Experiment Logging System

Table of contents

1 Automatic logging of experiments using a google form.

As well as recording which experiments were run in a physical lab book, the running of any "protocol_X" script will automatically log the experiment using an in-built google form. This process works by feeding the parameter "params" into a pre-made google form which then exports the results to a google sheet.

When any protocol is started by clicking on the "Run" button within the MATLAB editor, the function `get_input_parameters()` is run which opens a pop up window where the user can fill in details about the current experiment using a series of dropdown boxes (See figure). These values are assigned to fields within the struct `params` which is returned by the function.
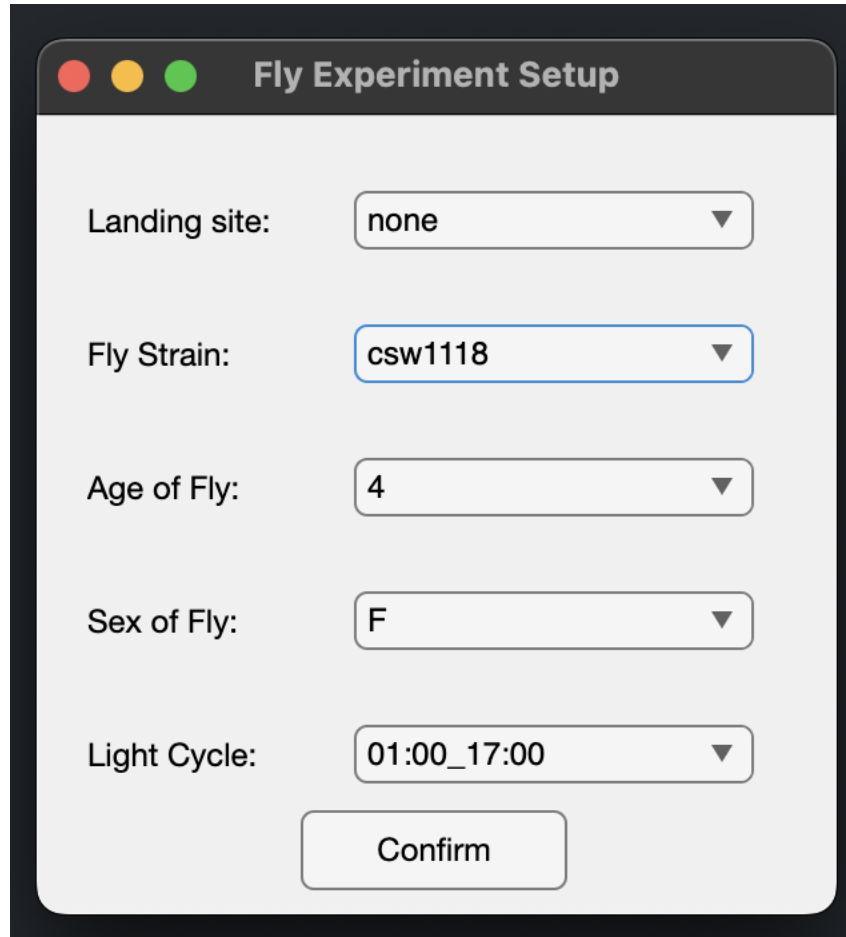
Figure 1: `get_input_parameters` GUI that pops up when you run any protocol.

Within the function `initialize_video_and_folder`, `params` is added to with the current date, the experiment time and the protocol which is being run. The in-built MATLAB function `input` is used within this function enable the user to type a string describing initial notes before the experiment has started. This string is also added to `params`. Within the protocol script itself, there is another call to `input` which can be used to enter notes at the end of experiment. These notes could relate to the general behaviour of the group of flies or some experimental abnormality. After this final `input` call, the function `export_to_google_sheets` is run which takes in the URL of the pre-made google form that will be filled by the values of `params`.

2 How to make the google form logging system

1 - Generate a pop up to take in data from the user. See `get_input_parameters` or `get_video_observations` in this repository for inspiration.

2 - Use the MATLAB function `input` if you want to include a long form string in your documentation.

3 - Make a google form with "questions" relating to the fields of data that you would like to record. The "questions" do not need to match exactly the naming of the struct that is returned from GUI.

4 - Set the "type" of each question to "short answer" question. Don't use "date" or "time" even if you have those fields.
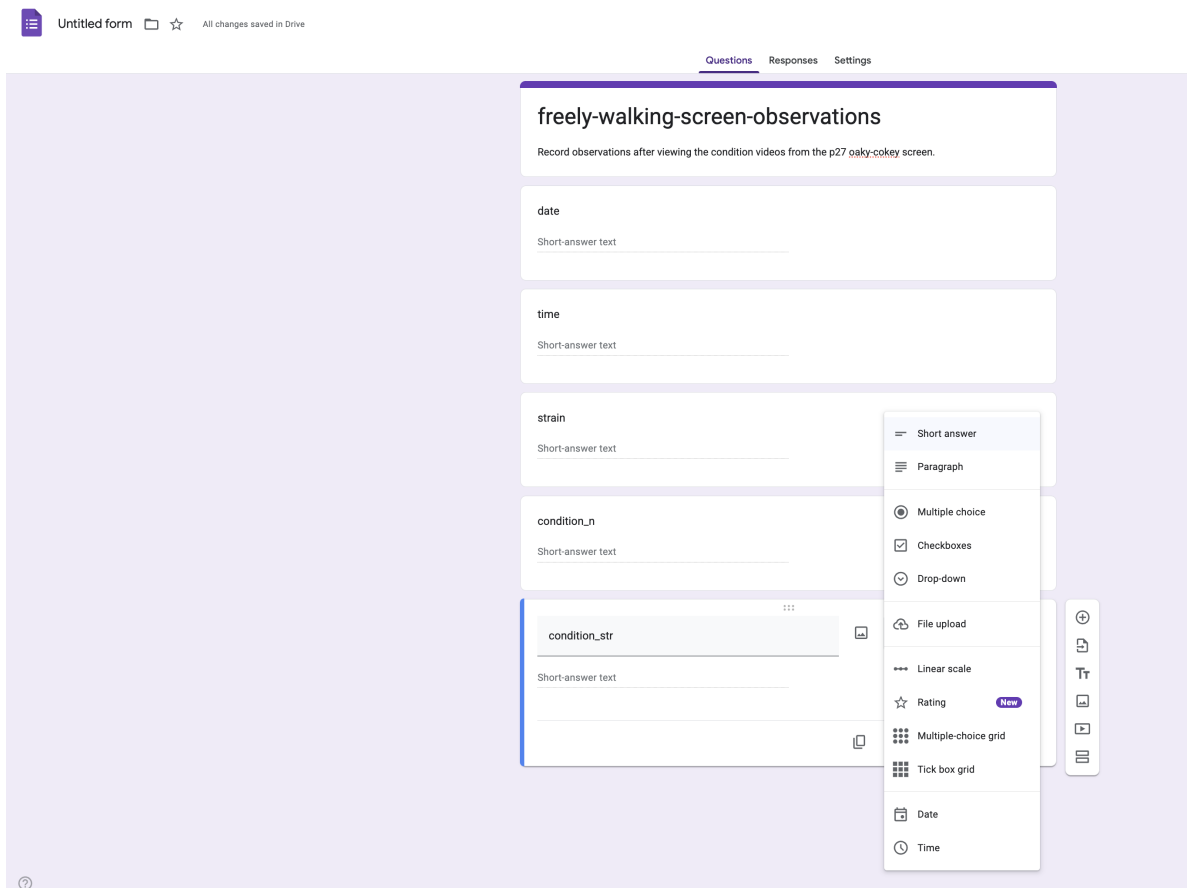

Figure 2: Select 'short answer' when creating the google form.

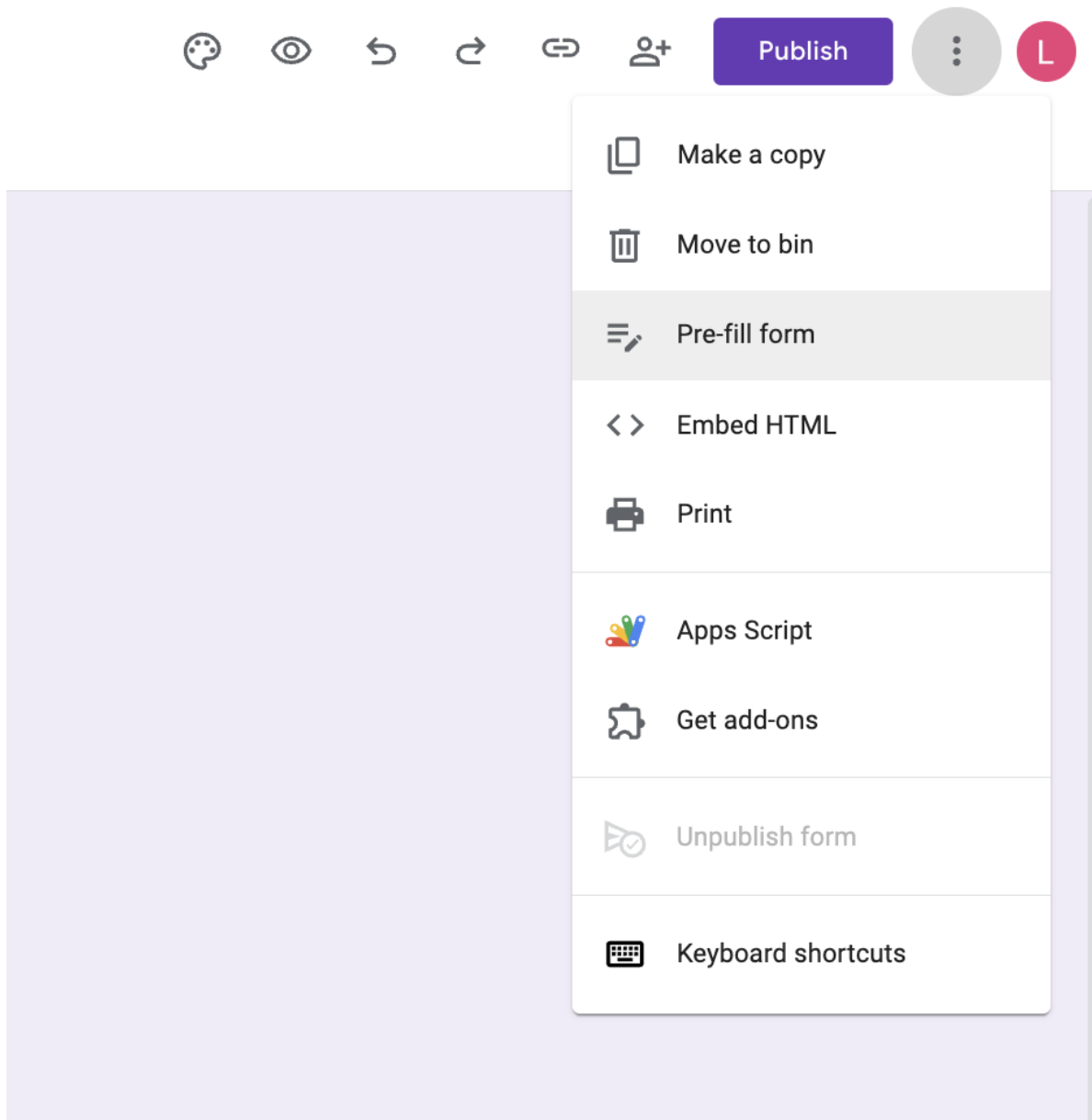**5** - Click on the three-dot menu ( ) in the top right and select "Pre-fill form".

Figure 3: Select 'Pre-fill form'

6 - Fill in sample responses and click "Get link" to generate a URL.

centring

None

turning

None

rebound

None

distribution

Sparse

locomotion

None

diversity

10

notes

Test case

Get link

Figure 4: Fill in the form with example entries then press 'Get link'.

7 - Copy the generated link (it will look something like this):

```
"https://docs.google.com/forms/d/e/1FAIpQLScgK2F-PiaHaW9AiqUjHNZ9D1WYpF1-
W19sEwlob2orr2wyfg/viewform?usp=pp_url&entry.104586054=10_23_25&entry.1144953751=10_04_27&ent
```

8 - From this link extract the "googleFormURL" and the entry IDs.

To find the "googleFormURL", get the first part of this URL up to "/viewform?....". Copy this initial part of the URL and after the last "/" add "formResponse". So, the final URL will be:

```
googleFormURL = "https://docs.google.com/forms/d/e/1FAIpQLScgK2F-PiaHaW9AiqUjHNZ9D1WYpF1-
W19sEwlob2orr2wyfg/formResponse"
```

9 - Link the MATLAB data with the questions in the google form.

Each number after `entry.` is the entry ID that corresponds to each field in your form, these need to be added within the `webwrite` call. This is what the code in MATLAB will look like:

```matlab
% Google Form Response URL (Make sure it ends with /formResponse)
googleFormURL = 'https://docs.google.com/forms/d/e/1FAIpQLScgK2F-PiaHaW9AiqUjHNZ9D1WYpF1-

% Use weboptions to set character encoding
options = weboptions('MediaType', 'application/x-www-form-urlencoded', 'Timeout', 30);

% Submit data using name-value pairs instead of struct
webwrite(googleFormURL, ...
    'entry.104586054', metadata.date, ...
    'entry.1144953751', metadata.time, ...
    'entry.1017789798', metadata.strain, ...
    'entry.462337747', metadata.condition_n, ...
    'entry.635492352', metadata.condition_str, ...
    'entry.1832623891', metadata.rep, ...
    'entry.932164432', observations.centring, ...
    'entry.324082057', observations.turning, ...
    'entry.1509506971', observations.rebound, ...
    'entry.350553119', observations.distribution, ...
    'entry.1178542927', observations.locomotion, ...
    'entry.1494282419', observations.diversity, ...
    'entry.183573490', notes, ...
    options); % Include options for correct encoding
```
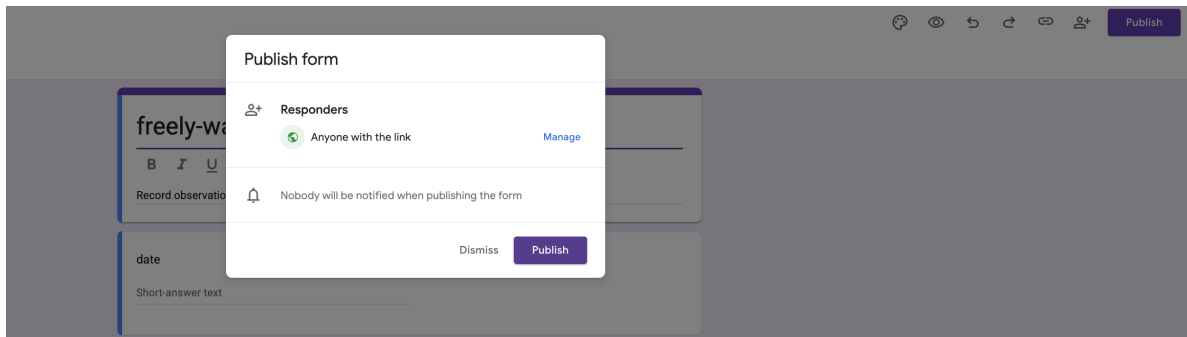
10 - Publish the form.

Figure 5: Publish the form to make it accessible.

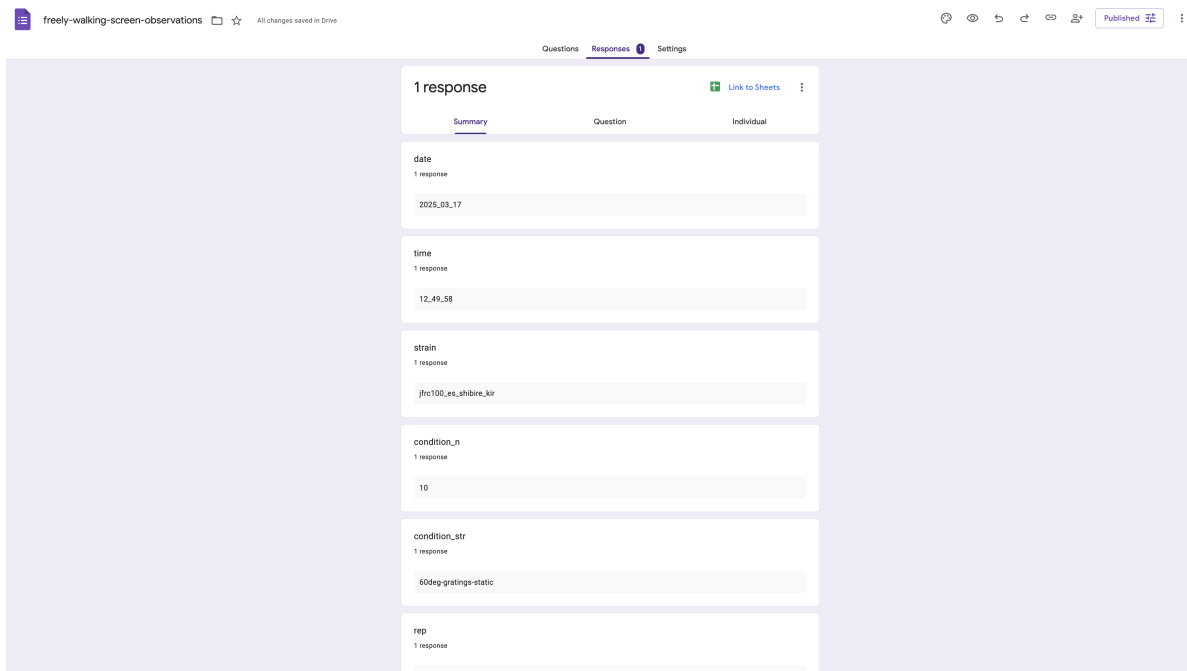**11** - Try running the script and see if a response is recorded within the form!

Figure 6: View the responses in the 'response' tab within the form.

Once you get responses you can open a google sheet and look at the combined responses there.

Figure 7: View the responses in the linked google sheet.