Clustering Neural Populations by State-space Factor Models

Ganchao Wei[1], Xiaojing Wang[1], Ian H. Stevenson[2, 3]

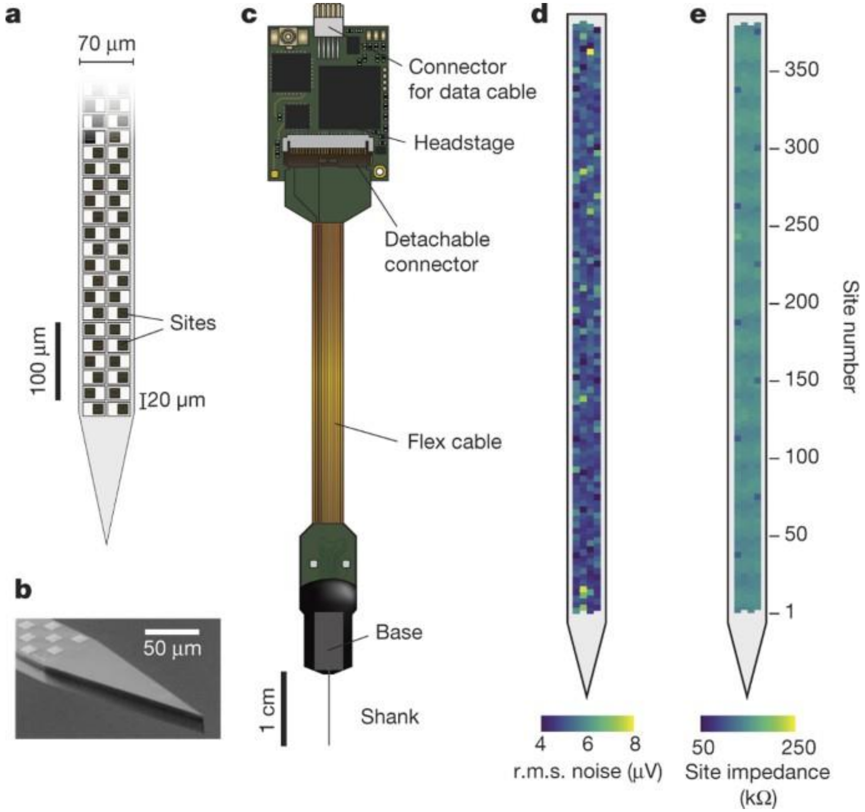
1 University of Connecticut, Department of Statistics

2 University of Connecticut, Department of Psychological Sciences

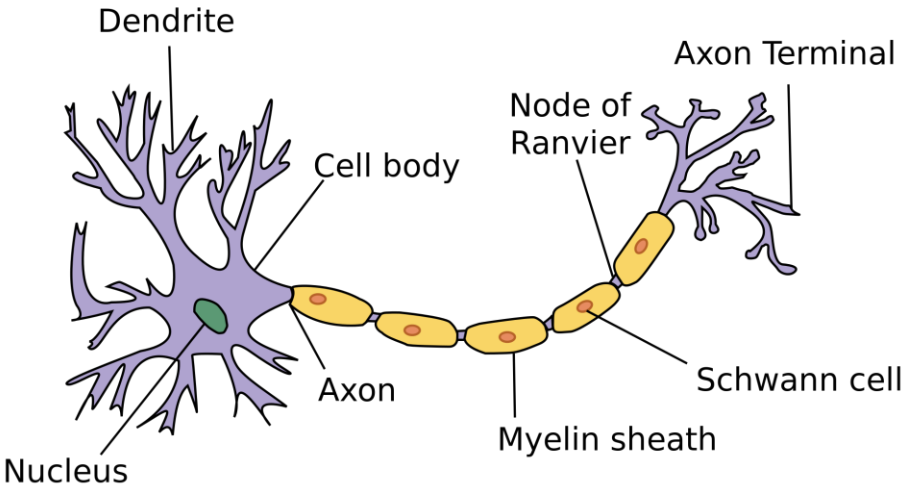
3 University of Connecticut, Department of Biomedical Engineering

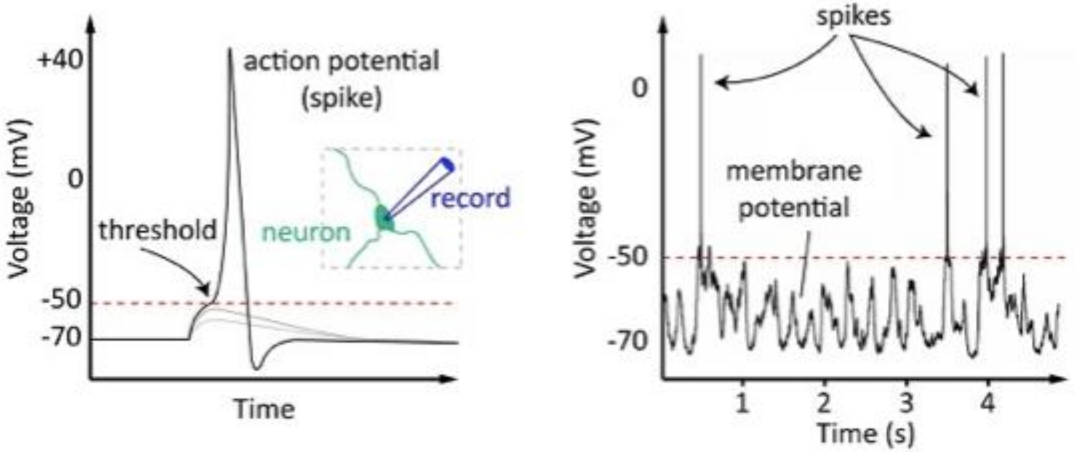
Introduction

- Neurons & neural spikes
- Modern techniques →
- study neurons in multi-population level
    - High-density silicon probes
    - large-scale calcium imaging methods
    - …



Jun et al., Nature 2017





Introduction

- Relationships within and between populations
  - Described by **low-dimensional latent state vectors**
  - usually modeled by **AR(1)** or **GP**.
- **BUT**… defining populations? **Difficult**!
  - Anatomical vagueness
  - Neurons in different sites can talk to each other, physically/ chemically
- One solution: do distance-based clustering at first, but…
  - How to define "distance"? (Tricky & loose information)
  - May bias the latent structures.

Introduction

- **Combine these two** (model-based clustering):
  - Let the latent structure help with clustering & vice versa
  - Capture latent structure by state-space factor models
- For neuroscience :
  - Help detect potential functional-related neurons (physically/ chemically)
  - Capture the temporal spiking pattern
- Beyond neuroscience:
  - Cluster general time series data
  - Extract low-dimensional structure at the same time

Model (1): Neural Populations-- SSFM

- **State-space factor model** (SSFM)
- **Observation**: $Y = (y_{it}) \in \mathbb{Z}_{\geq 0}^{N \times T}$ (N neurons, T steps)
- Given the cluster indicator z_i for neuron i:

$$y_{it} \sim Poi(\lambda_{it})$$
$$\log(\lambda_{it})\,|z_i = d_i^{(z_i)} + \boldsymbol{c}_i^{(z_i)} \boldsymbol{x}_t^{(z_i)}$$
$$\left(d_i^{(z_i)}, \boldsymbol{c}_i'^{(z_i)}\right)' \sim N_{p+\mathbf{1}}\left(\boldsymbol{\mu}_{dc}^{(z_i)}, \boldsymbol{\Sigma}_{dc}^{(z_i)}\right)$$

, where $\boldsymbol{c}_i^{(z_i)} \in \mathbb{R}^p$ and $\boldsymbol{x}_t^{(z_i)} \in \mathbb{R}^p$. $\boldsymbol{x}_t^{(z_i)}$ progresses linearly with a Gaussian noise:

$$\boldsymbol{x}_1^{(z_i)} \sim N_p(\boldsymbol{x}_0, \boldsymbol{Q}_0)$$
$$\boldsymbol{x}_{t+1}^{(z_i)} | \boldsymbol{x}_t^{(z_i)} \sim N_p(\boldsymbol{A}^{(z_i)} \boldsymbol{x}_t^{(z_i)} + \boldsymbol{b}^{(z_i)}, \boldsymbol{Q}^{(z_i)})$$

Model (1): Neural Populations-- SSFM

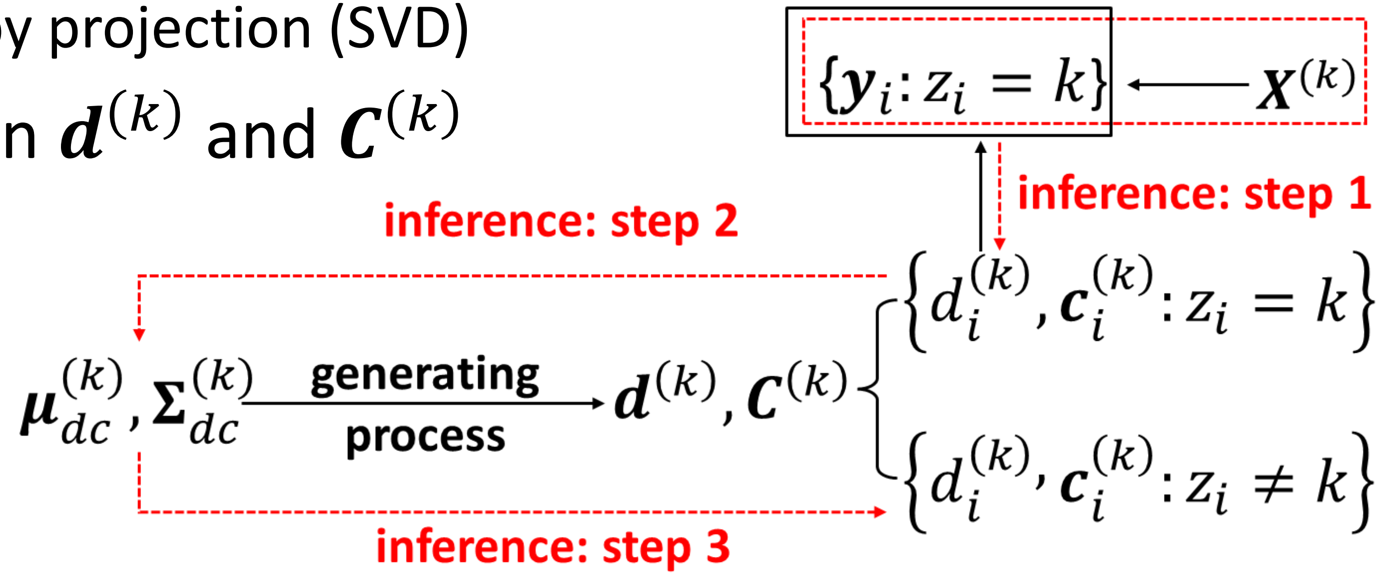
- **Remark 1**: Constraints for model identifiability
  - $\boldsymbol{X}^{(k)} = \left(\boldsymbol{x}_1^{(k)}, \dots, \boldsymbol{x}_T^{(k)}\right) \in \mathbb{R}^{p \times T}$: each row of has mean 0 and $\boldsymbol{X}^{(k)}\boldsymbol{X}'^{(k)} = \boldsymbol{I}_p$
  - + diagonal $\boldsymbol{A}^{(k)}$and $\boldsymbol{Q}^{(k)}$ = identifiable model
- **Remark 2**: $d_i^{(z_i)}$ and $\boldsymbol{c}_i^{(z_i)}$ are both **neuron**- and **cluster**-dependent
  - Auxiliary parameters $\left\{d_i^{(k)}, \boldsymbol{c}_i^{(k)}: z_i \neq k\right\}$ help clustering
  - The prior $\boldsymbol{\mu}_{dc}^{(z_i)}$ and $\boldsymbol{\Sigma}_{dc}^{(z_i)}$ help inference for these
- **Remark 3**: Decomposition of spiking features, 3 parts
  - The baseline firing rate $d_i^{(z_i)}$
  - A set (p) of centered and orthonormal temporal patterns $\boldsymbol{X}^{(k)}$
  - The “magnitude” of each temporal pattern $\boldsymbol{c}_i^{(z_i)}$

Model (2): Clustering– MFM

- **Model summary**:
  - cluster parameters: $\boldsymbol{\Theta}_k = \left\{\boldsymbol{d}^{(k)}, \boldsymbol{C}^{(k)}, \boldsymbol{\mu}_{dc}^{(k)}, \boldsymbol{\Sigma}_{dc}^{(k)}, \boldsymbol{X}^{(k)}, \boldsymbol{A}^{(k)}, \boldsymbol{b}^{(k)}, \boldsymbol{Q}^{(k)}\right\}$, with prior $\boldsymbol{H}$
  - $\boldsymbol{Y}_i = (Y_{i1}, \dots, Y_{iT})' \sim SSFM\left(\boldsymbol{\Theta}_{z_i}\right)$
- Unknown number of clusters → DPM? **Wrong!**
- Number of neural population is **finite but unknown**
- Put prior on cluster number → **mixture of finite mixtures (MFM)**
  - $K \sim p_k$ where p_k is a p.m.f. on $\{1,2, \dots\}$
  - $\boldsymbol{\pi} = (\pi_1, \dots \pi_k) \sim Dirichilet_k(\gamma, \dots, \gamma)$ given $K = k$
  - $Z_1, \dots, Z_N \overset{\text{i.i.d.}}{\sim} \boldsymbol{\pi}$ given $\boldsymbol{\pi}$
  - $\Theta_1, \dots \Theta_k \overset{\text{i.i.d.}}{\sim} \boldsymbol{H}$ given $K = k$
  - $\boldsymbol{Y}_i = (Y_{i1}, \dots, Y_{iT})' \sim SSFM\left(\boldsymbol{\Theta}_{z_i}\right)$ given $\boldsymbol{\Theta}_{1:K} and\ Z_{1:N}$

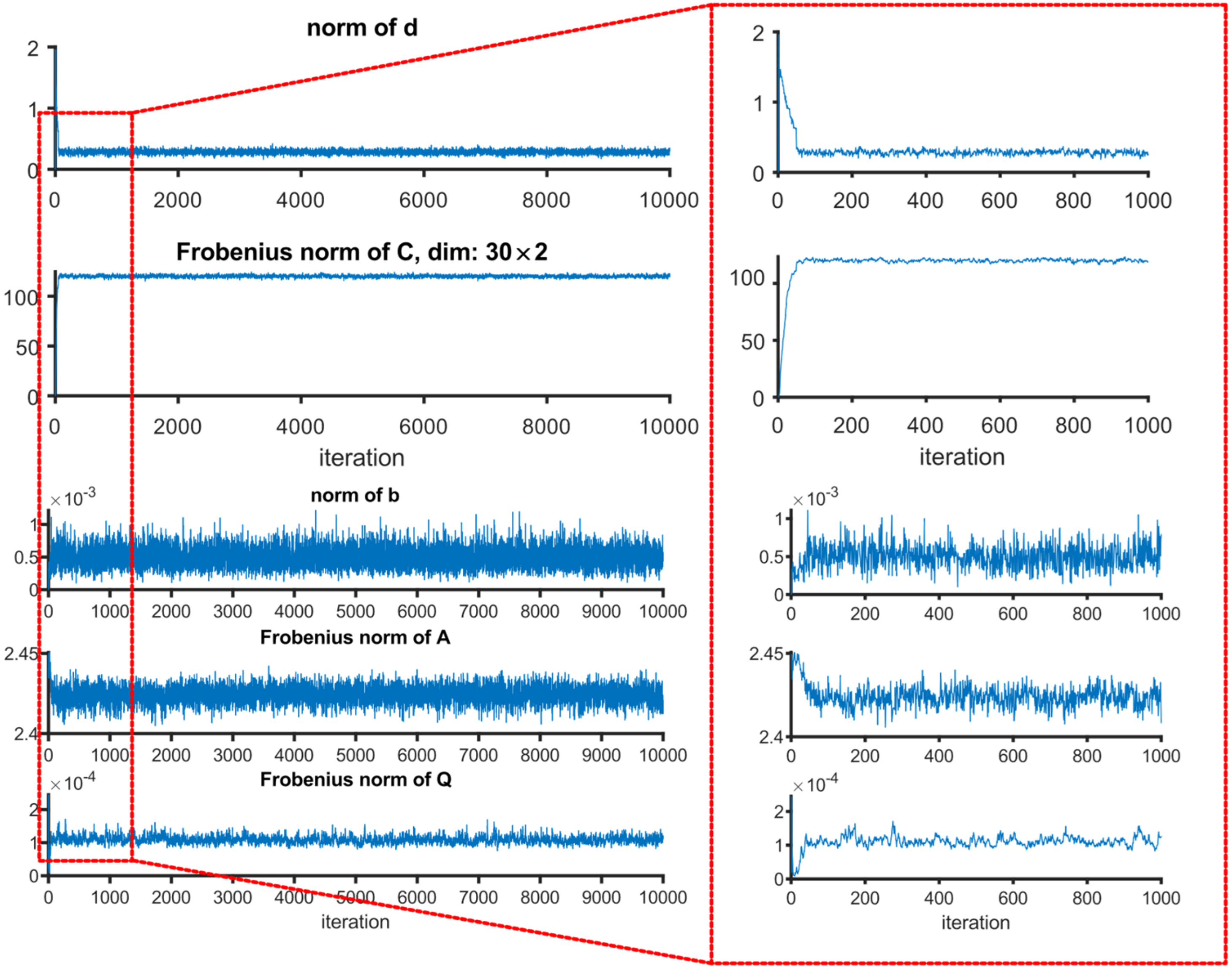
Inference

- Sample posteriors by MCMC
- SSFM-related parameters:
  - Poisson likelihood → particle MCMC? **Cumbersome…**
  - Unimodality & Markovian structure (tri-block diagonal Hessian) →Laplace approximation efficiently in $O(T)$
  - Constraints handled by projection (SVD)
- Auxiliary parameters in $\boldsymbol{d}^{(k)}$ and $\boldsymbol{C}^{(k)}$



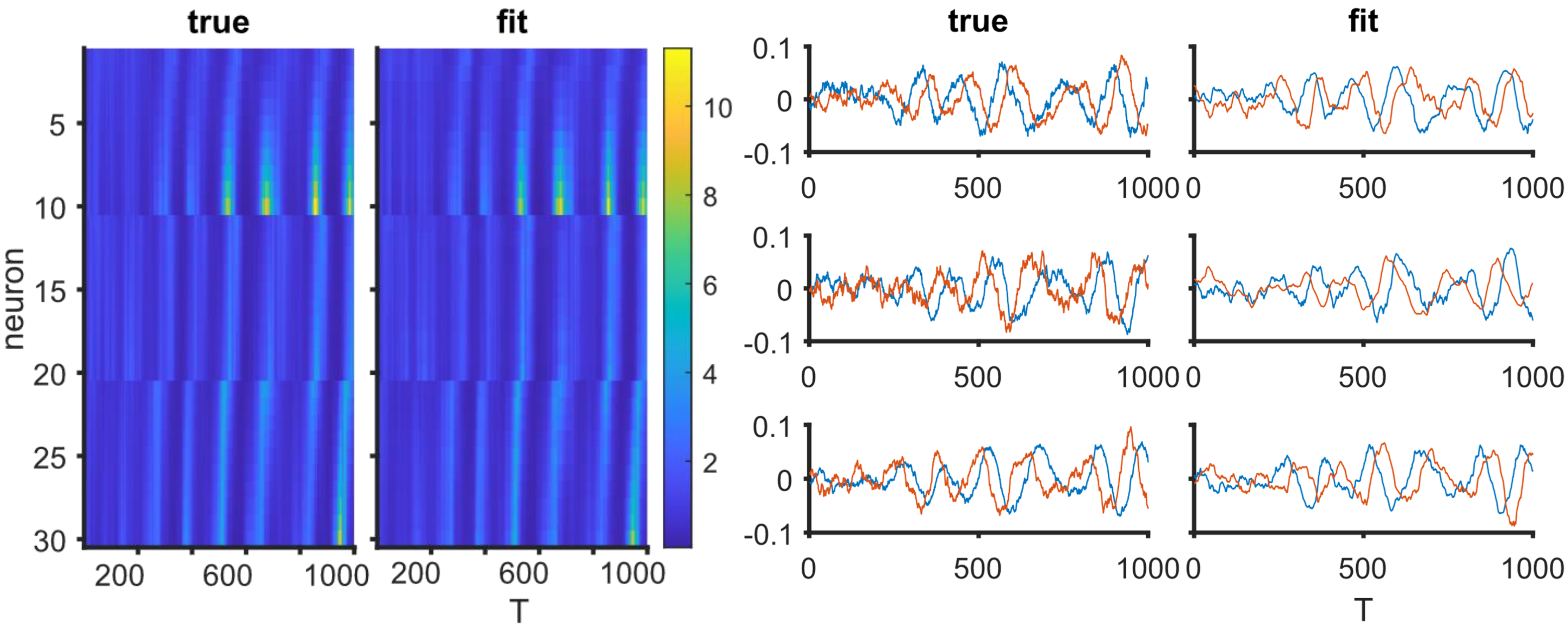
Simulation 1: Neurons with Known Labels

- 3 clusters, 10 neurons each.
- $p = 2$ and $T = 1000$
- MCMC: 10,000 iterations



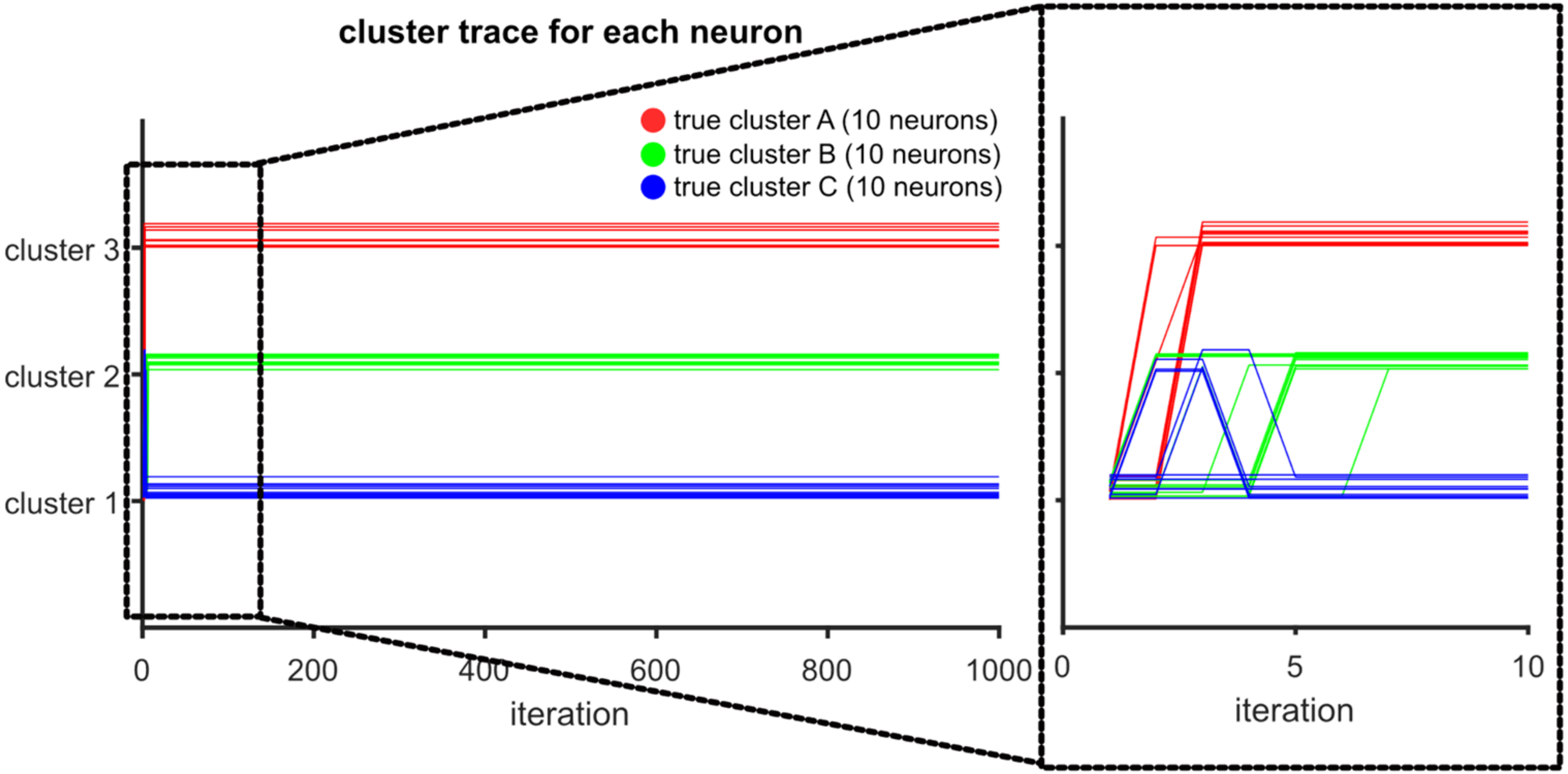
Simulation 1: Neurons with Known Labels

- Averages of fitted mean firing rate and latent sate (iter 1000- 10,000)



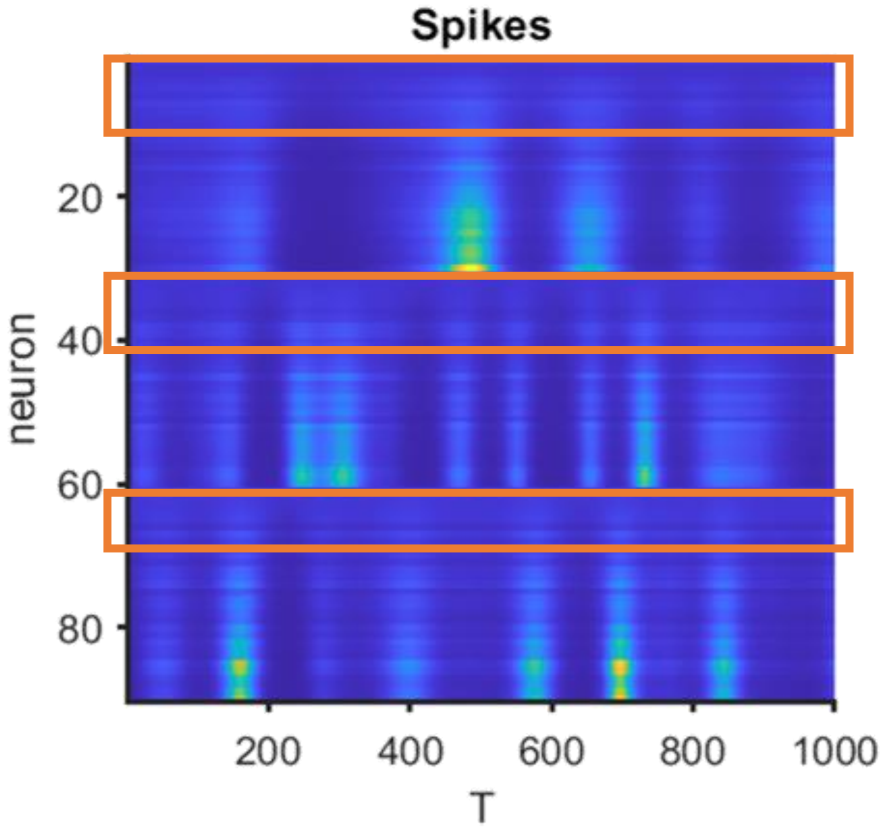
Simulation 2: Neurons with Unknown Labels

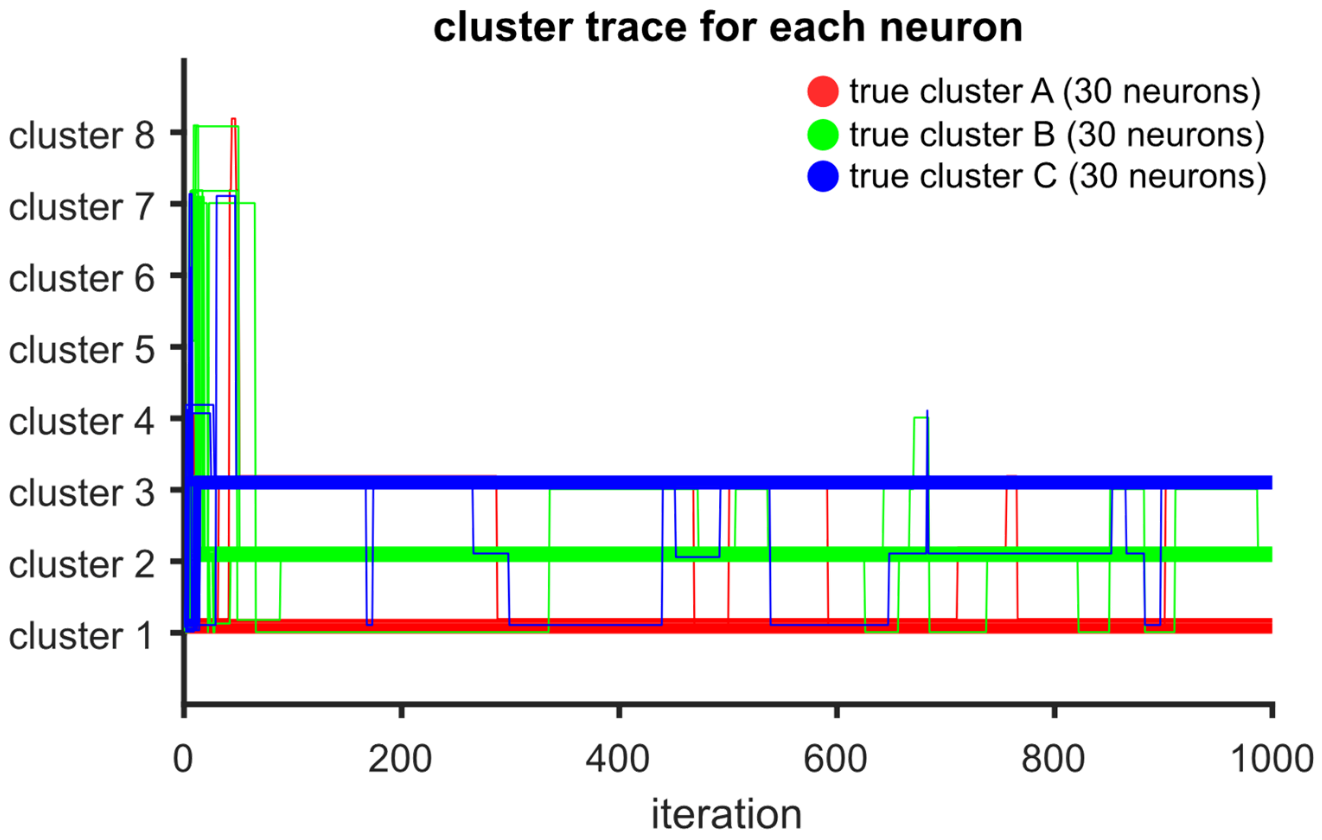
- same settings as in simulation 1 but with unknown labels



Simulation 3: A More Challenging Setting

- 30 neurons each.
- In each cluster, some neurons have weak signals (hard to cluster).



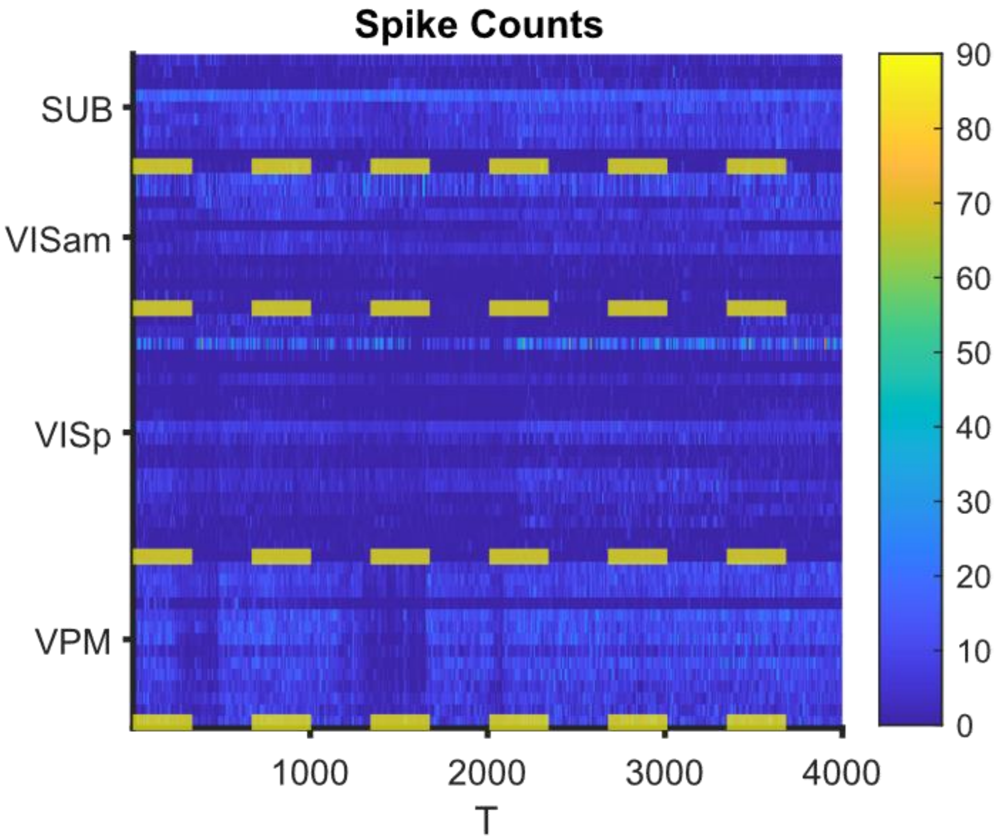


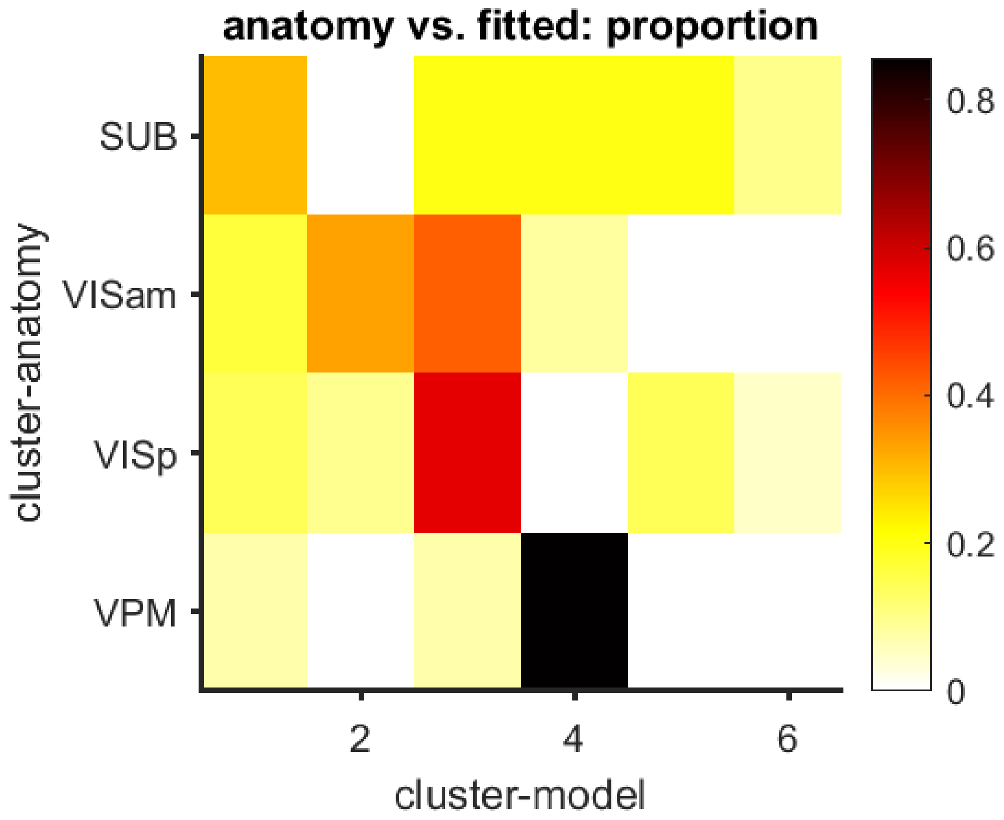
Application: Neuralpixels Data

- Data from The Alan Institute
- 57 neurons from 4 anatomical sites:
  - Subiculum (**SUB**): part of the hippocampus for spatial navigation/memory
  - 2 visual areas (**VISp** and **VISam**)
  - a part of thalamus (**VPM**): involved in sensation/movement
- Hard to cluster:
  - Activity in all these areas depend a bit on the movement of the animal.
  - Each area has different types of neurons within it, e.g. excitatory vs. inhibitory (~20-30%).

Application: Neuralpixels Data

- Use ~30 min recordings for clustering (bin size = 0.5s).
- Set p=4. The average results from iteration 1000 to 3000.



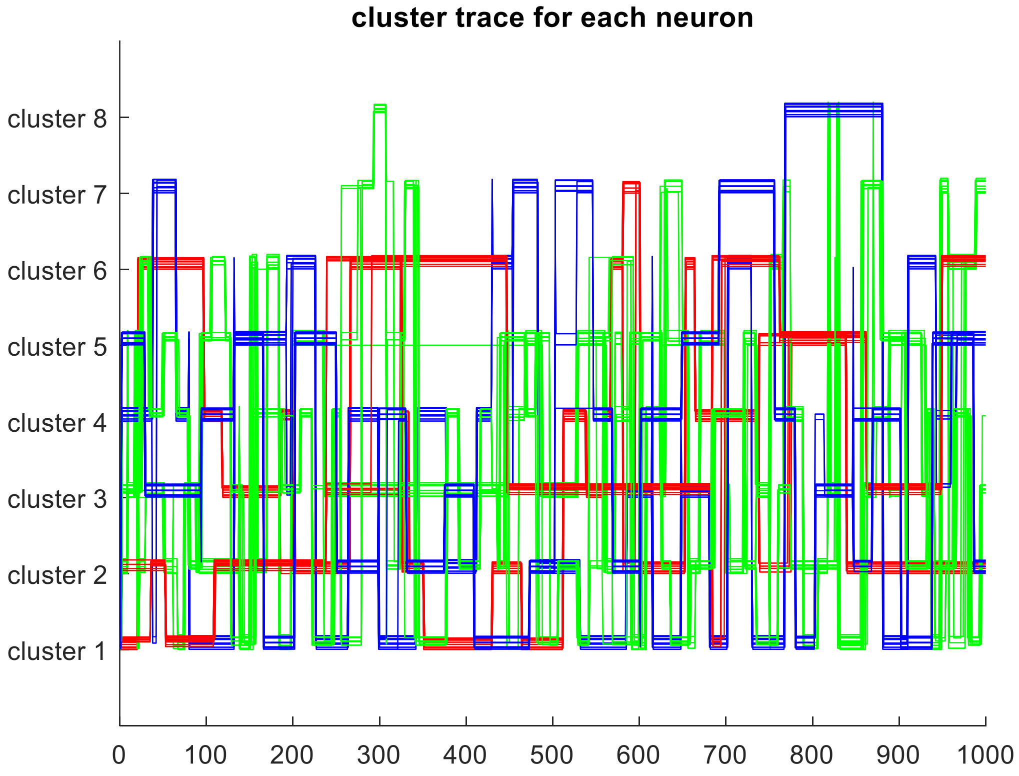


Appendix

- Why not update clusters based on prior $\boldsymbol{\mu}_{dc}^{(z_i)}, \boldsymbol{\Sigma}_{dc}^{(z_i)}$? $\rightarrow$
  - No need to make $d_i^{(z_i)}$ and $\boldsymbol{c}_i^{(z_i)}$ be **cluster**-dependent
  - No need to use auxiliary parameters
  - Update clusters by marginal likelihood $P(Y_i|\boldsymbol{\mu}_{dc}^{(z_i)}, \boldsymbol{\Sigma}_{dc}^{(z_i)}, \boldsymbol{X}^{(k)})$
  - Use the Laplace approximation
    $\int P\left(Y_i \middle| (d_i, \boldsymbol{c}_i')', \boldsymbol{X}^{(k)}\right) P\left((d_i, \boldsymbol{c}_i')' \middle| \boldsymbol{\mu}_{dc}^{(z_i)}, \boldsymbol{\Sigma}_{dc}^{(z_i)}\right) d\left((d_i, \boldsymbol{c}_i')'\right)$
- Looks promising, **but…super unstable**

Appendix (Simulation 2 revisit)

Raw trace



Sorted trace