Community detection in van Rossum network

Indra Ghosh, Áine Byrne

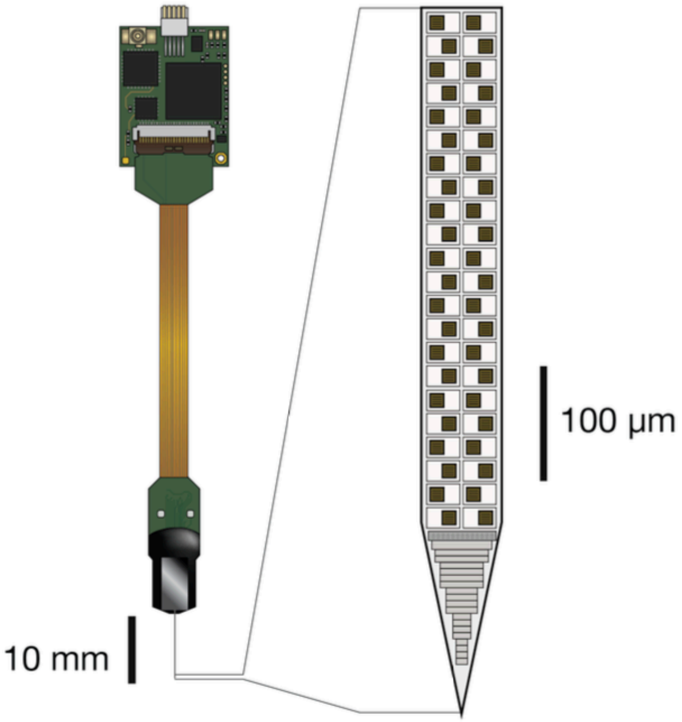
https://indrag49.github.io

13 May, 2026

University College Dublin

Email: indra.ghosh@ucd.ie

Neuropixels data



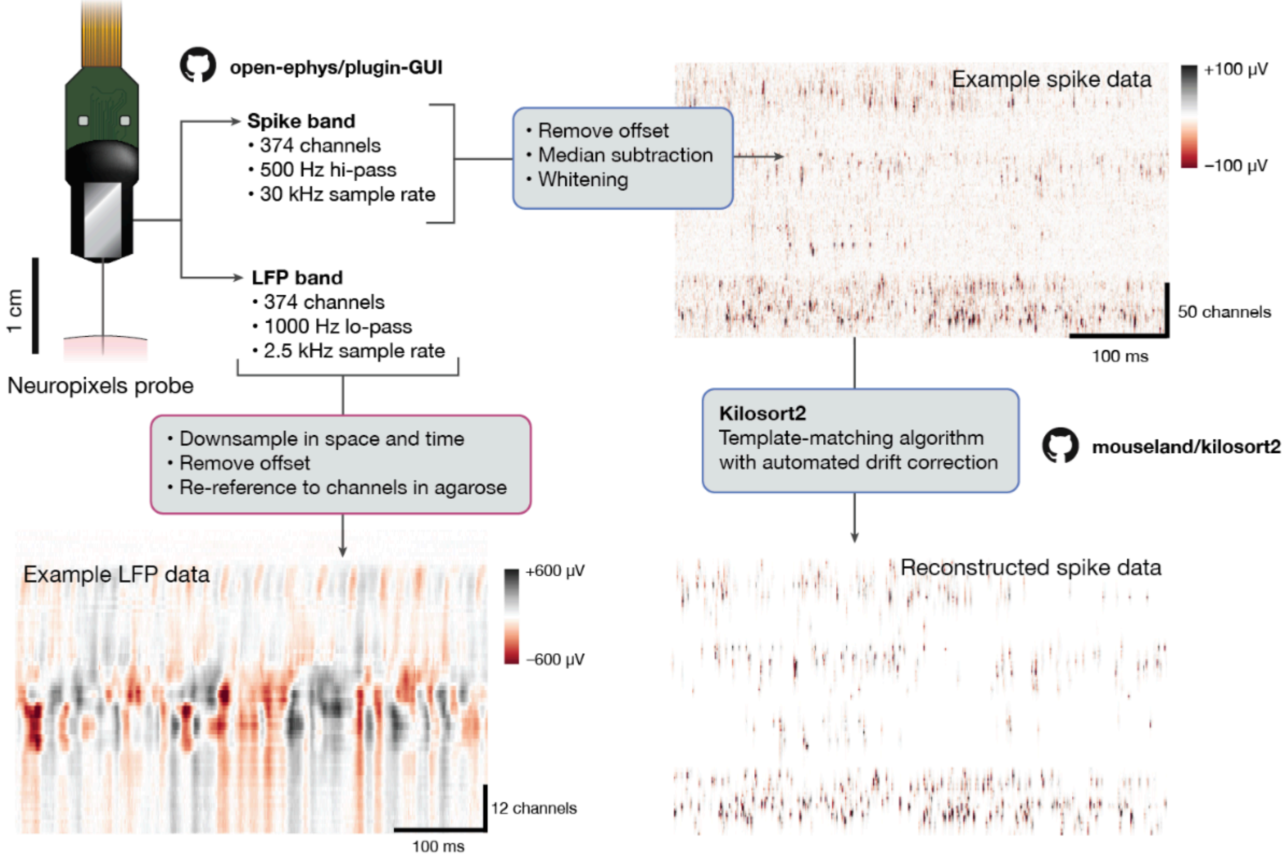


Image credit: Allen Institute for Brain Science. [https://brain-map.org/our-research/circuits-behavior/visual-coding], [https://allensdk.readthedocs.io/en/latest/visual_coding_neuropixels.html]

van Rossum distance

1. We convolve a discrete spike train data

$$\mathscr{K}(t) = \exp\left(-\frac{t}{t_R}\right) H(t),$$

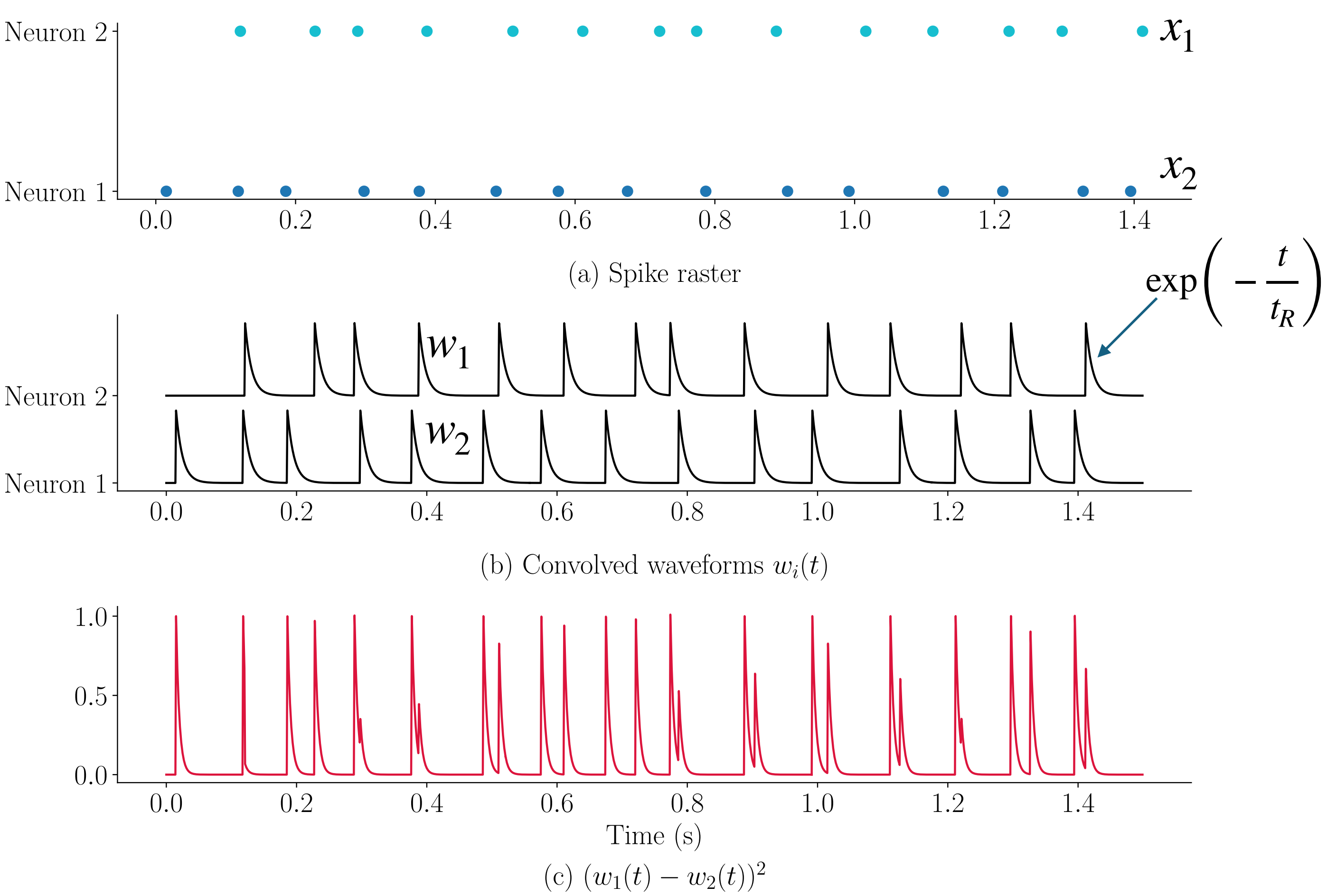
2. Convolution:

$$w_i(t) = \sum_k \exp\left(-\frac{t - t_i^k}{t_R}\right) H(t - t_i^k)\,.$$

3. Pairwise distance:

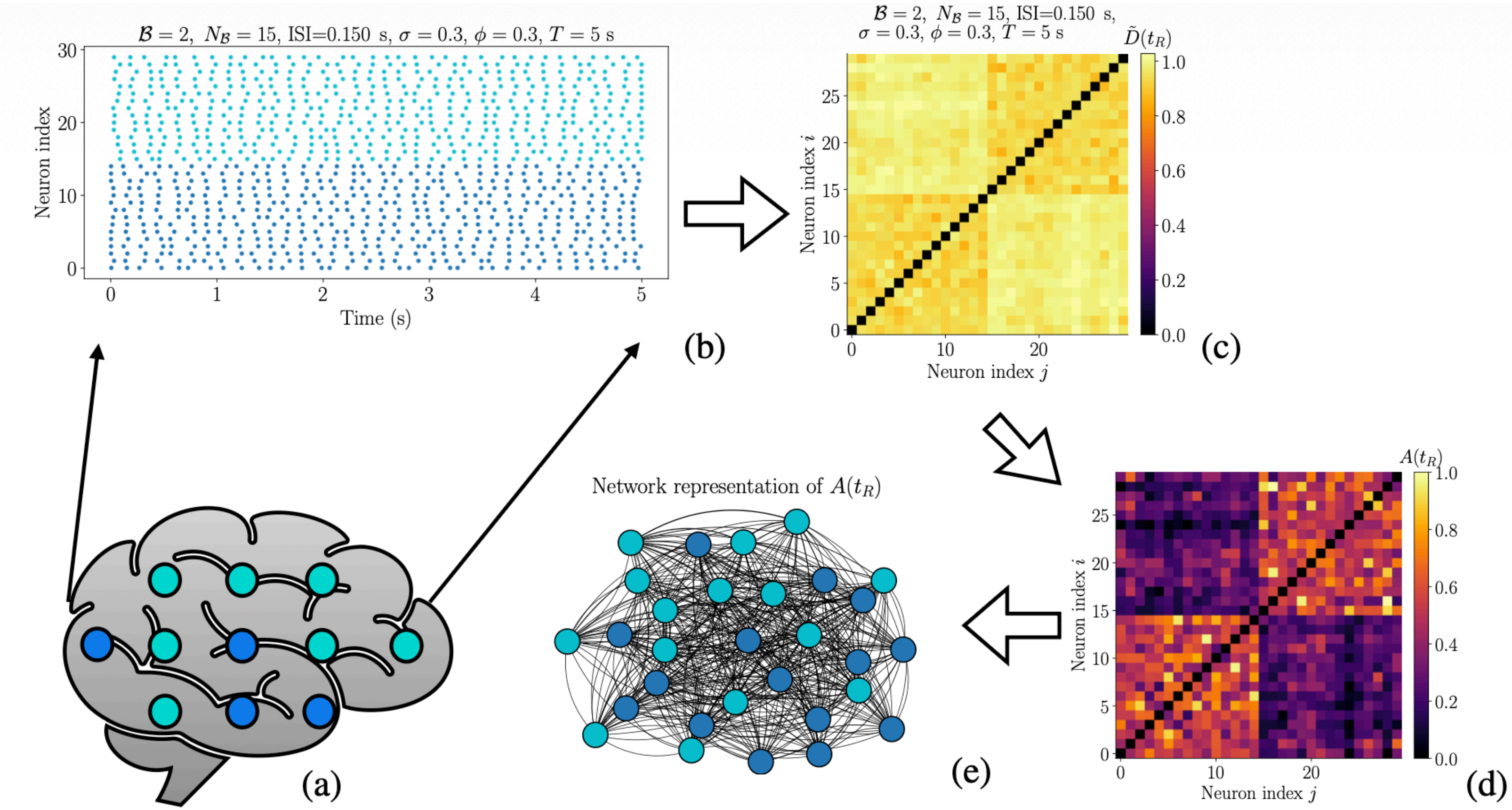
$$D_{ij}(t_R) = \sqrt{\frac{1}{t_R} \lim_{T\to\infty} \int_0^T (w_i(t) - w_j(t))^2 dt}\,.$$

4. Normalised: $\tilde{D}_{ij}(t_R) = D_{ij}(t_R)/\sqrt{\tilde{N}_{ij}}\,.$



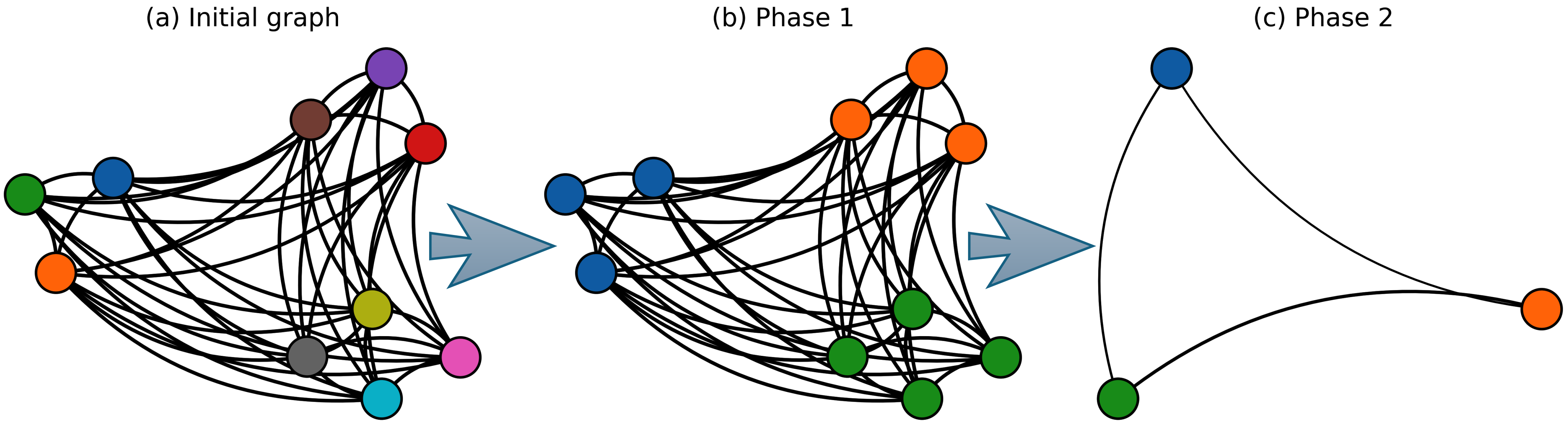
Principal workflow I

1. Synthetic raster: $t_b^k = (\phi_b + k) \times \mathrm{ISI} + \zeta^k$
2. Noise $\zeta^k \sim U[-\sigma_{\max}, \sigma_{\max}]$, where $\sigma_{\max} = \mathrm{round}(\mathrm{ISI} \times \sigma)$.
3. van Rossum distance matrix from raster: $\tilde{D}$.
4. Apply the min-max kernel to $\tilde{D}$ using $f(D) = \dfrac{D - \min(D)}{\max(D) - \min(D)}$.
5. Construct the similarity matrix $A = 1 - f(\tilde{D})$.

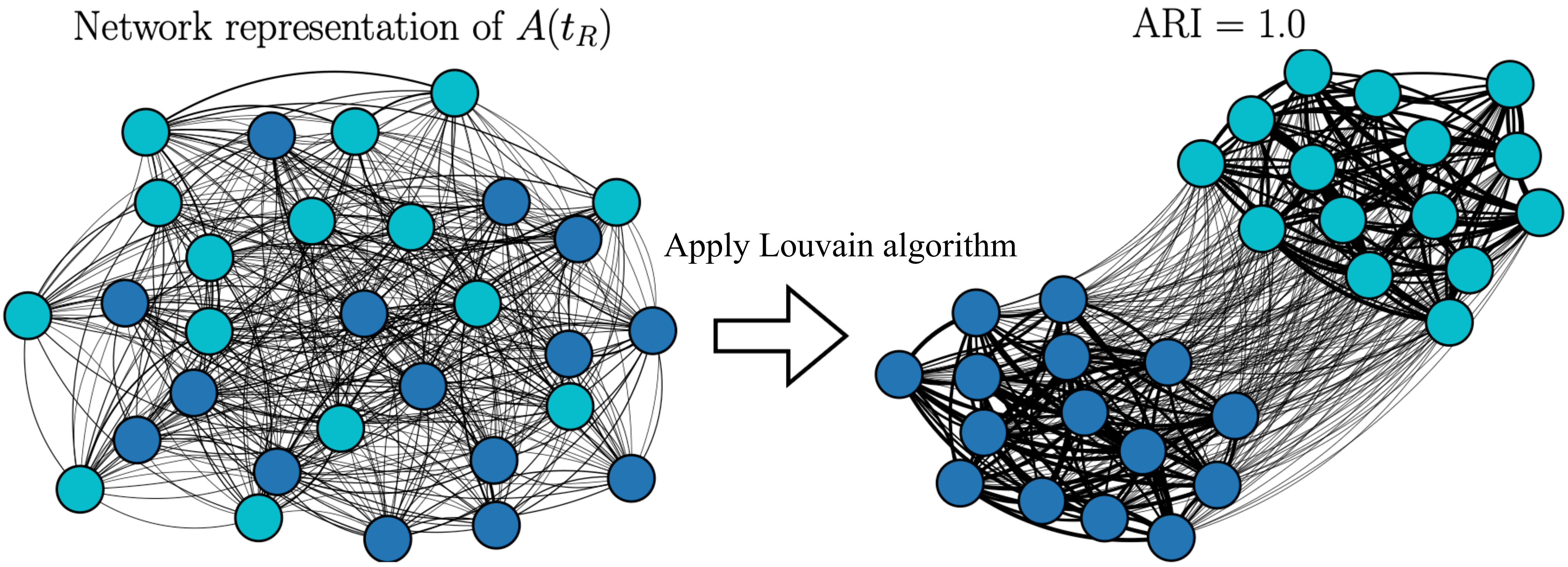


Louvain algorithm for community detection

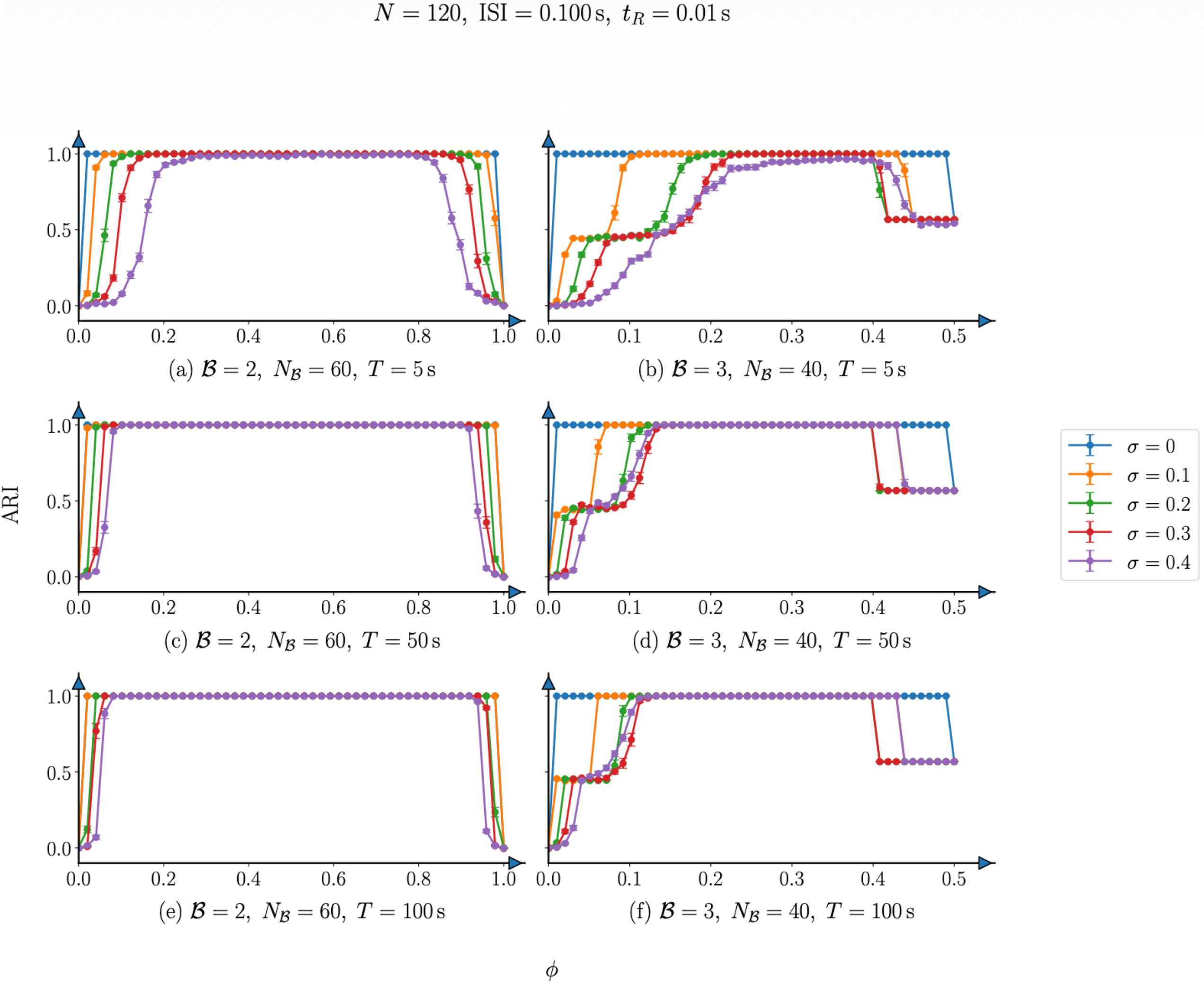
1. This algorithm depends on optimising the modularity of the network: $Q = \frac{1}{2m}\sum_{i=1}^{N}\sum_{j=1}^{N}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(c_i, c_j)\,.$
2. Here A_{ij}: element of the adjacency matrix, $k_i = \sum_j A_{ij}$, and c_i, c_j: the communities to which nodes i and j belong respectively, and $m = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} A_{ij}$.
3. Works in two phases:



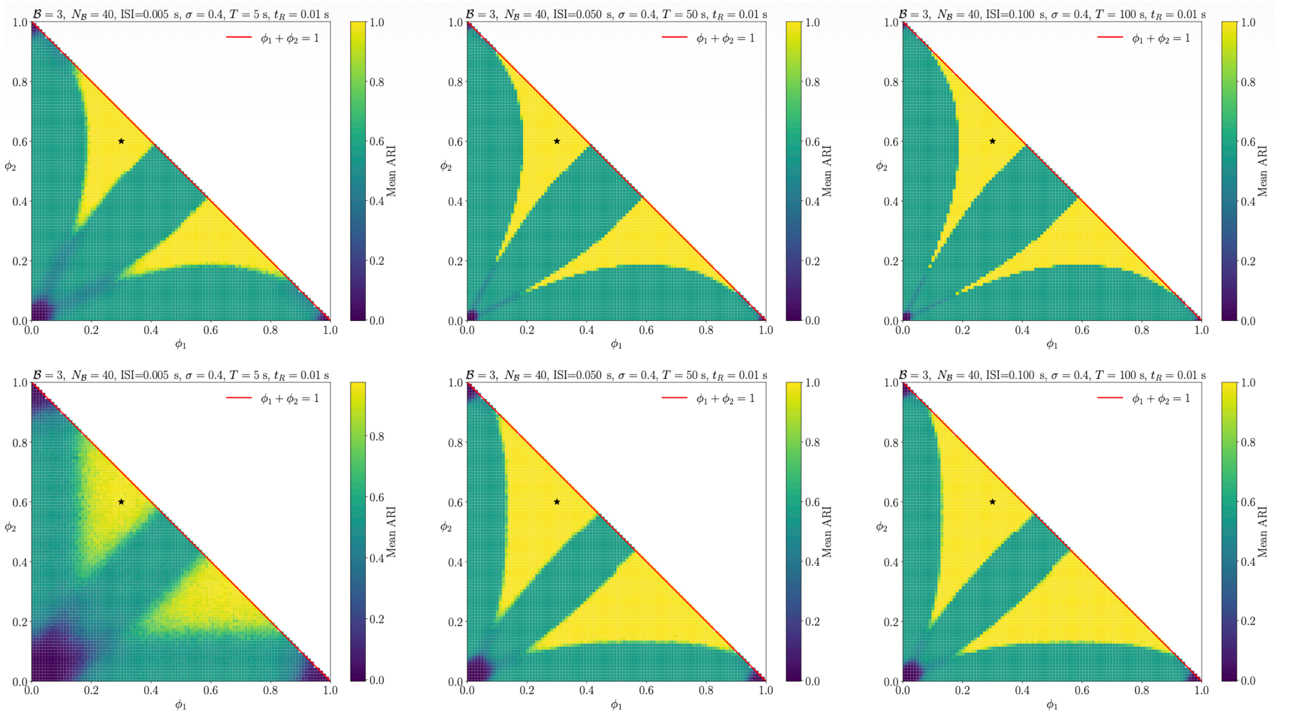
Principal workflow II



Results



$N = 120,\ \mathrm{ISI} = 0.100\,\mathrm{s},\ t_R = 0.01\,\mathrm{s}$
ARI
(a) $\mathcal{B} = 2,\ N_{\mathcal{B}} = 60,\ T = 5\,\mathrm{s}$
(b) $\mathcal{B} = 3,\ N_{\mathcal{B}} = 40,\ T = 5\,\mathrm{s}$
(c) $\mathcal{B} = 2,\ N_{\mathcal{B}} = 60,\ T = 50\,\mathrm{s}$
(d) $\mathcal{B} = 3,\ N_{\mathcal{B}} = 40,\ T = 50\,\mathrm{s}$
(e) $\mathcal{B} = 2,\ N_{\mathcal{B}} = 60,\ T = 100\,\mathrm{s}$
(f) $\mathcal{B} = 3,\ N_{\mathcal{B}} = 40,\ T = 100\,\mathrm{s}$
$\sigma = 0$
$\sigma = 0.1$
$\sigma = 0.2$
$\sigma = 0.3$
$\sigma = 0.4$
ϕ



$\mathcal{B} = 3,\ N_{\mathcal{B}} = 40$, ISI=0.005 s, $\sigma = 0.4$, $T = 5$ s, $t_R = 0.01$ s
$\mathcal{B} = 3,\ N_{\mathcal{B}} = 40$, ISI=0.050 s, $\sigma = 0.4$, $T = 50$ s, $t_R = 0.01$ s
$\mathcal{B} = 3,\ N_{\mathcal{B}} = 40$, ISI=0.100 s, $\sigma = 0.4$, $T = 100$ s, $t_R = 0.01$ s
$\phi_1 + \phi_2 = 1$
Mean ARI
ϕ_1
ϕ_2

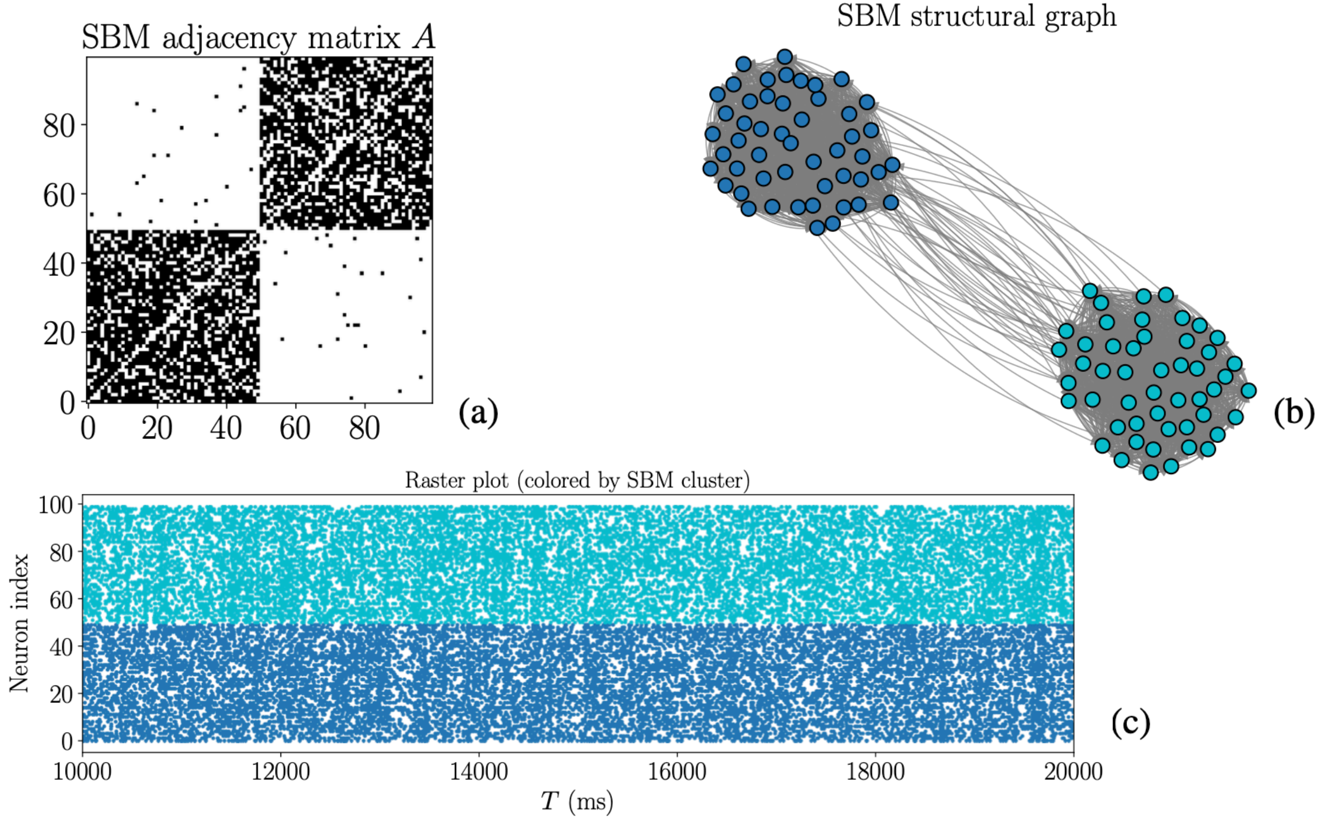
Application to LIF neurons

We apply this methodology to a stochastic block model (SBM) network of Leaky Integrate and Fire (LIF) neurons, subjected to external Poisson drive:

$$C_m \frac{dV_i}{dt} = g_{L,i}(V_L - V_i) + s_{E,i}(t)(V_E - V_i) + s_{I,i}(t)(V_I - V_i)\,.$$

$$\frac{ds_{E,j}}{dt} = -\frac{s_{E,j}}{\tau_E} + \sum_{i\in E} \frac{g_E}{\tau_E} A_{ij}\delta(t - t_i^{\text{spike}}) + \frac{f_{\mathscr{C}(j)}}{\tau_E}\frac{dQ_j(t)}{dt},$$

$$\frac{ds_{I,j}}{dt} = -\frac{s_{I,j}}{\tau_I} + \sum_{i\in I} \frac{g_I}{\tau_I} A_{ij}\delta(t - t_i^{\text{spike}})\,.$$



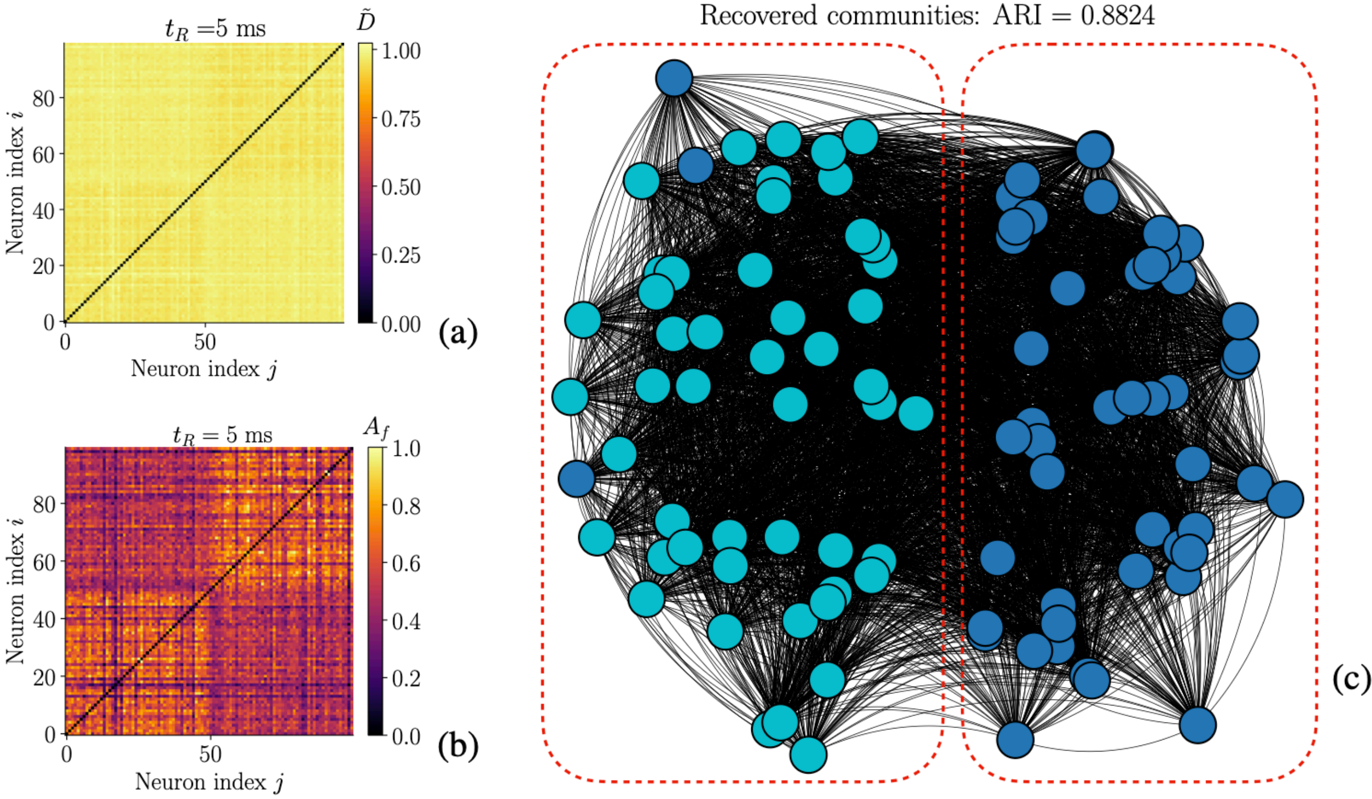
Application to LIF neurons

We apply this methodology to a stochastic block model (SBM) network of Leaky Integrate and Fire (LIF) neurons, subjected to external Poisson drive:

$$C_m \frac{dV_i}{dt} = g_{L,i}(V_L - V_i) + s_{E,i}(t)(V_E - V_i) + s_{I,i}(t)(V_I - V_i)\,.$$

$$\frac{ds_{E,j}}{dt} = -\frac{s_{E,j}}{\tau_E} + \sum_{i \in E} \frac{g_E}{\tau_E} A_{ij}\delta(t - t_i^{\text{spike}}) + \frac{f_{\mathscr{C}(j)}}{\tau_E}\frac{dQ_j(t)}{dt},$$

$$\frac{ds_{I,j}}{dt} = -\frac{s_{I,j}}{\tau_I} + \sum_{i \in I} \frac{g_I}{\tau_I} A_{ij}\delta(t - t_i^{\text{spike}})\,.$$



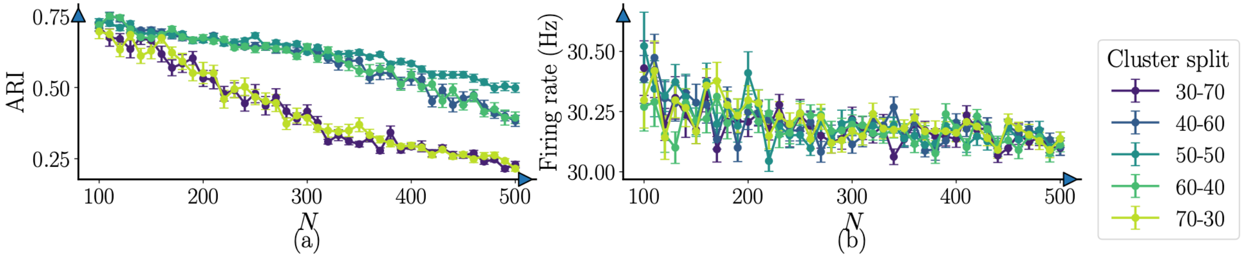
Application to LIF neurons

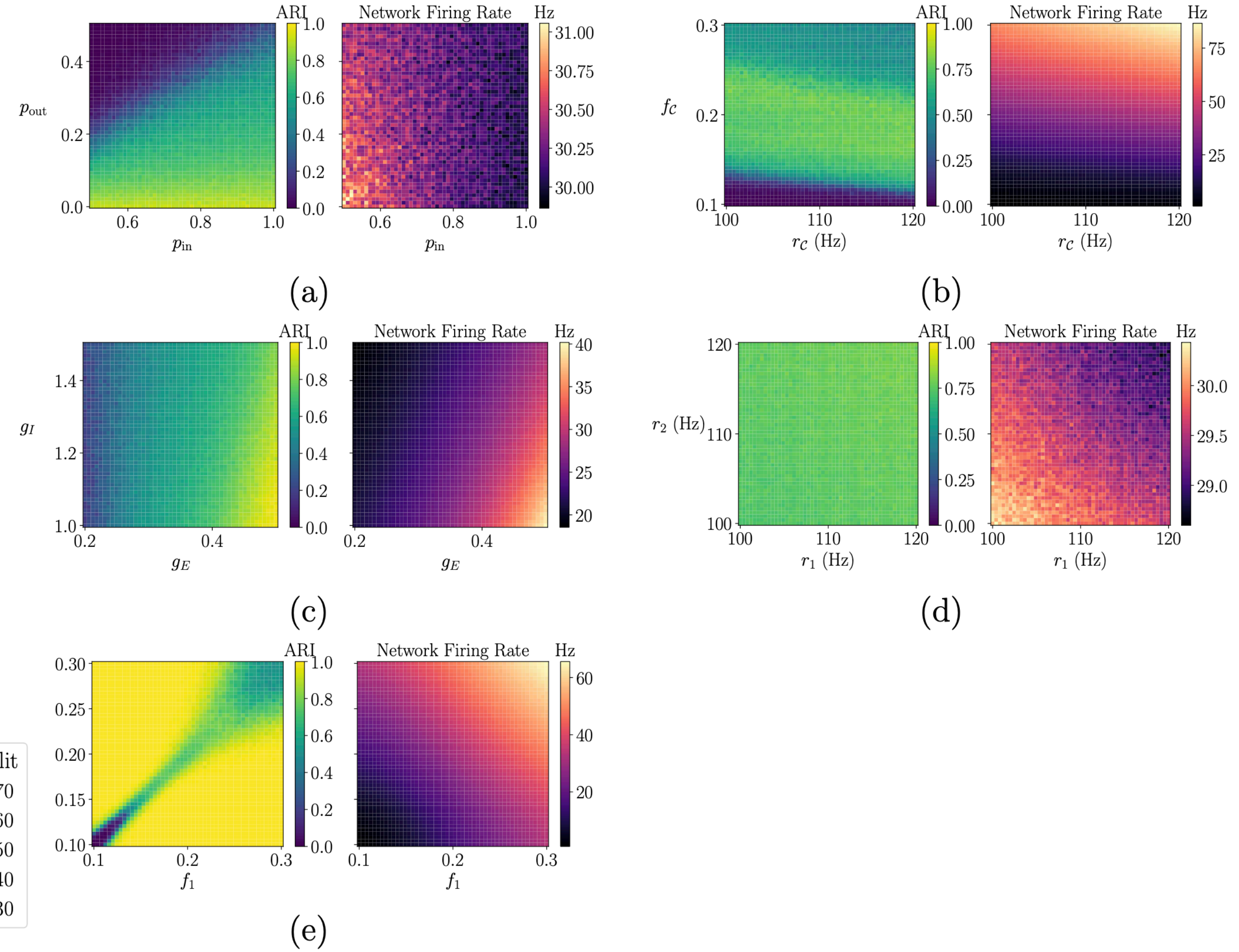
We apply this methodology to a stochastic block model (SBM) network of Leaky Integrate and Fire (LIF) neurons, subjected to external Poisson drive:

$$C_m \frac{dV_i}{dt} = g_{L,i}(V_L - V_i) + s_{E,i}(t)(V_E - V_i) + s_{I,i}(t)(V_I - V_i)\,.$$

$$\frac{ds_{E,j}}{dt} = -\frac{s_{E,j}}{\tau_E} + \sum_{i \in E} \frac{g_E}{\tau_E} A_{ij} \delta(t - t_i^{\text{spike}}) + \frac{f_{\mathscr{C}(j)}}{\tau_E} \frac{dQ_j(t)}{dt},$$

$$\frac{ds_{I,j}}{dt} = -\frac{s_{I,j}}{\tau_I} + \sum_{i \in I} \frac{g_I}{\tau_I} A_{ij} \delta(t - t_i^{\text{spike}})\,.$$



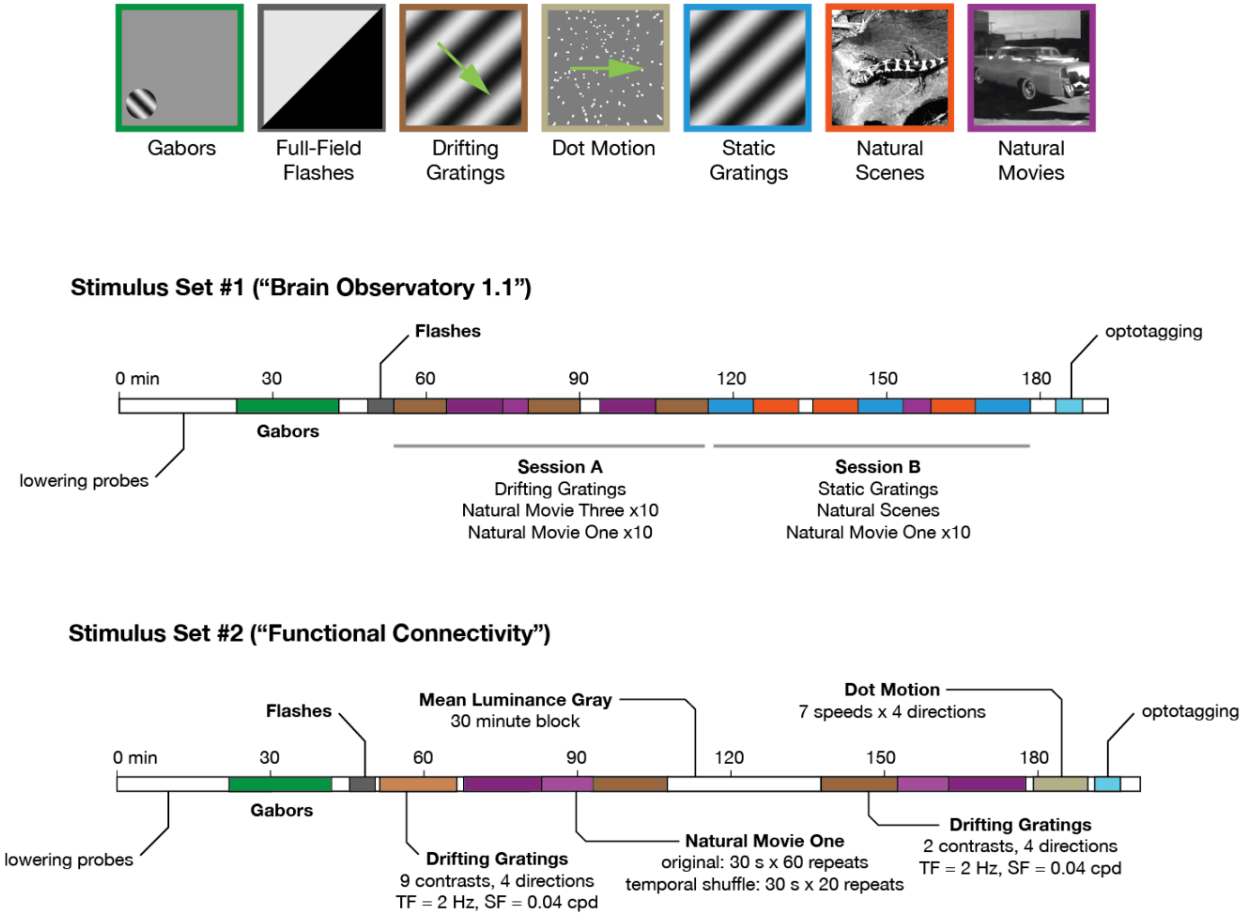


Applying on Neuropixels data from Allen Brain Institute

1. We also apply to visual coding Neuropixels data from the Allen Institute dataset
2. We only look at the “spontaneous” block between two stimuli for just the wild type mice.
3. Only the “functional connectivity” dataset.

Session ID	Specimen ID	Age (days)	Sex	Unit count	Channel count	Probe count
766640955	744912849	133.0	M	842	2233	6
767871931	753795610	135.0	M	713	2231	6
768515987	754477358	136.0	M	802	2217	6
771160300	754488979	142.0	M	930	2230	6
771990200	756578435	108.0	M	546	2229	6
774875821	759711152	114.0	M	649	2233	6
778240327	760938797	120.0	M	784	2234	6
778998620	759674770	121.0	M	793	2229	6
779839471	760960653	122.0	M	863	2220	6
781842082	760946813	126.0	M	833	2232	6
793224716	769319624	120.0	M	781	2229	6
819186360	800249587	128.0	F	531	1696	5
821695405	800250057	134.0	F	474	1856	5
847657808	827809884	126.0	F	874	2298	6

Table 1: Allen SDK extracellular electrophysiology (ecephys) data.

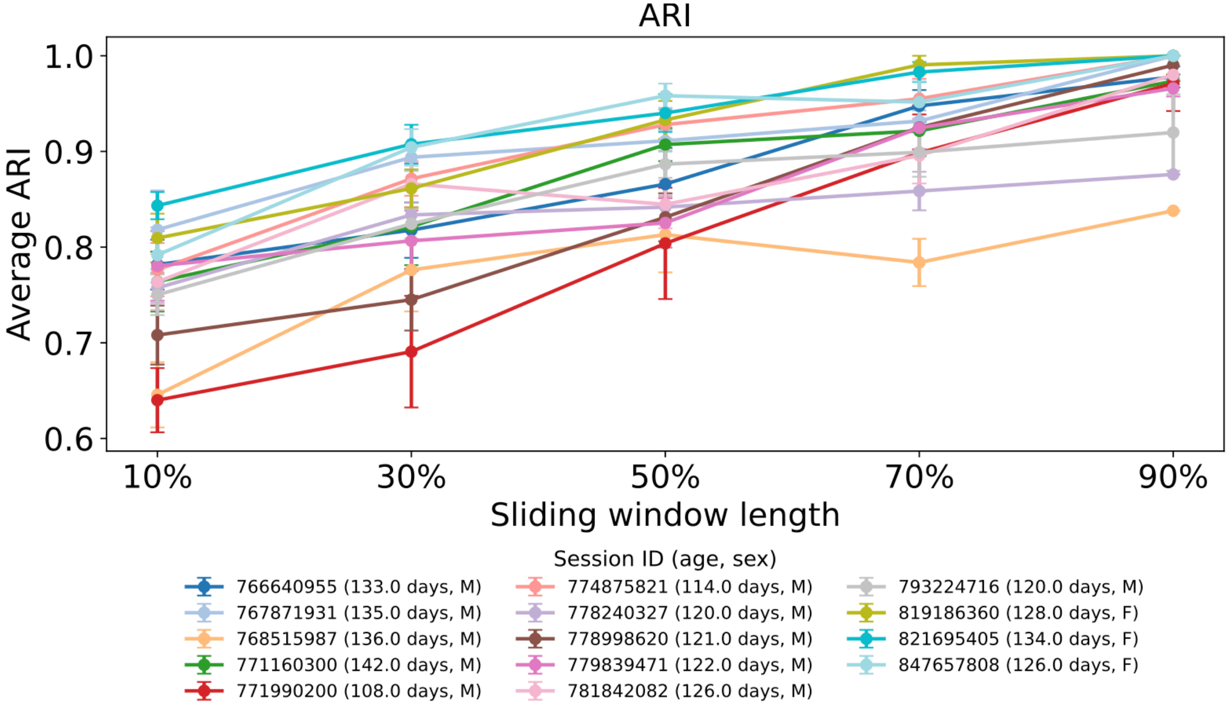


Visual stimuli sets showing “Brain Observatory 1.1” and “Functional Connectivity” types. Image credit: Allen Institute for Brain Science. [https://allensdk.readthedocs.io/en/latest/visual_coding_neuropixels.html]

Applying on Neuropixels data from Allen Brain Institute

Stimulus ID	Mean duration (s)	Median duration (s)
0	60.067	60.067
3646	288.993	288.992
3797	61.802	61.802
4338	120.601	120.601
40639	1802.514	1802.506
40940	1.001	1.001

Table 2: Stimulus IDs with mean and median durations in seconds.



Questions/comments/suggestions?

Created using ChatGPT 5.1