Spike train distances and neuronal coding

Satuvuori, E.

2018

document version
Publisher's PDF, also known as Version of record

Spike train distances and neuronal coding

PhD thesis

Eero Antero Satuvuori

October 5, 2018

To my family

CONTENTS

1

GENERAL INTRODUCTION

1.1 NEURONS, SPIKES AND SPIKE TRAINS

No! Try not. Do! Or do not. There is no try.
Yoda, -Star Wars Episode V

This seems an odd thing to say. One often fails when doing something for the very first time. Would that be counted as trying to do it? How I understand it is that one should always approach every new thing with self confidence or it is doomed to fail. While the quote has to be interpreted in some meaningful way for us to make sense of it, the basic unit of our brain, the neuron, actually functions with absolutes. A brain consists of potentially billions (1) of neurons and equally many or more other cells. Whenever enough input is presented to a neuron they fire an action potential and if not they will remain silent (2). They do. Or do not.

A single action potential lasts in the time range of $2 - 5ms$ depending on how it is defined. For this thesis the exact way spikes are produced is not very relevant, but for understanding how neurons work I will give a very short introduction to spike generation in neurons. Action potentials are based on difference in charged ion concentrations over the cell membrane of the neuron. The membrane has non-Ohmic resistance due to non-linear permeabilities for the different ions. A neuron fires an action potential when an area of the neuronal membrane reaches a *threshold* voltage. When this happens, the neuron's cell membrane is polarized very fast. The increase in voltage also causes another mechanism to drive the membrane potential down again but this happens slightly slower, however, and the membrane voltage has already got up by the time this happens. For a short moment both are in effect and during this time the neuron cannot fire another action potential. Once the *refractory period* is over and everything returns to normal, the neuron is ready to fire again given enough stimulation for the membrane to reach the threshold again.(2)

The action potential is initiated in the neuron at a place called axon hillock. Once the area fires an action potential, it also depolarizes the membrane nearby. This way the depolarization wave moves along the cell membrane. Most neurons have a long projection starting from the axon hillock called axon. You can think the cell membrane of an *axon* a as a water hose. Once the depolarization threshold is reached locally at one point, the depolarization effect progresses to polarize the membrane close by and the wave rapidly goes around the membrane and meets on the other side of the "hose". Once that happens, the depolarization wave starts moving along the hose until the end. In neurons the axons often diverge into multiple directions and have many endings. These endings are formed near the membrane of another neurons and are called *synapses*. At the synapses there is a different chemical machinery making use of the polarized membrane potential, which releases neurotransmitter molecules into the synapse. The released transmitters have an effect on the polarization of the next neuron either by exiting or inhibiting the increase of membrane potential at the axon hillock of the next neuron. Adding more neurons and connecting them with synapses we get what is called *neuronal network*.

There needs to be enough stimulation from the downstream neurons in order to invoke an action potential. Since neurons integrate effects of synapses from up to hundreds of thousands of other neurons, the timing of the action potentials in relation to each other is essential. While this often requires coordination of excitatory signals from multiple downstream neurons, the neuron itself does not discriminate which ones of the neurons connected to it actually caused the stimulation. For data-analysis purposes it is often enough to consider the times of the action potentials at the peak of the membrane potential due to the all-or-none paradigm of the neuron. The time discrete representation of an action potential is called a *spike* and series of spikes fired by the same neuron over time *spike trains*.

A lot of neuroscience research is concentrated on finding response patterns in spike trains of neurons. For a given stimulus a neuron that codes for something in the stimulus is expected to elicit similar response each time that differs for each stimulus coded. As if you put your finger on a hot cooking plate it will hurt every time. The same response is sent from your fingertip to your brain and each time the brain interprets the stimulus as pain. This is called *reliability* of response. However, defining when a response pattern is reliable, or in other words, similar to the response of another representation of the same stimulus, is not a simple task. The main issue is that similarity is not a well defined concept (See Fig. 1.1). It comes down to what are the properties considered in the

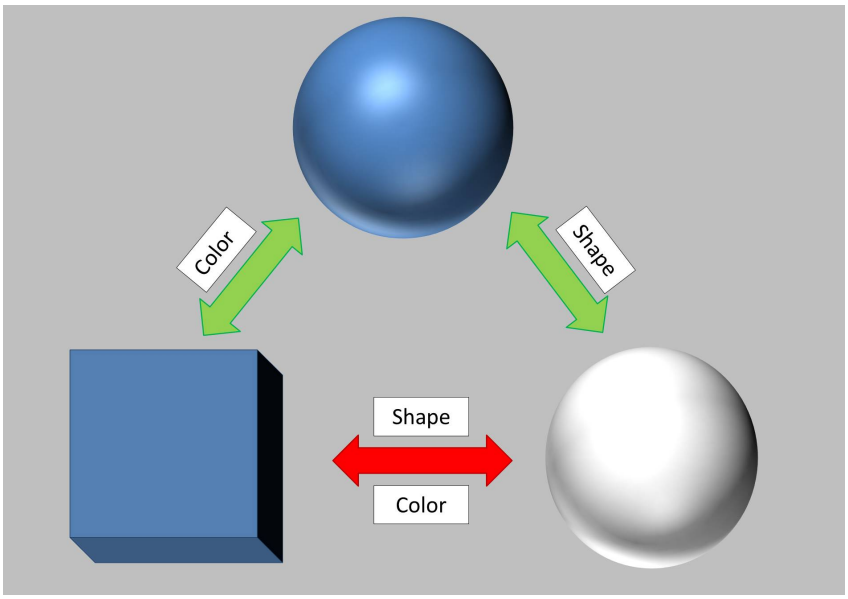

Figure 1.1: Visual example on importance of similarity description. A blue ball is similar to a white ball based on shape, but also to a blue cube based on color, while a white ball is not similar to the blue cube by either description.

similarity description and our selection will have an effect on the results obtained. The properties considered for assessing similarity of responses of neurons must come from the way the neurons transmit information or *code*.

1.2 NEURONAL CODING

There are two main approaches to neuronal coding. Since the neuron does not care where the stimulation actually came from, it is reasonable to assume that sufficiently high firing rate in the downstream neurons will eventually build up and elicit spikes in the next one. This assumption is called *rate coding*. Another approach considers a more compact coding, where each neuron time its spikes in order to achieve a spike upstream at the desired moment. This is called *time coding*. Both assumptions are perfectly reasonable, yet exclusive from the neuron's point of view.

The key to understanding neuronal coding is in the similarity of responses. The similarity can be self-similarity in response to the same stimulus at different times, or it can be similarity between responses of two different neurons. There are many ways for assessing if spike pat-

terns are similar or nots. One common way is to assume rate coding and count spikes in a predefined window or bin (one obtains rate by normalizing the spike count to the window length). This approach has two major problems. The first is that the definitions for the start and end times of the window are more or less arbitrary. The second problem is selecting appropriate time scale of and it relates to the first one, since rate is an average quantity over time. Considering rate as spikes over time does not take into account where in the window the spikes occurred and for different start and end times one might get very different results. For example, two spike trains would be deemed similar if they had 5 spikes each even if one has all spikes at the beginning and one at the end, but shortening the window so that the last spikes of the second spike train are outside of the window makes them dissimilar. Some methods use multiple adjacent or overlapping windows and counting spikes in each window. However, defining proper window length and starting position for the windows remains problematic. Additionally, this is no longer purely a rate based assessment, since it assumes that timing of spikes (bins) matters. Moreover, the length of the windows used defines the temporal resolution of the method.

An alternative to binning is to use binless spike train distances, which deal with just the time stamps of the spikes in order to determine similarity. ISI-distance (3, 4), SPIKE-distance (5, 6) and SPIKE-synchronization (7) use local rate estimates derived from the distance between two consecutive spikes called the *interspike interval* (ISI). This makes them independent from the rigid notion of bin size and allows for time scale independent assessments. This means that stretching and compressing a spike train will have no effect on the end result, while if one did that with a binning method, the bin size would have to be adjusted accordingly.

1.3 STRUCTURE OF THE THESIS

The main Chapters of this thesis are based on the following four publications:

- Chapter 2: *Measures of spike train synchrony for data with multiple time scales* (8), Journal of Neuroscience Methods, Volume 287, 1 August 2017, Pages 25-38

- Chapter 3 adapted from: *Leaders and followers: quantifying consistency in spatio-temporal propagation patterns* (9), New Journal of Physics, Volume 19, April 2017, Pages 043028

- Chapter 4: *Which spike train distance is most suitable for distinguishing rate and temporal coding?* (10), Journal of Neuroscience Methods, Volume 299, 1 April 2018, Pages 22-33
- Chapter 5: *Using spike train distances to identify the most discriminative neuronal subpopulation* Journal of Neuroscience Methods, September 2018, In press (11)

The Chapters above have been modified from the original open access publications but are not presented in chronological order. Especially, Chapter 3 has been shortened considerably from the original, since only part of the publication is about neuroscience.

1.3.1 *Chapter 2*

While time scale independence of ISI-distance and SPIKE-distance, and SPIKE-synchronization provide an effective way of assessing time local information in spike trains, in some cases it may be problematic when there are multiple time scales in the data. While it makes sense that spike trains that look similar at different time scales are assessed in the same way, there is a limit to when this should be done. For example if two neurons fire in perfectly opposite phases for a short period of time while being silent otherwise, it appears that the neurons are responding similarly when looking at the spike trains. However, since the methods do all assessment of similarity in relation to local rate estimated by ISIs, the moment when they fire is considered to be maximally different even when in the global scale this is not the case.

In Chapter 2, I explain a generalization to ISI-distance, SPIKE-distance, and SPIKE-synchronization to account for multiple time scales in the spike trains. The limitation of these generalizations is that they need a time scale parameter indicating what is a "too small" ISI, or reversely a too high rate, to be assessed in the local context and when should a global context be used instead. I also introduce a mathematical way to extract an estimate for an appropriate parameter from the ISI distribution of data. Only when using the estimate, the time scale free behaviour is maintained and if one uses a fixed parameter it is lost.

1.3.2 *Chapter 3*

A complementary approach to estimating similarity between spike trains is to take a set of simultaneous recordings from multiple sources and try to find if some units tend to fire before the others in a consistent way. In

Chapter 3, I explain how the spike trains can be sorted from leader to follower using methods extended from SPIKE-synchronization. They are called SPIKE-Order and Spike Train Order. Both provide complementary information that can be used to sort spike trains from leader to follower i.e. the one that tends to fire first to the one that tends to fire last and to test the statistical significance of the order found. This assumes reliable time coding and does not work for rate coding.

1.3.3 *Chapter 4*

Two of the most established spike train distances are Victor-Purpura distance (12) and the van Rossum distance (13). These measures contain parameters, which indicate time scale, and are loosely related to the window size used in binning methods. For one extreme of the parameter for each spike the methods search for an exact time matching pair in the other spike train and at the other extreme only the number of spikes in the entire window counts. It has been widely accepted that one end of the scale measures time coding and the other rate coding, and in between there is a tradeoff between the two.

The major difference between Victor-Purpura distance and van Rossum distance compared to the ISI-distance and the SPIKE-distance is not the parameter, but the way they approach similarity. The first two do comparisons solely between spike times of closest spikes in the other spike train, while the ISI-distance and the SPIKE-distance span a profile over time based on spike timing relative to the spike times in the other spike train over recording window. Thus, I refer to the first group as spike-resolved and the second one as time-resolved. There is a considerable difference in how these two types of measures consider some fundamental concepts of similarity between spike trains. In Chapter 4, I examine this, while also assessing the pitfalls of these measures.

1.3.4 *Chapter 5*

The Victor-Purpura distance and the van Rossum distance have been extended to deal with the important problem of population coding. Both extensions contain an additional parameter that indicates relative importance of a spike being fired by the same neuron in a population. Both extremes of the second parameter are easy enough to understand. Either all the spike patterns for repeated representations of a stimulus are compared against the responses of the same neuron each time, called labelled line (LL), or it does not matter at all which neuron fired the

spike and thus all spikes of the population are pooled for comparison, called summed population (SP). However, all the parameter values in between are difficult to interpret. In Chapter 5 I introduce a complementary approach for assessing if there is similarity in firing patterns of the population and for finding the subpopulation that is most effective at discriminating between different stimuli. I use different approaches to identify the subpopulations following either the LL or the SP coding assumptions.

1.4 cSPIKE SOFTWARE

During my studies, I wrote a freely available spike train analysis program called cSPIKE[1]. This software is used for doing all the spike train analysis concerning ISI-distance, SPIKE-distance and SPIKE-synchronization and their variants. All in all it includes over 40 functions for calculating different kinds of distances between spike trains.

The distance measures have been implemented before in a Matlab GUI package SPIKY (7) as well as in a Python package PySpike (14). Unlike the previous user friendly graphical user interface SPIKY, cSPIKE uses command line interface and is intended to be used by those, who are more familiar with Matlab coding. cSPIKE differs from the PySpike mainly by the implementation platform. Both use C++ for speed and run from command line interface. Additionally, the reduction of output data required for running cSPIKE when compared to SPIKY GUI allows the computation to be performed considerably faster.

1 http://www.fi.isc.cnr.it/users/thomas.kreuz/Source-Code/cSPIKE.html

2

EXTENDING MEASURES OF SPIKE TRAIN SYNCHRONY

Background: Measures of spike train synchrony are widely used in both experimental and computational neuroscience. Time-scale independent and parameter-free measures, such as the ISI-distance, the SPIKE-distance and SPIKE-synchronization, are preferable to time scale parametric measures, since by adapting to the local firing rate they take into account all the time scales of a given dataset.
New Method: In data containing multiple time scales (e.g. regular spiking and bursts) one is typically less interested in the smallest time scales and a more adaptive approach is needed. Here we propose the A-ISI-distance, the A-SPIKE-distance and A-SPIKE-synchronization, which generalize the original measures by considering the local relative to the global time scales. For the A-SPIKE-distance we also introduce a rate-independent extension called the RIA-SPIKE-distance, which focuses specifically on spike timing.
Results: The adaptive generalizations A-ISI-distance and A-SPIKE-distance allow to disregard spike time differences that are not relevant on a more global scale. A-SPIKE-synchronization does not any longer demand an unreasonably high accuracy for spike doublets and coinciding bursts. Finally, the RIA-SPIKE-distance proves to be independent of rate ratios between spike trains.
Comparison with Existing Methods: We find that compared to the original versions the A-ISI-distance and the A-SPIKE-distance yield improvements for spike trains containing different time scales without exhibiting any unwanted side effects in other examples. A-SPIKE-synchronization matches spikes more efficiently than SPIKE-Synchronization.
Conclusions: With these proposals we have completed the picture, since we now provide adaptive generalized measures that are sensitive to firing rate only (A-ISI-distance), to timing only (ARI-SPIKE-distance), and to both at the same time (A-SPIKE-distance).

Adapted from: *Measures of spike train synchrony for data with multiple time scales* (8)

2.1 INTRODUCTION

In neuroscience the neuronal action potential and its complex molecular behavior (2) is often reduced to time-discrete events called *spikes*. Due to the all-or-nothing paradigm of neurons together with the long silent periods, the time stamps of the spike events are considered to be an accurate enough description of the neuronal membrane potential (15). These sequences of consecutive spikes are called *spike trains*. While spike trains do not directly provide information about the connections between neurons, some form of link between two neurons is often inferred by the similarity of their spike trains. There are also other analysis methods apart from spike train distances for inferring the linkage (see e.g. (16)) A spike train distance does not take into account the specific type of linkage, but simply quantifies how (dis)similar the two spike trains are. This makes spike train distances universal and as such they can be applied to all systems that can be reduced to point processes. In addition to the obvious neuroscience applications, they have already been used to study inter-personal coordination (17) and social cognition (18) among many other fields.

Over the years many different measures have been developed in order to quantify similarities between two or more spike trains (see (19), (20) and (21) for an overview). The two most known time scale parametric measures, the Victor-Purpura (12) and the van Rossum distance (13), describe spike train (dis)similarity based on user-defined time scales to which the measures are mainly sensitive. One drawback of these measures is the fixed time scale, since it sets a boundary between rate and time coding for the whole recording. However, for real data which typically contain many time scales (such as regular spiking and bursts), this is difficult to detect with a measure that is mainly sensitive to only one of them (22).

The problem of having to choose one time scale has been eliminated in the three time-resolved and time scale independent measures ISI-distance (3, 4), SPIKE-distance (5, 6) and SPIKE-synchronization (7). The ISI-distance (3) is a measure of instantaneous rate dissimilarity. It uses the interspike intervals (ISIs) to estimate the local firing rate of spike trains and quantifies their differences in a time-resolved manner. The SPIKE-distance (5) compares the spike time accuracy between spike trains and uses instantaneous firing rates to adapt to the local time scale. Finally, SPIKE-synchronization (7) is a discrete time-resolved measure of similarity based on ISI-derived coincidence windows that are used to determine if two spikes from different spike trains are coincident or not. These measures have already been successfully applied in many different contexts;

for example they have been used to detect determinism in point processes (23), to find correlations between spike trains and behaviour in an inverse neurocontroller (24) and to evaluate a bio-inspired locomotion system in robotics (25).

Since they always adapt to the local firing rate, all three of these measures are time scale free. While they correctly identify the relative firing rate differences, they have no concept of actual time scales and treat all time scales as equally important. This has the consequence that for very small time scales even minor deviations from perfect synchrony lead to very high values of dissimilarity. However, for real data the smallest time scales are often not very relevant and any dissimilarities there can mostly be disregarded. Thus in this case the measures' focus on the local time scales results in a (spurious) amplification of dissimilarities which compared to the global time scales are rather negligible.

Here we address this problem by proposing generalizations to the three measures called adaptive ISI-distance (A-ISI-distance), adaptive SPIKE-distance (A-SPIKE-distance) and adaptive SPIKE-synchronization (A-SPIKE-synchronization). These generalized definitions add a notion of the relative importance of local differences compared to the global time scales. In particular, they start to gradually ignore differences between spike trains for ISIs that are smaller than a minimum relevant time scale (MRTS). The MRTS is implemented by an additional variable $\mathcal{T}$ which can either be defined as a parameter or estimated directly from the data.

In some neuroscience applications only the similarity of spike timing is important and rate differentiation is not a desired property. While the A-ISI-distance is sensitive to firing rate alone and the A-SPIKE-distance responds to differences in both rate and timing, there is currently no measure that focuses only on spike timing. Therefore, in a second step we extend the A-SPIKE-distance into the rate-independent adaptive SPIKE-distance (RIA-SPIKE-distance) which still identifies spike time differences but ignores any rate deviations between the spike trains.

The remainder of this paper is organized as follows. In Section 2.2 we describe the generalized definitions of the three measures, the A-ISI-distance (Section 2.2.1), the A-SPIKE-distance (Section 2.2.2), and A-SPIKE-synchronization (Section 2.2.3). In Section 2.2.4 we introduce a way to estimate the threshold value directly from the data. We then investigate using both simulated and real data how both the original measures and the adaptive generalizations deal with multiple time scales (Section 2.2.5). In Section 2.3 we add a rate-independent extension to A-SPIKE-distance (Section 2.3.1) and afterwards study the effects of the extension (Section 2.3.2). The implications of the extensions are discussed in Sec-

tion 2.4. Finally, in the 2.5.1 we cover some non-trivial subtleties of the definitions for all three measures. First we provide the definitions for the periods before the first and after the last spike in a spike train (where the interspike interval is not defined), and then we deal with the two special cases of empty spike trains and spike trains with only one spike. The two experimental datasets used in Section 2.2.5 are described in 2.5.2.

2.2 ADAPTIVE GENERALIZATIONS

In this Section we introduce the adaptive generalizations of the established measures ISI-distance (3), SPIKE-distance (5) and SPIKE-synchronization (7), which we will call A-ISI-distance, A-SPIKE-distance, and A-SPIKE-synchronization. All three generalizations are built on a minimum relevant time scale (MRTS) which is implemented via the threshold parameter $\mathcal{T}$. This threshold is used to determine if a difference between the spike trains should be assessed in a local context or in relation to the global time scales. This threshold is used for all three measures, but the way it is applied varies. The generalized measures fall back on the original definitions when $\mathcal{T} = 0$. In the following this is what we refer to whenever we talk of the original measures. In this case even the smallest time scales matter and all differences are assessed in relation to the local context only.

Note that the upcoming definitions only apply to the interval between the first and the last spike. In 2.5.1.1 and 2.5.1.2 they will be completed to range from the start of the recording t_s to the end of the recording t_e . Equally, some of the following equations are ill-defined when there are less than two spikes in a spike train. These special cases will be handled in 2.5.1.3 and 2.5.1.4.

Throughout the paper we denote the number of spike trains by N, indices of spike trains by n and m, spike indices by i and j and the number of spikes in spike train n by M_n. The spike times of spike train n are denoted by $\{t_i^{(n)}\}$ with $i = 1, \ldots, M_n$.

2.2.1 *Adaptive ISI-distance*

The A-ISI-distance measures the instantaneous rate difference between spike trains (see Fig. 2.1A). It relies on a time-resolved profile, meaning that a dissimilarity value is defined for each time instant. To obtain the profile, we assign to each time instant t the time of the previous spike

$$t_{\mathrm{P}}^{(n)}(t) = \max\left\{t_i^{(n)} | t_i^{(n)} \leq t\right\} \quad \text{for} \quad t_1^{(n)} \leqslant t \leqslant t_{M_n}^{(n)} \tag{2.1}$$

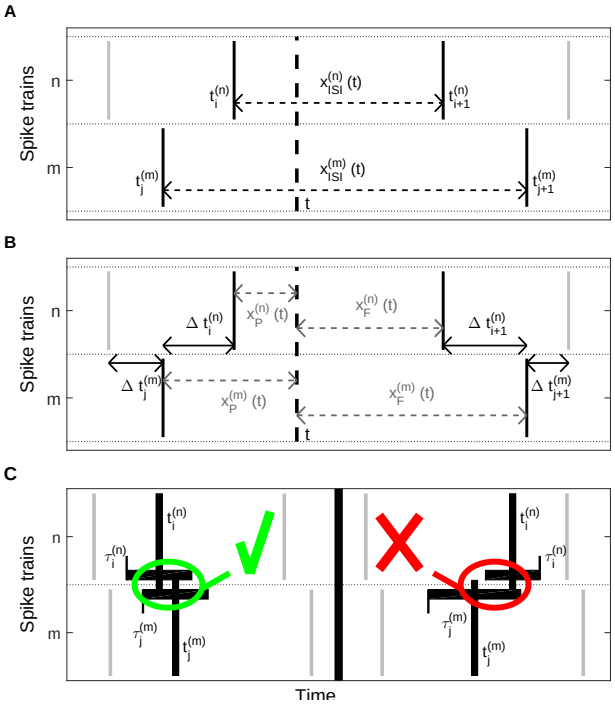

Figure 2.1: Schematic drawing for all three measures. (A) Illustration of the variables that define the ISI-distance. The instantaneous interspike intervals $x^{(n)}_{ISI}(t)$ are used as estimates of the local firing rate. (B) Additional variables employed in the definition of the SPIKE-distance. (C) Coincidence criterion for SPIKE-synchronization. The coincidence window of each spike is derived from its two surrounding interspike intervals. Here we illustrate two different examples. The two spikes on the left side are considered coincident since both lie in each other's coincidence windows. On the right there is no coincidence since the spike from the second spike train is outside of the coincidence window from the spike of the first spike train.

and the time of the following spike

$$t^{(n)}_{\mathrm{F}}(t) = \min\left\{t^{(n)}_i | t^{(n)}_i > t\right\} \quad \text{for} \quad t^{(n)}_1 \leqslant t \leqslant t^{(n)}_{M_n}. \tag{2.2}$$

From this for each spike train n an instantaneous ISI can be calculated as

$$x^{(n)}_{\mathrm{ISI}}(t) = t^{(n)}_{\mathrm{F}}(t) - t^{(n)}_{\mathrm{P}}(t). \tag{2.3}$$

For the A-ISI-distance we define the MRTS such that when the ISIs of both spike trains are smaller than a threshold value $\mathcal{T}$, this value is used instead. The pairwise A-ISI-profile is then defined as

$$I^A_{n,m}(t) = \frac{|x^{(n)}_{\mathrm{ISI}}(t) - x^{(m)}_{\mathrm{ISI}}(t)|}{\max\left\{x^{(n)}_{\mathrm{ISI}}(t), x^{(m)}_{\mathrm{ISI}}(t), \mathcal{T}\right\}}. \tag{2.4}$$

The multivariate A-ISI-profile is obtained by averaging over all pairwise A-ISI-profiles

$$I^A(t) = \frac{2}{N(N-1)} \sum_{n=1}^{N-1} \sum_{m=n+1}^{N} I^A_{n,m}(t). \tag{2.5}$$

This is a non-continuous piecewise constant profile and a final integration over time gives the A-ISI-distance

$$D^A_I = \frac{1}{t_e - t_s} \int_{t_s}^{t_e} I^A(t)dt. \tag{2.6}$$

If the threshold $\mathcal{T}$ is set to zero, the generalized ISI-distance D^A_I falls back to the original ISI-distance D_I.

2.2.2 *Adaptive SPIKE-distance*

The A-SPIKE-distance measures the accuracy of spike times between spike trains relative to local firing rates (see Fig. 2.1B). In order to assess the accuracy of spike events, each spike is assigned the distance to its nearest neighbor in the other spike train

$$\Delta t^{(n)}_i = \min_j \left(|t^{(n)}_i - t^{(m)}_j|\right). \tag{2.7}$$

These distances are then interpolated between spikes using for all times t the time differences to the previous spike

$$x^{(n)}_P(t) = t - t^{(n)}_i \quad \text{for} \quad t^{(n)}_i \leqslant t \leqslant t^{(n)}_{i+1}, \tag{2.8}$$

and to the following spike

$$x^{(n)}_F(t) = t^{(n)}_{i+1} - t \quad \text{for} \quad t^{(n)}_i \leqslant t \leqslant t^{(n)}_{i+1}. \tag{2.9}$$

These two quantities define a time-resolved dissimilarity profile from discrete values the same way as Eqs. 2.1 and 2.2 did for the A-ISI-distance.

The instantaneous weighted spike time difference for a spike train can then be calculated as the interpolation from one difference to the next

$$S_n(t) = \frac{\Delta t_i^{(n)}(t) x_F^{(n)}(t) + \Delta t_{i+1}^{(n)}(t) x_P^{(n)}(t)}{x_{\mathrm{ISI}}^{(n)}(t)} \quad , \quad t_i^{(n)} \leqslant t \leqslant t_{i+1}^{(n)}. \tag{2.10}$$

This function is analogous to the term $x_{\mathrm{ISI}}^{(n)}$ for the ISI-distance, with the only difference that it is piecewise linear instead of piecewise constant. It is also continuous.

The pairwise A-SPIKE-distance profile is obtained by averaging the weighted spike time differences, normalizing to the local firing rate average and, finally, weighting each profile by the instantaneous firing rates of the two spike trains

$$S_{m,n}^A(t) = \frac{S_n x_{\mathrm{ISI}}^m(t) + S_m x_{\mathrm{ISI}}^n(t)}{2 \left\langle x_{\mathrm{ISI}}^{n,m}(t) \right\rangle \max\left\{ \left\langle x_{\mathrm{ISI}}^{n,m}(t) \right\rangle, \mathcal{T} \right\}}. \tag{2.11}$$

We define the MRTS by using a threshold, that replaces the denominator of weighting to spike time differences if the mean is smaller than the threshold $\mathcal{T}$. This profile is analogous to the pairwise A-ISI-profile $I_{n,m}^A(t)$, but again it is piecewise linear, not piecewise constant. Unlike $S_n(t)$ it is not continuous, since typically it exhibits instantaneous jumps at the times of the spikes. The multivariate A-SPIKE-profile is obtained the same way as the multivariate A-ISI-profile, by averaging over all pairwise profiles

$$S^A(t) = \frac{2}{N(N-1)} \sum_{n=1}^{N-1} \sum_{m=n+1}^{N} S_{m,n}^A(t). \tag{2.12}$$

Finally, also the A-SPIKE-distance is calculated as the time integral over the multivariate profile

$$D_S^A = \frac{1}{t_e - t_s} \int_{t_s}^{t_e} S^A(t) dt. \tag{2.13}$$

For $\mathcal{T} = 0$ also the A-SPIKE-distance falls back to the SPIKE-distance.

The effect of applying the threshold can be seen in Fig. 2.2. The first event of five spikes is compressed more and more until it becomes a single burst in the fourth event. The original SPIKE-distance profile $S(T)$ has the same proportions of dissimilarity for all events, since it uses local context only and thus considers all time scales as equal, while the A-SPIKE-distance profile $S^A(t)$ is scaled down when the differences become small compared to the global time scales.

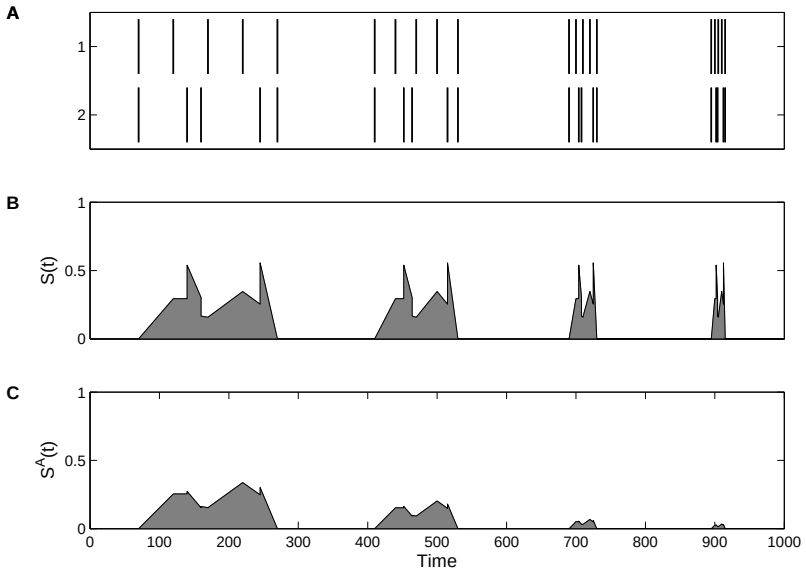

Figure 2.2: An example spike train pair and its SPIKE-distance and A-SPIKE-distance profiles. (A) Two spike trains consisting of four events with five spikes each. The sequence is the same for all four events, only the time scale is getting shorter and shorter. From a global perspective the first event consists of non-synchronous individual spikes, while the last event consists of coincident bursts. The two events in the middle are intermediates. (B) The SPIKE-distance considers only the local context and thus the profile shape is the same for all four events. (C) The A-SPIKE-distance takes into account also the global time scales. Like the SPIKE-distance it judges the first event as very dissimilar, but in contrast to the the SPIKE-distance it scales down the small spike time differences in the bursts and thus considers the coincident burst in the last event as very similar.

2.2.3 *Adaptive SPIKE-synchronization*

A-SPIKE-synchronization quantifies how many of the possible spike coincidences in a dataset are actually occurring (Fig. 2.1C). While the A-ISI-distance and the A-SPIKE-distance are measures of dissimilarity which obtain low values for similar spike trains, A-SPIKE-synchronization measures similarity. If all the spikes are coincident with a spike in all the other spike trains, its value will be one. In contrast, if none of the spikes are coincident, it will be zero.

The original SPIKE-synchronization (7) is parameter- and time scale-free, since it uses the adaptive coincidence detection first proposed for the measure event synchronization (26). The coincidence window, i.e., the time lag below which two spikes from two different spike trains,

$t_i^{(n)}$ and $t_j^{(m)}$, are considered to be coincident, is adapted to the local firing rate. Spikes are coincident only if they both lie in each other's coincidence windows.

For A-SPIKE-synchronization we generalize the definition by introducing a threshold, which decides if the window is determined locally or if the global time scales should be taken into account. As a first step, we define the ISI before the spike as

$$x_{iP}^{(n)} = \lim_{t \to t_i -} x_{\mathrm{ISI}}^{(n)}(t) \tag{2.14}$$

and the ISI after the spike as

$$x_{iF}^{(n)} = \lim_{t \to t_i +} x_{\mathrm{ISI}}^{(n)}(t). \tag{2.15}$$

The coincidence window for spike i of spike train n is defined by determining the minimum coincidence window size for a spike as half the length of the two ISIs adjacent to the spike

$$\tau_i^{(n)} = \tfrac{1}{2} \min \left\{ x_{iP}^{(n)}, x_{iF}^{(n)} \right\}, \tag{2.16}$$

and allowing asymmetric coincidence windows based on MRTS. This is done by replacing $\tau_i^{(n)}$ with the threshold value $\mathcal{T}$, if it is the smaller of the two. Since the threshold value is derived from ISIs and the coincidence window spans both sides of the spike, only half of the threshold spans each side. For the A-ISI- and the A-SPIKE-distance the changes induced by the threshold appear gradually, but for A-SPIKE-synchronization they occur as an abrupt jump from 0 to 1. Therefore, to compensate for the binary nature of A-SPIKE-synchronization, the threshold is divided by two, resulting in an overall factor of $1/4$. The coincidence windows of neighboring spikes are not allowed to overlap, and thus each side is limited to half the ISI even if the threshold is larger. Thus, the coincidence window before the spike is determined as

$$\tau_{iP}^{(n)} = \min \left\{ \max(\tfrac{1}{4}\mathcal{T}, \tau_i^{(n)}), \frac{1}{2} x_{iP}^{(n)} \right\} \tag{2.17}$$

and the coincidence window after the spike as

$$\tau_{iF}^{(n)} = \min \left\{ \max(\tfrac{1}{4}\mathcal{T}, \tau_i^{(n)}), \frac{1}{2} x_{iF}^{(n)} \right\}. \tag{2.18}$$

The combined coincidence window for spikes i and j is then defined as

$$\tau_{ij}^{(n,m)} = \begin{cases} \min \left\{ \tau_{iF}^{(n)}, \tau_{jP}^{(m)} \right\} & \text{if } \; t_i \leqslant t_j \\ \min \left\{ \tau_{iP}^{(n)}, \tau_{jF}^{(m)} \right\} & \text{otherwise} \end{cases}. \tag{2.19}$$

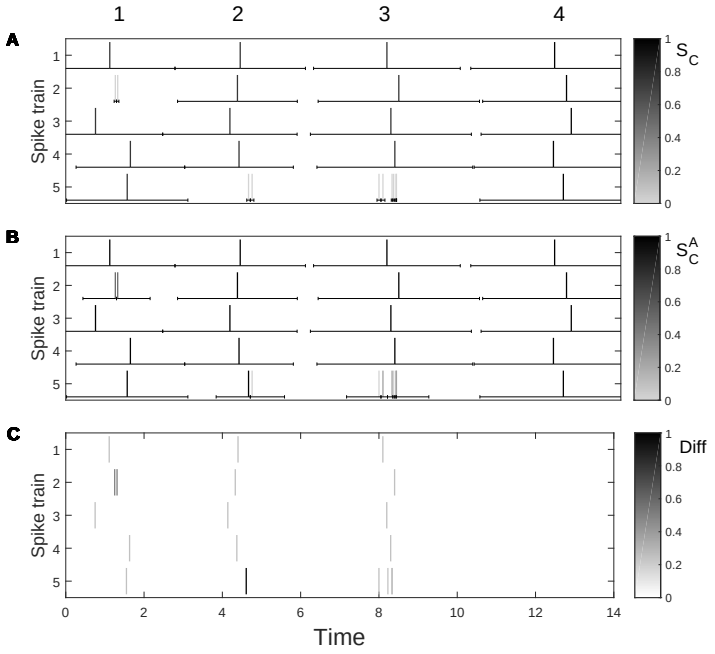

Figure 2.3: SPIKE-synchronization (A), A-SPIKE-synchronization (B) and their difference (C) illustrated using five spike trains with four simple events. For the original measure (A) the small interspike intervals of spike doublets (first and second event) or bursts (third event) result in an unreasonably high demand for spike timing accuracy. With the adaptive generalization (B) for all these cases the likelihood increases that at least one of the spikes is part of a coincidence. On the other hand, if there are no doublets or bursts (last event), nothing changes (best seen in C). Note that the color scales differ, for better visibility we use grey-black in A and B but white-black in C.

The coincidence criterion can be quantified by means of a coincidence indicator

$$C_i^{(n,m)} = \begin{cases} 1 & \text{if } \min_j \left\{ |t_i^{(n)} - t_j^{(m)}| \right\} < \tau_{ij}^{(n,m)} \\ 0 & \text{otherwise} \end{cases} . \tag{2.20}$$

This definition ensures that each spike can only be coincident with at most one spike in the other spike train. The coincidence criterion assigns either a one or a zero to each spike depending on whether it is part of a coincidence or not. For each spike of every spike train, a normalized coincidence counter

$$C_i^{(n)} = \frac{1}{N-1} \sum_{m \neq n} C_i^{(n,m)} \tag{2.21}$$

is obtained by averaging over all $N-1$ bivariate coincidence indicators involving the spike i in spike train n.

This way we have defined a coincidence indicator for each individual spike in the spike trains. In order to obtain one combined similarity profile, we pool the spikes of the spike trains as well as their coincidence indicators by introducing one overall spike index k. This yields one pooled set of coincidence indicators

$$\{C_k\} = \bigcup_n \left\{C_i^{(n)}\right\} \tag{2.22}$$

from which the A-SPIKE-synchronization profile $C^A(t_k)$ can be obtained via $C^A(t_k) = C(k)$. Finally, A-SPIKE-synchronization is defined as the average value of this discrete profile

$$S_C^A = \frac{1}{M}\sum_{k=1}^{M} C^A(t_k), \tag{2.23}$$

where M is the overall number of spikes. In Fig. 2.3 we illustrate how the asymmetric coincidence windows of A-SPIKE-synchronization allow for a better coverage of burst events which makes it easier to match spikes when compared to the original SPIKE-synchronization (A-SPIKE-synchronization with $\mathcal{T} = 0$). It is important to note that reducing differences below threshold adds coincidences and thus, since it is a measure of similarity, A-SPIKE-synchronization can only increase.

2.2.4 *Selecting the threshold value*

In neuroscience typical time scales are in the range of milliseconds or sometimes seconds and any time scales below this will not be considered relevant. In fields such as meteorology the respective time scales could be hours and days or even months and years. The relevant time scales clearly depend on the system under consideration. Setting the minimum relevant time scale (MRTS) for a given dataset might not be a simple task. To address this, we propose a method to extract a threshold value from the spike trains, that is based on the proportions of the different time scales present in the data.

It is important to note that the selected MRTS is not an indicator of a time scale of the system; it just determines the outcome of the adaptive generalizations. It is also not a hard set limiter neglecting everything below the threshold, but rather it marks the time scale from which on differences are considered in the global instead of the local context. Thus from this time scale on deviations from synchrony are treated as less and

less relevant the smaller they get, even if they are large in relation to the local time scales.

The purpose of the threshold is to act as an indicator of what globally is a high rate or inversely a small ISI. The original normalizations are based on the ISIs, so it is reasonable to determine the threshold from the pooled ISI-distribution. We use the ISIs after the edge effect has been corrected (see 2.5.1.1). The threshold should fulfill two main criteria. First, it needs to decrease proportionally to the spike count, so that increasing rates (or longer recordings with the same rate) do not change the threshold. Second, the threshold should respond to changes in the ISI-distribution so that it is able to adapt between single and multiple time scale data sets. In Fig. 2.4 we use a simple spike train motive of just four spikes to illustrate these two criteria.

The most straightforward threshold would be the mean length of the ISIs

$$\langle L_{\mathrm{ISI}} \rangle = \frac{\sum_{g=1}^{G} L_{\mathrm{ISI}}^{g}}{M_{ISI}} = \frac{L}{M_{ISI}}. \tag{2.24}$$

Here L_{ISI}^{g} denotes the ISI-length and M_{ISI} is the total number of ISIs in the pooled ISI-distribution. In the numerator the sum of the lengths of all ISI equals the overall length L of the pooled ISIs. Apart from edge effect corrections this is equal to the product of recording length and number of spike trains which is a constant. Thus while the mean of ISIs depends on the number of spikes (Fig. 2.4A), for a given number of spikes (number of ISIs) it is completely independent of how the ISIs are distributed around the mean (Fig. 2.4B). It adapts to the spike count but not to the proportions in which the ISIs appear in the data thus fulfilling the first but not the second criterion.

To fulfill both criteria one needs to not just count the interspike intervals but weight them by their length. This reduces the importance of short ISIs and allows the long ISIs to influence the threshold according to their contribution and not just number. It is equivalent to taking the mean of the second moments of the ISIs

$$\mathcal{T} = \sqrt{\langle L_{\mathrm{ISI}}^{2} \rangle} = \sqrt{\frac{\sum_{g=1}^{G} {L_{\mathrm{ISI}}^{g}}^{2}}{M_{ISI}}}. \tag{2.25}$$

Note that in order to obtain a value with the right dimension the square root of the average must be taken. This threshold value has roughly the same dependence on the number of spikes as the mean value (Fig. 2.4A), however, in contrast to the mean it is also sensitive to changes in the ISI-distribution. In summary, using $\mathcal{T}$ as the MRTS fulfills both criteria set for the threshold.

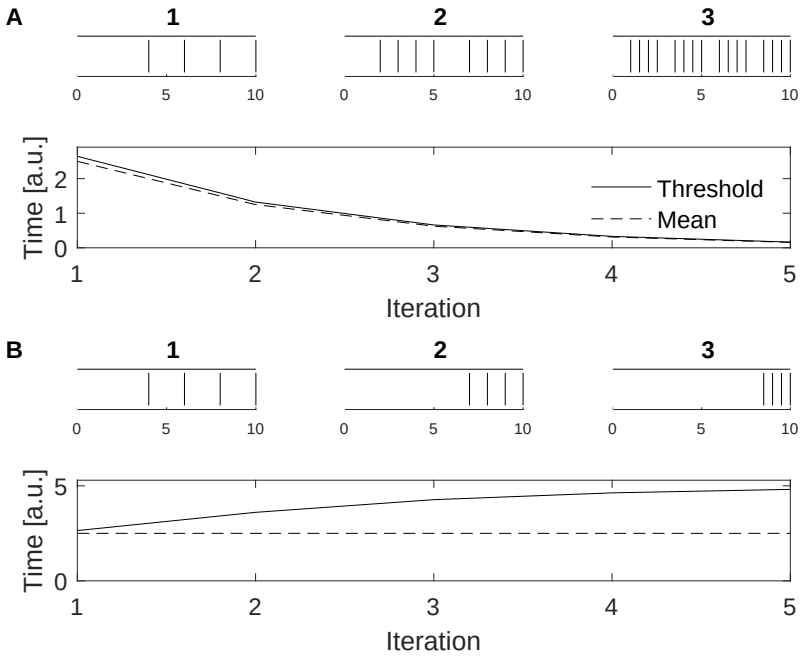

Figure 2.4: Threshold value vs. the mean of the ISI-distribution. (A) Dependence on the number of spikes (first criterion). In each iteration the number of spikes is increased by concatenating two half-length copies of the previous iteration. Both the mean and the threshold decrease with spike count. (B) Dependence on the ISI-distribution (second criterion). From iteration to iteration the ISI-distribution is changed by halving the three short ISIs and prolonging the long ISI accordingly. Since the spike count (and thus the number of ISIs) is kept constant, the mean does not respond to this change. However, the threshold correctly increases with the heightened importance of the long ISI.

2.2.5 *Results*

In this Section we investigate how both the adaptive generalizations (with automated thresholding) and the original measures deal with multiple time scales. For the A-ISI-distance and the A-SPIKE-distance we use a test spike train set consisting of simulated and real spike trains to study the effect of the generalized versions (Section 2.2.5.1). After that, in Section 2.2.5.2, we study on real MEA recordings how A-SPIKE-synchronization differs from SPIKE-synchronization. In Section 2.2.5.3 we systematically test the influence of the amount of bursts on the difference between adaptive and original measures. Finally, in Section 2.2.5.4

we investigate how the adaptive versions change the analysis of neuronal reliability in an experimental dataset.

2.2.5.1 *Adaptive ISI-distance and adaptive SPIKE-distance*

We address two points. First, we look at the sensitivity of the adaptive generalizations and verify that in the presence of bursts they perform better than the original measures. Second, we also make sure that the changes are specific, e.g., we confirm that in all other cases and especially if there are no bursts, the adaptive generalizations do not exhibit unwanted side effects.

To this aim, we use a test set composed of both artificial and real spike trains (Fig. 2.5) to compare A-ISI-distance to ISI-distance and A-SPIKE-distance to SPIKE-distance. We use two models to generate our samples. For the spike trains with perfect periodicity we use a time varying steady rate (fixed ISI) model. For samples with more variability in spike timing we used a Poisson spiking model, where the rate is fixed for a certain window at a time. In some cases we add small jitter noise to both models. The artificial spike trains 1-25 are designed to exhibit a variety of stereotypical spiking behaviours including both single and multiple time scales. The experimental spike trains 26-30 consist of short recordings from neuronal cultures on microelectrode arrays (see 2.5.2.1 for details). For the adaptive versions the threshold is estimated from the data (see Section 2.2.4) for each pair separately.

In the analysis every spike train is paired with all the others. Because for both the A-ISI-distance and the A-SPIKE-distance the MRTS $\mathcal{T}$ can only reduce but never increase the dissimilarity value, all pairs are found in the lower half of the scatter plot (Fig. 2.6). Furthermore, all values between pairs of spike trains are close to or on the diagonal, which means that both versions attain very similar values or even the same value. The differences between the two SPIKE-distances are slightly more pronounced than the differences between the two ISI-distances. Such seemingly small differences can still be of high significance since in a typical experimental setup it is rarely the absolute value of similarity that matters but rather the relative order of similarity for different conditions. Moreover, in real data the range of similarity values obtained is usually quite small which further increases the relative importance of small changes in similarity.

For one spike train at a time we then look at all its pairings and sort the results in ascending order according to the original versions. The results from the adaptive versions are arranged in the same order. If the

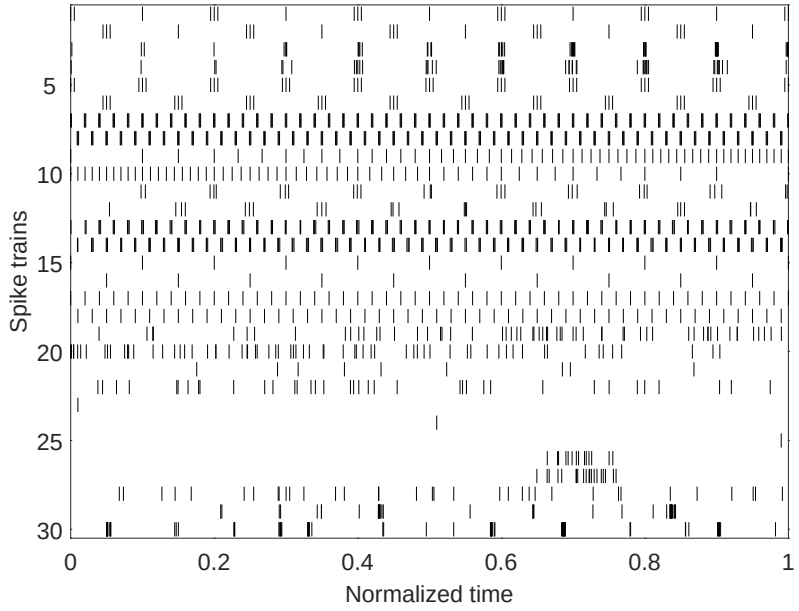

Figure 2.5: Spike train test set used to compare the generalized versus the original measures of spike train synchrony. Spike trains 1-25 are artificially constructed examples which cover a range of archetypical spiking patterns, whereas spike trains 26-30 are selected examples of neuronal spiking data from a neuronal culture recorded on a micro electrode array (see 2.5.2.1). All spike trains are normalized by their total length.

order of the spike train pairs does not match, there is a clear difference in the way the two measures consider spike train similarity.

We now investigate in more detail not only the largest absolute, but also the largest relative changes observed in Fig. 2.7. First, the largest absolute changes are identified by calculating the Euclidean distances between the results for the two spike train pairs. They typically take place for pairs of spike trains with large distances. Next, since deviations from near perfect synchrony are more prominent and easier to detect than differences between various levels of high dissimilarity, we also look at relative changes. These can be found by dividing each distance by its corresponding ISI- and A-ISI-distance or SPIKE- and A-SPIKE-distance average. For both distances they mostly occur for pairs of very similar spike trains.

For the A-ISI-distance, the spike train pairs showing the largest absolute change compared to the ISI-distance can be seen in Fig. 2.7A. The two measures show a different order of similarity; while the ISI-distance increases, the A-ISI-distance decreases from the first to the second pair.

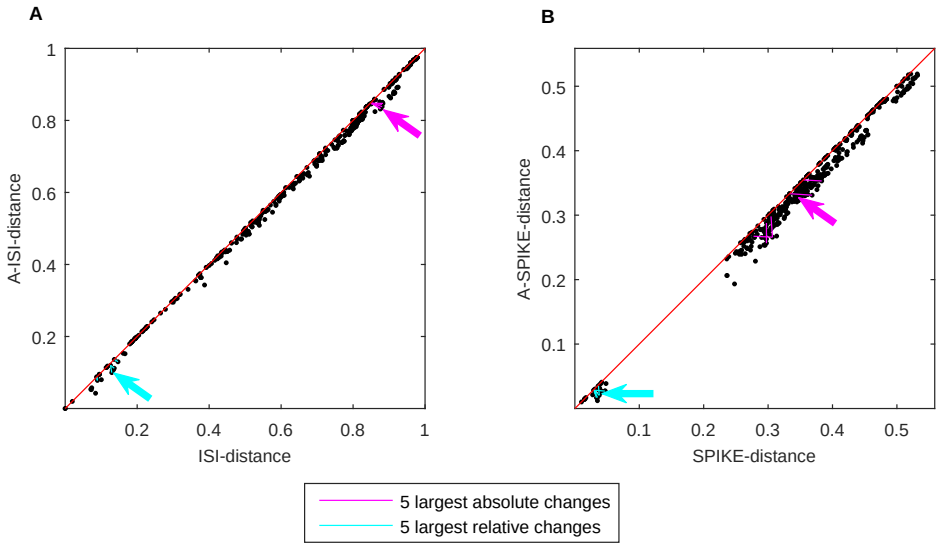

Figure 2.6: Scatter plots showing the A-ISI-distance between all pairs of spike trains versus the original ISI-distance (A) and the A-SPIKE-distance versus the SPIKE-distance (B). The diagonal line marks where the measures would show equal distance. The pairs were sorted according to rising order of the original distances- Thus, if the order changed for adaptive extension, there is a negative slope in a line connecting all the pairs. For each line with a negative slope we calculated its length using the Euclidean distance. The five largest absolute changes are indicated in magenta, the five largest relative changes in cyan. In addition, a magenta (cyan) arrow points to the very largest absolute (relative) change. Overall, while the changes are seemingly small on an absolute scale, the relative changes can be very significant.

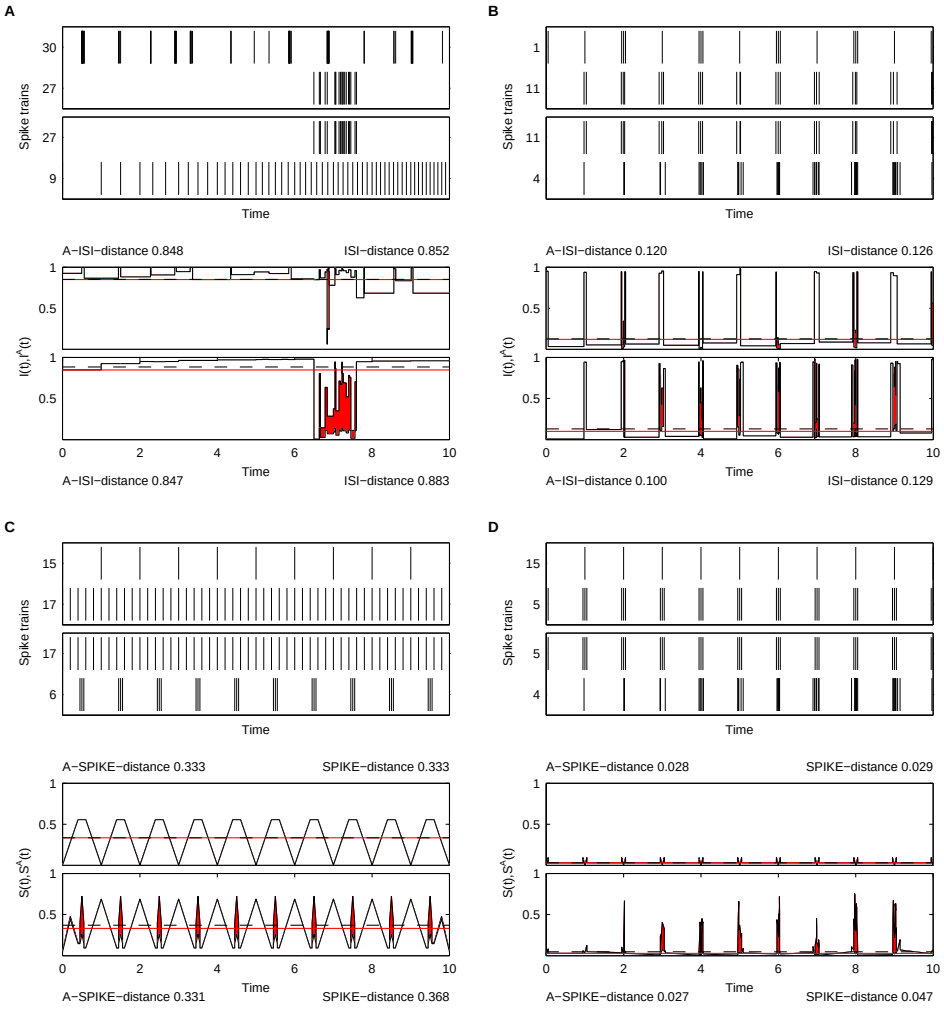

Figure 2.7: Largest absolute and largest relative change between ISI-distance and A-ISI-distance (A,B) as well as SPIKE distance and A-SPIKE-distance (C,D) for the spike train set shown in Fig. 2.5. In both cases the two measures show a different order of similarity. The original distances increase from the first to the second pair, while the adaptive extensions decrease. The first pairs attains distance values in between the second pairs, which results in a different order of similarity between the measures (see Fig. 2.6). Upper subplots: The spike train pairs with the largest changes. Lower subplots: Respective original distance vs. adaptive version profiles with the difference between the two profiles emphasized. The distance values for the first (second) pair are shown on top (at the bottom). They are also marked by a dashed line for the original distance and by a solid line for the adaptive distance.

Spike trains 27 and 30 in the first pair are seen very similarly (deviation $< 1\%$) by both measures. However, when spike train 27 is paired with spike train 9, the ISI-distance considers the local time scale only and thus has unreasonably high demands on the spikes of the burst in spike train 27 which leads to large fluctuations in similarity. For the A-ISI-distance on the other hand the burst matches another event with relatively high rate and treats this event as a coinciding burst. Outside of the burst the two measures agree that the spike trains are very dissimilar.

Fig. 2.7B depicts the two spike train pairs exhibiting the largest relative change between the two ISI-distances. While the first pair is seen as relatively similarly (deviation $< 5\%$), the main difference is found for the second spike train pair. Here the ISI-distance looks at the detailed structure and judges the interspike intervals within the bursts as very dissimilar, whereas the A-ISI-distance sees simply matching bursts and attains a considerably lower distance value than the ISI-distance (0.100 vs 0.129).

For the A-SPIKE-distance, the pairs of spike trains showing the largest absolute change compared to the SPIKE-distance are depicted in Fig. 2.7C. As there are no bursts in either of the two spike trains, both measures attain exactly the same value for the first spike train pair. This is a very good example for the specificity of the generalized version. On the other hand, the original distance considers the second spike train pair (periodic spiking versus periodic bursts) as much more dissimilar (increase $> 10\%$). In contrast to the SPIKE-distance, it rightly considers the spike time differences in the middle of two bursts as larger than the differences in the middle of the burst.

Finally, the largest relative change between the two SPIKE-distances is shown in Fig. 2.7D. Again, there is not much difference between the two distances for the first spike train pair. However, the SPIKE-distance considers the second spike train pair much more dissimilar ($> 62\%$ higher) due to the large relative deviations in spike timing within their coinciding bursts. In contrast, the A-SPIKE-distance puts much less weight on the differences within bursts, but still reacts to the spikes outside of the bursts. This is an example of the sensitivity of A-SPIKE-distance.

All these results show that the effect of both generalized versions is strongest in situations with multiple time scales in the spike trains. A prominent example are bursts embedded in long silent periods. In this case the long ISIs (of the silent periods) strongly influence the global time scales such that deviations of synchrony on the smallest time scales (within the bursts) are weighted less.

2.2.5.2 *Adaptive SPIKE-synchronization*

A-SPIKE-synchronization can not be meaningfully tested by using the spike train set of Fig. 2.5. The perfect periodicity in many spike trains makes analysis of the A-ISI-distance and the A-SPIKE-distance simple, but causes very abrupt changes in A-SPIKE-synchronization due to its binary nature. The values can be computed, but the largest differences are not meaningful with this data set, since many spike trains with bursts jump from zero to a large value and there is no way of ordering different pairs having zeros in the original measure. Thus, we here use a qualitative approach together with insights from the analysis of A-ISI-distance and A-SPIKE-distance.

As a side effect of being time scale adaptive, SPIKE-synchronization demands very high spike timing accuracy during fast firing. This leads to situations such as the one shown in Fig. 2.3 (Section 2.2.3). For spike trains 3 and 4 the spikes in the first event are considered coincident, but the doublet in between them in spike train 2 is not judged as coincident with either of them. In contrast, for A-SPIKE-synchronization the coincidence windows are adapted to the distribution of all ISIs in the data set and the two sides of the coincidence window are allowed to be of different length. With this change each of the spikes in the doublet becomes coincident with one of the spikes (the respective closest one) in spike trains 3 and 4.

As a by-product of the adaptation, A-SPIKE-synchronization also increases the coincidence window coverage within and at the edges of a burst and thus matches as many spikes as possible. For SPIKE-synchronization many of these spikes would be ignored due to the unreasonably small coincidence windows. This phenomenon occurs very often with real data. An example containing two small and one large burst event is shown in Fig. 2.8. In the first two events A-SPIKE-synchronization is able to detect a few additional coincidences compared to SPIKE-synchronization. The difference is much more pronounced for the third and largest event. Here for SPIKE-synchronization many potential matches are left out and this leads to a rather low overall value of 0.238. Instead, when A-SPIKE-synchronization is used, there are almost 45% more matched spikes within the burst and this strongly increases the overall synchronization value to 0.345.

Fig. 2.8C clearly shows that the additional spike matching of A-SPIKE-synchronization only occurs in the high frequency events for which small differences in the ISIs cause gaps between the coincidence windows of adjacent spikes. Coincidences outside of these high frequency events are not affected.

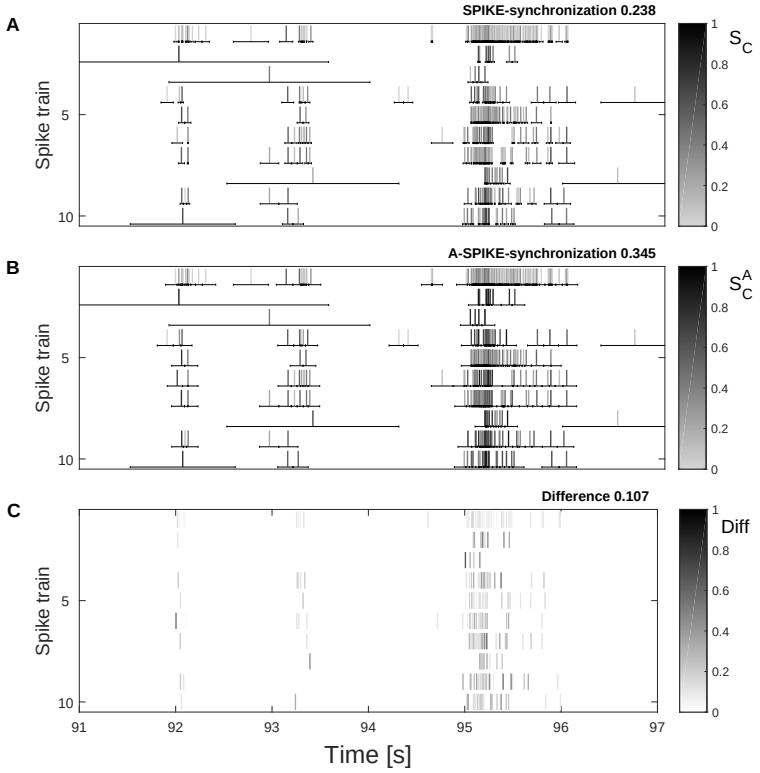

Figure 2.8: Real data example from MEA recordings (see 2.5.2.1 for more details). Ten spike trains from the data set are plotted and their coincidence windows are drawn as obtained by SPIKE-synchronization (A) and A-SPIKE-synchronization (B). The difference is plotted in C. Due to the adaptive coincidence windows, A-SPIKE-synchronization is able to match around 45% more spikes between bursting spike trains than SPIKE-synchronization. As in Fig. 2.3, the color scale is grey-black in A and B and white-black in C.

2.2.5.3 *Systematic evaluation of the influence of bursts*

Next we test how the effect of the automated threshold changes when spikes are forming tighter bursts (Fig. 2.9). To do this we first create two Poisson spike trains which are divided into four equally long segments. These segments are then increasingly compressed which prolongs the ISIs between them such that the total length remains constant (Fig. 2.9A). We use the relative length of the interburst intervals R as a parameter and track the difference between the adaptive and the original versions. The results for the ISI-distance and the SPIKE-distance are very similar and we only show the latter. From Fig. 2.9B we can see that the SPIKE-distance decreases almost linearly with R since the relative importance of the common silence in the interburst intervals increases.

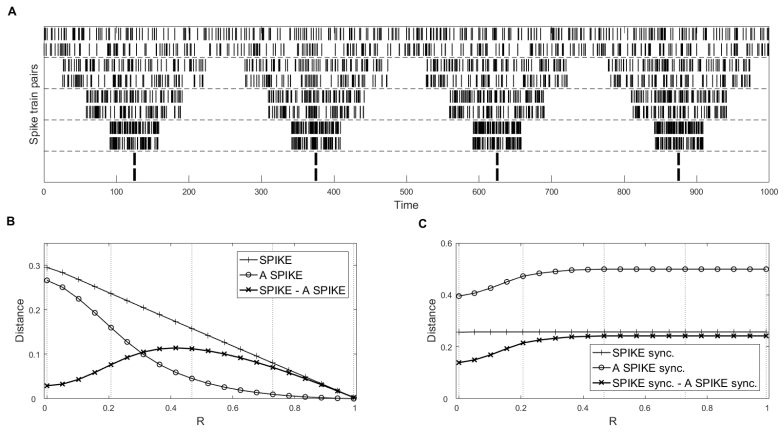

Figure 2.9: Effect of bursts on the adaptive versions evaluated by using the relative length R of interburst intervals. The values are averages over 10 realizations. (A) Five spike train pairs with increasing levels of burstiness for one example realization. (B) Effect of burstiness on the difference between A-SPIKE-distance and SPIKE-distance. The graph for the ISI-distance looks very similar and is thus omitted. (C) Equivalent results for A-SPIKE-synchronization. The R-values of the examples in (A) are marked in (B) and (C) as dotted vertical lines.

The adaptive version decreases sub-linearly with the largest absolute difference between the two measures occurring around $R = 0.4$. For higher R-values the reduction of the burst length overshadows the increases in similarity at burst times and the difference increases up to a point and then starts to decrease. The relative difference increases over the whole interval (data not shown, but can be appreciated by observing the difference approaching the SPIKE-distance value towards $R = 1$).

While for the SPIKE-distance the interburst intervals have an effect on the overall value, SPIKE-synchronization is sensitive to the matching of spikes only and is based on one coincidence indicator value per spike. Thus, the effect is increasing only until all possible spike pairs within the bursts are matched. For our example the increase saturates at $R = 0.4$ (Fig. 2.9C) at which point all possible spike pairings (encompassing roughly half of the spikes) have been identified. This is in agreement with what we demand from a distance sensitive to bursting structure for a systematic increase of the ratio between interbursts intervals and synchronous bursts.

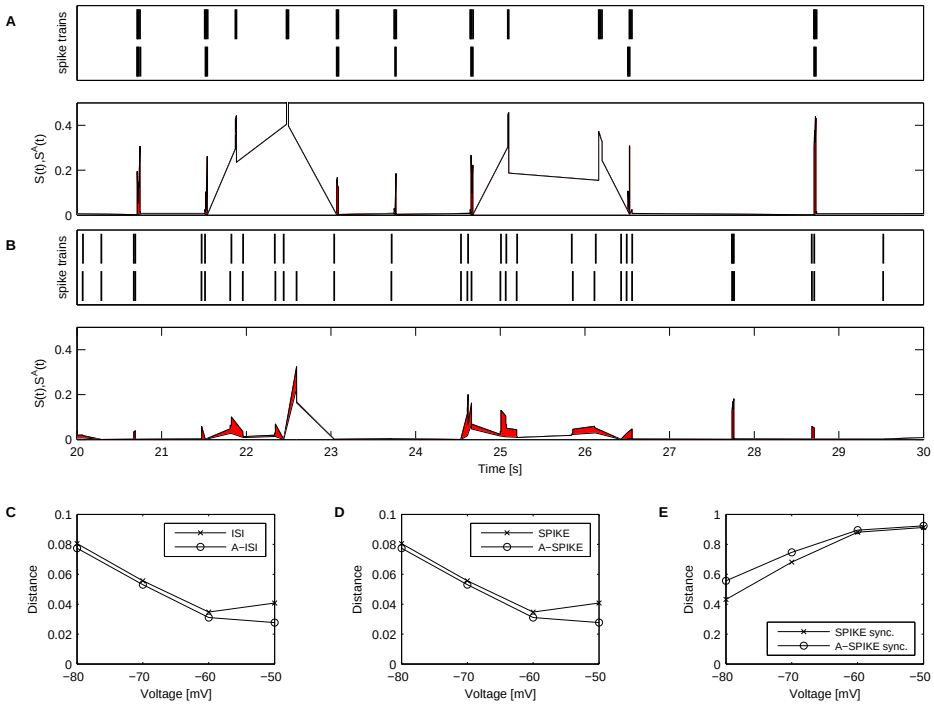

Figure 2.10: Analysis of neuronal responses to multiple presentations of frozen noise for four different levels of the membrane potential. (A) Spike train responses (top) to two of the three noise presentations for a membrane potential of $-80mV$ and corresponding profiles for both A-SPIKE-distance and SPIKE-distance (bottom). The difference between the two profiles is marked in red. (B) Same as in (A) but for a membrane potential of $-50mV$. Results of the original and the adaptive measures for spike train sets of all three trials at four different voltage levels for the ISI-distance (C), the SPIKE-distance (D) and SPIKE-synchronization (E).

2.2.5.4 *Application to real data: Reliability of neurons*

In order to demonstrate the effects of the adaptive generalization in a more realistic scenario, we re-analyze data previously used to study the effect of membrane potential resting state on neuronal reliability ((27), see 2.5.2.2 for details on the recordings). When in the original study frozen noise was injected into thalamocortical relay cells of rats, it was found that the reliability of the cell response increases with depolarization (27).

Here we use both the original versions and the adaptive generalizations of all three measures to assess the reliability of the responses from the two neurons for which all four levels of membrane potential were recorded. The adaptive versions use a threshold obtained from the data

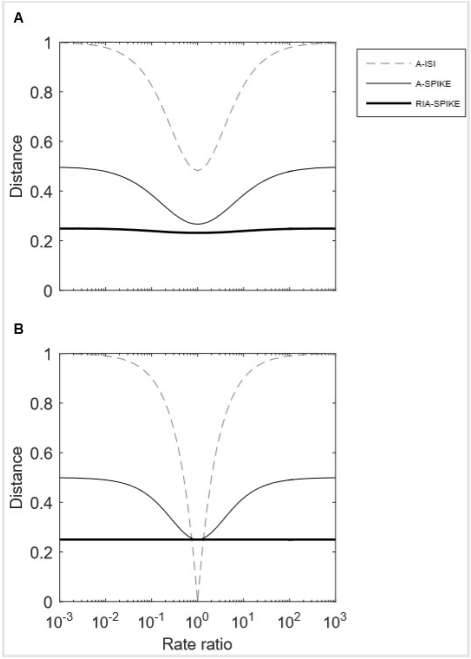

Figure 2.11: Rate-independent RIA-SPIKE-distance vs. A-SPIKE-distance and A-ISI- distance for Poisson (A) and steady-rate spike trains (B). (A) Distances for two Poisson spike trains with varying rate ratios. The overall number of spikes in the two spike trains is kept constant. Each data point is an average over 100 trials. In contrast to the clearly rate-dependent A-ISI- and A-SPIKE-distances, the rate-independent RIA-SPIKE-distance exhibits an almost constant curve. (B) For the steady-rate spike train curves each data point is an average over 100 trials with random phase shifts between the two spike trains. In this case the line for the RIA-SPIKE-distance is indeed constant.

by pooling all spike trains of each level and trial together. In Fig. 2.10 we show the results of the cell with the more prominent effect but we get similar results for the other cell as well. The cells analyzed were recorded three times for each holding membrane potential and reliability was assessed by trial to trial variations. For the highest hyperpolarization (Fig. 2.10A) the original SPIKE-distance yields spuriously high values for the local dissimilarity during the bursts, since it only evaluates the local context. Even when the A-SPIKE-distance takes the global context into account, both measures agree that there are large dissimilarities in the spike trains.

For the most depolarized state (Fig. 2.10B) the membrane potential is considerably closer to the action potential threshold. The patterns are

closely matching the burst positions of Fig. 2.10A, but also additional events appear. The neuron no longer responds in clearly distinguished bursts and it is considerably more difficult to determine where a burst begins or ends. Since the generalized version adapts to time scales found in all the spike trains, it is able to distinguish when a burst-like pattern emerges and considers them as more similar.

As can be seen in Figs. 2.10C and 2.10D, the original versions, without adaptation and only using the local context, attain a higher level of similarity for -60mV than for -50mV, which contradicts both the results in the original study and the results for SPIKE-synchronization (Fig. 2.10E). Since the adaptive versions are able to make use of the global context of all the spike trains, they attain results without this spurious dissimilarity and thus for higher membrane potentials the similarity increases monotonously.

A-SPIKE-synchronization works slightly differently (Fig. 2.10E). Due to the tight bursts that cause excessively small coincidence windows, the largest difference occurs for the hyperpolarized states. However, both versions agree that the reliability as quantified by spike to spike matching in the response patterns clearly show a monotonous increase over the baseline membrane potential. In summary, the results obtained by the A-SPIKE-distance and A-ISI-distance seem to be appropriate and more in line with the original results.

2.3 RATE-INDEPENDENT EXTENSION

Sometimes in neuroscience one is interested in the pure similarity of spike timing, independent of any differences in spike rates. Thus there is the need for a measure which can identify differences in spike timing but is able to ignore any differences in rate between the spike trains. Here we propose such a rate-independent extension for the A-SPIKE-distance.[1]

2.3.1 *RIA-SPIKE-distance*

In order to understand how rate-independence for A-SPIKE-distance is achieved, we need to separate Eq. 2.11 (Section 2.2.2) for the pairwise A-SPIKE-distance profile into its three components.

1 The A-ISI-distance is a measure of instantaneous rate difference and a rate-independent measure of rate difference makes little sense. A-SPIKE-synchronization is rate-dependent by definition, since it is calculated as the average value of spike-based coincidence indicators (Eqs. 2.20 and 2.23).

The first two components are the mean of spike time dissimilarity and the normalization to firing rate

$$S_{m,n}(t) = \frac{S_n(t) + S_m(t)}{2} \cdot \frac{1}{max\left\{\left\langle x_{\mathrm{ISI}}^{n,m}(t)\right\rangle, \mathcal{T}\right\}}, \tag{2.26}$$

where $S_n(t)$ and $S_m(t)$ are the weighted spike time differences for spike trains n and m defined by Eq. 2.10. The third component is a weighting of the spike time dissimilarity according to the firing rate difference that is applied to the first component

$$\frac{S_n(t)x_{\mathrm{ISI}}^m(t) + S_m(t)x_{\mathrm{ISI}}^n(t)}{\left\langle x_{\mathrm{ISI}}^{n,m}(t)\right\rangle}. \tag{2.27}$$

The rate-independent adaptive SPIKE-distance (RIA-SPIKE-distance) simply leaves out this last weighting and can thus be written as

$$S_{m,n}^{RIA}(t) = \frac{S_n(t) + S_m(t)}{2\max\left(\left\langle x_{\mathrm{ISI}}^{n,m}(t)\right\rangle, \mathcal{T}\right)}. \tag{2.28}$$

The RIA-SPIKE-distance shares all the properties of A-SPIKE-distance, but it only evaluates normalized spike timing differences, whereas the A-SPIKE-distance additionally uses differences in rate to determine similarity.

2.3.2 *Results*

In this Section we compare the RIA-SPIKE-distance to the regular A-SPIKE-distance regarding their response to differences in rate. First, in Fig. 2.11A we look at Poisson spike trains with different rate ratios. The regular A-SPIKE-distance exhibits a clear rate dependency obtaining its lowest value for spike trains with identical rates and increasing for higher rate differences. The RIA-SPIKE-distance on the other hand starts near 0.25 and remains relatively constant for all rate ratios. These deviations from perfect rate-independence occur because of the irregularities of the Poisson spike trains. When we repeat the same analysis with steady rate instead of Poisson spike trains (Fig. 2.11B), thereby removing the effects of the Poisson statistics, the RIA-SPIKE-distance exhibits indeed perfect rate-independence.

Regarding the original distances, in Fig. 2.11A they would show very similar behavior to the adaptive generalizations. Only for rate ratios close to 1 there would be a small increase due to coincident burst-like events within the Poisson spike trains. In Fig. 2.11B the curves would overlap perfectly since there is only one time scale in steady rate spike trains (both results not shown).

2.4 DISCUSSION

In this manuscript we introduce adaptive generalizations to the three existing measures ISI-distance, SPIKE-distance and SPIKE-synchronization as well as a rate-independent extension to the generalized SPIKE-distance. These new measures address two distinct problems.

The adaptive generalizations allow to disregard spike time differences that are not relevant on a more global scale. By means of a specifically constructed library of both stereotypical and real data spike trains, we can show that both A-ISI-distance and A-SPIKE-distance indeed yield improvements for pairs of spike trains containing different time scales without exhibiting any unwanted side effects in other examples. Thus the changes are both sensitive and specific. Regarding the size of the changes, even if they are seemingly small on an absolute scale, the relative changes can be very significant. For our test set the largest relative change reaches 29% for the A-ISI-distance and even up to 62% for the A-SPIKE-distance. With a more qualitative approach we then show that A-SPIKE-synchronization fixes the problem of SPIKE-synchronization which demands an unreasonably high accuracy for spike doublets and coinciding bursts. By introducing a global reference frame, it manages to match spikes more efficiently (for our test data we found an increase of 45%).

In order to test the adaptive measures methodologically we tested them in a controlled environment where two Poisson spike trains were split into bursts using increasingly large interburst intervals. We designed the adaptive extension to be sensitive to bursting structure, therefore for increasing relative length of interburst intervals we expect a larger difference between the original and the adaptive versions. We show that the relative difference indeed increases monotonously with an increase in the ratio between interbursts interval and bursts.

The absolute difference obtains its maximal value when the differences ignored in the bursts are large and the bursts are long enough in comparison to the total length of the recording. When very similar spike trains are compared their relative difference becomes dominant and internal structures of coinciding bursts become less relevant.

Additionally, we apply the measures to a dataset previously analyzed for reliability and find that the adaptive methods agree with the previous results better than the original versions. The A-ISI-distance and the A-SPIKE-distance seem to yield more reasonable results than the ISI-distance and the SPIKE-distance. On the other hand when the coincidence windows of the original version get spuriously small, A-SPIKE-synchronization can match spikes much more efficiently. The effect can

be especially meaningful in applications in which leader-follower relationships based on the temporal order of spikes are determined (28).

The rate-independent extension on the other hand focuses on spike time accuracy while disregarding rate differences in the two spike trains. The original SPIKE-distance considers spike time differences but also has a feature that takes into account the firing rate difference between the spike trains. However, sometimes only the spike time accuracy is of interest and for that purpose the RIA-SPIKE-distance disregards any deviations in firing rate. We can show that the RIA-SPIKE-distance is approximately rate-independent for Poisson spike trains (apart from minor statistical effects) and perfectly rate-independent for strictly periodic spike trains. With this final addition we have completed the picture, since we now have measures that are sensitive to rate only (A-ISI-distance), to timing only (ARI-SPIKE-distance), and to both at the same time (A-SPIKE-distance).

The adaptive generalizations are implemented for cases where we have prior knowledge of the system or where we want to reduce the importance of very small details. However, one has to be careful with this method. If the threshold parameter that defines the minimum relevant time scale (MRTS) is chosen too high, this can introduce spurious synchrony. To facilitate the selection, we introduce a method for automatically extracting the threshold from the spike train data. This is done by using the second moment the ISI-distribution of the whole dataset, thereby giving more weight to longer ISIs.

Here it is important to note that while this automated estimation of MRTS gives us a threshold value for each dataset, one has to be very careful when comparing results obtained with different threshold values. Thus, one cannot use the adaptive version for two recordings from the same source without using the same threshold for both recordings, even if the ISI-distributions differ. In such cases, the preferable option would be to combine the ISI-distributions before calculating the threshold and to use the resulting value for both recordings. However, this might not work in all cases. For example, recordings before and during an epileptic seizure can have very different ISI-distributions. This means that a globally meaningful threshold can not be extracted due to a very bi-modal distribution of all the ISIs from the whole recording. The resulting threshold would be in between the two modes which would cause the adaptive measures to basically consider one of the recordings as a long burst and the other as an almost silent period. Thus, in cases where a suitable threshold can not be found, it is preferable to just set it to zero and consider only local information. This is equivalent to using the original versions.

Many time scale parametric measures like the Victor-Purpura and the van Rossum distance use a parameter to define the time scale of the system. The threshold set for the adaptive versions is philosophically different in the sense that it does not define a single time scale, but sets a line below which the effects of the smaller time scales are being toned down. All different time scales are still considered at the same time, but weighted differently depending on how they compare to the threshold.

Other measures that deal with multiple time scales exist. For example, Lyttle and Fellous have proposed a metric to specifically assess the similarity of spike trains with bursts or common silent periods (29). While in the proposed adaptive measures the time scale parameter is limiting full time scale independence of the original measures, in many measures the time scale is a fixed value. With the method proposed by Lyttle and Fellous they can detect bursts as well as silent periods. However, this comes with a cost, since the method requires two time scale parameters and three additional parameters; length of minimum silent period, length of burst ISI, minimum number of spikes in a burst, scaling factor to decide how important bursts are in comparison to single spikes, and another factor to decide between importance of burst and silent period detection. While the large array of options gives the experimenter a powerful tool and provides more control over the analysis, it also increases the complexity of the overall experiment. This may cause problems, in particular when the data has many dimensions. Similarly, Rusu and Florian have introduced a new class of metrics (30). The max-metric and the modulus-metric are well suited for measuring distances between spike trains where information is encoded in bursts but single spike accuracy within burst is not relevant. The max-metric depends on the kernel chosen and a time scale parameter deciding its size. The modulus-metric is parameter free like the ISI-distance, the SPIKE-distance, and the SPIKE-synchronization. This is achieved by using a very simplified kernel. However, the results obtained with both methods are not normalized. Thus based on the dissimilarity value alone it is not possible to say anything about the similarity of the two spike trains, but only about the order of different pairs.

Another often used alternative to spike train distances are correlation measures (see e.g. (31)). However, these measures traditionally require windowing or binning and this creates the problem that their performance can depend crucially on the window length or bin size and also on the starting points and the overlap of the windows which clearly reduces the objectivity of the results.

The results confirmed our initial expectation that the main differences between the adaptive generalizations and the original measures is in

their assessment of the similarity of bursty data. Since bursts are ubiquitous and have been identified as an important area of neuroscience research (see e.g. (32, 33)), there is a strong need for this kind of similarity measurement. For the ISI-distance, a method has been proposed for evaluating the similarity of bursty data by identifying bursts and assigning spikes at the beginning of the bursts (34). However, burst detection is a notoriously difficult problem for which rather complicated methods have been developed (see for example (35)). Thus, a measure based on assigning spikes to bursts inherits the problems of burst detection. Another problem with the measure proposed in (34) is that it disregards differences in spiking behavior within the bursts. In contrast, our adaptive versions do not detect bursts at all, but automatically adapt their behavior whenever there are burst-like features in the data.

All the measures presented here are symmetric and thus invariant to the order of the spike trains. Recently we have developed a complementary directional approach consisting of two new measures called SPIKE-Order and Spike Train Order (28). This approach utilizes the adaptive coincidence detection of SPIKE-synchronization to first sort multiple spike trains from leader to follower and then to quantify the consistency of the spatio-temporal propagation patterns. A natural continuation of the work presented in this article would be to use the adaptive measures for this new approach as well.

2.5 APPENDIX

2.5.1 *Edge effect correction and treatment of special cases*

Here, we deal with some subtle details in the definitions of all three measures A-ISI-distance, A-SPIKE-distance and A-SPIKE-synchronization. First, in 2.5.1.1 and 2.5.1.2, we correct the edge effect by providing definitions for the periods before the first and after the last spike in a spike train (for which the interspike interval is not defined). This is necessary to guarantee that all measures are well-defined for the whole recording interval. Subsequently, in 2.5.1.3 and 2.5.1.4, we deal with the two special cases of empty spike trains and spike trains with only one spike. Even if some spike trains are empty or very sparse, all measures should still be defined in a way which is consistent with the regular definitions.

2.5.1.1 *Edge effect correction for A-ISI- and A-SPIKE-distance*

Since the A-ISI- and the A-SPIKE-distance are time-resolved and are based on ISIs defined by Eq. 2.3, there is ambiguity at the edges before the first spike and after the last spike. To resolve this ambiguity we need to add auxiliary spikes. For the beginning of the spike train, we assign an auxiliary spike at the maximum of the distance between the start of the observation interval and the first spike, and the first known ISI

$$t_{s_{aux}}^{(n)} = t_1^{(n)} - \max\left\{t_1^{(n)} - t_s, t_2^{(n)} - t_1^{(n)}\right\}. \tag{2.29}$$

This definition assumes that the rate stays the same at both sides of the spike unless the edge is too far away for this to be true, in which case the auxiliary spike is assigned at the edge. Analogously, the time of the auxiliary spike at the end is

$$t_{e_{aux}}^{(n)} = t_M^{(n)} + \max\left\{t_e - t_M^{(n)}, t_M^{(n)} - t_{M-1}^{(n)}\right\}. \tag{2.30}$$

If the first or last spike is at the edge, no edge correction is necessary at that end. This defines the ISI which is then used not only for the ISI-distance but also for the A-SPIKE-distance and A-SPIKE-synchronization.

An auxiliary spike used for the edge effect correction is basically treated as any other spike, for example they can be the nearest neighbor to a real spike. But there is one exception: In order to avoid artificial synchrony at the edges in the A-SPIKE-distance, they use the distance to the nearest neighbor from the first/last real spike

$$\Delta t_{s_{aux}}^{(n)} = \Delta t_1^{(n)} \qquad \text{and} \qquad \Delta t_{e_{aux}}^{(n)} = \Delta t_{M_n}^{(n)}. \tag{2.31}$$

2.5.1.2 *Edge effect correction for A-SPIKE-synchronization*

For the A-SPIKE-synchronization profile we first apply the edge effect correction described above and then calculate the coincidence windows following Eqs. 2.14 and 2.15.

For cases when there is a spike right at the edge, we use the one ISI that exists for setting the coincidence window of the spike to

$$\tau_1^{(n)} = \tfrac{1}{2} x_{1F}^{(n)} \qquad \text{and} \qquad \tau_M^{(n)} = \tfrac{1}{2} x_{MP}^{(n)}. \tag{2.32}$$

We also determine that an auxiliary spike can under no circumstance be part of a coincidence nor can it have a coincidence counter. Finally, an auxiliary spike does not count as a spike in the normalization.

2.5.1.3 *Special cases for A-ISI- and A-SPIKE-distance*

Empty spike trains and spike trains with only one spike do not provide the ingredients needed to apply Eq. 2.29 and 2.30.

In order to define the ISI of an empty spike train without any spikes, we assign auxiliary spikes to its edges, the beginning and the end of the recording interval. This is the only interval for which we can guarantee that there were no spikes.

However, while we can now use Eq. 2.3, Eq. 2.31 for the distance to the nearest neighbour of the auxiliary spikes is still ill-defined, since there are no real spikes. In this case a value is assigned exactly as in Eq. 2.7 and the nearest neighbor can either be a real or another auxiliary spike. A very reasonable implication of this definition is that two empty spike trains will be considered equal by both measures.

Similarly, it is not possible to assess the rate at either side of a single spike. The most reasonable auxiliary spike location is again at the edge of the recording. Thus for both cases, the auxiliary spikes are assigned at the edges as

$$t_{s_{\text{aux}}}^{(n)} = t_s \qquad \text{and} \qquad t_{e_{\text{aux}}}^{(n)} = t_e \tag{2.33}$$

and this completes the definitions for the A-ISI- and the A-SPIKE-distance.

2.5.1.4 *Special cases for A-SPIKE-synchronization*

For A-SPIKE-synchronization the situation is slightly different, since it is not continuous but only defined at the times of the spikes. This means that by definition an empty spike train cannot have synchronous spikes and thus has no value. In case all spike trains are empty, we set A-SPIKE-synchronization to $S_C^A = 1$, i.e. empty spike trains are considered to be perfectly synchronous. If a spike train contains only a single spike, we

use half the spike train length to define the coincidence window for the spike as

$$\tau_1^{(n)} = \tfrac{1}{2}(t_e - t_s). \tag{2.34}$$

These special cases complete the definition of A-SPIKE-synchronization.

2.5.2 *Experimental recordings*

2.5.2.1 *Microelectrode array recordings from mouse cortical cells*

The electrophysiological data analyzed in Sections 2.2.5.1 and 2.2.5.2 were recorded in the group of Prof. Jari Hyttinen at Tampere University of Technology / BioMediTech, Tampere, Finland. These recordings were performed prior to and independently from the design of this study.

Between 5,000 and 25,0000 commercially available primary mouse cortical cells (A15586, Gibco, Thermo Fisher) were plated on five microelectrode arrays (MEAs; four 60MEA200/30iR and one 60HDMEA30/10iR, all purchased from Multi Channel Systems, Reutlingen, Germany) following the protocol of Hales, Rolston, and Potter (36). The dishes were coated with poly-L-lysine (Sigma-Aldrich, St. Louis, MO, USA) and laminin (L2020-1MG, Sigma-Aldrich). The medium for the MEA cultures was replaced three times a week. All MEAs with cells were kept in an incubator (+37 °C, 5% CO2, 95% air) prior to and between recordings. Data were recorded three times a week between the 4th and the 35th day in vitro. Every recording lasted five minutes and was performed with 25 kHz sampling rate. Spike detection was carried out by setting an amplitude threshold at five times the standard deviation of the signal-noise level and the spike time stamps were stored with the Neuroshare Library for MATLAB (Multi Channel Systems). We used two recordings for our examples and test sets.

The five real data spike trains used in the test set (spike trains 26 to 30 in Fig. 2.5) were selected from these data by hand to represent different time scales but chosen such that spike numbers were quite constant and comparable to the artificial examples.

2.5.2.2 *Patch clamp recordings of rat thalamocortical relay cells*

The electrophysiological data analyzed in Section 2.2.5.4 were recorded at the Swammerdam Institute for Life Sciences, University of Amsterdam, the Netherlands. Again, these recordings were performed prior to and independently from the design of this study. The experiments carried out on brain slices from Wistar rats (Harlan, Netherlands; post-

natal days 12-16) were approved by the animal welfare committee of the University of Amsterdam.

For details on the animals, slice preparation and electrophysiological recordings, see (27). In the current-clamp measurements the cell was injected with current that consisted of a DC component with superimposed noise: a computer generated (MATLAB) time series of Gaussian distributed random numbers of a length of 300 s, filtered by an exponential filter with a time constant $\tau = 10$ ms and a standard deviation of $\sigma = 100$ pA. A slow feedback system controlled the background DC current to stabilize the membrane voltage at one of the specified values (-80 mV, -70 mV, -60 mV or -50 mV) before the actual recording started; after the start this DC current component was fixed. The same frozen (= an exactly reproduced computer generated) noise train was injected into the soma of the TCR neuron for every repetition of the experiment. Signals were filtered at 510 kHz and sampled at 1020 kHz.

The recordings consisted of trials from five different cells of which only two included trials for all four levels of membrane potential. The cells analyzed were recorded three times and reliability was assessed by trial to trial variations.

3

SPIKE TRAIN ORDER

Repetitive spatio-temporal propagation patterns are encountered in fields as wide-ranging as climatology, social communication and network science. In neuroscience, perfectly consistent repetitions of the same global propagation pattern are called a *synfire pattern*. For any recording of sequences of discrete events (in neuroscience terminology: sets of spike trains) the questions arise how closely it resembles such a synfire pattern and which are the spike trains that lead/follow. Here we address these questions and introduce an algorithm built on two new indicators, termed *SPIKE-Order* and *Spike Train Order*, that define the *Synfire Indicator* value, which allows to sort multiple spike trains from leader to follower and to quantify the consistency of the temporal leader-follower relationships for both the original and the optimized sorting. We demonstrate our new approach using artificially generated datasets before we apply it to analyze the consistency of propagation patterns in two real datasets from neuroscience (Giant Depolarized Potentials in mice slices) and climatology (El Niño sea surface temperature recordings). The new algorithm is distinguished by conceptual and practical simplicity, low computational cost, as well as flexibility and universality.

Adapted from: *Leaders and followers: quantifying consistency in spatio-temporal propagation patterns* (9)

3.1 INTRODUCTION

Recordings of spatio-temporal activity are ubiquitous in many scientific disciplines. Among the most prominent examples are large-scale electrophysiological measurements of neuronal firing patterns in experimental neuroscience (37, 38) and sensor data acquisition in seismology (39), oceanography (40), meteorology (41), or climatology (42). Other examples include interaction protocols in social communication (43, 44) or monitoring single-node dynamics in network science (45).

In all of these fields recordings often exhibit well-defined patterns of spatio-temporal propagation where some prominent feature first appears at a specific location and then spreads to other areas until potentially becoming a global event. A propagation pattern can also be seen as a recurring spike pattern or "unitary events", where the same spike pattern is repeated by the same neurons (16). However, in our work we do not expect repetitions of same event, but rather just look at the temporal order of all spikes. Such characteristic propagation patterns occur in phenomena such as avalanches (46), tsunamis (47), chemical waves and diffusion processes (48), and epileptic seizures (49). Further examples are the epidemic transmission of diseases (50), and, more recently, the spreading of memes on social networks (51) or in science (52).

In many cases spatio-temporal recordings can be represented as a two-dimensional plot where for each recording site the occurrence of certain discrete events (often obtained from threshold crossings in continuous data) are indicated by time markers. In neuroscience such a plot is known as a raster plot. A sequence of stereotypical neuronal action potentials (*spikes*, (53)) is a *spike train* and a set of spike trains exhibiting perfectly consistent repetitions of the same global propagation pattern is called a *synfire pattern*. In this paper we adapt this terminology and use all of these expressions not only in the context of neuronal spikes but also for any other kind of discrete events. However, note that our use of the term 'synfire pattern' differs slightly from the literature (see e.g. (54)). Here we define a synfire pattern as a sequence of global events in which all neurons fire in consistent order and the interval between successive events is at least twice as large as the propagation time within an event. An example of a rasterplot with spike trains forming a perfect synfire pattern is shown in Fig. 3.1a.

For any spike train set exhibiting propagation patterns the questions arise naturally whether these patterns show any consistency, i.e., to what extent do the spike trains resemble a synfire pattern, are there spike trains that consistently lead global events and are there other spike trains that invariably follow these leaders?

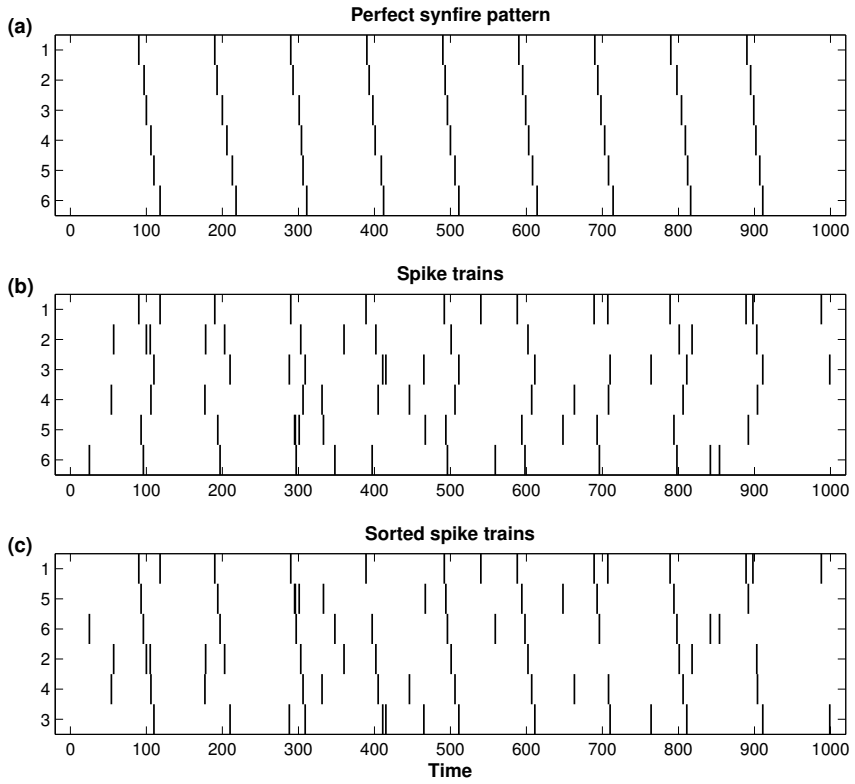

Figure 3.1: Motivation for SPIKE-Order and Spike Train Order. (a) Perfect synfire pattern. (b) Unsorted set of spike trains. (c) The same spike trains as in B but now sorted from leader to follower.

In this study we introduce a framework consisting of two directional measures (*SPIKE-Order* and *Spike Train Order*) that allows to define a value termed *Synfire Indicator* which quantifies the consistency of the leader-follower relationships in a rigorous and automated manner. This Synfire Indicator attains its maximal value of 1 for a perfect synfire pattern in which all neurons fire repeatedly in a consistent order from leader to follower (Fig. 3.1a).

The same framework also allows to sort multiple spike trains from leader to follower, as illustrated in Figs. 3.1b and 3.1c. This is meant purely in the sense of temporal sequence. Whereas Fig. 3.1b shows an artificially created but rather realistic spike train set, in Fig. 3.1c the same spike trains have been sorted to become as close as possible to a synfire pattern. Now the spike trains that tend to fire first are on top whereas spike trains with predominantly trailing spikes are at the bottom.

We demonstrate the new approach using artificially generated datasets before we apply it to analyze the consistency of propagation patterns in real dataset. The neurophysiological dataset consists of neuronal activity recorded from mice brain slices. These recordings typically exhibit a sequence of global events termed Giant Depolarized Potentials (GDPs) and one of the main questions we investigate is whether it is possible to identify neurons that consistently lead these events (potential hub neurons, see (55)).

The remainder of the article is organized as follows: In the Methods (Section 3.2) we first describe the coincidence detection (Section 3.2.1) and the symmetric measure SPIKE-Synchronization (Section 3.2.2). Subsequently, we introduce the new directionality approach consisting of the two measures SPIKE-Order and Spike Train Order (Section 3.2.3) as well as the Synfire Indicator (Section 3.2.4) before we discuss the use of SPIKE-Order surrogates to evaluate the statistical significance of the results in Section 3.2.5. The Results Section 5.5 consists of two Subsections detailing applications of the new approach to artificially generated datasets (Section 3.3.1) and neurophysiological data (Section 3.3.2). Conclusions are drawn in Section 3.4. Finally, both real dataset is described in the Appendix.

3.2 MEASURES

Analyzing leader-follower relationships in a spike train set requires a criterion that determines which spikes should be compared against each other. What is needed is a match maker, a method which pairs spikes in such a way that each spike is matched with at most one spike in each

of the other spike trains. This match maker already exists. It is the adaptive coincidence detection first used as the fundamental ingredient for the bivariate measure *event synchronization* (26, 56).

Event synchronization itself is symmetric and quantifies the overall level of synchrony from the number of quasi-simultaneous appearances of spikes. It was proposed along with an asymmetric measure termed *delay asymmetry* which evaluates the temporal order among matching spikes in the two spike trains.

However, unfortunately both event synchronization and delay asymmetry are defined for the bivariate case of two spike trains only, rely on sampled time profiles, and have a very non-intuitive normalization. For the symmetric variant we have already addressed these issues by proposing *SPIKE-Synchronization* (7), a renormalized multivariate extension of event synchronization.

The two new measures SPIKE-Order and Spike Train Order proposed here improve and extend the asymmetric measure delay asymmetry in the same way. In particular, instead of just quantifying bivariate directionality they open up a completely new application, since they allow us to sort the spike trains according to the typical relative order of their spikes and to quantify the consistency of this order using the Synfire Indicator.

All four approaches (bivariate/multivariate, symmetric/asymmetric) are time-resolved as well as parameter- and scale-free. Their calculation consists of two steps, adaptive coincidence detection followed by a combination of normalization and windowing. The first step, adaptive coincidence detection, is the same for all of these measures.

3.2.1 *Adaptive Coincidence Detection*

Most coincidence detectors rely on a coincidence window of fixed size τ (57, 20). However, since in many cases it is very difficult to judge whether two spikes are coincident or not without taking the local context into account (see Fig. 3.2a for an example), Quiroga et al. proposed a more flexible coincidence detection. This coincidence detection is scale- and thus parameter-free since the minimum time lag $\tau_{ij}^{(1,2)}$ at which two spikes $t_i^{(1)}$ and $t_j^{(2)}$ of spike trains (1) and (2) are no longer considered to be synchronous is adapted to the local firing rates according to

$$\tau_{ij}^{(1,2)} = \tfrac{1}{2} \min \left\{ t_{i+1}^{(1)} - t_i^{(1)}, t_i^{(1)} - t_{i-1}^{(1)}, t_{j+1}^{(2)} - t_j^{(2)}, t_j^{(2)} - t_{j-1}^{(2)} \right\}. \qquad (3.1)$$

For some applications it might be appropriate to additionally introduce a maximum coincidence window τ_{max} as a parameter. This way

additional knowledge about the data (such as typical propagation speed) can be taken into account in order to guarantee that two coincident spikes are really part of the same propagation front.

3.2.2 *SPIKE-Synchronization*

In normalization and windowing SPIKE-Synchronization (7) has evolved so substantially from event synchronization that here we refrain from going into any detail on the original measure, but rather just mention the main improvements. For a thorough introduction to event synchronization please refer to the original paper (26), a more detailed comparison of the two measures can be found in (7).

The main difference is that SPIKE-Synchronization (7) results in a discrete, not a continuous, spike-timing based profile. The coincidence criterion is quantified by means of a coincidence indicator

$$C_i^{(1,2)} = \begin{cases} 1 & \text{if} \min_j \left(|t_i^{(1)} - t_j^{(2)}| \right) < \tau_{ij}^{(1,2)} \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

which assigns to each spike either a one or a zero depending on whether this spike is part of a coincidence or not. Note that here, unlike for event synchronization, the minimum function and the '$<$' guarantee that a spike can at most be coincident with one spike (the nearest one) in the other spike train. In case a spike is right in the middle between two spikes from the other spike train there is no ambiguity since this spike is not coincident with either one of them.

This unambiguity, illustrated in Fig. 3.2b, is the essential property which allows the adaptive coincidence detection to act as a match-maker for the subsequent application of SPIKE-Synchronization. Fig. 3.2c shows examples, one with two coincident and one with two non-coincident spikes.

A multivariate version of SPIKE-Synchronization can be defined by generalizing the bivariate coincidence detection of Eq. 3.2 to all pairs of spike trains (n, m) with $n, m = 1, ..., N$ and N denoting the number of spike trains:

$$C_i^{(n,m)} = \begin{cases} 1 & \text{if} \min_j \left(|t_i^{(n)} - t_j^{(m)}| \right) < \tau_{ij}^{(n,m)} \\ 0 & \text{otherwise.} \end{cases} \tag{3.3}$$

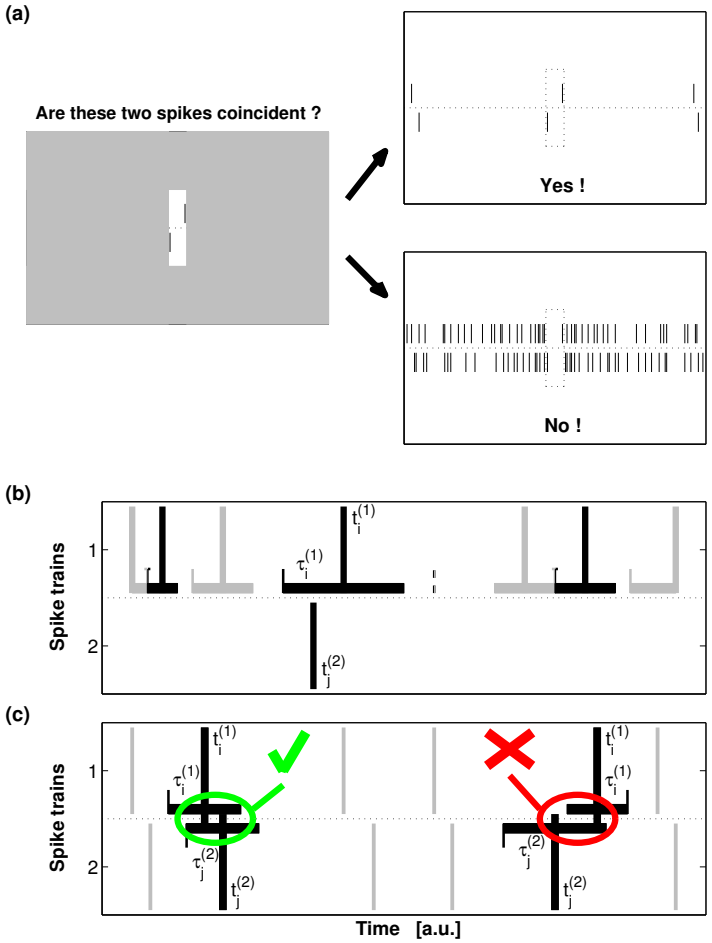

Figure 3.2: (a) This example demonstrates the usefulness of an adaptive coincidence detection. Depending on context the same two spikes (left) can appear as coincident (right, top) or as non-coincident (right, bottom). (b) Illustration of the adaptive coincidence detection. For clarity spikes and their coincidence windows are shown alternatively in bright and dark color. The first step assigns to each spike $t_i^{(1)}$ of the first spike train a potential coincidence window which does not overlap with any other coincidence window: $\tau_i^{(1)} = \min\{t_{i+1}^{(1)} - t_i^{(1)}, t_i^{(1)} - t_{i-1}^{(1)}\}/2$. Thus any spike from the second spike train can at most be coincident with one spike from the first spike train. Small vertical lines mark the times right in the middle between two spikes, and a line is dashed when it does not mark the edge of a coincidence window. (c) In the same way a coincidence window $\tau_j^{(2)} = \min\{t_{j+1}^{(2)} - t_j^{(2)}, t_j^{(2)} - t_{j-1}^{(2)}\}/2$ is defined for spike $t_j^{(2)}$ from the second spike train. For two spikes to be coincident they both have to lie in each other's coincidence window which means that their absolute time difference has to be smaller than $\tau_{ij} = \min\{\tau_i^{(1)}, \tau_j^{(2)}\}$ (which is equivalent to the shorter definition found in Eq. 3.1). For the two spikes $t_i^{(1)}$ and $t_j^{(2)}$ on the left side this is the case, whereas the spikes on the right side are not coincident.

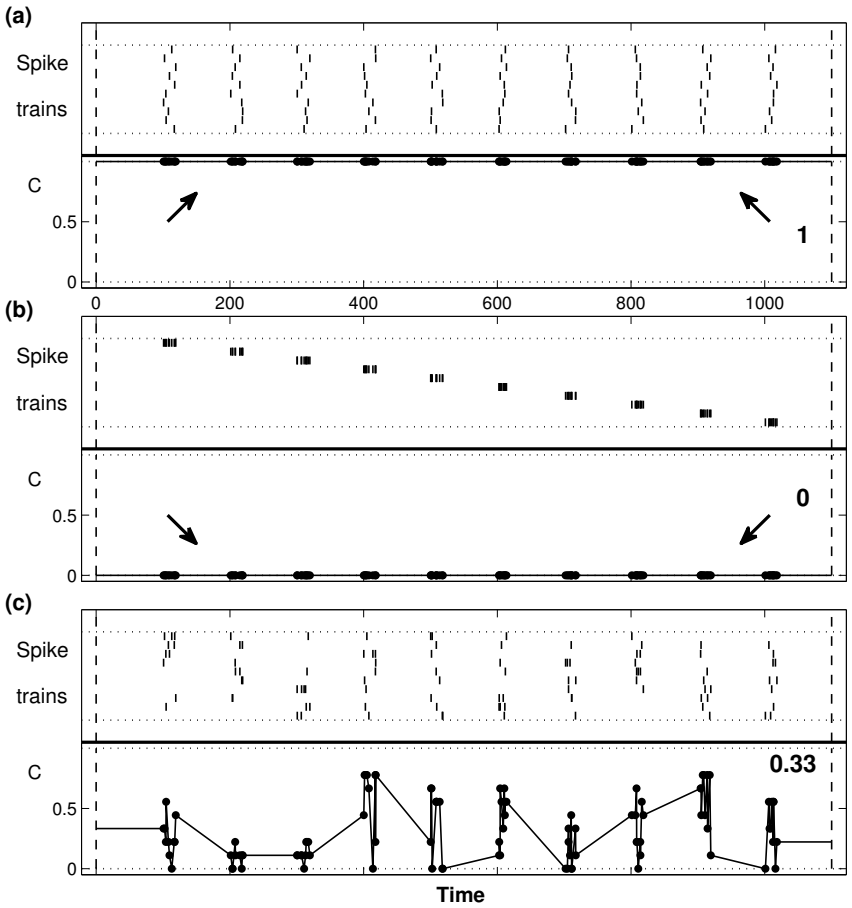

Figure 3.3: SPIKE-Synchronization. Note that the profile $C(t_k)$ is defined only at the times of the spikes but a better visualization is achieved by connecting the individual dots. By construction the pooled spike train of these examples is identical consisting of 10 evenly spaced bursts. The only difference is the distribution of the spikes among the individual spike trains which varies from maximum to minimum via intermediate synchrony. SPIKE-Synchronization correctly indicates these changes. (a) Maximum reliability results in the value one over the whole time interval. Each spike train contains one spike per firing event. (b) Synfire pattern of bursts resulting in minimum reliability corresponding to the value zero for the whole time interval. (c) A random distribution of spikes among spike trains yields intermediate values.

Here $\tau_{ij}^{(n,m)}$ is defined equivalent to Eq. 3.1. Subsequently, for each spike of every spike train a normalized coincidence counter

$$C_i^{(n)} = \frac{1}{N-1} \sum_{m \neq n} C_i^{(n,m)} \tag{3.4}$$

is obtained by averaging over all $N-1$ bivariate coincidence indicators involving the spike train n.

In order to obtain a single multivariate similarity profile we pool the spikes of all the spike trains as well as their coincidence counters:

$$\{C_k\} = \bigcup_n \left\{ C_{i(k)}^{(n(k))} \right\}, \tag{3.5}$$

where we map the spike train indices n and the spike indices i into a global spike index k denoted by the mapping $i(k)$ and $n(k)$.

Note that in case there exist perfectly coincident spikes, k counts over all of these spikes. From this discrete set of coincidence counters C_k the SPIKE-Synchronization profile $C(t_k)$ is obtained via $C(t_k) = C_k$. Finally, SPIKE-Synchronization is defined as the average value of this profile

$$S_C = \begin{cases} \frac{1}{M} \sum_{k=1}^{M} C(t_k) & \text{if } M > 0 \\ 1 & \text{otherwise} \end{cases} \tag{3.6}$$

with $M = \sum_{n=1}^{N} M^{(n)}$ denoting the total number of spikes in the pooled spike train.

This way we have used the same consistent framework for both the bivariate and the multivariate case. The former is just a special case of the latter. The interpretation is very intuitive: SPIKE-Synchronization quantifies the overall fraction of coincidences. It reaches one if and only if each spike in every spike train has one matching spike in all the other spike trains (or if there are no spikes at all), and it attains the value zero if and only if the spike trains do not contain any coincidences. Examples for both of these extreme cases can be found in Fig. 3.3a and 3.3b and one intermediate example (random distribution of spikes among spike trains) is shown in Fig. 3.3c. For a derivation of the expectation value for Poisson spike trains please refer to (58).

In the multivariate analysis proposed in this paper, SPIKE-Synchronization can be used to filter the input to the algorithm. In order to focus on propagation patterns within truly global events it is possible to set a threshold value C_{thr} for the SPIKE-Synchronization profile $C(t_k)$. This way only spikes with a coincidence value higher than this parameter C_{thr} are taken into account, all the other noisy background spikes are simply ignored. This kind of filter will be used in the analysis of the neurophysiological datasets in Section 3.3.2.

3.2.3 *SPIKE-Order and Spike Train Order*

SPIKE-Synchronization assigns to each spike of a given spike train pair a bivariate coincidence indicator. These coincidence indicators $C_i^{(n,m)}$, which are either 0 or 1, are then averaged over spike train pairs and converted into one overall profile $C(t_k)$ normalized between 0 and 1. In exactly the same manner SPIKE-Order and Spike Train Order assign bivariate order indicators to spikes. Also these two order indicators, the asymmetric $D_i^{(n,m)}$ and the symmetric $E_i^{(n,m)}$, which both can take the values -1, 0, or $+1$, are averaged over spike train pairs and converted into two overall profiles $D(t_k)$ and $E(t_k)$ which are normalized between -1 and 1. The SPIKE-Order profile $D(t_k)$ distinguishes leading and following spikes, whereas the Spike Train Order profile $E(t_k)$ provides information about the order of spike trains, i.e. it allows to sort spike trains from leaders to followers.

First of all, similar to the transition from the symmetric event synchronization to delay asymmetry, the symmetric coincidence indicator $C_i^{(n,m)}$ of SPIKE-Synchronization (Eq. 3.3) is replaced by the asymmetric SPIKE-Order indicator

$$D_i^{(n,m)} = C_i^{(n,m)} \cdot \operatorname{sign}\left(t_{j'}^{(m)} - t_i^{(n)}\right), \tag{3.7}$$

where the index j' is defined from the minimum in Eq. 3.2 as $j' = \arg\min_j(|t_i^{(1)} - t_j^{(2)}|)$.

The corresponding value $D_{j'}^{(m,n)}$ is obtained in an antisymmetric manner as

$$D_{j'}^{(m,n)} = C_{j'}^{(m,n)} \cdot \operatorname{sign}\left(t_i^{(n)} - t_{j'}^{(m)}\right) = -D_i^{(n,m)}. \tag{3.8}$$

Therefore, this indicator assigns to each spike either a 1 or a -1 depending on whether the respective spike is leading or following a coincident spike from the other spike train. The value 0 is obtained for cases in which there is no coincident spike in the other spike train ($C_i^{(n,m)} = 0$), but also in cases in which the times of the two coincident spikes are absolutely identical ($t_{j'}^{(m)} = t_i^{(n)}$).

The multivariate profile $D(t_k)$ obtained analogously to Eq. 3.5 is normalized between 1 and -1 and the extreme values are obtained if a spike is either leading ($+1$) or following (-1) coincident spikes in all other spike trains. It can be 0 either if a spike is not part of any coincidences or if it leads exactly as many spikes from other spike trains in coincidences as it follows. From the definition in Eqs. 3.7 and 3.8 it follows immediately that C_k is an upper bound for the absolute value $|D_k|$.

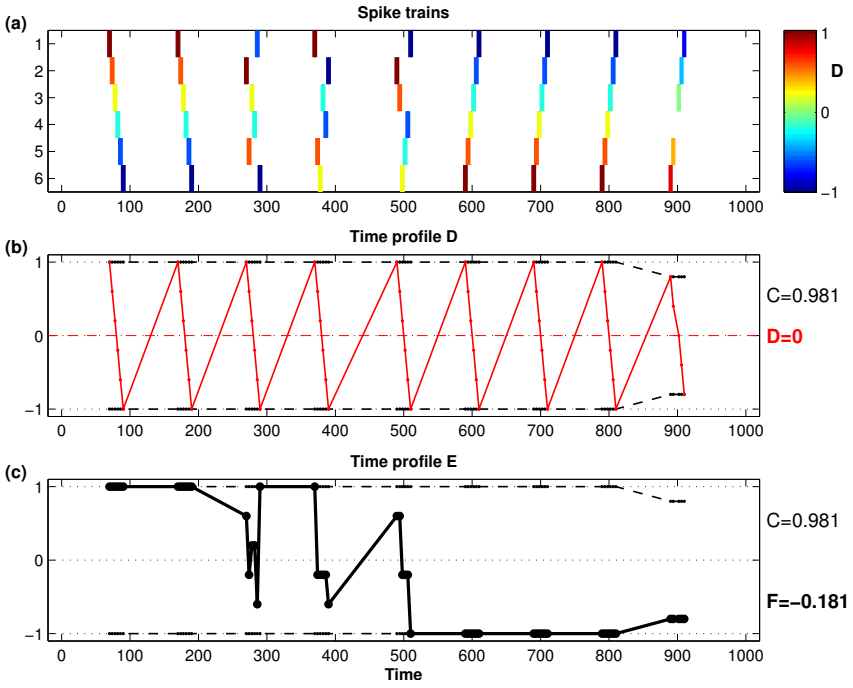

Figure 3.4: SPIKE-Order profile $D(t_k)$ and Spike Train Order profile $E(t_k)$ for an artificially created example dataset. (a) The rasterplot shows 6 spike trains which emit spikes in nine reliable events. For the first two events spikes fire in order, for the next three events the order is random whereas for the last four events the order is inverted. In the last event there is one spike missing. Spike thickness decodes the SPIKE-Synchronization value $C(t_k)$ (here almost constant), spike color the SPIKE-Order value $D(t_k)$. (b,c) The SPIKE-Synchronization profile $C(t_k)$ and its mirror profile (dashed black lines) act as envelope for both the SPIKE-Order profile $D(t_k)$ (b, red) and the Spike Train Order profile $E(t_k)$ (c, black). (b) The SPIKE-Order profile can not distinguish events with different firing order and by construction the average value is always $D = 0$. (c) On the other hand, in the Spike Train Order profile events with different firing order can clearly be distinguished. For the first two correctly ordered events the value 1 is obtained. The next three events exhibit random order and correspondingly the profile fluctuates rather wildly. Finally, the last four inversely ordered yield the value -1 except for the last event for which the absolute minimum value can not be obtained since one spike is missing. The average value, the Synfire Indicator F, is not 0 but negative which reflects the dominance of the inversely ordered events.

While the SPIKE-Order profile can be very useful for color-coding and visualizing local spike leaders and followers (Fig. 3.4a), it is not useful as an overall indicator of Spike Train Order (Fig. 3.4b). The profile is invariant under exchange of spike trains, i.e. it looks the same for all events no matter what the order of the firing is (in our example only the last event looks slightly different since one spike is missing). Moreover, summing over all profile values, which is equivalent to summing over all coincidences, necessarily leads to an average value of 0, since for every leading spike (+1) there has to be a following spike (−1).

So in order to quantify any kind of leader-follower information between spike trains we need a second kind of order indicator. The Spike Train Order indicator is similar to the SPIKE-Order indicator defined in Eqs. 3.7 and 3.8 but with two important differences. Both spikes are assigned the same value and this value now depends on the order of the spike trains:

$$E_i^{(n,m)} = C_i^{(n,m)} \cdot \begin{cases} \operatorname{sign}\left(t_{j'}^{(m)} - t_i^{(n)}\right) & \text{if} \quad n < m \\ \operatorname{sign}\left(t_i^{(n)} - t_{j'}^{(m)}\right) & \text{if} \quad n > m \end{cases} \tag{3.9}$$

and

$$E_{j'}^{(m,n)} = E_i^{(n,m)}. \tag{3.10}$$

This symmetric indicator assigns to both spikes a +1 in case the two spikes are in the correct order, i.e. the spike from the spike train with the lower spike train index is leading the coincidence, and a −1 in the opposite case. Once more the value 0 is obtained when there is no coincident spike in the other spike train or when the two coincident spikes are absolutely identical.

The multivariate profile $E(t_k)$, again obtained similarly to Eq. 3.5, is also normalized between 1 and −1 and the extreme values are obtained for a coincident event covering all spike trains with all spikes emitted in the order from first (last) to last (first) spike train, respectively (see the first two and the last four events in Fig. 3.4). It can be 0 either if a spike is not a part of any coincidences or if the order is such that correctly and incorrectly ordered spike train pairs cancel each other. Again, C_k is an upper bound for the absolute value of E_k.

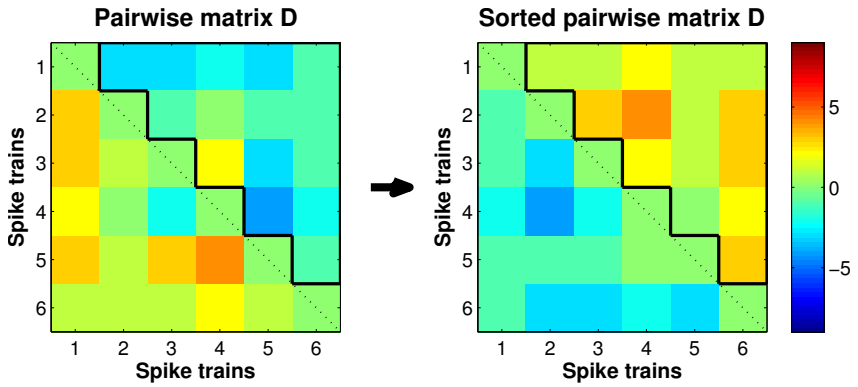

Figure 3.5: Pairwise cumulative SPIKE-Order matrix D before (left) and after (right) sorting for the example dataset from Fig. 3.4. The upper triangular matrix $D^{(n<m)}$, marked in black, is used to calculate the Synfire Indicator F, for both the unsorted spike trains (F_u, left) and the sorted spike trains (F_s, right). The thick black arrow in between the two matrices indicates the sorting process.

3.2.4 *Synfire Indicator*

In contrast to the SPIKE-Order profile D_k, for the Spike Train Order profile E_k it does make sense to define an average value, which we term the Synfire Indicator:

$$F = \frac{1}{M} \sum_{k=1}^{M} E(t_k). \tag{3.11}$$

The interpretation is very intuitive. The Synfire Indicator F quantifies to what degree the spike trains in their current order resemble a perfect synfire pattern. It is normalized between 1 and -1 and attains the value 1 (-1) if the spike trains in their current order form a perfect (inverse) synfire pattern. This means that all spikes are coincident with spikes in all other spike trains and that all orders from leading (following) to following (leading) spike consistently reflect the order of the spike trains.

It is 0 either if the spike trains do not contain any coincidences at all or if among all spike trains there is a complete symmetry between leading and following spikes.

The Spike Train Order profile $E(t_k)$ for our example is shown in Fig. 3.4c. In this case the order of spikes within an event clearly matters. The Synfire Indicator F is slightly negative indicating that the current order of the spike trains is actually closer to an inverse synfire pattern.

Given a set of spike trains we now would like to sort the spike trains from leader to follower such that the set comes as close as possible to a synfire pattern. To do so we have to maximize the overall number of

correctly ordered coincidences and this is equivalent to maximizing the Synfire Indicator F. However, it would be very difficult to achieve this maximization by means of the multivariate profile $E(t_k)$. Clearly, it is more efficient to sort the spike trains based on a pairwise analysis of the spike trains. The most intuitive way is to use the anti-symmetric cumulative SPIKE-Order matrix

$$D^{(n,m)} = \sum_i D_i^{(n,m)} \tag{3.12}$$

which sums up orders of coincidences from the respective pair of spike trains only and quantifies how much spike train n is leading spike train m (Fig. 3.5).

Hence if $D^{(n,m)} > 0$ spike train n is leading m, while $D^{(n,m)} < 0$ means m is leading n. If the current Spike Train Order is consistent with the synfire property, we thus expect that $D^{(n,m)} > 0$ for $n < m$ and $D^{(n,m)} < 0$ for $n > m$. Therefore, we construct the overall SPIKE-Order as

$$D_< = \sum_{n<m} D^{(n,m)}, \tag{3.13}$$

i.e. the sum over the upper right tridiagonal part of the matrix $D^{(n,m)}$.

After normalizing by the overall number of possible coincidences, we arrive at a second more practical definition of the Synfire Indicator:

$$F = \frac{2D_<}{(N-1)M}. \tag{3.14}$$

The value is identical to the one of Eq. 3.11, only the temporal and the spatial summation of coincidences (i.e., over the profile and over spike train pairs) are performed in the opposite order.

Having such a quantification depending on the order of spike trains, we can introduce a new ordering in terms of the spike train index permutation $\varphi(n)$. The overall Synfire Indicator for this permutation is then denoted as F_φ. Accordingly, for the initial (**u**nsorted) order of spike trains φ_u the Synfire Indicator is denoted as $F_u = F_{\varphi_u}$.

The aim of the analysis is now to find the optimal (**s**orted) order φ_s as the one resulting in the maximal overall Synfire Indicator $F_s = F_{\varphi_s}$:

$$\varphi_s : F_{\varphi_s} = \max_\varphi\{F_\varphi\} = F_s. \tag{3.15}$$

This Synfire Indicator for the sorted spike trains quantifies how close spike trains can be sorted to resemble a synfire pattern, i.e., to what extent coinciding spike pairs with correct order prevail over coinciding spike pairs with incorrect order. Unlike the Synfire Indicator for the

unsorted spike trains F_u, the optimized Synfire Indicator F_s can only attain values between 0 and 1 (any order that yields a negative result could simply be reversed in order to obtain the same positive value). For a perfect synfire pattern we obtain $F_s = 1$, while sufficiently long Poisson spike trains without any synfire structure yield $F_s \approx 0$.

The complexity of the problem to find the optimal Spike Train Order is similar to the well-known travelling salesman problem (59). For N spike trains there are $N!$ permutations φ, so for large numbers of spike trains finding the optimal Spike Train Order φ_s is a non-trivial problem and brute-force methods such as calculating the F_φ-value for all possible permutations are not feasible. Instead, one has to make use of methods such as parallel tempering (60) or simulated annealing (61) to search for the optimal order. Here we choose simulated annealing, a probabilistic technique which approximates the global optimum of a given function in a large search space. In our case this function is the Synfire Indicator F_φ (which we would like to maximize) and the search space is the permutation space of all spike trains. We start with the F_u-value from the unsorted permutation and then visit nearby permutations using the fundamental move of exchanging two neighboring spike trains within the current permutation. The update of the Synfire Indicator when exchanging the spike trains k and $k+1$ is simply given by $\Delta F = -2D^{(k,k+1)}$. All moves with positive ΔF are accepted while the likelihood of accepting moves with negative ΔF is decreased along the way according to a standard slow cooling scheme. The procedure is repeated iteratively until the order of the spike trains no longer changes or until a predefined end temperature is reached.

In Fig. 3.6 we show the complete SPIKE-Order analysis including the results for the sorted spike trains. The sorting of the spike trains maximizes the Synfire Indicator as reflected by both the normalized sum of the upper right half of the pairwise cumulative SPIKE-Order matrix (Eq.3.14, Fig. 3.6c) and the average value of the Spike Train Order profile $E(t_k)$ (Eq.3.11, Fig. 3.6d). Finally, the sorted spike trains in Fig. 3.6e are now ordered such that the first spike trains have predominantly high values (red) and the last spike trains predominantly low values (blue) of $D(t_k)$.

The complete analysis returns results consisting of several levels of information. Time-resolved (local) information is represented in the spike-coloring and in the profiles D and E. The pairwise information in the SPIKE-Order matrix reflects the leader-follower relationship between two spike trains at a time. The Synfire Indicator F characterizes the closeness of the dataset as a whole to a synfire pattern, both for the unsorted (F_u) and for the sorted (F_s) spike trains. Finally, the sorted order

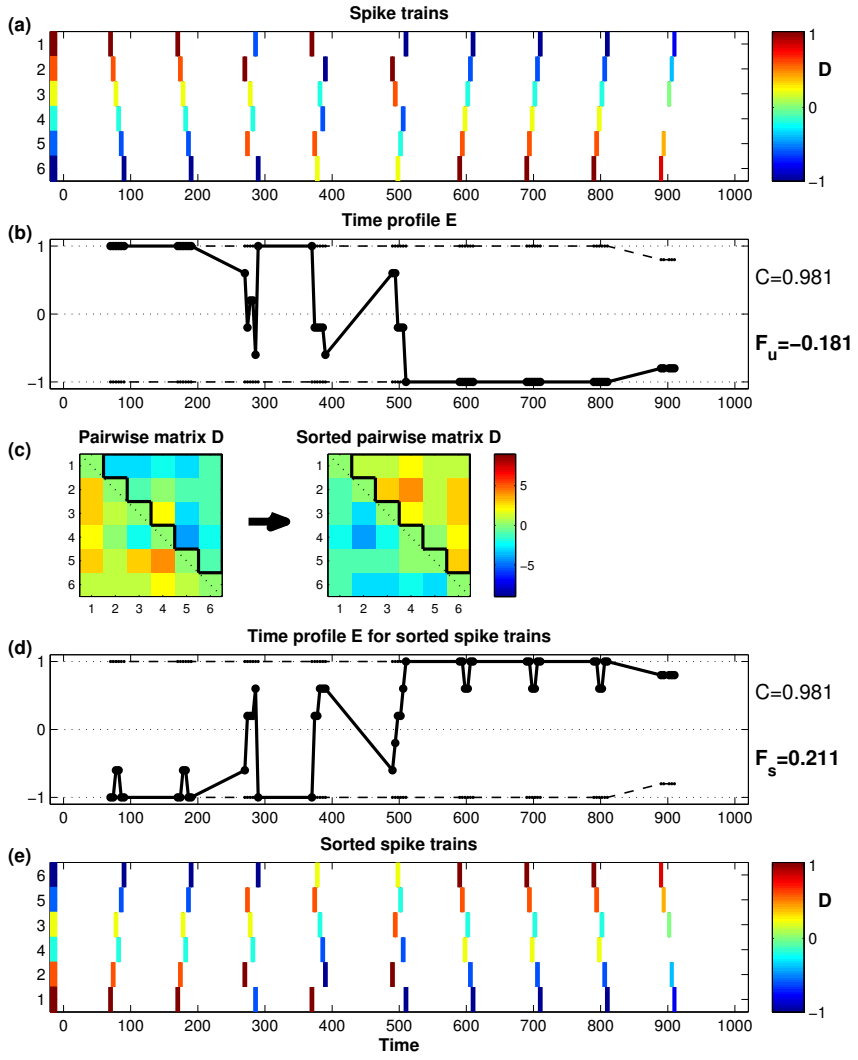

Figure 3.6: Complete illustration of SPIKE-Order using our example dataset from Fig. 3.4. (a) Unsorted spike trains with the spikes color-coded according to the value of the SPIKE-Order $D(t_k)$. (b) Spike Train Order profile $E(t_k)$. The Synfire Indicator F_u for the unsorted spike trains is slightly negative. (c) Pairwise SPIKE-Order matrix D before and after sorting. The optimal order maximizes the upper triangular matrix. (d) Spike Train Order profile $E(t_k)$ and its average values, the Synfire Indicator F_s for the sorted spike trains. (e) Sorted spike trains.

of the spike trains is a very important result in itself since it identifies the leading and the following spike trains.

3.2.5 *Statistical significance*

As a last step in the analysis we evaluate the statistical significance of the optimized Synfire Indicator F_s. What we would like to estimate is the likelihood that for the given total number of coincidences the prevalence of correctly ordered spike pairs (as quantified by the optimized Synfire Indicator) could have been obtained by chance. If all coincident spike pairs would be independent, the probability distribution would be strictly binomial and we could calculate this likelihood analytically. However, the pairwise spike orders in coincident events involving multiple spike trains are not independent from each other, and so instead we estimate the likelihood numerically using a set of carefully constructed spike order surrogates.

For each surrogate (Fig. 3.7a) we maintain the coincidence structure of the original spike trains by preserving the SPIKE-Synchronization values of every individual spike. However, we destroy the spike order patterns by swapping the order of the two spikes in a sufficient number of randomly selected coincident spike pairs. Note that the generation of surrogates takes place not on the level of spike times but on the level of order values (the x-axis in Fig. 3.7a is labeled 'time index', not 'time'). Spike trains with swapped spike times would have different interspike intervals, and this would alter the results of the coincidence criterion in Eq. 3.1 and change the value of SPIKE-Synchronization. This in turn would make the desired evaluation of pure spike order effects difficult.

In the implementation, from one spike order surrogate to the next the number of spike order swaps is set to the number of coincident spikes in the spike train set, such that all possible spike order patterns can be reached. Only for the first surrogate, since it starts from the original spike trains, we swap twice as many coincidences in order to account for transients. After each swap we take extra care that all other spike orders that are affected by the swap are updated as well. For example, if a swap changes the order between the first and the third spike in an ordered sequence of three spikes, we also swap both the order between the first and the second as well as the order between the second and the third spike.

For each spike train surrogate we repeat exactly the same optimization procedure in the spike train permutation space that is done for the original dataset. The original Synfire Indicator is deemed significant if

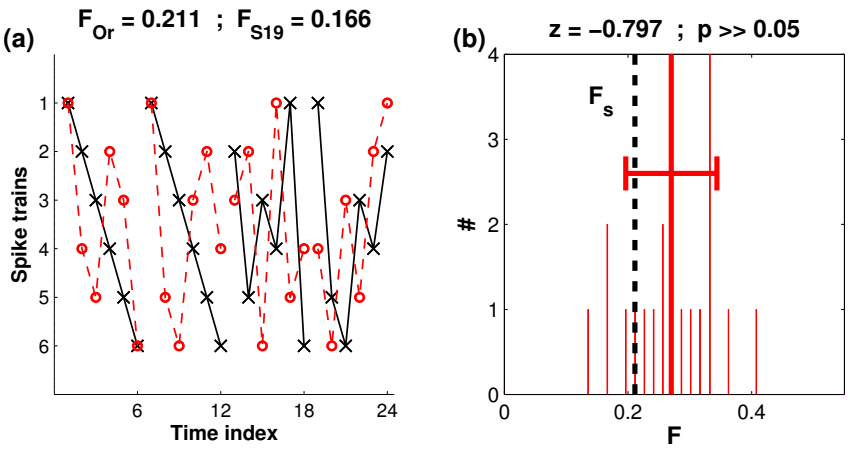

Figure 3.7: Statistical significance: Surrogate analysis for the example dataset from Fig. 3.4. (a) Spike order patterns for original (black) and one randomized surrogate (red). For clarity only the first four events are shown. For the first two events the synfire-order of the original is destroyed in the surrogates whereas for the next two events both sequences are equally unordered. (b) Histogram for 19 surrogates. Thick lines denote mean and standard deviation. Since the value for the original dataset (black) is not maximum, the optimally sorted spike trains do not exhibit a statistically significant synfire pattern.

it is higher than the Synfire Indicator obtained for all of the surrogate datasets (this case will be marked by two asterisks). Here we use $s = 19$ surrogates for a significance level of $p^* = 1/(s+1) = 0.05$. As a second indicator we state the z-score, e.g., the deviation of the original value x from the mean μ of the surrogates in units of their standard deviation σ:

$$z = \frac{x - \mu}{\sigma}. \tag{3.16}$$

Results of the significance analysis for our standard example are shown in the histogram in Fig. 3.7b. In this case the absolute value of the z-score is smaller than one and the p-value is larger than p^* and the result is thus judged as statistically non-significant.

In case the initial sorting of the spike trains is used to test a specific hypothesis there also exists a straightforward procedure to test the statistical significance of the Synfire Indicator F_u for the unsorted spike trains. In this case no optimization of the Synfire Indicator is required, rather the Synfire Indicator F_u for the initial sorting is compared against Synfire Indicators obtained for random permutations of the spike trains. This kind of significance test will be used in Section 3.3.2.

3.3 RESULTS

In the following we apply our new algorithm to artificially generated datasets (Section 3.3.1)and neurophysiological data (Section 3.3.2). For the Figures we use the same full layout introduced in Fig. 3.6 to which we add the significance analysis of Fig. 3.7b.

3.3.1 *Application to artificially generated data*

We start with examples covering the two extreme cases of a perfect synfire pattern and a completely random spike train set. First, in Fig. 3.8 we apply the algorithm to a perfect inverse synfire pattern for which the spike trains are initially sorted from follower to leader. Therefore, the Synfire Indicator of the unsorted spike trains yields its minimum value of $F_u = -1$. Sorting just reverses the order of the spike trains and in consequence the maximum value of $F_s = 1$ is obtained. Any shuffling of spike orders necessarily destroys the synfire pattern and thus leads to much lower values of the Synfire Indicator. Accordingly, the surrogate test (Fig. 3.8f) shows that the statistical significance of the original Synfire Indicator is very high.

The other extreme case is Poisson spike trains (Fig. 3.9) for which the arrival times of spikes are completely random and without any preferred order. For this realization the Synfire Indicator F_u for the unsorted spike trains happens to be slightly negative indicating that the spike trains are closer to an inverse synfire pattern than to a synfire pattern. The absolute value F_s after sorting is higher. The fact that both of these values are non-zero is due to the finite size effect caused by the limited number of spikes. For more and more spike trains and/or spikes the expectation value even for the sorted case would converge towards zero. As expected, the surrogate test shows that the order for the original spike trains is not statistically distinct from the order of the surrogate spike trains (there is no preferred order that can be destroyed by the shuffling) and, accordingly, the value of the original Synfire Indicator is revealed to be clearly non-significant.

The third example in Fig. 3.10 shows a mixture of these two extremes, Poisson spike train interspersed with spike trains that contain a perfect inverse synfire pattern (plus random spikes). Sorting the spike trains restores the correct order of the synfire pattern spike trains within the Poisson spike train. The Synfire Indicator for the sorted spike trains F_s for this mixed example is actually almost identical to the value obtained for the Poisson spike trains in Fig. 3.9, but this time the surrogate test

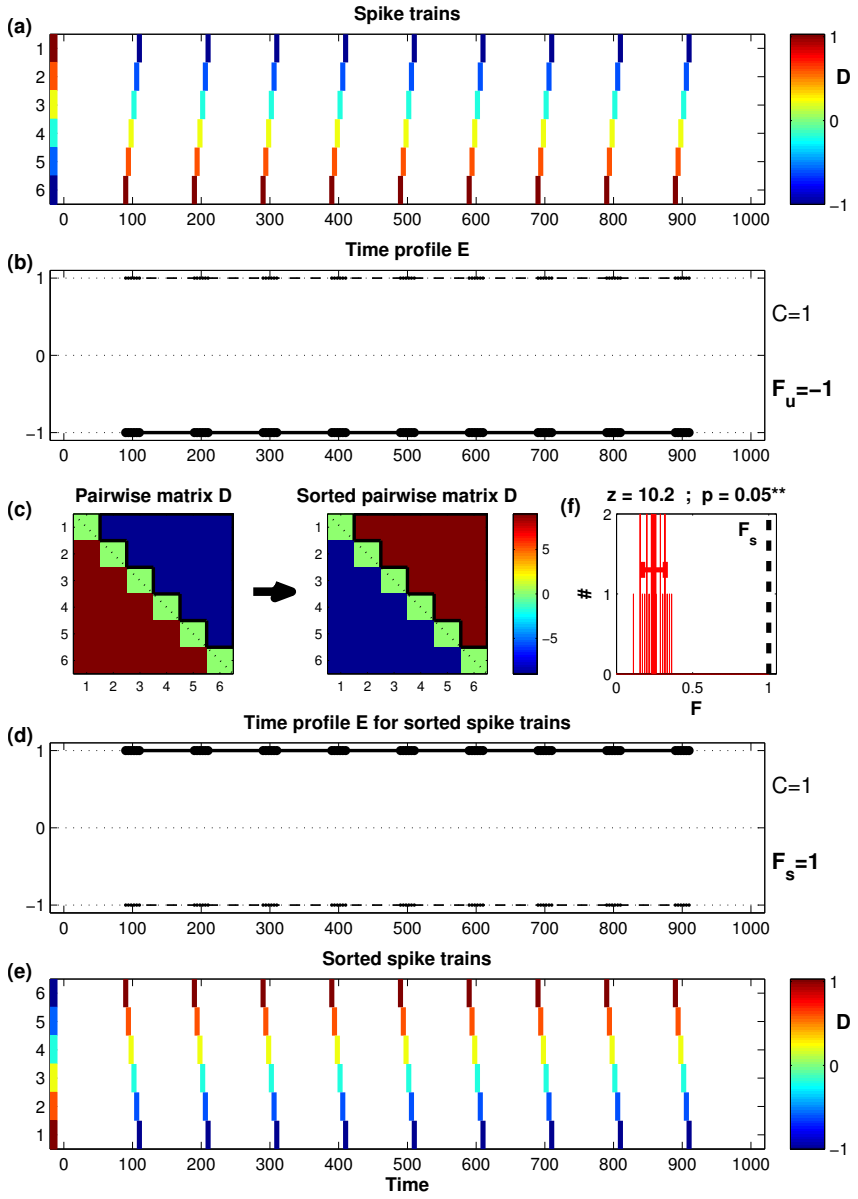

Figure 3.8: SPIKE- and Spike Train Order analysis for a perfect inverse synfire pattern. The plot follows the layout of Fig. 3.6 with the histogram of the surrogate test (see Fig. 3.7b) for statistical significance added as subplot f. For the unsorted spike trains a minimal Synfire Indicator of $F_u = -1$ is obtained, while sorting results in the maximum value of $F_s = 1$. According to the surrogate test the statistical significance of the result is very high.

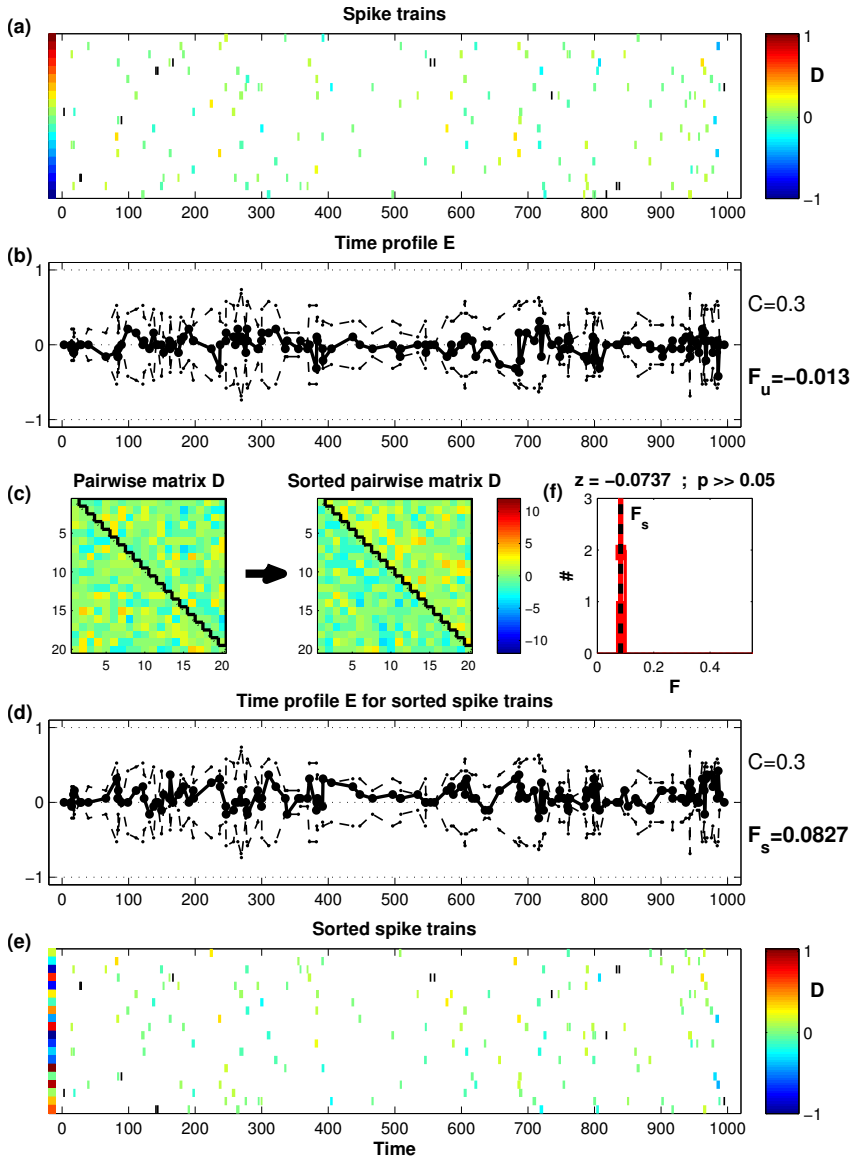

Figure 3.9: SPIKE- and Spike Train Order analysis for 20 Poisson spike trains. Since the number of spike trains is too large to label the spike trains in the top and in the bottom subplot with numbers we use color coding at the left side to label them. Both before and after sorting the Synfire Indicator is very close to zero. The surrogate analysis reveals the result to be non-significant.

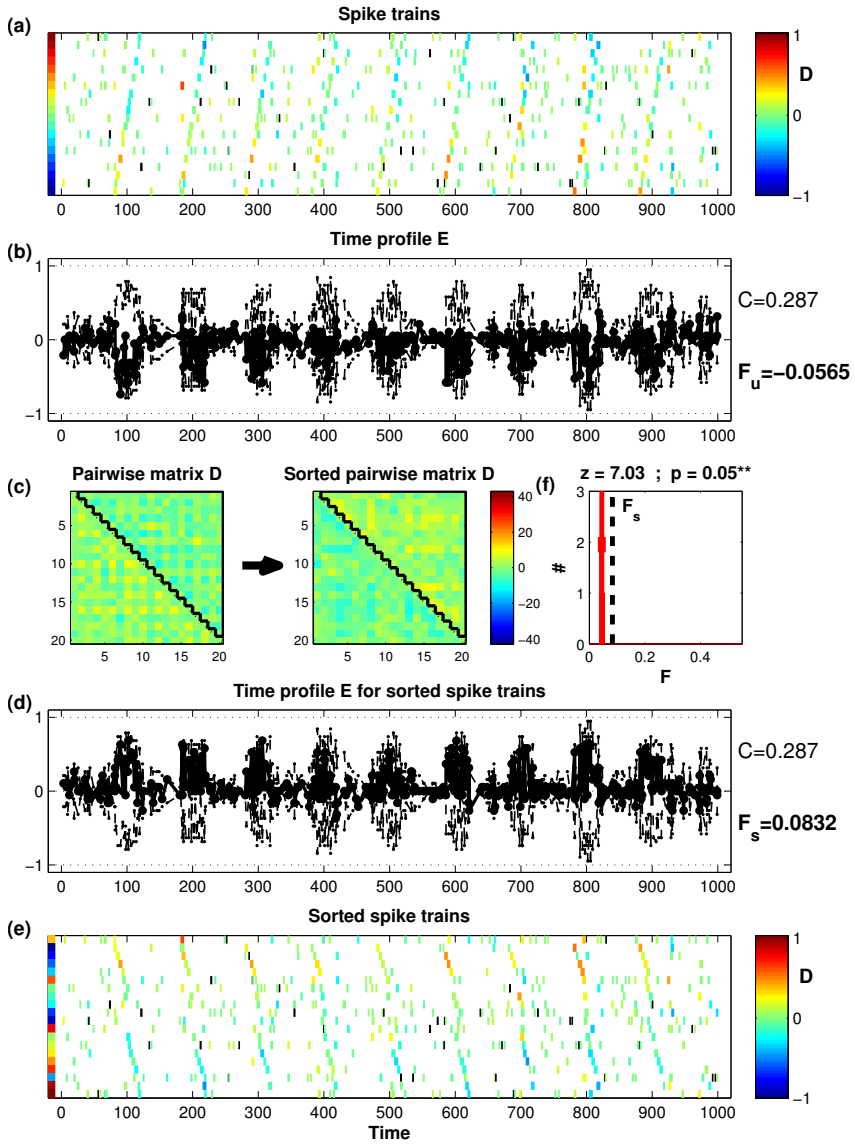

Figure 3.10: SPIKE- and Spike Train Order analysis for Poisson spike train interspersed with spike trains that contain random spikes but also a perfect inverse synfire pattern. The order contained within the synfire pattern spike train is distinct enough to make the Synfire Indicator for the sorted spike trains statistically significant.

reveals the value to be highly significant. These two examples combined illustrate nicely that the Synfire Indicator and the surrogate analysis provide complementary information. In the mixture example of Fig. 3.10 there are many more random Poisson spikes than ordered synfire pattern spikes. According to the Synfire Indicator, these two types of spikes together appear to be as ordered as the spikes of the shorter but purely random Poisson spike trains in Fig. 3.9. However, the Synfire Indicator is strongly influenced by the statistics of the dataset and thus is in itself not sufficient to reliably compare two datasets with widely different number of spike trains and spikes. The surrogate analysis, on the other hand, can be used to compare datasets of different size since by preserving the spike numbers in the surrogates it explicitly takes the statistics of each dataset into account.

3.3.2 *Application to real data*

In order to apply the Spike Train Order algorithm to real neurophysiological data, we analyzed data recorded via fast multicellular calcium imaging in acute CA3 hippocampal brain slices from juvenile mice. In the juvenile hippocampus, the CA3 region is the origin of a stereotypical network phenomenon of wavelike propagating activity termed giant depolarizing potentials (GDPs (62)). In previous studies, GDPs have been used to investigate the topology of networks and the role of hub cells (55) as well as to reveal the deterministic and stochastic processes underlying spontaneous, synchronous network bursts (63). Due to the distinct architecture and the repetitive nature of the GDPs this experimental setup offers a very suitable test case for our synfire pattern analysis (for more background and a detailed description of the experimental methods refer to Appendix 3.5.1).

The first dataset analyzed in Fig. 3.11 includes 13 GDPs over a bit more than 6 minutes. Almost all GDPs involve the whole network. Here as for all other neurophysiological datasets analyzed initially the spike trains are sorted according to their firing rate such that the sparsely spiking neurons are on top and the most active neurons at the bottom (Fig. 3.11a). This specific sorting allows us to test the hypothesis that the neurons which fire almost exclusively within the GDPs and are very sparse on background activity might have a stronger role in initiating GDPs and tend to lead, whereas the more regularly spiking neurons might tend to follow. If this would be the case one would expect a very high value for the initial Synfire Indicator F_u. However, according to Fig. 3.11b the actual value is very close to zero and actually slightly negative. A sta-

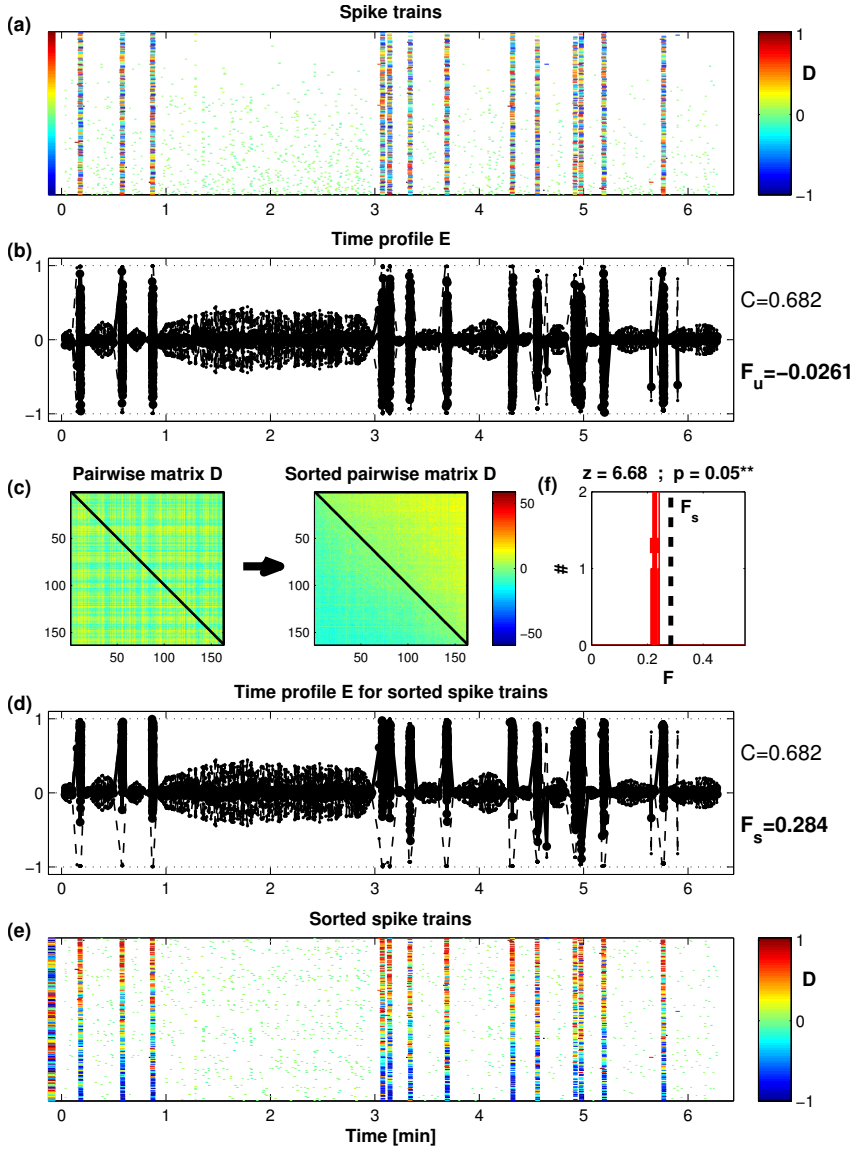

Figure 3.11: SPIKE-Order for real data recorded in an acute hippocampal slice from a juvenile mouse. Note how the color-coding of the spikes according to their SPIKE-Order D helps to overcome the low temporal resolution of the Figure and to resolve the spike order within the GDPs. (a) Initially the spike trains are sorted according to their firing rate starting with the most sparse spike trains. The messy color-patterns reveal that this is completely uncorrelated to the spike order within the GDPs. (f) After sorting, there is a fairly consistent transition from spike trains with predominantly leading spikes (red) in the GDPs to spike trains with predominantly following spikes (blue).

tistical significance test using random permutations of spike trains (see Section 3.2.5) indeed proves the Synfire Indicator of the unsorted spike trains F_u to be non-significant (result not shown). A further indicator for this is the fact that the order of the sorted spike trains is very different from the initial order, as can be seen by comparing the color bars on the left of Fig. 3.11a and Fig. 3.11e. The color-coding of the GDPs exhibits typically a slightly noisy transition from leader (red) to follower (blue). The Synfire Indicator for the sorted spike trains F_s is also much higher (Fig. 3.11d). Finally, the surrogate analysis (Fig. 3.11f) shows this result to be highly significant.

However, the spiking in Fig. 3.11 consists not only of the GDPs. Most neurons exhibit at least to some extent spontaneous background activity, the ones at the top of the initial sorting less than the ones at the bottom. The spikes in this background activity are typically coincident with only few other spikes and do not take part in any propagation patterns (note their green color which indicates SPIKE-Order values close to zero). So in the context of our synfire pattern analysis this is just noise that leads to a decrease of the Synfire Indicator. There is a straightforward way to disregard these background spikes by setting a threshold value C_{thr} for the SPIKE-Synchronization profile $C(t_k)$. Only spikes with a coincidence value higher than C_{thr} are taken into account, all other spikes are simply ignored. The result of this background correction can be seen in Fig. 3.12 for the same dataset already used in Fig. 3.11. Focusing the analysis on the reliable GDPs leads to an increase of the Synfire Indicator from 0.284 to 0.438.

As already mentioned before, one of the main results of our analysis is the sorted order of the spike trains itself. For these neurophysiological data it allows to identify the leading and the following neurons in the network and to project this information back on the recording setup. This is shown in Fig. 3.13 where we have color-coded the optimized Spike Train Order obtained in Fig. 3.12 within a 2D-plot of the neurons recorded from the hippocampal slice. For this example there appears to be a clear overall propagation from right to left but there is also a considerable degree of variability which might be due to a non-trivial connectivity within the network.

In Fig. 3.14 we apply the SPIKE-Order analysis to a second dataset recorded from a different slice, again focusing on the order within the global events only. Here we also added one new feature, the mean value of the Spike Train Order $E(t_k)$ for each global event (we use the maxima and minima of the SPIKE order profile $D(t_k)$ to delineate the GDPs). This again emphasizes the time-resolved nature of the SPIKE order and the Spike Train Order indicators.

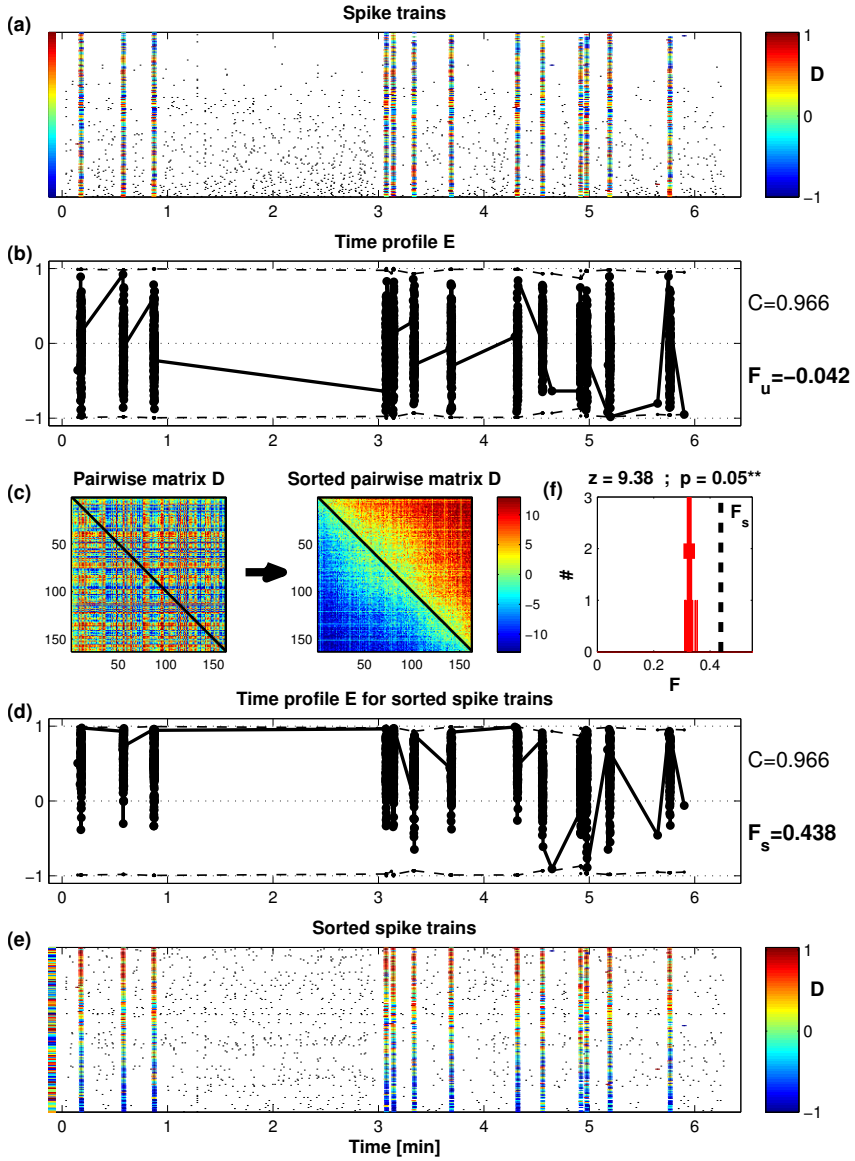

Figure 3.12: SPIKE-Order for the real data already analyzed in Fig. 3.11 but this time the analysis of SPIKE-Order was restricted to spikes with a SPIKE-Synchronization value of at least 0.7. This simple thresholding allows to focus the analysis on the reliable events and to disregard all spikes between the events (these are not colored and thus remain black). This results in an increase of the overall value of SPIKE-Order from 0.284 to 0.438.

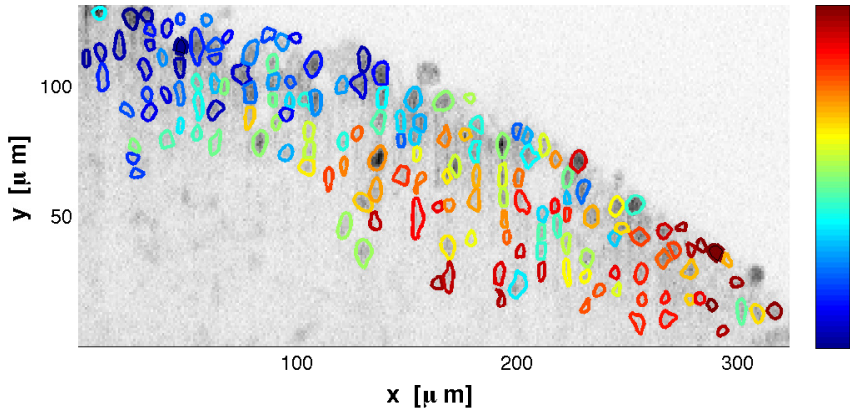

Figure 3.13: Projection of the optimized Spike Train Order on the 2D-photo of the hippocampal slice. The Regions of Interest (ROIs) which denote filled and identified cells in the CA3 region are color-coded from leader (index 1, red) to follower (index 163, blue) using the optimized Spike Train Order of Fig. 3.12. The very first leader (lower right) and the very last follower (upper left) are marked by filled contours.

Overall, we have analyzed neurophysiological datasets from four hippocampal slices exhibiting an average of 7.75 GDPs. We obtained an average value for SPIKE-Synchronization of 0.59 before focusing on the GDPs (as in Fig. 3.11) and 0.92 after (as in Fig. 3.12). With or without this focus the Synfire Indicator for the initial spike train sorting F_u was very close to zero and in all cases proved to be non-significant when tested against Synfire Indicators obtained for random permutations of the Spike Train Order. Since the initial sorting was based on overall firing rate of the neurons, this signifies that the hypothesis that the low-firing neurons which are basically only active during the GDPs might have a stronger role in initiating GDPs can be rejected. For the sorted spike trains the Synfire Indicator F_s was 0.20 for all spikes and 0.42 for the spikes within the GDPs only. Suppressing the effect of the noisy background spikes in the analysis thus leads to an average increase of the Synfire Indicator by about 110%. Finally, according to the surrogate analysis described in Section 3.2.5 the Synfire Indicator for the sorted spike trains F_s yielded a statistically significant result for all datasets analyzed.

So overall we can conclude that the GDPs recorded in brain slices from juvenile mice are distinguished by a very high consistency of their spatio-temporal propagation patterns. However, it is interesting to note that this consistency does not hold when comparing different slices. In the datasets analyzed in this paper we find examples of both propagation

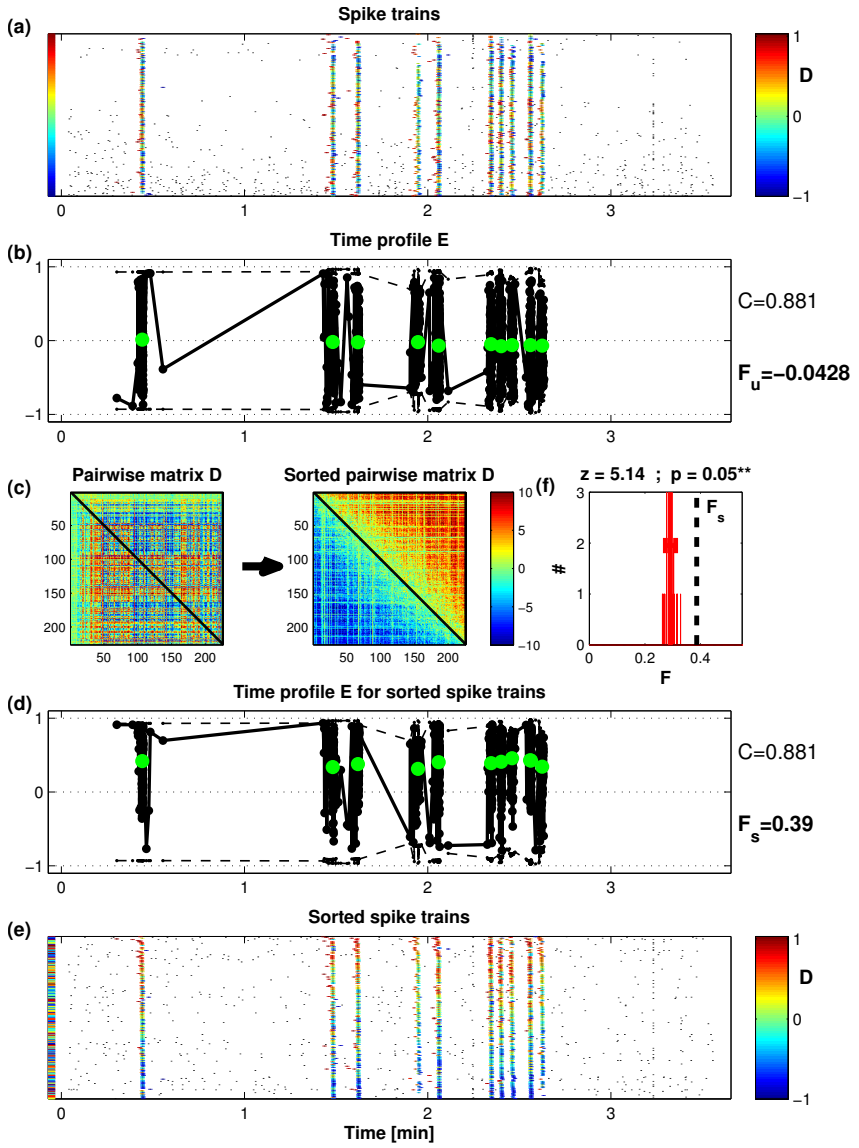

Figure 3.14: SPIKE-Order for a second dataset for which again the analysis of SPIKE-Order was restricted to spikes with a SPIKE-Synchronization value of at least 0.7. In this case the focus on the GDPs let the Synfire Indicator increase from 0.296 (result not shown) to 0.389. In addition, here we also calculated one average Spike Train Order value per GDP (green points) which illustrates once more the time-resolved nature of the method.

in the direction of CA2 as well as propagation towards the dentate gyrus. This is consistent with results reported in (63).

3.4 DISCUSSION

Over the last years a wide variety of measures to quantify the synchrony between spike trains have been introduced. Three recent proposals, *ISI-distance* (64, 4), *SPIKE-distance* (5, 6), and *SPIKE-Synchronization* (7, 58), share the desirable property of being time-resolved and parameter-free (time scale independent). However, their bivariate versions are symmetric and in consequence their multivariate versions are invariant to changes in the order of spike trains. None of these measures is designed to provide information about the directionality of the propagation patterns.

In the present study we address this issue. First we use an adaptive coincidence detection as match maker in order to identify pairs of coincident spikes. Then we define two measures, the asymmetric *SPIKE-Order D* and the symmetric *Spike Train Order E*, which are particularly useful in a bivariate representation (pairwise matrix) and as a time-resolved multivariate profile, respectively. From these two measures we can derive the *Synfire Indicator F*, a condensed scalar value that quantifies the overall consistency of the spatio-temporal propagation patterns in a rigorous and automated way. Its maximization allows to sort multiple spike trains from leader to follower. This is meant purely in the sense of temporal sequence of the spikes. The question asked is: For which spike trains do spikes tend to occur first and for which do they tend to occur last? We use simulated annealing to search among all permutations of spike trains for the sorting that resembles as closely as possible a synfire pattern, a perfectly consistent repetition of the same global propagation pattern. In a final step we evaluate the statistical significance of the optimized permutation using a set of carefully constructed spike train surrogates.

We first illustrate the merits of our new approach using artificially generated datasets and then apply it to real dataset.

The new algorithm is conceptually simple, of low computational cost and comes with an intuitive and straightforward visualization, including a color-coded rasterplot. It substantially improves on all the bivariate functionalities of its predecessor directional measure *delay asymmetry* (no need for sampled profiles, more intuitive normalization etc.) and could thus also be used in the context of the pairwise matrices, both normalized or cumulative, used in complex network theory (65, 66). However, one of the main advantages of the new algorithm is its multivariate na-

ture which opens up completely new kinds of application such as spike train sorting.

One important advantage that the method shares with other techniques of spike train analysis is the high flexibility in the definition of events. For example, when looking at the synchronization of neuronal bursts instead of individual spikes one can define the events as the onset, the center or the offset of the activity (e.g., the first, the middle or the last spike of each burst). In cases in which a burst of spikes is considered to be equivalent to a single spike one could introduce some kind of meta-events and then look at coincidences between these meta-events.

The application of our measures is also not restricted to truly discrete data. Continuously sampled data can be reduced to a spike train where the only information maintained is the timing of the individual events. Often these event times are obtained in a manner similar to how the neuronal spike times are extracted from recordings of neuronal membrane potentials (usually done via some kind of thresholding). Examples of sampled data to which measures of spike train synchrony have been applied include EEG data (26, 6, 67) and, outside of neuroscience, stock market velocity (68) - and rainfall events (65).

The algorithm is particularly suited for datasets with a high value of SPIKE-Synchronization. According to the coincidence criterion (Eq. 3.1) these are spike trains that include sequences of global events for which the interval between successive events is at least twice as large as the propagation time within an event. For these datasets the Synfire Indicator evaluates the consistency of the order within these well separated global events. The universality of the phenomenon, repetitive propagation patterns, makes our new algorithm applicable in a wide array of fields such as medical sciences, seismology, oceanography, meteorology or climatology. For example, the duration of an epileptic seizure is typically much shorter than the interval between two successive seizures. Also the time it takes a storm front to cross a specific region is typically much smaller than the time to the next storm. Many other repetitive propagation phenomena exhibit similar ratios of characteristic time scales.

In order to understand the scope of our proposed algorithm it is important to understand what it is not designed to achieve. The method deals purely with relative order, it does not consider the length of absolute delays. Moreover, while the instantaneous coincidence criterion makes the method time scale independent, parameter-free and easy to use, it also renders it insensitive to patterns involving spikes that are not immediately adjacent. Many other, typically more complicated, methods have been designed to characterize the detailed spatio-temporal structure in

large neuronal networks (69, 70) or to detect hierarchically structured spike-train communities (71, 72). The method is also not designed to detect neuronal synfire chains (in the strict sense of the word) in massively parallel data. For this task other statistical methods based on some forms of pattern detection have been developed (73, 74).

Another caveat concerns causality. While a significant value of the Synfire Indicator F_s in our algorithm clearly shows the presence of a preferred temporal order of some signals with respect to others, it does not necessarily prove a driver-responder relationship. There are other methods that have been developed for this kind of system dynamics analysis (e.g., (75)). But even for such methods causality is always a strong claim. In fact, the two signals might be driven by a common hidden source and a consistent leader (follower) could just indicate a drive with a smaller (larger) delay. Similarly, internal delay loops in one of the two systems can also fool the interpretation.

There are a number of possible directions for future research, both from a methods and from a data point of view. Regarding the algorithm, for the coincidence detection it would be straightforward to limit the range of allowed time lags by incorporating information about the expected speed of propagation (65). One could introduce a minimum time lag in order to ensure causality and/or limit the maximally allowed time lag in order to focus on meaningful propagation of activity. In principle the range of allowed time lags could even be selected individually for each pair of spike trains depending on the known properties of the connectivity between the respective two neurons. Importantly, even with such type of time lag restrictions in place, it has still to be guaranteed that each spike can be part of at most one coincidence.

A follow-up task for our neurophysiological data would be to investigate to what extent the neurons that are identified as leading by our analysis are identical to the so-called hub neurons (55), i.e. neurons with a much higher than average degree of connectivity within the network.

3.5 APPENDIX

3.5.1 *Neurophysiological dataset*

The neurophysiological data analyzed in Section 3.3.2 were recorded via fast multicellular calcium imaging in acute CA3 hippocampal slices from juvenile mice. The CA3 region has a strong recurrent excitatory connectivity (76). This distinct feature is suggested to be crucial for memory encoding and pattern completion and thus memory retrieval (77). During memory retrieval in rodents, population bursts of the CA3 lead to high frequency stimulation of the efferent regions, so called sharp wave ripples (78). In the juvenile hippocampus, due to a higher chloride reversal potential in the CA3 pyramidal cells, the GABA-ergic system is excitatory (62). GABA-ergic interneurons have been shown to serve as so called hub neurons that trigger the GDPs (55).

The recordings were performed by the group of Heinz Beck at the Department of Epileptology, University of Bonn, Germany, prior to and independently from the design of this study. Transversal acute brain slices ($300\mu m$ thick) were prepared from 5 to 10-day-old (P5-P10) C57BL/6 mice (Charles River, n = 19 slices). Slice preparation, calcium imaging and data analysis were performed as previously described in (79). For AM-loading of brain slices with OGB1-AM we used a protocol modified from (80). Multicellular calcium imaging was done using a homemade single planar illumination microscope (SPIM) modified from (81). Movies were recorded at a frame rate of 200 Hz over a minimal length of 5 min up to 30 min to record a sufficient amount of spontaneous activity. Time points of cell activity from the imaging data were defined as the onsets of Ca^2 events in fluorescence traces of all individual cells using the maximum of the second derivative of each event (82).

In order to test the Spike Train Order algorithm, datasets were chosen that exhibited at least three global GDPs during the recording ($n = 5$). For one dataset the surrogate analysis described in Section 3.2.5 proved to be unfeasible due to its excessive density of spontaneous activity. Therefore this dataset was discarded from further analysis, so the final number of datasets analyzed was $n = 4$.

4

RATE AND TEMPORAL CODING IN SPIKE TRAIN DISTANCES

Background: It is commonly assumed in neuronal coding that repeated presentations of a stimulus to a coding neuron elicit similar responses. One common way to assess similarity are spike train distances. These can be divided into spike-resolved, such as the Victor-Purpura and the van Rossum distance, and time-resolved, e.g. the ISI-, the SPIKE- and the RI-SPIKE-distance.
New Method: We use independent steady-rate Poisson processes as surrogates for spike trains with fixed rate and no timing information to address two basic questions: How does the sensitivity of the different spike train distances to temporal coding depend on the rates of the two processes and how do the distances deal with very low rates?
Results: Spike-resolved distances always contain rate information even for parameters indicating time coding. This is an issue for reasonably high rates but beneficial for very low rates. In contrast, the operational range for detecting time coding of time-resolved distances is superior at normal rates, but these measures produce artefacts at very low rates. The RI-SPIKE-distance is the only measure that is sensitive to timing information only.
Comparison with Existing Methods: While our results on rate-dependent expectation values for the spike-resolved distances agree with (22), we here go one step further and specifically investigate applicability for very low rates.
Conclusions: The most appropriate measure depends on the rates of the data being analysed. Accordingly, we summarize our results in one table that allows an easy selection of the preferred measure for any kind of data.

Adapted from: *Which spike train distance is most suitable for distinguishing rate and temporal coding?* (10)

4.1 INTRODUCTION

Neurons respond to stimulation with discrete events called spikes and a consecutive sequence of such spikes over time form a spike train (53). The effect of a spike is to release neuronal transmitters at synapses, and with enough input from the other neurons a downstream neuron fires a spike (2).

Neurons can code information in the rate and/or in the timing of the spikes (15). Neuronal information whether coded in spike rate or in spike timing is linked to the length of the encoding window. The rate at which a relevant property of the neural code can change, and the maximum rate at which changes in the stimulus can be represented, is the rate at which the neural code can adjust to the new representation. A definition for rate and time coding schemes has been provided by (83) as follows:

- *"It is generally accepted that a rate encoding scheme is one in which the relevant information encoded about the stimulus is correlated only with the number of elicited spikes within the encoding window and is not correlated with any aspect of the temporal pattern of the spikes within the encoding window."*

- *"In a temporal encoding scheme, the relevant information is correlated with the timing of the spikes within the encoding window, over and above any information that might be correlated with the number of spikes within the window."*

It is important to note that this definition of temporal coding does not require rate correlation actually to be present. Also it is not limited to single spike correlations but applies to any correlations in spike patterns that would not be expected due to rate alone.

There is plenty of evidence suggesting that information is coded in rate averaged over short periods of time. An example is rate coding in directional tuning within the motor cortex (84). However, another view is that for very fast reaction sensory and motor systems this kind of temporal averaging is too slow (85). When the organism needs to make decisions based on sensory stimuli in time scales of hundreds of milliseconds it is not possible to average over time at every stage of the pathway from stimulus detection to reaction. Thus it has been proposed that for each neuron first the spike latency of a response could code for intensities of a stimulus forming tuning curves similar to those of rate (85). The problem is that there is no specific trigger instant from when to calculate

time delays. However, spike time coding has been shown to be very generally used in relaying among others gustatory, somatosensory, olfactory, auditory, and visual information (86, 87, 88, 89, 90) at least as long as relevant timing signals are available to the experimenter (85). Using a simulation of ganglion cells it could be demonstrated that information transfer from multiple cells firing only one single spike can exceed that of pure rate coding (91).

Whichever coding type is used, the main assumption behind all neuronal coding research is that repeated presentations of the same stimulus result in a similar spike train responses, whereas presentations of different stimuli typically yield very dissimilar responses. One very common approach to measure such similarity are spike train distances (see e.g. (92, 19)), which are designed to assess similarity based on rate and timing within spike trains. In this paper we deal with four commonly used as well as the more recent spike train distances. The spike-resolved Victor-Purpura distance (12, 93), and the van Rossum distance (13) are defined by using spikes as the main elements of the measures while the time-resolved ISI-distance (3, 4), the SPIKE-distance (5, 6), and the RI-SPIKE-distance (8) are based on time.

The Victor-Purpura distance (12, 93) and the van Rossum distance (13) utilize time scale parameters q and τ respectively. For the extreme time scale parameter values $q = 0$ and τ approaching infinity, respectively, the distances evaluate spike count as an indicator of rate. While this works in some cases, it ignores any temporal correlations in spike timing within the counting window. In order to be more and more sensitive not only to rate but also to timing information, the time scale parameter has to be moved towards the opposite side of the parameter range. For this reason, these parameters are often taken as deciding between rate or time coding (94). Spike-resolved spike train distances have been applied to identify rate and time coding in recordings e.g. from the auditory system (in songbirds, (95, 96)) or the visual system (salamander retinas, (97), or visual cortex of cats, (98)). On the other hand, the time-resolved ISI-distance (3, 4), SPIKE-distance (5, 6), and RI-SPIKE-distance (8), are time scale free and thus do not require a parameter, but they assess timing information in the data in relation to the local time scale.

The existence of rate and time coding in neural representations within the brain (or even their coexistence in different brain regions) is a very important field of investigation. Since spike train distances are used to quantify the (dis)similarity of spike trains are thus one of the most used tools to tackle this issue, it is equally important to understand their definitions of similarity. Therefore, in the first part of this study we investigate the specific sensitivities of the measures to rate and time cod-

ing. Any distance responding to rate difference between the spike trains is sensitive to rate coding. If a distance is sensitive to spike timing, it causes deviations from the distance obtained for pure rate coding (rate difference) if such can be defined for the measure. This definition also implies that if a timing responsive distance is not sensitive to rate difference, it assesses purely timing information. These distinctions allow us to quantify the operational ranges of the distances for detecting time coding using pairs of independent steady rate Poisson spike trains (with reasonably high rates) as surrogates for random spike trains with no timing information.

In the second part of this study we look at limitations of the different spike train distances regarding their ability to deal with very low spike rates in the data. Neuronal spike trains have a very strong restriction regarding the information they can convey over the spike generating process that is called the floor effect. This is because they are discrete samples from a distribution and the smallest units measurable is a single spike. As an example one might have a neuron firing at 1Hz for a 1s recording and the spike count distribution consists mostly of values 0, 1, or 2 spikes. Equally, if a neuron fires at 2Hz, it still exhibits a considerable amounts of spike trains with 0, 1 or 2 spikes, even if it is more likely to also produce spike trains with 4 or 5 spikes. Since differences between 10Hz and 20Hz are more visible from fewer recordings, this floor effect only affects the low end of the rate scale. Here, due to the insufficient sampling it is not possible to carry out a meaningful statistical analysis of the spike train distances based on the number of spikes. Instead, in this part of the study we use sampling over multiple realizations of steady rate Poisson processes to estimate the minimal rate that is needed in order to still obtain reliable estimates of timings in the data.

The remainder of the paper is organized as follows. In the Methods (Sec. 4.2) we introduce the spike train distances used in this study and introduce the statistical method used. The Results (Sec. 4.3) are divided into three parts. In the first part we analyze both the spike-resolved and the time-resolved distances for the normal case where the total rate of the processes is reasonably high and thus far from exhibiting the floor effect. In the second part we examine the functionality of both types of distances for rates so very low that the floor effect takes place. A series of simple examples specifically constructed to illustrate the most important implications of the results obtained is presented in the third part. Finally, in Sec. 4.4 we discuss the results and present our conclusions.

4.2 METHODS

There are many different ways to quantify similarity in spike trains. In this Section we qualitatively describe four established and one recently proposed spike train distances. These spike train distances can be divided into two main categories. The first category (Sec. 4.2.1) contains measures where spikes are the main element for constructing the distance. For the second category (Sec. 4.2.2) the main unit of the analysis is time since the values are assigned over time rather than over spikes. The mathematical definitions for the spike-resolved and time-resolved spike train distances can be found in 4.5.1 and 4.5.2, respectively. Finally, in Sec. 4.2.3 we summarize the analysis and the statistical methods used in this study.

4.2.1 *Spike-resolved spike train distances*

The spike-resolved distances are based on the idea that for each spike there should be a matching spike in the other spike train. Thus even if not defined exactly in this manner, they will try finding pairs for spikes and consider unpaired spikes as rate difference. Thus, they consider rate first and timing second, since any excess spikes are always considered in the distance and only after that the timing differences are added.

Here we describe the two spike-resolved distances proposed by Victor and Purpura and by van Rossum which so far have been most commonly used in the context of neuronal coding. Other spike-resolved measures such as SPIKE-synchronization (7) are not considered in the scope of this study.

4.2.1.1 *Victor-Purpura distance*

The Victor-Purpura distance (12, 93) is based on finding the minimal cost for transforming one spike train into the other using the three elementary operations of deleting, inserting and shifting spikes. While both deletion and insertion carry a cost of 1, the parameter q is used to evaluate the cost of shifting a spike. The theoretical range of values the Victor-Purpura distance can obtain for a pair of spike trains will always fall in range of $[n_2 - n_1, n_2 + n_1]$, where n_1 and n_2 are the spike counts of the two spike trains. With a q-value of zero, shifting spikes to coincide costs nothing and in this case the total distance between spike trains is just the difference in spike count $(n_2 - n_1)$, since the extra spikes in the spike train with the higher rate need just to be deleted to convert one to another. On the other hand, very high q values require more timing accuracy between

Distance	Reasonably high rates			Very low rates		
	Timing	Rate	Synchrony	Timing	Rate	Synchrony
Victor-Purpura	x (✓*)	✓✓	x (✓*)	✓	✓	✓
van Rossum	x (✓*)	✓	x (✓*)	✓	✓	✓
ISI	✓	✓	✓✓	x	x	x
SPIKE	✓	✓	✓✓	x	x	x
RI-SPIKE	✓✓	x	✓	x	x	x
✓✓: very well suited ✓: can perform x: avoid using *: timing information available only for near equal rates						

Table 4.1: An overview of all spike train distances. Reasonably high rates denotes the normal case of rates that are high enough to avoid any floor effect (more than 4 spikes overall), whereas very low rates refers to rates so low that the floor effect takes place (total number of spikes equal to or lower than 4). Timing means pure spike timing information, rate pure spike count difference, and synchrony taking into account time local similarities in both rate and spike timing.

spikes, since the distance is the sum of all spikes in the spike trains that do not exactly match a spike in the other spike train (up to $n_2 + n_1$). Thus the parameter q is often taken as an indicator of the relative importance of rate and time coding (94).

4.2.1.2 *van Rossum distance*

The van Rossum distance (13) ties the spikes to a more biological context by using a kernel, which can be considered as the effect a spike will have on the postsynaptic neuron (see also (99)). An exponentially decaying kernel with time constant τ is applied to each spike and differences in the effect patterns are considered when calculating the distance. The larger is τ, the longer lasts the effect of a spike.

Although the resulting profiles are time-dependent there is no time-normalization. Rather, the values are obtained in a spike-based manner, since each spike is convolved with a kernel function. For this reason the distance obtained from the time-resolved profile is actually spike-resolved. Often the Victor-Purpura distance and the van Rossum distance are considered interchangeable with parameter conversion $\tau = 1/q$ (compare (100)).

4.2.2 *Time-resolved spike train distances*

For the time-resolved distances it is more important when a spike occurs in relation to its neighbours than if there is a match for it in the other spike train. As a consequence timing is more important than rate and in the distance rate is seen through differences in ISI-lengths rather than the spike count. Thus, these measures consider spike timing first and rate differences enter only as a consequence of that timing.

The three time-resolved measures used in this study are the ISI-distance, the SPIKE-distance and the RI-SPIKE-distance. These measures are calculated by integrating instantaneous dissimilarity values over continuous time rather than summing over discrete spikes. All of these distances are time scale independent since they do not require a time scale parameter.

The time-resolved profiles used to calculate these distances are obtained by comparing the spike trains based on previous and following spikes of both spike trains at each time moment. Due to missing information before the first and after the last spike, edge effect corrections are applied and special cases of empty spike trains and one spike are treated separately by adding auxiliary spikes (See 2.5.1).

4.2.2.1 *ISI-distance*

The ISI-distance (3, 4) assesses the dissimilarity of the spike trains based on instantaneous rate synchrony, where the rate is estimated from the inverse of the local interspike intervals (ISIs). It is calculated by averaging the local rate dissimilarity over the total length of the recording. The ISI-distance obtains the minimum value zero for identical local rates everywhere which means not only perfectly identical spike trains but also spike trains consisting of constant and equal interspike intervals with a global phase shift. It can grow arbitrarily close to one for periodic spike trains with ever larger rate differences.

4.2.2.2 *SPIKE-distance*

While the ISI-distance assesses local rate dissimilarity, the SPIKE-distance (5, 6) additionally takes into account difference in spike timing. For the simple case of spike trains with steady rates this means that whereas the ISI distance will assess any two processes with the same rate as identical, whereas the SPIKE-distance also evaluates the phase shift. The SPIKE-distance obtains its minimum value zero for exactly identical spike trains only. The theoretical upper limit is one (since this is the limit for the time profile), but in practice the maximum value is 0.55 due to how spikes can

be arranged to be as far from each other as possible in the spike trains (as can be shown using, e.g., an evolutionary algorithm).

4.2.2.3 *RI-SPIKE-distance*

The latest time-resolved measure is called rate independent (RI)-SPIKE-distance (8). While both the ISI-distance and the SPIKE-distance are sensitive to differences in rate, in the RI-SPIKE-distance this sensitivity has been removed. This allows the measure to purely focus on spike timing information.

4.2.3 *Analysis and statistical considerations*

In this study we use the definition of time coding as correlations beyond rate (83) to investigate how the sensitivity of the different spike train distances to rate and time coding depends on the rate of the spike trains. To address this question we follow common practice (see e.g. (101, 102, 103)) and use pairs of independent steady rate Poisson spike trains as surrogates for random spike trains with fixed rate and no timing information. We sample the distances over multiple realizations in order to calculate the expectation values and to estimate the statistical significance. Any spike trains from spike generating processes that are more similar than expected can be considered containing information beyond pure randomly distributed spikes. The results consist of two parts, one where we deal with reasonably high overall spike rates and one where we look at the special case of very low rates.

In the first part (Sec. 4.3.1) we reduce the dimension by keeping the sum of the two rates fixed and analyze how the rate ratio determines the expectation value, i.e., the distance value that results of rate coding alone. We then assess the ability of the spike train distances to detect timing information in the data via their operational ranges for temporal coding which we define as the range of values the distances can obtain after differences in rate have already been taken into account.

In the second part (Sec. 4.3.2) we examine the floor effect which can occur for spike trains with very low rates. Rate can be estimated by averaging over large numbers of redundant cells or by averaging over repeated presentations of the same stimulus (104). The rate is by definition an average quantity and not a property of a single spike train. While counting spikes will always provide an estimate of the rate, the question of similarity is more complicated for spike train distances, since they also contain timing information. Thus, first we compare in very general terms, using simple spike counting, estimates of the total rate of two

spike train pairs, one pair with equal rates and one pair with a very high rate ratio. In order to have any timing information that can be compared in the data, there must be at least one spike in both spike trains. We again use Poisson models to identify for both spike train pairs the lowest rate at which this is violated with only 5% probability. In the next step we evaluate for both kinds of spike train distances the full dependence of the distance values on the two rates of the independent Poisson processes. Once more we use a 5% confidence layer as indicator of a value being outside of the distribution obtained for pure rate.

In both parts of this study we work with spike trains of unit length 1s, since this way the rate exactly equals the expectation value for the number of spikes in one spike train. For the spike-resolved Victor-Purpura and van Rossum distance we also look at the influence of the respective time scale parameter. All our results for both the spike-resolved and the time-resolved spike train distances are gathered in Table 4.1.

4.3 RESULTS

In the first part (Sec. 4.3.1) we investigate the rate-dependent sensitivity of the different distances to temporal coding. Whereas this analysis is carried out for reasonably high rates, in the second part (Sec. 4.3.2) we focus on very low spike rates for which the floor effect can occur. Each time we first discuss the spike-resolved and then the time-resolved spike train distances. In the third part we illustrate with simple examples the most important differences between the two kinds of spike train distances.

4.3.1 *Detecting time coding*

Identifying time coding can be difficult, since many spike train distance measures respond to changes in both rate and timing. In order to detect time coding one has to know what is the expected distance when only rate is considered. To this aim, we first calculate the expectation values for pure rate code using Poisson spike trains of varying rate ratios by averaging over a sufficient number of realizations (here we use 1000) such that all timing fluctuations cancel out. For the spike-resolved Victor-Purpura distance it is straightforward to eliminate the effect of the rate coding. Here the operational range for time coding, i.e., the range of values that can be obtained beyond rate difference, is easily obtained by subtracting the spike count difference (Victor-Purpura distance

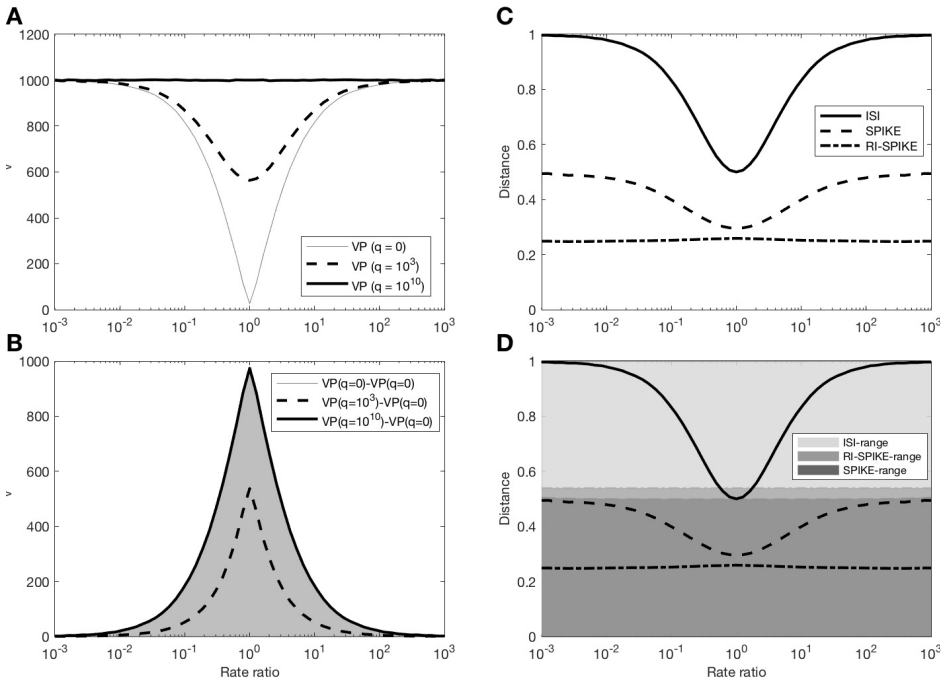

Figure 4.1: Statistics of two Poisson spike trains with a fixed total rate of 1000Hz divided among the two spike trains. (A) Dependence of the Victor-Purpura distance on the rate ratio for three different values of the time scale parameter q. (B) The operational range for spike time coding is marked in grey. (C) Same dependence for the ISI-distance, the SPIKE distance (compare (58)) and the RI-SPIKE-distance (see (8)). (D) Operational ranges for spike time coding. For all distances and rate ratios values can go down to almost zero which is indicated by increasingly darker shades of grey.

for $q = 0$). For the time-resolved distances this is not possible and the operational range becomes simply the range of values.

4.3.1.1 *Spike-resolved spike train distances*

First we estimate the operational range for time coding for the spike-resolved distances. As a first step in Fig. 4.1A we show the dependence of the Victor-Purpura distance on the rate ratio for three different values of the time scale parameter q. For large values of the parameter value q the distance always attains its maximum value independent of the rate ratio. Since the curves are made such that the sum of the rate of the two processes is 1000Hz for a pair of spike trains of unit length 1s, the result is a constant line at $D_V = n_1 + n_2 = 1000$ (compare Section 4.2.1.1). The total rate is kept reasonably high in order not to run into the floor effect. The only region where the parameter q has any reasonable effect is within the first decade around a rate ratio of one.

In Fig. 4.1B we depict the operational range for timing information obtained by subtracting the spike count difference ($q = 0$) which is always part of the total distance value independently of the time scale parameter. We can see that while for equal rate processes timing information covers almost the whole range of the distance, the operational range is lost very fast with increasing rate difference. As a result, ever larger portions of any distance value come from spike count difference. This holds true even for very large q-values which supposedly indicate timing information.

Thus, in order to obtain spike timing information using the Victor-Purpura distance, the spike trains must have very similar rates. As a consequence of how the distance is defined, the minimum distance for any spike train pair is obtained for $q = 0$. For this parameter value the distance equals exactly the spike count difference between the spike trains. In contrast, when q approaches infinity, the distance becomes spike count over both spike trains and considers spike trains with a smaller overall number of spikes as more similar. The only region where the distance has any time coding detection capability is when rates are almost identical and q is in some intermediate range. While the parameter q is often taken as deciding the relative importance of rate and time coding (94), this is not the whole story. We refrain from examining the van Rossum distance in detail, because it behaves similarly to the Victor-Purpura distance but requires normalization between parameter values since the maximum distance value depends on the choice of tau.

Due to these findings we suggest that one should not use the Victor-Purpura distance nor the van Rossum distance for detecting timing infor-

mation for spike trains that do not have nearly identical rates. Additionally, since the $q = 0$ case for the Victor-Purpura distance is simply spike count and estimates purely rate we regard it as well suited for detecting rate coding. These findings are marked at the top of the first main column of Table 4.1 for the normal case of reasonably high rates.

4.3.1.2 *Time-resolved spike train distances*

In Fig. 4.1C we show the rate ratio dependence for the ISI-distance, the SPIKE-distance and the RI-SPIKE-distance. The ISI-distance and the SPIKE-distance consider not only timing but also rate information. Accordingly, when only rate is considered, they obtain their minimum value for pairs of spike trains with equal rate. In contrast, the RI-SPIKE-distance ignores differences in rate and thus has a flat response independent of the rate ratio.

Fig. 4.1D depicts the operational ranges of all three measures. For any given rate ratio the operational range is defined as the overall range of values from minimum to maximum, i.e. the range that is covered due to deviations from the expectation value caused by the influence of spike timing. All three distances can obtain minimum values arbitrarily close to zero for any rate ratio (an extreme example for large rate ratios would be a infinitesimally narrow multi-spike burst matching a single spike). Regarding the maximum values, the ISI-distance is able to cover the whole interval by approaching the value of one arbitrarily close (the higher rate spike train is evenly distributed, the lower rate spike train is concentrated on the edges). This is not the case for the SPIKE-distance and the RI-SPIKE distance which yield the maximum values of 0.5 and 0.54, respectively, for spike train pairs with alternating spikes and the excess spikes of the higher rate spike train concentrated at the edges. [1]

The important result here is that, in contrast to the Victor-Purpura distance, all three distances cover a wide range of values even for very high rate ratios. For the ISI-distance and the SPIKE-distance this full operational range means that they are always able to identify both local rate and local spike timing information at the same time. This makes them the best candidates for evaluating synchrony in general. The RI-SPIKE-distance, on the other hand, was specifically designed to ignore differences in rate, so it is obvious that it should not be used for detect-

1 Note that the maximum values for the SPIKE- and the RI-SPIKE-distance shown here were obtained for sufficiently high spike numbers also in the lower rate spike train. Otherwise the edge effect can lead to slightly higher values (e.g. up to 0.61 instead of 0.5 for the SPIKE-distance). All results for maximum and minimum values were obtained and/or confirmed with an evolutionary algorithm (see, e.g. (105)).

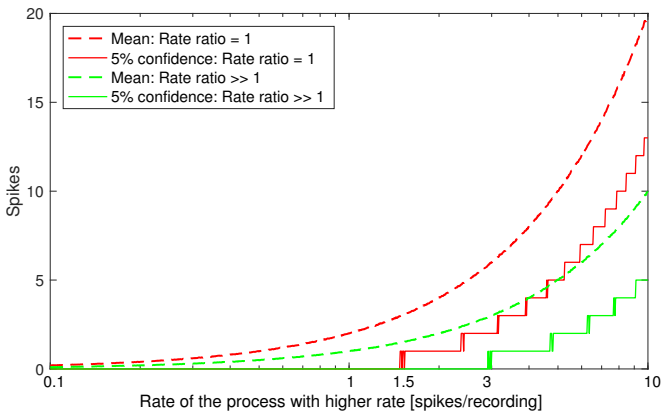

Figure 4.2: The floor effect for spike trains with a very low rate. Mean overall rates (dashed lines) and overall numbers of spikes in the two spike trains at 5% confidence (solid lines) over 10,000 realizations for a spike train pair with a rate ratio of one (red curves) and a pair with a very high rate ratio (green curves). While the mean values which perfectly match the expectation values (curves not visible) are growing linearly with the rate of the process with higher rate (note the logarithmic x-scale), the actual spike counts can only attain discrete values. Moreover, due to the floor effect it takes 1.5 spikes for spike train pairs with rate ratio one and 3 spikes for spike train pairs with very high rate ratio to get at least a single spike in either of the two spike trains at 95% probability.

ing rate coding. However, this very property makes it best suited for detecting pure timing information.

These results are marked in Table 4.1 at the bottom of the first main column for the standard case of reasonably high rates.

4.3.2 *Floor effect*

Even though the rate is a continuous property that can assume any value, the spike trains are sampling the rate with discrete samples. Thus, while for reasonably high rates the "resolution" of the sampling is high enough, as there are sufficient discrete samples available to give a good estimate of the actual rate, at the low end the sampling is not sufficient, since there are only a few discrete values the spike count can assume. This is called the floor effect.

In order to study the floor effect we first generate 10,000 realizations of two pairs of unit Poisson spike trains, one pair with equal rates (rate ratio = 1) and one pair for which the rate of one of the spike trains is lowered considerably (rate ratio $\gg 1$). Subsequently, for each of these

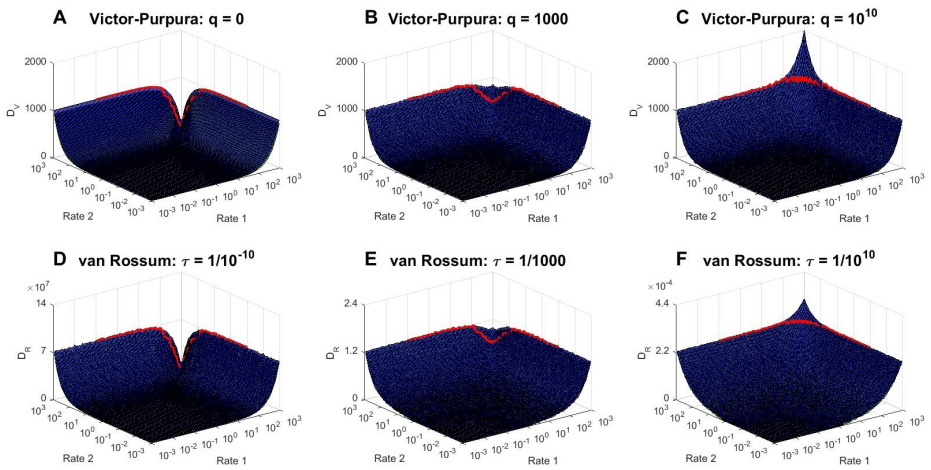

Figure 4.3: Statistical assessment of the Victor-Purpura distance and the van Rossum distance for three different values of the time scale parameter. The blue surface represents the mean of 1000 spike train pair realizations with corresponding rates. The green 5% confidence boundary trails the mean surface so close that it can not be seen. The red lines show fixed rate producing on average 1000 spikes in total in the two spike trains and thus corresponds to the curves shown in Fig. 4.1A. The parameter values for the Victor-Purpura distance are (A) $q = 0$, (B) $q = 1000$ and (C) $q = 10^{10}$. For the van Rossum distance we use the parameter values $\tau = 1/q$. Only for (D) we avoid division by zero by setting $\tau = 1/10^{-10}$. The other two cases yield (E) $\tau = 1/1000$ and (F) $\tau = 1/10^{10}$. Note that for the Victor-Purpura distance the range of values is only determined by the number of spikes and thus independent of the parameter value. For the van Rossum distance the time scale dependent kernel size changes the range of the distance axis as well.

pairs we calculate the mean and the 5% confidence boundary from the distribution of total spike counts of the sum of the two trains. This means that at any given rate there is only a 5% chance of getting spike trains with less spikes than this boundary.

As can be seen from Fig. 4.2, the average values follow the actual rate linearly, but the 5% confidence boundary is discrete, since spike trains can only have an integer number of spikes. Due to this floor effect, most of the rate information is not contained within one short spike train. One needs to have a good sampling of the process either from multiple repetitions or from longer recordings in order to get a meaningful rate estimate. While the mean of the rate is exactly the true rate of the processes, one can have empty spike trains at over 5% chance when the corresponding confidence curve is at 0. We can see from Fig. 4.2 that in order to reach non-empty spike trains at 95% confidence, one should obtain spike trains with an average rate of 3 spikes for high rate ratios and 1.5 for equal rates. This means that if one knows that the two spike trains are from the same process, they need to have a minimum rate of 1.5 spikes/recording. Note that this factor 2 between the 5% boundary for a rate ratio of 1 and the 5% boundary for a high rate ratio is observed over all rates. The reason is that for very high rate ratios the spike train with the lower rate tends to be empty and thus non zero samples are only drawn from one spike train instead of two.

4.3.2.1 *Spike-resolved spike train distances*

The rate dependence surface is plotted for many different rate combinations and three different parameter values for the Victor-Purpura distance in Fig. 4.3A-C, and for the corresponding parameter values with the van Rossum distance (Fig. 4.3D-F). The first thing to note is that the general shape of the surfaces is very alike between the two spike-resolved distances using the parameter conversion $\tau = 1/q$. In the following we discuss the Victor-Purpura distance in more detail, since the definition for it is more intuitive.

As can be seen in Fig. 4.3A-C, the smooth mean surfaces trace exactly the total rate of the processes for high rate ratios identical to Fig. 4.2 mean spike counts and the 5% confidence layer traces so close by to the mean that they cannot be distinguished. It is important to note that there is no visible artefact from the floor effect in the distance measure but the curves are smooth, since the distance is always primarily spike count difference. This is caused by the property that the parameter q only has an effect in a very narrow area near identical rates as we have already

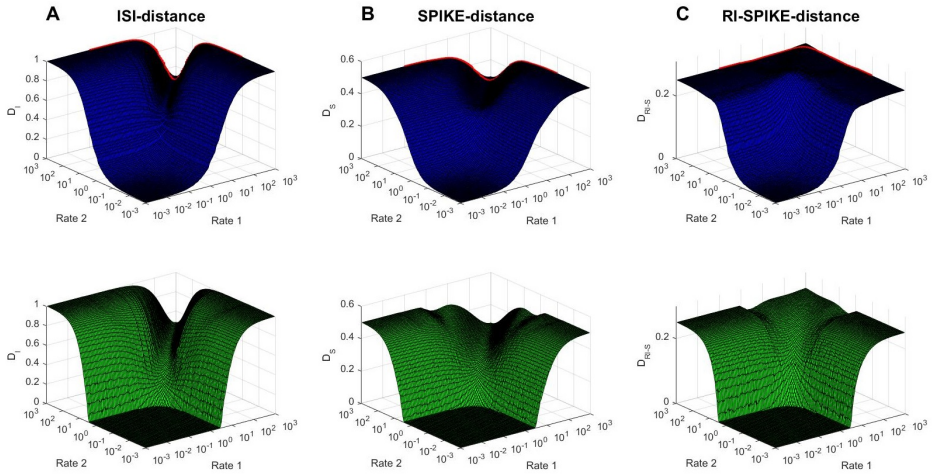

Figure 4.4: Same as Fig. 4.3, but this time for the ISI-distance (A), the SPIKE-distance (B) and the RI-SPIKE-distance (C). For the sake of visibility we separate the mean vales (blue, top) and the 5% confidence boundary (green, bottom) in two separate subplots. The red line represents the fixed rate equivalent to the curves in Fig. 4.1C.

seen in Sec. 4.3.1. For low rates the Victor-Purpura distance is almost flat and close to zero since the range $[n_2 - n_1, n_2 + n_1]$ tends to vanish.

In Fig. 4.3D-F we can see that for the van Rossum parameter $\tau = 1/q$ the distance performs very similar to the Victor-Purpura distance. However, a comparison of distances obtained with different parameter values can be done with the Victor-Purpura distance, but not with the van Rossum distance since there the range of values also depends on the time scale parameter.

Both distances can be used for very low rates without artefacts, since they assess rate first and timing second. These findings are marked at the top of the second main column of Table 4.1.

4.3.2.2 *Time-resolved spike train distances*

In Fig. 4.4 are again plotted the mean and 5% confidence boundary for the distribution, but this time for the ISI-distance, the SPIKE-distance and the RI-SPIKE-distance. The means of the distance values grow once the random processes start introducing spikes. In Fig. 4.4A one can see the mean rate surface of the ISI-distance. While the surface increases almost linearly up to 1Hz, the scaling to total time makes the distance settle to an expectation value that depends on the rate ratio. However, it is important to note, that the 5% confidence layer does only get above zero after a threshold rate is reached. This means that below this thresh-

old there is at least a 0.05 chance of getting two empty spike trains due to the floor effect (compare again Fig. 4.2).

In Fig. 4.4B we can see that the general shape of the mean surface for the SPIKE-distance shares the rate-dependent nature of the ISI-distance. On the other hand, the RI-SPIKE-distance (Fig. 4.4C) by construction shows no rate dependence. This means that the distance is purely based on spike timing and ignores rate differences in the spike trains.

In contrast to the spike-resolved Victor-Purpura and van Rossum distances, the time-resolved ISI-distance, SPIKE-distance and RI-SPIKE-distance can attain any distance value between zero and the maximum even for very low rates. While both kinds of distances are affected by the same floor effect, the definition of time coding by (83) as being over and above any information coded in rate is not satisfied for time-resolved distances, since the measures attempt using timing information even before there is sufficient rate. Only once it becomes unlikely to have two empty spike trains the boundary starts to increase and the values obtained start being reliable. This increase is not only manifested for empty spike trains. Already spike trains from processes with lower rates are more likely to be similar by chance than those obtained with higher rates. From this we can conclude that spike train distance values can only be considered non-random, when the rate of the spike-generating process is high enough not to produce empty (or quasi empty) spike trains and when the similarity of a pair is below the 5% confidence interval surface. Even for these the similarity has to be very pronounced not to be considered as being drawn from a random distribution.

The ISI-distance, the SPIKE-distance and the RI-SPIKE-distance are time-resolved which allows an instantaneous assessment of similarity. The normalization of the measures means that the spikes are assessed in relation to the length of the local ISIs. This time scale independence allows comparisons of spike trains with very different rates.

The reason why the ISI-distance, the SPIKE-distance and the RI-SPIKE-distance can draw any values even for spike trains that have very few if any spikes is an artefact from the normalization used for these measures. Since the values are time-resolved, and even a single spike needs to be comparable, the measures apply edge effect corrections (8). For this reason already for such extreme cases as a spike train pair composed of an empty spike train and a spike train with just one spike the distances can obtain virtually all of their range (depending on the location of this single spike). While the time-resolved nature gives the measures the advantage of being able to assess similarity in time, instead of just assessing timings of pairs of spikes, it becomes a downside for very low spike counts when not enough information is available to form mean-

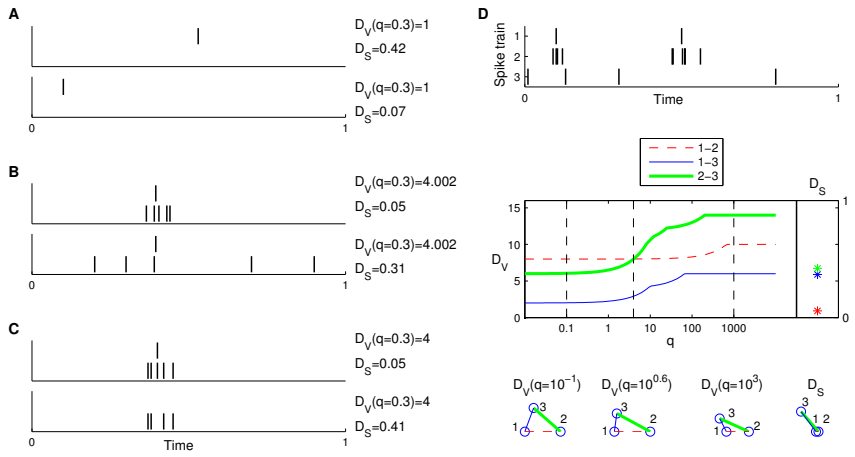

Figure 4.5: Simple examples illustrating some of the most important differences between the Victor-Purpura distance D_V and the SPIKE-distance D_S. (A) Floor effect: SPIKE-distance looks for timing in a single spike. (B) Spike trains with different rates: Victor-Purpura distance ignores timing information of extra spikes. (C) The Victor-Purpura distance is insensitive to exactly matching spikes. (D) Simple clustering example: In contrast to the SPIKE-distance, for spike trains with different rates the Victor-Purpura distance can never really focus on the timing information, even for large q-values.

ingful time-resolved profiles. Because of this we do not suggest using time-resolved measures for processes with a very low rate. These results for the ISI-distance, the SPIKE-distance and the RI-SPIKE-distance can be found in Table 4.1 at the bottom of the second main column for very low rates.

4.3.3 *Examples*

In previous Sections we have done statistical analysis of the spike train distances. In this Section we give a brief overview of the implications of the results using constructed examples (Fig 5). In Fig 4.5A we see typical examples of spike trains near the floor effect. There is hardly any visible rate and thus determining timing is problematic. Since the Victor-Purpura distance is primarily assessing rate and only second single spike timing, it gets the distance of one for both spike train pairs. However, the SPIKE-distance always incorporates timing and thus it can reach a very large variety of different values (0.42 and 0.07 in these example spike trains) even when there is very little timing information. This is due to the inflated contribution of the edge effects.

In Fig. 4.5B we see another situation, where we have two spikes at a short distance of each other. It is clearly visible that in the first spike train pair there is more timing information beyond rate than in the second. However, since the Victor-Purpura distance considers only the closest spike pairs, the distance will be the same independently of how the four other spikes are arranged. This is again due to rate first assessment of similarity. The SPIKE-distance uses timing assessment over time rather than over spikes and indicates a clear distinction between the two cases.

However, even the Victor-Purpura distance itself is ambiguous in respect to whether the value is obtained from rate or from timing. In Fig. 4.5C we see two very different scenarios. In the top spike train there is a perfectly coincident pair of spikes plus a few additional spikes in the second spike train. If we compare this to the exactly identical spike trains with only the two coinciding spikes removed we get exactly the same distance. In both cases this value comes purely from rate difference. However, there is a considerable difference in timing correlation between the two spike trains. This is clearly reflected by the values obtained for the SPIKE-distance.

In Fig. 4.5D we show how serious these effects are. As we mentioned already in Section 4.3.1, the Victor-Purpura distance and the van Rossum distance can hardly distinguish any timing information once the rate ratio is 10-fold or more. However the effects are realized already for much lower rate differences. In this example we consider three spike trains (2, 10 and 4 spikes). The first two spike trains clearly contain timing correlations, while the third does not correlate with either of the two. We can observe that the rate difference obtained with low q values (here $q = 10^{-2}$) is basically the spike count difference between pairs. The result is that the spike trains 1 and 3 have more similar rate than any other pair. Then by tracking the distances for increasing q-values the order of the distances remains the same until single spike timing reaches its peak at $q = 10^{0.6}$. Here the distances from 1 to 2 and 2 to 3 are equal. However, still the distance between 1 and 3 prevails as the smallest one indicating that they are the most similar pair. While after this q-value the distance from 1 to 2 is closer than 2 to 3, the pair with built in time correlations, 1 and 2 never reaches the range where it would be the most similar. In Fig. 4.5D relative distances are drawn as distance triangles with the respective distances for different q. For comparison we have added the distance triangle obtained for the spike trains using the SPIKE-distance, which, since it is not restricted to comparing pairs of spikes and thus rate first, finds the most time correlated pair with ease.

4.4 DISCUSSION AND CONCLUSIONS

Spike train distances can be constructed in a few different ways. The basic components common to all are the spikes in time. Some spike train distances are evaluated over values attached to spikes, like the Victor-Purpura distance (12, 93), and the van Rossum distance (13), and thus difference in spike count becomes a dominant feature. We call these spike-resolved distance measures. For the ISI-distance (3, 4), the SPIKE-distance (5, 6), and the RI-SPIKE-distance (8) effects of spikes are evaluated in relation to time and these distances are thus time-resolved.

In this study we asked two questions: How does the sensitivity of the different spike train distances to rate and time coding depend on the rates of the two processes and how high a rate is needed in order to obtain reliable estimates of timings in the data? To answer these questions we used two independent steady rate Poisson spike trains as surrogates for rate only coding neurons and calculated both the expectation values and the 5% confidence boundary over multiple realizations. The results are gathered in Table 4.1.

The first key finding of the analysis of time coding is that the spike-resolved Victor-Purpura distance compares the spike trains spike for spike and thus they are always sensitive to differences in spike counts even for parameter values seemingly indicating time coding. For large spike count differences the spike-resolved distances do not obtain the ability to assess timing information beyond spike pairs and thus in many cases most of the distance comes from mismatch in spike counts rather than timings, independently of the time scale parameter. As a result, for the Victor-Purpura distance timing information is only available for spike trains with almost identical rates (as illustrated in Fig. 4.5B and 4.5D). Since the behaviour of the van Rossum distance in response to rate differences closely resembles that of the Victor-Purpura distance, it also has the same problem (in addition to its normalization issues for different tau-values). These results are consistent with those obtained by (22) for a similar analysis of rate differences.

The second key finding is that the time-resolved measures perform better in assessing timings in the normal case of reasonably high rates. Also these measures can provide a meaningful instantaneous similarity profile within the coding window. Since they assess similarity in time, the exact spike count becomes less important and the actual timing of events becomes more relevant. However, they suffer from artefacts when the rates of the spike generation processes is so low that the floor effect takes place (see Fig. 4.5A for an example). For spike trains with only a few spikes one should use the spike-resolved Victor-Purpura or van

Rossum distance, since they assess first similarity in spike count and then apply timing information assessment only for pairs of spikes.

Investigation for neuronal coding has been going on for decades and the two most prominent approaches are to find similarities in responses in rate and in timing. However, while the distinction between the coding types is clear in a philosophical sense as presented by (83), the exact nature of time coding as being "over and above any information that might be correlated with the number of spikes within the window" is not uniquely defined. It depends on the type of correlation chosen.

For the spike-resolved distances sensitivity to rate never goes away since the information contained in the timing accuracy of spike pairs is always added on top of the rate information. For the Victor-Purpura distance, even for high q-values the relative importance of timing gets smaller for increasingly different rates. Moreover, this timing information only includes the distances between spikes that are needed to match all the spikes of the shorter spike train with their nearest spikes in the other spike train. While the importance of these differences does increase with higher q-values (Figure 4.1B) it is still capped by the maximum cost of 2 per spike pair (delete and add instead of shift). The timing of the other $n_2 - n_1$ spikes in the longer spike train is ignored entirely (see Fig. 4.5B), a loss of information that again increases with larger rate difference. Additionally, it is ambiguous if the distance obtained with parameter values indicating timing truly come from timing as shown in Fig. 4.5C. Therefore, while it may sound the intuitive thing to do, one cannot simply take the distance with a parameter indicating time coding and subtract the rate coding distance to get the timing correlations in the data. For the time-resolved ISI-distance, SPIKE-distance and RI-SPIKE-distance the measures have been defined as the integral over a dissimilarity profile that covers the whole recording time. In this case the intuitive difference between coinciding burst and steady rate is accounted for, since the assessment is not done spike by spike. On the other hand, for the ISI-distance or the SPIKE-distance there is no single value to be obtained for rate coding in order to subtract the time coding information content without a surrogate. The only measure we have shown to be independent of rate information is the RI-SPIKE-distance. This is an important notion, since one of the most essential questions in the analysis of neuronal coding is if spike trains contain information beyond rate and this measure is able to provide exactly this assessment.

For the investigation of neuronal coding the argument between rate and time coding types hinges crucially on the inaccuracy of the definition of time coding. Additionally the coding types are slightly mixed through the concept of the encoding window. If one estimates rate over

an encoding window and then splits it into multiple bins, which are essentially shorter encoding windows, the result will be an assessment of timing and timing accuracy depending on the bin size.

Based on our analysis we would advice against using the spike-resolved Victor-Purpura or van Rossum distances if one is interested in timings in the data when the rates are sufficiently high to avoid any floor effect. Also, the original interpretation of the parameters q and τ as precision of temporal coding ((94), but see also (22)) is slightly misleading in the light of this study, since it only works for nearly identical rates. In all cases, the information about differences in rate is always included. This seems to be consistent with results found in (106). As a result, it might be useful to reassess some older studies, where the Victor-Purpura distance and the van Rossum distance have been used for the distinction of time coding from rate coding in neuronal data such as the studies conducted by (95) and (96) for songbird data.

If one is interested in rate, the Victor-Purpura distance and the van Rossum distance can provide information of the rate difference between the spike trains. However, this information is equally available from the spike count and the true strength of the measures lies in assessing synchrony of spike trains with a very low rate, where they can provide a distance at the same time based on rate difference and on timings of single spikes. The results for the very low rate can be relevant in real data analysis, but most of the time there are enough spikes to use the time-resolved distances. The time-resolved distances always assess timing information due to how they are defined.

For simplicity the results in this paper have been obtained for spike trains of unit length 1s, since for a spike train of length one any rate will produce on average the same number of spikes as the rate. However, the values will scale with the recording length. This is very important, since a rate requirement of 1.5Hz for a spike train of unit length 1s will translate to 15Hz for a 100ms recording. Also the analysis performed here can be used to estimate window or bin sizes for methods that need to split a spike train into smaller segments. For this, assuming a steady rate over a recording, one can obtain a very good rate estimate via a better sampling of the process, since the original non-divided spike train is less likely to suffer from the floor effect.

It is important to note that from a statistical point of view the rate of the process and the length of the recording are inversely proportional. Recording a 10Hz process for 0.1s gives exactly the same amount of information about the distribution of the process as recording a 1Hz process for 1s. Thus it is possible to obtain the statistical significance of the rate of a steady rate process either by taking one long recording or

averaging over multiple short ones. Taking multiple short samples of a process may be more laborious than using one longer recording. On the other hand, it is harder to ensure stationary of the process over longer time than over multiple repetitions. Another problem with neurons is to ensure that the rate of the process does not change faster than it is sampled. However, assessing change in the rate of the process in relation to sampling is outside the scope of this paper.

In this study random spike trains were simulated as steady rate Poisson processes. While this approach is often used, it does not match many experimental ISI-distributions (e.g. (101, 107)). It is important to use a meaningful surrogate when evaluating whether the distance could have been obtained by chance. While there have been some studies on spike train surrogates (108, 109, 110), there is yet no simple answer as to which null hypothesis to test for and how to adapt the surrogates to the specific null hypothesis that is tested.

Here we compared four established spike train distance measures and one recently proposed measure. The same study could be conducted on other measures, e.g. on SPIKE-synchronization (7) or on other new classes of measures (30), to see how these approaches perform under the same conditions. It would also be interesting to see if spike-resolved and time-resolved distances all share some common characteristics. Also different kinds of normalizations and integrations in the measure descriptions may share common features. If one were able to construct a theoretical framework for the distances, perhaps by combining desired properties from complementary measures it could be possible to construct a measure that works universally in all cases. In the meantime, we suggest referring to Table 4.1 when deciding which of these measures to use for which kind of data.

4.5 APPENDIX

4.5.1 *Spike-resolved spike train distances*

In this paper we evaluate two spike-resolved distances, the Victor-Purpura distance (12, 93) and the van Rossum distance (13).

4.5.1.1 *Victor-Purpura distance*

The Victor-Purpura distance D_V (12, 93) is calculated by finding the smallest path to convert one spike train into the other using three elementary steps:

1. Deleting a spike with a cost of 1.
2. Inserting a spike with a cost of 1.
3. Shifting a spike to coincide with another spike in the other spike train with a cost of $q|\Delta t|$.

The time scale parameter q determines how far away two spikes can be in order for it to cost less than achieving the same by using steps 1 and 2. This parameter is thus considered as an indicator of the relative importance between time and rate coding.

4.5.1.2 *van Rossum distance*

The van Rossum distance D_R (13) first transforms discrete spikes into continuous functions by convolving each spike with an exponential kernel

$$H(t)\exp\left(-\frac{t}{\tau}\right), \tag{4.1}$$

where $H(t)$ is the Heaviside step function, t is time and τ the time constant. Using the resulting waveforms $\tilde{x}(t)$ and $\tilde{y}(t)$ the distance can then be calculated as

$$D_R(\tau) = \frac{1}{\tau}\int_0^\infty |\tilde{x}(t) - \tilde{y}(t)|^2 dt. \tag{4.2}$$

Quite recently a markage trick has been presented which significantly reduces the computational cost of calculating the van Rossum distance (111).

4.5.2 *Time-resolved spike train distances*

In this paper we also investigate three time-resolved spike train distances, the ISI-distance (3, 4), the SPIKE-distance (5, 6), and the very recently proposed RI-SPIKE-distance (8). Note that in (8) all of these distances have been adapted for data with multiple time scales. To see how these adaptive versions behave please refer to 4.5.2.4.

4.5.2.1 *ISI-distance*

The ISI-distance D_I (3, 4) measures the instantaneous rate difference between spike trains. It relies on a time-resolved profile, meaning that a dissimilarity value is defined for each time instant. To obtain this profile, we assign to each time instant t the time of the previous spike

$$t_{\mathrm{P}}^{(n)}(t) = \max\left\{t_i^{(n)} | t_i^{(n)} \leq t\right\} \quad \text{for} \quad t_1^{(n)} \leqslant t \leqslant t_{M_n}^{(n)} \tag{4.3}$$

and the time of the following spike

$$t_{\mathrm{F}}^{(n)}(t) = \min\left\{t_i^{(n)} | t_i^{(n)} > t\right\} \quad \text{for} \quad t_1^{(n)} \leqslant t \leqslant t_{M_n}^{(n)}. \tag{4.4}$$

From this for each spike train n an instantaneous interspike interval (ISI) can be calculated as

$$x_{\mathrm{ISI}}^{(n)}(t) = t_{\mathrm{F}}^{(n)}(t) - t_{\mathrm{P}}^{(n)}(t). \tag{4.5}$$

The pairwise ISI-profile is then defined as

$$I_{n,m}(t) = \frac{|x_{\mathrm{ISI}}^{(n)}(t) - x_{\mathrm{ISI}}^{(m)}(t)|}{\max\left\{x_{\mathrm{ISI}}^{(n)}(t), x_{\mathrm{ISI}}^{(m)}(t)\right\}}. \tag{4.6}$$

The multivariate ISI-profile is obtained by averaging over all pairwise ISI-profiles:

$$I(t) = \frac{2}{N(N-1)} \sum_{n=1}^{N-1} \sum_{m=n+1}^{N} I_{n,m}(t). \tag{4.7}$$

Finally, integration over time gives the distance value

$$D_I = \frac{1}{t_e - t_s} \int_{t_s}^{t_e} I(t) dt. \tag{4.8}$$

Here, t_s and t_e denote the start and the end of the recording, respectively.

4.5.2.2 *SPIKE-distance*

The SPIKE-distance D_S (5, 6) measures the relative spike timing between spike trains normalized to local firing rates. In order to assess the accuracy of spike events, each spike is assigned the distance to its nearest neighbor in the other spike train

$$\Delta t_i^{(n)} = \min_j(|t_i^{(n)} - t_j^{(m)}|). \tag{4.9}$$

These distances are then interpolated between spikes using for all times t the time differences to the previous spike

$$x_P^{(n)}(t) = t - t_i^{(n)} \quad \text{for} \quad t_i^{(n)} \leqslant t \leqslant t_{i+1}^{(n)}, \tag{4.10}$$

and to the following spike

$$x_F^{(n)}(t) = t_{i+1}^{(n)} - t \quad \text{for} \quad t_i^{(n)} \leqslant t \leqslant t_{i+1}^{(n)}. \tag{4.11}$$

This defines a time-resolved dissimilarity profile from discrete values the same way as Eqs. 4.3 and 4.4 did for the ISI-distance. The instantaneous weighted spike time difference for a spike train can then be calculated as the interpolation from one difference to the next

$$S_n(t) = \frac{\Delta t_i^{(n)}(t) x_F^{(n)}(t) + \Delta t_{i+1}^{(n)}(t) x_P^{(n)}(t)}{x_{\mathrm{ISI}}^{(n)}(t)} \quad , \quad t_i^{(n)} \leqslant t \leqslant t_{i+1}^{(n)}. \tag{4.12}$$

This function is analogous to the term $x_{\mathrm{ISI}}^{(n)}$ for the ISI-distance, with the only difference that it is piecewise linear instead of piecewise constant. It is also continuous.

The pairwise SPIKE-distance profile is then obtained by averaging the weighted spike time differences, normalizing to the local firing rate average and, finally, weighting each profile by the instantaneous firing rates of the two spike trains

$$S_{m,n}(t) = \frac{S_n x_{\mathrm{ISI}}^m(t) + S_m x_{\mathrm{ISI}}^n(t)}{2\left\langle x_{\mathrm{ISI}}^{n,m}(t)\right\rangle^2}. \tag{4.13}$$

From this the multivariate profile and the distance value can be calculated similar to Eqs. 4.7 and 4.8.

4.5.2.3 *RI-SPIKE-distance*

The rate-independent SPIKE-distance (RI-SPIKE-distance) D_S^{RI} (8) is similar to the SPIKE-distance, but leaves out the weighting by rate difference by substituting Eq. 4.13 with

$$S_{m,n}^{RI}(t) = \frac{S_n(t) + S_m(t)}{2\left\langle x_{\mathrm{ISI}}^{n,m}(t)\right\rangle}. \tag{4.14}$$

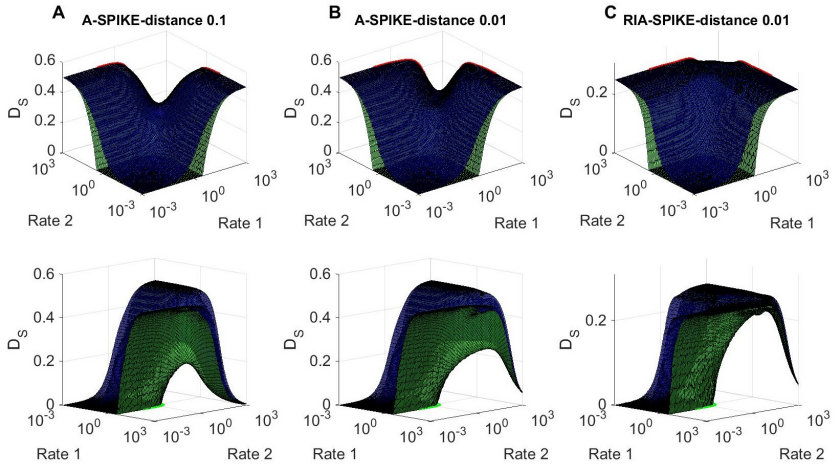

Figure 4.6: Adaptive spike train distances A-SPIKE-distance and RIA-SPIKE-distance with fixed parameters $\mathcal{T}$. In each case we present two different surface projections. (A) A-SPIKE-distance with $\mathcal{T} = 0.1$ applied to two Poisson spike trains of unit length. When both spike trains exhibit high rates compared to the global parameter $\mathcal{T}$ the spike trains are considered more similar. (B) A-SPIKE-distance with $\mathcal{T} = 0.01$. With smaller details still considered important the drop moves to higher frequencies. (C) RIA-SPIKE distance with $\mathcal{T} = 0.01$. The RIA-SPIKE distance does not take into account rate, but starts ignoring differences in spike times once both rates become high.

The RI-SPIKE-distance shares all the properties of the SPIKE-distance, but it only evaluates normalized spike timing differences, whereas the SPIKE-distance additionally uses differences in rate to determine similarity.

Again, the multivariate profile and the distance value can be obtained analogous to Eqs. 4.7 and 4.8.

4.5.2.4 *Adaptive spike train distances*

In (8) all three of the time-resolved distances described in Secs. 4.5.2.1-4.5.2.3 have been adapted for data containing multiple time scales by adding a notion of the relative importance of local differences compared to the global time scales. The adaptive versions start to gradually ignore differences between spike trains for ISIs that are smaller than a minimum relevant time scale (MRTS). The MRTS is implemented by an additional parameter $\mathcal{T}$ which can either be set by the user or estimated directly from the data.

In the present study we basically evaluated these adaptive versions with the threshold parameter $\mathcal{T}$ set to zero, which is equivalent to us-

ing the original distances. This makes sense since the generalizations are primarily designed to reduce the importance of small time scales in datasets containing multiple time scales, but the steady rate Poissonians analyzed here contain only one time scale.

The only reasonable comparison using the adaptive versions is when the threshold is fixed to a constant value. This causes higher rates to be considered as less significant for dissimilarity (once the spike trains get very dense, relative differences in ISIs or spike times hardly matter). So for completeness, in Fig. 4.6 we provide the results for the adaptive distance to be compared with Fig. 4.4. Higher rates with a rate ratio close to 1 are considered as more similar because the differences are small in comparison to the threshold. For a smaller threshold (Fig. 4.6B) the area where similarity is enforced moves to higher frequencies. The adaptive versions are designed to work with datasets containing multiple time scales such as regular spiking and bursts and for a fixed threshold the spike trains with a higher rate are considered as long bursts.

We also looked at the results for an automated threshold. Here the graphs look almost identical to the $\mathcal{T} = 0$ case (results not shown). However, and more importantly, the results can not really be compared in a meaningful way, since the threshold is different for different spike train pairs (see (8), for details).

5

THE MOST DISCRIMINATIVE NEURONAL SUBPOPULATIONS

Background: Spike trains of multiple neurons can be analyzed following the summed population (SP) or the labeled line (LL) hypothesis. Responses to external stimuli are generated by a neuronal population as a whole or the individual neurons have encoding capacities of their own. The SPIKE-distance estimated either for a single, pooled spike train over a population or for each neuron separately can serve to quantify these responses.
New Method: For the SP case we compare three algorithms that search for the most discriminative subpopulation over all stimulus pairs. For the LL case we introduce a new algorithm that combines neurons that individually separate different pairs of stimuli best.
Results: The best approach for SP is a brute force search over all possible subpopulations. However, it is only feasible for small populations. For more realistic settings, simulated annealing clearly outperforms gradient algorithms with only a limited increase in computational load. Our novel LL approach can handle very involved coding scenarios despite its computational ease.
Comparison with Existing Methods: Spike train distances have been extended to the analysis of neural populations interpolating between SP and LL coding. This includes parametrizing the importance of distinguishing spikes being fired in different neurons. Yet, these approaches only consider the population as a whole. The explicit focus on subpopulations render our algorithms complimentary.
Conclusions: The spectrum of encoding possibilities in neural populations is broad. The SP and LL cases are two extremes for which our algorithms provide correct identification results.

Adapted from: *Using spike train distances to identify the most discriminative neuronal subpopulation* (11)

5.1 INTRODUCTION

The nervous system is believed to employ large populations of neurons to code and broadcast information. Population coding can be considered less vulnerable and, hence, a more reliable and robust manner than coding via single neurons (112). In neuronal recordings population coding can appear in two ways. First, all the neurons in the recorded population contribute equally (113). Patterns of activity within the population are irrelevant for coding as all that matters is whether or not any of the neurons fires. There, the information being conveyed is that of a single spike train generated by the population as a whole. In contrast to this so-called *summed population (SP) hypothesis*, each neuron may have a unique and distinguishable role (114, 115, 116). In this case, the population is best decoded neuron-by-neuron, which is referred to as the *labeled line (LL) hypothesis* (117). Examples for the relevance of each coding scheme in experimental data can be found, e.g., in (118) (SP) and (119) (LL).

When recording a neuronal population after stimulus presentation, usually only some of the neurons encode the stimulus while others might be involved in different tasks or may exhibit a seemingly erratic activity independent of the stimulus. The responses of these non-coding neurons do not contribute to stimulus discrimination but rather act as a noisy disturbance if included in the analysis. We evaluated different methods to distinguish coding from non-coding neurons under either the SP- or the LL-hypothesis. As will be shown below, the two presumptions require different ways for evaluating stimulus discrimination.

Spike train distances are a useful means to assess neuronal coding by clustering responses to repeated presentations of a given set of stimuli. If the distance is chosen to be sensitive to the distinguishing features in the spike trains, a small distance between responses to the same stimulus and a large distance between responses to different stimuli can be obtained. While this kind of analysis has been mainly carried out for individual neurons (see e.g. (97, 95, 96)), current technical advances (e.g., (37, 38)) allow for studying neuronal coding in simultaneously recorded populations of neurons (120, 121). It has been shown that sensory information is typically not localized in individual neurons (122) but appears to be distributed over larger neuronal populations (123, 124). However, the coding via individual neurons and the summation of an entire population are the extreme case in a broad spectrum of possibilities (125, 126). In fact, recent evidence points at some intermediate scenario in which a comparably small number encodes information not only in a robust but also very efficient way (127, 128). Ince and colleagues (129) reported that sensory cortical circuits may process information using small but highly

informative ensembles consisting of a few privileged neurons. In the context of brain computer interfaces (BCIs), it was found that a reduced set of carefully selected important neurons exceeded BCI performance levels of the full ensemble (130).

The search for an optimal coding population requires fine-tuned analyses under both the SP- and the LL-hypothesis. For these two cases we show how to separate relevant from irrelevant subpopulations by identifying the subpopulation of neurons amongst all possible ones that discriminates best a given set of stimuli.

5.2 SPIKE TRAIN DISTANCES FOR NEURONAL DECODING

Spike train distances can measure the extent to which in a coding population repeated presentations of the same stimulus yield similar spike train responses, while different stimuli result in dissimilar responses. To simulate this, we considered the following setup. N neurons are simultaneously recorded upon repeated presentations of different stimuli – in a real experiment this is typically done with a multi-electrode array. The number of stimuli S and the number of repetitions R yield an overall number of trials by means of $T = S \cdot R$. Different spike trains are here denoted as $t_{n,s,r}$ with $n = 1, \ldots, N$, $s = 1, \ldots, S$ and $r = 1, \ldots, R$ indexing neurons, stimuli, and repetitions, respectively. Across simulations we selected a subset of neurons to be the coding subpopulation. The goal was, hence, to identify that subset, i.e. the neuronal subpopulation that collectively could distinguish between stimuli.

Spike train distances quantify the similarity of neuronal activity based on rate and timing within spike trains (see e.g. (117, 21, 19)). Over the years, many different distances have been proposed, including time scale dependent measures such as the Victor-Purpura distance (12) or the van Rossum distance (13) but also time scale independent approaches like the ISI-distance (3, 4) and the SPIKE-distance (5, 6). Here, we employed the SPIKE-distance D (6) as it offers the possibility of time scale and parameter-free assessments via the relative spike timing between spike trains normalized to the local firing rates (10). The smaller its value, the more similar the spike trains are, with $D = 0$ indicating identical spike trains. A detailed description of the SPIKE-distance can be found in the Appendix.

Neuronal coding can be assessed by determining the matrix of pairwise spike train distances over all trials (see e.g. (95, 96, 97)). How do these distances cluster in response to different stimuli? Identifying clusters depends on the presumed type of encoding. As said, we distinguish

between the SP- and the LL-hypotheses, i.e. we either combine all neurons into a single population or treat each neuron separately. In both cases, we determined distance matrices and estimated their stimulus discrimination performance to quantify how a subpopulation succeeds in discerning different stimuli.

For the summed population case, we compared three fundamentally different algorithms for finding the population that is able to most efficiently discriminate between a set of stimuli. (i) For comparably small neuronal populations one can perform a brute force search in which pairwise distance matrices and their stimulus discrimination performance are calculated for all possible subpopulations. (ii) A gradient algorithm used by (129) relies on a restricted number of subpopulations: from a given starting subpopulation one searches for the optimum performance by simply following a maximum local ascent. There are two alternatives. (129) followed a bottom-up variant that starts with the best individual neuron and gradually adds neurons. In addition, we also considered a top-down variant that iteratively subtracts neurons starting from the full population. (iii) A conventional albeit heuristic approach taken from statistical physics is simulated annealing. It is known for being less prone to getting stuck in suboptimal solutions as the aforementioned gradient ascent optimization methods.

For the labeled line case, we introduce a novel algorithm for identifying the most discriminative LL population. It performs a selection process that evaluates each individual neuron separately and forms the optimized subpopulation by combining the best neurons from every pair of stimuli.

5.3 DATA

For both the SP and the LL case we used a Poisson neuron model with an absolute refractory period of 2 ms. We always considered pure time coding. All coding and non-coding neurons had the same baseline rate M. For our first examples we also assumed the coding of the optimal subpopulation to be noiseless and perfect, though later on this assumption was weakened by adding noise.

5.3.1 *Summed Population (SP)*

We simulated spike train responses for a group of neurons that code in unison but not individually. These coding neurons were complemented by non-coding ones, which were simulated separately as neurons with-

out responses beyond baseline activity. We show the generation of the SP data using an example with $S = 4$ stimuli and $R = 5$ repetitions each, so overall $T = 20$ trials. The population comprised of $N = 7$ neurons, the first $c = 3$ of which were coding perfectly for the different stimuli and the last four were non-coding. Fig. 5.1 depicts four exemplary spike train raster plots: responses to the first two repeated presentations of the first two stimuli. The same data also serve to illustrate the further procedures in Figs. 5.2 and 5.3.

The pooled spike train of the first c coding spike trains was generated randomly but different for each of the S stimuli. Subsequently, for each of the R trials of every stimulus the spikes of the pooled spike train were evenly distributed among the c coding neurons. For the $N - c$ noisy, non-coding neurons, a trial was independent of every other trial irrespective of the stimulus. Throughout procedures, we ensured that for each trial the expectation value for the rate was the same for all individual neurons. Hence, the SP activity of the coding neurons for a given stimulus agreed exactly across trials but the activity of both coding and non-coding individual neurons remained largely random. As a result, the coding subpopulation discriminated the different stimuli perfectly, while all its 'superpopulations' (populations which contain it as subpopulation) as well as all its subpopulations did not perform likewise well.

5.3.2 *Labeled Line (LL)*

To study the LL case, we combined a set of stimuli with a population of neurons by varying the responsiveness of the neurons to these stimuli with the following setting in mind: Usually, every stimulus consists of a combination of different features and the individual neurons are either sensitive to these features or not. This implies that a stimulus may be coded by more than one neuron but also that for a diverse set of stimuli a combination of neurons is required to discriminate between all of them. Similarly, every individual neuron may be sensitive to more than one stimulus, to only one or even to none of the stimuli.

When simulating the spike trains, we assumed that for a neuron not sensitive to any of the features present in a stimulus, every stimulus repetition yielded random firing. On the contrary, a neuron sensitive to a certain stimulus responded very reliably (consistently) to repeated presentations of that very stimulus. In essence, we created a single spike train and copied it. In order to make the resulting spike trains more realistic, we used different realizations of jitter noise up to ± 5 ms) for every repetition. We note that changing the amplitude of the noise altered

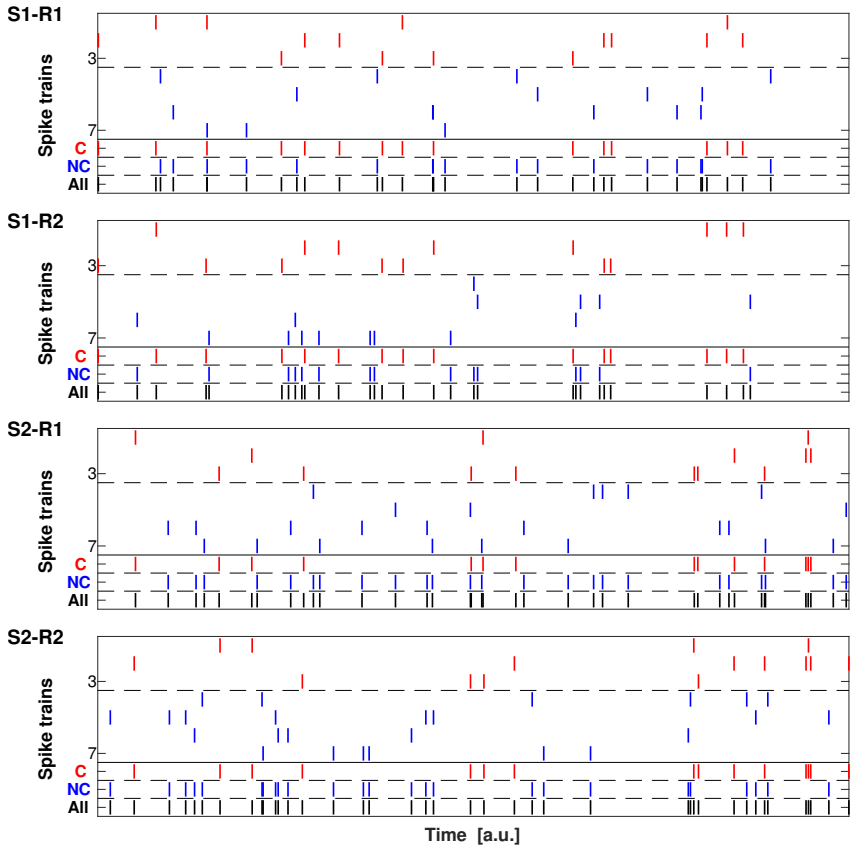

Figure 5.1: Summed population coding: Simulated spike train responses obtained from the first two of $R=5$ repetitions for the first two of $S=4$ stimuli (i.e. overall four out of $T=20$ trials). There are $N=7$ neurons of which $c=3$ form the coding subpopulation (in red) whereas the remaining 4 neurons are just noisy (in blue). Below the spikes of the individual neurons we depict the pooled spike train of the coding subpopulation ("C", red), the noisy (non-coding) subpopulation ("NC", blue) and the whole population ("All", black). By construction, the pooled spike train of the coding subpopulation was identical for different repetitions of the same stimulus (i.e., for the first two and for the last two rasterplots), while the pooled response of the non-coding subpopulation was random.

the reliability (consistency) of the responses and we were able to control the responsiveness of the neurons to certain stimuli. As in the SP case, we always used a (statistically) constant rate for all spike trains, while controlling for reliability.

5.4 METHODS

Our approach to assess population coding is to search for the subpopulation with maximum discriminative power across the responses to repeated presentations of a set of stimuli. Since the summed population and the labeled line hypotheses make different assumptions about neuronal coding, this task is addressed using fundamentally different algorithms. However, both analyses rely on similar pairwise distance matrices of the SPIKE-distance D, which in the case of SP are calculated for neuronal subpopulations, in the case of LL for individual neurons. The two algorithms also share the same basic discrimination performance $P^{s,\bar{s}}$ which quantifies for all repetitions R of each pair of stimuli $s, \bar{s}$ the degree to which identical stimuli give rise to similar responses and different stimuli result in dissimilar responses:

$$\begin{aligned} P^{s,\bar{s}} &= \frac{1}{R^2} \sum_{r,\bar{r}} D^{s,r;\bar{s},\bar{r}} - \frac{1}{2R(R-1)} \sum_{s,r,\bar{r}\neq r} D^{s,r;s,\bar{r}} \\ &= \left\langle D^{s,r;\bar{s},\bar{r}} \right\rangle_{r,\bar{r}} - \left\langle D^{s,r;s,\bar{r}} \right\rangle_{s,r,\bar{r}\neq r} . \end{aligned} \tag{5.1}$$

Hence, the larger the mean inter-stimuli distance and the smaller the mean intra-stimulus distance, the better the two stimuli can be distinguished.

In the SP case for the pooled spike trains of every subpopulation under evaluation the discrimination performance is computed for all stimulus pairs at the same time. The most discriminative SP subpopulation is the one that yields the highest average performance. In contrast, for LL every individual neuron is evaluated separately. Since often different stimulus pairs will be distinguished best by different neurons, the discrimination performance is optimized for one stimulus pair at a time. For every stimulus pair the algorithm identifies the discriminative neurons and selects the best one. Together, the selected neurons form the most discriminative LL subpopulation.

5.4.1 *Summed population (SP)*

5.4.1.1 *Discrimination Performance*

The first step of the summed population analysis for any given subpopulation $\mathcal{K}$, stimulus s and repetition r, is to pool the spike trains from all the neurons of this subpopulation according to

$$t_{\mathcal{K},s,r} = \bigcup_{k\in\mathcal{K}} t_{k,s,r}. \tag{5.2}$$

The matrix of all pairwise spike train distances between all $T = S \cdot R$ pooled responses can be readily determined. We denote them as

$$D_{\mathcal{K}}^{s,r;\bar{s},\bar{r}} = D\left(t_{\mathcal{K},s,r}, t_{\mathcal{K},\bar{s},\bar{r}}\right). \tag{5.3}$$

If a neuron population is able to discriminate a pair of stimuli, one can expect low values of the spike train distances between different repetitions of the same stimulus (intra-stimulus), but high values for different stimuli (inter-stimulus). In the SP case all stimulus pairs are evaluated at the same time. Accordingly, one can introduce the discrimination performance of a subpopulation $\mathcal{K}$ as

$$\begin{aligned} P_{\mathcal{K}} &= \frac{1}{S(S-1)} \sum_{s,\bar{s}\neq s} P_{\mathcal{K}}^{s,\bar{s}} \\ &= \frac{1}{S(S-1)R^2} \sum_{s,\bar{s}\neq s,r,\bar{r}} D_{\mathcal{K}}^{s,r;\bar{s},\bar{r}} - \frac{1}{SR(R-1)} \sum_{s,r,\bar{r}\neq r} D_{\mathcal{K}}^{s,r;s,\bar{r}} \\ &= \left\langle D_{\mathcal{K}}^{s,r;\bar{s},\bar{r}} \right\rangle_{s,\bar{s}\neq s,r,\bar{r}} - \left\langle D_{\mathcal{K}}^{s,r;s,\bar{r}} \right\rangle_{s,r,\bar{r}\neq r}, \end{aligned} \tag{5.4}$$

with $P_{\mathcal{K}}^{s,\bar{s}}$ given in Eq. 5.1 and here specified via subscript to the subpopulation $\mathcal{K}$. The better subpopulation $\mathcal{K}$ is able to distinguish the different stimuli, the higher the value of $P_{\mathcal{K}}$. Note that in Eq. 5.1 we could reduce the computational cost by making use of the fact that the initial loop $\langle$inter-stimuli$\rangle - \langle$intra-stimulus$\rangle$ over stimulus pairs can be transformed into the mean inter-stimulus distance minus the mean intra-stimulus distance for the entire discrimination matrix.

Fig. 5.2 shows the pairwise distance matrices and their discrimination performance values for three subpopulations of the data set exemplarily shown in Fig. 5.1. In this noise-free example, the coding subpopulation (the first three neurons in Fig. 5.2A) was able to discriminate perfectly and, accordingly, we obtained a very high value of P. The non-coding subpopulation (last four neurons in Fig. 5.2A) could not distinguish between the different stimuli. Its discrimination performance P was very close to the expected zero value (Fig. 5.2B). Finally, for the pairwise distance matrix of the full population (Fig. 5.2C), which contained both the coding and the noisy non-coding subpopulation, we found an intermediate discrimination performance P.

5.4.1.2 *Algorithms*

Since the measure $P_{\mathcal{K}}$ in Eq. 5.4 quantifies the discrimination performance for every given subpopulation, it can serve to search the space of

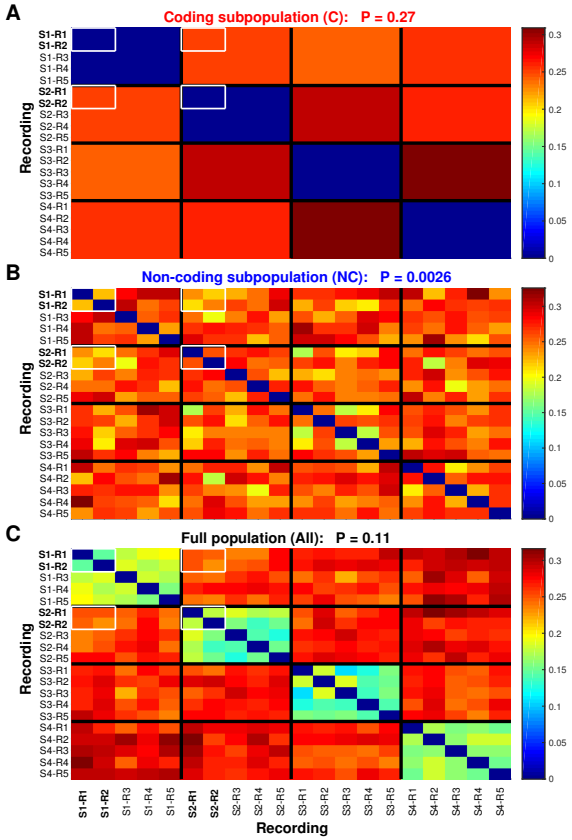

Figure 5.2: Summed population coding: Stimulus-dependent clustering for the seven neurons (three coding, four non-coding) of Fig. 5.1: Pairwise spike train distance matrices of all $T = 20$ trials consisting of $S = 4$ stimuli with $R = 5$ repetitions each for three different subpopulations: A. The coding subpopulation consisting of first three neurons (C, red) distinguish the stimulus perfectly. The different stimuli can be distinguished easily because in this noise-free case high distances are obtained for inter-stimuli realizations and zero values for intra-stimuli realizations. Accordingly a very large discrimination performance is obtained. B. Evaluating the summed activity of the last four neurons, the non-coding subpopulation (NC, blue) leads to seemingly random distances and stimulus discrimination fails. Such a distance matrix results in a very low discrimination performance. C. For the full population (All, black) the intra-stimulus submatrices can still be distinguished but are much less pronounced. Accordingly, the discrimination performance P attains some intermediate value. The sub-matrices resulting from the four examples (first two repetitions of the first two stimuli) given in Fig. 5.1 are marked by white boxes.

all possible subpopulations for the best SP-coding subpopulation $\mathcal{K}_{\text{opt}}$, defined as

$$\mathcal{K}_{\text{opt}} : P^{\text{SP}}_{\mathcal{K}_{\text{opt}}} = \max_{\mathcal{K}} \{P_{\mathcal{K}}\}. \tag{5.5}$$

As mentioned in Section 5.2, there are three different approaches to this task.

(i) Brute force

One can determine the stimulus discrimination performance $P_{\mathcal{K}}$ for the summed activity of every possible subpopulation $\mathcal{K}$, and identifies the subpopulation that provides the maximum performance. Since all possible subpopulations are evaluated, the brute force approach is guaranteed to find the best subpopulation. Its result can thus serve as a ground truth for other less exhaustive algorithms. Evaluating all possible subpopulations, however, is not feasible for very large datasets because the number of possible subpopulations increases exponentially with the number of neurons N:

$$K_{\text{bf}} = \sum_{k=1}^{N} \binom{N}{k} = 2^N - 1. \tag{5.6}$$

For example, for $N = 125$ the individual terms from different subpopulation sizes add up to $K_{\text{bf}} = 4.25 \cdot 10^{37}$, which is far beyond the limits of current soft- and hardware implementations. More restrictive algorithms are needed that explore only a (relevant) subspace of the numerous subpopulations.

(ii) Gradient algorithms

The idea behind gradient algorithms is to evaluate the discrimination performance for a restricted number of neuronal subpopulations. There are two variants: The *bottom-up variant* used by (129) starts with the best individual neuron and builds up the population by adding in each iteration the best remaining neuron. The alternative *top-down variant* starts from the complete population and iteratively subtracts one neuron at a time. Both variants are illustrated in Fig. 5.3 using the example from Figs. 5.1 and 5.2. In this example, both gradient variants correctly identified the first 3 neurons as the coding subpopulation. Importantly, for either algorithm the number of combinations for which the stimulus discrimination performance had to be calculated amounts only to

$$K_{\text{grad}} = \sum_{k=1}^{N} k = \tfrac{1}{2} N(N+1), \tag{5.7}$$

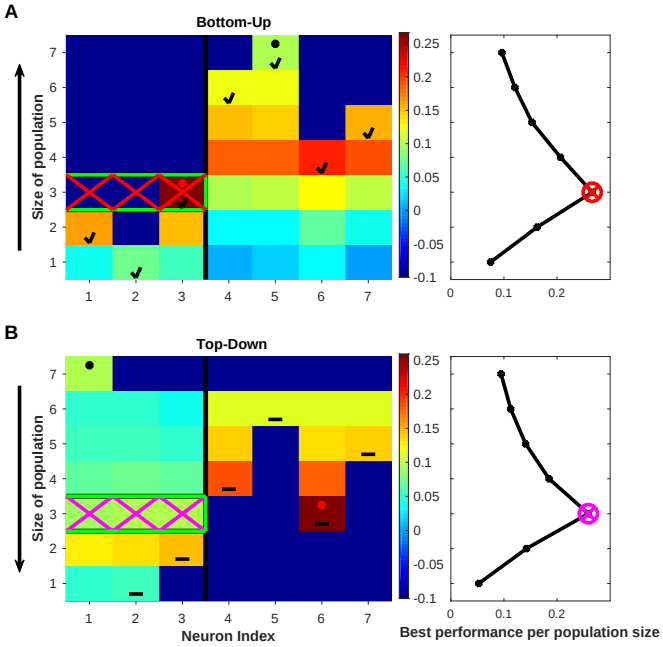

Figure 5.3: Summed population coding: Color-coded discrimination performance for different neuronal subpopulations within the example from Figs. 5.1 and 5.2, in which a subpopulation consisting of the first 3 out of $N = 7$ neurons code for the different stimuli while the remaining non-coding neurons fire just randomly. Every matrix element depicts the performance of one specific subpopulation. The right panels show the maximum performance for a given subpopulation size. A. The bottom-up algorithm starts with the discrimination performances of the individual neurons. In every subsequent iteration one adds the neuron that complements the current subpopulation best (indicated by small black ticks) and this is repeated until the full population size is reached. B. The top-down algorithm begins with the discrimination performance of the complete population (depicted in the first column of the top row). In every iteration, one discards the neuron that contributes the least to the discrimination (marked by short black horizontal lines) until just one neuron remains. — The red and the black dots in both matrices mark the coding and the full subpopulations whose pairwise distance matrices are shown in Fig. 5.2A and 5.2C, respectively (the non-coding subpopulation in Fig. 5.2B is never visited). For both algorithms, the maximum overall population sizes (red circle in A, magenta circle in B) is correctly obtained for the coding subpopulation (marked by crosses in red resp. magenta and confirmed by the green rectangles indicating the ground truth results of the brute force approach).

which is much smaller than K_{bf} and, thus, feasible even for very large N. For $N = 125$, the individual terms from different subpopulation sizes add up to only $K_{\mathrm{grad}} = 7875$.

(iii) Simulated annealing

Simulated annealing (61) is a heuristic approach, which – in principle – allows to find the global maximum without having to explore the whole search space, though there is no guarantee that the optimum solution will at all be found and, if so, that this will be in fewer steps than the brute force search. However, simulated annealing, in contrast to the two gradient algorithms, has the ability to recover from local maxima. Suboptimal solutions are hence much less likely to occur.

One uses a random permutation of neurons as an initial subpopulation $\mathcal{K}_0$. The n-th step in the search is to add or remove a randomly chosen neuron to or from the current subpopulation $\mathcal{K}_{n-1}$ resulting in a new population $\mathcal{K}_n$. Addition or removal is applied with equal probability, except for the boundary populations of one neuron and all neurons, for which the only possible steps are to add or to remove a neuron, respectively. Whether or not the addition/removal is accepted depends on the new discrimination performance P_n relative to the current one P_{n-1}. The corresponding acceptance probability is set by

$$q_n = \exp\left\{-\frac{|P_n - P_{n-1}|}{\mathcal{T}_{n-1}}\right\}, \tag{5.8}$$

where $\mathcal{T}_{n-1}$ is a pseudo-temperature that allows moving 'downhill' in order to not get stuck in a local and thus suboptimal maximum. Steps with $P_n > P_{n-1}$ are always accepted. The likelihood of accepting steps with $P_n \leq P_{n-1}$ is also finite but decreases according to a gradual and stepwise cooling scheme in which $\mathcal{T}_n$ is held constant for a certain number of iterations chosen depending on the number of neurons.

By means of a path of N_0 random test steps from the starting population one can set the initial temperature to

$$\mathcal{T}_0 = -\tfrac{1}{\ln(0.95)} \left\langle |P_n - P_{n-1}| \right\rangle_{n=1,\ldots,N_0}, \tag{5.9}$$

which guarantees a fair mobility in the beginning because even downhill steps are accepted with a likelihood of 95% (131). The stopping criterion for the algorithm is that between two successive temperature changes the population remains unchanged, i.e. $\mathcal{K}_n = \mathcal{K}_{n-1}$. During the whole iteration one tracks the highest discrimination performance value reached thus far and in case the final value is worse than this best value along the path, it can not be the global maximum and thus the algorithm resets the temperature to $\mathcal{T}_0$ and continues with increased mobility.

5.4.2 *Labeled Line (LL) – Discrimination Performance & Algorithm*

The assumption underlying LL coding is that neurons individually encode different properties or features of a stimulus. Hence, in this case every neuron must be evaluated separately. This actually makes things much easier since instead of having to deal with distance matrices in the space of all possible subpopulations as in the SP case, it is now sufficient to calculate only N distance matrices, one for each neuron. A single neuron typically codes for one specific stimulus feature only. Therefore, in order to discriminate a large and broad set of S stimuli, the complementary information provided by many neurons needs to be combined. Identifying the most discriminative LL population is thus equivalent to finding discriminative neurons for as many stimulus pairs as possible. For two stimuli to be distinguishable there must be at least one individual neuron sensitive to their difference. In case more than one neuron is found for a stimulus pair, the most discriminative one is selected.

Fig. 5.4 illustrates this procedure using a very schematic and simplified example. For clarity we use two very distinct features (color and vehicle type), but in a typical recording these often would be two different features of the same sensory mode. Our starting point is a set of S different stimuli (Fig. 5.4A) whose repeated presentations elicit $T = S \times R$ spike train responses (Fig. 5.4B). From these responses a pairwise distance matrix D is computed for every individual neuron $n = 1, ..., N$ (Fig. 5.4C). Looping over stimulus pairs transforms each of these $T \times T$ distance matrices into a $S \times S$ discrimination matrix that indicates the stimulus pairs this particular neuron is able to discriminate. For two different stimuli s and $\bar{s}$ to be distinguishable, their intra-stimulus distances $D^{s,s}$ and $D^{\bar{s},\bar{s}}$ and their inter-stimuli distances $D^{s,\bar{s}}$ (both pooled over all repetitions of the respective stimuli) should stem from different distributions. If they were to stem from the same distribution, the two stimuli could not be discriminated.

We seek to verify the hypothesis that responses to two different stimuli can be discriminated. To this end, we employ three two-sample t-tests $t^{\bar{s},\bar{s}}_{s,s} = t\,(D^{s,s}, D^{\bar{s},\bar{s}})$, $t^{s,\bar{s}}_{s,s} = t\,(D^{s,s}, D^{s,\bar{s}})$, and $t^{s,\bar{s}}_{\bar{s},\bar{s}} = t\,(D^{\bar{s},\bar{s}}, D^{s,\bar{s}})$ at a significance level of $\alpha = 0.001$. From these three tests one can form a logical discrimination matrix such that for neuron n the discrimination between stimuli s and $\bar{s}$ reads

$$M^{s,\bar{s}}_n = \begin{cases} 1 & \text{if } t^{\bar{s},\bar{s}}_{s,s} \vee t^{s,\bar{s}}_{s,s} \vee t^{s,\bar{s}}_{\bar{s},\bar{s}} = \mathit{true} \\ 0 & \text{otherwise.} \end{cases} \tag{5.10}$$

The two stimuli can be discriminated by this neuron (i.e. $M^n_{s,\bar{s}} = 1$) whenever at least one of the three tests yields significant differences. In the

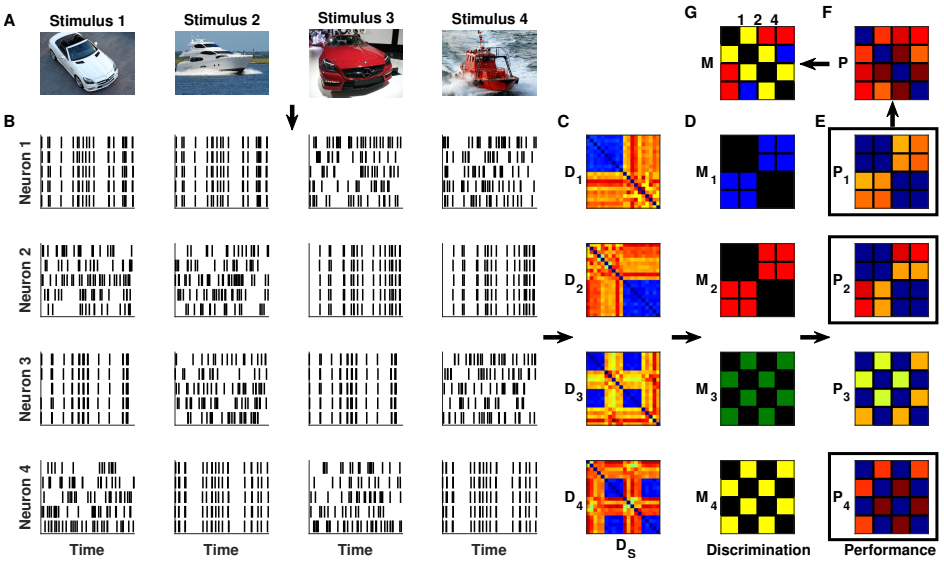

Figure 5.4: Labeled line coding: Schematic example in which each of the $N = 4$ neurons is sensitive only to one specific feature of the different stimuli. Neuron 1 responds to white and neuron 2 to red objects, neuron 3 to cars and neuron 4 to ships. A. The $S = 4$ stimuli were chosen such that they combine these features. B. Spike trains responses of each neuron to $R = 5$ repetitions of every stimulus. C. Pairwise distance matrices D_n (using the SPIKE-distance D) over all $T = 20$ trials. D. Corresponding discrimination matrices M_n. Black is always 0, whereas each neuron has its own color representing 1 for the stimulus pairs it can distinguish. E. Performance matrices P_n. The value is zero for stimulus pairs that can not be discriminated and otherwise the higher the better the discrimination. F. The population performance matrix P collects for each stimulus pair the highest values obtained for any of the four individual neurons. G. The corresponding neurons are indicated in the population discrimination matrix M which is colored according to which neuron achieves the best discrimination performance ($P_n^{s,\bar{s}}$) for that stimulus pair. The overall performance is $P^{LL} = 0.148$ and is obtained for the optimized LL population $\mathcal{K}_{\text{opt}}^{\text{LL}} = [1\,2\,4]$ (written on top, the corresponding performance matrices are marked in E by black rectangles). Note that the color coding in both subplots D and G is discrete and used to label the individual neurons.

example of Fig. 5.4, all neurons respond to just one single feature of the stimuli, the first two to color (white and red) and the last two to vehicle type (car and ship). Thus, all of the neurons are able to discriminate among some of the stimuli but not among others (Fig. 5.4D).

Next, in order to identify the best LL subpopulation for discrimination, we define the discrimination performance for each stimulus pair $(s, \bar{s})$ and every neuron n as

$$\hat{P}_n^{s,\bar{s}} = M_n^{s,\bar{s}} P_n^{s,\bar{s}} \tag{5.11}$$

with $P_n^{s,\bar{s}}$ given in Eq. 5.1, supplemented by the subscript n to index individual neurons. High values of $\hat{P}_n^{s,\bar{s}}$ are obtained for large inter-stimuli and small intra-stimulus distances, while for the stimuli pairs a neuron cannot discriminate the value vanishes (cf. Fig. 5.4E). From these individual discrimination performance matrices the population performance matrix can be obtained as

$$\hat{P}_{\max}^{s,\bar{s}} = \max_n \left\{ \hat{P}_n^{s,\bar{s}} \right\}, \tag{5.12}$$

which takes for every stimulus pair the best discrimination performance of all the individual neurons (see Fig. 5.4F). The population discrimination matrix

$$M_{\max}^{s,\bar{s}} = \begin{cases} \arg \hat{P}_{\max}^{s,\bar{s}} & \text{if } \hat{P}_{\max}^{s,\bar{s}} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.13}$$

indicates for every stimulus pair the best neuron (cf. Fig. 5.4G) and from this matrix the optimized LL population is obtained by uniting all neurons that contribute to the discrimination, i.e.

$$\mathcal{K}_{\text{opt}}^{\text{LL}} = \bigcup_{s \neq \bar{s}, M_{\max}^{s,\bar{s}} > 0} M_{\max}^{s,\bar{s}}. \tag{5.14}$$

Finally, the LL discrimination performance of the full population for the whole stimulus set is the mean of the discrimination performances over all stimulus pairs, that is,

$$P^{\text{LL}} = \left\langle \hat{P}_{\max}^{s,\bar{s}} \right\rangle_{s \neq \bar{s}}. \tag{5.15}$$

In our example, from Fig. 5.4G we can extract the best selection from the two color neurons (both neuron 1 and neuron 2) and from the two 'vehicle type' neurons (just the ship neuron, neuron 4) yielding neurons 1, 2, and 4 as the optimized LL population which obtains an labeled line discrimination performance of $P^{\text{LL}} = 0.148$.

5.5 RESULTS

5.5.1 *Summed Population (SP)*

(i) Brute force

This is the preferred algorithm because it guarantees that the best discrimination performance is found. However, the number of subpopula-

tions that have to be evaluated increases exponentially with the number of neurons, see Eq. 5.6, rendering this algorithm applicable only for comparably small numbers of neurons. In this study whenever possible we used it as benchmark to verify the correctness of the solutions found by the other algorithms and to evaluate their decrease in computational cost. Being able to obtain the ground truth this way was most important for the examples with noisy spike trains where the actual result was not known beforehand.

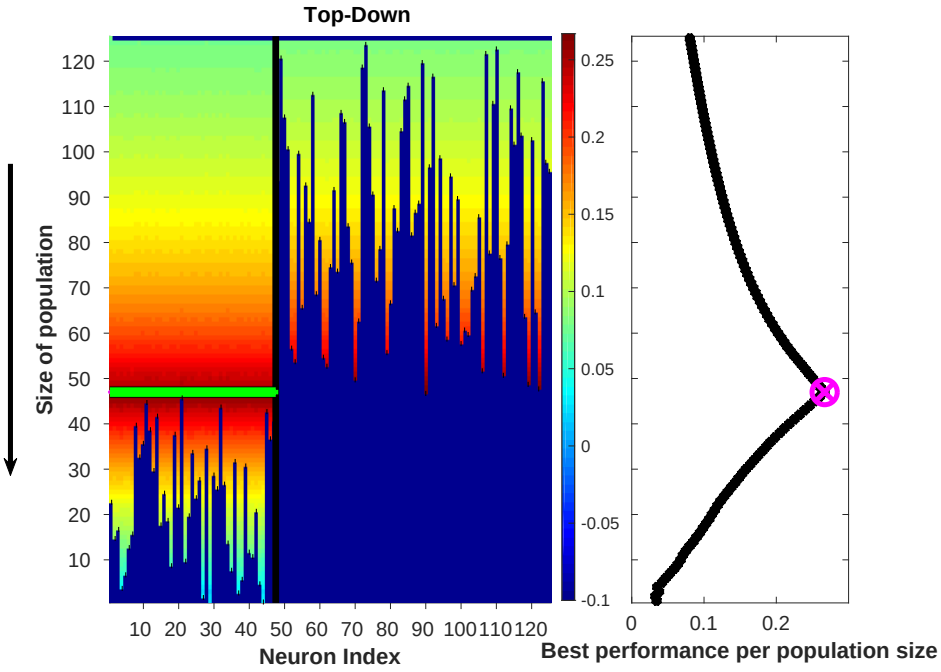

Figure 5.5: Summed population coding: Same setup and format as Fig. 5.3B but this time the top-down gradient algorithm was applied to a neuronal population of size $N = 125$ which corresponded to a total number of more than 10^{37} possible subpopulations. In this simulation the subpopulation consisting of the first 47 neurons (marked in green) coded for the different stimuli while the remaining non-coding neurons fired randomly. Even though it evaluated just less than 8.000 subpopulations, the algorithm correctly identified the coding subpopulation as indicated by the maximum over all population sizes (magenta circle in curve on the right).

(ii) Gradient algorithms

We illustrate the appropriateness of using gradient algorithms, i.e a proof-of-principle, using a noise-free case. In Fig. 5.5 we used a similar ex-

ample as in Fig. 5.3, again with $S = 4$ different stimuli which were repeated $R = 5$ times each. This time, however, the population consisted of $N = 125$ neurons, a number in the range of real life experiments. This corresponds to more than $K_{\mathrm{bf}} = 4 \cdot 10^{37}$ possible subpopulations and thus renders the brute force approach absolutely unfeasible. For the gradient algorithm (top-down variant) the discrimination performance had to be determined for only 7875 subpopulations. The first $c = 47$ coded perfectly and this coding subpopulation was correctly identified as the one with the maximum discrimination performance $P^{\mathrm{SP}}_{\mathcal{K}_{\mathrm{opt}}}$.

Next, we ran two simulations that are constructed such that each time one of the two variants of the gradient algorithm did not find the best subpopulation since it got trapped in a local maximum. The two cases employed essentially the same setup, but the first case was noiseless whereas in the second case we applied noise that disrupted the timing information of each spike with 50% chance. For the first simulation we generated a population of $N = 10$ neurons made up of three different subpopulations. The first four individual neurons are each able to discriminate between all the stimuli on their own. Next, a subpopulation of three neurons could discriminate only collectively, i.e. as a population, but with a slightly lower discrimination than the individual neurons. Finally, the last three neurons were non-coding. For a population of this size the brute force approach is still feasible and its result served as ground truth.

In the simulation without noise shown in Fig. 5.6, the best discrimination performance should have been obtained by the very best individual neuron. This was indeed the result of the bottom-up variant (Fig. 5.6A). However, the top-down variant failed and erroneously indicated the neuronal subpopulation consisting of the middle four neurons as the winner (Fig. 5.6B). This is because it had to follow the iterative procedure of singling out one neuron at a time for elimination and, hence, could not treat the middle population as a single entity. Therefore, at each step breaking up the population was being considered a very bad option and falsely avoided to the very end. The individually coding neurons were eliminated first and thus never evaluated on their own, which left the performance of the collectively coding subpopulation as the best one encountered along the path.

Fig. 5.6C depicts the SP discrimination performances given in Eq. 5.4 calculated for each of the individual neurons. Each of the first three neurons had a very large discriminative power far superior to the individual neurons of the collectively coding subpopulation which were still better than the non-coding neurons. In Fig. 5.6D we show the winners chosen by each of the different algorithms. The top-down gradient variant was

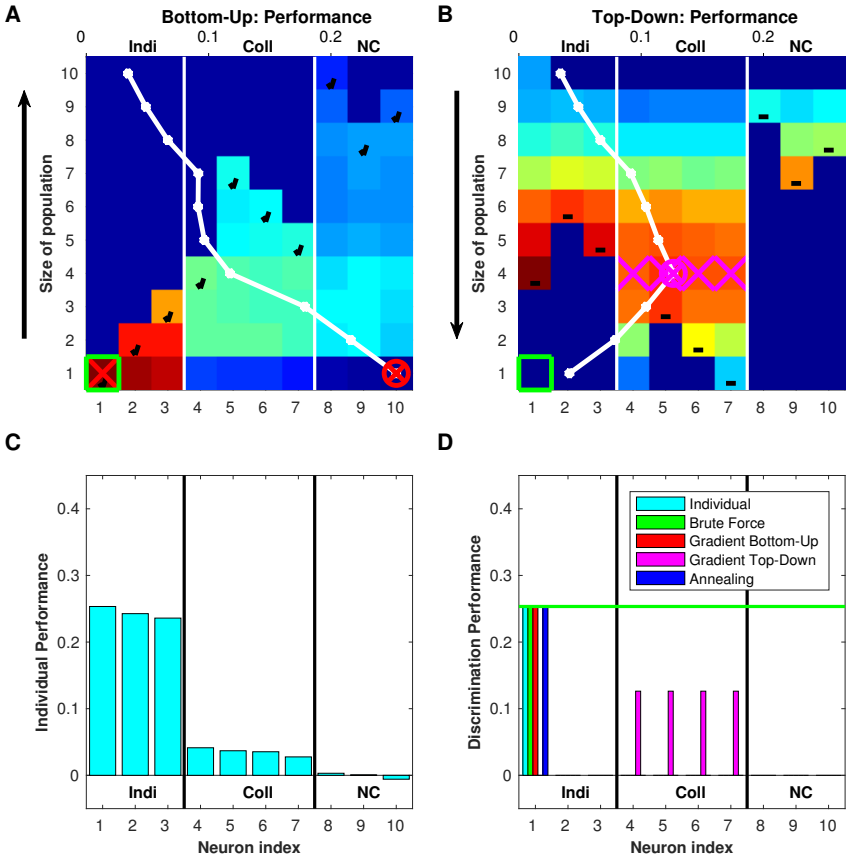

Figure 5.6: Summed population coding: Example where the top-down gradient algorithm failed. The population consists of three neurons that code the different stimuli individually (Indi), four neurons that code them as a collective (Coll) and three non-coding neurons (NC). The top panels A and B follow the setup of the bottom-up and the top-down algorithms in Fig. 5.3A and B, respectively. For the sake of legibility now the curves of the maximum performance per population size are superimposed in white (axis on top). The optimal solution is the single best individually coding neuron (neuron #1 on the left) and this neuron was indeed correctly identified by all algorithms apart from one. The top-down algorithm always discarded the neuron contributing the least and since it could not discard the whole collective unit it got stuck in a local maximum. C. Performance of individual neurons. D. Neuronal subpopulations identified as winners by the different algorithms. The horizontal green line indicates the optimal discrimination performance as verified by the brute force algorithm.

the only algorithm that did not succeed in identifying the very first indi-

vidual neuron as the most discriminative subpopulation (as verified by the brute force approach).

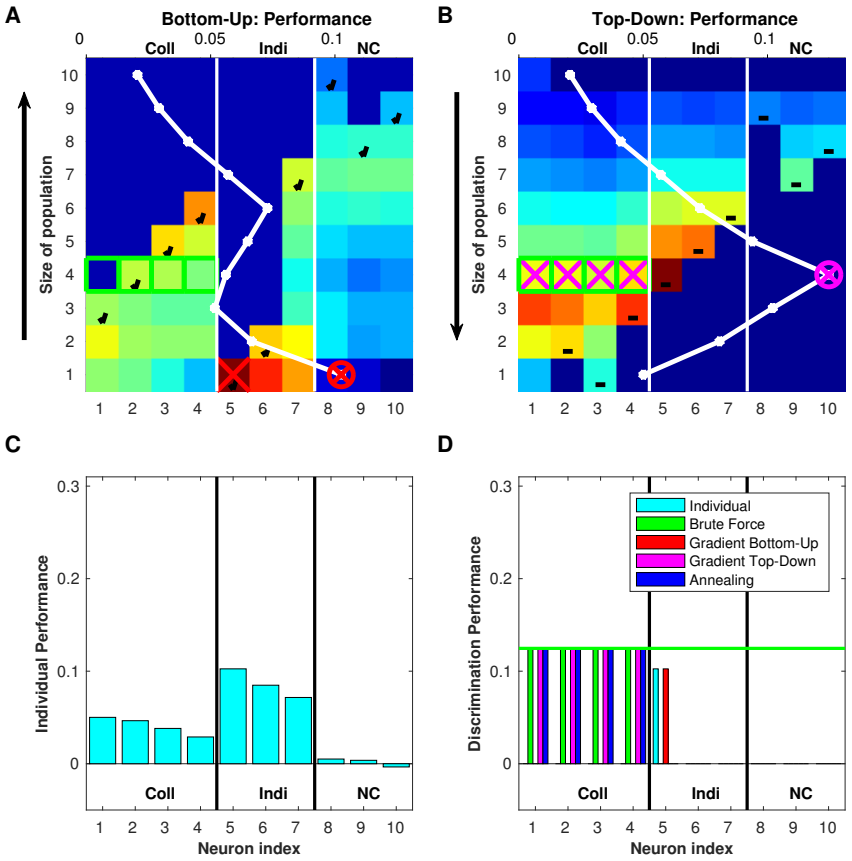

Figure 5.7: Summed population coding: Similar to Fig. 5.6 but in this example we added some noise which made the bottom-up gradient algorithm fail. The discriminative performance of the individual neurons was degraded to such an extent that here the four summed population neurons performed better (to stay with our convention that the winning subpopulation comes always first, we reversed the order of these two groups of neurons). The bottom-up algorithm could never add the SP-population as a whole and thus always had to pick the best individual neuron remaining. This took it to a local maximum right at the beginning, and from then on the discrimination performance of the very first individual neuron apparently remained the optimal solution along the whole path.

In the second simulation summarized in Fig. 5.7, we used exactly the same setup but added so much noise to the first three individually coding neurons that they did no longer outperform the four collectively coding neurons which thus together should have been identified as the most discriminative subpopulation. In this case the bottom-up gradient

algorithm failed (Fig. 5.7A), whereas the top-down variant managed to find the correct solution (Fig. 5.7B). The reasoning is exactly inverse to the first case. The iterative 'one-neuron at a time' scheme did not allow to include the collectively coding population consisting of four neurons in one step and since each of them on their own was worse than the three individually coding neurons (see Fig. 5.7C), these latter neurons were added first. Again, Fig. 5.7D summarizes the results of the different algorithms. There, the bottom-up gradient algorithm incorrectly identified the best individual neuron as most discriminative.

The respective failures of the top-down and the bottom-up variants in these two simulations were both due to the fact that the gradient algorithms follow a steepest ascent approach where at each step they take the locally optimal choice. The well-known problem with this *greedy* approach is that it does not necessarily lead to the global optimum. In this specific context it meant that once an incorrect neuron was added (bottom-up) / a correct neuron was excluded (top-down), the gradient algorithms had no way to correct this 'mistake' and these bad choices remained. We would like to add that we also found cases in which neither variant was able to find the correct solution. So running both algorithms and picking the overall best performance can also not provide a guarantee that the optimal solution is found.

(iii) Simulated annealing

Since gradient algorithms are much faster than the brute force approach and successful under idealized conditions, they can be used for first testing. However, our examples illustrate that they can generally not be relied upon in more realistic settings. Fortunately, simulated annealing provides a recovery mechanism that considerably reduces the likelihood of getting stuck in a local maximum.

Both Fig. 5.6D and Fig. 5.7D include the successful simulated annealing algorithm. In these two cases as well as in all other cases that we looked at, simulated annealing was indeed able to identify the most discriminative neuronal subpopulation. However, this increased reliability of the simulated annealing algorithm compared to the two gradient variants comes with a prize, an increased computational cost.

The runtime of an algorithm consists mostly of two parts, the number of subpopulations that have to be evaluated and the time it takes to evaluate each of these subpopulations. As summarized in Fig. 5.8, we compared the different algorithms regarding the numbers of subpopulations visited. As expected, for the brute force algorithm that number

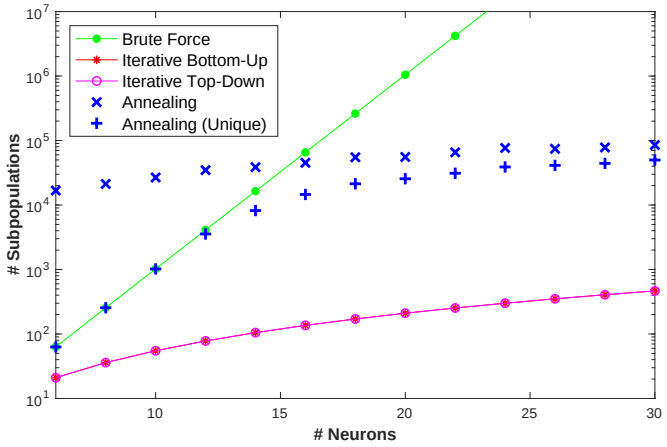

Figure 5.8: Comparison of the computational cost for the different algorithms for summed population coding: Subpopulations evaluated versus number of neurons. Note that the values for simulated annealing are averages over ten trials.

increased exponentially with the number of neurons. For the two variants of the gradient algorithm the numbers were identical in line with Eq. 5.7, and much smaller than for the brute force algorithm. In the case of simulated annealing, our selected examples revealed values in between these two extremes for large enough numbers of neurons. Here we had to divide the number of subpopulations evaluated into two distinct categories. The first one is the actual number visited by the random walk, the second one is the number of uniquely evaluated subpopulations. We added the latter because some of the subpopulations might have been revisited many times. We note that in our implementation the discrimination performance for every subpopulation is determined only at the first visit and stored so that at all repeated visits this value can be readily retrieved via an unambiguous mapping to the subpopulation space. This makes revisiting a subpopulation considerably cheaper than calculating a new one. While for small population sizes the actual number of subpopulations evaluated might be even higher than for brute force, it starts to gain more speed compared to the brute force approach as soon as the algorithm no longer has to visit all solutions. This also implies that for a small number of neurons it is always preferable to use the brute force approach, because it not only gains speed, but, unlike the simulated annealing, also guarantees the ground truth solution.

5.5.2 *Labeled line (LL)*

For the LL case, we investigated a more complex example than Fig. 5.4. However, we used a schematic design to sketch a general idea of some of the complications that may occur. In this example $R=5$ repetitions of $S=8$ stimuli were presented to $N = 10$ neurons (Fig. 5.9).

The coding and discrimination capabilities of every individual neuron can be seen best in the subplots of Figs. 5.9A and 5.9B. Neurons 1 and 2 both coded for the first four of the eight stimuli, but neuron 1 seemed to respond to one feature common in all of these stimuli whereas neuron 2 responded to each of these stimuli in a different way (Fig. 5.9A). This implies a sensitivity to four different features that were present in just one of these stimuli each. Because of this, only neuron 2 was able to distinguish between these four stimuli themselves, whereas both neurons were able to discriminate between any of the first four stimuli and any of the last four (compare M_1 and M_2 in Fig. 5.9B). Neurons 3 and 4 were each sensible to two of these four features (neuron 3 equally and neuron 4 differently) and neurons 5 and 6 to just one. So among the fist six neurons there was a hierarchy of specificity with neurons 1 and 2 the least and neurons 5 and 6 the most specific. All these neurons also showed different levels of reliability, which caused that only three of them (neurons 1, 2 and 4) obtained maximum discrimination values for some stimulus pairs (Fig. 5.9, C and D) and thus contributed to the optimal LL population (Fig. 5.9E). An even higher level of redundancy could be observed for neurons 7 and 8, which both coded exactly for the same two stimuli, but here only neuron 8 was reliable enough to enter the optimal LL population. Neuron 9 did not respond to any of the stimuli, i.e. this neuron was either just noisy or sensitive only to one (or more) feature(s) that were not present in any of the stimuli.

In the simpler example of Fig. 5.4, all off-diagonal elements of the population performance matrix P were positive which means that all pairs of stimuli could successfully be discriminated. The optimized LL population consisted of three neurons, one vehicle type neuron and two color neurons, which came about because both color neurons happened to be most discriminative for some of the stimulus pairs. Here it would take only two neurons to perfectly discriminate, as long as a color neuron was combined with a vehicle type neuron. This nicely illustrates the subtle distinction between coding and discrimination: with only the 'white' neuron 1 and the 'car' neuron 3 all stimuli could be discriminated even though none of these two neurons actually coded for stimulus 4, the 'red ship'. While this contained a case of discrimination without coding, exactly the opposite case occurred in Fig. 5.9. There, for stimulus pair 7

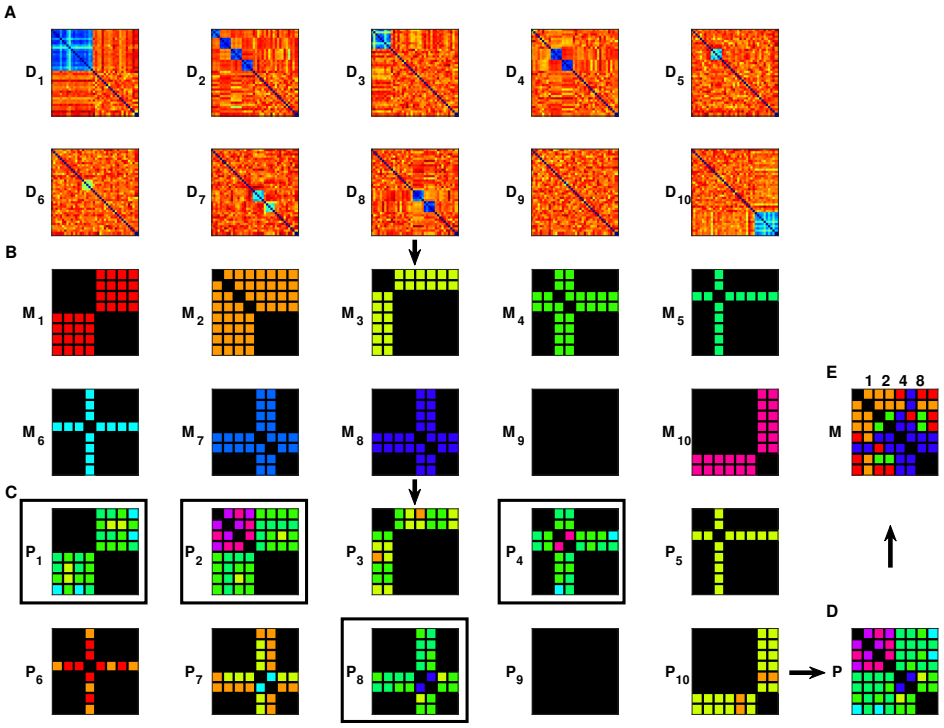

Figure 5.9: Labeled Line Coding: A reasonably complex example based on $R = 5$ repetitions, $S = 8$ stimuli and $N = 10$ neurons. A. Pairwise spike train distance matrices D_n over all $T = 40$ trials. Both neurons 1 and 2 responded to the first four stimuli, but whereas the first reacted with just one common response, the second did so with a different response for each of these stimuli. Neuron 3 was only sensitive to the first two (equally) and neuron 4 only to the third and fourth stimulus (separately) who were also covered by neuron 5 and 6 but only one at a time. The two neurons 7 and 8 coded for the fifth and sixth stimulus but with varying degrees of reliability, while neuron 9 did not respond to any of the stimuli. Neuron 10 was sensitive to the last two stimuli reacting to both of them with the same response. B. Discrimination matrices M_n, colored such that the stimuli pairs that could be discriminated are marked with a unique color for each neuron. C. Corresponding performance matrices P_n that were used to identify the best performance for each stimulus pair. D. Maximization of the P_n-matrices results in the population performance matrix P. The black off-diagonal element indicates stimuli 7 and 8 which could not be discriminated by any of the neurons of this population. E. The population discrimination matrix M gathers the neurons that contributed to the optimized LL subpopulation $\mathcal{K}^{\mathrm{LL}}_{\mathrm{opt}} = [1\,2\,4\,8]$ (again marked in C by black rectangles). Their overall performance in this example was $P_{LL} = 0.139$.

and 8 we obtained a zero outside of the diagonal indicating that these two stimuli could not be discriminated by this set of neurons, although neuron 10 was responding to both of these stimuli. Therefore, this is a case of coding without discrimination. In general, two stimuli cannot be discriminated whenever the population contains only neurons that are not sensitive to any of the features that distinguish them. Some of the neurons might still code the stimuli but just not the distinction. In Fig. 5.4, a red car and a red ship could not be discriminated if only 'color' neurons were evaluated.

5.6 DISCUSSION

5.6.1 *Summary*

We evaluated methods for identifying the most discriminative subpopulation when looking at neuronal responses to repeated presentations of a set of stimuli. Central to our studies were two different assumptions about neuronal population coding. According to the *summed population (SP) hypothesis* all neurons in a population potentially contribute to the coding of an external stimulus, whereas under the *labeled line (LL) hypothesis* stimulus encoding is realized through individual neurons. In both cases our analysis relied on the computation of pairwise distance matrices and a basic discrimination performance that quantifies the degree to which identical (different) stimuli yield similar (dissimilar) spike train responses. However, since the two hypotheses include different assumptions about neuronal coding they required complementary algorithmic approaches.

For SP we compared three approaches that search the space of all possible subpopulations for the maximum discriminative performance over all stimulus pairs: (i) A *brute force* search evaluating all possible subpopulations; (ii) two variants of a steepest ascent or *gradient algorithm*; and (iii) *simulated annealing*. By definition, approach (i) provides the best results as evaluating all possible combinations guarantees that the global maximum is found. However, the number of possible subpopulations increases exponentially with the number of neurons (Eq. 5.6), rendering a brute force approach feasible only for rather small populations. Gradient algorithms (ii), like the one used in (129), overcome this limitations, as the number of subpopulations that has to be evaluated increases only quadratically with the number of neurons (Eq. 5.7). Unfortunately, while useful under very idealized conditions, in general these approaches have problems in finding the correct solution. In contrast, our simulated an-

nealing approach (iii) proved much more reliable in finding the maximal discriminative power than both variants of the gradient approach, and this despite only a moderate increase in computational cost in the majority of cases studied here (see Fig. 5.8 for an example).

For LL we introduced an optimization algorithm which finds the most discriminative subpopulation by evaluating every neuron separately. First, for every individual stimulus pair the algorithm identifies the discriminative neurons and selects the best one. These best neurons are then combined to form the optimal LL-population. As shown in Fig. 5.9, the algorithm can handle quite involved coding scenarios, even though its computational complexity is much lower than in the SP case (because we only have to deal with one distance matrix per neuron). Moreover, we are guaranteed to find the best subpopulation since this time no search in a very high-dimensional subpopulation space is needed.

5.6.2 *Comparisons*

To provide a more intuitive understanding of the relationship between the SP and the LL hypothesis, consider a single neuron that in itself is able to discriminate the whole set of stimuli perfectly. Such a neuron is where both hypothesis meet. It could be considered either as a perfect SP population of size one or as an LL neuron that yields a perfect discrimination matrix on its own. While for a few stimuli one neuron alone might indeed be able to do the job, the distinction of a large and complex set of stimuli typically works along several feature dimensions and a population of neurons is needed to work in unison and complement each other. Conversely, in order to obtain a robust and universal subpopulation in an experimental setting, it is essential to test the population on a stimulus set that is as diverse as possible. In general, for both the SP and the LL case one should always keep in mind that the results obtained are a function of both the stimulus set presented and the neuronal subpopulation recorded.

For such a complex stimulus set, SP and LL hypotheses can be seen as two distinct ways neurons as a population may collaborate to carry out the task of the perfect neuron. In the SP case they divide the spikes of each individual response among themselves, while in the LL case the responses to different stimuli are distributed among the neurons. It would also be possible to construct all kind of SP and LL mixtures by combining subdivision of spikes among neurons (SP coding blocks) with separation of stimulus sensitivity between different neurons (LL subpopulations) at any level. However, since this can get arbitrarily complex, a modification

of the algorithms to address these mixed cases seems to be computationally out of reach.

There are other potential modifications. The current LL algorithm identifies for each stimulus pair all the neurons that are sensitive to discrimination and selects only the very best one. Overall this will result in a rather minimal LL-subpopulation. However, in real-life applications additional criteria such as redundancy and reliability might be very important. To create more stability it could be preferable to include more rather than less neurons (130). This could be particularly useful whenever two neurons distinguish the same stimulus pair but use different features to do so. The inclusion of both of these neuron would guarantee access to complementary information that might be essential to discriminate more diverse stimulus sets. Then, both algorithms are designed to identify the most *discriminative* subpopulation which is why we always look at pairs of two stimuli. If instead one were interested in finding the *coding* subpopulation, the algorithms could be modified to evaluate responsiveness to individual stimuli.

For the sake of simplicity we restricted our simulations to cases with uniform rates. However, recent studies on the observed variability in firing rates have emphasized the prevalence of log-normal distributions (132) and the roles of 'soloists and choristers' (133). This raises the question of the relative importance of coding by sparse-firing vs. higher-frequency neurons. In the SP case varying firing rates may imply that the population spike train is divided among the population with a non-uniform probability distribution. However, this will only change the gradient towards the optimal solution and not the solution itself. Thus, our algorithm will not be affected. Likewise, in the LL case, the stimulus pairs are assessed individually and therefore the rate is less important. The neurons that display the most consistently distinct responses for different stimuli will be selected regardless of the actual rates.

Our algorithms employ the SPIKE-distance as a fundamental measure for comparing spike trains. The only other approaches that use spike train distances to evaluate the coding properties of neuronal ensembles are population extensions for the Victor-Purpura distance (125) and the van Rossum distance (126). These population measures interpolate between the two extreme cases of summed population and labeled line coding by means of a parameter that determines the importance of distinguishing spikes fired in different cells (minimal for SP, maximal for LL). While the applications to the pooled spike train of the full population (SP) or to all individual spike trains separately (LL) are straightforward ((134) resp. (135)), it is not obvious how to interpret intermediate cases. More importantly, these approaches never deal with subpopulations but

always consider the population as a whole. This hampers comparison of results because these population extensions have not been designed to answer questions about the extent to which a part of the population contributes to stimulus discrimination and much less to identify the most discriminative subpopulation. In short, our algorithms and these population extensions address complimentary questions.

5.6.3 *Outlook*

So far we have applied both the SP and the LL algorithms to simulated datasets only. We either knew the ground truth a priori or we could apply the brute force approach to obtain the ground truth. After having passed this verification test, the next step will be to analyze experimental datasets in which a neuronal population is recorded upon repeated presentations of a set of stimuli. These can be data from animal models in both sensory and motor regions, but also recordings of intracranial neuronal spiking from patients undergoing seizure monitoring prior to epilepsy surgery (136).

Once the algorithms are applied to experimental data their results may serve to address further fundamental questions about neuronal coding. One of them regards the size of the most discriminative subpopulation and how it compares to the size of the full population and to the size of the subpopulation that conveys the same information as the whole (see, e.g., the information theoretic analysis carried out in Ince et al., 2013). Moreover, it would be interesting to investigate the spatial location of the discriminative neurons and search for properties that distinguish them from other neurons. For example, one may evaluate their overall firing rates and their level of connectivity (132) as well as their coupling to the overall firing of the population (133).

Another potential application for our algorithms are BCIs (see, e.g., (137, 138)), in particular, the kind of BCI that works with multi-unit spike train read-out (139). Current BCI systems are following the so-called mass-effect principle and rely on rather crude population averages like mean firing rates over many neurons (140). This could be improved significantly by increasing the signal-to-noise ratio (SNR) by selecting the most informative neurons (130) and making explicit use of the temporal structure of spike trains (141). Algorithms for finding the most discriminative subpopulation based on the spike timing sensitive SPIKE-distance could lead to refined and more targeted estimates of ensemble activity. Comparing the SP and LL algorithms on the same dataset may provide further insights on how neural circuits encode. In fact, the success or fail-

ure of clinical BCI applications will depend on our efforts to understand population coding.

6

EPILOGUE

6.1 ON THE CONCEPT OF SIMILARITY

While I was studying Bioengineering at Tampere University of Technology, I had this revelation in one of the classes. The lecturer showed us a slide of an opened radio with tags for each of the components and their descriptions.*"This is how we perceive a radio. We have defined the concepts and constructed the device based on our knowledge of how each individual component works."* Next the lecturer showed us a slide of the same radio with the tags removed. *"Now let us imagine that the radio is a biological system. We have very little prior knowledge. How do we approach the complexity of a system that we do not know much about? Obviously if we have only one radio, we cannot do much in fear of breaking it. However, if we had more radios, we can start removing parts one by one."* In the slide, the tags started to get new names. *"DNTN: Do Not Touch Node, VN: Vital Node, NVIN: Not Very Important Node"*. This made sense, since we are essentially reverse-engineering the system. But then it hit me and I realized something even more vital. No matter how far we dissect the radio, we will always use some frame of reference that we are familiar with. In my case the system I was studying was already the brain. Most importantly, since we are talking about the brain, there can be the unconscious assumption that it bears similarity to our own electric circuit constructions, in which you have a conductor between elements and any information transmitted in the conductor is coded and can be read in between. Does the brain really work that way? Are neurons but conductors and computation elements, or are spikes just the result of the computation done on some other level?

Using similarity in rate or spike timing is an essential part in our attempt to assess neuronal coding. We must, however, always be critical towards the methods used, since similarity is not a well defined concept. Rather there are many ways to define what is similar. Which of the three are the most similar with each other, a mouse, an apple, and a pig? Do we look at size? Speed? Colour? Genetic sequence? The basis of similar-

ity in our case lies by far in the assumption that we know how the brain transmits information. There is evidence of both rate and timing being involved in the process. However, these two concepts are intertwined on a very basic level. If a neuron is silent and needs to transmit information via spikes and it does that with a reliable spike pattern, will the transmitted information be in the increased rate or in the spike timing, or will the increased rate and timing be a direct result of how the previous neuron layer had been activated, which again was activated by the previous, and so on? It might be that the coding is, for example, in the connections and synaptic strengths and the spike pattern similarity that we observe is but a consequence, since we are not aware of a neuron that functions in time scales that would be able to "read" the spike pattern over time. Even if we identify a coding neuron, the next one will still just "see" near instantaneous changes in the membrane potential. The time scales in which neurons primarily operate and the time scale of the spike pattern that should contain the information do not match. Thus, our way of separating rate and timing could be completely alien to the neurons. In fact, it might be that both concepts are but derived for us to understand what is going on.

This is not a totally new idea. There is plenty of literature about how neuronal connections may affect network behaviour (see e.g. (142) and references therein) as well as how the synapses and Glia-cells interact at synapse and local level (143). The point I am trying to make is that when we consider neuronal coding, it is not the neuron that codes anything. It is merely responding to stimulation it gets over time. If there are similarities in the spike pattern elicited by a neuron, it is more a property of the underlying input network than the neuron itself. The main difference in this respect between a single neuron and the network is memory. A single neuron can only have a very limited memory of the spikes it has elicited in the past, yet a network consisting of more neurons and connections can have this capability.

Then one may ask how the network is utilizing this capability. If you have a robust network response, in which the timing of the spikes will always be similar given a certain initial activation pattern, is this time coding? If so, what will happen if one of the neurons is lost or connections change? If the network is less robust and as a consequence the spike pattern looks a little different each time, but activity in certain neuron layers is similar, is this rate coding? How about if and when the neuron we are recording is part of a network layer itself and thus it is but a single unit among many? If and when the neuron has connections back to the previous layer thereby influencing the spike pattern it will receive in the next moments? Perhaps we are approaching the problem

through concepts that are more easy for us to grasp in order to create order within chaos?

6.2 USING THE ALGORITHMS

Even if the spike train patterns that we can record from single neurons are not a result of the neuron itself coding something, it can relay information of how the underlying network has responded as well as what is being relayed forwards (or backwards). I any case the information is present in the spike pattern for us to decode. Thus, measures such as spike train distances may be used to assess correlations in neuronal coding as long as one does not mix the measure with how the neurons actually work. There exist plenty of metrics that can tell us about economic situations without being actually money moving around, or how to reward people for a job well done by defining what we consider to be a good job, like measuring scientific quality by number of publications. An important thing about metrics is that you will get what you measure. No more, no less.

In the past few years we have expanded existing measures and introduced new ways of assessing spike trains. These approaches can be readily applied in neuroscience but also in fields such as brain-computer interfaces and robotics. For example for prosthesis control discrimination between different states is exactly what is needed. Our expanded measures can be used for assessing similarity in multiple time scale systems and for providing additional information about leader-follower relationships.

The adaptive extensions of the measures that I explained in Chapter 2 provide a way to limit the time scale free nature of the measures. This is especially useful when there are multiple time scales in the data. The parameter does not introduce a single time scale like those of the parameters of the Victor-Purpura distance and the van Rossum distance, but rather sets a limit after which differences are assessed based on the minimum relevant time scale. However, these extensions should be used with care, since while they are able to provide additional information, they pose very strong limitations for how to use the measures. First of all assessing the correct time scale manually can be difficult unless one has prior knowledge of the system. Second, the automated extraction can be used, but there one has to be careful not to compare values obtained with different time scale parameters with each other, since values obtained are only valid in relation to the time scale parameter. In this respect these extensions bear similarity with the Victor-Purpura dis-

tance and the van Rossum distance for which one would not compare values obtained with different time scales. Additionally the automated extraction cannot detect the useful time scale parameter. It is simply an estimate based on distributions of ISIs.

The main topic of this thesis concerned neuroscience but in practise there are many fields in which spike train distances can be applied. In Chapter 3 I talked mainly about neuroscience, but in the original publication (9) we also applied the method to climate data. In this thesis the Victor-Purpura distance and the van Rossum distance have been associated with both being spike-resolved and having a time scale parameter, while the ISI-distance and the SPIKE-distance having no parameter and being time resolved. These properties are not linked. The SPIKE-synchronization used for identifying the leader-follower relationships is a time scale free spike-resolved measure. It uses local rate for determining the time scale and assesses similarity purely in spike pairs. Similarly the leader-follower assessment is done under the assumption of robust spike responses and spikes are compared to single spikes in the other spike trains. This combination has a different approach to similarity from all those above and is very useful if the spike generation process can be expected to be robust, as it is equally sensitive to rate as any spike-resolved measure. However, the way it is used for identifying synfire-patterns does not care about the rate mismatch, since the main task for the algorithm is merely to identify spike pairs and then the time order of the spikes in the pairs is used.

The most important thing to consider when applying the methods is to understand what the methods are designed to identify and sometimes what we think they do isn't what they actually do. In Chapter 4 I showed the analysis we did on some of the most used spike train distances and how they behave under different conditions. The time resolved distances perform very well under the assumption that spikes are generated with slight randomness in their timing and number, but not when the number of spikes in the recording window is very low. On the other hand if we assume that each spike is generated with precision and robustness in the spike numbers, also spike-resolved distances will perform well. Their strength is that they are robust in assessing similarity of low rate spike trains, but are not well equipped when one adds noise to the spike generation process and spike count is different from one realization to the next, since they are blind to timing of spikes without a pair in the other spike train. It is very important that one is aware of these limitations when using the methods for neuroscience or even when applying them in other fields.

This is especially important since often for Victor-Purpura distance the parameter value yielding best discrimination performance is thought to provide additional information about the tradeoff between rate and time coding. This is a "dangerous" assumption as I have shown in Chapter 4 and should be used with care. The parameter value indicates timing correlation only for spike trains with very similar rates. Additionally, as noted in the previous Section, one should not mix the way neurons work with the metrics.

So far the spike train distances have mainly concentrated in classifying single neuron responses. Another complementary approach has been made possible by developments in high throughput sensor technology. With the increasing availability of recordings from the same sample means that it might be possible to analyse the responses of multiple neurons together rather than each one individually. There is no conceptual leap from single neuron recordings once we remember that the single neuron response is a a sort of a reflection of input layer population response. The main difference is that in the population robustness on the single neuron level might not be required and thus it would be possible to see discrimination on the population level that is not detectable for a single neuron.

When assuming that a population codes together there are two main approaches that I explained in Chapter 5. The more traditional approach is labelled line, where each recorded unit shows features that alone are enough for discrimination. The other approach is summed population, where the spikes are distributed among the neurons. This is a slightly more complex scenario, since identification of the population that elicits the spikes is computationally heavy. One important thing to notice is that this kind of population coding can only be expected if the population response is robust. If it would be simply rate-response without discriminable spike-pattern, the distribution of rate among multiple neurons would produce redundant neurons that would alone be able to discriminate. Thus the summed population concept only makes sense if one expects that spike timing is important, but which neuron fired the spike is not.

7

REFERENCES

[1] S. Herculano-Houzel, "The human brain in numbers: a linearly scaled-up primate brain," *Frontiers in Human Neuroscience*, vol. 3, p. 31, 2009.

[2] M. F. Bear, B. W. Connors, and M. A. Paradiso, *Neuroscience: Exploring the Brain*. Philadelphia, PA, USA: Lippincott Williams & Wilkins, 2007.

[3] T. Kreuz, J. S. Haas, A. Morelli, H. D. I. Abarbanel, and A. Politi, "Measuring spike train synchrony," *Journal of Neuroscience Methods*, vol. 165, p. 151, 2007.

[4] T. Kreuz, D. Chicharro, R. G. Andrzejak, J. S. Haas, and H. D. I. Abarbanel, "Measuring multiple spike train synchrony," *Journal of Neuroscience Methods*, vol. 183, p. 287, 2009.

[5] T. Kreuz, D. Chicharro, M. Greschner, and R. G. Andrzejak, "Time-resolved and time-scale adaptive measures of spike train synchrony," *Journal of Neuroscience Methods*, vol. 195, p. 92, 2011.

[6] T. Kreuz, D. Chicharro, C. Houghton, R. G. Andrzejak, and F. Mormann, "Monitoring spike train synchrony," *Journal of Neurophysiologyogy*, vol. 109, p. 1457, 2013.

[7] T. Kreuz, M. Mulansky, and N. Bozanic, "SPIKY: A graphical user interface for monitoring spike train synchrony," *Journal of Neurophysiology*, vol. 113, p. 3432, 2015.

[8] E. Satuvuori, M. Mulansky, N. Bozanic, I. Malvestio, F. Zeldenrust, K. Lenk, and T. Kreuz, "Measures of spike train synchrony for data with multiple time scales," *Journal of Neuroscience Methods*, 2017.

[9] T. Kreuz, E. Satuvuori, M. Pofahl, and M. Mulansky, "Leaders and followers: quantifying consistency in spatio-temporal propagation patterns," *New Journal of Physics*, vol. 19, no. 4, p. 043028, 2017.

[10] E. Satuvuori and T. Kreuz, "Which spike train distance is most suitable for distinguishing rate and temporal coding?," *Journal of Neuroscience Methods*, vol. 299, pp. 22 – 33, 2018.

[11] E. Satuvuori, M. Mulansky, A. Daffertshofer, and T. Kreuz, "Using spike train distances to identify the most discriminative neuronal subpopulation," *Journal of Neuroscience Methods*, 2018.

[12] J. D. Victor and K. P. Purpura, "Nature and precision of temporal coding in visual cortex: A metric-space analysis," *Journal of Neurophysiology*, vol. 76, p. 1310, 1996.

[13] M. C. W. van Rossum, "A novel spike distance," *Neural Computation*, vol. 13, p. 751, 2001.

[14] M. Mulansky and T. Kreuz, "Pyspike - A python library for analyzing spike train synchrony," *Software X*, vol. 5, p. 183, 2016.

[15] R. Quian Quiroga and S. Panzeri, *Principles of Neural Coding*. Boca Raton, FL, USA: CRC Taylor and Francis, 2013.

[16] S. Grün, M. Diesmann, F. Grammont, A. Riehle, and A. Aertsen, "Detecting unitary events without discretization of time," *Journal of Neuroscience Methods*, vol. 94, no. 1, pp. 67 – 79, 1999.

[17] T. C. Rabinowitch and A. Knafo-Noam, "Synchronous rhythmic interaction enhances childrenś perceived similarity and closeness towards each other.," *PLoS ONE*, vol. 10, p. e0120878, 2015.

[18] L. Zapata-Fonseca, D. Dotov, R. Fossion, and T. Froese, "Time-series analysis of embodied interaction: Movement variability and complexity matching as dyadic properties," *Frontiers in Psychology*, vol. 7, 2016.

[19] J. D. Victor, "Spike train distance," *Encyclopedia of Computational Neuroscience*, pp. 2808–2814, 2015.

[20] R. Naud, F. Gerhard, S. Mensi, and W. Gerstner, "Improved similarity measures for small sets of spike trains.," *Neural Computation*, vol. 23, p. 3016, 2011.

[21] T. Kreuz, "Measures of spike train synchrony," *Scholarpedia*, vol. 6 (10), p. 11934, 2011.

[22] D. Chicharro, T. Kreuz, and R. G. Andrzejak, "What can spike train distances tell us about the neural code?," *Journal of Neuroscience Methods*, vol. 199, pp. 146–165, 2011.

[23] R. G. Andrzejak, F. Mormann, and T. Kreuz, "Detecting determinism from point processes," *Physical Review E*, vol. 90, p. 062906, 2014.

[24] S. Dura-Bernal, K. Li, S. A. Neymotin, J. T. Francis, J. C. Principe, and W. W. Lytton, "Restoring behavior via inverse neurocontroller in a lesioned cortical spiking model driving a virtual arm," *Frontiers in Neuroscience*, vol. 10, 2016.

[25] A. Espinal, H. Rostro-Gonzalez, M. Carpio, E. I. Guerra-Hernandez, M. Ornelas-Rodriguez, H. J. Puga-Soberanes, M. A. Sotelo-Figuero, and P. Melin, "Quadrupedal robot locomotion: A biologically inspired approach and its hardware implementation," *Computational Intelligence and Neuroscience*, p. 5615618, 2016.

[26] R. Quian Quiroga, T. Kreuz, and P. Grassberger, "Event synchronization: A simple and fast method to measure synchronicity and time delay patterns," *Physical Review E*, vol. 66, p. 041904, 2002.

[27] F. Zeldenrust, P. J. P. Chameau, and W. J. Wadman, "Reliability of spike and burst firing in thalamocortical relay cells," *Journal of Computational Neuroscience*, vol. 5, no. 3, pp. 317–334, 2013.

[28] T. Kreuz, E. Satuvuori, M. Pofahl, and M. Mulansky, "Leaders and followers: Quantifying consistency in spatio-temporal propagation patterns," *New Journal of Physics*, 2017.

[29] D. Lyttle and J. M. Fellous, "A new similarity measure for spike trains: Sensitivity to bursts and periods of inhibition.," *Journal of Neuroscience Methods*, vol. 199, p. 296, 2011.

[30] C. V. Rusu and R. V. Florian, "A new class of metrics for spike trains," *Neural Computation*, vol. 26, no. 2, pp. 306–348, 2014.

[31] C. S. Cutts and S. J. Eglen, "Detecting pairwise correlations in spike trains: an objective comparison of methods and application to the study of retinal waves," *Journal of Neuroscience*, vol. 34, no. 43, pp. 14288–14303, 2014.

[32] S. M. Sherman, "Tonic and burst firing: Dual modes of thalamocortical relay," *Trends in Neurosciences*, vol. 24, pp. 122–126, 2001.

[33] E. M. Izhikevich, N. S. Desai, E. C. Walcott, and F. C. Hoppensteadt, "Bursts as a unit of neural information: selective communication via resonance," *Trends in Neurosciences*, vol. 26, pp. 161–167, 2003.

[34] J. Qu, R. Wang, Y. Du, and C. Yan, *Advances in Cognitive Neurodynamics (V)*, ch. An Improved Method of Measuring Multiple Spike Train Synchrony, p. 777. Springer Singapore, 2016.

[35] F. E. Kapucu, J. M. A. Tanskanen, J. E. Mikkonen, L. Ylä-Outinen, S. Narkilahti, and J. A. K. Hyttinen, "Burst analysis tool for developing neuronal networks exhibiting highly varying action potential dynamics," *Frontiers in Computational Neuroscience*, vol. 6, p. 38, 2012.

[36] C. M. Hales, J. D. Rolston, and S. M. Potter, "How to culture, record and stimulate neuronal networks on micro-electrode arrays (MEAs)," *Journal of Visualized Experiments*, vol. 39, p. e2056, 2010.

[37] M. E. Spira and A. Hai, "Multi-electrode array technologies for neuroscience and cardiology," *Nature Nanotechnology*, vol. 8, no. 2, pp. 83–94, 2013.

[38] C. Lewis, C. Bosman, and P. Fries, "Recording of brain activity across spatial scales," *Current Opinion in Neurobiology*, vol. 32, pp. 68–77, 2015.

[39] S. Maranò, D. Fäh, and Y. M. Lu, "Sensor placement for the analysis of seismic surface waves: sources of error, design criterion and array design algorithms," *Geophysical Journal International*, p. ggt489, 2014.

[40] J. Heidemann, M. Stojanovic, and M. Zorzi, "Underwater sensor networks: applications, advances and challenges," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 370, no. 1958, pp. 158–175, 2012.

[41] C. L. Muller, L. Chapman, C. Grimmond, D. T. Young, and X. Cai, "Sensors and the city: A review of urban meteorological networks," *International Journal of Climatology*, vol. 33, no. 7, pp. 1585–1600, 2013.

[42] M. J. Menne, I. Durre, R. S. Vose, B. E. Gleason, and T. G. Houston, "An overview of the global historical climatology network-daily database," *Journal of Atmospheric and Oceanic Technology*, vol. 29, no. 7, pp. 897–910, 2012.

[43] D. Lazer, A. S. Pentland, L. Adamic, S. Aral, A. L. Barabasi, D. Brewer, N. Christakis, N. Contractor, J. Fowler, M. Gutmann, *et al.*, "Life in the network: the coming age of computational social science," *Science (New York, NY)*, vol. 323, no. 5915, p. 721, 2009.

[44] L. K. Gallos, D. Rybski, F. Liljeros, S. Havlin, and H. A. Makse, "How people interact in evolving online affiliation networks," *Physical Review X*, vol. 2, no. 3, p. 031014, 2012.

[45] S. Boccaletti, G. Bianconi, R. Criado, C. I. Del Genio, J. Gómez-Gardeñes, M. Romance, I. Sendiña-Nadal, Z. Wang, and M. Zanin, "The structure and dynamics of multilayer networks," *Physics Reports*, vol. 544, no. 1, pp. 1–122, 2014.

[46] P. Lacroix, J.-R. Grasso, J. Roulle, G. Giraud, D. Goetz, S. Morin, and A. Helmstetter, "Monitoring of snow avalanches using a seismic array: Location, speed estimation, and relationships to meteorological variables," *Journal of Geophysical Research: Earth Surface*, vol. 117, no. F1, 2012.

[47] E. Pelinovsky, *Waves in Geophysical Fluids: Tsunamis, Rogue Waves, Internal Waves and Internal Tides*, ch. Hydrodynamics of Tsunami Waves, pp. 1–48. Vienna: Springer Vienna, 2006.

[48] Y. Kuramoto, *Chemical oscillations, waves, and turbulence*, vol. 19. Springer Science & Business Media, 2012.

[49] G. Baier, M. Goodfellow, P. N. Taylor, Y. Wang, and D. J. Garry, "The importance of modeling epileptic seizure dynamics as spatiotemporal patterns," *Frontiers in Physiology*, vol. 3, p. 281, 2012.

[50] V. Belik, T. Geisel, and D. Brockmann, "Natural human mobility patterns and spatial spread of infectious diseases," *Physical Review X*, vol. 1, no. 1, p. 011001, 2011.

[51] X. Wei, N. C. Valler, B. A. Prakash, I. Neamtiu, M. Faloutsos, and C. Faloutsos, "Competing memes propagation on networks: A network science perspective," *Selected Areas in Communications, IEEE Journal on*, vol. 31, no. 6, pp. 1049–1060, 2013.

[52] T. Kuhn, M. Perc, and D. Helbing, "Inheritance patterns in citation networks reveal scientific memes," *Physical Review X*, vol. 4, no. 4, p. 041036, 2014.

[53] F. Rieke, D. Warland, R. de Ruyter van Steveninck, and W. Bialek, *Spikes: Exploring the neural code*. Cambridge, Massachusetts: Institute of Technology, 1996.

[54] M. Abeles, "Synfire chains," *Scholarpedia*, vol. 4, no. 7, p. 1441, 2009.

[55] P. Bonifazi, M. Goldin, M. A. Picardo, I. Jorquera, A. Cattani, G. Bianconi, A. Represa, Y. Ben-Ari, and R. Cossart, "Gabaergic hub neurons orchestrate synchrony in developing hippocampal networks," *Science*, vol. 326, p. 1419, 2009.

[56] T. Kreuz, F. Mormann, R. G. Andrzejak, A. Kraskov, K. Lehnertz, and P. Grassberger, "Measuring synchronization in coupled model systems: A comparison of different approaches.," *Journal of Physics D*, vol. 225, p. 29, 2007.

[57] W. M. Kistler, W. Gerstner, and J. Leo van Hemmen, "Reduction of hodgkin-huxley equations to a single-variable threshold model.," *Neural Computation*, vol. 9, p. 1015, 1997.

[58] M. Mulansky, N. Bozanic, A. Sburlea, and T. Kreuz, "A guide to time-resolved and parameter-free measures of spike train synchrony," *Event-based Control, Communication, and Signal Processing (EBCCSP)*, vol. 2015 International Conference on, pp. 1–8, 2015.

[59] D. L. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook, *The traveling salesman problem: a computational study*. Princeton University Press, 2011.

[60] D. J. Earl and M. W. Deem, "Parallel tempering: Theory, applications, and new perspectives," *Physical Chemistry Chemical Physics*, vol. 7, no. 23, pp. 3910–3916, 2005.

[61] K. A. Dowsland and J. M. Thompson, "Simulated annealing," in *Handbook of Natural Computing*, pp. 1623–1655, Springer, 2012.

[62] Y. Ben-Ari, E. Cherubini, R. Corradetti, and J. L. Gaiarsa, "Giant synaptic potentials in immature rat ca3 hippocampal neurones.," *The Journal of Physiology*, vol. 416, no. 1, pp. 303–325, 1989.

[63] H. Takano, M. McCartney, P. I. Ortinski, C. Yue, M. E. Putt, and D. A. Coulter, "Deterministic and stochastic neuronal contributions to distinct synchronous ca3 network bursts," *Journal of Neuroscience*, vol. 32, no. 14, pp. 4743–4754, 2012.

[64] T. Kreuz, S. Luccioli, and A. Torcini, "Coherence resonance due to correlated noise in neuronal models," *Neurocomputing*, vol. 70, p. 1970, 2007.

[65] N. Malik, N. Marwan, and J. Kurths, "Spatial structures and directionalities in monsoonal precipitation over south asia.," *Nonlinear Processes in Geophysics*, vol. 17, p. 371, 2010.

[66] N. Boers, B. Bookhagen, H. Barbosa, N. Marwan, J. Kurths, and J. Marengo, "Prediction of extreme floods in the eastern central andes based on a complex networks approach," *Nature Communications*, vol. 5, p. 5199, 2014.

[67] R. S. Rosário, P. T. Cardoso, M. A. Muñoz, P. Montoya, and J. G. V. Miranda, "Motif-synchronization: A new method for analysis of dynamic brain networks with eeg," *Physica A*, vol. 439, p. 7, 2015.

[68] W. X. Zhou and D. Sornette, "Evidence of a worldwide stock market log-periodic anti-bubble since mid-2000.," *Physica A: Statistical Mechanics and its Applications*, vol. 330, p. 543, 2003.

[69] G. Schneider, M. N. Havenith, and D. Nikolić, "Spatiotemporal structure in large neuronal networks detected from cross-correlation," *Neural Computation*, vol. 18, no. 10, pp. 2387–2413, 2006.

[70] K. S. Gansel and W. Singer, "Detecting multineuronal temporal patterns in parallel spike trains," *Frontiers in Neuroinformatics*, vol. 6, p. 18, 2012.

[71] Y. N. Billeh, M. T. Schaub, C. A. Anastassiou, M. Barahona, and C. Koch, "Revealing cell assemblies at multiple levels of granularity," *Journal of Neuroscience Methods*, vol. 236, pp. 92–106, 2014.

[72] E. Russo and D. Durstewitz, "Cell assemblies at multiple time scales with arbitrary lag constellations," *eLife*, vol. 6, p. e19428, 2017.

[73] S. Schrader, S. Grün, M. Diesmann, and G. L. Gerstein, "Detecting synfire chain activity using massively parallel spike train recording," *Journal of Neurophysiology*, vol. 100, no. 4, pp. 2165–2176, 2008.

[74] G. L. Gerstein, E. R. Williams, M. Diesmann, S. Grün, and C. Trengove, "Detecting synfire chains in parallel spike data," *Journal of Neuroscience Methods*, vol. 206, no. 1, pp. 54–64, 2012.

[75] R. G. Andrzejak and T. Kreuz, "Characterizing unidirectional couplings between point processes and flows," *Europhysics Letters*, vol. 96, p. 50012, 2011.

[76] D. G. Amaral and M. P. Witter, "Hippocampal formation, paxinos g., the rat nervous system, 1995, 443-493."

[77] S. Leutgeb and J. K. Leutgeb, "Pattern separation, pattern completion, and new neuronal codes within a continuous ca3 map," *Learning & memory*, vol. 14, no. 11, pp. 745–757, 2007.

[78] J. J. Chrobak and G. Buzsáki, "High-frequency oscillations in the output networks of the hippocampal–entorhinal axis of the freely behaving rat," *Journal of Neuroscience*, vol. 16, no. 9, pp. 3056–3066, 1996.

[79] U. B. Hedrich, C. Liautard, D. Kirschenbaum, M. Pofahl, J. Lavigne, Y. Liu, S. Theiss, J. Slotta, A. Escayg, M. Dihné, *et al.*, "Impaired action potential initiation in gabaergic interneurons causes hyperexcitable networks in an epileptic mouse model carrying a human nav1. 1 mutation," *Journal of Neuroscience*, vol. 34, no. 45, p. 14874, 2014.

[80] V. Crépel, D. Aronov, I. Jorquera, A. Represa, Y. Ben-Ari, and R. Cossart, "A parturition-associated nonsynaptic coherent activity pattern in the developing hippocampus," *Neuron*, vol. 54, p. 105, 2007.

[81] T. F. Holekamp, D. Turaga, and T. E. Holy, "Fast three-dimensional fluorescence imaging of activity in neural populations by objective-coupled planar illumination microscopy," *Neuron*, vol. 57, p. 661, 2008.

[82] D. A. Henze, G. R. González-Burgos, N. N. Urban, D. A. Lewis, and G. Barrionuevo, "Dopamine increases excitability of pyramidal neurons in primate prefrontal cortex," *Journal of Neurophysiology*, vol. 84, p. 2799, 2000.

[83] F. Theunissen and J. P. Miller, "Temporal encoding in nervous systems: A rigorous definition," *Journal of Computational Neuroscience*, vol. 2, p. 149, 1995.

[84] A. P. Georgopoulos, "Current issues in directional motor control," *Trends in Neurosciences*, vol. 18, no. 11, pp. 506–510, 1995.

[85] R. VanRullen, R. Guyonneau, and S. J. Thorpe, "Spike times make sense," *Trends in Neurosciences*, vol. 28, no. 1, pp. 1–4, 2005.

[86] P. M. Di Lorenzo and J. D. Victor, "Taste response variability and temporal coding in the nucleus of the solitary tract of the rat," *Journal of Neurophysiology*, vol. 90, p. 1418, 2003.

[87] M. A. Harvey, H. P. Saal, J. F. Dammann III, and S. J. Bensmaia, "Multiplexing stimulus information through rate and temporal codes in primate somatosensory cortex," *PLoS Biology*, vol. 11, no. 5, p. e1001558, 2013.

[88] K. MacLeod, A. Backer, and G. Laurent, "Who reads temporal information contained across synchronized and oscillatory spike trains?," *Nature*, vol. 395, no. 6703, p. 693, 1998.

[89] M. Fukushima, P. L. Rauske, and D. Margoliash, "Temporal and rate code analysis of responses to low-frequency components in the birds own song by song system neurons," *Journal of Comparative Physiology A*, vol. 201, no. 12, pp. 1103–1114, 2015.

[90] D. S. Reich, F. Mechler, and J. D. Victor, "Temporal coding of contrast in primary visual cortex: when, what, and why," *Journal of Neurophysiology*, vol. 85, no. 3, pp. 1039–1050, 2001.

[91] R. Van Rullen and S. J. Thorpe, "Rate coding versus temporal order coding: what the retinal ganglion cells tell the visual cortex," *Neural Computation*, vol. 13, no. 6, pp. 1255–1283, 2001.

[92] C. Houghton and J. Victor, "Measuring representational distances–the spike-train metrics approach," *Visual Population Codes–Toward a Common Multivariate Framework for Cell Recording and Functional Imaging*, pp. 391–416, 2010.

[93] J. D. Victor and K. P. Purpura, "Metric-space analysis of spike trains: theory, algorithms and application," *Network: Comput. Neural Syst*, vol. 8, p. 127, 1997.

[94] J. D. Victor, "Spike train metrics," *Current Opinion in Neurobiology*, vol. 15, p. 585, 2005.

[95] L. Wang, R. Narayan, G. Graña, M. Shamir, and K. Sen, "Cortical discrimination of complex natural stimuli: Can single neurons match behavior?," *Journal of Neuroscience*, vol. 27, pp. 582–589, 2007.

[96] C. Tang, D. Chehayeb, K. Srivastava, I. Nemenman, and S. J. Sober, "Millisecond-scale motor encoding in a cortical vocal area," *PLoS Biology*, vol. 12, no. 12, p. e1002018, 2014.

[97] E. Chichilnisky and F. Rieke, "Detection sensitivity and temporal resolution of visual signals near absolute threshold in the salamander retina," *Journal of Neuroscience*, vol. 25, no. 2, pp. 318–330, 2005.

[98] J. M. Samonds and A. Bonds, "From another angle: differences in cortical coding between fine and coarse discrimination of orientation," *Journal of Neurophysiology*, vol. 91, no. 3, pp. 1193–1202, 2004.

[99] C. Houghton, "Studying spike trains using a van rossum metric with a synapse-like filter," *Journal of Computational Neuroscience*, vol. 26, p. 149, 2009.

[100] B. Schrauwen and J. V. Campenhout, "Linking non-binned spike train kernels to several existing spike train metrics.," *Neurocomputing*, vol. 70, pp. 1247–1253, 2007.

[101] W. R. Softky and C. Koch, "The highly irregular firing of cortical cells is inconsistent with temporal integration of random epsps," *Journal of Neuroscience*, vol. 13, no. 1, pp. 334–350, 1993.

[102] D. P. Hanes, K. G. Thompson, and J. D. Schall, "Relationship of presaccadic activity in frontal eye field and supplementary eye field to saccade initiation in macaque: Poisson spike train analysis," *Experimental Brain Research*, vol. 103, no. 1, pp. 85–96, 1995.

[103] M. J. Berry II and M. Meister, "Refractoriness and neural precision," in *Advances in Neural Information Processing Systems*, pp. 110–116, 1998.

[104] W. Bialek, F. Rieke, R. R. d. R. van Steveninck, and D. Warland, "Reading a neural code," in *Advances in Neural Information Processing Systems*, pp. 36–43, 1990.

[105] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: a Practical Approach to Global Optimization*. Springer Science & Business Media, 2006.

[106] V. Lopes-dos-Santos, S. Panzeri, C. Kayser, M. E. Diamond, and Q. Quian Quiroga, "Extracting information in spike time patterns with wavelets and information theory," *Journal of Neurophysiology*, vol. 113, no. 3, pp. 1015–1033, 2015.

[107] R. Baddeley, L. F. Abbott, M. C. Booth, F. Sengpiel, T. Freeman, E. A. Wakeman, and E. T. Rolls, "Responses of neurons in primary and inferior temporal visual cortices to natural scenes," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 264, no. 1389, pp. 1775–1783, 1997.

[108] P. Rapp, I. Zimmerman, E. Vining, N. Cohen, A. M. Albano, and M. Jimenez-Montano, "The algorithmic complexity of neural

spike trains increases during focal seizures," *Journal of Neuroscience*, vol. 14, no. 8, pp. 4731–4739, 1994.

[109] T. Schreiber and A. Schmitz, "Surrogate time series," *Physica D*, vol. 142, p. 346, 2000.

[110] S. Louis, G. L. Gerstein, S. Grüen, and M. Diesmann, "Surrogate spike train generation through dithering in operational time," *Frontiers in Computational Neuroscience*, vol. 4, p. 127, 2010.

[111] C. Houghton and T. Kreuz, "On the efficient calculation of van rossum distances," *Network: Computation in Neural Systems*, vol. 23, p. 48, 2012.

[112] A. Berkowitz, "Population coding," *Encyclopedia of Neuroscience*, p. 757, 2009.

[113] E. T. Rolls, A. Treves, and M. J. Tovee, "The representational capacity of the distributed encoding of information provided by populations of neurons in primate temporal visual cortex," *Experimental Brain Research*, vol. 114, no. 1, pp. 149–162, 1997.

[114] D. Huber, L. Petreanu, N. Ghitani, S. Ranade, T. Hromádka, Z. Mainen, and K. Svoboda, "Sparse optical microstimulation in barrel cortex drives learned behaviour in freely moving mice," *Nature*, vol. 451, no. 7174, p. 61, 2008.

[115] R. Q. Quiroga and S. Panzeri, "Extracting information from neuronal populations: information theory and decoding approaches," *Nature Reviews Neuroscience*, vol. 10, no. 3, p. 173, 2009.

[116] A. Georgopoulos, J. Lurito, M. Petrides, A. Schwartz, and J. Massey, "Mental rotation of the neuronal population vector," *Science*, vol. 243, no. 4888, pp. 234–236, 1989.

[117] C. Houghton and J. Victor, "Measuring representational distances–the spike-train metrics approach," in *Understanding Visual Population Codes – Towards a Common Multivariate Framework for Cell Recording and Functional Imaging* (N. Kriegeskorte and G. Kreiman, eds.), pp. 391–416, Cambridge, MA, USA: MIT Press, 2011.

[118] S. Panzeri, F. Petroni, R. S. Petersen, and M. E. Diamond, "Decoding neuronal population activity in rat somatosensory cortex: Role of columnar organization," *Cerebral Cortex*, vol. 13, no. 1, pp. 45–52, 2003.

[119] D. S. Reich, F. Mechler, and J. D. Victor, "Independent and redundant information in nearby cortical neurons," *Science*, vol. 294, no. 5551, pp. 2566–2568, 2001.

[120] A. Berényi, Z. Somogyvári, A. J. Nagy, L. Roux, J. D. Long, S. Fujisawa, E. Stark, A. Leonardo, T. D. Harris, and G. Buzsáki, "Large-scale, high-density (up to 512 channels) recording of local circuits in behaving animals," *Journal of Neurophysiology*, vol. 111, no. 5, pp. 1132–1149, 2013.

[121] A. M. Packer, L. E. Russell, H. W. Dalgleish, and M. Häusser, "Simultaneous all-optical manipulation and recording of neural circuit activity with cellular resolution in vivo," *Nature Methods*, vol. 12, no. 2, p. 140, 2015.

[122] H. Safaai, M. von Heimendahl, J. M. Sorando, M. E. Diamond, and M. Maravall, "Coordinated population activity underlying texture discrimination in rat barrel cortex," *Journal of Neuroscience*, vol. 33, no. 13, pp. 5843–5855, 2013.

[123] A. B. Graf, A. Kohn, M. Jazayeri, and J. A. Movshon, "Decoding the activity of neuronal populations in macaque primary visual cortex," *Nature Neuroscience*, vol. 14, no. 2, p. 239, 2011.

[124] I. Arandia-Romero, R. Nogueira, G. Mochol, and R. Moreno-Bote, "What can neuronal populations tell us about cognition?," *Current Opinion in Neurobiology*, vol. 46, pp. 48–57, 2017.

[125] D. Aronov, D. S. Reich, F. Mechler, and J. D. Victor, "Neural coding of spatial phase in v1 of the macaque monkey," *Journal of Neurophysiology*, vol. 89, p. 3304, 2003.

[126] C. Houghton and K. Sen, "A new multineuron spike train metric," *Neural Computation*, vol. 20, p. 1495, 2008.

[127] B. A. Olshausen and D. J. Field, "Sparse coding of sensory inputs," *Current Opinion in Neurobiology*, vol. 14, no. 4, pp. 481–487, 2004.

[128] A. C. Kwan and Y. Dan, "Dissection of cortical microcircuits by single-neuron stimulation in vivo," *Current Biology*, vol. 22, no. 16, pp. 1459–1467, 2012.

[129] R. A. Ince, S. Panzeri, and C. Kayser, "Neural codes formed by small and temporally precise populations in auditory cortex," *Journal of Neuroscience*, vol. 33, no. 46, pp. 18277–18287, 2013.

[130] J. C. Sanchez, J. M. Carmena, M. A. Lebedev, M. A. Nicolelis, J. G. Harris, and J. C. Principe, "Ascertaining the importance of neurons to develop better brain-machine interfaces," *IEEE Trans Biomed Eng*, vol. 51, no. 6, pp. 943–953, 2004.

[131] W. Ben-Ameur, "Computing the initial temperature of simulated annealing," *Computational Optimization and Applications*, vol. 29, no. 3, pp. 369–385, 2004.

[132] G. Buzsáki and K. Mizuseki, "The log-dynamic brain: how skewed distributions affect network operations," *Nature Reviews Neuroscience*, vol. 15, no. 4, p. 264, 2014.

[133] M. Okun, N. A. Steinmetz, L. Cossell, M. F. Iacaruso, H. Ko, P. Barthó, T. Moore, S. B. Hofer, T. D. Mrsic-Flogel, M. Carandini, *et al.*, "Diverse coupling of neurons to populations in sensory cortex," *Nature*, vol. 521, no. 7553, p. 511, 2015.

[134] D. M. Schneider and S. M. Woolley, "Discrimination of communication vocalizations by single neurons and groups of neurons in the auditory midbrain," *Journal of Neurophysiology*, vol. 103, no. 6, pp. 3248–3265, 2010.

[135] E. L. Mackevicius, M. D. Best, H. P. Saal, and S. J. Bensmaia, "Millisecond precision spike timing shapes tactile perception," *Journal of Neuroscience*, vol. 32, no. 44, pp. 15309–15317, 2012.

[136] I. Fried, K. A. MacDonald, and C. L. Wilson, "Single neuron activity in human hippocampus and amygdala during recognition of faces and objects," *Neuron*, vol. 18, no. 5, pp. 753–765, 1997.

[137] M. A. Lebedev and M. A. Nicolelis, "Brain–machine interfaces: past, present and future," *Trends in Neurosciences*, vol. 29, no. 9, pp. 536–546, 2006.

[138] J. N. Mak and J. R. Wolpaw, "Clinical applications of brain-computer interfaces: current state and future prospects," *IEEE Reviews in Biomedical Engineering*, vol. 2, pp. 187–199, 2009.

[139] M. L. Homer, A. V. Nurmikko, J. P. Donoghue, and L. R. Hochberg, "Sensors and decoding for intracortical brain computer interfaces," *Annual Review of Biomedical Engineering*, vol. 15, pp. 383–405, 2013.

[140] M. A. Nicolelis and M. A. Lebedev, "Principles of neural ensemble physiology underlying the operation of brain–machine interfaces," *Nature Reviews Neuroscience*, vol. 10, no. 7, p. 530, 2009.

[141] J. C. Sanchez, J. C. Principe, T. Nishida, R. Bashirullah, J. G. Harris, and J. A. B. Fortes, "Technology and signal processing for brain-machine interfaces," *IEEE Signal Processing*, vol. 25, p. 29, 2008.

[142] A. Arenas, A. Daz-Guilera, J. Kurths, Y. Moreno, and C. Zhou, "Synchronization in complex networks," *Physics Reports*, vol. 469, no. 3, pp. 93 – 153, 2008.

[143] M. M. Halassa, T. Fellin, and P. G. Haydon, "The tripartite synapse: roles for gliotransmission in health and disease," *Trends in Molecular Medicine*, vol. 13, no. 2, pp. 54 – 63, 2007.

8

ACKNOWLEDGEMENTS

To begin with, I feel that I cannot express big enough thanks in writing to all those who made this thesis possible. I want to give a general thanks to all the parties I met, worked with and who were part of these three years. However, there are people who should be addressed separately for their invaluable help and support.

Especially I want to thank my two supervisors Thomas Kreuz and Andreas Daffertshofer. Without their persistence and support this thesis might not have realized. Also I want to give equal thanks to Roberto Livi and Duccio Fanelli for all the help they provided navigating the Italian university system and to Bob van Dijk for many interesting discussions. Additionally I want to give thanks to all the people who made COSMOS what it is. I want to thank all the principal investigators and especially our project coordinator Arkady Pikovsky, vice coordinators Michael Rosenblum and Antonio Politi, as well as our wonderful project manager Caroline Reid. Also big thanks to all the early stage researchers in COSMOS I had the privilege to work with. Special thanks go to Bastian Pietras, Nicolas Deschle, Clement Zankoc, and Irene Malvestio for many insightful discussions. In addition to the people in COSMOS I also want to thank Nebojsa Bozanic and Mario Mulansky with whom I had the privilege to work closely with during the first year. Then, I also want to give my thanks to the members of the reading committee for evaluating this thesis and for many useful comments for improving the thesis.

The biggest thanks of all goes to all of my family; to my parents Harri and Elina, my siblings Meeri and Esa, and especially to my wife Sanni who has been taking care of our baby girl Fiona for the last eight months and has been unbelievably resilient and supportive during these three years.

9

SUMMARY

A neuron transmits information by a moving potential difference over the cell membrane called action potential and by releasing neurotransmitters to other neurons as a result. The time discrete representation of an action potential is called a *spike*. The series of spikes fired by the same neuron over time are called *spike trains*

A lot of neuroscience research is concentrated on finding response patterns in spike trains of neurons. For a given stimulus a neuron that codes for something in the stimulus is expected to elicit similar response each time. This is called *reliability* of response. However, defining when a response pattern is reliable or not is not a simple task. One may argue that similarity itself is not a well-defined concept. In a way this boils down to defining which properties of the response are considered to describe similarity. And, of course, selecting them will affect the results of any comparison. The properties considered for assessing similarity of responses of neurons must come from the way the neurons transmit information or *code*.

Similarity can be self-similarity in response to the same stimulus at different times, or it can be similarity between responses of two different neurons. There are many ways to assess if spike patterns are similar or not that use different assumptions.

Over the years different measures have been developed in order to quantify similarities between two or more spike trains. The two most popular time scale parametric measures, the Victor-Purpura and the van Rossum distance, describe spike train (dis)similarity based on user-defined time scales to which the measures are mainly sensitive. A drawback of these measures is the fixed time scale, since it sets a boundary between rate and time coding for the recording. However, for real data which typically contain many time scales (such as regular spiking and bursts), this is difficult to detect with a measure that is mainly sensitive to only one of them.

The problem of having to choose one time scale has been eliminated in the three time-resolved and time scale independent measures ISI-dis-

tance, SPIKE-distance and SPIKE-synchronization. Since they always adapt to the local firing rate, all three of these measures are time scale free. While they correctly identify the relative firing rate differences, they have no concept of actual time scales and treat all time scales as equally important. As a the consequence, for very small time scales even minor deviations from perfect synchrony lead to very high values of dissimilarity. For real data the smallest time scales are often not very relevant and any dissimilarities there can mostly be disregarded. Thus the measures' focus on the local time scales results in a (spurious) amplification of dissimilarities which compared to the global time scales are rather negligible.

In Chapter 2 I addressed the problem by proposing generalizations to the three measures called adaptive ISI-distance (A-ISI-distance), adaptive SPIKE-distance (A-SPIKE-distance) and adaptive SPIKE-synchronization (A-SPIKE-synchronization). These generalized definitions add a notion of the relative importance of local differences compared to the global time scales. In particular, they start to gradually ignore differences between spike trains for ISIs that are smaller than a minimum relevant time scale. The adaptive generalizations allow to disregard spike time differences that are not relevant on a more global scale.

The methods were developed for use in neuroscience, but in fact they work the same with any point process. In neuroscience typical time scales are in the range of milliseconds or sometimes seconds and any time scales below this will not be considered relevant. In fields such as meteorology the respective time scales could be hours and days or even months and years. The relevant time scale clearly depends on the system under consideration. Setting the minimum relevant time scale for a given dataset might not be a simple task. To address this, I proposed a method to extract a threshold value from the spike trains, that is based on the proportions of the different time scales present in the data. The extended methods are intended to be used exclusively when there are multiple time scales in the data.

In Chapter 3 I explained a complementary method for identifying similarity between spike trains which quantifies the consistency of the leader-follower relationships within a spike train set. The framework consists of two directional measures (*SPIKE-Order* and *Spike Train Order*) that allows to define a value termed *Synfire Indicator* which quantifies the consistency of the leader-follower relationships in a rigorous and automated manner, and also to sort multiple spike trains from leader to follower. The SPIKE-Order profile was used for color-coding and visualizing local spike leaders and followers and Spike Train Order functioned as an overall indicator of leader-follower consistency. A set of spike trains ex-

hibiting perfectly consistent repetitions of the same global propagation pattern is called a *synfire pattern*. The synfire patterns were evaluated using spike-to-spike matching and expecting that the system where the spikes were recorded from functions with robust single spike timing.

In Chapters 2 and 3 I explained new methods for point processes. In Chapter 4 I took a step back and had a look on how the similarity descriptions in some of the most commonly used spike train distances actually match to what is expected of them. The analysis was based on the two main approaches to neuronal coding. Since a neuron does not discriminate where the stimulation it receives actually came from, it seems reasonable to assume that high enough firing rate in the downstream neurons will eventually build up and elicit spikes in the next one. This assumption is called *rate coding*. An alternative approach considers a more compact coding, where each neuron time their spikes carefully together in order to achieve the wanted spike upstream at a certain time. This referred to as *time coding*. Both the assumptions are perfectly reasonable, yet exclusive from the neuron's point of view. As quoted in Chapter 4:

- *"It is generally accepted that a rate encoding scheme is one in which the relevant information encoded about the stimulus is correlated only with the number of elicited spikes within the encoding window and is not correlated with any aspect of the temporal pattern of the spikes within the encoding window."*
- *"In a temporal encoding scheme, the relevant information is correlated with the timing of the spikes within the encoding window, over and above any information that might be correlated with the number of spikes within the window."*

This definition of temporal coding does not require rate correlation actually to be present. Also it is not limited to single spike correlations but applies to any correlations in spike patterns that would not be expected due to rate alone.

In Chapter 4 I used that definition of time coding as correlations beyond rate to investigate how the sensitivity of the different spike train distances to rate and time coding depends on the rate of the spike trains. In this study I asked two questions: How does the sensitivity of the different spike train distances to rate and time coding depend on the rates of the two processes and how high a rate is needed in order to obtain reliable estimates of timings in the data? The analysis was conducted using independent steady rate Poisson spike trains as surrogates for random spike trains with fixed rate and no timing information.

The first finding is that the spike-resolved Victor-Purpura distance compares the spike trains spike-for-spike and thus they are always sensitive to differences in spike counts even for parameter values seemingly indicating time coding. For large spike count differences the spike-resolved distances do not obtain the ability to assess timing information beyond spike pairs and thus in many cases most of the distance comes from mismatch in spike counts rather than timings, independently of the time scale parameter. As a result, for the Victor-Purpura distance timing information is only available for spike trains with almost identical rates. Since the behaviour of the van Rossum distance in response to rate differences closely resembles that of the Victor-Purpura distance, it also has the same problem (in addition to its normalization issues for different tau-values).

The second finding is that the time-resolved measures perform better in assessing timings in the normal case of reasonably high rates. These measures can also provide a meaningful instantaneous similarity profile within the coding window. Since they assess similarity in time, the exact spike count becomes less important and the actual timing of events becomes more relevant. However, they suffer from artefacts when the rates of the spike generation processes is so low that the floor effect takes place. For spike trains with only a few spikes one should use the spike-resolved Victor-Purpura or van Rossum distance, since they assess first similarity in spike count and then apply timing information assessment only for pairs of spikes.

The nervous system is believed to employ large populations of neurons to code and broadcast information. Population coding can be considered less vulnerable and, hence, a more reliable and robust manner than coding via single neurons. In Chapter 5 I evaluated existing approaches and introduced new algorithms for identifying the most discriminative subpopulations from a population of recorded neurons.

In neuronal recordings population coding can appear in two ways. First, all the neurons in the recorded population contribute equally. Patterns of activity within the population are irrelevant for coding as all that matters is whether or not any of the neurons fires. There, the information being conveyed is that of a single spike train generated by the population as a whole. In contrast to this so-called *summed population (SP) hypothesis*, each neuron may have a unique and distinguishable role. In this case, the population is best decoded neuron-by-neuron, which is referred to as the *labeled line (LL) hypothesis*. The coding via individual neurons and the summation of an entire population are the extreme case in a broad spectrum of possibilities. In fact, recent evidence points at some interme-

diate scenario in which a comparably small number encodes information not only in a robust but also very efficient way.

The search for an optimal coding population requires fine-tuned analyses under both the SP- and the LL-hypothesis. For these two cases I showed how to separate relevant from irrelevant subpopulations by identifying the subpopulation of neurons amongst all possible ones that discriminates best a given set of stimuli.

For the SP case, I compared three fundamentally different algorithms for finding the subpopulation that is able to most efficiently discriminate between a set of stimuli. First SP algorithm is a simple brute force solution that goes through every possible permutation. This is computationally heavy and even for relatively small data sets is not feasible. Computationally lighter gradient algorithms were also tested and I ran two simulations that are constructed such that each time one of the two variants of the gradient algorithm did not find the best subpopulation since it got trapped in a local maximum. Since gradient algorithms are much faster than the brute force approach and successful under idealized conditions, they can be used for first testing. However, our examples illustrate that they can generally not be relied upon in more realistic settings. I also constructed a third algorithm using simulated annealing that provides a recovery mechanism that considerably reduces the likelihood of getting stuck in a local maximum and providing correct result more consistently even if slightly increasing computation load.

For the LL case, I introduced a novel algorithm for identifying the most discriminative LL population by evaluating every neuron separately. First, for every individual stimulus pair the algorithm identifies the discriminative neurons and selects the best one. These best neurons are then combined to form the optimal LL-population. The algorithm can handle quite involved coding scenarios, even though its computational complexity is much lower than in the SP case. Moreover, one is guaranteed to find the best subpopulation since this time no search in a very high-dimensional subpopulation space is needed. I also identified that there is a mismatch between definitions of discrimination and coding, even if the two concepts seem to be linked. In reality one may have coding without discrimination and vice versa.

In this thesis I have introduced new methods and evaluated existing ones. The most important thing to consider when applying methods is to understand what they are designed for. What we think they should do is sometimes not what they are actually doing. Having a clear understanding of how the measures we use evaluate the data is crucial for proper interpretation of the results. My thesis was devoted to this challenge. My comparative studies and newly proposed methods provide

an encompassing approach to the analysis of spike trains and general point processes, and with that, hopefully paved the way to application in neuroscience and beyond.