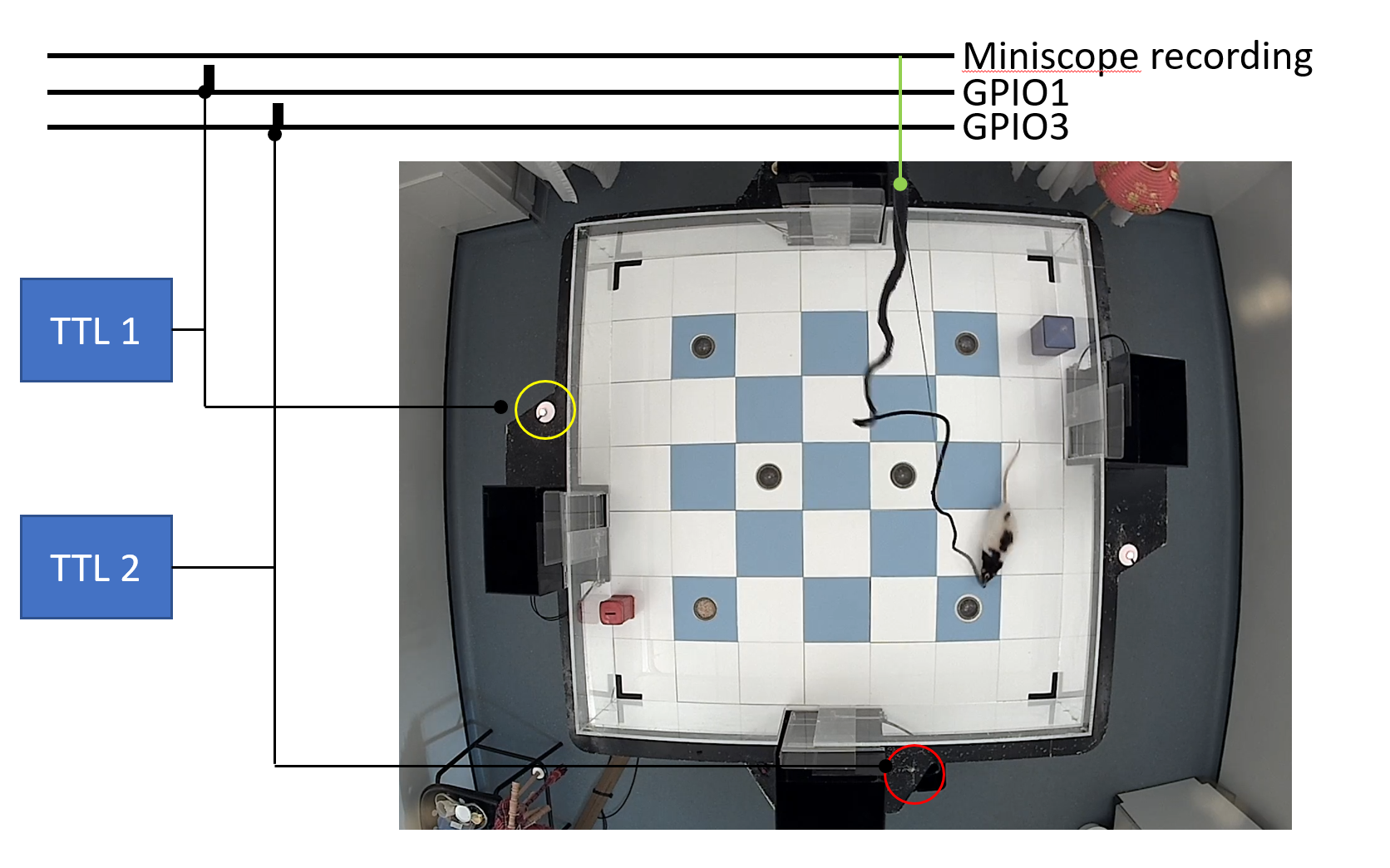
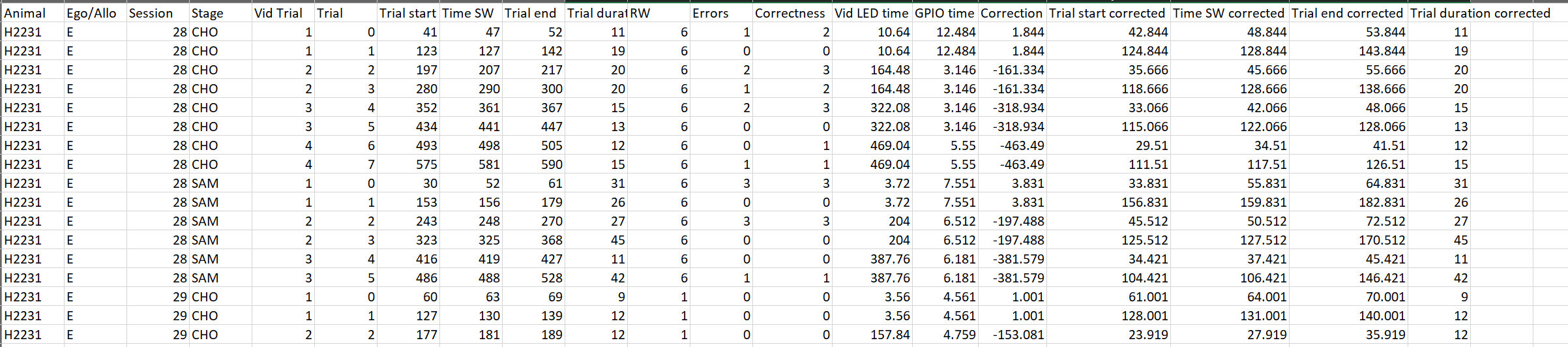
**Data Synchronisation**

The synchronisation is done by means of a TTL impulse timestamping the GPIO channel and a light bulb in the video camera field. Two separate TTLs channels were employed for safety reasons. The video recording and the miniscope recording start independently and are synchronised with the synchronous TTL timestamp on the GPIO channel/LED switch in the video frame.



The resulting data is saved as follows:



Trial: trial number

Vid Trial: trial number in video (each video recording contains 2 trials)

Trial start: Start of the trial (animal moving out of startbox) as time in video

Time SW: Time where animal reached the sandwell

Trial End: Return to Starbox (video timestamp)

Vid LED time: Time where LED light connected to TTL1 is switched on.

GPIO time: Time when the TTL impulse is timestamped on the GPIO channel

Correction: GPIO/LED correction: the animal positions detected in video will be time shifted and aligned with calcium recording

Trial start corrected, Time SW corrected, Trial End corrected: same as above but aligned to calcium recording timeline

**Example Data**

This folder contain example data to run the Jupyter notebook files. Individual files are saved using the same Folder network structure so that the Home directory should be changed to “Example Data”. Animal and Stage, and extra folders, should be modified following code annotation.

The following code is provided as Jupyter notebooks in Python.

The code can be run via Anaconda Navigator (https://www.anaconda.com/products/navigator)

**Place Cells.ipynb**

This Jupyter notebook contains the code to calculate place cells map from aligned calcium and extracted video data. Mutual information and maps are calculated here.

**Place Cells Remapping.ipynb**

This Jupyter notebook contains code to calculate Place cells properties such as stability (to evaluate remapping) and direction dependency.

**Trial\_identification\_code.ipynb**

This Jupyter notebook contains code for the automatic identification of trial starts from animal coordinates.

**Startbox activity Export.ipynb**

This Jupyter notebook contains code to export the information content of cells active in the startbox in the 10s prior to trial start.

**Common SB cells.ipynb**

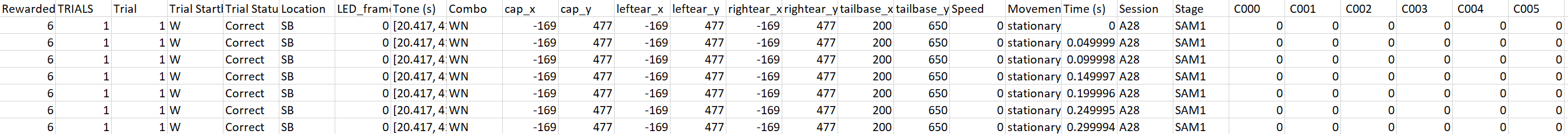
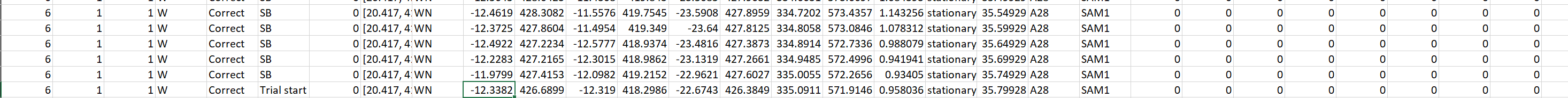
This Jupyter notebook contains code to calculate the fraction of common cells across startboxes before trial start.

**arena.py** and **place\_cells.py** contain functions used in the Jupyter notebooks.

Our code design saves intermediate steps as data tables in csv format that are then read by separate Jupyter notebooks for later analysis. Data are grouped by animal for convenience.   
in ALL\_META.csv . Below are an example that describse the format.

**ALL\_META.csv**

This table is generated by concatenation of all trials after automatic trial identification and quality check. Each animal has an individual table

….

Trial Start: Starting startbox

Trial Status: Automatic classification of correct of incorrect trials based on trajectory

Location: Position of animal in that time frame; SB = startbox. Trial start: start of the trial. Outbound: animal traveling to sandwell

LED\_frame: synchronising LED status

Combo: use of startboxes in trial (start/end)

Cap\_x, …. Tailbase\_y: x and y coordinates of animal body parts (head, ears, tail) after correction and translation.

Speed: movement speed

Movement: classification as moving or stationary

Time (s) : time

Session: Session type and number

Stage: SAM or CHO trial

C000…. Event vector of cells; 0: no events, non-zero value: event amplitude (note: events are binarized in analysis