Here we are going to learn to load data from the International Brain Lab, and make a simple raster plot. This is mainly an exercise in programming.

To see what the IBL data looks like, go to https://viz.internationalbrainlab.org.

First job is to install the library needed to access IBL data:

```
pip install ONE-api
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: ONE-api in /usr/local/lib/python3.8/dist-packages (1.16.3)
Requirement already satisfied: packaging in /usr/local/lib/python3.8/dist-packages (from ONE-api) (21.3)
Requirement already satisfied: tqdm>=4.32.1 in /usr/local/lib/python3.8/dist-packages (from ONE-api) (4.64.1)
Requirement already satisfied: flake8>=3.7.8 in /usr/local/lib/python3.8/dist-packages (from ONE-api) (6.0.0)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.8/dist-packages (from ONE-api) (6.0)
Requirement already satisfied: iblutil>=1.1.0 in /usr/local/lib/python3.8/dist-packages (from ONE-api) (1.4.0)
Requirement already satisfied: boto3 in /usr/local/lib/python3.8/dist-packages (from ONE-api) (1.26.41)
Requirement already satisfied: pandas>=1.2.4 in /usr/local/lib/python3.8/dist-packages (from ONE-api) (1.3.5)
Requirement already satisfied: requests>=2.22.0 in /usr/local/lib/python3.8/dist-packages (from ONE-api) (2.23.0)
Requirement already satisfied: numpy>=1.18 in /usr/local/lib/python3.8/dist-packages (from ONE-api) (1.21.6)
Requirement already satisfied: pycodestyle<2.11.0,>=2.10.0 in /usr/local/lib/python3.8/dist-packages (from flake8>=3.7.8->ONE-api) (2.10
Requirement already satisfied: mccabe<0.8.0,>=0.7.0 in /usr/local/lib/python3.8/dist-packages (from flake8>=3.7.8->ONE-api) (0.7.0)
Requirement already satisfied: pyflakes<3.1.0,>=3.0.0 in /usr/local/lib/python3.8/dist-packages (from flake8>=3.7.8->ONE-api) (3.0.1)
Requirement already satisfied: colorlog>=6.0.0 in /usr/local/lib/python3.8/dist-packages (from iblutil>=1.1.0->ONE-api) (6.7.0)
Requirement already satisfied: pyarrow in /usr/local/lib/python3.8/dist-packages (from iblutil>=1.1.0->ONE-api) (9.0.0)
Requirement already satisfied: numba in /usr/local/lib/python3.8/dist-packages (from iblutil>=1.1.0->ONE-api) (0.56.4)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.8/dist-packages (from pandas>=1.2.4->ONE-api) (2022.6)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.8/dist-packages (from pandas>=1.2.4->ONE-api) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.8/dist-packages (from python-dateutil>=2.7.3->pandas>=1.2.4->ONE-api)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.8/dist-packages (from requests>=2.22.0->ONE-api) (2022.12.7)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.8/dist-packages (from requests>=2.22.0->ONE-api) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.8/dist-packages (from requests>=2.22.0->ONE-api) (2.10)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.8/dist-packages (from requests>=2.22.0-
Requirement already satisfied: s3transfer<0.7.0,>=0.6.0 in /usr/local/lib/python3.8/dist-packages (from boto3->ONE-api) (0.6.0)
Requirement already satisfied: botocore<1.30.0,>=1.29.41 in /usr/local/lib/python3.8/dist-packages (from boto3->ONE-api) (1.29.41)
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in /usr/local/lib/python3.8/dist-packages (from boto3->ONE-api) (1.0.1)
Requirement already satisfied: llvmlite<0.40,>=0.39.0dev0 in /usr/local/lib/python3.8/dist-packages (from numba->iblutil>=1.1.0->ONE-api
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.8/dist-packages (from numba->iblutil>=1.1.0->ONE-api) (5.1.0
Requirement already satisfied: setuptools in /usr/local/lib/python3.8/dist-packages (from numba->iblutil>=1.1.0->ONE-api) (57.4.0)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.8/dist-packages (from importlib-metadata->numba->iblutil>=1.1.0->ONE-
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.8/dist-packages (from packaging->ONE-api) (3.0.9)
```

Now import libraries we need, and open a connection to the IBL public data server

```
import numpy as np
import matplotlib.pyplot as plt
from one.api import ONE
one = ONE(base_url='https://openalyx.internationalbrainlab.org', password='international', silent=True)
```

Now we are going to load information about the spikes and trials for one experiment. Each experiment has a unique "experiment ID", which is a long string identifying the experiment. This particular one is an experiment made at New York University, where they recorded from the motor cortex and striatum. You can get more information by putting the eID string in the search box at https://viz.internationalbrainlab.org.

We will load two data objects: dictionaries containing information about the spikes, and about the trials.

```
eID = 'ebe2efe3-e8a1-451a-8947-76ef42427cc9'

spikes = one.load_object(eID, 'spikes', 'alf/probe00/pykilosort')
trials = one.load_object(eID, 'trials')
```

To see what is in the dictionaries we can use the .keys() function. You can also access data using . notation - so `spikes.times` gives the same as `spikes['times']`.

Our main interest is in `spikes.times` - the time of each recorded action potential ; `spikes.clusters` - the neuron each action potential was assigned to ; `trials.stimOn_times` - the time of stimulus onset on each trial ; and `trials.choice` - which gives the subject's choice on that trial (-1 or +1). All times are in seconds. We are going to make a raster of the spikes for neuron #41, so we start by getting the times of these spikes:

```
print(spikes.keys())
print(trials.keys())

my_spike_times = spikes.times[spikes.clusters==41]


print(my_spike_times)
```

```
dict_keys(['amps', 'clusters', 'depths', 'samples', 'templates', 'times'])
dict_keys(['goCueTrigger_times', 'stimOff_times', 'feedbackType', 'contrastLeft', 'contrastRight', 'rewardVolume', 'goCue_times', 'choic
[9.51507143e-01 1.96194201e+00 2.54947623e+00 ... 4.66518843e+03
 4.66527446e+03 4.66572263e+03]
```

Now we are ready to make the raster! And this is where you won't see the code since you need to do it yourself!

Write some code that:

1. finds all spikes that occur between 100 ms before and 1s after a stimulus onset
2. plots these in a scatter plot, with time relative to stimulus onset on the x-axis and trial number on the y-axis
3. colors each spike by the choice made by the subject on that trial.

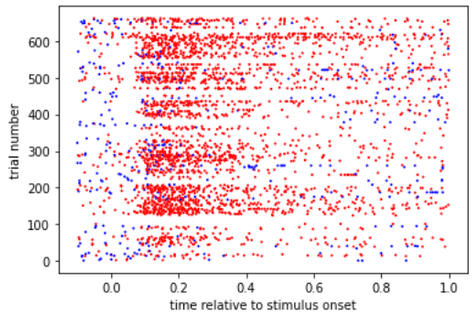
There are lots of ways to do this. Using a loop over trials is slower but probably easiest. You might want to build up python lists of numpy arrays containing spike times relative to stimulus onset; trials numbers; and choice IDs. Then concatenate them with np.concatenate, and plot

everything with plt.scatter.

A faster way to do it would be to use `numpy.searchsorted`. That's not necessary here but try it if you are feeling ambitious!

Output should be something like

Show code



What do you conclude about what this neuron does? Try making some more rasters to explore other things about this neuron, for example by time aligning to response_times or feedback_times, coloring by feedback_type, or sorting rows by reaction time. Can you find any other behavioral or sensory correlates of this neuron?