Pynapple IO
Standard analysis in systems neuroscience

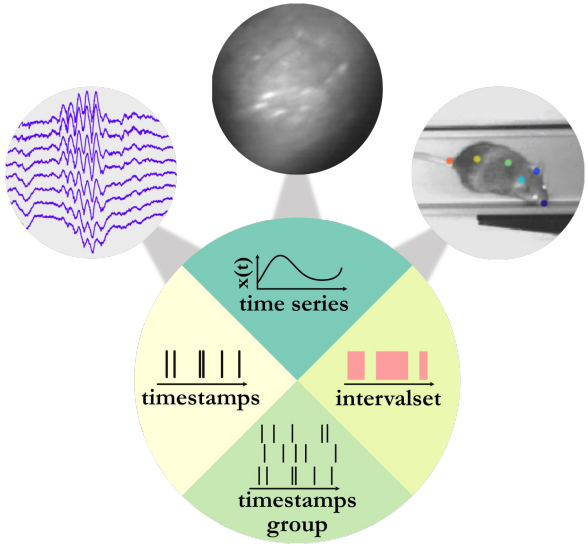
FENS 2024 workshop

Guillaume Viejo

FLATIRON INSTITUTE

Input-output (IO)

Core



FLATIRON INSTITUTE

Loading data

From **spikeinterface** : a unified framework for spike sorting

Raw Data Formats

For raw recording formats, we currently support:

- AlphaOmega `read_alphaomega()`
- Axona `read_axona()`
- BlackRock `read_blackrock()`
- Binary `read_binary()`
- Biocam HDF5 `read_biocam()`
- CED `read_ced()`
- EDF `read_edf()`
- IBL streaming `read_ibl_streaming_recording()`
- Intan `read_intan()`
- MaxWell `read_maxwell()`
- MCS H5 `read_mcsh5()`
- MCS RAW `read_mcsraw()`
- MEArec `read_mearec()`
- Mountainsort MDA `read_mda_recording()`
- Neuralynx `read_neuralynx()`
- Neurodata Without Borders `read_nwb_recording()`
- Neuroscope `read_neuroscope_recording()`
- Neuroexplorer `read_neuroexplorer()`
- NIX `read_nix()`
- Open Ephys Legacy `read_openephys()`
- Open Ephys Binary `read_openephys()`
- Plexon `read_plexon()`
- Plexon 2 `read_plexon2()`
- Shybrid `read_shybrid_recording()`
- SpikeGLX `read_spikeglx()`
- SpikeGLX IBL compressed `read_cbin_ibl()`
- SpikeGLX IBL stream `read_streaming_ibl()`
- Spike 2 `read_spike2()`
- TDT `read_tdt()`
- Zarr `read_zarr()`

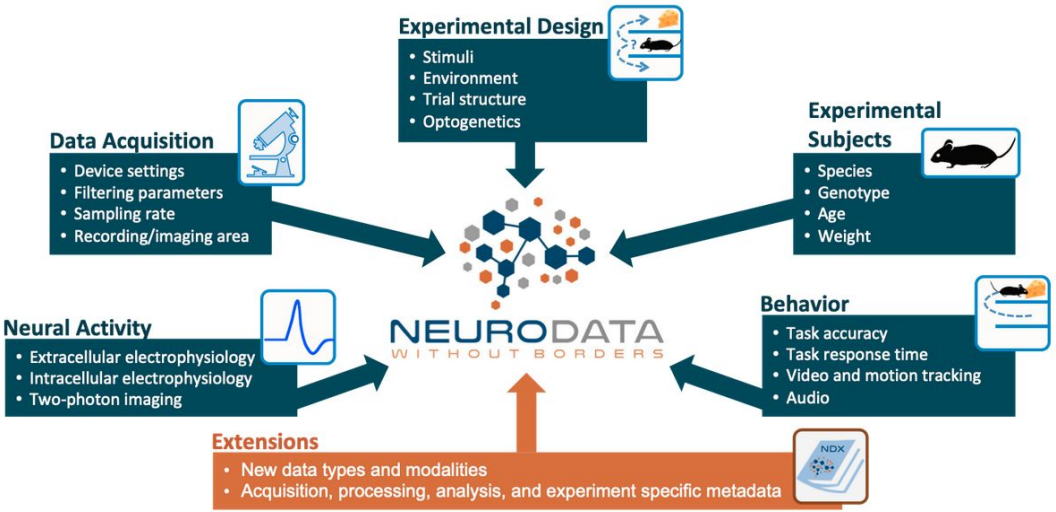
Sorted Data Formats

For sorted data formats, we currently support:

- BlackRock `read_blackrock_sorting()`
- Combinato `read_combinato()`
- Cell explorer `read_cellexplorer()`
- HerdingSpikes2 `read_herdingspikes()`
- HDsort `read_hdsort()`
- Kilosort1/2/2.5/3 `read_kilosort()`
- Klusta `read_klusta()`
- MClust `read_mclust()`
- MEArec `read_mearec()`
- Mountainsort MDA `read_mda_sorting()`
- Neurodata Without Borders `read_nwb_sorting()`
- Neuroscope `read_neuroscope_sorting()`
- Neuralynx spikes `read_neuralynx_sorting()`
- NPZ (created by SpikeInterface) `read_npz_sorting()`
- Plexon spikes `read_plexon_sorting()`
- Plexon 2 spikes `read_plexon2_sorting()`
- Shybrid `read_shybrid_sorting()`
- Spyking Circus `read_spykingcircus()`
- Trideclous `read_tridesclous()`
- Wave Clus `read_waveclus()`
- YASS `read_yass()`

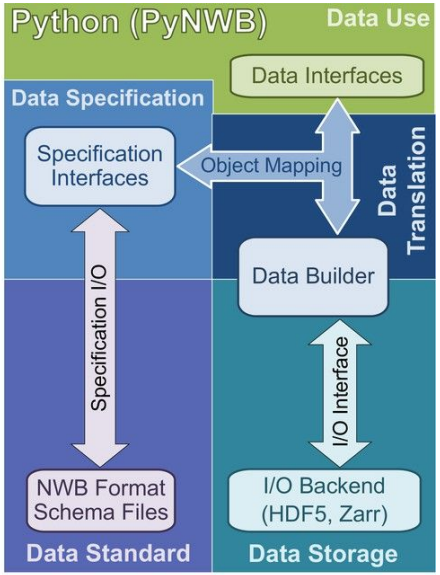
On universal standard : the neurodata without borders format (NWB)

Neurodata Without Borders (NWB) is a data standard for neurophysiology, providing neuroscientists with a common standard to share, archive, use, and build common analysis tools for neurophysiology data.



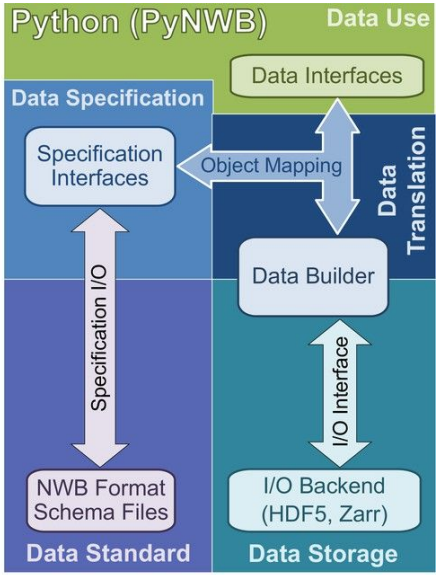
Teeters, J. L., Godfrey,K., Young, R., Dang, C., Friedsam, C., Wark, B., ... & Sommer, F. T. (2015). Neurodata without borders: creating a common data format for neurophysiology. *Neuron*, *88*(4), 629-634.

Rübel, O., Tritt, A., Ly, R., Dichter, B. K., Ghosh, S., Niu, L., ... & Bouchard, K. E. (2022). The neurodata without borders ecosystem for neurophysiological data science. *Elife*, *11*, e78362.



Teeters, J. L., Godfrey,K., Young, R., Dang, C., Friedsam, C., Wark, B., ... & Sommer, F. T. (2015). Neurodata without borders: creating a common data format for neurophysiology. *Neuron*, *88*(4), 629-634.

Rübel, O., Tritt, A., Ly, R., Dichter, B. K., Ghosh, S., Niu, L., ... & Bouchard, K. E. (2022). The neurodata without borders ecosystem for neurophysiological data science. *Elife*, *11*, e78362.



pynwb Public

A Python API for working with Neurodata stored in the NWB Format

Python 158 78

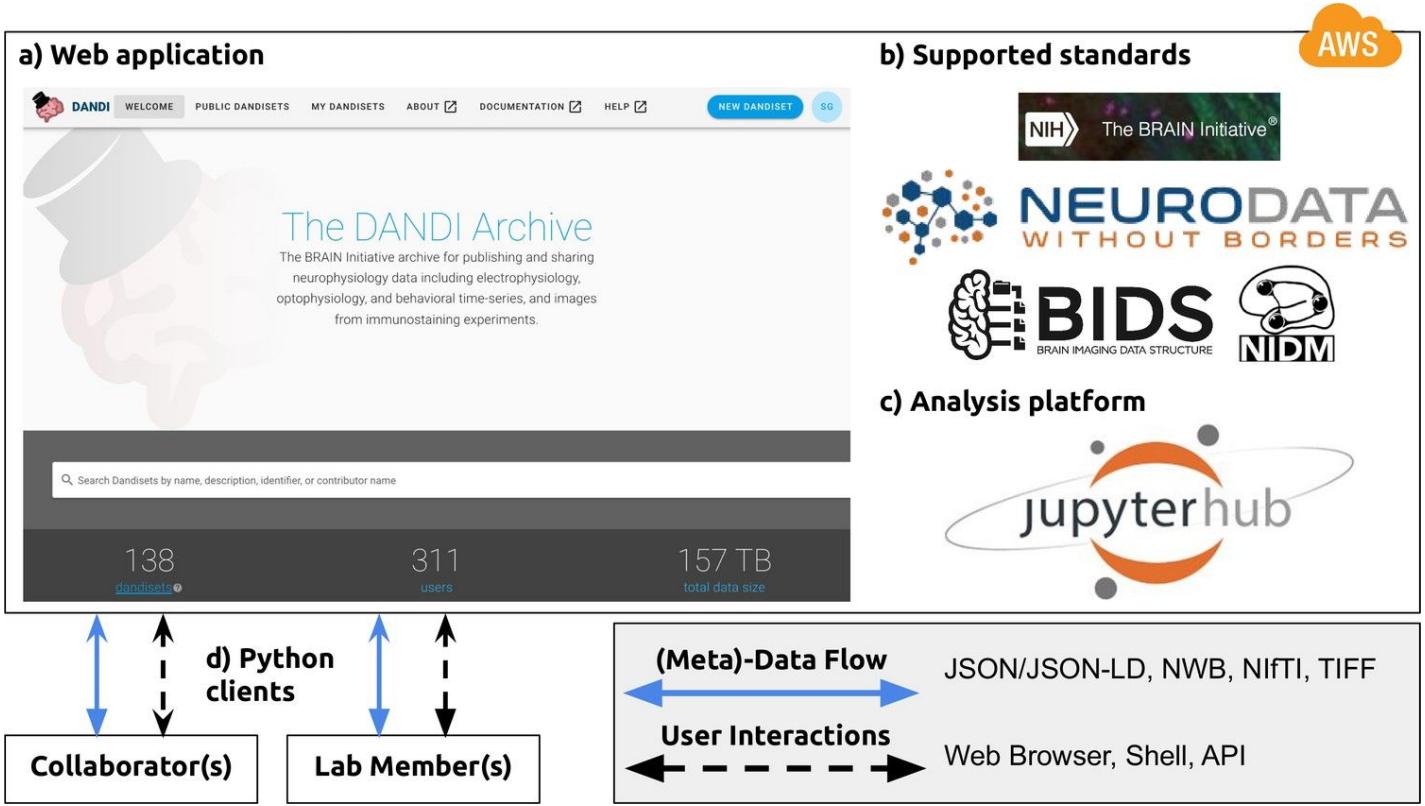
$ pip install pynwb

Teeters, J. L., Godfrey,K., Young, R., Dang, C., Friedsam, C., Wark, B., ... & Sommer, F. T. (2015). Neurodata without borders: creating a common data format for neurophysiology. *Neuron*, *88*(4), 629-634.

Rübel, O., Tritt, A., Ly, R., Dichter, B. K., Ghosh, S., Niu, L., ... & Bouchard, K. E. (2022). The neurodata without borders ecosystem for neurophysiological data science. *Elife*, *11*, e78362.

Sharing is caring : the open data era

NWB is foundational for the DANDI data repository to enable collaborative data sharing.



Rübel, O., Tritt, A., Ly, R., Dichter, B. K., Ghosh, S., Niu, L., ... & Bouchard, K. E. (2022). The neurodata without borders ecosystem for neurophysiological data science. *Elife*, *11*, e78362.

Loading NWB with pynapple

NWB interface

In [1]:

```
1 import pynapple as nap
2
3 nwb = nap.load_file("A2929-200711.nwb")
```

FLATIRON INSTITUTE

NWB interface

In [1]:
```
1 import pynapple as nap
2
3 nwb = nap.load_file("A2929-200711.nwb")
```

In [2]:
```
1 nwb
```

A2929-200711.nwb

Keys	Type
position_time_support	IntervalSet
epochs	IntervalSet
z	Tsd
y	Tsd
x	Tsd
rz	Tsd
ry	Tsd
rx	Tsd

NWB interface

In [1]:
```
import pynapple as nap

nwb = nap.load_file("A2929-200711.nwb")
```

In [2]:
```
nwb
```

A2929-200711.nwb

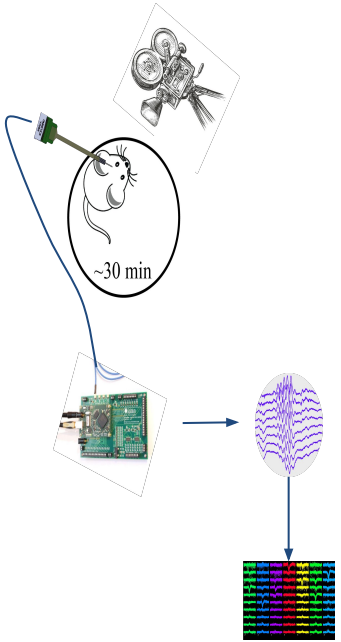
Keys	Type
position_time_support	IntervalSet
epochs	IntervalSet
z	Tsd
y	Tsd
x	Tsd
rz	Tsd
ry	Tsd
rx	Tsd

In [3]:
```
nwb["position_time_support"]
```

Out[3]:

	start	end
0	670.6407	1199.99495

NWB interface



pypi package 0.4.4 | Minimal and Full Tests passing | Upload Package to PyPI passing
codecov 91% | docs passing | license BSD-3-Clause

NeuroConv

Automatically convert neurophysiology data to NWB

https://github.com/catalystneuro/neuroconv

```
In [1]:
1 import pynapple as nap
2
3 nwb = nap.load_file("A2929-200711.nwb")
```

```
In [2]:
1 nwb
```

A2929-200711.nwb

Keys	Type
position_time_support	IntervalSet
epochs	IntervalSet
z	Tsd
y	Tsd
x	Tsd
rz	Tsd
ry	Tsd
rx	Tsd

```
In [3]:
1 nwb["position_time_support"]
```

Out[3]:

	start	end
0	670.6407	1199.99495

The Brain Imaging Data Structure (BIDS)

FLATIRON INSTITUTE

BIDS in 10 seconds

- Folder organisation

```
project/
└── subject
    └── session
        └── datatype
```

- Filename template

key1 - value1 _ key2 - value2 _ suffix .extension

- Metadata files (json sidecar)

```
{
    "key": "value",
    "key2": "value2",
    "key3": {
        "subkey1": "subvalue1"
    }
}
```

Pynapple interface to a BIDS dataset

Datasets interface

```python
import numpy as np
import pynapple as nap

# mkdocs_gallery_thumbnail_path = '_static/treeview.png'

project_path = "../../your/path/to/MyProject"

project = nap.load_folder(project_path)

print(project)
```

Out:

```
MyProject
└── sub-A2929
```

Datasets interface

```
import numpy as np
import pynapple as nap

# mkdocs_gallery_thumbnail_path = '_static/treeview.png'

project_path = "../../your/path/to/MyProject"

project = nap.load_folder(project_path)

print(project)
```

Out:

```
MyProject
└── sub-A2929
```

```
project.view
```

Out:

```
MyProject
└── sub-A2929
    └── ses-A2929-200711
        ├── derivatives
        │   ├── spikes.npz      |        TsGroup
        │   ├── sleep_ep.npz    |        IntervalSet
        │   ├── position.npz    |        TsdFrame
        │   └── wake_ep.npz     |        IntervalSet
        ├── pynapplenwb
        │   └── A2929-200711    |        NWB file
        ├── x_plus_y.npz        |      Tsd
        └── stimulus-fish.npz   |        IntervalSet
```

Datasets interface

```
session = project["sub-A2929"]["ses-A2929-200711"]

print(session)
```

Out:

```
ses-A2929-200711
├── derivatives
├── pynapplenwb
├── x_plus_y.npz     |         Tsd
└── stimulus-fish.npz        |         IntervalSet
```

```
project.view
```

Out:

```
MyProject
└── sub-A2929
    └── ses-A2929-200711
        ├── derivatives
        │   ├── spikes.npz       |         TsGroup
        │   ├── sleep_ep.npz     |         IntervalSet
        │   ├── position.npz     |         TsdFrame
        │   └── wake_ep.npz      |         IntervalSet
        ├── pynapplenwb
        │   └── A2929-200711     |         NWB file
        ├── x_plus_y.npz     |         Tsd
        └── stimulus-fish.npz        |         IntervalSet
```

Datasets interface

```
session = project["sub-A2929"]["ses-A2929-200711"]

print(session)
```

Out:

```
ses-A2929-200711
├── derivatives
├── pynapplenwb
├── x_plus_y.npz     |        Tsd
└── stimulus-fish.npz       |        IntervalSet
```

```
print(session.expand())
```

Out:

```
ses-A2929-200711
├── derivatives
│   ├── spikes.npz      |        TsGroup
│   ├── sleep_ep.npz    |        IntervalSet
│   ├── position.npz    |        TsdFrame
│   └── wake_ep.npz     |        IntervalSet
├── pynapplenwb
│   └── A2929-200711    |        NWB file
├── x_plus_y.npz     |        Tsd
└── stimulus-fish.npz       |        IntervalSet
None
```

Datasets interface

```
print(session["derivatives"]["spikes"])
```

Out:

Index	rate	group	location
0	7.3	0	adn
1	5.73	0	adn
2	8.12	0	adn
3	6.68	0	adn
4	10.77	0	adn
5	11	0	adn
6	16.52	0	adn
7	2.2	1	ca1
8	2.02	1	ca1
9	1.07	1	ca1
10	3.92	1	ca1
11	3.31	1	ca1
12	1.09	1	ca1
13	1.28	1	ca1
14	1.32	1	ca1

```
print(session.expand())
```

Out:

```
📂 ses-A2929-200711
├── 📂 derivatives
│   ├── spikes.npz      |        TsGroup
│   ├── sleep_ep.npz    |        IntervalSet
│   ├── position.npz    |        TsdFrame
│   └── wake_ep.npz     |        IntervalSet
├── 📂 pynapplenwb
│   └── A2929-200711    |        NWB file
├── x_plus_y.npz    |        Tsd
└── stimulus-fish.npz       |        IntervalSet
None
```

Datasets interface

```
session["derivatives"].doc("spikes")


session["derivatives"].doc("position")
```

Out:

```
┌─ ../../your/path/to/MyProject/sub-A2929/ses-A2929-200711/derivatives/spikes.npz ─┐
│ time : 2023-07-11 12:40:10.338066                                                │
│ info : Neurons recorded simultaneously in ADN and CA1                            │
└──────────────────────────────────────────────────────────────────────────────────┘
┌─ ../../your/path/to/MyProject/sub-A2929/ses-A2929-200711/derivatives/position.npz ─┐
│ time : 2023-07-11 12:40:10.364061                                                  │
│ info : Position and head-direction of the mouse recorded with Optitrack            │
└────────────────────────────────────────────────────────────────────────────────────┘
```

```
print(session.expand())
```

Out:

```
📂 ses-A2929-200711
├── 📂 derivatives
│   ├── spikes.npz      |        TsGroup
│   ├── sleep_ep.npz    |        IntervalSet
│   ├── position.npz    |        TsdFrame
│   └── wake_ep.npz     |        IntervalSet
├── 📂 pynapplenwb
│   └── A2929-200711    |        NWB file
├── x_plus_y.npz    |        Tsd
└── stimulus-fish.npz       |        IntervalSet
None
```

Datasets interface : saving a pynapple object

```
tsd = position["x"] + 1
epoch = nap.IntervalSet(start=np.array([0, 3]), end=np.array([1, 6]))

session.save("x_plus_1", tsd, description="Random position")
session.save("stimulus-fish", epoch, description="Fish pictures to V1")
```

Datasets interface : saving a pynapple object

```
tsd = position["x"] + 1
epoch = nap.IntervalSet(start=np.array([0, 3]), end=np.array([1, 6]))

session.save("x_plus_1", tsd, description="Random position")
session.save("stimulus-fish", epoch, description="Fish pictures to V1")
```

```
session.expand()
```

Out:

```
ses-A2929-200711
├── derivatives
│   ├── spikes.npz      |       TsGroup
│   ├── sleep_ep.npz    |       IntervalSet
│   ├── position.npz    |       TsdFrame
│   └── wake_ep.npz     |       IntervalSet
├── pynapplenwb
│   └── A2929-200711    |       NWB file
├── x_plus_y.npz    |       Tsd
├── stimulus-fish.npz       |       IntervalSet
└── x_plus_1.npz    |       Tsd
```

Datasets interface : saving a pynapple object

```
tsd = position["x"] + 1
epoch = nap.IntervalSet(start=np.array([0, 3]), end=np.array([1, 6]))

session.save("x_plus_1", tsd, description="Random position")
session.save("stimulus-fish", epoch, description="Fish pictures to V1")
```

```
session.doc("stimulus-fish")
```

Out:

```
┌─ ../../your/path/to/MyProject/sub-A2929/ses-A2929-200711/stimulus-fish.npz ─┐
│ time : 2023-10-03 17:48:57.817177                                           │
│ info : Fish pictures to V1                                                  │
└─────────────────────────────────────────────────────────────────────────────┘
```

Datasets interface : saving a pynapple object

```
tsd = position["x"] + 1
epoch = nap.IntervalSet(start=np.array([0, 3]), end=np.array([1, 6]))

session.save("x_plus_1", tsd, description="Random position")
session.save("stimulus-fish", epoch, description="Fish pictures to V1")
```

```
session["x_plus_1"]
```

Out:

```
Time (s)
----------  --------
670.6407    0.957143
670.649     0.956137
670.65735   0.955147
670.66565   0.954213
670.674     0.953244
...
1199.9616   0.953244
1199.96995  0.953244
1199.97825  0.953244
1199.9866   0.953244
1199.99495  0.953244
dtype: float64, shape: (63527,)
```

Pynapple IO

In [1]:
```
import pynapple as nap

nwb = nap.load_file("A2929-200711.nwb")
```

In [2]:
```
nwb
```

A2929-200711.nwb

Keys	Type
position_time_support	IntervalSet
epochs	IntervalSet
z	Tsd
y	Tsd
x	Tsd
rz	Tsd
ry	Tsd
rx	Tsd

In [3]:
```
nwb["position_time_support"]
```

Out[3]:

	start	end
0	670.6407	1199.99495

```
project = nap.load_folder(project_path)
```

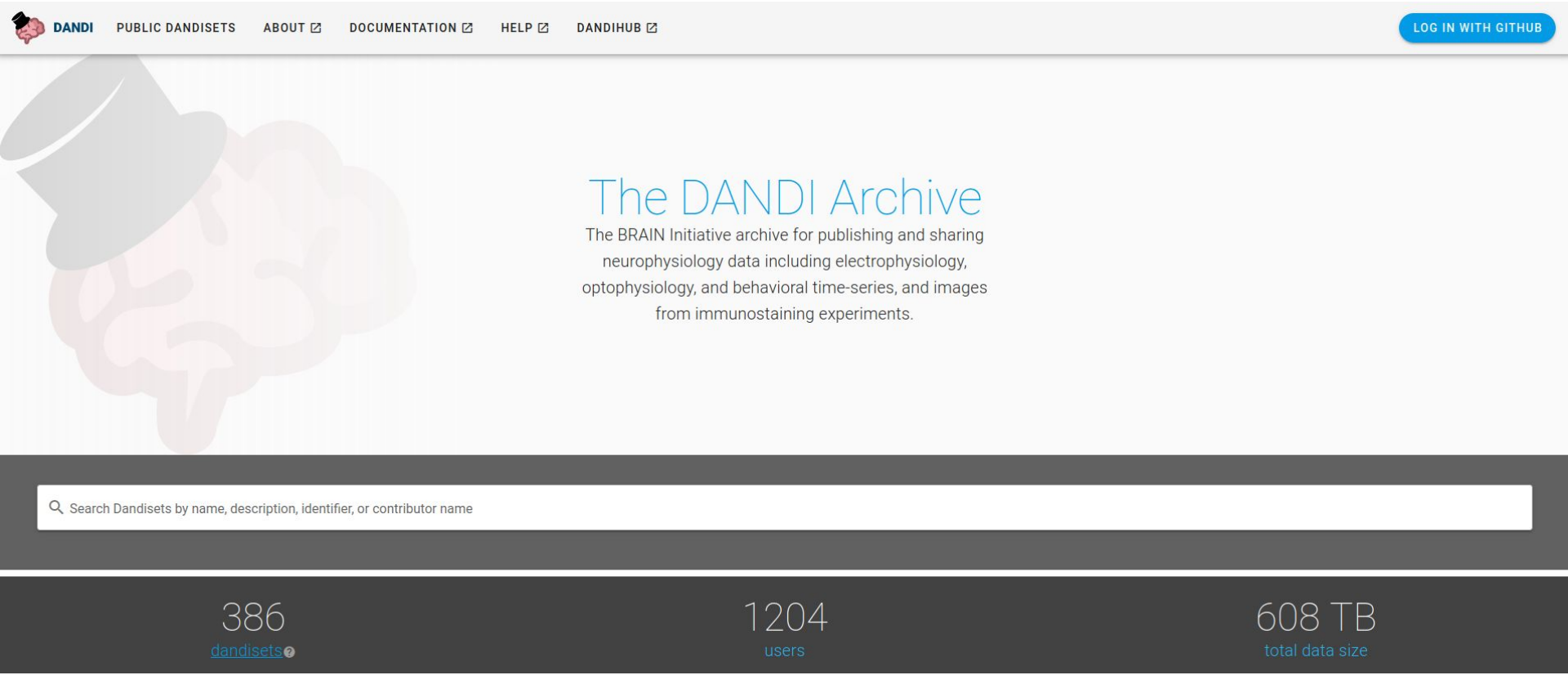
```
project.view
```

Out:

```
MyProject
└── sub-A2929
    └── ses-A2929-200711
        ├── derivatives
        │   ├── spikes.npz      |       TsGroup
        │   ├── sleep_ep.npz    |       IntervalSet
        │   ├── position.npz    |       TsdFrame
        │   └── wake_ep.npz     |       IntervalSet
        ├── pynapplenwb
        │   └── A2929-200711    |       NWB file
        ├── x_plus_y.npz        |     Tsd
        └── stimulus-fish.npz   |       IntervalSet
```

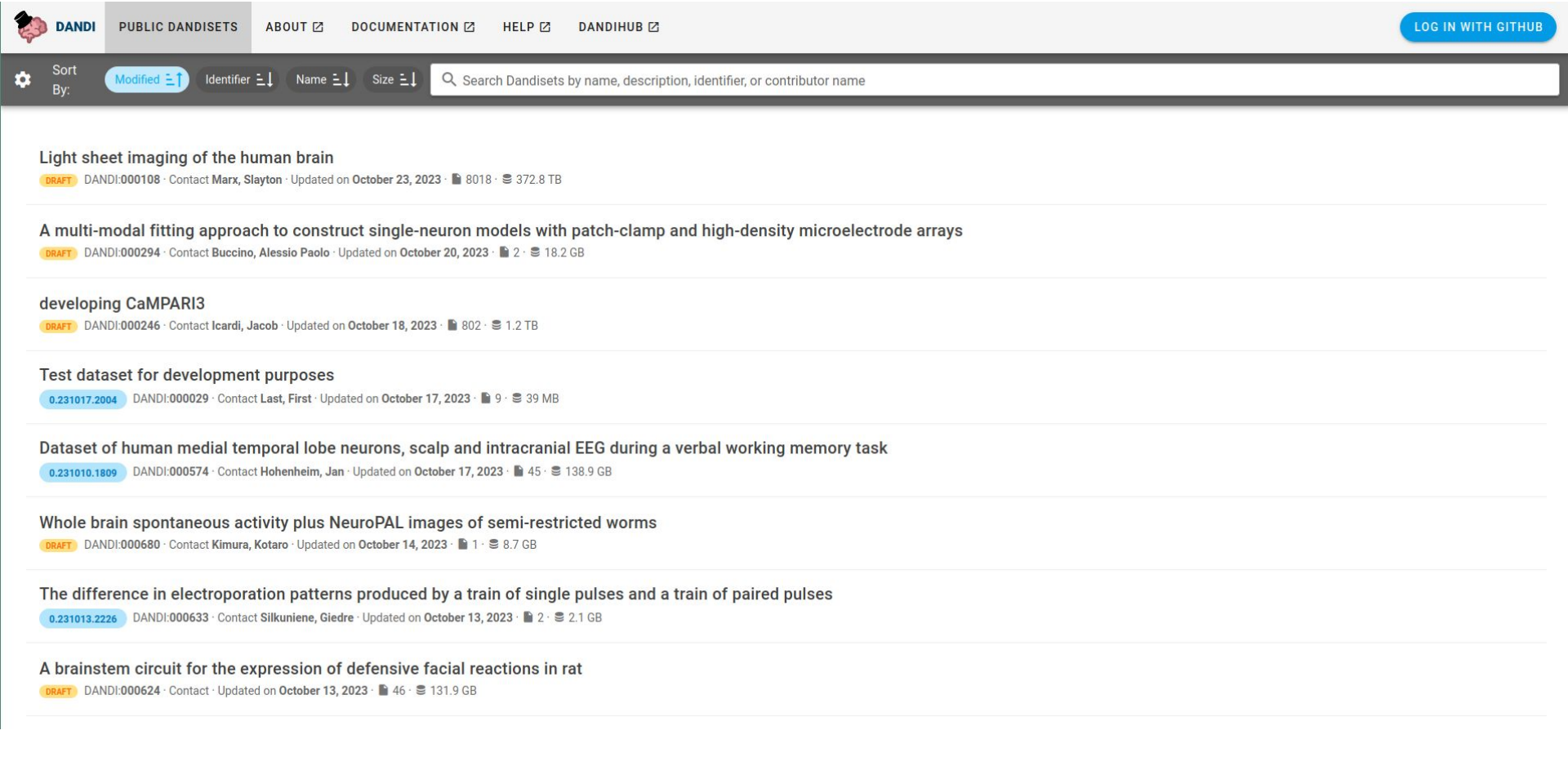
What if I don’t have data yet.

https://dandiarchive.org/



https://dandiarchive.org/

FLATIRON INSTITUTE



https://dandiarchive.org/

FLATIRON INSTITUTE

DANDI | PUBLIC DANDISETS | ABOUT | DOCUMENTATION | HELP | DANDIHUB | LOG IN WITH GITHUB

Sort By: Modified | Identifier | Name | Size

Search Dandisets by name, description, identifier, or contributor name

Light sheet imaging of the human brain
DRAFT DANDI:000108 · Contact Marx, Slayton · Updated on October 23, 2023 · 8018 · 372.8 TB

A multi-modal fitting approach to construct single-neuron models with patch-clamp and high-density microelectrode arrays
DRAFT DANDI:000294 · Contact Buccino, Alessio Paolo · Updated on October 20, 2023 · 2 · 18.2 GB

developing CaMPARI3
DRAFT DANDI:000246 · Contact Icardi, Jacob · Updated on October 18, 2023 · 802 · 1.2 TB

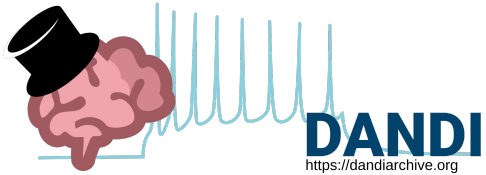
Test dataset for development purposes
0.231017.2004 DANDI:000029 · Contact Last, First · Updated on October 17, 2023 · 9 · 39 MB

Dataset of human medial temporal lobe neurons, scalp and intracranial EEG during a verbal working memory task
0.231010.1809 DANDI:000574 · Contact Hohenheim, Jan · Updated on October 17, 2023 · 45 · 138.9 GB

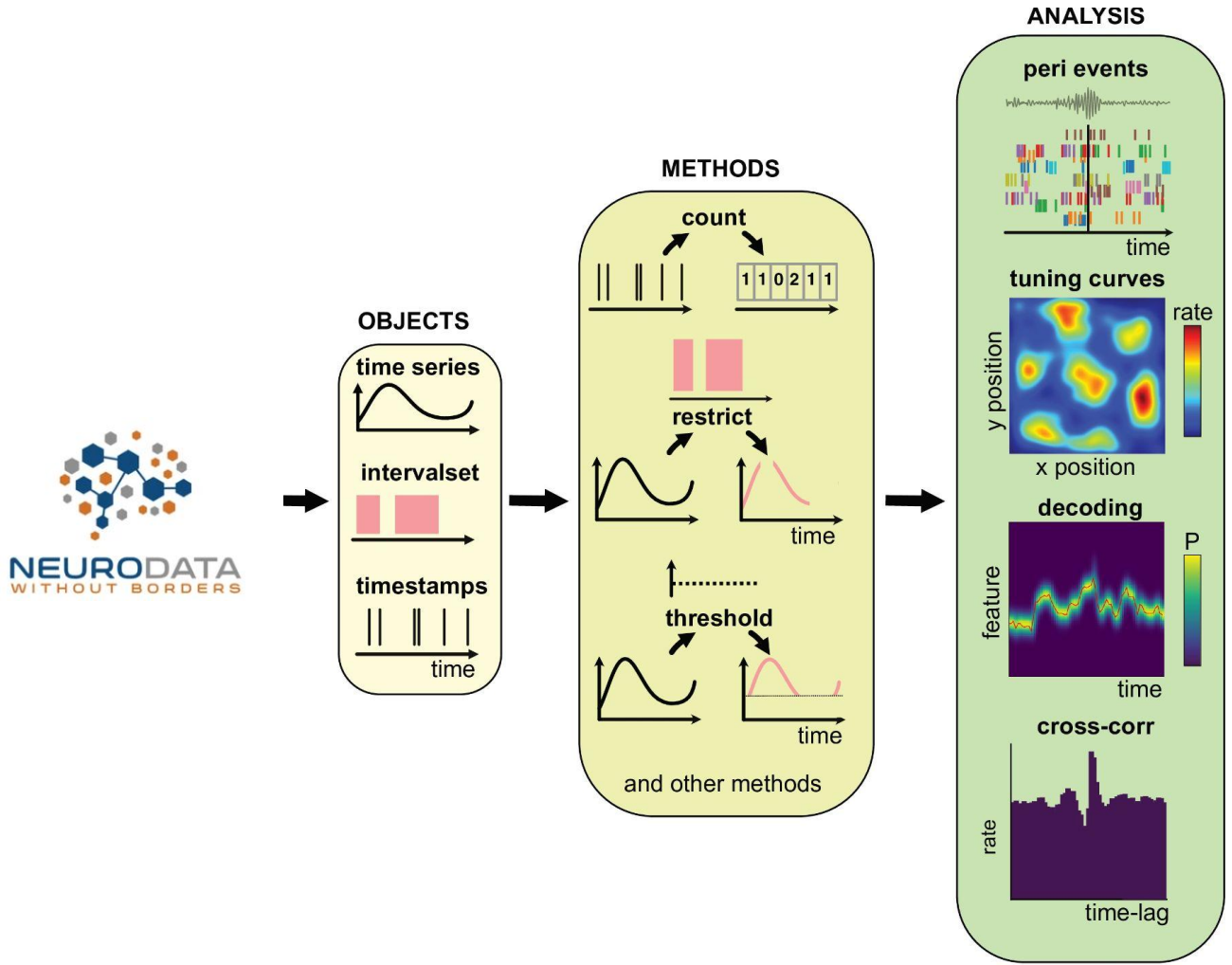
Whole brain spontaneous activity plus NeuroPAL images of semi-restricted worms
DRAFT DANDI:000680 · Contact Kimura, Kotaro · Updated on October 14, 2023 · 1 · 8.7 GB

The difference in electroporation patterns produced by a train of single pulses and a train of paired pulses
0.231013.2226 DANDI:000633 · Contact Silkuniene, Giedre · Updated on October 13, 2023 · 2 · 2.1 GB

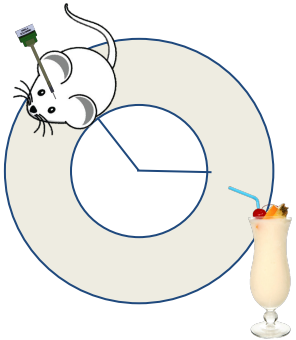
A brainstem circuit for the expression of defensive facial reactions in rat
DRAFT DANDI:000624 · Contact · Updated on October 13, 2023 · 46 · 131.9 GB

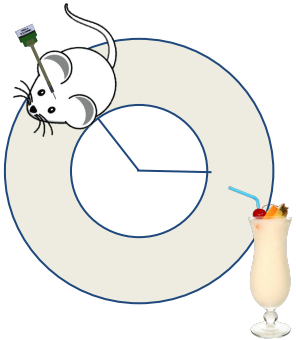


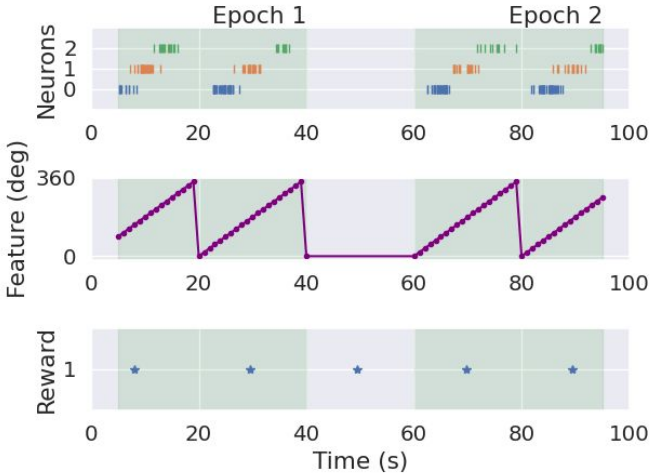

```
In [1]:
1 import pynapple as nap
2
3 nwb = nap.load_file("A2929-200711.nwb")
```

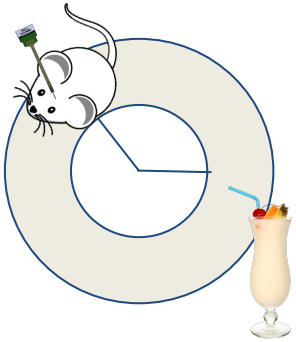


FLATIRON INSTITUTE



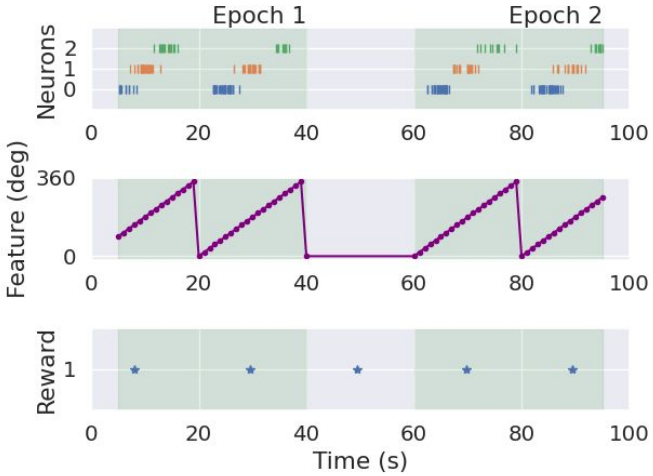



Epoch 1
Epoch 2
Neurons
2
1
0
0
20
40
60
80
100
Feature (deg)
360
0
0
20
40
60
80
100
Reward
1
0
20
40
60
80
100
Time (s)



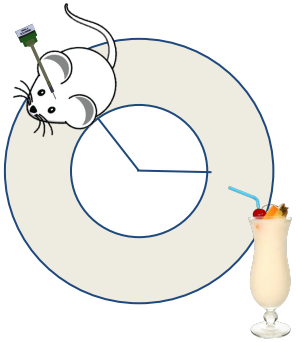
```
In [48]: ep
Out[48]:
   start   end
0    5.0  40.0
1   60.0  95.0
```

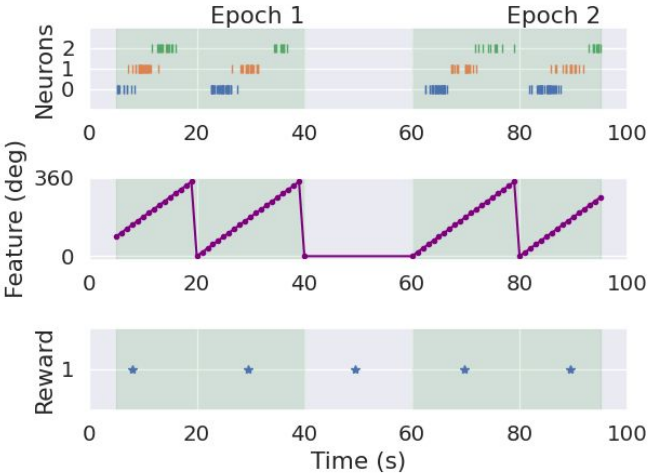
```
In [49]: spikes
Out[49]:
  Index    Freq. (Hz)
-------  ------------
      0          2.07
      1          1.08
      2          0.66
```

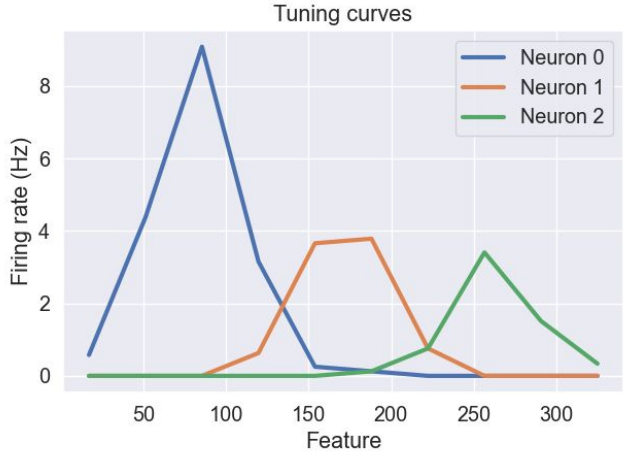



```
In [50]: feature
Out[50]:
Time (s)
0.0       0.0
1.0      18.0
2.0      36.0
3.0      54.0
4.0      72.0
         ...
95.0    270.0
96.0    288.0
97.0    306.0
98.0    324.0
99.0    342.0
Length: 100, dtype: float64
```

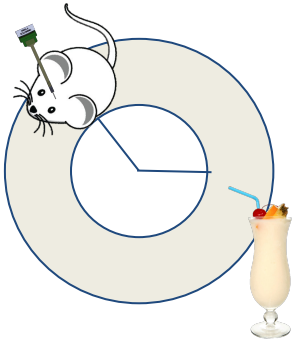
```
In [51]: reward
Out[51]:
Time (s)
6.227442    NaN
15.192588   NaN
22.722009   NaN
23.191450   NaN
25.678101   NaN
```

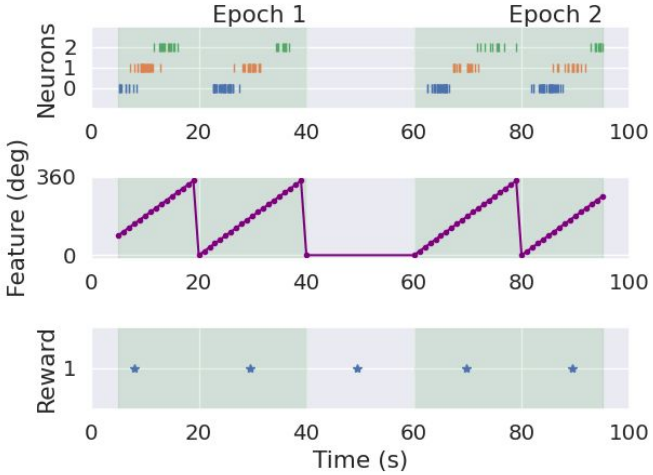


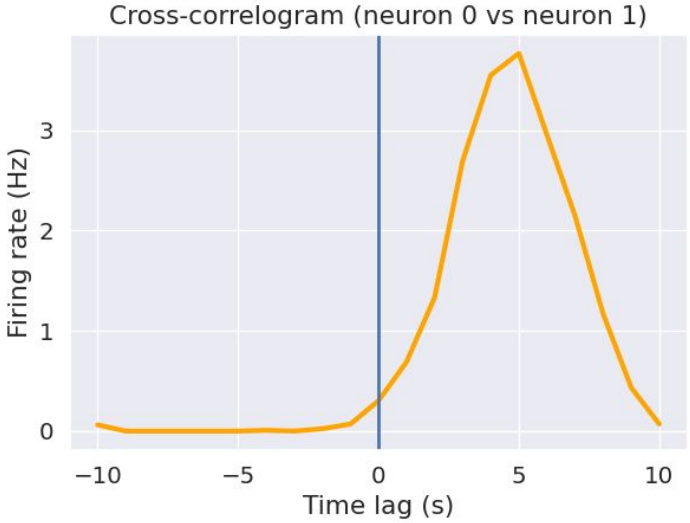




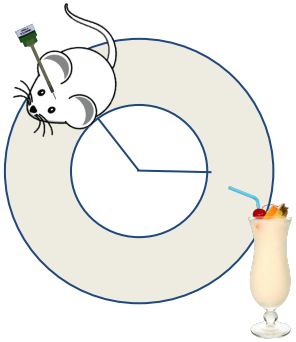

```
tuning_curve = nap.compute_1d_tuning_curves(
    spikes,
    feature,
    nb_bins=10,
    ep=ep)
```

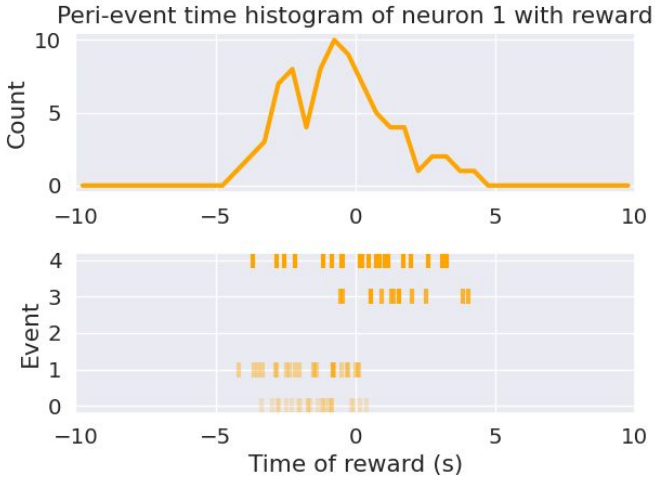


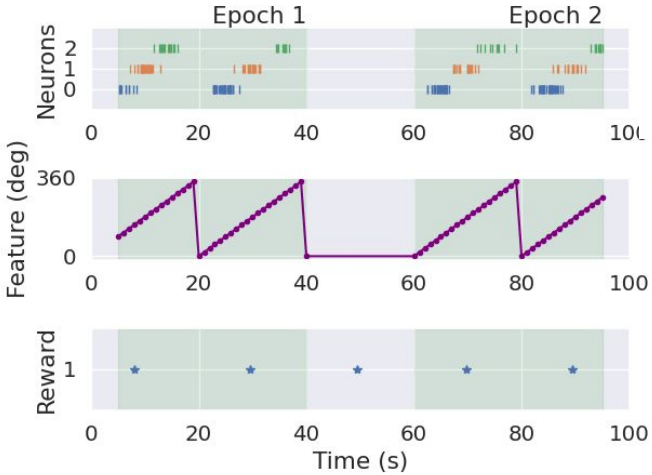





```
cross_corr = nap.compute_crosscorrelogram(
    spikes,
    binsize = 1,
    windowsize = 10,
    ep = ep,
    norm = False)
```

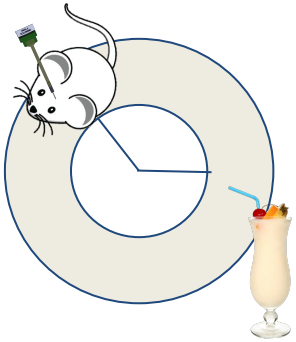


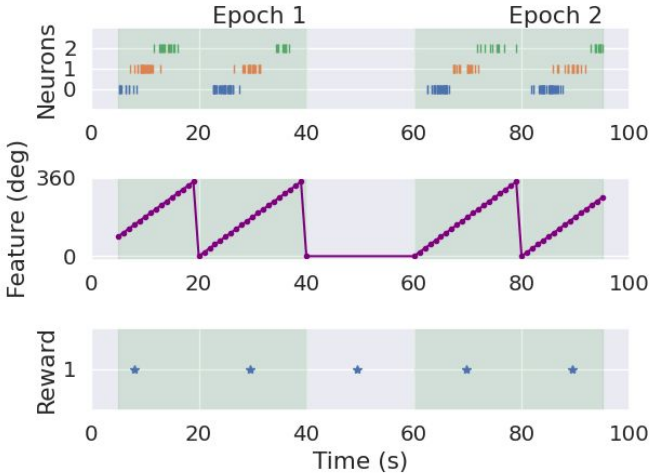


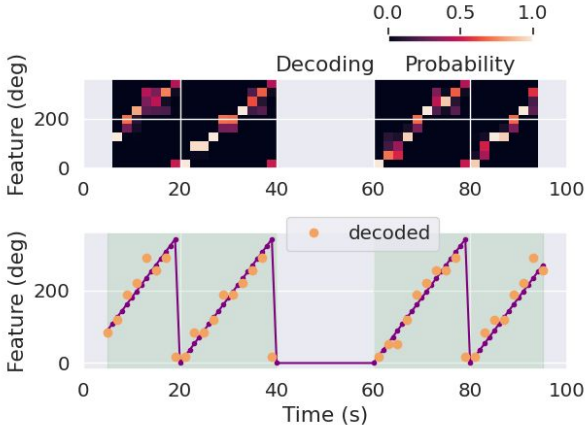



```
In [20]: perievent
Out[20]:
  Index    Freq. (Hz)    ref_times
-------  ------------  -----------
      0          1.15      10.79
      1          1.15      31.1445
      2           nan      49.1539
      3          0.65      67.7832
      4             1      89.4536
```

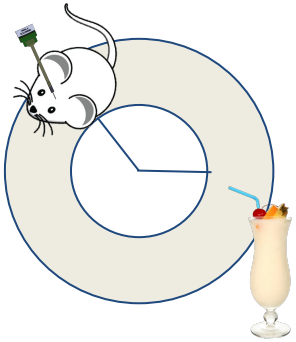
```python
perievent = nap.compute_perievent(
    spikes,
    tref = reward,
    minmax=(-10, 10))
```

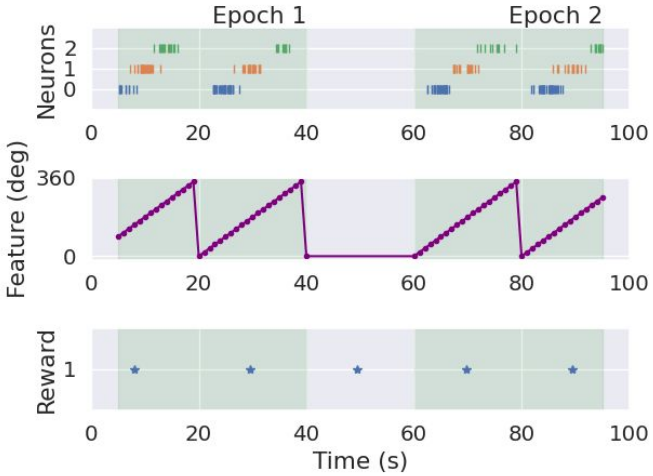


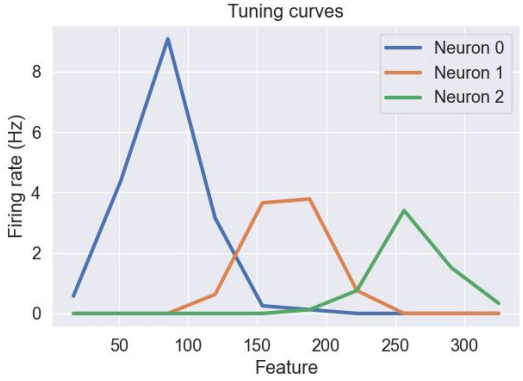


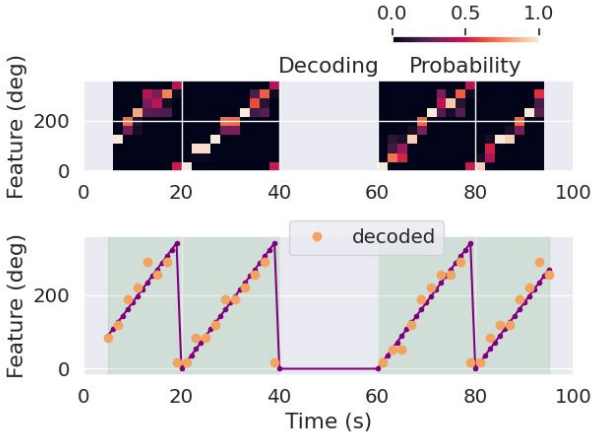


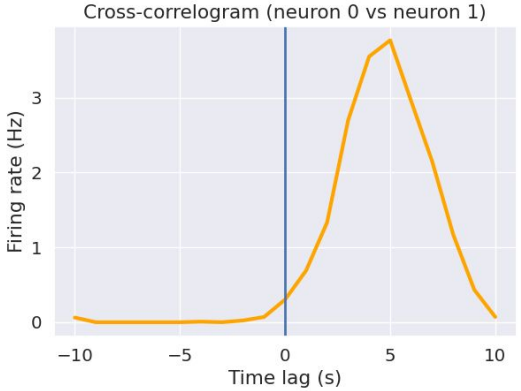

```
decoded,proba = nap.decode_1d(
    tuning_curve,
    spikes,
    ep,
    binsize = 2,
    feature=feature)
```

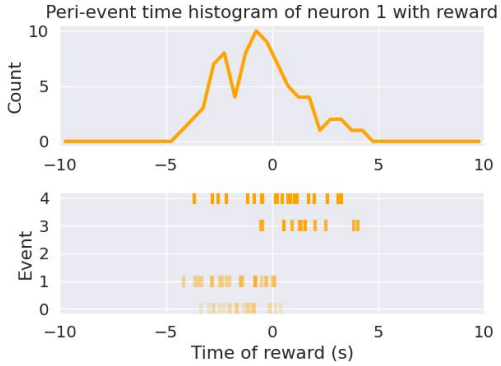


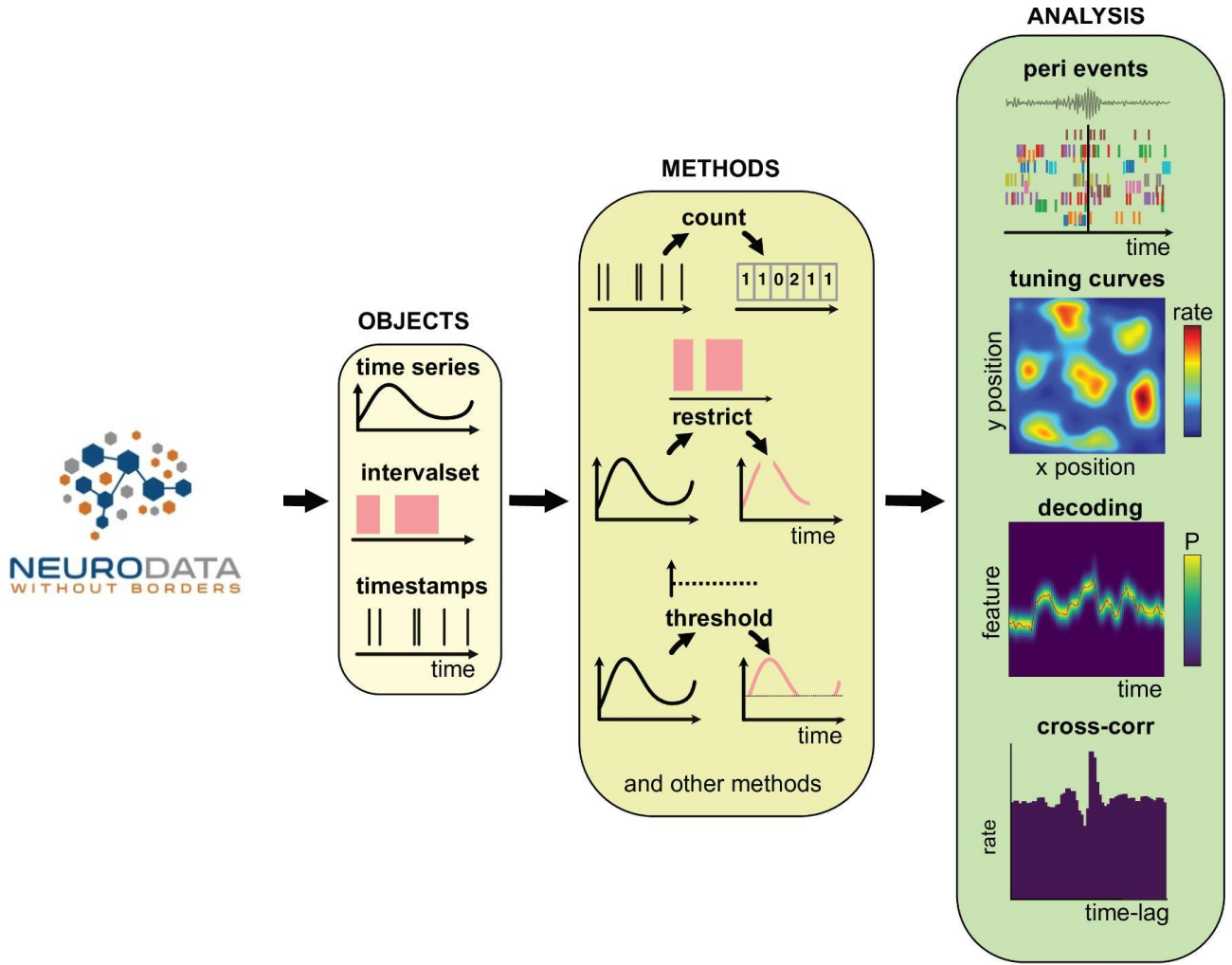

Epoch 1
Epoch 2
Neurons
Feature (deg)
Reward
Time (s)


Tuning curves
Neuron 0
Neuron 1
Neuron 2
Firing rate (Hz)
Feature


Decoding
Probability
Feature (deg)
decoded
Time (s)


Cross-correlogram (neuron 0 vs neuron 1)
Firing rate (Hz)
Time lag (s)


Peri-event time histogram of neuron 1 with reward
Count
Event
Time of reward (s)



FLATIRON INSTITUTE

Future developments


Pynapple

Pynasuite

pynalog Public

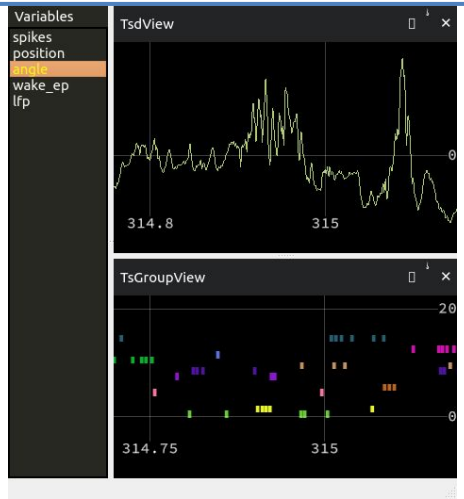
Logging manager for data analysis with pynapple

0 GPL-3.0 0 0

pynaviz Private

Life made easier 🦜🍍🚧

1 GPL-3.0 0 6 1



Pynasuite

Pynapple

pynalog Public

Logging manager for data analysis with pynapple

0 GPL-3.0 0 0

fastplotlib Public

Next-gen fast plotting library running on WGPU using the pygfx rendering engine

Python ☆ 307 31

Kushal Kolar

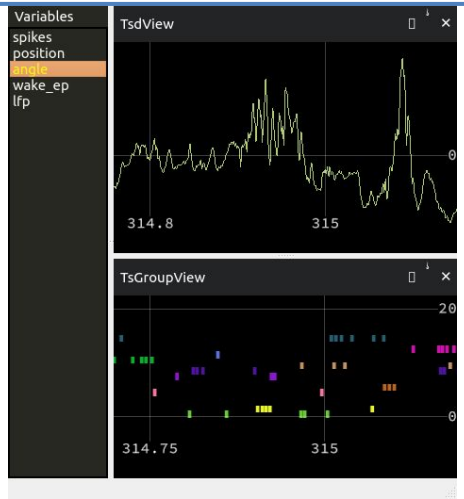
Caitlin Lewis

FLATIRON INSTITUTE

pynaviz Private

Life made easier 🦜🍍🚧

☆ 1 GPL-3.0 0 6 1



Pynasuite

Pynapple

pynalog Public

Logging manager for data analysis with pynapple

☆ 0 GPL-3.0 0 0

fastplotlib Public

Next-gen fast plotting library running on WGPU using the pygfx rendering engine

Python ☆ 307 31

Kushal Kolar

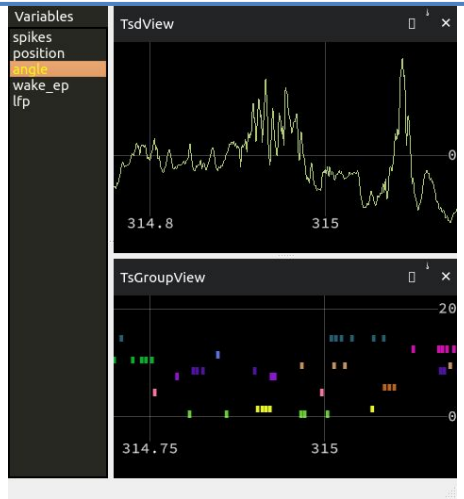
Caitlin Lewis

FLATIRON INSTITUTE

pynaviz Private

Life made easier 🦜🍍🚧

☆ 1 GPL-3.0 0 6 1



Pynasuite

Pynapple

pynajax Public

Jax backend for pynapple

Python ☆ 5 MIT

pynalog Public

Logging manager for data analysis with pynapple

☆ 0 GPL-3.0 0 0


pynajax
Public
Jax backend for pynapple
Python
5
MIT

```python
import pynapple as nap
import numpy as np

tsd = nap.Tsd(t=np.arange(100), d=np.random.randn(100))

tsd.convolve(np.ones(11))
```

FLATIRON INSTITUTE

pynajax Public

Jax backend for pynapple

Python ☆ 5 MIT

```python
import pynapple as nap
import numpy as np

tsd = nap.Tsd(t=np.arange(100), d=np.random.randn(100))

tsd.convolve(np.ones(11))
```

FLATIRON INSTITUTE

pynajax Public

Jax backend for pynapple

Python ☆ 5 MIT

$ pip install pynajax

```python
import pynapple as nap
import numpy as np

nap.nap_config.set_backend("jax")

tsd = nap.Tsd(t=np.arange(100), d=np.random.randn(100))

tsd.convolve(np.ones(11))
```

```python
import pynapple as nap
import numpy as np

tsd = nap.Tsd(t=np.arange(100), d=np.random.randn(100))

tsd.convolve(np.ones(11))
```

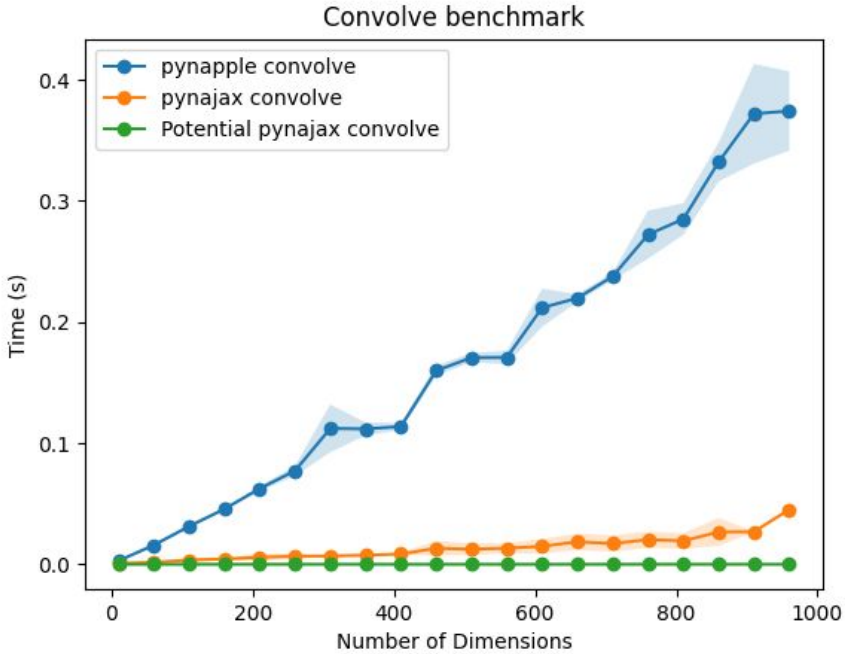
$ pip install pynajax (soon)

```python
import pynapple as nap
import numpy as np

nap.nap_config.set_backend("jax")

tsd = nap.Tsd(t=np.arange(100), d=np.random.randn(100))

tsd.convolve(np.ones(11))
```



Pynapple

$ pip install pynapple

https://twitter.com/thepynapple

pynapple-org.slack.com

FLATIRON INSTITUTE

Adrien Peyrache

Dhruv Mehrotra

Sofia Skromne Carrasco

Daniel Levenstein

Selen Calgin

Sara Mahallati

Alex Efremov

Henry Denny

Gilberto R Vite

Francesco Battaglia

Edoardo Balzani

Luigi Petrucco

David Spalla

Luke Sjulson

Time to code again


FLATIRON
INSTITUTE