The International Brain Laboratory

Brain-Wide Map Data – Hands-On Training

295501 neurons
547 insertions
194 brain regions
115 mice performing the task

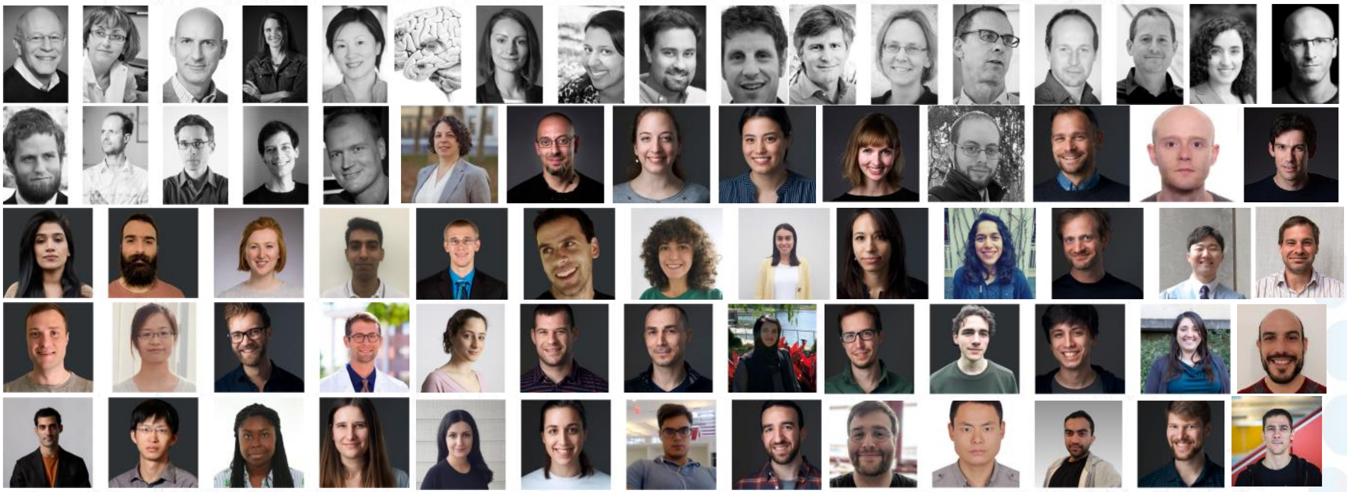
Karolina Socha
PostDoc in Carandini and Harris Labs

INTERNATIONAL BRAIN LABORATORY

NeuroDataShare 2023
22 February 2023, UCL-SWC, London

The IBL: 22 labs working to understand decision-making

INTERNATIONAL BRAIN LABORATORY



IBL Public Data

1. **Brain-Wide Map (BWM) Data** (recordings from 194 brain regions during perceptual task)
2. Reproducible Ephys Data (recordings in the same location)
3. Behavior Data (only behavior during learning the task)
4. Future: individual projects (mesoscale imaging, photometry, optogenetics)

Future Tutorials: COSYNE 2023

INTERNATIONAL
BRAIN
LABORATORY

DATE: 9 March 2023

TIME: 10:00 - 11:00

LOCATION: Av. Duluth, Convention Floor, Fairmont The Queen Elizabeth

IBL Public Data

INTERNATIONAL BRAIN LABORATORY

Access to data:
https://viz.internationalbrainlab.org/

Methods:
https://figshare.com/articles/preprint/Data_release_-_Brainwide_map_-_Q4_2022/21400815

Data Type:
https://docs.google.com/document/d/1OqIqqakPakHXRAwceYLwFY9gOrm8_P62XIfCTnHwstg/edit#

Example scripts:
https://github.com/int-brain-lab/UCL_NeuroDataShare2023

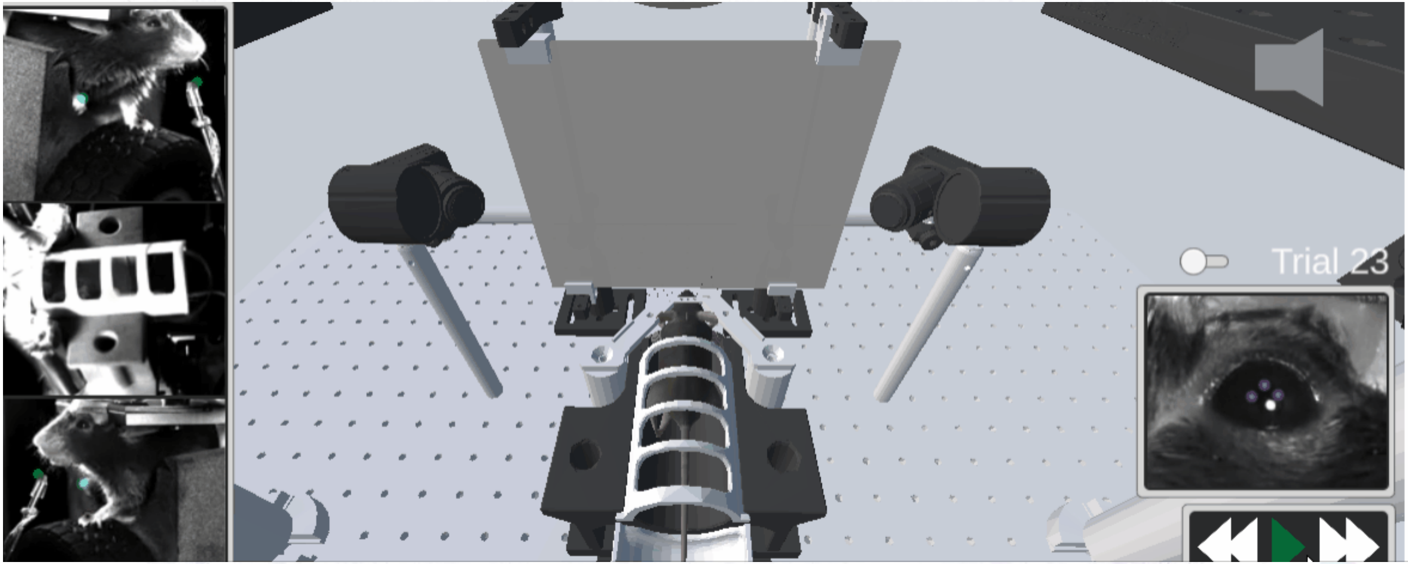
BWM Data

1. Experimental design
2. Data collection strategy
3. Data Integration
4. Getting started:
   installation
   loading data
   searching sessions
5. Summary

Experimental setup

INTERNATIONAL
BRAIN
LABORATORY

3 cameras allowing to record behavioral responses
(licking, face motion, movement)

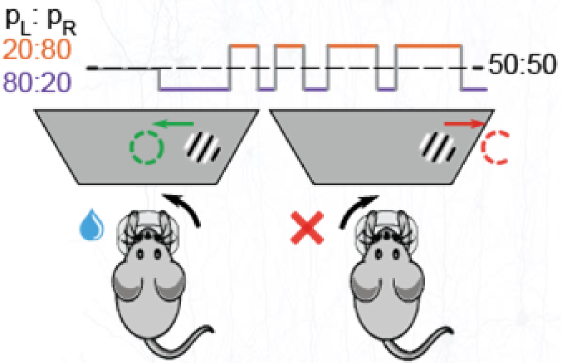


eid: 1a166c4f-4c53-422e-8473-b3cff85e6750
danlab/DY_009/2020-02-29/001

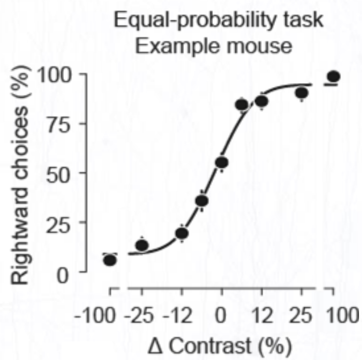
IBL decision-making task

INTERNATIONAL BRAIN LABORATORY

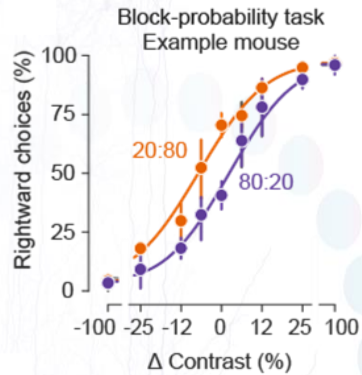
Visual perceptual task



50-50 block
(first 90 trials)



20-80 block
(from 90th trial)



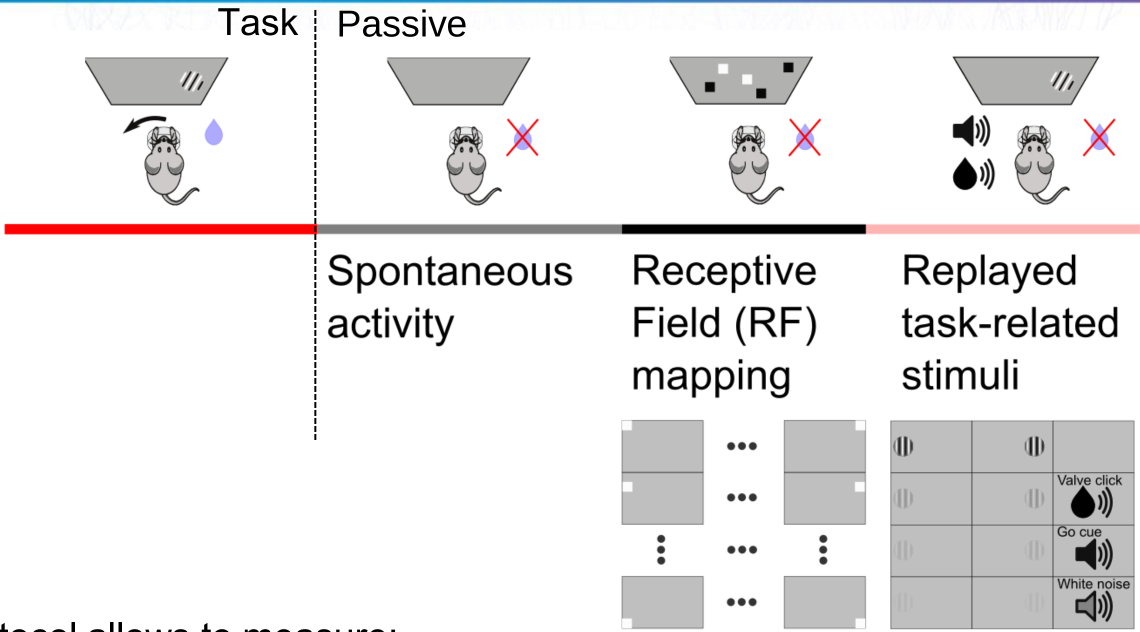
Task allows to measure neural correlates of:

- Sensory stimuli (visual detection)
- Motor response (wheel movement, or direction of movement)
- Reward signal
- Bias/Prior (at low stimuli contrasts)

The IBL et al., *Standardized and reproducible measurement of decision-making in mice*, eLife 2021

IBL passive (after the task, data not released)

INTERNATIONAL BRAIN LABORATORY



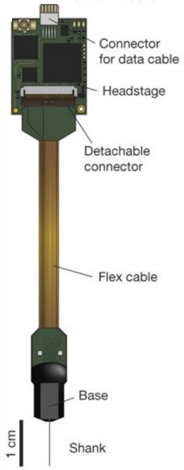
Protocol allows to measure:

- Post-task activity (spontaneous)
- Receptive fields mapping (visual responses)
- Sensory task-related responses (valve clicks, sound cues, visual stimuli)
- Spontaneous movement, licking
- Brain state modulations (activity during task vs passive)

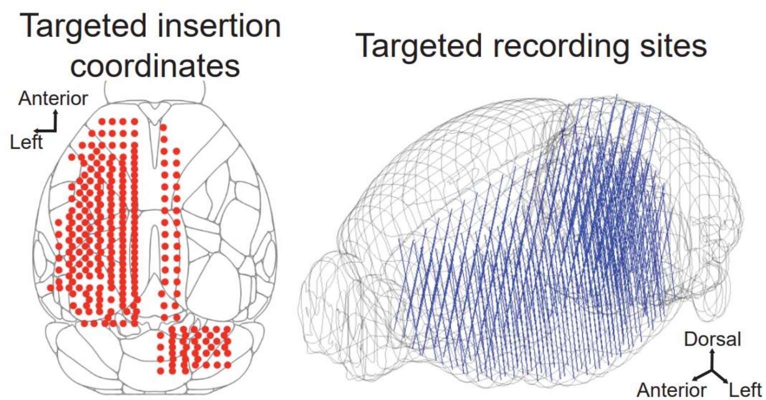
Recording neural activity during behavior

INTERNATIONAL BRAIN LABORATORY

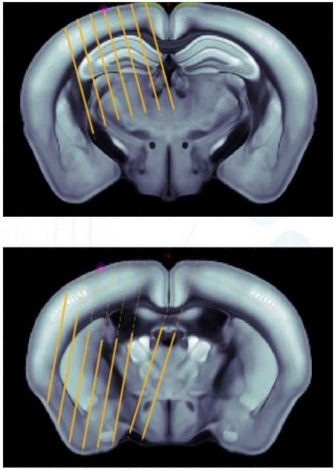
Neuropixels probes
385 active channels



Recordings strategy



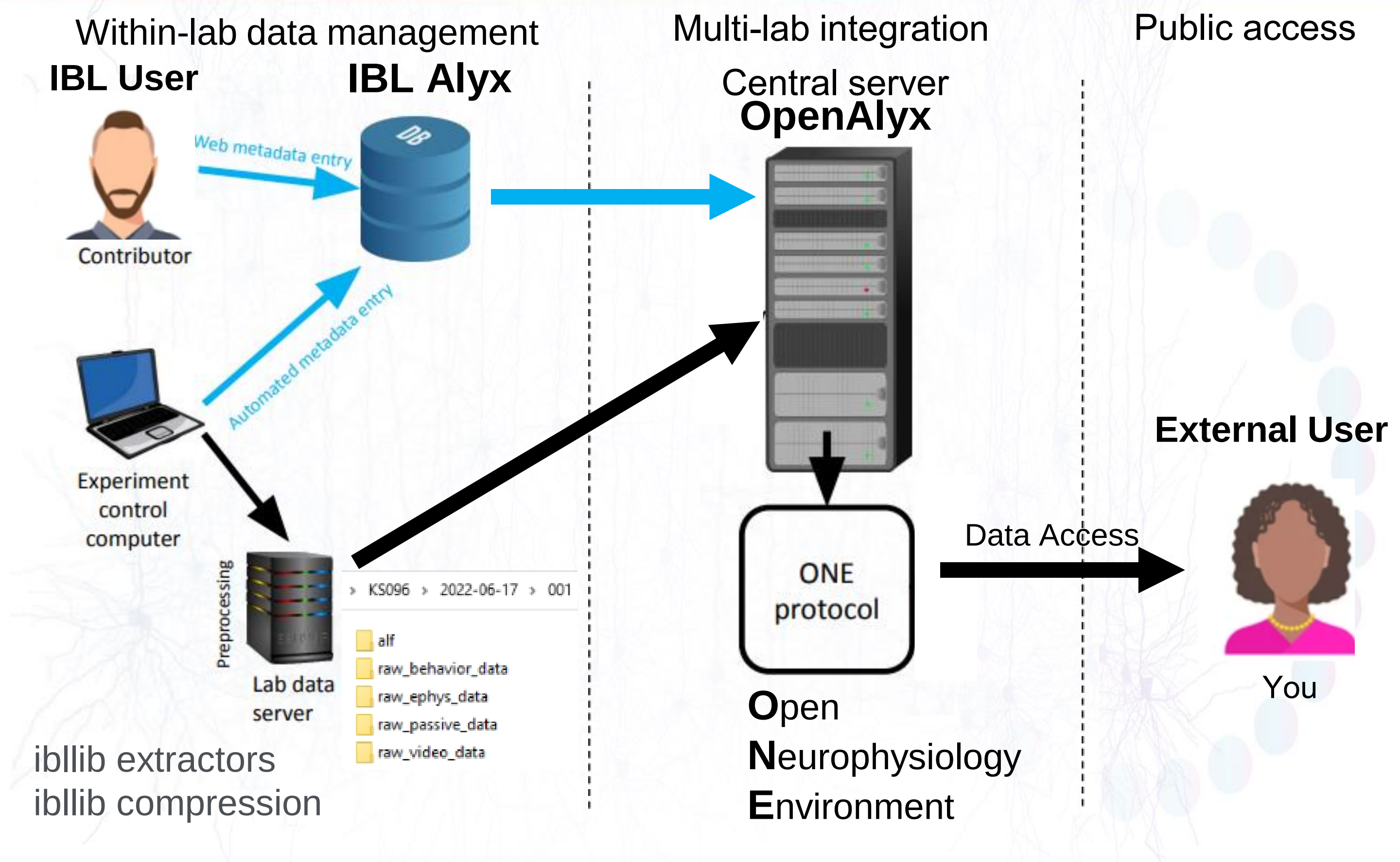
Example recordings sites



Recordings acquisition:
spikeGLX (Bill Karsh, HHMI)

BWM Data Release:
295501 neurons
547 insertions
194 brain regions
115 mice

Data flow within and outside IBL



Data Integration

INTERNATIONAL BRAIN LABORATORY

1. **Raw data**
- Subject-level metadata (Alyx database)
- Behavioral performance (*.json)
- Electrophysiology recordings (compressed *.cbin, *.meta)
- Behavioral video (compressed *.mp4)
- Histological probe track reconstruction (Alyx database)
- Ambient data (temperature, humidity, etc)

2. **Post-processing data (extraction spikes, task signals, and video details)**
- Spike sorting (pykilosort)
- Video tracking (DeepLabCut)
- Task signals extraction (wheel, photodiode, task event signals)
- Synchronization with the probe

3. **Quality Control**
- Hardware
- Electrophysiology recording, videos
- Behavioral performance
- Single units

IBL folder structure

INTERNATIONAL BRAIN LABORATORY

lab / mouse / day / session / *folder_name*

cortexlab / KS091 / 2022-07-06 / 001 / folder_name

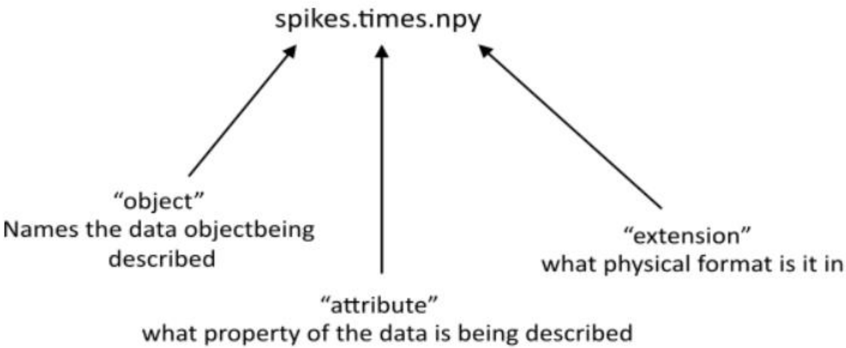
folder_name can be:

- ***alf* – extracted data, including spike-sorted data**
- *raw_ephys_data/probe00* , *raw_ephys_data/probe01*
- *raw_video_data*

```
cortexlab/KS091
|--2022-07-06/
|   |-- 001/
|   |   |--alf/
|   |   |   |-- probe00/
|   |   |   |   |-- pykilosort/
|   |   |   |-- probe01/
|   |   |   |   |-- pykilosort/
|   |   |-- raw_ephys_data/
|   |   |   |-- probe00/
|   |   |   |-- probe01/
|   |   |-- raw_video_data
```

IBL data type: ALF (ALyx Files)

INTERNATIONAL BRAIN LABORATORY



Filename	Size
spikes.times.npy	nSpikes
spikes.amps.npy	nSpikes
spikes.clusters.npy	nSpikes
clusters.amps.npy	nClusters
clusters.mlapv.npy	nClusters x 3

Standard attributes

Event series	`*.times	*.* _times[_timescale]` `e.g. trials.reward_times`
Interval series	`*.intervals[_timescale]` `e.g. tones.intervals`
Continuous timeseries	`*.timestamps` `e.g. video_trialStart.timestamps`

IBL data objects: *trials*, *wheel*, *camera*, *spikes*, *clusters*
https://docs.google.com/document/d/1OqIqqakPakHXRAwceYLwFY9gOrm8_P62XIfCTnHwstg/edit#heading=h.nvzaz0fozs8h

NPY in matlab: https://github.com/kwikteam/npy-matlab/

IBL data structure

INTERNATIONAL
BRAIN
LABORATORY

1. **Raw data (compressed; bin files (mtscomp) https://github.com/int-brain-lab/mtscomp; avi video (ffmpeg))**

2. **ALF (ALyx Files, extracted data)**

A. Post-processing session data, specific to session

- Trial data: trials.table
- Wheel data: wheel.position, wheel.timestamps
- DLC data: camera.dlc, camera.times, camera.ROIMotionEnergy, ROIMotionEnergy.position, licks.times
- Passive data: _ibl_passiveRFM.times, _ibl_passiveStims.table, _ibl_passiveGabor.table

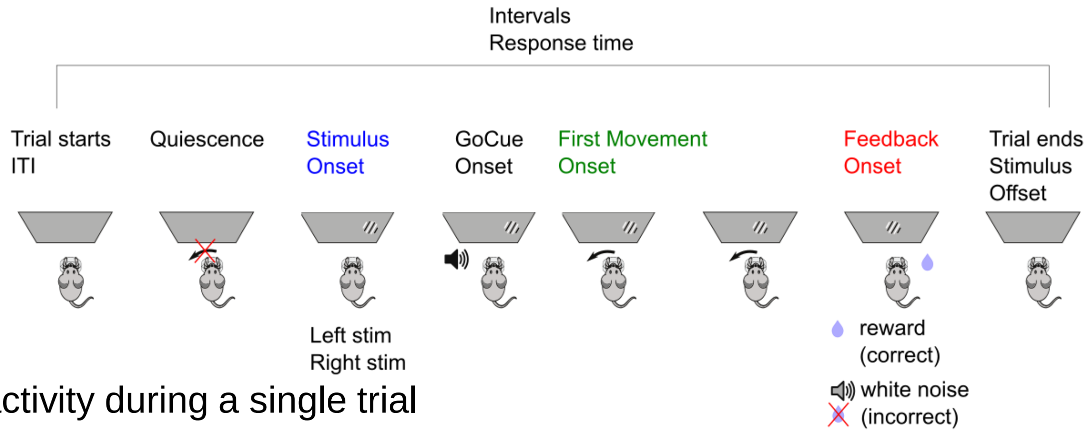
B. Post-processing ephys data, specific to probe

- Ephys data: _spikeglx_sync.channels, _spikeglx_sync.times, _iblqc_ephysSpectralDensity.power, _iblqc_ephysTimeRms.rms
- Spike-sorted data: spikes.times, spikes.clusters, spikes.depths, clusters.metrics, clusters.acronym

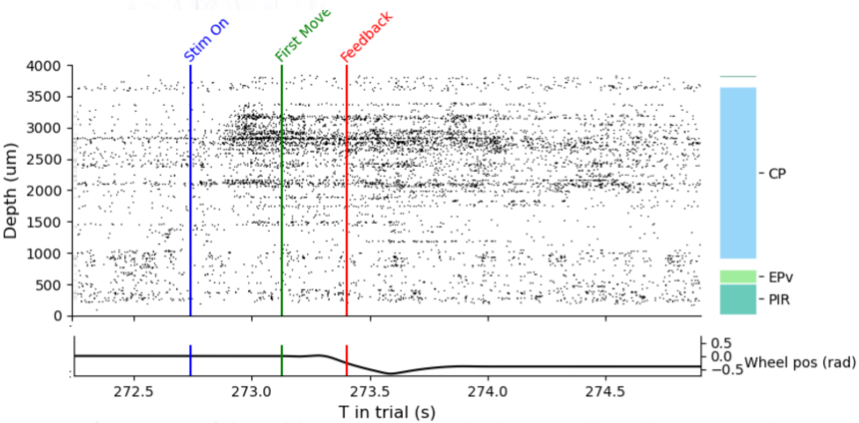
Task trial structure and wheel data

INTERNATIONAL BRAIN LABORATORY

Trial structure



Example neural activity during a single trial

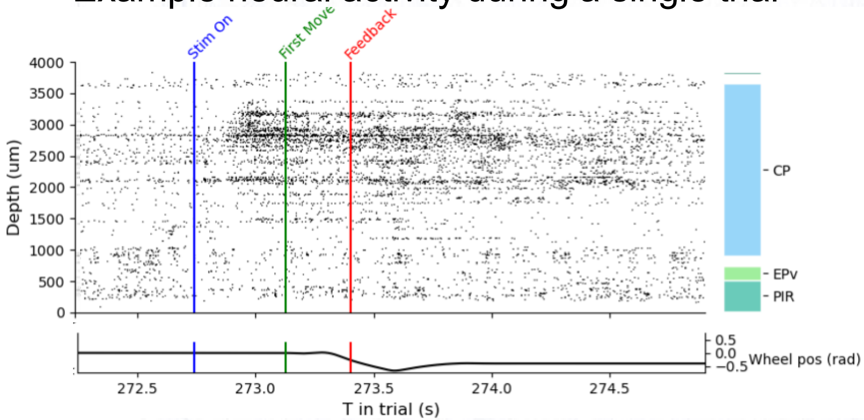


trials.table:
feedback_times,
firstMovement_times, *stimOn_times*,
goCue_times

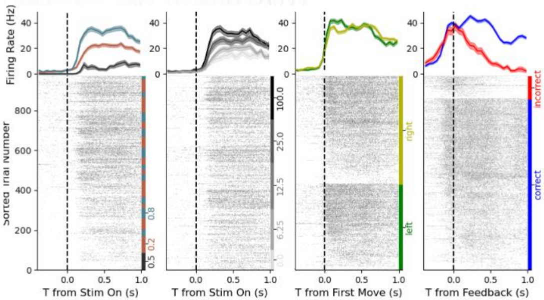
wheel:
_ibl_wheel. position : absolute linear displacement of wheel (cm).
_ibl_wheel. timestamps

Task data – single trial details

Example neural activity during a single trial



Single unit activity aligned to task events



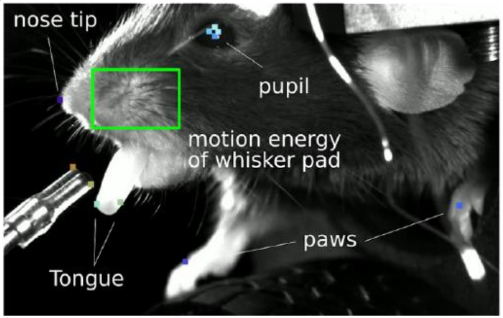
Cluster #493, Caudoputamen

timestamps:
feedback_times, *firstMovement_times*, *stimOn_times,* *goCue_times*

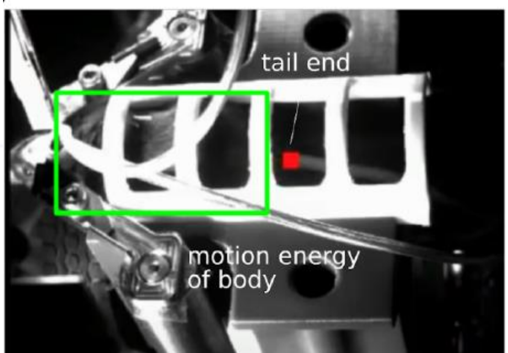
details :
choice : -1 (turn **CCW**), +1 (turn **CW**), or 0 (**no-go**)
contrastLeft : **contrast** of left-side stimulus **0, 6.25%, 12.5%, 25%, 100%**
contrastRight : contrast of right-side stimulus (nan if stimulus is on the other side)
feedbackType : -1 for negative (**incorrect**), 1 for positive (**correct**), 0 for **no-go**
probabilityLeft : **0.5**, **0.2**, **0.8**

DLC data from videos

Right camera (150Hz)
Left camera (60Hz)



Body camera (30 Hz)



DLC data

Details

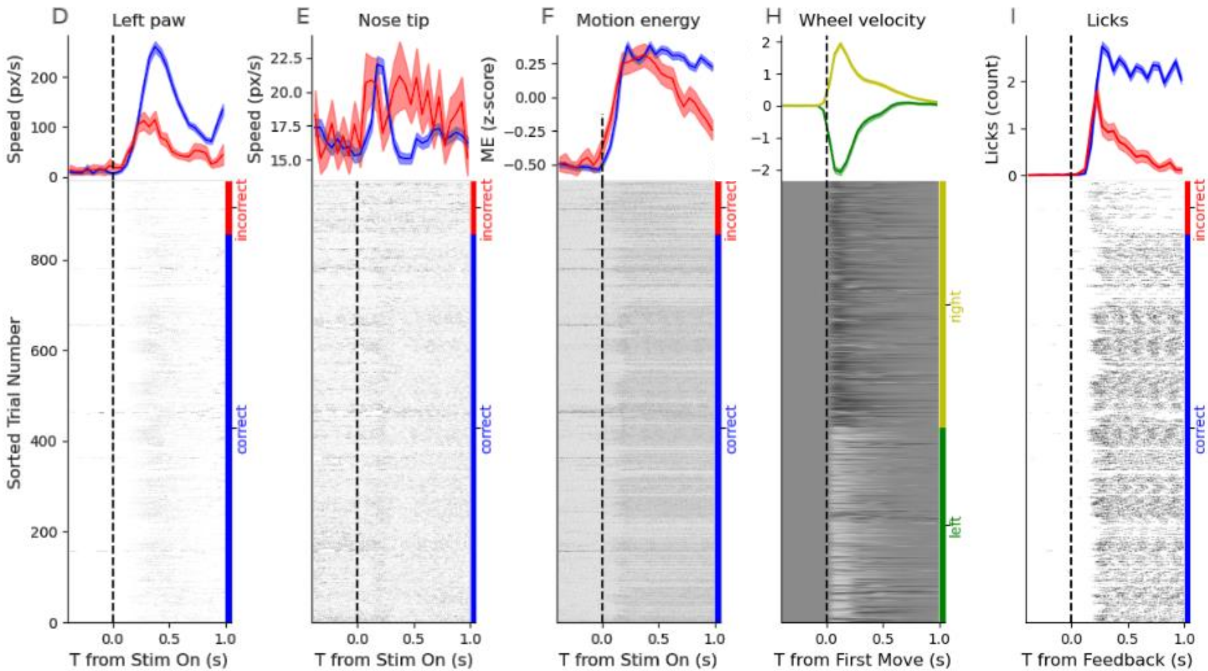
Dataset type	Collection	Dataset name
camera.dlc - LEFT	alf	_ibl_leftCamera.dlc.pqt
camera.dlc - RIGHT	alf	_ibl_rightCamera.dlc.pqt
camera.dlc - BODY	alf	_ibl_bodyCamera.dlc.pqt
camera.times - LEFT	alf	_ibl_leftCamera.times.npy
camera.times - RIGHT	alf	_ibl_rightCamera.times.npy
camera.times - BODY	alf	_ibl_bodyCamera.times.npy
camera.ROIMotionEnergy - LEFT	alf	leftCamera.ROIMotionEnergy.npy
camera.ROIMotionEnergy - RIGHT	alf	rightCamera.ROIMotionEnergy.npy
camera.ROIMotionEnergy - BODY	alf	bodyCamera.ROIMotionEnergy.npy
ROIMotionEnergy.position - LEFT	alf	leftROIMotionEnergy.position.npy
ROIMotionEnergy.position - RIGHT	alf	rightROIMotionEnergy.position.npy
ROIMotionEnergy.position - BODY	alf	bodyROIMotionEnergy.position.npy
camera.features - LEFT	alf	_ibl_leftCamera.features.pqt
camera.features - RIGHT	alf	_ibl_rightCamera.features.pqt
licks.times	alf	licks.times.npy

Raw video data

_iblrig_Camera.raw - LEFT	raw_video_data	_iblrig_leftCamera.raw.mp4
_iblrig_Camera.frameData - LEFT	raw_video_data	_iblrig_leftCamera.frameData.bin

Example plots using DLC variables

INTERNATIONAL BRAIN LABORATORY



How to find session, or specific probe ?

INTERNATIONAL BRAIN LABORATORY

Every session and probe have a unique number:

Probe ID (pid): *00b05238-aa75-4846-a480-c5ffef4529dc*

- Brain region location: clusters.acronym
- Ephys data: spikes.times, spikes.clusters

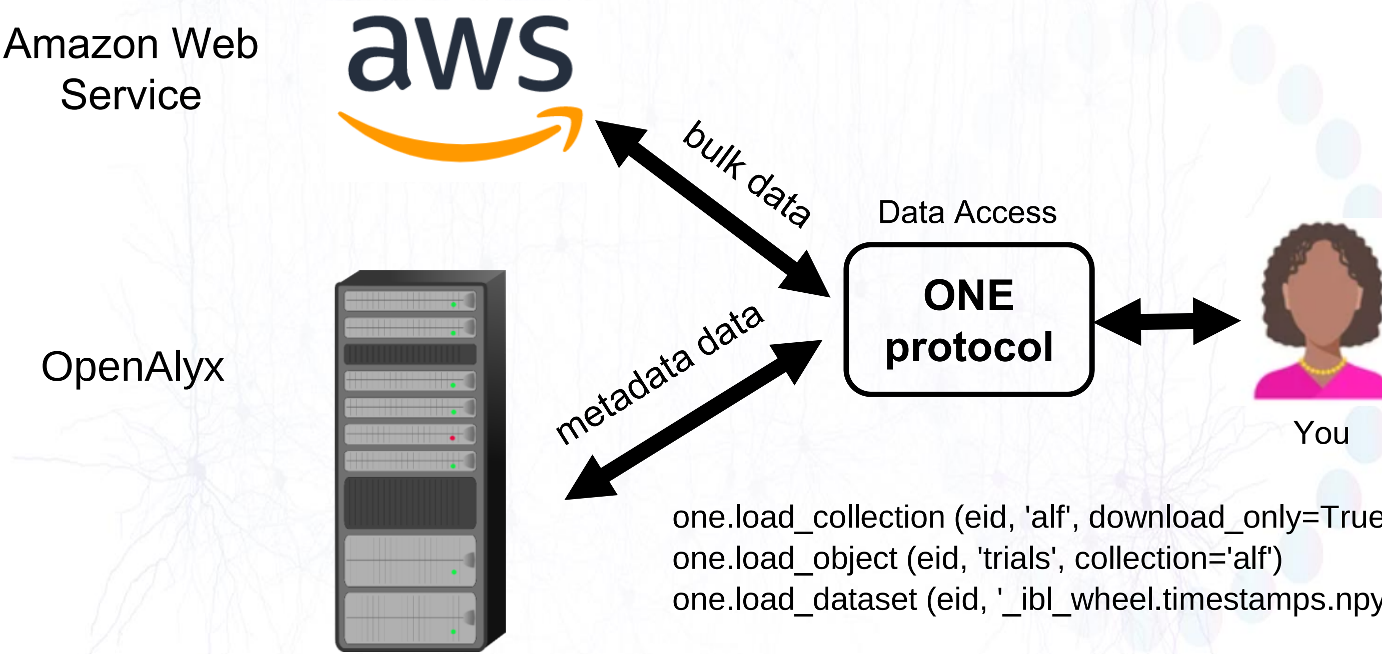
Experiment ID (eid): *258b4a8b-28e3-4c18-9f86-1ea2bc0dc806*

- task trials data: trials.table
- wheel data: wheel.position, wheel.timestamps
- dlc data: camera.dlc, camera.times, licks.times, camera.ROIMotionEnergy
- passive data: _ibl_passiveRFM.times, _ibl_passiveStims.table, _ibl_passiveGabor.table

Getting started
- access public data with ONE

INTERNATIONAL BRAIN LABORATORY

IBL promotes open science, hence developed tools and packages are in **python** +



one.load_collection (eid, 'alf', download_only=True)
one.load_object (eid, 'trials', collection='alf')
one.load_dataset (eid, '_ibl_wheel.timestamps.npy')

Getting started
- example notebook scripts

1. Installation
2. Searching sessions
3. Downloading sessions
4. Loading spikes, and trials to jupyter notebook
5. Query session
6. Plotting data

Documentation:
https://int-brain-lab.github.io/iblenv/notebooks_external/data_download.html

Getting started
- creating env and installation

INTERNATIONAL
BRAIN
LABORATORY

Step01: Create environment

```
conda create --name ibl python=3.9
conda activate ibl
```

Step02: Install IBL packages

```
pip install ONE-api
pip install ibllib
```

ONE-api*: a set of functions that allow you to search the database and download the bulk data to your local computer*

Step 03: Setting up credentials

```
from one.api import ONE
pw = 'international'
one = ONE(base_url='https://openalyx.internationalbrainlab.org',
password=pw, silent=True)
```

Getting started
- cloning BWM github repository

INTERNATIONAL BRAIN LABORATORY

Step04: Clone BWM git repository

```
git clone https://github.com/int-brain-lab/paper-brain-wide-map.git
cd paper-brain-wide-map
pip install -e .
```

List of all sessions:
paper-brain-wide-map / brainwidemap / fixtures / ***2022_10_bwm_release.csv***
List of terms: ['pid', 'eid', 'probe_name', 'session_number', 'date', 'subject', 'lab']

	pid	eid	probe_name	session_number	date	subject	lab
0	56f2a378-78d2-4132-b3c8-8c1ba82be598	6713a4a7-faed-4df2-acab-ee4e63326f8d	probe00	1	2020-02-18	NYU-11	angelakilab
1	47be9ae4-290f-46ab-b047-952bc3a1a509	56956777-dca5-468c-87cb-78150432cc57	probe01	1	2020-02-21	NYU-11	angelakilab
2	6be21156-33b0-4f70-9a0f-65b3e3cd6d4a	56956777-dca5-468c-87cb-78150432cc57	probe00	1	2020-02-21	NYU-11	angelakilab
...	...	...	...	...	...	...	...
542	8bf0f1a4-0d8c-4df3-a99e-f7c81c809652	993c7024-0abc-4028-ad30-d397ad55b084	probe01	1	2020-09-16	CSH_ZAD_029	zadorlab

Getting started
- cloning BWM github repository

INTERNATIONAL BRAIN LABORATORY

Step05: Clone git repository with examples for UCL_NeuroDataShare2023

```
git clone https://github.com/int-brain-lab/UCL_NeuroDataShare2023.git
```

Example01_LaunchONE-api_SearchSe...
Example02_DownloadSession.ipynb
Example03_DownloadRawData.ipynb
Example04_LoadTrials_Spikes.ipynb
Example05_QuerySessions_BrainRegi...
Example06_QueryNeurons_GoodIBLU...
Example07_LoadWheel_plotTrace.ipy...
Example08_LoadData_PlotPSTH.ipynb
Example09_LoadData_PlotFR.ipynb
Example10_ReactionTimes.ipynb

Mayo Faulkner (IBL-BWM tutorials)

Example01: Launch ONE-api and search ephys sessions

INTERNATIONAL BRAIN LABORATORY

1. Using file ***2022_10_bwm_release***

```
from brainwidemap import bwm_query
import numpy as np
from one.api import ONE
one = ONE()
ba = AllenAtlas()

# load info about all released sessions
bwm_df=bwm_query(one=None, alignment_resolved=True, return_details=False, freeze='2022_10_bwm_release')
```

```
Loading bwm_query results from fixtures/2022_10_bwm_release.csv
```

```
bwm_df
```

	pid	eid	probe_name	session_number	date	subject	lab
0	56f2a378-78d2-4132-b3c8-8c1ba82be598	6713a4a7-faed-4df2-acab-ee4e63326f8d	probe00	1	2020-02-18	NYU-11	angelakilab
1	47be9ae4-290f-46ab-b047-952bc3a1a509	56956777-dca5-468c-87cb-78150432cc57	probe01	1	2020-02-21	NYU-11	angelakilab
2	6be21156-33b0-4f70-9a0f-65b3e3cd6d4a	56956777-dca5-468c-87cb-78150432cc57	probe00	1	2020-02-21	NYU-11	angelakilab
3	1e176f17-d00f-49bb-87ff-26d237b525f1	a8a8af78-16de-4841-ab07-fde4b5281a03	probe00	1	2020-01-22	NYU-12	angelakilab
4	701026df-e170-4ca7-88aa-eb0b95ef6ba1	a8a8af78-16de-4841-ab07-fde4b5281a03	probe01	1	2020-01-22	NYU-12	angelakilab
...	...	...	...	...	...	...	...
542	8bf0f1a4-0d8c-4df3-a99e-f7c81c809652	993c7024-0abc-4028-ad30-d397ad55b084	probe01	1	2020-09-16	CSH_ZAD_029	zadorlab
543	5d570bf6-a4c6-4bf1-a14b-2c878c84ef0e	fece187f-b47f-4870-a1d6-619afe942a7d	probe01	1	2020-09-17	CSH_ZAD_029	zadorlab
544	f7c93877-ec05-4091-a003-e69fae0f2fa8	fece187f-b47f-4870-a1d6-619afe942a7d	probe00	1	2020-09-17	CSH_ZAD_029	zadorlab
545	675952a4-e8b3-4e82-a179-cc970d5a8b01	c7bd79c9-c47e-4ea5-aea3-74dda991b48e	probe01	1	2020-09-19	CSH_ZAD_029	zadorlab
546	79f44ba1-c931-4346-82eb-f628a9374045	c7bd79c9-c47e-4ea5-aea3-74dda991b48e	probe00	1	2020-09-19	CSH_ZAD_029	zadorlab

547 rows × 7 columns

Example01_LaunchONE-api_SearchSessions

Example01: Launch ONE-api and search ephys sessions

INTERNATIONAL BRAIN LABORATORY

2. Using **one.search**

```
example_sess.keys()
```

```
dict_keys(['id', 'subject', 'start_time', 'number', 'lab', 'projects', 'url', 'task_protocol'])
```

subject = 'SWC_054’

sessions = **one.search**(subject=subject)
- this gives a list of only **eids** of all detected sessions

```
In [58]: ## searching sessions from sepcific subject
         subject = 'SWC_054'
         # query sessions endpoint
         sessions = one.search(subject=subject)
         print(f'No. of detected sessions: {len(sessions)}')
         sessions

         No. of detected sessions: 57

Out[58]: ['25731502-95bd-4aa7-b5e9-87414a3c4be6',
          '6bb5da8f-6858-4fdd-96d9-c34b3b841593',
          '671c7ea7-6726-4fbe-adeb-f89c2c8e489b',
          'eebacd5a-7dcd-4ba6-9dff-ec2a4d2f19e0',
          '5c7d2345-1f0e-40e5-aad7-2c6133b71b09',
          '6c6983ef-7383-4989-9183-32b1a300d17a',
```

Example01_LaunchONE-api_SearchSessions

Example01: Launch ONE-api and search ephys sessions

INTERNATIONAL
BRAIN
LABORATORY

3. Using **one.alyx.rest**

insertions = **one.alyx.rest**('insertions', 'list', subject=subject)
- this gives **metadata** about session

```
In [59]: ## query insertions endpoint ONE.ALYX.REST
         insertions = one.alyx.rest('insertions', 'list', subject=subject)
         print(f'No. of detected insertions: {len(insertions)}')

         # after quering , each insertion has it is own metadata
         insertions

         No. of detected insertions: 10

Out[59]: [{'id': '7909c0aa-c074-4e19-aabf-b8167c682a5b',
           'session': '6bb5da8f-6858-4fdd-96d9-c34b3b841593',
           'model': '3B2',
           'session_info': {'subject': 'SWC_054',
            'start_time': '2020-10-11T20:00:42.054571',
            'number': 1,
            'lab': 'mrsicflogellab',
            'id': '6bb5da8f-6858-4fdd-96d9-c34b3b841593',
            'task_protocol': '_iblrig_tasks_ephysChoiceWorld6.4.2'},
           'name': 'probe00',
           'json': {'qc': 'WARNING',
            'n_units': 139,
            'xyz_picks': [[-2013, -3024, -43],
             [-2013, -3099, -292],
             [-2013, -3125, -393],
             [-2013, -3149, -568],
```

Example02: Navigate and load data

INTERNATIONAL BRAIN LABORATORY

Use **bwm_query** to generate data frame table with IBL sessions list and get eids and pids from this table:

In [11]:
```
from brainwidemap import bwm_query
import numpy as np
from one.api import ONE
one = ONE()
ba = AllenAtlas()

# load info about all released sessions
bwm_df=bwm_query(one=None, alignment_resolved=True, return_details=False, freeze='2022_10_bwm_release')
```

```
Loading bwm_query results from fixtures/2022_10_bwm_release.csv
```

In [10]:
```
bwm_df
```

Out[10]:

	pid	eid	probe_name	session_number	date	subject	lab
0	56f2a378-78d2-4132-b3c8-8c1ba82be598	6713a4a7-faed-4df2-acab-ee4e63326f8d	probe00	1	2020-02-18	NYU-11	angelakilab
1	47be9ae4-290f-46ab-b047-952bc3a1a509	56956777-dca5-468c-87cb-78150432cc57	probe01	1	2020-02-21	NYU-11	angelakilab

Example02: Download data

INTERNATIONAL BRAIN LABORATORY

1. Using **one.load_collection**

eid=‘288bfbf3-3700-4abe-b6e4-130b5c541e61’
sessions = **one.load_collection**(eid, 'alf', download_only=True)

The format of the returned datasets gives the path of the collection followed by the dataset. e.g in the case of alf/trials.table.pqt, **alf** is the collection and trials.table.pqt is the dataset.

The collection is important as it differentiates datasets with the same name e.g spikes.times in alf/probe00/pykilosort and spikes.times in alf/probe01/pykilosort.

```
collections = one.list_collections(eid)
print(collections)
```

```
['alf/probe01/pykilosort', 'alf/probe00', 'alf/probe00/pykilosort', 'alf/probe01', 'raw_ephys_data/probe00',
obe01', 'alf', 'raw_passive_data', 'raw_ephys_data', 'raw_behavior_data', 'spike_sorters/pykilosort/probe01',
'spike_sorters/pykilosort/probe00', 'spike_sorters/ks2_matlab/probe00', 'spike_sorters/ks2_matlab/probe01']
```

Example02: Download data

INTERNATIONAL BRAIN LABORATORY

2. Using **one.load_dataset**

eid=‘288bfbf3-3700-4abe-b6e4-130b5c541e61’

spike_times = **one.load_dataset** (eid, 'spikes.times.npy’, collection='alf/probe00/pykilosort')

A single dataset can be downloaded and loaded into memory by passing in the eid and dataset as arguments into the one.load_dataset method,

```
# Download and load the left camera timestamps
left_cam_times = one.load_dataset(eid, '_ibl_leftCamera.times.npy')

# Download and load the spikes times for probe00
spike_times = one.load_dataset(eid, 'spikes.times.npy', collection='alf/probe00/pykilosort')
```

Example02: Download data

INTERNATIONAL BRAIN LABORATORY

3. Using **one.load_object**

eid='288bfbf3-3700-4abe-b6e4-130b5c541e61'

trials = **one.load_object**(eid, 'trials', collection='alf')

A group of attributes (e.g amps, depths, metrics) belonging to the same object (e.g clusters) can be downloaded and loaded in one command using the one.load_object method

```
# Load in all trials datasets
trials = one.load_object(eid, 'trials', collection='alf')
wheel = one.load_object(eid, 'wheel', collection='alf')

# Only download the clusters object for probe01
clusters = one.load_object(eid, 'clusters', collection=f'alf/{pname}/pykilosort', download_only=True)

# Only download the spikes object for probe01
spikes = one.load_object(eid, 'spikes', collection=f'alf/{pname}/pykilosort',  download_only=True)
```

Example03: Download raw data

INTERNATIONAL
BRAIN
LABORATORY

Use **one.load_datasets** to load *lf.cbin , *ap.cbin data (use PID)

In [*]:
```
from one.api import ONE
import spikeglx
one = ONE()

pid = 'da8dfec1-d265-44e8-84ce-6ae9c109b8bd'
eid, probe = one.pid2eid(pid)

band = 'ap' # either 'ap','lf'

# Find the relevant datasets and download them
dsets = one.list_datasets(eid, collection=f'raw_ephys_data/{probe}', filename='*.lf.*')
data_files, _ = one.load_datasets(eid, dsets, download_only=True)
bin_file = next(df for df in data_files if df.suffix == '.cbin')
```

```
K:\Flatiron\ONE\alyx.internationalbrainlab.org\hoferlab\Subjects\SWC_043\2020-09-21\001\raw_ephys_data\probe00\_spikegl
```

Use **one.load_datasets** to load *mp4 video files (use EID)

In [ ]:
```
from one.api import ONE
import ibllib.io.video as vidio

one = ONE()
eid = '4ecb5d24-f5cc-402c-be28-9d0f7cb14b3a'
label = 'body' # 'left', 'right' or 'body'

# Load raw video
video_body = one.load_dataset(eid, f'*{label}Camera.raw*', collection='raw_video_data')
```

More examples: https://int-brain-lab.github.io/iblenv/loading_examples.html

Example04: Load session data to jupyter notebook

INTERNATIONAL BRAIN LABORATORY

Use **SessionLoader** to load e.g. trials, wheel data

```
# import session loader
from one.api import ONE
from brainbox.io.one import SessionLoader
one = ONE()
eid='6713a4a7-faed-4df2-acab-ee4e63326f8d'
# instantiate session loader
sess_loader = SessionLoader(one=one, eid=eid)
```

```
SessionLoader(one=One (online, https://alyx.internationalbrainlab.org), session_path=WindowsPath('K:/Flatiron/ONE/alyx.internat
ionalbrainlab.org/angelakilab/Subjects/NYU-11/2020-02-18/001'), eid='6713a4a7-faed-4df2-acab-ee4e63326f8d')
```

```
# Load in trials data
sess_loader.load_trials()

# Load in wheel data
sess_loader.load_wheel()

sess_loader.data_info

sess_loader.trials.keys()
```

```
Index(['stimOff_times', 'goCueTrigger_times', 'intervals_bpod_0',
       'intervals_bpod_1', 'firstMovement_times', 'goCue_times',
       'probabilityLeft', 'response_times', 'feedbackType', 'rewardVolume',
       'contrastRight', 'choice', 'feedback_times', 'stimOn_times',
       'contrastLeft', 'intervals_0', 'intervals_1'],
      dtype='object')
```

Example04: Load session data to jupyter notebook

INTERNATIONAL BRAIN LABORATORY

trials table structure:

```
sess_loader.trials
```

	stimOff_times	goCueTrigger_times	intervals_bpod_0	intervals_bpod_1	firstMovement_times	goCue_times	probabilityLeft	response_times	feedbackType
0	315.645189	268.879047	0.000000	62.662702	312.82632	268.879856	0.5	313.593986	-1.0
1	377.427764	317.356906	63.199299	126.375502	NaN	317.357932	0.5	377.358385	-1.0
2	386.672888	381.061309	126.884199	133.690002	385.42132	381.062242	0.5	385.593944	1.0
3	406.660838	389.144464	134.121899	153.677802	404.07032	389.145186	0.5	404.595102	-1.0

Example04_LoadTrials_Spikes

Example04: Load session data to jupyter notebook

INTERNATIONAL BRAIN LABORATORY

Use **SpikeSortingLoader** to load spikes data

The SpikeSorting loader can be used in spike sorting data for a single insertion. It can be instantiated with an ONE instance and either a **pid or and eid, pname** combination

```
pid='56f2a378-78d2-4132-b3c8-8c1ba82be598'
pname='probe00'
# instantiate with a pid
spike_loader = SpikeSortingLoader(pid=pid, one=one)

# alternatively instantiate with an eid and probe name
spike_loader = SpikeSortingLoader(eid=eid, one=one, pname=pname)

# Download and load data
spikes, clusters, channels = spike_loader.load_spike_sorting()
# Assign brain location information from channels to clusters
clusters = spike_loader.merge_clusters(spikes, clusters, channels)
spikes
```

```
{'depths': array([439.90316772, 273.93023682, 445.34341431, ...,  28.31001663,
        191.40383911,  41.63357162]),
 'clusters': array([ 99,  64, 324, ...,   6,  28,   9], dtype=uint32),
 'times': array([8.38230849e-03, 1.27823046e-02, 1.41823034e-02, ...,
        4.68878040e+03, 4.68878233e+03, 4.68878343e+03]),
 'amps': array([1.48818473e-04, 7.21604797e-05, 7.99972536e-05, ...,
        1.45808202e-04, 8.26257018e-05, 1.57378919e-04])}
```

```
clusters.keys()
```

```
dict_keys(['uuids', 'depths', 'channels', 'cluster_id', 'amp_max', 'amp_min', 'amp_median', 'amp_std_dB', 'contamination', 'con
tamination_alt', 'drift', 'missed_spikes_est', 'noise_cutoff', 'presence_ratio', 'presence_ratio_std', 'slidingRP_viol', 'spike
_count', 'firing_rate', 'label', 'x', 'y', 'z', 'acronym', 'atlas_id', 'axial_um', 'lateral_um'])
```

Example04_LoadTrials_Spikes

Example05: Query sessions

Use **one.alyx.rest** to query sessions from specific brain region

```
In [16]: # Find sessions recorded in specific brain region

         # loading data with SC
         #from oneibl.one import ONE
         from one.api import ONE
         import matplotlib.pyplot as plt
         import pandas as pd
         import numpy as np
         one = ONE()

         ## FIND EIDS IN SPECIIC BRAIN REGIONS
         # example brain regions
         #brainregions_acronyms=["LGD","CP","MOp","VISp","ZI","SNr"]
         brain_region="LGD"
         insertions = one.alyx.rest('insertions', 'list', task_protocol='ephys',
                                    atlas_acronym=brain_region,
                                    project='ibl_neuropixel_brainwide_01', no_cache=True)

         probe_insertions = [p['name'] for p in insertions]
         eid_insertions = [s['session'] for s in insertions]
         pid_insertions=[p['id'] for p in insertions]
         subject_insertions=[m['session_info']['subject'] for m in insertions]
         start_time_insertions = [k['session_info']['start_time'][0:10] for k in insertions]

         data_in={'subject':subject_insertions,'day':start_time_insertions,'probe':probe_insertions,
                  'eid':eid_insertions,'pid':pid_insertions,'brain_region':brain_region}

         # create data frame
         df_experiments=pd.DataFrame(data_in)

         print('found', len(eid_insertions), 'probe recordings from', brain_region)

         found 43 probe recordings from LGD
```

Example05_QuerySessions_BrainRegion

Example06: Query Good-IBL units

INTERNATIONAL BRAIN LABORATORY

In [2]:
```
from one.api import ONE
from brainbox.io.one import SpikeSortingLoader
from ibllib.atlas import AllenAtlas

one = ONE()
ba = AllenAtlas()

pid = 'da8dfec1-d265-44e8-84ce-6ae9c109b8bd'
#LOAD SPIKES
sl = SpikeSortingLoader(pid=pid, one=one, atlas=ba)
spikes, clusters, channels = sl.load_spike_sorting()
clusters = sl.merge_clusters(spikes, clusters, channels)

#Filter GOOD - IBL - UNITS
good_clusterIDs = clusters['cluster_id'][clusters['label'] == 1]
good_clusterIDs
```

Out[2]:
```
array([  0,   1,   6,  35,  37,  38,  44,  49,  51,  53,  54,  63,  80,
        99, 109, 114, 127, 128, 129, 130, 132, 167, 169, 177, 188, 198,
       208, 217, 232, 243, 244, 249, 259, 261, 274, 288, 296, 297, 305,
       312, 313, 320, 325, 326, 327, 328, 339, 340, 341, 355, 356, 360,
       364, 367, 375, 378, 384, 393, 394, 395, 399, 401, 403, 406, 407,
       416, 425, 435, 436, 438, 446, 448, 452, 456, 458, 459, 461, 462,
       464, 469, 472, 475, 476, 479, 482, 488, 489, 494, 495, 500, 505,
       507, 509, 510, 518, 519, 527, 528, 529, 531, 536, 537, 543, 549,
       550, 554, 556, 560, 567, 570, 575, 577, 578, 581, 583, 587, 588,
       590, 594, 595, 597, 598, 599, 601, 607, 612, 613, 615, 619, 622,
       625, 626, 639, 641, 645, 648, 654, 665, 666, 667, 668, 669, 673,
       675, 677, 683, 685, 687, 715, 741, 743, 777, 782, 785, 792, 793,
       804, 812, 821, 826, 830, 856, 861, 869, 913], dtype=int64)
```

Criteria Good-IBL unit:

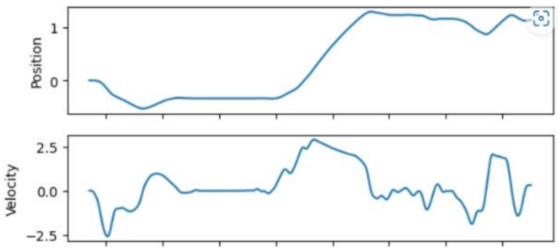
1) amplitude > 50 uV
2) noise cut-off < 20
3) refractory period violation

Example06_QueryNeurons_GoodIBLUnits

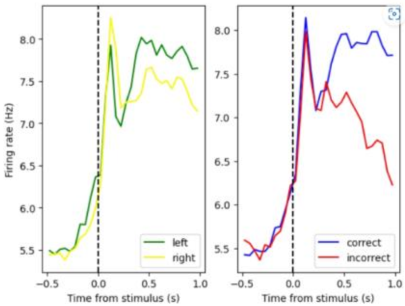
Practical examples

INTERNATIONAL BRAIN LABORATORY

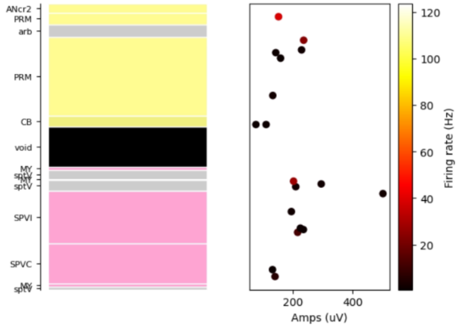
Example07_LoadWheel



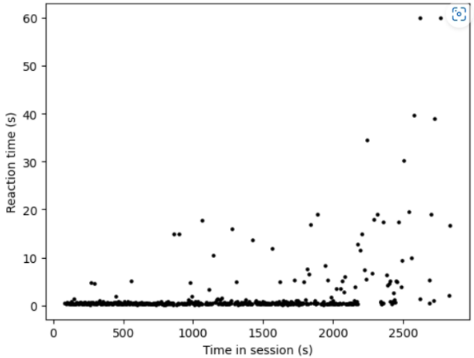
Example08_LoadData_PlotPSTH



Example09_LoadData_PlotFR



Example10_ReactionTimes



Summary

IBL modular data architecture allows:

- organization of data within a lab
- integration of data from multiple labs
- Flexible access during project development
- pipelined analysis, display on an interactive website
- multiple access methods
- Modules can be used by individual labs, large or small collaborations

Useful links

ONE documentation
https://int-brain-lab.github.io/iblenv/notebooks_external/one_quickstart.html#
Viz website
https://viz.internationalbrainlab.org/app
Main website
https://www.internationalbrainlab.com
Task being performed by the mouse-
https://doi.org/10.7554/eLife.63711
Neural data that has been recorded
https://int-brain-lab.github.io/iblenv/notebooks_external/data_release_repro_ephys.html