Master 1 internship report

Statistical methods for the functional clustering of single neuron activity

Victor Piriou
Supervised by Thibault Lagache (Biological Image Analysis Unit, Institut Pasteur, Paris)

From August 2 to September 10, 2021

Abstract

The goal of this internship was to study algorithms to identify, from the activity of single neurons, groups of neurons that tend to activate at the same time. Different approaches exist to solve this problem. One of them considers that this problem is analogous to the detection of communities in a graph. Identifying such ensembles is not obvious since the topological organization of the communities obtained depends strongly on the strategy employed by the chosen algorithm. Therefore, rather than simply applying generic community detection algorithms, a method that adapts to the nature of the problem has been designed and tested.

Acknowledgements

I would like to thank the Institut Pasteur and especially Mr. Lagache for his help and supervision throughout the internship as well as Mr. Olivo-Marin for having welcomed me.

Contents

Introduction

Context

The concept of neural ensembles (also called cell assemblies) was invented in 1949 by the Canadian neuropsychologist Donald Hebb to describe a network of neurons whose activation is synchronous. Furthermore, it has been shown that the activation of neural ensembles is correlated with various brain functions such as motor control or memory formation. Thus, neural ensembles are increasingly considered as the basis of neural code. To understand how neural networks represent and update information, it is necessary to know the composition and activity of these ensembles.

Since it is now possible to reliably quantify the activity of all the neurons of certain organisms such as the Hydra by calcium imaging, it seems important to have a method allowing to localize and characterize these neuronal ensembles from the activity signals. The neuronal discharges being strongly irregular, the direct correlation between signals is not exploitable. Therefore, we are only interested in the times at which the action potentials are emitted since they are highly informative.

However, the extraction of these ensembles is complex. Spike detection is prone to errors. All neurons can activate independently of the ensembles to which they belong. A neuron may not belong to any ensemble, in which case it contributes to the increase of noise in the dataset. When an ensemble is active, not all of its neurons are necessarily recruited. The ensembles can be of different sizes, have different activity rates. Although the majority of neurons that are not isolated belong to a single ensemble, a neuron can belong to several ensembles. It is therefore assumed that one ensemble cannot be totally included in another.

Contributions

This internship is a continuation of the IMA project carried out during the second semester. During this project, a simple and naive method of detecting the peaks of action potentials had been tested, some similarity metrics (such as cosine similarity) between the time series of neurons' activity had been compared and a hierarchical algorithm of community extraction - the Louvain algorithm - had been tested on graphs whose adjacency matrix is obtained by binarizing the similarity matrix.

However, the Louvain algorithm as such does not allow to detect overlapping communities. This internship has therefore allowed the development and testing of a new approach to detect overlapping ensembles. Moreover, a method based on Bayesian inference has been tested, although it does not allow any possibility of overlapping.

Chapter 1

Constitution of a ground truth

In order to be able to evaluate the clustering methods, a simulator of neural ensemble activities has been developed.

The input parameters are:

- the number of neurons;
- the number of ensembles;
- the proportion of neurons belonging to at least one ensemble;
- the maximum size that an ensemble can have;
- the minimum size that an ensemble can have;
- for each ensemble: the minimum proportion of neurons belonging to several ensembles;
- for each ensemble: the maximum proportion of neurons belonging to several ensembles;
- the number of time steps;
- the duration of a time step;
- the duration of an activation of an ensemble;
- the frequency of activation of the ensembles;
- the average probability that a neuron is activated when at least one of the ensembles to which it belongs is activated (this probability follows a normal distribution);
- the probability that a neuron will spontaneously activate when none of the ensembles to which it belongs is active.

For each neuron, the output data we are interested in are:

- the list of ensembles to which the neuron belongs;
- the series S_i representing the activity of neuron i over time. S_i^t is 1 if a peak of activity occurred at the t^{th} time step, otherwise S_i^t is 0.

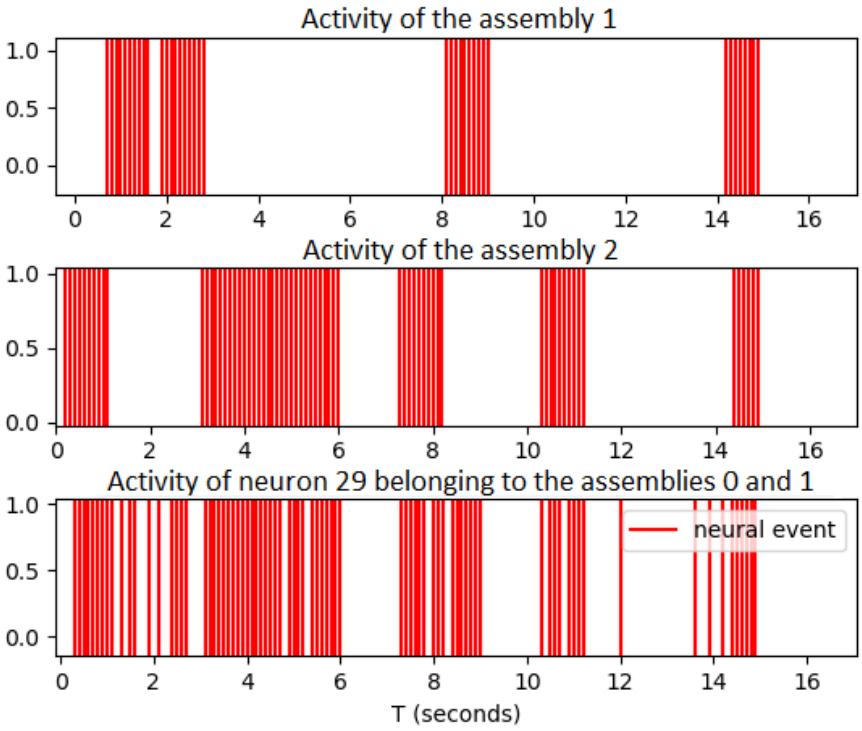

Figure 1.1: **Comparison between the activity of a neuron and the activity of its assemblies**. When at least one of the assemblies (to which neuron 29 belongs) is activated, the neuron is likely to activate.

Chapter 2

Descriptions of the methods experimented

2.1 Bayesian inference

Giovanni Diana *et al.* proposed a method [1] based on Bayesian inference to obtain the activity of assemblies and identify the neurons they contain. The method is robust and does not require any parameter. Nevertheless, it does not allow the possibility of assigning a neuron to more than one assembly. A new formulation would be needed to allow this method to assign to several assemblies. Moreover, it takes as input neurons that all belong to at least one ensemble. However, in reality, not all neurons are involved in assemblies and the short duration of the recording of neuron activities does not always allow to observe the activation of each assembly.

2.2 Novel algorithm

2.2.1 Summary

The method we have developed treats the problem sequentially in the sense that the activities of the assemblies are first estimated (from neurons belonging to a single assembly) and then exploited to classify the neurons belonging to several assemblies. It requires four parameters: α_1, α_2, α_3 and α_4.

Step 1 For each pair of neurons, we determine if their activities are significantly similar. The description of this step is detailed in section 2.2.2.

Step 2 From step 1, we deduce the adjacency matrix of a graph, which we construct. In this graph, there is a node for each neuron. Two nodes are therefore connected if the activities of the corresponding neurons are significantly similar.

Step 3 The nodes with a number of connections lower than α_1 are deleted.

Step 4 The neurons (corresponding to the nodes deleted in step 3) are classified as nodes not belonging to any assembly.

Step 5 The local clustering coefficient of each node is calculated. This step is detailed and justified in section 2.2.3.

Step 6 The nodes with a local clustering coefficient lower than a threshold α_2 are deleted. These nodes correspond, for the most part, to neurons belonging to several assemblies. This step is detailed and justified in section 2.2.3.

Step 7 The Louvain algorithm is then applied to the graph in order to partition it into communities. The number of communities obtained is greater than or equal to the number of neural assemblies to find.

Step 8 It is possible that, among the communities obtained in step 7, some have a number of nodes lower than α_3. The neurons of the nodes of such communities are classified as neurons not belonging to any assembly. These communities are removed from the list of obtained communities.

Step 9 At this stage, there are normally as many communities as there are neuronal assemblies. For each assembly, we now try to know the time steps during which it is active. By averaging the binary series of neurons of each of the communities found, we obtain, for each ensemble, a vector indicating the probability that the assembly is active at a given time step. By binarizing these vectors with a threshold α_4, we deduce, for each ensemble, the times when the assembly is active.

Step 10 Neurons not belonging to any assembly were classified in steps 3 and 8. Neurons belonging to a single assembly were classified in step 7. It now remains to classify the neurons that probably belong to several assemblies. They were identified in step 6. For each neuron i and each assembly j, we compare the activity frequency p_i^j of neuron i over the activity times (determined in step 9) of assembly j, against the activity frequency p_i^0 of neuron i outside the activity of any assembly. If p_i^j is significantly greater than p_i^0 then neuron i is classified as a neuron belonging to assembly j. For each neuron, the hypothesis of belonging to a assembly is therefore treated independently for each assembly. The description of this step is detailed in section 2.2.4.

2.2.2 Identification of statistically coupled neural activities

A simple way to test whether two binary series are statistically coupled is to compare the number of concurrent spikes of these two series at a certain threshold.

The probability that k spikes from a neuron 2 are concurrent with spikes from a neuron 1 is equal to :

$$P(k) = C_k^{n_2}(\tfrac{n_1}{T})^k(\tfrac{1-n_1}{T})^{n_2-k},$$

where:

- T is the number of time steps;
- n_1 is the number of spikes of neuron 1;
- n_2 is the number of spikes of neuron 2.

The p-value of the test "there are at least k spikes in common" is:

$$\text{p-value} = \sum_{i=1}^{k-1} P(k).$$

Nevertheless, this test is naive in the sense that it considers that the spikes of each neuron are independent of each other (*i.e.* there is no spike burst or variation between the intensities of the action potentials).

Another test of independance [2] (thought in 2D but adapted in 1D), not making the hypothesis of independence between the spikes, was tested. Since it is much more expensive in terms of computation time and the performance of the naive test is good enough, this last test was used to evaluate the performance of the method.

2.2.3 Identification of neurons that belong to several assemblies

Then, the nodes corresponding to neurons that belong to several communities are also removed. To identify such a node, we rely on its local clustering coefficient C_i which is defined for a node i by the proportion between the number of edges that exist between its neighbors divided by the number of edges that could potentially exist between its neighbors:

$$C_i = \frac{2\left|\left\{e_{jk}:\ (v_j,v_k)\in N_i, e_{jk}\in E\right\}\right|}{k_i(k_i-1)},$$

where:

- k_i is the number of vertices included in the set N_i of neighbors of vertex i;
- E is the set of edges of the graph.

The clustering coefficient of a node is correlated to the probability that the neuron of this node belongs to several neural assemblies. Indeed, if the detection of spikes as well as the determination of statistically coupled activities are perfect, then the vertices of the communities form cliques[1] in the graph since the neurons of the same assembly are all supposed to have a similar activity. Therefore, we can assume that the vertices in the same assembly tend to have an clustering coefficient close to 1 while the vertices at the intersection of several assemblies have a much lower clustering coefficient.

The nodes with an clustering coefficient lower than a threshold α_1 are therefore deleted. This threshold is related to the confidence we have in the test of independence of the binary series.

[1] Subsets of vertices of a graph all adjacent to each other.

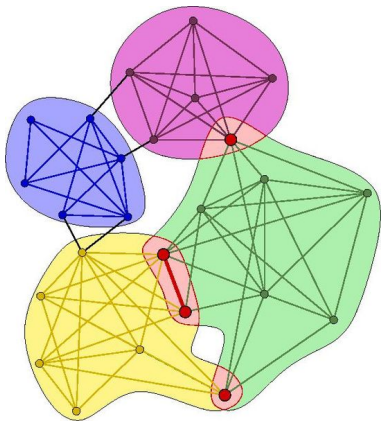

Figure 2.1: **Graph with a typical structure of graphs representing neural ensembles**. The red nodes, belonging to several communities, have a significantly lower clustering coefficient than most nodes belonging to a single community, whose clustering coefficient is close to 1.

2.2.4 Classification of neurons that belong to several assemblies

A neuron that belongs to an ensemble is more active when the assembly to which it belongs is active.

To classify neurons that belong to several ensembles, we can therefore perform the test whose null hypothesis is:

$$\text{H0: } p_i^j = p_i^0,$$

where:

- p_i^j is the frequency of activity of neuron i on the n_j time steps where the assembly j is active;
- p_i^0 is the frequency of activity of neuron i on the n_0 time steps where all assemblies are inactive.

Testing the difference between these two proportions is the same as testing whether this difference follows a normal distribution of zero mean. When we divide this difference by its standard deviation, we obtain a random variable T that follows a centered reduced normal distribution:

$$T = \frac{p_i^j - p_i^0}{\sqrt{p(1-p)\left(\frac{1}{n_j} + \frac{1}{n_0}\right)}} \sim \mathcal{N}(0;1),$$

with:

$$p = \frac{n_j p_i^j + n_0 p_i^0}{n_j + n_0}.$$

Chapter 3

Performance evaluation of the methods

3.1 Test conditions

For each assembly, the proportion of neurons belonging to at least one other assembly follows a uniform distribution between 0.1 and 0.3. The probability that a neuron activates, when at least one of the assemblies to which the neuron belongs is activated, follows a normal distribution centered in 0.7 and of standard deviation 0.2. The probability that a neuron activates when none of the ensembles to which it belongs is active, is assembly to 0.05. The duration of an activation of an assembly is fixed to 1 second. A assembly is active during 1% of the time. A time step lasts 0.1 second and each simulation lasts 150 seconds. The size of each assembly varies randomly between a and b such that:

$$a = 0.8 \times \left(\frac{\text{total number of neurons} - \text{number of neurons not belonging to any assembly}}{\text{number of assemblies}}\right),$$

$$b = 1.2 \times \left(\frac{\text{total number of neurons} - \text{number of neurons not belonging to any assembly}}{\text{number of assemblies}}\right).$$

3.2 Bayesian inference

Although the Bayesian inference-based method assigns each neuron to a single assembly, it was tested on simulations of partially overlapping assemblies activity to assess its ability to classify neurons that belong to multiple assemblies into at least one of the assemblies to which those neurons are expected to belong. All neurons in these simulations belong to at least one assembly.

3.2.1 Evaluation measure used

To evaluate the accuracy of a group of neurons, the simulator assignments are compared with the assignments obtained with the tested algorithm.

This performance is defined by:

$$\text{Performance } (\mathcal{A}, \mathcal{A}') = \frac{1}{|L|} \sum_{i \in L} I_i \ ,$$

with:

$$I_i = \begin{cases} 1 & \text{if } A_i \cap A'_i \neq \emptyset \\ 0 & \text{if } A_i \cap A'_i = \emptyset \end{cases},$$

where:

- $\mathcal{A} = \{A_i \mid 0 \leq i < N\}$ is the set of assignments of the simulator;
- N is the number of observed neurons;
- $\mathcal{A}' = \{A'_i \mid 0 \leq i < N\}$ is the set of assignments obtained with the algorithm;
- L is the set of identifiers of the neurons whose accuracy we want to evaluate.

3.2.2 Results

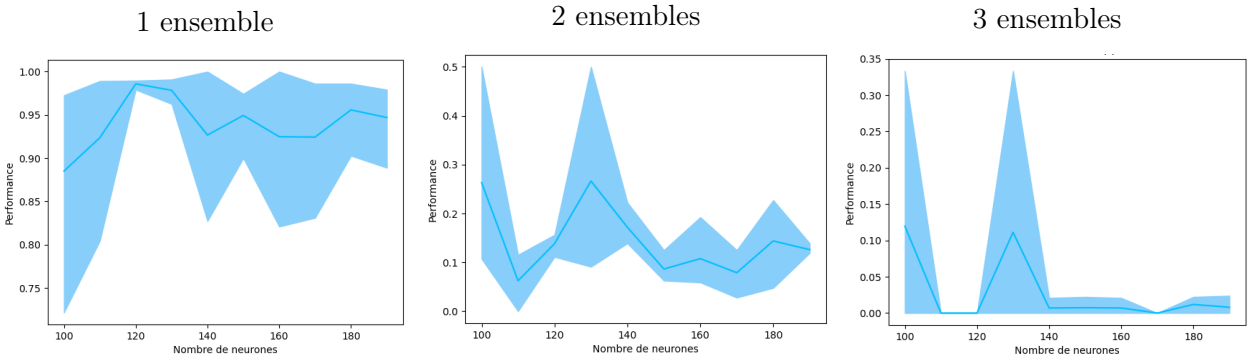

Figure 3.1: **Performance of groups of neurons belonging to 1, 2 and 3 assemblies.** The number of assemblies in this simulation is set to 3. The line is the average over 3 iterations.

The algorithm is very good at finding neurons that belong to a single ensemble. However, it struggles to classify neurons belonging to several assemblies. It seems to detect as many communities as there are possible combinations of assemblies.

3.3 Novel algorithm

The proportion of neurons belonging to at least one assembly follows a uniform distribution between 0.6 and 0.9. The nodes with a number of connections lower than $\alpha_1 = 3$ have been removed. The nodes with an agglomeration coefficient lower than $\alpha_2 = 0.62$ have been removed from the graph. The communities with a number of nodes lower than $\alpha_3 = 5$ were not considered. The threshold α_4 used to binarize the activity vectors of the assemblies is set to 0.3.

3.3.1 Evaluation measure used

To evaluate the accuracy of a set of neurons, the simulator assignments are compared with the assignments obtained with the tested algorithm.

This performance is defined by:

$$\text{Performance } (\mathcal{A}, \mathcal{A}') = \frac{1}{|L|} \sum_{i \in L} \frac{\left|A_i \cap A'_i\right|}{\left|A_i \cup A'_i\right|},$$

3.3.2 Results

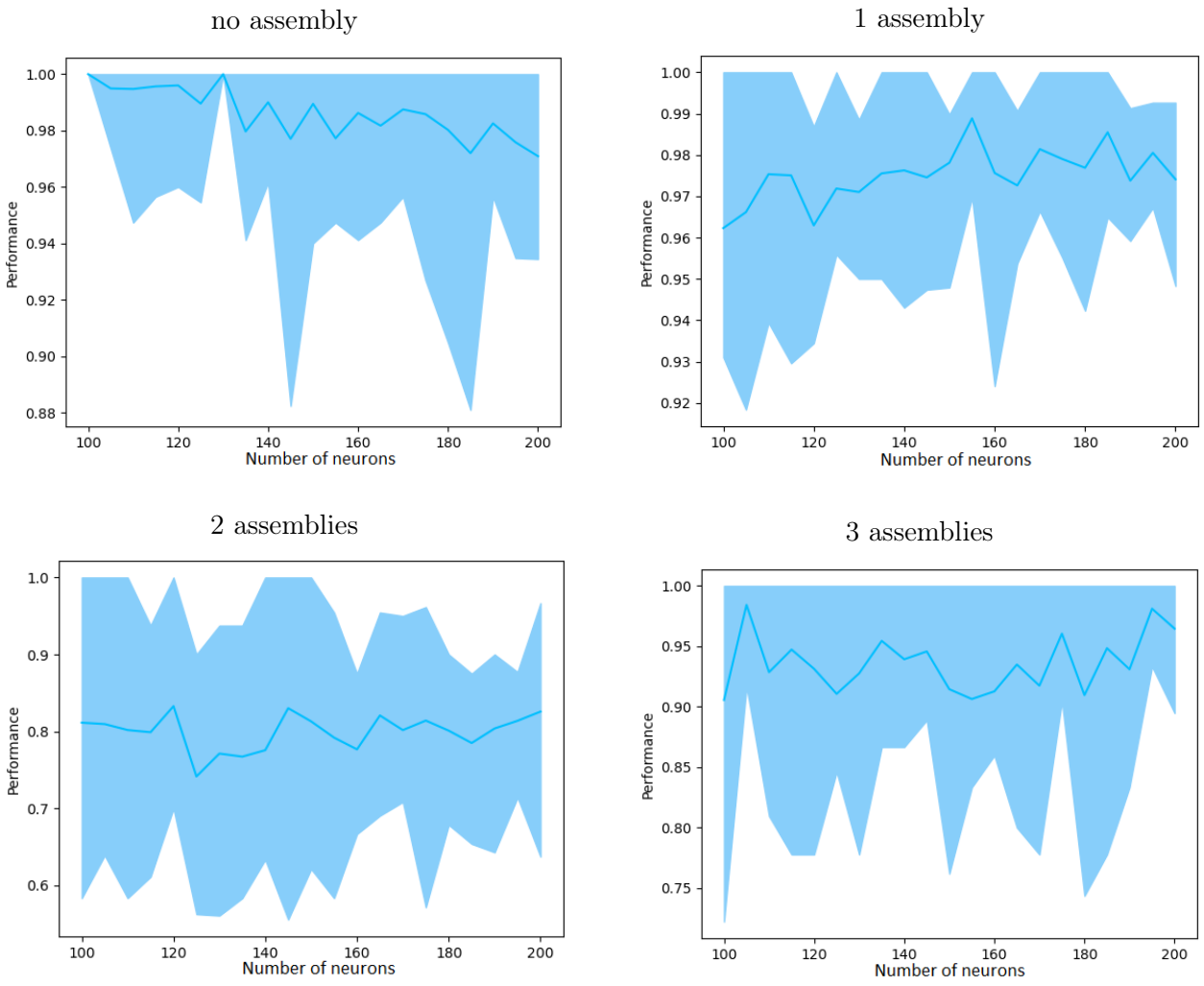

Figure 3.2: **Performance of groups of neurons belonging to 0, 1, 2 and 3 assemblies.** The number of asemblies in this simulation is set to 3. The line is the average over 10 iterations. The average performance is excellent. The same test was performed on simulations of 2, 4 and 5 assemblies, the results are also excellent. Nevertheless, when the estimation of the number of assemblies is wrong, the results are much less satisfactory.

Conclusion and perspectives

The objective of this work was to investigate methods for extracting neural ensembles from the activity of single neurons. A method based on Bayesian inference was tested in order to evaluate its ability to classify neurons that belong to several assemblies, into one of the assemblies to which they are supposed to belong. A method based on the Louvain algorithm was developed and tested.

In the continuity of this work, it would be a question of comparing the developed method with the other state of the art methods concerning the detection of overlapping neural ensembles in order to know if this new method really brings something new in terms of reliability and speed for example.

Appendix A

Methodological trials

A.1 Alternative to the statistical test

An alternative to hypothesis testing is to search, for each neuron to be classified, the A^* assignment that maximizes the likelihood:

$$A^* = \underset{A}{\operatorname{argmax}}\, \mathcal{L}(S_i \mid A) = \prod_{j=1}^{T} (p_A^j)^{S_i^j} \cdot (1 - p_A^j)^{1 - S_i^j},$$

where:

- p_A^i is the probability that at least one of the assemblies of A is active at time i;
- S_i is the binary series of the neuron i that we want to classify;
- T is the number of time steps of each series.

This method works very well provided that we have a high number of neurons per assembly so that the estimation of ensembles activities is sufficiently reliable.

A.2 Another way to identify neurons belonging to a single community

The first idea that was tested to identify the « cores » of the communities was inspired by a Monte Carlo algorithm based on the Louvain algorithm. The idea is to run the algorithm several times and to swap the order of the nodes in which they are evaluated when optimizing the modularity of the graph (which is the goal of the Louvain algorithm). By doing this, we obtain different partitions at each execution. This allows us to deduce a matrix $\mathbf{P} = [p_{ij}]_{n \times n}$ (n being the number of neurons) giving the probability that the pair of nodes i and j are in the same partition. For example, if $p_{ij} = 1$ it means that nodes i and j always appear in the same community. On the contrary, $p_{ij} = 0$ means that nodes i and j never appear in the same community. By binarizing $\mathbf{P} = [p_{ij}]_{n \times n}$ with a threshold α, we transform this matrix

into an adjacency matrix of a new graph. The core of a community is the largest connected component of this new graph in this community (the one of the partition with the highest modularity).

This idea was abandoned since running the Louvain algorithm several times on our graph (while making sure that the order of the nodes is randomly swapped) always gave the same partition.

Appendix B

Attempt to evaluate another method

Geoffrey J. Goodhill *et al.* proposed another method [3] to extract overlapping neural ensembles. This method, based on spectral partitioning, has been tested and compared with our algorithm on some simulations. It seems to give very similar results to our algorithm. More time would have been needed to test it more rigorously.

Bibliography

[1] Giovanni Diana, Thomas T. J. Sainsbury, and Martin P. Meyer. "Bayesian inference of neuronal assemblies". en. In: *PLOS Computational Biology* 15.10 (Oct. 2019). Publisher: Public Library of Science, e1007481. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1007481. URL: https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1007481 (visited on 09/09/2021).

[2] Frédéric Lavancier et al. "Testing independence between two random sets for the analysis of colocalization in bioimaging". en. In: *Biometrics* 76.1 (2020). _eprint: https://onlinelibrary.wiley.com pp. 36–46. ISSN: 1541-0420. DOI: 10.1111/biom.13115. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/biom.13115 (visited on 09/07/2021).

[3] Jan Mölter, Lilach Avitan, and Geoffrey J. Goodhill. "Detecting neural assemblies in calcium imaging data". In: *BMC Biology* 16.1 (Nov. 2018), p. 143. ISSN: 1741-7007. DOI: 10.1186/s12915-018-0606-4. URL: https://doi.org/10.1186/s12915-018-0606-4 (visited on 09/09/2021).