Network Reconstruction from Perturbation Time Course Data

Gregory R Smith[1,*], Deepraj Sarmah[2,*], Mehdi Bouhaddou[3,4], Alan D. Stern[5], James Erskine[2], and Marc R Birtwistle[2,#]

[1]Department of Neurology, Center for Advanced Research on Diagnostic Assays, Icahn School of Medicine at Mount Sinai, New York, New York, USA

[2]Department of Chemical and Biomolecular Engineering, Clemson University, Clemson, SC

[3]J. David Gladstone Institutes, San Francisco, CA 94158, USA

[4]Department of Cellular and Molecular Pharmacology, University of California San Francisco, San Francisco, CA, 94158, USA

[5]Department of Pharmacological Sciences, Icahn School of Medicine at Mount Sinai, New York, New York, USA

*These authors contributed equally

[#]To whom all correspondence should be addressed: mbirtwi@clemson.edu

Summary

Network reconstruction is an important objective for understanding biological interactions and their role in disease mechanisms and treatment. Yet, even for small systems, contemporary reconstruction methods struggle with critical network properties: (i) edge causality, sign and directionality; (ii) cycles with feedback or feedforward loops including self-regulation; (iii) dynamic network behavior; and (iv) environment-specific effects. Moreover, experimental noise significantly impedes many methods. We report an approach that addresses the aforementioned challenges to robustly and uniquely infer edge weights from sparse perturbation time course data that formally requires only one perturbation per node. We apply this approach to randomized 2 and 3 node systems with varied and complex dynamics as well as to a family of 16 non-linear feedforward loop motif models. In each case, we find that it can robustly reconstruct the networks, even with highly noisy data in some cases. Surprisingly, the results suggest that incomplete perturbation (e.g. 50% knockdown vs. knockout) is often more informative than full perturbation, which may fundamentally change experimental strategies for network reconstruction. Systematic application of this method can enable unambiguous network reconstruction, and therefore better prediction of cellular responses to perturbations such as drugs. The method is general and can be applied to any network inference problem where perturbation time course experiments are possible.

Introduction

Networks underlie much cellular and biological behavior, including transcriptional, protein-protein interaction, signaling, metabolic, cell-cell, endocrine, ecological, and social networks, among many others. As such, identifying and then representing their structure has been a focus of many for decades now. This is not just from experimental perspectives alone, but predominantly computational with a variety of statistical methodologies that integrate prior knowledge from interaction databases with new experimental data sets (Angulo et al., 2017; Barabási and Albert, 1999; Califano et al., 2012; Calvano et al., 2005; Dorel et al., 2018; Hackett et al., 2020; Hein et al., 2015; Hill et al., 2017a, 2017b; Ideker et al., 2001, 2002; Liu et al., 2013; Ma'ayan et al., 2005; Margolin et al., 2006; Mazloom et al., 2011; Mehla et al., 2015; Molinelli et al., 2013; Pe'er et al., 2001; Pósfai et al., 2013; Schraivogel et al., 2020; Shannon et al., 2003; Stein et al., 2015; Wynn et al., 2018).

Network structure is usually represented as either an undirected or a directed graph, with edges between nodes specifying the system. There are four main areas where current approaches to reconstructing networks struggle to capture important features of biological networks. The first is causality of edges (Hackett et al., 2020; Hill et al., 2017a; Shipley, 2016). Commonly employed correlational methods predominantly generate undirected, non-causal edges. However, a complete and predictive understanding of the biological networks that dictate cellular behavior requires causal information. Second is cycles. Cycles such as feedback or feedforward loops are nearly ubiquitous in biological systems and central to their function (Mangan and Alon, 2003; Reeves, 2019). This also includes an important type of cycle: self-regulation of a node, that is, an edge onto itself, which is rarely considered (Fournier et al., 2007). Third is that biological networks are often dynamic and quantitative. Two notable examples are circadian and p53

oscillators (Bell-Pedersen et al., 2005; Stewart-Ornstein et al., 2017), where dynamics are key to biological function. Directionality and edge signs (i.e. positive or negative) dictate dynamics. Fourth is identifying causal edges from environmental variables. For example, if a growth factor is present, how does that growth factor influence production and/or degradation of other nodes? Is it direct, or through other explicitly considered nodes? The experimental design and data requirements to uniquely identify the ***causal, dynamic, directed and signed*** edge structures in biological networks containing all types of cycles remains a largely open but significant problem. Any such design should ideally be feasible to implement with current experimental technologies.

Modular Response Analysis (MRA) approaches, first pioneered by Kholodenko and colleagues in 2002 (Kholodenko et al., 2002; Santra et al., 2018) inherently deal with cycles and causality by prescribing systematic perturbation experiments followed by steady-state measurements. The premise for data requirements is to measure the entire system response to at least one perturbation for each node. Thus, an n node system requires n experiments, if the system response can be measured in a global fashion (i.e. all nodes measured at once). The original instantiations struggled with the impact of experimental noise, but total least squares MRA and Monte Carlo sampling helped to improve performance (Andrec et al., 2005; Santos et al., 2007). Incomplete and prior knowledge can be handled as well using both maximum likelihood and Bayesian approaches (Gross and Blüthgen, 2020; Halasz et al., 2016; Klinger et al., 2013; Santra et al., 2013). However, these approaches are based on steady-state data, or fixed time point data, limiting abilities to deal with dynamic systems. There is a formal requirement for small perturbations, which are experimentally problematic and introduce issues for estimation with noisy data. Subsequent approaches have recommended the use of large perturbations as a trade off in dealing with noisy data, but the theory still formally requires small perturbations

(Thomaseth et al., 2018). Lastly, there are two classes of biologically-relevant edges that MRA does not comprehensively address: self-regulation of a node, which is often normalized causing it to not be uniquely identifiable and effects of an environmental stimulus (basally present or administered) on a node which is not estimated.

In addition to perturbations, another experimental design feature that can inform causality is a time-series. One MRA variant (Sontag et al., 2004) uses time-series perturbation data to uniquely infer a signed, directed network that can predict time-dependent network behavior. However, the data requirements are higher than MRA. In an *n* node system, multiple nodes may need to be distinctly perturbed more than once, such as both production and degradation of a transcript, or phosphorylation and dephosphorylation of a protein. This can be experimentally challenging both in terms of scale and finding suitable distinct perturbations for a node. Moreover, as is often the case, noise in the experimental data severely limits inference accuracy (due to estimation of 2nd derivatives), which has limited its practical application. Thus, there remains a need for inference methods that can reconstruct signed, directed networks from realistic perturbation time course experiments that capture dynamics, can uniquely estimate edge properties related to self-regulation and environmental stimuli, and finally that function in the presence of typical experimental noise levels.

Here we describe a novel, MRA-inspired approach that specifies feasible perturbation time course experiments and a least squares parameter estimation method that is capable of handling significant experimental noise. This approach is capable of robustly and uniquely reconstructing causal, dynamic, directional, and signed networks that include cycles, self-regulation, and environmental stimulus effects. Perturbations can be large, and we surprisingly find that in some circumstances, incomplete perturbation (e.g. partial knockdown) is more

informative than complete perturbation (e.g. gene knockout). For an *n*-node system, *n* perturbation time courses are required, and thus in principle the method scales reasonably as network size increases. We test our approach using simulated time-series perturbation data with known network topology under increasing levels of simulated noise. The approach has good accuracy and precision for identifying network structure in randomly generated 2 and 3 node networks that contain a wide variety of cycles and oscillatory properties. For the investigated cases, we find between 7 to 11 time points yielded reasonable results. We then apply the approach to a family of 16 non-linear feedforward loop network models and find that it can reconstruct networks with reasonable specificity and sensitivity (AUROC~0.93-0.99, depending on noise level), even with highly noisy data in some cases. Surprisingly, application to the non-linear models suggested that incomplete perturbation (e.g. partial knockdown vs. knockout) is more informative than complete inhibition. While challenges remain for expanding to other and larger systems with more parameters, the proposed algorithm is able to robustly reconstruct a wide range of networks with good specificity and sensitivity using feasible time course experiments, all while addressing several important limitations of current inference approaches.

Results

Formulation of Sufficient Experimental Data Requirements for Network Reconstruction

Consider a 2-node network with four directed, weighted edges (Fig. 1a). A stimulus may affect each of the two nodes differently and its effect is quantified by $S_{1,ex}$ and $S_{2,ex}$, respectively (e.g. Methods, Eq. 16). We also allow for basal/constitutive production in each node ($S_{i,b}$). Let $x_i(k)$ be the activity of node i at time point t_k. The network dynamics can be cast as a system of ordinary differential equations (ODEs) as follows

$$\frac{dx_1}{dt} \equiv f_1(x_1(k), x_2(k), S_{1,ex}, S_{1,b}) \equiv f_1(k); \frac{dx_2}{dt} \equiv f_2(x_1(k), x_2(k), S_{2,ex}, S_{2,b}) \equiv f_2(k) \,. \qquad (1)$$

The network edges can be connected to the system dynamics through the Jacobian matrix **J** (Kholodenko et al., 2002; Santra et al., 2018; Sontag et al., 2004),

$$\mathbf{J} \equiv \begin{pmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{pmatrix} \equiv \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{pmatrix} \qquad (2)$$

The network edge weights (F_{ij}'s) describe how the activity of one node affects the dynamics of another node in a causal and direct sense, given the explicitly considered nodes (though not necessarily in a physical sense). In MRA, these nodes may be individual species or "modules". In order to simplify a complex network it may often be separated into "modules" comprising smaller networks of inter-connected species with the assumption that each module is generally insulated from other modules except for information transfer through so-called communicating species (Kholodenko et al., 2002). Cases where such modules may not be completely isolated are explored elsewhere (Lill et al., 2019).

What experimental data are sufficient to uniquely estimate the signed directionality of the network edges and thus infer the causal relationships within the system? Fundamentally, we know that perturbations and/or dynamics are important for inferring causality (Shipley, 2016). Previous work showed how steady-state data combined with small perturbations of each node can provide estimates for the edge weights through Modular Response Analysis (Kholodenko et al., 2002). Extensions to dynamic data relied on estimation of 2nd derivatives and were not robust to experimental noise. They also formally required small perturbations, which are often not practical.

Consider a simple experimental setup of three time course experiments that each measure x_1 and x_2 dynamics in response to a stimulus (Fig. 1b-g). One time course is in the presence of no perturbation (vehicle), one has a perturbation of Node 1, and one has a perturbation of Node 2. Consider further that the perturbations are reasonably specific, such that the perturbation of x_1 has negligible *direct* effects on x_2, and vice versa, and that they may be large. Experimentally, this would be an shRNA or gRNA that is specific to a particular considered node, or that a small molecule inhibitor is used at low enough dose to predominantly inhibit the targeted node. A well-posed estimation problem can be formulated (see Methods) that, in principal, allows for unique estimation of the Jacobian elements as a function of time with the following set of linear algebra relations:

$$\begin{bmatrix} y_1(t_{k+1}) \\ y_{1,2}(t_k) \end{bmatrix} = \begin{bmatrix} \Delta_t x_1(t_{k+1}) & \Delta_t x_2(t_{k+1}) \\ \Delta_{p,2} x_1(t_k) & \Delta_{p,2} x_2(t_k) \end{bmatrix} \begin{bmatrix} F_{11}(t_k) \\ F_{12}(t_k) \end{bmatrix} \tag{3}$$

$$\begin{bmatrix} y_2(t_{k+1}) \\ y_{2,1}(t_k) \end{bmatrix} = \begin{bmatrix} \Delta_t x_1(t_{k+1}) & \Delta_t x_2(t_{k+1}) \\ \Delta_{p,1} x_1(t_k) & \Delta_{p,1} x_2(t_k) \end{bmatrix} \begin{bmatrix} F_{21}(t_k) \\ F_{22}(t_k) \end{bmatrix} \tag{4}$$

Here, $y_{i,j}$ refers to a measured first-time derivative of node i in the presence of node j perturbation (if used), and Δ to a difference with respect to perturbation (subscript p) or time (subscript t) (see Methods). Note, we do not use data from the perturbation of node i for estimation of node i edges (see Methods). Importantly, this formally relaxes constraints on how large the perturbation can be, following recent recommendations (Thomaseth et al., 2018) (see also Methods).

Note that this formulation is generalizable to an n dimensional network. With n^2 unknown parameters in the Jacobian matrix, n equations originate from the vehicle perturbation and n-1 equations originate from each of the n perturbations (discarding equations from Node i with Perturbation i). This results in $n + n*(n-1) = n + n^2 - n = n^2$ independent equations.

The above provides a basis for expecting the proposed time course data to contain sufficient information to constrain network reconstruction uniquely. However, the effects of stimulus (which may comprise both basal production and external stimuli) still remain to be determined. This was not explored in previous MRA theory but is an important network feature. Analytically determining the effects of stimulus requires evaluation of specific scenarios. For example, in the case that there is no basal production, the stimulus strengths $S_{1,ex}$ and $S_{2,ex}$ are time-invariant step changes, additive (with respect to the right hand side of the ODEs), and a steady-state exists, they can be determined from steady-state data $x_{1,ss}$ and $x_{2,ss}$ and the estimated Jacobian values F_{ij} (See Methods, Eq. 19).

$$\begin{aligned} S_{1,ex} &= -(F_{11}x_{1,ss} + F_{12}x_{2,ss}) \\ S_{2,ex} &= -(F_{21}x_{1,ss} + F_{22}x_{2,ss}) \end{aligned} \quad (5)$$

In the examples used hereafter, we focus on such step change, time invariant stimuli that are a predominant form of experimental investigation.

Using Sufficient Simulated Data to Reconstruct a Network

As an initial test of the above formulation, we used a simple 2 node, single activator network where Node 1 activates Node 2, and both nodes have first-order degradation (-1 diagonal elements) (Fig. 1a—see Methods for equations). Following a stimulus that increases the activity of each node, we simulated an 11-point time course experiment. This simulation was done for the vehicle control and for each perturbation case (Node 1 and Node 2) to generate the necessary simulation data per the theoretical considerations above (Fig. 1c,e,g, left panel). Here, we modeled perturbations as complete inhibition, where a perturbation of Node 1 makes its value 0 at all times. Solving Eqs. 3-4 to infer the Jacobian elements at each time point yielded good agreement between the median estimates and the ground truth values (Fig. 1h, "Analytic Solution", No Noise). Using the node activity data corresponding to the last time point in the time course and the median estimates of Jacobian elements, the stimuli $S_{1,ex}$ and $S_{2,ex}$ were also determined (Eq.19-20) and reasonably agree with the ground truth values.

How does this approach fare when data are noisy? We performed the estimation with the same data but with a relatively small amount of simulated noise added (10:1 signal-to-noise—Fig 1c,e,g). The resulting estimates are neither accurate nor precise, varying on a scale more than ten times greater than each parameter's magnitude with median predictions both positive and negative regardless of the ground truth value (Fig. 1i). The stimulus strengths $S_{1,ex}$ and $S_{2,ex}$ are estimated to be negative, while the ground truth is positive.

Although the analytic equations suggest the sufficiency of the perturbation time course datasets to uniquely estimate the edge weights, in practice even small measurement noise corrupts estimates obtained from direct solution of these equations. Therefore, we considered an alternative representation by employing a least squares estimation approach rather than solve the linear equations directly. Least squares methods were shown to improve traditional MRA-based approaches (Andrec et al., 2005; Santos et al., 2007), but had never been formulated for dynamic problems. A major hurdle was how to model the effect of a perturbation without (i) adding additional parameters to estimate or (ii) requiring strong functional assumptions regarding perturbation action. We solved this here by implementing a novel means of directly using the already-available experimental measurements within the context of the least-squares estimation (see Methods). For a given set of guesses for edge weight and stimulus parameters, one can integrate to obtain a solution for the dynamic behavior of the resulting model, which can be directly compared to data in a least-squares sense. We applied this new approach to the single activator model, 10:1 signal-noise ratio case above where the analytic approach failed. This new estimation approach was able to infer the network structure accurately and precisely (Fig. 1j). We conclude that analytic formulations can be useful for suggesting experimental designs that should be sufficient for obtaining unique estimates for a network reconstruction exercise, but in practice directly applying those equations may not yield precise nor accurate estimates. Alternatively, using a least-squares formulation seems to work well for this application.

Reconstruction of Random 2 and 3 Node Networks

To investigate the robustness of the least-squares estimation approach, we applied it to increasingly complex networks with larger amounts of measurement noise and smaller numbers of time points (Fig. 2). We focused on 2 and 3 node networks. We generated 50 randomized 2

and 3 node models, where each edge weight is randomly sampled from a uniform distribution over the interval [-2,2], and the basal and external strength from [0,2] (Fig. 2a, S1a, S2a). Each random network was screened for stability. Unstable networks were omitted (any network with positive real part of a Jacobian eigenvalue). Many networks (29/50 for 2 node and 26/50 for 3 node) displayed potential for oscillatory behavior (non-zero imaginary parts of eigenvalues). For each random model, we generated a simulated dataset based on the prescribed experimental design, using complete inhibition as the perturbation. We considered time courses with different numbers of time points (3, 7, 11 and 21), and added 10:1 signal-to-noise, 5:1 signal-to-noise, and 2:1 signal-to-noise to the data.

For each random network model, number of time points, and noise level, we evaluated the fidelity of the proposed reconstruction approach in terms of signed directionality (Fig. 2c-f). We overall found reasonable agreement between inferred and ground truth values, even at the higher noise levels and low number of timepoints. Expectedly, the overall classification accuracy increases with more time points and decreases with higher noise levels. But, surprisingly, even in the worst case investigated of 3 timepoints and 2:1 signal-to-noise ratio, classification accuracy was above 70% (2 node) and 60% (3 node). Increasing the number of nodes decreases performance, with 3-node reconstruction being slightly worse than 2-node reconstruction, other factors held constant.

We wondered whether the magnitude of an edge weight influenced its classification accuracy, since small edge weights may be more difficult to discriminate from noise. We found that edge weights with smaller absolute values, which are expected to have a smaller influence on the networks, were more likely to be classified incorrectly, and the estimates get progressively better with greater edge weights (Fig. S1b-e, S2b-e).

We also wondered whether models with complex dynamics such as oscillations would be easier or harder to reconstruct. For models with potential oscillatory behavior, the classification accuracy is very similar (Fig. S3a, S3b) for all 50 random models. This indicates that networks with potential oscillatory behavior can be reconstructed just as well as the ones without oscillatory behavior.

We conclude that using least squares estimates and an experimentally feasible number of timepoints, the network parameters of 2 and 3 node systems can be determined with a high degree of accuracy, even in the presence of noise. Models with oscillatory properties can be reconstructed as accurately as models without these properties. Expanding to larger networks will decrease the accuracy of the estimates, but a greater number of timepoints or more replicates to improve precision at each time point could help.

Application to Non-Linear Feed Forward Loop Networks Demonstrates Partial Perturbations are More Informative than Full Perturbations

The previous applications were idealized cases, where the parameters F_{ij} were time-invariant and the model used to generate simulated data was the same as the one use to fit the data. Here, we challenged the proposed algorithm further with a series of well-studied non-linear feed forward loop (FFL) network models that have time-varying Jacobian elements (Fig. 3a, Table 1) (Mangan and Alon, 2003; Reeves, 2019). Such FFL motifs are strongly enriched in multiple organisms and are important for signaling functions such as integrative control, persistence detection, and fold-change responsiveness (Goentoro and Kirschner, 2009; Goentoro et al., 2009; Nakakuki et al., 2010).

The FFL network has three nodes (x_1, x_2, and x_3), and the external stimulus acts on x_1 ($S_{1,ex}$). There is no external stimulus on x_2 and x_3; however, there may be basal production of x_2 ($S_{2,b}$) and x_3 ($S_{3,b}$),. Each node exhibits first-order decay (F_{ii}=-1). The parameters F_{12}, F_{13}, and F_{23} represent connections that do not exist in the model; we call these null edges, but we allow them to be estimated. The relationship between x_1 and x_2 (F_{21}), between x_1 and x_3 (F_{31}), or between x_2 and x_3 (F_{32}) can be either activating or inhibitory. Furthermore, x_1 and x_2 can regulate x_3 through an "AND" gate (both needed) or an "OR" gate (either sufficient) (Fig. 3a). These permutations give rise to 16 different FFL structures (Table 1).

To generate simulated experimental data from these models, we first integrated the system of ODEs starting from a *zero* initial condition to find the steady state in the absence of stimulus. We then introduced the external stimulus and integrated the system of ODEs (see Methods) to generate time series perturbation data consistent with the proposed reconstruction algorithm, using full inhibitory perturbations. We used 11 timepoints for all 16 non-linear models, based on its accuracy in the 3-node model analysis above.

We first noticed that even in the absence of added noise, a surprising number of inferences were incorrect (Fig. 3b, f). Model #1 (Table 1, Fig. 3b-c) is used as an example, where F_{21}, F_{31} and F_{32} are activators with an AND gate, and F_{31} is incorrectly predicted as null (Fig. 3b—compare ground truth to 100% inhibition). To understand the reason for the incorrect estimation, we looked at the node activity dynamics across the perturbation time courses (Fig. 3d). All three nodes start from an initial steady state of zero, but Node 3 is zero for all three perturbation cases. This is because of the following. Since x_1 is required for the activation of x_2 and x_3, complete inhibition of x_1 completely blocks both x_2 and x_3 activation. But, because both x_1 and x_2 are required for the activation of x_3, completely inhibiting x_2 activity also completely

inhibits x_3. Thus, given this experimental setup, it is impossible to discern if x_1 directly influences x_3 or if it acts solely through x_2.

We thus reasoned that full inhibitory perturbation may suppress the information necessary to correctly reconstruct the network, but that a partial perturbation experiment may contain enough information available to make a correct estimate. If this were to be true, then upon applying partial perturbations (we chose 50% here), Node 3 dynamics should show differences across the perturbation time courses. Simulations showed that this is the case (Fig. 3e). Subsequently, we found that for partial perturbation data, F_{31} is correctly identified as an activator. More broadly, we obtain perfect classification from noise-free data across all 16 FFL networks when partial perturbation data are used, as opposed to 5/16 networks having discrepancy with full perturbation data (Fig. 3f). The fits to simulated data from the reconstructed model align very closely, despite model mismatch (Fig. S4). We conclude that in these cases of non-linear networks, a partial inhibition is necessary to estimate all the network parameters accurately. Thus, moving forward, we instead applied 50% perturbation to all simulation data and proceeded with least squares estimation.

Classification of Non-Linear Feed Forward Loop Parameters from Noisy Data

The above analysis prompted us to use a partial (50%) perturbation strategy, since it classified each edge for each model in the absence of noise correctly. What classification performance do we obtain in the presence of varying levels of experimental noise? We first devised the following strategy to assess classification performance. We generated 50 bootstrapped datasets for each network structure/signal-to-noise pair, and thus obtained 50 sets of network parameter estimates. To classify the network parameters, we used a symmetric cutoff of a percentile window around the median of these 50 estimates (Fig. 4a). We illustrate this

approach with three different example edges and associated estimates, one being positive (Edge 1), one being negative (Edge 2), and one being null (Edge 3). Given the window of values defined by the percentile cutoff being chosen, if the estimates in this window are all positive, then the network parameter would be classified as positive. Similarly, if the estimates in this window are all negative, then the parameter would be classified as negative. Finally, if the estimates in the window cross zero (i.e. span both positive and negative terms), then it would be classified as null. First, consider the case that the percentile window is just set at the median with no percentile span. Then, the classifications for true positives and negatives are likely to be accurate while the null parameters are likely to be incorrectly categorized as either positive or negative (Fig. 4a). If we increase the percentile window span slightly (e.g. between the 40th and 60th percentile, middle panel), we can categorize null edges better, while maintaining good classification accuracy of both true positive and negative edges. However, if we relax the percentile window too much, (e.g. between the 10th and 90th percentile, far right panel) we may categorize most parameters as null, including the true positive and negatives. Thus, it is clear there will be an optimal percentile cutoff that maximizes true positives and minimizes false positives as the threshold is shifted from the median to the entire range.

Now, we applied this classification strategy to the 16 FFL model estimates from data with different noise levels. We varied the percentile window from the median only (50) to the entire range of estimated values (100) and calculated the true and false positive rates for all edges across all 16 FFL models, which allowed generation of receiver operator characteristic (ROC) curves (Fig. 4b). For each noise level, we chose the percentile window that yielded a 5% false positive rate (13-87 percentile for 10:1 Signal:Noise, 19-81 percentile for 5:1 Signal:Noise, and 21-79 percentile for 2:1 Signal:Noise). Using this simple cutoff classifier, we observed good

classification performance across all noise levels according to traditional area under the ROC curve metrics (10:1 AUC=0.99, 5:1 AUC=0.9, 2:1 AUC=0.92).

How does classification accuracy break down by FFL model and edge type? To evaluate the performance for each of the 16 FFL cases, we calculated the fraction of the 12 links in each FFL model that was classified correctly as a function of signal-to-noise, given the percentile windows determined above (Fig. 4c). We also looked at the fraction of the 16 models where each of the 12 links were correctly classified (Fig. 4d). Perfect classification is a value of one, which is the case for no noise, and for many cases with 10:1 signal-to-noise.

In general, as noise level increases, prediction accuracy decreases, as expected. Although for some models and parameters, performance at 2:1 signal-to-noise is poor, in some cases it is surprisingly good. This suggests that the proposed method can yield information even in high noise cases; this information is particularly impactful for null, self-regulatory, and stimulus edges. High noise has strong effects on inference of edges that are either distinct across models, time variant or reliant on other node activities (F_{21}, F_{31}, F_{32}) (Fig. 4cd, S5). $F_{21,}$ which is reliant on activity of x_1, is inferred better than F_{31} and F_{32}. This may be caused by the fact that data for x_3 is highly reliant on both x_1 and x_2, whereas data for x_2 is only reliant on x_1.

Comparing across models, we find that Models 1-8 are reconstructed slightly better than Models 9-16 (Fig. 4c) when noise is high. This performance gap is predominantly caused by $S_{3,b}$ misclassification—basal production of Node 3 (Fig. S5). What is the reason for the possible misclassification of $S_{3,b}$ in Models 9-16? We know that $S_{3,b}$ depends on the initial values of x_1, x_2 and x_3 and the estimated values of F_{31}, F_{32} and F_{33} (See Methods, Eq. 20). For Models 1-8, $x_1(t=0)$ and $x_2(t=0)$ are both zero and therefore $S_{3,b}$ is effectively only dependent on estimated value of F_{33} and $x_3(t=0)$ (Fig. S4 and Methods). But for Models 9-16, $x_2(t=0)$ is non-zero and $S_{3,b}$

is dependent on the estimated values of both F_{32} and F_{33}, in addition to $x_2(t=0)$ and $x_3(t=0)$, which increases the variability of $S_{3,b}$ estimates. Therefore with high levels of noise, $S_{3,b}$ is more likely to be mis-classified in Models 9-16, whereas this does not happen in Models 1-8 (Fig 4c,d, S5). In the future, including stimulus and basal production parameters in the least squares estimations themselves, rather than further deriving algebraic relations to estimate them, will likely help improve reliability.

We conclude that (i) when dealing with non-linear biological networks, complete perturbations such as genetic knockouts may fundamentally impede one's ability to deduce network architecture and (ii) this class of non-linear networks can be reconstructed with reasonable performance using the proposed strategy employing partial perturbations.

Discussion

Despite intensive research focus on network reconstruction, there is still room to improve discrimination between direct and indirect edges (i.e. causality), particularly when biologically-ubiquitous feedback and feedforward cycles are present that stymie many statistical or correlation-based methods, and also given that experimental noise is inevitable. The presented simple method prescribes a realistic experimental design for causal inference of signed, directed edges when typical levels of noise are present. It allows estimation of self-regulation edges as well as those for basal production and external stimuli. For 2 and 3 node networks, the method can successfully handle both random linear networks as well as more complex, non-linear feed forward loop structures. Prediction accuracy was strong in many cases even with simulated noise that exceeds typical experimental variability (2:1 signal-to-noise). The method presented here is quite general, and could be applied not only to molecular biology, but also vastly different fields where perturbation time course experiments are possible, and where network structures are important to determine.

Although complete inhibition is often used for perturbation studies (e.g. CRISPR-mediated gene knockout), we found that partial inhibition is important to fully reconstruct the considered non-linear FFL networks. Since many biological networks share the same types of non-linear features contained within the considered FFL models, this is not likely to be the only case when partial inhibition will be important. We are thus inclined to speculate it may be a generally important experimental design criterion moving forward. However, there is a tradeoff between perturbation strength and feasibility, since the effects of small perturbations are masked by noise (Thomaseth et al., 2018). Partial inhibition is often “built-in” to certain assay types, such as si/shRNA or pharmacological inhibition that are titratable to a certain extent.

One major remaining challenge is scaling to larger networks. Here, we limited our analysis to 2 and 3 node networks. Conveniently, the number of necessary perturbation time courses needed grows linearly (as opposed to exponentially) with the number of considered nodes. Furthermore, as long as system-wide or omics-scale assays are available, the experimental workload also grows linearly. This is routine for transcriptome analyses (Stark et al., 2019), and is becoming even more commonplace for proteomic assays (e.g. mass cytometry (Spitzer and Nolan, 2016), cyclic immunofluorescence (Lin et al., 2016), mass spectrometry (Aksenov et al., 2017), RPPA (Akbani et al., 2014)) (Lin et al., 2016a). Thus, the method is arguably experimentally scalable to larger networks.

Increasing the network size will quadratically increase the number of unknown parameters. Reducing this parameter space and obtaining good initial guesses will be important. Imposing prior knowledge can also reduce the parametric space, such as in Bayesian Modular Response Analysis (Santra et al., 2013), or with functional database information (Wu et al., 2010). As network size grows, the sparseness of the Jacobian will increase, so judicious allocation of non-zero elements will be important. Checking estimated Jacobians for emergent properties such as degree distributions for scale-free networks (Barabási and Albert, 1999) can provide additional important constraints. The approach used in this paper is accommodative of such prior knowledge and in principle can be scaled up for larger network size. Lastly, large estimation problems may be broken into several smaller problems to be merged subsequently, which is likely to yield large computational speed up by allowing parallelization of much smaller tasks.

In conclusion, the proposed approach to network reconstruction is systematic, scalable and feasible, robustly operating in the presence of experimental noise and accepting data from

large perturbations. It addresses important features of biological networks that current methods struggle to account for: causality/directionality/sign, cycles (including self-regulation), dynamic behavior (including oscillatory properties) and environmental stimuli. It does so while leveraging dynamic data of the network and only requires one perturbation per node for completeness. We expect this approach to be broadly useful not only for reconstruction of biological networks, but to enable using such networks to build more predictive models of disease and response to treatment, and more broadly, to other fields where such networks are important for system behavior.

Acknowledgments

We would like to thank Clemson University and the CCIT team for the generous allotment of time and support in the Palmetto cluster for running the simulations in this paper.

Author Contributions

MRB, GRS, and DS conceived of the work. DS, GRS, MB, MRB, and JE performed analyses. DS, MB, and GRS made the figures. DS, GRS, MB, and MRB wrote the manuscript.

Declaration of Interests

The authors declare no competing interests.

Funding

MRB acknowledges funding from Mount Sinai, Clemson University, the National Institutes of Health Grants R01GM104184 and an IBM faculty award. MB and ADS were supported by a National Institute for General Medical Sciences-funded Integrated Pharmacological Sciences Training Program grant (T32GM062754).

References

Akbani, R., Becker, K.-F., Carragher, N., Goldstein, T., Koning, L. de, Korf, U., Liotta, L., Mills, G.B., Nishizuka, S.S., Pawlak, M., et al. (2014). Realizing the Promise of Reverse Phase Protein Arrays for Clinical, Translational, and Basic Research: A Workshop Report: The RPPA (Reverse Phase Protein Array) Society. Mol. Cell. Proteomics *13*, 1625–1643.

Aksenov, A.A., da Silva, R., Knight, R., Lopes, N.P., and Dorrestein, P.C. (2017). Global chemical analysis of biology by mass spectrometry. Nat. Rev. Chem. *1*, 1–20.

Dorel, M., Klinger, B., Gross, T., Sieber, A., Prahallad, A., Bosdriesz, E., Wessels, L.F.A., and Blüthgen, N. (2018). Modelling signalling networks from perturbation data. Bioinformatics *34*, 4079–4086.

Fournier, T., Gabriel, J.P., Mazza, C., Pasquier, J., Galbete, J.L., and Mermod, N. (2007). Steady-state expression of self-regulated genes. Bioinformatics *23*, 3185–3192.

Gross, T., and Blüthgen, N. (2020). Identifiability and experimental design in perturbation studies. BioRxiv 2020.02.03.931816.

Hackett, S.R., Baltz, E.A., Coram, M., Wranik, B.J., Kim, G., Baker, A., Fan, M., Hendrickson, D.G., Berndl, M., and McIsaac, R.S. (2020). Learning causal networks using inducible transcription factors and transcriptome□wide time series. Mol. Syst. Biol. *16*.

Lill, D., Rukhlenko, O.S., Mc Elwee, A.J., Kashdan, E., Timmer, J., and Kholodenko, B.N. (2019). Mapping connections in signaling networks with ambiguous modularity. Npj Syst. Biol. Appl. *5*, 19.

Raue, A., Schilling, M., Bachmann, J., Matteson, A., Schelke, M., Kaschek, D., Hug, S., Kreutz, C., Harms, B.D., Theis, F.J., et al. (2013). Lessons Learned from Quantitative Dynamical Modeling in Systems Biology. PLOS ONE *8*, e74335.

Reeves, G.T. (2019). The engineering principles of combining a transcriptional incoherent feedforward loop with negative feedback. J. Biol. Eng. *13*, 62.

Schraivogel, D., Gschwind, A.R., Milbank, J.H., Leonce, D.R., Jakob, P., Mathur, L., Korbel, J.O., Merten, C.A., Velten, L., and Steinmetz, L.M. (2020). Targeted Perturb-seq enables genome-scale genetic screens in single cells. Nat. Methods 1–7.

Shipley, B. (2016). Cause and Correlation in Biology: A User's Guide to Path Analysis, Structural Equations and Causal Inference with R (Cambridge: Cambridge University Press).

Spitzer, M.H., and Nolan, G.P. (2016). Mass Cytometry: Single Cells, Many Features. Cell *165*, 780–791.

Stark, R., Grzelak, M., and Hadfield, J. (2019). RNA sequencing: the teenage years. Nat. Rev. Genet. *20*, 631–656.

Thomaseth, C., Fey, D., Santra, T., Rukhlenko, O.S., Radde, N.E., and Kholodenko, B.N. (2018). Impact of measurement noise, experimental design, and estimation methods on Modular Response Analysis based network reconstruction. Sci. Rep. *8*, 16217.

Wu, G., Feng, X., and Stein, L. (2010). A human functional protein interaction network and its application to cancer data analysis. Genome Biol. *11*, R53.

Figure Legends

Figure 1. **(a)** Two-node network with Jacobian elements labeled (Single Activator Model). Green arrows are stimuli and basal production terms. **(b)** Time course experimental design with vehicle perturbation. The vehicle may be the solvent like DMSO for inhibition with a drug, or a nontargeting si/shRNA for inhibition with si/shRNA. **(c,e,g)** Simulated time course data for Vehicle perturbation (c), Node 1 perturbation (e), Node 2 perturbation (g) from the network in (a). Left Column: no added noise; Right Column 10:1 signal-to-noise added. **(d,f)** Time course experimental design with perturbation of Node 1 (d) and Node 2 (f). **(h-j)** Actual versus inferred model parameters ($S_{1,b}$, $S_{1,ex}$, F_{11}, F_{12} , $S_{2,b}$, $S_{2,ex}$, F_{21}, F_{22}) for direct solution of Eq. 3-4 in the absence (h) or presence (i) of noise or with 10:1 signal-to-noise and the least-squares approach (j). In h-i, error bars are standard deviation across time points.

Figure 2. **(a)** Connections around a node in an n-Node Model. $S_{i,b}$ and $S_{i,ex}$ are the basal production and external stimulus terms acting on Node *i* respectively. F_{ii} is the self-regulation term; *Fij* the effect of Node *j* on Node *i* and F_{ji} the effect of Node *i* on Node *j*. **(b)** Example of different signal-to-noise ratio effects on time course data. **(c,d)** Ground truth versus estimated edge weights across all 50 random networks and noise levels for data from four different total timepoints (3,7,11,21) for 2 node (c) and 3 node (d) networks. If the edge weights fall in the green shaded quadrants, they are said to be correctly classified whereas if they fall in the red shaded quadrants they are said to be incorrectly classified. **(e,f)** Fraction of network parameters correctly classified in 50 randomly generated 2 Node networks (e) and 3 Node networks (f) with different noise levels and total timepoints.

Figure 3. **(a)** Feedforward loop (FFL) network models. Across all 16 models (Table 1), F_{11}, F_{22}, and F_{33} values are fixed at -1 and F_{12}, F_{13}, and F_{23} values are fixed at 0. F_{21}, F_{31}, and F_{32} values can be positive or negative depending on the model. The combined effect of x_1 and x_2 on x_3 is described by either an AND gate or an OR gate. There are 16 possible model structures (Table 1). **(b)** 100% inhibitory perturbations may not provide accurate classification even without noise. In Model #1, F_{31} is positive (ground truth) but is estimated as null. **(c)** Specific structure of Model #1. **(d)** Node activity simulation data for 100% inhibition in Model #1, implying that it is impossible to infer F_{31} from such data. **(e)** Node activity simulation data for 50% inhibition in Model #1, showing potential to infer F_{31}. **(f)** Fraction of model parameters correctly classified in all the 16 non-linear models without noise, for 100% inhibition vs 50% inhibition.

Figure 4. (a) Classification scheme for a distribution of parameter estimates. Going from left to right panels, the same parameter distribution with an actual (ground truth) value of positive (+), negative (-), or null (0), respectively, is estimated using different percentile windows centered on the median. The percentile "window" is the median value for the leftmost panel (rigorous classification), between 40th and 60th percentile in the second panel, and between 10th and 90th percentile in the third panel (conservative classification). Going from rigorous to conservative (left to right), an intermediate between the two gives a good classification performance. **(b)** ROC curves across all parameters for all 16 FFL models. Different color lines are different noise levels. **(c)** Fraction of correctly classified model parameters for different noise levels broken

down by FFL model type. **(d)** Fraction of each model parameter correctly classified for different noise levels broken down by parameter type.

Supplementary Figure Legends

Figure S1: (a) Random 2 node network with Jacobian elements labeled. Green arrows are basal production and external stimulus terms. **(b,c,d,e)** Fraction of network parameters correctly classified in 50 randomly generated 2 node networks with an absolute value between 0 to 0.5 (b), 0.5 to 1 (c), 1 to 1.5 (d) and 1.5 to 2 (e).

Figure S2: (a) Random 3 node network with Jacobian elements labeled. Green arrows are basal production and external stimulus terms. **(b,c,d,e)** Fraction of network parameters correctly classified in 50 randomly generated 3 node networks with an absolute value between 0 to 0.5 (b), 0.5 to 1 (c), 1 to 1.5 (d) and 1.5 to 2 (e).

Figure S3: (a,b) Fraction of network parameters correctly classified in 29 two node networks (a) and 26 three node networks (b) with potential for oscillatory behavior (non-zero imaginary parts of eigenvalues of Jacobian elements).

Figure S4: Simulated, noise-free experimental data (dots) and model-generated fits (lines) for each model structure. Different perturbations—vehicle (P0), perturb x1 (P1), perturb x2 (P2), perturb x3 (P3)—are across the columns and different model structures (1-16) are down the rows. Each node (1-3) is a different color as indicated in the legend.

Figure S5: Detailed results from the Non-Linear models from Figure 3,4 depicting whether a model parameter was correctly (1) or incorrectly (0) predicted for each model structure (1-16) under each noise level. Correct predictions were classified based on the optimal percentile cutoffs identified for each noise level.

Methods

Deriving Sufficiency Conditions for Unique Estimation of Jacobian Elements

The first-order partial derivatives comprising **J** (Eq. 2) can be approximated by a first-order Taylor series expansion of Eq. 1 about a time point *k*

$$f_1(k+1) \approx f_1(k) + \frac{\partial}{\partial x_1}\left(f_1(k)\right).\left(x_1(k+1) - x_1(k)\right) + \frac{\partial}{\partial x_2}\left(f_1(k)\right)\cdot\left(x_2(k+1) - x_2(k)\right) \qquad (6)$$

$$f_2(k+1) \approx f_2(k) + \frac{\partial}{\partial x_1}\left(f_2(k)\right)\cdot\left(x_1(k+1) - x_1(k)\right) + \frac{\partial}{\partial x_2}\left(f_2(k)\right)\cdot\left(x_2(k+1) - x_2(k)\right) \qquad (7)$$

Eq. 6-7 may be written more succinctly as

$$\begin{aligned} y_1(k+1) &\approx F_{11}(k)\cdot\Delta_t x_1(k+1) + F_{12}(k)\cdot\Delta_t x_2(k+1) \\ y_2(k+1) &\approx F_{21}(k)\cdot\Delta_t x_1(k+1) + F_{22}(k)\cdot\Delta_t x_2(k+1) \end{aligned} \qquad (8)$$

where

$$y_i(k+1) \equiv f_i(k+1) - f_i(k); \Delta_t x_i(k+1) \equiv x_i(k+1) - x_i(k)\,. \qquad (9)$$

The approximation in Eq. 8 becomes more accurate as more time points are measured. Also, the edge weights are potentially time-dependent, although this is rarely considered when describing biological networks.

How do we estimate the edge weights in Eq. 8 and thus reconstruct the network? Time series data can inform x_i's and f_i's as a function of time, following application of a stimulus. Given such stimulus-response data, however, for each time point there are only two equations for four unknowns, an underdetermined system for which more data are needed.

Consider now stimulus-response time course data in the presence of single perturbations. Let p_i be a variable that reflects the strength and/or presence of different potential perturbations:

p_1 represents inhibition of x_1 and p_2 represents inhibition of x_2. If p_j is not explicitly written, its value is zero and/or it has no effect. Now, the ODEs also become a function of the perturbation variables

$$f_{i,j}(k) \equiv f_i(k, p_j) = f_i(x_1(k), x_2(k), p_j) \qquad (10)$$

The 1st order Taylor series expansions for cases with perturbations become

$$y_{1,1}(k) \approx F_{11}(k) \cdot \Delta_{p,1} x_1(k) + F_{12}(k) \cdot \Delta_{p,1} x_2(k) + \frac{\partial}{\partial p_1}\left(f_1(k)\right) \cdot p_1 \qquad (51)$$

$$y_{1,2}(k) \approx F_{11}(k) \cdot \Delta_{p,2} x_1(k) + F_{12}(k) \cdot \Delta_{p,2} x_2(k) + \frac{\partial}{\partial p_2}\left(f_1(k)\right) \cdot p_2 \qquad (12)$$

$$y_{2,1}(k) \approx F_{21}(k) \cdot \Delta_{p,1} x_1(k) + F_{22}(k) \cdot \Delta_{p,1} x_2(k) + \frac{\partial}{\partial p_1}\left(f_2(k)\right) \cdot p_1 \qquad (13)$$

$$y_{2,2}(k) \approx F_{21}(k) \cdot \Delta_{p,2} x_1(k) + F_{22}(k) \cdot \Delta_{p,2} x_2(k) + \frac{\partial}{\partial p_2}\left(f_2(k)\right) \cdot p_2 \qquad (14)$$

where

$$y_{i,j}(k) \equiv f_{i,j}(k) - f_i(k); \Delta_{p,j} x_i(k) \equiv x_i(k, p_j) - x_i(k) \quad (15)$$

Here, we have expanded with respect to the perturbation, rather than with respect to time as previously. However, since the reference point is the same, the Jacobian elements remain identical in these equations. Thus, now we have six potential equations with which to estimate the four Jacobian elements. It is also interesting to note that the Jacobian elements, or network, may be affected by the perturbation, but we do not necessarily have to know those effects mathematically, since the reference point is the same. However, we must make some

determination as to how the perturbations p_1 and p_2 directly affect Node 1 and Node 2 dynamics f_1 and f_2 to account for the perturbation variable partial derivatives.

By design, the Node 1 perturbation has significant *direct* effects on Node 1 dynamics, and similarly for the Node 2 perturbation on Node 2 dynamics. Using equations including $\partial f_1/\partial p_1$ and $\partial f_2/\partial p_2$ require precise definition of perturbation strength and their effects on dynamics, which could be difficult to determine experimentally and implement in simulations. Moreover, it involves using perturbation data that is likely to be far away from the reference point, which makes the first-order Taylor series approximation less accurate. Therefore, we do not employ equations involving such terms. On the other hand, if the Node 1 perturbation has negligible *direct* effect on Node 2 dynamics, that is, the effects on Node 2 dynamics are through the network (i.e. p_1) is not explicit in f_2), and similarly the Node 2 perturbation has negligible *direct* effect on Node 1 dynamics, then $\partial f_2/\partial p_1$ and $\partial f_1/\partial p_2$ can be approximated as equal to zero in the above equations. This mild condition is often the case experimentally. The only determining factor for the suitability of the Taylor series truncation is the spacing of time points, and the effects of a perturbation *through* the network, not on its target node. From this, the main set of linear equations presented in Eq. 3-4 are obtained.

General Estimation Model Equations

We employ the following general model for a two-node network: -

$$\frac{dx_1}{dt} = f_1(x_1, x_2) = S_1 + F_{11}x_1 + F_{12}x_2$$
$$\frac{dx_2}{dt} = f_2(x_1, x_2) = S_2 + F_{21}x_1 + F_{22}x_2 \qquad (16)$$

Here, S_1 and S_2 are the stimuli strengths on Node 1 and Node 2 respectively, and F_{11}, F_{12}, F_{21} and F_{22} are the network edge weights (Figure 1a). In many systems, there may be a basal or constitutive production driving the node activities, besides an external stimulus. For these cases, the Stimulus term (S_i), may be considered as an addition of these two effects- the basal production term ($S_{i,b}$) and the external stimulus ($S_{i,ex}$). Then the two-node model can be represented by the following equations-

$$\frac{dx_1}{dt} = S_{1,b} + S_{1,ex} + F_{11}x_1 + F_{12}x_2$$
$$\frac{dx_2}{dt} = S_{2,b} + S_{2,ex} + F_{21}x_1 + F_{22}x_2 \qquad (17)$$

Or more generally,

$$\frac{dx_i}{dt} = S_{i,b} + S_{i,ex} + \sum_{j=1}^{n} F_{ij}x_j \ , \qquad (18)$$

where n is the total number of nodes.

When a steady state exists, the dx_i/dt terms become zero and it becomes easy to represent the stimulus terms as a function of the node activities (x_i) and network edges (F_{ij}).

$$S_{i,b} + S_{i,ex} = -(\sum_{j=1}^{n} F_{ij}x_{i,ss}) \qquad (19)$$

This is helpful in understanding that the perturbation time course data also generally constrain not only the edge weights, but also the stimulus terms. For a system at a steady state without an external stimulus, for example at t=0:

$$S_{i,b} = -(\sum_{j=1}^{n} F_{ij} x_{i,ss}) \qquad (20)$$

The Two-node Single Activator model

The two-node single activator model (Fig. 1a, S1a) is described by

$$\begin{aligned} \frac{dx_1}{dt} &= f_1(x_1, x_2) = 1 - x_1 \\ \frac{dx_2}{dt} &= f_2(x_1, x_2) = 1 + 1.5x_1 - 0.8x_2 \end{aligned} \qquad (21)$$

Here, $S_{1,ex}=1$, $F_{11}=-1$, $F_{12}=0$, $S_{2,ex}=1$, $F_{21}=1.5$, $F_{22}=-0.8$. The basal production terms are both zero, for simplicity, and the initial conditions for $x_1(t=0)$ and $x_2(t=0)$ are zero. The stimulus terms $S_{i,\,ex}$ are calculated through Eq. 19, using the median values of F_{ij} and the $x_i(t=10)$, when the system reaches near steady state.

Random Two-node and Three-node models

The random 2 node network is described by

$$\begin{aligned} \frac{dx_1}{dt} &= f_1(x_1, x_2) = S_{1,b} + S_{1,ex} + F_{11}x_1 + F_{12}x_2 \\ \frac{dx_2}{dt} &= f_2(x_1, x_2) = S_{2,b} + S_{2,ex} + F_{21}x_1 + F_{22}x_2 \end{aligned} \qquad (22)$$

Values for $S_{1,b}$, $S_{2,b}$, $S_{1,ex}$ and $S_{2,ex}$ are sampled from a uniform distribution over the range [0,2] and values for F_{11}, F_{12}, F_{21}, and F_{22} are sampled from a uniform distribution over the range [-2,2] using the MATLAB function `rand`. To capture basal activity, we use a two-step approach. First, starting from node activity values of zero, without the external stimulus on Node 1 and Node 2 ($S_{1,ex}=S_{2,ex}=0$ in Eq 22) we simulate until the network reaches steady-state with just basal production driving the network behavior. Then, we introduce the external stimulus on

Node 1 and Node 2, integrate the ODEs, and sample evenly spaced time-points using `ode15s` in MATLAB with default settings. We sample 3,7, 11, and 21 evenly spaced time points across a time course, from 0 to 10 arbitrary time units in all the cases.

The random 3 node networks use the same sampling rules as the 2 node networks with the following equations.

$$\begin{aligned}
\frac{dx_1}{dt} &= f_1(x_1, x_2, x_3) = S_{1,b} + S_{1,ex} + F_{11}x_1 + F_{12}x_2 + F_{13}x_3 \\
\frac{dx_2}{dt} &= f_2(x_1, x_2, x_3) = S_{2,b} + S_{2,ex} + F_{21}x_1 + F_{22}x_2 + F_{23}x_3 \\
\frac{dx_3}{dt} &= f_3(x_1, x_2, x_3) = S_{3,b} + S_{3,ex} + F_{31}x_1 + F_{32}x_2 + F_{33}x_3
\end{aligned} \tag{23}$$

Non-Linear Models

The non-linear feedforward loop models (Mangan and Alon, 2003) are described by:

$$\begin{aligned}
\frac{dx_1}{dt} &= f_1(x_1, x_2, x_3) = 1 - x_1 \\
\frac{dx_2}{dt} &= f_2(x_1, x_2, x_3) = f(x_1, K_{x_1x_2}) - x_2 \\
\frac{dx_3}{dt} &= f_3(x_1, x_2, x_3) = G(x_1, K_{x_1x_3}, x_2, .K_{x_2x_3}) - x_3
\end{aligned} \tag{24}$$

When an AND gate is present

$$G(x_1, K_{x_1x_3}, x_2, K_{x_2x_3}) = f(x_1, K_{x_1x_3}) * f(x_2, K_{x_2x_3}) \tag{25}$$

When an OR gate is present

$$G(x_1, K_{x_1x_3}, x_2, K_{x_2x_3}) = fc(x_1, K_{x_1x_3}, K_{x_2x_3}, x_2) + fc(x_2, K_{x_2x_3}, K_{x_1x_3}, x_1) \tag{26}$$

For a given u, v □ {x_1, x_2, x_3} and K, K_u, K_v □ {$K_{x_1x_2}$, $K_{x_1x_3}$, $K_{x_2x_3}$}:

If u activates its target, then:

$$f(u,K)=\frac{\left(\frac{u}{K}\right)^2}{1+\left(\frac{u}{K}\right)^2}\ ;\ fc(u,K_u,K_v,v)=\frac{\left(\frac{u}{K_u}\right)^2}{1+\left(\frac{u}{K_u}\right)^2+\left(\frac{v}{K_v}\right)^2} \qquad (27)$$

If u represses its target, then:

$$f(u,K)=\frac{1}{1+\left(\frac{u}{K}\right)^2}\ ;\ fc(u,K_u,K_v,v)=\frac{1}{1+\left(\frac{u}{K_u}\right)^2+\left(\frac{v}{K_v}\right)^2} \qquad (28)$$

Effectively, an external stimulus of '$S_{1,ex}=1$', acts on Node 1 at $t=0$ and is propagated through the network. There is no external stimulus acting on Node 2 and Node 3. However, in many cases there is basal production in one or both of Node 2 and Node 3. This leads to a non-zero steady state of the network, when there is a basal production, before the external stimulus is introduced.

To capture basal activity, we use a two-step approach. First, starting from node activity values of zero, without the external stimulus on Node 1 ($S_{1,ex}=0$), we simulate until the network reaches steady-state. Then, we introduce the external stimulus on Node 1, integrate the ODEs, and sample 11 evenly spaced time-points using `ode15s` in MATLAB with default settings and steady- state node values without the external stimulus as the initial conditions. We chose 11 timepoints because it yields good classification accuracy for the above random 3 node model even in presence of noisy data. For each of the 16 non-linear models, the values of the parameters (*K, Ku, Kv*), were varied and chosen so that the resulting node activity data are responsive to the stimulus and perturbations (Fig. S4, See Supplementary Code for values).

Modeling Perturbations

Precisely modeling perturbations can be a challenge, since experimentally, there may be several ways of causing a perturbation with different mechanisms such as siRNAs, competitive/non-competitive/uncompetitive inhibition, etc. It may be hard to quantify how much a perturbation is affecting a node, in terms of its dynamics (i.e. right hand sides of the ODEs). Therefore, we employ the following approaches which circumvent the need to model how each perturbation mechanistically manifests in the ODEs during parameter estimation. There are two cases to consider: (i) when we have a perturbation of node *i* and we need to simulate node *i* dynamics; (ii) when we have a perturbation of node *i* and we need to simulate other node *j* dynamics. To illustrate the approach, we use the above-described 2 node model with an example of a Node 1 perturbation. Recall that

$$\begin{aligned} \frac{dx_1}{dt} &= S_{1,b} + S_{1,ex} + F_{11}x_1 + F_{12}x_2 \\ \frac{dx_2}{dt} &= S_{2,b} + S_{1,ex} + F_{21}x_1 + F_{22}x_2 \end{aligned} \qquad (29)$$

For case (i), we have to obtain values for x_1 under perturbation of Node 1. We refer to the perturbed time-course as $x_{1,1}$. In experimental situations, $x_{1,1}$ would be measured directly. To obtain simulation data for $x_{1,1}$ we use the following:

$$x_{1,1}(k) = p_1 \times x_1(k), \qquad (30)$$

where x_1 is obtained from the simulations without perturbations, and recall that k refers to time point k. For a 50% inhibition, p=0.5 and for a complete inhibition, p=0.

For case (ii), we have to obtain the values for x_2 under perturbation of Node 1, which we refer to as $x_{2,1}$. To do this, we have to integrate the ODE for dx_2/dt, but using $x_{1,1}$ values, as follows

$$\frac{dx_{2,1}}{dt} = S_{2,b} + S_{2,ex} + F_{21}x_{1,1}(k) + F_{22}x_{2,1} \qquad (31)$$

Here, x_2 has been replaced with $x_{2,1}$ for clarity. To solve this equation, we simply use the "measured" $x_{1,1}$ time course directly in the ODE.

When data are generated by simulations, there is little practical limit to temporal resolution, but with real data, to solve Eq. 31 one will need values for $x_{1,1}$ at multiple time points where measurements are not available. We therefore fit $x_{1,1}$ dynamics to a polynomial using `polyfit` in MATLAB, and use the polynomial to interpolate given a required time point. In this work, we have used order one less than the number of time points, but the functional form is quite malleable so long as it captures the data trends. The approach is general and scales to three and more nodes.

Simulated Noise

Normally distributed white (zero mean) noise is added to simulated time courses point-wise with

$$y = x + N(0, d \cdot x) \qquad (32)$$

where x is the simulation data point, y is the noisy data point, and d represents the noise level. Signal-to-noise ratio of 10:1, 5:1 and 2:1 are, respectively d = 0.1, 0.2, and 0.5. Normally distributed samples are obtained using `randn` in MATLAB.

Parameter Estimation

For the two-node model, the entire network, with and without perturbations, can be explained by the following system of equations

$$\begin{aligned}
\frac{dx_1}{dt} &= S_{1,b} + S_{1,ex} + F_{11}x_1 + F_{12}x_2 \\
\frac{dx_2}{dt} &= S_{2,b} + S_{2,ex} + F_{21}x_1 + F_{22}x_2 \\
\frac{dx_{2,1}}{dt} &= S_{2,b} + S_{2,ex} + F_{21}x_{1,1} + F_{22}x_{2,1} \\
\frac{dx_{1,2}}{dt} &= S_{1,b} + S_{1,ex} + F_{11}x_{1,2} + F_{12}x_{2,2}
\end{aligned} \tag{33}$$

where $x_{1,1}$ and $x_{2,2}$ are the perturbed node values, from either simulated or experimental data. Eight parameters ($S_{1,b}$, $S_{1,ex}$, F_{11}, F_{12} , $S_{2,b}$, $S_{2,ex}$, F_{21}, F_{22}) need to be estimated to fully reconstruct this network. We seek a set of parameters that minimizes deviation between simulated and measured dynamics.

For an initial guess, the node edge parameters (F_{ij}) are randomly sampled from a uniform distribution over the interval [-2,2] and the stimulus parameters ($S_{i,ex}$) are sampled from a uniform distribution over the interval [0,2]. Using data at $t=0$, which corresponds to a steady state without $S_{i,ex}$, the $S_{i,b}$ can be estimated during each iteration of the estimation as follows-

$$\begin{aligned}
\hat{S}_{1,b} &= -(\hat{F}_{11}x_1(t=0) + \hat{F}_{12}x_2(t=0)) \\
\hat{S}_{2,b} &= -(\hat{F}_{21}x_1(t=0) + \hat{F}_{22}x_2(t=0))
\end{aligned} \tag{34}$$

For an n-node model, this equation can be scaled accordingly to obtain each $\hat{S}_{i,b}$.

For these initial guesses we compute the activity data using Eq. 33. The perturbation data $x_{k,k}$ is used in the perturbation equations as detailed above (Eq. 31). Let $\hat{x}_i$, and $\hat{x}_{i,j}$ denote the

predicted nodal values for non-perturbed and perturbed cases respectively. For a total of n nodes and N_t timepoints, the objective function is the sum of squared errors Φ

$$\Phi = \sum_{k=1}^{N_t}\left[\left(\sum_{i=1}^{n}\left(x_i(k)-\hat{x}_i(k)\right)^2\right)+\sum_{i=1}^{n}\sum_{j\neq i}\left(x_{i,j}(k)-\hat{x}_{i,j}(k)\right)^2\right] \qquad (35)$$

Note here that we do not use data from node j, when perturbation j was used (per the derivation). The MATLAB function `fmincon` is used to minimize Φ by changing edge weights and stimulus terms within the range [-10,10].

We employ "multi-start" by running the estimation 10 times, starting from different randomly generated initial starting points (Raue et al., 2013). The estimated parameters and their respective final sum of squared errors (Φ) are saved and the estimated parameter set corresponding to the minimum Φ is chosen as the final parameter set.

Parameter Estimation for Non-Linear Models

For estimating the Non-Linear models, we start with a prior knowledge that $S_{1,b}$ is always *zero* and $S_{2,ex}$ and $S_{3,ex}$ are always zero as well, which is directly evident from x_1 initial conditions and x_2, x_3 stimulus response in the presence of a complete Node 1 perturbation. The equations for the non-perturbation case become as follows

$$\begin{aligned}
\frac{dx_1}{dt} &= S_{1,ex} + F_{11}x_1 + F_{12}x_2 + F_{13}x_3 \\
\frac{dx_2}{dt} &= S_{2,b} + F_{21}x_1 + F_{22}x_2 + F_{23}x_3 \qquad (36) \\
\frac{dx_3}{dt} &= S_{3,b} + F_{31}x_1 + F_{32}x_2 + F_{33}x_3
\end{aligned}$$

Since the system is at steady-state before the external stimulus, the basal production parameter can be estimated during each iteration of the estimation as-

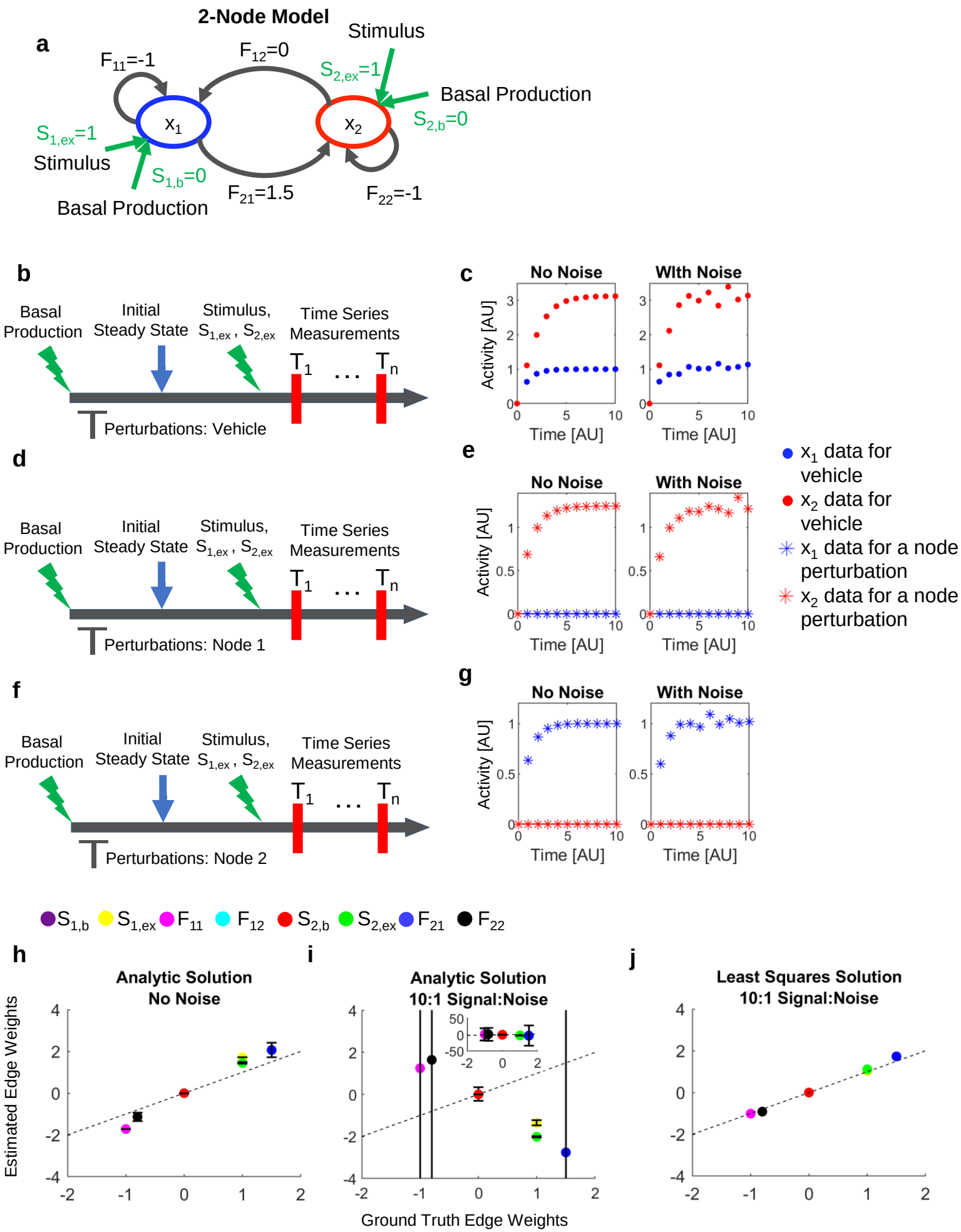
$$\hat{S}_{2,b} = -(\hat{F}_{21}x_1(t=0) + \hat{F}_{22}x_2(t=0) + \hat{F}_{23}x_3(t=0))$$
$$\hat{S}_{3,b} = -(\hat{F}_{31}x_1(t=0) + \hat{F}_{32}x_2(t=0) + \hat{F}_{33}x_3(t=0)) \quad (37)$$

where $\hat{F}_{i,j}$ are the current model parameter estimates and $x_i\,(t=0)$ are the x values at the initial system steady state before the induction of external stimulus.

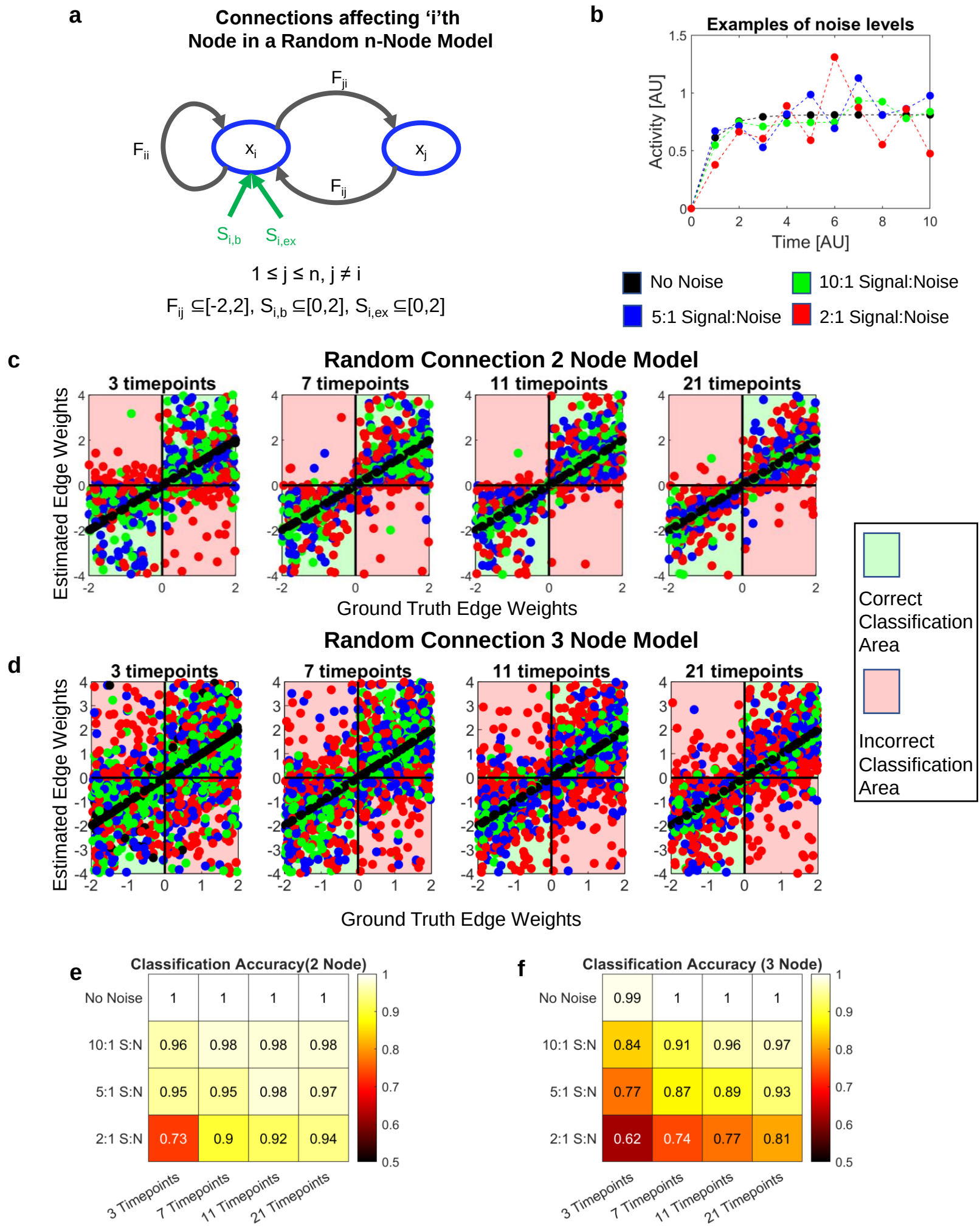
Bootstrapping Simulated Data for the Non-Linear Model Cases

To generate multiple parameter set estimates to classify edge weights for the non-linear model cases, we employ a bootstrapping approach. In an experimental scenario, each data point will have a mean and a standard deviation, and upon a distributional assumption (e.g. normal), one can then resample datasets to obtain measures of estimation uncertainty. We use the simulated data as the mean, and then vary the standard deviation as described above to generate 50 bootstrapped datasets for each of the 16 considered models. Estimation is carried out for each of the 50 datasets using multi-start, which yields 50 best-fitting parameter sets for each model. Uncertainty analysis and classification error is based on these sets.

Smith et al., 2020. Figure 1



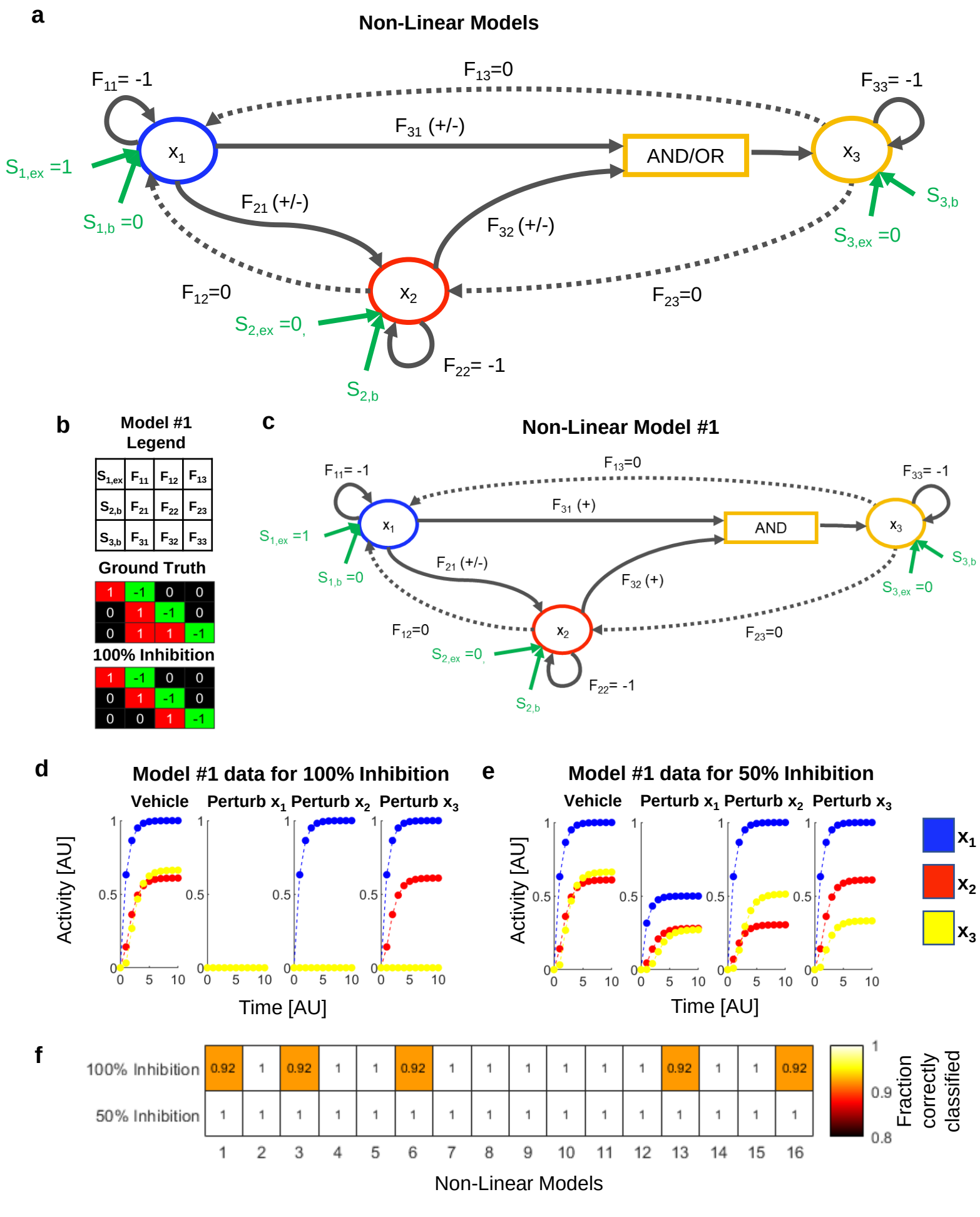
Smith et al., 2020. Figure 2



Model#	F_{21}	Gate	F_{31}	F_{32}
1	Activator	AND	Activator	Activator
2	Activator	AND	Activator	Repressor
3	Activator	AND	Repressor	Activator
4	Activator	AND	Repressor	Repressor
5	Activator	OR	Activator	Activator
6	Activator	OR	Activator	Repressor
7	Activator	OR	Repressor	Activator
8	Activator	OR	Repressor	Repressor
9	Repressor	AND	Activator	Activator
10	Repressor	AND	Activator	Repressor
11	Repressor	AND	Repressor	Activator
12	Repressor	AND	Repressor	Repressor
13	Repressor	OR	Activator	Activator
14	Repressor	OR	Activator	Repressor
15	Repressor	OR	Repressor	Activator
16	Repressor	OR	Repressor	Repressor

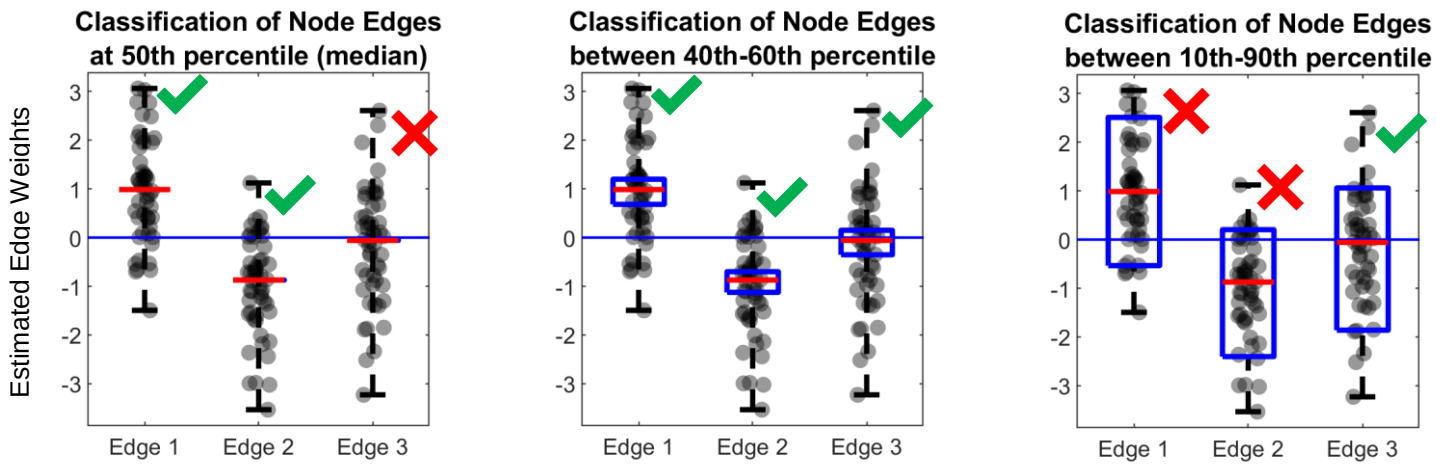
Table 1. Structure of each of the 16 non-linear models.

Smith et al., 2020. Figure 3



Smith et al., 2020. Figure 4

a

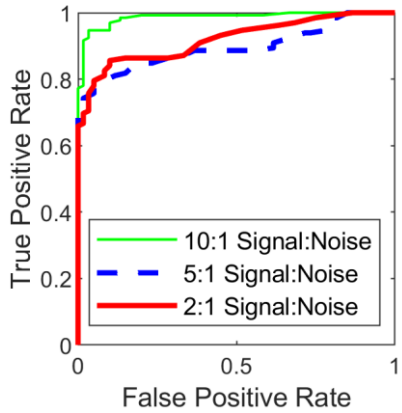


	Edge 1	Edge 2	Edge 3
Actual	+	-	0
Predicted	+	-	-

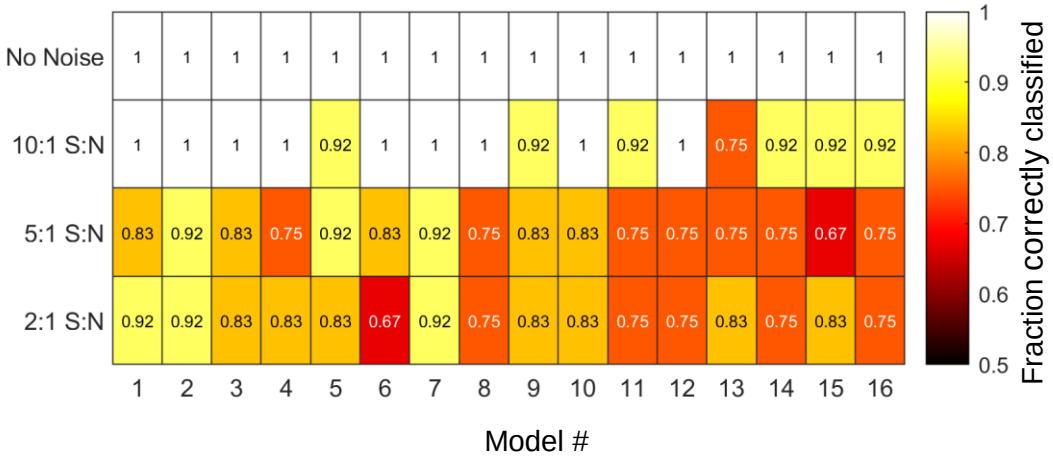
	Edge 1	Edge 2	Edge 3
Actual	+	-	0
Predicted	+	-	-

	Edge 1	Edge 2	Edge 3
Actual	+	-	0
Predicted	0	0	0

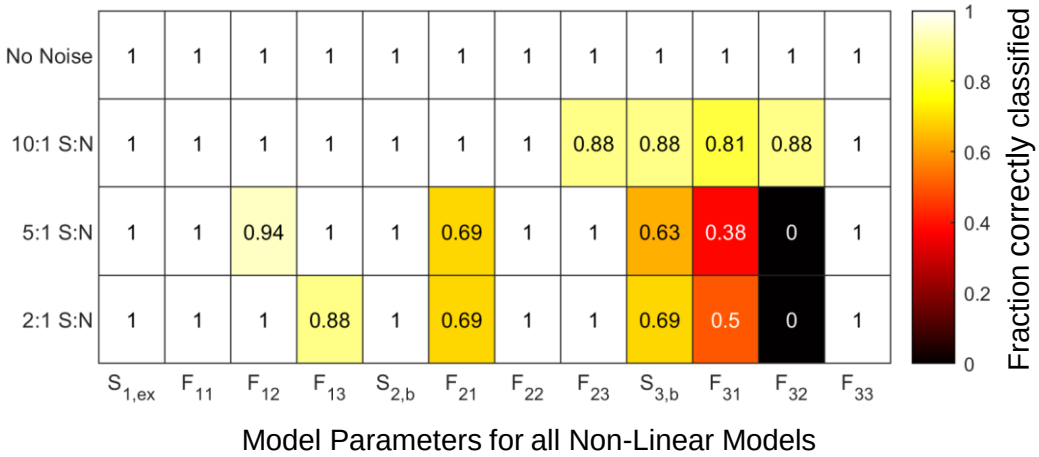
b



c

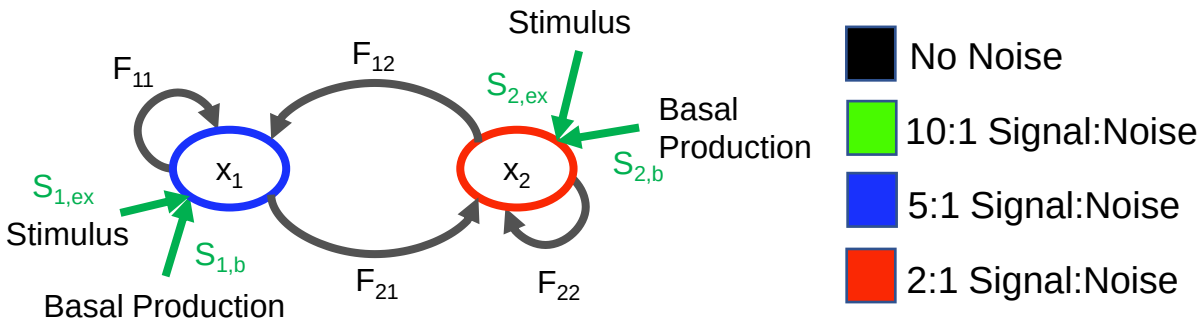


d



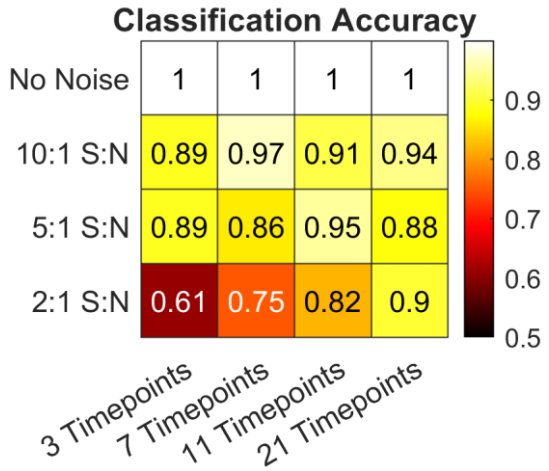
Smith et al., 2020. Supplementary Figure 1

a **Random Connection 2-Node Model**



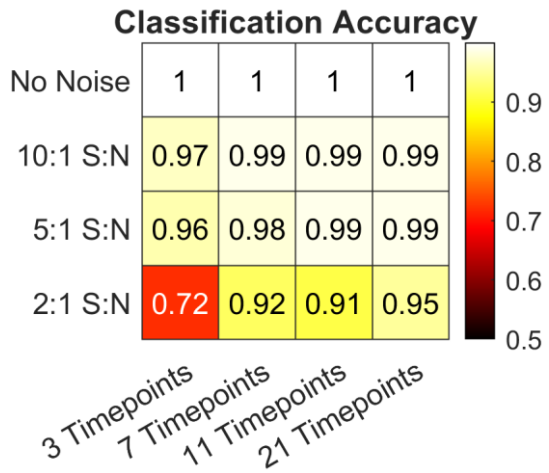
$F_{ij} \subseteq [-2,2]$, $S_{i,b} \subseteq [0,2]$, $S_{i,ex} \subseteq [0,2]$

b



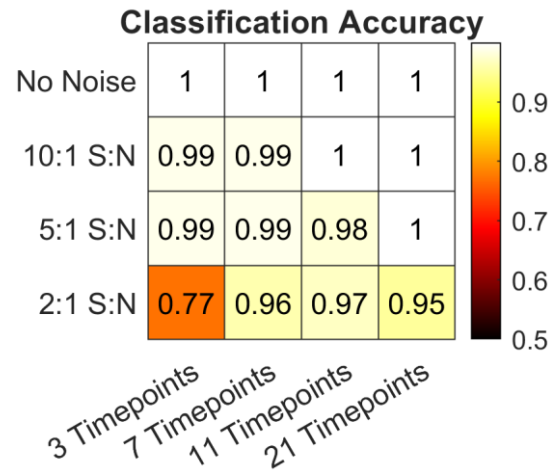
|Edge Weights| $\subseteq [0,0.5]$

c



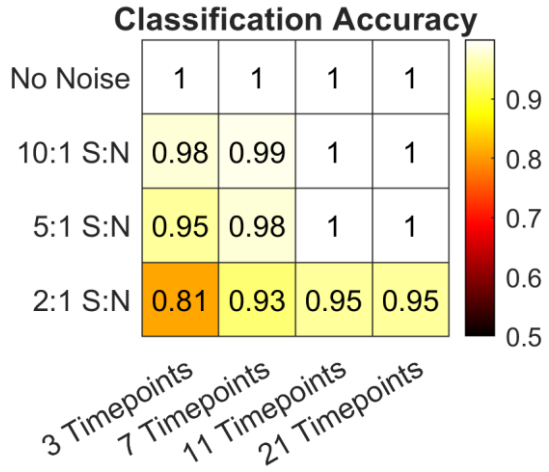
|Edge Weights| $\subseteq [0.5,1]$

d



|Edge Weights| $\subseteq [1,1.5]$

e

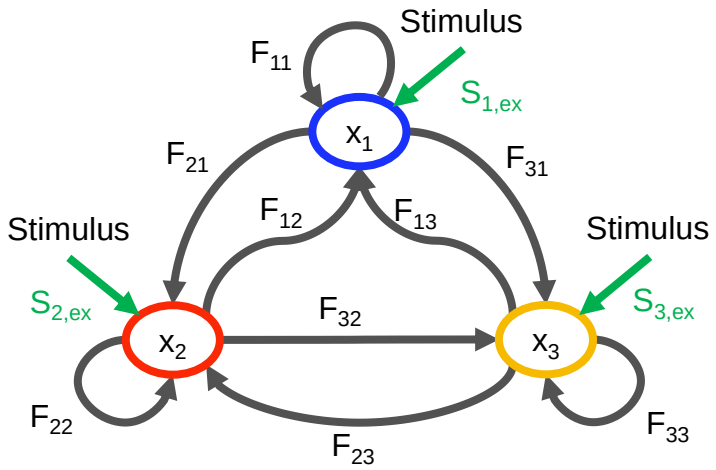


|Edge Weights| $\subseteq [1.5,2]$

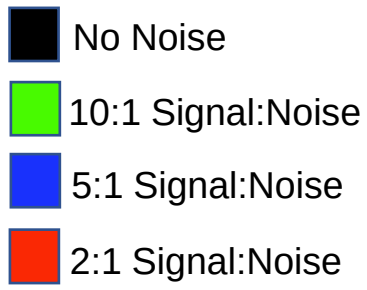
Smith et al., 2020. Supplementary Figure 2

a

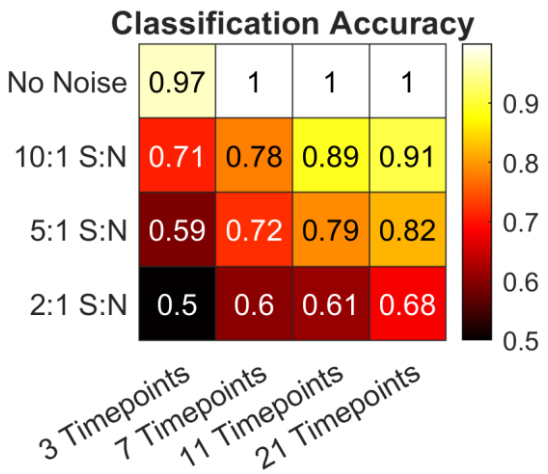
Random Connection 3-Node Model



$F_{ij} \subseteq [-2,2]$, $S_{i,b} \subseteq [0,2]$, $S_{i,ex} \subseteq [0,2]$

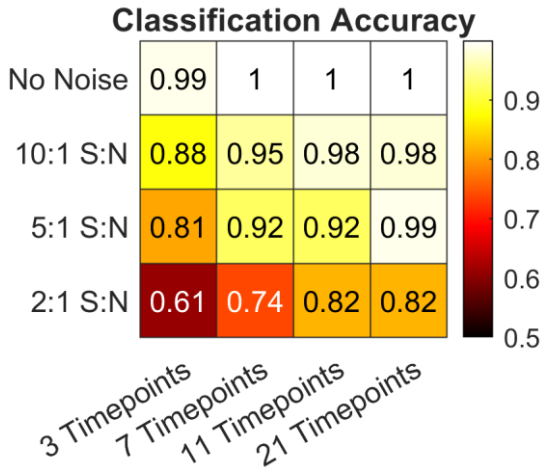


b



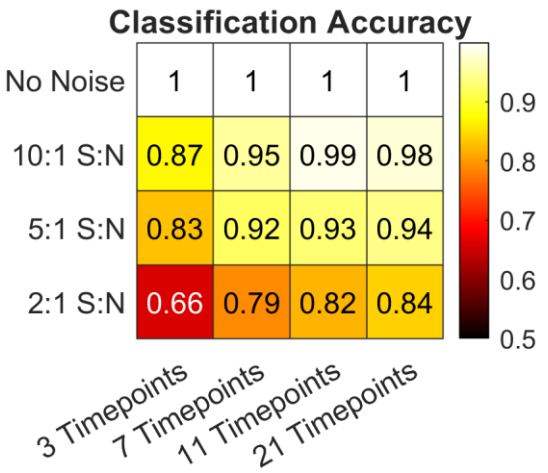
|Edge Weights| $\subseteq [0,0.5]$

c



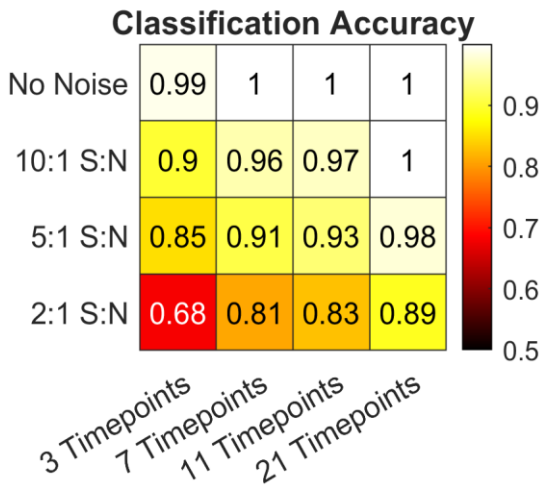
|Edge Weights| $\subseteq [0.5,1]$

d



|Edge Weights| $\subseteq [1,1.5]$

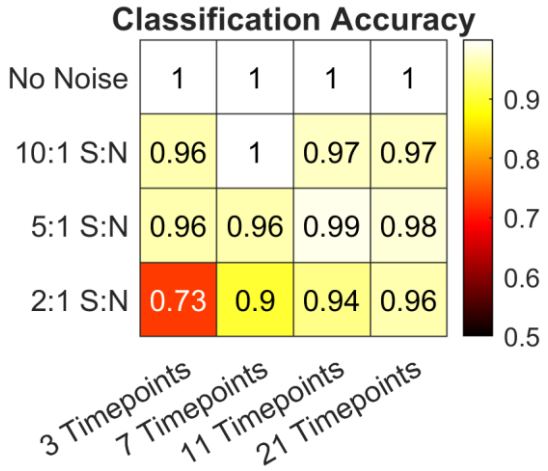
e



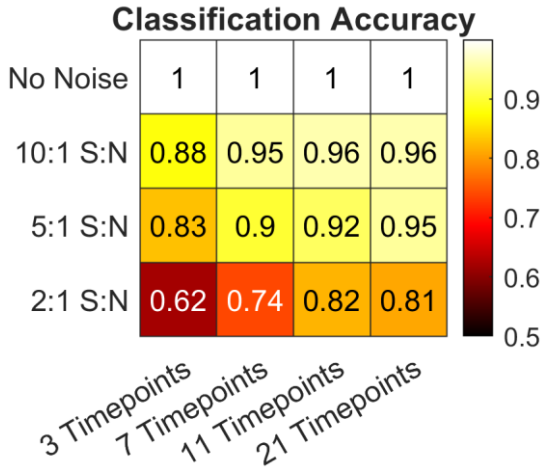
|Edge Weights| $\subseteq [1.5,2]$

Smith et al., 2020. Supplementary Figure 3

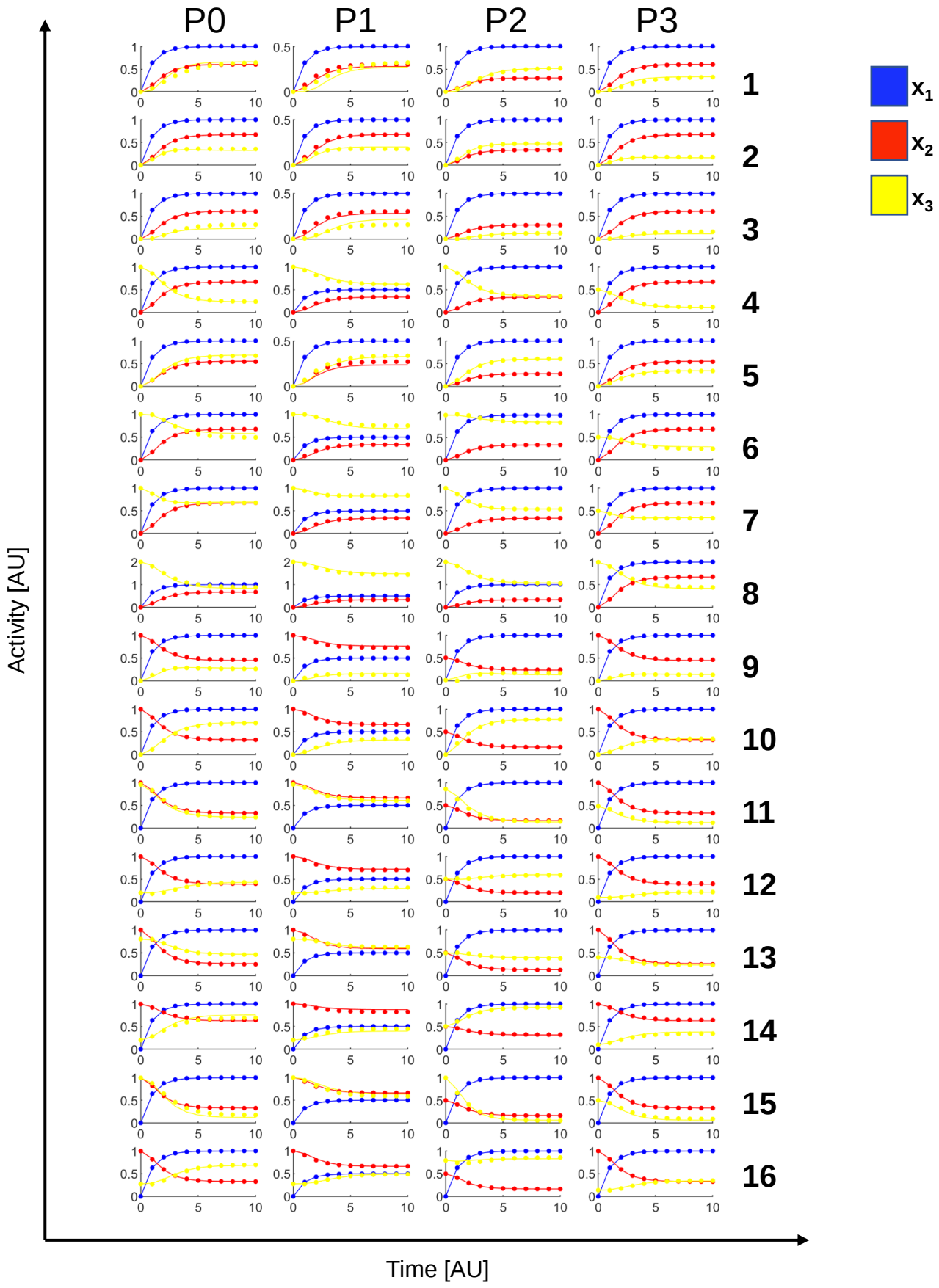
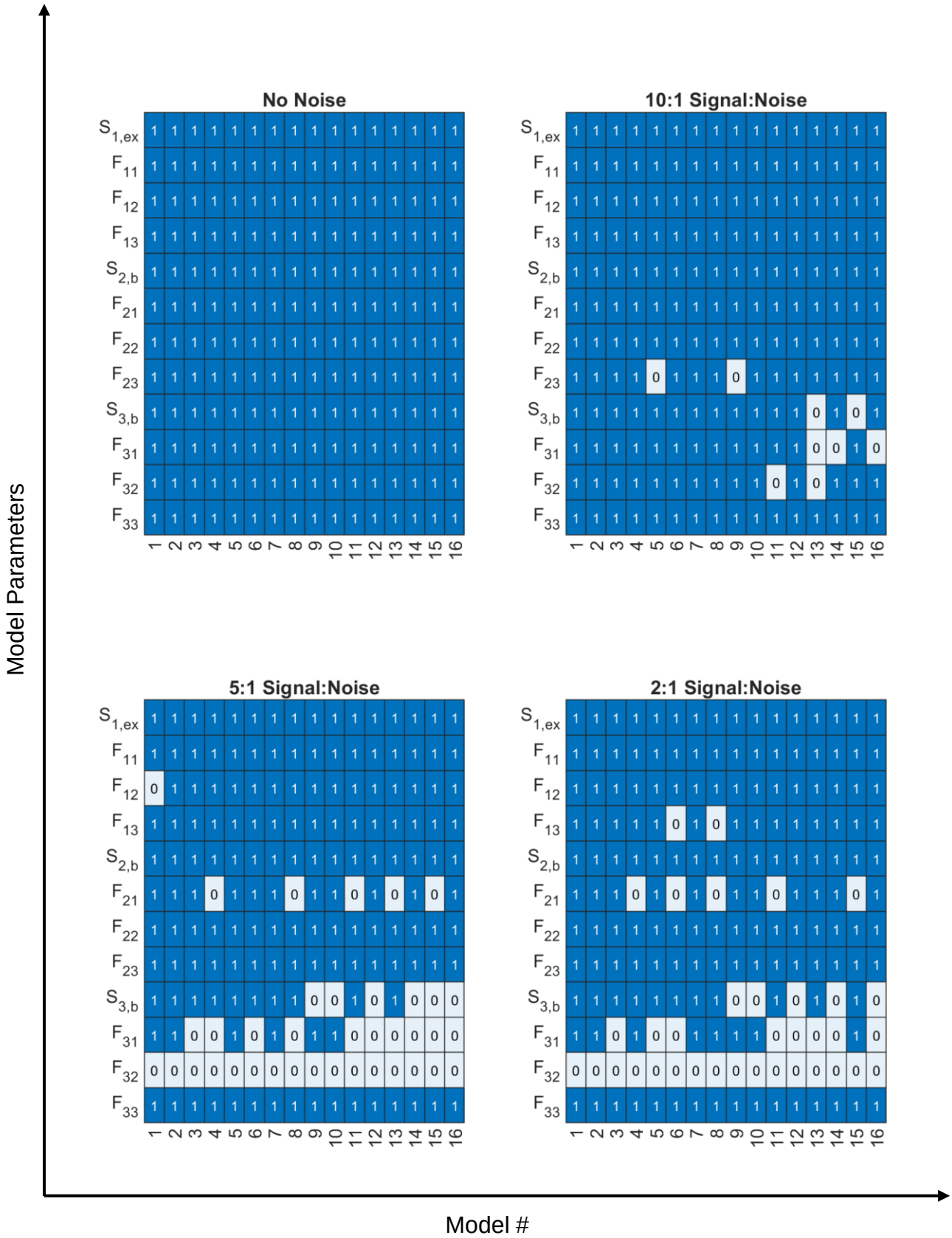
a **Random Connection 2-Node Models With Oscillatory behaviour**



b **Random Connection 3-Node Models With Oscillatory behaviour**



Smith et al., 2020. Supplementary Figure 4



Smith et al., 2020. Supplementary Figure 5