Building Data Pipelines for Neurophysiology

Thinh Nguyen
Dimitri Yatsenko

ODIN - June 11, 2025

Overview

1. **Intro - data pipelines in science operations**
    a. **DataJoint Elements**
2. **Building scientific pipelines - interactive tutorials**
    a. Pipeline for electrophysiology
    b. Pipeline for motion sequencing (MoSeq) - demo
    c. General DataJoint tutorials
3. **Pipeline Operations**

Open-Source Data Pipelines for Neurophysiology

DataJoint Elements

DataJoint Elements

Modular pipelines

Common language

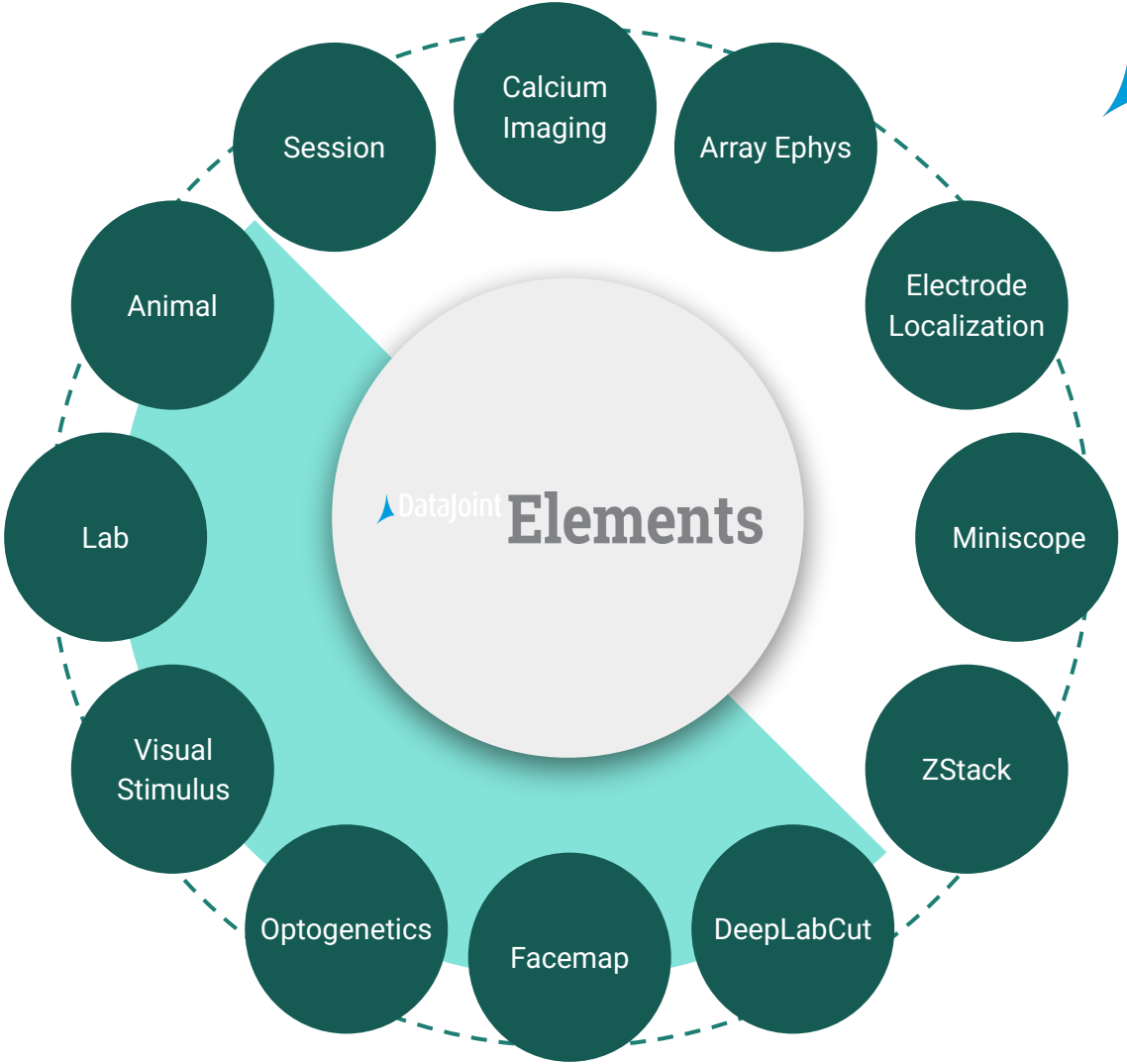
A flexible open standard for scientists to define all aspects of a study — so it can be ***understood***, ***validated***, ***shared***, and ***automed***.

Standardized modules

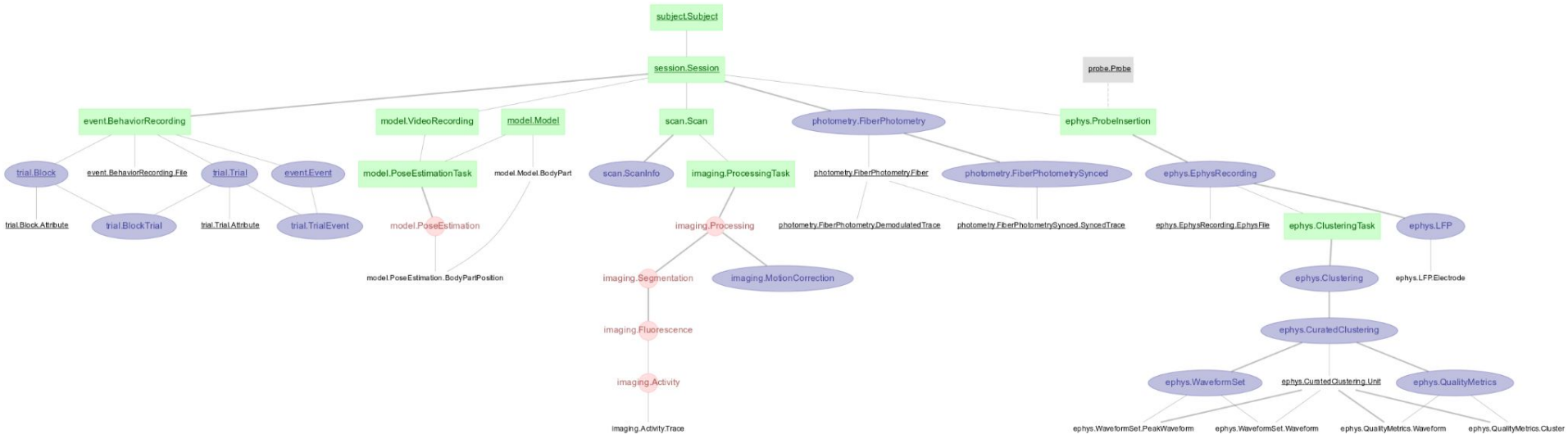
Validated open-source modules. Integration, interfaces, customization

Uniform processes

Navigation, automation, queries, visualization, sharing, publishing



Multimodal data pipeline



Operant Behavior (trials, events)	Pose Estimation (DLC)	Calcium Imaging (Suite2p)	Fiber Photometry	Ephys (SpikeGLX, Kilosort)

Open-source tools and informatics resources

Community, licensing, governance.

open ephys

NEURODATA WITHOUT BORDERS

DANDI

Suite2p

Facemap

DeepLabCut: a software package for animal pose estimation

ALLEN INSTITUTE

Allen Brain Observatory

CaImAn

Miniscope

UCLA

neuro PIXELS

Kilosort

MoSeq

bonsai

VISUAL REACTIVE PROGRAMMING

ScanImage

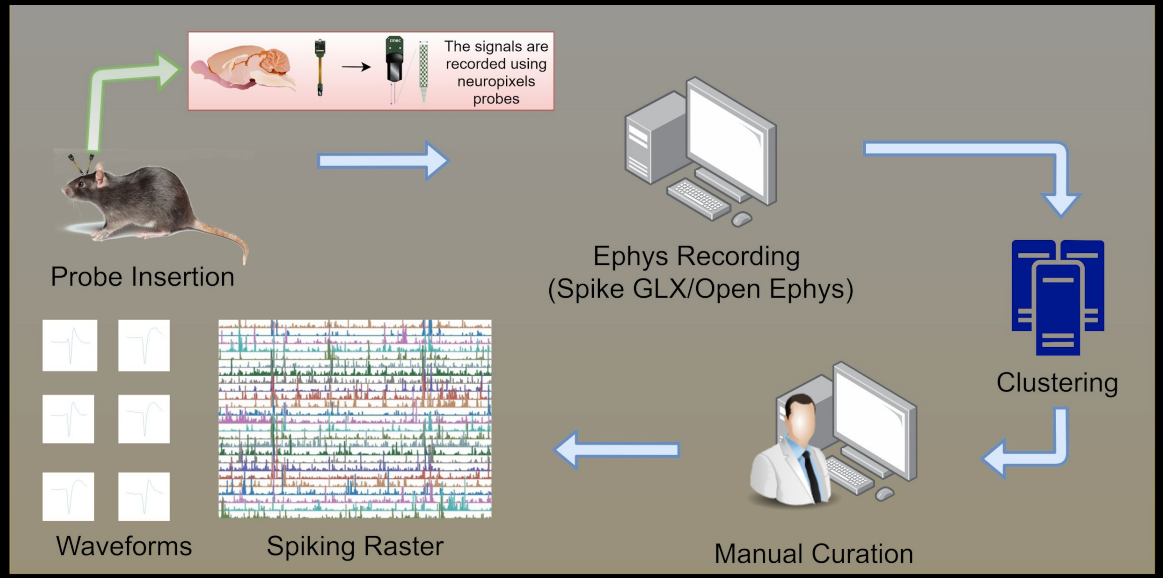
SpikeInterface

incf International Neuroinformatics Coordinating Facility

Element Array Electrophysiology

https://github.com/datajoint/element-array-ephys

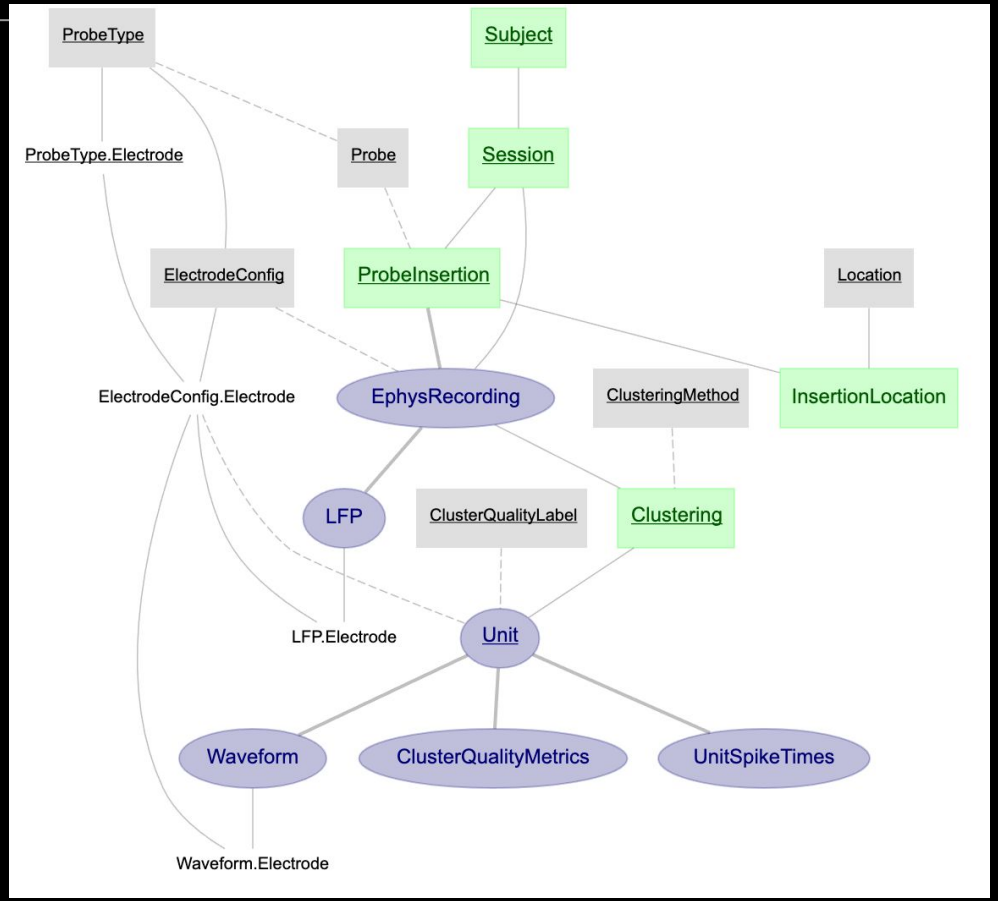
Acquisition hardware	Neuropixels 1.0, 2.0, 3A
Acquisition software	SpikeGLX, OpenEphys
Preprocessing	Kilosort 2.5 Kilosort 3.0 pykilosort
Analysis	Events & trias NWB export
Projects	Princeton U19 Mesoscale Activity Project International Brain Lab Moser Group Loren Frank Lab Columbia U19 Allen Institute - Mindscope



Element Array Electrophysiology

https://github.com/datajoint/element-array-ephys

Acquisition hardware	Neuropixels 1.0, 2.0, 3A
Acquisition software	SpikeGLX, OpenEphys
Preprocessing	Kilosort 2.5 Kilosort 3.0 pykilosort
Analysis	Events & trias NWB export
Projects	Princeton U19 Mesoscale Activity Project International Brain Lab Moser Group Loren Frank Lab Columbia U19 Allen Institute - Mindscope

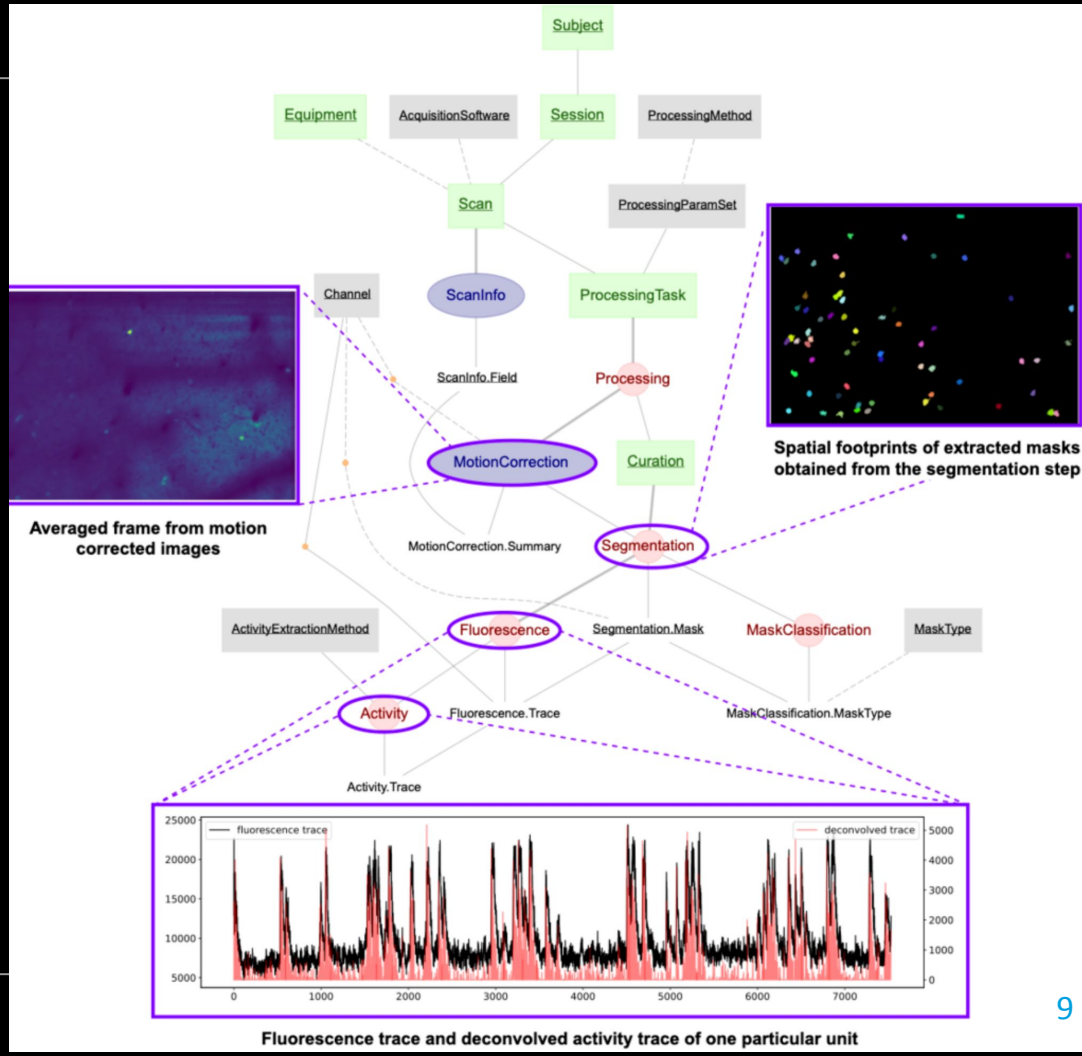


Element Calcium Imaging

Calcium Imaging

https://github.com/datajoint/element-calcium-imaging

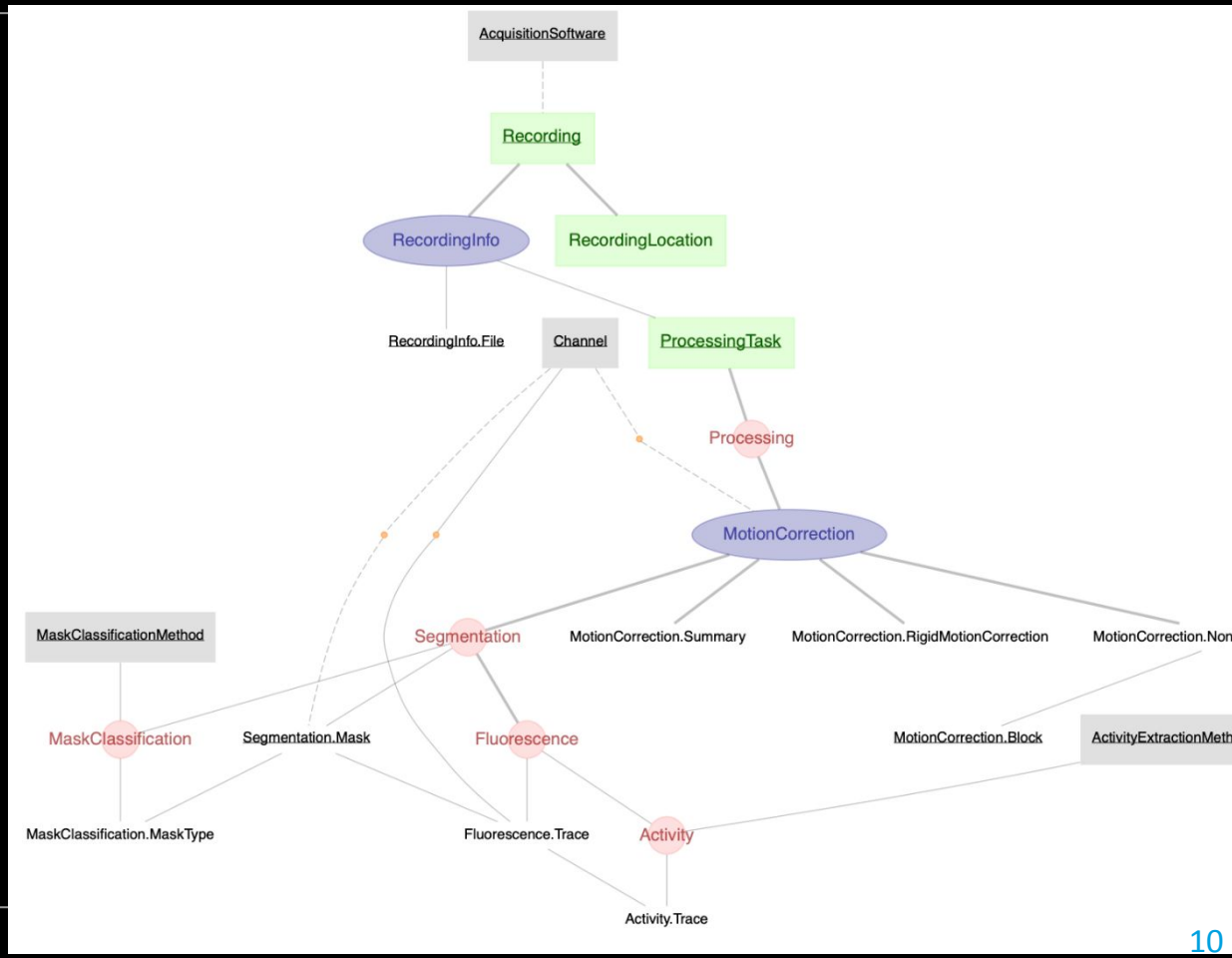
Acquisition	ScanImage Scanbox Nikon NIS Elements Prairie View
Preprocessing	Suite2p, CaImAn EXTRACT
Analysis	Events & trials
Projects	Princeton U19, Lu Lab @ Indiana U, Moser Group, Tobias Rose Lab



Miniscope Calcium Imaging

https://github.com/datajoint/element-miniscope

Acquisition hardware	UCLA Miniscope
Acquisition software	Miniscope-DAQ-V3, Miniscope-DAQ-V4
Preprocessing	CaImAn
Analysis	Events & trials
Projects	Columbia U19



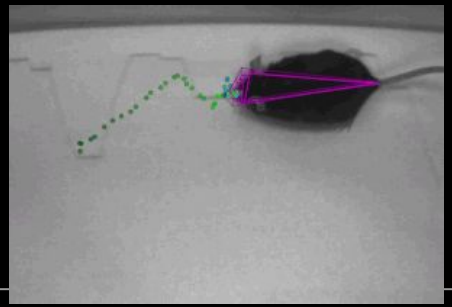
DataJoint

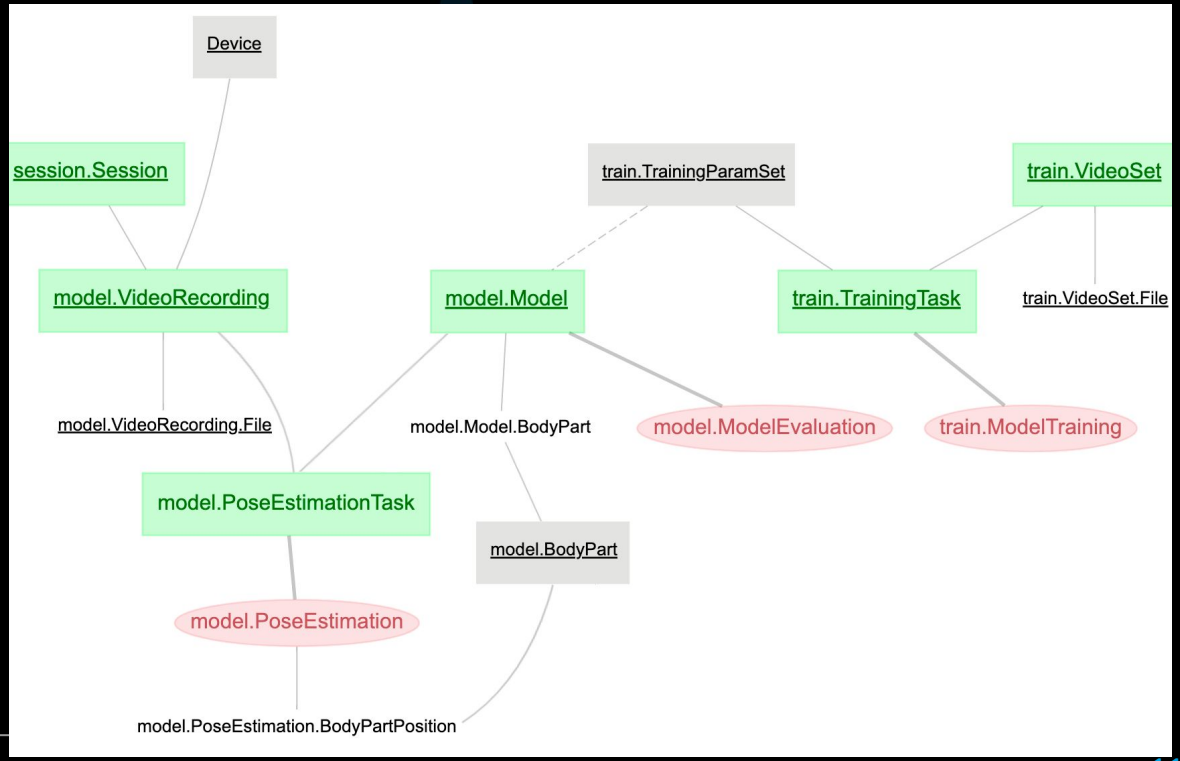
Element DeepLabCut

https://github.com/DeepLabCut/DeepLabCut

Analysis
- Video Management
- Model Training
- Pose Estimation

Projects
- Mesoscale Activity Project
- Mathis Lab @ EPFL
- Lu Lab @ Indiana U
- Rose Lab @ Bonn U
- Moser Group





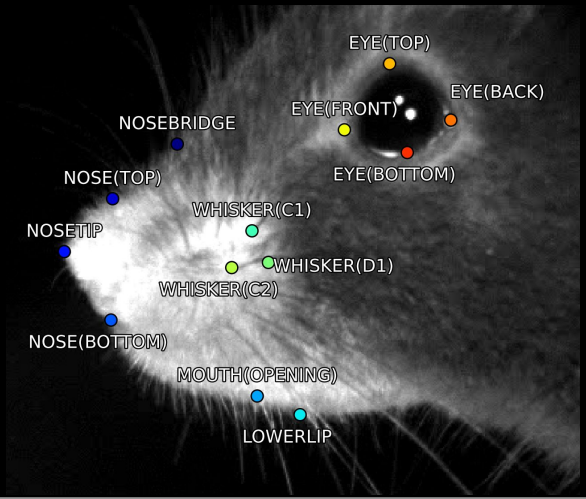
DataJoint

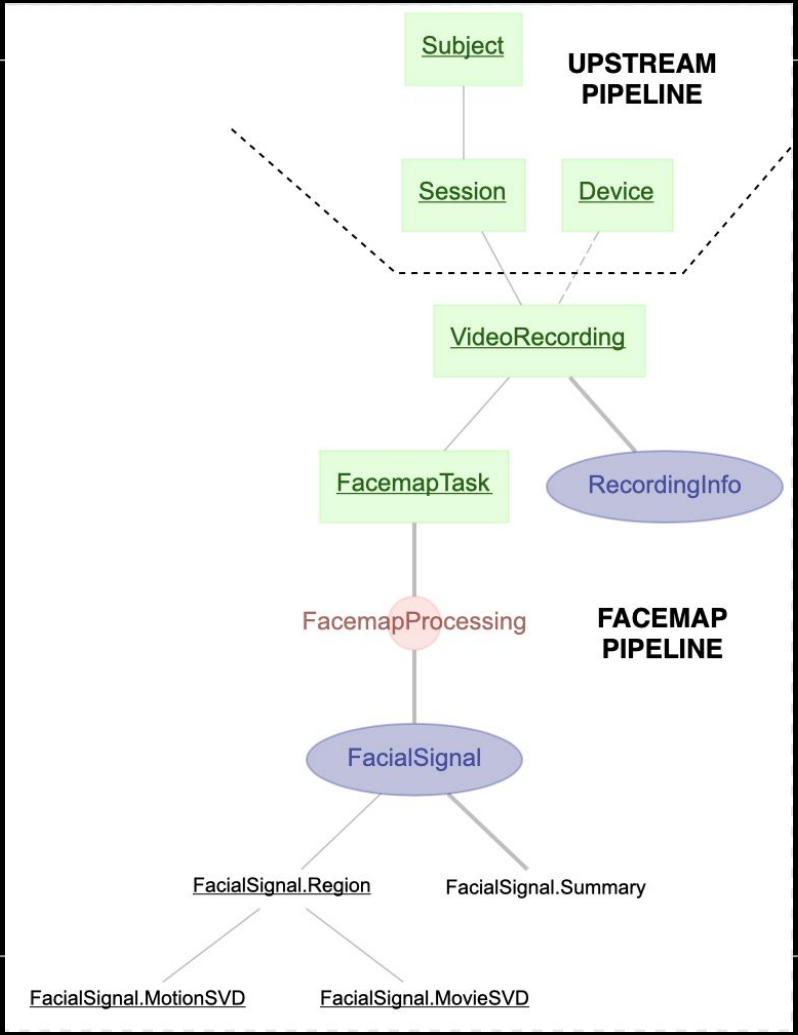
Facemap Element

https://github.com/datajoint/element-facemap

Analysis Facemap

Projects Lu Lab @ Indiana U





Interactive Tutorial

DataJoint

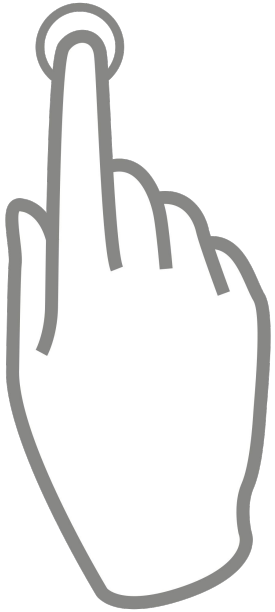
A New Operating System for Science

element-array-ephys

https://github.com/datajoint/element-array-ephys

- Create a fork

- Start Codespace (~10min)



Neuropixels probe

https://www.neuropixels.org/

https://open-ephys.org/neuropixels

nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄ Subscribe

nature > letters > article

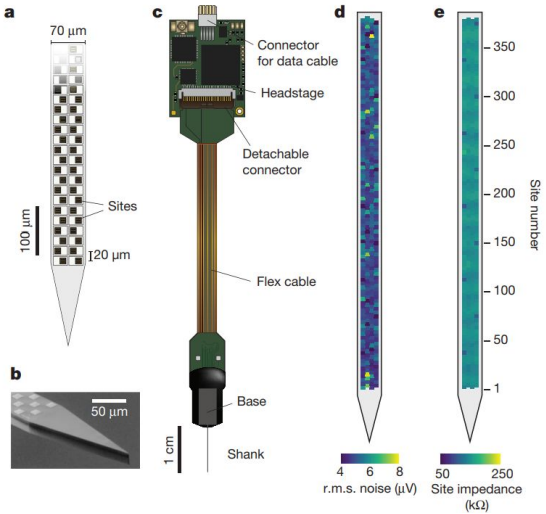
Letter | Published: 09 November 2017

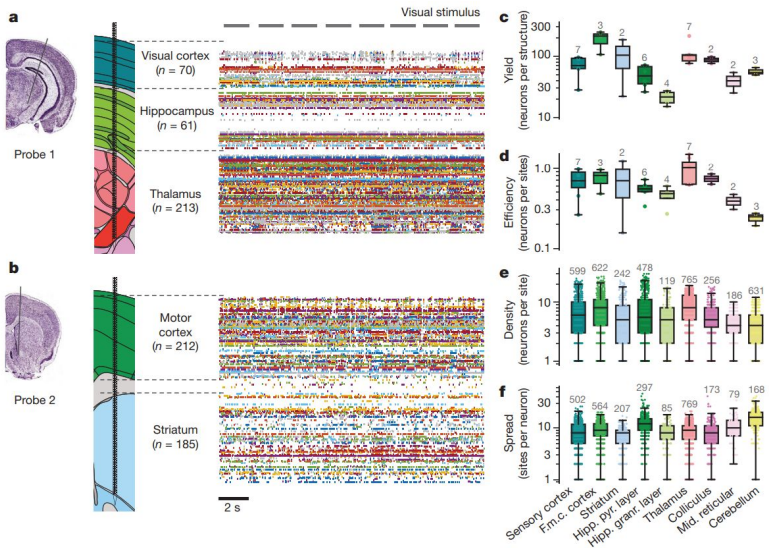
Fully integrated silicon probes for high-density recording of neural activity

James J. Jun, Nicholas A. Steinmetz, Joshua H. Siegle, Daniel J. Denman, Marius Bauza, Brian Barbarits, Albert K. Lee, Costas A. Anastassiou, Alexandru Andrei, Çağatay Aydın, Mladen Barbic, Timothy J. Blanche, Vincent Bonin, João Couto, Barundeb Dutta, Sergey L. Gratiy, Diego A. Gutnisky, Michael Häusser, Bill Karsh, Peter Ledochowitsch, Carolina Mora Lopez, Catalin Mitelut, Silke Musa, Michael Okun, ... Timothy D. Harris ✉ + Show authors

Nature **551**, 232–236 (2017) | Cite this article

https://www.nature.com/articles/nature24636





Spike Sorting

"The procedure of 'spike sorting' consists of various processing steps to extract single-neuron spiking activity from extracellular recordings"

IOP Publishing Prog. Biomed. Eng. 4 (2022) 022005 https://doi.org/10.1088/2516-1091/ac6b96

Progress in Biomedical Engineering

TOPICAL REVIEW

Spike sorting: new trends and challenges of the era of high-density probes

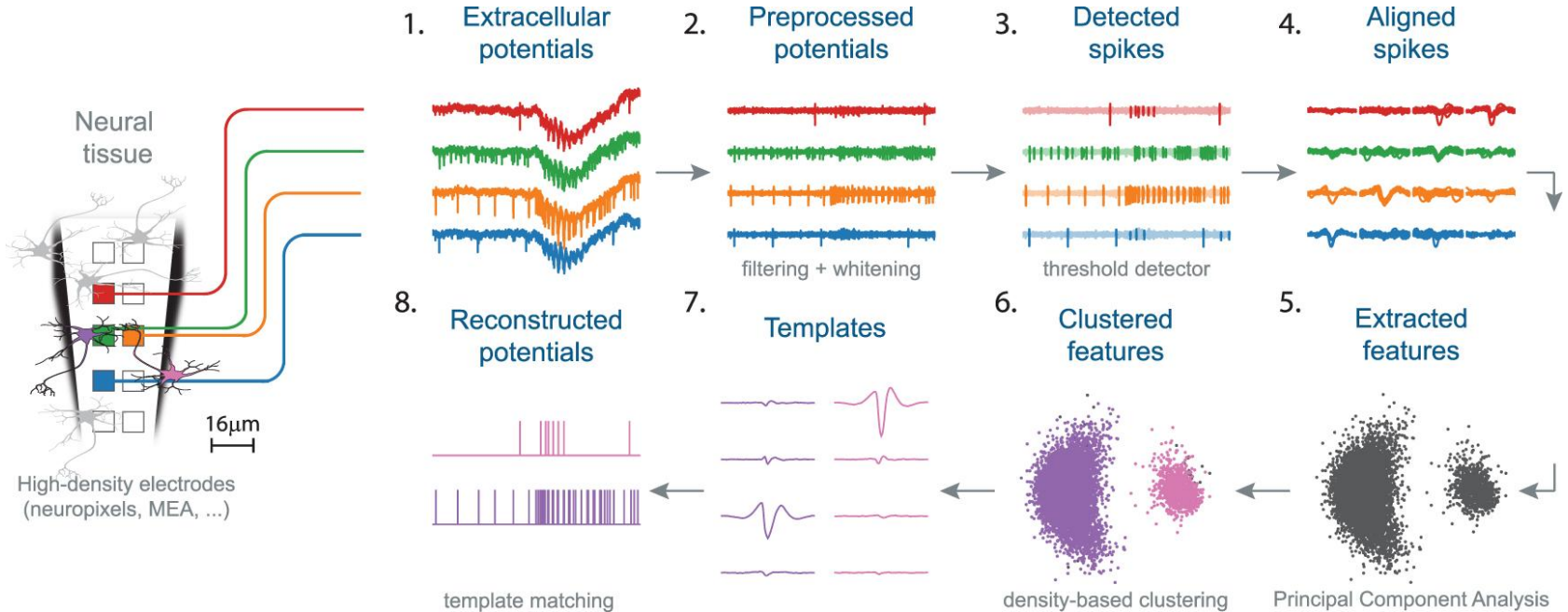
OPEN ACCESS

RECEIVED 7 January 2022
REVISED 29 March 2022
ACCEPTED FOR PUBLICATION 28 April 2022
PUBLISHED 17 May 2022

Alessio P Buccino[1,*], Samuel Garcia[2] and Pierre Yger[3]

[1] Department of Biosystems Science and Engineering, ETH Zurich, Switzerland
[2] Centre de Recherche en Neuroscience de Lyon, CNRS, Lyon, France
[3] Sorbonne Université, INSERM, CNRS, Institut de la Vision, F-75012 Paris, France
[*] Author to whom any correspondence should be addressed.

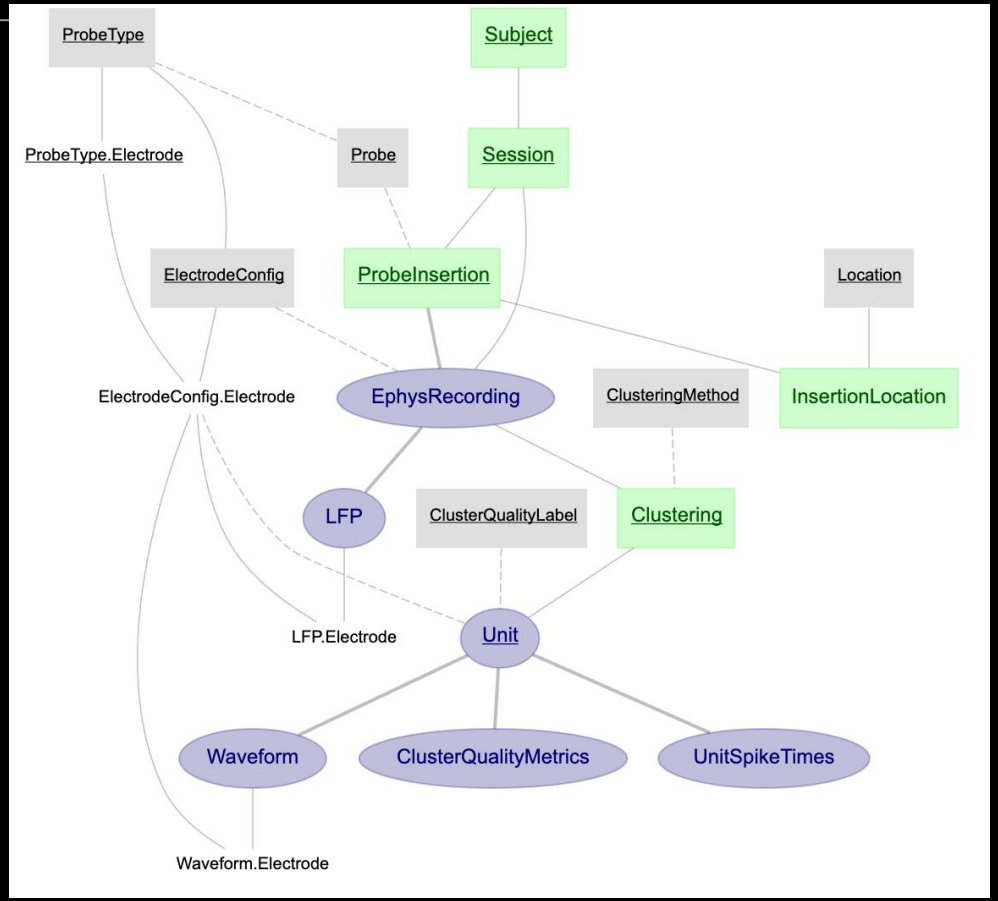
E-mail: alessio.buccino@bsse.ethz.ch



Array Electrophysiology

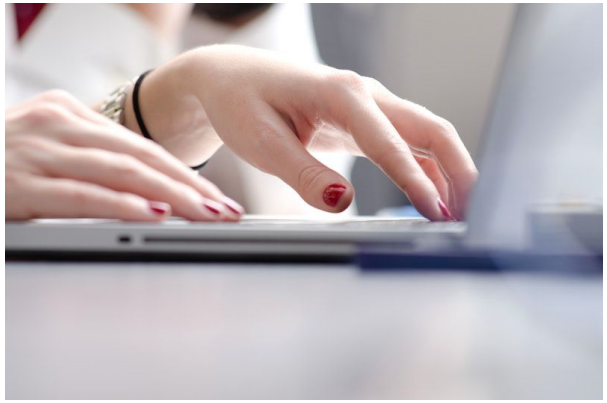
https://github.com/datajoint/element-array-ephys

Probe	Neuropixels 1.0, 2.0, 3A, ...
Acquisition software	SpikeGLX, OpenEphys
Preprocessing	kilosort SpikeInterface LFP extraction
Analysis	Events & trias NWB export
Projects	Princeton U19 Mesoscale Activity Project International Brain Lab Moser Group Loren Frank Lab Columbia U19 Allen Institute - Mindscope



Hands-on tutorial

Today's goal is to give a high level description of tables and dependencies, and a quick tutorial to learn how to flow your data through the element-array-ephys pipeline.



element-moseq

nature neuroscience

Resource | Published: 21 September 2020

Revealing the structure of pharmacobehavioral space through motion sequencing

Alexander B. Wiltschko, Tatsuya Tsukahara, Ayman Zeine, Rockwell Anyoha, Winthrop F. Gillis, Jeffrey E. Markowitz, Ralph E. Peterson, Jesse Katon, Matthew J. Johnson & Sandeep Robert Datta

Nature Neuroscience **23**, 1433–1443 (2020) | Cite this article

Abstract

Understanding how genes, drugs and neural circuits influence behavior requires the ability to effectively organize information about similarities and differences within complex behavioral datasets. Motion Sequencing (MoSeq) is an ethologically inspired behavioral analysis method that identifies modular components of three-dimensional mouse body language called 'syllables'. Here, we show that MoSeq effectively parses behavioral differences and captures

nature methods

Article | Open access | Published: 12 July 2024

Keypoint-MoSeq: parsing behavior by linking point tracking to pose dynamics

Caleb Weinreb, Jonah E. Pearl, Sherry Lin, Mohammed Abdal Monium Osman, Libby Zhang, Sidharth Annapragada, Eli Conlin, Red Hoffmann, Sofia Makowska, Winthrop F. Gillis, Maya Jay, Shaokai Ye, Alexander Mathis, Mackenzie W. Mathis, Talmo Pereira, Scott W. Linderman & Sandeep Robert Datta

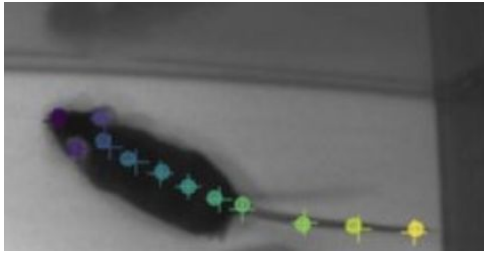
Nature Methods **21**, 1329–1339 (2024) | Cite this article

Abstract

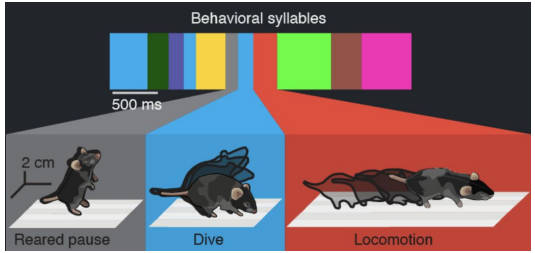
Keypoint tracking algorithms can flexibly quantify animal movement from videos obtained in a wide variety of settings. However, it remains unclear how to parse continuous keypoint data into discrete actions. This challenge is particularly acute because keypoint data are susceptible to high-frequency jitter that clustering algorithms can mistake for transitions between actions. Here we present keypoint-MoSeq, a machine learning-based platform for identifying behavioral modules ('syllables') from keypoint data without human supervision.

Keypoint-MoSeq

MoSeq applies unsupervised machine learning algorithms to segment continuous mouse behavior into interpretable behavioral motifs (like rears, turns and pauses) called “syllables” from pose estimation data.

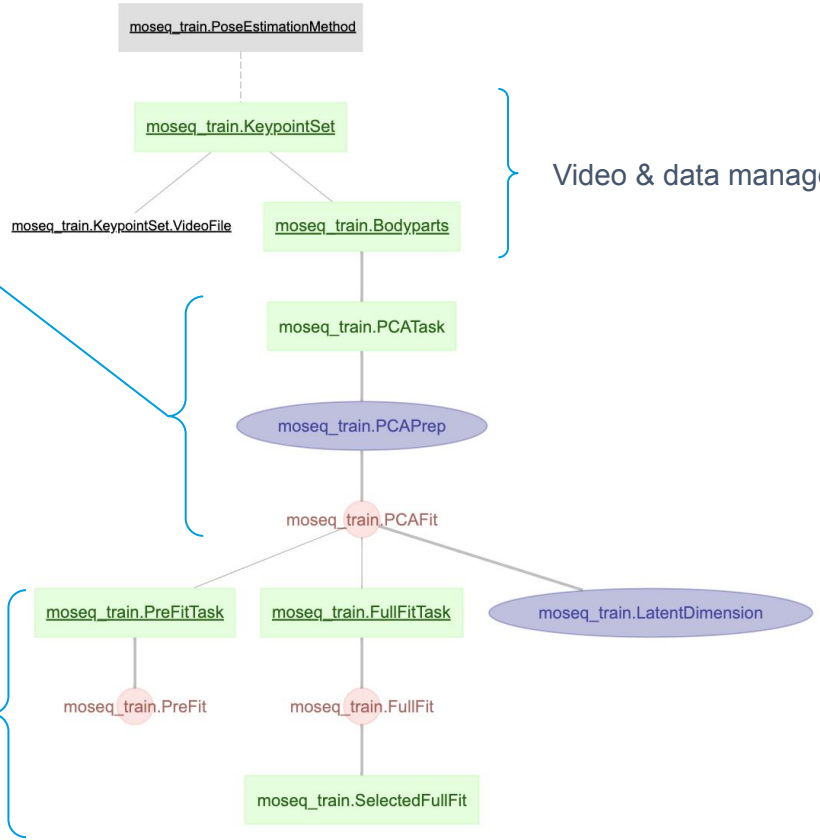


Analysis
1. Model Training
2. Model Inference



MoSeq Model Training pipeline

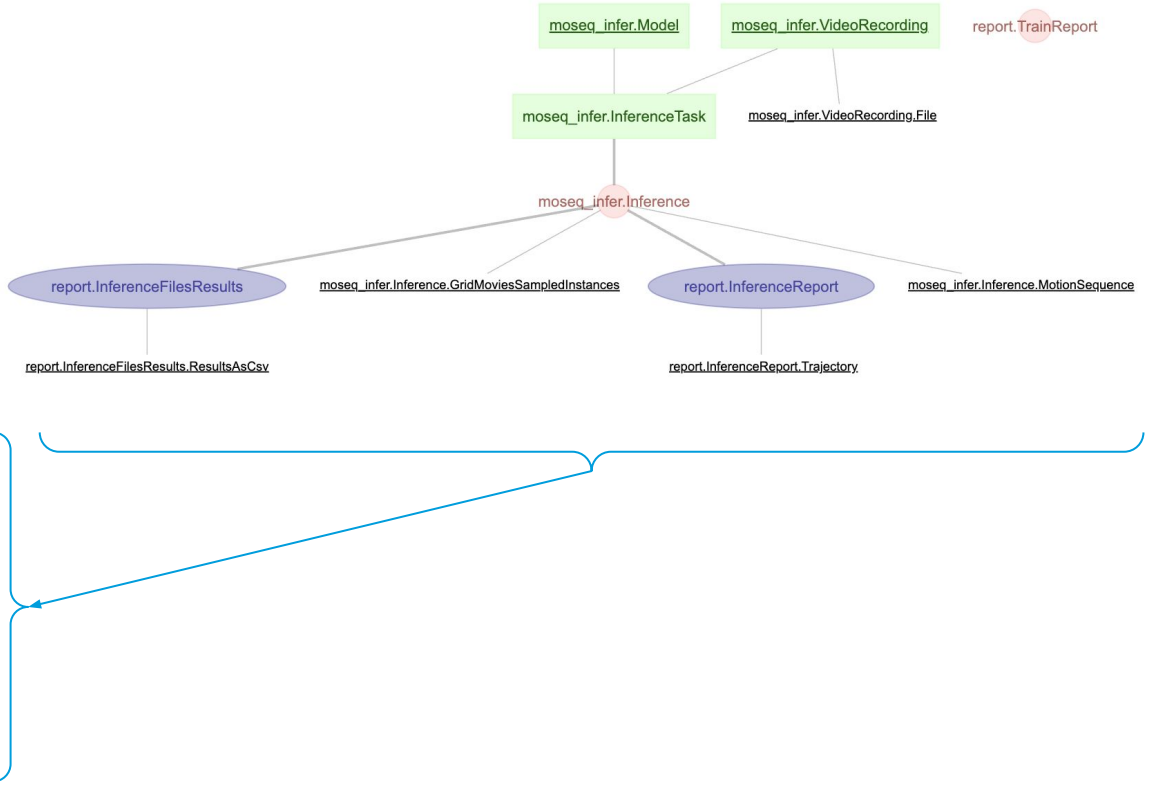
- Project setup
  - Edit the config file
  - Load data
  - Fit PCA
- Model fitting
  - Setting kappa
  - Initialization
  - Fitting an AR-HMM
  - Fitting the full model
  - Sort syllables by frequency
  - Extract model results
    - [Optional] Save results to csv
  - Apply to new data
- Visualization
  - Trajectory plots
  - Grid movies
  - Syllable Dendrogram



Video & data management

MoSeq Model Inference pipeline

- Project setup
  - Edit the config file
  - Load data
  - Fit PCA
- Model fitting
  - Setting kappa
  - Initialization
  - Fitting an AR-HMM
  - Fitting the full model
  - Sort syllables by frequency
  - Extract model results
    - [Optional] Save results to csv
  - Apply to new data
- Visualization
  - Trajectory plots
  - Grid movies
  - Syllable Dendrogram

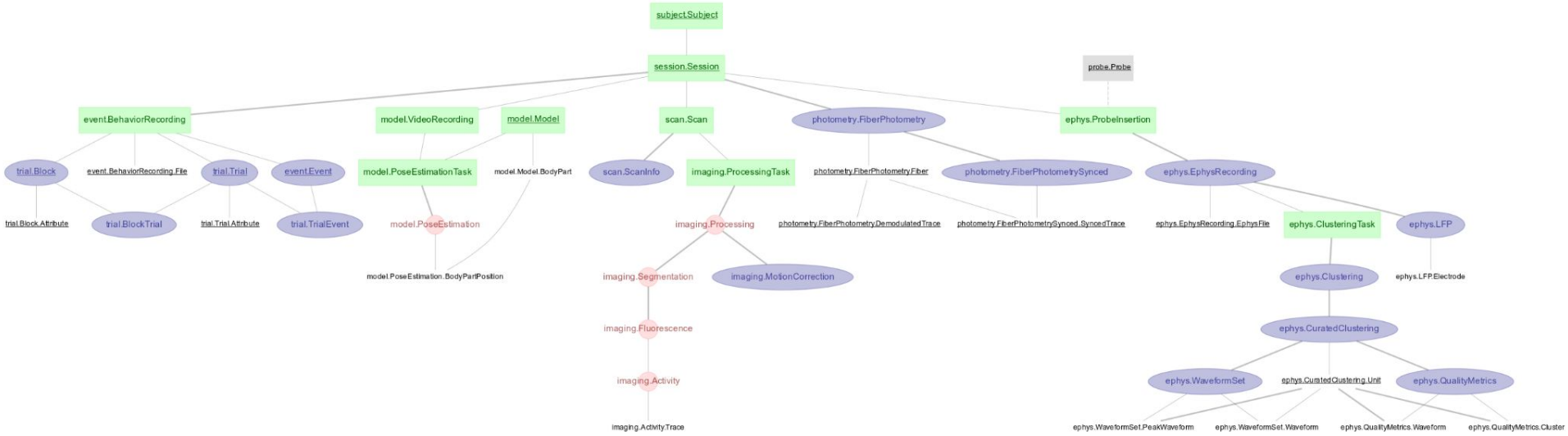


Pipeline Operations

DataJoint

A New Operating System for Science

The computational data pipeline - in DataJoint



Operant Behavior (trials, events)

Pose Estimation (DLC)

Ephys (SpikeGLX, Kilosort)

Fiber Photometry

Calcium Imaging (Suite2p)

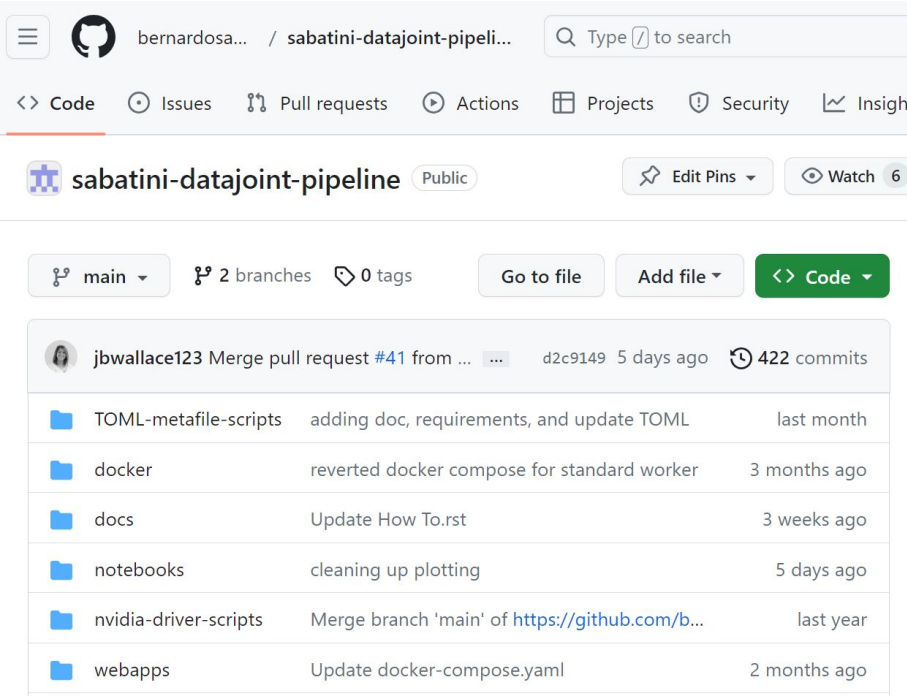
Where is your code?

DataJoint

A New Operating System for Science

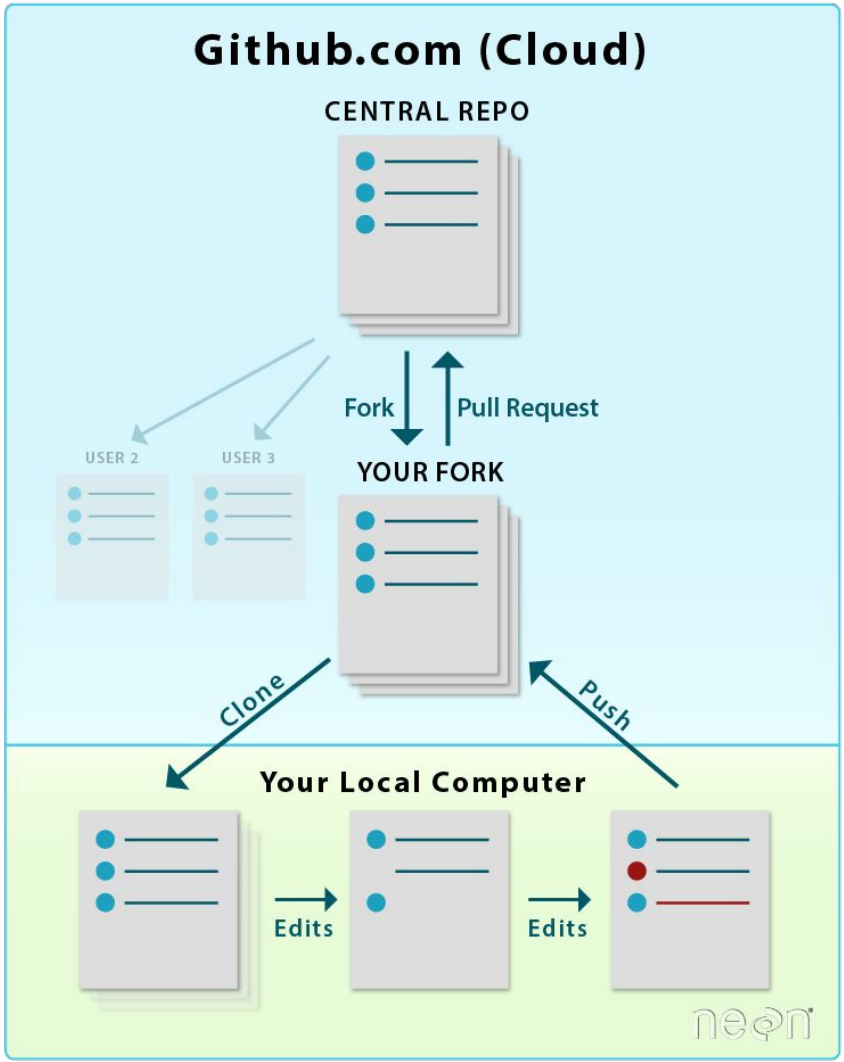
Where is your code?

- The code management process for this project uses Git and GitHub
- Github repositories https://github.com/bernardosabatinilab/sabatini-datajoint-pipeline



How to collaborate with the code

- Depending on the guideline for each individual lab
- We recommend the “GitHub standard fork and pull request workflow”



https://zenodo.org/badge/latestdoi/186890040

You wrote the code!
How about organizing your data?

DataJoint
A New Operating System for Science

Where is your data?

- Raw data files are
  - located on-premise storage
  - uploaded and stored as files on the cloud - Amazon Web Service (AWS)
- Processed data are stored directly with the the database (MySQL) - located on the cloud - AWS
- Data access are granted and managed via database credentials
- With valid credentials, ones can access the data via datajoint-python or datajoint-matlab

How/where to run your code?

DataJoint

A New Operating System for Science

How are the computations orchestrated and executed?

- Execution of the computation steps in the pipeline can be done in local environment or on the cloud
- DataJoint keeps track of executed jobs and remaining jobs
- Resources to manage
  - Containerized environments - Docker
  - Compute resources - labs’ workstation, on-prem servers/HPC, cloud VMs

How are the computations orchestrated and executed?

Locally on your laptop

- codebase installed
- script to run `.populate()

Orchestrated across multiple computers

- codebase installed
- script to run `.populate()
- containerized environment (Docker)

Orchestrated with cloud computing

- codebase installed
- script to run `.populate()
- containerized environment (Docker)
- Cloud resource provider (AWS, Azure, GCP)
- DevOps resources (e.g. Kubernetes, Terraform, etc.)

To put it all together

Code
- GitHub
- Pipeline code
- Docker

File storage
- File system
- Network drive
- Cloud object storage (AWS S3)

DataJoint

Compute
- Personal laptop
- Lab's workstations
- HPC
- Cloud computing

Relational Database
- Local MySQL server
- AWS RDS
- Percona cluster