



**Kempner**  
INSTITUTE



**HARVARD**  
UNIVERSITY

# Spike Sorting on an HPC Cluster

March 4, 2025

# Objectives

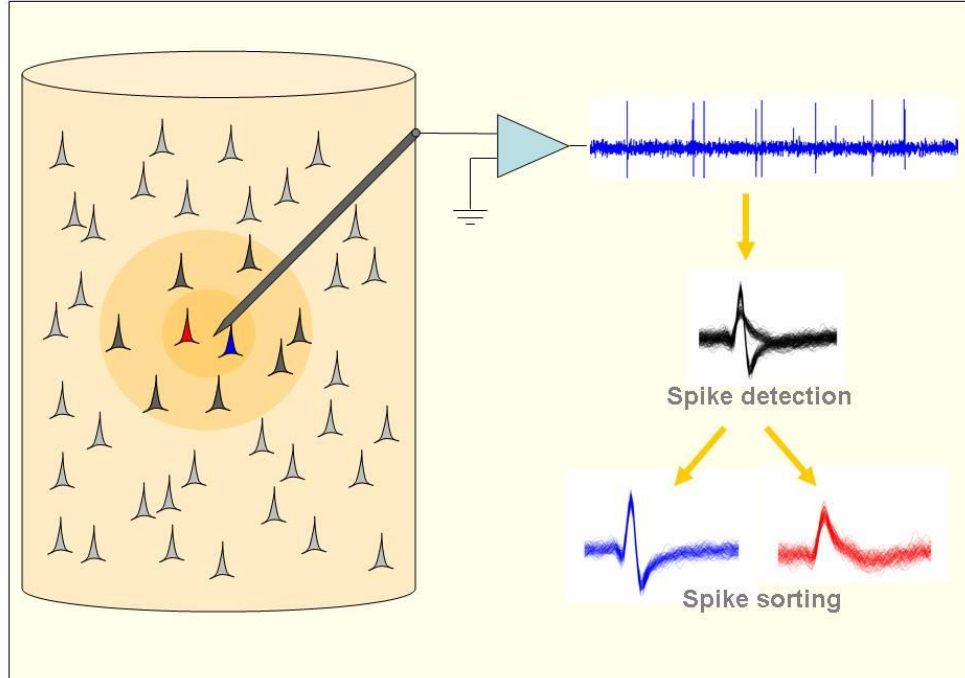
By the end of this workshop, you will be able to:

- Run Kilosort 4.0 efficiently on an HPC cluster (O2 or FASRC)
- Use the web app to curate/visualize spike sorted results
- Know parameter options to customize Kilosort

# Agenda

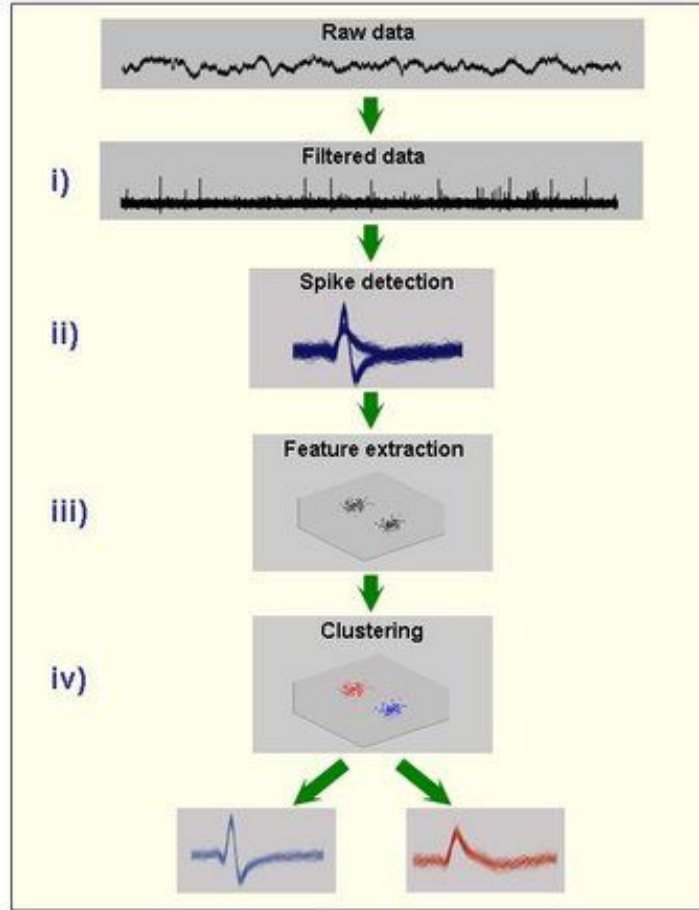
- 1 Intro to spike sorting & why do it on an HPC cluster
- 2 Getting started  
Getting set up with data/container, submitting a job
- 3 Spike sorting outputs  
Web app, how to work with spike interface objects
- 4 Getting more advanced  
Hyperparameters, multiple data directories
- 5 Open session for troubleshooting

# Intro to Spike Sorting



[http://www.scholarpedia.org/article/Spike\\_sorting](http://www.scholarpedia.org/article/Spike_sorting)

# Intro to Spike Sorting



[http://www.scholarpedia.org/article/Spike\\_sorting](http://www.scholarpedia.org/article/Spike_sorting)

# Lots of Spike Sorting Algorithms

algorithms	no drift	medium drift	high drift	fast drift	step drift	step drift aligned	runtime (minutes)
Kilosort4	<b>526</b>	<b>534</b>	<b>477</b>	<b>506</b>	<b>436</b>	<b>512</b>	25.4 ± 0.7
Kilosort3	495	454	283	446	253	415	69.0 ± 2.1
Kilosort2.5	479	456	321	460	274	433	24.0 ± 0.5
Kilosort2	469	454	408	444	110	71	23.6 ± 0.7
Kilosort [6]	163	147	54	111	7	4	51.8 ± 0.5
IronClust [1, 19]	331	301	239	254	1	1	34.9 ± 0.6
MountainSort4 [3]	316	283	124	262	2	1	82.8 ± 4.4
SpyKING CIRCUS [2]	348	301	135	285	6	4	65.6 ± 2.1
SpyKING CIRCUS 2 [20]	123	131	62	93	7	1	32.9 ± 2.3
HDSort [21]	159	5	19	10	1	1	42.5 ± 1.0
Herding Spikes [22]	25	21	6	13	0	0	24.7 ± 0.2
Tridesclous2 [23]	7	12	3	2	0	1	7.8 ± 0.2

Pachitariu et al, Nature Methods 2024

# Spike Sorting on an HPC Cluster

Kempner Research & Engineering team members, in collaboration with Sabatini lab members, got a workflow in place to perform spike sorting using an HPC Cluster

<https://github.com/KempnerInstitute/ephys-spike-sorting>

# Contributors to AIND Ephys Pipeline by the Allen Institute

**Alessio Buccino**

Electrophysiology Pipeline Engineer  
Scientific Computing Group  
Allen Institute

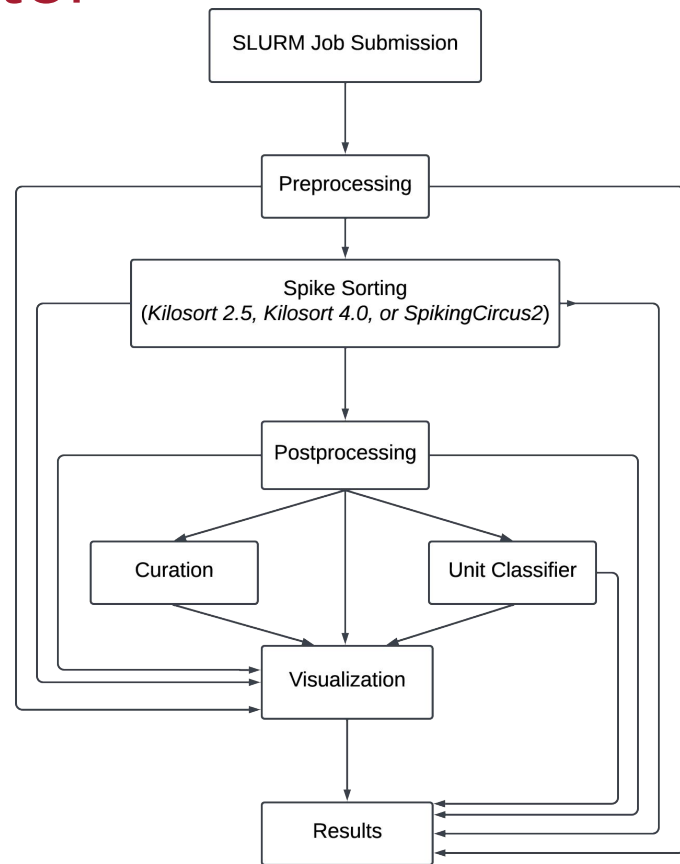
**David Feng**

Sr. Director  
Scientific Computing Group  
Allen Institute



# Spike Sorting on an HPC Cluster

- This workflow is based on AIND Ephys Pipeline by the Allen Institute
- Uses **Nextflow**, which enables reproducible scientific workflows using software containers



# Why use this workflow?

- Avoid local installation and environment creation issues
- Streamlines the analysis process by including preprocessing, spike sorting, post-processing, automated curation using quality control metrics, and unit classification
- Efficiently uses compute resources by maximizing CPU and GPU utilization dramatically reducing compute times
- Eliminates the need for manual curation GUIs by displaying results in an accessible web app

# Choosing a Spike Sorter

This pipeline works with Kilosort 2.5, Kilosort 4.0, & SpyKING CIRCUS.

Our example today uses Kilosort 4.0

algorithms	no drift	medium drift	high drift	fast drift	step drift	step drift aligned	runtime (minutes)
Kilosort4	<b>526</b>	<b>534</b>	<b>477</b>	<b>506</b>	<b>436</b>	<b>512</b>	$25.4 \pm 0.7$
Kilosort3	495	454	283	446	253	415	$69.0 \pm 2.1$
Kilosort2.5	479	456	321	460	274	433	$24.0 \pm 0.5$
Kilosort2	469	454	408	444	110	71	$23.6 \pm 0.7$
Kilosort [6]	163	147	54	111	7	4	$51.8 \pm 0.5$
IronClust [1, 19]	331	301	239	254	1	1	$34.9 \pm 0.6$
MountainSort4 [3]	316	283	124	262	2	1	$82.8 \pm 4.4$
SpyKING CIRCUS [2]	348	301	135	285	6	4	$65.6 \pm 2.1$
SpyKING CIRCUS 2 [20]	123	131	62	93	7	1	$32.9 \pm 2.3$
HDSort [21]	159	5	19	10	1	1	$42.5 \pm 1.0$
Herding Spikes [22]	25	21	6	13	0	0	$24.7 \pm 0.2$
Tridesclous2 [23]	7	12	3	2	0	1	$7.8 \pm 0.2$

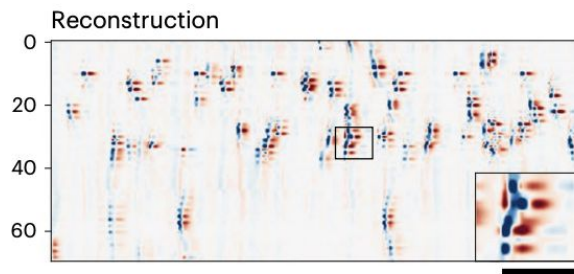
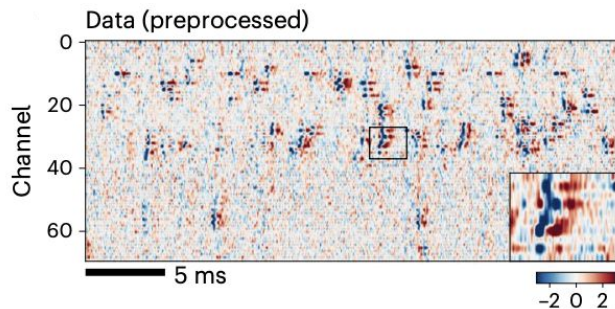
Pachitariu et al, Nature Methods 2024

# Kilosort 4.0

Feature extraction pipeline goals:

- Detect all spikes (even if overlapping)
- Extract spike features after subtracting the influence of the background

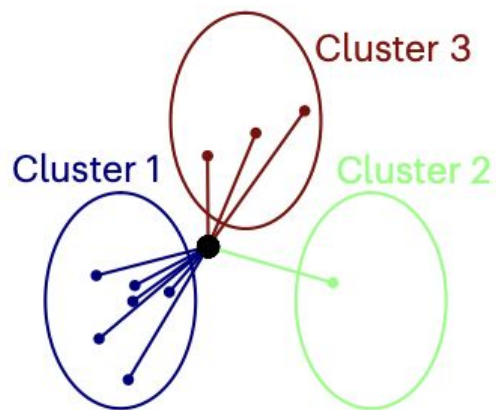
Feature extraction pipeline



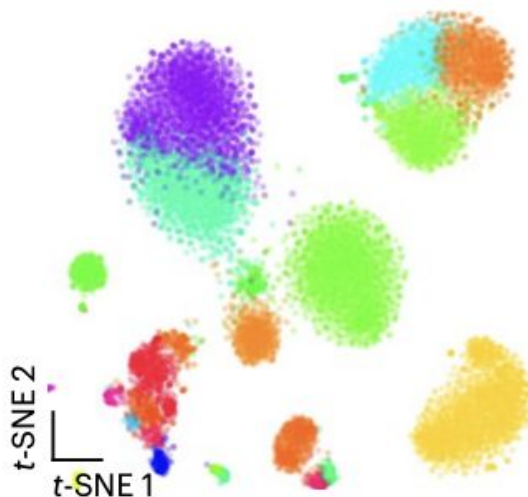
Pachitariu et al, Nature Methods 2024

# Kilosort 4.0

Neighbor clustering



Initial clustering  
(27 clusters)



Only refractory units  
(9 clusters)



Pachitariu et al, Nature Methods 2024

# Agenda

- 1 Intro to spike sorting & why do it on an HPC cluster
- 2 Getting started  
Getting set up with data/container, submitting a job
- 3 Spike sorting outputs  
Web app, how to work with spike interface objects
- 4 Getting more advanced  
Hyperparameters, multiple data directories
- 5 Open session for troubleshooting

# Overview of files & paths

Pipelines on O2 and FASRC have the same file structure but some differences within those files:

[ephys-spike-sorting/pipeline/hms\\_cluster](#)

[ephys-spike-sorting/pipeline/kempner\\_cluster](#)

- 1) **spike\_sort.slm:**
  - a) batch script you'll submit to run job
  - b) sets directory paths, activates conda environment, calls nextflow pipeline to run main\_slurm.nf with specified parameters
- 2) **Nextflow\_slurm.config:**
  - a) Specifies some config options for nextflow
- 3) **Main\_slurm.nf:**
  - a) Runs several processes in parallel (filtering, data curation, data conversion, etc), then runs spike sorting, then puts it all together (you'll see multiple jobs running on the cluster)

# Prepping your data

We have example data on both the FASRC and O2 cluster you can use for this workshop

To prepare your own data:

- 1) Transfer your data to the cluster  
If you have data on O2 but are using the Kempner cluster, you can use Globus!
- 2) Ensure each experiment's data is in its own directory
- 3) Ensure it matches this expected data structure: recording & meta data in parent directory  
(can use spikeGLX, openEphys, or NWB formatting)

```
data_dir
├── 20240805_M100_4W50_g0_t0.imec0.ap.bin
└── 20240805_M100_4W50_g0_t0.imec0.ap.meta
```

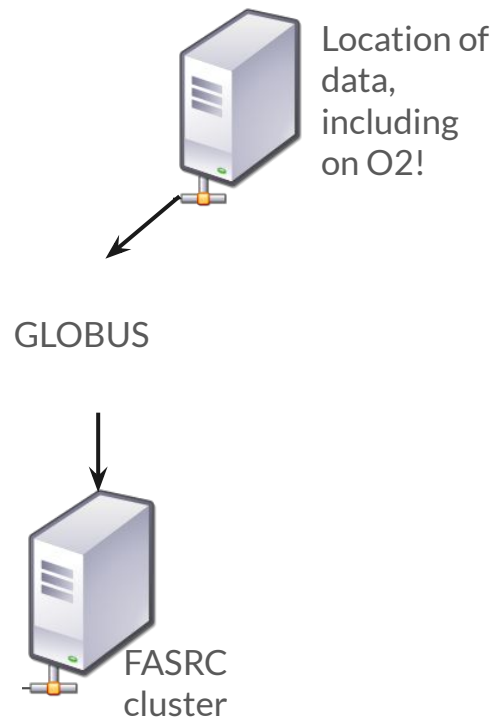


# Globus

File sharing service designed for the secure and efficient transfer of large datasets

Brief intro to Globus can be found in [Kempner Computing Handbook](#)

Can be used to transfer from O2 to FASRC



# Setting up your container

To ensure a seamless pipeline, we'll use a **singularity container**, which allows us to work in a reproducible environment identical to what the developers used.

## FASRC/O2 cluster users:

- Cached containers are accessible from a shared directory! No need to do anything
- Location of containers:  
/n/holyhfs06/LABS/kempner\_shared/Everyone/workflow/ephys-spike-sorting-2024/containers
- /n/app/singularity/containers/shared/sabatini/EPHYSv2\_Singularity\_Image/

## Other cluster users:

- Use the environment/pull\_singularity\_containers.sh script to pull copies of the containers to a location of your choice (specified by the CONTAINER\_DIR variable)
- Set the environment variable EPHYS\_CONTAINER\_DIR to the container directory

# Final Steps

## 1) Decide on directory paths & update in spike\_sort.slm

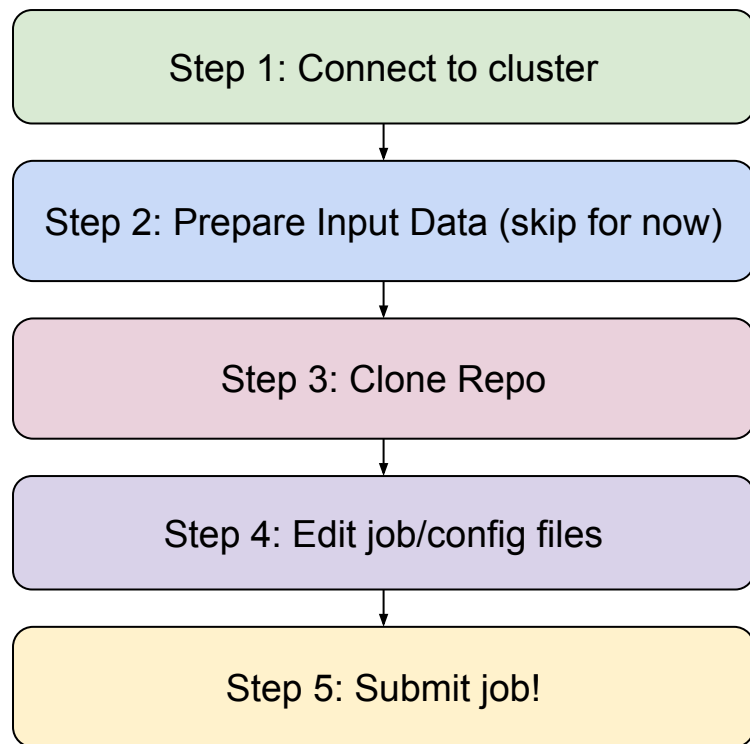
- **DATA\_PATH**: Specifies the location of your input data.
- **WORK\_DIR**: A temporary work directory used by the pipeline during execution. e.g. `"/scr_tmp_dir"`
- **RESULTS\_PATH**: Defines where the pipeline will store the generated output files. e.g. `"/output"`
- **PIPELINE\_PATH**: Location of nextflow pipeline and nextflow config files. Usually `"/repo_path/pipeline/kempner_cluster"` or `"/"`

## 2) Decide on slurm account and partitions

Update in spike\_sort.slm, nextflow\_slurm.config, main\_slurm.nf

Remember that Kempner partitions require GPUs! You should use a non-GPU partition in nextflow\_slurm.config and spike\_sort.slm, like test or serial\_requeue, if you have access

# Exercise: Submitting Job on Example Data



- 1) Complete steps 1, 3, 4, and 5 of either the [HMS workflow](#) (if using O2) or the [Kempner workflow](#) (if using FASRC). Use our example data for now
  - a) **O2:** `/n/scratch/users/b/bad395/public-data`
  - b)
  - c) **FASRC:**  
`/n/holyifs06/LABS/kempner_shared/Everyone`  
`/workflow/ephys-spike-sorting-2024/data/sample_data_1/dir1/20240108_M175_4W50_g0_imec0/`
- 2) If you have extra time and have your own data, try to submit a job with your data!


Ask for help if you get stuck

# Agenda

- 1 Intro to spike sorting & why do it on an HPC cluster
- 2 Getting started  
Getting set up with data/container, submitting a job
- 3 Spike sorting outputs  
Web app, how to work with spike interface objects
- 4 Getting more advanced  
Hyperparameters, multiple data directories
- 5 Open session for troubleshooting

# Output Structure

```
curated/          postprocessed/  processing.json  visualization_output.json  
data_description.json  preprocessed/  spikesorted/
```



Provides visualizations of timeseries, drift maps, and the sorting output using Figurl






## Example outputs on O2:

/n/scratch/users/b/bad395/public-data/output/

## Example outputs on FASRC Cluster:

/n/holyifs06/LABS/kempner\_shared/Everyone/workflow/ephys-spike-sorting-2024/data/output/

# Computed QC metrics

Metric	Icon	Description	Reference
Spike Count		Spike count in an epoch	
Firing rate		Mean spike rate in an epoch	
Presence ratio		Fraction of epoch in which spikes are present	
Amplitude cutoff		Estimate of miss rate based on amplitude histogram	
ISI violations		Rate of refractory-period violations	
Isolation distance		The Mahalanobis distance from a specified unit within as many spikes belong to the specified unit as to other units	Harris et al. Neuron 32.1 (2001): 141-149.
L-ratio		The Mahalanobis distance and chi-squared inverse cdf (given the assumption that the spikes in the cluster distribute normally in each dimension) are used to find the probability of cluster membership for each spike.	Schmitzer-Torbert and Redish. J Neurophys 91.5 (2004): 2259-2272.
$d'$		The classification accuracy between units based on linear discriminant analysis (LDA).	Hill et al. (2011) J Neurosci 31, 8699-9705
Nearest-neighbors		Non-parametric estimate of unit contamination using nearest-neighbor classification.	Chung et al. (2017) Neuron 95, 1381-1394
Silhouette score		A standard metric for quantifying cluster overlap	
Maximum drift		Maximum change in spike position throughout recording	
Cumulative drift		Cumulative change in spike position throughout recording	

- QC metrics for each unit is calculated and reported within the webapp
- The following are used to *flag* a unit as ‘passing’ or ‘failing’ QC:
  - ISI violation ratio  $< 0.5$
  - presence ratio  $> 0.8$
  - amplitude cutoff  $< 0.1$

\*all metrics were adapted from Ecephys ([tutorial here](#))

# Exercise: Using the Web App

Please use this URL:

[https://figurl.org/f?v=npm://@fi-sci/figurl-sortingview@12/dist&d=sha1://5f70c5552\[...\]ck0\\_imec0.ap\\_recording1%20-%20kilosort4%20-%20Sorting%20Summary](https://figurl.org/f?v=npm://@fi-sci/figurl-sortingview@12/dist&d=sha1://5f70c5552[...]ck0_imec0.ap_recording1%20-%20kilosort4%20-%20Sorting%20Summary)

- 1) Compare two SUAs – one that failed and one that passed QC (i.e. units 3 & 4) – do you agree with the automated call?
- 2) Why did Unit 4 fail QC? (*\*Hint: refer to the computed QC metrics slide*)
- 3) Re-label a unit as a different label (e.g. MUA → SUA)
- 4) When you export to JSON, what is exported? Is it all the data?



# Agenda

- 1 Intro to spike sorting & why do it on an HPC cluster
- 2 Getting started  
Getting set up with data/container, submitting a job
- 3 Spike sorting outputs  
Web app, how to work with spike interface objects
- 4 Getting more advanced  
Hyperparameters, multiple data directories
- 5 Open session for troubleshooting

# Multiple data directories

DATA\_PATH="Top Level Directory"

*./multijob\_submission\_wrapper.sh spike\_sort.slm*

**Data Directories:** dir\_1, dir\_2, dir\_3, ...dir\_N

**New:** ./pipeline\_job\_dir1, ./pipeline\_jobdir\_2, ...../pipeline\_jobdir\_N

**New Job Scripts:** spike\_sort.slm.1, spike\_sort.slm.2,...spike\_sort.slm.N

*Submits all these N Slurm sbatch jobs*

Results will be stored in ./output\_dir\_1, ./output\_dir\_2,...

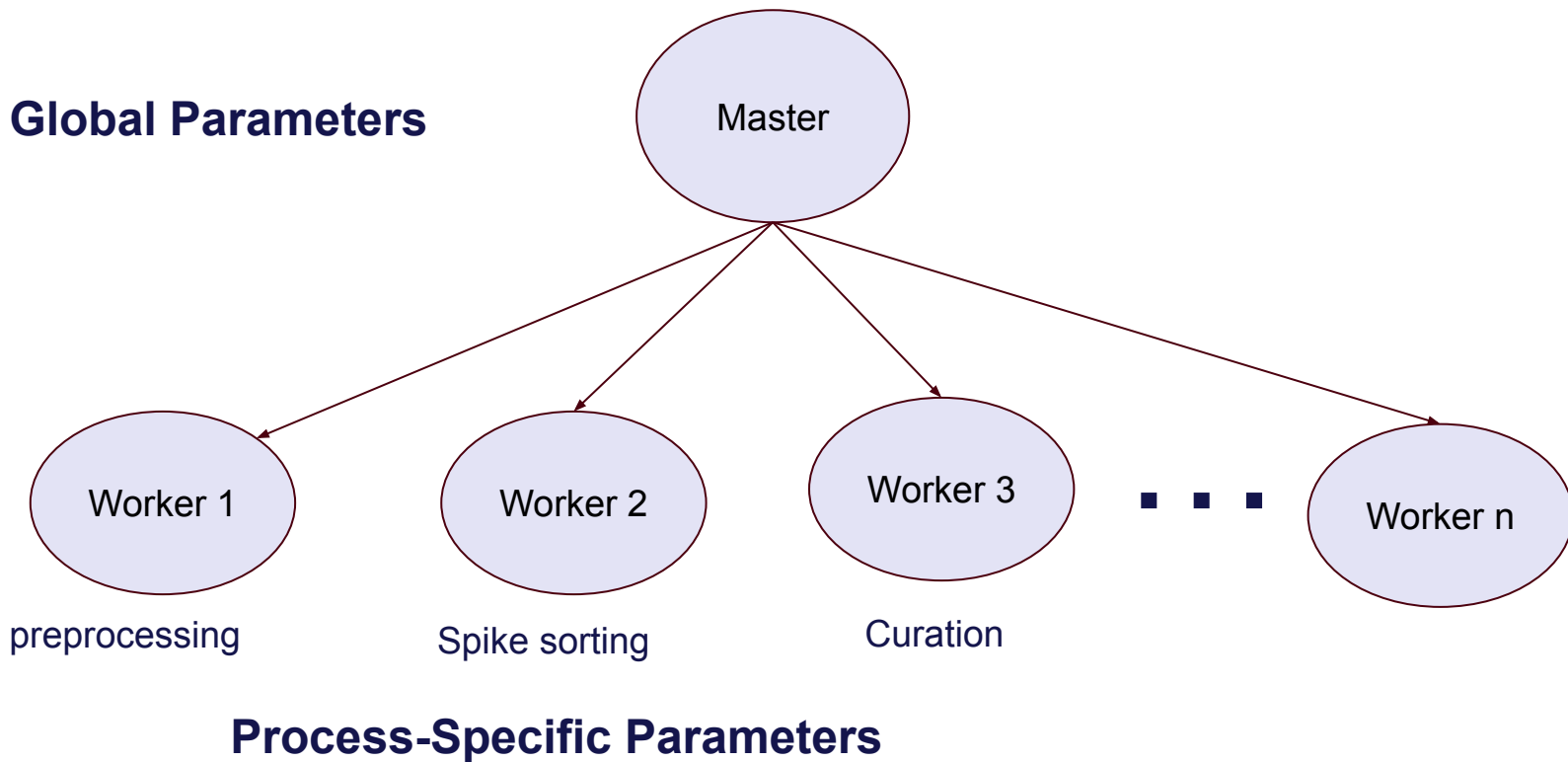
```
sample_data_1
├── dir1
│   ├── 20240108_M175_4W50_g0_imec0
│   │   ├── 20240108_M175_4W50_g0_t0.imec0.ap.bin
│   │   └── 20240108_M175_4W50_g0_t0.imec0.ap.meta
│   └── dir2
│       ├── 20240110_M175_4W50_g0_imec0
│       │   ├── 20240110_M175_4W50_g0_t0.imec0.ap.bin
│       │   └── 20240110_M175_4W50_g0_t0.imec0.ap.meta
│       └── dir3
│           ├── 20240425_M187_4A25_g0_imec0
│           │   ├── 20240425_M187_4A25_g0_t0.imec0.ap.bin
│           │   ├── 20240425_M187_4A25_g0_t0.imec0.ap.meta
│           │   ├── 20240425_M187_4A25_g0_t0.imec0.lf.bin
│           │   └── 20240425_M187_4A25_g0_t0.imec0.lf.meta
│           └── dir4
│               ├── 20240511_M202_4W50_g0_imec0
│               │   ├── 20240511_M202_4W50_g0_t0.imec0.ap.bin
│               └── 20240511_M202_4W50_g0_t0.imec0.ap.meta
```

# Pipeline Parameters

**Global Parameters** - number of Jobs, Sorter Algorithm

**Process-Specific Parameters** - denoising, motion correction, ...

# Global vs Process-Specific Parameters



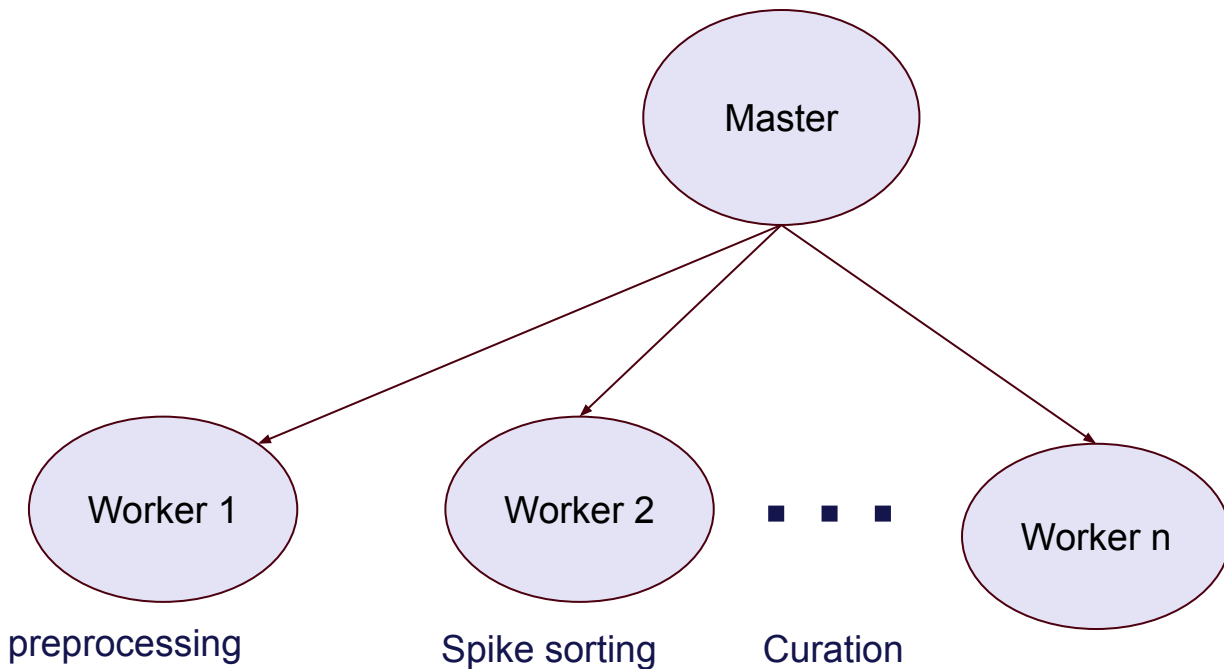
# Global Parameters

**n\_jobs** (integer, jobs to run in parallel)

**sorter** {kilosort25, kilosort4, or spykingcircus2} (kilosort4 preferred)

**runmode** {full, fast} (full is the default, which does not skip any step. Fast skips some steps like motion correction in favor of speed)

# Process-Specific Parameters



`job_dispatch_args`

`preprocessing_args`

`spikesort_args`

`nwb_subject_args`

`nwb_ecephys_args`

# job\_dispatch\_args

**debug** ( Whether to run in DEBUG mode)

**input {aind,spikeglx,nwb}** (Which 'loader' to use (aind | spikeglx | nwb))

For details on process-specific arguments, check [this description](#), from the Allen Institute for Neural Dynamics.

# preprocessing\_args

**denoising** {cmr, destripe}

**max-bad-channel-fraction** MAX\_BAD\_CHANNEL\_FRACTION (Maximum fraction of bad channels to remove)

**motion** {skip, compute, apply}

**motion-preset**{dredge, dredge\_fast, rigid\_fast, kilosort\_like, ...}

For details on process-specific arguments, check [this description](#), from the Allen Institute for Neural Dynamics.



# spikesort\_args

**skip-motion-correction** (Whether to skip the sorter-specific motion correction. Default: False)

**min-drift-channels** MIN\_DRIFT\_CHANNELS (Minimum number of channels to enable motion correction. Default is 96.)

For details on process-specific arguments, check [this description](#), from the Allen Institute for Neural Dynamics.

# Example

```
nextflow -C $PIPELINE_PATH/nextflow_slurm.config \  
-log $RESULTS_PATH/nextflow.log \  
run $PIPELINE_PATH/main_slurm.nf \  
-work-dir $WORK_DIR \  
--n_jobs 4 \  
--sorter kilosort4 \  
--runmode fast \  
--job_dispatch_args "--input $INPUT_DATA_TYPE" \  
--preprocessing_args "--denoising cmr "
```

# Exercise: Modify the pipeline

Draft a pipeline job which skips the sorter-specific motion correction

# Exercise Solution

```
nextflow_slurm.config \
```

```
-log $RESULTS_PATH/nextflow.log \
```

```
run $PIPELINE_PATH/main_slurm.nf \
```

```
-work-dir $WORK_DIR \
```

```
--n_jobs 4 \
```

```
--sorter kilosort4 \
```

```
--runmode fast \
```

```
--job_dispatch_args "--input $INPUT_DATA_TYPE"
```

```
--spikesort_args "--skip-motion-correction True "
```

# Agenda

- 1 Intro to spike sorting & why do it on an HPC cluster
- 2 Getting started  
Getting set up with data/container, submitting a job
- 3 Spike sorting outputs  
Web app, how to work with spike interface objects
- 4 Getting more advanced  
Hyperparameters, multiple data directories
- 5 Open session for troubleshooting



**Kempner**  
INSTITUTE



**HARVARD**  
UNIVERSITY

**Thank you**