bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

# Active Learning of Cortical Connectivity from Two-Photon Imaging Data

Martín Bertrán[1◑*], Natalia Martínez[1◑], Ye Wang[2], David Dunson[2], Guillermo Sapiro[1,3‡], Dario Ringach[4‡]

**1** Electrical and Computer Engineering, Duke University, Durham, North Carolina, USA;
**2** Statistical Science Program, Duke University, Durham, North Carolina, USA;
**3** BME, CS, and Math, Duke University, Durham, North Carolina, USA;
**4** Neurobiology and Psychology, Jules Stein Eye Institute, Biomedical Engineering Program, David Geffen School of Medicine, University of California, Los Angeles, California, USA;

◑These authors contributed equally to this work.
‡Equal senior contributors.
* martin.bertran@duke.edu

## Abstract

Understanding how groups of neurons interact within a network is a fundamental question in system neuroscience. Instead of passively observing the ongoing activity of a network, we can typically perturb its activity, either by external sensory stimulation or directly via techniques such as two-photon optogenetics. A natural question is how to use such perturbations to identify the connectivity of the network efficiently. Here we introduce a method to infer sparse connectivity graphs from *in-vivo*, two-photon imaging of population activity in response to external stimuli. A novel aspect of the work is the introduction of an oracle which, at any point in time, can recommend a distribution of external stimuli that will optimally refine the inferred network. Unlike existing system identification techniques, this "active learning" method automatically focuses its attention on key undiscovered areas of the network, instead of targeting global uncertainty indicators like parameter variance. We show how active learning leads to faster inference while, at the same time, provides confidence intervals for the network parameters. We present simulations on artificial small-world networks to validate the methods and apply the method to real data. Analysis of frequency of motifs recovered show that cortical networks are consistent with a small-word topology model.

## Introduction

A fundamental question of system neuroscience is how large groups of neurons interact, within a network to perform computations that go beyond the individual ability of each one. One hypothesis is that the emergent behavior in neural networks results from their organization into a hierarchy of modular sub-networks, or motifs, each performing simpler computations than the network as a whole [1].

To test this hypothesis and to understand brain networks in general we need to develop methods that can reliably measure network connectivity, detect recurring motifs, elucidate the computations they perform, and understand how these smaller

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

modules are combined into larger networks capable of performing increasingly complex computations.

Here we focus on the first of these problems, which is a pre-requisite to the rest: the identification of network connectivity from *in-vivo*, two-photon imaging data. Advances in two-photon imaging are giving us the first look at how large ensembles of neurons behave *in-vivo* during complex behavioral tasks [2–4]. Developing methods capable of analyzing the connectivity between a large number neurons, from noisy, stochastic activations, and limited recording time, is a significant challenge.

It is often the case that we can probe the networks under study, instead of merely observing their ongoing activity. For example, in studying visual cortex we can select a specific visual stimulus [5–8], or we can stimulate individual neurons directly via two-photon optogenetics [9, 10]. This active observation has been shown critical for system identification beyond brain networks, and is the direction here pursued.

We introduce a method that infers a sparse connectivity graph from available simultaneous recording of external stimuli and individual neural spiking rates, and recommends the future distribution of external stimuli to apply in order to optimally refine the inferred network. We show how such iterative "active learning" leads to faster inference while at the same time providing confidence measurements for the computed network components. The proposed decision-making approach takes into account information we may already have about network connectivity, the stochastic nature of the neural responses, and the uncertainty of connection weights.

Our framework consists of modelling the neuron spiking rates using a Generalized Linear Model (GLM) [11], using past spiking rates and applied stimuli as regressors. The coefficients of these regressors in the GLM model make the edge weights of the directed network. A variable selection approach is used to make the network sparse (set most edges to zero). Active learning is then used to decide, at any given point in time, the next sets of stimuli that allow for optimal inference of the network connectivity. Fig 1 shows a visual representation of the proposed framework.

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

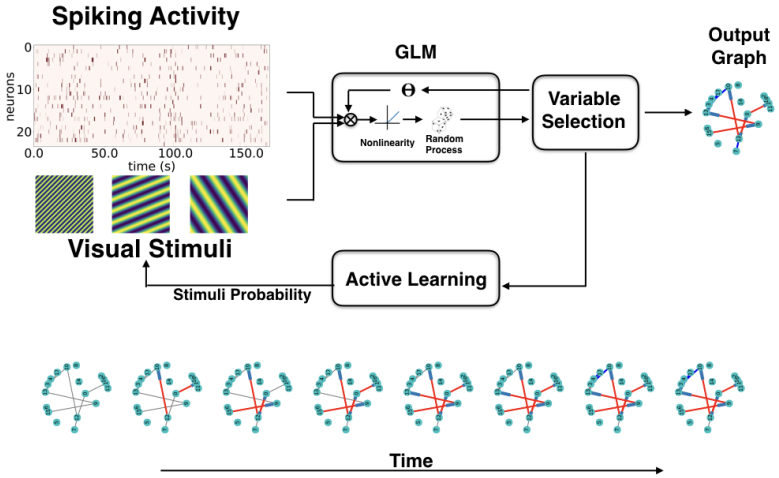



**Fig 1.** Recordings of spiking activity of a neuron population and the presented visual stimuli are fed into a GLM model. The GLM and Variable Selection blocks work in tandem to decide which connections are relevant for explaining the system's behaviour (the data) and building the directed connectivity graph (network). The active learning component analyzes the data obtained so far to optimize the visual stimuli to be presented for the next step of data acquisition, this is done to reduce graph uncertainty. This process is iteratively repeated. The bottom row shows how the network is gradually reconstructed as a function of acquired samples. Gray edges represent yet undiscovered edges present in the network, while red and blue edges represent discovered excitatory and inhibitory edges respectively.

While the proposed framework is general, we illustrate it by applying it to two-photon imaging data from mouse primary visual cortex. We also validate the method's effectiveness on *in-silico* network simulations.

# Materials and methods

Following the brief description of the data acquisition, we then describe the foundation of the proposed active network inference framework. In doing so, we use terminology that will be relevant for the particular application at hand: the estimation of connectivity in neural networks. However, the framework is general enough to be applied in other contexts and using other modalities.

## Data acquisition

Animals: All procedures were approved by UCLA's Office of Animal Research Oversight (the Institutional Animal Care and Use Committee), and were in accord with guidelines set by the US National Institutes of Health. The present study used data already collected for other studies. Thus, no new animal experiments were performed for the purposes of the present study. A detailed account of the experimental methods can be found elsewhere [12]. A brief description follows.

Imaging: Imaging of GCaMP6f expressed in primary visual cortex was performed using a resonant, two-photon microscope (Neurolabware, Los Angeles, CA) controlled

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

by Scanbox acquisition software (Scanbox, Los Angeles, CA). The light source was a Coherent Chameleon Ultra II laser (Coherent Inc, Santa Clara, CA) running at 920nm. The objective was an x16 water immersion lens (Nikon, 0.8NA, 3mm working distance). The microscope frame rate was 15.6Hz (512 lines with a resonant mirror at 8kHz). Eye movements and pupil size were recorded via a Dalsa Genie M1280 camera (Teledyne Dalsa, Ontario, Canada) fitted with a 740 nm long-pass filter that looked at the eye indirectly through the reflection of an infrared-reflecting glass. Images were captured at an average depth of 260 µm.

Sequences of pseudo-random sinusoidal gratings [5, 13] and sparse noise stimuli were generated in real-time by a Processing sketch using OpenGL shaders (see `http://processing.org`). A detailed description is provided in [14]. The duration of the sequences was either 20 or 30 min, and gratings were updated 4 times a second on a screen refreshed at 60Hz. In a 20 min sequence, each combination of orientation and spatial frequency appeared at least 22 times on average.

In all experiments we used a BenQ XL2720Z screen which measured 60 cm by 34 cm and was viewed at 20 cm distance, subtending 112 x 80 degrees of visual angle. The screen was calibrated using a Photo-Research (Chatsworth, CA) PR-650 spectro-radiometer, and the result used to generate the appropriate gamma corrections for the red, green and blue components via an nVidia Quadro K4000 graphics card. The contrast of the stimulus was 80%. The center of the monitor was positioned with the center of the receptive field population for the eye contralateral to the cortical hemisphere under consideration. The location of the receptive fields were estimated by an automated process where localized, flickering checkerboards patches, appeared at randomized locations within the screen. This experiment was run at the beginning of each imaging session to ensure the centering of receptive fields on the monitor.

Image processing: The image processing pipeline was the same as described in detail elsewhere [12]. Briefly, calcium images were aligned to correct for motion artifacts. Following motion stabilization, we used a Matlab graphical user interface (GUI) tool developed in our laboratory to define regions of interest corresponding to putative cell bodies manually. Following segmentation, we extracted signals by computing the mean of the calcium fluorescence within each region of interest and discounting the signals from the nearby neuropil. Spikes were then estimated via deconvolution [15]. The present results are based on the inferred spiking activity.

## Generalized linear models: Poisson point process

Let $X = \{X_1, ..., X_{n_c}\}$ and $R = \{R_1, ..., R_{n_r}\}$ be two sets of random variables called the target variables and regressor variables respectively. In our case, the target variables $X$ are the spike time series of the $n_c$ observed neurons. The regressors $R$ are $n_r$ time series that carry information about the spiking activity of a neuron, they consist of the past spiking activity of all the observed neurons and the history of the presented external stimuli.

We say that the set $X$ is a Poisson Point Process if the conditional probability distribution (CPD) is of the form

$$X_c|R \sim Poisson\Big(f_c\Big(R\Big)\Big).$$

where $f_c$ is a real-valued function, $\forall\, c = 1, ..., n_c$.

A Poisson Generalized linear model (GLM) is a special case of a general Poisson point process where

$$X_c|R \sim Poisson(f(b_c + \sum_{i \in PA_c} R_i \times w_{ic})).$$

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

$f : \mathbf{R} \to \mathbf{R}^+$ is a predefined link function, $w_{ic}$ are the influence weights, $b_c$ is the bias weight, and $PA_c$ is the parent set of $X_c$, and is the subset of the regressors $R$ that carry information on the behaviour of neuron $c$. In our case $PA_c$ consists of visual stimuli and past neuron activity that directly affect the spiking rate of neuron $c$. The parameter $w_{ic}$ represents the magnitude of that influence.

The overall goal of network learning is finding the set $(PA_c, w_{ic})$, for each neuron $c$, that best represents the recorded data (stimuli and neural activity).

In the following sections we will present the model in more detail and the chosen algorithm for selecting the relevant regressors $PA_c$ and influence weights $w_{ic}$. Then we will introduce an experimental design method to select the stimuli that are more relevant in discovering the structure of the network. We want to emphasize that we are especially interested in correctly inferring the regressor set $PA_c$, this can be seen as a binary classification problem, since a given regressor either belongs or does not belong on $PA_c$ for any given neuron $c$. This will directly translate into the decision of which sets of stimuli are relevant. This decision will be based on improving the $PA_c$ classification performance.

## Model description

The system under study consists of a set of $n_c$ neurons $\{C\}$ and $n_s$ stimuli or external inputs $\{S\}$, that can be used to perturb the neuron's activity.

To differentiate the influence of spiking activity and visual stimuli we will split the set of regressors $R$ explicitly into spike activity $X_j(t), j = 1, .., n_c$, and stimuli activity $I_i(t), i = 1, .., n_s$.

To incorporate information about the past observations we further redefine the regressors as $\hat{X}_j(t) = \sum_{u=t-D_c^u}^{t-D_c^l} X_j(u) = (\mathbf{1}_{[t-D_c^l, t-D_c^u]} * X_j)(t), j = 1, .., n_c$, and $\hat{I}_i(t) = \sum_{u=t-D_s^u}^{t-D_s^l} I_i(u) = (\mathbf{1}_{[t-D_s^l, t-D_s^u]} * I_i)(t), i = 1, .., n_s$, which are the convolution of past spiking activity $(X_j, j = 1..n_c)$ and past stimuli activity $(I_i, i = 1..n_s)$ with the boxcar influence function up to delays $D_c^u$ and $D_s^u$ respectively.

To model the spiking train of neuron $c$, we also define the sets of relevant regressors of neuron $c$: $C'_c = \{PA_c \cap C\}$ and $S'_c = \{PA_c \cap S\}$.

Under these conditions, the spiking train of neuron $c$ can be modeled as

$$\eta_c(t) = b_c + \langle W_c^T, \{\hat{X}_u(t)\}_{u \in C'_c} \rangle + \langle H_c^T, \{\hat{I}_v(t)\}_{v \in S'_c} \rangle \; \forall t, \tag{1}$$

$$\lambda_c(\eta_c(t)) = \frac{\log(1 + e^{\kappa \eta_c(t)})}{\kappa}, \tag{2}$$

$$X_c(t) | \{\hat{X}_u(t)\}_{u \in C'_c}, \{\hat{I}_v(t)\}_{v \in S'_c} \sim \text{Poisson}(\lambda_c(\eta_c(t)), \tag{3}$$

where $< . >$ denotes inner product. The parameters of interest are:

- $W_c \in R^{|C'_c|}$: Edge weights between parent neurons in $C'_c$ and $c$. These weights collectively represent the inter-neuron connectivity matrix
$$W = \{w_{ic} : i, c = 1, ..., n_c\} = \begin{cases} 0 \leftrightarrow i \notin PA_c, \\ W_c(i) \leftrightarrow i \in PA_c. \end{cases}$$

- $H_c \in R^{|S'_c|}$: Edge weights between parent stimuli in $S'_c$ and $c$. These weights collectively represent the direct stimuli response matrix
$$H = \{h_{jc} : j = 1, ..., n_s, \; c = 1, ..., n_c\} = \begin{cases} 0 \leftrightarrow j \notin PA_c, \\ H_c(j) \leftrightarrow j \in PA_c. \end{cases}$$

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

- $b_c \in R$: Bias of neuron $c$. This number encodes the base spiking rate of neuron $c$ independently of the state of the other regressors.

Note that the model is time homogeneous, i.e., $W_c$, $H_c$ and $b_c$ do not depend on time. The motivation for using the link function Eq. (2) instead of the canonical log link function is that Eq. (2) is pseudo linear in the parameter $\eta_c$. $\kappa$ is a static calibration constant to ensure the link function is operating in the linear approximation region when no strong inhibitions are active. Strong inhibitions can still drive the link function into the nonlinear operation region.

## Parameter estimation

For each observed neuron $c$ we want to find the regressor sets $C'_c$ and $S'_c$ that best explain the data without over-fitting.

We first define the likelihood function of the proposed model,

$$\begin{aligned}\mathcal{L}(\mathrm{X_c}|\{\hat{\mathrm{X}}\}_{\mathrm{C'_c}}, \{\hat{\mathrm{I}}\}_{\mathrm{S'_c}}, \mathrm{W_c}, \mathrm{H_c}) &= \log[\mathrm{f}(\mathrm{X_c}|\{\hat{\mathrm{X}}\}_{\mathrm{C'_c}}, \{\hat{\mathrm{I}}\}_{\mathrm{S'_c}}, \mathrm{W_c}, \mathrm{H_c})] \\ &\propto \sum_{t=1}^{m}[X_c(t) log[\lambda_c(\eta_c(t))] - \lambda_c(\eta_c(t))].\end{aligned} \tag{4}$$

A good regressor should provide a significant improvement in the model likelihood, and should have a tight confidence interval around its estimated edge weight. These notions are formalized using the Bayesian Information Criterion (BIC) [16] and the Wald test [17] respectively. A derivation of both in the context of our model is provided below.

To estimate the values for a set of parameters $\theta = \{W_c, H_c, b_c\}$ we utilize the standard Maximum Likelihood Estimation (MLE) framework. The MLE estimate is denoted as $\hat{\theta} = \{\hat{W}_c, \hat{H}_c, \hat{b}_c\}$ and is obtained as the solution of

$$\hat{\theta} = \{\hat{\mathrm{W}}_\mathrm{c}, \hat{\mathrm{H}}_\mathrm{c}, \hat{\mathrm{b}}_\mathrm{c}\} = \mathrm{argmax}_{\{\mathrm{W_c}, \mathrm{H_c}, \mathrm{b_c}\}} \sum_{\mathrm{t=1}}^{\mathrm{m}}[X_c(t) log[\lambda_c(\eta_c(t))] - \lambda_c(\eta_c(t))]. \tag{5}$$

Furthermore, the parameter obtained from this estimation asymptotically follows a Normal distribution around the true value $\theta_0$ [18]. Under further regularity conditions [19], the variance of the estimator can be computed as shown in Eq. (6) (this will form the theoretical basis for the notion of tight confidence intervals),

$$\begin{aligned}\lim_{t\to\infty} P(|\hat{\theta} - \theta_0| > \epsilon) = 0 \; \forall \epsilon > 0, \\ \hat{\theta} \approx N[\theta_0, \{I(\theta_0)\}^{-1}]. \\ \mathcal{I}(\theta_0) = -E[\partial^2 \mathcal{L}/\partial\theta_0 \partial\theta'_0].\end{aligned} \tag{6}$$

The quantity $\mathcal{I}(\theta_0)$ is the Fisher information matrix [19]. Since we do not have access to the true parameter $\theta_0$ we use the observed Fisher information ($\mathcal{I}(\hat{\theta}) = -E[\partial^2 \mathcal{L}/\partial\hat{\theta}\partial\hat{\theta}']$) as an approximation. The quantity $\mathcal{I}\{(\theta_0)\}^{-1}$ is a lower bound for the variance of any unbiased estimator, as stated by the Cramér-Rao lower bound [20, 21]. The observed Fisher information of the model can be found on Eq. (27) in the Appendix.

The MLE estimators combined with the observed Fisher information provide a confidence interval for each parameter of interest. We start from the null hypothesis $H_0$

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

that the edge is irrelevant (zero value), and use the Wald test [17], to accept or reject this. The probability of parameter $\theta_j$ belonging to the null hypothesis can be computed as

$$\frac{\hat{\theta}_j}{[\hat{\mathcal{I}}(\hat{\theta})]^{-1}_{j,j}} \sim \mathcal{X}_1^2, \tag{7}$$

$$p_{H_0} = 1 - F_{\mathcal{X}_1^2}(\frac{\hat{\theta}_j}{[\hat{\mathcal{I}}(\hat{\theta})]^{-1}_{j,j}}), \tag{8}$$

where $\mathcal{X}_1^2$ is the chi-square distribution with one degree of freedom, $F_{X_1^2}(z)$ is the cumulative distribution function $\mathcal{X}_1^2$ evaluated at $z$, and $p_{H_0}$ is the p-value associated with the null hypothesis.

At this point we have derived a measure of the probability of the parameter being different from zero.

To measure how informative a given regressor is we use the Bayesian information criteria (BIC). This quantity decreases with a higher likelihood $\hat{L}$ and increases with the number of parameters currently used in the model ($|\widehat{PA}_c|$ ), and the number of observations ($m$):

$$BIC = \ln(m) \times |\widehat{PA}_c| - 2 \times \hat{\mathcal{L}}. \tag{9}$$

When comparing two models, the model with the lowest BIC value is preferred. The quantity $\ln(m) \times |\widehat{PA}_c|$ penalizes model complexity, reducing the number of noisy weak connections.

For each observed neuron $c$, our objective is finding the parent set estimate $\widehat{PA}_c$ that minimizes the BIC, subject to a p-value restriction $\gamma$ which forces the selected regressors to have a tight confidence interval:

$$\begin{aligned} \{\widehat{PA}_c\} = \text{argmin}_{PA_c}\{BIC(\hat{\mathcal{L}}[\{PA_c\}]\}, \\ \text{s.t. } max(p_{H_0}(PA_c)) \leq \gamma. \end{aligned} \tag{10}$$

In the following section we will briefly explain how we approach this minimization.

## Selecting relevant regressors

The optimization problem presented in Eq. (10) can be stated concisely as the problem of finding the set of regressors $\widehat{PA}_c$ that yields the best BIC score (Eq. 9) subject to a p-value restriction. This is done to ensure that the regression model has good prediction capabilities and generalizes well to non observed data points.

The problem as stated in Eq. (10) is combinatorial in nature and cannot be directly optimized. There is a rich literature on model selection using various search strategies and evaluation criteria [22–28]. Usual model evaluation criteria include BIC and Akaike Information Criterion (AIC) [29] among others, while search algorithms include stochastic search variable selection [28], forward model selection, backward elimination, and stepwise methods in general, among others [23].

For this particular problem, we decided to use a greedy elastic-forward subset selection algorithm; an extensive overview of subset selection strategies can be found in [23]. In addition, we take several randomly selected subsets from the training dataset, each containing a fraction $\nu$ of available samples, and we evaluate the BIC performance and p-value restriction for regressor candidates across all bootstrapping subsets to find

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

candidates that are consistently relevant across subsets. In our experiments the fraction $\nu$ is set to 0.7. We show the performance of varying the $\nu$ parameter on simulated data in the Appendix.

The algorithm starts from $PA_c = \emptyset$, and iteratively includes regressors that improve the median BIC score across the random subsets, as well as the BIC score over the full dataset, while satisfying the p-value constraint. The algorithm is described in detail in the Appendix. The results section compares the performance of this algorithm against the LASSO method for variable selection [30].

Now that we have described the model and algorithm, we proceed to describe the active learning strategy.

## Experimental design - Active learning

Our goal is to develop a method to select, at any time, the optimal action (or network perturbation) that is expected to yield the maximum information about its currently computed connectivity. For the purposes of this paper, our action set will consist of selecting which set of visual stimuli will be presented next.

We are interested in gathering samples from network connections (edges) that show a promising improvement in the likelihood of the model but have not yet been added to it, we refer to these edges as candidate edges. That means we want to improve the parent set estimate $\widehat{PA}_c$, $c = 1, ..., n_c$, and edge weight estimates $w_{ic}$, $i = 1, ..., n_r$, $c = 1, ..., n_c$, as in equations (10) and (5) respectively. The samples are collected by presenting stimuli that directly or indirectly generate activations in the parent nodes of the promising candidate edges.

To address this, we will define a relevance score for each potential stimulus. This score will consider the effects of presenting one stimulus over random selection on each candidate edge, weighted by the potential log-likelihood improvement of adding those edges to the current network model. This in effect means that stimuli that directly trigger neurons associated with good candidate edges will be presented more often than other stimuli during the next intervention. The exact formulation is presented next.

### Defining a score for each stimuli

Define $\hat{W}^l$, $\hat{H}^l$ as the estimated adjacency and stimuli response matrices up to sample $m_l$, where $l$ is an iteration counter. Our objective is to obtain a probability distribution vector $P^{l+1} = [p_1^{l+1}, ..., p_s^{l+1}, ..., p_{n_s}^{l+1}]$ for presenting each stimuli $s = 1, ..., n_s$ at intervention $l + 1$.

Lets start by estimating the effect on every neuron $c = 1, ..., n_c$ of presenting each stimuli $s = 1, ..., n_s$ more frequently. For each stimuli $s$, we define a surrogate probability distribution vector associated with it: $\hat{P}_s = \{\hat{p_s}_j : j = 1, ..., n_s\}$, where stimuli $s$ has the highest occurrence probability. Given the previous observations, we want to predict the expected change in spiking rate of each neuron $c$ when preferentially applying stimuli $s$ using $\hat{P}_s$, as opposed to the baseline where all stimuli are presented equally: $\hat{P} = \{\frac{1}{n_s}\}$. Formally the probability vector that favors stimuli $s$ ($\hat{P}_s$) is defined as

$$\begin{aligned} \hat{P}_s &= \{\hat{p_s}_j : j = 1, ..., n_s\}, \\ \hat{p_s}_j &= (1 - \beta) \times \delta_{s-j} + \beta \times \frac{1}{n_s}, \end{aligned} \tag{11}$$

where $\hat{p_s}_j$ is the $j$-th element of the probability vector $\hat{P}_s$ and corresponds to the probability of presenting stimuli $j$ in the surrogate probability distribution vector

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

associated with $s$, $\beta$ is a smoothing constant that satisfies $0 \leq \beta \leq 1$ and controls the overall probability of other stimuli appearing, and $n_s$ is the number of available stimuli. We will define $\widetilde{IM}_{s,c}$ as the impact of stimuli $s$ on neuron $c$.

The impact of stimuli $s$ on neuron $c$ for iteration $l+1$ ($\widetilde{IM}_{s,c}[l+1]$) is presented in Eq. (12) next. This corresponds to the effect of using a stimuli distribution $\hat{P}_s$ on neuron $c$ when compared to using the baseline (uniform) distribution $\hat{P}$. The quantity $\widetilde{IM}_{s,c}[l+1]$ shows the impact of preferentially presenting stimuli $s$ on the spiking rate of neuron $c$ ($\lambda_c$) according to our previous $m_l$ observations,

$$\widetilde{IM}_{s,c}[l+1] = \frac{E[\lambda_c|\hat{W}^l, \hat{H}^l, \{S\}_{P=\hat{P}_s}]}{E[\lambda_c|\hat{W}^l, \hat{H}^l, \{S\}_{P=\hat{P}}]}. \tag{12}$$

Next we introduce the log-likelihood score of each of the output edges of neuron $c$ that did not satisfy the parameter selection criteria in Eq. (10) and therefore $c \notin \widehat{PA}_{c_i}$ for some neuron $c_i$. This is the log-likelihood difference (noted as $\hat{L}_{c,c_i}[m_l]$) between the network model up to sample $m_l$ (where $c \notin \widehat{PA}_{c_i}$) and a network model where $c$ is included as a parent of $c_i$:

$$\hat{L}_{c,c_i}[m_l] = -\hat{L}[\hat{X}_{c_i}(m_l)|\widehat{PA}_{c_i}(m_l)] + \hat{L}[\hat{X}_{c_i}(m_l)|(\widehat{PA}_{c_i} \cup c)(m_l)]. \tag{13}$$

Note that Eq. (13) is always non-positive, a highly negative value indicates a strong possibility of neuron $c$ influencing neuron $c_i$. By acquiring more samples from this interaction (samples where the candidate parent node is active), we can either disprove this notion, or gather enough evidence to add this edge into the regressor set (by satisfying the BIC and p-value criteria for adding an edge to the model).

We therefore define the score of stimuli $s$ associated with inter neuron edges $W$ as

$$\widehat{SC}_{s,W}[l+1] = \sum_{c \in C} [\widetilde{IM}_{s,c}[l+1] \times \sum_{c_i : c \notin \widehat{PA}_{c_i}(m_l)} \frac{\hat{L}_{c,c_i}[m_l]}{|\{c_i : c \notin \widehat{PA}_{c_i}(m_l)\}|}]. \tag{14}$$

We have so far assigned a score that considers the impact of preferentially applying a stimuli $s$ on the inter neuron connectivity matrix $W$. It is important to note that the set $\{c_i : c \notin \widehat{PA}_{c_i}(m_l)\}$ refers to the set of cells that up to sample $m_l$ were not included as children of cell $c$, and therefore are considered as candidate edges for the purpose of the score. In this score, the first summation considers the impact of stimuli $s$ over each cell $c$ multiplied by the second summation, which is the mean log-likelihood difference over all the cells that were not considered as children of neuron $c$ up to sample $m_l$.

In a similar fashion, we also need to consider the effect of prioritizing any given stimuli $s$ on the stimuli response matrix $H$. In this case, the impact of prioritizing stimuli $s$ over stimuli $s_i$ is the quotient of the $s_i$ entry of the stimuli probability distribution vector associated with $s$ ($\hat{P}_s$) over the baseline (uniform) probability distribution vector ($\hat{P}$),

$$\begin{aligned} \widetilde{IM}_{s,s_i}[l+1] &= \frac{\hat{p}_{s,s_i}}{\frac{1}{n_s}} \\ &= (1-\beta) \times n_s \times \delta_{s-s_i} + \beta. \end{aligned} \tag{15}$$

The log-likelihood score of the output edges of $s$ that did not satisfy the parameter selection criteria in Eq. (10) and the score of stimuli $s$ associated with the stimuli response matrix $H$ can be analogously defined:

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

$$\hat{L}_{s_i,c_i}[m_l] = -\hat{L}[\hat{X}_{c_i}(m_l)|\widehat{PA}_{c_i}(m_l)] + \hat{L}[\hat{X}_{c_i}(m_l)|(\widehat{PA}_{c_i} \cup s_i)(m_l)], \tag{16}$$

$$\widehat{SC}_{s,H}[l+1] = \sum_{s_i \in S} \left\{ \widetilde{IM}_{s,s_i}[l+1] \times \sum_{c_i : s_i \notin \widehat{PA}_{c_i}(m_l)} \frac{\hat{L}_{s_i,c_i}[m_l]}{|\{c_i : s_i \notin \widehat{PA}_{c_i}(m_l)\}|} \right\}. \tag{17}$$

Finally, the combined score given to stimuli $s$ is

$$\widehat{SC}_s[l+1] \quad = \widehat{SC}_{s,W}[l+1] + \widehat{SC}_{s,H}[l+1]. \tag{18}$$

At this point we have a score for each stimuli $s$ that is able to capture how informative this stimuli might be based on the impact it has on edges that are not included in the model so far. The next step is mapping these scores into a probability vector. For that we first compute the z-score of each stimuli; this is done as a normalization step of the score values, and allows the detection of outlying stimuli. The z-scores are then converted into a probability vector with the use of the well known softmax function. To avoid giving unnecessarily small or large probabilities to any given stimuli, we truncate the computed z-score into the $[-2, 2]$ range.

The formulation is as follows:

$$Z_s[l+1] \quad = \frac{\widehat{SC}_s[l+1] - mean(\{\widehat{SC}_s[l+1]\}_{s=1}^{n_s})}{std(\{\widehat{SC}_s[l+1]\}_{s=1}^{n_s})}, \tag{19}$$

$$Z_s^*[l+1] \quad = max(min(Z_s[l+1], 2), -2), \tag{20}$$

and the probability distribution vector for presenting each stimuli at intervention $l+1$ ends up being:

$$P^{l+1} = \left\{ \frac{\exp Z_s^*[l+1]}{\sum_{j=1}^{n_s} \exp Z_j^*[l+1]};\ s = 1, ..., n_s \right\}. \tag{21}$$

The use of the z-score as a normalization step allows the algorithm to dynamically pick up on the "relative quality" of the stimulation actions, and the truncation of the score provides a limit on how frequently or infrequently any given stimuli can be shown.

### Active learning

The active learning strategy consists of iteratively evaluating the current network model using Algorithm 1 in the Appendix, then using equations (18) and (21) to compute the stimuli distribution probabilities for the next time interval. The full algorithm is described in Algorithm 2 in the Appendix.

# Results

## Simulated data

In order to validate the method before applying it to real datasets, we generated a number of artificial datasets where the connectivity is known.

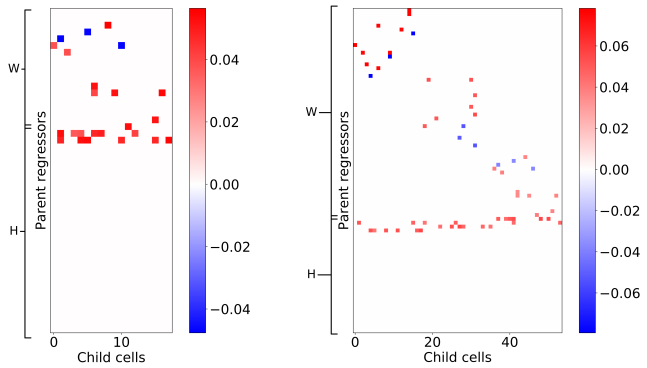
Network topology was simulated using the small-world Watts-Strogatz model [31]. This type of network architecture has been used to model functional cortical

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

connectivity in cats and macaques [32, 33], and has been theorized to be of use in understanding human functional connectivity [34].

We defined two separate networks SW1CL and SW3CL, network SW1CL is a single small-world cluster network with 18 neurons, while network SW3CL has three separate small-world clusters, each cluster has 18 neurons. All clusters have an average connectivity degree of 0.03. Edge weights for network SW1CL were drawn from a normal distribution $N(0.05, 0.005)$. Edge weights for the three clusters in network SW3CL were drawn from normal distributions $N(0.075, 0.005)$, $N(0.05, 0.005)$ and $N(0.035, 0.005)$ respectively. Thirty percent of inter-neuron edges were made inhibitory; this is consistent with the observed values on our real datasets.

These simulated networks were presented with 30 possible excitatory stimuli, most of which were designed to have no effect on the network. This was done to test that the active learning algorithm has the ability of navigating through confounders. Stimuli edge weights were drawn from the normal distribution $N(0.05, 0.005)$. Fig. 2 shows the connectivity matrix $W$ and stimuli response matrix $H$ for networks SW1CL and SW3CL. Boxcar influence functions as defined for Eq. (1) were set to $D_s^l = D_c^l = 5$ and $D_s^u = D_c^u = 2$ $(\mathbf{1}_{[t-2,t-5]}(t))$. These values were selected so that the average spiking rate of the simulated neurons were similar to the ones obtained in real data.




(a) Network SW1CL connectivity matrix (b) Network SW3CL connectivity matrix

**Fig 2.** Adjacency matrices for networks SW1CL and SW3CL. Red entries in the adjacency matrices denote an excitatory relation between the regressor and the child neuron, while blue entries denote inhibitory connections. The block diagonal structure present in the $W$ matrix for network SW3CL evidences the three cluster structure of the network. On both networks, we can observe the large number of stimuli that have no effect on the network.

In the following sections we will present two experiments. The first experiment will show the performance difference in regressor selection when using the proposed Algorithm 1 compared against the Lasso method [30]. The second experiment will compare the performance in regressor selection when stimuli are chosen according to active learning Algorithm 2 versus random stimuli selection.

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Algorithmic performance will be presented based on precision, recall, and $F_1$ metrics:

$$\begin{aligned} \text{precision} &= \frac{\sum_{c\in C}|\widehat{PA}_c \cap PA_c|}{\sum_{c\in C}|\widehat{PA}_c|}, \\ \text{recall} &= \frac{\sum_{c\in C}|\widehat{PA}_c \cap PA_c|}{\sum_{c\in C}|PA_c|}, \\ F_1 &= 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \end{aligned} \tag{22}$$

$\widehat{PA}_c$ is again the recovered set of regressors, and $PA_c$ is the true set of regressor edges for neuron $c$.

### Relevant regressors

We first checked the performance of the regressor selection methods on networks SW1CL and SW3CL. We compare the performance of the elastic-forward BIC selection Algorithm 1 with bootstrapping versus standard Lasso [30] regression using the pyglmnet implementation [35].

Stimuli were sampled uniformly with replacement, each stimuli was presented for 4 consecutive frames. Spiking trains for the simulations were sampled from a Poisson random process with a spiking rate corresponding to the ground truth model from Eq. (3).

The $l_1$ regularization parameter for the Lasso method was selected using an oracle to provide the best possible $F_1$ score. This method was selected for comparison because it is one of the most common approaches to variable selection. The modified log-likelihood function used for the Lasso method was:

$$\mathcal{L}_{\mathrm{l_1}}(\mathrm{X_c}|\{\hat{\mathrm{X}}\}_{\mathrm{C}'_c}, \{\hat{\mathrm{I}}\}_{\mathrm{S}'_c}, \mathrm{W_c}, \mathrm{H_c}) \propto \sum_{t=1}^{m}[X_c(t) log[\lambda_c(\eta_c(t))] - \lambda_c(\eta_c(t))] - l_1||W_c||_1 - l_1||H_c||_1. \tag{23}$$

Fig. 3 shows the results; 10 independent trials were used to provide confidence intervals for the metrics.

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

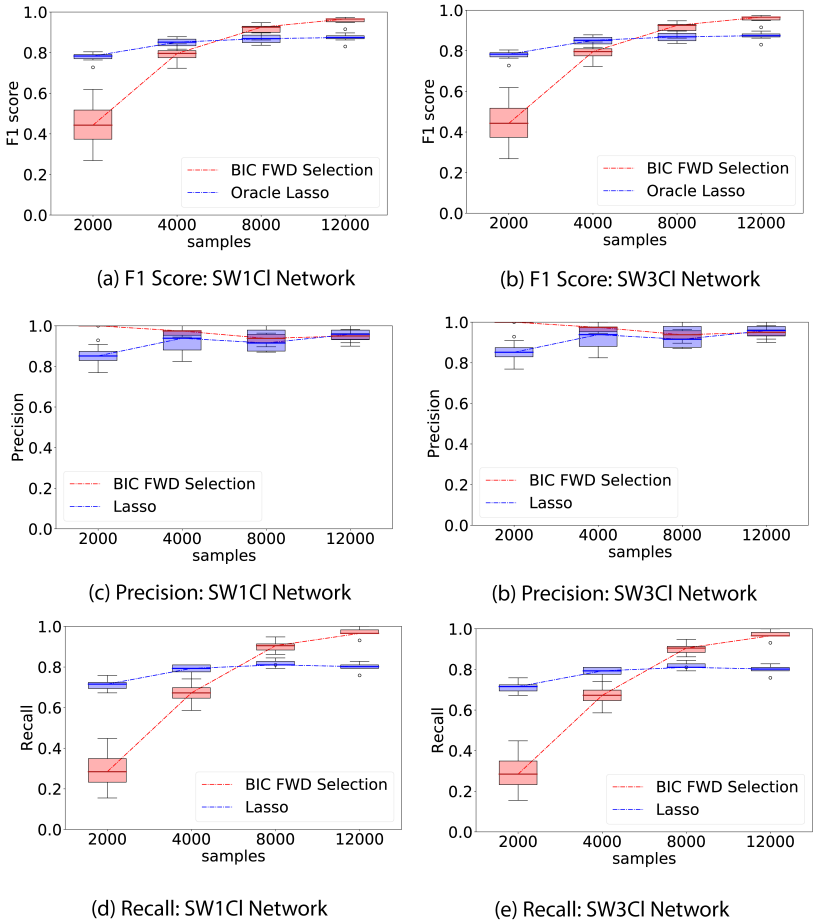



**Fig 3.** Whisker plot of performance indicators as a function of number of samples; elastic-forward BIC selection is shown in red, oracle lasso in blue. Whisker plot is obtained from 10 independent trials. The elastic-forward BIC selection outperforms oracle lasso for larger sample sizes. This performance improvement is more noticeable in the SW3CL network, where edge weights are more diverse

Fig. 3 shows that the elastic-forward BIC method described in Algorithm 1 outperforms lasso for larger sample sizes, even when the $l_1$ regularization parameter is selected using an oracle. The improvement is more noticeable for network SW3Cl which has diverse edge weights. In all tested cases, the precision metric in edge recovery was better for elastic-forward BIC subset selection.

### Active learning: Stimuli selection

We now evaluate the performance of the proposed active learning method, Algorithm 2. We compare it against uniformly sampling from all 30 possible stimuli.

Both strategies start from the same initial 500 samples, and each intervention adds an additional 500 samples. At the beginning of each intervention step, we compute the best network estimate so far using Algorithm 1, and show the performance in recovering the set of regressor edges $PA_c$ using the $F_1$, precision and recall metrics. At this stage, the active learning strategy described in Algorithm 2 recomputes the stimuli probability distribution to apply for the following 500 samples. Active learning parameter $\beta$ was set

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

to $\beta = 1/4$.

Fig. 4 compares the performance of uniform (Random) stimuli sampling and active learning (AL) sampling, while Fig. 5 shows the performance difference only on the entries of the stimuli response matrix $H$. Experiments were repeated 10 times to provide confidence intervals for the metrics.

Active learning outperforms random sampling by a large margin, the inter-quartile ranges for random sampling and active learning do not overlap over a significant sample count. As expected, both strategies converge for large sample sizes, but the process is sped-up by selecting the correct set of stimuli. The most noticeable performance difference is obtained when recovering the stimuli response edges $H$, since these are the edges we have direct influence on.

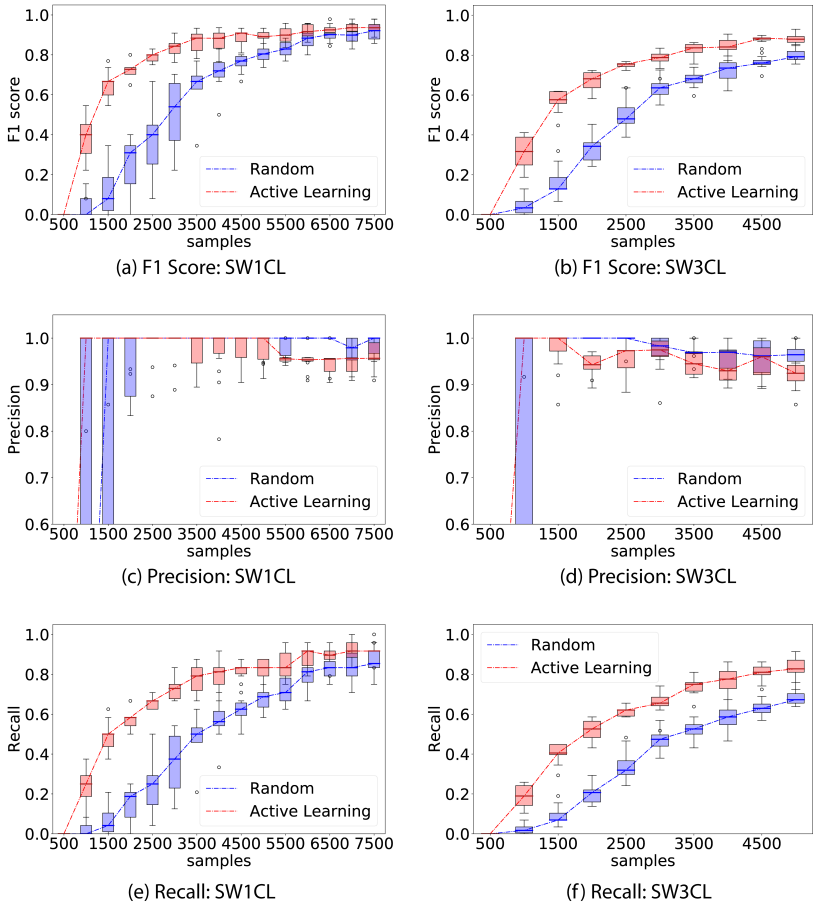



**Fig 4.** Comparison of performance between the proposed active learning (AL) method versus uniformly sampling (Random) from all stimuli. The experiment consisted of 500 sample interventions, with an initial 500 sample observation. Whisker plots are obtained from 10 independent trials. Left column show $F_1$, precision, and recall performance on network SW1CL and right column on network SW3CL.

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

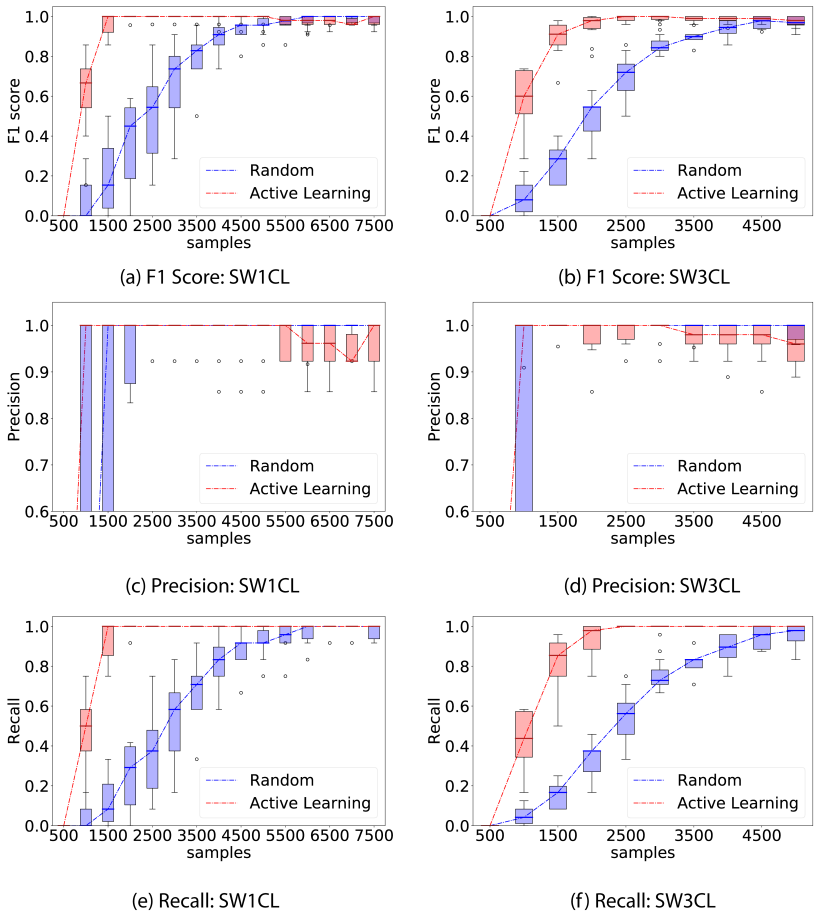



**Fig 5.** Comparison of performance between the proposed active learning (AL) method versus uniformly sampling (Random) from all stimuli only over the direct stimuli response matrices $H$. The experiment consisted of 500 sample interventions, with an initial 500 sample observation. Whisker plots are obtained from 10 independent trials. Left column shows $F_1$, precision, and recall performance on network SW1CL and right on network SW3CL.

Fig. 6 shows a visual representation of the edge discovering process over network SW1CL using active learning versus random sampling. Fig. 7 shows the difference between ground truth and the active learning and random sampling estimates as a function of interventions. Here we can clearly see that $H$ edges are quickly recovered using active learning, while $W$ edges show a slower improvement that is network dependent.

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

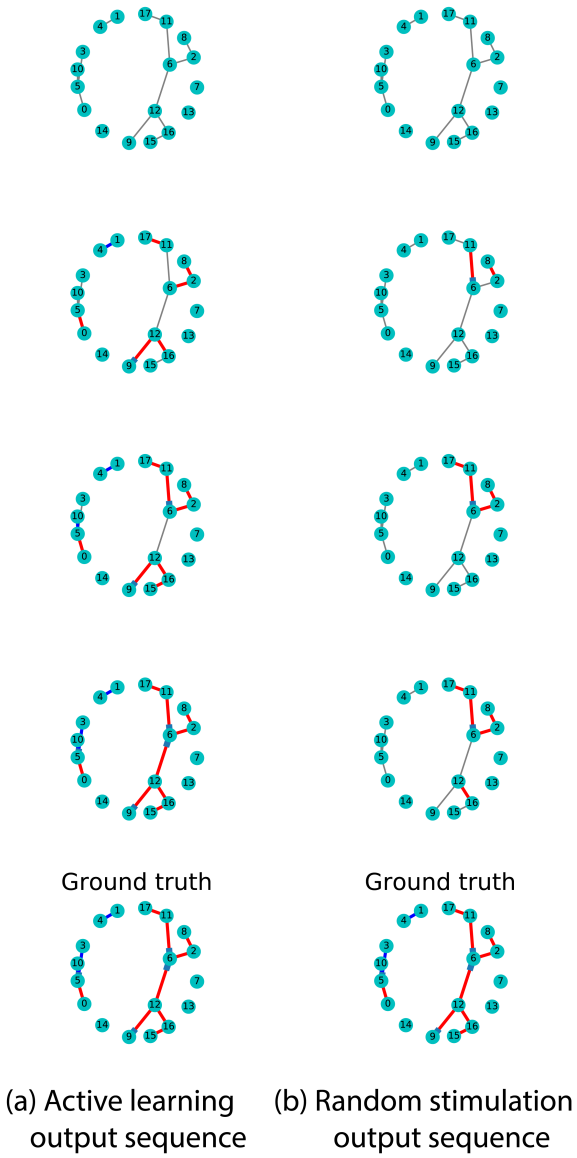



**Fig 6.** Comparison of the edge discovering process for network SW1CL using active learning versus random stimulation. Rows show inferred connections over a simulated cluster as a function of samples. Red and blue edges show correctly detected excitatory and inhibitory connections respectively, while grey edges show connections as not yet detected. Left column shows detected edges as a function of time when the active learning stimulation policy is used, right column shows the same cluster with a completely random stimulation policy. Rightmost cluster shows the ground truth. This example shows a clear advantage in edge recovery when using active learning compared to random stimulation.

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

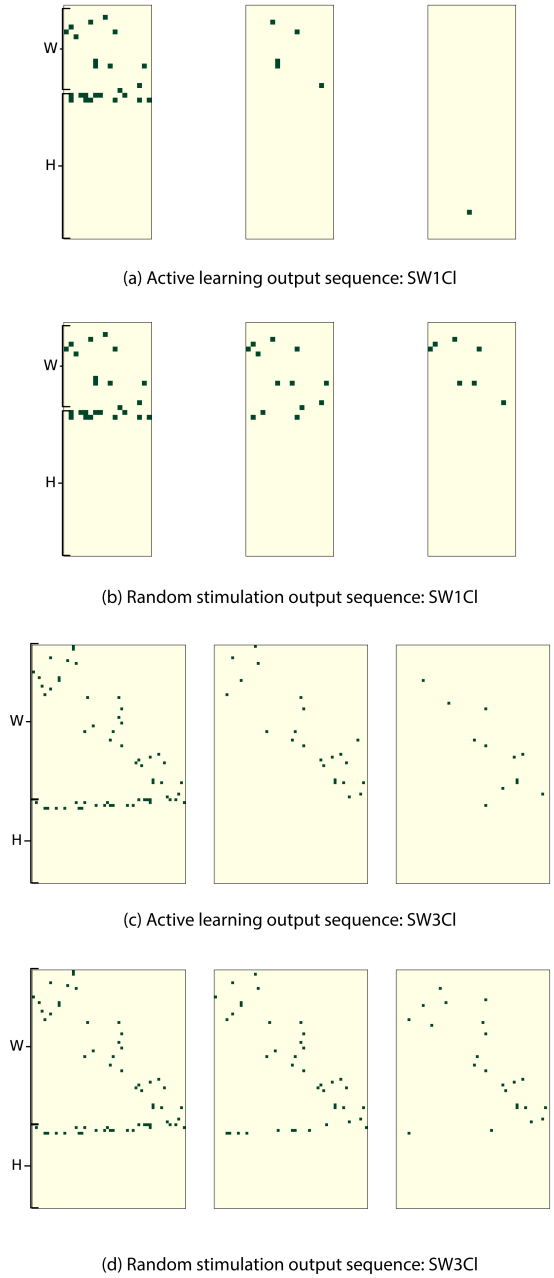



**Fig 7.** Comparison of the edge discovering process for networks SW1CL and SW3CL using active learning versus random stimulation. Rows show misclassified edges in the adjacency matrices $W$ and $H$ as a function of samples. Rows 1 and 3 show misclassified edges as a function of time when the active learning stimulation policy is used, while rows 2 and 4 show the same network probed with a random stimulation policy. The misclassified edge matrix under active learning quickly becomes sparse as the number of misclassifications goes to zero, random stimulation produces the same results but in a longer time frame.

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

## Real data

We worked with two datasets: lt3-000-002 and lt3-000-003, hereafter called datasets 1 and 2, containing a population of 57 and 63 neurons respectively. The presented visual stimuli consisted of sinusoidal gratings defined using Hartley basis functions of the form:

$$H(k_x, k_y) = A \times \text{cas}([k_x x + k_y y]\frac{2\pi}{M}), \tag{24}$$

where the function $cas(x)$ is $cos(x) + sin(x)$, $A = \pm 1$, $(x, y)$ are the pixel coordinates in the monitor, $M$ is the image size, and $k_x$, $k_y = \{-12, ..., 12\}$ are the frequency components. Parameters $A$, $k_x$, $k_y$ were uniformly sampled from all possible values, and each stimuli persisted for 4 frames.

To prevent excessive data fragmentation, the stimuli basis $(A, k_x, k_y)$ was encoded into an $(r, \phi)$ pair using Eq.(25) with the $r$ and $\phi$ parameters discretized into 7 values each,

$$\begin{aligned} r &= \sqrt{k_x^2 + k_y^2}, \\ \phi &= \text{atan}(k_y, k_x). \end{aligned} \tag{25}$$

For both spike and stimuli regressors, we used $\mathbf{1}_{[t-2,t-7]}(t)$ as the influence function as defined for Eq. (1) ($D_s^l = D_c^l = 7$ and $D_s^u = D_c^u = 2$).

$7,000$ samples of each dataset were used for training, and 2000 samples were reserved for model validation. Models were computed under two conditions, the first model used all available regressors (full model), while the second model was restricted to self regression coefficients and direct stimuli to neuron connectivity (AR model).

We first show the predictive power of both models when evaluated over the validation samples, that is to say, we evaluate the log likelihood (Eq. (4)) of both models over the test samples.

We then utilize the model that we obtained from all samples in the dataset as a template for simulations ("ground truth"), and compare the simulated performance between the proposed active learning Algorithm 2 and random sampling.

We go on to show that observed neurons tend to respond more to low frequency stimuli. We then show the recovered adjacency matrix (networks) for both datasets, and spiking trains time series for neurons belonging to the largest cliques in the network.

### Recovered models

We now show the recovered connectivity matrices $W$ and stimuli response matrices $H$ for both datasets for the full model and the AR model.

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

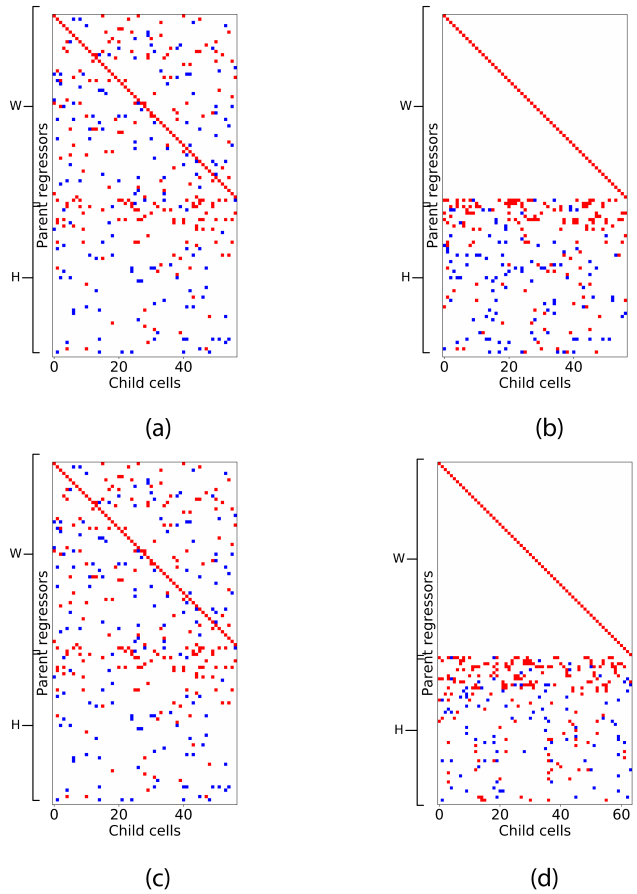


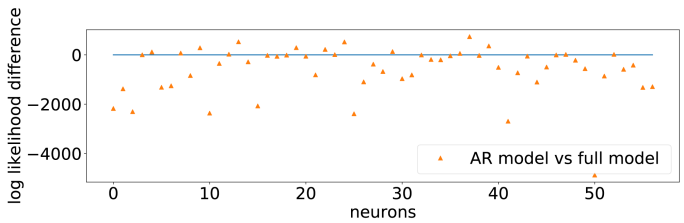

**Fig 8.** Recovered adjacency matrices for datasets 1 and 2. Top and bottom rows show the recovered adjacency matrices for datasets 1 and 2 respectively. Columns from left to right show the full model and the AR model respectively. Inhibitory connections are shown in blue, and excitatory connections are shown in red. We can observe that self regression coefficients are always added to the model. We also observe that the overall sparsity of the recovered network is consistent across datasets , and that the first few rows of the $H$ matrices show a heavy concentration of excitatory connections. These rows correspond to low spatial frequency ($r$) values in the Hartley basis functions (Eq.( 24)).

The recovered inter-neuron connectivity matrix $W$ for both datasets was 93% sparse. Additionally, 64% and 73% of recovered functional edges of this matrix were excitatory for datasets 1 and 2 respectively. Both datasets also show a large number of stimuli connections in the $H$ matrix corresponding to low spatial frequency stimulation values ($r$), and consistently selected the self regression coefficient as an important regressor. This was consistent for both regression models.

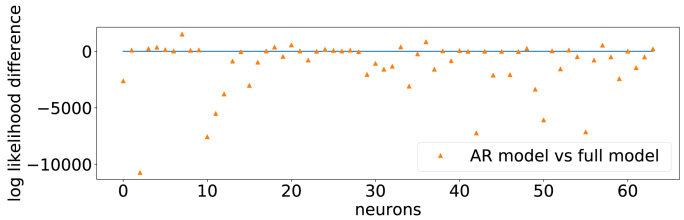
## Out of sample prediction power

To test the prediction power of the regressor model, we first evaluate the log likelihood of the full model and AR model over the test samples. We compute $\eta_c$ and $\lambda_c$ for each neuron on the test samples using the observed spike trains and visual stimuli as presented in equations (1) and (2). We then evaluate the log likelihood according to Eq. (4). The results are shown in Fig. 9.

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.




(a) Dataset 1




(b) Dataset 2

**Fig 9.** Log likelihood difference of full model and AR model over the test samples. The graph shows that, on average, spiking rate predictions are better on the full model than on the auto-regressive model. This grounds the idea of neuron to neuron interaction as being predictive of neuron behaviour.

Fig. 9 shows that, for most neurons, the full model generalizes well to out of sample (unobserved) data, when compared to the AR model. This shows that inter-neuron connectivity is predictive of spiking rates.

Long range prediction is also possible using the recovered model parameters $(\hat{W}, \hat{H}, \hat{b})$ and the visual stimuli sequence to be presented $(\{I_k(t)\})$. Instead of using the observed past spike trains $X_j(t)$ as regressors, we use the expected spiking rate $\lambda_j(t)$. This experiment iteratively computes the expected spiking rate for each neuron in the network using a fully observed external stimulation sequence and the past computed expected spiking rate. It is important to note that here we are computing the entire behaviour of the system given a stimuli sequence.

Formally, we define the long range spiking rate as

$$\lambda_c^{\mathrm{lr}}(t) = E[X_c \mid \{\hat{\lambda}_j^{\mathrm{lr}}(t)\},\, \{\hat{I}_k(t)\},\, \hat{W},\, \hat{H},\, \hat{b}], \tag{26}$$

where $\lambda_c^{\mathrm{lr}}(t)$ is the expected spiking rate at time $t$ for neuron $c$, $\{\hat{\lambda}_j^{\mathrm{lr}}(t)\} = \{\sum_{t-D_c^u}^{t-D_c^l} \lambda_j^{\mathrm{lr}}(t)\}$ is the set of past expected spiking rates as seen through the boxcar influence function, and $\{\hat{I}_k(t)\} = \{\sum_{t-D_s^u}^{t-D_s^l} I_k(t)\}$ is the applied visual stimuli as seen through the boxcar influence function.

Here we compare the difference between the log likelihood when simulating the 2000 test samples with the stimuli that were presented versus a simulation of the system under a random equally likely stimuli selection.

The log likelihood of the sequences is computed against the observed spike trains for each neuron, experiments are repeated over 20 trials to provide error estimates. Results are shown in Fig. 10.

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

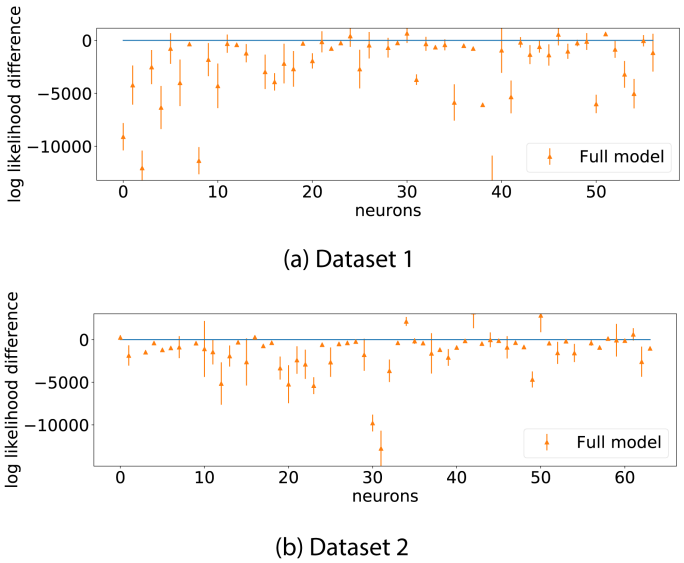



**Fig 10.** Difference in log likelihood of forecasted sequences, real stimulation sequence versus randomized stimulation sequences. Error bars represent one standard deviation over 20 trials.

Iterated simulations like these integrate the influence of stimuli to neuron connectivity and neuron to neuron connectivity. The large majority of neurons for both datasets showed a likelihood increase when presented with the true stimulation sequence, showing that the recovered models really capture interactions between regressors and spiking rates.

### Active learning on real data

In lieu of validating the proposed active learning framework on live animals, we utilize the recovered full model networks from Fig. 8 as ground truth. This is the exact same process as presented in the active learning results on simulated data, but using the recovered inter-neuron and stimuli response matrices ($W$, $H$) for real data as the ground truth model. We compare the active learning technique against uniformly sampling from all 49 possible stimuli.

Both strategies start from the same initial $1,000$ samples, and each intervention adds an additional $1,000$ samples. As a reminder, the ground truth network was recovered from $9,000$ samples. At the beginning of each intervention step, we compute the best network estimate so far using Algorithm 1, and show the performance in recovering the set of regressor edges $PA_c$ using the $F_1$, precision and recall metrics as defined in Eq. (22). At this stage, the active learning strategy recomputes the stimuli probability distribution to apply for the following samples. These results are shown in Fig. 11. Additionally, Fig. 12 shows the stimuli probability distribution ($P^{l+1}$) obtained from Algorithm 1, averaged over all realizations.

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

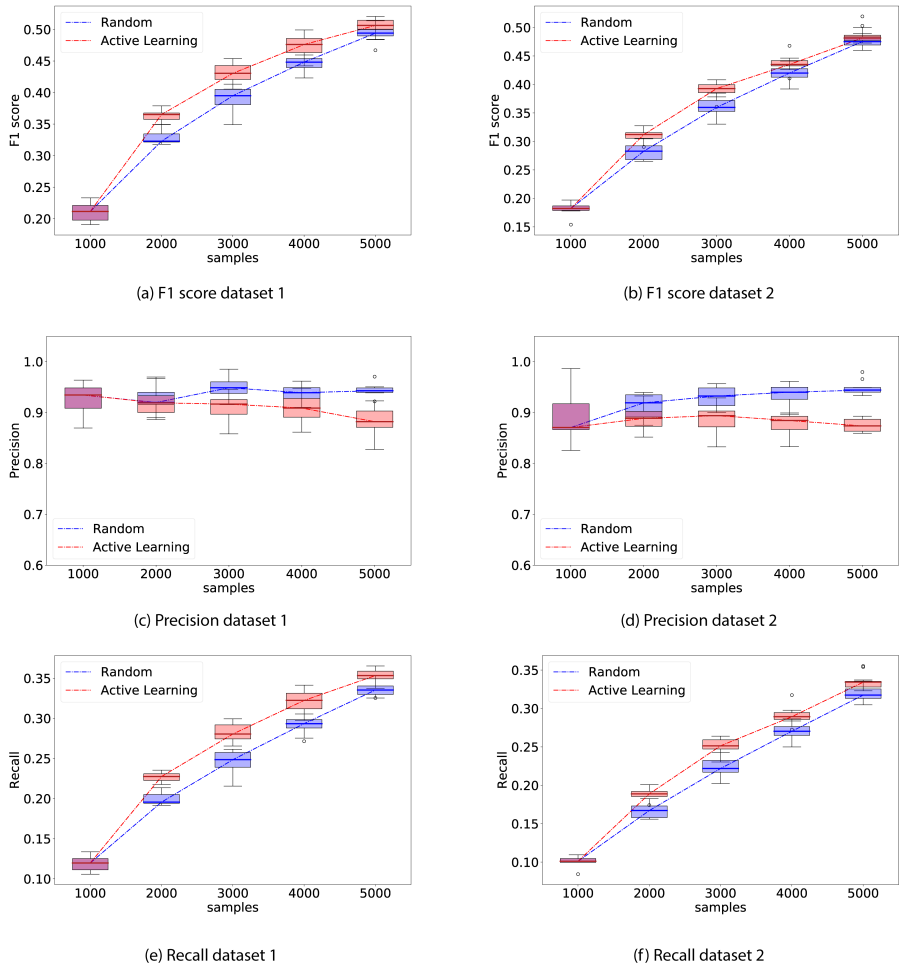
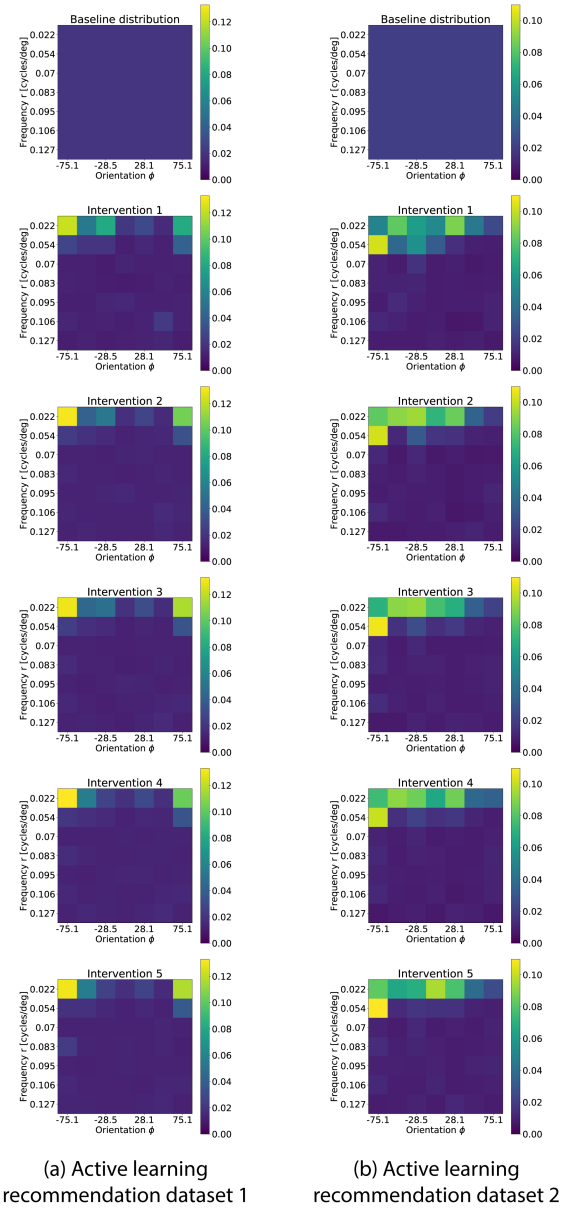



**Fig 11.** Comparison of performance between the proposed active learning (AL) method versus uniformly sampling from all stimuli. The experiment consisted of $1,000$ sample interventions, with an initial $1,000$ sample observation. Whisker plot is obtained from 10 independent trials. Left column shows $F_1$, precision, and recall performance on network recovered from dataset 1 and right column shows $F_1$, precision, and recall performance on network dataset 2.

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.




(a) Active learning recommendation dataset 1

(b) Active learning recommendation dataset 2

**Fig 12.** Distribution of recommended stimuli ($P^{l+1}$) averaged across all realizations as a function of the number of interventions. Initially, the distribution is uniform (top). The experiment consisted of $1,000$ sample interventions, with an initial $1,000$ sample observation. Left column shows the stimuli probability distribution history for dataset 1, while right column shows the distribution for dataset 2.

The $F_1$ score of the active learning experiment is consistently better than random stimuli selection. While the performance gain is not large for these networks, the result spreads are tight and consistent; there is thereby no reason for not using the active learning strategy over random stimulation. We also observe that the AL algorithm preferentially presents low frequency stimuli, even though no explicit variable in the AL algorithm distinguishes between low and high frequency stimuli. In the following section we will show this is a reasonable result, since we found that neurons in these datasets tend to respond considerably more to low frequency stimuli.

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

## Network analysis

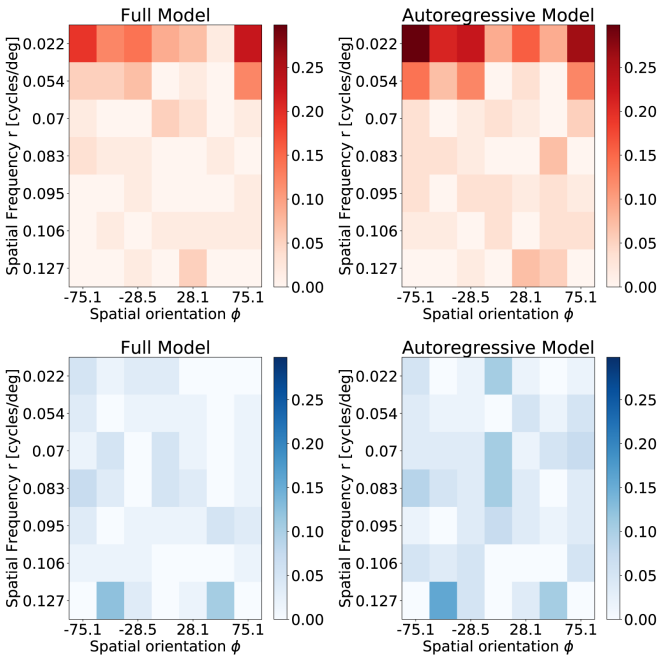
Now that we showed the performance of the proposed methods we will present some observations over the recovered biological models.

We first analyze the obtained input response matrices $H$ for both datasets. Fig. 13 shows the percentage of neurons in each dataset that directly respond to each stimulation pattern. We can conclude that low frequency stimulation patterns have an out-sized proportion of directly responding neurons when compared to higher frequency patterns. Moreover, when we view Fig. 13 in conjunction with Fig. 8 we can additionally conclude that low frequency stimulation patterns have a net excitatory effect on the observed neurons, while mid and high frequency patterns have a more balanced overall effect.

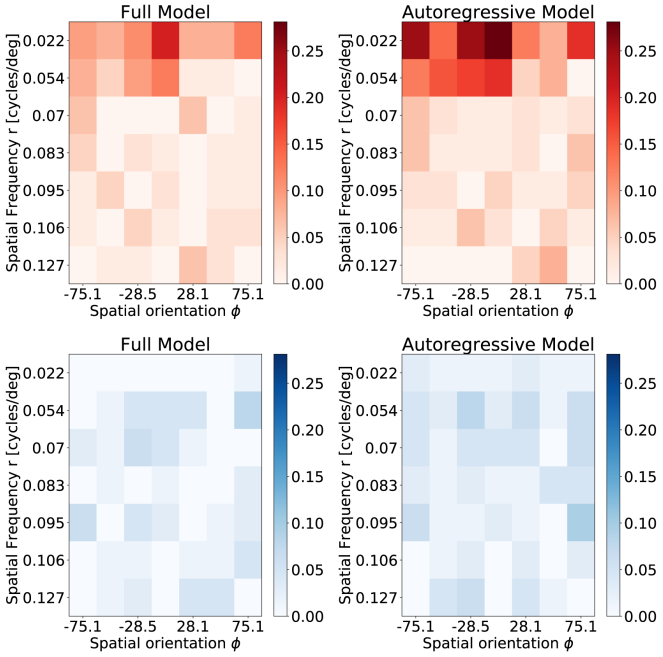
For each dataset, we then examine the adjacency matrices $W$ and extract two of the largest cliques in the network. Fig. 14 shows these cliques, and the spike trains of all neurons in the clique.

We additionally count the occurrence rate of motif triplets in the recovered networks, and do a simple hypothesis test to check if these motifs could have arisen from a small-world network topology, Fig. 15 show these results. The high p-values observed in Fig. 15 show that the recovered network motifs are at least compatible with the small-world topology hypothesis. We also compared the distribution of number of children and parents per cell between the inferred networks and a small-world network ensemble. In Fig. 16 we can see that the inferred networks are compatible with a small-world topology in terms of number children and parents per neuron. Finally, we count the percentage of cells involved in multiple cliques, and further show the percentage of cells involved in multiple cliques of a set size. The percentage counts also appear to be within the expected counts of a small-world network, this is shown in Fig. 17. This is in accordance with other observations like the ones in [32–34].

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.




(a) Dataset 1




(b) Dataset 2

**Fig 13.** Percentage of neurons that have an excitatory or inhibitory response to each possible visual stimuli. Visual stimuli are represented in matrix form, where rows represent spatial frequency $r$ and columns spatial orientation $\phi$. The value for each entry in the matrix is the percentage of neurons in the datasets that show an excitatory (red) or inhibitory (blue) response to that visual stimuli. This visualization shows that both datasets show a large number of directly responding neurons for low spatial frequency visual stimuli.

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

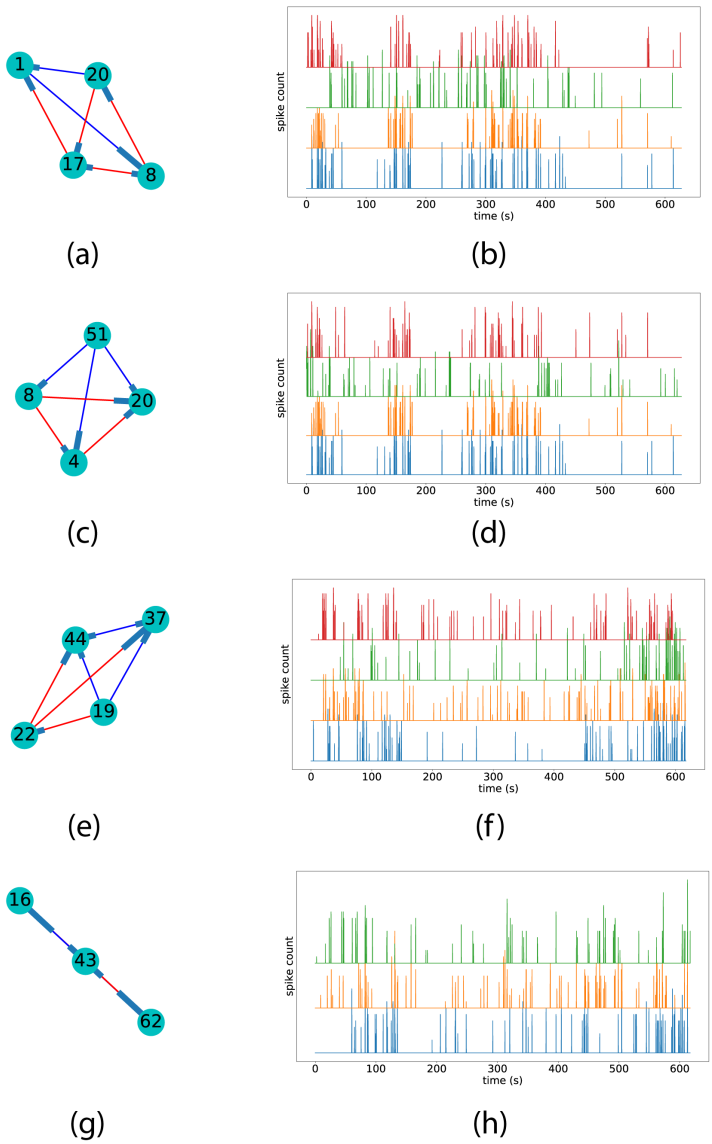



**Fig 14.** Figures a), c), e), and g) show the excitatory (red) and inhibitory (blue) edges detected for neuron cliques in datasets 2 (a) and c)) and 3 (e) and g)). Figures b), d), f), and h) show the spike time series of the neurons in the clique. The nodes are numbered according to the corresponding neuron index. We can visually see that spike trains from neurons in the selected cliques show similar spiking behaviour throughout the experiment.

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

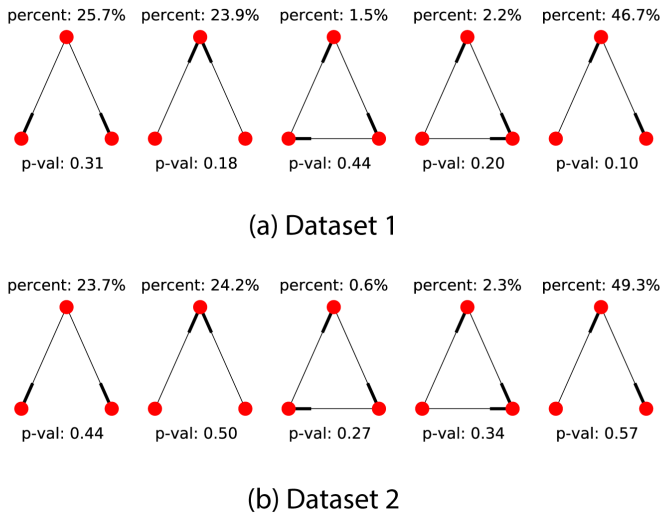



**Fig 15.** We count the occurrences of motif triplets for both datasets (we ignore edge weight and sign) by enumerating all neuron triplet combinations in the recovered networks and checking for graph isomorphism against all 5 motif triplet types. Top and bottom rows show results for datasets 1 and 2 respectively. We compare the obtained motif counts against a base model of small-world network topology and show the obtained p-values. These p-values are obtained by computation of the mean and standard deviation of each motif type in a small-world network with the same node count and edge density. The relatively large p-values obtained show that the small-world model is a good fit for the recovered network topology.

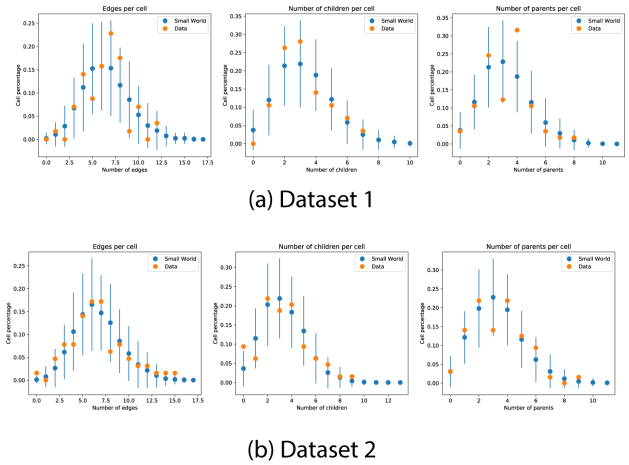
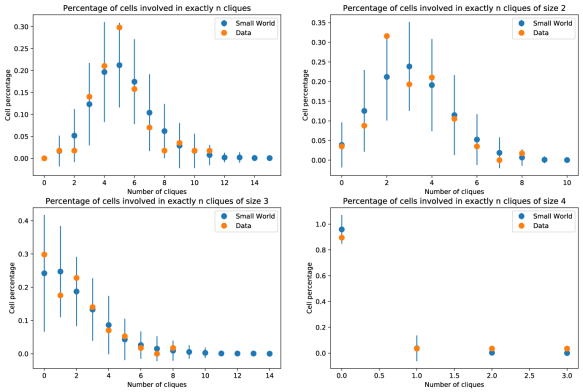
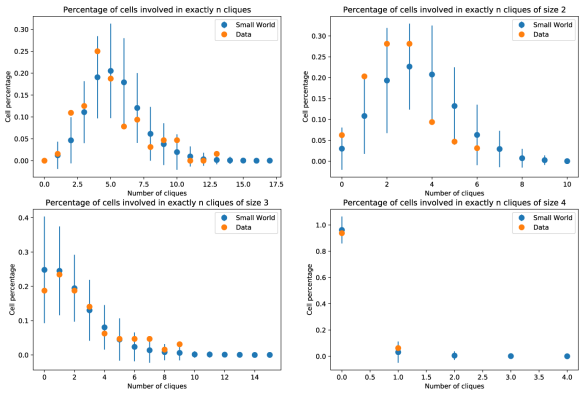



**Fig 16.** We compare the edge density distribution of the recovered inter neuron connectivity matrices in both datasets. Edge counts shown from left to right are all edges (number of neurons connected to node, either as parent or child), outbound edges (number of child nodes), and inbound nodes (number of parent nodes). The edge counts are compared against a base model of small-world network topology, error bars denote two standard deviations obtained from simulation of small-world networks with the same number of nodes and connectivity degree.

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.




(a) Dataset 1




(b) Dataset 2

**Fig 17.** We count the percentage of cells participating in multiple cliques. The counts are compared against a base model of small-world network topology, error bars denote two standard deviations obtained from simulation of small-world networks with the same number of nodes and connectivity degree.

# Discussion

There is a large body of work done in reverse-engineering neural circuits from data across different modalities [36, 37]. Some of these methods learn a directed graph, which provides a compact and interpretable way of encoding Granger causal [38] relations between neurons and covariates. Several works have been published on the use of Gaussian Bayesian Networks to learn connectivity strengths from data [39, 40]. For cases where the framework is not applicable, as in the case of spike train time series, Generalized Linear Models (GLMs) have been successfully used [41–48].

Parameters for these models are most commonly estimated using Maximum Likelihood Estimation or Maximum a Posteriori estimation [42–48].The parameters obtained from the GLM models can be interpreted as a directed graph capturing

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

dependencies between variables of interest, or nodes (neuron spiking rates and visual stimuli). The presence of an edge in this graph represents a directed influence from one node to another, and the weight of the edge represents the magnitude of that influence. The graph can be made sparse with the use of subset selection, where only a limited number of edges are assigned non-zero weights (no influence). Subset selection can be performed using deviance tests [49] or by the use of priors [46].

In parallel, there is a corresponding push for actively estimating the best stimuli subset for network inference, with variants based on mutual information and Gaussian approximations of MAP parameters [42, 44, 45, 47, 50, 51]

Methods like the one proposed in [50] use of mutual information for intervention selection, but rely on a specific Gaussian Bayesian Network framework, making them unsuitable for count data. Methods like [52] generalize D-Optimal factorial design for GLMs and to multi-level regressor covariates, but require full control over the regressor covariates.

The proposed variable selection algorithm is focused on subset selection of parent regressors. It is posed as an optimization problem where the objective is to find a set of regressors that minimize the BIC score, subject to a confidence interval restriction. Using BIC as the score to optimize fosters prediction improvement while penalizing model complexity. The use of p-value as a restriction criteria ensures that the regressors in the model are highly significant. A local minimal set of regressors is constructively obtained following a simple rule-set.

The variable selection method was tested on simulated data and compared against oracle Lasso. It performed worse than oracle lasso for very small sample sizes, but otherwise proved to be better on the $F_1$ and recall metrics. On settings where no ground truth is available, Lasso would require some other sub-optimal method for parameter tuning. The p-value restriction parameter present in the elastic-forward model selection algorithm is easily interpretable as the desireable confidence level on the regressor parameters, this makes the parameter easy to set beforehand for any experiment.

The proposed active learning algorithm follows a simple design philosophy, it looks for promising edges not added into the model so far, and increases the appearance frequency of stimuli that drive up the spiking rate of the parent nodes of these edges. To achieve this, it defines a score for each possible stimuli based on the previous learned model, it takes into account the spiking rate difference (impact) of presenting one stimulus more frequently than the others on every parent node in the system, and weighs it by the potential log-likelihood improvement of adding every edge associated with this parent node to the model. The log-likelihood value of an edge is tightly related with the BIC score the elastic-forward model selection attempts to optimize.

Active learning proved to be faster than random stimulation in recovering edges on the simulated networks, this was especially true for edges whose spiking rate could be greatly affected by changing the stimuli distribution (first order connections). On the simulations from networks recovered from real data, the performance of active learning was consistently better than random stimulation, and the measured metrics had a tighter spread. There is therefore no reason not to use active learning during data acquisition.

We measured three basic properties of the recovered inter-neuron connectivity networks from on real data: edge distribution, motif type distribution, and number of cliques per neuron. Counts were compared to simulations of small-world networks with an identical number of nodes and connectivity degree. All three properties measured fell well within the expected values for these types of networks, pointing at a small-world-like structure in the recovered inter-neuron connectivity networks.

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

# Conclusion

In this paper we propose a simple framework for actively learning network connectivity for GLM models by selecting external forcing actions. The algorithm has the advantage of making relatively few assumptions on the exact distribution of the model, and the amount of control the experimenter has over the regressor covariates.

The use of a greedy regressor selector using BIC and Wald testing allows for an easy identification of edges that seem beneficial for the model, but do not yet have a sufficient number of interaction samples to be included in the model. By utilizing external triggers to these interactions, the algorithm prioritizes interventions that provide information over uncertain edges.

The greedy regressor selector outperforms the oracle Lasso in identifying the proper regressor subsets in simulations for non-small sample sizes, even when accounting for the oracle selection of the $l_1$ prior.

Even on the recovered real datasets, the use of the active learning algorithm proved to be beneficial as well. The algorithm is very quick at recovering directly connected edges, making its application in conjunction with optogenetics an interesting proposition.

Finally, we note the method is not restricted to one modality or domain, it can be applied in any situation where there is a family of possible actions available to probe the activity of a network, in both biological and artificial systems.

We note that for all measured properties, the recovered network structure on the real datasets was consistent with a small-world topology.

# Appendix

## Derivation of the observed Fisher information

Given a neuron $c$ and the MLE estimators $\hat{\theta} = \{\hat{W}_c, \hat{H}_c, \hat{b}_c\}$, the element $k, j$ of the observed Fisher information matrix $\hat{\mathcal{I}}_c(\hat{\theta})$ for the given model can be expressed as

$$[\hat{\mathcal{I}}_c(\hat{\theta})]_{k,j} = -\sum_{i=1}^{m}[-\frac{y_c(t)}{\lambda_c(t)^2}[\frac{\partial\lambda_c(t)}{\partial\eta_c}]^2 R_k(t)R_j(t) + [\frac{y_c(t)}{\lambda_c(t)} - 1]\frac{\partial^2\lambda_c(t)}{\partial\eta_c^2}R_k(t)R_j(t)], \quad (27)$$

where $R$ is the concatenation of all the considered regressors (neurons and stimuli),

$$R(t) = [\hat{X}(t), \hat{I}(t), 1] \in \mathbf{R}^{m\times[n_s+n_c+1]}, \quad (28)$$

and the first and second derivatives of $\lambda_c(t)$ are,

$$\frac{\partial\lambda_c(t)}{\partial\eta_c} = \frac{exp[k \times \eta_c(t)]}{1 + exp[k \times \eta_c(t)]}, \quad (29)$$

$$\frac{\partial^2\lambda_c(t)}{\partial\eta_c^2} = \frac{k \times exp[k \times \eta_c(t)]}{(1 + exp[k \times \eta_c(t)])^2}. \quad (30)$$

## Regressor selection: fraction parameter $\nu$

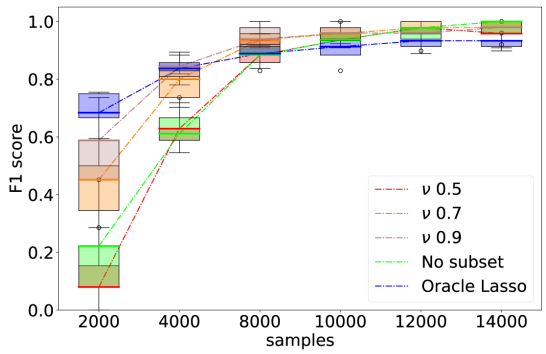
We evaluate the effect of varying the sample fraction $\nu$ used for the random subset selection step of the elastic-forward model selection method. The $\nu$ parameter is tested

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

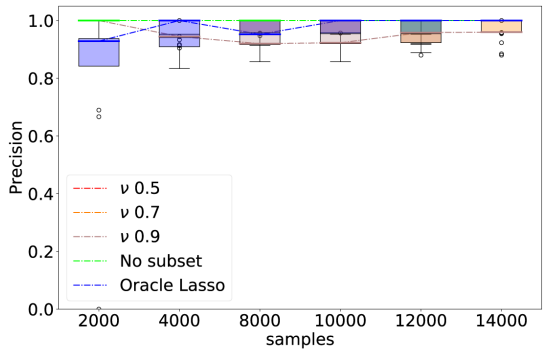
in the 0.5 to 0.9 ranges. This is compared against the baseline oracle lasso method and model selection where no random subset selection is performed (no-subset).

All methods are evaluated on samples drawn from simulated network SW1CL using the $F_1$, precision, and recall performance metrics. Results are shown in Fig. 18.

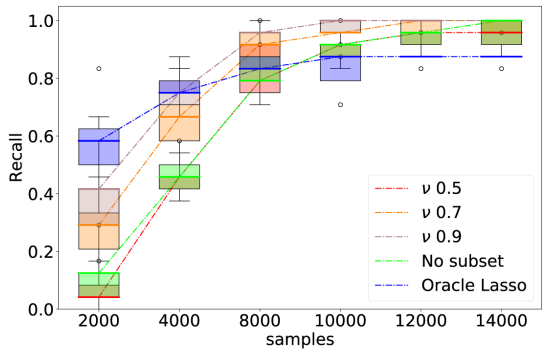
bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.




(a) F1 Score




(b) Precision




(c) Recall

**Fig 18.** Whisker plot of performance indicators as a function of number of samples; elastic-forward BIC selection is compared using several $\nu$ parameters against both oracle Lasso and the special case where the whole dataset is used at once, without splitting into random subsets (no-subset). The plots show relatively little difference between the various $\nu$ parameters, but the use of the $\nu$ parameter is better performing overall to both oracle lasso and no-subset model selection

We can observe from Fig. 18 that all $\nu$ parameters perform similarly for all but the smallest sample sizes. Additionally, the use of no-subset model selection had a large performance decrease for small sample sizes when compared to all $\nu$ parameters, but

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

similar performance on larger sample sizes.

## Table of defined variables

| Variable symbol | Variable Definition |
|---|---|
| $m$ | number of acquired samples |
| $n_c$ | number of neurons |
| $n_r$ | number of regressors |
| $X = \{X_i\}_{i=1,...,n_c}\ X_i \in \mathbf{R}^m$ | spike time series of target neurons |
| $R = \{R_i\}_{i=1,...,n_r}\ R_i \in \mathbf{R}^m$ | regressor time series |
| $b_c$ | bias of target neuron $c$ |
| $w_{ic}$ | influence weight of regressor $i$ on neuron $c$ |
| $PA_c$ | set of parent regressors of neuron $c$ |
| $n_s$ | number of external stimuli |
| $C$ | set of $n_c$ neurons |
| $S$ | set of $n_s$ external stimuli |
| $I = \{I_i\}_{i=1,...,n_s}\ I_i \in \mathbf{R}^m$ | external stimuli time series |
| $\hat{X} = \{\hat{X}_i\}_{i=1,...,n_c}$ | past neuron activity |
| $\hat{I} = \{\hat{I}_i\}_{i=1,...,n_s}$ | past stimuli activity |
| $D_s^u,\ D_s^l$ | upper and lower bounds of boxcar influence function for stimuli |
| $D_c^u,\ D_c^l$ | upper and lower bounds of boxcar influence function for neurons |
| $C_c'$ | set of parent neurons of neuron $c$ |
| $S_c'$ | set of parent stimuli of neuron $c$ |
| $W_c \in \mathbf{R}^{\lvert C_c' \rvert}$ | edge weights between parent neurons of $c$ and neuron $c$ |
| $H_c \in \mathbf{R}^{\lvert S_c' \rvert}$ | edge weights between parent stimuli of $c$ and neuron $c$ |
| $W = \{w_{ij} : i, j = 1, ..., n_c\}$ | adjacency matrix |
| $H = \{h_{ij} : i = 1, ..., n_s,\ j = 1, ..., n_c\}$ | stimuli response matrix |
| $\lambda_c$ | instantaneous Poisson spiking rate of neuron $c$ |
| $\kappa$ | spiking rate non-linearity calibration constant |
| $\{\hat{X}\}_{C_c'} = \{\hat{X}_j\}_{j \in C_c'}$ | past neuron activity of parent neurons of $c$ |
| $\{\hat{I}\}_{S_c'} = \{\hat{I}_j\}_{j \in S_c'}$ | past stimuli activity of parent stimuli of $c$ |
| $\hat{\theta} = \{\hat{W}_c, \hat{H}_c, \hat{b}_c\}$ | MLE estimate of model parameters of neuron $c$ |
| $\hat{W},\ \hat{H}$ | estimators of adjacency matrix $W$ and stimuli response matrix $H$ |
| $\widehat{PA}_c$ | estimated parent set of neuron $c$ |
| $\gamma$ | p-value restriction |
| $l$ | active learning iteration counter |
| $m_l$ | number of samples acquired up to active learning iteration $l$ |
| $\hat{W}^l,\ \hat{H}^l$ | estimators of adjacency matrix $W$ and stimuli response matrix $H$ up to iteration $l$ |
| $P^{l+1}$ | probability distribution vector for presenting each stimuli $S$ at intervention $l+1$ |
| $\hat{P}_s$ | surrogate stimuli probability distribution vector, where stimuli $s$ is presented more frequently |
| $\hat{P}$ | uniform stimuli probability distribution vector |
| $\{S\}_{P=\hat{P}_s}$ | sequence of stimuli drawn from distribution $\hat{P}_s$ |
| $\widehat{IM}_{s,c}$ | relative impact on spiking rate of neuron $c$ when using stimuli probability distribution $\hat{P}_s$ |
| $\hat{L}_{c,c_i}$ | log likelihood difference in neuron $c$ between model with and without regressor neuron $c_i$ |
| $\widehat{IM}_{s,s_i}$ | relative impact on appearance rate of stimuli $s$ when using stimuli probability distribution $\hat{P}_s$ |
| $\hat{L}_{c,s_i}$ | log likelihood difference in neuron $c$ between model with and without regressor stimuli $s_i$ |
| $\widehat{SC}_{s,W}[l+1]$ | score of stimuli $s$ associated with inter neuron matrix $W$ |
| $\widehat{SC}_{s,H}[l+1]$ | score of stimuli $s$ associated with stimuli response matrix $H$ |
| $\widehat{SC}_s[l+1]$ | full score of stimuli $s$ |
| $\beta$ | active learning smoothing constant |

**Table 1.** Summary of defined variables

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

# Algorithms

**Algorithm 1** Elastic-forward selection

**Require:** $X_c,\ \hat{R}_r,\ k,\ \gamma$

▷ $X_c$ sequence of observations of neuron $c$
▷ $\hat{R}_r$ is the full set of regressor observations
▷ $k$ maximum number of regressors to add per step
▷ $\gamma$ minimum allowed regressor p-value

$r' = \{\}$ ▷ Initialize set of active regressors to empty set (only bias is included).
**repeat**

$\hat{\theta}_{r'};\ \hat{L_{r'}};\ \hat{I}(\hat{\theta}_{r'});\ pvals_{r'};\ \mathrm{BIC}_{r'} = \mathrm{EvaluateRegressors}(X_c;\ \hat{R}_r;\ r')$

$r^{\dagger}, \{j_i\} = \mathrm{Forward\,Model\,Proposal}(X_c,\ \hat{R}_r,\ r',\ \gamma,\ k,\ \mathrm{BIC}_{r'})$
▷ Both sub routines are explained in algorithms 3 and 4
**if** $\{j_i\} = \emptyset$ **then**
▷ No suitable candidate found
**return** $(\hat{\theta}_{r'};\ \hat{L_{r'}};\ \hat{I}(\hat{\theta}_{r'});\ \mathrm{pvals}_{r'};\ \mathrm{BIC}_{r'})$
**end if**
success = False
$n = |\{j_i\}|$
$r_{\mathrm{best}} = r'$
$\mathrm{BIC}_{\mathrm{best}} = \mathrm{BIC}_{r'}$
**while** not sucess **do**

$r^{\ddagger} = r' + \{j_i\}[:n]$
$\hat{\theta}_{r^{\ddagger}};\ \hat{L_{r^{\ddagger}}};\ \hat{I}(\hat{\theta}_{r^{\ddagger}});\ pvals_{r^{\ddagger}};\ \mathrm{BIC}_{r^{\ddagger}} = \mathrm{EvaluateRegressors}(X_c;\ \hat{R}_r;\ r^{\ddagger})$
▷ Update best set found so far using Algorithm 3
**if** $max(\mathrm{pvals}_{r^{\ddagger}}) \leq \gamma$ and $\mathrm{BIC}_{r^{\ddagger}} \leq \mathrm{BIC}_{\mathrm{best}}$ **then**
$r_{\mathrm{best}} = r^{\ddagger}$
$\mathrm{BIC}_{\mathrm{best}} = \mathrm{BIC}_{r^{\ddagger}}$
**end if**
▷ Already found best set in the descending sequence, update and exit loop
**if** $\mathrm{BIC}_{r^{\ddagger}} \geq \mathrm{BIC}_{\mathrm{best}}$ and $r_{\mathrm{best}} \neq r'$ **then**
success = True
$r' = r_{\mathrm{best}}$
**end if**
$n = n - 1$
**end while**
**until** True

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

**Algorithm 2** Iterative Active Learning

**Require:** $X,\ I,\ k,\ \gamma,\ \beta$

▷ $X$ Initial observations for all neurons
▷ $I$ Initial applied stimuli
▷ $k$ maximum number of regressors to add per step
▷ $\gamma$ minimum allowed regressor p-value
▷ $\beta$ active learning smoothing parameter

**for** $l \in$ interventions **do**
  $\hat{R}_r \leftarrow$ BuildRegressors$(X, I)$ ▷ Build regressor sequences from $X$ and $I$ as defined for Eq. (1)
  **for** $c \in$ neurons **do**
    ▷ Perform model selection using Algorithm 4
    $(\hat{\theta}_{r'}[c], \hat{L_{r'}}[c], \hat{I}(\hat{\theta}_{r'})[c], pvals_{r'}[c], \mathrm{BIC}_{r'}[c]) = \mathrm{ElasticForwardSelection}(\mathrm{X_c}, \hat{\mathrm{R}}_\mathrm{r}, \mathrm{k}, \gamma)$

    ▷ Get log-likelihood difference of considering all regressors that are not parent edges of neuron $c$
    $R_{nonparent(c)} = \{r{:}\hat{\theta}_{r'}[c] = 0\}$

    **for** $r \in R_{nonparent(c)}$ **do**
      $L_{r,c} \leftarrow$ Compute Loglikelihood difference ▷ Use Eq. (13) for neuron regressor and Eq. (16) for stimuli regressor
    **end for**
  **end for**

▷ Compute the impact and score of each stimuli according to previous observations
  **for** $s \in stimuli$ **do**
    **for** $c \in neurons$ **do**
      $IM_{s,c} \leftarrow$ Compute impact of stimuli $s$ on neuron $c$ ▷ Use Eq. (12)
    **end for**

    **for** $s_i \in stimuli$ **do**
      $IM_{s,s_i} \leftarrow$ Compute impact of stimuli $s$ on stimuli $s_i$ ▷ Use Eq. (15)
    **end for**
    $SC_s \leftarrow$ Compute score of stimuli $s$ ▷ Use Eqs. (14,17,18)
  **end for**

  $P^{l+1} \leftarrow$ Compute Stimuli Probability Vector ▷ Use Eqs. (19,21)
  $(X, I) \leftarrow$ AcquireSamples ▷ Acquire samples of $X$ and $I$ drawing $I$ with probability $P^{l+1}$

**end for**

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

**Algorithm 3** Sub routine: Evaluate Regressors

**Require:** $X_c,\ \hat{R}_r,\ r^\dagger$

▷ $X_c$ sequence of observations of neuron $c$

▷ $\hat{R}_r$ is the full set of regressor observations ($\hat{R}_r = [\hat{X}, \hat{I}, 1]$)

▷ $r^\dagger$ subset of regressors to consider

$\hat{\theta}_{r^\dagger} = \text{argmax}_{\theta_{r^\dagger}} \mathcal{L}(X_c; \hat{R}_{r^\dagger}; \theta_{r^\dagger})$

$\hat{L}_{r^\dagger} = \mathcal{L}(X_c; \hat{R}_{r^\dagger}; \hat{\theta}_{r^\dagger})$

$\hat{I}(\hat{\theta}_{r^\dagger}) = -E[\frac{\partial^2 \hat{\mathcal{L}}}{\partial\theta_a \partial\theta_b}]$

$\text{pvals}_{r^\dagger} = chi2sf(\hat{\theta}_{r^\dagger}/diag([\hat{I}(\hat{\theta}_{r^\dagger})]^{-1}))$ ▷ Compute $\mathcal{X}^2_{df=1}$ survival function for each parameter in $r^\dagger$

$\text{BIC}_{r^\dagger} = -2\hat{L}_{r^\dagger} + \ln(n) \times \left|r^\dagger\right|$

**return** $\hat{\theta}_{r^\dagger};\ \hat{L}_{r^\dagger};\ \hat{I}(\hat{\theta}_{r^\dagger});\ \text{pvals}_{r^\dagger};\ \text{BIC}_{r^\dagger}$

**Algorithm 4** Sub routine: Forward Model Proposal

**Require:** $X_c,\ \hat{R}_r,\ r,\ \gamma,\ k,\ \text{baseBIC}$

▷ Draw $n_{\text{splits}}$ sets from data and regressor sequence randomly, each set will contain $p\%$ of the total sample count

$\{X_{c,i}, \hat{R}_{r,i}\}_{i=1,\ldots,n_{\text{splits}}} = \text{Random Samples}(X, \hat{R}_r, p\%, n_{\text{splits}})$

▷ Add regressors individually and compute indicators

**for** $j \notin r$ **do**

$r^\dagger = r + \{j\}$

**for** $i = 1, ..., n_{\text{splits}}$ **do**

$_;\ _;\ _;\ \text{pvals}^\dagger_{\text{split}}[i];\ \text{BIC}^\dagger_{\text{split}}[i] = \text{EvaluateRegressors}(X_{c,i};\ \hat{R}_{r,i};\ r^\dagger)$

**end for** ▷ Evaluate BIC score and p-values on the full dataset as well. Max function provides an extra safeguard against bad regressors

$_;\ _;\ _;\ \text{pvals}^\dagger_{\text{full}};\ \text{BIC}^\dagger_{\text{full}} = \text{EvaluateRegressors}(X_c;\ \hat{R}_r;\ r^\dagger)$

$\text{pval}_{\text{score}}[j] = max(median(\text{pvals}^\dagger_{\text{split}}), \text{pvals}^\dagger_{\text{full}})$

$\text{BIC}_{\text{score}}[j] = max(median(\text{BIC}^\dagger_{\text{split}}), \text{BIC}^\dagger_{\text{full}})$

**end for**

▷ Sort candidates by best BIC in set, and select at most k of them, subject to $\gamma$ restriction

$\{j_i\} = \text{argsort}(\text{BIC}_{\text{score}}[\text{BIC}_{\text{score}} < \text{baseBIC} \wedge \text{p} - \text{value}_{\text{score}} < \gamma])[0:k]$

**return** $r^\dagger = r + \{j_i\},\ \{j_i\}$

# Acknowledgments

We thank Prof. Gregory Randall for important discussions during the early stages of this work. This work has been supported by NIH, NSF, and DoD.

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

# References


[1] Herbert A Simon. The architecture of complexity. In *Facets of systems science*, pages 457–476. Springer, 1991.

[2] Aman B Saleem, Aslı Ayaz, Kathryn J Jeffery, Kenneth D Harris, and Matteo Carandini. Integration of visual motion and locomotion in mouse visual cortex. *Nature Neuroscience*, 16(12):1864–1869, 2013.

[3] Aslı Ayaz, Aman B Saleem, Marieke L Schölvinck, and Matteo Carandini. Locomotion controls spatial integration in mouse visual cortex. *Current Biology*, 23(10):890–894, 2013.

[4] A Moses Lee, Jennifer L Hoy, Antonello Bonci, Linda Wilbrecht, Michael P Stryker, and Cristopher M Niell. Identification of a brainstem circuit regulating visual cortical state in parallel with locomotion. *Neuron*, 83(2):455–466, 2014.

[5] Dario L Ringach, Guillermo Sapiro, and Robert Shapley. A subspace reverse-correlation technique for the study of visual neurons. *Vision Research*, 37(17):2455–2464, 1997.

[6] Ian Nauhaus, Kristina J Nielsen, and Edward M Callaway. Nonlinearity of two-photon ca2+ imaging yields distorted measurements of tuning for v1 neuronal populations. *Journal of Neurophysiology*, 107(3):923–936, 2011.

[7] Vincent Bonin, Mark H Histed, Sergey Yurgenson, and R Clay Reid. Local diversity and fine-scale organization of receptive fields in mouse visual cortex. *Journal of Neuroscience*, 31(50):18506–18521, 2011.

[8] Judson P Jones and Larry A Palmer. An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology*, 58(6):1233–1258, 1987.

[9] Adam M Packer, Darcy S Peterka, Jan J Hirtz, Rohit Prakash, Karl Deisseroth, and Rafael Yuste. Two-photon optogenetics of dendritic spines and neural circuits. *Nature Methods*, 9(12):1202–1205, 2012.

[10] Adam M Packer, Lloyd E Russell, Henry WP Dalgleish, and Michael Häusser. Simultaneous all-optical manipulation and recording of neural circuit activity with cellular resolution in vivo. *Nature Methods*, 12(2):140–146, 2015.

[11] John Ashworth Nelder and R Jacob Baker. *Generalized Linear Models*. Wiley Online Library, 1972.

[12] Dario L Ringach, Patrick J Mineault, Elaine Tring, Nicholas D Olivas, Pablo Garcia-Junco-Clemente, and Joshua T Trachtenberg. Spatial clustering of tuning in mouse primary visual cortex. *Nature Communications*, 7, 2016.

[13] Brian J Malone and Dario L Ringach. Dynamics of tuning in the fourier domain. *Journal of Neurophysiology*, 100(1):239–248, 2008.

[14] Patrick J Mineault, Elaine Tring, Joshua T Trachtenberg, and Dario L Ringach. Enhanced spatial resolution during locomotion and heightened attention in mouse primary visual cortex. *Journal of Neuroscience*, 36(24):6382–6392, 2016.

[15] Philipp Berens, Jeremy Freeman, Thomas Deneux, Nicolay Chenkov, Thomas McColgan, Artur Speiser, Jakob H Macke, Srinivas Turaga, Patrick Mineault, and Peter Rupprecht. Community-based benchmarking improves spike inference from two-photon calcium imaging data. *bioRxiv*, page 177956, 2017.

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

[16] Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[17] Frank E Harrell, Kerry L Lee, Robert M Califf, David B Pryor, and Robert A Rosati. Regression modelling strategies for improved prognostic prediction. *Statistics in Medicine*, 3(2):143–152, 1984.

[18] Lucien Le Cam. *Asymptotic methods in statistical decision theory*. Springer Science & Business Media, 2012.

[19] Erich L Lehmann and George Casella. *Theory of Point Estimation*. Springer Science & Business Media, 2006.

[20] Harald Cramér. *Mathematical Methods of Statistics (PMS-9)*, volume 9. Princeton University Press, 2016.

[21] C Radhakrishna Rao. Information and the accuracy attainable in the estimation of statistical parameters. In *Breakthroughs in Statistics*, pages 235–247. Springer, 1992.

[22] George H John, Ron Kohavi, and Karl Pfleger. Irrelevant features and the subset selection problem. In *Machine learning: Proceedings of the Eleventh International Conference*, pages 121–129, 1994.

[23] Alan Miller. *Subset Selection in Regression*. CRC Press, 2002.

[24] Matthew RE Symonds and Adnan Moussalli. A brief guide to model selection, multimodel inference and model averaging in behavioural ecology using akaike's information criterion. *Behavioral Ecology and Sociobiology*, 65(1):13–21, 2011.

[25] David Posada, Thomas R. Buckley, and Jeffrey Thorne. Model selection and model averaging in phylogenetics: Advantages of akaike information criterion and bayesian approaches over likelihood ratio tests. *Systematic Biology*, 53(5):793–808, 2004.

[26] Kenneth P Burnham and David R Anderson. Multimodel inference: understanding aic and bic in model selection. *Sociological Methods & Research*, 33(2):261–304, 2004.

[27] Emmanuel Candes and Terence Tao. The dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, pages 2313–2351, 2007.

[28] Edward I George and Robert E McCulloch. Variable selection via gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.

[29] Hirotogu Akaike. Information theory and an extension of the maximum likelihood principle. In *Selected Papers of Hirotugu Akaike*, pages 199–213. Springer, 1998.

[30] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[31] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *Nature*, 393(6684):440–442, 1998.

[32] Olaf Sporns and Jonathan D Zwi. The small world of the cerebral cortex. *Neuroinformatics*, 2(2):145–162, 2004.

[33] Shan Yu, Debin Huang, Wolf Singer, and Danko Nikolić. A small world of neuronal synchrony. *Cerebral Cortex*, 18(12):2891–2901, 2008.

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

[34] Danielle Smith Bassett and ED Bullmore. Small-world brain networks. *The Neuroscientist*, 12(6):512–523, 2006.

[35] Pavan Ramkumar. Pyglmnet. `https://github.com/glm-tools/pyglmnet`, commit = 962b6dc, 2017.

[36] Elad Schneidman, Michael J Berry, Ronen Segev II, and William Bialek. Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature*, 440(7087):1007, 2006.

[37] Luis MA Bettencourt, Greg J Stephens, Michael I Ham, and Guenter W Gross. Functional structure of cortical neuronal networks grown in vitro. *Physical Review E*, 75(2):021915, 2007.

[38] Judea Pearl. *Causality*. Cambridge University Press, 2009.

[39] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. Using bayesian networks to analyze expression data. *Journal of computational biology*, 7(3-4):601–620, 2000.

[40] Joseph L Natale, David Hofmann, Damián G Hernández, and Ilya Nemenman. Reverse-engineering biological networks from large data sets. *arXiv preprint arXiv:1705.06370*, 2017.

[41] Il Memming Park, Miriam LR Meister, Alexander C Huk, and Jonathan W Pillow. Encoding and decoding in parietal cortex during sensorimotor decision-making. *Nature Neuroscience*, 17(10):1395–1403, 2014.

[42] Ben Shababo, Brooks Paige, Ari Pakman, and Liam Paninski. Bayesian inference and online experimental design for mapping neural microcircuits. In *Advances in Neural Information Processing Systems*, pages 1304–1312, 2013.

[43] Yuriy Mishchencko, Joshua T Vogelstein, and Liam Paninski. A bayesian approach for inferring neuronal connectivity from calcium fluorescent imaging data. *The Annals of Applied Statistics*, pages 1229–1261, 2011.

[44] Jeremy Lewi, Robert Butera, and Liam Paninski. Efficient active learning with generalized linear models. In *Artificial Intelligence and Statistics*, pages 267–274, 2007.

[45] Liam Paninski, Jonathan Pillow, and Jeremy Lewi. Statistical models for neural encoding, decoding, and optimal stimulus design. *Progress in Brain Research*, 165:493–507, 2007.

[46] David Pfau, Eftychios A Pnevmatikakis, and Liam Paninski. Robust learning of low-dimensional dynamics from large neural ensembles. In *Advances in Neural Information Processing Systems*, pages 2391–2399, 2013.

[47] Jeremy Lewi, Robert Butera, and Liam Paninski. Real-time adaptive information-theoretic optimization of neurophysiology experiments. In *Advances in Neural Information Processing Systems*, pages 857–864, 2007.

[48] Ian H Stevenson, James M Rebesco, Lee E Miller, and Konrad P Körding. Inferring functional connections between neurons. *Current Opinion in Neurobiology*, 18(6):582–588, 2008.

[49] Sanggyun Kim, David Putrino, Soumya Ghosh, and Emery N Brown. A granger causality measure for point process models of ensemble neural spiking activity. *PLoS Computational Biology*, 7(3):e1001110, 2011.

bioRxiv preprint doi: https://doi.org/10.1101/268599; this version posted February 20, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

[50] Hyunghoon Cho, Bonnie Berger, and Jian Peng. Reconstructing causal biological networks through active learning. *PloS One*, 11(3):e0150611, 2016.

[51] David JC MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.

[52] Jie Yang and Abhyuday Mandal. D-optimal factorial designs under generalized linear models. *Communications in Statistics-Simulation and Computation*, 44(9):2264–2277, 2015.