Statistical Analysis of Pairwise Connectivity

Georg Krempl[1](✉), Daniel Kottke[2], and Tuan Pham[2]

[1] Utrecht University, 3584 CC Utrecht, The Netherlands
g.m.krempl@uu.nl
[2] Kassel University, 34121 Kassel, Germany
{daniel.kottke,tuan.pham}@uni-kassel.de
https://www.uu.nl/staff/GMKrempl

Abstract. Analysing correlations between streams of events is an important problem. It arises for example in Neurosciences, when the connectivity of neurons should be inferred from spike trains that record neurons' individual spiking activity. While recently some approaches for inferring delayed synaptic connections have been proposed, they are limited in the types of connectivities and delays they are able to handle, or require computation-intensive procedures. This paper proposes a faster and more flexible approach for analysing such delayed correlated activity: a statistical **A**nalysis of the **C**onnectivity of spiking **E**vents (ACE), based on the idea of hypothesis testing. It first computes for any pair of a source and a target neuron the inter-spike delays between subsequent source- and target-spikes. Then, it derives a null model for the distribution of inter-spike delays for *uncorrelated* neurons. Finally, it compares the observed distribution of inter-spike delays to this null model and infers pairwise connectivity based on the Pearson's χ^2 test statistic. Thus, ACE is capable to detect connections with a priori unknown, non-discrete (and potentially large) inter-spike delays, which might vary between pairs of neurons. Since ACE works incrementally, it has potential for being used in online processing. In an experimental evaluation, ACE is faster and performs comparable or better than four baseline approaches, in terms of AUPRC (reported here), F1, and AUROC (reported on our website), for the majority of the 11 evaluated scenarios.

Keywords: Machine learning from complex data · Event streams · Neurosciences · Neural connectomics · Connectivity inference

1 Introduction

An important problem in various applications is detecting correlations between streams of events. This is of particular importance in Neurosciences, where it arises for example when inferring the functional connectivity of neurons [PGM67]. Given spike trains with recordings of the neuron's individual spike activity, the objective is to detect correlations between the spike activities of pairs or networks of neurons. Most of the existing approaches are designed for

C. Soares and L. Torgo (Eds.): DS 2021, LNAI 12986, pp. 138–148, 2021.
https://doi.org/10.1007/978-3-030-88942-5_11

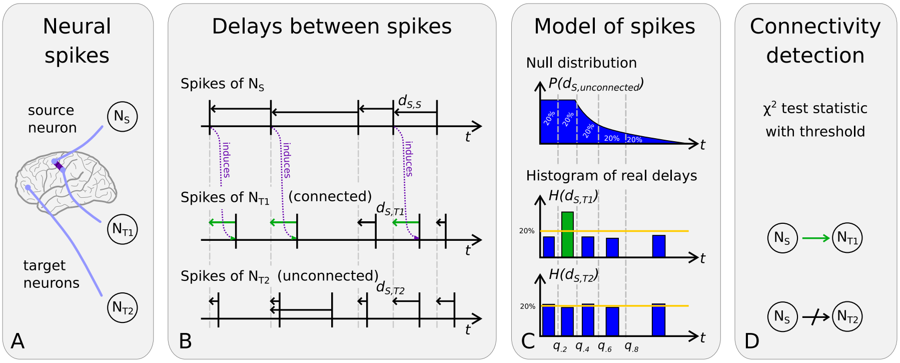

Fig. 1. Schematic visualisation of ACE (**A**nalysis of **C**onnectivity of spiking **E**vents).

detection of correlated synchronous activity, considering solely events within the same discretised time interval. More recently, the detection of delayed synaptic connections has gained attention. However, existing methods have limitations in the types of connectivities and delays they are able to handle, for example due to requiring range-parameters for expected delays, or they require computation-intensive procedures, such as computing cross-correlation histograms or performing cross-evaluations of parameter values.

We propose a faster and more robust approach for the detection of delayed correlated activity. The principle of this statistical **A**nalysis of **C**onnectivity of spiking **E**vents (ACE) of neural spikes is illustrated in Fig. 1. This statistical approach follows the idea of hypothesis testing: Starting (A) with data in the form of spike trains that are recorded for several neurons, the aim is to infer for any pair of source (N_S) and target neuron (N_T) the pairwise connectivity between them. For this purpose, we compute in step (B) for any pair of neurons the inter-spike intervals, i.e. the delays between their subsequent spikes ($d_{S,S}$, $d_{S,T1}$ and $d_{S,T2}$). In the third step (C), we use the inter-spike intervals of a potential source neuron to determine the null distribution of delays for *unconnected* target neurons (shown on top as $P(d_{S,unconnected})$). If a target neuron is *not* connected to the source neuron, the observed distribution of the inter-spike intervals should follow this distribution (shown on the bottom as $H(d_{S,T2})$). In contrast, if this observed distribution (shown at the center as $H(d_{S,T1})$) differs sufficiently, we assume that these two neurons are connected. Thus, in step (D) we use the Pearson's χ^2 test statistic to determine the connectivity for each pair of neurons.

As a consequence, and in contrast to existing methods, our approach works incrementally. It requires neither a cross-correlation histogram (faster), nor prespecified range-parameters for expected delays (more flexible), but assumes that the signal reaches the target neuron faster than the source neuron fires again. This makes it particularly interesting for online processing. Summarising, the contributions of our statistical **ACE** approach are:

- A statistical, principled approach based on hypothesis testing, by modelling the null distribution of delays between pairs of *unconnected* neurons,
- which is fast and robust (no sensitive parameters),
- does not assume a particular type of connectivity pattern,
- capable to detect connections with unknown, non-discrete delays, that might vary in length between connections.
- Based on the distribution of the F1 score for different datasets with similar neuron types (with known ground truth), experiments indicate that the threshold is a robust parameter (transferable to other data sets, see [KKP20]).

2 Related Work

There is a rich literature on analysing neuronal spike train data for inferring connectivity. This comprises recent reviews, e.g., [MYD18], and a recent machine learning challenge on neural connectomics [BGL+17]. Following [MYD18], we distinguish model-based approaches from model-free ones. An important limitation of model-based approaches is that they rely on assumptions on the data generating process. Their structure and function depend on a large number of factors, resulting in a variety of models and approaches [MYD18]. These include autoregressive models, which are fast but assume a directed linear interaction and generalised linear models, which despite recent extensions to handle transmission delays remain limited to small and uniform delays within the network.

In contrast, model-free approaches rely on principles from descriptive statistics, information theory, and supervised learning. This comprises approaches based on the correlation between the activities of neurons, which was the key component in the winning approach of the neural connectomics challenge [BGL+17]. While in the simplest form only simultaneous spikes are considered, the extension to cross correlation allows for a delay τ between spike times [IHH+11]. Extending the idea presented at the neural connectomics challenge, [Moh14] suggested to use inverse covariance estimates together with an initial convolution filter to preprocess the data. The convolution kernel and other parameter are learned by optimising the binomial log-likelihood function [Moh14] on a training data set, where ground truth is known. In the above mentioned challenge they report comparable AUC scores and accuracy compared to the winner, while being noticeably faster.

A further group of model-free approaches are based on information theoretic approaches. Their applicability for neural connectomics was investigated in [GNMM09], comparing methods based on Mutual Information, Joint-Entropy, Transfer Entropy (TE) and Cross-Correlation. This study revealed Transfer Entropy and Joint-Entropy being the best of the aforementioned methods. Transfer Entropy (TE) is equivalent to Granger causality for Gaussian variables [BBS09], which describes a statistical hypothesis test that measures the ability of predicting future events of a time series based on past events of related time series. However, pairwise Granger causality only detects direct correlations. This leads to problems when two neurons are driven by a common third neuron with different delays [CDHD08]. In [IHH+11], two different TE-based approaches are proposed, one using Delayed TE (DTE) and one using Higher Order TE (HOTE). DTE calculates the TE for multiple delays (e.g. 1–30 ms), which is extended in HOTE by considering multiple bins for each delay.

Recently, statistical methods for analysing synchrony across neurons were reviewed in [HAK13]. Therein, it is emphasised that the statistical identification of synchronous spiking presumes a null model that describes spiking without synchony. Then, if the observed synchrony is not consistent with the distribution under the null model, i.e. there is more synchony than expected by chance, the null hypothesis of no synchrony is rejected. For hypothesis testing, [HAK13] focus their discussion on deriving the test statistics from a cross-correlation histogram (CCH). This shows the observed frequency of different time delays between two spike trains, but is scaling dependent and computationally costly.

Convolutional Neural Networks have also been used [Rom17] to learn connectivity directly from calcium imaging data. This scored worse than the correlation-based approaches, and required extremely high computation time [BGL+17].

3 A Method for Analysing Potential Connectivity in Events

The new method ACE (**A**nalyser for **C**orrelated spiking **E**vents) is a statistical approach following the idea of hypothesis testing. Observing the inter-spike intervals of a potential source neuron, we determine the null distribution of delays for *unconnected* target neurons. If the real observed distribution differ sufficiently (using the Pearson's χ^2 test statistic), we infer that these two neurons are connected. In contrast to existing methods, our approach works incrementally without using a cross-correlation histogram (faster) and without providing range-parameters for expected delays (more flexible).

The complete analysis pipeline of our algorithm is provided in Fig. 1: In the first step, we compute the delays of two consecutive spikes of all neurons to determine the neuron's parameters λ and RP (Sect. 3.1, Fig. 1B (top)). Thereby, we are able to reconstruct the neuron's delay distribution and to determine the null model for the delay distribution between two neurons (Sect. 3.2, Fig. 1C (top)). As we estimate the null model in advance, we have information on the expected delay distribution and can use the histogram with intervals according to the quantiles to estimate the distribution of observed delays (Sect. 3.3). Then, the real delays from the source neurons to the target neurons are determined and the histogram is completed (Fig. 1C (bottom)). The χ^2 statistic provides a score for distinguishing connected and unconnected neurons (Sect. 3.4, Fig. 1D). A threshold is used to determine if the connection score was sufficiently large. A detailed description follows in the next subsections, while the threshold specification is discussed in the evaluation section.

3.1 Modeling Spiking Behaviour of a Single Neuron

As we will show in Sect. 3.2, the null-distribution for the delays between unconnected source and target neurons depends on the distribution of time intervals (delays) d between consecutive spikes of the source neuron (N_S). To model the random variable ($X_{N_S \rightarrow N_S}$) corresponding to these delays, we use an exponential distribution as a simplification of the gamma distribution, in accordance to [Pil09]. This yields two parameters: the refractory period RP, describing the time a neuron is inhibited to spike again, and the firing rate λ, defining the shape of the exponential distribution. The probability density function (pdf) of $X_{N_S \rightarrow N_S} = EXP(\lambda) + RP$ and its first two moments are given as:

$$f_{N_S \rightarrow N_S}(d) = \begin{cases} \lambda \exp(-\lambda(d - RP)) & d \geq RP \\ 0 & d < RP \end{cases} \tag{1}$$

$$E(X_{N_S \rightarrow N_S}) = E(EXP(\lambda) + RP) = E(EXP(\lambda)) + RP = {}^1\!/_\lambda + RP \tag{2}$$

$$V(X_{N_S \rightarrow N_S}) = V(EXP(\lambda) + RP) = V(EXP(\lambda)) = {}^1\!/_{\lambda^2} \tag{3}$$

The expectation value $E(X_{N_S \rightarrow N_S})$ and the variance $V(X_{N_S \rightarrow N_S})$ can be incrementally calculated [BDMO03] to find the values for both parameters RP and λ.

$$\lambda = {}^1\!/\sqrt{V(X_{N_S \rightarrow N_S})} \qquad\qquad RP = E(X_{N_S \rightarrow N_S}) - {}^1\!/_\lambda \tag{4}$$

3.2 Determining the Null-Distribution for Uncorrelated Neurons

Our approach follows the idea of a statistical test: Instead of deriving models for cases when a source neuron (N_S) is connected to a target neuron (N_T), we develop a model to describe the delays d if N_S and N_T are *unconnected*. Observing a spike at the target neuron at time t, and knowing the time t_S of the source neuron's last spike, this delay is $d = t - t_S$.

If the source neuron (N_S) is not connected to the target neuron (N_T), spikes of N_T seem to appear randomly from the perspective of N_S as they are *independent*. Instead of using the real spike time points, we could also use an equal number of randomly chosen time points. Thus, the null-distribution solely depends on the firing frequency of the source neuron, which is defined by its parameters RP and λ. Determining the distribution of delays $d = t - t_S$ corresponds to estimating the probability $P(X_{N_S \rightarrow N_S} > d)$ that N_S has not spiked again within $[t_s, t]$:

$$P(X_{N_S \rightarrow N_S} > d) = 1 - \int_0^d f_{N_S \rightarrow N_S}(d')\, \mathrm{d}d' = \begin{cases} \exp(-\lambda(d - RP)) & d \geq RP \\ 1 & 0 \leq d < RP \end{cases} \tag{5}$$

$$\int_0^\infty P(X_{N_S \rightarrow N_S} > d)\, \mathrm{d}d' = RP + \frac{1}{\lambda} \int_0^\infty \lambda \exp(-\lambda d)\, \mathrm{d}d' = RP + \frac{1}{\lambda} \tag{6}$$

Using the normalised probability from above, we obtain the distribution $X_{N_S \rightarrow N_?}$ of delays between N_S and an unconnected neuron $N_?$:

$$f_{N_S \rightarrow N_?}(d) = \frac{P(X_{N_S \rightarrow N_S} > d)}{\int_0^\infty P(X_{N_S \rightarrow N_S} > d')\, \mathrm{d}d'} = \begin{cases} \frac{\exp(-\lambda(d - RP))}{RP + {}^1\!/_\lambda} & d \geq RP \\ \frac{1}{RP + {}^1\!/_\lambda} & 0 \leq d < RP \end{cases} \tag{7}$$

Summarising, this null model describes the distribution of delays between two unconnected neurons. Hence, our model is based on (but not similar) to the interspike intervals of neuron N_S which depends on the refractory period (RP) and the firing rate (λ).

3.3 Estimating the Distribution of Observed Delays

To compare the true distribution of observed delays to the distribution under the null model, we build a histogram with B bins such that every bin should contain the same amount of delays following the null distribution. The delay interval of bin $b \in \{1, \ldots, B\}$ is given in Eq. 8 with F^{-1} being the quantile function (inverse cumulative distribution function) of the null distribution:

$$\mathcal{I}_b = \left[F^{-1}\left(\frac{b-1}{B}\right), F^{-1}\left(\frac{b}{B}\right) \right[\tag{8}$$

Given the source neuron's RP and λ, this quantile function is:

$$F^{-1}(q) = \begin{cases} RP - \frac{\ln\left(1-(q-\frac{RP}{RP+1/\lambda})\cdot(\lambda RP+1)\right)}{\lambda} & q > \frac{RP}{RP+1/\lambda} \\ q \cdot (RP + 1/\lambda) & q \leq \frac{RP}{RP+1/\lambda} \end{cases} \tag{9}$$

3.4 Infering Connectivity Using the Pearson's χ^2-test Statistic

Following the null hypothesis (neurons are not connected), the previously mentioned histogram should be uniformly distributed. Hence, the frequencies H_b of bin b should be similar to $H_b \approx N/B$ ($N = \sum H_b$, which is the total number of delays). Our method uses the Pearson's χ^2-Test statistic to find a threshold for distinguishing connected and unconnected neurons.

$$\chi^2 = \sum_{b=1}^{B} \frac{(H_b - \frac{N}{B})^2}{\frac{N}{B}} \tag{10}$$

Instead of calculating the p value, we directly use the χ^2 statistic to determine a threshold as the degrees of freedom $(B-1)$ are similar for every pair of neurons.

4 Experimental Evaluation

We evaluate our algorithm to the most used baseline techniques [MYD18] regarding its detection quality and its robustness to parameters like the detection threshold. All code and data are available at our repository[1].

[1] https://bitbucket.org/geos/ace-public.

4.1 Baseline Algorithms and Performance Scores

We compare our algorithm with the method proposed by [Moh14] (denoted as *IC*), which is as the winner of the Neural Connectomics Challenge based on inverse covariance but faster, and the higher order transfer entropy (*HOTE*) approach [IHH+11]. Besides the threshold that all connectivity detection algorithms have in common, both methods require an additional binning parameter that influences the performance. Following the suggestions, we use realistic standard parameters and two variants with 20 and 50 bins. The experiments were implemented in MATLAB (HOTE was provided by the author of [IHH+11]).

As in the challenge and common in literature, we compare the algorithms' connection scores using the Area Under the Precision-Recall Curve (AUPRC) (see [Pow11] and [MYD18]). This describes the relationship between precision $\frac{TP}{TP+FP}$ and recall $\frac{TP}{TP+FN}$ at different thresholds [MYD18].

4.2 Sensitivity of the Algorithms

To evaluate the sensitivity of algorithms to different neural patterns, in our first series of experiments we used artificially generated data with varying neural parameters, which are summarised in Table 1. Each parameter range has been chosen according to animal studies [Izh06] and related comparisons [MYD18].

To evaluate the detection capabilities for each algorithm, we show the **AUPRC scores** in Table 1. Our algorithm outperforms all competitors except for the data set NU_H with higher number of neurons (equal performances with the HOTE approach) and the data set with high delays (DE_H) which is more difficult for all algorithms. To explain our poor performance on the latter, we need to recall that the self-initiated firing rate is between 17 ms and 36 ms, calculated as the interval given in expected latency (default $[10, 25)$) plus the RP ($[7, 11)$) for the DE_H data set. If we observe delays longer than the source neuron's inter-spike intervals (here: $9\,\text{ms} \leq d < 120\,\text{ms}$) it is likely that the source neuron spiked again before its signal reaches the target neuron. Hence, we are not able to link the spike of the source with the spike of the target neuron which makes it impossible to find the respective connection. Fortunately, this behaviour is rare in real neural systems [Izh06].

Unfortunately, it is not possible to set the detection threshold of the algorithms to a fixed value. In this section, we aim to evaluate the sensitivity of the respective parameter mentioned in Table 1. Therefore, we use each of the three different configurations (low, mid, high) as one fold of an experiment. For each fold, we tune the detection threshold on the remaining folds and calculate the *F1 score* accordingly. Except for the data set with varying delays, our approach shows superior performance than the baseline algorithms although the AUPRC score differences have not been that large. This indicates that our detection score is more robust to changes in the number of neurons (NU), the latency (LA), the number of connections (CO) and noise (NO). The F1 scores for the delay (DE) data sets for our approach ACE are: DE_L 0.6164, DE_M = ST 0.7539 and

DE_H = 0.1100. We see that the previously discussed low performance on DE_H is the reason for the low mean score.

In the experiments, we presented two variants of IC and HOTE as they need an additional parameter which highly influences the results. Our **parameter** B has not such an influence on the performance and the runtime (see also Sect. 4.3) of ACE. This is visualised in Fig. 3 which plots the AUPRC w. r. t. the number of bins B on the ST data set. The plot shows that the performance generally increases with higher resolution, but only marginally beyond $B = 100$, our default for further experiments.

Table 1. Area Under the Precision-Recall Curve values for all algorithms on data sets with varying characteristics. The default values for unvaried characteristics are 100 neurons (in the dataset), [10, 25) ms + RP expected latency (between two consecutive spikes), 1% as relative number of connections, [5, 9) ms delay (between two connected neurons), and +0 ms noise (when determining spike times). The length of the spike stream is 30s and the refractory period (RP) is uniformly between 7 and 11 ms. The data set name is composed by the characteristic abbreviation and a suffix for either low, mid or high. Thus, the setting with 200 Neurons is called NU_H.

	Characteristic	ACE	IC20	IC50	HOTE20	HOTE50
ST	Defaults	**0.8626**	0.2028	0.1409	0.7832	0.7761
NU_L	50 neurons	**0.8519**	0.0834	0.0616	0.6310	0.6186
NU_H	200 neurons	0.9504	0.2586	0.1038	0.9552	**0.9553**
LA_L	[1, 10) ms + RP latency	**0.8652**	0.0219	0.0138	0.5579	0.5546
LA_H	[25, 50) ms + RP latency	**0.8135**	0.2500	0.1430	0.8126	0.8119
CO_L	5‰ connections	**0.8850**	0.2656	0.1620	0.8738	0.8739
CO_H	2% connections	**0.8086**	0.0406	0.0317	0.5952	0.5894
DE_L	[2, 5) ms delay	0.7598	0.2770	0.1528	**0.7671**	0.7670
DE_H	[9, 120) ms delay	0.1334	0.1687	0.1053	0.1505	**0.3669**
NO_M	+[0, 3) ms noise	**0.7838**	0.2248	0.1225	0.7246	0.7224
NO_H	+[0, 5) ms noise	**0.8518**	0.2246	0.1021	0.8322	0.8309

4.3 Computational Complexity

To provide a computational run time complexity bound of ACE, let N denote the number of neurons, M the number of spikes over all neurons, and B the number of bins used in the histogram. ACE's first step is estimating the refractory period RP and firing rate λ for each neuron by iterating once over all its spikes, suming up to $O(M)$ constant time operations for all neurons. Second, for each neuron's bin the frequencies according to its null model are computed, resulting overall in $O(N \cdot B)$. Third, the histograms of all neurons are updated after each spike, requiring to insert the observed delay into the corresponding bin. Using a k-d-tree, this requires overall $O(M \cdot N \cdot \log(B))$. Fourth, the χ^2-test statistic is computed and tested for each pair of neurons, requiring $O(N^2 \cdot B)$ operations. The last two steps dominate, giving an overall complexity of $O(M \cdot N \cdot \log(B))$ or $O(N^2 \cdot B)$, respectively. In practice, the third step might be the bottleneck, as the number of spikes depends on the number of neurons, i.e., $M > N$, and the number of bins is small (e.g., $B = 100$). In contrast, HOTE's time complexity is $O(N^2 \cdot (F \cdot D \cdot R + 2^R))$, with firing rate F, recording duration after discretisation D, and R being the total order $(k + l + 1)$ used in the calculations. IC's time complexity is $O(N^2 \cdot T + N^2 \cdot \log(N))$, with T as number of considered time lags.

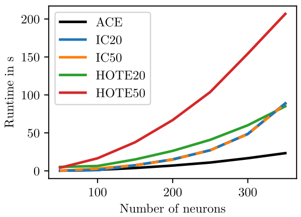

Fig. 2. Runtime over data sets with varying number of neurons.

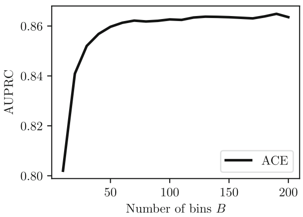

Fig. 3. AUPRC scores of ACE with varying bin sizes on the ST dataset.

Figure 2 shows the empirical runtime[2] of all strategies according to the number of neurons in the data set. Those runtimes are obtained by creating datasets that duplicate the spike trains from NE_L and evaluating them 10 times. One can see that ACE is the fastest. IC20 and IC50 only differ slightly in runtime. HOTE20's runtimes are higher than the ones from IC20 and IC50 for $N \leq 300$. The number of bins affects HOTE's runtime, making HOTE50 the slowest.

5 Conclusion

This paper proposed ACE for detecting correlated, but delayed event patterns in streams, e.g., for detecting delayed connectivity of neurons. Using a null model for the distribution of inter-spike delays of *uncorrelated* neurons, ACE employs principles from hypothesis testing. Against this null-distribution, the distribution of observed inter-spike delays is compared using a Pearson's χ^2 test statistic.

In an experimental evaluation, this algorithm was compared against recently proposed approaches based on inverse covariance and higher order transfer entropy, on data sets with varying characteristics based on our own data generator. On all data sets, ACE is faster and performs comparable or better in terms of AUPRC, F1 and AUROC (see [KKP20]) score, except for networks with very long inter-spike delays that interfere with uncorrelated spike activity. In particular, ACE performs also better on the publicly available, state-of-the-art benchmark data generator with realistic spike characteristics. ACE has only

[2] Experiments were performed using a Intel(R) Core(TM) i7-6820HK CPU @ 2.70 GHz, 16 GB RAM.

two parameters, both being very robust and transferable between data sets of similar characteristic. ACE is fast and flexible, allowing to detect connections with a priori unknown, non-discrete delays, that might vary in length between connections. Furthermore, due to its incremental nature, ACE has potential for being used in online processing.

References

BBS09. Barnett, L., Barrett, A.B., Seth, A.K.: Granger causality and transfer entropy are equivalent for Gaussian variables. Phys. Rev. Lett. **103**(23), 238701 (2009)

BDMO03. Babcock, B., Datar, M., Motwani, R., O'Callaghan, L.: Maintaining variance and k-medians over data stream windows. In: Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pp. 234–243. ACM (2003)

BGL+17. Battaglia, D., Guyon, I., Lemaire, V., Orlandi, J., Ray, B., Soriano, J. (eds.): Neural Connectomics Challenge. TSSCML, Springer, Cham (2017). https://doi.org/10.1007/978-3-319-53070-3

CDHD08. Cadotte, A.J., DeMarse, T.B., He, P., Ding, M.: Causal measures of structure and plasticity in simulated and living neural networks. PloS One **3**(10), e3355 (2008)

GNMM09. Garofalo, M., Nieus, T., Massobrio, P., Martinoia, S.: Evaluation of the performance of information theory-based methods and cross-correlation to estimate the functional connectivity in cortical networks. PloS One **4**(8), e6482 (2009)

HAK13. Harrison, M.T., Amarasingham, A., Kass, R.E.: Statistical identification of synchronous spiking. In: Di Lorenzo, P.M., Victor, J.D. (eds.) Spike Timing: Mechanisms and Function, pp. 77–120 (2013)

IHH+11. Ito, S., Hansen, M.E., Heiland, R., Lumsdaine, A., Litke, A.M., Beggs, J.M.: Extending transfer entropy improves identification of effective connectivity in a spiking cortical network model. PLOS One **6**(11), e27431 (2011)

Izh06. Izhikevich, E.M.: Polychronization: computation with spikes. Neural Comput. **18**(2), 245–282 (2006)

KKP20. Krempl, G., Kottke, D., Minh, T.P.: ACE: a novel approach for the statistical analysis of pairwise connectivity. arXiV (2020)

Moh14. Mohler, G.: Learning convolution filters for inverse covariance estimation of neural network connectivity. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 891–899 (2014)

MYD18. de Abril, I.M., Yoshimoto, J., Doya, K.: Connectivity inference from neural recording data: challenges, mathematical bases and research directions. Neural Netw. **102**, 120–137 (2018)

PGM67. Perkel, D.H., Gerstein, G.L., Moore, G.P.: Neuronal spike trains and stochastic point processes: II. Simultaneous spike trains. Biophys. J. **7**(4), 419–440 (1967)

Pil09. Pillow, J.W.: Time-rescaling methods for the estimation and assessment of non-Poisson neural encoding models. In: Advances in Neural Information Processing Systems, pp. 1473–1481 (2009)

Pow11. Powers, D.M.: Evaluation: from precision, recall and f-measure to ROC, informedness, markedness and correlation. J. Mach. Learn. Technol. **2**(1), 37–63 (2011)

Rom17. Romaszko, L.: Signal correlation prediction using convolutional neural networks. In: Battaglia, D., Guyon, I., Lemaire, V., Orlandi, J., Ray, B., Soriano, J. (eds.) Neural Connectomics Challenge. TSSCML, pp. 47–60. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-53070-3_4