Journal of Computational Neuroscience 29: 231-252 (2010)

Noname manuscript No.
(will be inserted by the editor)

Causal Pattern Recovery from Neural Spike Train Data using the Snap Shot Score

Christoph Echtermeyer · **Tom V. Smulders** · **V. Anne Smith**

Received: date / Accepted: date

Abstract We present a new approach to learning directed information flow networks from multi-channel spike train data. A novel scoring function, the Snap Shot Score, is used to assess potential networks with respect to their quality of causal explanation for the data. Additionally, we suggest a generic concept of plausibility in order to assess network learning techniques under partial observability conditions. Examples demonstrate the assessment of networks with the Snap Shot Score, and neural network simulations show its performance in complex situations with partial observability. We discuss the application of the new score to real data and indicate how it can be modified to suit other neural data types.

Keywords neuronal assembly analysis · spike train · causal network · neural information flow

Christoph Echtermeyer
School of Biology, University of St Andrews, St Andrews, KY16 9TS, United Kingdom
Tel.: +44-(0)1334-463362
Fax: +44-(0)1334-463366
E-mail: ce86@st-andrews.ac.uk

Tom V. Smulders
Institute of Neuroscience, The Henry Wellcome Building for Neuroecology, Newcastle University, Newcastle upon Tyne, NE2 4HH, United Kingdom
Tel.: +44-(0)191-2225790
Fax: +44-(0)191-2225622
E-mail: tom.smulders@ncl.ac.uk

V. Anne Smith
School of Biology, University of St Andrews, St Andrews, KY16 9TS, United Kingdom
Tel.: +44-(0)1334-463368
Fax: +44-(0)1334-463366
E-mail: vas1@st-andrews.ac.uk

1 Introduction

Understanding the brain boils down to understanding its interplay on many different spatial and temporal scales. Therefore, fundamental knowledge about the nervous system's connectivity is crucial for deeper insights into its machinery. It is thus not surprising that significant efforts have been spent for the development of both anatomical and analytical approaches in order to reveal yet another jigsaw piece of brain circuitry.

Current technical advances in spatial and temporal resolution have led to an increasing amount of physiological data with many channels recorded simultaneously (e.g. [57, 35, 37, 76, 68]). The increasing dimensionality of the data raises the need for powerful computational tools to detect and extract critical features for further analysis. Our work joins the class of analysis tools for electrophysiological recordings - in particular multi-channel spike trains. The Snap Shot Score efficiently reveals effective connectivity (over a range of time-lags) between recorded units and thereby offers a new perspective on the data.

In general, collecting spike train data from a system is insufficient to reconstruct structural connectivity between units, as neurons are sampled with relatively low spatial density. Even if data could be collected from all neurons, their *functional connectivity*, i.e. their correlated activity, does still not convey enough information for an unambiguous decision on their causal interactions (Fig. 1). A network representing functional relationships is thus not an attempt to suggest structural connectivity, but a visualisation aid for data analysis. In order to be able to draw conclusions from such a network, its semantic must be defined by an underlying model. Different model types might favour different kinds of connectivity pattern; for example, tree-like (Fig. 1a), chain-like considering hidden units (Fig. 1b), or chain-like structures among observed units only (Fig. 1c). Choosing any pattern preference reduces ambiguity of good explanations (i.e. networks) for the data, such that they become more consistent. The model type also determines the interpretation of networks; for example, links might represent cause-effect relationships. Neuron models according to the integrate and fire paradigm [72, 1] contain such cause-effect relationships, which renders corresponding networks easily interpretable. Connectivity suggested by the model might differ from the actual anatomical connectivity and is thus called *effective connectivity* [25, 71]. The method introduced in this paper implicitly uses a causal model to reveal effective connectivity from spike train data.

To date, extra-cellular neural recordings can only be made from a fraction of neurons in the neural system, such that most neurons remain hidden units with respect to the recorded data. The predominant number of non-observed units affects the detection of functional and effective relations: Observed units, connected by unobserved chains, can show dependence over different time-lags depending on the length of the connecting chain. This is commonly taken into account by analysing relationships for a series of different time-lags; in contrast, the technique presented in this paper accounts for multiple time-lags simultaneously. Our method favours connectivity patterns for which responses are explained by short time-lags, but it will also suggest connections with larger time-lags if they explain the data significantly better. This reflects Occam's razor [48] as an accepted model selection criterion, when direct (short time-lag) connections are considered to be simpler than indirect (long time-lag) connections. The range of time-lags considered by our method can be adapted to suit the specific characteristics of the data.

Different analysis techniques for constructing neural information flow networks from neural electrophysiological data exist [9]. (See Table 1 for a classification of analysis methods.) These methods can be categorised according to how they use the data: as spike times or transformations to the frequency domain. Spike time methods utilise the full precision

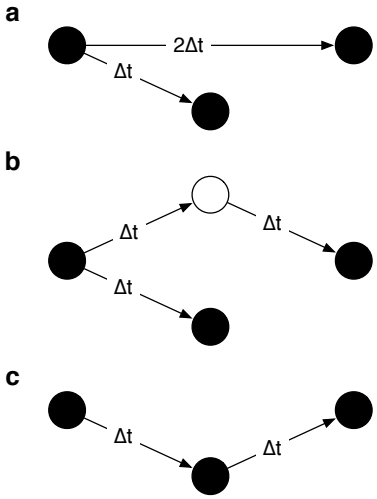


Fig. 1 Correlation does not imply causation: Networks visualising information flow between nodes from left to right with different time-lags. Filled nodes are observable while the white node cannot be observed. Mere correlation between observed nodes cannot distinguish between alternatives: **a** hub-node with connections of different time-lags, **b** homogenous time-lags involving non-observed nodes, **c** chain structure among observed units.

of spike train data, while frequency methods smooth out minor variations. Some methods of either category average over multiple trials (and thus require several repeated recordings under same conditions), while others can be applied to single trial recordings. Additionally, techniques differ in whether or not they require spike trains of individual neurons. Methods fitting neuron models to the data require single unit data (i.e. all spikes on one channel originate from exactly one neuron), whereas other approaches can handle non-spike sorted multi-unit data (with outputs of multiple neurons per channel). Some methods are restricted to the analysis between pairs of channels and might thus fail to pick up relationships that are conditional on more than one channel; multivariate analysis techniques can detect these complex patterns. Our new method aims to combine the advantages of both spike time and frequency domain methods, by using a mixture of these concepts. It can reveal multivariate relationships and is applicable to both single and multi-unit spike train data without requiring multiple trial recordings.

2 The Snap Shot Score

In analogy to Hebbian learning [33, p.62], revealing an information flow network can be summarised as: *cells that fire together, shall wire together*. Discovering, or *learning*, such a network with our method requires the assessment of many potential networks using a scoring function, which assigns a high value to networks that give a good causal explanation of the data and a low value otherwise. Links in the learned network represent an excitatory influence of the starting node on the destination node. If the number of potential networks is sufficiently small, all of them can be evaluated, but in general too many networks exist to be evaluated exhaustively. In such cases, search heuristics or Monte Carlo Markov Chain (MCMC) methods can be used to select a subset of promising networks to score. Any

Table 1 Classification overview of techniques that can be used for inference of neural information flow networks from electrophysiological data. Legend: ■ required/intended use, □ possible use, n/a not applicable.

method (based on)	spike time /frequency	multiple trials	single-unit	multi-unit	analysis type
Joint Peristimuls Time Histogram (JPSTH) [27,2]	time	■	■	□	pairwise
Cross-Correlation [60]	frequency[a]	■	■	■	pairwise
Information Theory [65,8,18]	frequency[a]	■	■	□	multivariate
single neuron model [56,49]	time	□	■	n/a	multivariate
Gravity [28,26,47]	time	□	■	□	multivariate
Dynamic Bayesian Network (DBN) [70]	frequency (Hz)	□	n/a	■	multivariate
Partial Directed Coherence (PDC) [67,6,5,74]	frequency (Hz)	□	■	■	multivariate
Generalised Linear Model (GLM) [14,58,75,62]	time	□	■	■	multivariate
Granger causality [31,11]	frequency (Hz)	□	□	■	multivariate
Direct Transfer Function (DTF) [39,20]	frequency (Hz)	□	□	■	multivariate
Snap Shot Score (SSS)	both	□	■	□	multivariate

[a] Frequency meant in terms of a frequentist's probability estimate [17]. Sufficient amounts of data are required for these estimates, although not necessarily multiple identical trials.

generic search method that operates on a discrete search space is suitable for our method. We refer the reader to the manifold literature on such methods: greedy search [16]; evolutionary algorithms [7,4], such as genetic algorithms [78] or particle swarm optimisation [40,19]; and the large class of MCMC methods: Metropolis-Hastings [51,32], simulated annealing [43,13], and Gibbs sampling [12,66], for example.

The core of our method, the Snap Shot Score (SSS), can account for relations over multiple time-lags. This is achieved by converting each spike train with a low-pass filter to an activity level series. All spike times are preserved in the activity level series, and additionally – in inter spike intervals – it is enriched by information about past neural activity. The actual score values are then calculated using both the spike trains and activity level series, by taking snapshots of the activity level at all spike times. In order to account for multivariate relationships, i.e. situations in which an effect has more than one cause, multiple activity level series need to be joined before calculating the score. The mathematical description of our method follows next, followed by interpretations and examples.

We consider spike trains of n channels being given by the n-dimensional time series $s = (s_{k,t})_{t=1,\ldots,T}^{k=1,\ldots,n}$ with $s_{k,t} = 1$ if a spike was detected at time t on channel k and $s_{k,t} = 0$

otherwise. We define corresponding *activity level series* $a = (a_{k,t})_{t=1,\dots,T}^{k=1,\dots,n}$ with

$$a_{k,t} = \max_{j=0,\dots,t-1} s_{k,t-j} - j \cdot d \tag{1}$$

for some *decay constant* $d \in [0,1]$. Throughout most of this paper we consider $d = 3^{-1}$ such that the activity level of channel k at time t is determined by $s_{k,t}, s_{k,t-1}$, and $s_{k,t-2}$, only.[1] The further a spike occurred in the past, the less influence it has on the activity level. This is because the potential weight of a spike $1 - j \cdot d$ decreases as j, the number of time-lags into the past, increases. Spikes in the more recent past (i.e. a smaller j) or the present ($j = 0$) have higher weight and thus supersede any spikes that occurred earlier. This is due to the maximum taken in equation (1). Any spike further in the past (from time t) than $\lceil 1/d \rceil$, the number of time-bins it takes for activity to fully decay, does not contribute at all to the activity level series (at time t): Such spikes result in a negative value for $1 - j \cdot d$; thus, any subsequent spike, or even current silence (0), would be selected by the maximum.

The *joined activity level series* $a_{(k_1,\dots,k_m)}$ of channels $k_1, \dots, k_m$ is defined as the maximum over channels for each time t:

$$a_{(k_1,\dots,k_m),t} = \max_{j=1,\dots,m} a_{k_j,t} \ . \tag{2}$$

If $m = 1$, the joined activity level series (*join*) is thus of a single channel. This join is identical to the activity level series of that channel. We use the term activity level series to identify both joins and activity level series of single channels.

Definition 1 (Snap Shot Score) For a given pair (a,s) of activity level series $a = (a_t)_{t=1,\dots,T}$ and a spike train $s = (s_t)_{t=1,\dots,T}$, the Snap Shot Score is defined as

$$SSS(a,s;\Delta t) = \frac{\sum_{t=1}^{T-\Delta t} a_t \cdot s_{t+\Delta t}}{\sum_{t=1}^{T-\Delta t} a_t} \tag{3}$$

if $\sum_{t=1}^{T-\Delta t} a_t \neq 0$, and 0 otherwise. The parameter $\Delta t \in \mathbb{N}_{>0}$ is called the *shift constant*; it defines the minimal time-lag with which causal effects are assumed to occur. Throughout this paper we consider $\Delta t = 1$, i.e. at least one time step between a cause and its effect.

The SSS quantifies the excitatory effect of an activity level series on a spike train (Fig. 2): The score value is determined by spikes occurring within the *lag-window* W defined by the inclusive boundaries *minimal response-lag* (= shift constant Δt) and *maximal response-lag* ($=\lceil d^{-1} \rceil + \Delta t - 1$). Potential information flow networks are assessed by identifying each data channel with one network node. Every node is then assigned a score value depending on the nodes linked to it. Commonly, a link's source node is called a *parent* of the destination node (*child*).[2] Using this terminology, the child-node is scored by applying the SSS to its spike train and the join of all parent-channels. A node without any parents is also assigned a score value, which requires the join $a_{(1,\dots,n)}$ of all channels. With the child's spike train s, the score of the parentless node is $SSS\left(a_{(1,\dots,n)}, s; \Delta t\right)$ if this value is non-zero, and 1 otherwise. Finally, the score of the full network is the product of all its nodes' scores.

[1] Other choices for the decay constant are possible: the smaller d is chosen, the larger the range of time-lags considered for detection of interrelations. Extreme values where $d = 0$ (activity level constantly 1 once a spike occurred on the channel) or $d \approx 0$ (activity decaying extremely slow) are unlikely to deliver sensible results. We have chosen $d = 1/3$ to keep examples (Section 3) expressive and clear. For real data, the decay constant can be derived from the anticipated maximal causal lag (in time-bins) or by using a parameter series as outlined in example 3, later.

[2] A loop-link renders a node parent and child at the same time. We will refer to such configurations as *self-exciting*.

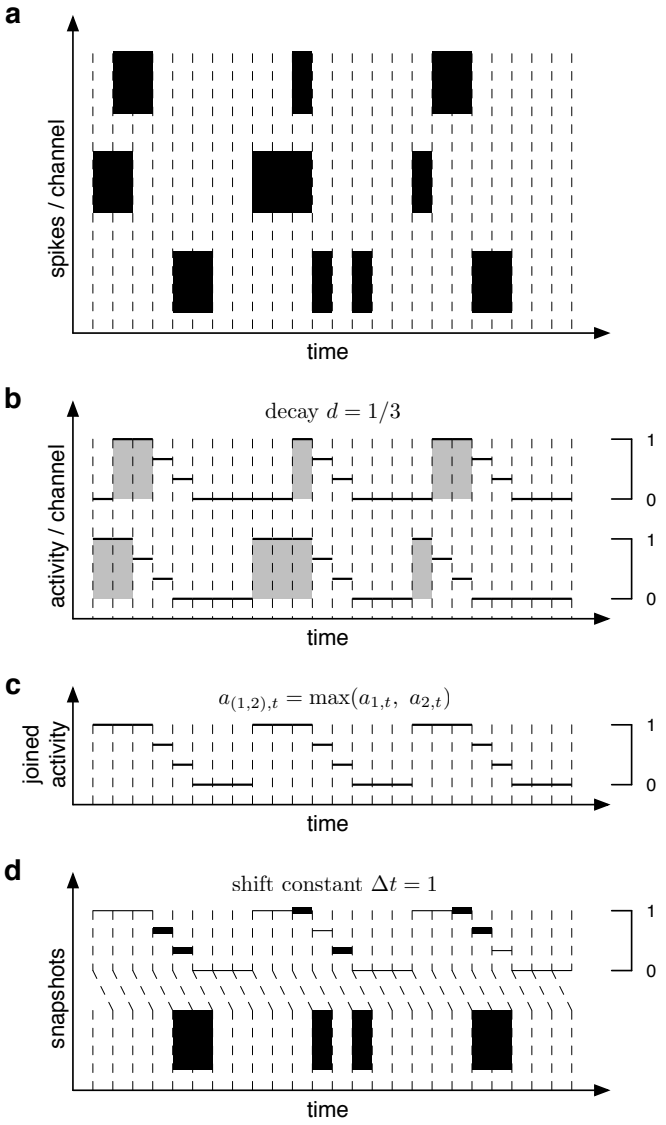


Fig. 2 Graphical interpretation of the Snap Shot Score. **a** Spike trains for three channels. The channel on the bottom is excited by the upper two channels. **b** Activity level series of upper two channels for given activity level decay (decay constant $d = 1/3$). **c** Joined activity level series (join) of upper two channels. **d** Snapshots of joined activity level series taken at spike times of bottom channel corrected by shift constant ($\Delta t = 1$). Normalising the snapshot-values $\left(\frac{2}{3}+\frac{1}{3}+1+\frac{1}{3}+1+\frac{2}{3}=4\right)$ by the accumulated joined activity $\left(1+1+1+\frac{2}{3}+\frac{1}{3}+1+1+1+\frac{2}{3}+\frac{1}{3}+1+1+1+\frac{2}{3}+\frac{1}{3}=12\right)$ yields the SSS value $\left(\frac{4}{12}=\frac{1}{3}\right)$.

The SSS can be easily interpreted by dividing both the numerator and denominator of (3) by $T-\Delta t$ to render both terms time-averages, which can be understood as probability estimates; the left hand side is then a conditional probability [22], such that (3) informally reads as:

$$P\left(\begin{array}{c}\textit{spikes will follow,}\\ \textit{given that activity is high}\end{array}\right)=\frac{P(\textit{spikes following high activity})}{P(\textit{high activity})}\ . \tag{4}$$

The model underlying the SSS assumes that the spike trains are generated by a stochastic process $\mathbf{X}_t=\left(X_t^{(1)},\ldots,X_t^{(n)}\right)$, where each neuron's future activity depends on past neural activity, i.e.

$$P\left(X_{t+1}^{(i)}\middle|\mathbf{X}_t,\mathbf{X}_{t-1},\ldots,\mathbf{X}_0\right)\ . \tag{5}$$

As discussed below, the SSS can be used to learn a network, which gives more specific information about the process $\mathbf{X}_t$: First, dependencies over time are limited by the score's lag-window W; secondly, parent-child relations in the learned network describe which subset of neurons $pa(X^{(i)})$ was found to be relevant for each neuron $X^{(i)}$, namely, its parents. Equation (5) thus simplifies to

$$P\left(X_{t+1}^{(i)}\middle|pa(X^{(i)})_{\tilde{t}\in\{(t+1)-l\ |\ l\in W\}}\right) \tag{6}$$

and describes the process $\mathbf{X}_t$ more concretely. Equation (6) does not imply one particular neuron model in order to interpret learned networks. Instead, any neuron model for which this process is a reasonable characterisation can be chosen. This could, for instance, be a leaky integrate and fire neuron, where the leakage current is chosen such that temporal summation of synaptic inputs only occurs over a time-window corresponding to the score's lag-window W. Stochastic dependencies in the resulting spiking process $\mathbf{X}_t$ would then match equation (6). Changing assumptions about the neuron model leads to a distinct understanding of the network. The large number of possible models prevents discussing all of them; we have thus chosen to present an easily interpretable template model in which the learned links between observed units stand for chains of hidden units. The aim of this model is to give an illustrative example of how to interpret recovered links; we do not suggest it to reflect physical connectivity. The model consists of the following assumptions:

- All neurons act as unreliable relay units, i.e. spikes received through synaptic transmission are forwarded to connected neurons with a certain probability.
- All postsynaptic potentials are excitatory, and synaptic transmission takes one time-bin per synapse.
- Each recovered link between observable neurons represents a connection between these neurons via a number of synapses (*connection length*). This number of synapses ranges from Δt (minimal response-lag) to $\lceil d^{-1}\rceil+\Delta t-1$ (maximal response-lag). The connections are formed by chains of hidden units, one less than synapses in the chain. For shift $\Delta t=1$ and decay constant $d=3^{-1}$ (as chosen before), there are 1 to 3 synapses between connected units, i.e. observable neurons either connect directly to each other or by up to 2 hidden units in between.

This descriptive model gives a simple interpretation for links, which can assist in explaining features of the SSS, particularly, the trade off between explanatory power and model complexity. In this model, complexity depends on both the number of links to a given node, i.e. the number of neurons upon which a neuron's firing is dependent, and the length of connecting chains of hidden nodes. Examples in section 3 will demonstrate how the SSS does not add additional parents unless the explanatory power is sufficiently increased, and

how the SSS prefers parents that fire with minimal rather than maximal response-lag, corresponding to shorter hidden chains. Thus, the SSS aims at explaining the dependence among (observed) units by linking them using few and short connections.

Learning an information flow network from data generally involves scoring many potential structures. Ideally, the highest scoring one would be found. Because of the score's decomposability, the best scoring network can be assembled from each node's best scoring parent configuration. Thus, full network scores need not to be calculated for learning, but it is sufficient to determine each node's optimal parent configuration. In order to identify these with certainty, all 2^n possible joins for each node would have to be evaluated.[3] However, for practical dimensions (like a 60 electrode array, for example), there are far too many joins for an exhaustive evaluation. To circumvent this problem, the set of information flow networks to score can be limited to ones with sparse connectivity, or limits can be placed on the number of parents per node. The number of potential child-parent relations might also be reduced by excluding connections ruled out by factual knowledge (like large physical distance between electrodes, for example). Additionally or alternatively, as mentioned previously, search heuristics and Monte Carlo methods can be used to select promising networks to assess [16,7,4,78,40,19,51,32,43,13,12,66]. Once dependencies have been revealed, they can be visualised as a network (graphical model [3,59,45]), which connects nodes associated with certain spike train channels to indicate excitatory influence between the observed units.

Network inference can be assisted by prior knowledge about the studied system, which can be used to derive a *link-acceptance-threshold* (LAT) for each network node. Any parent configuration with a score value lower than the child node's LAT will be rejected. This selection removes irrelevant links and can lead to sparser, more relevant networks. The LAT reflects the best explanation for the data at a particular level of complexity. The actual level of complexity is determined by the prior information at hand: Knowledge about the studied system constrains the space of potential parent configurations for each node. For example, self-excitation might be excluded, or observed units are known to only have few interaction partners. The space of potential configurations can thus be restricted to a particular level of complexity, i.e. number of parents. Configurations at the highest permitted level of complexity determine the LAT, which is the highest score value of these configurations. This highest scoring configuration (*LAT-configuration*) reflects the best explanation for the data at a level of complexity, which could not be limited further by using prior knowledge (represented by the LAT-configuration). Better explanations than the LAT-configuration might exist: these are simpler configurations with scores equal or above the LAT. Calculating the SSS for several parent configurations can reveal such superior explanations. Ultimately, we seek to find the simplest among the best scoring configurations consistent with prior knowledge. Thus, any configuration with a score value below LAT should be omitted from result lists, as it gives a worse explanation for the data than prior knowledge (LAT-configuration).

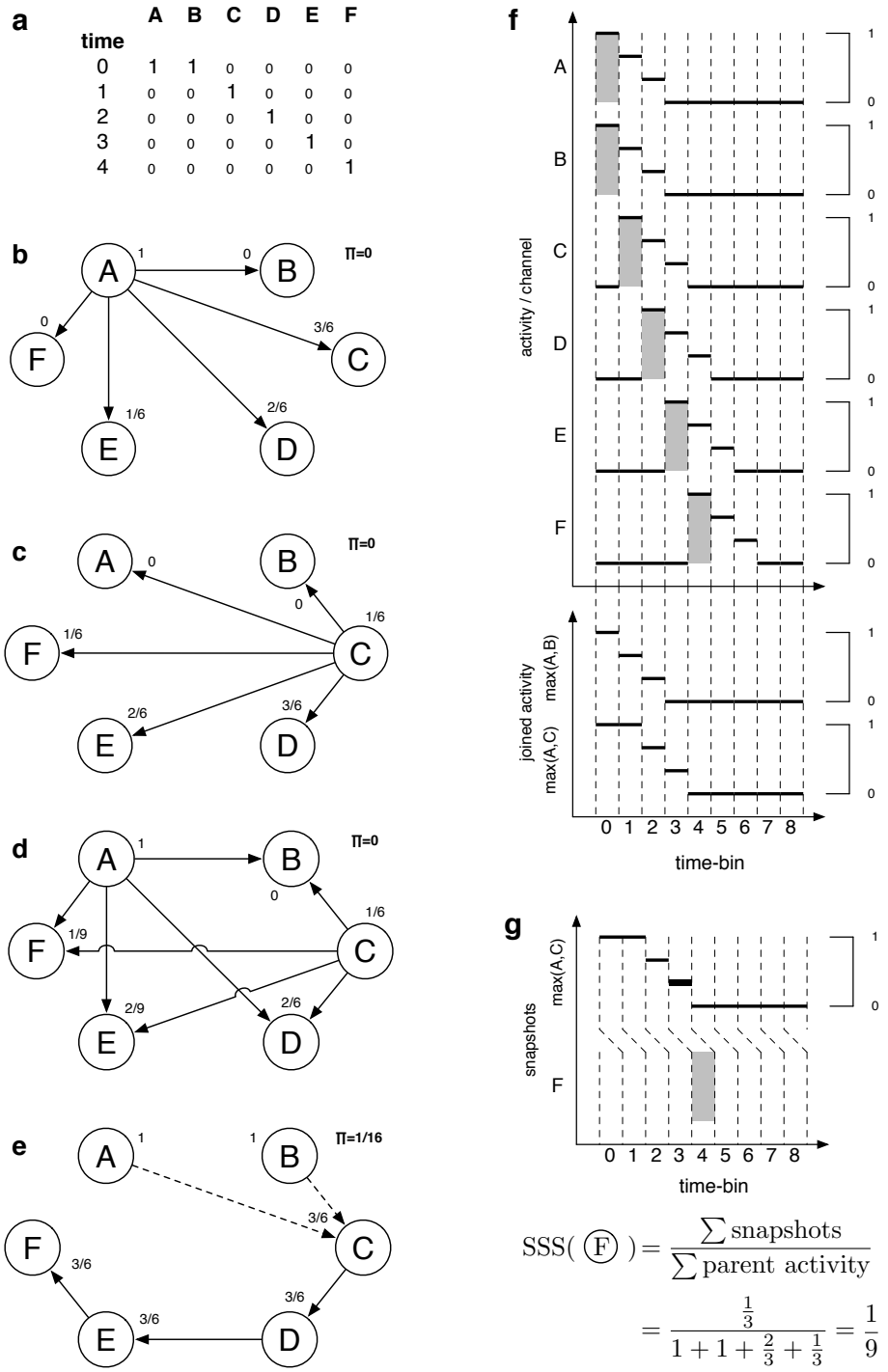
[3] Implementation note: Profiling implementations (in C [41] and Python [77]) of our method revealed that calculating joins is computationally much more expensive than calculating the SSS value. Instead of recalculating a join for different nodes, we suggest to perform scoring join-wise, i.e. scoring all nodes with a join once it has been computed.

3 Illustrative Examples

To illustrate the functionality of our method, we discuss a series of three examples: First, we clarify the nature of the SSS by working out basic features of the score. We then evaluate all possible parent configurations of a node for different sample data sets and discuss the score's quantification of these. Finally, we apply the method to data from neural network simulations, to demonstrate its performance in complex scenarios. This last example will also address how good parameter setting of the SSS can be found in practice.

Example 1 (Snap Shot Score features) In order to illustrate first basic features of the score, we consider an unnatural data-set with 6 channels (Fig. 3a). We calculated the activity level series (Fig. 3f) and scored selected information flow networks (Fig. 3b-e); the links of these networks were chosen to illustrate central features of the SSS:

- The score value is zero if putative cause and effect do not appear within the lag-window of minimal and maximal response-lag: $\left[\Delta t,\ \lceil d^{-1}\rceil + \Delta t - 1\right] = [1,3]$ (Fig. 3b). This is especially true when effects precede supposed causes (Fig. 3c). Note that in figure 3bcd, node B's contradictory parent configurations render the full networks inconsistent, which is reflected by their zero score values.
- The score value is maximal if putative cause and effect occur exactly at the minimal time-lag, and it is lower if effects occur later (Fig. 3b). The SSS thus favours units as causal ones to which the response-lag is minimal (Nodes D, E, F in Fig. 3b vs. 3c).
- Addition of more parents may get penalised by the SSS: If higher complexity is not balanced out by a significant explanation benefit, the score value decreases (Nodes D, E, F in Fig. 3c vs. 3d).
- Different parent configurations can have the same score value if spike trains are identical for different units (Fig. 3e). It is unsurprising that the score cannot distinguish between alternative explanations where different, but identical in spike train, units are parents: any unbiased method would be expected to present such equivalent alternatives. It is not optimal that the score cannot distinguish between identical single units and their join: one would rather penalise the multi-parent structure. But, as we will discuss in the following example in more detail, for realistic data, it is extremely unlikely that two units have precisely identical spike trains; it is thus extremely unlikely that several networks are assigned the same high score value.



$$\mathrm{SSS}(\,\textcircled{\mathrm{F}}\,) = \frac{\sum \text{snapshots}}{\sum \text{parent activity}} = \frac{\frac{1}{3}}{1+1+\frac{2}{3}+\frac{1}{3}} = \frac{1}{9}$$

Fig. 3 Legend separate in box on page 11.

Fig. 3 legend: Simplistic spike trains and SSS values of selected networks (left), activity level series used for scoring and detailed calculation example (right). **a** Spike trains of 6 units (*A*-*F*). **b-e** Snap Shot Score values shown for selected parent-child configurations (near child nodes) and for full networks ($\prod = \ldots$ next to network). **b** Unit *A* is single parent of all other nodes. Scores of nodes *B* and *F* are zero, as these units do not *respond* within the defined time-lag-window: Unit *B* undershooting minimal response-lag ($\Delta t = 1$); unit *F* overshooting maximal response-lag ($\lceil d^{-1} \rceil + \Delta t - 1 = 3$ time-steps). Decreasing non-zero scores of nodes *C* through *E* reflect the Snap Shot Score's preference of short time-lags. **c** Unit *C* is single parent of all other nodes. Scores of nodes *A* and *B* are zero, as their response undershoots the minimal response-lag: Links $C \rightarrow A$ and $C \rightarrow B$ are directing *backwards in time*. Scores of nodes *D*, *E*, and *F* are larger for parent node *C* than for parent node *A* (b), as their response-lag to unit *C* is smaller than to *A*. **d** Units *A* and *C* are joined parents of nodes *B*, *D*, *E*, and *F*. All non-zero scores are smaller than those where unit *C* is the only parent (c); the explanatory benefit of two parents (*A* and *C*) does not balance out raised complexity of the network. **e** Nodes *A* or *B* are parents of node *C* either exclusively or jointly; a chain is formed of *C* to *D* to *E* to *F*. For the data in (a), these three resulting networks are the best scoring ones. Their structure differs with respect to the parent configuration of node *C*, but links between nodes *C* to *F* are unambiguous. **f** Activity level series of all units (top) and joins $a_{(A,B)}$ of *A* and *B*, and $a_{(A,C)}$ of *A* and *C*, respectively (bottom). Join $a_{(A,B)}$ is used in (e) and equals individual activity level series of *A* and *B*. Join $a_{(A,C)}$ was used to calculate scores in (d). **g** Detailed score calculation for node *F* with parents *A* and *C* as shown in (d).

While the preceding example showed how the SSS works for selected child-parent configurations, the following one illustrates how it operates on the full space of parent configurations of one particular node. The systems discussed are of low dimensionality in order to be able to display results appropriately; systems of higher dimensions will be discussed in example 3, later.

Example 2 (Exhaustive network space evaluation) To illustrate the Snap Shot Score's selectivity, we consider different activity patterns (4 channels) and evaluate the score of node 1 for all its possible parent configurations (Fig. 4). We compare scores of individual configurations against the mean score of all configurations for illustrative purposes, only.[4] Despite the low dimensionality of the depicted situations, several characteristics of the SSS can be seen. For example, joining silent channels has no effect on the score value: configurations in Fig. 4a have the same score value whether they contain the silent channel 4 as a parent or not. Also, joining completely identical channels is effectless compared to using only one of them (Fig. 4b, channels 2 and 3). Both of these effects are due to the max-operation in (2). More precisely, the score stays unaltered if the activity level series of the channel to join does not raise the activity level any further (Fig. 4c, join of channels 1 and 2 equals channel 1). This occurs in only three special cases: (1) joins of identical spike trains, (2) joins where all spike trains are a precise subset of one of the spike trains, and (3) joins including silent channels. This is not optimal; one would rather penalise the effectless complication of a structure. However, this behaviour is a consequence of the simplicity of the score. For real data, it seems unlikely that identical spike trains, or spike trains that echo precisely a subset of another, are observed on different channels; thus, the first two special cases are expected to have little effect in practical application. The third special case can be handled by

[4] Generally the mean score is unknown, because an exhaustive evaluation is computationally impossible in practical dimensions.

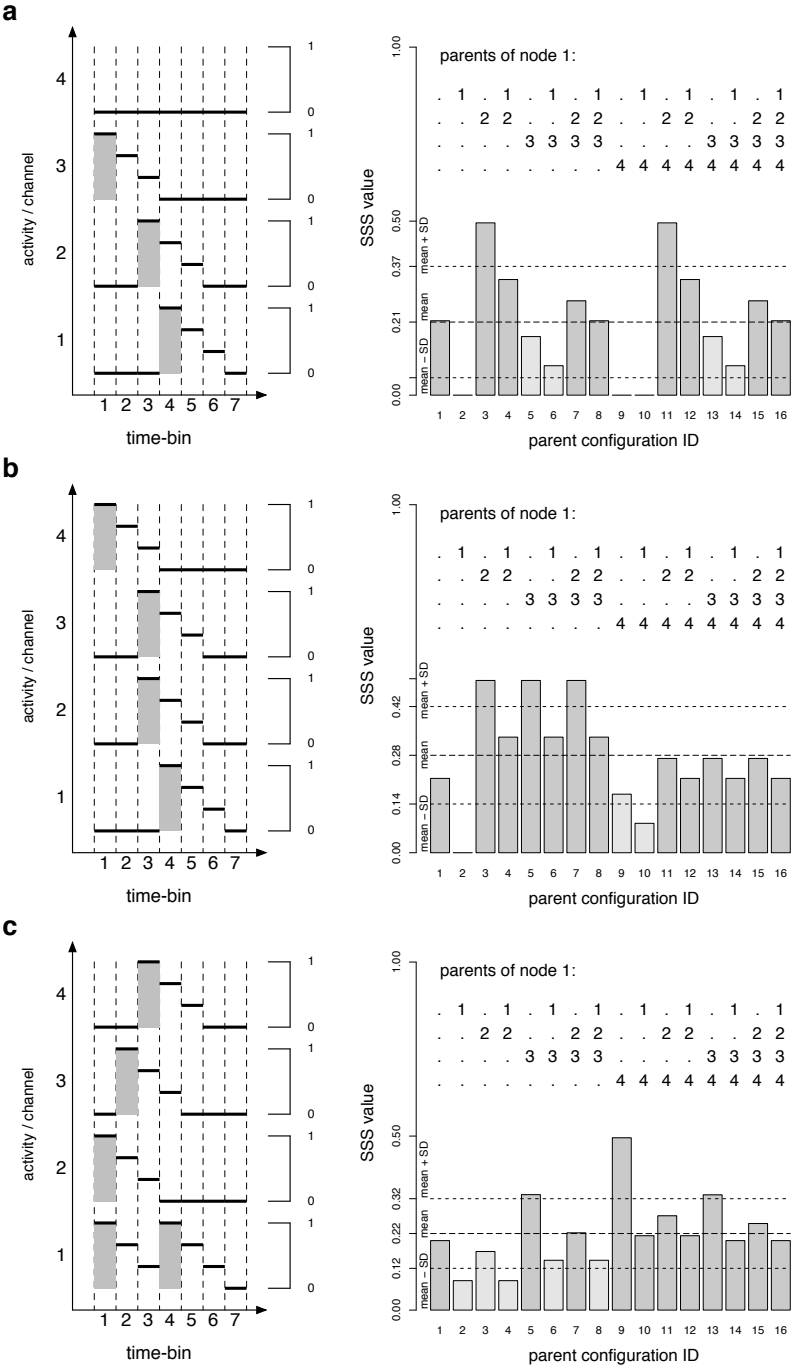


Fig. 4 Spike train patterns and activity level series (left) and corresponding Snap Shot Score values of channel 1 for all its possible parent configurations (right). Spikes are indicated by grey bars with superimposed lines indicating the resulting activity levels. The adjacent bar plot shows the score values of all possible parent configurations for node 1 (parent nodes given above each bar). Mean score value and mean plus/minus one standard deviation (root mean square) shown by dashed lines for illustrative purposes. Bars indicating the score values are coloured light grey if below the link-acceptance threshold (score of most complex configuration 16) and dark grey otherwise. **a** Channel 4 silent; joining it has no effect on SSS value. **b** Channels 2 and 3 identical; joining more than one of them has no effect on SSS values. **c** Close spikes on channel 1; self-exciting configuration (number 2) with positive SSS value.

preprocessing to remove inactive channels (as would be likely practice in any case). Thus, in practice the SSS values are likely to be different for every parent configuration (Fig. 4a, configurations 1-8). The different score values can be used to order parent configurations hierarchically for subsequent inspection and result selection.

In Fig. 4a-c, the SSS assigns distinct top-scores to its most favoured configurations. To grasp the score's characteristics for more variable data, the spike patterns in Fig. 4 were concatenated in three different ways to yield the spike trains shown in Fig. 5. Interpreting the resulting spike trains with respect to the question *"Which channels are exciting channel 1?"* can be harder or easier than before; the SSS values reflect this: In Fig. 5a, channel 1 seems to be clearly excited by channel 2, as all parent configurations but one of those including this channel have score values above the mean. However, except from the clear peak for configuration number 3, the score is high and *undecided* about some others, especially configurations 7 and 11, and configurations 4 and 5. The spike train does not contain enough information to clearly prefer one of these parent configurations over the other; we find high scores approaching one standard deviation (SD) above the mean for all four configurations. A similar situation occurs in Fig. 5b: One configuration is clearly favoured, too, but high scores (about one SD above the mean) are reached by only two other configurations. The spike train in Fig. 5b is thus more expressive to the SSS than the preceding one (Fig. 5a).

Score value peaks become flat when the data contains unclear information: In Fig. 5c spikes on channels $2,3$ and 4 all occur *as favoured* (with time-lag 1) once, but also with larger lags another time. We find the prominence of the top configuration number 5 less distinct than in other cases and the overall score distribution close to the mean score. More informative data would be needed for the formation of a distinct peak; indeed, repeating one of the two spike patterns (Fig. 4bc) in the concatenation twice already leads to fewer and more distinctly favoured configurations (not shown).[5] In both Figures 4 and 5, even numbered configurations include node 1 as its own parent; for configuration number 2, node 1 is its only parent. This exclusive self-exciting configuration can have non-zero score values when two spikes on channel 1 occur close enough to each other, i.e. within the lag-window (Figs. 4c, 5c); any two spikes that are too close or too far apart do not contribute to the score value (Fig. 4ab). If (exclusive) self-exciting configurations seem implausible for the single-unit spike train data, they should not be considered.

In the final example, we assess the performance of our method by using simulated data to demonstrate its potential and value for learning effective connectivity networks in practice. We begin the example by introducing a plausibility concept to assess any network inference technique under partial observability conditions. We apply this concept to the SSS to show its performance in a series of different situations.

Example 3 (Neural network simulations) To indicate how the SSS performs in more realistic situations, we simulate a feed-forward network (Fig. 6a) with an integrate and fire model (see [18, pp.162] or [29, chapter 4.1] for example) and learn information flow networks from the simulated data. The neural simulation was set-up as follows: Neural baseline activity is given by uncorrelated homogeneous Poisson processes [22, pp.446]. For different simulations, the rate parameter was varied ($\lambda = 10^{-1}, 15^{-1}, 25^{-1}, 30^{-1}, 40^{-1}, 50^{-1}$), corresponding to level of *spontaneous spiking activity*. Spikes propagate according to the network connectivity (Fig. 6a) with a delay of one time-bin (1 msec) per link. Each neuron integrates spikes received from its parents over time; the *synaptic efficiency* was varied to allow $2,3,4,$

[5] For independent random spike trains, SSS values of all parent configurations lie within mean $\pm$ SD and approach mean score ($SD \searrow 0$) for increasing length of spike trains (not shown).

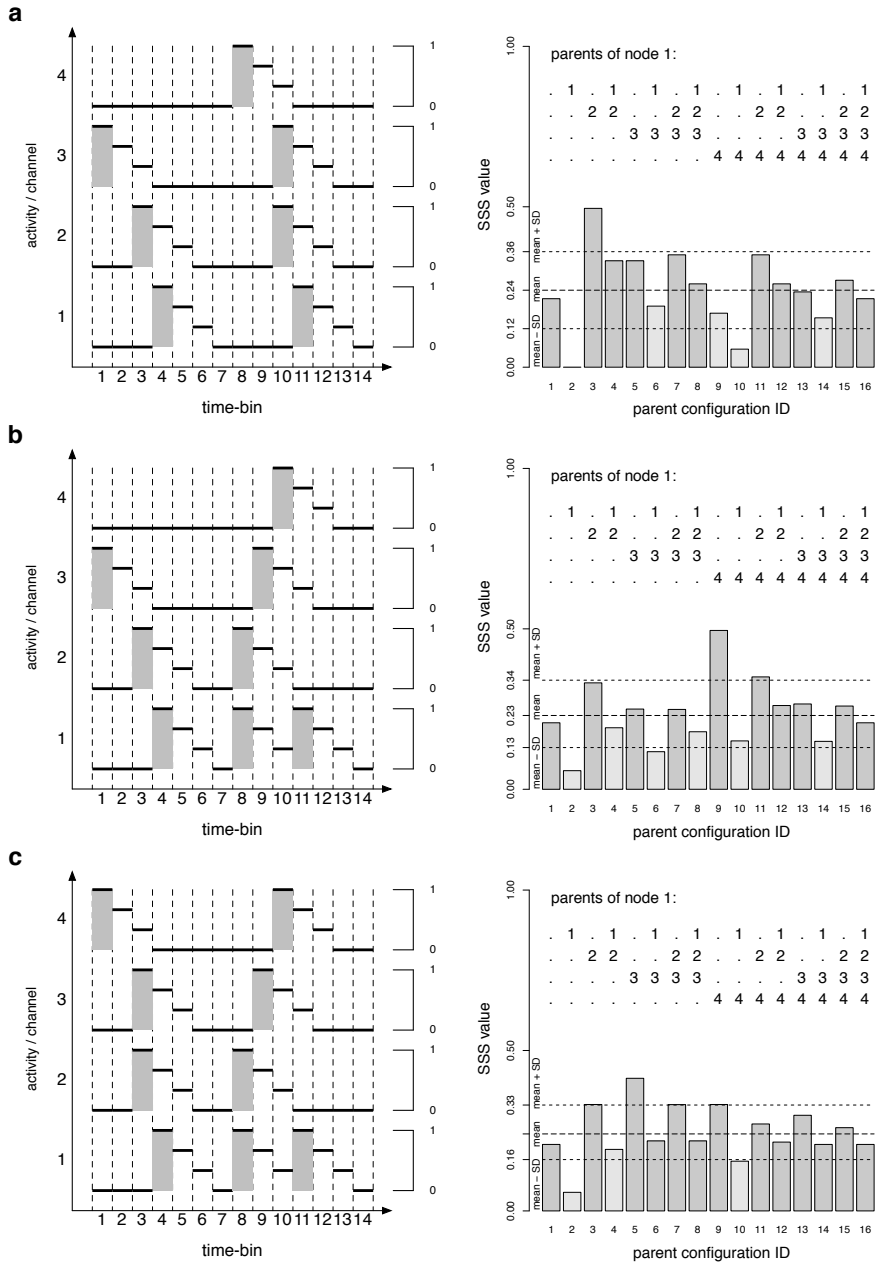


Fig. 5 Exhaustive score evaluations of all parent configurations for channel 1 with same semantics as in Fig. 4. Spike train patterns are concatenations of spike patterns in Fig. 4 (**a** Fig. 4ab, **b** Fig. 4ac, **c** Fig. 4bc). **a** Distinct peak and four parent configurations with score value close to mean plus one SD. **b** Distinct peak and two parent configurations with score value close to mean plus one SD. **c** Score value of most configurations close to mean score and vague peak only.

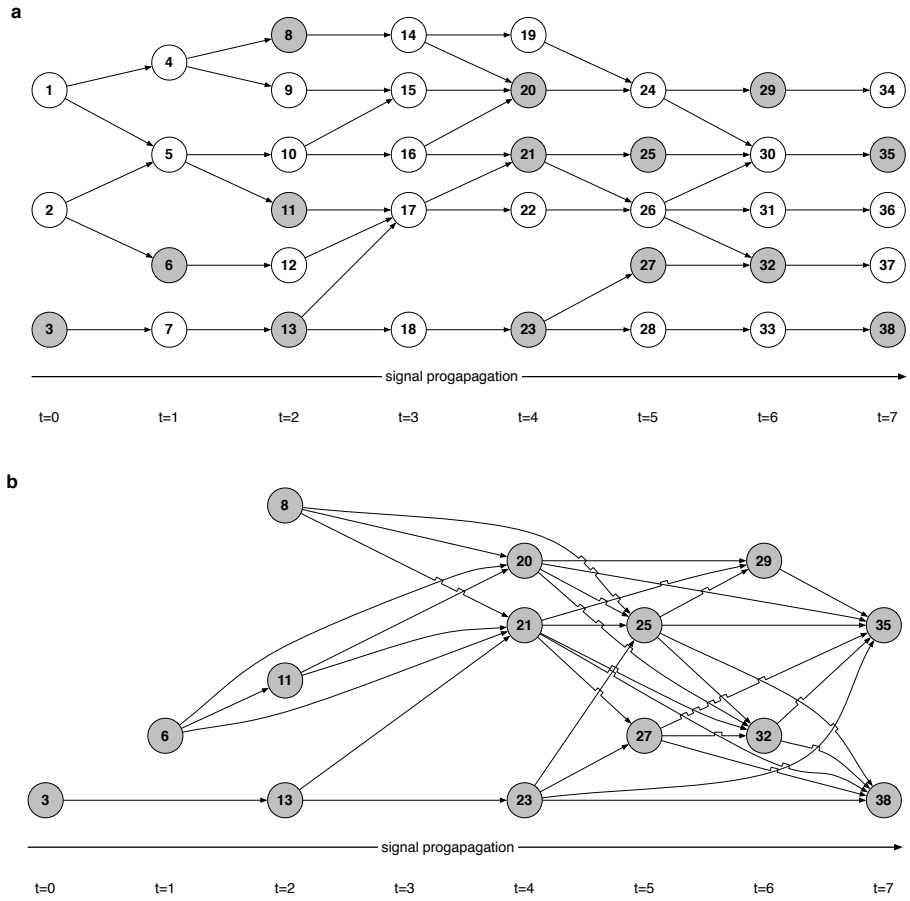


Fig. 6 **a** Simulated feed-forward network (38 nodes, 47 links) with observable units (14 nodes $\approx 36.8\%$) shaded in grey and hidden units unshaded. **b** Observable nodes from network shown in (a) with all 34 plausible links ($\approx 18.7\%$ out of 182 possible links) for plausible lags $l_{min} = 1$ and $l_{max} = 3$.

or 5 received spikes to evoke a spike in the receiving neuron. In each simulation, all neurons are stimulated by uncorrelated activity of equal rate; synaptic efficiency is equal for all neurons. Every such parameter combination was simulated 10 times for different *data lengths* (5, 10, and 30 seconds, 1, 5, and 10 minutes). In accordance with practical situations, we assume that the simulated system is not completely observable, and only spike trains from a subset of units (*observable nodes*) can be collected and used for learning. Learned networks only contain links between these observable units, such that their quality cannot be determined by a simple link-by-link comparison with the full network, in which many observable units are connected via hidden ones. We therefore analyse the full network in order to classify each possible link between observable nodes as plausible or implausible. For this classification, we introduce a concept of (link) plausibility, which is not restricted to use with the SSS, but can be used for the assessment of any network learning technique using neural network simulations. The general idea of the concept is to determine plausibility of

any link between observable units from the known simulated network. For example, plausible links can be ones which exist in the full network (e.g. $\underline{21} \rightarrow \underline{25}$, $\underline{23} \rightarrow \underline{27}$, and $\underline{27} \rightarrow \underline{32}$) or certain links for which a directed path from the link's starting node to its end node exists (e.g. $\underline{3} \rightarrow$ *(7)* $\rightarrow \underline{13}$, $\underline{13} \rightarrow$ *(17)* $\rightarrow \underline{21}$, and $\underline{23} \rightarrow$ *(28)* $\rightarrow$ *(33)* $\rightarrow \underline{38}$).[6] Links that contradict the full network by connecting nodes in the opposite direction of information flow in the feed-forward network (e.g. $\underline{13} \rightarrow \underline{3}$, $\underline{21} \rightarrow \underline{3}$, and $\underline{35} \rightarrow \underline{3}$) are implausible, for example. The system's partial observability can lead to plausible links for which no directed path between units exists: A common trigger can cause co-ordinated firing between nodes, which are connected via the triggering node only (e.g. $\underline{6} \leftarrow$ *(2)* $\rightarrow$ *(5)* $\rightarrow \underline{11}$). A link connecting the two co-ordinated nodes is thus plausible if one node could fire within the *plausible lag-window* $[l_{min}, l_{max}]$ of the other node (in the right order). Plausibility of a link thus depends on the full network, which nodes are observable, and parameters specifying the minimal and maximal plausible lag (l_{min}, l_{max}). This is formalised in the following

Definition 2 (plausibility) Let a and b denote two observable nodes. Node a is called a plausible parent of b if there exists a node s (in the full network) for which directed paths to both a and b exist such that:

1. their lengths[7] $l(s \rightarrow a)$ and $l(s \rightarrow b)$ fulfil $l_{min} \leq l(s \rightarrow b) - l(s \rightarrow a) \leq l_{max}$, and
2. a path $a \rightarrow b$ exists, which does not include another plausible parent of b.

Node a is called an implausible parent of b otherwise. The link $a \rightarrow b$ is called (im)plausible whenever a is a(n im)plausible parent of b.

The first condition in the definition assures that dependence between nodes a and b can arise within the plausible lag-window. The second condition refines the set of plausible parents by rejecting those for which the relationship can only propagate through another plausible parent of b. (For example, for $(l_{min}, l_{max}) = (1,3)$, the link $\underline{23} \rightarrow$ *27* $\rightarrow \underline{32}$ fulfils the first but not the second condition. Comments on the generic concept of plausibility can be found in the last paragraph of this example.) We used an algorithmic implementation (Algorithm 1) of the definition in order to analyse the full network (Fig. 6a). Throughout this example, the plausible lags are chosen $(l_{min}, l_{max}) = (1,3)$ to determine all plausible links (Fig. 6b). We have chosen l_{min} according to the minimal lag between observable nodes in the full network and l_{max} such that a modest percentage of links will be plausible.[8] In this setting, the plausible lag-window and the lag-window of the SSS are both $[1,3]$. This matching is expected to yield the best results because links in learned networks only connect nodes with appropriate time-lags: ones within the SSS lag-window. As the lag-windows are identical, all links that are plausible could be learned, but not those spanning over time-lags outside the plausible lag-window. To show that the performance of the SSS does not depend on this matching, we will vary the decay constant d (and thus the lag-window) later. We will also account for the fact that the probability of finding plausible links by chance depends on their percentage. (See P-value calculation below.)

We ran several series in which both the parameters of the neural simulation and the score were altered systematically in order to assess their influence on the quality of the

[6] Start- and end-nodes underlined, nodes on path in full network (given for illustration) italic, hidden nodes in brackets.

[7] A series of directed links is called a *path*, if the origin of all links equals their predecessors' destination. Length $l(a \rightarrow b)$ of directed path from node a to b is defined as number of links on path. The length of a path $a \rightarrow b$ directly corresponds to the time-lag a signal needs to propagate from a to b. In our neural simulation, the time-lag in time-bins (1 msec) is equal to the length of a path.

[8] For $l_{min} = 1$ there exist 12 ($\approx 6.6\%$ out of 182 possible links, $l_{max} = 1$), 27 ($\approx 14.8\%$, $l_{max} = 2$), 34 ($\approx 18.7\%$, $l_{max} = 3$), 37 ($\approx 20.0\%$, $l_{max} = 4$) plausible links.

Algorithm 1 Determination of all plausible links (Definition 2) for a partially observable network. The algorithm begins with preliminary determination of reachability sets for each node. These consist of all nodes to which a directed path exist. The main loop then implements Definition 2 in two steps: First, potential plausible parents are determined; these are observable nodes, which fulfil condition 1 in the definition. In the second step, condition 2 of the definition is checked, and nodes violating the condition are removed as potential plausible parents. The remaining nodes fulfil both conditions of plausibility and are returned in the end.

Input: Full network, observable nodes
Parameters: minimal and maximal plausible lag (l_{min}, l_{max})

for all nodes s_i **do**
  determine the reachability set R_i of all nodes that can be reached from s_i via directed paths; $R_i = \{s_i\}$ if s_i isolated node without any links
end for
for all observable nodes o **do**
  (First determine all nodes ppa_o that fulfil condition 1 of definition 2)
  initialise its plausible parent set $ppa_o := \emptyset$
  for all nodes s_i where $o \in R_i$ **do**
    for all nodes $pa \in R_i$ **do**
      if paths $s_i \to o, s_i \to pa$ with $l_{min} \leq l(s_i \to o) - l(s_i \to pa) \leq l_{max}$ exist **then**
        $ppa_o := ppa_o \cup \{pa\}$
      end if
    end for
  end for
  (Remove all nodes from ppa_o that do not fulfil condition 2 of definition 2)
  for all parents $pa \in ppa_o$ of node o **do**
    if $\forall$ paths $pa \to o = (pa, x_1, \ldots, x_k, o) : ppa_o \cap \{x_1, \ldots, x_k\} \neq \emptyset$ **then**
      $ppa_o := ppa_o - \{pa\}$
    end if
  end for
end for
return all plausible parent sets ppa_o

recovered networks. (The full network, observable nodes, and plausible links were left unaltered.) For each parameter combination, the full network is simulated with an integrate and fire model to yield spike train data. Data channels corresponding to observable nodes are then used to determine each node's best scoring parent configuration with no more than 3 parents. Composing these parent configurations yields the recovered network, which is analysed with respect to its plausibility: Links of learned networks are classified as either *hit* or *miss*, depending on whether they are plausible or not (respectively). We then determined the *recovery rate* (the percent the hits represent of all possible plausible links [Fig. 6b]),[9] *precision* (the percent of all the recovered links that are hits), and the corresponding P-value using the hypergeometric distribution [22, pp.43].[10]

The performance of the SSS is compared to that of cross-correlation [60], which results in comparable computational costs. In order to account for different time-lags in the cross-correlation analysis, the correlation between any two channels A and B has been evaluated

[9] Note that the definition of the recovery rate corresponds to sensitivity. We have called it differently because we do not expect the recovery rate to reach 100% (as is explained in the text later), which the reader might assume if confronted with the familiar but misleading term sensitivity.

[10] The chance level for at least h hits out of p plausible links out of N total links with k links learned is $\sum_{i=h}^{\min\{p,k\}} q_i$, where $q_i = \frac{\binom{p}{i}\binom{N-p}{k-i}}{\binom{N}{k}}$.

from time-shifted data: shifting B's data forward (relative to A) by $1, 2$, or 3 time-bins. The maximal correlation between channels was then assigned to the corresponding link $A \to B$. Links with maximum correlation equal or above a threshold α are *learned*; all remaining ones are not. In practice, the threshold α would have to be chosen by the user, which is not possible for the large number of simulations performed here (1,440 simulations). Instead, for each analysed data-set, the threshold has been chosen to yield optimal performance: According to the Neyman-Pearson Lemma ([54] or [18, pp.119]), no better choice for α exists than the one which yields the highest likelihood-ratio (i.e. recovery rate / [100 - precision]). The threshold α is chosen according to this optimal trade-off and thus yields the best performance that can be reached with this technique.

Parameters of the neural simulation (spontaneous activity level, synaptic efficiency) do not have any physical correspondence in this sparsely connected network; we refrain from showing their individual influence on network inference, as their combined effect is fully reflected in the simulated spike train. Instead, the simulation output, the spike train, is characterised by the amount of information it contains about the network: Uncorrelated Poisson spike trains that were used to stimulate network activity do not convey any information about the network itself; only spikes that were induced by post-synaptic potentials provide information about network connectivity. Therefore, each spike train is characterised by quantifying the proportion of externally induced spikes and evoked ones. Evoked spikes increase the total number of spikes (i.e. stimulating and evoked ones), and we call the percental increase the simulated system's *impetus*.[11] When the impetus is low, the data is similar to the uncorrelated stimulation spike trains (representing inherent spontaneous activity of model neurons or excitation originating from un-modelled units). A higher impetus indicates a more autonomous system with higher self-dynamics. In such systems, the spike trains are more informative about network connectivity, which is expected to improve network recovery. The impetus was determined for every data set used for network learning in order to show its effect on the performance of our method. We note that the impetus cannot be calculated for real data; its purpose is purely to provide a scale for the informative value of spike trains in the simulation. Knowing the impetus for real data would be useful to relate the results of our simulations to the quality of networks learned from these data. However, experimenters can only estimate the impetus for their data roughly. For example, recordings within a feed-forward type structure are expected to exhibit a higher impetus than from an area with many converging external inputs.

In simulation series, we observe the expected relation between quality (impetus) and amount of data (length of spike train) and the grade (recovery rate, precision) of networks learned from the data (Fig. 7): In order to learn networks of a certain grade, a lower impetus can be compensated by longer recordings; longer recordings at a fixed impetus generally improve the quality of the learned networks; and, for a particular data length, recovered networks are generally better for higher impetuses, i.e. more informative data.

To understand why the recovery rate does not exceed 35% for any learned network (Fig. 7a), we note that not all of the links that are classified as plausible are expected to be recovered. This is because, generally, some redundancy among them exists; for example, in figure 6b links $20 \to 29$, $21 \to 29$ are both plausible, but given the joint excitation of nodes 20 and 21 by unit 16, they might be similar enough such that one of them is sufficient to explain spiking of unit 29. The set of plausible links is therefore a superset of those that are expected

[11] The $\mathit{impetus} = 100 \cdot \#\mathit{evoked\ spikes}/\#\mathit{stimulating\ spikes}$ where number of $\mathit{evoked\ spikes} = \#\mathit{spikes\ in\ simulation\ output} - \#\mathit{stimulating\ spikes}$. If, for example, *impetus*=0%, then no spikes were evoked by the stimulation and the simulation output is equal to uncorrelated random spike trains. For *impetus*=100%, the simulation output is a mixture of two halves: stimulation spikes and evoked spikes.

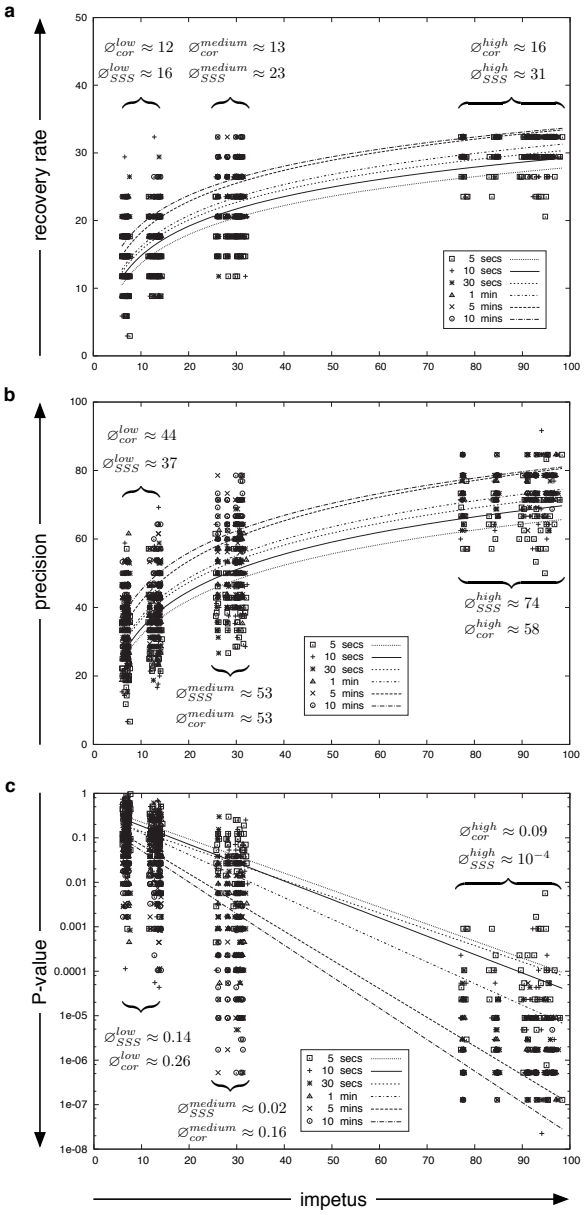


Fig. 7 Quality of recovered networks depending on impetus and recording length ($d = 3^{-1}$, $\Delta t = 1$ fixed). Data points represent one learned network each; trend lines fitted to points of each data-length. Average values of the SSS, $\varnothing_{SSS}$, of three impetus-regions *low*, *medium*, and *high* (indicated by curly brackets) given above/below data points. For the correlation approach, no data points are shown, but these are summarised in the corresponding average values $\varnothing_{cor}$. **a** Recovery rate of all plausible links for different impetuses and different lengths of data. Higher impetuses and longer data sets result in better recovery rates. Longer data sets can compensate for lower impetuses. **b** Precision of recovered links for different impetuses and different lengths of data. High impetuses and longer recordings improve precision. Longer data sets can compensate for lower impetuses. **c** P-values of precision shown in (b) on logarithmic scale. Networks with a higher precision are less likely to be revealed by chance. Lower P-values thus indicate better performance.

to be revealed by network inference. Out of this superset, the SSS predominantly recovers links that connect nodes and their closest plausible parent, but not more distant ones. (This can be explicitly seen in Fig. 9, which will be discussed later.) Additional plausible links that do not sufficiently increase the explanatory quality of the network are omitted due to the SSS's preference for simpler networks.

In contrast, the precision would optimally reach 100%, such that all recovered links are indeed plausible. Fig. 7b shows the relationship between impetus and precision, which is high when the impetus is high. Reaching such precision by chance is extremely unlikely (Fig. 7c); however, at lower impetuses, the percentage of learned links that are implausible increases and causes precision to drop down, as is to be expected. For the lowest range of impetuses [0-20], cross-correlation (on average) yields better precision than the SSS. However, for higher impetuses as well as for P-values and recovery rates, the SSS shows equal or better performance than cross-correlation (for which the threshold α has been chosen to yield optimal results).

We used the same framework to investigate the robustness of learning performance for a fixed data length with respect to different choices for the decay constant. The corresponding results are depicted in Fig. 8. Excluding the extreme case $d = 1$, we find all settings performing similarly well over a range of impetuses. Setting $d = 1$ results in a significantly worse performance, as the resulting lag-window $[1,1]$ causes the SSS to find relationships of lag 1 only. Links with larger time-lags (e.g. $\underline{3} \rightarrow$ *(7)* $\rightarrow \underline{13}$, $\underline{13} \rightarrow$ *(18)* $\rightarrow \underline{32}$, and $\underline{23} \rightarrow$ *(28)* $\rightarrow$ *(33)* $\rightarrow \underline{38}$) can thus only be found through spurious firing within one time-bin. This limitation accounts for the comparatively low performance of this setting.

Realistic situations may allow for only a few recordings of the studied system under similar conditions. We mimic this situation by simulating 10 data sets of 30 seconds in length, each using the same simulation parameters, such that all spike trains exhibit about the same 30% impetus.[12] For each of the 10 data sets, five networks were learned, one for each decay constant $d \in \{5^{-1}, 4^{-1}, 3^{-1}, 2^{-1}, 1\}$, $\Delta t = 1$. For each decay constant, the networks from the 10 data sets were averaged. We find the earlier observations (Fig. 8) confirmed: On average, networks learned with decay constant $d = 1$ contain fewer plausible links than those learned with smaller decay constants (Fig. 9a vs. 9b-e). Independent of the decay constant, implausible links show lower percentages of recovery than plausible ones (Fig. 9a-e). Furthermore, most plausible links are recovered consistently, while implausible ones vary for different decay constants (Fig. 9b-e). In order to account for links learned from spurious relationships, we averaged over all learned networks ($d \in \{5^{-1}, 4^{-1}, 3^{-1}, 2^{-1}, 1\}$); this reduces the frequency of implausible links significantly, while preserving plausible ones with high percentages (Fig. 9f).

We refrained from running series to increase the shift constant Δt, as the structure of the simulated network (Fig. 6a) and its implementation as a dynamical system directly suggest $\Delta t = 1$ to be the optimal choice. For real data, the shift constant is best chosen based on the investigator's expectation about the minimal response-lag over which relationships may occur; for example, electrode placement (close to each other / in different brain regions) during data recording may suggest smaller or larger time-lags. Expressing the expected time-lag as number of corresponding time-bins in the spike train yields the optimal shift constant.

In this simulation example, we find the SSS performing well for a large range of impetuses irrespective of (non-extreme) choices for the decay constant. This parameter insen-

[12] The data sets sum up to a total recording time of 5 minutes. The impetus was found between 29.7% and 30.0%. For an impetus of 30%, the simulation output consists of 3 evoked spikes per 10 uncorrelated stimulation spikes (on average).

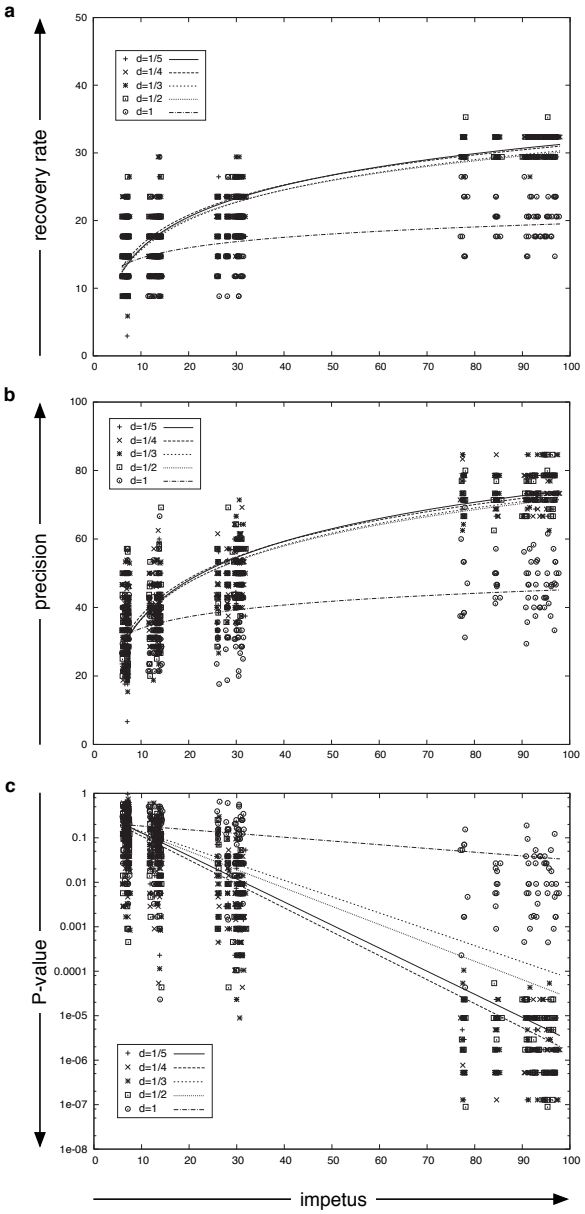


Fig. 8 Quality of recovered networks depending on impetus and decay constant (recording length 30 seconds, $\Delta t = 1$ fixed). Data points represent one learned network each; trend lines fitted to points of each choice of decay constant. **a** Recovery rate of plausible links for different impetuses and choices of decay constant d. Decay constants $d \in \{5^{-1}, 4^{-1}, 3^{-1}, 2^{-1}\}$ result in similar good recovery rate. Significantly worse recovery rate for decay constant $d = 1$ due to inability to consider dependence over multiple time-lags. **b** Corresponding precision for impetuses and decay constants shown in (a). As in (a), performance is similar for decay constant $d \in \{5^{-1}, 4^{-1}, 3^{-1}, 2^{-1}\}$, but worse for $d = 1$ (for the same reason). **c** P-values of precision shown in (b) on logarithmic scale.

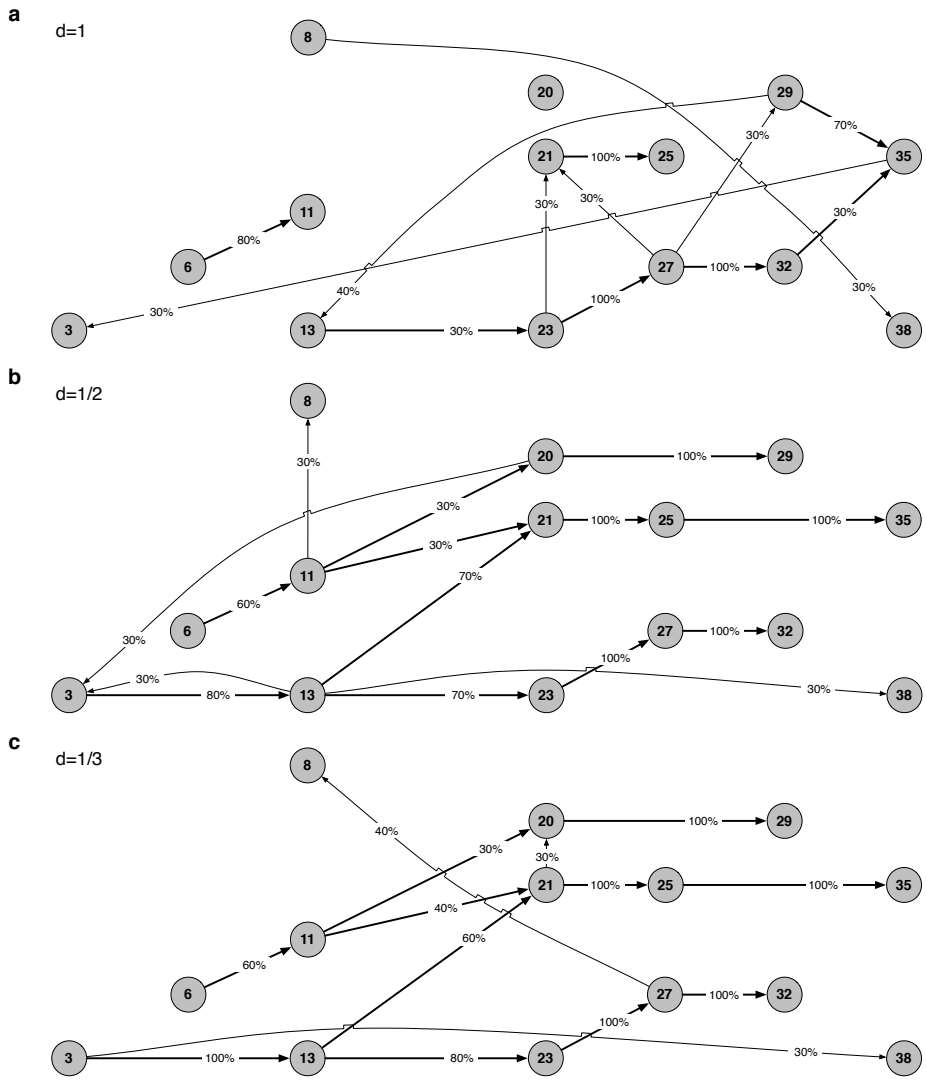


Fig. 9 (part 1) Average of networks learned for data of fixed length (30 seconds) and similar impetuses ($\approx 30\%$), using different decay constants ($d \in \{5^{-1}, 4^{-1}, 3^{-1}, 2^{-1}, 1\}$). Links that can be found in at least 30% of learned networks (for each choice of d) shown with percental frequencies on link. Plausible links are marked bold (Fig. 6b). **a** ($d = 1$) Except for one link ($\underline{13} \rightarrow$ *(18)* $\rightarrow \underline{23}$), the only plausible links recovered are those with lag 1. The decay constant is not small enough to capture dependence over larger time-lags. **b** ($d = 2^{-1}$) Good recovery of plausible links with lags 1 and 2. Link $\underline{13} \rightarrow$ *(18)* $\rightarrow \underline{23}$ recovered with higher percentage than in (a). **c** ($d = 3^{-1}$) Similar recovery of plausible links as in (b), but different implausible links (recovered in $\geq 30\%$ of networks) due to wider lag-window.

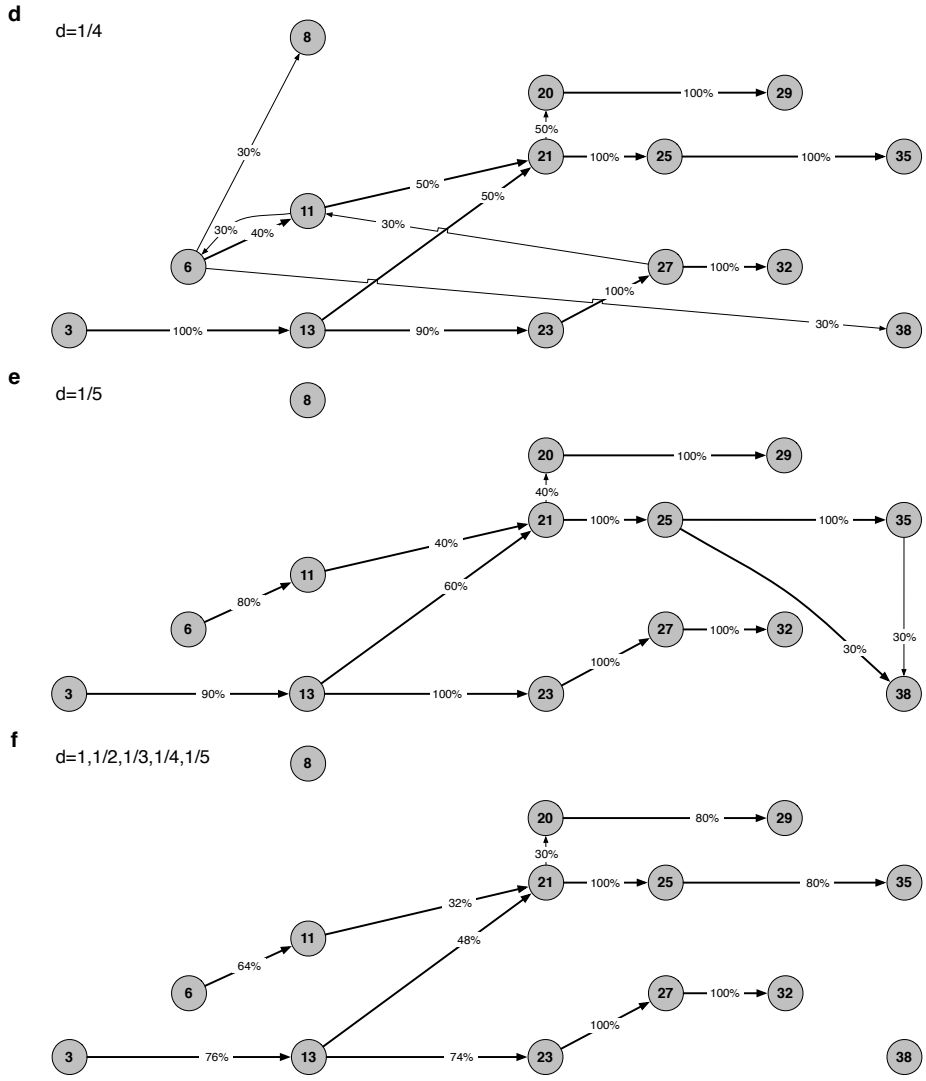


Fig. 9 (part 2) d ($d = 4^{-1}$) Similar recovery of plausible links as in (b) and (c), but more implausible links (recovered in $\geq 30\%$ of networks) than in (b) and (c). **e** ($d = 5^{-1}$) Plausible links recovered as in (b-d) plus link $\underline{25} \rightarrow \underline{38}$, but different implausible links than in (a-d). The differences among implausible links in (a-e) indicate that they are learned from spurious relationships over different time-lags. **f** ($d \in \{5^{-1}, 4^{-1}, 3^{-1}, 2^{-1}, 1\}$) Average over all networks (a-e) shows good recovery of plausible links. Several implausible links learned from spurious relationships average out (frequency below 30%).

sitivity is important for practical applications where doubts about parameter settings exist. In such situations, our simulation results suggest a potential approach for analysis: A fixed shift-constant Δt needs to be chosen (as described above) and should then be used together with a series of decay constants d (as seen in the example). For each of these parameter combinations, networks need to be learned. The decay constant should be gradually decreased,

until a further decrease does not lead to substantial differences in learned networks. This result stabilisation indicates that the decay constant is sufficient, as the resulting lag-window is wide enough to capture all relevant dependencies. Practically, the average of learned networks can indicate which links are consistently found (Fig. 9f). The largest decay constant for which (most of) these links are learned is optimal. (A decay of $d = 1/2$ would thus be a reasonable choice for Fig. 9.) In conclusion, parameters in practical application can be determined in three steps: First, based on the experimenter's knowledge, the shift constant Δt is selected. In doubt, Δt should be chosen smaller rather than larger in order to prevent ignoring relationships with short time lags. In the second step, networks are learned from the data using a series of decay constants d. Finally, the learned networks are analysed and compared with respect to each other, in order to infer the optimal decay for which the width of the lag-window will be minimal, but sufficient to represent identified relations in the data. As discussed before, the shift- and decay-constant determine the interpretation of learned networks.

We would like to emphasise again that the classification of links as plausible or implausible is not restricted to be used for assessment of the SSS. The formalised concept of plausibility can generally account for situations where partial observability results in (multiple) causal explanations that are not present in the full network; such explanation is plausible if it is consistent with the full network. The presented idea can be used for simulation-based assessments of any recovery method under conditions of partial observability. We see two important advantages in using the plausibility approach: (1) The formal specification about expected results is independent of subjective classification of links, and (2) The algorithmic implementation of the concept facilitates a fully automatic assessment process of many different networks. Both points arise from the fact that the plausibility approach minimises manual interaction in the assessment of results. Because it is neither specific to the simulated neuron model nor the technique used to reveal networks from data, it can thus be used for large scale comparisons between different methods.

4 Discussion and Conclusions

In this paper, we presented a novel analysis method for multi-channel spike train data, which can detect stochastic relations over multiple time-lags. Simulations were used to demonstrate the method's ability to cope with different impetuses of the studied system. In comparison to an optimally performing cross-correlation analysis over different time-lags, the SSS showed superior performance in most cases. Examples have demonstrated the SSS's preference of short and few connections. However, modifications of our method can change the preferred connection type. For example, changing the activity level calculation[13] can alter the score's preference of connection lengths (e.g. preferring moderately long connections over both longer or shorter ones). Accordingly, changing the join-operation by replacing the *maximum* function in (2) with the *minimum* or *average* of inputs, for example, affects the favoured integration of multiple inputs (*maximum*: logical or; *minimum*: logical and; *average*: mean). The diversity of potential modifications indicates that the SSS can be adapted to other time series data as well (e.g. calcium imaging [73], fMRI [36,50], EEG [55]). However, we do not discuss such adaptations in this paper.

[13] Activity level series correspond to rate-limited firing rate series that were computed using a causal kernel. A *causal kernel* function f satisfies $\mathrm{supp}(f) \cap \mathbb{R}_{<0} = \{x \in \mathbb{R} | f(x) > 0\} \cap \mathbb{R}_{<0} = \emptyset$ [18, p.14]. Such a kernel does not make use of *information gained in the future*. The activity level of channel k at time t is given by $a_{k,t} = \sum_{j=0,\ldots,t} f(t - j \cdot s_{k,t-j})$.

In the special case $d = 1$, $\Delta t = 1$, activity decays completely within one time-bin, such that the activity level series equals the spike train it is calculated from. The lag-window W is thus of width 1, rendering the process $\mathbf{X}_t$ describing the spike trains a 1st order Markov process, such that (6) reduces to

$$P\left(X_{t+1}^{(i)} | pa(X^{(i)})_t\right) . \tag{7}$$

In this special case, the SSS can be related to another network learning approach: Dynamic Bayesian networks (DBNs) restricted to 1st Markov order can encode such dependencies [52,30,53]. Increasing the Markov order of DBNs raises computational costs for learning such networks, whereas a widened lag-window does not affect the costs of our method. However, it is only in the special case ($d = 1$, $\Delta t = 1$) where the nature of resulting networks from both approaches match.

DBN learning approaches have been applied to biological data of different kinds, e.g. gene expression profiles [52,24,61,42,79], fMR images [38,46,64,63,10], and also neural electrophysiological multi-unit data [70]. So far the only publication of which we are aware applying DBNs to (simulated) spike train data is Eldawlatly et al. [21], who used a coarse representation of 3 msec time-bins for the data. We speculate that the sparsity of DBN learning approaches applied to spike trains is due to the scoring functions that are commonly used (e.g. the Bayesian Dirichlet (BD) score [15,34], Bayesian Dirichlet equivalence (BDe) score [34], or the minimal description length (MDL) [44], which is equivalent [23] to the Bayesian information criterion (BIC) [69]). These scores are not specifically adapted to any data-type, i.e. states of variables do not possess any semantics and are thus treated equally. This can be problematic for their application to spike train data, as inspection of the BD scores shows: these scores favour parent-child relations for which parents' states are good predictors of the child's states most of the time (irrespective of their semantics). The enormous imbalance in the number of spiking to non-spiking bins can cause the BD scores to favour parent-child relations for which the non-spiking of the parents can predict non-spiking of the child; occasional spikes are basically ignored, as they are clearly outnumbered. In practice, this causes search heuristics to deliver unstable results (unpublished observations using the BDe score). However, this is neither a failure of the search heuristic nor the scoring function, as can be seen in the most extreme situation where all variables are constants: Any combination of parents $pa(X^{(i)})$ can then be used to predict $X^{(i)}$ in (7) equally well. For neural data with rare spike events – appearing to be nearly constant – it is thus likely that several equally high scoring networks exist, although their connectivity might substantially differ. Therefore learning DBNs from single unit spike train data may require a scoring function which is particularly adapted by accounting for the semantics of states and weighting them accordingly. While the model underlying the BD scores is fixed, the MDL principle (or BIC equivalently) can be applied with a variety of paradigms. It thus seems likely that they can be adopted to data with low spiking rates in order to infer DBNs.

The SSS has been specifically designed for spike train data and assigns high score values to networks if parent nodes' spikes are good predictors of child spikes. In detail: High scores are assigned to parents whose spikes precede those of the child reliably with the right timing - at the beginning of the lag-window. How well the timing matches is determined by snapshots of the parents' activity level, which are normalised in order to control for random matchings due to constantly high activity of the parents. Periods in the data in which neither the child nor its parents show any activity do not affect the score value. This is because (1) no snapshots are triggered and (2) a zero activity level does not affect normalisation. Thus, spiking and non-spiking bins in the data are distinguished, and this enables the SSS to

account for low spiking rates and the biological significance of action potentials. The design of the score further places emphasis on computational efficiency, in order to facilitate evaluation of large data-sets of higher dimension, as is required for realistic data sets. The score can be used to learn DBNs or dependencies like shown in (6) at the same computational cost. The revealed stochastic relations are visualised as excitatory causal networks to assist the interpretation of the data. Such a method would be useful for other neural time series data types as well and, as indicated earlier, modifications to the activity level calculation could render an appropriate adaptation of the SSS possible.

Further research is needed to fully investigate the score's characteristics and potential usage. In particular, neural simulations of different network topologies and scales are necessary to picture and understand the score. Additionally, it will be interesting to see how the SSS concept can be harnessed for the detection of inhibitory effects; likewise, it might prove useful for synchrony detection. In order to apply the SSS to different types of time series data, adequate transformation of these data to activity levels and snapshot triggers need to be developed. Future work will address these questions.

The Python [77] implementation of the SSS is available for download as supplementary information and from the web-site `http://biology.st-andrews.ac.uk/vannesmithlab`. In the future, the method will also be provided as a web-service on the platform built by the *Code Analysis, Repository and Modelling for E-Neuroscience* (CARMEN) project.

Acknowledgements We thank two anonymous reviewers for a thorough review and helpful comments. This work was supported by the CARMEN e-science project (www.carmen.org.uk) funded by the EPSRC (EP/E002331/1).

References

1. L. F. Abbott. Lapicque's introduction of the integrate-and-fire model neuron (1907). *Brain research bulletin*, 50(5-6):303–4, 1999.
2. A. M. H. J. Aertsen, G. L. Gerstein, M. K. Habib, and G. Palm. Dynamics of neuronal firing correlation - modulation of effective connectivity. *Journal of Neurophysiology*, 61(5):900–917, 1989.
3. E. M. Airoldi. Getting started in probabilistic graphical models. *PLoS Computational Biology*, 3(12):e252, 2007.
4. D. Ashlock. *Evolutionary Computation for Modeling and Optimization*. Springer, 2004.
5. L. Astolfi, F. Cincotti, D. Mattia, M. G. Marciani, L. A. Baccalá, F. D. Fallani, S. Salinari, M. Ursino, M. Zavaglia, and F. Babiloni. Assessing cortical functional connectivity by partial directed coherence: Simulations and application to real data. *IEEE Transactions on Biomedical Engineering*, 53(9):1802–1812, 2006.
6. L. A. Baccalá and K. Sameshima. Partial directed coherence: a new concept in neural structure determination. *Biological Cybernetics*, 84(6):463–74, 2001.
7. T. Bäck. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, USA, 1996.
8. A. Borst and F. E. Theunissen. Information theory and neural coding. *Nature Neuroscience*, 2(11):947–57, 1999.
9. E. N. Brown, R. E. Kass, and P. P. Mitra. Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nature Neuroscience*, 7(5):456–461, 2004.
10. J. Burge, T. Lane, H. Link, S. Qiu, and V. P. Clark. Discrete dynamic Bayesian network analysis of fMRI data. *Human Brain Mapping*, 30(1):122–37, 2009.
11. A. J. Cadotte, T. B. DeMarse, P. He, and M. Ding. Causal measures of structure and plasticity in simulated and living neural networks. *PLoS ONE*, 3(10):e3355, 2008.
12. G. Casella and E. I. George. Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.
13. V. Cerny. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51, 1985.

14. E. S. Chornoboy, L. P. Schramm, and A. F. Karr. Maximum-likelihood identification of neural point process systems. *Biological Cybernetics*, 59(4-5):265–275, 1988.
15. G. F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992.
16. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Greedy algorithms. In *Introduction to Algorithms*, pages 370–404. MIT Press, 2nd edition, 2001.
17. R. T. Cox. Probability, frequency and reasonable expectation. *American Journal of Physics*, 14(1):1–13, 1946.
18. P. Dayan and L. F. Abbott. *Theoretical neuroscience: computational and mathematical modeling of neural systems*. The MIT Press, 1st paperback edition, 2005.
19. R. Eberhart, Y. Shi, and J. Kennedy. *Swarm Intelligence*. Artificial Intelligence. Morgan Kaufmann, 2001.
20. M. Eichler. On the evaluation of information flow in multivariate systems by the directed transfer function. *Biological Cybernetics*, 94(6):469–82, 2006.
21. S. Eldawlatly, Y. Zhou, R. Jin, and K. Oweiss. Reconstructing functional neuronal circuits using dynamic Bayesian networks. In *30th Annual International IEEE Engineering in Medicine and Biology Society (EMBS) Conference*, volume 2008, pages 5531–4, Vancouver, British Columbia, Canada, 2008.
22. W. Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. John Wiley and Sons, 3rd edition, 1950.
23. N. Friedman. Learning belief networks in the values and hidden variables. In *14th International Conference on Machine Learning (ICML 1997)*, pages 125–133, Nashville, Tennessee, USA, 1997. Morgan Kaufmann.
24. N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3-4):601–620, 2000.
25. K. J. Friston. Functional and effective connectivity in neuroimaging: A synthesis. *Human Brain Mapping*, 2:56–78, 1994.
26. G. L. Gerstein and A. M. Aertsen. Representation of cooperative firing activity among simultaneously recorded neurons. *Journal of Neurophysiology*, 54(6):1513–28, 1985.
27. G. L. Gerstein and D. H. Perkel. Simultaneously recorded trains of action potentials: analysis and functional interpretation. *Science*, 164(881):828–30, 1969.
28. G. L. Gerstein, D. H. Perkel, and J. E. Dayhoff. Cooperative firing activity in simultaneously recorded populations of neurons: detection and measurement. *Journal of Neuroscience*, 5(4):881–9, 1985.
29. W. Gerstner and W. M. Kistler. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, Cambridge, 1st edition, 2002.
30. Z. Ghahramani. Learning dynamic Bayesian networks. *Adaptive Processing of Sequences and Data Structures*, 1387:168–197, 1998.
31. C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, 1969.
32. W. K. Hastings. Monte-Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
33. D. O. Hebb. *The Organization of Behavior*. Wiley, New York, 1949.
34. D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks - the combination of knowledge and statistical-data. *Machine Learning*, 20(3):197–243, 1995.
35. M. O. Heuschkel, M. Fejtl, M. Raggenbass, D. Bertrand, and P. Renaud. A three-dimensional multi-electrode array for multi-site stimulation and recording in acute brain slices. *Journal of Neuroscience Methods*, 114(2):135–48, 2002.
36. P. Jezzard, P. M. Matthews, and S. M. Smith. *Functional MRI: An Introduction to Methods*. Oxford University Press, Oxford, 1st edition, 2001.
37. J. L. Johnson and J. P. Welsh. Independently movable multielectrode array to record multiple fast-spiking neurons in the cerebral cortex during cognition. *Methods*, 30(1):64–78, 2003.
38. Z. W. Junning Li and M. McKeown. Dynamic Bayesian networks (DBNs) demonstrate impaired brain connectivity during performance of simultaneous movements in Parkinson's disease. In *3rd IEEE International Symposium on Biomedical Imaging: Nano to Macro*, pages 964–967, 2006.
39. M. J. Kaminski and K. J. Blinowska. A new method of the description of the information flow in the brain structures. *Biological Cybernetics*, 65(3):203–10, 1991.
40. J. Kennedy and R. Eberhart. Particle swarm optimization. In *IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948, Perth, WA, Australia, 1995.
41. B. W. Kernighan and D. M. Ritchie. *The C Programming Language*. Prentice Hall, 2nd edition, 1988.
42. S. Kim, S. Imoto, and S. Miyano. Dynamic Bayesian network and nonparametric regression for nonlinear modeling of gene networks from time series gene expression data. *Biosystems*, 75(1-3):57–65, 2004.

43. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
44. W. Lam and F. Bacchus. Learning Bayesian belief networks: an approach based on the MDL principle. *Computational Intelligence*, 10(3):269–293, 1994.
45. S. L. Lauritzen. *Graphical Models*. Oxford University Press, Oxford, 1996.
46. J. Li, Z. J. Wang, and M. J. McKeown. A framework for group analysis of fMRI data using dynamic Bayesian networks. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5992–5, 2007.
47. B. G. Lindsey and G. L. Gerstein. Two enhancements of the gravity algorithm for multiple spike train analysis. *Journal of Neuroscience Methods*, 150(1):116–127, 2006.
48. D. Madigan and A. E. Raftery. Model selection and accounting for model uncertainty in graphical models using Occam's window. *Journal of the American Statistical Association*, 89(428):1535–1546, 1994.
49. V. A. Makarov, F. Panetsos, and O. de Feo. A method for determining neural connectivity and inferring the underlying network dynamics using extracellular spike recordings. *Journal of Neuroscience Methods*, 144(2):265–79, 2005.
50. P. M. Matthews and P. Jezzard. Functional magnetic resonance imaging. *Journal of Neurology Neurosurgery and Psychiatry*, 75(1):6–12, 2004.
51. N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculation by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.
52. K. Murphy and S. Mian. Modelling gene expression data using dynamic Bayesian networks. Technical report, MIT Artificial Intelligence Laboratory, 1999.
53. K. P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, 2002.
54. J. Neyman and E. S. Pearson. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London*, Series A, 231:289–337, 1933.
55. P. L. Nunez and R. Srinivasan. Electroencephalogram. *Scholarpedia*, 2(2):1348, 2007.
56. D. Q. Nykamp. Revealing pairwise coupling in linear-nonlinear networks. *SIAM Journal on Applied Mathematics*, 65(6):2005–2032, 2005.
57. H. Oka, K. Shimono, R. Ogawa, H. Sugihara, and M. Taketani. A new planar multielectrode array for extracellular recording: application to hippocampal acute slice. *Journal of Neuroscience Methods*, 93(1):61–7, 1999.
58. M. Okatan, M. A. Wilson, and E. N. Brown. Analyzing functional connectivity using a network likelihood model of ensemble neural spiking activity. *Neural Computation*, 17(9):1927–1961, 2005.
59. J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, UK, 2000.
60. D. H. Perkel, G. L. Gerstein, and G. P. Moore. Neuronal spike trains and stochastic point processes. II. Simultaneous spike trains. *Biophysical Journal*, 7(4):419–40, 1967.
61. B. E. Perrin, L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet, and F. d'Alche Buc. Gene networks inference using dynamic Bayesian networks. *Bioinformatics*, 19 Suppl 2:ii138–48, 2003.
62. J. W. Pillow, J. Shlens, L. Paninski, A. Sher, A. M. Litke, E. J. Chichilnisky, and E. P. Simoncelli. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454(7207):995–U37, 2008.
63. J. C. Rajapakse, Y. Wang, X. Zheng, and J. Zhou. Probabilistic framework for brain connectivity from functional MR images. *IEEE Transactions on Medical Imaging*, 27(6):825–33, 2008.
64. J. C. Rajapakse and J. Zhou. Learning effective brain connectivity with dynamic Bayesian networks. *Neuroimage*, 37(3):749–60, 2007.
65. F. Rieke, D. Warland, R. d. R. van Steveninck, and W. Bialek. *Spikes: exploring the neural code*. MIT Press, 1st paperback edition, 1999.
66. C. P. Robert and G. Casella. The multi-stage Gibbs sampler. In *Monte Carlo statistical methods*, pages 337–370. Springer, 2nd edition, 2004.
67. K. Sameshima and L. A. Baccalá. Using partial directed coherence to describe neuronal ensemble interactions. *Journal of Neuroscience Methods*, 94:93–103, 1999.
68. T. Sato, T. Suzuki, and K. Mabuchi. A new multi-electrode array design for chronic neural recording, with independent and automatic hydraulic positioning. *Journal of Neuroscience Methods*, 160(1):45–51, 2007.
69. G. Schwarz. Estimating dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.
70. V. A. Smith, J. Yu, T. V. Smulders, A. J. Hartemink, and E. D. Jarvis. Computational inference of neural information flow networks. *PLoS Computational Biology*, 2(11):e161, 2006.
71. O. Sporns, D. R. Chialvo, M. Kaiser, and C. C. Hilgetag. Organization, development and function of complex brain networks. *Trends in Cognitive Sciences*, 8(9):418–25, 2004.
72. R. B. Stein. A theoretical analysis of neuronal variability. *Biophysical Journal*, 5:173–94, 1965.

73. C. Stosiek, O. Garaschuk, K. Holthoff, and A. Konnerth. In vivo two-photon calcium imaging of neuronal networks. *Proceedings of the National Academy of Sciences of the United States of America*, 100(12):7319–7324, 2003.
74. D. Y. Takahashi, L. A. Baccalá, and K. Sameshima. Connectivity inference between neural structures via partial directed coherence. *Journal of Applied Statistics*, 34(10):1259–1273, 2007.
75. W. Truccolo, U. T. Eden, M. R. Fellows, J. P. Donoghue, and E. N. Brown. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of Neurophysiology*, 93(2):1074–1089, 2005.
76. V. Tsytsarev, M. Taketani, F. Schottler, S. Tanaka, and M. Hara. A new planar multielectrode array: recording from a rat auditory cortex. *Journal of Neural Engineering*, 3(4):293–8, 2006.
77. G. van Rossum and et al. Python language website, http://www.python.org/.
78. D. Whitley. A genetic algorithm tutorial. *Statistics and Computing*, 4(2):65–85, 1994.
79. M. Zou and S. D. Conzen. A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics*, 21(1):71–9, 2005.