DESCRIPTION

This code is used for the following within our manuscript:

(1) alignment of fluorescent signals (“averaged\_signals.npy” files) and behavioral data (“.mat”) files from our imaging datasets,

(2) clustering of unique active lever response patterns among neuronal ensembles

(3) decoding of active lever pressing behavior based on the observe population and ensemble dynamics.

Dependencies for this script take about 1 hour to install, and the code itself takes about 1-1.5 hours to run for the sample dataset (the decoding section takes about 45 minutes to run, whereas the other sections take much less time).

DEPENDENCES

Python v2.7 or greater

Jupyter Notebook

The following modules must also be installed within the Python environment

Numpy

Matplotlib

OS

Seaborn

Pandas

Sci-kit Learn

Scipy

Patsy

Sys

Re

Random

We executed the code using a MacBook Pro using Python 2.7. However, the code can be formatted for use in Windows or Linux, and for Python versions >2.7.

DIRECTIONS

Following installation of Python and associated dependencies, start by opening “Population - PVT Self-Admin Sample Code FINAL.ipynb” in Jupyter which can be found in the “Sample Population Code” folder.

The first 3 cells of code are for importing the downloaded modules (1), defining the frame rate for image acquisition (2), and for defining functions that will be used in later cells (3). Each cell can be run for the sample data without adjustment.

In cell 4, we will define your user directory information. “basedir” should indicate the location of the downloaded folder “Sample Population Code”. “doibasedir” is the location of your day of interest, which for our downloadable sample data will be the same as “basedir” plus the folder “Sample Day”. We can analyze multiple animals within the sample day folder in the next set of lines, by changing the variable “animals\_of\_interest”. Note that each animal folder has one or two fields of view (FOVs), which will all be automatically analyzed when indicating the animal’s name. In the unmodified version of the code, we analyze all six sample mice simultaneously. Once we have selected this information, the cell can be executed and should deliver a heatmap and averaged line of the sample data, which should look like Figure 1 J-K.

Our next section is Clustering. You’ll notice that within the Clustering cells, we perform and describe a Principal Components Analysis based on the active lever press responses and use that analysis to inform a Spectral Clustering Algorithm. Spectral clustering will allow us to separate unique cell response patterns, and subsequently plot them. We save the cluster identities as a loadable numpy file to each FOV folder. Results might vary slightly run-to-run, but we observed 2 Principal Components (Extended Data Fig. 2A) and 3 Clusters (Extended Data Fig. 2C; Figure 1L)

Our final section is Decoding, where we use the activity of all neurons (“population analysis”) or the activity of each ensemble (“ensemble analysis”) to predict active lever pressing behavioral events. This section of the code will load previously analyzed clustering data from above, as clustering results are saved to each mouse’s folder. The final two cells of code for the decoding analysis allow you to visualize the population level decoding data as a CDF plot (Extended Data Fig. 3A, and ensemble level decoding data as a CDF plot (Figure 1M). Note that the cluster identities (e.g., the inhibited ensemble) may be rearranged in order, but they correspond to the order described within the clustering analysis above (rather than the data in the paper).