Assembly Detection in Continuous Neural Spike Train Data

Christian Braune[1], Christian Borgelt[2], and Sonja Grün[3,4]

[1] Otto-von-Guericke-University of Magdeburg
Universitätsplatz 2, D-39106 Magdeburg, Germany
[2] European Centre for Soft Computing
Calle Gonzalo Gutiérrez Quirós s/n, E-33600 Mieres (Asturias), Spain
[3] Institute of Neuroscience and Medicine (INM-6), Research Center Jülich, Germany
[4] Theoretical Systems Neurobiology, RWTH Aachen University, Aachen, Germany
christian.braune@ovgu.de, christian@borgelt.net, s.gruen@fz-juelich.de

Abstract. Since Hebb's work on the organization of the brain [16] finding cell assemblies in neural spike trains has become a vivid field of research. As modern multi-electrode techniques allow to record the electrical potentials of many neurons in parallel, there is an increasing need for efficient and reliable algorithms to identify assemblies as expressed by synchronous spiking activity. We present a method that is able to cope with two core challenges of this complex task: *temporal imprecision* (spikes are not perfectly aligned across the spike trains) and *selective participation* (neurons in an ensemble do not all contribute a spike to all synchronous spiking events). Our approach is based on modeling spikes by influence regions of a user-specified width around the exact spike times and a clustering-like grouping of similar spike trains.

Keywords: spike train, ensemble detection, Hebbian learning, continuous data, multidimensional scaling

1 Introduction and Motivation

Modern multi-electrode arrays (MEAs) allow to record electrical potentials simultaneously at many positions in a brain area [5]. In the raw recordings *spikes* (i.e. brief, sharp increases in the extracellular electrical potentials recorded by the MEAs, also called *action potentials*) are detected and processed by so-called *spike sorting* [18] into the spikes of individual neurons according to their shapes and amplitudes. The result is formally a set of point processes (lists of spike times, one per neuron), which is called *parallel spike trains*.

Recordings of parallel spike trains are essential if one tries to understand how a (biological) neural network encodes and processes information. One of many competing theories is that information is encoded and processed by *temporal coordination* of spiking activity, in particular, *synchronous spiking activity*, a theory that was introduced by D.O. Hebb [16]. If it is correct, groups of neurons, called *cell assemblies*, can be expected to emit spikes in a fairly synchronized fashion. In order to be able to test this theory, efficient and reliable algorithms are needed that can detect such synchronous spiking activity in MEA recordings.

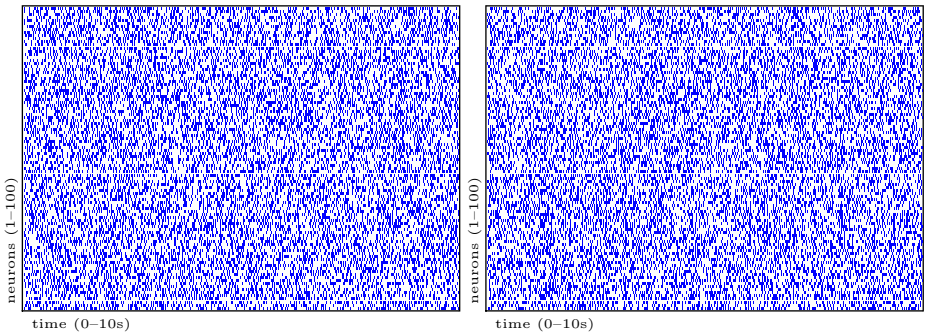


Fig. 1. Two sets of parallel spike trains, one of which shows only random noise (independent stationary Poisson processes, left), while the other (right) exhibits several occurrences of synchronous activity of an assembly of 20 (randomly selected) neurons (stationary Poisson processes with injected coincident spiking events, cf. [14, 13, 3]).

To demonstrate that this is a challenging task, Figure 1 shows *dot displays* of two sets of artificially generated parallel spike trains, one of which exhibits synchronous activity: by pure visual inspection it is essentially impossible to discover the assembly, because the 20 assembly neurons are randomly selected from the total of 100 neurons. However, detection algorithms face two even harder challenges, which are not present in the data shown in Figure 1: *temporal imprecision*, that is, spikes cannot be expected to be perfectly aligned across the spike trains, and *selective participation*, that is, the neurons in an assembly cannot all be expected to contribute a spike to all synchronous spiking events of the assembly. The most common approach to deal with the former problem is to discretize the originally continuous signal into time bins of equal length and to consider spikes as synchronous if they lie in the same time bin [12]. This allows further analysis by means of, for example, cross-correlograms [21], with which the correlation of two (discretized) time series is measured and graphically displayed.

However, time binning has severe disadvantages: depending on how the time bin boundaries fall between the spikes, two spikes that are actually very close together in time may be considered as not synchronous (because they end up in different time bins) [12]. On the other hand, spikes that are almost as far apart as the time bin length may still be considered as synchronous if they happen to fall into the same time bin. This can lead to a severe loss and distortion of synchrony information [12]. In this paper, we try to overcome this problem by avoiding time binning. We rather model each spike by an influence region in which other spikes (of other neurons) may be considered as synchronous to it. Based on this model, we then extend the approach presented in [4], which is based on time-binning the data, to a continuous time treatment of spike trains, thus making much better use of the time resolution of the recordings.

2 Related Work

Using pairwise comparisons of neurons or their corresponding spike trains with each other has been shown to yield promising results in several publications over the last years (such as in [10, 8, 25]).

A classical approach to detect joint spiking patterns in parallel spike trains is the so-called *accretion method* [11], which works with time-binned data. Starting from single neurons, this method iteratively accretes neurons into sequences as long as another neuron shows significantly correlated activity (χ^2 test) with the already accreted neurons. However, accretion suffers from several drawbacks: it works on sequences instead of sets, thus incurring high costs from redundant detections (memory consumption, speed), but at the same time may miss assemblies due to a restriction of the branching factor of the search. Alternatives consist in improved approaches that exploit ideas from frequent item set mining (such as [2]) and enumerate all subsets of neurons as long as a certain quality criterion (e.g. minimum support, significant statistical test result) is met. However, the need to enumerate large numbers of sets of neurons (exponential in the number of neurons in the worst case) still makes them computationally costly.

In order to overcome this problem, we proposed a method that interprets time-binned spike trains as binary vectors in a high-dimensional space (one dimension per time bin) [4]. Dimensionality reduction techniques such as Sammon's Mapping [24] or multi-dimensional scaling [6] onto a single dimension (placing new data points onto the real line such that the pairwise distances of the new points resemble the original distances as closely as possible) yield a sorting criterion for the spike trains. Thus, instead of testing all pairs and subsets of neurons, a single traversal of the sorted set of spike trains suffices to identify assemblies.

Still, this method still suffers from a problem called temporal jitter. The spikes produced by the neurons may neither be perfectly timed nor does the recording process ensure that they are recorded at the same time. The previously mentioned methods can cope with this problem only as long as two originally coincident events fall into the same time bin. In [12] a method is proposed that shifts the spike trains against each other to cope with that problem. Several shifts of integer multiples of the sampling resolution are performed and each time the number of (exact) coincidences is counted. The sum over all these coincidences is used as a measure of synchronicity between two trains. Yet, this approach still requires an the spike train to be binned, making this method prone to effects such as clipping. Also we may argue that although a coincidence may have been found in this way, its significance may be less when the offset between the two original spikes is larger.

A different approach considering the likelihood that a spike has been generated as the result of spikes in different spike trains can be found in [19]. Here a maximum-likelihood approach is used to calculate the influence strengths between several spike trains at once. This comes at the cost of an increased computational need, that we want to avoid.

Table 1. Contingency table for two binary vectors A and B.

	B = 0	B = 1	
A = 0	n_{00}	n_{01}	n_{0*}
A = 1	n_{10}	n_{11}	n_{1*}
	n_{*0}	n_{*1}	$N = n_{**}$

Table 2. Different distance measures used for binary vector comparison.

Jaccard [17]	$d_{\text{Jaccard}} = \frac{n_{10}+n_{01}}{n_{11}+n_{10}+n_{01}}$
Tanimoto [23, 26]	$d_{\text{Tanimoto}} = \frac{2(n_{10}+n_{01})}{n_{11}+n_{00}+2(n_{10}+n_{01})}$
Dice [7]	$d_{\text{Dice}} = \frac{n_{10}+n_{01}}{2n_{11}+n_{10}+n_{01}}$
Correlation [9]	$d_{\text{Correlation}} = \frac{1}{2} - \frac{n_{11}n_{00}-n_{01}n_{10}}{2\sqrt{(n_{10}+n_{11})(n_{01}+n_{00})(n_{11}+n_{01})(n_{00}+n_{10})}}$
Yule [27]	$d_{\text{Yule}} = \frac{n_{01}n_{10}}{n_{11}n_{00}-n_{01}n_{10}}$
Hamming [15]	$d_{\text{Hamming}} = \frac{n_{01}+n_{10}}{n_{**}}$

3 Finding Assemblies in Continuous Spike Train Data

Most of the previously mentioned methods rely on time-binning the spike trains and thus suffer from the disadvantages (loss of synchrony information) pointed out above. To cope with this problem, [22] introduces the notion of so-called *influence maps*: intervals in which a spike may have influence. This approach is similar to convolving spike trains with a kernel and using the overlap of such convolved spike trains as a synchrony measure (see, e.g. [20]), but relies on different kernels and can be generalized to more than two spike trains. The overlap of two influence maps measures the amount or degree of synchrony of the two spikes the influence maps were generated from. The highest degree of synchrony can obviously be achieved by two perfectly aligned spikes. The sum of the degrees of synchrony of individual spikes may be used as an indicator for the correlation between two spike trains. We exploit this idea to improve our approach [4].

In the discrete, time-binned case, the (dis-)similarity between two given spike trains, represented as binary vectors with one element per time bin, is computed from the elements of a 2×2 contingency table (see Table 1). This table records how often both, neither, or only one of two considered neurons produce a spike in a time bin. There is an abundance of (dis-)similarity measures that can be defined on such a table, from which we selected the ones listed in Table 2. They cover some of the most interesting features of common binary distance measures.

3.1 Generalized Contingency Tables

The measures listed in Table 2 are designed for binary vectors. However, none of them actually requires the implied restriction to integer entries in the contingency table. Therefore we may generalize them by computing contingency tables with

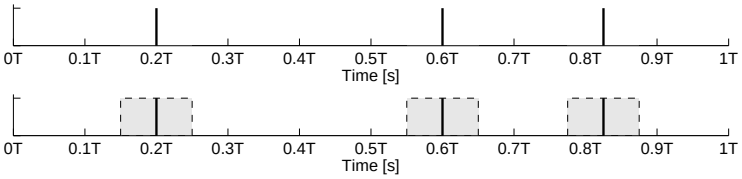


Fig. 2. Continuous spike train with three spikes at $0.2T$, $0.6T$ and $0.825T$, without and with influence maps ($r = 0.05T$).

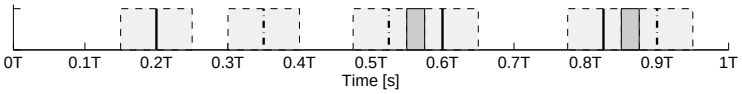


Fig. 3. Two continuous spike trains, one with three spikes at $0.2T$, $0.6T$ and $0.825T$ (solid) and one with spikes at $0.35T$, $0.525T$ and $0.9T$ (dash-dotted). Overlap of influence maps in darker gray.

real-valued entries by exploiting the influence maps introduced in [22]. The fundamental idea may also be viewed as using binary vectors with a super-countably infinite number of elements, one for each time in the recording period.

Formally, let $\boldsymbol{t} = \{t_1, t_2, \ldots, t_n\}, 0 \leq t_i \leq T$, be a sequence of spike times for a given neuron, where T is the recording period. A user now chooses an influence map, that is, a function $f : \mathbb{R} \to \mathbb{R}_0^+$, which describes an influence region in which spikes of other neurons may be considered as synchronous (possibly only to some degree). In principle, these maps may have any shape, but for reasons of simplicity we confine ourselves here to symmetric rectangular functions of a user-specified width r. That is, we consider $\forall t \in \boldsymbol{t}$:

$$f_t : \mathbb{R} \to \{0, 1\}, \quad x \mapsto f_t(x) = \begin{cases} 1, \text{ if } |x - t| \leq \frac{r}{2}, \\ 0, \text{ otherwise,} \end{cases}$$

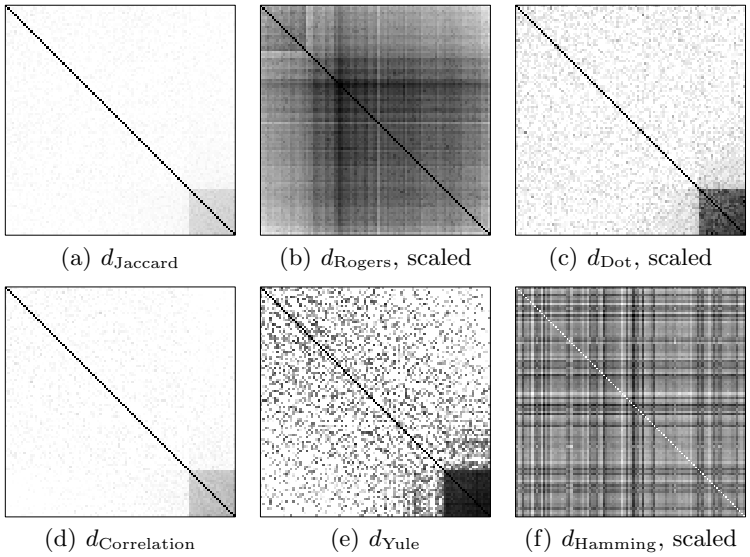
as illustrated in Figure 2. This effectively turns the sequence of spike times into a set of intervals $M_T = \{[a, b] \mid \exists t \in \boldsymbol{t} : a = \max(0, t - \frac{r}{2}) \wedge b = \min(t + \frac{r}{2}, T)\}$. However, we have to take care of the fact that these intervals may overlap, which makes the function view easier to handle. Instead of merging intervals, we combine the functions f_t for all $t \in \boldsymbol{t}$, which yields the function

$$f_{\boldsymbol{t}} : \mathbb{R} \to \{0, 1\}, \quad x \mapsto f_{\boldsymbol{t}}(x) = \max_{t \in \boldsymbol{t}} f_t(x) = \begin{cases} 1, \text{ if } \exists t \in \boldsymbol{t} : |x - t| \leq \frac{r}{2}, \\ 0, \text{ otherwise.} \end{cases}$$

This function describes where spikes of the considered neuron have influence and can be considered synchronous with spikes of other neurons.

Consider now two neurons a and b with their sets $\boldsymbol{t}_a$ and $\boldsymbol{t}_b$ of spike times, which give rise to functions $f_{\boldsymbol{t}_a}$ and $f_{\boldsymbol{t}_b}$. From these we derive the four functions

$$f_{ab}^{(i,j)} : \mathbb{R} \to \{0, 1\}, \quad x \mapsto f_{ab}^{(i,j)}(x) = \begin{cases} 1, \text{ if } f_{\boldsymbol{t}_a} = i \text{ and } f_{\boldsymbol{t}_b} = j, \\ 0, \text{ otherwise,} \end{cases}$$



(a) d_{Jaccard} (b) d_{Rogers}, scaled (c) d_{Dot}, scaled

(d) $d_{\text{Correlation}}$ (e) d_{Yule} (f) d_{Hamming}, scaled

Fig. 4. Distance matrices obtained from the respective distance measures. Distances are encoded on a gray scale, where items have a lower distance the blacker the corresponding square. 100 neurons with an injected assembly of size 20. Data points have been pre-sorted such that the assembly appears in the lower right corner of the plot. Distance measures marked as *scaled* have $d_{\min} = 0$ and $d_{\max} = 1.0$.

for all $(i, j) \in \{0, 1\}^2$, which describe whether the argument x is inside the influence region of a spike for both (if $(i, j) = (1, 1)$), neither (if $(i, j) = (0, 0)$) or only one of the neurons (if $(i, j) = (0, 1)$ or $(i, j) = (1, 0)$). An example for $(i, j) = (1, 1)$ is shown in Figure 3. Intuitively, these four functions indicate where in a (infinitely dimensional) binary vector with one element for each $x \in [0, T]$ we have a situation that corresponds to an element of the 2×2 contingency table shown in Table 1. Note that for all $x \in [0, T]$ exactly one of the four functions $f_{ab}^{(i,j)}(x)$, $(i, j) \in \{0, 1\}^2$, has the value 1, while the other three are 0. Therefore we can easily derive the elements of a generalized contingency table as

$$n_{ij} = \frac{1}{r} \int_0^T f_{ab}^{(i,j)}(x) dx$$

for all $(i, j) \in \{0, 1\}^2$. Note that it is $n_{00} + n_{01} + n_{10} + n_{11} = n_{**} = \frac{T}{r}$. This shows how one can solve the counting problem, which consists in the fact that the entries of a standard contingency table are counters of vector elements, but we cannot count elements of a binary vector with super-countably many elements. Note that the solution chosen here is in line with a time-binned approach, where the width of the time bins, which corresponds to the width of the influence maps in our approach, determines the total count n_{**} in exactly the same way.

3.2 Evaluating Distance Measures

Being able to efficiently calculate distances between continuous spike trains, we can evaluate the distance measures. As we want to discriminate between neurons belonging to an assembly and those that exhibit only random noise, we take a first look at plots of the distance matrices. In these plots large distances correspond to whiter cells, while darker cells indicate a high similarity between two data points. Figure 4 shows example plots for different distance measures.

As can be seen, only Jaccard, Dot, Correlation and Yule allow for a (visually) clear discrimination of the assembly from the rest of the neurons. In addition, Yule's distance suffers from many small distances between neurons exhibiting random noise, which can impede the detection. The remaining two distance measures show no significant differences between neurons belonging to the injected assembly and other neurons. Obviously the possible values for the distance measure is in both cases not fully exhausted such that a visual inspection may not be sufficient to reject these two distance measure at this stage. But even normalizing the values in the distance matrix so that they lie in the range $[0, 1]$ does not lead to better result. Hence we will only consider the four distance measures Jaccard, Dot, Correlation and Yule in the following.

3.3 Sorting by Non-Linear Mapping and Assembly Detection

To find a suitable ordering of the neurons for further testing it is necessary to choose a sorting criterion. In [4] we already proposed to use Sammon's non-linear mapping [24] to reduce the dimensionality of the data set to only one dimension. The resulting scalar can then be used to sort the neurons. In a linear traversal of the sorted list the final tests are performed and the assemblies are detected.

Originally, in the discrete case, we performed χ^2 tests for independence to distinguish between correlated and non-correlated neurons. However, this test is only properly applicable if we have actually countable events like coincidences, which in the discrete case is the number of time bins. Only in this case the χ^2 value is properly interpretable. In our current setup, however, even though we can achieve an analogous sum of the contingency table entries (see Section 3.1), applying the test in the same way appears to be a somewhat dubious procedure.

Similarly, choosing a time resolution (other than the width r of the influence maps) to give a unit to the entries in the contingency table does not appear to be appropriate, since the result of the χ^2 test (whether it is significant or not) depends directly on this choice. Suppose, for instance, that we have a recording of one second length and choose one second as the time unit. As a consequence, no entry in the contingency table can be higher than 1 (if measured in this unit). Thus, for a χ^2 distribution with one degree of freedom, we would have to lower the level of significance to 0.68 in order to be able to reject the null hypothesis of independence. On the other hand, using a millisecond resolution for the same data may enable us to reject the null hypothesis, because of the wider range of possible χ^2 values. Hence we refrain from such an approach.

However, as the calculation of the χ^2 value and the corresponding p-value is only used to accept or reject whether a neurons is correlated with its neighbor (in the sorted list of neurons), we can also argue that the distance measure itself provides all relevant information. Neurons belonging to the same assembly have lower distances between each other than to the rest of the neurons. This leads to the following simple idea: we sort the neurons w.r.t. the value that is assigned to them by Sammon's non-linear mapping (or some other multi-dimensional scaling method) to one dimension. Then we find the index k of the neuron such that

$$k = \operatorname*{argmax}_{1 \leq i \leq n} |x_i - x_{i+1}|,$$

which indicates the largest gap between two consecutive neurons (all indices refer to the sorted list of neurons). We set the decision boundary in this gap and thus obtain the two sets of neuron indices

$$A = \{i \mid i \geq 1 \wedge i \leq k\} \quad \text{and} \quad B = \{i \mid i > k \wedge i \leq n\}.$$

We then report the set as assembly that has the least average inner-set distance:

$$d_A = \frac{1}{k-1}|x_k - x_1| \quad \text{and} \quad d_B = \frac{1}{n-k-1}|x_n - x_{k+1}|$$

4 Evaluation

To evaluate our method we artificially generate a set of $n = 50$ spike trains (independent stationary Poisson processes with a firing rate of $\lambda = 20$Hz, which is similar to the frequency with which cortical neurons fire) of $T = 10$s length, into which synchronous spiking activity of 10 neurons was injected (coincidence firing rate $\lambda_c = 5$Hz, background rate appropriately adapted to obtain equal overall firing rates of 20Hz). Coincident spikes are copied from a *mother process* into the spike trains with different copy probabilities p. In addition, the spike times are uniformly dithered by ± 5ms (independently for each neuron) to simulate temporal imprecision as it may be present in a real life data set. The width of the influence maps was chosen to be 15ms to be able to capture a jittered spike at least partially if the dithering moved the coincidences far away from each other. The possible outcomes that may occur in our test setup are:

1. The assembly has been correctly detected. No neuron is missing, no additional neuron has been found (depicted as *perfect*). These are true positives.
2. The whole assembly has been found, but at least one additional neuron was found (depicted as *too many*).
3. Only a subset of the assembly's neurons has been found. At least one neuron was missing (depicted as *too few*).
4. Only a subset of the assembly's neurons has been found. At least one neuron was missing. In addition at least one neuron has been found to belong to the assembly that is actually independent (depicted as *too few/too many*).

5. Every neuron that has been reported belongs to the group of independent neurons and not a single neuron of the true assembly has been reported (depicted as *wrong*). These are false positives.

For each test 100 trials were run and the number of each type of error was recorded. The results of these tests are shown in Figure 5.

We obtained no results for the dot product distance measure because during the first run the test was canceled due to the surprisingly bad quality of the results (see Figure 5, bottom) even if the copy probability is one. Further investigation showed that the quality is only bad if the dimensionality reduction is performed to produce only one dimension. Two dimensional plots reveal, that an algorithm which calculates decision boundaries at arbitrary angles would very well be able to distinguish between assembly and noise neurons in case of a dot product distance measure (see Figure 7).

To compare the results obtained with the binned approach we conducted exactly the same experiments for half of the copy probability settings as well. The distance measures used are exactly the same, only that the spike trains are treated as binary vectors. The numbers $n_{ij}, i, j \in \{0, 1\}$ are calculated in the usual way. What we can see is that the binning method is slightly more prone to not finding the whole assembly or considering too many neurons while never reporting any assembly that has to be labelled as *wrong*. The later may be attributed to the fact that spikes that fall into the same time bin are counted as if they were perfectly aligned, thus leading to a higher density of the assembly as such, while the continuous calculation uses a more differentiated view on such coincidences.

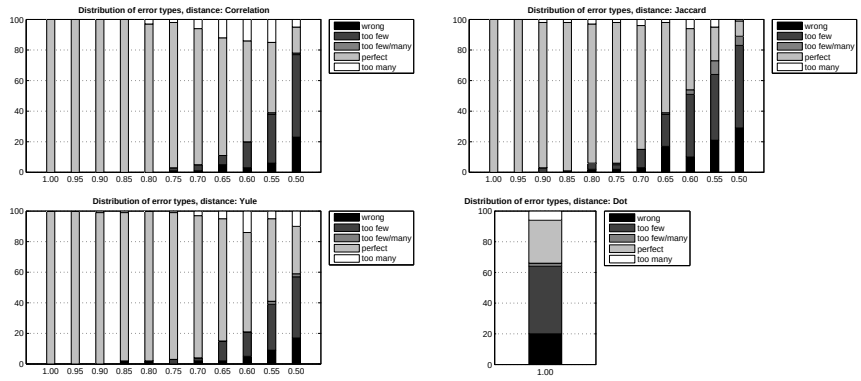


Fig. 5. Assembly detection quality under varying conditions for the selective participation of neurons in synchronous spiking activity. The horizontal axis shows the probability with which a spike is copied from the coincidence mother process. For the dot product as the distance measure, experiments for lower copy probabilities were not conducted, because of the disappointing results for a copy probability of 1.

In contrast the method proposed by used is able to detect the assembly as a whole even when the copy probability is decreasing. Perfect findings are much more common but only at the cost of reports of totally *wrong* assemblies. Having a closer look at those false positives revealed that these were mostly single noise neurons that were so different from all other neurons that the multidimensional scaling placed them farther apart from the rest of all neurons than the gap between assembly and noise neurons was.

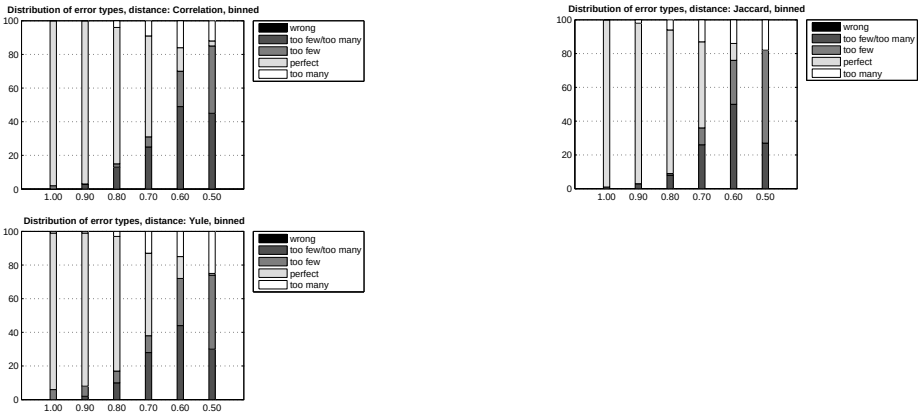


Fig. 6. Assembly detection quality under varying conditions for the selective participation of neurons in synchronous spiking activity. The horizontal axis shows the probability with which a spike is copied from the coincidence mother process. The spike trains have undergone a binning before being analysed with the same procedure to compare results.

5 Conclusion and Future Work

In this paper we presented a simple and efficient method for detecting assemblies of neurons in continuous spike train data, which avoids the otherwise almost exclusively applied time binning. As a consequence, the loss and distortion of synchrony information resulting from binning the data can be avoided. The additional challenges of *temporal imprecision* (imperfect spike alignment) and *selective participation* (copy probabilities less than 1) can be handled with this method while still recovering most of the assemblies at least partially. Although the performance of our algorithm is fairly good, there are still many open problems that need to be addressed. In the first place, additional tests on different types of data need to be performed. Compared to algorithms that work on discretized data the rate of false positives is significantly higher. This is partly due to the multi-dimensional scaling that increases the likelihood of neighboring data points having a small distance to each other. On the other hand a lot of information about the structure of the data is lost, if multi-dimensional scaling to only one dimension is performed. First tests have shown that this one dimension contains

about three times as much information as the next best dimensions, but keeping two or more dimensions and finding assemblies with common clustering methods or support vector machines may be worth looking into.

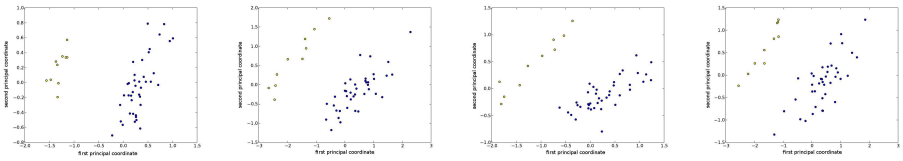


Fig. 7. Two dimensional plots of 50 neurons, assembly of size 10 (yellow) and dot product as distance measure. Classification has been performed with only the x-axis available. The upper left image shows the case where an assembly was completely found. The upper right image shows the case, where only a single (noise) neuron was reported. The lower left image shows the case, where some neurons were missed. The lower right image shows the case, where too many neurons were found. On the right hand side the distribution of error types for a copy probability of 1.0 is shown.

Acknowledgments

This work was partly supported by Helmholtz Alliance on Systems Biology, European Union (FP7-ICT-2009-6, BrainScales).

References

1. Allen, J.: Maintaining knowledge about temporal intervals. *Communications of the ACM* 26(11):832–843. ACM Press, New York, NY, USA (1983)
2. Berger, D., Borgelt, C., Diesmann, M., Gerstein, G., Grün, S.: An accretion based data mining algorithm for identification of sets of correlated neurons. *18th Annual Computational Neuroscience Meeting: CNS*2009* 10(Suppl 1), 18–23 (2009)
3. Berger, D., Borgelt, C., Louis, S., Morrison, A., Grün, S.: Efficient identification of assembly neurons within massively parallel spike trains. *Computational Intelligence and Neuroscience*, Article ID 439648, DOI:10.1155/2010/439648. Hindawi Publishing Corporation, New York, NY, USA (2010)
4. Braune, C., Borgelt, C., Grün, S.: Finding ensembles of neurons in spike trains by non-linear mapping and statistical testing. *Advances in Intelligent Data Analysis X* LNCS 7014:55–66. Springer, Berlin / Heidelberg, Germany (2011)
5. Buzsáki, G.: Large-scale Recording of Neuronal Ensembles. *Nature Neuroscience* 7:446–461. Nature Publishing Group/Macmillian, New York, NY, USA (2004)
6. Cox, T.F.; Cox; M.A.A.: *Multidimensional Scaling (2nd ed.)*. Chapman & Hall, London, United Kingdom (2000)
7. Dice, L.R.: Measures of the amount of ecologic association between species. *Ecology* 26:297–302. Ecological Society of America, Ithaca, NY, USA (1945)
8. Casey O. Diekman and P.S. Sastry and K.P. Unnikrishnan: *Statistical significance of sequential firing patterns in multi-neuronal spike trains.* Journal of Neuroscience Methods, 182(2):279–284 (2009)

9. Edwards, A.: *An introduction to linear regression and correlation.* WH Freeman, New York, NY, USA (1984)
10. Feldt, S. and Waddell, J. and Hetrick, V. L. and Berke, J. D. and Zochowski, M.: *Functional clustering algorithm for the analysis of dynamic network data*, Physical Review E, 79(5):056104-1–056104-9 (2009)
11. Gerstein, G., Perkel, D., Subramanian, K.: Identification of functionally related neural assemblies. *Brain Research* 140(1):43–62. Elsevier, Amsterdam, Netherlands (1978)
12. Grün, S., Diesmann, M., Grammont, F., Riehle, A., and Aertsen, A.: Detecting unitary events without discretization of time. *Journal of Neuroscience Methods* 94:67–79. Elsevier, Amsterdam, Netherlands (1999)
13. Grün, S., Abeles, M., Diesmann, M.: Impact of higher-order correlations on coincidence distributions of massively parallel data. *Dynamic Brain – from Neural Spikes to Behaviors* LNCS 5286: 96–114. Springer, Berlin / Heidelberg, Germany (2008)
14. Grün, S., Diesmann, M., Aertsen, A.: Unitary events in multiple single-neuron activity. I. Detection and significance. *Neural Computation* 14(1):43–80. MIT Press, Cambridge, MA, USA (2002)
15. Hamming, R.: Error detecting and error correcting codes. *Bell Systems Tech. Journal* 29:147–160. Bell Laboratories, Murray Hill, NJ, USA (1950)
16. Hebb, D.: *The organization of behavior.* J. Wiley & Sons, New York, USA (1949)
17. Jaccard, P.: Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin de la Société Vaudoise des Sciences Naturelles* 37:547–579. France (1901)
18. Lewicki, M.: A review of methods for spike sorting: The detection and classification of neural action potentials. *Network* 9(4):R53–R78. Informa, St. Helier, Jersey, France (1998)
19. Mohammad Shahed Masud and Roman Borisyuk: *Statistical technique for analysing functional connectivity of multiple spike trains.* Journal of Neuroscience Methods, 196(1):201–219 (2011)
20. Pazienti, A., Maldonado, P., Diesmann, M., Grün, S.: Effectiveness of systematic spike dithering depends on the precision of cortical synchronization. *Brain Research* 1225:39–46. Elsevier, Amsterdam, Netherlands (2008)
21. Perkel, D.H. and Gerstein, G.L. and Moore, G.P.: Neuronal spike trains and stochastic point processes: II. Simultaneous spike trains. *Biophysical Journal* 7(4):419–440. Elsevier, Amsterdam, Netherlands (1967)
22. Picado-Muiño, D.; Castro-León, I.; Borgelt, C.: Fuzzy frequent pattern mining to identify frequent neuronal patterns in parallel spike trains. Submitted to IDA 2012
23. Rogers, D., Tanimoto, T.: A computer program for classifying plants. *Science* 132:1115–1118 American Association for the Advancement of Science, Washington, DC, USA (1960)
24. Sammon, J.: A nonlinear mapping for data structure analysis. *IEEE Trans. Comput.* 18(5):401–409. IEEE Press, Piscataway, NJ, USA (1969)
25. Shmiel, T. and Drori, R. and Shmiel, O. and Ben-Shaul, Y. and Nadasdy, Z. and Shemesh, M. and Teicher, M. and Abeles, M.: *Temporally precise cortical firing patterns are associated with distinct action segments* Journal of neurophysiology, 96(5):2645–2652 (2006)
26. Tanimoto, T.: IBM Internal Report (November 17, 1957)
27. Yule, G.: On the association of attributes in statistics. *Philosophical Transactions of the Royal Society of London*, Series A, 194:257–319 (1900)