Quickstart

Table of contents

Welcome!

This page will walk you through everything you need to **run**, **create**, and **analyse** a freely-walking experiment. If you are new to this setup, read through this page first, then follow the links to the detailed pages as needed. For hardware and software requirements, see the Rig Specs page.

Training guide

A comprehensive training guide is available as a PDF at `docs/training_guide/training_guide.pdf` in the freely-walking-optomotor repository. The training guide covers the full data pipeline in detail — from running experiments through to the DATA struct — and includes diagrams, code walkthroughs, and a glossary of key terms. It is a useful companion to this documentation site, especially for understanding the processing functions and data structures in depth.

1 Prerequisites

Before running an experiment, ensure the following are in place:

- ☐ LED arena powered on and controller connected
- ☐ **BIAS** (SimpleBiasCameraInterface) open — IR illumination on, camera acquiring at 30 fps
- ☐ **PControl** connected to the LED panels in MATLAB
- ☐ `freely-walking-optomotor` repository cloned and configured — see Configuration & Paths
- ☐ Required MATLAB toolboxes installed: Statistics and Machine Learning, Image Processing, Circular Statistics
- ☐ G4_Display_Tools installed (provides the `panel_com` functions)
- ☐ FlyTracker installed (for post-experiment tracking)

See the Rig Specs page for full hardware and software details.

---

2 How to run a protocol

1. Power on the LED arena and the controller.
2. Open **BIAS** (SimpleBiasCameraInterface) — ensure the IR illumination is on and the camera is acquiring at 30 fps.
3. Open **PControl** in MATLAB and verify it connects to the LED panels.
4. Use the live camera view to confirm the flies are visible in the arena.
5. Load the flies into the arena and allow them to settle.
6. Run the MATLAB protocol script (e.g. `protocol_27.m`). The script handles the full stimulus sequence including acclimation periods, stimulus presentation, and data logging.
7. After the protocol finishes, transfer the data folder to the processing computer and run FlyTracker to generate tracking output — or let the automated pipeline handle this for you.

The protocol used for the screen was protocol 27. It presents 12 stimulus conditions (gratings, bars, curtains, reverse-phi, flicker, and a fixation bar) with 15 s stimulus trials separated by 20 s dark intervals, repeated twice. A 5-minute dark acclimation precedes the stimulus block.

---

3 How to make a protocol

The easiest way to create a new protocol is to start from the **protocol template** (`src/protocols/protocol_template.m`) in the freely-walking-optomotor repository. The template is based on `protocol_27` and includes:

- Annotated sections for each part of the experiment (workspace setup, timing parameters, condition definitions, stimulus presentation, data saving)
- A comprehensive speed reference table mapping speed values to temporal frequencies for all common patterns
- Commented examples for different stimulus routing options (standard gratings, curtains, fixation bars, mixed routing)
- Automatic condition numbering (the 7th column of `all_conditions` is generated from the row index)

To create a new protocol:

1. Copy `protocol_template.m` and rename it (e.g. `protocol_XX.m`)
2. Edit the header comments to describe the scientific question and stimuli
3. Define your `all_conditions` matrix with the desired patterns, speeds, and durations
4. Choose the appropriate presentation function(s) in the stimulus routing section
5. Test with a short acclimation time (`t_acclim_start = 30`) before running the full experiment

Patterns for the G3 arena can be generated using the `G4_pattern_generator_gui`, but for this screen all patterns and position functions were made from scratch using custom scripts:

- Pattern scripts: script_to_make_patterns/
- Protocol scripts: protocols/ (includes a subfolder with helper functions)
- See the Protocols and Patterns index pages for documentation on each stimulus

---

4 How to analyse the data

The analysis pipeline is described in detail on the Analysis page. In brief, data processing proceeds in three levels:

1. **Level 1** — per-cohort processing via `process_freely_walking_data()`, which generates overview figures and results files for each individual vial of flies.
2. **Level 2** — cross-cohort analysis via `process_screen_data()`, which combines data across all strains and cohorts and creates comparison plots against the empty-split control flies.
3. **Level 3** — statistical testing via `make_summary_heat_maps_p27()`, which generates p-value heatmaps comparing each strain to the empty-split control.

If you are analysing data from a different protocol, see Analysing a new protocol.

For interactive data exploration, see the Dashboard page — it provides a browser-based interface for comparing strains and conditions without writing code.

5 What happens automatically

If you are running experiments at Janelia, the automated pipeline handles the data flow after you finish recording. Data is automatically copied from the acquisition PC, tracked with FlyTracker, and processed with the MATLAB pipeline. You can monitor the progress of all experiments via the pipeline status page.