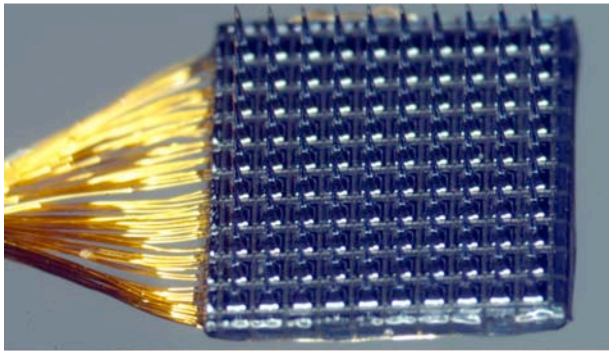
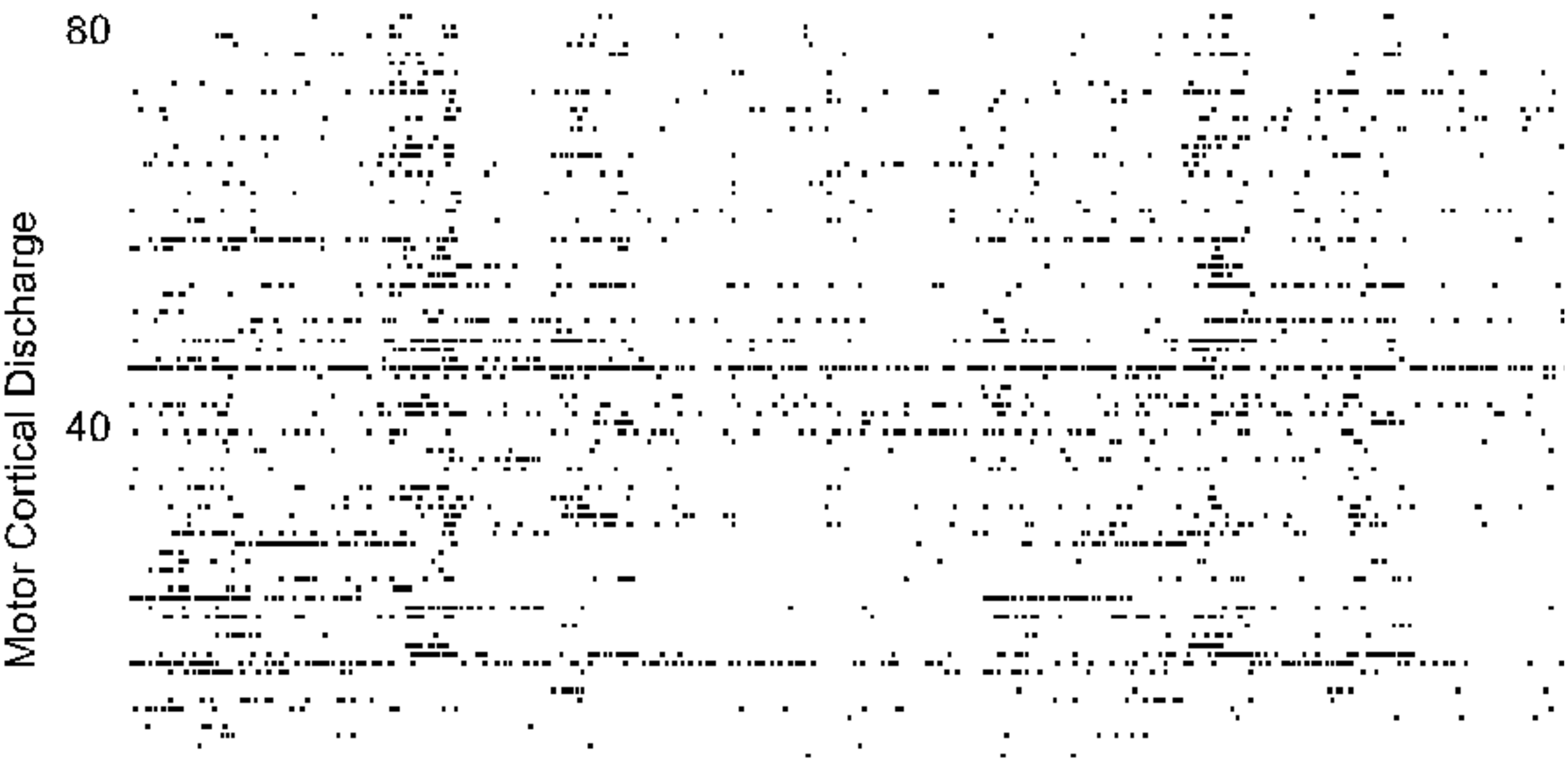
Effective connectivity reveals induced network reorganization in behaving animals

Sara A. Solla
Northwestern University
KITP, October 22, 2010

Inferring Functional Connectivity

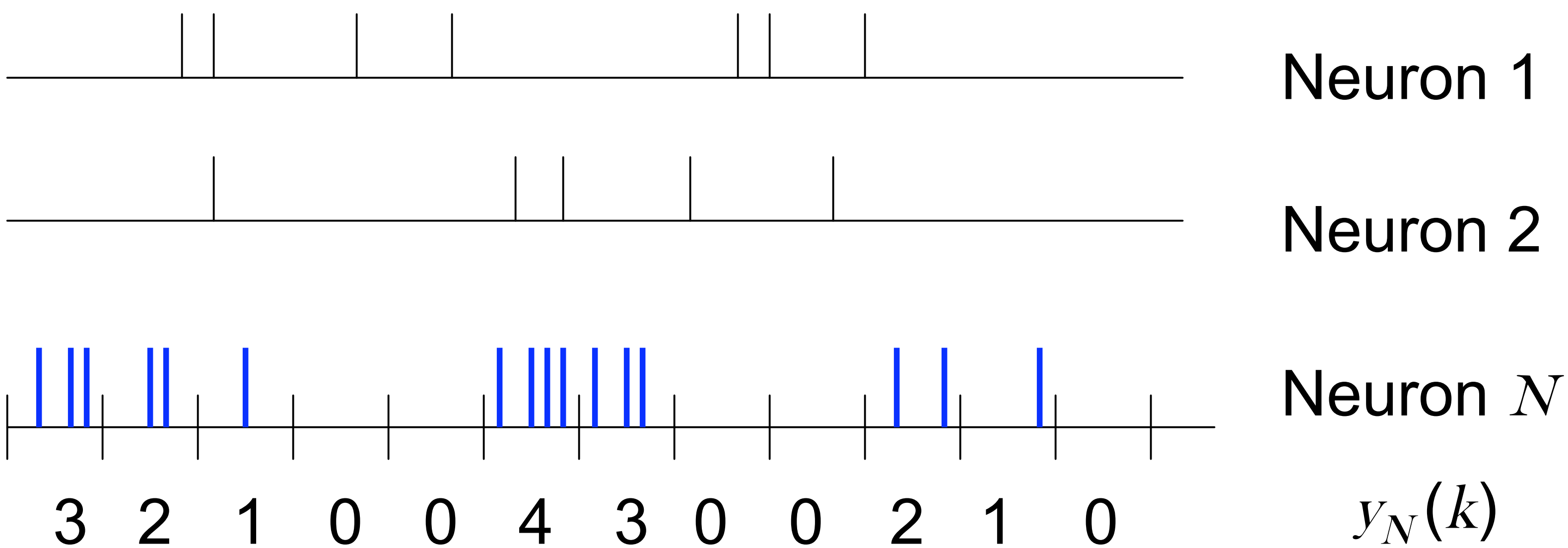


- High-density multi-electrode arrays:
- Record simultaneously one hundred neurons
- Characterize the dynamics of neural circuits.



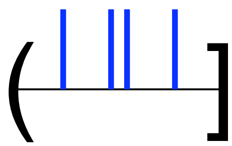
Analysis of neural activity

- Consider a population of N neurons whose spiking activity is observed during a time interval $(0,T]$.
- The interval is divided into K bins of size $\Delta = T/K$, labeled by an index $1 \leq k \leq K$.
- In each interval k we observe the number of spikes $y_i(k)$ emitted by neuron i, for all $1 \leq i \leq N$.



Neural activity as a Poisson process

$\left(t_{k-1}, t_k\right]$ $1 \le k \le K$, K bins of size $\Delta = T/K$



$y_i(k)$ = with probability $P(y_i(k))$

- Data: $\{y_i(k)\}$, for $1 \le k \le K$ and for $1 \le i \le N$.
- The spiking activity of neuron i at time interval k is modeled as a Poisson process with mean $\lambda_i(k)$.
- The probability of observing precisely $y_i(k)$ spikes emitted by neuron i at interval k is given by:

$$P(y_i(k) \mid \lambda_i(k)) = \frac{\left(\lambda_i(k)\right)^{y_i(k)} \; e^{-\lambda_i(k)}}{y_i(k)!}$$

Likelihood of spiking data

What is the probability of the observed data $\{y_i(k)\}$ given the parameters $\{\lambda_i(k)\}$?

$$P_T(\{y_i(k)\} \mid \{\lambda_i(k)\}) = \prod_{i=1}^{N} \prod_{k=1}^{K} \frac{(\lambda_i(k))^{y_i(k)} \; e^{-\lambda_i(k)}}{y_i(k)!}$$

The likelihood is defined as the logarithm of the probability:

$$L_T\left(\{y_i(k)\} \mid \{\lambda_i(k)\}\right) = \ln\left\{\prod_{i=1}^{N} \prod_{k=1}^{K} \frac{(\lambda_i(k))^{y_i(k)} \; e^{-\lambda_i(k)}}{y_i(k)!}\right\} =$$

$$= \left\{\sum_{i=1}^{N} \sum_{k=1}^{K} \left(y_i(k) \ln \lambda_i(k) - \lambda_i(k) - \ln\left(y_i(k)!\right)\right)\right\}$$

Maximum likelihood

Given the data $\{y_i(k)\}$, find the parameters $\{\lambda_i(k)\}$ that maximize L_T:

$$L_T\left(\left\{y_i(k)\right\}\right) \propto \left\{ \sum_{i=1}^{N} \sum_{k=1}^{K} \left(y_i(k) \ln \lambda_i(k) - \lambda_i(k) \right) \right\}$$

A term that does not depend on $\left\{\lambda_i(k)\right\}$ has been dropped.

MODEL: $$\ln \lambda_i\left(k\right) = \alpha_{i0} + \sum_{j=1}^{N} \sum_{m=1}^{\tau_N} \alpha_{ij}(m)\, y_j(k-m)$$

Given the data $\{y_i(k)\}$, find the parameters $\{\alpha\}$ that maximize L_T:

$$L_T\left(\left\{y_i(k)\right\} \mid \left\{\alpha\right\}\right) \propto \left\{ \sum_{i=1}^{N} \sum_{k=1}^{K} \left(y_i(k) \left[\alpha_{i0} + \sum_{j=1}^{N} \sum_{m=1}^{\tau_N} \alpha_{ij}(m)\, y_j(k-m) \right] - \right.\right.$$

$$\left.\left. - \exp\left[\alpha_{i0} + \sum_{j=1}^{N} \sum_{m=1}^{\tau_N} \alpha_{ij}(m)\, y_j(k-m) \right] \right) \right\}$$

Generalized Linear Model (GLM)

$$\ln \lambda_i(k) = \alpha_{i0} + \sum_{j=1}^{N} \sum_{m=1}^{\tau_N} \alpha_{ij}(m)\, y_j(k-m)$$

WHY??

The parameter $\lambda_i(k)$ is the time-dependent mean of a Poisson process. In a Generalized Linear Model, it is the logarithm (link function) of the mean that is expressed as a linear combination of the observed variables. The likelihood of the observed ensemble spiking activity can then be expressed in terms of the various linear kernels, to obtain the maximum likelihood Generalized Linear Model (GLM).

GLM: exponential family

The exponential family of probability distributions is of the form:

$$\rho_y(y \mid \delta, \varphi) = \exp\left\{ \frac{y\delta - b(\delta)}{a(\varphi)} + c(y, \varphi) \right\}$$

Here, y is the random variable whose probability density function is given by ρ_y. The distribution is parametrized by δ, the canonical parameter, and φ, the dispersion parameter. The functions a(.), b(.), and c(.,.) need to be specified, and define the family.

The term $c(y, \varphi)$ plays an important role: it provides a normalization function that guarantees $\int dy\, \rho_y(y \mid \delta, \varphi) = 1$ for all δ, φ.

GLM: exponential family

Consider the family of canonical exponential distributions with canonical parameter δ and dispersion parameter φ:

$$\rho_y(y \mid \delta, \varphi) = \exp\left\{ \frac{y\delta - b(\delta)}{a(\varphi)} + c(y, \varphi) \right\}$$

WHY CARE? because the normal, Bernoulli, binomial, multinomial, Poisson, gamma, geometric, chi-square, beta, and a few other distributions are all exponential distributions.

GLM: exponential family

$$\rho_y(y \mid \delta, \varphi) = \exp\left\{ \frac{y\delta - b(\delta)}{a(\varphi)} + c(y, \varphi) \right\}$$

Since $\int dy\, \rho_y(y \mid \delta, \varphi) = 1$ for all δ, φ then:

$$\frac{\partial}{\partial \delta} \int dy\, \rho_y(y \mid \delta, \varphi) = 0 \qquad \Rightarrow \qquad \mathrm{E}(y) = b'(\delta)$$

$$\frac{\partial^2}{\partial \delta^2} \int dy\, \rho_y(y \mid \delta, \varphi) = 0 \qquad \Rightarrow \qquad \mathrm{Var}(y) = a(\varphi)\; b''(\delta)$$

Note that the canonical parameter δ fully determines the mean E(y), while the variance Var(y) requires additional information provided by the dispersion parameter through $a(\varphi)$.

McCullagh, Nelder, *Generalized Linear Models* (1989)

Poisson distribution

$$\rho_y(y \mid \lambda) = \frac{\lambda^y \; e^{-\lambda}}{y!}$$

The distribution is properly normalized:

$$\sum_{y=0}^{\infty} \frac{\lambda^y \; e^{-\lambda}}{y!} = e^{-\lambda} \sum_{y=0}^{\infty} \frac{\lambda^y}{y!} = e^{-\lambda} \, e^{+\lambda} = 1$$

and has moments:

$$\mathrm{E}(y) = \langle y \rangle = \lambda$$

$$\mathrm{Var}(y) = \langle (y - \langle y \rangle)^2 \rangle = \lambda$$

Poisson distribution

The relations: $\begin{cases} \mathrm{E}(y) = b'(\delta) \\ \mathrm{Var}(y) = a(\varphi)\ b''(\delta) \end{cases}$

hold for any probability density function within the exponential family. When applied to the Poisson case, they imply:

$$\mathrm{E}(y) = b'(\delta) = \lambda$$

$$\mathrm{Var}(y) = a(\varphi)\ b''(\delta) = \lambda$$

For Poisson statistics, $\mathrm{E}(y) = \mathrm{Var}(y) = \lambda$ implies:

$$b'(\delta) = a(\varphi)\ b''(\delta) \quad \Longrightarrow \quad \begin{cases} a(\varphi) = 1 \\ b(\delta) = e^{\delta} = \lambda \end{cases}$$

GLM: linear predictor

Since $b(\delta)$= exp(δ)= λ, the canonical parameter is δ = log λ. The dispersion parameter is not needed: $a(\varphi)$ =1. Then:

$$\rho_y(y \mid \lambda) = \exp\{y \ln \lambda - \lambda + c(y)\} = \frac{\lambda^y \; e^{-\lambda}}{e^{-c(y)}}$$

The normalization condition requires $c(y)$= - ln (y!)

In a generalized linear model based on an exponential distribution, the canonical parameter δ is constructed as a linear combination of all observed variables that can *explain* the random variable y. This linear predictor is related to the expectation value E(y) through a link function g :

$$\mathrm{E}(y) = g^{-1}(\delta) \qquad\qquad g\big(\mathrm{E}(y)\big) = \delta$$

GLM: Poisson distribution

In the Poisson case, δ = log λ = log (E(y)), and the nonlinear link function g is the logarithm:

$$\lambda = \mathrm{E}(y) = g^{-1}(\delta) = \exp(\delta)$$

$$\delta = g(\lambda) = \ln(\lambda)$$

At last!! This is why we write:

$$\ln \lambda_i\left(t \mid H(t), \{\alpha\}\right) = \alpha_{i0} + \sum_{j=1}^{N} \sum_{m=1}^{\tau_N} \alpha_{ij}(m)\, y_j(t-m)$$

Here $H(t)$ refers to the spiking history for all times $t' < t$.

Generalized Linear Model for Poisson statistics

$$\delta_i\left(t \mid H(t),\{\alpha\}\right) = \ln \lambda_i\left(t \mid H(t),\{\alpha\}\right)$$

$$= \alpha_{i0} + \sum_{j=1}^{N} \sum_{m=1}^{\tau_N} \alpha_{ij}(m)\, y_j(t-m)$$

$$\lambda_i\left(t \mid H(t),\{\alpha\}\right) = \exp\left\{\alpha_{i0} + \sum_{j=1}^{N} \sum_{m=1}^{\tau_N} \alpha_{ij}(m)\, y_j(t-m)\right\}$$

Linear-Nonlinear (NL) model! Here, the kernel parameter $\alpha_{ij}(m)$ quantifies the effect that the spiking activity of neuron j at time bin (t-m) has on the spiking activity of neuron i at time bin t.

Likelihood of spike train

$$L_T\left(\left\{y_i(k)\right\}\right) \propto \left\{\sum_{i=1}^{N}\sum_{k=1}^{K}\left(y_i(k)\ln\lambda_i(k) - \lambda_i(k)\right)\right\}$$

$$\ln\lambda_i\left(k\right) = \alpha_{i0} + \sum_{j=1}^{N}\sum_{m=1}^{\tau_N}\alpha_{ij}(m)\,y_j(k-m)$$

Given the data $\{y_i(k)\}$, find the parameters $\{\alpha\}$ that maximize L_T.

- Why is this an easy problem?
- How do we solve it?

$$L_T\left(\left\{y_i(k)\right\} \mid \left\{\alpha\right\}\right) \propto \left\{\sum_{i=1}^{N}\sum_{k=1}^{K}\left(y_i(k)\left[\alpha_{i0} + \sum_{j=1}^{N}\sum_{m=1}^{\tau_N}\alpha_{ij}(m)\,y_j(k-m)\right] - \right.\right.$$

$$\left.\left. -\exp\left[\alpha_{i0} + \sum_{j=1}^{N}\sum_{m=1}^{\tau_N}\alpha_{ij}(m)\,y_j(k-m)\right]\right)\right\}$$

Simple example: one neuron

$$L_T\left(\{y_1(k)\} \mid \{\alpha\}\right) \propto \left\{ \sum_{k=1}^{K} \left(y_1(k) \left[\alpha_{10} + \sum_{m=1}^{\tau_N} \alpha_{11}(m)\, y_1(k-m) \right] - \right.\right.$$

$$\left.\left. - \exp\left[\alpha_{10} + \sum_{m=1}^{\tau_N} \alpha_{11}(m)\, y_1(k-m) \right] \right) \right\}$$

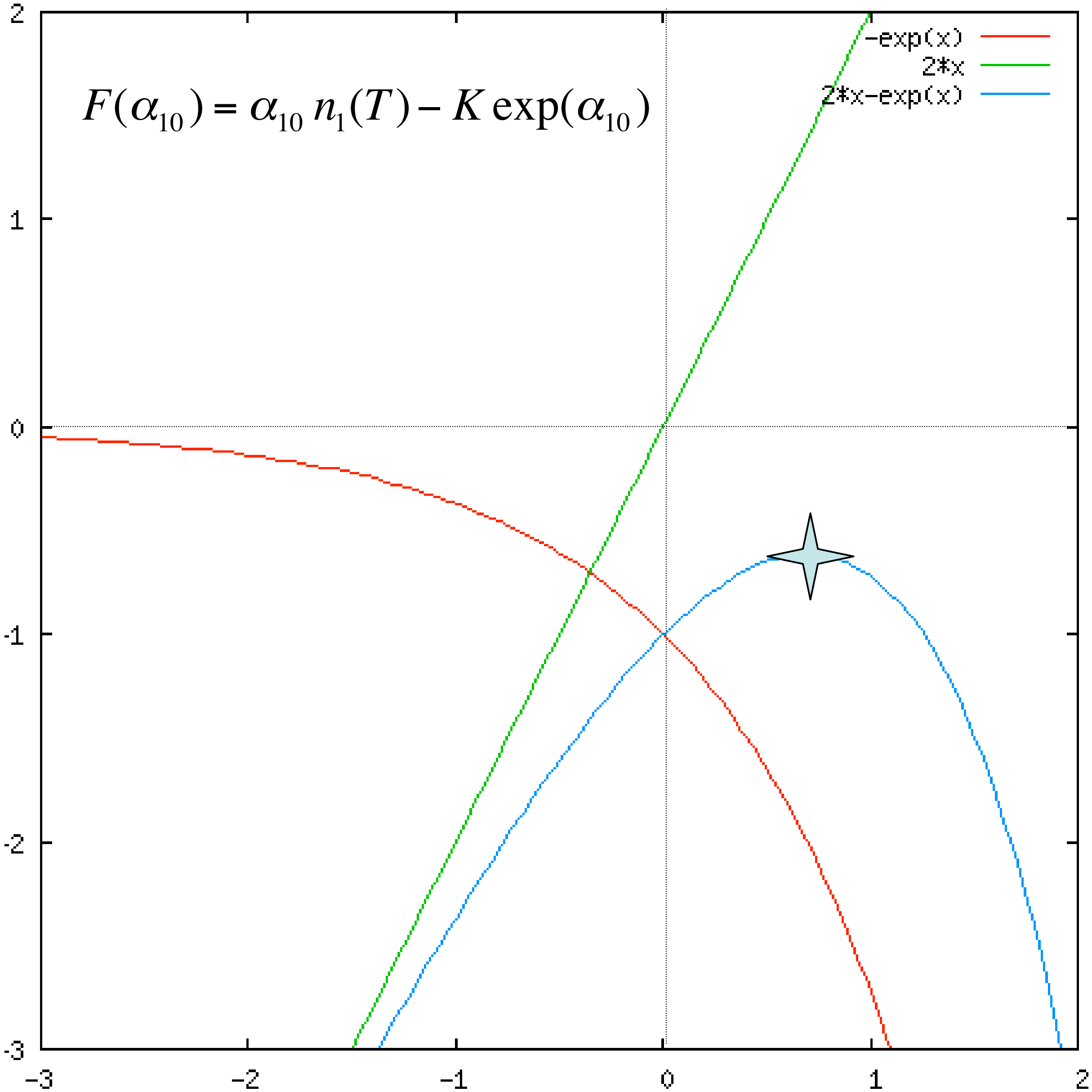
Determine the background activity: no autoregressive kernel

$$L_T\left(\{y_1(k)\} \mid \alpha_{10}\right) \propto \left\{ \sum_{k=1}^{K} \left(y_1(k) \left[\alpha_{10}\right] - \exp\left[\alpha_{10}\right] \right) \right\} =$$

$$= \alpha_{10} \sum_{k=1}^{K} y_1(k) - \sum_{k=1}^{K} \exp\left(\alpha_{10}\right) = \alpha_{10}\, n_1(T) - K \exp(\alpha_{10})$$

Here, $n_1(T)$ is the total number of spikes emitted by neuron 1 during the total time T, and $K = T/\Delta$ is the number of bins.

Simple example: likelihood function



Maximum at:

$$\alpha_{10}^{*} = \ln\left(\frac{n_1(T)}{K}\right)$$

$$\lambda^{*} = \exp\left(\alpha_{10}^{*}\right) = \frac{n_1(T)}{K}$$

Time-independent λ: the average number of spikes per bin.

Here: $\frac{n_1(T)}{K} = 2$

$$\alpha_{10}^{*} = \ln(2) \quad \Rightarrow$$

$$\lambda^{*} = \exp\left(\alpha_{10}^{*}\right) = 2$$

Iterative gradient ascent

Consider a network of N neurons. The data is of the form $\{y_i(k)\}$, for $1 \le k \le K, 1 \le i \le N$. The GLM for the likelihood of the data is:

$$L_T\left(\{y_i(k)\} \mid \{\alpha\}\right) \propto \left\{ \sum_{i=1}^{N} \sum_{k=1}^{K} \left(y_i(k) \left[\alpha_{i0} + \sum_{j=1}^{N} \sum_{m=1}^{\tau_N} \alpha_{ij}(m) y_j(k-m) \right] - \right.\right.$$

$$\left.\left. - \exp\left[\alpha_{i0} + \sum_{j=1}^{N} \sum_{m=1}^{\tau_N} \alpha_{ij}(m) y_j(k-m) \right] \right) \right\}$$

We want to find the maximum likelihood parameters $\{\alpha^*\}$ using an iterative gradient ascent method with adaptive step. To implement this algorithm, we need to compute the first and second derivatives of the likelihood with respect to the parameters $\{\alpha\}$.

Likelihood: first derivative

The gradient that drives the uphill search is given by:

$$\left.\frac{\partial L_T\left(\{y_i(k)\} \mid \{\alpha\}\right)}{\partial \alpha_{ij}(m)}\right|_{(\mu)} = \sum_{k=1}^{K} \left\{ \left[y_i(k) - < y_i(k) >^{(\mu)} \right] y_j(k-m) \right\}$$

The update of the parameter $\alpha_{ij}(m)$ is given by the product of the activity $y_j(k\text{-}m)$ of the *presynaptic* neuron *j* at time lag m and the difference between the actual activity $y_i(k)$ of the *postsynaptic* cell and our current estimate of it. The rule presynaptic activity x postsynaptic error is a famous learning rule, called the Delta Rule.

I have italicized *presynaptic* and *postsynaptic* because I do not mean to imply that the parameter $\alpha_{ij}(m)$ is an actual synaptic strength.

Likelihood: second derivative

The components of the Hessian matrix of second derivatives that controls the size of the uphill steps are given by:

$$\left.\frac{\partial^2 L_T\left(\{y_i(k)\} \mid \{\alpha\}\right)}{\partial\alpha_{ij}(m)\,\partial\alpha_{ij'}(m')}\right|_{(\mu)} = -\sum_{k=1}^{K}\left\{< y_i(k) >^{(\mu)} y_j(k-m)\; y_{j'}(k-m')\right\}$$

Now there are two *presynaptic* neurons: neuron j at time lag m and neuron j' at time lag m'. Their activities are multiplied, and this product is weighted by our current estimate of the activity of the *postsynaptic* neuron. Note the overall minus sign! The variables y represent number of spikes emitted during a bin of size Δ. These variables, and their averages, are always non negative. Thus, every component of the Hessian matrix is negative - the surface is everywhere downward concave.

Likelihood maximization

The algorithm can now be written as follows:

$$\vec{\alpha}^{(\mu+1)} = \vec{\alpha}^{(\mu)} + \mathrm{E}^{(\mu)} \nabla L^{(\mu)}$$

Here, $\vec{\alpha}$ is a listing of all the parameters needed to specify the model; ∇L is the gradient of the likelihood function L, obtained by taking a derivative of L with respect to every parameter in $\vec{\alpha}$; and E is the matrix of step sizes, obtained by inverting the Hessian matrix of second derivatives of the likelihood function.

If the model requires p parameters, then both $\vec{\alpha}$ and ∇L are p-dimensional vectors, and E is a p x p matrix. For instance, p=1+τ_N for the autoregressive model:

$$\lambda_i\left(t \mid H_i(t), \{\alpha\}\right) = \exp\left\{\alpha_{i0} + \sum_{m=1}^{\tau_N} \alpha_{ii}(m)\, y_i(t-m))\right\}$$

Uphill iteration

Given the data $\{y_i(m)\}$ and the current value $\{\alpha^{(\mu)}\}$ of the parameters, construct:

1] $$< y_i(k) >^{(\mu)} = \exp\left[\alpha_{i0}^{(\mu)} + \sum_{j'=1}^{N} \sum_{m'=1}^{\tau_N} \alpha_{ij'}^{(\mu)}(m')\, y_{j'}(k-m')\right]$$

Once you have the $<y_i(k)>^{(\mu)}$, you do not need the parameters any more. Build the components of the gradient vector:

2] $$\left.\frac{\partial L_T\left(\{y_i(k)\} \mid \{\alpha\}\right)}{\partial \alpha_{ij}(m)}\right|_{(\mu)} = \sum_{k=1}^{K} \left\{\left[y_i(k) - < y_i(k) >^{(\mu)}\right] y_j(k-m)\right\}$$

Uphill iteration

Build the components of the Hessian matrix:

3] $$\left.\frac{\partial^2 L_T\left(\{y_i(k)\} \mid \{\alpha\}\right)}{\partial \alpha_{ij}(m)\, \partial \alpha_{ij'}(m')}\right|_{(\mu)} = -\sum_{k=1}^{K} \left\{ < y_i(k) >^{(\mu)} \; y_j(k-m)\; y_{j'}(k-m') \right\}$$

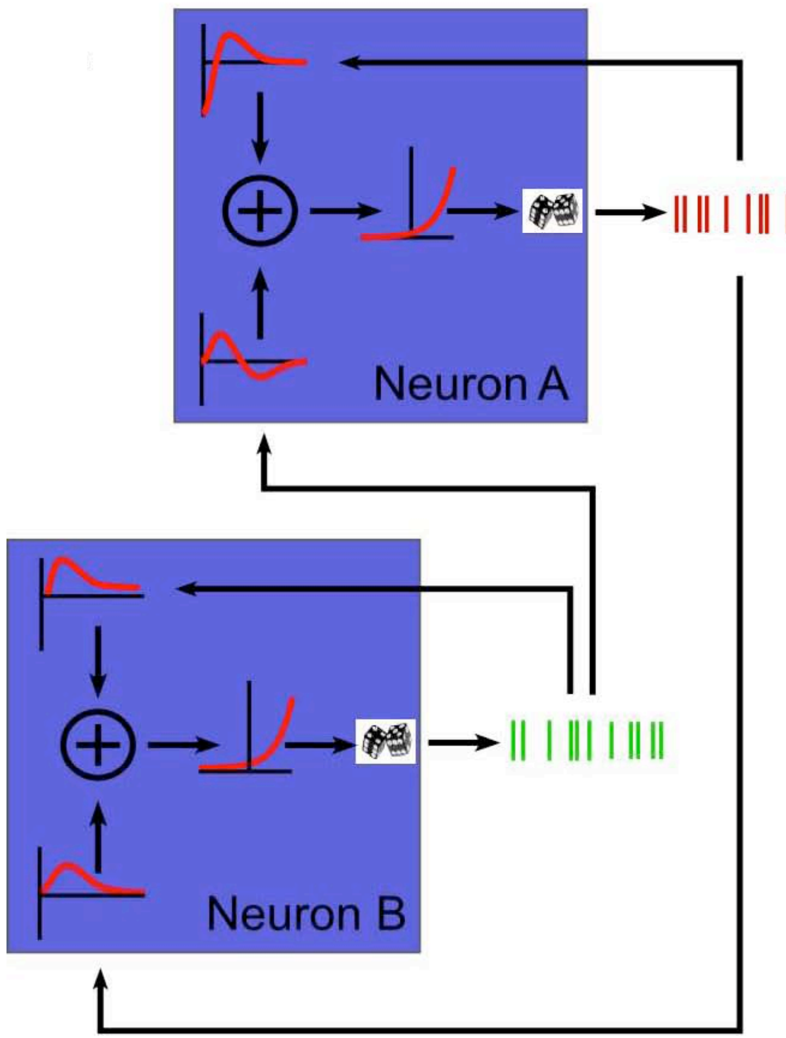
Invert the Hessian matrix of second derivatives to obtain the matrix Epsilon of step sizes:

4] $$\mathrm{E} = -\frac{1}{H}$$

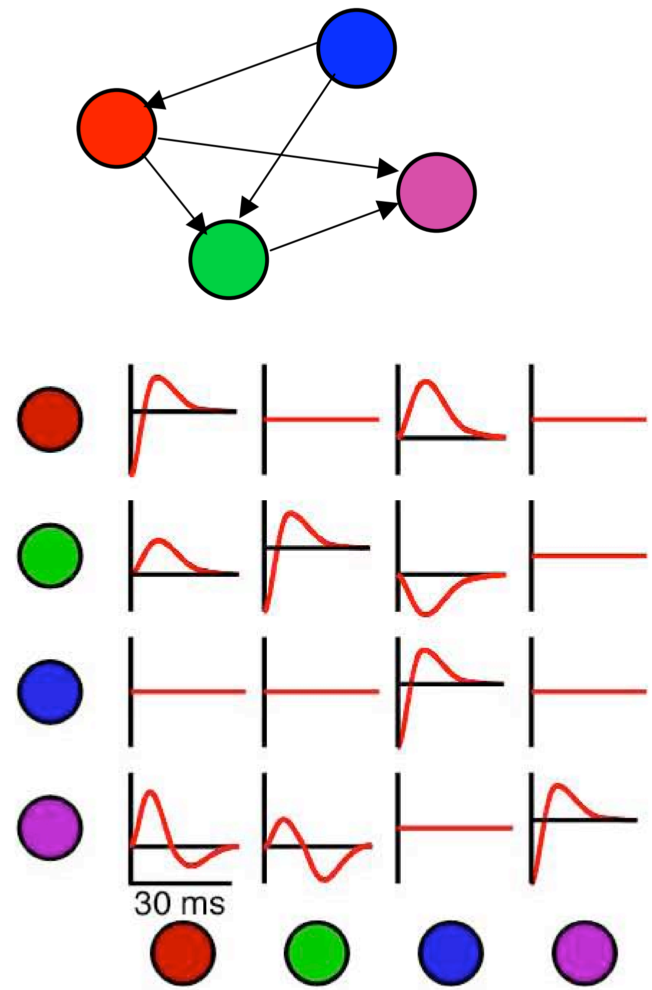
Multiply the matrix E and the gradient ∇L to obtain the update:

5] $$\vec{\alpha}^{(\mu+1)} = \vec{\alpha}^{(\mu)} + \mathrm{E}^{(\mu)} \nabla L^{(\mu)}$$

GLM model: functional connectivity

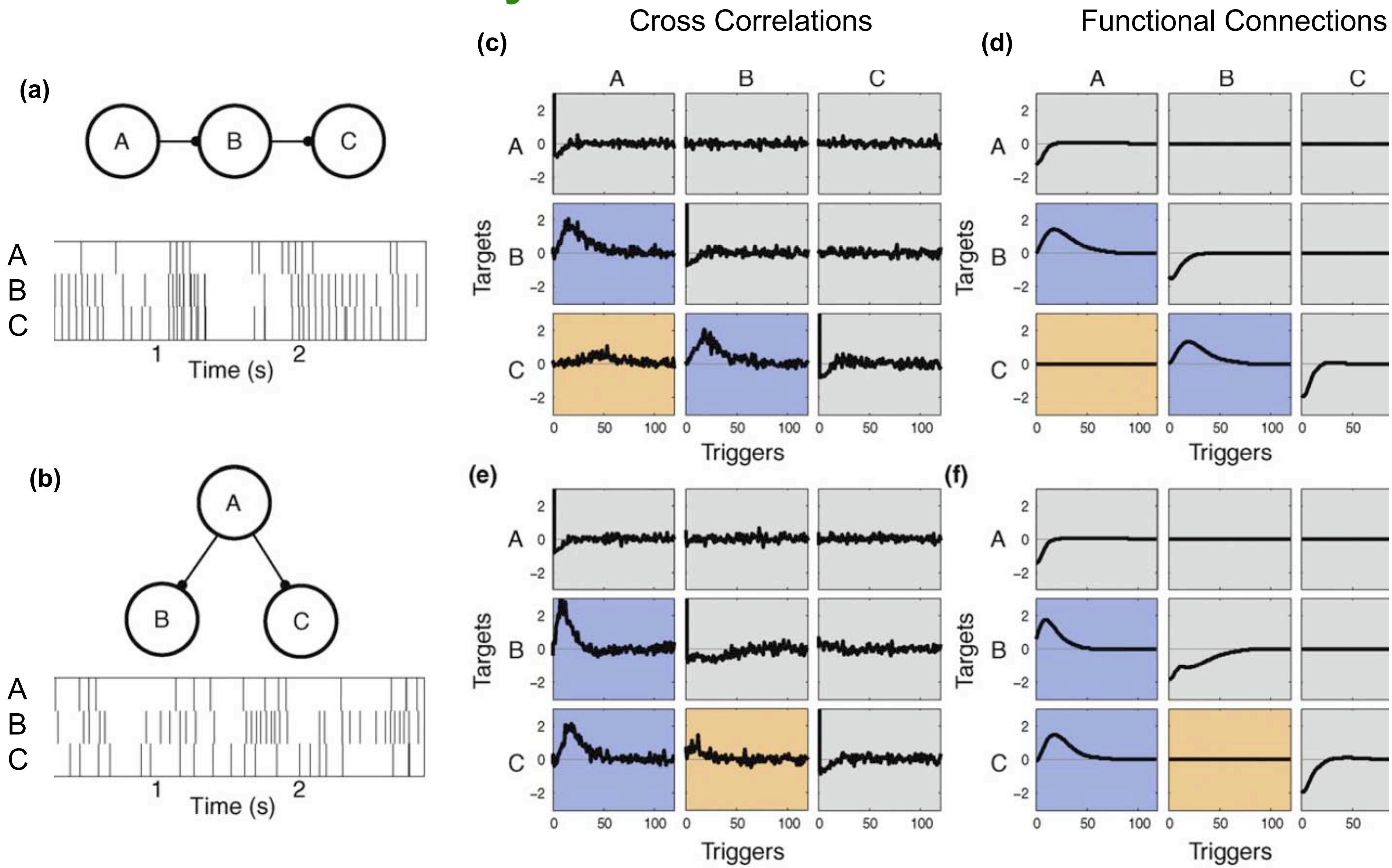


Generative model



Connectivity kernels

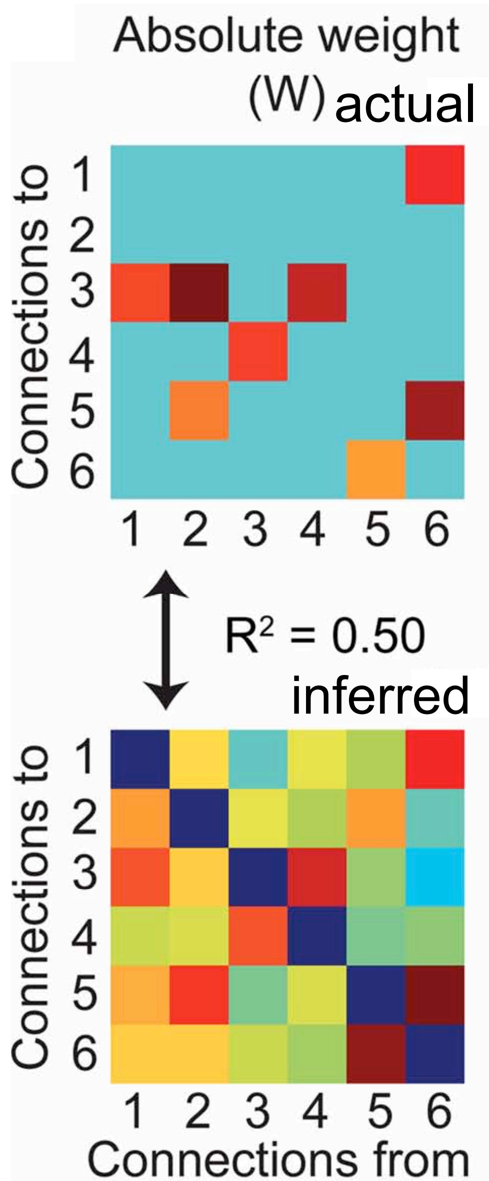
Functional connectivity: beyond correlations



Functional connectivity: network reconstruction

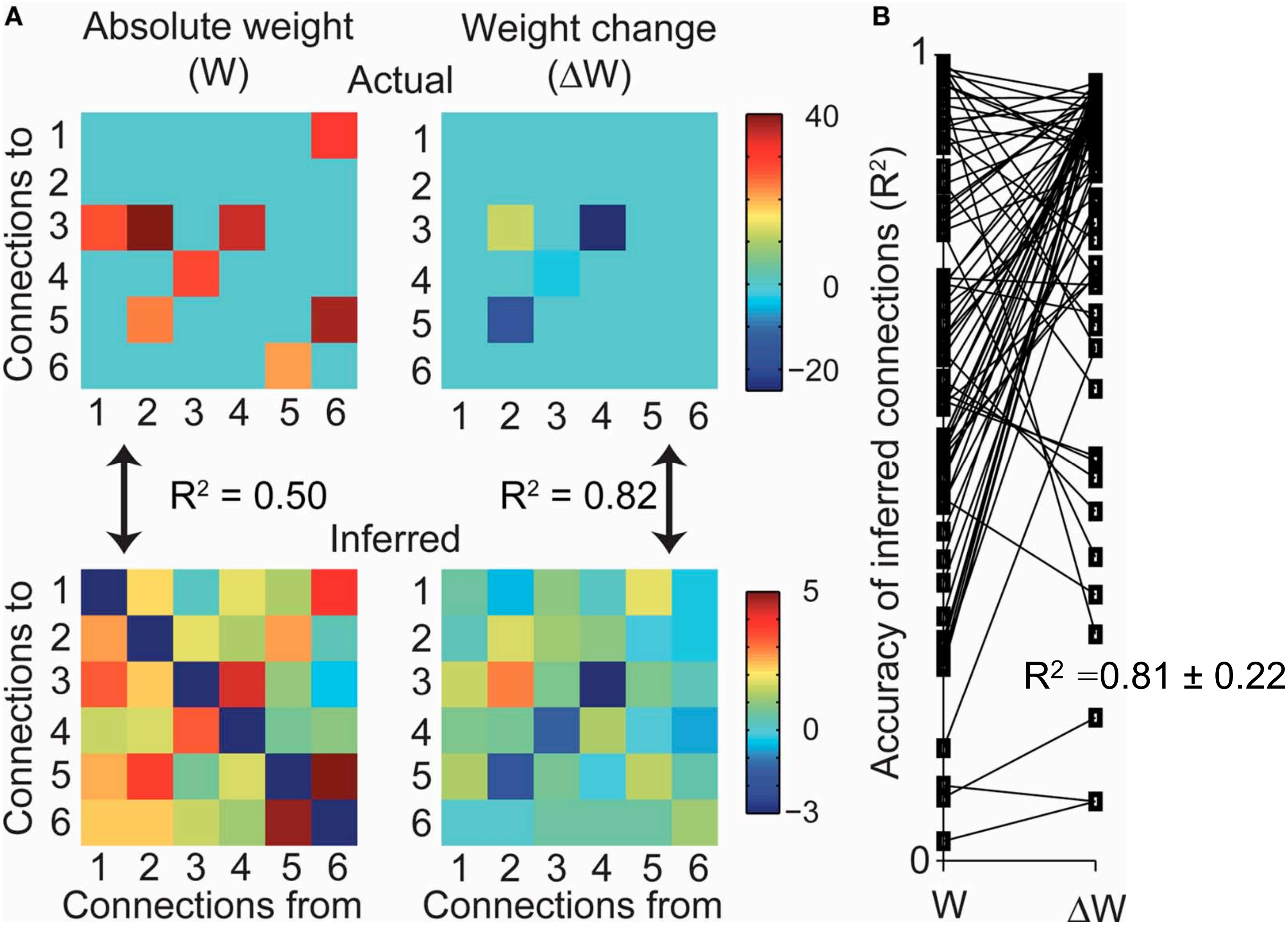
80 random draws of N=6 neurons from a network of n=10,000 Izhikevich neurons

$$W_{ij} = \sum_m \left| \alpha_{ij}(m) \right|$$



$R^2 = 0.61 \pm 0.25$

Functional connectivity: monitoring connectivity changes



Induced connectivity changes

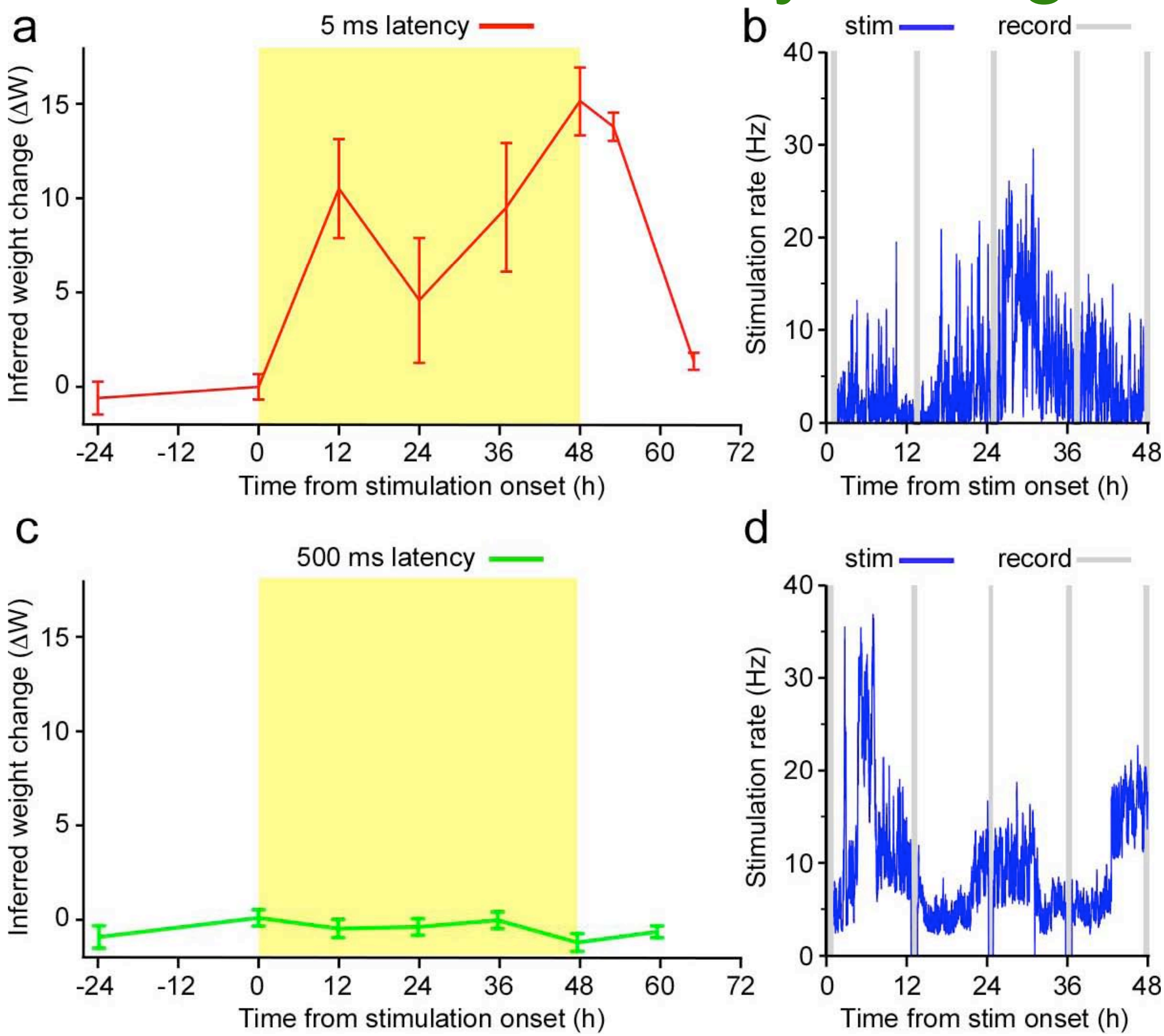
GOAL: induce in vivo changes in functional connectivity

SYSTEM: rat forelimb sensorimotor cortex

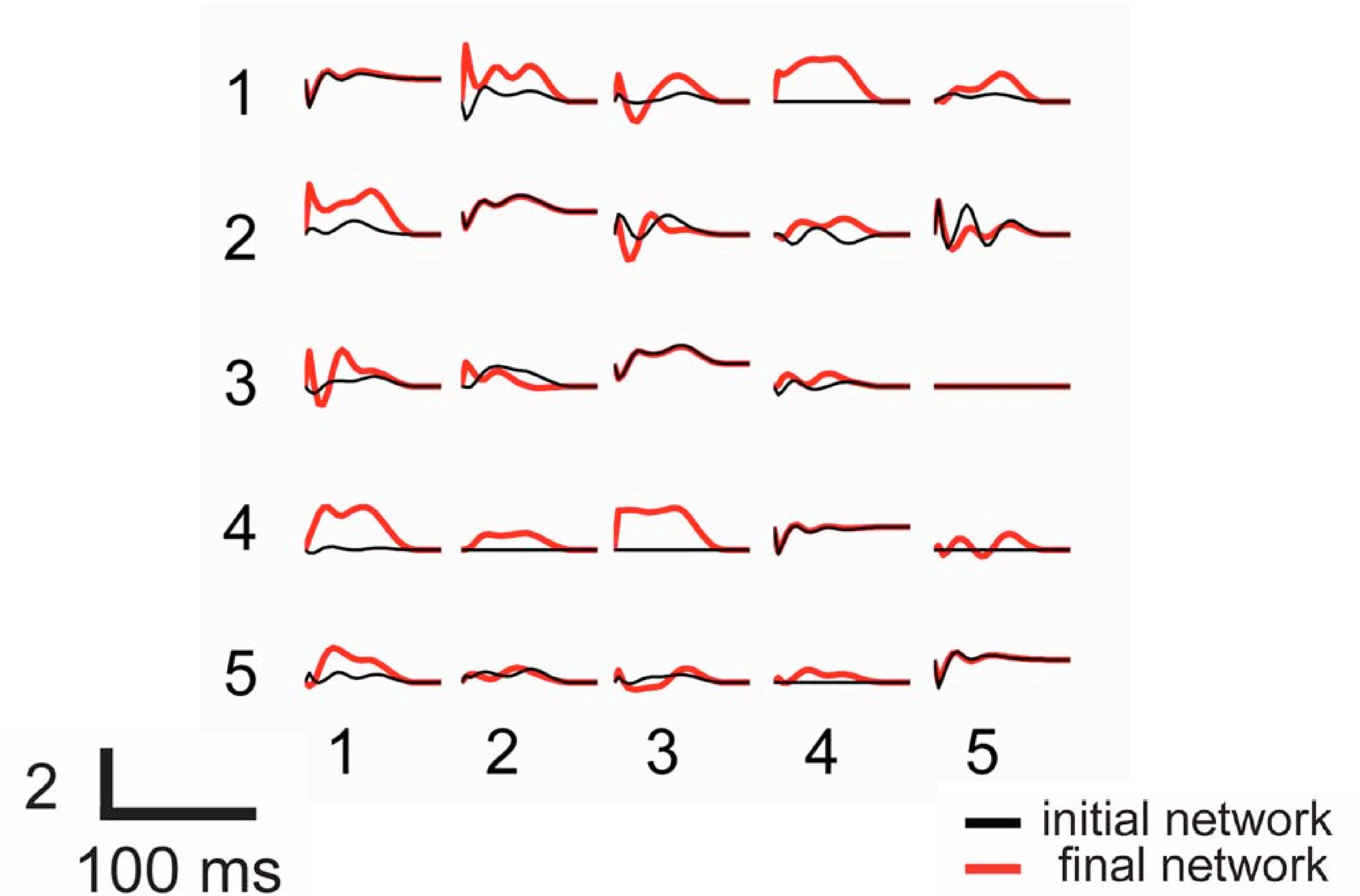
Fine wire electrodes used to record the activity of N = 4-9 neurons. One neuron is randomly chosen as the *trigger* and used to control the stimulation. Every spike of the trigger neuron was followed by a stimulation pulse delivered through a *target* electrode at a fixed latency.

The electrical stimulation was an exact, time-lagged replica of the trigger spike train.

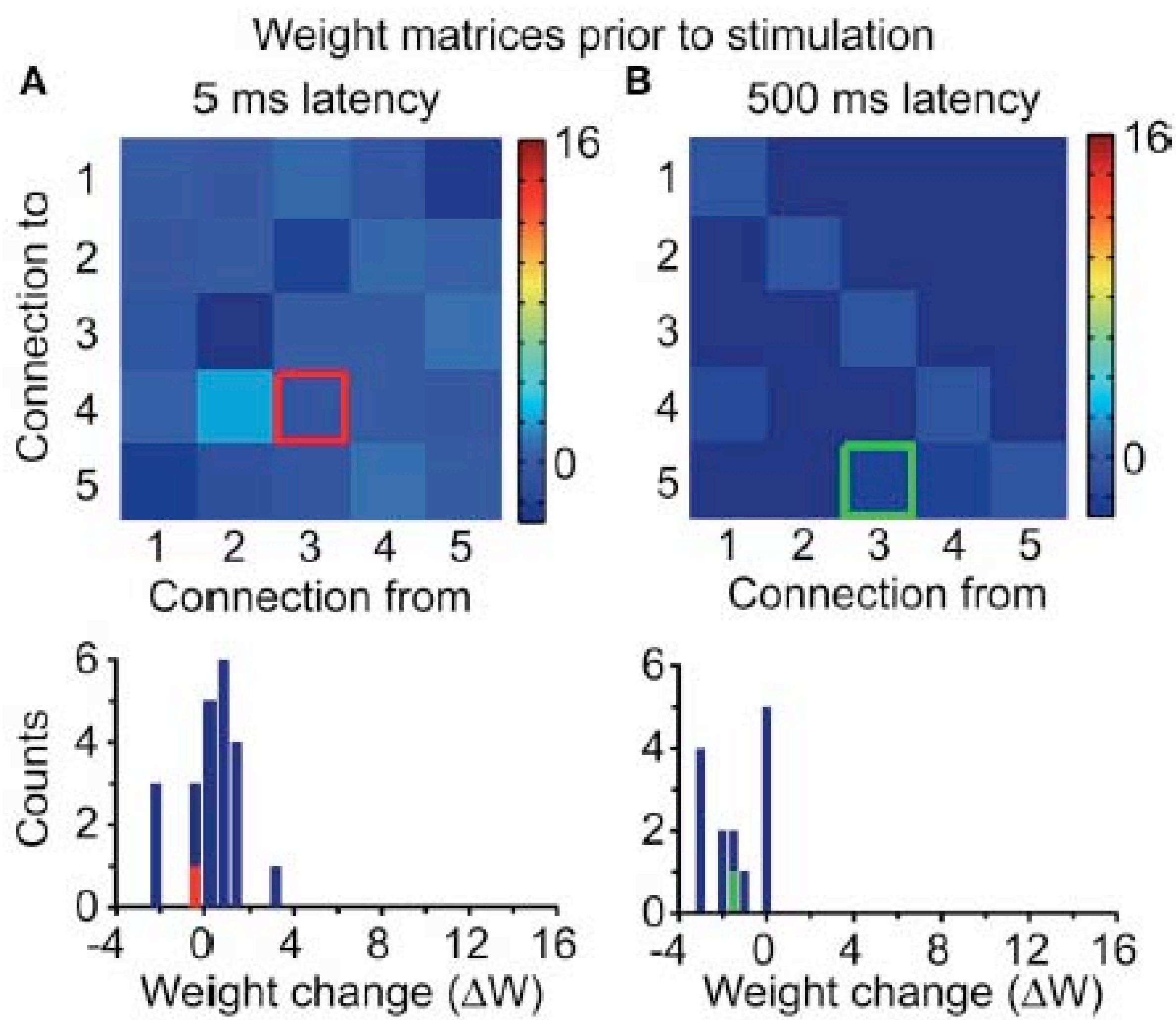
Induced connectivity changes



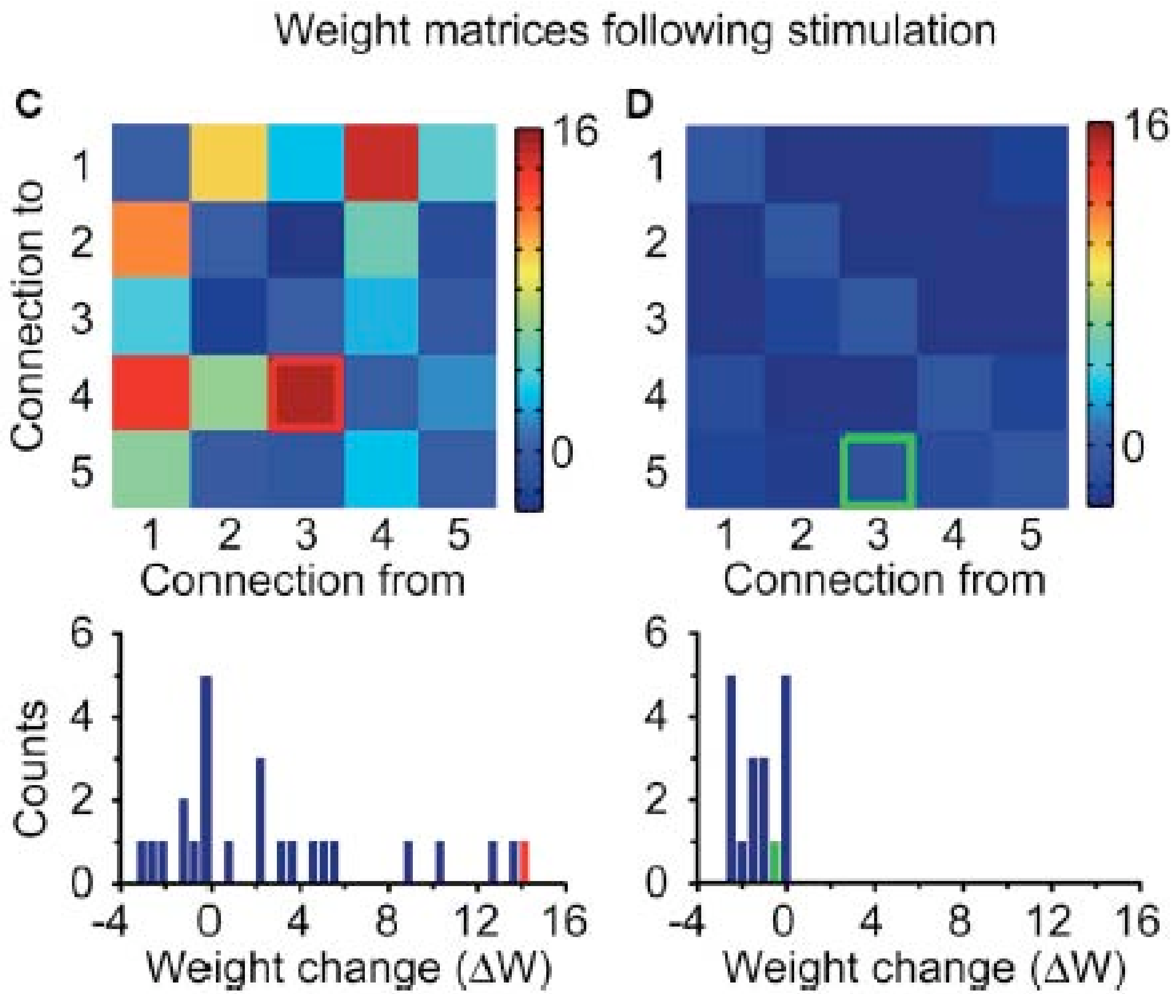
Connectivity kernels



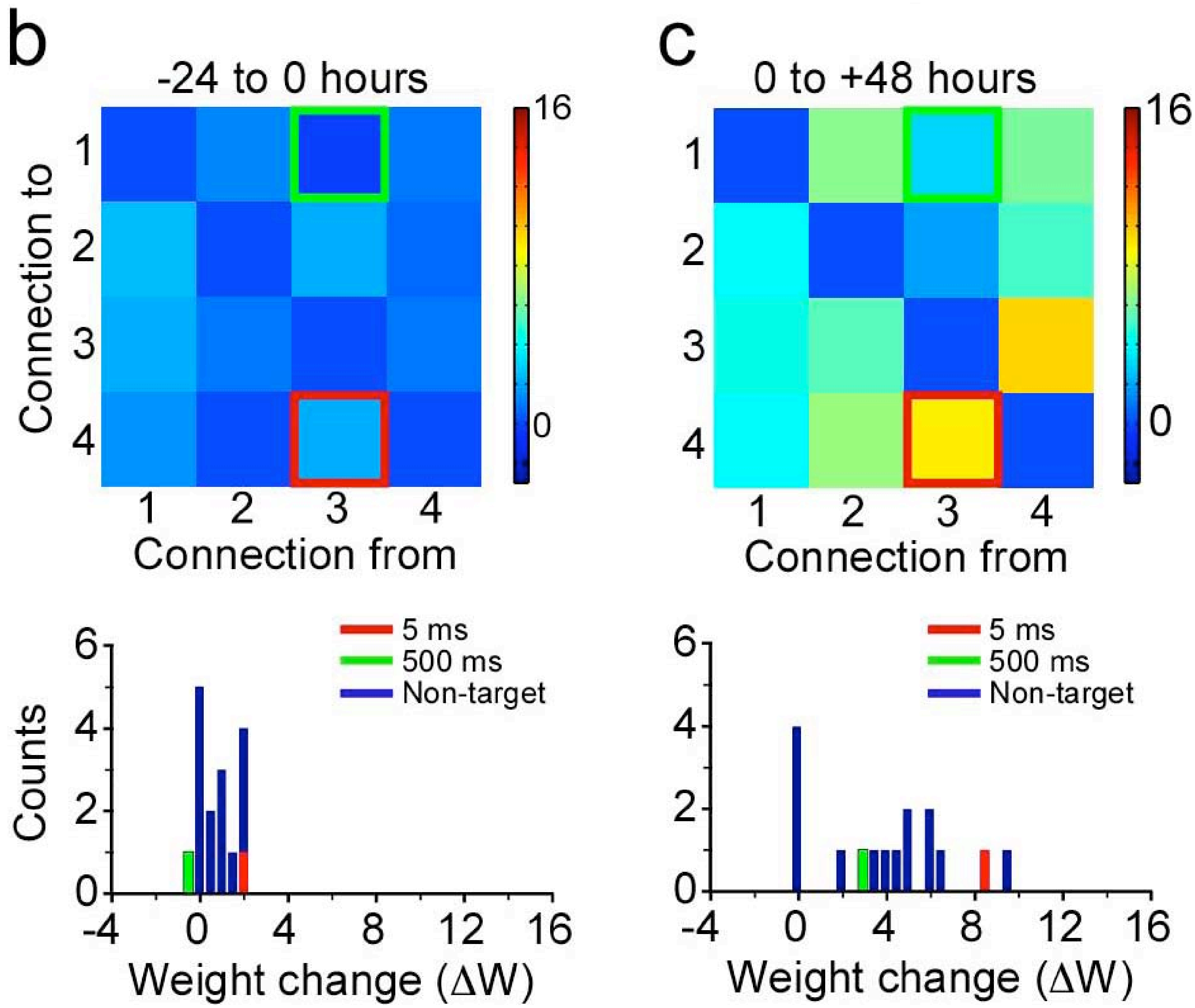
Connectivity changes: no stimulation



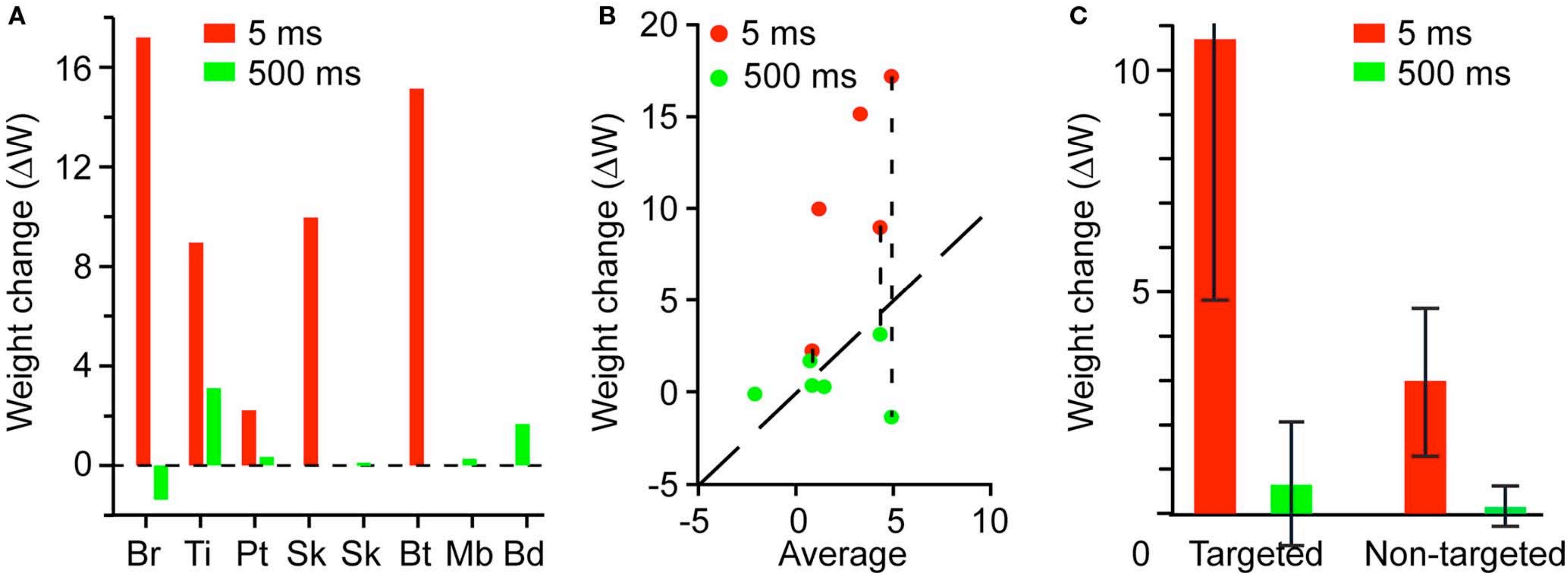
Connectivity changes induced by stimulation



Induced connectivity changes



Induced connectivity changes



Summary

• In vivo activity-triggered stimulation has been shown to induce changes consistent with STDP.

• Changes in synaptic efficacy have been detected without invoking changes in stimulus-evoked postsynaptic activity.

• Generalized Linear Models have been shown to provide a useful tool for monitoring induced changes in connectivity.