is extremely important to verify the polarity.* Use an adjustable DC power supply, with overcurrent protection. Start using half of the calculated nominal voltage for the LED circuit, and never exceed 90% of the maximal calculated voltage of the LED circuit (based on the maximal functioning voltage of the LED provided by the manufacturer). E.g. when wiring series of 10 LEDs rated at 3.2 V each, the maximal calculated voltage will be 32 V (and the maximal value you should not exceed is 28.8 V).

Text S3. Assembly of the Peltier element temperature control circuit.

Assembly was carried out according to the manufacturer's user manual, which can be found online at <https://www.quick-cool-shop.de/en/download/User%20Manual%20English%20Controller%20QC-PC-CO-CH1%2020160318.pdf>. In our setup, control is achieved with the QC-PC-D-CH1. Thus, it is not necessary to wire the provided potentiometer.

Text S4. INFERNO reversal flagging pipeline user guide.

Prerequisite software: **Cygwin** (<https://cygwin.com/install.html>) with the **python 2.7** package selected during installation (labelled as **python27** in the **Cygwin** package manager). All scripts described below can be downloaded from the afferent GitHub repository (see **GitHub repository** section in the main article): `INFERNO_pipeline_8fps.sh`, `INFERNO_reversal_flagger_8fps.py`; Both scripts include supplementary comments that explain the function of code blocks. Comments are canonically preceded by a '#', which prevents code compilers and/or interpreters from running the comments as code.

The MWT output folder for each movie from an experimental day should be moved into a centralized folder, which is then used as input for the `INFERNO_pipeline_8fps.sh` (*e.g.* `./output_replicate_1` and `./output_replicate_2` moved to `./centralized_folder`).

The **Cygwin** terminal is launched and the current directory is set to the absolute path of the centralized folder with the following generalized terminal command:

```
cd /path/to/centralized_folder
```

Alternatively, users can type 'cd[spc]' into the terminal and drag and drop the centralized folder into the terminal, which will automatically fill in the absolute path.

N.B. For clarity, '[spc]' is used to explicitly mark a whitespace character.

New users should configure the `INFERNO_pipeline_8fps.sh` in a text editor, so that line 13 references the actual path where the `INFERNO_reversal_flagger_8fps.py` was saved on their system instead of the generalized path provided:

```
python2.7/path/to/INFERNO_reversal_flagger_8fps.py
```

The `INFERNO_pipeline_8fps.sh` is executed by inputting its absolute path into the terminal, generalized below as:

```
/path/to/INFERNO_pipeline_8fps.sh
```

Alternatively, users can drag and drop the `INFERNO_pipeline_8fps.sh` file into the terminal, which will automatically fill in the absolute path.

Once the `INFERNO_pipeline_8fps.sh` has finished executing, users can navigate to the newly generated output subfolder within the centralized folder, which will contain a `.txt` file for each MWT input folder. The `.txt` file contains reversal data identified with the `INFERNO_reversal_flagger_8fps.py`, following the layout provided in Fig. S3.