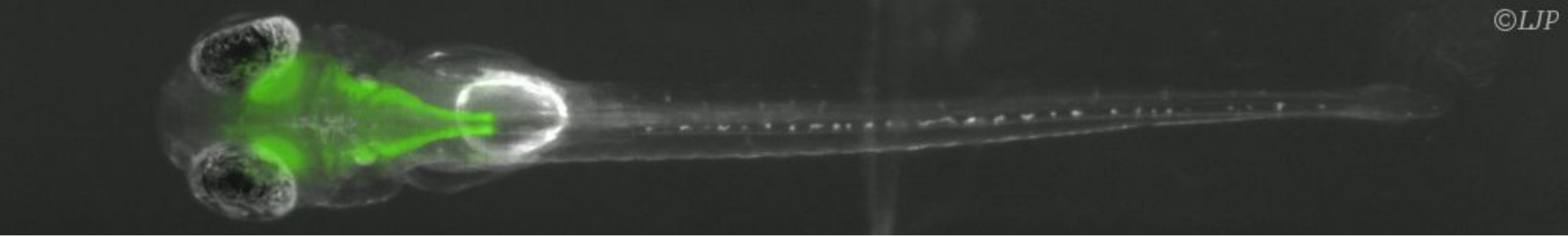
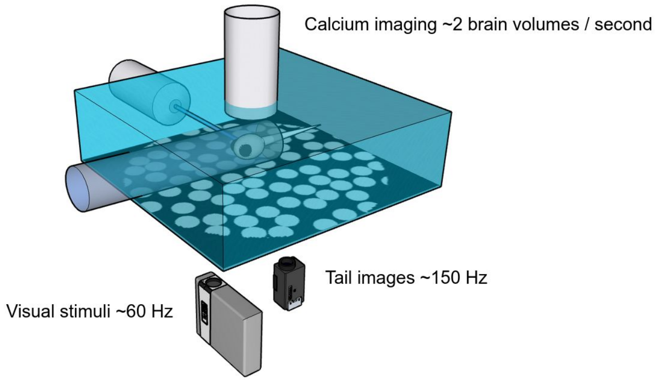


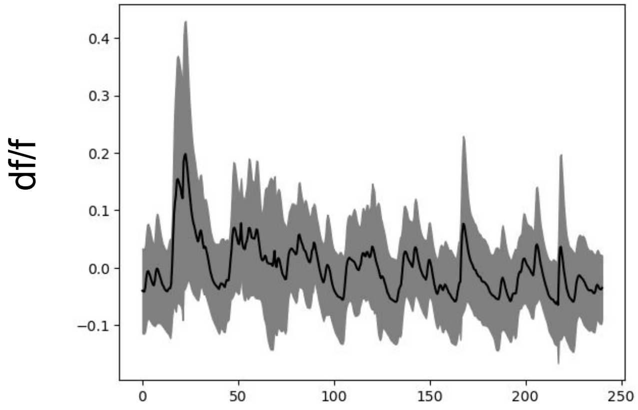
Fish Team !!

Presentation by :
- Akanksha Gupta
- Alice Fermigier
- Faezeh Rabbani
- Nicole Ortiz
- Noé Hamou
- Ourania Semelidou
- Theo Gauvrit

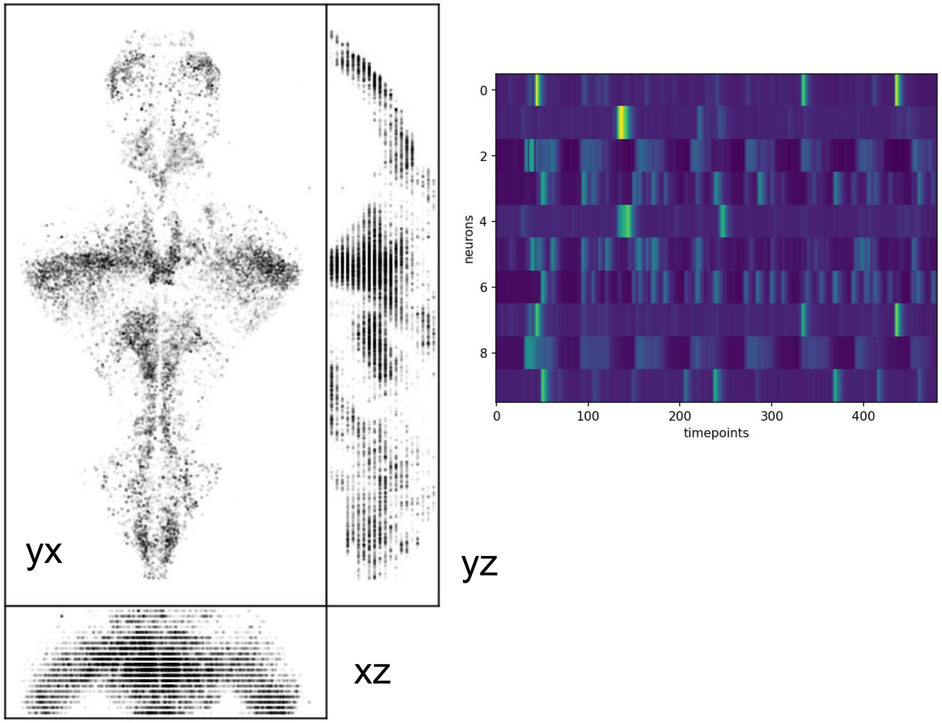
Teaching :
- Leonardo Demarchi
- Matteo Dommanget-Kott
- Georges Debrégeas

Day 1: Processing and data visualization





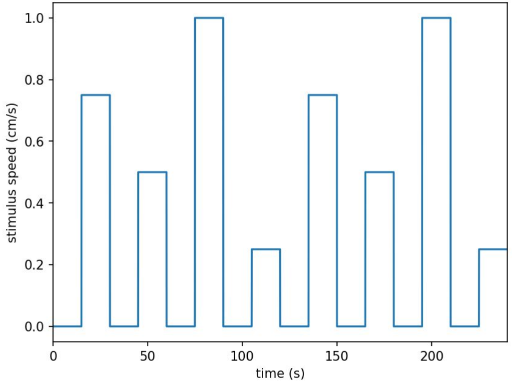
Average neuronal activity of 50 neurons



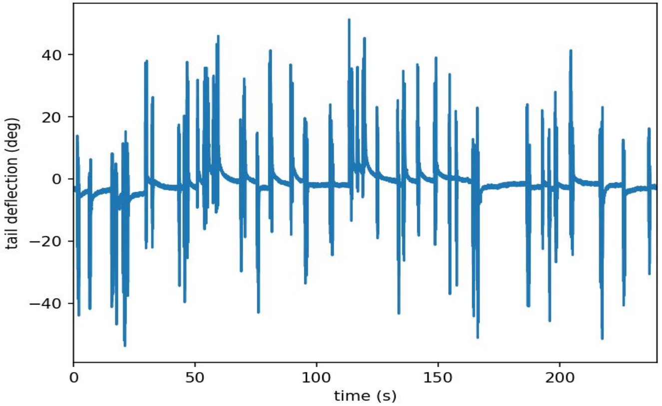
Zebrafish whole brain showing variations in neuronal activity

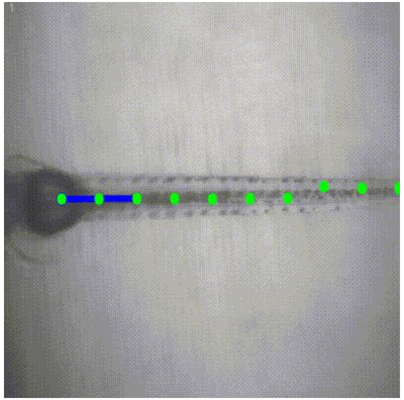
30971 neurons, 481 timepoints

Optomotor response in the zebrafish

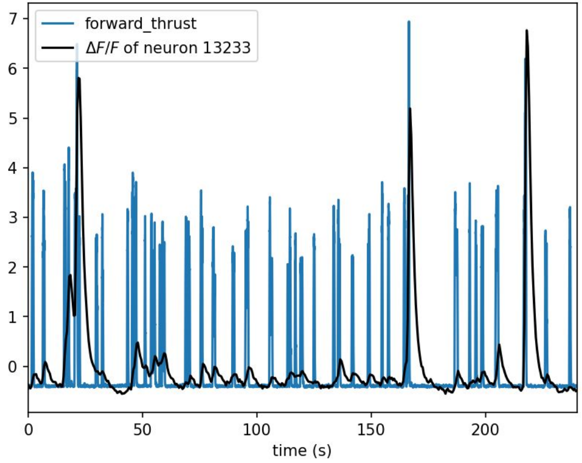


8 trials of visual stimuli with varying speeds





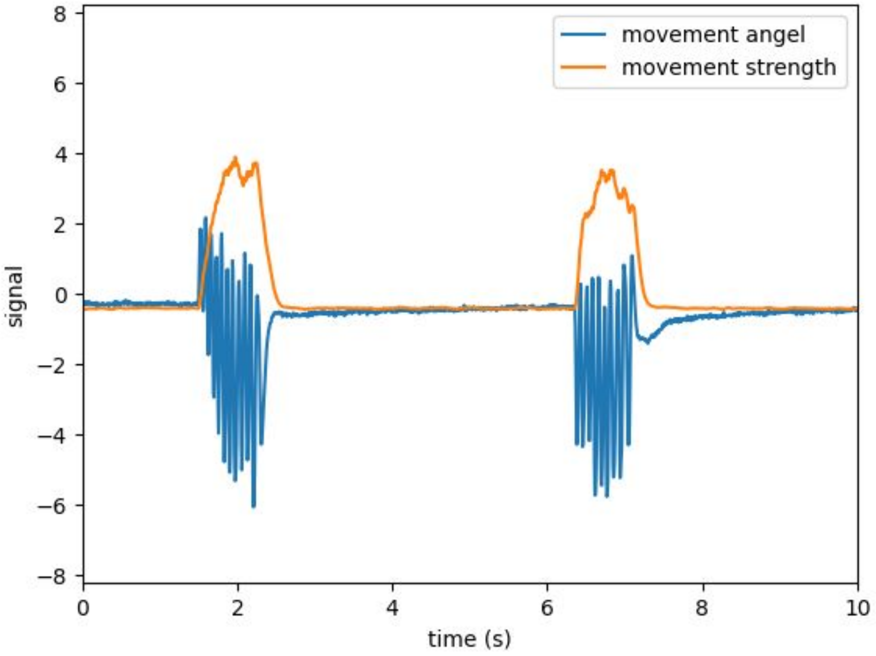
Zebrazoom

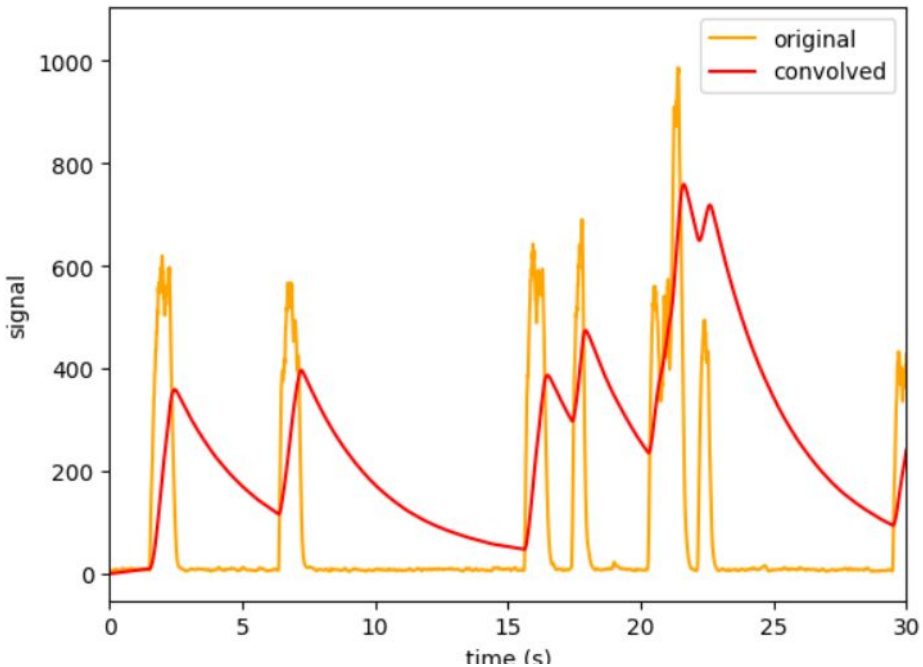


Day 2: **Correlation and regression**

- Correlation between neural activity and external signals
- Multiple linear regression

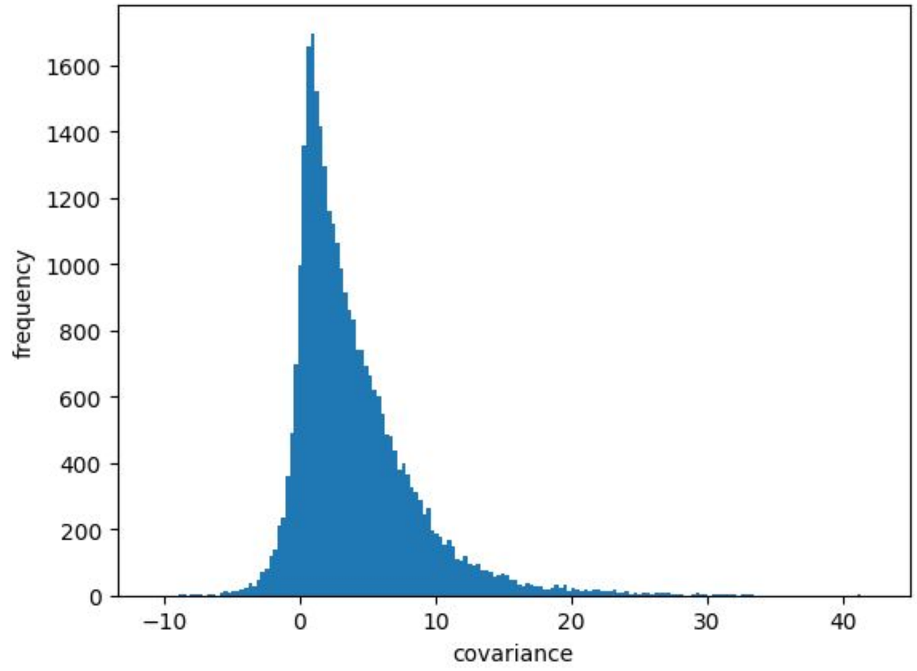
convolving data





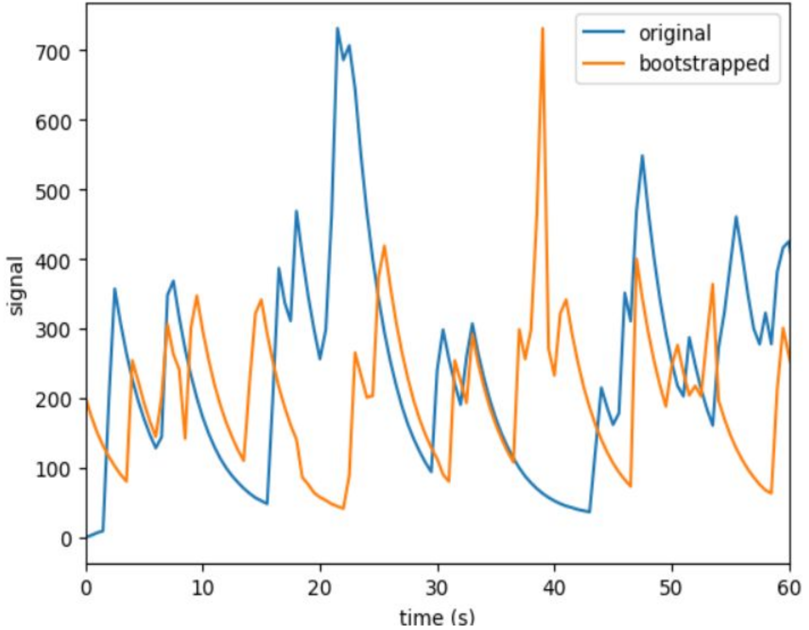
movement strength

Covariance between the behavioral and the neural data

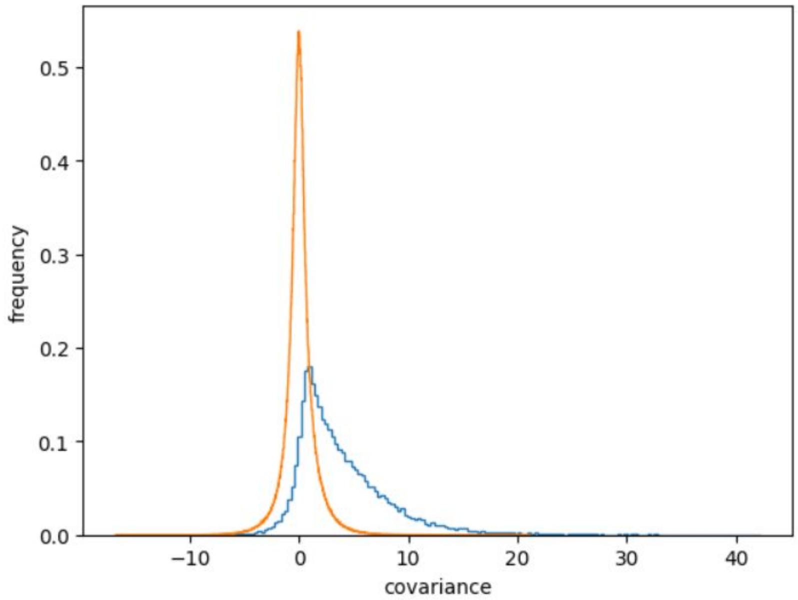


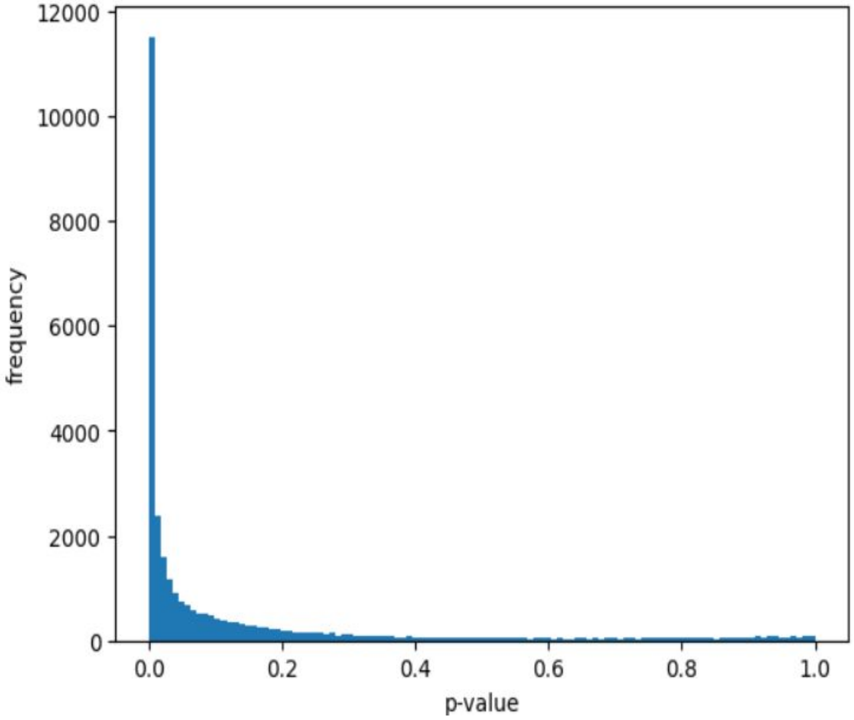
$$Cov(x,y) = \overline{(x-\bar{x})\cdot(y-\bar{y})}$$

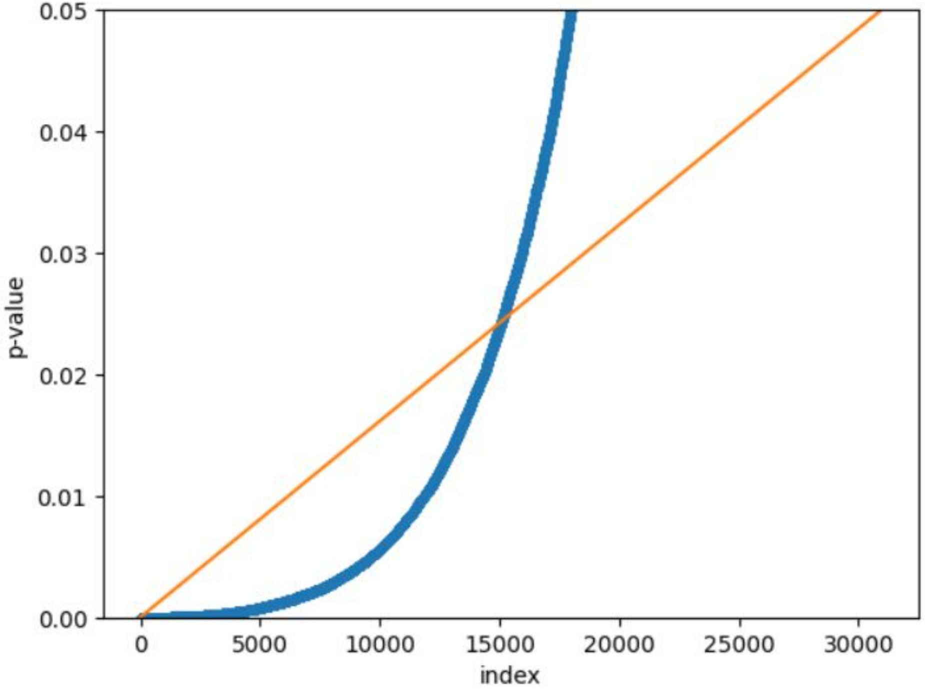
statistical significance



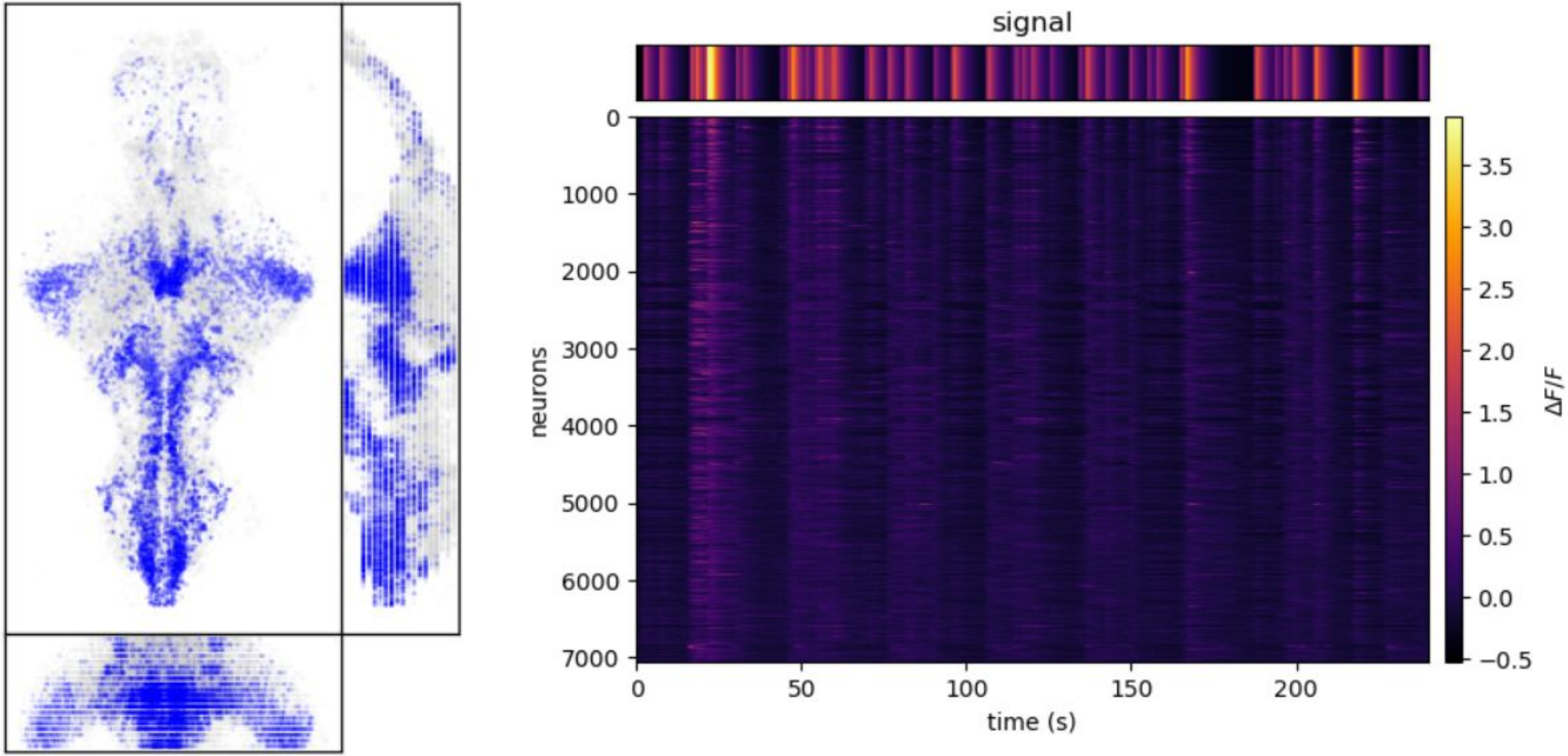
bootstrapping signal



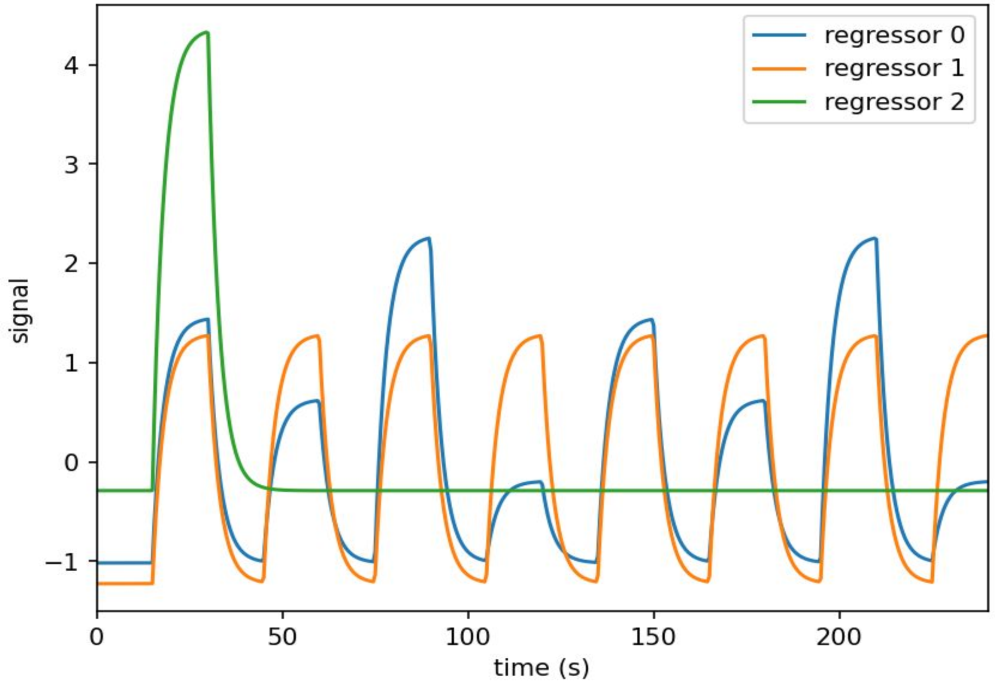




Benjamini−Hochberg $P_k \leq \frac{k}{m}\alpha.$


signal
neurons
0
1000
2000
3000
4000
5000
6000
7000
time (s)
0
50
100
150
200
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
ΔF/F

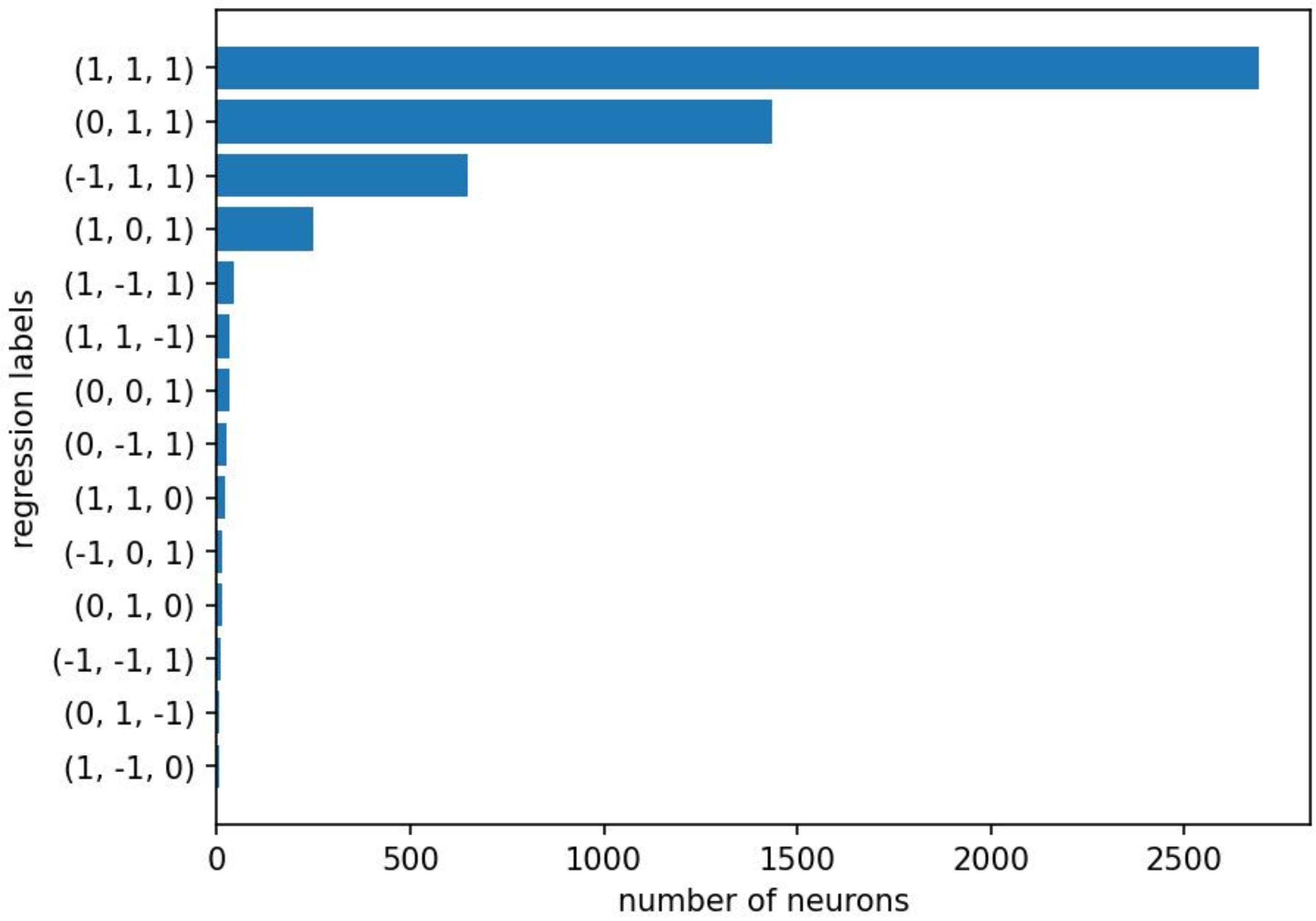
linear regression



$$y_n(t) = \sum_i \beta_{n,i} x_i(t) + \epsilon(t)$$

$$\sum_t (y_n(t) - \sum_i \beta_{n,i} x_i(t))^2$$

neurons that encode multiple different signal



(1, 1, 1)
(0, 1, 1)
(-1, 1, 1)
(1, 0, 1)
(1, -1, 1)
(1, 1, -1)
(0, 0, 1)
(0, -1, 1)
(1, 1, 0)
(-1, 0, 1)
(0, 1, 0)
(-1, -1, 1)
(0, 1, -1)
(1, -1, 0)
regression labels
0
500
1000
1500
2000
2500
number of neurons

3 - Dimensionality Reduction and clustering

- Can we cluster neurons into groups ?

- Can we interpret the activity of those clusters using behavior/stimuli ?

—> K-means

k-Means Clustering

Finding groups of neurons with similar activity

1- Initialise

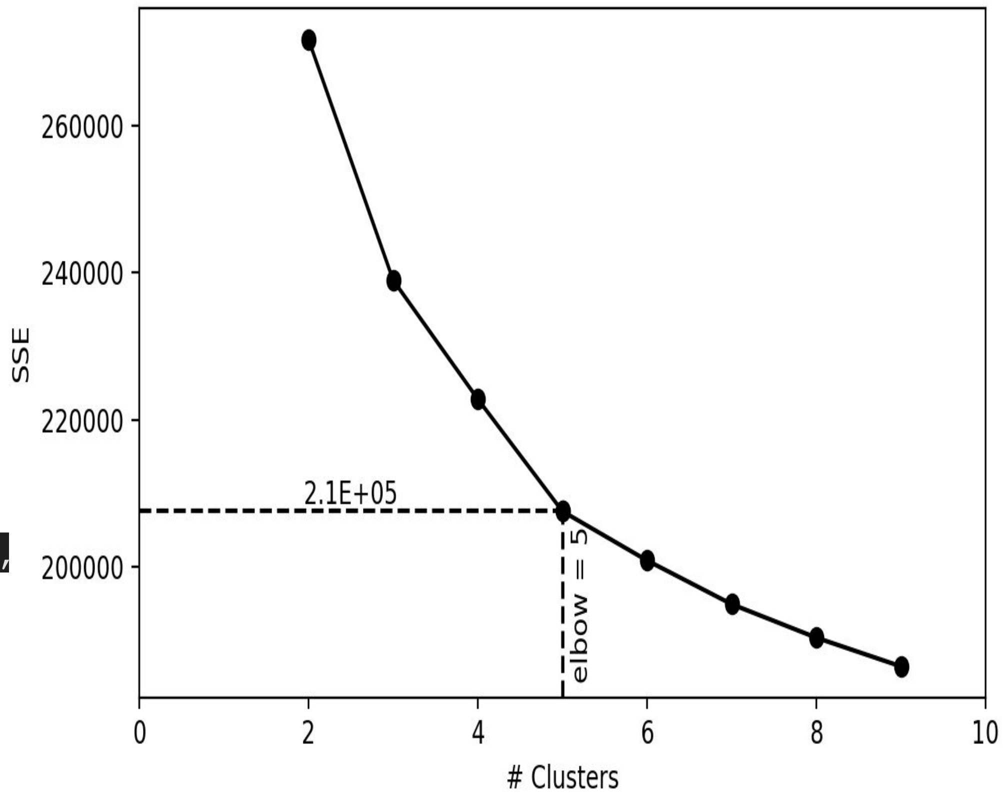
```
kmeans = KMeans(n_clusters=k, **kmeans_params)
```

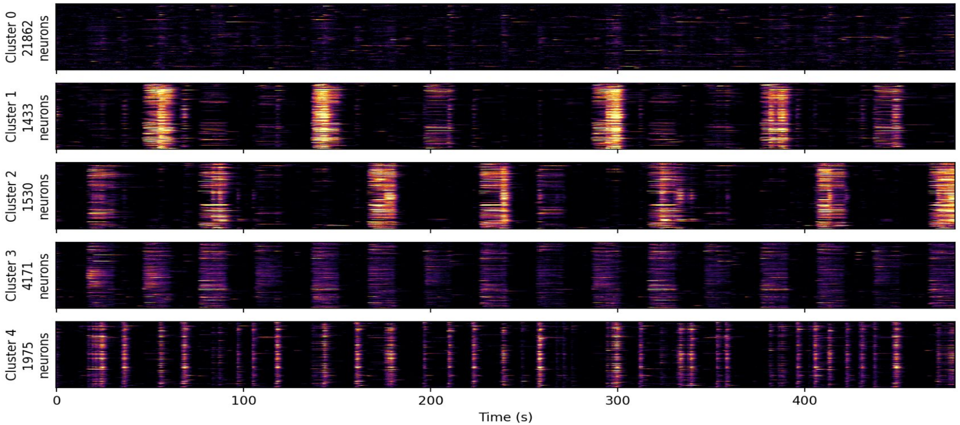
2- Train it

```
kmeans.fit(dffs)
```

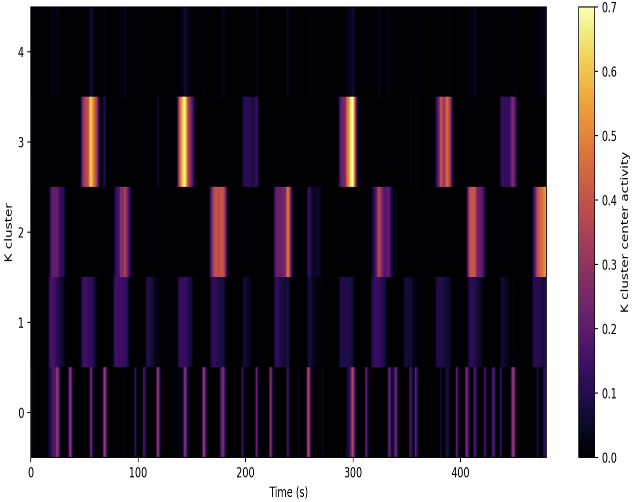
To find the optimal number of clusters

```
kl = KneeLocator(Ks, SSEs, curve="convex",
direction="decreasing")
```

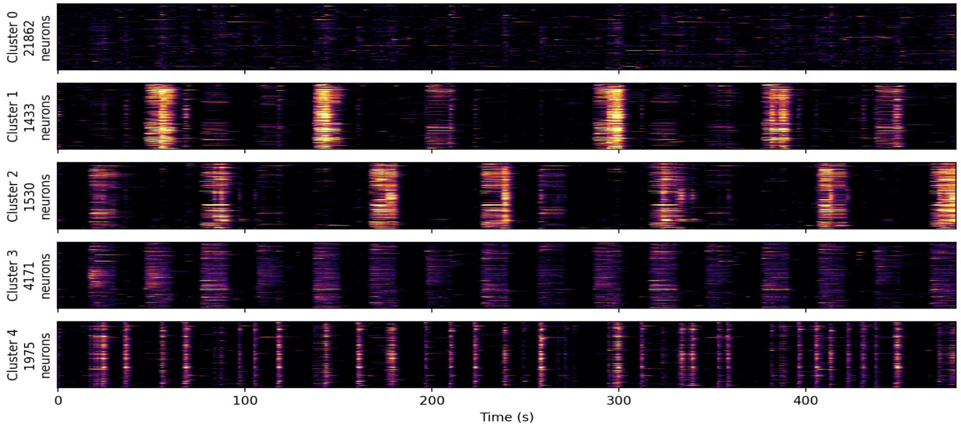


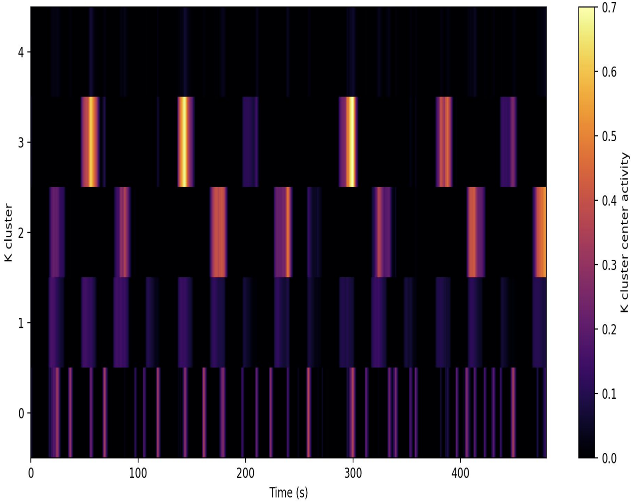


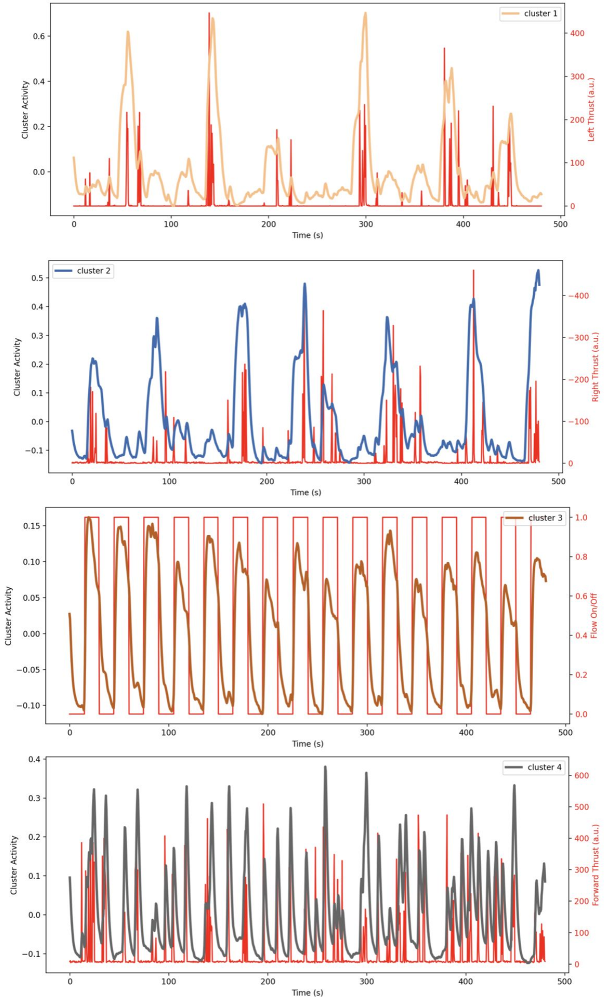
Cluster 0
21862
neurons
Cluster 1
1433
neurons
Cluster 2
1530
neurons
Cluster 3
4171
neurons
Cluster 4
1975
neurons
0
100
200
300
400
Time (s)

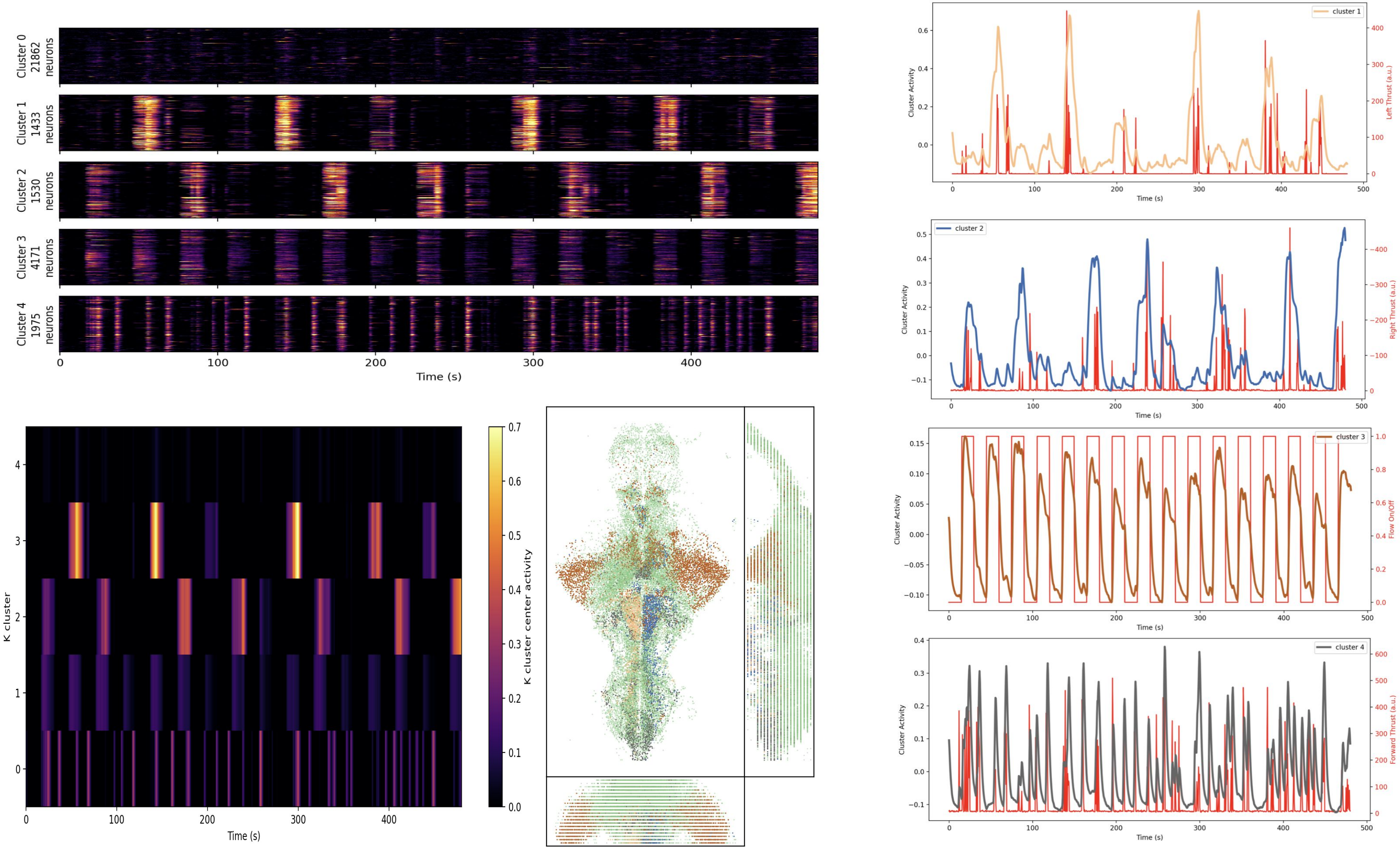


K cluster
4
3
2
1
0
0
100
200
300
400
Time (s)
K cluster center activity
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0


Cluster 0
21862
neurons
Cluster 1
1433
neurons
Cluster 2
1530
neurons
Cluster 3
4171
neurons
Cluster 4
1975
neurons
Time (s)

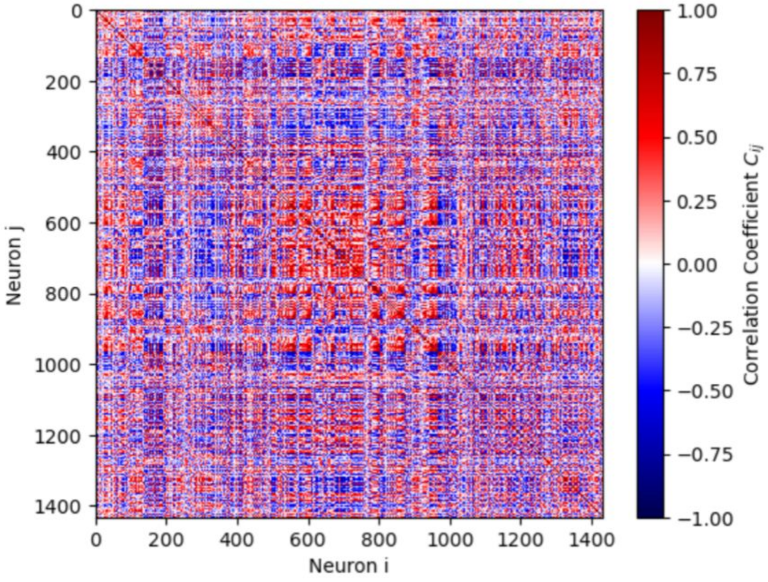

K cluster
Time (s)
K cluster center activity

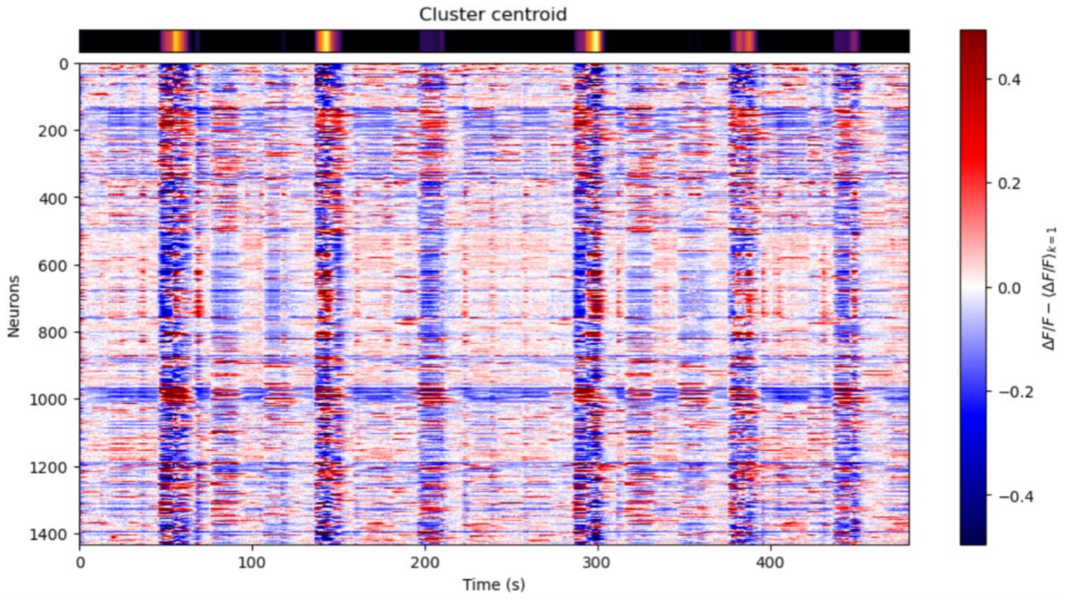

cluster 1
Cluster Activity
Left Thrust (a.u.)
Time (s)
cluster 2
Cluster Activity
Right Thrust (a.u.)
Time (s)
cluster 3
Cluster Activity
Flow On/Off
Time (s)
cluster 4
Cluster Activity
Forward Thrust (a.u.)
Time (s)



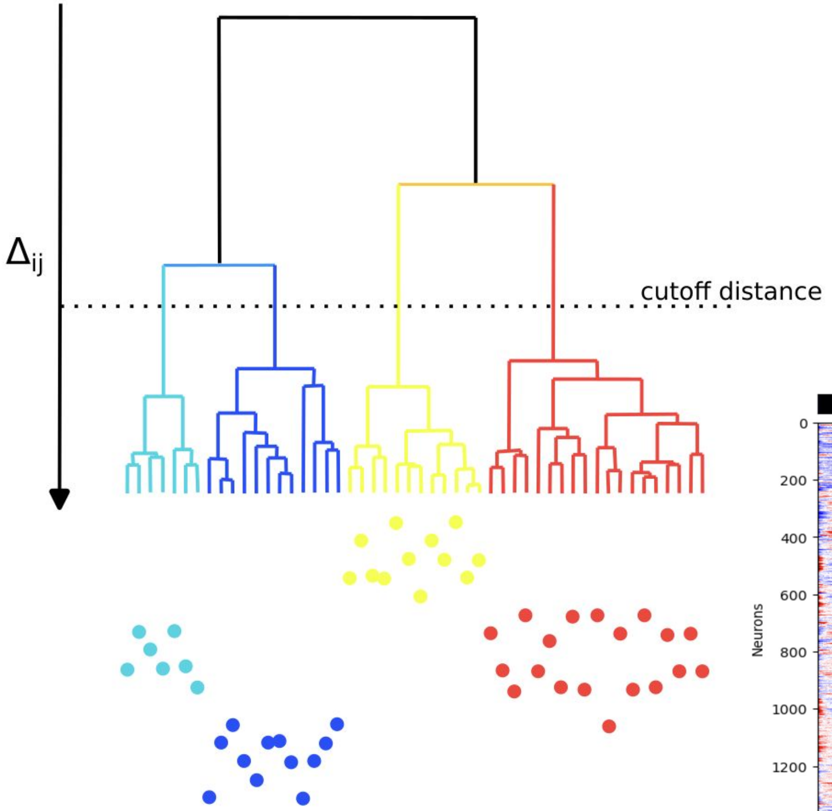
Cluster 0
21862
neurons
Cluster 1
1433
neurons
Cluster 2
1530
neurons
Cluster 3
4171
neurons
Cluster 4
1975
neurons
Time (s)
K cluster
K cluster center activity
Time (s)
cluster 1
Cluster Activity
Left Thrust (a.u.)
Time (s)
cluster 2
Cluster Activity
Right Thrust (a.u.)
Time (s)
cluster 3
Cluster Activity
Flow On/Off
Time (s)
cluster 4
Cluster Activity
Forward Thrust (a.u.)
Time (s)

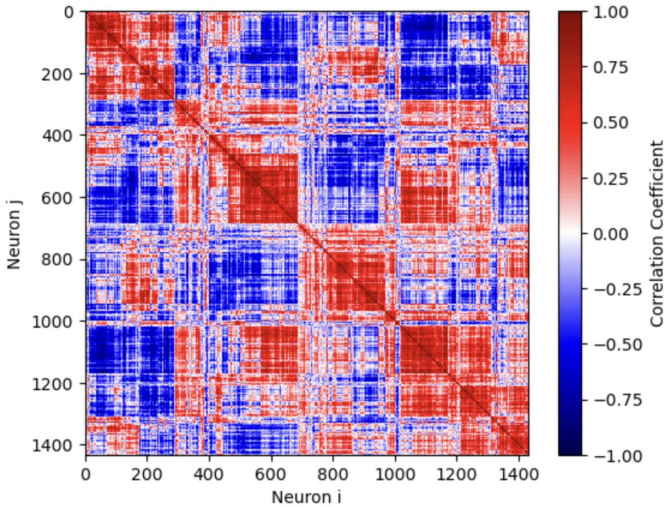
Investigating Individual Clusters

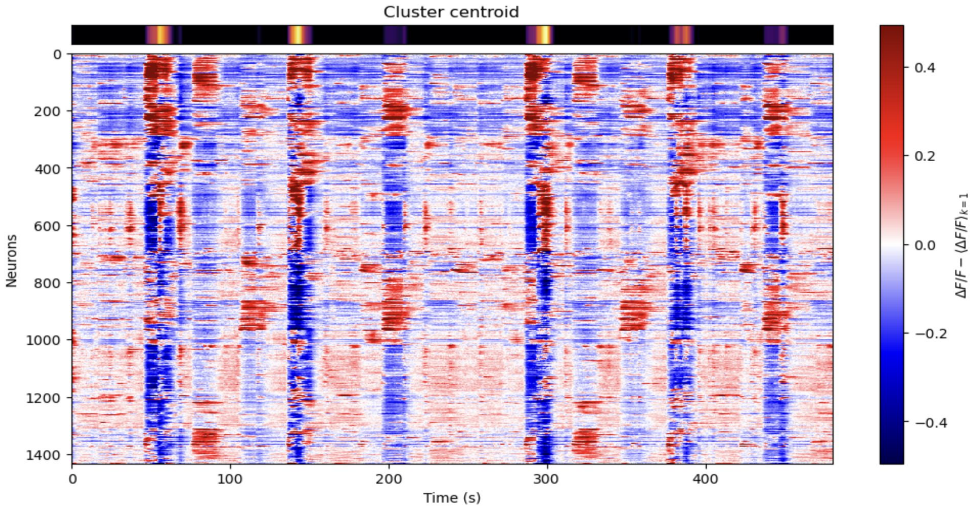




Hierarchical Clustering







3 - Dimensionality reduction and clustering

- Can we find modes in the activity of the brain ?

- Can we interpret those modes using behavior/stimuli ?

—> PCA

3 - Dimensionality reduction and clustering - **PCA**

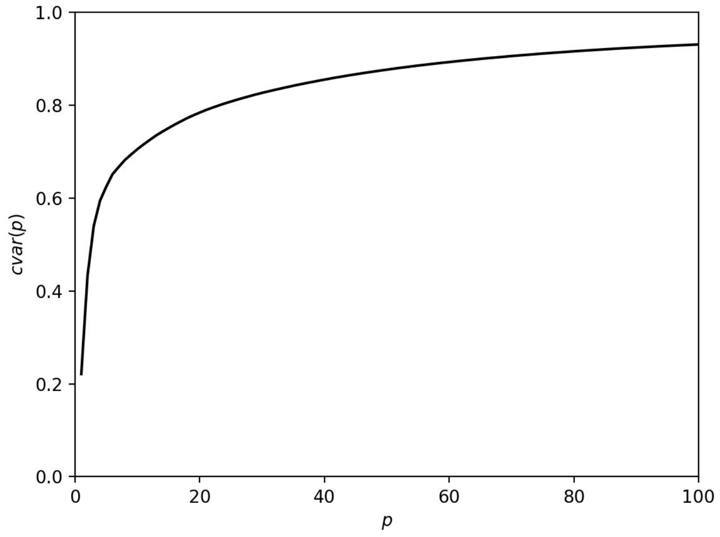
Finding how many components to keep

1-Make a PCA with a large number of components (ex:100)

```
pca = PCA(n_components=100)
pca.fit(dffs.T)
Y = pca.transform(dffs.T)
```

2-Plot the cumulative explained variance

```
cum_exp_var = np.cumsum(pca.explained_variance_ratio_)
```



3 - Dimensionality reduction and clustering - **PCA**

Finding how many components to keep

1-Make a PCA with a large number of components (ex:100)

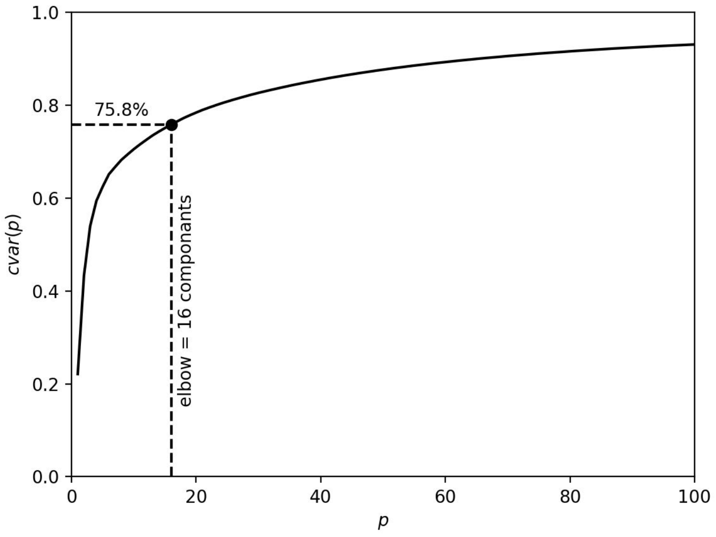
```
pca = PCA(n_components=100)
pca.fit(dffs.T)
Y = pca.transform(dffs.T)
```

2-Plot the cumulative explained variance

```
cum_exp_var = np.cumsum(pca.explained_variance_ratio_)
```

3-Find the elbow in the curve

```
kl = KneeLocator(x, cum_exp_var, curve="concave", direction="increasing")
```

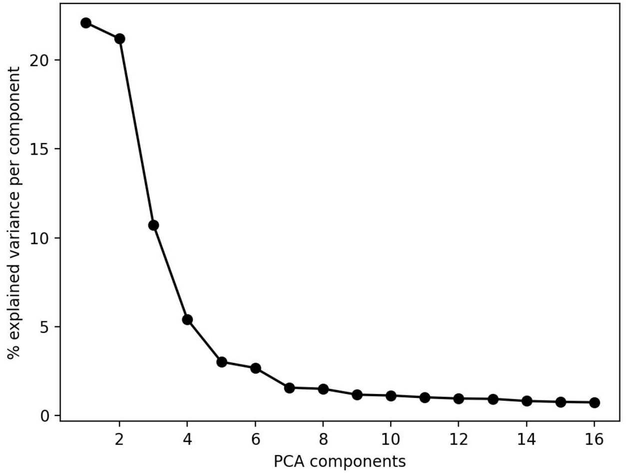


3 - Dimensionality reduction and clustering - **PCA**

Apply on the data

1-Make a PCA with 16 components

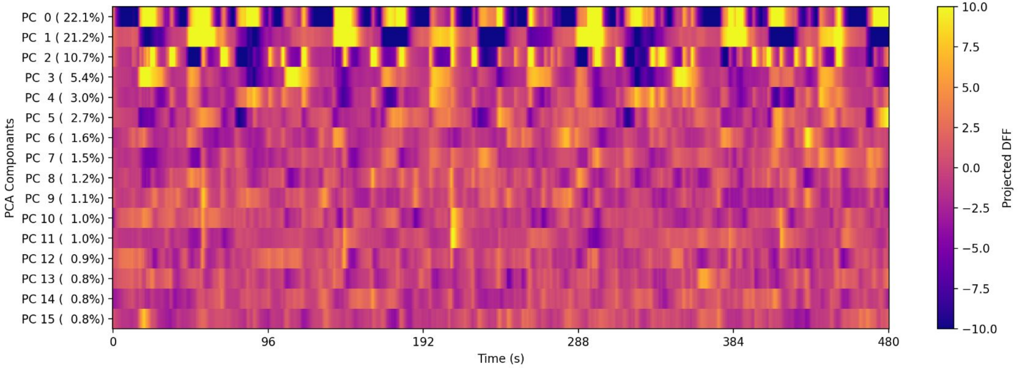
```
pca = PCA(n_components=16)
pca.fit(dffs.T)
Y = pca.transform(dffs.T)
```



2-Look at explained variance for each component

```
exp_var = pca.explained_variance_ratio_
exp_var_tot = exp_var.sum()
```

```
Total explained variance = 75.8%
```



3 - Dimensionality reduction and clustering - **PCA**

Apply on the data

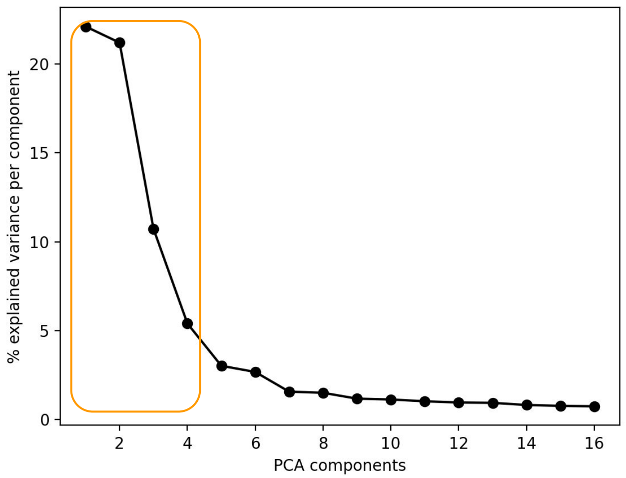
1-Make a PCA with 16 components

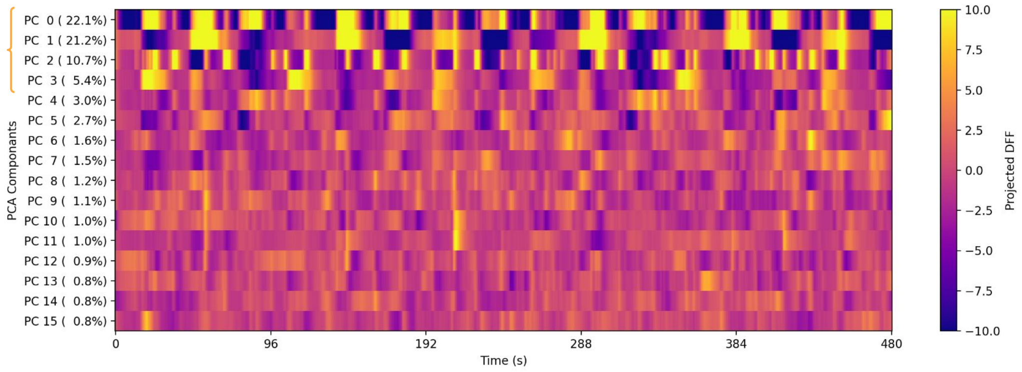
```
pca = PCA(n_components=16)
pca.fit(dffs.T)
Y = pca.transform(dffs.T)
```

2-Look at explained variance for each component

```
exp_var = pca.explained_variance_ratio_
exp_var_tot = exp_var.sum()
```

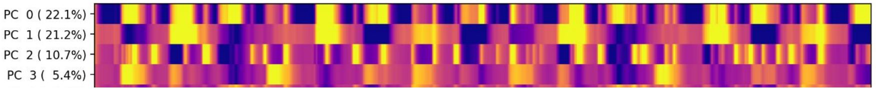
```
Total explained variance = 75.8%
```



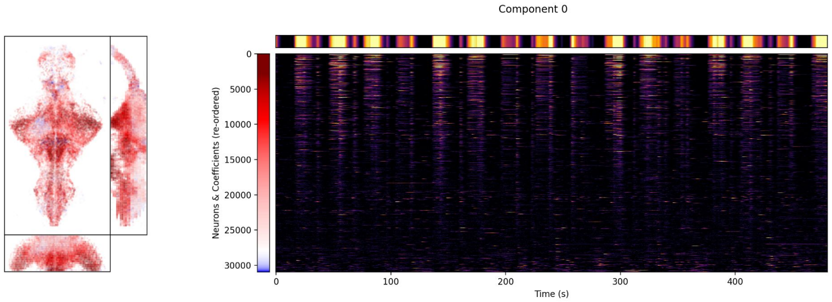


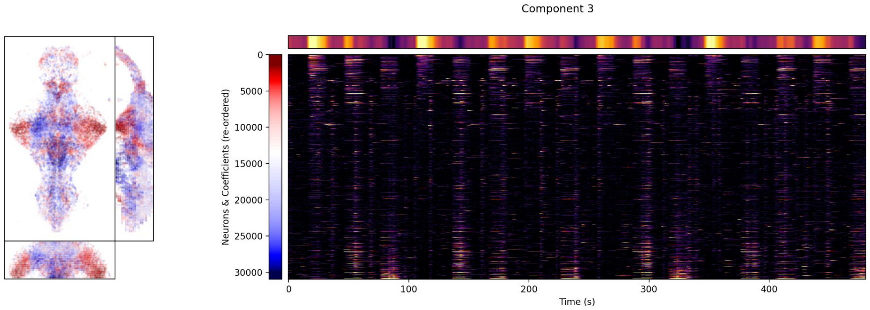
3 - Dimensionality reduction and clustering - **PCA**

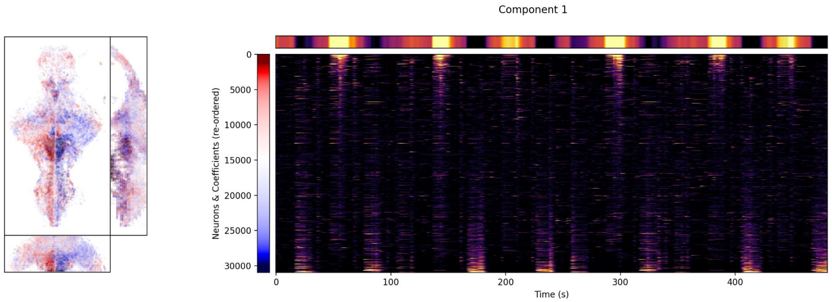
Apply on the data

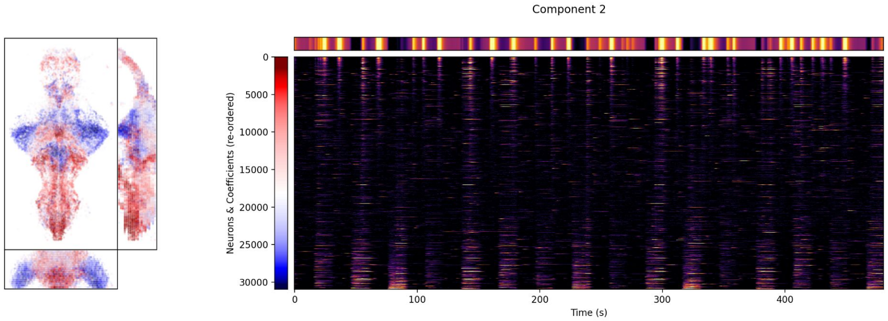


3-Take a look at how much each component contribute to the activity pattern of each neuron









3 - Dimensionality reduction and clustering

- Can we find modes in the activity of the brain ?

- Can we interpret those modes using behavior/stimuli ?

—> PCA

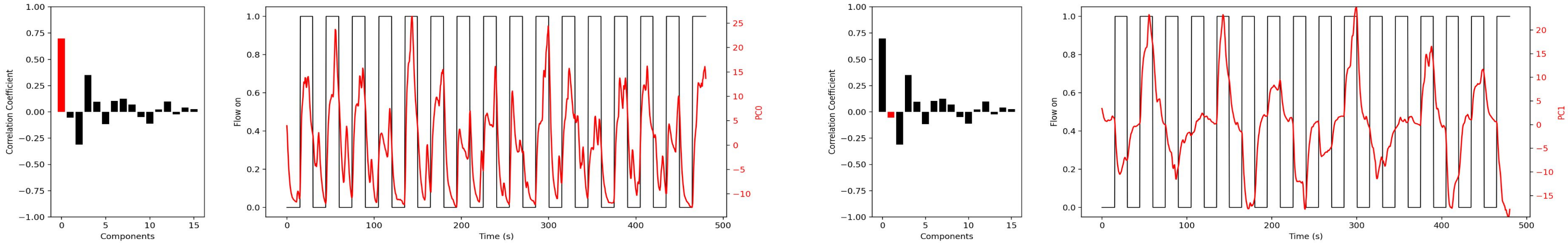
3 - Dimensionality reduction and clustering - **PCA**

How can we interpret PCA components?

Calculate the correlation coefficient of projected ΔF/F activity on each PCA component (Y) with behaviour (here presence of stimulus (flow_on).

```
finite = np.isfinite(flow_on)
C = np.corrcoef(flow_on[finite], Y[finite].T)
cor_comp_dir = C[0,1:]
```

Look at the projected ΔF/F activity on each PCA component aligned with the stimulus (flow on)

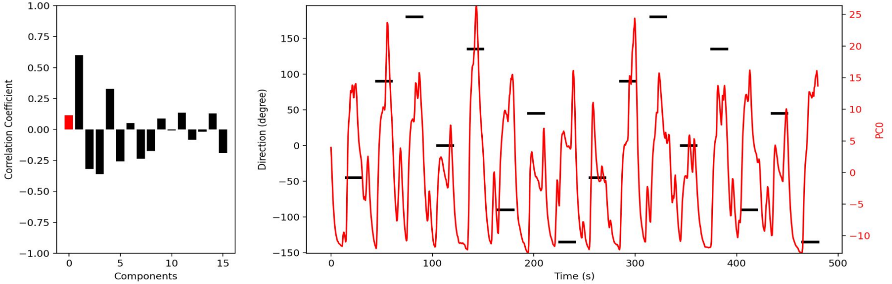


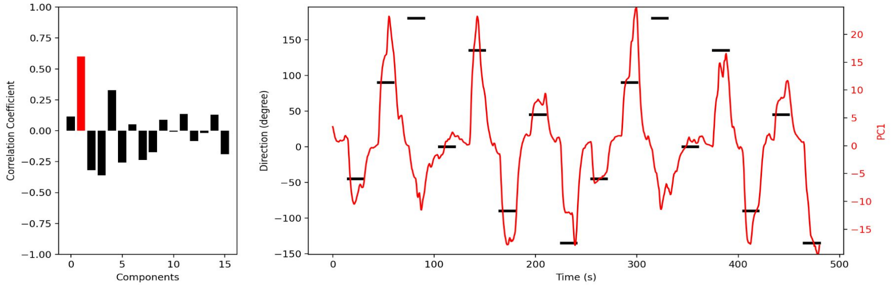
How can we interpret PCA components?

Calculate the correlation coefficient of projected ΔF/F activity on each PCA component (Y) with behaviour (here presence of stimulus (flow_on).

```
finite = np.isfinite(flow_on)
C = np.corrcoef(flow_on[finite], Y[finite].T)
cor_comp_dir = C[0,1:]
```

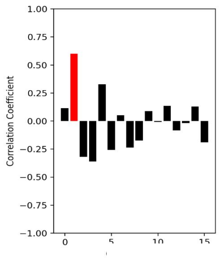
Look at the projected ΔF/F activity on each PCA component aligned with the stimulus (direction of flow)

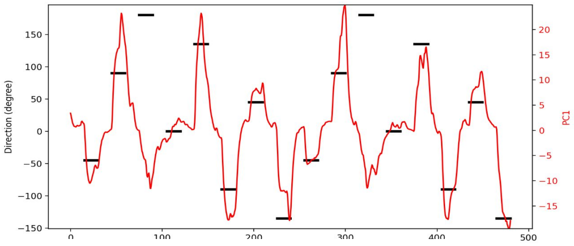


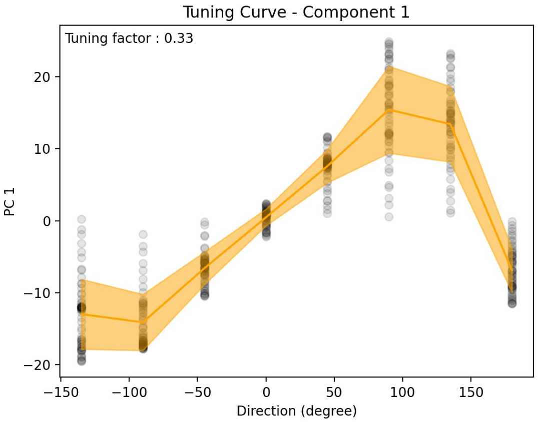


How can we interpret PCA components?

Look at the tuning curves to determine which PCA components are tuned to certain stimulus directions



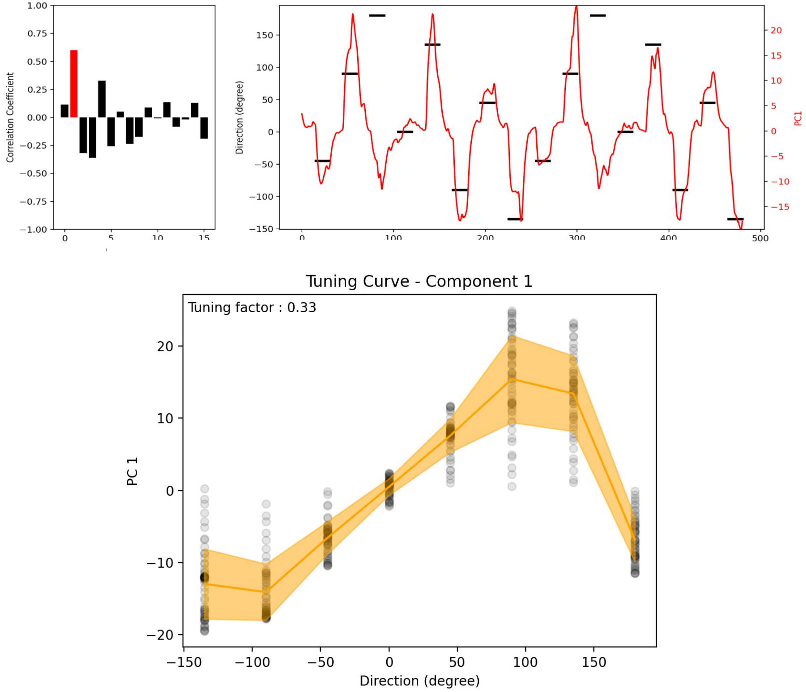


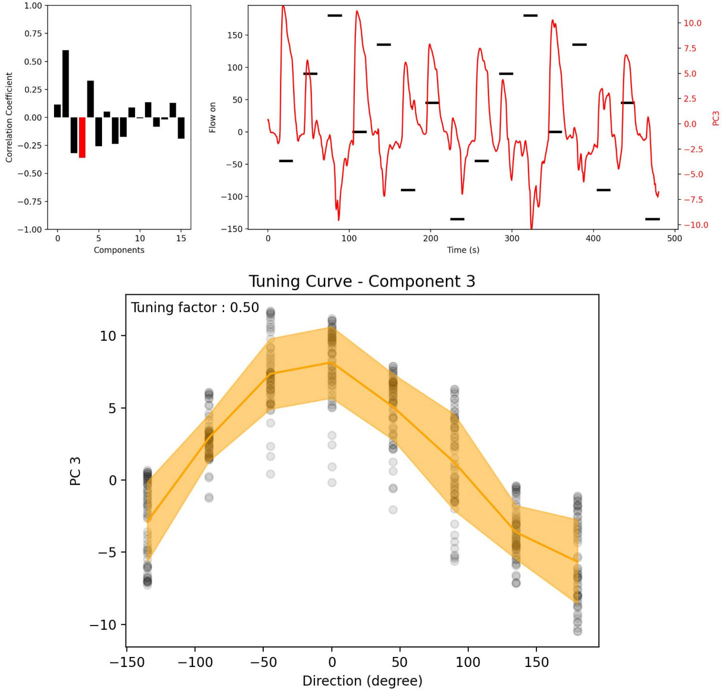


3 - Dimensionality reduction and clustering - **PCA**

How can we interpret PCA components?

Look at the tuning curves to determine which PCA components are tuned to certain stimulus directions





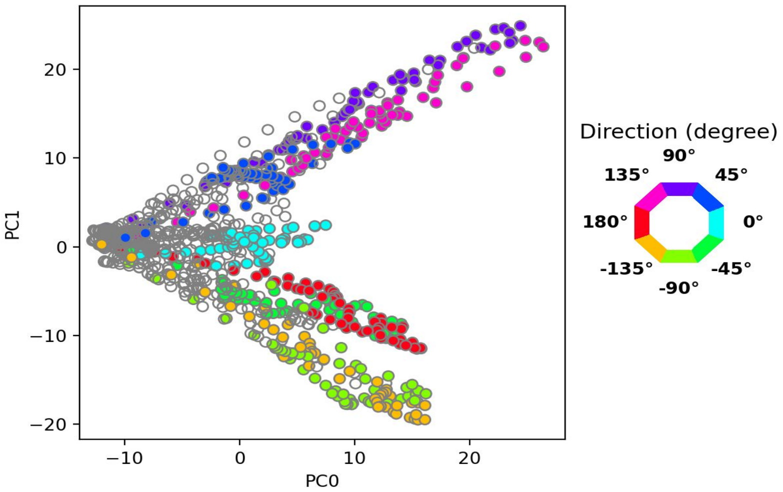
How can we interpret PCA components?

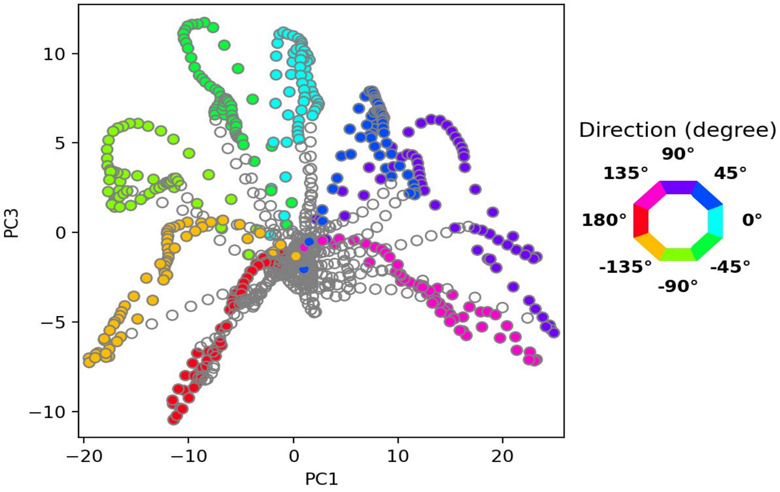
Is there way of plotting the data that would be informative in relation to stimulus direction?

3 - Dimensionality reduction and clustering - **PCA**

How can we interpret PCA components?

Is there way of plotting the data that would be informative in relation to stimulus direction?



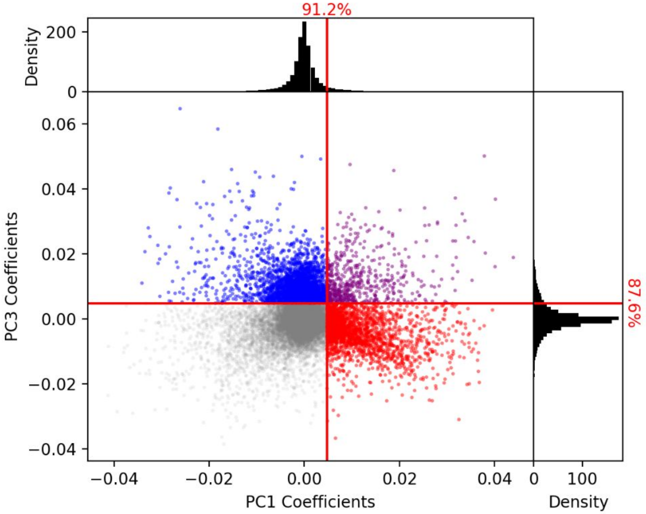


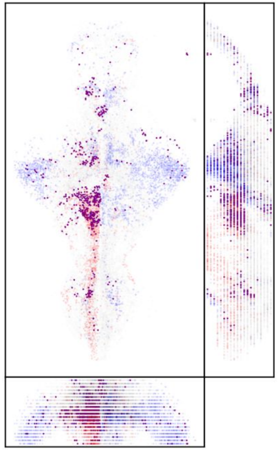
It seems that PC1 and PC3 are sufficient to encode the direction of the stimulus

3 - Dimensionality reduction and clustering - **PCA**

How can we interpret PCA components?

We can try to find a subgroup of neurons that seems to contribute the most to PC1 and PC3 (aka “clustering”)



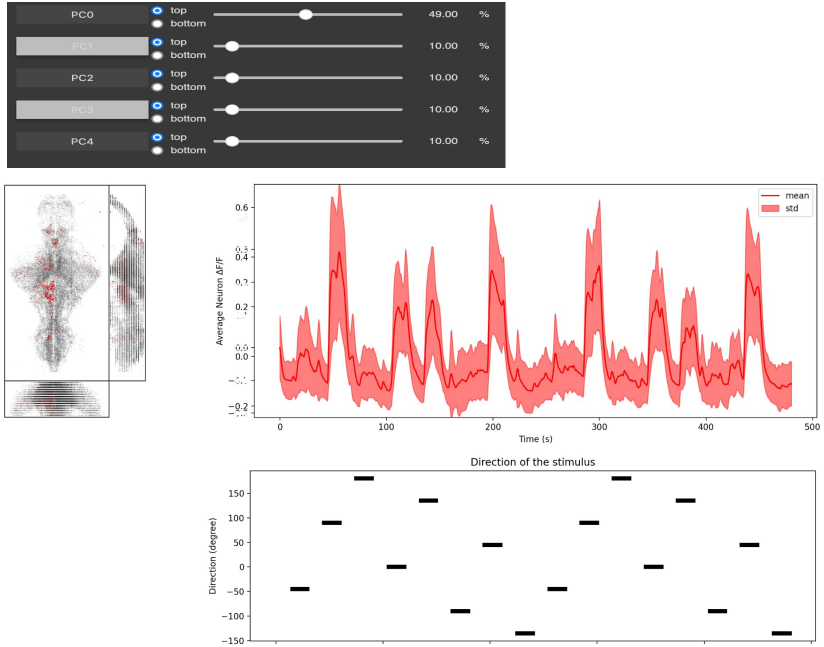


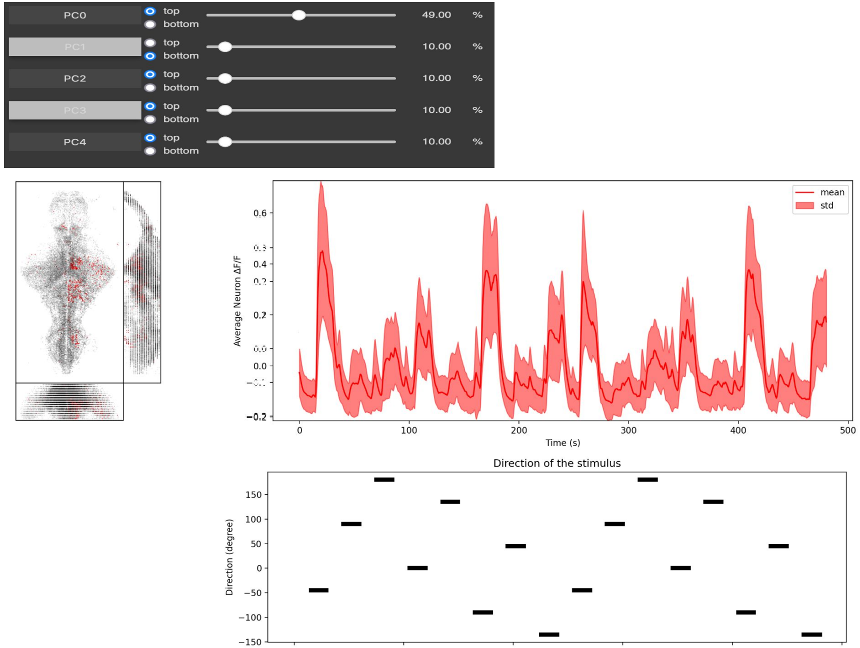
They seem to cluster in space, which is good!

3 - Dimensionality reduction and clustering - **PCA**

How can we interpret PCA components?

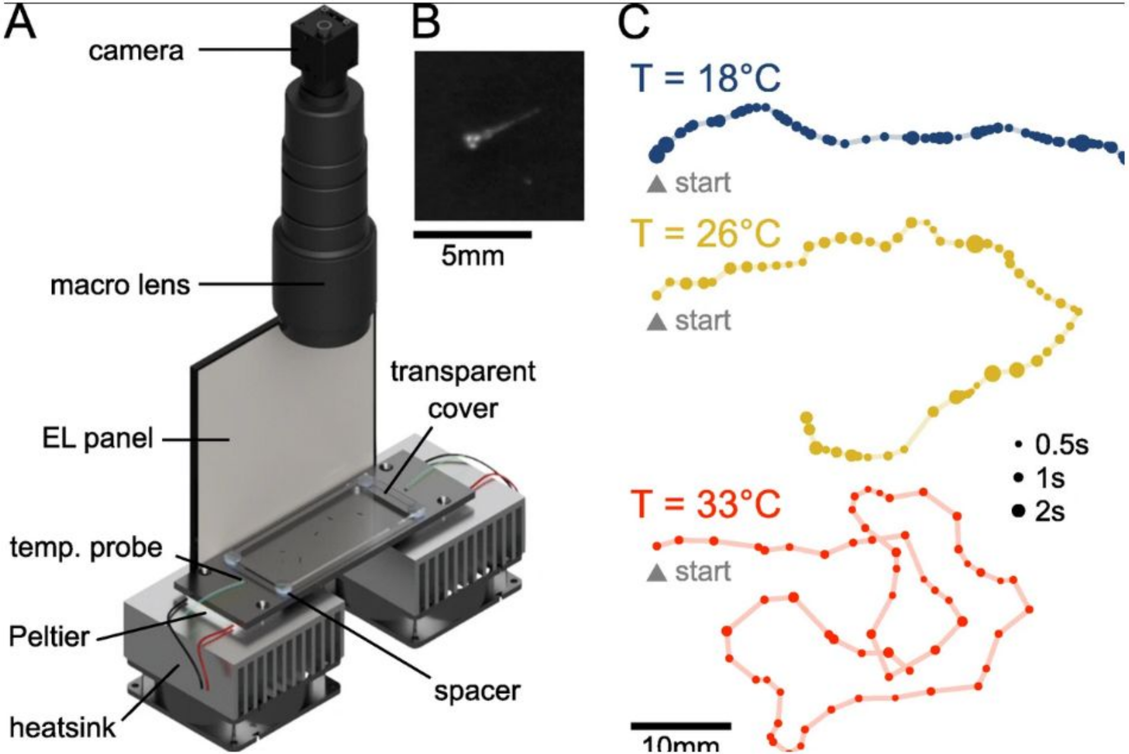
Finally, let's validate that our subgroups of neurons have a biological meaning by looking at their activity :



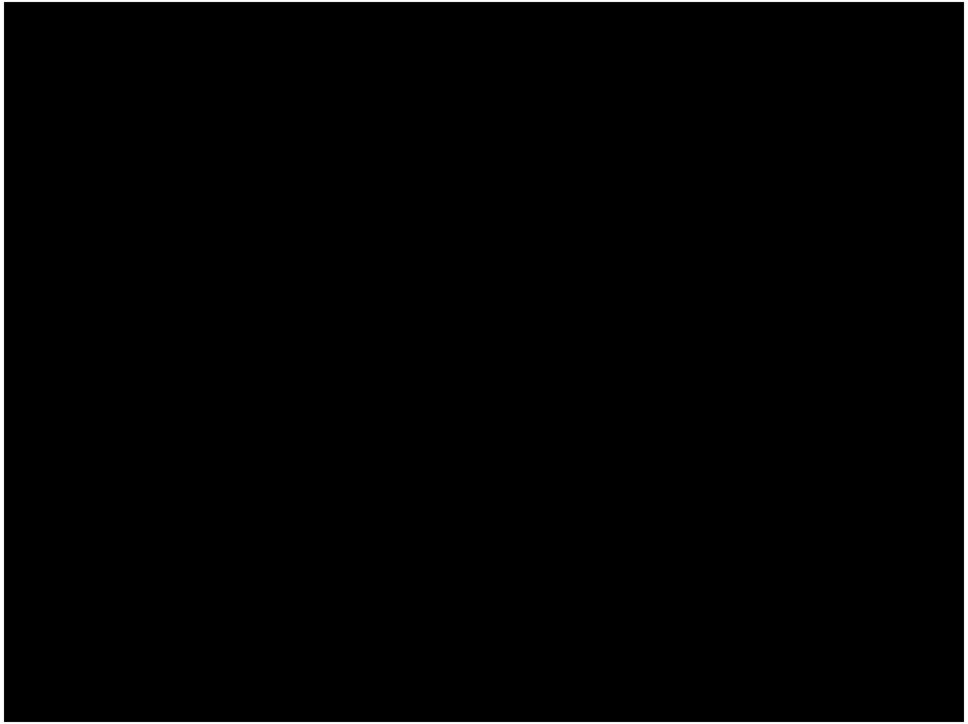


(The proper way to do this would be to plot the tuning curves between direction and average dFF)

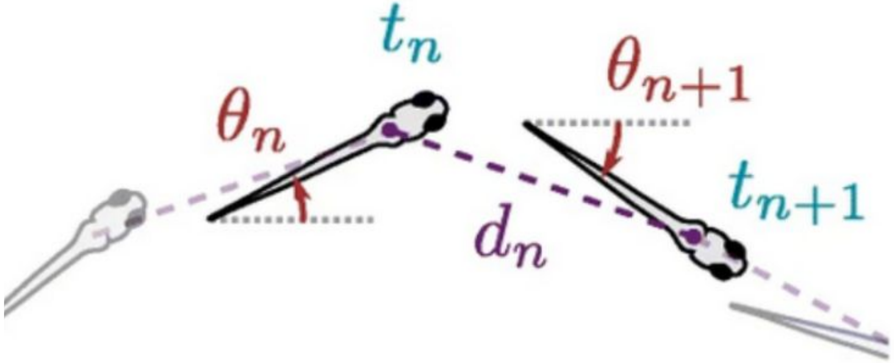
Day 4: Markov Chains (MCs) and Hidden Markov Models (HMMs)



Day 4: Markov Chains (MCs) and Hidden Markov Models (HMMs)

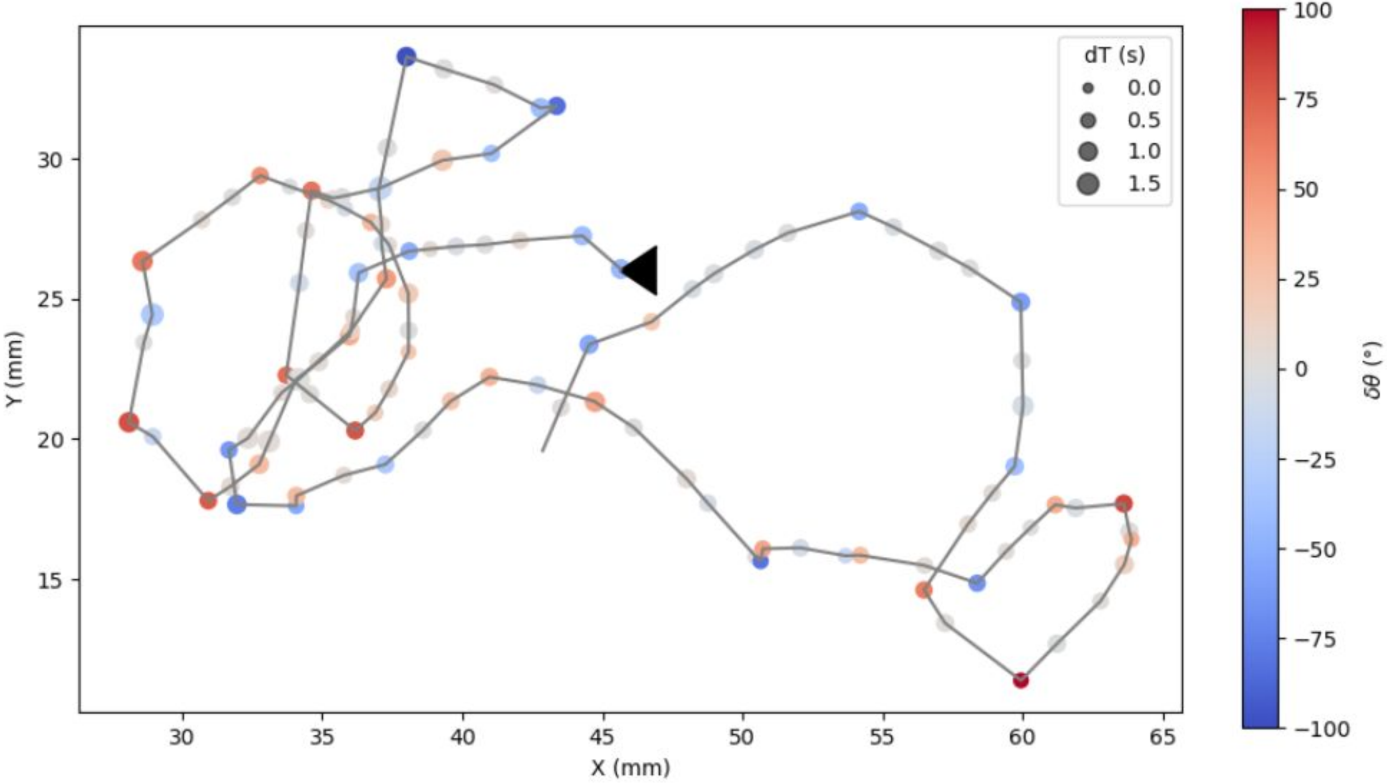


Day 4: Markov Chains (MCs) and Hidden Markov Models (HMMs)



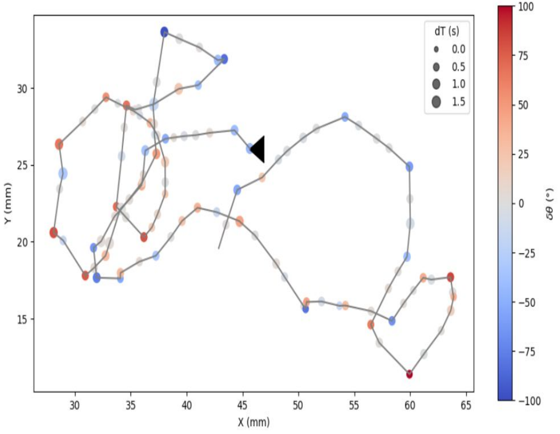
$$\delta t_n = t_{n+1} - t_n$$
$$d_n = \|\vec{r}_{n+1} - \vec{r}_n\|$$
$$\delta\theta_n = \theta_{n+1} - \theta_n \pmod{2\pi}$$

Day 4: Markov Chains (MCs) and Hidden Markov Models (HMMs)

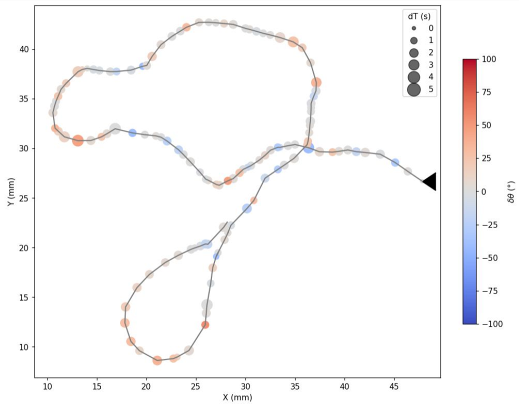


Day 4: Markov Chains (MCs) and Hidden Markov Models (HMMs)

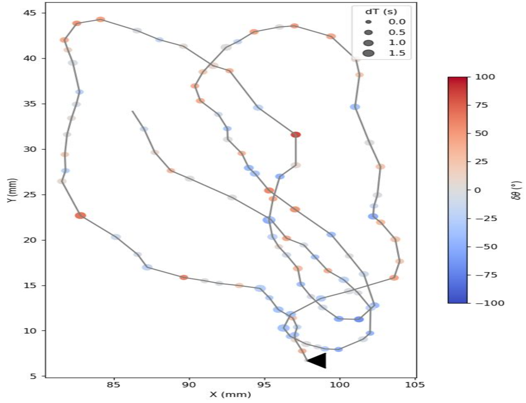
26°C



18°C



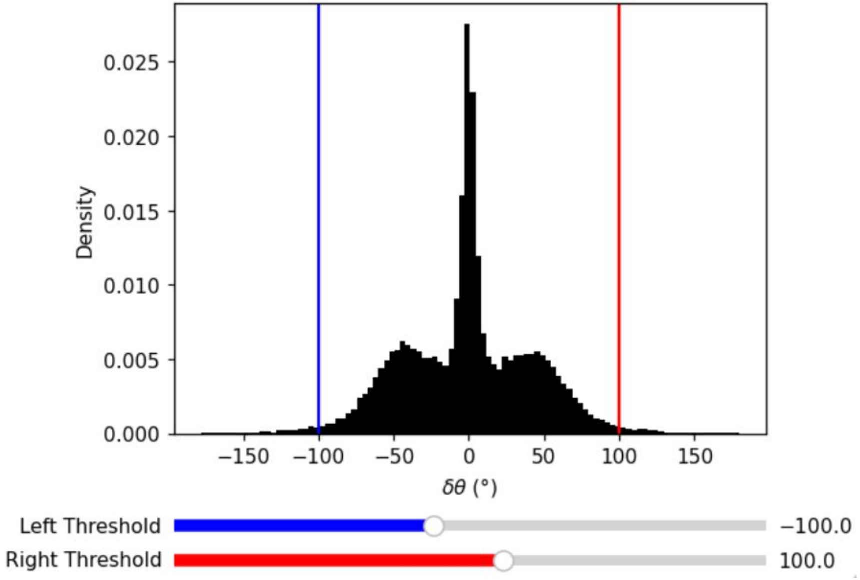
33°C



Day 4: **Markov Chains (MCs)** and Hidden Markov Models (HMMs)

1. Classifying Bouts

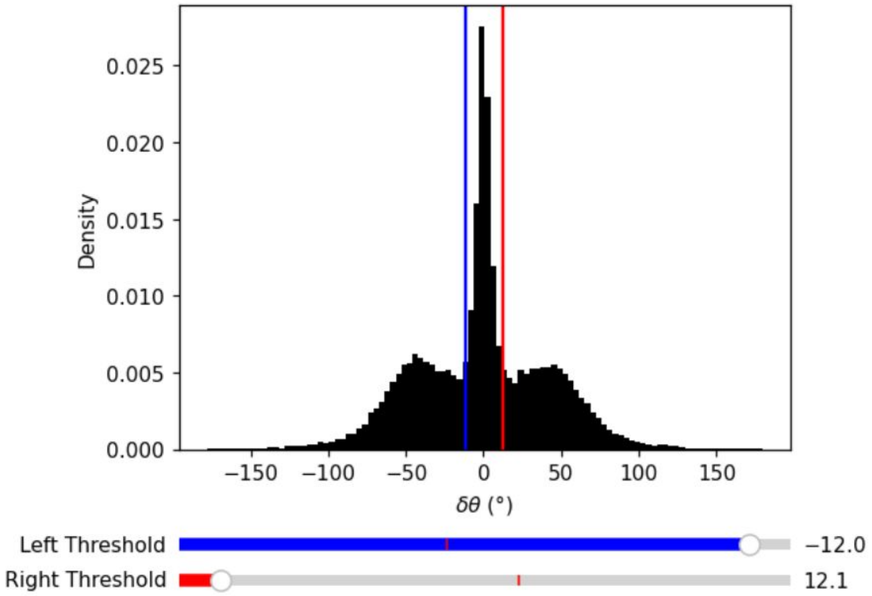
- if $\delta\theta \approx 0$: they will essentialy be going forward.
- if $\delta\theta < 0$: they will be going left.
- if $\delta\theta > 0$: they will be going right.



Day 4: **Markov Chains (MCs)** and Hidden Markov Models (HMMs)

1. Classifying Bouts

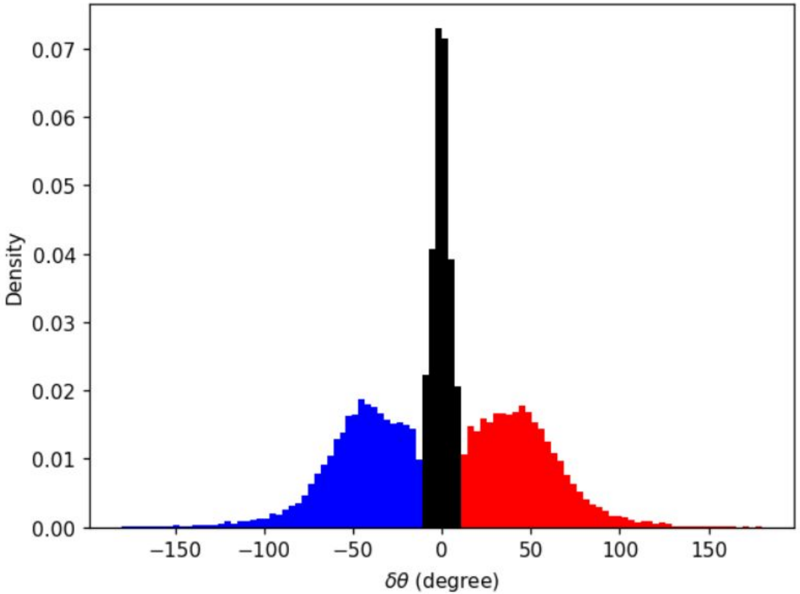
- if $\delta\theta \approx 0$: they will essentialy be going forward.
- if $\delta\theta < 0$: they will be going left.
- if $\delta\theta > 0$: they will be going right.



Day 4: **Markov Chains (MCs)** and Hidden Markov Models (HMMs)

1. Classifying Bouts

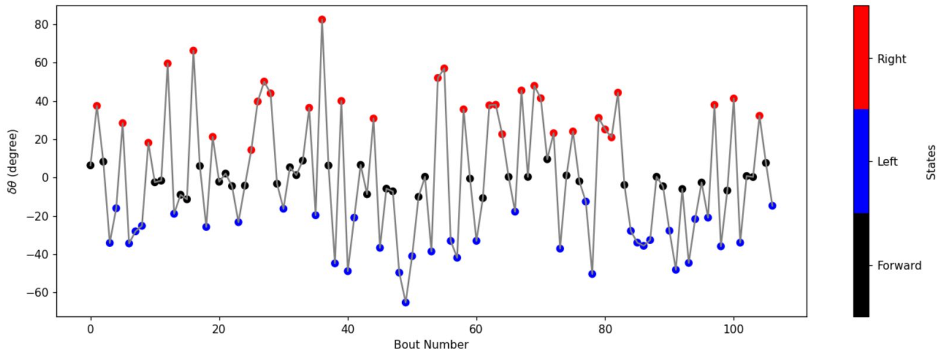
- if $\delta\theta \approx 0$: they will essentialy be going forward.
- if $\delta\theta < 0$: they will be going left.
- if $\delta\theta > 0$: they will be going right.



Day 4: **Markov Chains (MCs)** and Hidden Markov Models (HMMs)

1. Classifying Bouts

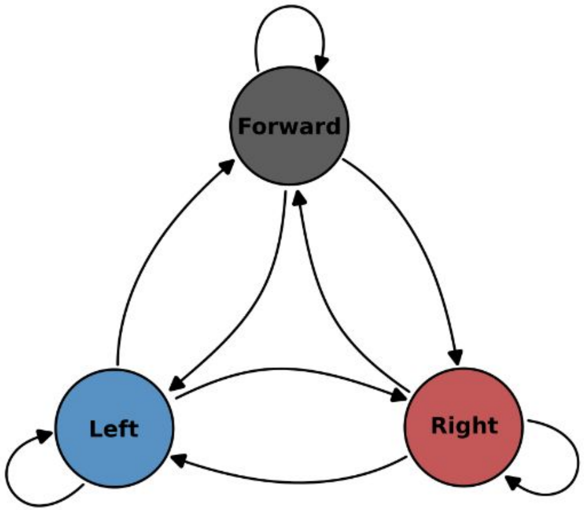
- if $\delta\theta \approx 0$: they will essentialy be going forward.
- if $\delta\theta < 0$: they will be going left.
- if $\delta\theta > 0$: they will be going right.

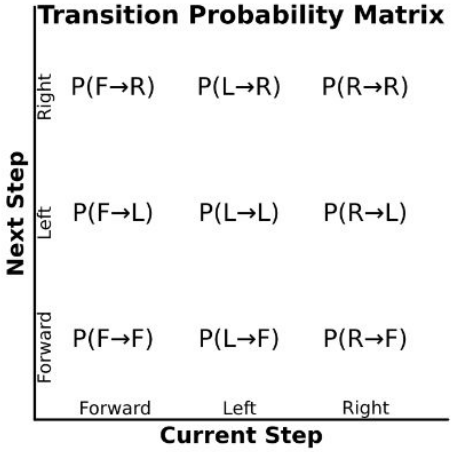


Day 4: **Markov Chains (MCs)** and Hidden Markov Models (HMMs)

2. Bout Probabilities and transitions

$$P(\text{next}|\text{current}) = P(\text{current} \rightarrow \text{next}) = P(c \rightarrow n)$$

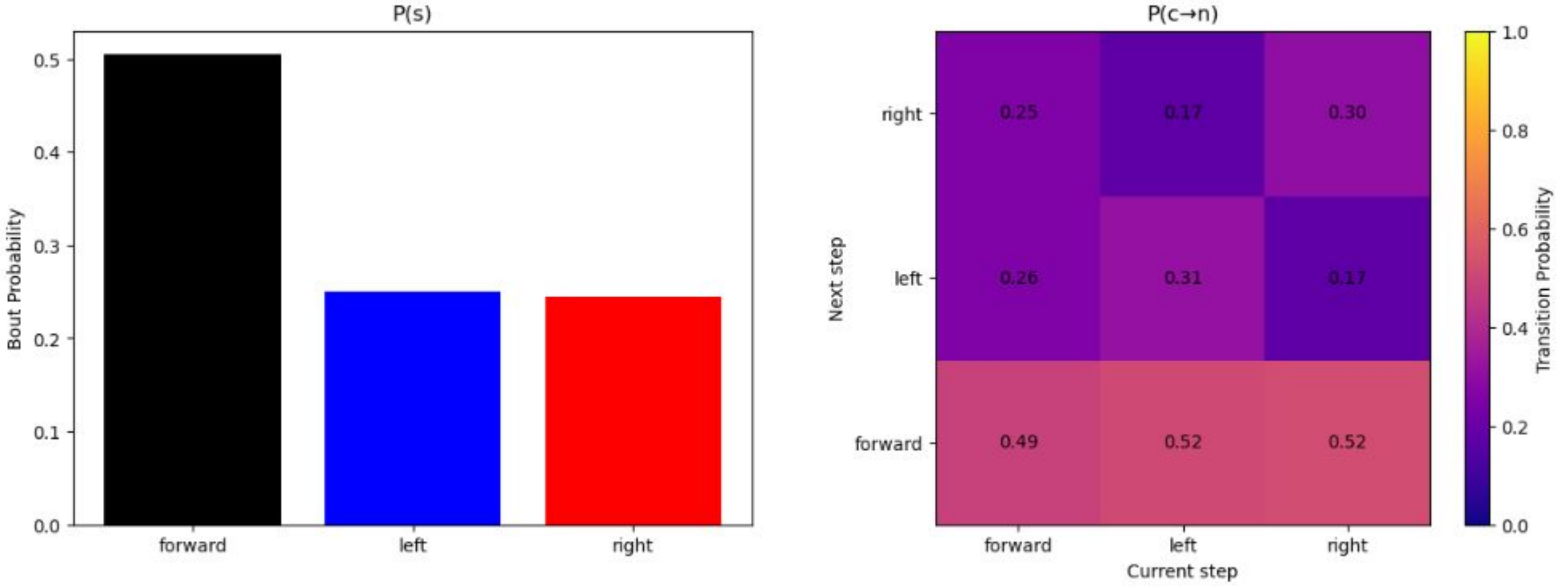




Normalisation : $\sum_{n \in F,L,R} P(c \rightarrow n) = 1 \quad \forall c \in F, L, R$

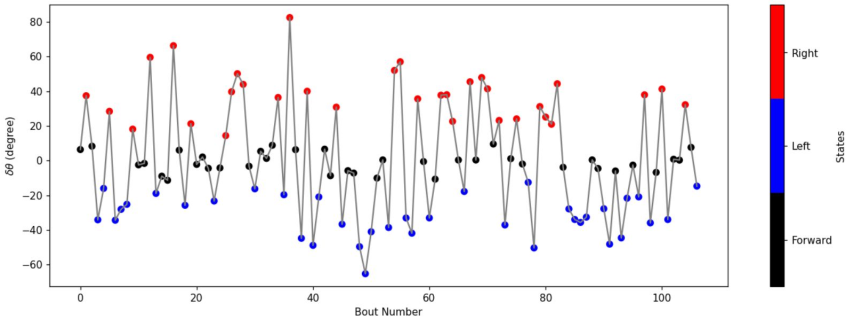
Day 4: **Markov Chains (MCs)** and Hidden Markov Models (HMMs)

2. Bout Probabilities and transitions

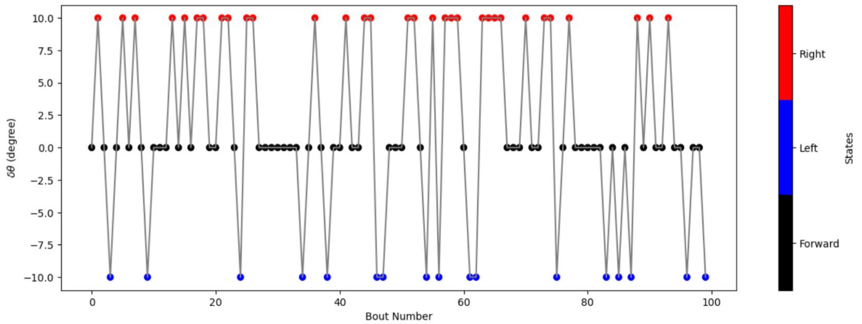


Day 4: **Markov Chains (MCs)** and Hidden Markov Models (HMMs)

Angle Sequence Generation



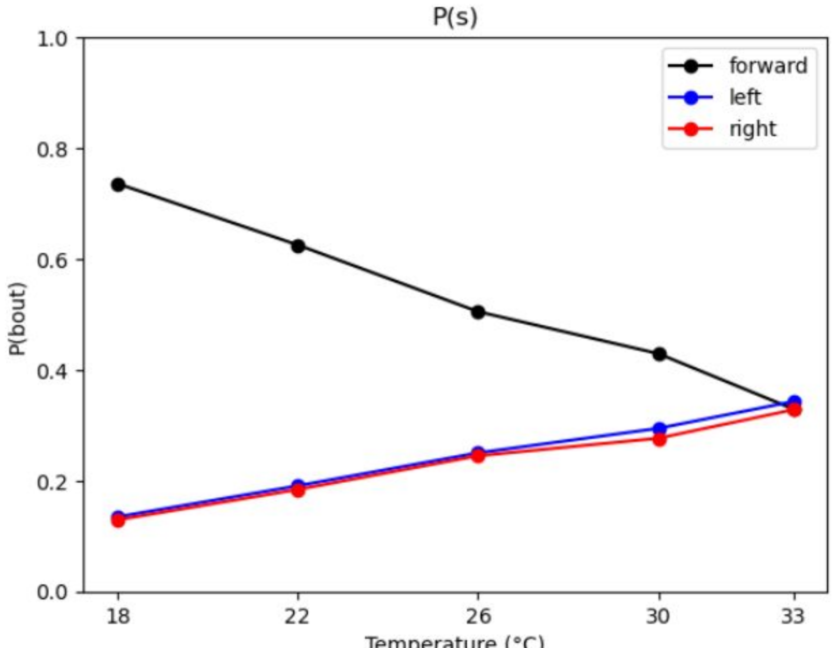
Data



Markov Model

Day 4: **Markov Chains (MCs)** and Hidden Markov Models (HMMs)

Bout Probabilities change according to the temperature

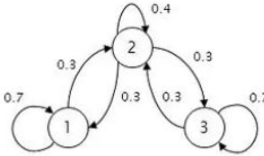


Day 4: Markov Chains (MCs) and **Hidden Markov Models (HMMs)**

1.Stationary distribution

πT=π   π : stationary distribution

- A Markov chain:



$$T = \begin{bmatrix} 0.7 & 0.3 & 0.0 \\ 0.3 & 0.4 & 0.3 \\ 0.0 & 0.3 & 0.7 \end{bmatrix}$$

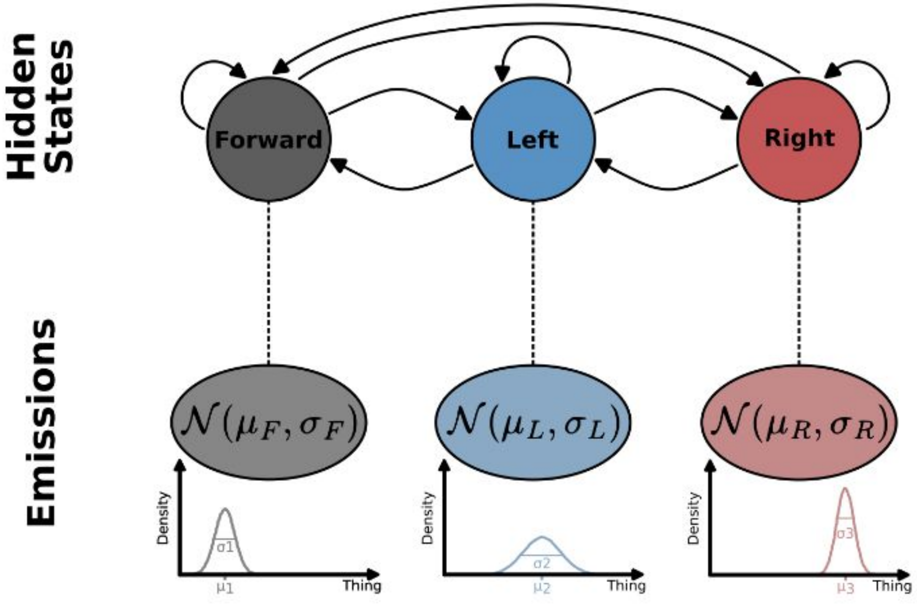
- The stationary distribution is $\underset{\sim}{\pi} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$

$$\begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 0.7 & 0.3 & 0.0 \\ 0.3 & 0.4 & 0.3 \\ 0.0 & 0.3 & 0.7 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$$

$$T^{\alpha} = T \times T \times \ldots \times T$$

$$\alpha \to \infty, T_c^{\alpha} = \pi \;\; \forall c.$$

Day 4: Markov Chains (MCs) and **Hidden Markov Models (HMMs)**



Transition Probability Matrix

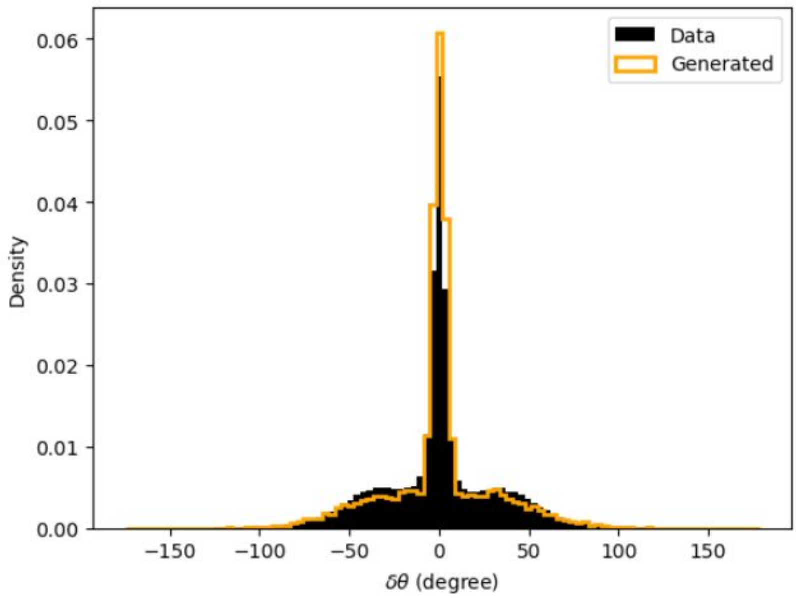
Next Step \ Current Step	Forward	Left	Right
Right	P(F→R)	P(L→R)	P(R→R)
Left	P(F→L)	P(L→L)	P(R→L)
Forward	P(F→F)	P(L→F)	P(R→F)

Emission Matrix

Hidden State	Emission Distributions
Forward	$\mathcal{N}(\mu_F, \sigma_F)$
Left	$\mathcal{N}(\mu_L, \sigma_L)$
Right	$\mathcal{N}(\mu_R, \sigma_R)$

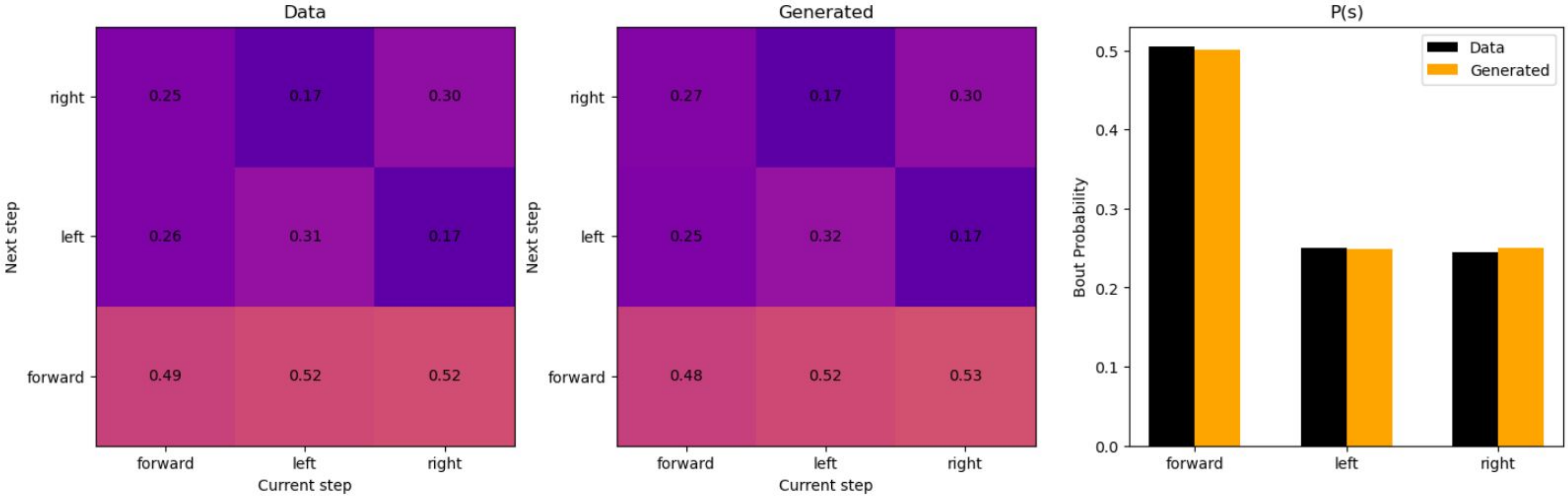
Day 4: Markov Chains (MCs) and **Hidden Markov Models (HMMs)**

2. Generating a distribution



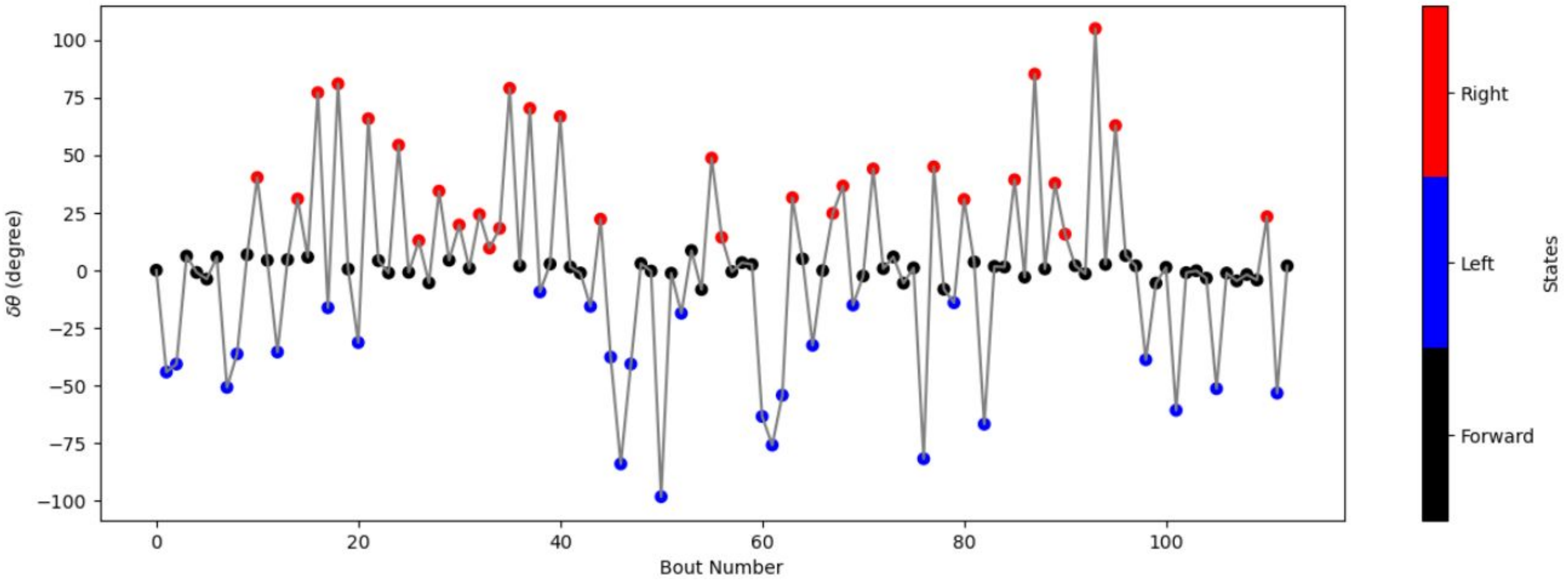
Day 4: Markov Chains (MCs) and **Hidden Markov Models (HMMs)**

3. Checking the a distribution



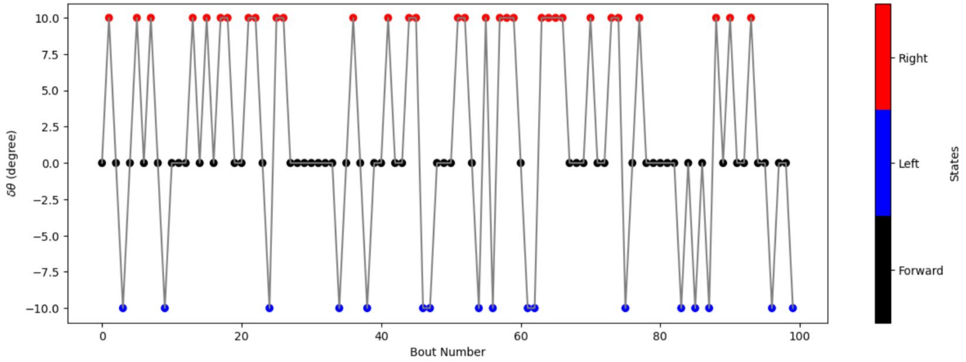
Day 4: Markov Chains (MCs) and **Hidden Markov Models (HMMs)**

3. Classification of the observations

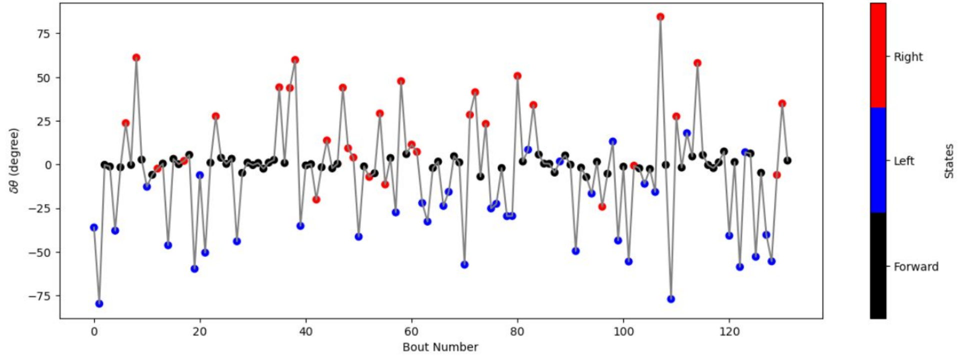


Day 4: Markov Chains (MCs) and **Hidden Markov Models (HMMs)**

4. Classification of the generated sequence



Markov Model

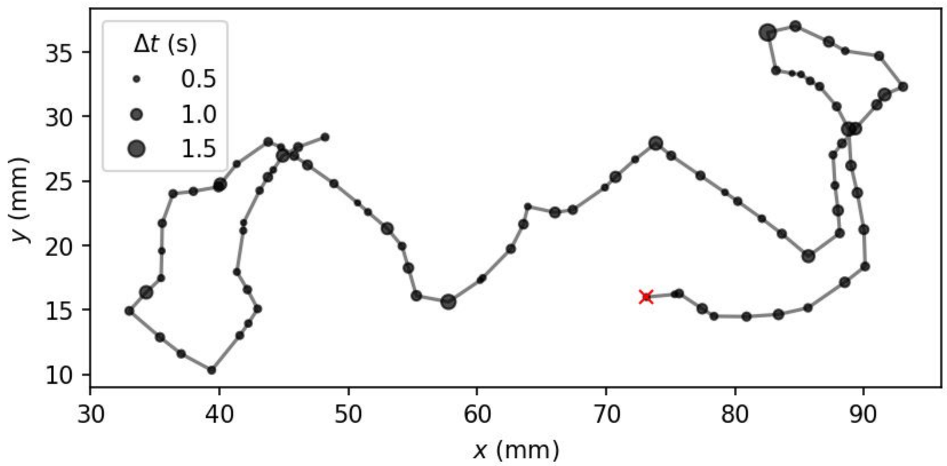


Hidden Markov Model

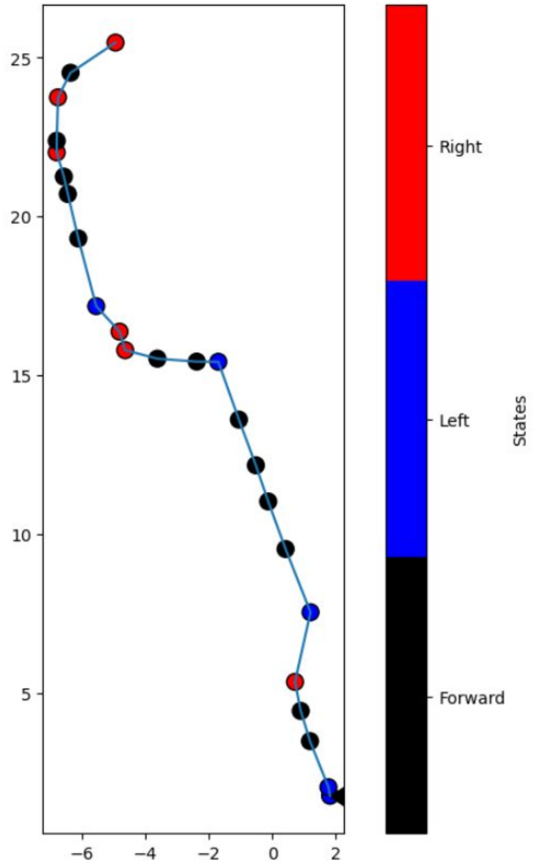
Day 4: Markov Chains (MCs) and **Hidden Markov Models (HMMs)**

5. Generate a trajectory (26°C)

Data



Generated



Day 4: Markov Chains (MCs) and **Hidden Markov Models (HMMs)**

6. Mean square reorientation (MSR)

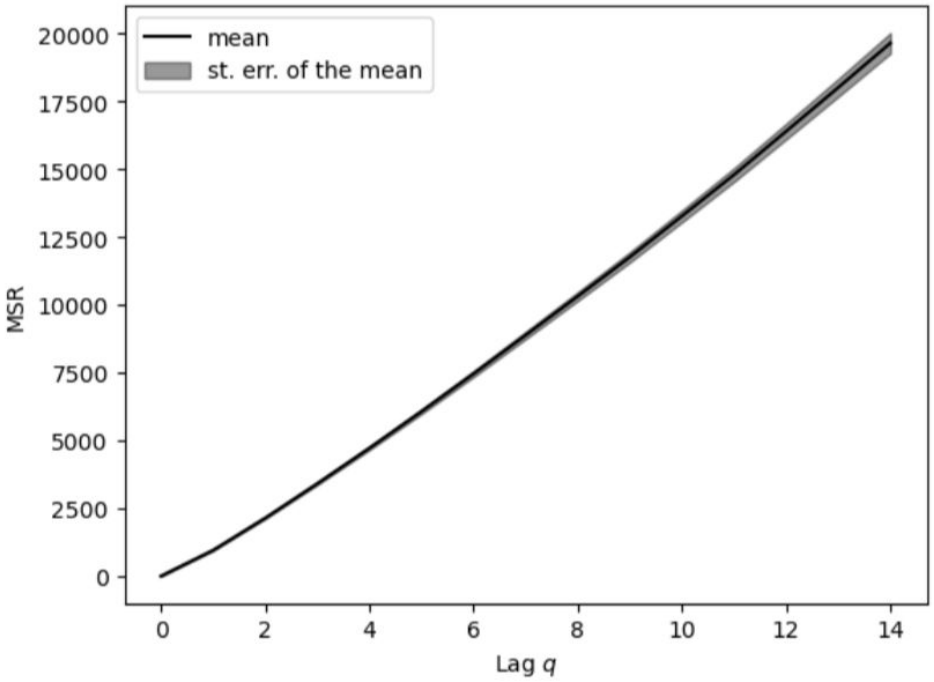
$$M_q = \left\langle (\theta_{n+q} - \theta_n)^2 \right\rangle_n$$

$$= \left\langle \left(\sum_{i=0}^{q-1} \delta\theta_{n+i} \right)^2 \right\rangle_n$$

$$= \frac{1}{N-q+1} \sum_{n=0}^{N-q} \left(\sum_{i=0}^{q-1} \delta\theta_{n+i} \right)^2$$

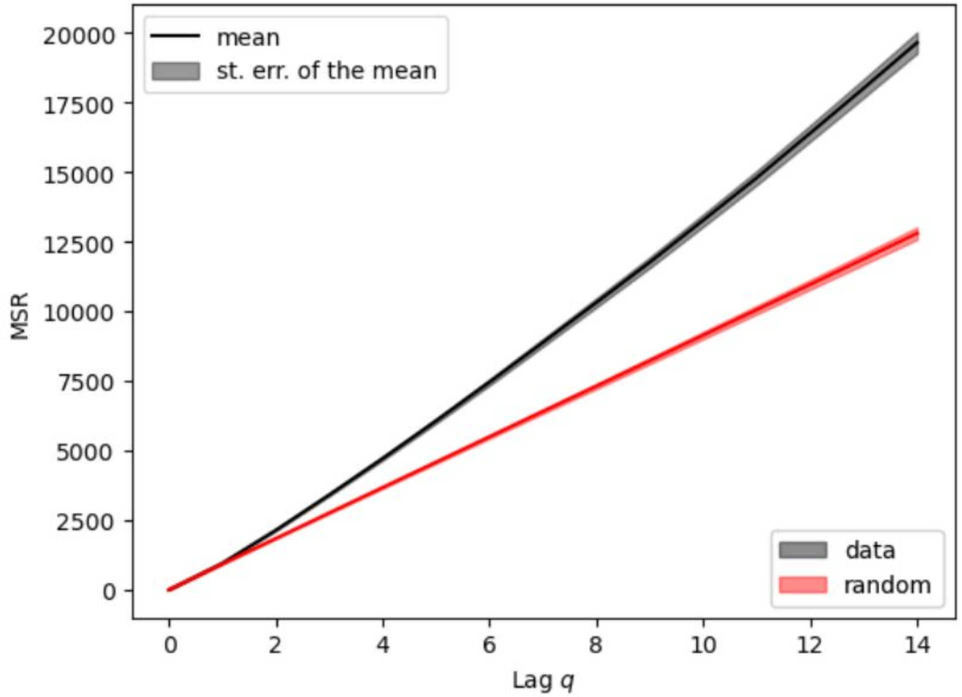
Day 4: Markov Chains (MCs) and **Hidden Markov Models (HMMs)**

6. Mean square reorientation (MSR)



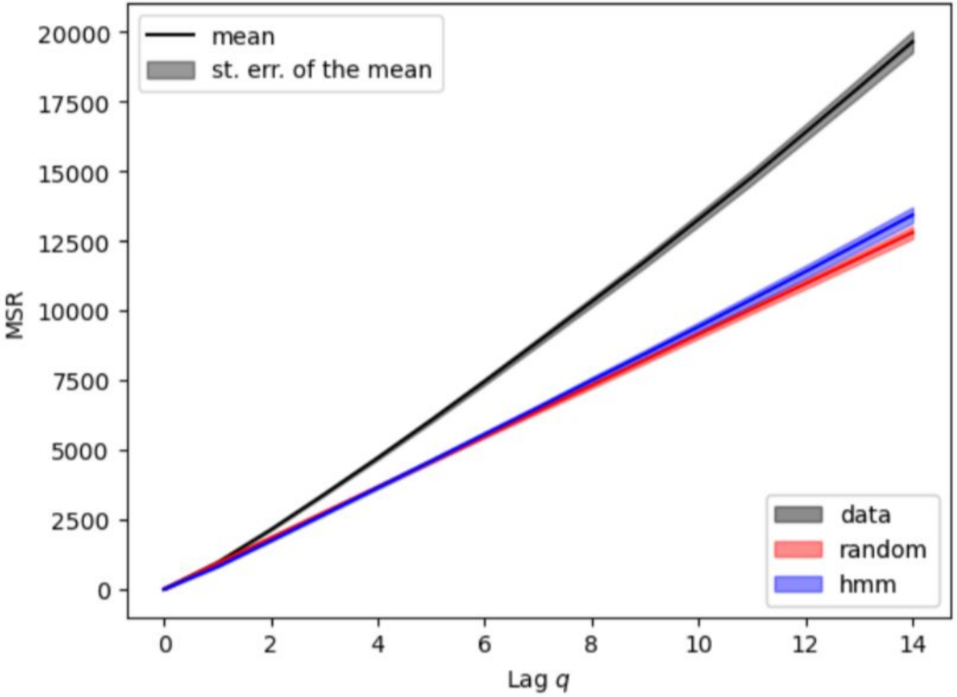
Day 4: Markov Chains (MCs) and **Hidden Markov Models (HMMs)**

6. Mean square reorientation (MSR)



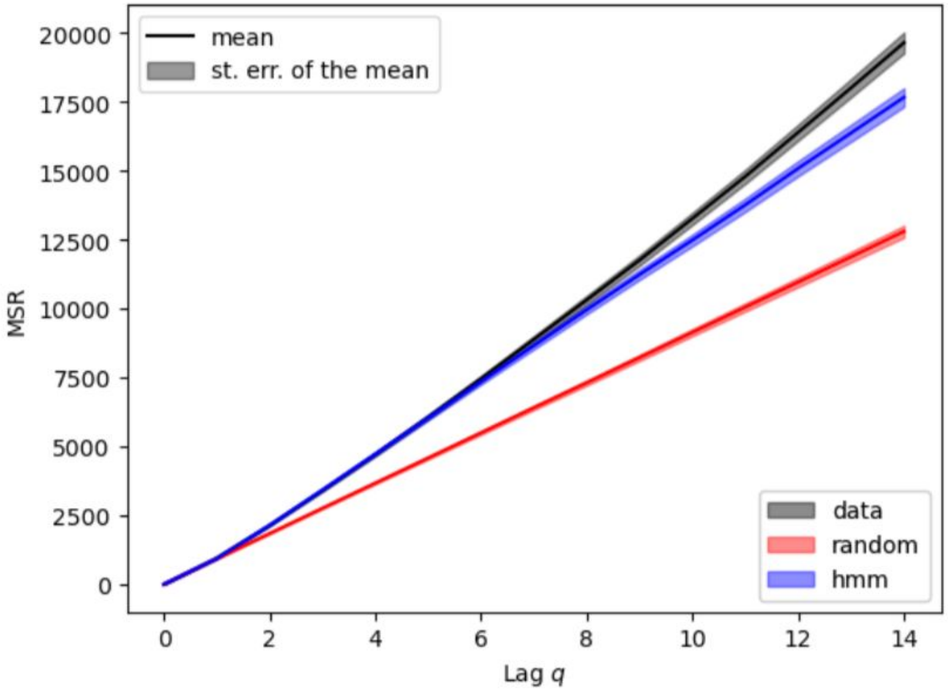
Day 4: Markov Chains (MCs) and **Hidden Markov Models (HMMs)**

6. Mean square reorientation (MSR)



Day 4: Markov Chains (MCs) and **Hidden Markov Models (HMMs)**

6. Mean square reorientation (MSR)



Thank you!