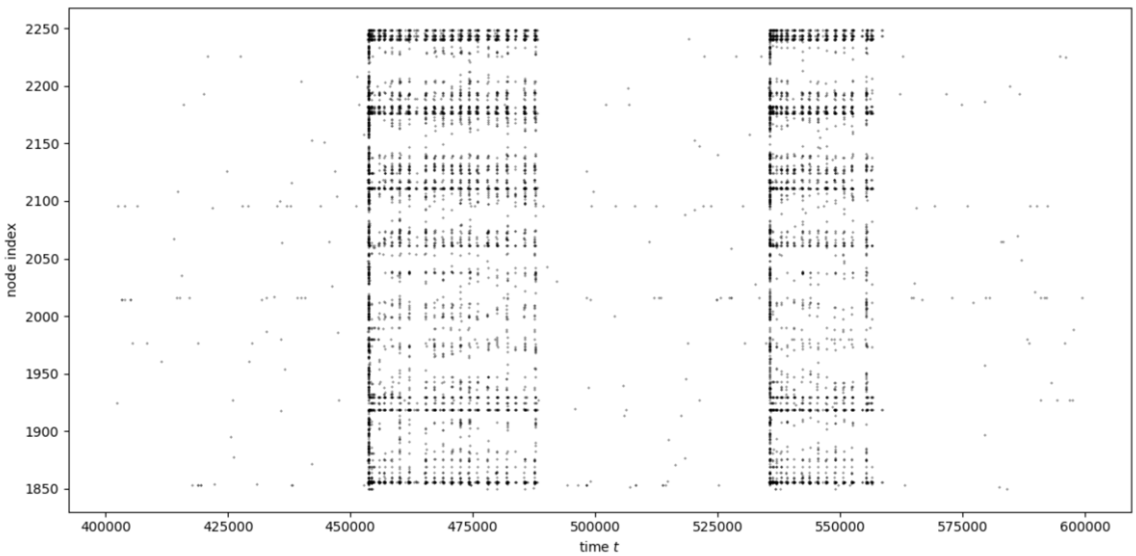
Table of content

1. Raster plots

For clarity, spikes for node 1850-2250 over time interval [400000,600000] are plotted.

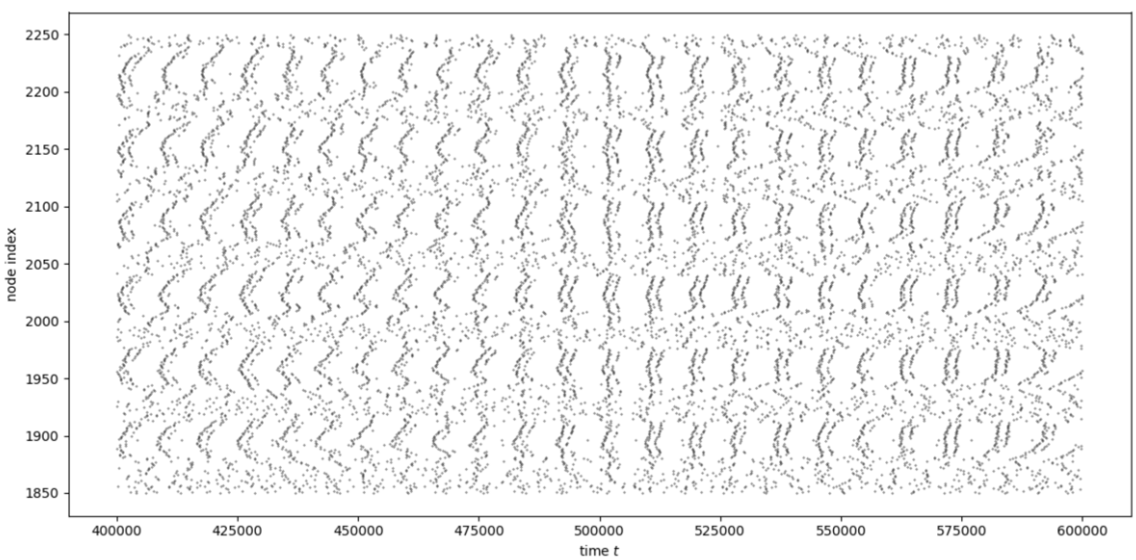
Measured spikes



Spike pattern persists for 50000 time units.

Simulated spikes

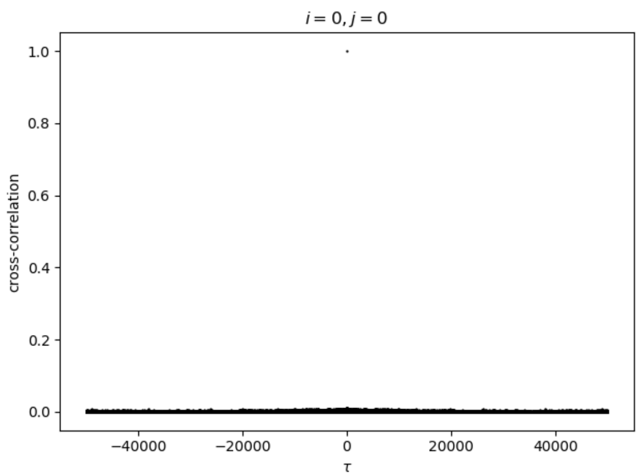
Synaptic FHN model with params $\epsilon = 0.1, \alpha = 0.95, \beta_0 = 0.01, \beta_1 = 1, y_0 = 0$



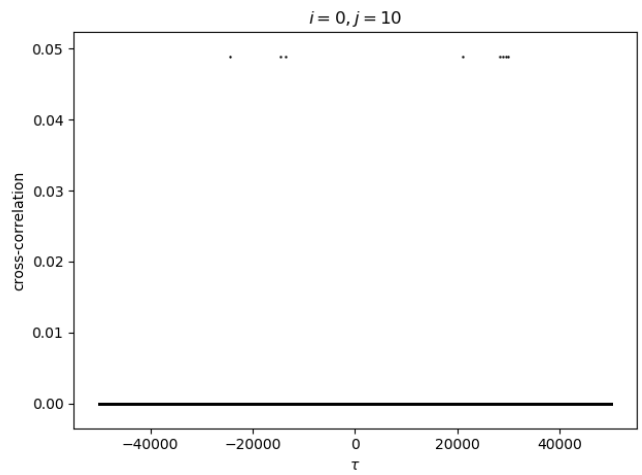
2. FNCCH method

CCH for i=0 and j=0,10,20,30,40

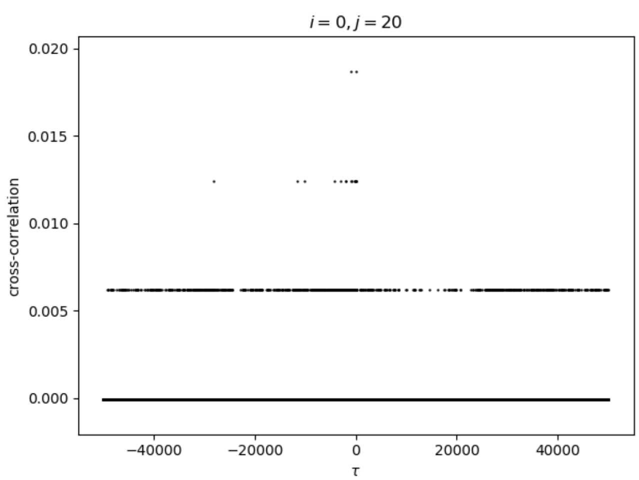
i=0, j=0



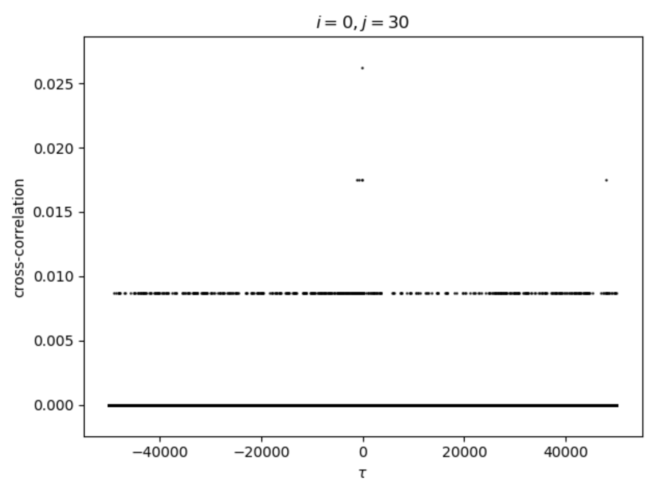
i=0, j=10



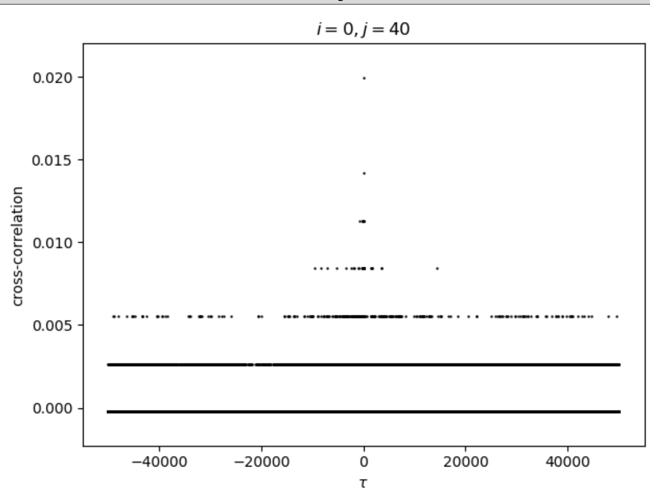
i=0, j=20



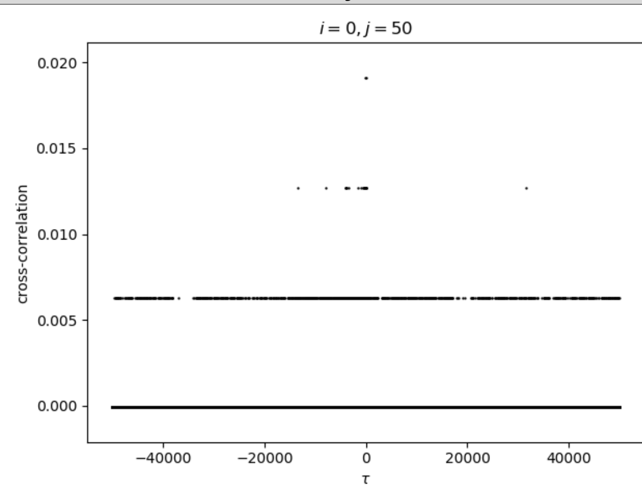
i=0, j=30

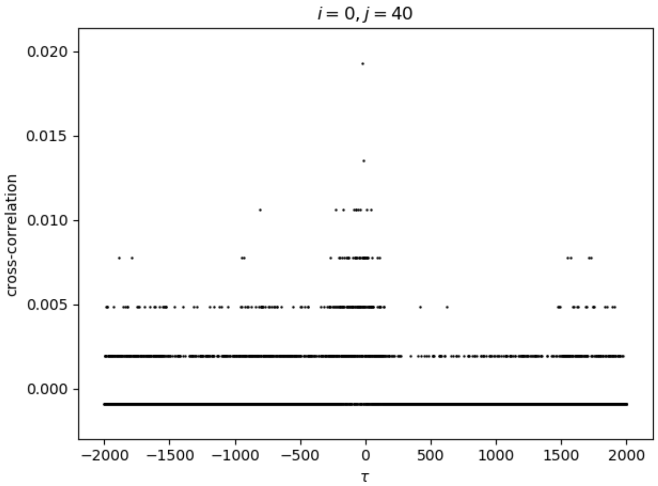
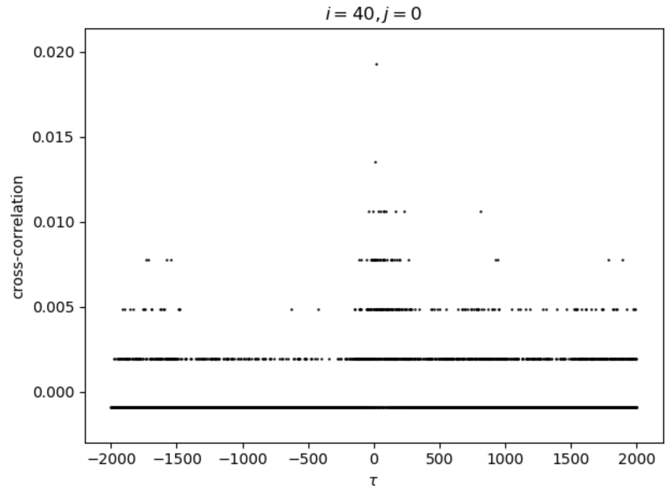
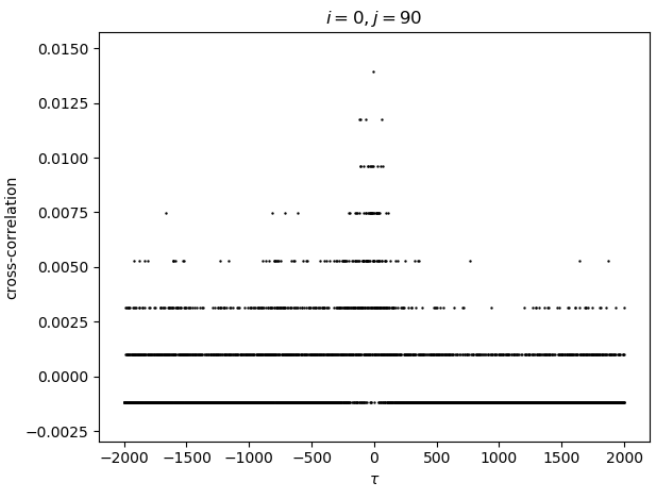
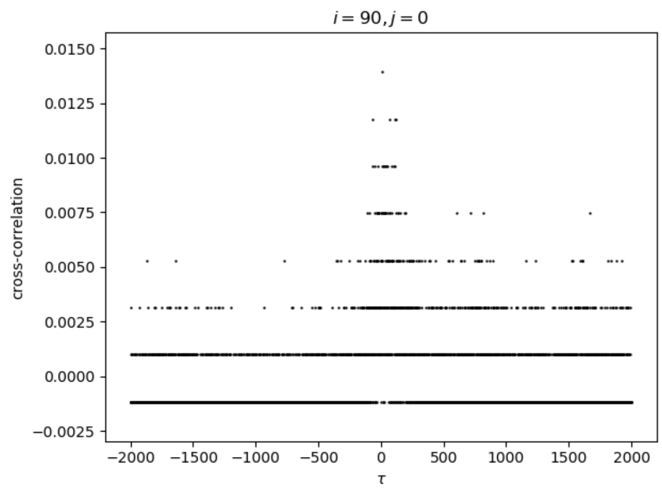


i=0, j=40



i=0, j=50



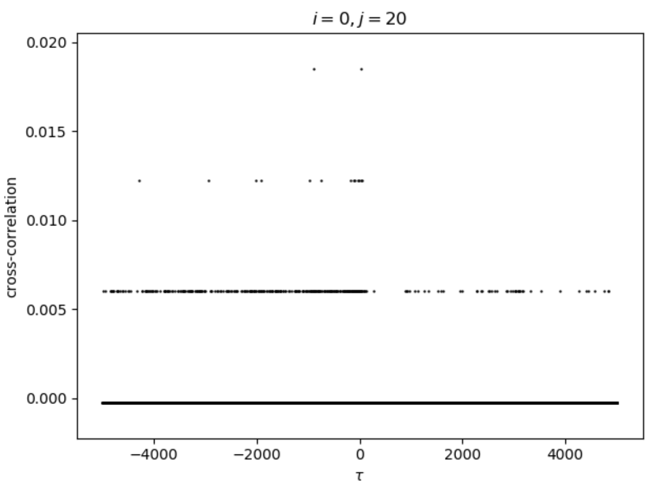
<table>
<tr><th colspan="2">Verifying symmetry: Cij(tau) = Cji(-tau)</th></tr>
<tr><th>i=0, j=40</th><th>i=40, j=0</th></tr>
<tr><td>
</td><td>
</td></tr>
<tr><th>i=0, j=90</th><th>i=90, j=0</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

Some suggestion

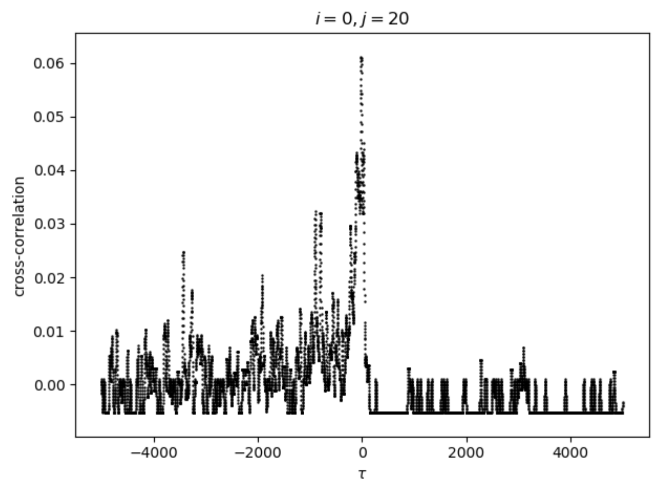
Cross-correlations are based on spike time points, but this could underestimate the "actual" correlations as statistical noise could "displace" the spike points near where the spikes actually occur. Of course the most accurate calculations would be based on the values of the time series but this is too computational costly. One suggestion is to add a small "padding" near the spike time points. For example, detected spikes for a certain neuron occur at t=[200,400]. With a padding of p=2 time units, t is modified to [198,199,200,201,202,398,…]. Here spikes are assumed to take place over a time period instead of a time point. Also, p < time scale of spikes so that the computed CC lower-bounds the "actual" CC [Guess Only], and it has to be ensured that ti+p < tj-p for all spike time points ti<tj. The method sacrifices computational speed on the order of p^2 (in calculating CC).

i=0, j=20

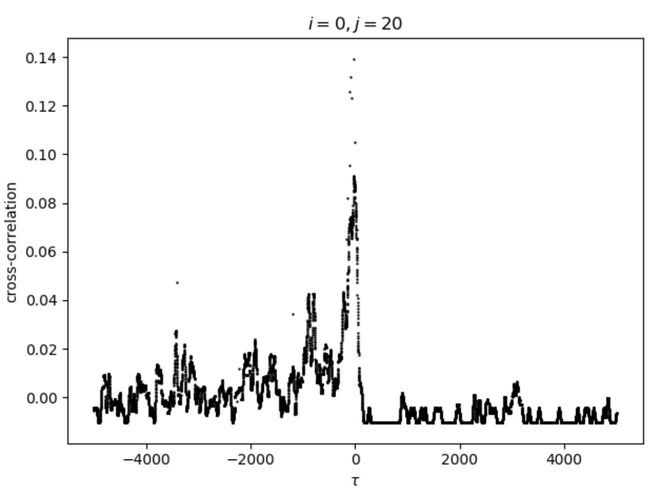
Padding p=0 (original)



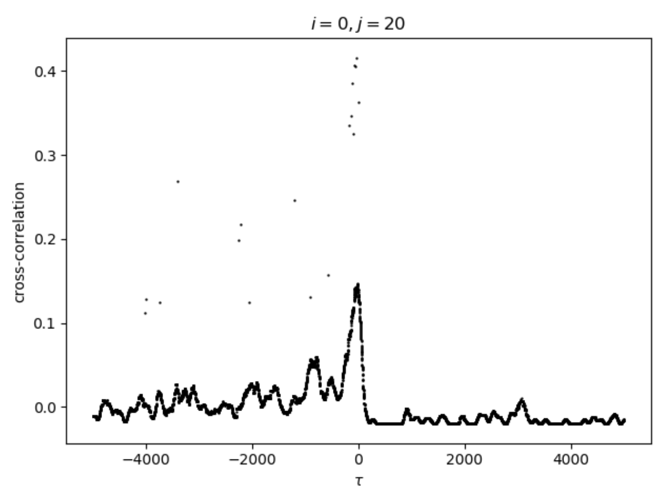
Padding p=10



Padding p=20



Padding p=40

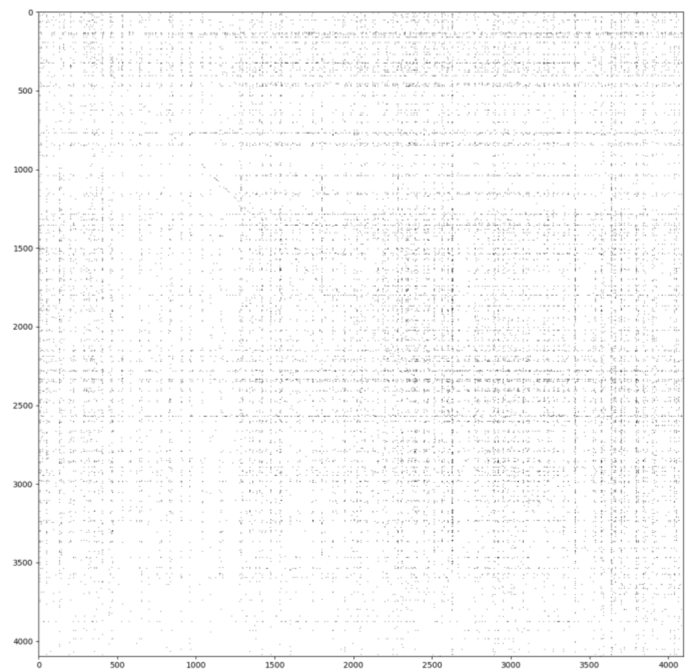
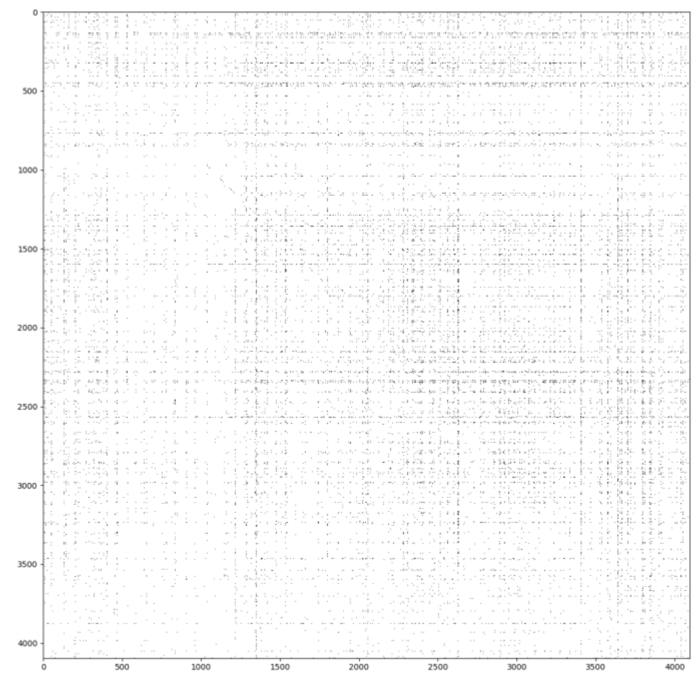


FNCCH on measured spike data

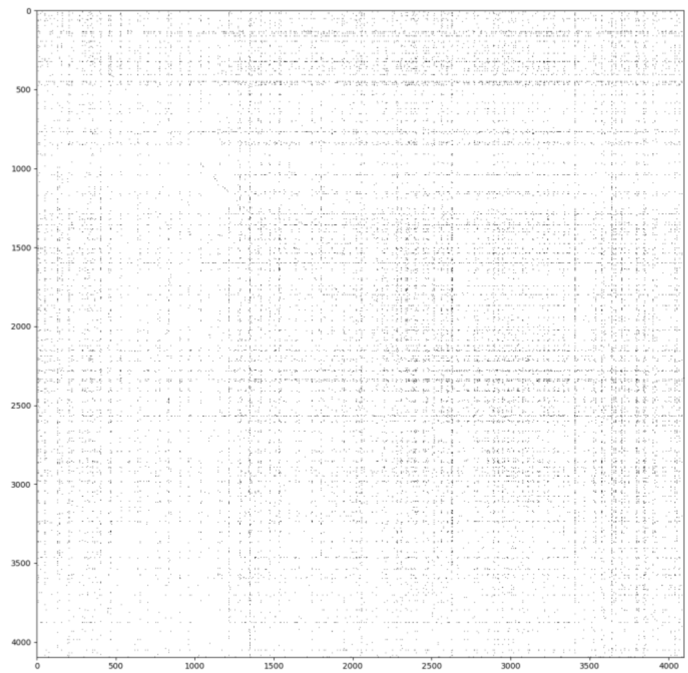
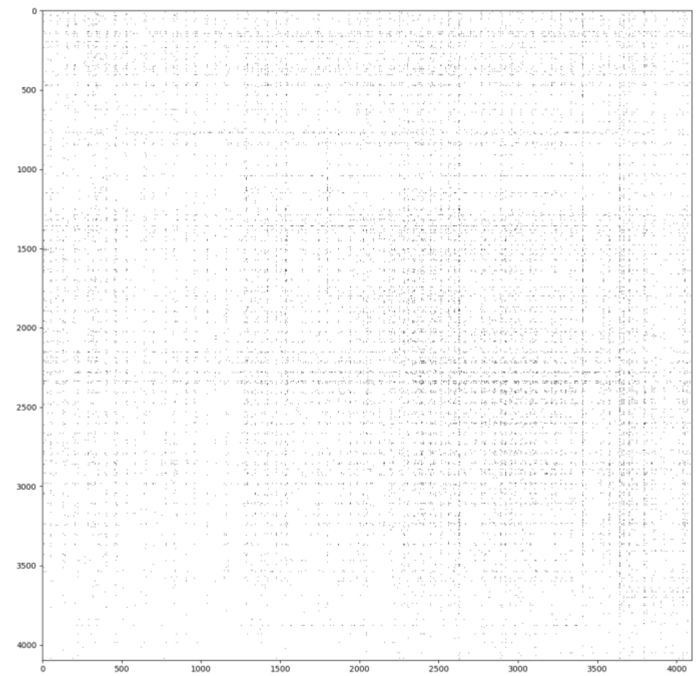
Observation: peaks in CCHs occur in the close neighborhood of $\tau=0$ => could use shorter W=2000 or 4000
gij are found to be all positive for both W=2000 and 4000

gij>0 is not unexpected. CC = 0 for most τ, and suppose CC(τ) = some positive const if $\tau = \tau'$ and 0 otherwise. Then the normalized CC(τ) ~ unnormalized CC(τ) as mean CC ~ 0. Thus, the extracted CC*~CC(τ')>0 so gij>0 almost always.

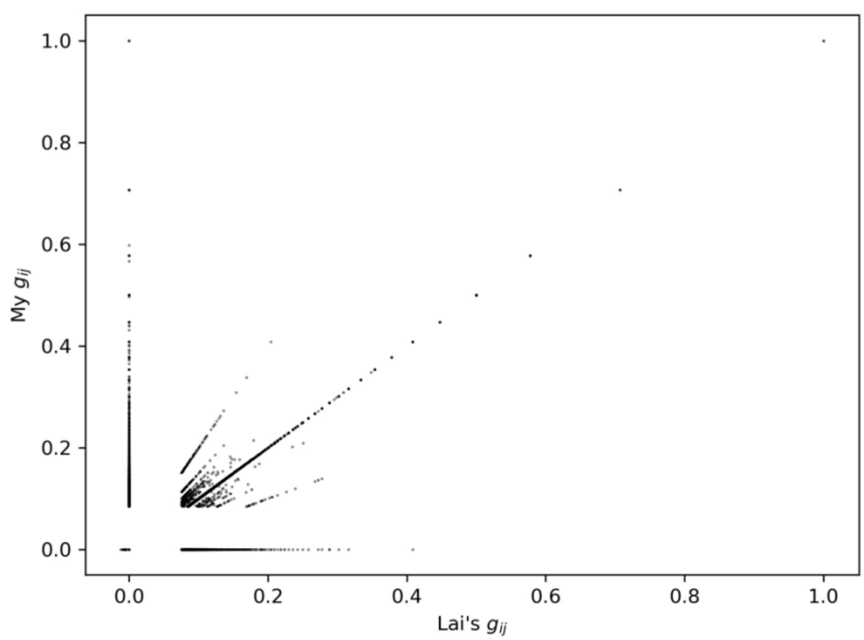
In the reconstructed adjacency matrix, grid/line patterns and a slight diagonal pattern are seen.

Adjacency matrix (FNCCH)	
W=2000	**W=4000**
	
Connection prob. = 1.44%	Connection prob. = 1.47%

Comparison with Lai's (collaborator) FNCCH reconstructed matrix

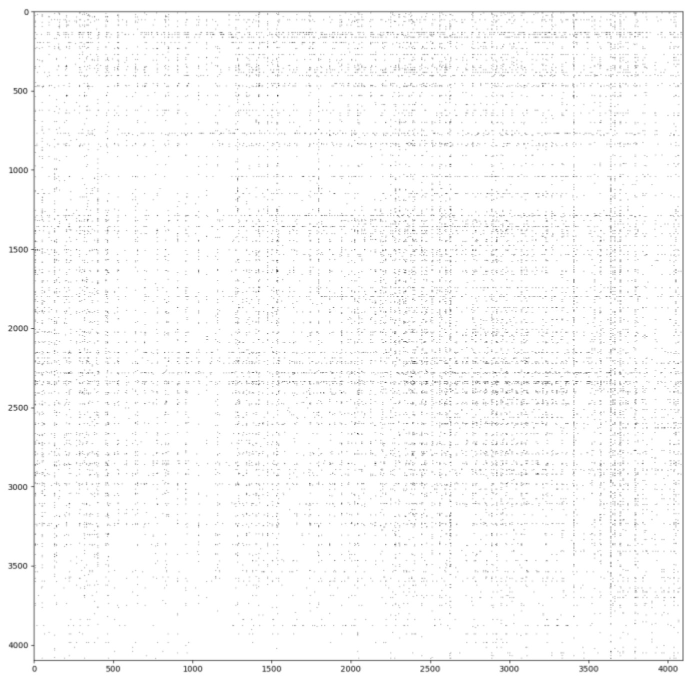
<table>
<tr><th colspan="4">Adjacency matrix (FNCCH)</th></tr>
<tr><th colspan="2">(a) W=4000</th><th colspan="2">(b) Lai's</th></tr>
<tr><td colspan="2">
Connection prob. = 1.47% (denser)</td><td colspan="2">
Connection prob. = 1.01% (less dense)</td></tr>
<tr><th colspan="4">Regarding Lai's matrix as the true matrix</th></tr>
<tr><th>Sensitivity (Correctness for existing links)</th><th>Specificity (Correctness for non-existing links)</th><th>Error rate (FN/nL)</th><th>Error rate (FP/nL)</th></tr>
<tr><td>0.6537</td><td>0.9918</td><td>0.3463</td><td>0.8098</td></tr>
</table>

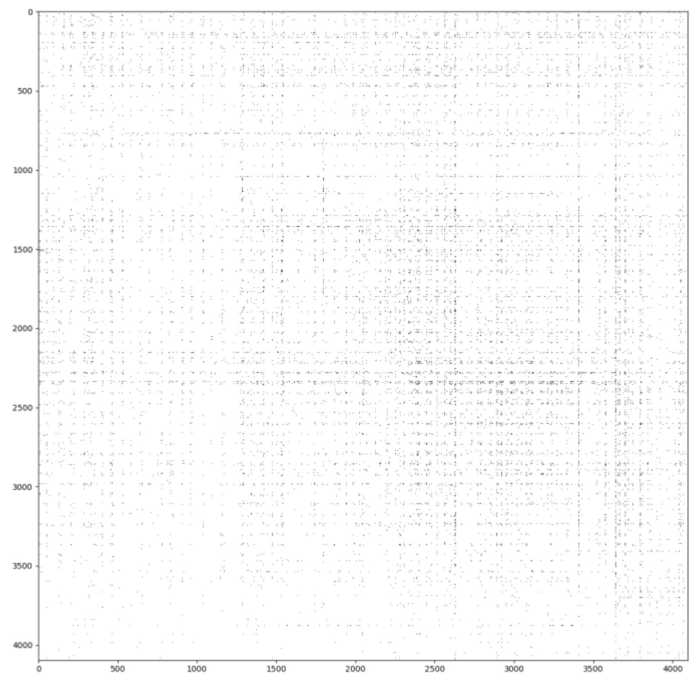
Comparison of our calculated correlations



Adjacency matrix (FNCCH)

(a) W=200 (Physical criteria applied)	(b) Lai's





Regarding Lai's matrix as the true matrix			
Sensitivity (Correctness for existing links)	**Specificity (Correctness for non-existing links)**	**Error rate (FN/nL)**	**Error rate (FP/nL)**
0.9157	0.9968	0.0843	0.3111

Notes on Functional Normalized Cross-Correlation Histogram method

We follow the procedure employed in [1] to identify the links between the electrodes (number of electrodes=N_{node}) in the MEA measurement. Consider two nodes (electrodes) i and j with their corresponding binary (0 or 1) spiking time-series data $f_i(t)$ and $f_j(t)$, the time-lag normalized cross-correlation is given by

$$C_{ij}(\tau) = \frac{1}{\sqrt{N_i N_j}} \sum_{s=1}^{N_i} f_i(t_s) f_j(t_s + \tau), \tag{1}$$

where t_s is the spiking time of the s^{th} spike, N_i and N_j are the total number of spikes of nodes i and j respectively. A time window of width W is chosen for the time-lag with $\tau \in [-\frac{W}{2}, \frac{W}{2}]$. The value τ^* at which the maximum of $|C_{ij}(\tau) - \overline{C_{ij}}|$ occurs is located, where $\overline{C_{ij}} \equiv \frac{1}{W}\sum_{t=-W/2}^{W/2} C_{ij}(t)$ is the average correlation over the time window W. If $C_{ij}(\tau^*) \equiv C^*_{ij}$ is a peak (trough), then an excitatory (inhibitory) link is identified from i to j. The link is also spatio-temporally filtered by the physical constraint that $\tau^* >$ the minimum delay time of 1 ms, and the propagation speed cannot exceed the maximum signal propagation speed $v_{max} \simeq 400$mm/s. Finally a hard thresholding procedure is applied to select the links of significant magnitude: the means (μ) and standard deviations (σ) are computed for all the $N_{node} \times N_{node}$ matrix elements of $|C^*_{ij}|$ separately for the excitatory and inhibitory links, and links of magnitude less than $\mu + n\sigma$ will be discarded. We choose $n = 2$ and $n = 1$ respectively for thresholding the excitatory and inhibitory links respectively. The magnitude of the filtered and thresholded C^*_{ij} is identified as the connection strength of the link $j \to i$.

Column 1 and Column 2 are nodes n1 and n2 (node index runs from 1 to 4095)
Column 3 = value of tau^* at which the maximum value occurs;
if tau^* >0 then link points from node n1 to node n2; if tau^*<0 then link points from node n2 to node n1
Column 4 = max value of |Cij(tau^*)-<Cij(tau)>|, with the minus sign
when Cij(tau^*) < <Cij(tau)> ==> this is an inhibitory link
Column 5 = Cij(tau^*)
Column 6 = <Cij(tau)> = 1/W sum_{tau= -W/2}^{W/2} Cij(tau)

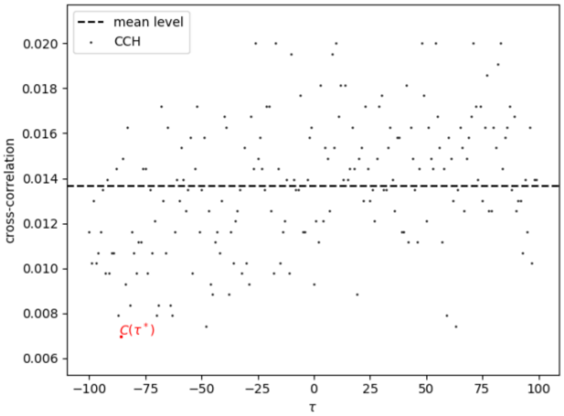
Reference
```
656 648 -8  0.20704928  0.209656954  0.00260767364
57 521 -86 -0.00668063294  0.0069768345  0.0136574674
26 337 5  0.0148042757  0.0300830323  0.0152787566
26 1857 -100 -0.00848391652  0.00425651763  0.0127404341
```

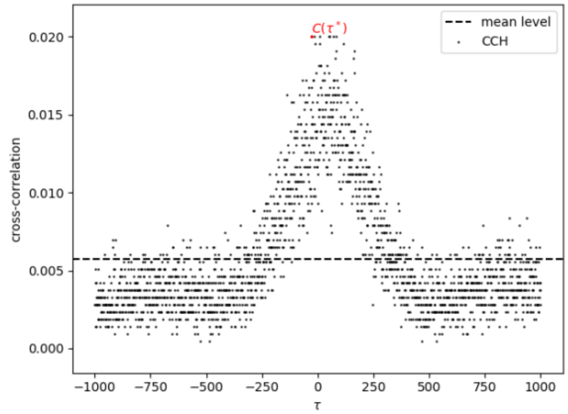
w=200
```
656 648 -8 0.20703625525257888 0.20965696734438366 0.0026207120918047954
57 521 -86 -0.006688458909711 0.006976834745873783 0.013665293655584783
26 337 5 0.014775889673903402 0.030083032907665577 0.015307143233762175
26 1857 -100 -0.00845450809755311 0.004256517607326926 0.012711025704880035
```

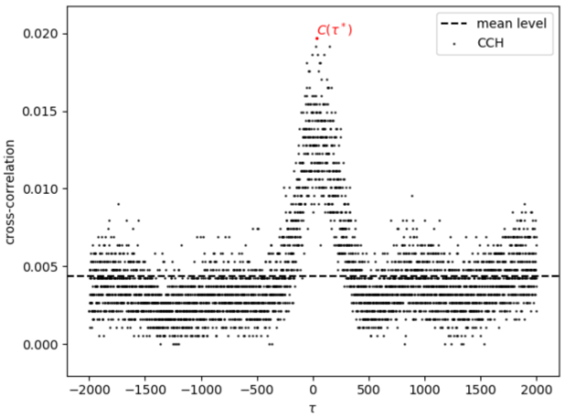
w=2000
```
656 648 -8 0.20939489613520318 0.20965696734438366 0.00026207120918047953
57 521 -26 0.014237394104746434 0.02000025960483818 0.005762865500091746
26 337 5 0.024581678272742477 0.030083032907665577 0.0055013546349230984
26 1857 35 0.014377984412849442 0.019686393933887034 0.0053084095210375925
```

w=4000
```
656 648 -8 0.20952593173979342 0.20965696734438366 0.00013103560459023976
57 521 -26 0.015112986865353593 0.02000025960483818 0.004887272739484585
26 337 5 0.025805481654007513 0.030083032907665577 0.004277551253658065
26 1857 35 0.01530191476598984 0.019686393933887034 0.004384479167897192
```

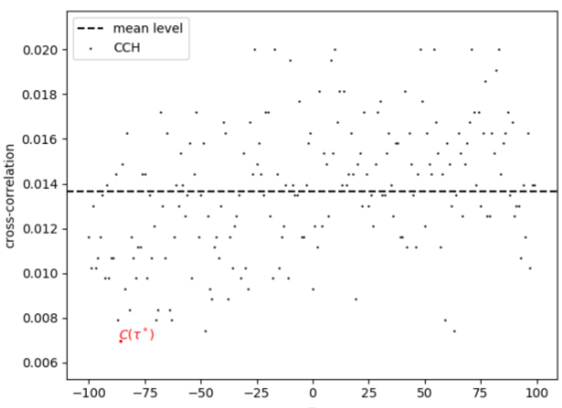
i=26, j=1857

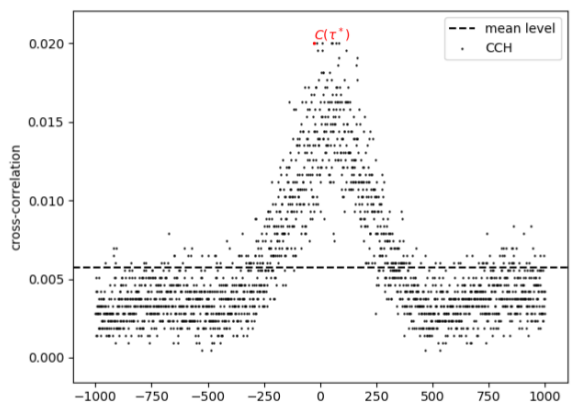


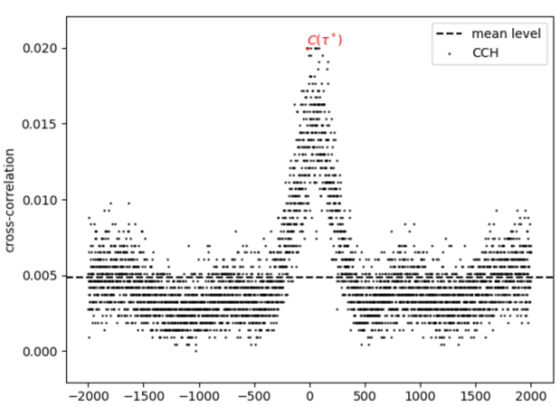




i=56, j=520



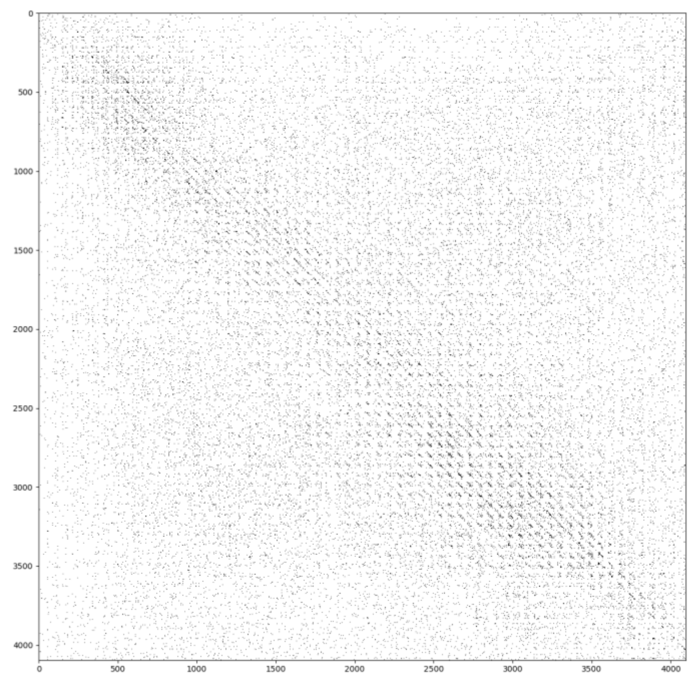
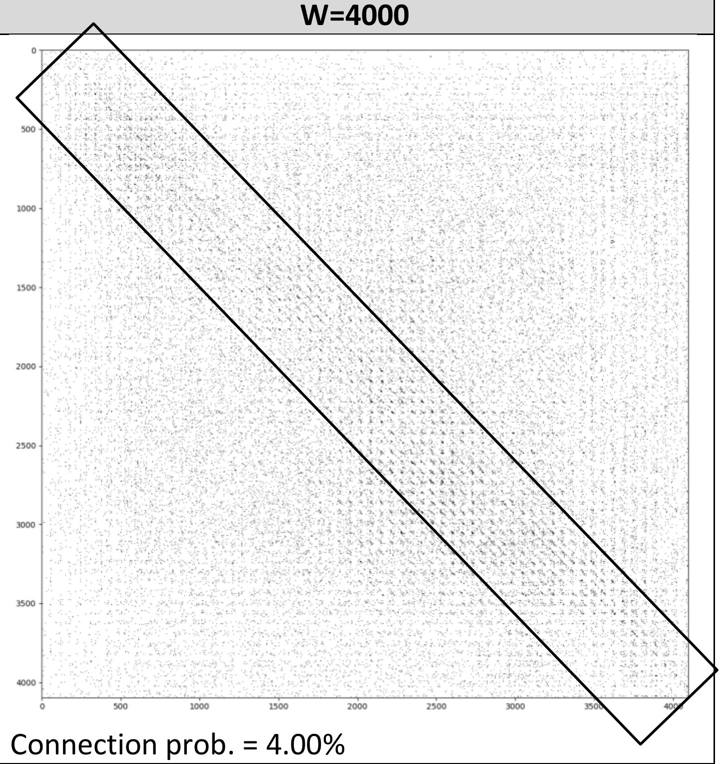




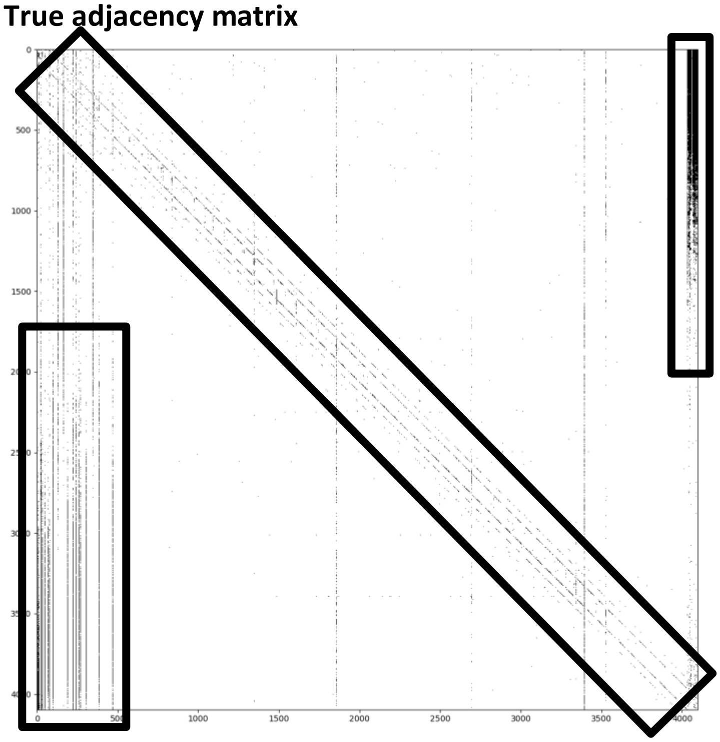
FNCCH on FHN model data

In the reconstructed adjacency matrix, a clear diagonal pattern is seen, with no special pattern for non-diagonal region. The diagonal pattern is consistent with the true adjacency matrix. But it fails to recover some other patterns, such as the line pattern for small j and the concentration of points for small i and large j.

Adjacency matrix (FNCCH)

W=2000	W=4000
 Connection prob. = 3.06%	 Connection prob. = 4.00%

True adjacency matrix



Connection prob. = 1.42%

The three key features are boxed:
- line pattern (large i, small j)
- concentration of points (small i, large j)
- diagonal line pattern

3. PRE covariance method

PRE method on FHN model data

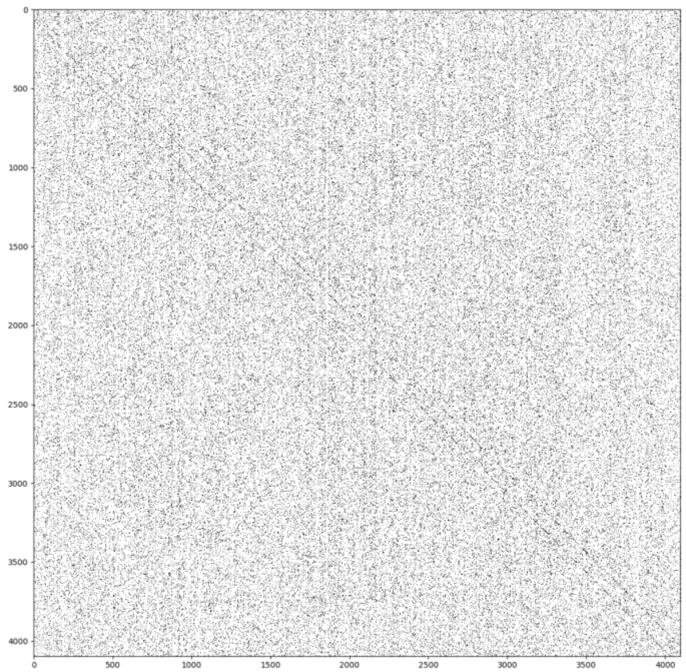
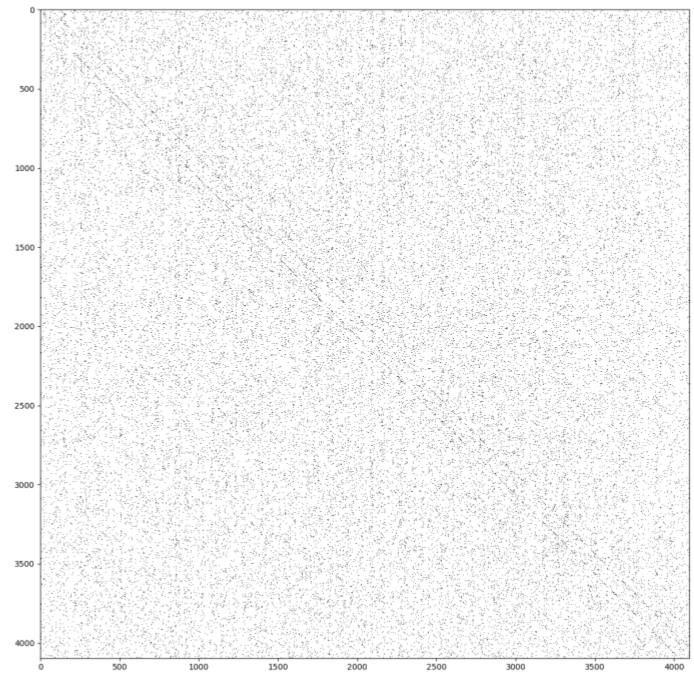
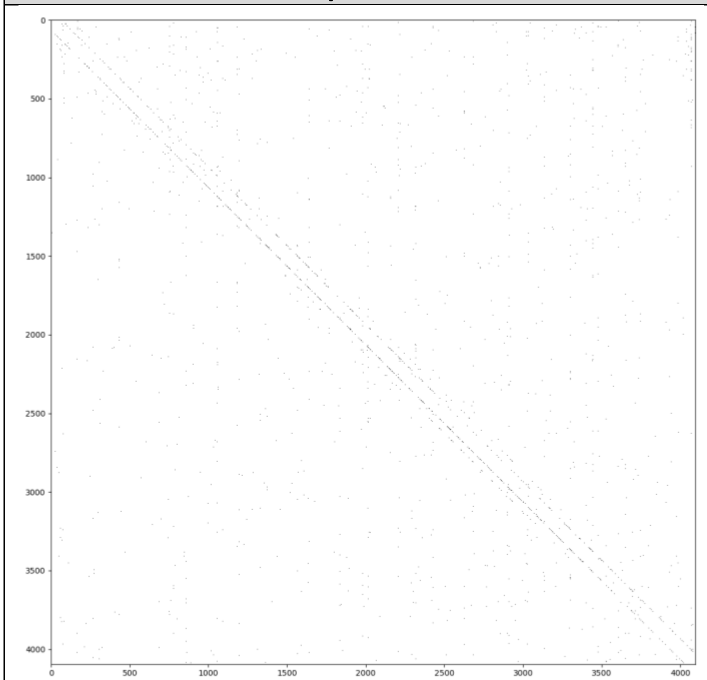
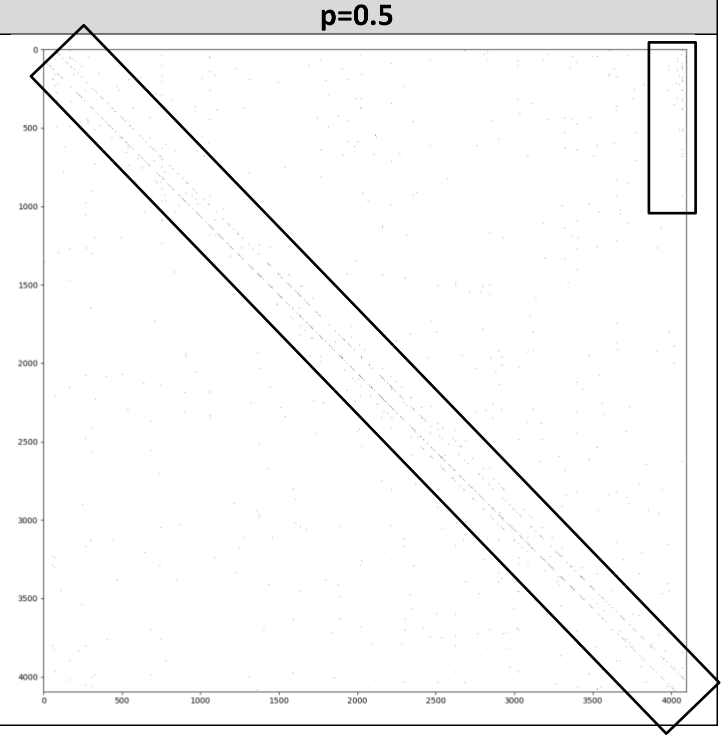
Algorithm. $\mathbf{M} = \frac{1}{\tau}\log \mathbf{K}_\tau \mathbf{K}_0^{-1}$. For non-diagonal entries of $\mathbf{M}$, for each j, Mij (i.e. each column) is classified using Gaussian mixture model into **three groups** [negative, near zero, positive], with, for example, fitted means [-2.05, 0.02, 2.10] returned from the model. For each i, a vector of probabilities of Mij belonging to each group is calculated, and the adjacency matrix entry (i,j) is inferred to be 1 if the probability of Mij belonging to the negative or positive group > probability threshold p=0.5 (i.e. Mij is sufficiently "different" from zero).

For threshold >= 0.5, there is at most one group that the node is identified to locate in. For threshold < 0.5, there can be more than one group. The node is then identified to locate in the group with higher probability. For example, if probability vector = [0.42, 0.45, 0.03], with threshold = 0.4, it can belong to the negative or near zero group, but 0.45>0.42 so it belongs to near zero group.

The time series of the FHN model for 2e6 steps were stored. The first 2e5-step series are extracted to compute the 2e5-step Mij, as a preliminary test. Using p=0.5, the reconstructed adjacency matrix is "noisy" but a "double-lined" diagonal pattern can still be seen, which is consistent with the true adjacency matrix. Other patterns are not obvious.

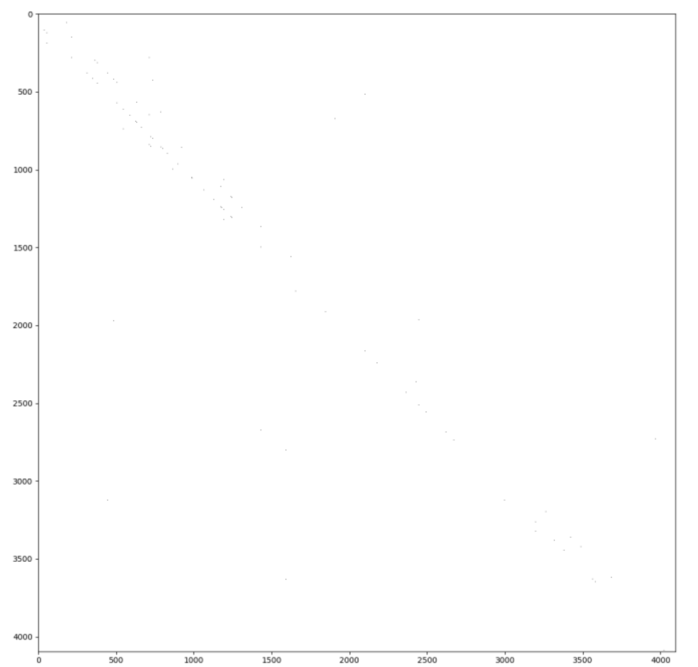
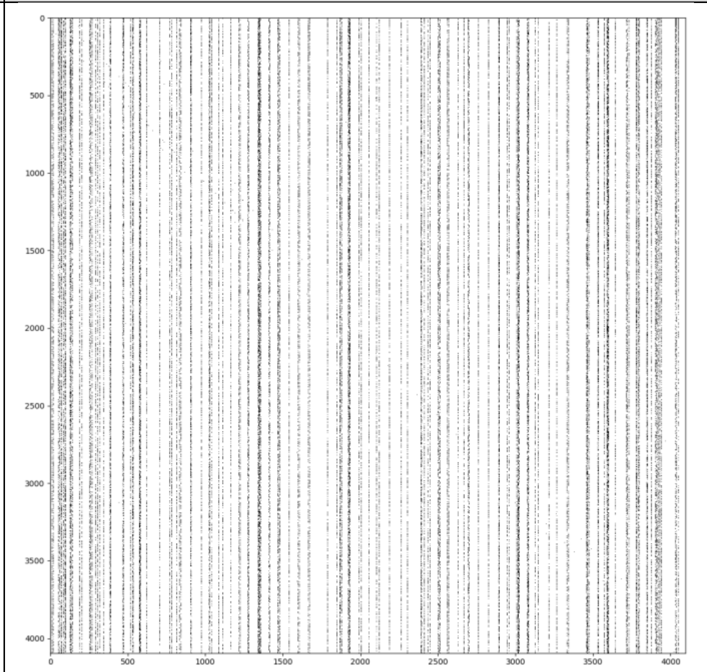
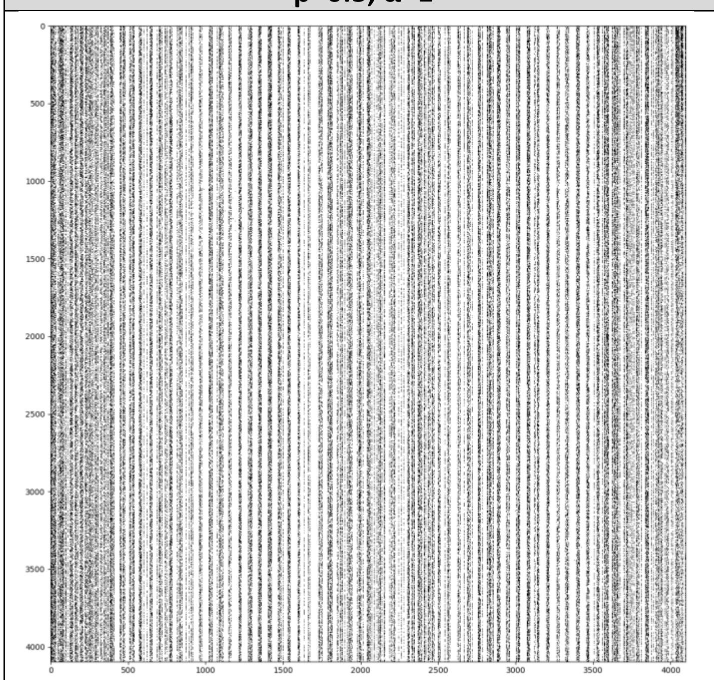
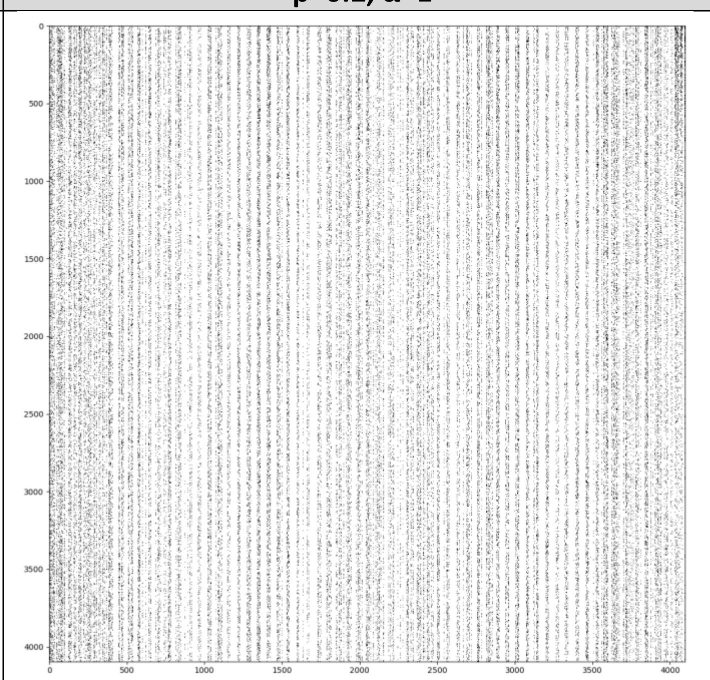
The 1e6-step series are extracted to compute the 1e6-step Mij, and this time the reconstructed adjacency matrix is much "cleaner", retaining the "double-lined" diagonal pattern with sparse points in the non-diagonal region. A slight concentration of points is seen at the right corner (small i and large j), which corresponds to the true matrix.

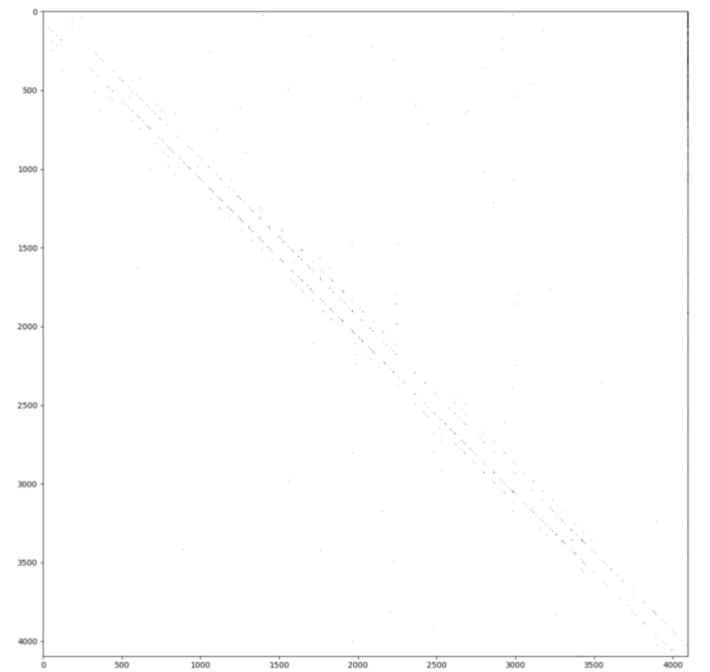
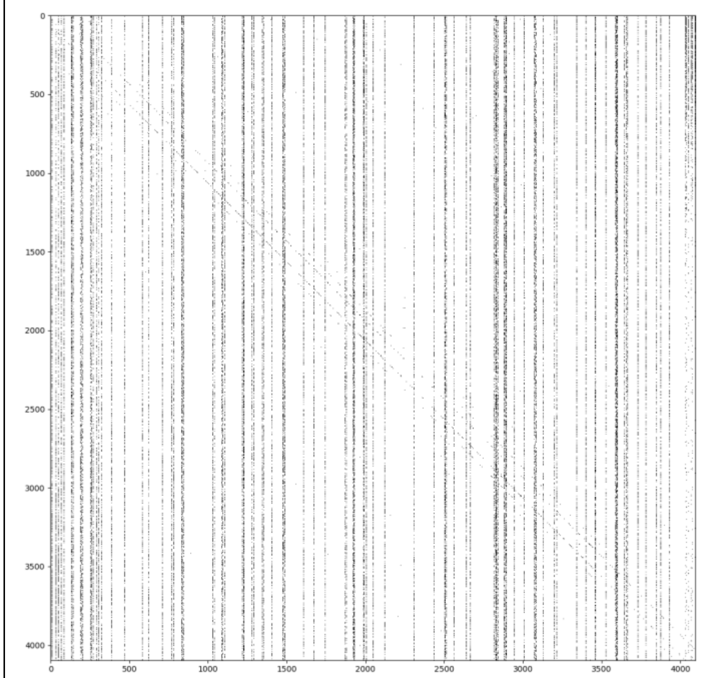
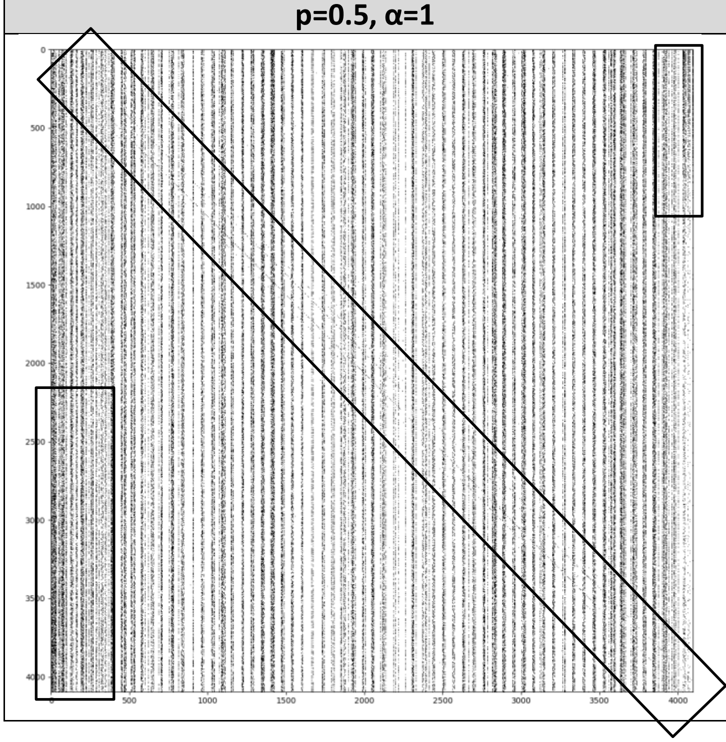
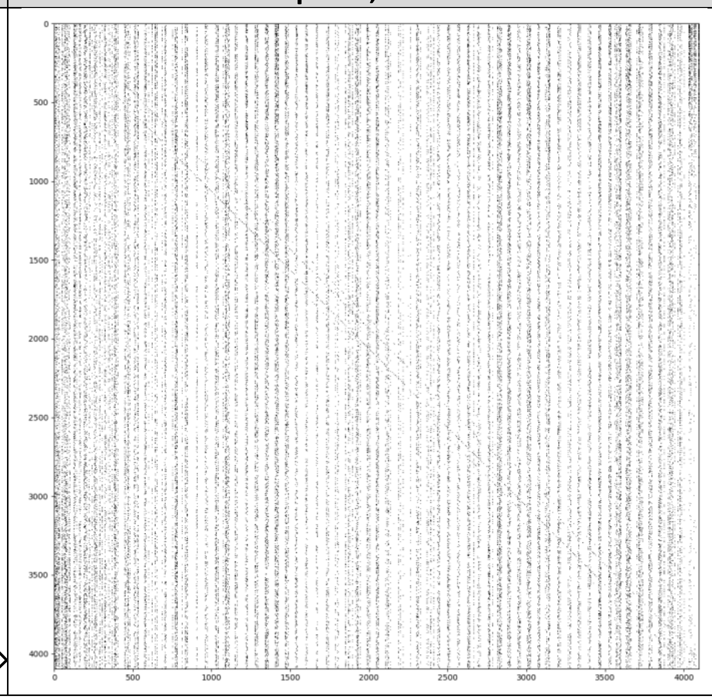
Three groups [negative, near zero, positive]

<table>
<tr><th colspan="2">Adjacency matrix (PRE method)</th></tr>
<tr><th colspan="2">2e5 steps Mij</th></tr>
<tr><th>p=0.5</th><th>p=0.8</th></tr>
<tr><td></td><td></td></tr>
<tr><th colspan="2">1e6 steps Mij</th></tr>
<tr><th>p=0.4</th><th>p=0.5</th></tr>
<tr><td></td><td></td></tr>
</table>

Two groups [near zero, non-zero]

Algorithm. $\mathbf{M} = \frac{1}{\tau}\log \mathbf{K}_\tau \mathbf{K}_0^{-1}$. For non-diagonal entries of $\mathbf{M}$, for each j, Mij (i.e. each column) is classified using Gaussian mixture model into **two groups** [near zero, non-zero], with, for example, fitted means [0.02, 2.10] returned from the model. If the difference between the fitted means < α*sigma(near zero group) (i.e. the two fitted Gaussians are "too close"), then all other nodes are inferred to be unconnected to node j (i.e. column j of adjacency matrix left empty). For each i, a vector of probabilities of Mij belonging to each group is calculated, and the adjacency matrix entry (i,j) is inferred to be 0 (unconnected) if the probability of Mij belonging to the near-zero group > probability threshold p=0.5.

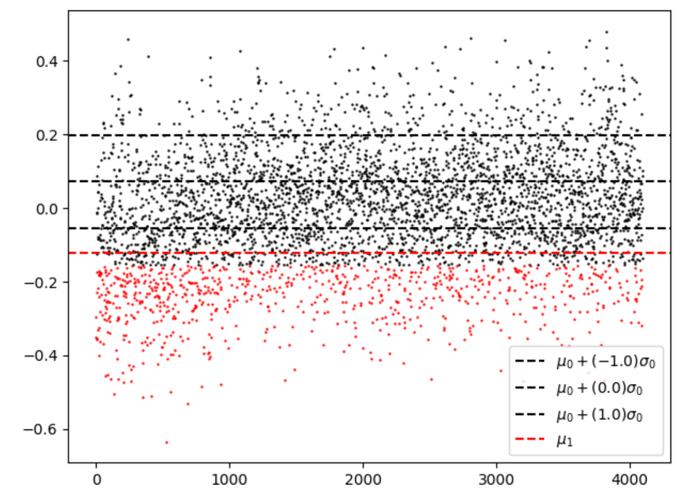
<table>
<tr><th colspan="2">1e6 steps Mij</th></tr>
<tr><th>p=0.5, α=2</th><th>p=0.5, α=1.5</th></tr>
<tr><td></td><td></td></tr>
<tr><th>p=0.5, α=1</th><th>p=0.2, α=1</th></tr>
<tr><td></td><td></td></tr>
</table>

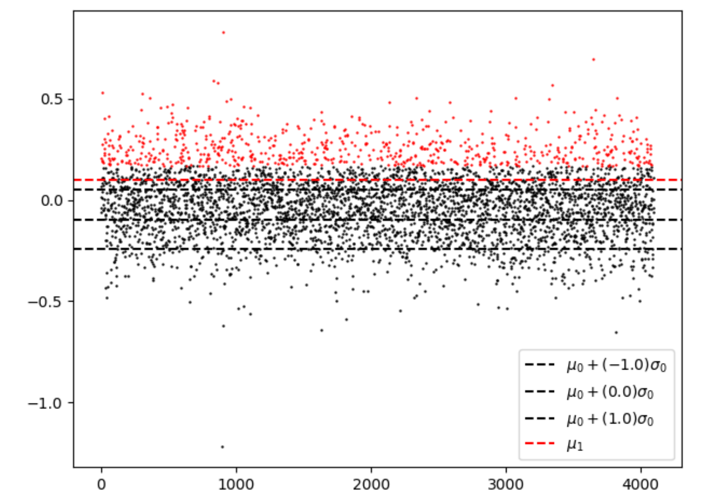
2e6 steps Mij	
p=0.5, α=2	**p=0.5, α=1.5**
	
p=0.5, α=1	**p=0.2, α=1**
	

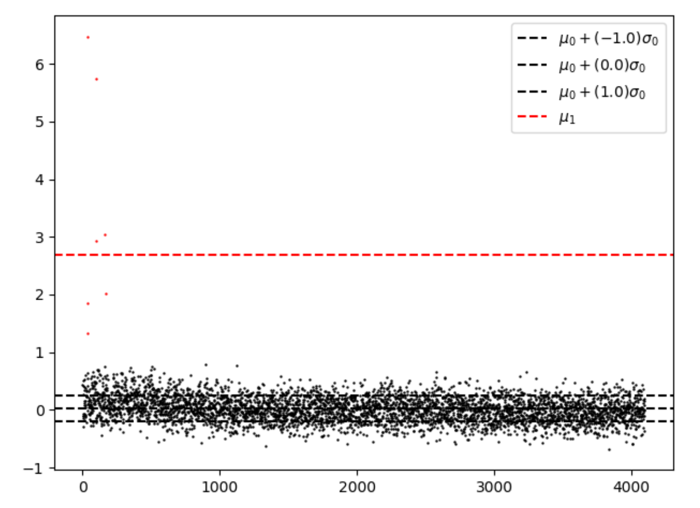
Visualizing the classification

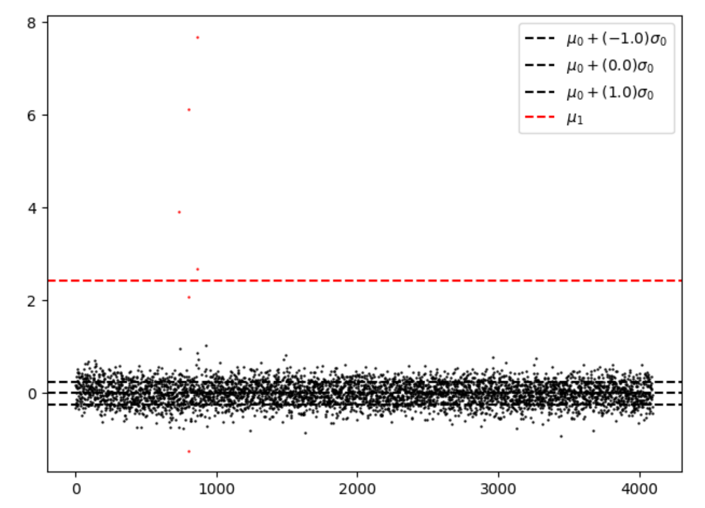
Black = near-zero group (null) with params µ0,σ0; Red = non-zero group with params µ1,σ1
x-axis = node index, y-axis = Mij

The classifications using settings (p=0.2,α=1) and (p=0.5,α=1) are visualized. (p=0.2,α=1) is a more stringent requirement in that in a column, for an Mij, if the posterior probability belonging to null group > 0.2, then it is determined to belong to the null. Equivalently, to belong to the non-zero group, the probability has to be > 0.8 (a more stringent requirement). Data points have to be significantly further away from µ0 in order to be classified into the non-zero group, as seen from the following plots.

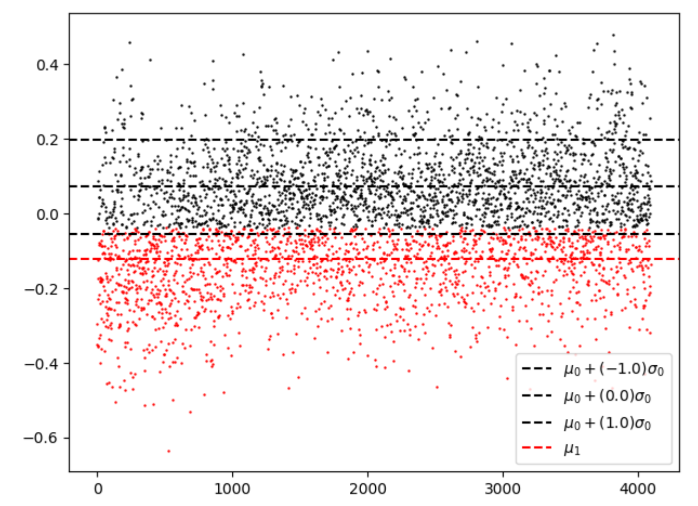
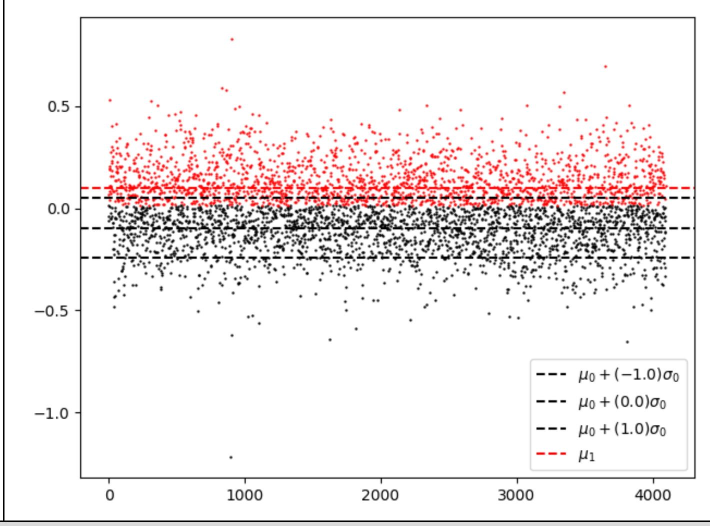
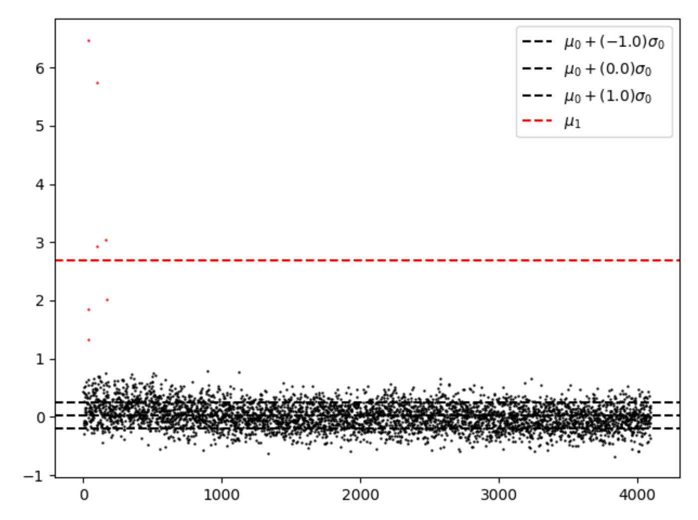
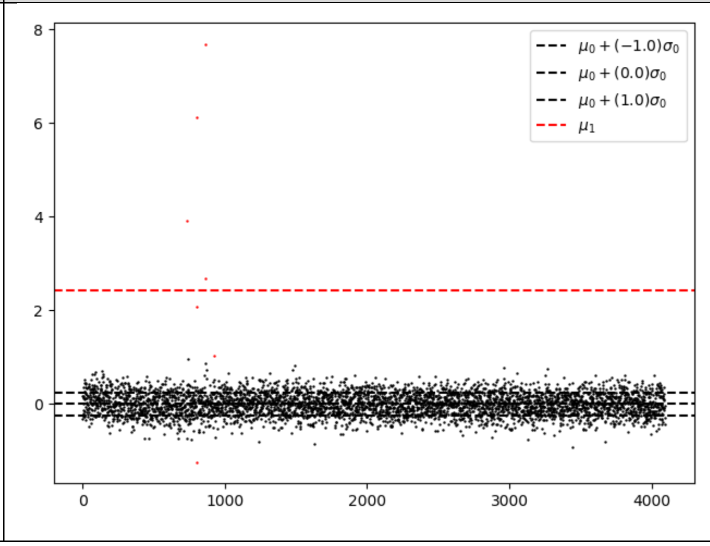
<table>
<tr><th colspan="2">2e6 steps Mij – p=0.2, α=1</th></tr>
<tr><th colspan="2">Mild separation</th></tr>
<tr><th>j=200</th><th>j=900</th></tr>
<tr><td>


</td><td>


</td></tr>
<tr><th colspan="2">Extreme separation</th></tr>
<tr><th>j=100</th><th>j=800</th></tr>
<tr><td>


</td><td>


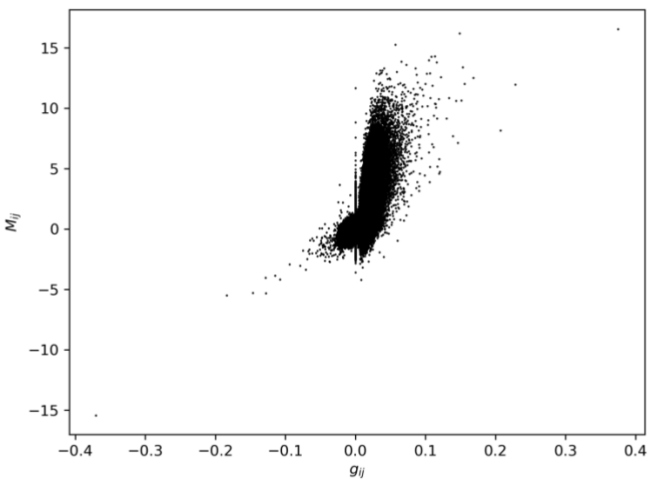
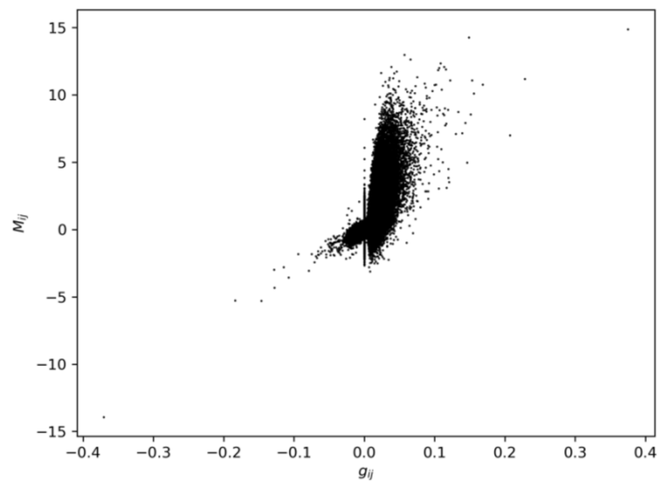
</td></tr>
</table>

2e6 steps Mij – p=0.5, α=1	
Mild separation	
j=200	**j=900**
	
Extreme separation	
j=100	**j=800**
	

Sensitivity & Specificity

Method Params	Sensitivity (Correctness for existing links)	Specificity (Correctness for non-existing links)	Error rate (FN/nL)	Error rate (FP/nL)
FNCCH				
W=2000	0.0295	0.9858		
W=4000	0.0286	0.9855		
PRE method (Using 1e6-step Mij)				
Three groups [negative, near zero, positive]				
p=0.4	0.0784	0.9991		
p=0.5	0.0778	0.9995		
Two groups [near zero, non-zero]				
p=0.5, α=2	0.0079	0.9999	0.9921	0.0008
p=0.5, α=1.5	0.2234	0.9335	0.7766	4.6329
p=0.5, α=1	0.4250	0.8104	0.5750	13.2057
p=0.2, α=1	0.2301	0.9228	0.7699	5.3754
PRE method (Using 2e6-step Mij)				
Two groups [near zero, non-zero]				
p=0.5, α=2	0.0667	0.9999	0.9333	0.0094
p=0.5, α=1.5	0.2558	0.9568	0.7442	3.0090
p=0.5, α=1	0.4547 (highest)	0.8585	0.5453	9.8568
p=0.2, α=1	0.2798	0.9428	0.7202	3.9839

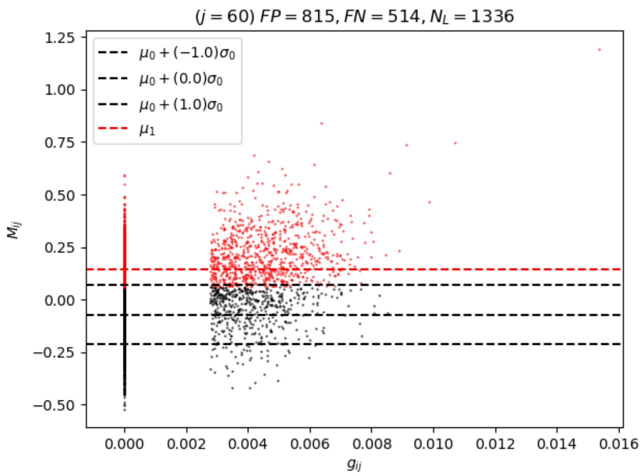
Mij vs gij plot

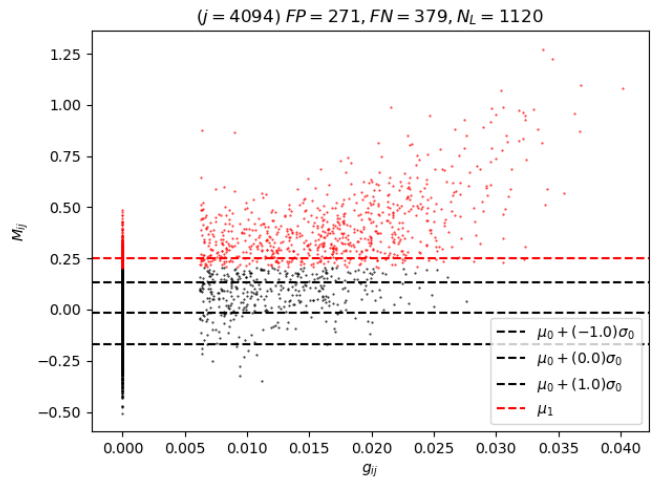
1e6 steps	2e6 steps
	

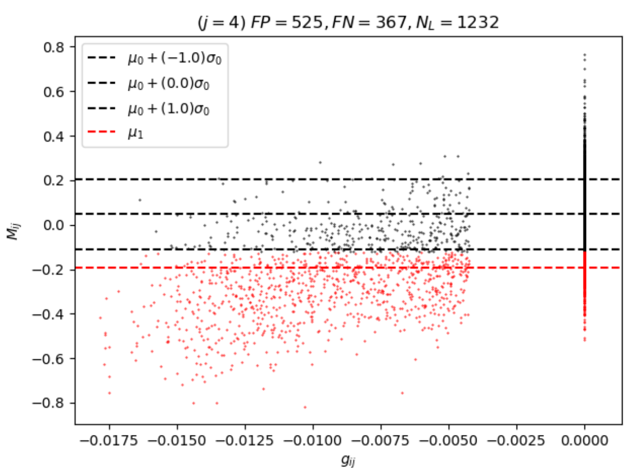
Observations

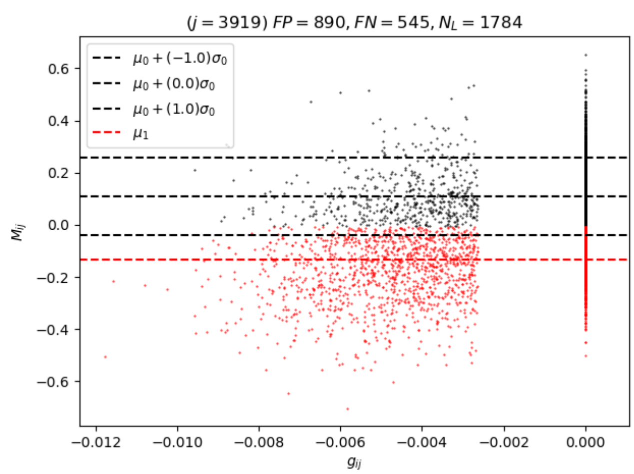
1. Mij do not correlate linearly with gij; instead, slopes corresponding to +ve & -ve gij are different.
2. The line of data points at gij=0 has a scale comparable to points with gij<0, so there can be more errors (e.g. FP) when classifying inhibitory nodes. Classifying excitatory nodes is relatively accurate, in particular points in the top-right region.
3. More time steps lead to higher accuracy in calculation of Mij, e.g. the middle line of points and points with gij<0 “contract” and become more concentrated (i.e., have smaller errors). But the improvement is not significant.

Mij vs gij plot for nodes with large nL [to visualize the classification]

<table>
<tr><th colspan="2">Excitatory (out-strength > 0)</th></tr>
<tr><th>j=60</th><th>j=4094</th></tr>
<tr><td>


</td><td>


A slight linear pattern is seen
</td></tr>
<tr><th colspan="2">Inhibitory (out-strength < 0)</th></tr>
<tr><th>j=4</th><th>j=3919</th></tr>
<tr><td>


A slight linear pattern is seen
</td><td>


</td></tr>
</table>

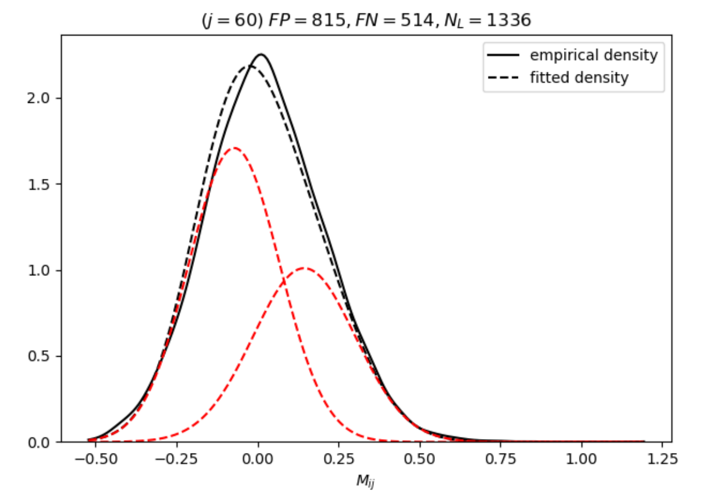
Plots for other nodes with large nL

1. (excitatory) https://drive.google.com/file/d/1NRdMvhLI19GgR-2axGIamTR_S3QIEUsz/view?usp=sharing
2. (inhibitory) https://drive.google.com/file/d/1nxA-iU2w1-M1Y3Rck0IBr0NFvjORN6FN/view?usp=sharing

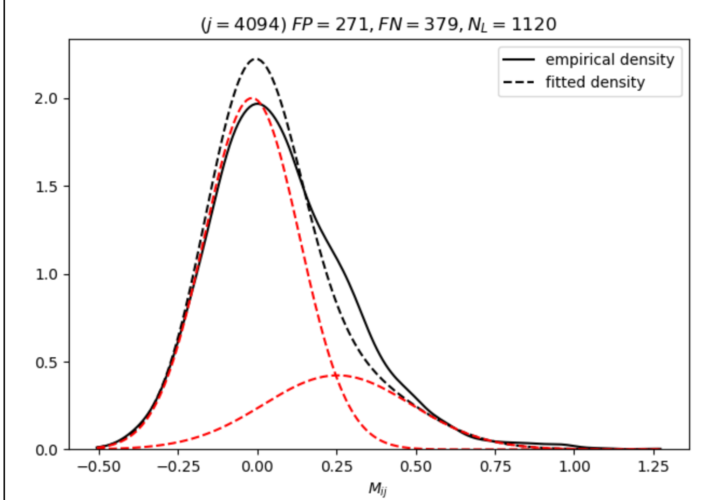
Distribution of Mij for nodes with large nL

Excitatory (out-strength > 0)

j=60

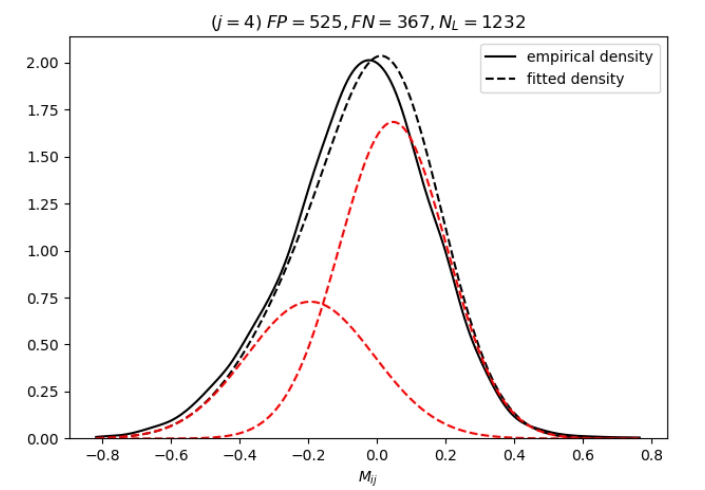


j=4094

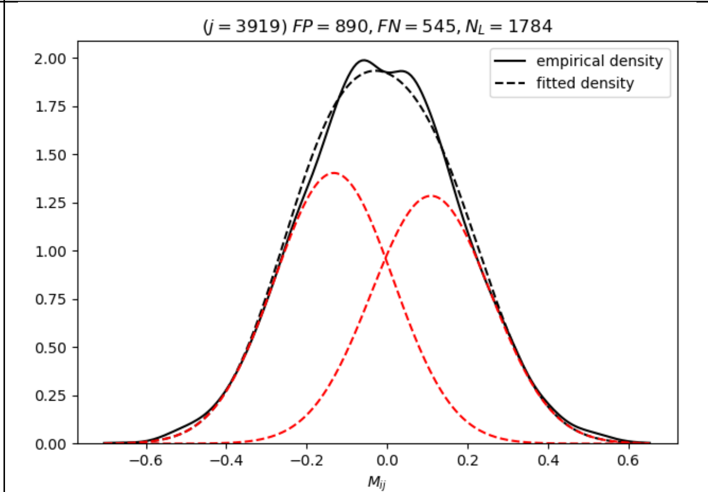


Inhibitory (out-strength < 0)

j=4



j=3919



Plots for other nodes with large nL

1. (excitatory) https://drive.google.com/file/d/1jFySeEc2sjwMtShl0ucX2cwSxqabz5wG/view?usp=sharing
2. (inhibitory) https://drive.google.com/file/d/1krdCMajYD-up8asd1q1Kew2L4hFLIV4f/view?usp=sharing

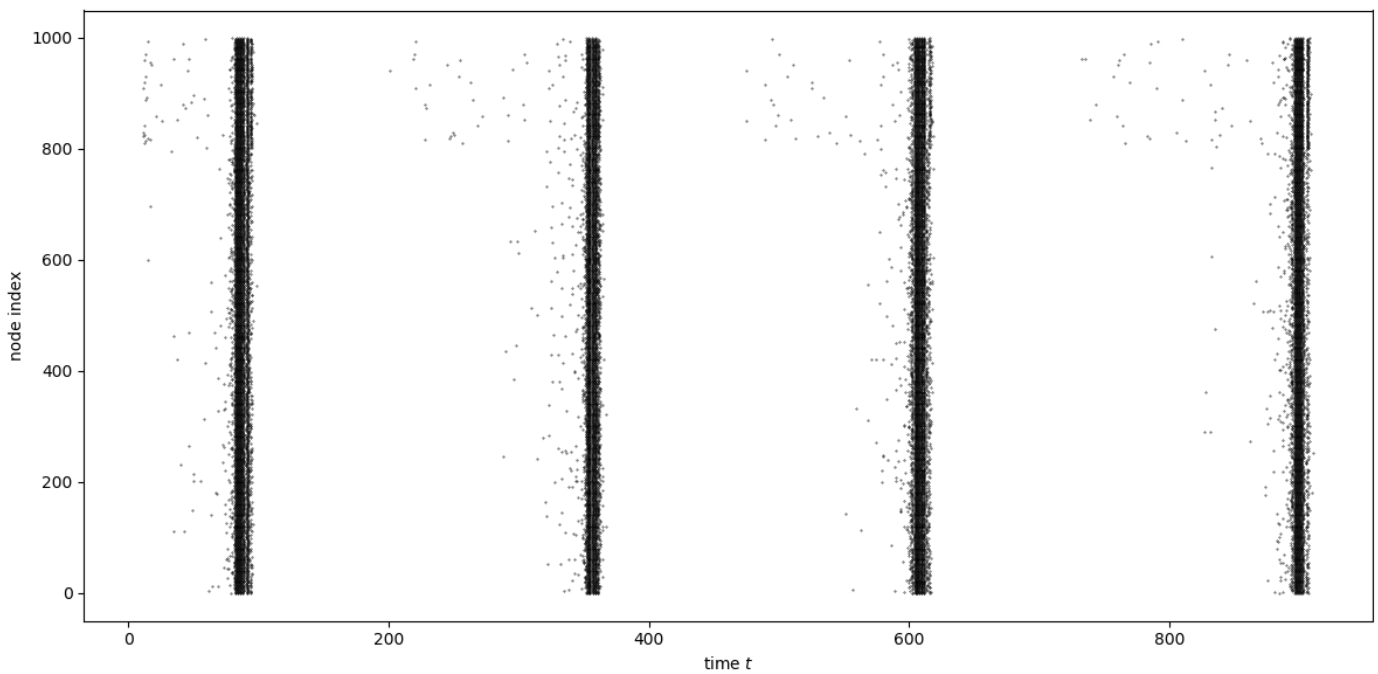
4. Spiking neuron model [EM Izhikevich]

Code in original paper

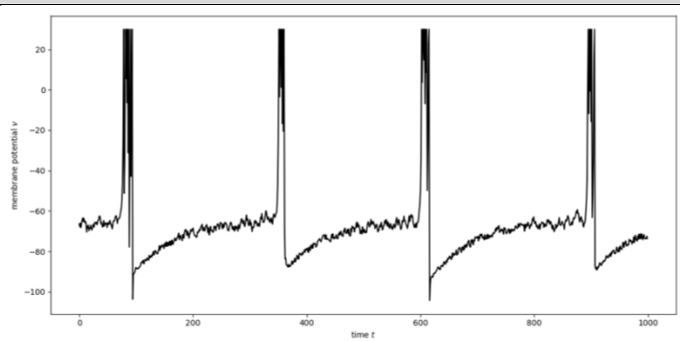
Tunable params: a,b,c,d

```
% Created by Eugene M.  Izhikevich, February 25, 2003
% Excitatory neurons    Inhibitory neurons
Ne=800;                 Ni=200;
re=rand(Ne,1);          ri=rand(Ni,1);
a=[0.02*ones(Ne,1);     0.02+0.08*ri];
b=[0.2*ones(Ne,1);      0.25-0.05*ri];
c=[-65+15*re.^2;        -65*ones(Ni,1)];
d=[8-6*re.^2;           2*ones(Ni,1)];
S=[0.5*rand(Ne+Ni,Ne),  -rand(Ne+Ni,Ni)];

v=-65*ones(Ne+Ni,1);    % Initial values of v
u=b.*v;                 % Initial values of u
firings=[];             % spike timings

for t=1:1000            % simulation of 1000 ms
  I=[5*randn(Ne,1);2*randn(Ni,1)]; % thalamic input
  fired=find(v>=30);    % indices of spikes
  firings=[firings; t+0*fired,fired];
  v(fired)=c(fired);
  u(fired)=u(fired)+d(fired);
  I=I+sum(S(:,fired),2);
  v=v+0.5*(0.04*v.^2+5*v+140-u+I); % step 0.5 ms
  v=v+0.5*(0.04*v.^2+5*v+140-u+I); % for numerical
  u=u+a.*(b.*v-u);                 % stability
end;
plot(firings(:,1),firings(:,2),'.');
```

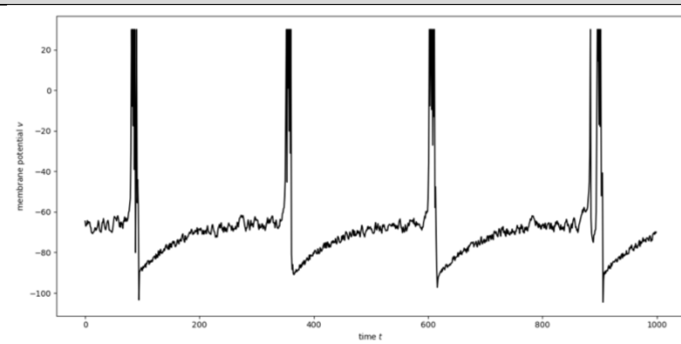
Raster plot of spikes (Definition: v>=30)



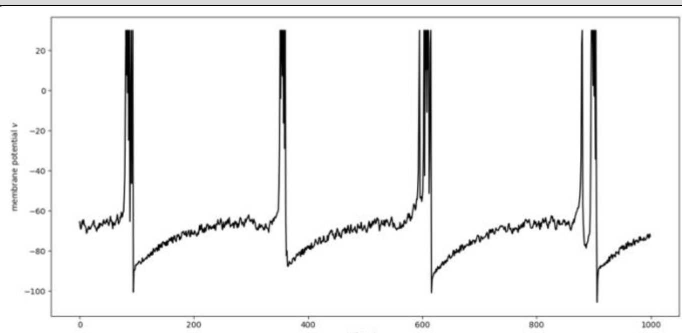
Node 0



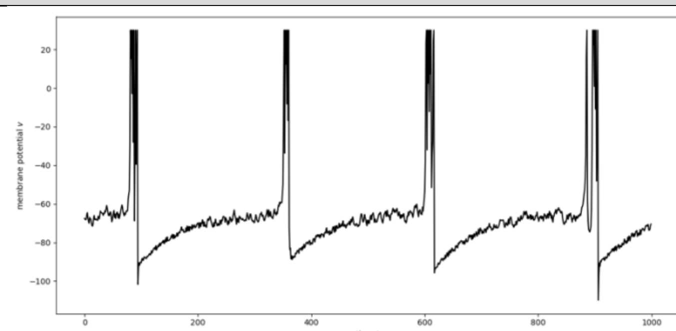
Node 1



Node 2



Node 3



5. Conductance-based spiking neuron model

Tunable params: **a,b,c,d,beta,tau_exc,tau_inh,V_E,V_I**

$$\dot{v}_i = 0.04\, v_i^2 + 5 v_i + 140 - u_i + I_i + \underbrace{\xi_i}_{\text{noise}}$$

$$\dot{u}_i = a(bv - u)$$

$a = 0.02$

$b = 0.2$

$c = -65$

$d = 8$

if $v_i \geq 30$

$v_i \to c$

$u_i \to u_i + d$

$$I_i(t) = G_i^{Exc}(t)\left(V_E - V_i(t)\right) - G_i^{Inh}(t)\left(V_i(t) - V_I\right)$$

$$G_i^{exc} = \beta \sum_{j=1,\, g_{ij}>0} g_{ij} \sum_k e^{-\frac{(t - t_{j,k})}{\tau_{exc}}}\, \theta(t - t_{j,k})$$

Heaviside step function

$$G_i^{inh} = \beta \sum_{j=1,\, g_{ij}>0} |g_{ij}| \sum_k e^{-\frac{(t - t_{j,k})}{\tau_{inh}}}\, \theta(t - t_{j,k})$$

(Preliminary Test) Model 1 [Success]

Ne,Ni

Ne=80, Ni=20

a,b,c,d

```
re=rand(Ne,1);            ri=rand(Ni,1);
a=[0.02*ones(Ne,1);       0.02+0.08*ri];
b=[0.2*ones(Ne,1);        0.25-0.05*ri];
c=[-65+15*re.^2;          -65*ones(Ni,1)];
d=[8-6*re.^2;             2*ones(Ni,1)];
```

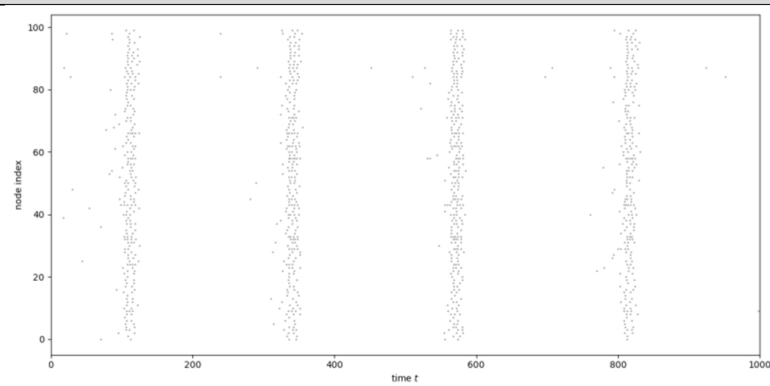
Coupling matrix S

```
S=[0.5*rand(Ne+Ni,Ne),  -rand(Ne+Ni,Ni)];
```

beta,tau_exc,tau_inh,V_E,V_I

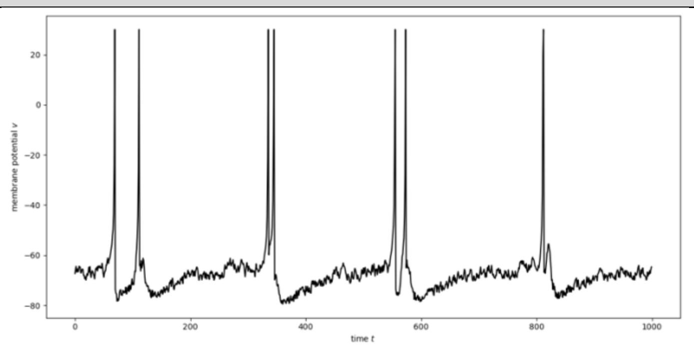
beta=0.02 (so that current I_i has the order of 1-10)
tau_exc=5, tau_inh=6
V_E=0, V_I=-80

Raster Plot

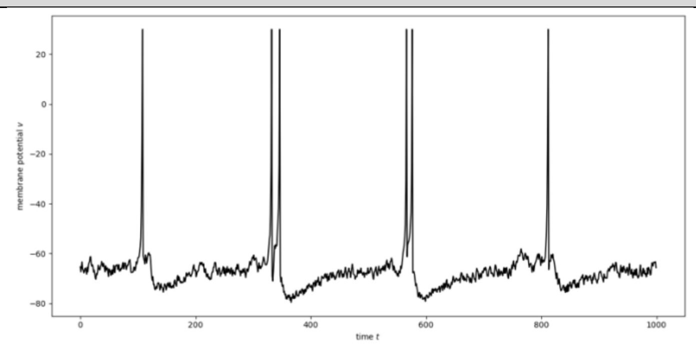


Time Series

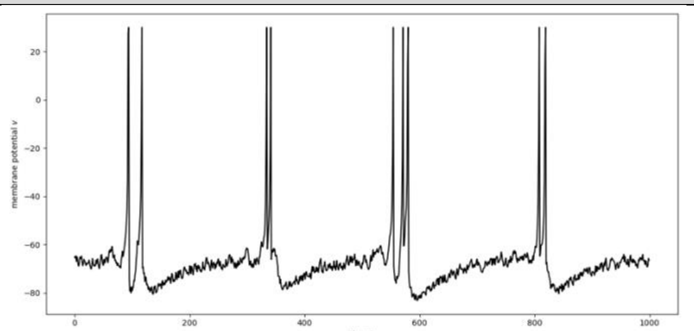
Node 0



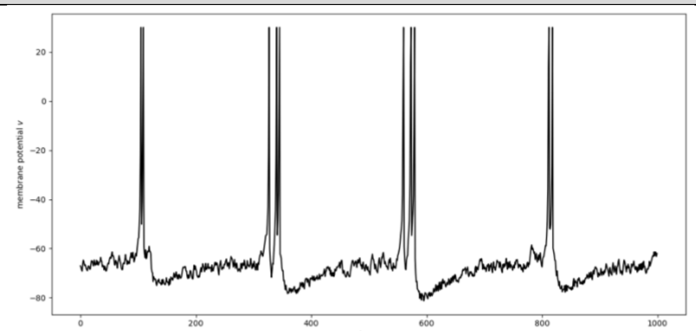
Node 1



Node 2



Node 3



(Preliminary Test) Model 2 [Failed]

Ne,Ni

Ne=400, Ni=100

a,b,c,d

```
re=rand(Ne,1);            ri=rand(Ni,1);
a=[0.02*ones(Ne,1);       0.02+0.08*ri];
b=[0.2*ones(Ne,1);        0.25-0.05*ri];
c=[-65+15*re.^2;          -65*ones(Ni,1)];
d=[8-6*re.^2;             2*ones(Ni,1)];
```

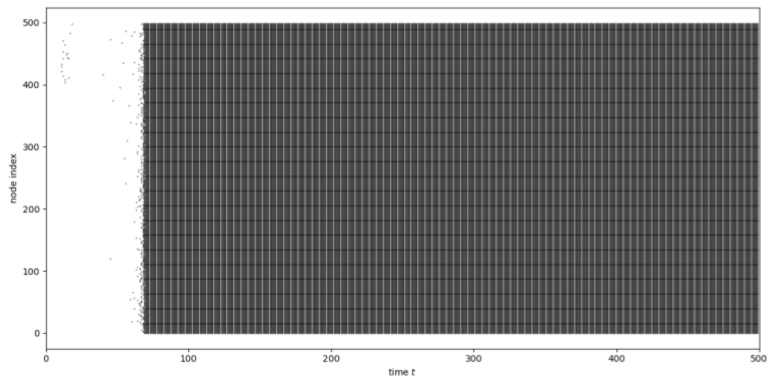
Coupling matrix S [maybe too dense]

```
S=[0.5*rand(Ne+Ni,Ne), -rand(Ne+Ni,Ni)];
```

beta,tau_exc,tau_inh,V_E,V_I

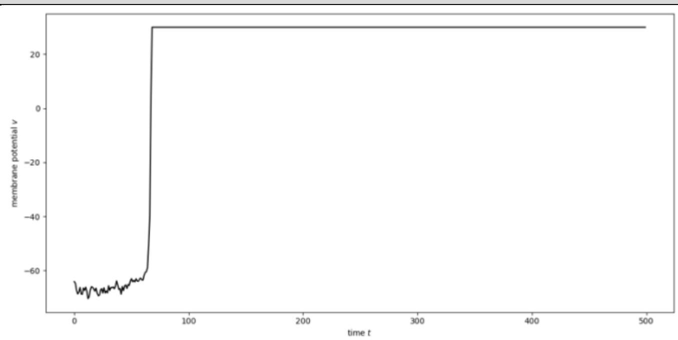
beta=0.02 (so that current I_i has the order of 1-10)
tau_exc=5, tau_inh=6
V_E=0, V_I=-80

Raster Plot

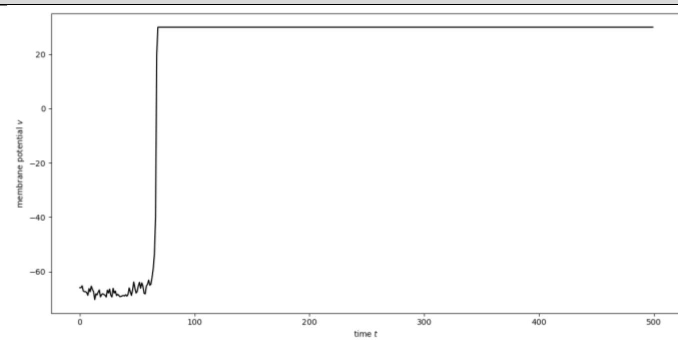


Time Series

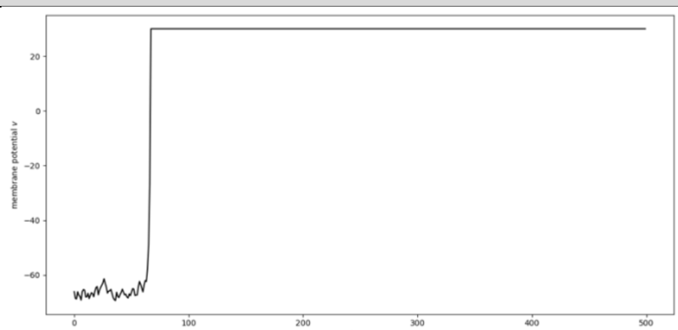
Node 0



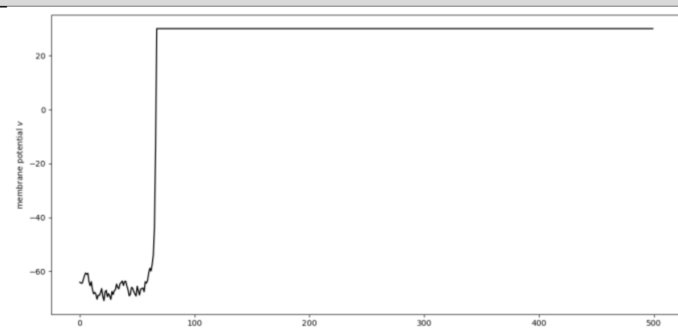
Node 1



Node 2



Node 3



(Preliminary Test) Model 3 [Success]

Ne,Ni

Ne=400, Ni=100

a,b,c,d

```
re=rand(Ne,1);          ri=rand(Ni,1);
a=[0.02*ones(Ne,1);     0.02+0.08*ri];
b=[0.2*ones(Ne,1);      0.25-0.05*ri];
c=[-65+15*re.^2;        -65*ones(Ni,1)];
d=[8-6*re.^2;           2*ones(Ni,1)];
```

Coupling matrix S

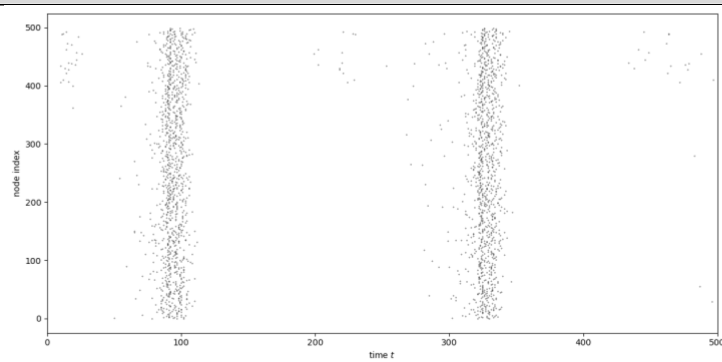
```
S=[0.5*rand(Ne+Ni,Ne),  -rand(Ne+Ni,Ni)];
```

Then filtered using connection probability 0.2

beta,tau_exc,tau_inh,V_E,V_I

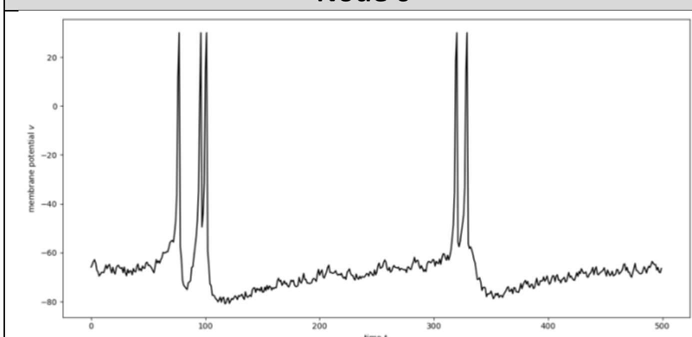
beta=0.02 (so that current I_i has the order of 1-10)
tau_exc=5, tau_inh=6
V_E=0, V_I=-80

Raster Plot

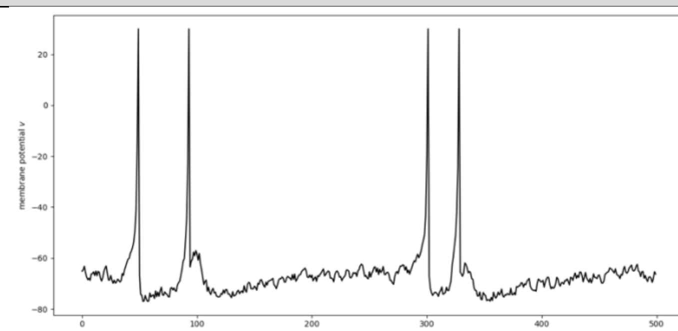


Time Series

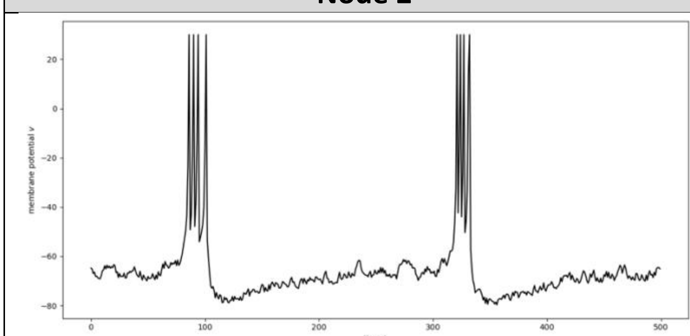
Node 0



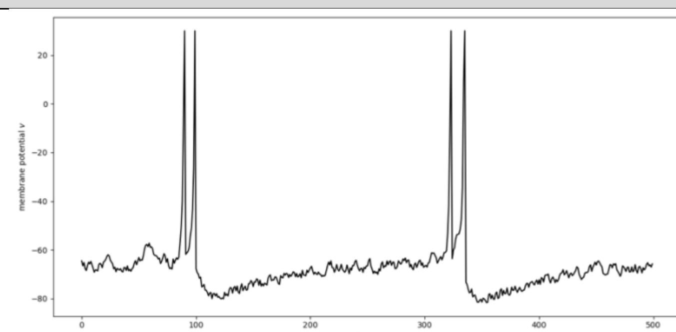
Node 1



Node 2



Node 3



DIV25 [Success]

Ne,Ni

Excitatory nodes have out-strengths >= 0; Ne=3425
Inhibitory nodes have out-strengths < 0; Ni=599

a,b,c,d

Excitatory ----------------------- Inhibitory

```
re=rand(Ne,1);           ri=rand(Ni,1);
a=[0.02*ones(Ne,1);      0.02+0.08*ri];
b=[0.2*ones(Ne,1);       0.25-0.05*ri];
c=[-65+15*re.^2;         -65*ones(Ni,1)];
d=[8-6*re.^2;            2*ones(Ni,1)];
```

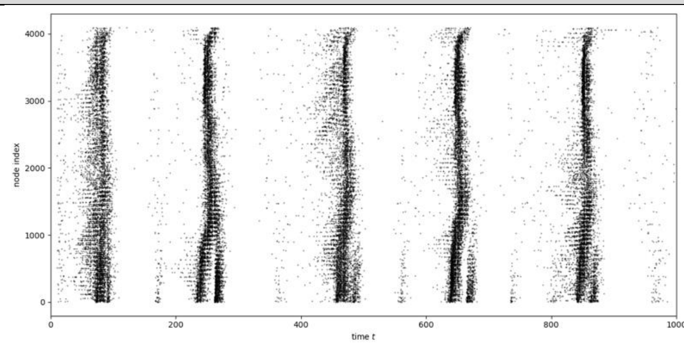
Coupling matrix S

DIV25

beta,tau_exc,tau_inh,V_E,V_I

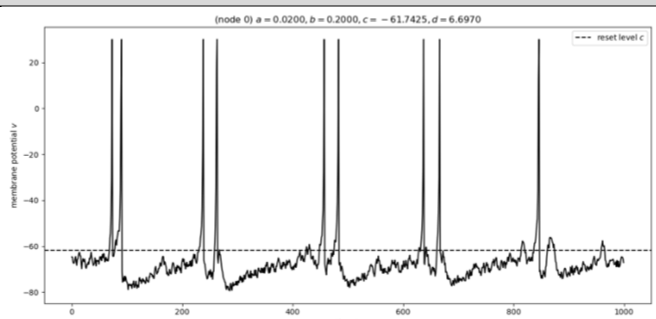
beta=1
tau_exc=5, tau_inh=6
V_E=0, V_I=-80

Raster Plot

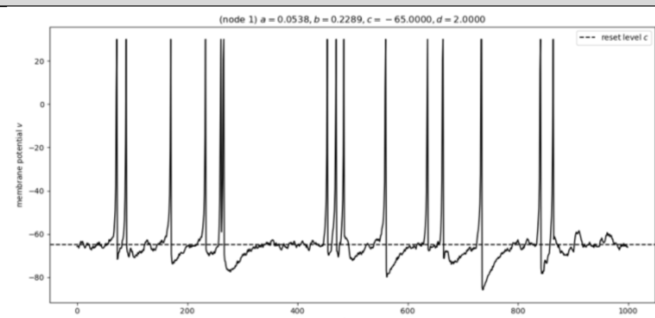


Time Series

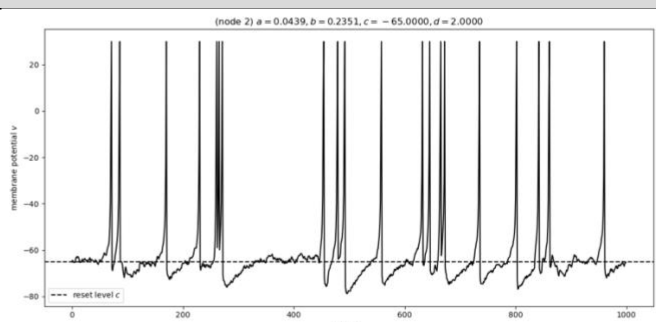
Node 0



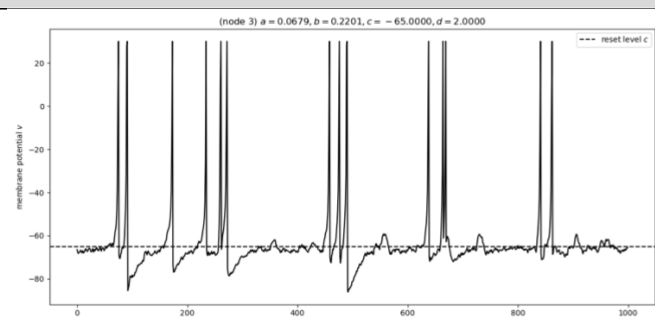
Node 1



Node 2

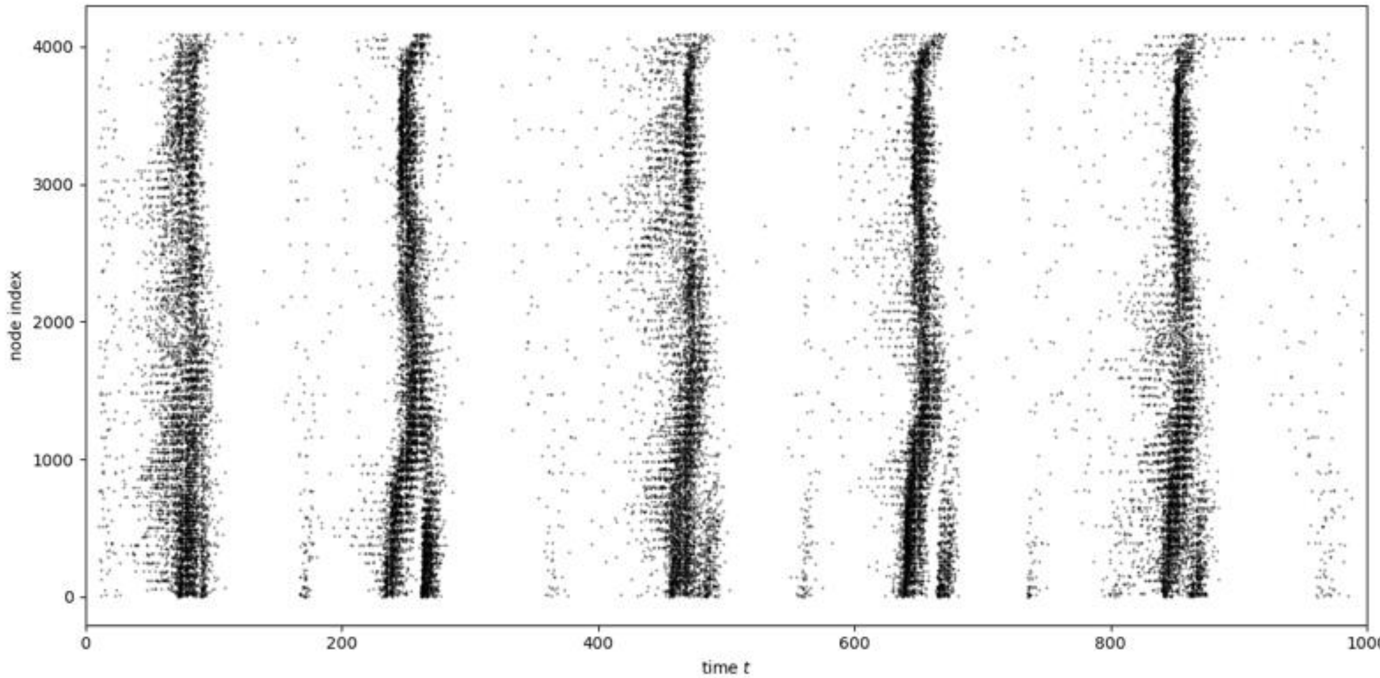


Node 3

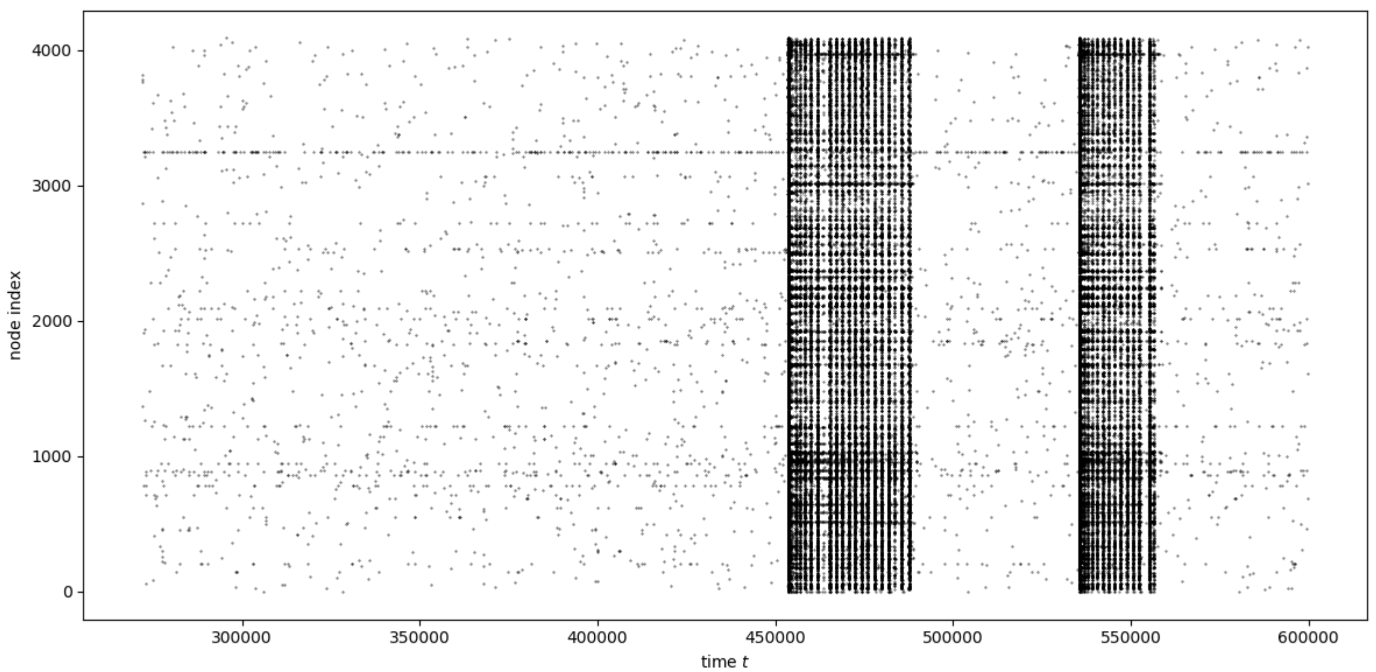


Raster Plot Comparison

Conductance-based Spiking Neuron Model

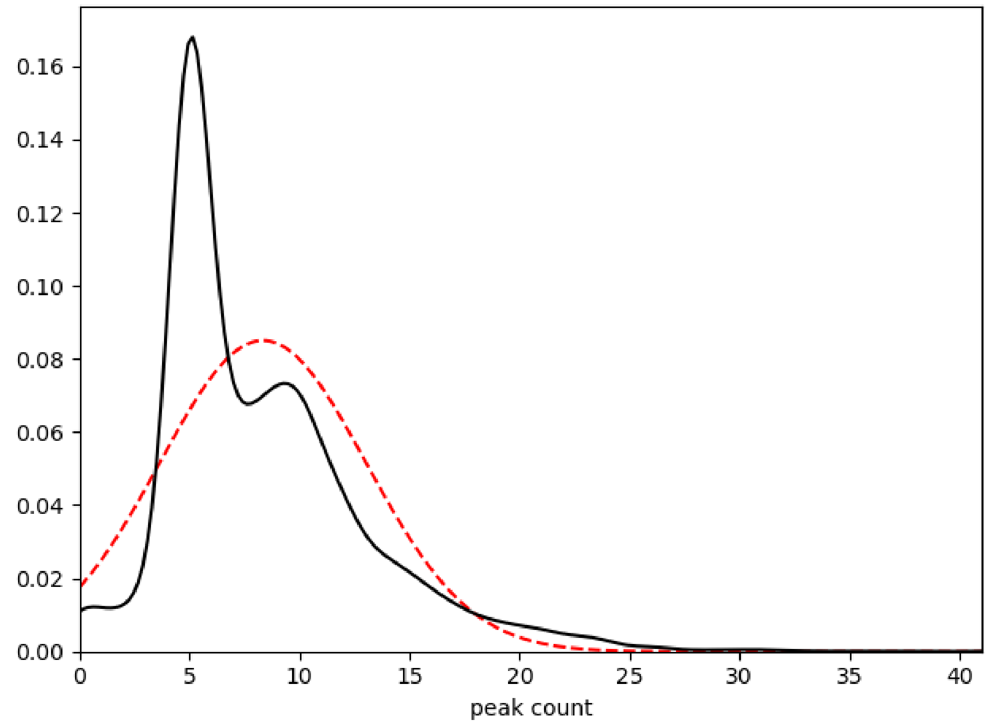


Experimental

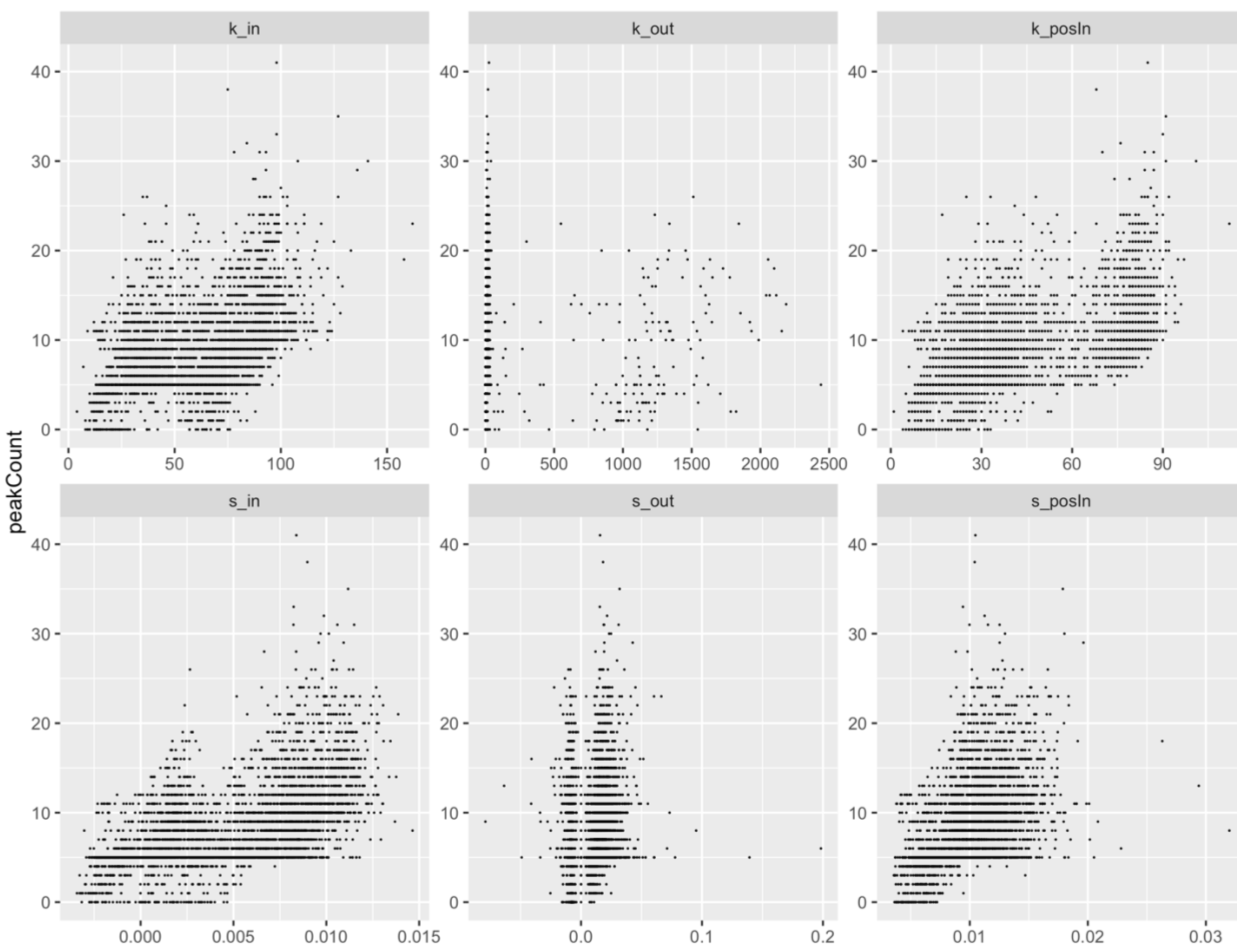


1000 time steps

Spike Count Distribution

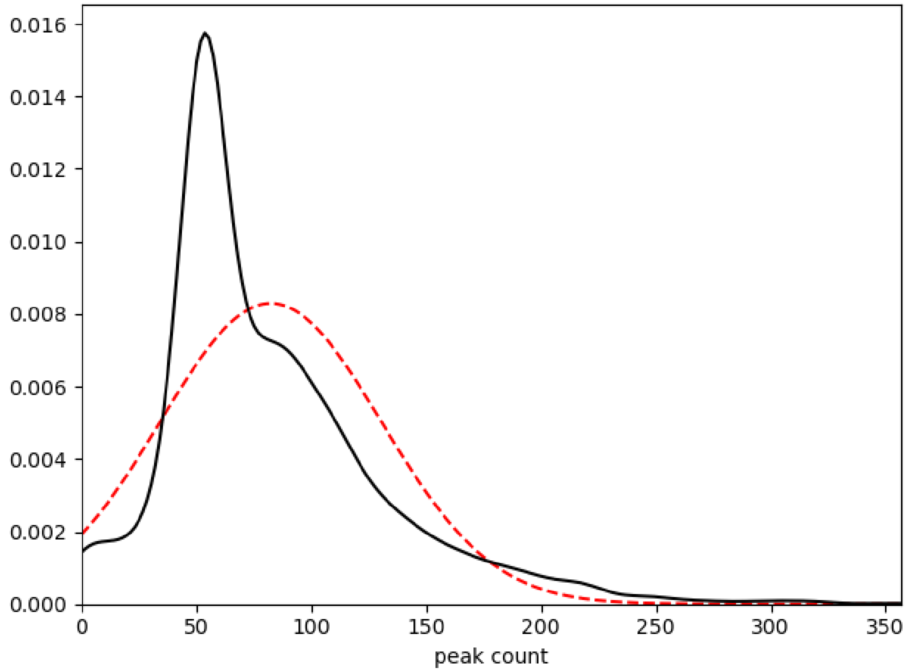


Spike Count vs Network Features

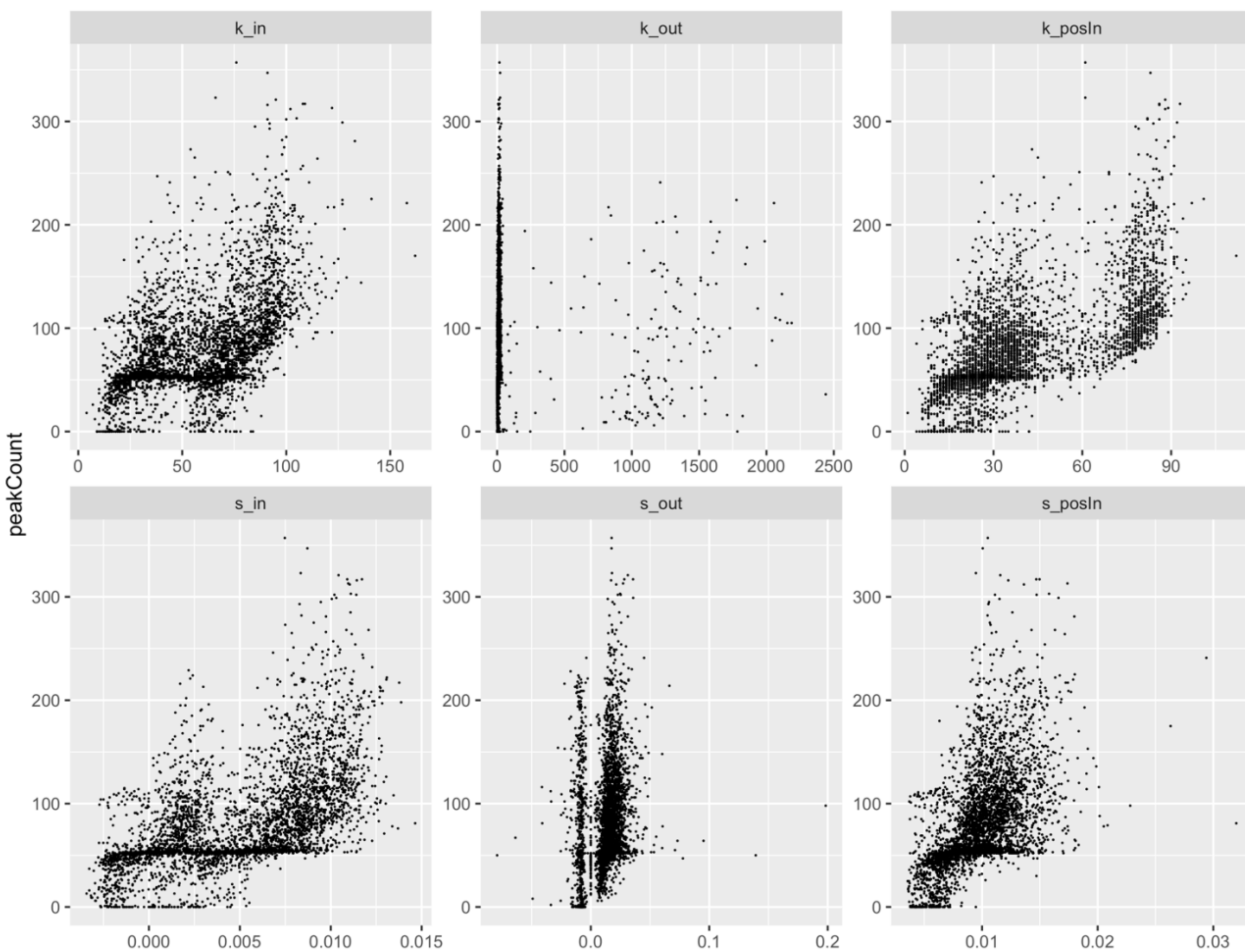


10000 time steps

Spike Count Distribution



Spike Count vs Network Features



6. Shuffled networks

DIV25 Shuffled Networks (10000 time steps)
Ne,Ni
Excitatory nodes have out-strengths >= 0 Inhibitory nodes have out-strengths < 0
a,b,c,d

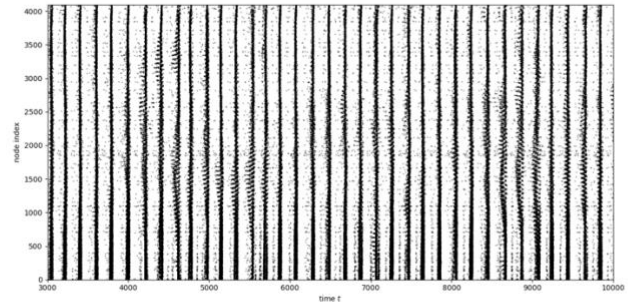
Excitatory ----------------------- Inhibitory
`re=rand(Ne,1);          ri=rand(Ni,1);`
`a=[0.02*ones(Ne,1);      0.02+0.08*ri];`
`b=[0.2*ones(Ne,1);       0.25-0.05*ri];`
`c=[-65+15*re.^2;         -65*ones(Ni,1)];`
`d=[8-6*re.^2;            2*ones(Ni,1)];`

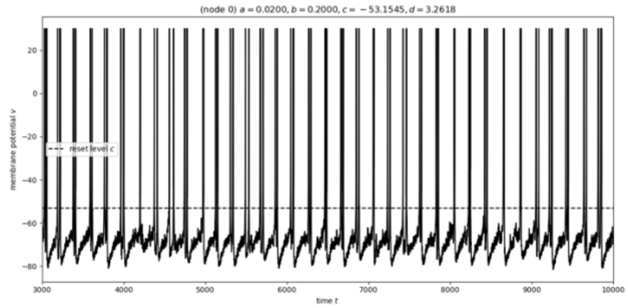
Coupling matrix S
DIV25 couplings under shuffling / replacement **(Ref 1) Non-zero gij replaced with N(μ,σ) [μ=mean(gij!=0), σ=sd(gij!=0)]** **(Ref 2) Rows shuffled** **(Ref 3) Cols shuffled** **(Ref 4) Random network with N(μ,σ) couplings** **(Ref 5) Entries shuffled**
beta,tau_exc,tau_inh,V_E,V_I
beta=1 tau_exc=5, tau_inh=6 V_E=0, V_I=-80

Raster Plot Comparison (10000 time steps)

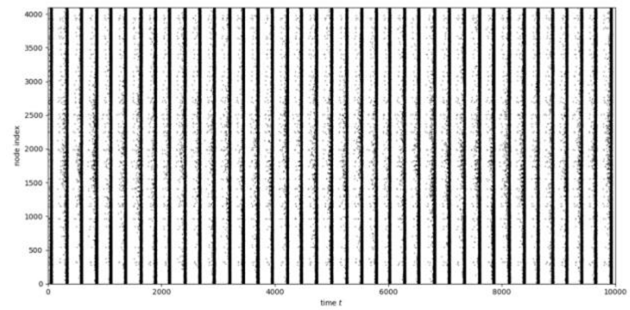
Raster Plot	Node 0 Time Series

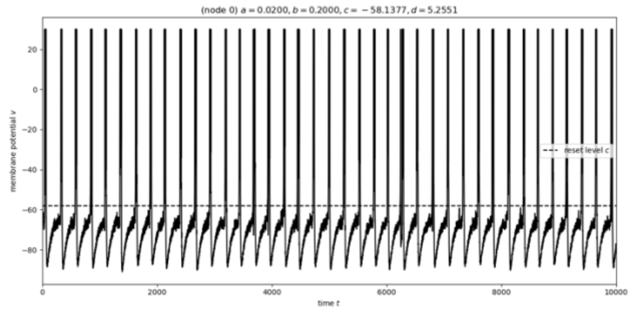
Original (DIV25)



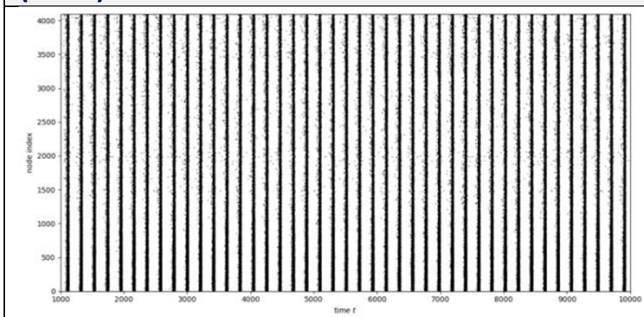


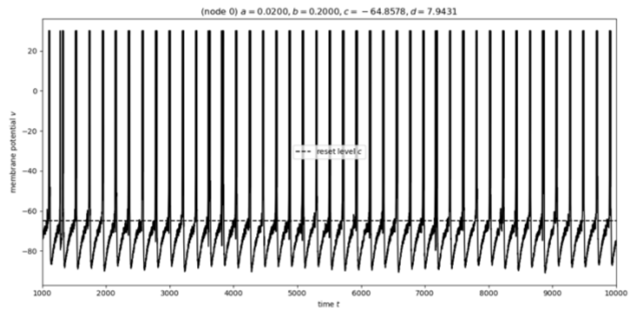
(Ref 1) Non-zero gij replaced with N(μ,σ)



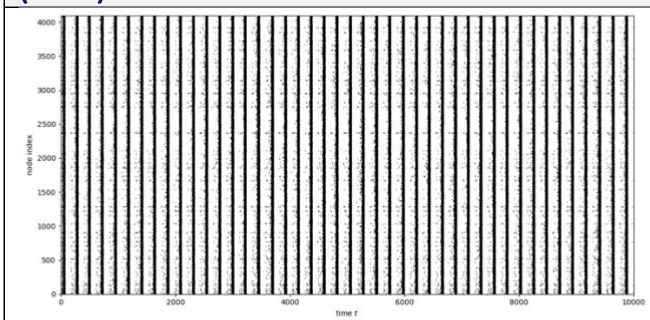


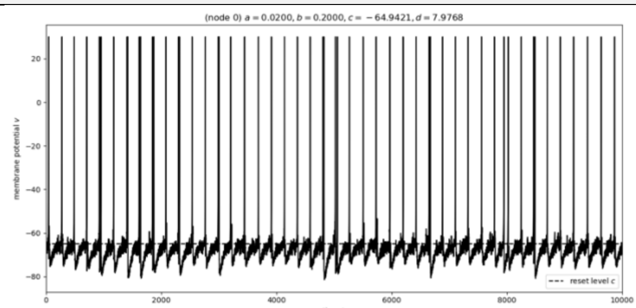
(Ref 2) Rows shuffled



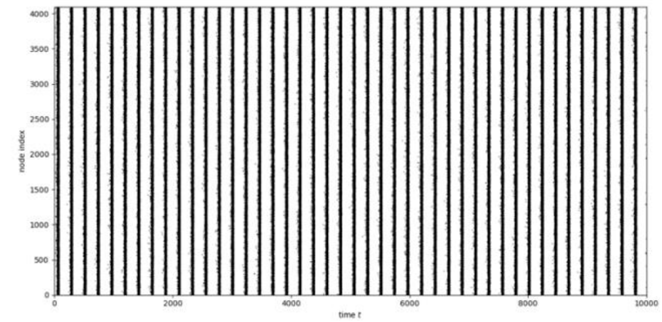


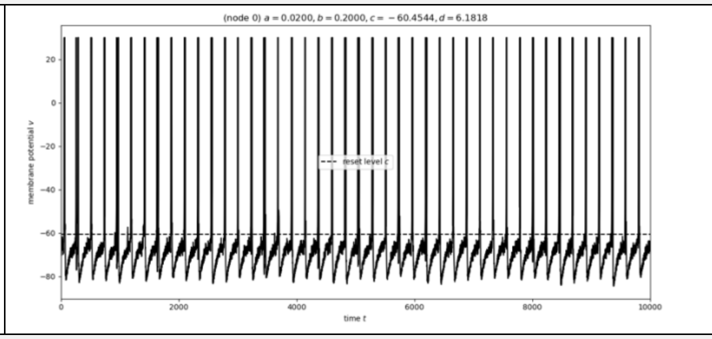
(Ref 3) Cols shuffled



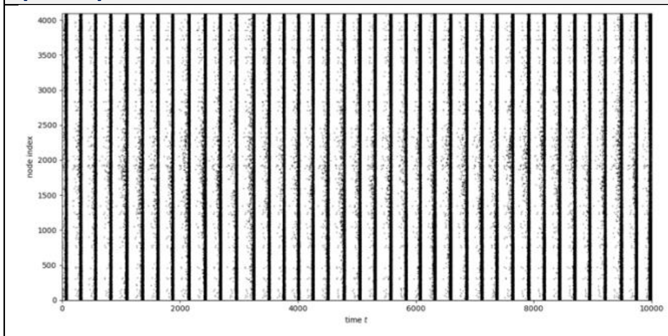


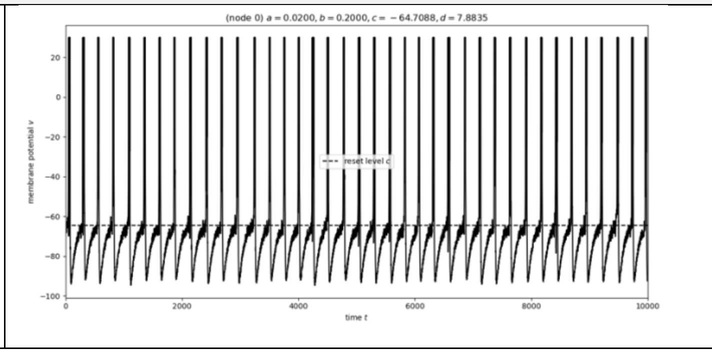
(Ref 4) Random network with N(μ,σ) couplings





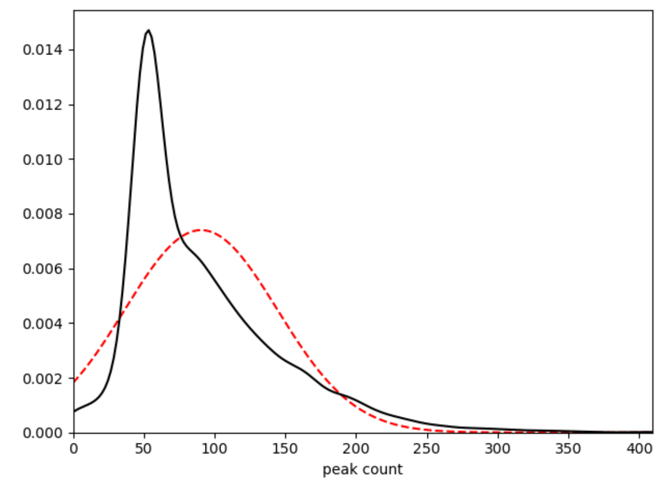
(Ref 5) Entries shuffled



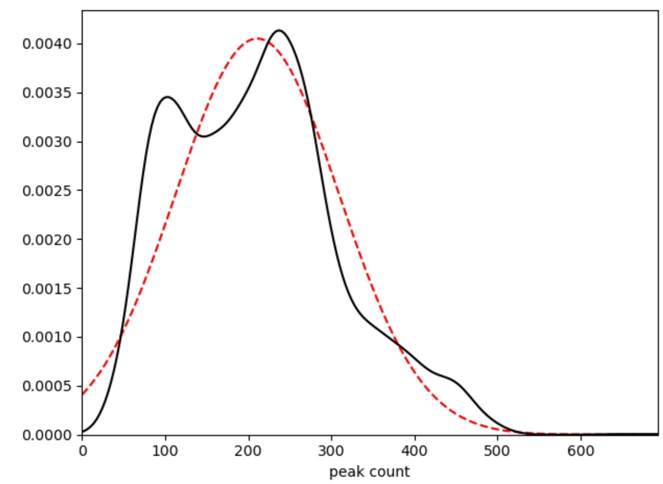


Spike Count Distribution (10000 time steps)

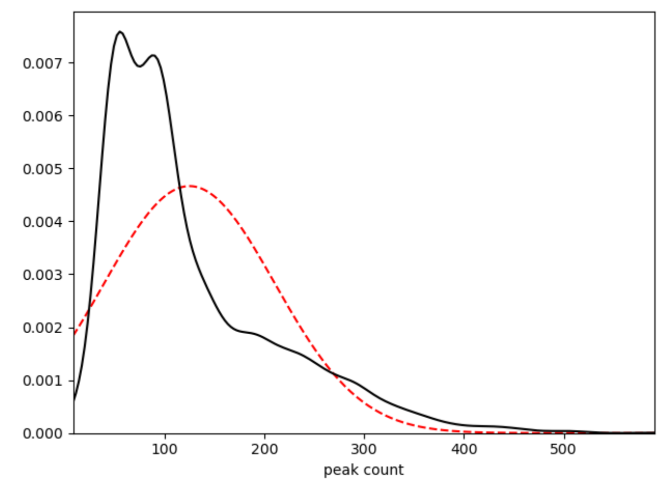
Original (DIV25)



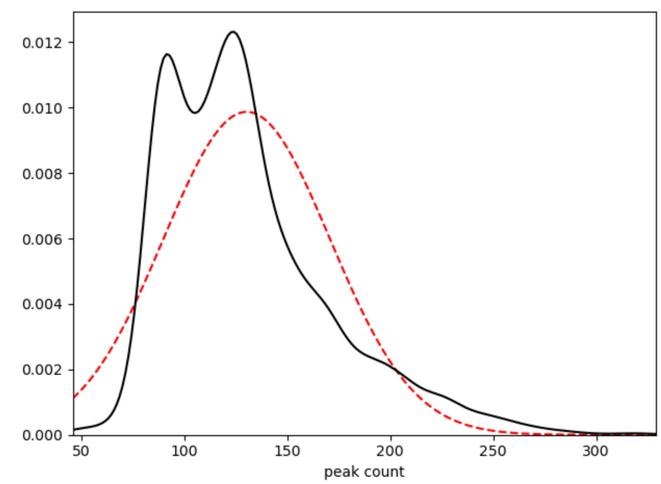
(Ref 1) Non-zero gij replaced with N(μ,σ)



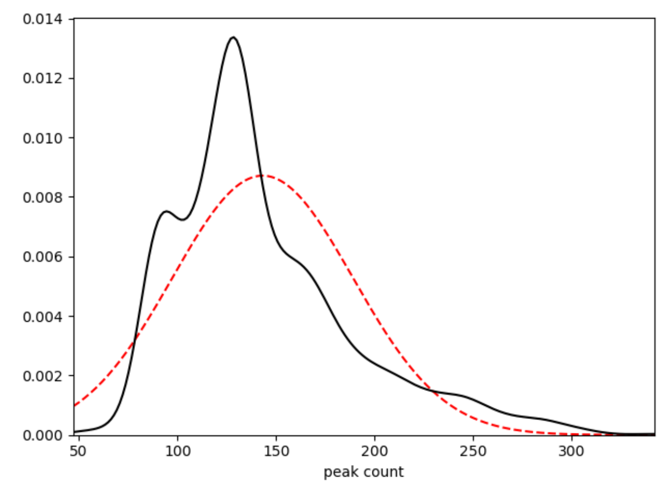
(Ref 2) Rows shuffled



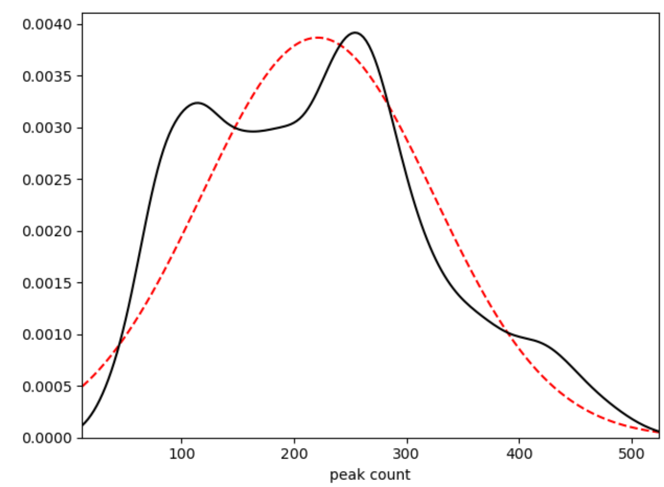
(Ref 3) Cols shuffled



(Ref 4) Random network with N(μ,σ) couplings

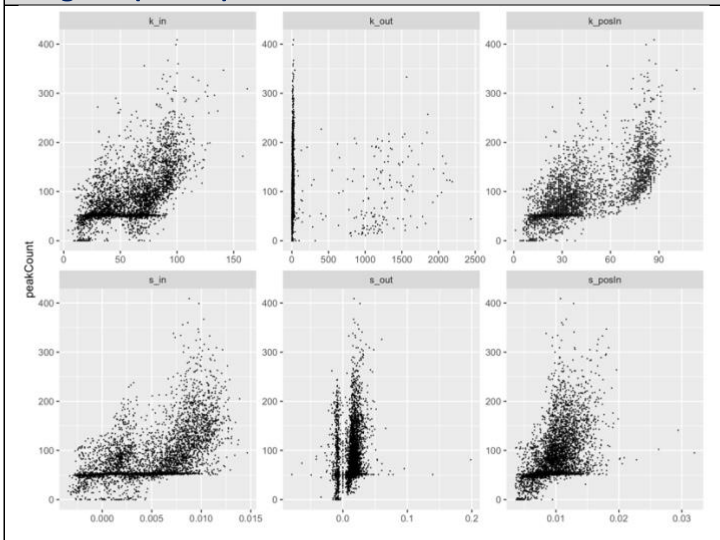


(Ref 5) Entries shuffled

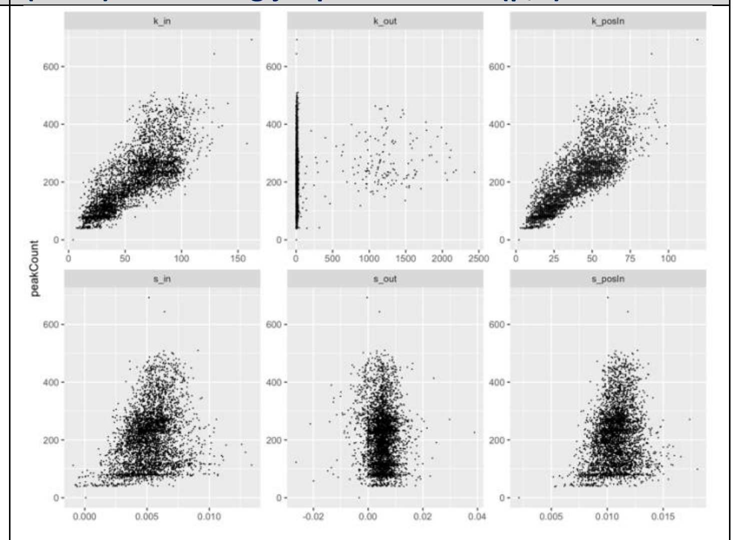


Spike Count vs Network Features (10000 time steps)

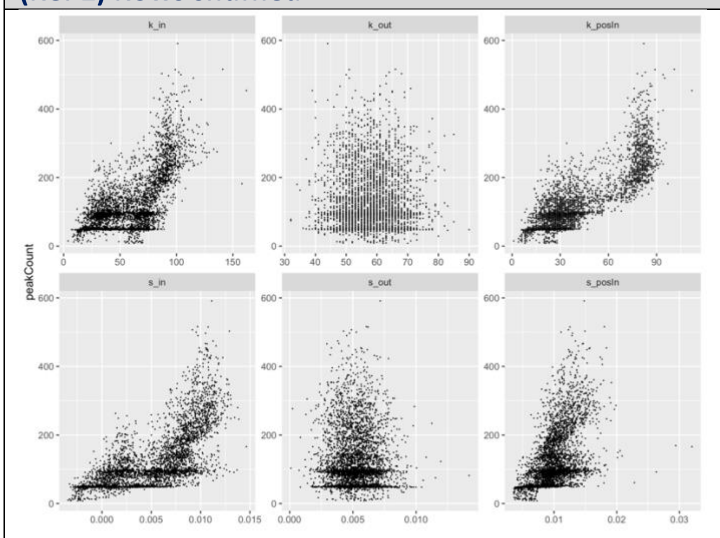
Original (DIV25)



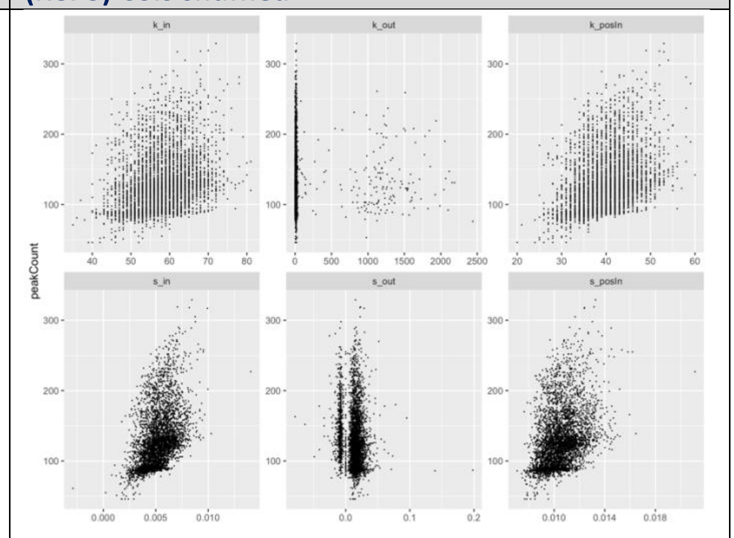
(Ref 1) Non-zero gij replaced with N(μ,σ)



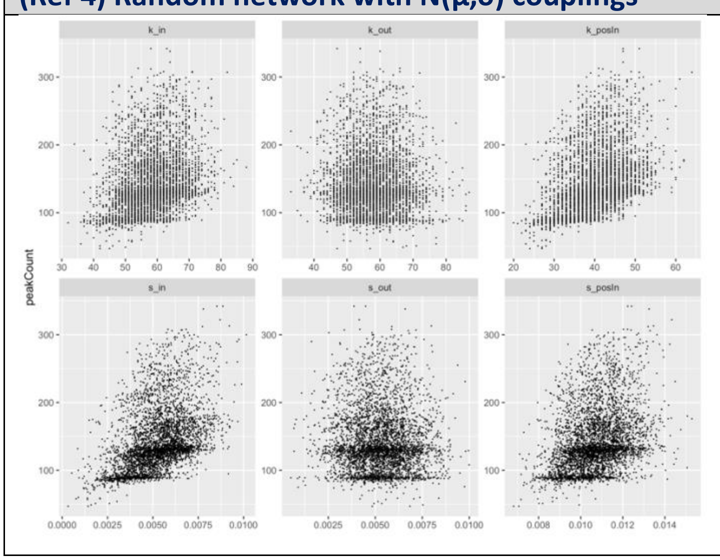
(Ref 2) Rows shuffled



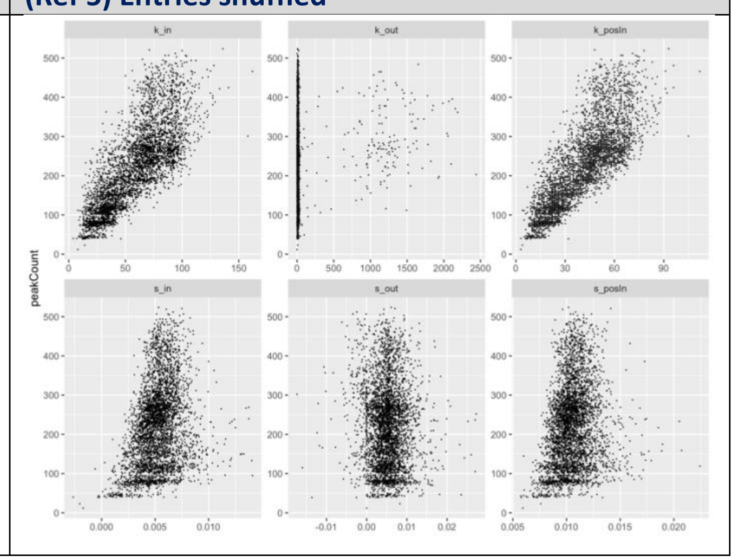
(Ref 3) Cols shuffled



(Ref 4) Random network with N(μ,σ) couplings



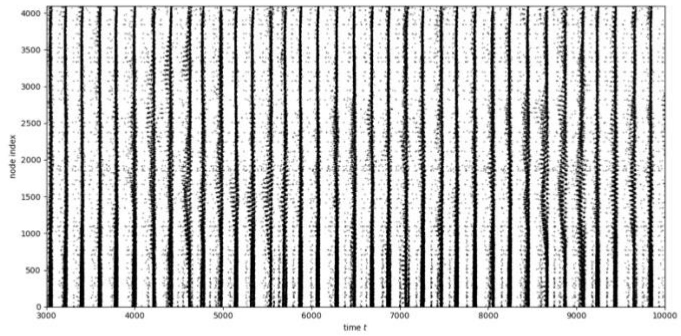
(Ref 5) Entries shuffled



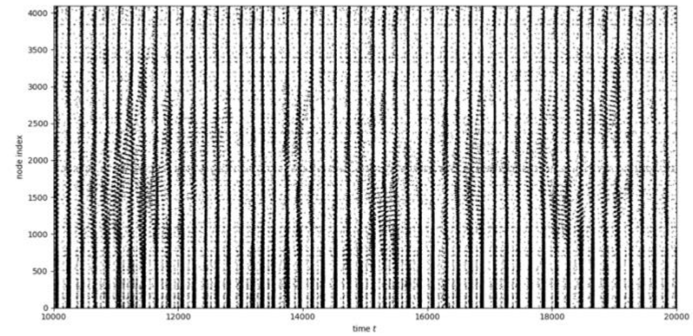
DIV25 Original Network (10000 vs 20000 time steps)

Raster Plot

0 to 10000 time steps

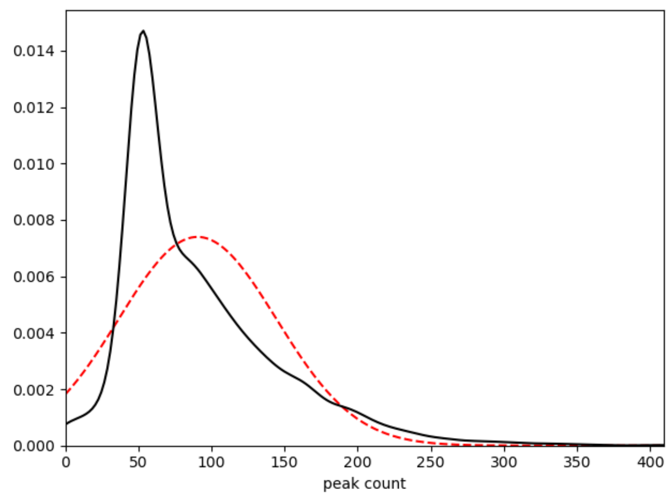


10000 to 20000 time steps

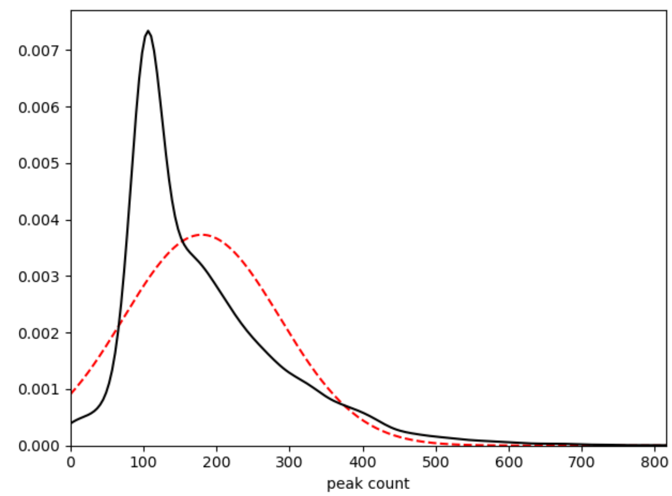


Spike Count Distribution

10000 time steps

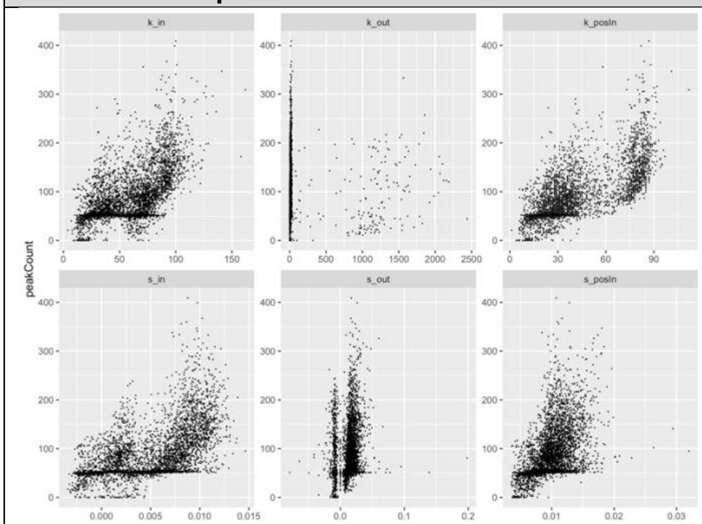


20000 time steps

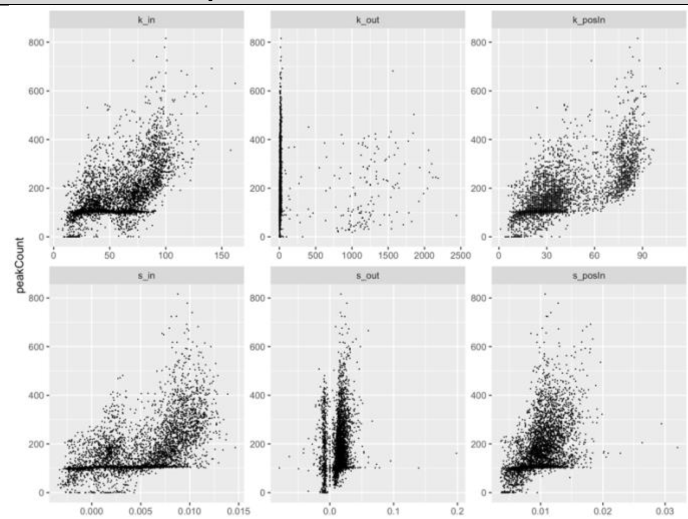


Spike Count vs Network Features

10000 time steps



20000 time steps



7. Analysis of spike dynamics

Spike count statistics (10000 time steps)

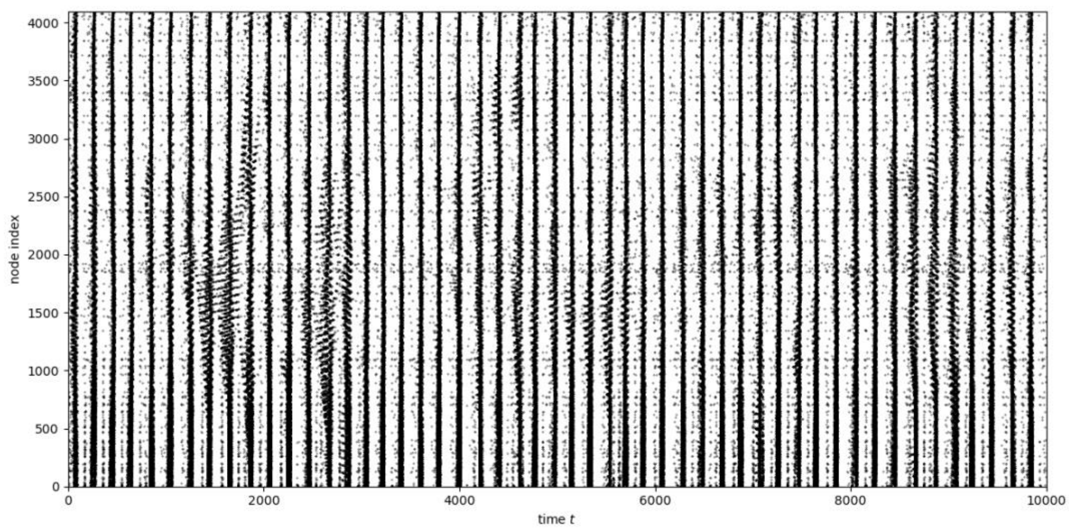
Network	Min	Max	Median	Mean	Skew	Kurtosis	S(95%)
Original	0	409	74	90.46	1.43	**2.66**	**1.01**
Ref 1	0	693	207	210.76	0.53	-0.05	0.32
Ref 2	8	591	95	124.77	1.42	**1.85**	**1.03**
Ref 3	46	329	123	130.42	1.16	1.45	0.82
Ref 4	47	342	132	143.22	1.13	1.24	0.89
Ref 5	12	524	222	221.05	0.41	-0.44	0.15

S(q) = (Mean(X|X>q-quantile)-Median(X))/Std(X) – Ref(q)

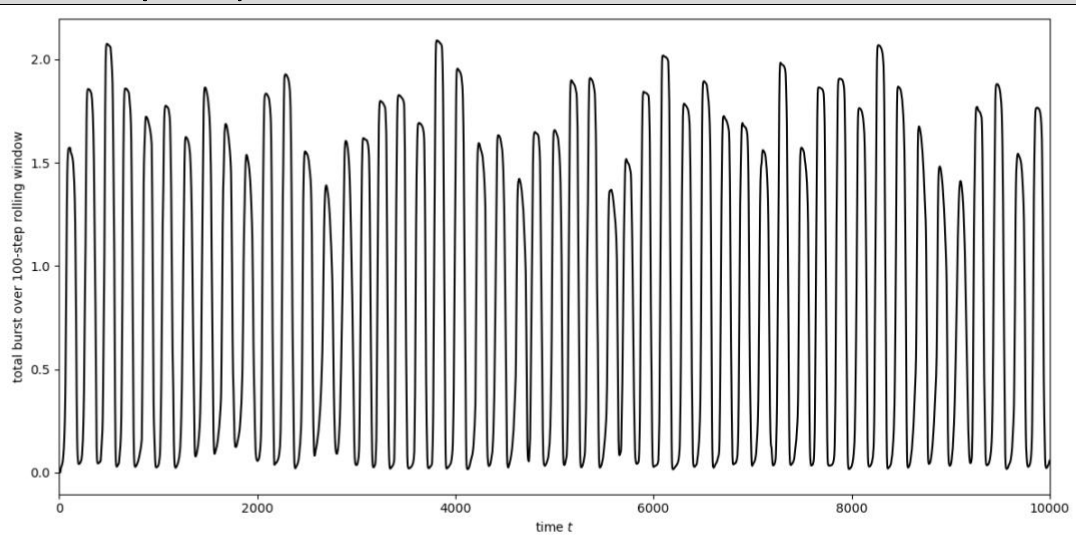
Ref(q) = Mean(N(0,1)|N(0,1)>q-quantile)

Defining spike burst (Original network – 10000 steps)

Raster Plot



Burst Plot (w=100)



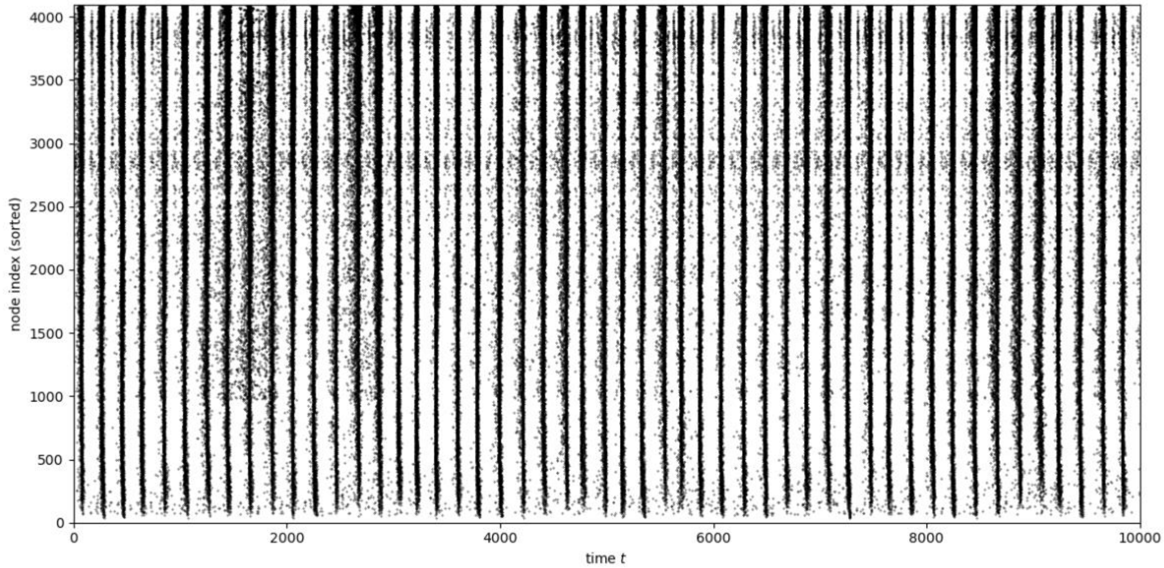
Burst(t) = Sum(Spike[min(t-w,0):t])/(#nodes)

Choice of w: < spike time scale ~ 200 time steps

Identify the peaks to locate the simultaneous spiking activity

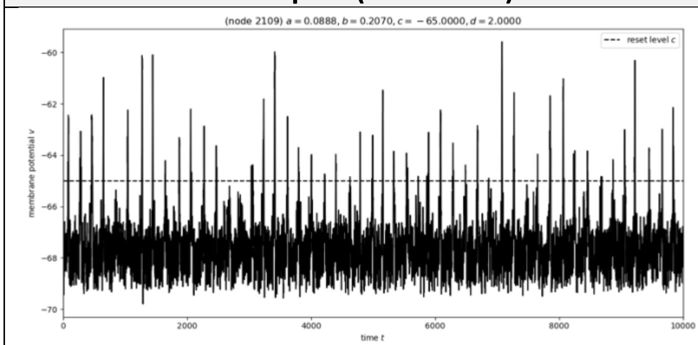
High-spike, moderate-spike & low-spike behavior (Original network – 10000 steps)

Raster Plot (sorted according to spike counts)

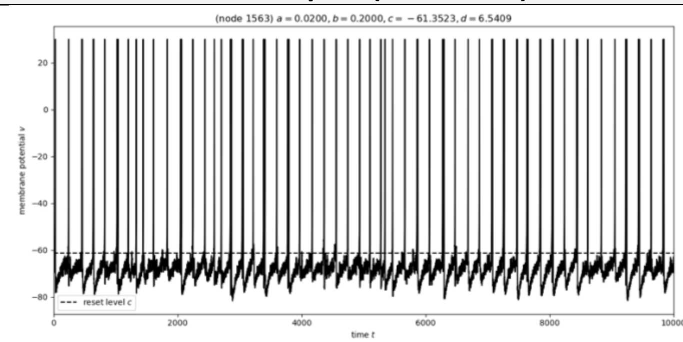


Time Series (low-, median-, high-spike nodes)

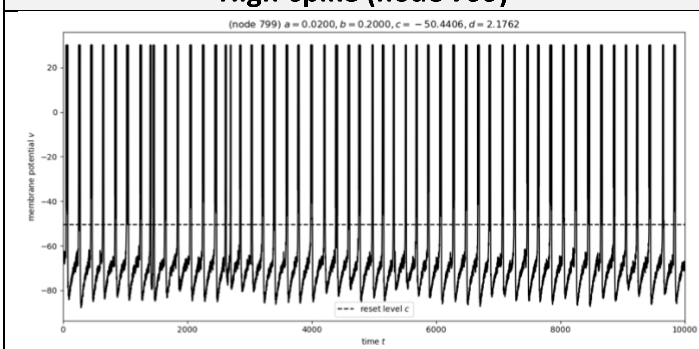
Low-spike (node 2109)



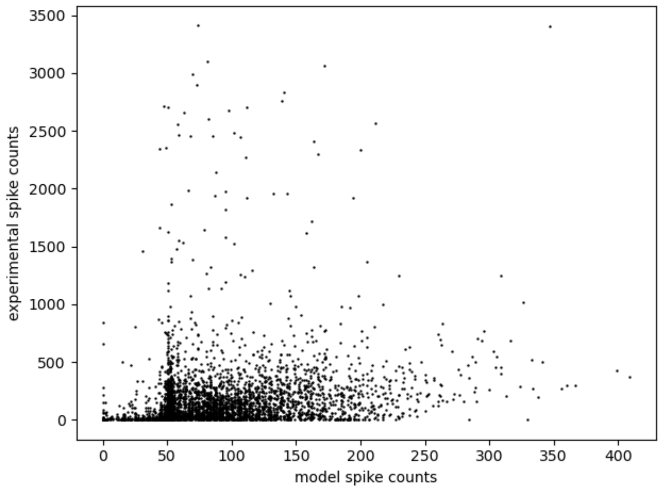
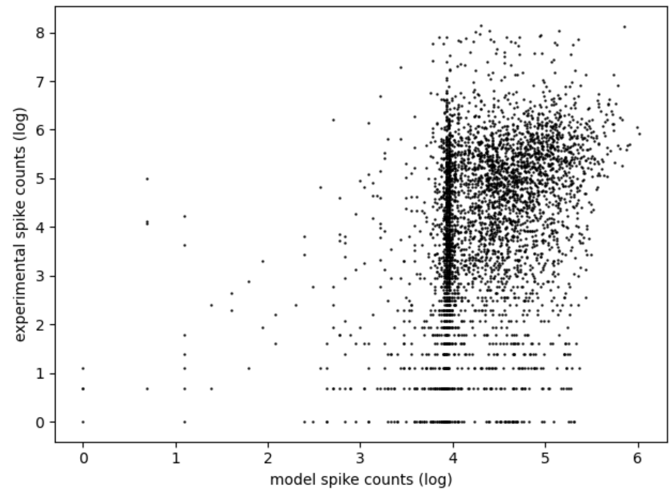
Median-spike (node 1563)



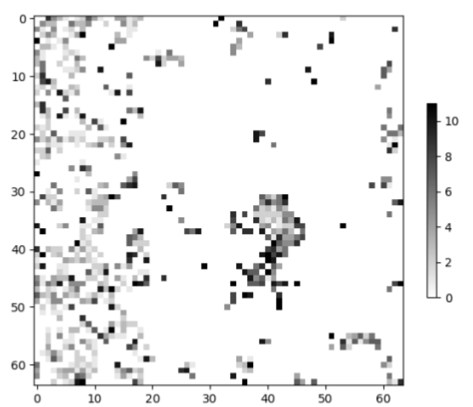
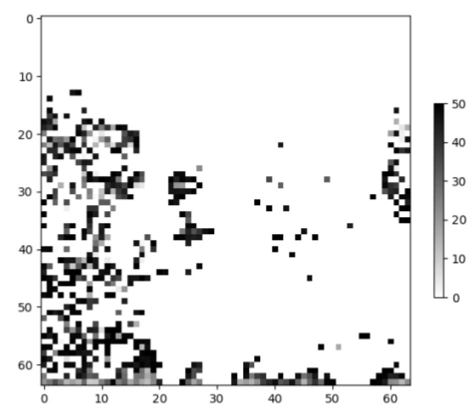
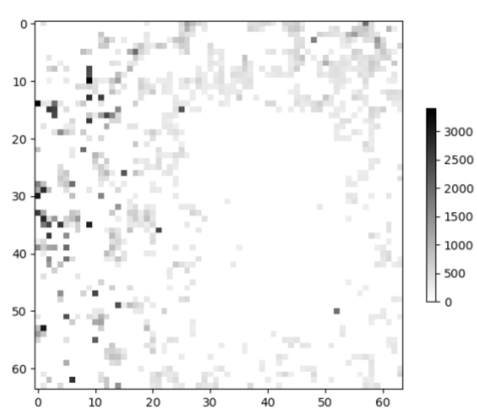
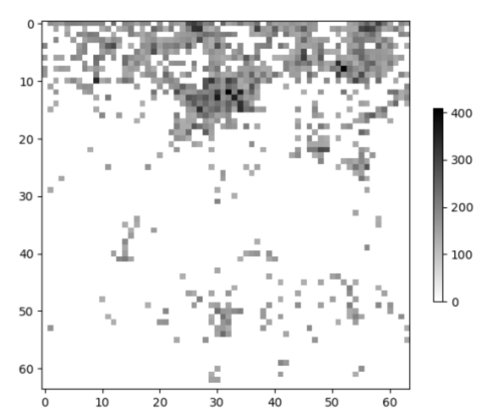
High-spike (node 799)



Experimental vs model spike counts (Original network – 10000 steps)

Linear scale	Log scale
	

Where do high-spike & low-spike nodes locate physically?

Experimental	Model (Original network – 10000 steps)
< 20-percentile	
	
> 80-percentile	
	

8. Summary of FYP Part II work

1. Network reconstruction with FNCCH and Covariance Relation (PRE method) [P2-21]

- **FNCCH applied to measured spike data**: (1) Thresholds for gij>0 & gij<0 and (2) Physical constraints, including min delay time & signal propagation speed, are applied. Majority of the reconstructed gij are positive while a small portion are negative.
- **FNCCH & Covariance Method applied to FHN model data**: Covariance Method gives ~45% sensitivity while FNCCH gives ~3% sensitivity.
- **Limitations on the accuracy of Covariance Method**: The line of (Mij,gij=0) points due to noise obscures the classification of Mij=0 and Mij!=0 group. Require longer time series s.t. non-zero gij give significantly non-zero Mij. The effect of noise in the FHN model caps the sensitivity of the reconstruction method.

2. Conductance-based spiking neuron model [P22-40]

- **Modification of EM Izhikevich's basic spiking neuron model to incorporate conductance**
- **DIV25 coupling strengths applied and dynamics are still stable through the choice of a,b,c,d params**: The model generates spontaneous & simultaneous spiking activity, and the spike count distribution exhibits a long tail with a wide range (e.g. for 2e6 time steps, range = [0,800]). Thus, the model shows a richer and more realistic dynamics than FHN model.
- **Relation btw. long-tailed sin/sout & long-tailed spike counts**: DIV25 gij have long-tailed sin/sout and when applied to spiking neuron model, spike counts are long-tailed -> Any association? Using shuffled networks, ref 2, which shuffles each column of coupling strength matrix G thus preserving in-measures (e.g. kin, sin), most closely replicates the dynamics of the original network, in terms of spike count distribution & correlation btw. spike counts and network features. Other shuffled networks "destroy" the structure of sin and are not able to recover a long-tailed spike count distribution. Hence, sin is identified to be an important factor of spike count. (This is also intuitively true)
- **Is the model output realistic**? (1) Measured spike counts & model spike counts are compared and there exist a (slight) correlation. (2) Physical locations on the measurement grid of the high/low-spike nodes roughly coincide. Note that in this stage, the model is uncalibrated and the choice of a,b,c,d params are conveniently chosen, based on the example params in Izhikevich's paper. Therefore, the accuracy of the model is still improvable.

9. Report outline

1. **Methods of Network Reconstruction**
   - [Theory] FNCCH
   - [Results] FNCCH applied to DIV25 measured spike data [explain physical constraints]
   - [Theory] Covariance method (PRE method)
   - [Theory] FHN model
   - [Results] Performance of FNCCH & Covariance Method (when applied to FHN model)
   - [Results] Limitations of the covariance method [explain how model noise hinders Mij classification]
2. **Spiking Neuron Model**
   - [Theory] EM Izhikevich's basic spiking neuron model
   - [Theory] Conductance-based model
   - [Results] Applying DIV25 gij and network dynamics [spike counts & correlation w/ net features]
   - [Results] Do model outputs resemble measured data?
3. **Network Features and Spike Dynamics**
   - [Theory] Shuffled networks
   - [Results] Methodology & key conclusions from FYP Part I
   - [Results] Network features & dynamics of the shuffled networks [distribution of features & spike counts]
   - [Results] What drive the long-tailed spike count distribution?