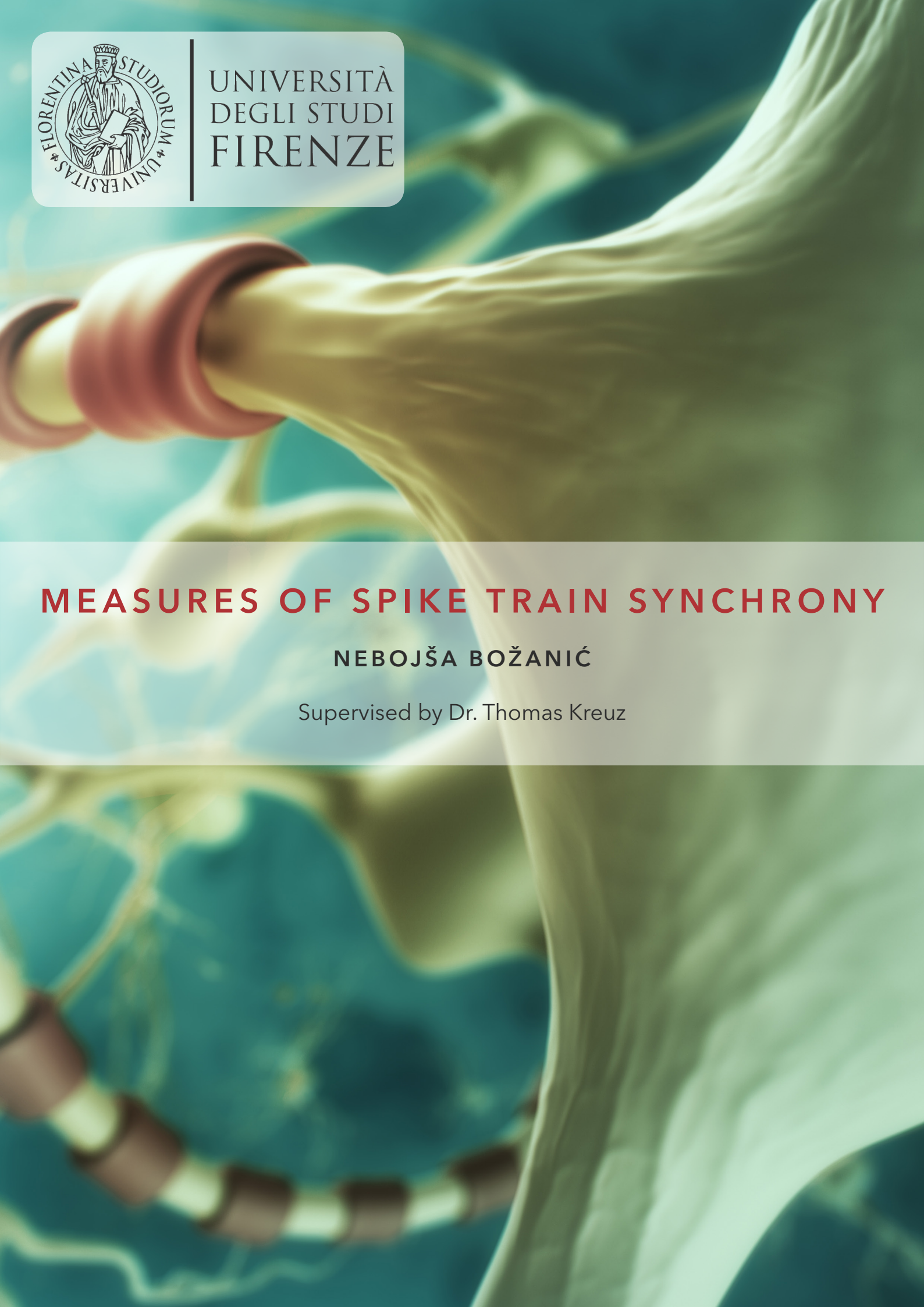


UNIVERSITÀ DEGLI STUDI FIRENZE

MEASURES OF SPIKE TRAIN SYNCHRONY

NEBOJŠA BOŽANIĆ

Supervised by Dr. Thomas Kreuz

Nebojša Božanić: Measures of Spike Train Synchrony.

PhD in Nonlinear Dynamics and Complex Systems, Department for Information and Communications Technology, University of Florence, © Autumn 2015.

Cover design by *Miroslav Maletić*.

ABSTRACT

In experimental neuroscience techniques for recording large-scale neuronal spiking activity are developing very fast. This leads to an increasing demand for algorithms capable of analyzing large amounts of spike train data. One of the most crucial and demanding tasks is the identification of similarity patterns with a very high temporal resolution and across different spatial scales. To address this task, in recent years time-resolved measures of spike train synchrony such as the ISI-distance and the SPIKE-distance have been proposed. Here we add the complementary measure SPIKE-synchronization, a sophisticated multivariate coincidence detector with a very intuitive interpretation.

In the first Results chapter we present SPIKY, an interactive graphical user interface that facilitates the application of these three time-resolved measures of spike train synchrony to both simulated and real data. SPIKY, which has been optimized with respect to computation speed and memory demand, also comprises a spike train generator and an event detector that makes it capable of analyzing continuous data. Finally, the SPIKY package includes additional complementary programs aimed at the analysis of large numbers of datasets and the estimation of significance levels.

In the second Results chapter we deal with the very important problem of latency variations in real data. By means of a validated setup we can show that the parameter-free SPIKE-distance outperforms two time-scale dependent standard measures. In summary, in this thesis we provide several important measures and corrections that when applied to the right experimental datasets could potentially lead to an increased understanding of the neural code – the ultimate goal of neuroscience.

PUBLICATIONS

[1] N. Bozanic, M. Mulansky, and T. Kreuz. "SPIKY." In: *Scholarpedia* 9.12 (2014), p. 32344.

[2] T. Kreuz, M. Mulansky, and N. Bozanic. "SPIKY: A graphical user interface for monitoring spike train synchrony." In: *Journal of neurophysiology* 113.9 (2015), pp. 3432–3445.

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

[4] N. Bozanic, M. Mulansky, and T. Kreuz. "Latency correction using the time-scale independent SPIKE-distance." In preparation.

[5] T. Loncar-Turukalo, G. Mijatovic, N. Bozanic, F. Stoll, D. Bajic, and E. Procyk. "Time-Frequency Characterization of Local Field Potential in a Decision Making Task." In: *Proceedings of the 37th Annual International Conference Of The IEEE Engineering In Medicine And Biology Society (EMBC)*. 2015.

[6] J. M. Rivera-Rubio, B. Kasap, N. Bozanic, A. Kai, and A. A. Bharath. "Populaiton Coding via Matrix Equilibration." In preparation.

[7] J. M. Rivera-Rubio, B. Kasap, N. Bozanic, and A. A. Bharath. "Reconstructing place fields with visual stimuli." In preparation.

OTHER WORKS

During the three years of this doctorate I was also involved in several side projects which have already led or which will lead to published manuscripts in the near future (publications [5]-[7] on the previous page). However, since these studies did not really fit into the context of this thesis I refrained from including any of their results, but for completeness I here provide their abstracts.

FIRST PAPER [5]

AUTHORS T. Loncar-Turukalo, G. Mijatovic, **N. Bozanic**, F. M. Stoll, D. Bajic, and E. Procyk

TITLE Time-Frequency Characterization of Local Field Potential in a Decision Making Task

JOURNAL Proceedings of the 37th Annual International Conference Of The IEEE Engineering In Medicine And Biology Society (EMBC). 2015.

ABSTRACT This study seeks to characterize the neuronal mechanisms underlying voluntary decisions to check/verify. In order to describe and potentially decode decisions from brain signals we analyzed intracortical recordings from monkey prefrontal regions obtained during a cognitive task requiring self-initiated as well as cue-instructed decisions. Using local field potentials (LFP) and single units, we analyzed power spectral density, oscillatory modes, power profiles in time, single unit firing rate, and spike-phase relationships in the β band. Our results point toward specific but variable activation patterns of oscillations in β band from separate recordings, with task-dependent frequency preference and amplitude

modulation of power. The results suggest relationships between particular LFP oscillations and functions engaged at specific time in the task.

SECOND PAPER [6]

AUTHORS J. M. Rivera-Rubio, B. Kasap, **N. Bozanic**, K. Arulkumaran, and A. A. Bharath

TITLE Population Coding via Matrix Equilibration

JOURNAL In preparation

ABSTRACT Divisive normalisation has been proposed as a model to describe non-linear population coding effects observed within biological sensory neurons. Different forms of normalization have also been applied in machine learning, such as L_2 normalisation across features for each sample.In this paper, we explore the relationship between divisive normalization and matrix equilibration, a technique that scales matrix rows and columns. Using mixed matrix norms, and introducing partial mixed matrix norms, we interpret divisive normalization in terms of matrix equilibration. Two broad classes of normalization are explored, synchronous and asynchronous. We evaluate the effect of divisive normalisation based on the partial mixed matrix norms in two use cases. The first involves modelling the receptive fields of artificial place cells that use visual information from hand-held cameras. The second is within a Gaussian RBM. Our conclusion is that the use of partial mixed norms provide methods of scaling ensembles of neurons and their responses over experiments in a variety of ways, some of which may improve the performance of artificial networks. The freedom to select different combinations of norms provides the potential to improve performance, but improvements are both highly context dependent, and network dependent.

THIRD PAPER [7]

AUTHORS J. M. Rivera-Rubio, B. Kasap, **N. Bozanic**, and A. A. Bharath

TITLE Reconstructing place fields with visual stimuli

JOURNAL In preparation

ABSTRACT We propose a computational model that simulates the behaviour of biological place cells from visual inputs. In contrast with previous rate coding models place cells, the proposed model does not use distance- to-boundary calculations, but rather takes input in the form of pixel data, simulating, instead rectified simple cells in V1. The proposed model is tested in two ways. First, it is applied to data acquired from hand-held and wearable cameras as a user navigates along an approximate one-dimensional track; we show that place-cell responses similar to those observed in single-cell recordings, can be constructed. We explore the effect on joint population coding of synthetic place cells using different forms of population normalisation. Finally, a decoding of place-cell activity is proposed. This decoding is applied both to the place-cell simulations and to recordings acquired from navigating rats moving along linear tracks.

ACKNOWLEDGMENTS

This thesis is actually an homage to my supervisor *Thomas Kreuz*. If I ever do something in my career it would mostly be because of his 3-year supervision and his persistence from which only I had benefits. I am grateful that I conducted my research under his guidelines. I also thank *Mario Mulansky*, who also partially supervised my work and also hosted me during the time when I wrote this thesis - so thanks to *Mario* and *Mariana*.

Thanks or 'Merci' *Alban Levy* and 'Hvala' *Aleksandar Božanić* for carefully reading some parts of the thesis. Also big thanks to *Miroslav Maletić* for the design part of the thesis.

Time spent at the Institute for Complex Systems (*Istituto dei Sistemi Complessi - ISC*) which is part of the Italian National Research Council (*Consiglio Nazionale delle Ricerche - CNR*) was truly dynamic and chaotic, but beautiful. I would like to thank all *CNR* personnel. It was a pleasure to be part of the team led by *Alessandro Torcini*. Special thanks goes to my colleagues and friends, among others *David Ciccio Angulo-Garcia*, *Stefano Luccioli*, *Simona Olmi*, *Stefano Lepri* and our new team member *Eero Räisänen*.

I express my gratitude to *John van Opstal* and his group for hosting me at the Radboud University, Nijmegen, Netherlands, *Joshua D Berke* and his group for hosting me at the University of Michigan, Ann Arbor, MI, USA as well as to *Anil Bharath* and his group for hosting me at the Imperial College London, UK. Also special thanks to *Tatjana Lončar-Turukalo* for hosting me at the Faculty of Technical Sciences, Novi Sad, Serbia and *Emmanuel Procyk* for hosting me at the Stem cell and Brain Research Institute, Lyon, France. I would like to thank all of my hosts and their lab members for interesting and sometimes inspiring discussions.

The whole research activity would not be possible without the funding support from the European Commission, through the Marie Curie Initial Training Network 'Neural Engineering Transformative Technologies (NETT)', project 289146. I also acknowledge the Serbian Ministry of Youth and Sports.

I would like to thank *Gorana Mijatović*, *Bahadir Kasap*, *Andreea Sburlea*, *Alessandro Barardi*, and other project fellows, with whom I went through my PhD together.

And finally, I want to thank *Branka*, *Dragan*, and *Aleksandar*, and the rest of my family for their unconditional love and support. My life in Florence would not be the same without *Alexandros Markatselis*, *Irena Stanković*, *Fabio De Francesco* and *Valeria Betró* who made me feel like home. I want to thank my friends in Serbia and the rest of the world, and finally I want to express my gratitude to *Nina Vasković* for her kind affection.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AP Action Potential

ISI Inter Spike Interval

PSTH Peri-Stimulus Time Histogram

CDF Cumulative Distribution Function

GUI Graphical User Interface

MEX Matlab Executable

Part I

INTRODUCTION

1

MOTIVATION

The brain consists of many large interconnected networks of nerve cells, also called 'neurons'. These neurons communicate among each other via action potentials (or simply 'spikes') which are abrupt and transient changes of membrane potential. Sequences of such spikes are called spike trains. Since in the brain both the outer and the inner world is to a very large extent represented in such spike trains, spike trains form the language of the brain. Therefore, their analysis can help to decipher the so-called neuronal code. A natural and very common approach to do so is to look at spike train synchrony and among the most important tools for this are spike train distances.

Spike train distances are measures of the degree of synchrony between spike trains which yield low values for very similar and higher values for dissimilar spike trains. They are applied in two major scenarios: simultaneous and non-simultaneous recordings.

The first scenario is the *simultaneous* recording of a neuronal population, typically in a spatial multi-channel setup. If different neurons emit spikes at the same time, these spikes are considered truly "synchronous" (Greek: "occurring at the same time"). Synchronization between individual neurons has been proven to be of high prevalence in many different neuronal circuits (Tiesinga et al., 2008; Shlens et al., 2008). However, as of now many open questions remain regarding the spatial scale of spike train synchrony and the nature of interactions (pairwise or higher order, see Nirenberg and Victor, 2007) as well as its functional significance for neuronal coding and information processing (Kumar et al., 2010). In epilepsy for example, the analysis of the varying similarity patterns of *simultaneously* recorded ensembles of neurons could lead to a better understanding of the mechanisms of seizure generation, propagation, and ter-

mination (Truccolo et al., 2011; Bower et al., 2012). Another example for the importance of synchrony in neural populations can be deduced from Izhikevich's observation that neurons do not fire on their own, but as a result of incoming spikes from other neurons Izhikevich, 2007. Similarly, synaptic plasticity is often assumed to work according to this famous quote: *Cells that fire together, wire together* (Hebb's rule, Hebb, 1949). A recent study addressed the connection between synchronous firing of neurons and their wiring and found that also the other way around, synchronous activity can lead to the establishment of cell assemblies Munz et al., 2014.

In the second scenario, the neuronal spiking response is recorded in different time intervals and synchrony is evaluated between such recordings of the same neuron. In order to allow a meaningful comparison, there has to be a temporal reference point which is typically set by some kind of trigger (e.g., the onset of an external stimulation). There are two prominent applications for this *successive* trials scenario. Repeated presentation of the same stimulus addresses the reliability of individual neurons (Mainen and Sejnowski, 1995) while different stimuli are used to investigate neuronal coding and to find the features of the response that provide the optimal discrimination (e.g., Victor, 2005, for a more general introduction to neural coding cf. Quian Quiroga et al., 2002). These two applications are related since for a good understanding one needs both a pronounced discrimination between stimuli (high inter-stimulus spike train distances) and a high reliability for the same stimulus (low intra-stimulus spike train distances).

The main objective in neuroscience is to understand the neural code, i.e. the relationship between the stimulus and the spike train(s). In an abstract sense, measures of spike train synchrony as discussed in this thesis can help with this problem by providing a structure in the spike train space. This idea is visualized in Figure 1.1. Different stimuli can be connected to different spike train responses that are clustered according to spike train synchrony measures.

Electrophysiology and other modern recording techniques are developing fast. For both simultaneous population and successive trial recordings they often provide more

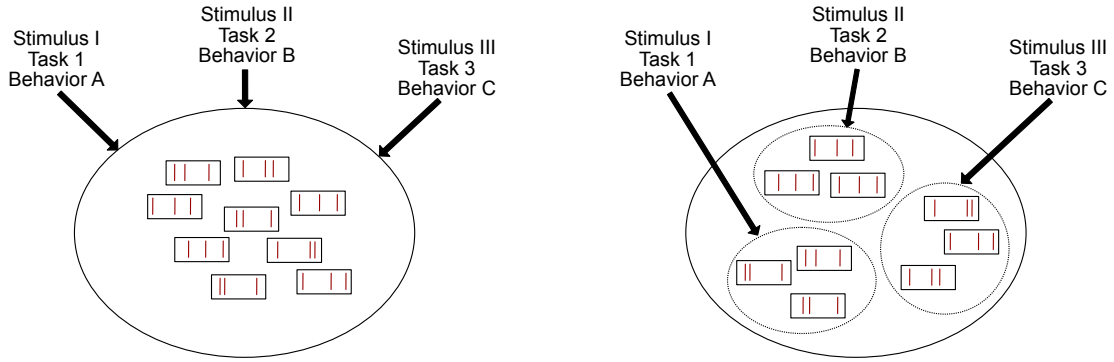


Figure 1.1: Visualization of the introduction of a structure in the space of spike trains by spike train distance measures [Diagrams created by Mario Mulansky].

data than available methods of spike train analysis can handle. There is a lack of algorithms able to identify multiple spike train patterns across different spatial scales and with a high temporal resolution.

However, in recent years, two time-resolved measures have been proposed: The ISI-distance (Kreuz et al., 2007) and the SPIKE-distance (Kreuz et al., 2013). Both methods rely on instantaneous estimates of spike train dissimilarity which makes it possible to track changes in instantaneous clustering, i.e. time-localized patterns of (dis)similarity among multiple spike trains. Additionally, both measures are parameter-free and time-scale independent. Furthermore, the SPIKE-distance also comes in a causal variant (Kreuz et al., 2013) which is defined such that the instantaneous values of dissimilarity are derived from past information only. This way, time-resolved spike train synchrony can be estimated in real-time. Both measures have already been widely used in various contexts (e.g., for the most recent measure, the SPIKE-distance, Papoutsi et al., 2013; Di Poppa and Gutkin, 2013; Sacré and Sepulchre, 2014). Another time-scale independent and time-resolved method is event synchronization (Quian Quiroga et al., 2002), a sophisticated coincidence detector which quantifies the level of synchrony from the number of quasi-simultaneous appearances of spikes. Originally, it was proposed and used in a bivariate context only. In this thesis it is adapted to fit into the time-resolved framework and extended to the multivariate case. Since this involves

substantial changes of the original event synchronization, to avoid confusion we term the new, modified measure SPIKE-Synchronization.

With all of these measures, spike trains can be analyzed on different spatial and temporal scales, accordingly there are several levels of information extraction (Kreuz, 2012). This diversity of representations make a straightforward implementation of the measures in one simple program/function unfeasible. What is needed is an intuitive and interactive tool for analyzing neuronal spike train data which at the same time provides high computational speed, efficient memory management, and applicability to large datasets.

In the first main part of this thesis we address this need and present the graphical user interface SPIKY (Bozanic et al., 2014; Kreuz et al., 2015). Given a set of real or simulated spike train data (importable from many different formats), SPIKY calculates the measures of choice and allows the user to switch between many different visualizations such as measure profiles, pairwise dissimilarity matrices, or hierarchical cluster trees. SPIKY also includes the possibility of generating movies which are very useful in order to track the varying patterns of (dis)similarity. SPIKY has been optimized with respect to both computation speed (by using MEX-files, i.e. C-based Matlab executables) and memory demand (by taking advantage of the piecewise linear nature of the dissimilarity profiles). Finally, the SPIKY-package includes two complementary programs. The first program SPIKY_loop is meant to be used for the analysis of a large number of datasets. The second program SPIKY_loop_surro is designed to evaluate the statistical significance of the results obtained for the original dataset by comparing them against the results of spike train surrogates generated from that dataset.

A typical problem when working with experimental spike train recordings is spike train latency. Latency refers to a constant global shift of spike times between different spike trains that could be recorded either from different neurons or from the same neuron in different trials. It is caused by various biophysical limitations and it largely contributes to a spurious lowering of synchrony or correlation in the analysis of real data. Therefore, identifying latency within a set of spike trains is a crucial preanalysis

step in spike train analysis. In the second main part of this thesis we propose the use of the time-scale independent and parameter-free SPIKE-distance to estimate the latency in simulated data. In contrast to the standard methods which are time-scale dependent, no parameter optimization is required and there are no problems with varying time scales.

ORGANIZATION OF THE THESIS

The remainder of the thesis is organized as follows: In Chapter 2 we provide elementary notions of neuroscience, morphology of the neuron, in particular the concepts of action potentials (which has been assumed to be the centerpiece of the intercellular signaling), recording techniques and spike sorting. This chapter provides the origin of spike trains. In Chapter 3 we motivate different scenarios of spike train (dis)similarity, introduce different spike train (dis)similarity measures and provide some details about their implementation. In Chapter 4, the different ways to extract information from spike train data are presented. SPIKY, our graphical user interface for monitoring spike train synchrony, is presented in Chapter 5. Additionally, there we discuss refined definitions as well as some improvements realized in the new implementation of the measures.

In Chapter 6 we address a very important issue, latency variations in real data. We show how spike train analysis with both time-scale dependent and independent measures are affected by a noisy variation of the onset of the stimuli, i.e. by shifting spike trains. We compare different methods and we provide all details of the setup, as well as a comparison of their performance (Bozanic et al.,).

Finally, in Chapter 7 we summarize the thesis (including all methods, the SPIKY program and the latency correction) and present an outlook on future developments.

2

BACKGROUND

The objective of this chapter is to introduce some elementary notions of computational neuroscience, in particular the concepts of spikes, their generation, their propagation and important sources of their unreliability or mis-detection. The biological background is presented in a 'minimalistic but non-reductionistic' manner. For a more thorough study of neurophysiology we refer to Kandel et al., 2000.

The human brain consists of $8.6 \cdot 10^{10}$ neurons with 70 km of fiber, hundreds, or even thousands, different types of neurons, over $1.5 \cdot 10^{15}$ synapses, and more than 100 different kinds of neuro transmitters (Azevedo et al., 2009; Pakkenberg et al., 2003). Some neurons such as Purkinje cells have about 200 000 input connections, others, like retinal ganglion, have 500 (Eliasmith and Anderson, 2004; Gerstner and Kistler, 2002). The complexity of the brain is even more striking when one takes into account that neurons are outnumbered by 'supporter' cells, so called glia cells, by a factor of about 10 (or up to 50) to 1 (Kandel et al., 2000).

Figure 2.1 can give a partial insight into how the brain looks from within, but in reality it is much more denser. Neither the brain nor the cortex consist exclusively of neuronal cells. As we mentione, there is a large number of glia cells, that are required for energy supply and structural stabilization of brain tissue. Since glia cells are assumed not to be directly involved in information processing (but see Otis and Sofroniew, 2008 for a different point of view), we will not discuss them any further.

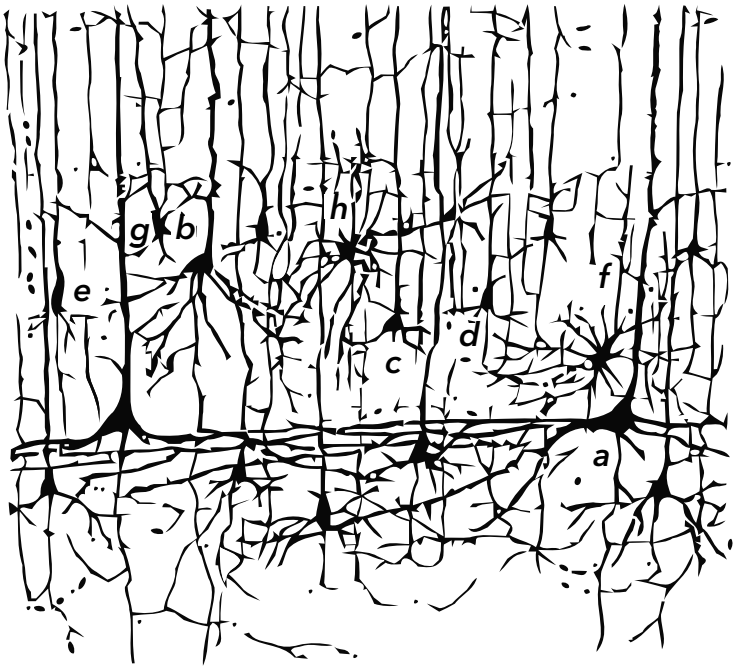


Figure 2.1: Neurons in the mammalian cortex. A drawing of Santiago Ramon y Cajal (1909). Modern staining procedures reveal that there are many more neurons.

2.1 NEURONS

A neuron, or a nerve cell, is an electrosensitive cell capable of generating an Action Potential (AP). The main components of neurons are the soma (the cell body), the axon, and the dendrites (see Figure 2.2). The body and the dendrites form the receptive regions, while the axon is the main transmitter. Besides these parts, particular zones of interests are where two neurons, or a neuron and another cell, are making functional contacts via synapses. According to the type of transmission there are chemical and electrical synapses. The presynaptic part is most often a synaptic expansion of presynaptic neuron, while the postsynaptic part is typically a dendrite or the body of the postsynaptic neuron.

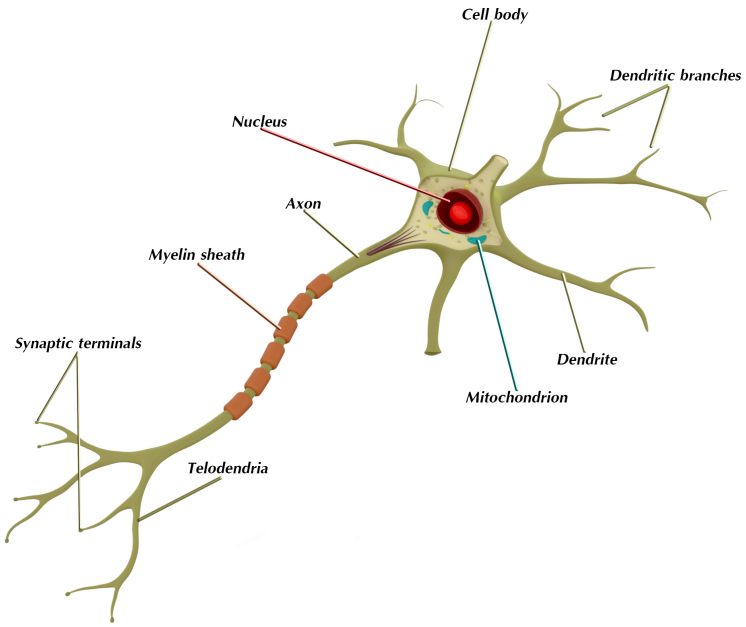


Figure 2.2: Morphology of a single neuronal cell.

The CLASSIFICATION OF NEURONS is done considering many factors such as shape, length of the axon, size of the cell body, the kind of receptor-effector system etc. The two most important classifications for neuroscientists are based on the effect to other neurons (excitatory, inhibitory, or modulatory) and the way neuron spikes (tonic / regular spiking, fast spiking, or phasic bursting) [1].

The nervous system, in order to function in the way it is assumed today uses fast signalization via action potentials, so-called spikes. Not only regular neurons are able to communicate via spikes, muscle cells and endocrine cells do so as well. Also note that while some neurons (such as the amacrine retinal cell) are indeed non-spiking, their overall importance is rather negligible.

1 Neurons can change the way they spike, and this biophysical property will be incorporated in our simulated latency data in Chapter 6.

2.2 ACTION POTENTIALS (SPIKES)

An action potential is a propagating change of potential whose amplitude is not decreasing over time and distance from the place where it originated.

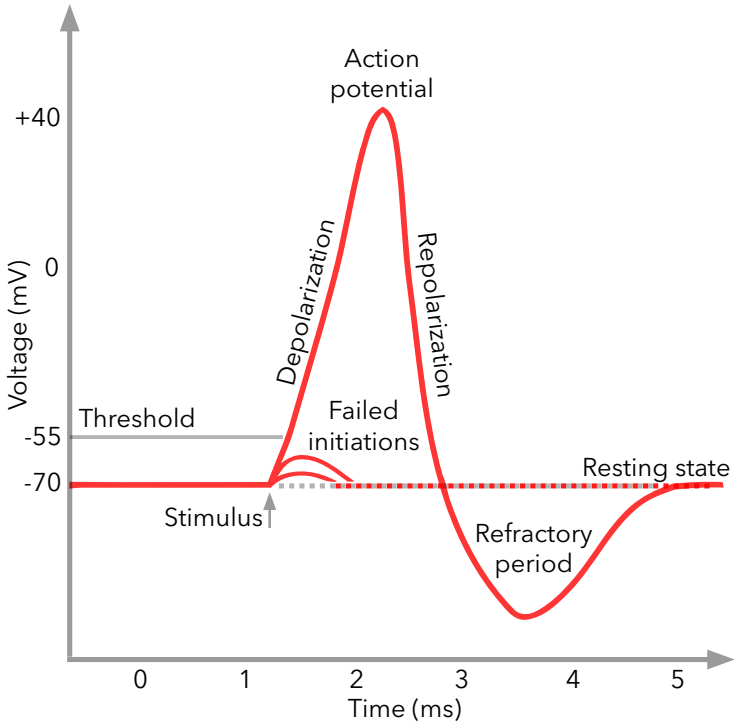


Figure 2.3: Idealized shape of an action potential, adapted from Guyton and Hall, 2006.

Neurons generate the impulse with an 'all or nothing' principle. Inputs from other neural cells have to be strong enough so that the membrane potential of the neuron reaches the threshold. Only then an AP will be generated. The shape of an AP is stereotypical and independent of the stimulus strength (Figure 2.3), however, the spiking frequency depends to a great extent on the stimulus strength. A neuronal AP is typically generated at the axon's initial segment, which is also in charge of propagating it to other cells. After an AP is emitted, the neuron usually can not spike again immediately. This so-called refractory period is divided in two phases of absolute and

relative insensibility. The period of absolute insensibility starts with the peak of the AP and it lasts ~0.4ms and in this period the emergence of another AP is not possible at all, whereas during the relative refractory period which lasts $1-2$ms it is very unlikely. An important consequence of the refractory period is that it is impossible that a neuron can emit more than one spike at the same time.

2.3 SPIKE DETECTION AND SPIKE SORTING

Experiments on brain activity, whether in vivo or in vitro, are typically done by placing electrodes in the region of interest. Most common techniques provide intracellular recordings of a membrane potential of a single isolated neuron (single-unit recordings) or measure the mean extracellular field potential generated by electrical activities of several (nearby) neurons (multi-unit recordings). The prerequisite for any further analysis is the extraction of the spike times from the measured membrane potential.

Under the assumption that both the shape of the spike and the background activity carry minimal information, neuronal responses are typically reduced to the much simpler form of a spike train, where the only information maintained is the timing of the single spikes. Typically, the *spike detection* is performed using some sort of threshold criterion (static or adaptive), either for the time series itself or for its derivative. Thereby the continuous time series is transformed into a discrete series of spikes.

In many experimental situations where the extra-cellular field potential is measured, the profiles contain spike events from more than one neuron (multi-unit recordings). Then, additionally to spike detection also *spike sorting* must be performed. This involves isolating the neural signals and assigning each recorded waveform to the neuron of origin (Lewicki, 1998). Spike sorting techniques (Martínez and Quiroga, 2013) are based on the fact that action potentials recorded from the same cell tend to have a stereotypical spike shape determined by the cell's morphology and biophysical properties, but also by its position relative to the recording electrode. Spike sorting is then usually done in two steps. The first step is feature extraction, where a number of signif-

icant features are obtained from the spike events, which allow to separate the different clusters afterwards. The second step is clustering, where the spikes are assigned into different groups corresponding to different neurons. The clustering is based on the features obtained before.

2.4 SPIKE TRAIN

The resulting sequence of spike times of a single neuron is then called a *spike train*, and is defined as:

$$s = \{t_i\}, \quad \text{with} \quad t_i < t_{i+1}, \tag{1}$$

where the t_i represent the times of the spikes. A plot in which the spikes of several spike trains are plotted versus time is called a raster plot (see Figure 2.4 for an example).

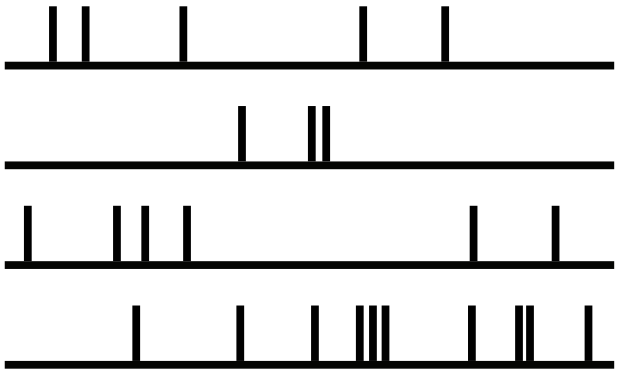


Figure 2.4: Example of a raster plot with four spike trains.

Part II

METHODS

3

MEASURES OF SPIKE TRAIN SYNCHRONY

Measures of spike train synchrony (or inversely spike train distances) are estimators of the (dis)similarity between two or sometimes more spike trains. Such measures are important tools for many applications. Among others, they can be used to quantify the reliability of responses upon repeated presentations of a stimulus (Mainen and Sejnowski, 1995), to evaluate the information transfer between synaptically coupled neurons (Reyes, 2003), or to test the performance of neuronal models (Jolivet et al., 2008).

Before we provide the very detailed definitions of the spike train distances which are central to this thesis we first introduce two standard measures that we use for comparison. The Peri-Stimulus Time Histogram (PSTH) is a measure of overall firing rate (Section 3.1.1), while the cross correlation is a measure of linear correlation (Section 3.1.2). Only then we describe the spike train distances which as such are divided in two groups, time-scale dependent (Section 3.2) or time-scale independent (Section 3.3).

3.1 STANDARD MEASURES

3.1.1 *Peri-Stimulus Time Histogram*

The first measure often applied to the spike trains is the Peri-Stimulus Time Histogram (PSTH) which we use in this thesis occasionally as a multivariate standard measure to compare against. However, it is important to note that strictly speaking it is not a measure of spike train synchrony but rather one of the most widely used firing rate

estimators. It is based on the average firing rate over some time interval. The most common way to estimate it is given by dividing the spike train into bins,

$$\mathrm{PSTH}(t) = \frac{\Delta N(t)}{\Delta t}, \tag{2}$$

where $\Delta N(t)$ is the number of spikes emitted in a time window $[t, t + \Delta t)$. Having more than one spike train, the resulting PSTH is just an average value across the trains. A common situation is one in which multiple trials are aligned to some stimulus marker (e.g. the start of a trial or some other behavioral significant event of interest). The resulting plot is known as a PSTH (Mitra and Bokil, 2007).

3.1.2 *Cross correlation*

After being able to obtain rate estimates (see previous Section) we also need a standard method that estimates the degree of (linear) correlation between spike trains.

For discrete data such as spike trains it reads

$$(t^{(1)} \star t^{(2)})[n] \stackrel{\text{def}}{=} \sum_{i=-\infty}^{\infty} t^{(1)}[i]\ t^{(2)}[i+n]. \tag{3}$$

From two de-meaned signals the linear correlation is calculated as

$$(t^{(1)} \star t^{(2)})[n] \stackrel{\text{def}}{=} \sum_{i=-\infty}^{\infty} (t^{(1)}[i]\ - \langle t^{(1)} \rangle)(t^{(2)}[i+n] - \langle t^{(2)} \rangle). \tag{4}$$

Finally, we obtain the normalized version, also called Pearson correlation coefficient,

$$r_{t^{(1)}t^{(2)}} = \frac{\sum_{i=1}^{n} (t_i^{(1)} - t^{(\bar{1})})(t_i^{(2)} - t^{(\bar{2})})}{\sqrt{\sum_{i=1}^{n} (t_i^{(1)} - t^{(\bar{1})})^2} \sqrt{\sum_{i=1}^{n} (t_i^{(2)} - t^{(\bar{2})})^2}} \tag{5}$$

or, equivalently, shorter

$$r_{t^{(1)}t^{(2)}} = \frac{\mathrm{cov}(t^{(1)}, t^{(2)})}{\sigma(t^{(1)})\sigma(t^{(2)})}. \tag{6}$$

This value is limited to the range $[0, 1]$ and attains a maximum value of 1 for completely correlated signals (also just scaled), a minimum value of -1 for completely anti-correlated signals, and values close to 0 for linearly independent (uncorrelated) signals.

In Chapter 6 the cross correlation will be one of two measures against which we compare regarding the performance to correct for latency variations.

3.2 TIME-SCALE DEPENDENT SPIKE TRAIN DISTANCES

Arguably the most prominent task for spike train analysis is to address questions regarding the nature of the neuronal code (for an overview, see Victor, 2005). This task is typically addressed by spike train distances which consider spike trains to be points in an abstract metric space and quantify their dissimilarity by non-negative values. Some of these spike train distances depend on a parameter which determines the temporal scale in the spike trains to which the distance is sensitive. While in one limit of the parameter range these distances are sensitive to the difference in spike number, they detect spike coincidences in the other limit. These limits reflect the characteristics of a rate code and a coincidence code, respectively.

In a typical experimental setup, single-unit responses are recorded for repeated presentations of a set of stimuli. The pairwise spike train distances between the different trials are calculated for different values of the timescale parameter. To each resulting distance matrix a cluster analysis is applied and the discrimination performance is calculated which measures how well the responses to different stimuli can be distinguished. The timescale τ_d for which the highest discrimination performance is obtained (i.e., the one which yields lowest distances between responses to the same stimulus and highest distances between responses to different stimuli) is assumed to be the discriminative precision of the neural code.

By far the most widely used time-scale dependent measure is the Victor–Purpura distance (Victor and Purpura, 1996; Victor and Purpura, 1997). Other distances with

a time scale are the van Rossum distance (van Rossum, 2001), and the Schreiber et al. similarity measure (Schreiber et al., 2003). Comparisons of these measures on simulated data can be found in (Schrauwen and Van Campenhout, 2007; Paiva et al., 2010; Chicharro et al., 2011. As it can also be seen in these three studies, all of these measures are very similar to each other. Thus, in this thesis the only time-scale dependent spike train distance that will be used for comparison in Chapter 6 is the Victor–Purpura distance.

3.2.1 *Victor-Purpura distance*

The Victor-Purpura distance (Victor and Purpura, 1996) is an adaptation of a distance between two finite strings - Levenshtein distance, that is counting the minimum cost needed to transform one string into the other (Levenshtein, 1966). Operations defined are insertion, deletion, and substitution of a single character[1]. It was designed for spell checking, but it is also used in genetics to measure variation between DNA.

One of the main differences is that Victor-Purpura applied to spike trains is based on a set of real numbers instead of a set of characters. Now positive real numbers represent the spike times of the neurons. The cost of insertion and deletion is by definition 1 unit and the cost of shifting the spike is q/unit which defines the time-scale of the method. While the most prominent modification is that a substitution has a weight, it is not the same unit cost as insertion or deletion. Boundaries for the Victor-Purpura can be easily calculated by setting the cost parameter q to zero and to infinity. The cost being zero allow the spikes to freely move without any cost, making the total distance equal to the difference in the number of spikes. Having the cost that limits to infinity is making spikes impossible to shift, making the total distance equal to the number of non-coincident spikes. Thus varying the cost parameter changes the method from a firing rate distance (for smaller cost parameters) to a precise timing

1 The Damerau - Levenshtein distance is based on the Levenshtein distance with one more operation - transposition of two adjacent characters (Damerau, 1964).

distance (for larger cost parameters). Since the method satisfies all metric conditions the Victor Purpura distance is a metric (Victor and Purpura, 1996). For more on metric properties please refer to the next section.

In addition to this distance which is sensitive to the timing of individual spikes, two complementary cost-based distances have been proposed. These distances are sensitive either to Inter Spike Intervals (ISIs) or to temporal patterns of so-called spike motifs (Victor and Purpura, 1997).

3.3 TIME-SCALE INDEPENDENT SPIKE TRAIN DISTANCES

Complementary to the time-scale dependent approaches, in recent years spike train distances have been proposed which are parameter-free and time-scale-adaptive. While not allowing the functional characterization and precision analysis described above, single-valued methods give an objective and comparable estimate of neuronal variability. They can be preferable in applications for real data for which there is no validated knowledge about the relevant time scales. The computational cost is reduced since there is no need for parameter optimization. In fact, it is not at all guaranteed that there exists an optimal parameter. For example, spike trains that include different time-scales such as regular spiking and bursting might result in misleading conclusions, since any fixed parameter will misrepresent either one of these dynamics.

For these reasons time-scale independent spike train distances form the class of measures that is implemented in the graphical user interface SPIKY which will be introduced in Chapter 5. These measures include the ISI-distance (Kreuz et al., 2007) and the SPIKE-distance (Kreuz et al., 2013) as well as SPIKE synchronization (Kreuz et al., 2015), an improved multivariate extension of event synchronization (Quian Quiroga et al., 2002).

These measures share several properties, however, there are also a few conceptual differences between the ISI- and the SPIKE-distance on the one hand, and SPIKE-synchronization on the other hand. All three measures rely on instantaneous values

which are normalized between zero and one. The same holds true for the respective temporal averages, the distance values D_I and D_S and the SPIKE-synchronization S_C. However, while the two distances are measures of dissimilarity which yield the value zero for identical spike trains, SPIKE-synchronization is a measure of similarity with high values denoting similar spike trains.

While all three measures are time-resolved, the ISI-distance and the SPIKE-distance even have a continuous domain since there exists a unique definition of an instantaneous value ($I(t)$ and $S(t)$, respectively, for every single time instant. The resulting dissimilarity profiles are either piecewise constant (ISI-distance) or piecewise linear (SPIKE-distance). SPIKE-synchronization is time-resolved as well, however, its domain is discrete since instantaneous values $C(t_k)$ are only defined at the times of the spikes. Incidentally, the same distinction holds true regarding the range of values that can be obtained for the measures: it is continuous for the two distances and discrete for SPIKE-synchronization.

All three measures can also be applied to more than two spike trains (spike train number $N > 2$). For the ISI- and the SPIKE-distance this extension is simply the average over all bivariate distances. Extending SPIKE-synchronization is even more straightforward. Essentially, the same spike-based definition holds for both the bivariate and the multivariate case.

Note that previous definitions of the three measures were to a large part not complete since they did not state explicitly how to define values at the edge of the spike trains, i.e. before the first spike and after the last spike of each spike train. Due to the finite recording time there is an ambiguity regarding the definitions of the initial distance to the preceding spike, the final distance to the following spike, as well as the very first and the very last ISIs. Here we deal with this issue and introduce edge corrections whose aim is to make use of as much of the available information as possible. Similarly, so far no information was provided about how to deal with special cases such as empty spike trains, and spike trains with just a single spike. Here we will complete the definitions in each of the three individual measure descriptions. Fol-

lowing Mulansky et al., 2015 we will state for each of the three measures whether it fulfills the three mathematical properties of a metric and derive the expectation values for Poisson spike trains (see Table 1).

The definition of a metric: the function is considered to be metric if it maps two points in space $t_i^{(1)}$ and $t_j^{(2)}$, into a real number $D(t_i^{(1)}, t_j^{(2)})$. We consider spike train distances to be metrics if they satisfy the following conditions:

1. Non-negativity with coincidence axiom

$$D(t_i^{(1)}, t_j^{(2)}) \geqslant 0 \tag{7}$$

with equality if and only if they are identical spike trains

$$t_i^{(1)} = t_j^{(2)}) \iff D(t_i^{(1)}, t_j^{(2)}) = 0 \tag{8}$$

2. Symmetry

$$D(t_i^{(1)}, t_j^{(2)}) = D(t_j^{(2)}, t_i^{(1)}), \tag{9}$$

3. Triangle inequality

$$D(t_i^{(1)}, t_j^{(2)}) \leqslant D(t_i^{(1)}, t_k^{(3)}) + D(t_k^{(3)}, t_j^{(2)}). \tag{10}$$

In addition to these three basic measures we will also describe the real-time (causal) and the forward variant as well as a new rate-independent version of the SPIKE-distance. First, the causal variant (Section 3.3.1.3) is defined such that the instantaneous values of dissimilarity are derived from past information only so that time-resolved spike train synchrony can be estimated in real-time (Kreuz et al., 2013). In neuronal population recordings being able to monitor spike train synchrony in realtime would be a necessary condition for a prospective epileptic seizure prediction algorithm (Mormann et al., 2007), but it could also be very useful for the rapid online decoding needed to control prosthetics (Hochberg et al., 2006; Sanchez et al., 2008). Second, the forward variant (Section 3.3.1.4) which instead relies on future information only could be used

	ISI: $I(t)$	SPIKE: $S(t)$	SPIKE-SYNC: C_k
Profile type	piecewise constant	piecewise linear	discrete
Integration	$\frac{1}{T}\int I(t)\,dt$	$\frac{1}{T}\int S(t)\,dt$	$1-\frac{1}{M}\sum_k C_k$
Codomain	$[0,1]$	$[0,1)$	$[0,1]$
Identity	0	0	0
Mean value (Poisson)	$\frac{1}{(1+r)^2}+\frac{1}{(1+r^{-1})^2}$	$\frac{1}{2}-\frac{1}{5}e^{-(\log r)^2/8}$	$1-\frac{1}{r^{-1}+r+2}$

Table 1: : Overview of the mathematical properties of different spike train synchronization measures (Mulansky et al., 2015).

in triggered temporal averaging in order to evaluate the (causal) effect of certain spikes or of specific stimuli features on future spiking. Third and finally, the SPIKE-distance is dependent on the ratio between firing rates of spike trains (Mulansky et al., 2015), however, in many applications it can be desirable to use a rate independent measure. To this aim, in Section 3.3.1.5 we introduce a variant of SPIKE-distance that is indeed rate independent.

3.3.1 *The ISI- and the SPIKE-distance*

The first step in the calculation of the ISI- and the SPIKE-distance is to transform the sequences of discrete spike times $\{t_i^{(n)}\}, i = 1, ..., M_n$(with M_n denoting the number of spikes for spike train n) into dissimilarity profiles $I(t)$ and $S(t)$, respectively. Averaging over time yields the respective distance value.

The multivariate extension is obtained as the average over all bivariate distances. Since this average over all pairs of spike trains commutes with the average over time, it is possible to achieve the same kind of time-resolved visualization as in the bivariate

case by first calculating the instantaneous average $S^a(t)$ (here for the SPIKE-distance) over all pairwise instantaneous values $S^{mn}(t)$,

$$S^a(t) = \frac{2}{N(N-1)} \sum_{m=1}^{N-1} \sum_{n=m+1}^{N} S^{mn}(t). \tag{11}$$

The dissimilarity profiles of both measures are based on three piecewise constant quantities (see Figure 3.1). These are the time of the preceding spike

$$t_P^{(n)}(t) = \max(t_i^{(n)} | t_i^{(n)} \leqslant t) \quad t_1^{(n)} \leqslant t \leqslant t_{M_n}^{(n)}, \tag{12}$$

the time of the following spike

$$t_F^{(n)}(t) = \min(t_i^{(n)} | t_i^{(n)} > t) \quad t_1^{(n)} \leqslant t \leqslant t_{M_n}^{(n)}, \tag{13}$$

as well as the ISI

$$x_{ISI}^{(n)}(t) = t_F^{(n)}(t) - t_P^{(n)}(t). \tag{14}$$

The ambiguity regarding the definition of the very first and the very last ISI is resolved by placing for each spike train auxiliary leading spikes at time $t = 0$ and auxiliary trailing spikes at time $t = T$).

3.3.1.1 *The ISI-distance*

The ISI-distance, proposed as a bivariate measure in Kreuz et al., 2007 and extended to the multiple spike train case in Kreuz et al., 2009 was the first spike train distance directly defined as the temporal average of an instantaneous dissimilarity profile. This profile is calculated as the absolute value of the instantaneous ratio between the ISIs $x_{ISI}^{(1)}$ and $x_{ISI}^{(2)}$ (see Figure 3.1) according to:

$$I(t) = \frac{|x_{ISI}^{(1)}(t) - x_{ISI}^{(2)}(t)|}{\max(x_{ISI}^{(1)}(t), x_{ISI}^{(2)}(t))} \tag{15}$$

Since the ISI-values only change at the times of spikes, the dissimilarity profile is piecewise constant (with discontinuities at the spikes). The ISI-ratio equals zero for

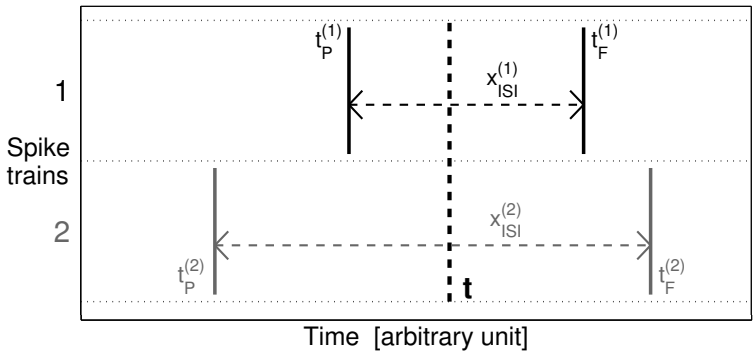


Figure 3.1: ISI-distance. Illustration of the local quantities used to define the dissimilarity profile $I(t)$ for an arbitrary time instant t.

identical ISIs in the two spike trains, and approaches one in intervals in which one spike train is much faster than the other. The ISI-distance is defined as the temporal average of this absolute ISI-ratio:

$$D_I = \frac{1}{T}\int_0^T dt I(t). \tag{16}$$

EDGE CORRECTION For the first ISI we use the maximum of the distance between the start of the observation interval and the first spike, and the first known ISI:

$$x_{ISI}(0 \leqslant t < t_1) = \max\{t_1, t_2 - t_1\}. \tag{17}$$

Similarly for the last ISI:

$$x_{ISI}(t_M \leqslant t < T) = \max\{T - t_M, t_M - t_{M-1}\}. \tag{18}$$

If the first or last spike is at the edge, no edge correction is necessary. However, the above definitions still apply, as the first term in the maximum always becomes zero.

SPECIAL CASES: EMPTY SPIKE TRAIN OR SPIKE TRAINS WITH A SINGLE SPIKE If a spike train is empty, we simply use the length of the spike train as ISI, e.g.

$$x_{ISI} = T. \tag{19}$$

We omitted the superscript indicating the spike train (1) or (2) for brevity noting that these definitions apply to either or both of the two spike trains if they are empty. Note furthermore, that this is equivalent to adding auxiliary spikes at t_s and t_e and use the definition from Equation 14.

If a spike train contains only a single spike, e.g. $M_1 = 1$, we use the distances to the beginning and end of the spike train as ISIs:

$$x_{ISI}(0 \leqslant t < t_1) = t_1, \qquad \text{and} \qquad x_{ISI}(t_1 \leqslant t < T) = T - t_1. \tag{20}$$

This is again equivalent to adding auxiliary spikes at $t = 0$ and $t = T$ and use the definition from Equation 14. Note that this definition also applies if the single spike is at one of the edges.

METRIC PROPERTIES From the normalization in Equation 15, it is immediately obvious that the ISI-profile is bounded by $0 \leqslant I(t) < 1$, and the value $I(t) = 0$ is only obtained for identical ISIs $\nu^{(1)}(t) = \nu^{(2)}(t)$. Consequently, also the ISI-distance is bound: $0 \leqslant D_I < 1$. The value of exactly zero, $D_I = 0$, is only reached for identical spike trains. Furthermore, the ISI-distance is clearly symmetric: $D_I(s_1, s_2) = D_I(s_2, s_1)$ by construction. Finally, it can be shown that the ISI-distance also fulfills the triangle inequality: $D_I(s_1, s_2) + D_I(s_2, s_3) \geqslant D_I(s_1, s_3)$, a proof is presented in Appendix B of Lyttle and Fellous, 2011. It follows, that mathematically the ISI-distance is a *metric*[2], and the set of all spike trains on some interval $[T_1, T_2]$ together with D_I form a *metric space*.

EXPECTATION VALUE FOR POISSON SPIKE TRAINS WITH $r = \lambda_1/\lambda_2$ For the ISI-distance it is possible to compute the expectation value $\langle D_I \rangle$ for random Poisson spike trains analytically. Poisson spike trains are determined by a single parameter: the rate λ. Hence, the average ISI-distance will be a function of the two rates of the spike trains $\langle D_I(\lambda_1, \lambda_2) \rangle$. However, note that as the ISI-distance is

2 To be precise, when the edge-correction is employed, the value $D_I = 0$ can also be found for two perfectly periodic but shifted spike trains. These shifted spike trains can be considered as an equivalent class thus making the ISI-distance a 'pseudo-metric' in the strict mathematical sense.

independent on the global time scale, it can only depend on the ratio of the two rates: $r = \lambda_1/\lambda_2$. A straightforward calculation for Poisson spike trains with rate ratio r (see Appendix A in Mulansky et al., 2015) indeed gives:

$$\langle D_I(r)\rangle = \frac{1}{(1+r)^2} + \frac{1}{(1+r^{-1})^2}. \tag{21}$$

Note how $\langle D_I(r)\rangle$ is symmetric in r and r^{-1}, a direct consequence from the fact that D_I is symmetric in s_1 and s_2. For Poisson spike trains with equal rates, $r = 1$, we thus find $\langle D_I\rangle = 1/2$, while in the limit where one spike train is much faster than the other one, $r \to 0, \infty$ we have $\langle D_I\rangle \to 1$. This is visualized in Figure 3.2.

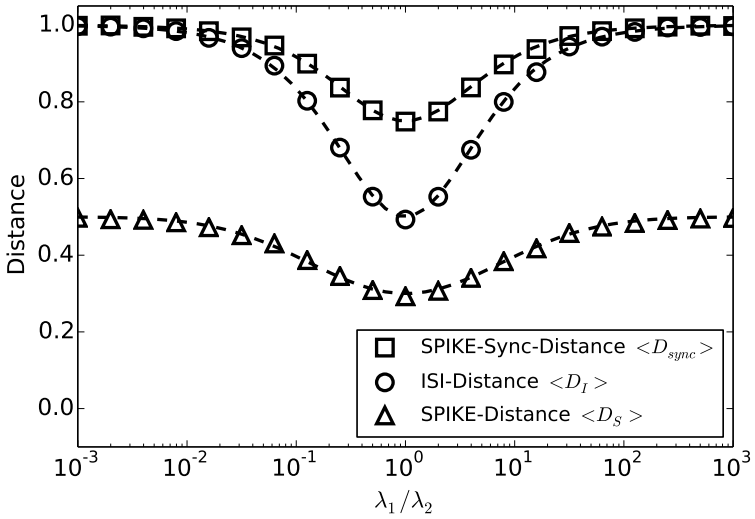


Figure 3.2: Numerical results for ISI-distance (circles), SPIKE-distance (triangles) and SPIKE-Synchronization-distance (squares) of two Poisson spike trains with a total of $M \approx 20000$ spikes in dependence of the rate ratio $r = \lambda_1/\lambda_2$. The black dashed lines represent the analytical result for the ISI-distance Equation 21 the empirical curve for the SPIKE-distance Equation 31, and the analytical result for the SPIKE-Synchronization distance Equation 42

3.3.1.2 *The SPIKE-distance*

The SPIKE-distance (see Kreuz et al., 2011, for the original proposal and Kreuz et al., 2013; Kreuz, 2012, for the definite version presented here) is the centerpiece of SPIKY. In contrast to the ISI-distance, it considers the exact timing of the spikes.

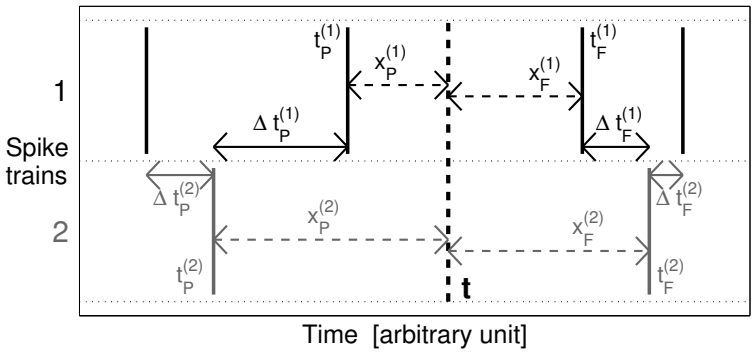


Figure 3.3: SPIKE-distance. Illustration of the additional local quantities needed for the calculation of the dissimilarity profile $S(t)$.

The dissimilarity profile is calculated in two steps: First for each spike a spike time difference is calculated, then for each time instant the relevant spike time differences are selected, weighted, and normalized. Here 'relevant' means local; each time instant is surrounded by four corner spikes: the preceding spike from the first spike train $t_P^{(n)}$, the following spike from the first spike train $t_F^{(n)}$, the preceding spike from the second spike train $t_P^{(m)}$, and, finally, the following spike from the second spike train $t_F^{(m)}$. To each of these corner spikes one assigns the distance to the nearest spike in the other spike train, for example, for the previous spike of the first spike train:

$$\Delta t_P^{(n)}(t) = \min_i(|t_P^{(n)}(t) - t_i^{(m)}|) \tag{22}$$

and analogously for $t_F^{(n)}$, $t_P^{(m)}$, and $t_F^{(m)}$ (see Figure 3.3). Subsequently, for each spike train separately, a locally weighted average is employed such that the differences for

the closer spike dominate; for each spike train the weighting factors are the intervals to the previous and to the following spikes:

$$x_P^{(n)}(t) = t - t_P^{(n)}(t) \tag{23}$$

and

$$x_F^{(n)}(t) = t_F^{(n)}(t) - t. \tag{24}$$

The local weighting for the spike time differences of the first spike train reads:

$$S_n(t) = \frac{\Delta t_P^{(n)}(t) x_F^{(n)}(t) + \Delta t_F^{(n)}(t) x_P^{(n)}(t)}{x_{ISI}^{(n)}(t)}, \tag{25}$$

and analogously $S_2(t)$ is obtained for the second spike train. Averaging over the two spike train contributions and normalizing by the mean ISI yields:

$$S''(t) = \frac{S_n(t) + S_m(t)}{2\langle x_{ISI}^{(n)}(t)\rangle_n}. \tag{26}$$

This quantity sums the spike time differences for each spike train weighted according to the relative distance of the corner spike from the time instant under investigation. This way relative distances within each spike train are taken care of, while relative distances between spike trains are not. In order to get these ratios straight and to account for differences in firing rate, in a last step the two contributions from the two spike trains are locally weighted by their instantaneous ISI. This leads to the definition of the dissimilarity profile:

$$S(t) = \frac{S_1(t) x_{ISI}^{(2)}(t) + S_2(t) x_{ISI}^{(1)}(t)}{2\langle x_{ISI}^{(n)}(t)\rangle_n^2}. \tag{27}$$

Again, the overall distance value is defined as the temporal average of the dissimilarity profile:

$$D_S = \frac{1}{T}\int_0^T dt S(t). \tag{28}$$

Since the dissimilarity profile $S(t)$ is obtained from a linear interpolation of piecewise constant quantities, it is piecewise linear (with potential discontinuities at the spikes). Both the dissimilarity profile $S(t)$ and the SPIKE-distance D_S are bounded in the interval $[0, 1]$. The distance value $D_S = 0$ is obtained for identical spike trains only.

EDGE CORRECTION To resolve the ambiguity at the edges, i.e. before the first spike t_1 (which can be either from the first or the second spike train) and after the last spike t_M, we define the spike time differences in these intervals as the one of the first (last) actual spike, e.g. for the first spike train:

$$\Delta t_P^{(1)}(t_s \leqslant t < t_1) = \Delta t_P^{(1)}(t_1) \qquad \text{and} \qquad \Delta t_F^{(1)}(t_{M_1} \leqslant t < t_e) = \Delta t_F^{(1)}(t_{M_1}), \tag{29}$$

and similarly for the second spike train. Additionally, for computing $\Delta t_P^{(1)}(t_1)$ we add an auxiliary spike at the beginning t_s of the second spike train as a potentially closest spike to $t_1^{(1)}$. Accordingly, an auxiliary spike is added at the end t_e of the second spike train as potential closest spike to $t_{M_1}^{(1)}$. Similarly, auxiliary spikes are added to the first spike train as potentially closest spikes to $t_1^{(2)}$ and $t_{M_2}^{(2)}$. Finally, for the ISIs at the beginning and end we use the same edge correction as for the ISI-distance above given in Equation 17 and Equation 18. This completes the definitions at the edges and allows to consistently compute the SPIKE-distance profile in the first and last interval.

SPECIAL CASES: EMPTY SPIKE TRAIN OR SPIKE TRAINS WITH A SINGLE SPIKE In cases where spike trains have less than two spikes we perform the following procedure. Suppose the first spike train has less then two spikes, then we define for both start and end:

$$\Delta t_P^{(1)}(t_s) = 0, \qquad \text{and} \qquad \Delta t_F^{(1)}(t_e) = 0. \tag{30}$$

Together with the corresponding definition for the ISIs in this case, cf. Equation 19 and Equation 20, we have all contributions of the first spike train to the SPIKE-distance. The second spike train is treated in the same way if it has less than two spikes.

METRIC PROPERTIES The normalization in Equation 27 again ensures that the SPIKE-profile is bound to $0 \leqslant S(t) \leqslant 1$. Hence, the same bounds hold for the SPIKE-distance $0 \leqslant D_S = \int S(t)dt < 1$. Furthermore, $D_S(s_1, s_2) = 0$ only if $s_1 = s_2$,

and the SPIKE-distance is also symmetric $D_S(s_1, s_2) = D_S(s_2, s_1)$. However, it is not a metric as it is possible to construct spike trains that violate the triangular inequality: $D_S(s_1, s_2) + D_S(s_2, s_3) \not\geq D_S(s_1, s_3)$.

EXPECTATION VALUE FOR POISSON SPIKE TRAINS WITH $r = \lambda_1/\lambda_2$ Unfortunately, the complicated definition of the SPIKE-distance makes an analytic computation of the expectation value $\langle D_S \rangle$ for Poisson spike trains intractable. However, it is clear that the SPIKE-distance should again only depend on the rate ratio $r = \lambda_1/\lambda_2$ and approach the value $D_S = 0.5$ in the limit $r \to 0, \infty$. This is seen in Figure 3.2. Similarly to the ISI-distance and SPIKE-Synchronization, the SPIKE-distance exhibits a clear minimum for spike trains with equal rates $\lambda_1 = \lambda_2$, ie. $r = 1$. As we are unable to obtain an exact analytic result for $\langle D_S \rangle$ at this point, we provide an empirical approximation. Therefore, we use the following function, which already incorporates the properties mentioned above ($\lim_{r\to 0,\infty} \langle D_S \rangle = 0.5$, minimum at $r = 1$):

$$\langle D_S \rangle = \frac{1}{2} - \alpha e^{-(\log r)^2/(2\beta^2)}. \tag{31}$$

From visual inspection, we find that with $\alpha = 0.2$, $\beta = 2$, Equation 31 provides a good approximation of the average SPIKE-distance for Poisson spike trains as shown in Figure 3.2.

3.3.1.3 *Realtime SPIKE-distance*

In contrast to the dissimilarity profile $S(t)$ of the regular SPIKE-distance (Kreuz et al., 2013), the dissimilarity profile $S_r(t)$ of the realtime SPIKE-distance can be calculated online because it relies on past information only. From the perspective of an online measure, the information provided by the following spikes, both their position and the length of the ISI, is not yet available. Like the profile of the regular SPIKE-distance, this causal variant is also based on local spike time differences but now only two corner spikes are available, and the spikes of comparison are restricted to past spikes, e.g., for the preceding spike of the first spike train as shown in Equation 22. Since there

are no following spikes available, there is no local weighting. There is no ISI either, so the normalization is achieved by dividing the average corner spike difference by twice the average time interval to the preceding spikes (Equation 23). This yields a causal indicator of local spike train dissimilarity:

$$S_r(t) = \frac{\Delta t_P^{(1)}(t) + \Delta t_P^{(2)}(t)}{4\langle x_P^{(n)}(t)\rangle_n}. \tag{32}$$

3.3.1.4 *Forward SPIKE-distance*

The dissimilarity profile $S_f(t)$ of the forward SPIKE-distance (Kreuz et al., 2015) is 'inverse' to the profile of the realtime SPIKE-distance. Instead of relying on past information only, it relies on forward information only. It can be used in triggered temporal averaging in order to evaluate the (causal) effect of certain spikes or of specific stimuli features on future spiking. Again, for each time instant there are just two corner spikes and the potential nearest spikes in the other spike train are future spikes only. Thus, the spike time difference for the following spike of the first spike train reads analogously to Equation 22 and accordingly for the following spike of the second spike train. In analogy to Equation 32, an indicator of local spike train dissimilarity is obtained as follows:

$$S_f(t) = \frac{\Delta t_F^{(1)}(t) + \Delta t_F^{(2)}(t)}{4\langle x_F^{(n)}(t)\rangle_n}. \tag{33}$$

3.3.1.5 *Rate-independent SPIKE-distance*

The SPIKE-distance is dependent on the ratio between firing rates of spike trains (Mulansky et al., 2015), that can be undesirable property in some analysis, where one is interested in spike time related behavior only, and does not want any influence of spike rates. Therefore, we also provide its rate-independent variant. It is based on the SPIKE-distance so all variables are calculated in the same manner, as well as the edge

correction, and special cases, such as empty spike trains or spike trains with a single spike. The rate-independent SPIKE distance is defined as:

$$S(t) = \frac{S_1(t) + S_2(t)}{2\langle x_{ISI}^{(n)}(t)\rangle_n}. \tag{34}$$

As seen in Figure 3.4, the dependence on ratio does not appear anymore. However, in cases where the ratio of the firing rates actually carries information and is thus a desirable influence, it remains of course still possible to use the regular SPIKE-distance.

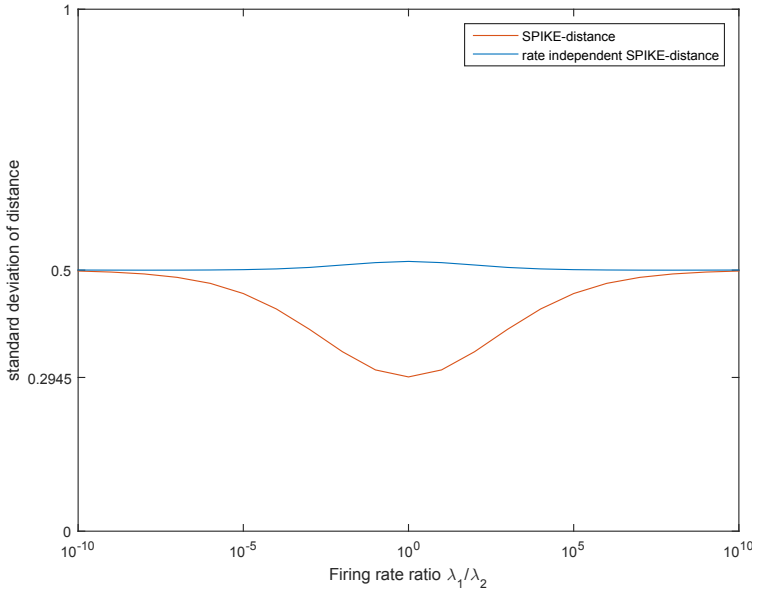


Figure 3.4: SPIKE-distance: Original vs Rate independent

3.3.2 *SPIKE-synchronization*

SPIKE-synchronization (Kreuz et al., 2015) quantifies the degree of synchrony from the relative number of quasi-simultaneous appearances of spikes. Since it builds on the same bivariate and adaptive coincidence detection that was used for event synchro-

nization (Quian Quiroga et al., 2002; see also Kreuz et al., 2007), SPIKE-synchronization is parameter- and scale-free as well.

Coincidence detection typically uses a coincidence window $\tau_{ij}^{(1,2)}$ which denotes the time lag below which two spikes from two different spike trains, $t_i^{(1)}$ and $t_j^{(2)}$, are considered to be coincident. For both event synchronization and SPIKE synchronization this coincidence window is adapted to the local spike rates (see Figure 3.5):

$$\tau_{ij}^{(1,2)} = \min\{t_{i+1}^{(1)} - t_i^{(1)}, t_i^{(1)} - t_{i-1}^{(1)}, t_{j+1}^{(2)} - t_j^{(2)}, t_j^{(2)} - t_{j-1}^{(2)}\}/2. \tag{35}$$

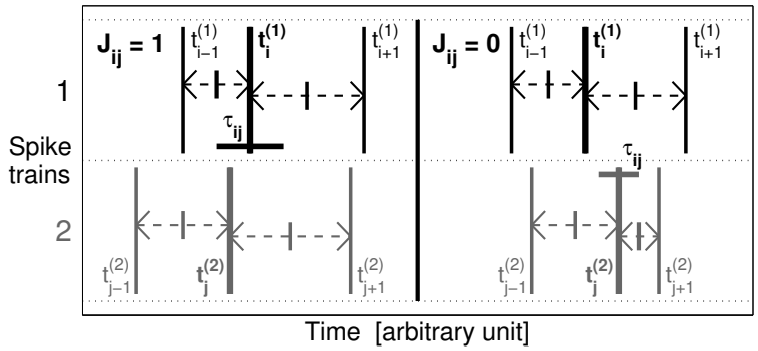


Figure 3.5: SPIKE-synchronization. Illustration of the adaptive coincidence detection (which was originally proposed for event synchronization). While in the first half the middle spikes $t_i^{(n)}$ and $t_j^{(m)}$ are coincident, the middle spikes in the second half are not.

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

$$C_i^{(1)} = \begin{cases} 1 & \text{if } \min_j(|t_i^{(1)} - t_j^{(2)}|) < \tau_{ij}^{(1,2)} \\ 0 & \text{otherwise} \end{cases} \tag{36}$$

(and analogously for $C_j^{(2)}$) which assigns to each spike either a one or a zero depending on whether it is part of a coincidence or not. The minimum function in Equation 36 takes already into account that a spike can at most be coincident with one spike (the nearest one) in the other spike train. This is a consequence of the adaptive definition of $\tau_{ij}^{(1,2)}$ in Equation 35 and the '$<$' in Equation 36 (which has been changed from the

'$\leqslant$' used in the original definition of event synchronization). In case a spike is right in the middle between two spikes from the other spike train there is no ambiguity any more since there is no coincidence.

This way we have defined a coincidence indicator for each individual spike of the two spike trains. In order to obtain one combined similarity profile we pool the spikes of the two spike trains as well as their coincidence indicators by introducing one overall spike index k. In case there exists, exact matches (pairs of perfectly coincident spikes) k counts over both spikes. This yields one unified set of coincidence indicators C_k in which according to Equation 35 and Equation 36 each coincidence leads to a pair of consecutive ones.

From this discrete set of coincidence indicators 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

$$S_C = \frac{1}{M} \sum_{k=1}^{M} C(t_k) \tag{37}$$

with $M = M_1 + M_2$ denoting the total number of spikes in the pooled spike train. The interpretation is very intuitive: S_C quantifies the fraction of all spikes in the two spike trains that are coincident. It is zero for spike trains without any coincidences, and reaches one if and only if the two spike trains consist only of pairs of coincident spikes.

The extension to the case of more than two spike trains ($N > 2$) is straightforward. First, bivariate coincidence detection is performed for each pair of spike trains (n, m). Generalizing Equation 36 gives the coincidence indicators

$$C_i^{(n,m)} = \begin{cases} 1 & \text{if } \min_j(|t_i^{(n)} - t_j^{(m)}|) < \tau_{ij}^{(n,m)} \\ 0 & \text{otherwise} \end{cases} \tag{38}$$

where $\tau_{ij}^{(n,m)}$ is defined as in Equation 35, but for arbitrary spike trains n and m. Subsequently, for each spike of every spike train a normalized coincidence counter

$$C_i^{(n)} = \frac{1}{N-1} \sum_{m \neq n} C_i^{(n,m)}. \tag{39}$$

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

As in the bivariate case, pooling leads to just one set of normalized coincidence counters C_k each of which can obtain any one out of this finite set of values: $0, 1/(N-1), ..., (N-1)/(N-1) = 1$. Again, the multivariate SPIKE-Synchronization profile $C(t_k)$ is obtained via $C(t_k) = C(k)$ and its average value yields the multivariate SPIKE-Synchronization

$$S_C = \frac{1}{M} \sum_{k=1}^{M} C(t_k) \tag{40}$$

where $M = \sum_n^N M_n$ again denotes the overall number of spikes.

Note that Equation 40 is completely analogous to Equation 37. Moreover, setting $n = 1$, $m = 2$, and $N = 2$ in Equation 38 and Equation 39 retrieves Equation 36. Therefore, the bivariate equations are just a special case of the more general multivariate formulation.

Accordingly, the interpretation of SPIKE synchronization as the overall fraction of coincident spikes is general and holds for both bivariate and multivariate datasets. SPIKE-Synchronization is zero if and only if the spike trains do not contain any coincidences, and reaches one if and only if each spike in every spike train has one matching spike in all the other spike trains. Examples for both of these extreme cases can be found in subplots B and D of Figure 3.6.

EXTENDED CAPTION OF FIGURE 3.6: Comparison of PSTH and SPIKE-synchronization. For the latter we added a dashed similarity profile which serves as a visual aid only. A | Multivariate example with 50 spike trains. In the first half within the noisy

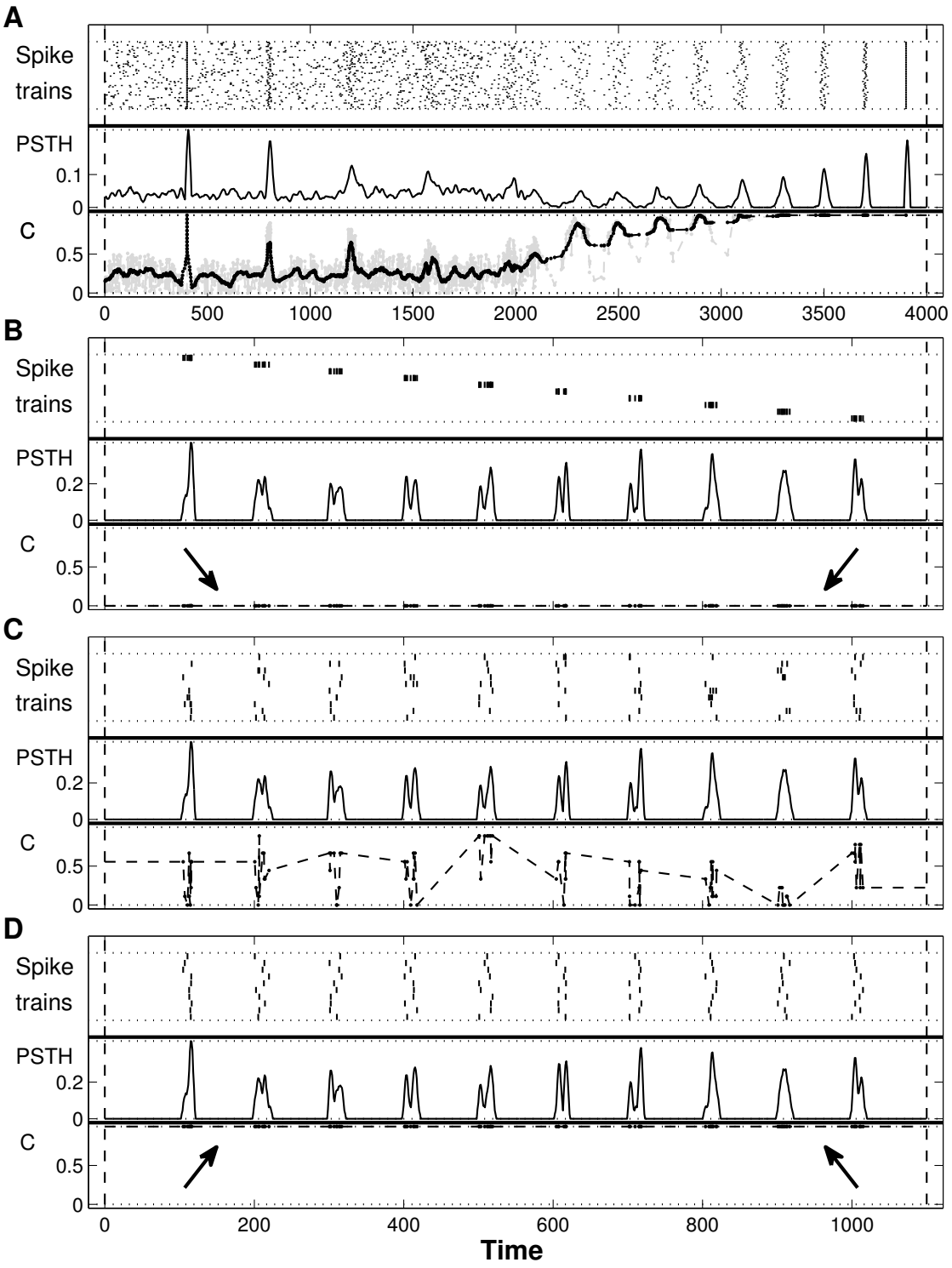


Figure 3.6: Comparison of PSTH and SPIKE-synchronization. A | Multivariate example with 50 spike trains. B | Synfire chain of bursts. C | Random distribution of spikes among spike trains. D | High reliability. Each spike train contains one spike per firing event. More detailed explanation in the paragraph below. The only difference is the distribution of the spikes among the individual spike trains which varies from low via intermediate to high synchrony. Whereas the PSTH is the same for all three examples, SPIKE-synchronization correctly indicates the increase in synchrony

background there are 4 regularly spaced spiking events with increasing jitter. The second half consists of 10 spiking events with decreasing jitter but now without any noisy background. In the noisy first half PSTH and the smoothed SPIKE-synchronization exhibit very similar profiles. The fact that the firing events become more distinct in the second half is indicated by the smoothed SPIKE-synchronization as a gradual increase to synchronization. In the PSTH the peaks become more and more narrow. B-D| 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 low via intermediate to high synchrony. Whereas the PSTH is the same for all three examples, SPIKE-synchronization correctly indicates the increase in synchrony (note that in subplot B SPIKE-synchronization attains the value zero and in subplot D the value one over the whole time interval, see arrows). B| Synfire chain of bursts. C| Random distribution of spikes among spike trains. D| High reliability. Each spike train contains one spike per firing event.

In contrast to $I(t)$ and $S(t)$, the time-resolved SPIKE-synchronization $C(t_k)$ is a measure of similarity. The Peri-Stimulus Time Histogram is oriented the same way so it makes sense to compare these two measures (Figure 3.6). The SPIKE-Synchronization profile $C(t_k)$ is only defined at the times of the spikes but a better visualization can be achieved by connecting the individual dots. For larger spike train datasets (here example A) it also makes sense to smooth the profile with a moving average of appropriate order. Figure 3.6A demonstrates the similarities and dissimilarities between the PSTH and SPIKE-synchronization on a rather general example. Figure 3.6B-D shows that SPIKE-synchronization, in contrast to the PSTH, is a true measure of spike train synchrony (see also Kreuz et al., 2011). Since the PSTH is invariant to shuffling spikes among the spike trains, it yields the same value regardless of how spikes are distributed among the different spike trains.

EDGE CORRECTION As the SPIKE-Synchronization is only defined at the spike times, we only have to take care of the coincidence windows for the first (last) spike in the spike trains. There, as we don't have the distance to the previous (following)

spike train we simply neglect these and only use the following (previous) ISIs to compute the adaptive coincidence window τ.

SPECIAL CASES: EMPTY SPIKE TRAIN OR SPIKE TRAINS WITH A SINGLE SPIKE In case we have empty spike trains, we artificially define the SPIKE-Synchronization to be $C = 1$, i.e. empty spike trains are considered to be synchronous.

If a spike train contains only a single spike, we simply use the spike train interval to define the coincidence window, i.e.

$$\tau = \frac{T}{2}. \tag{41}$$

METRIC PROPERTIES In contrast to the ISI- and SPIKE-profiles that quantify dissimilarity, the SPIKE-Synchronization profile is a similarity measure. Furthermore, the bivariate SPIKE-Synchronization profile is two-valued, $S_{sync} = 0, 1$ representing the absence or presence of a coincidence. The overall SPIKE-Synchronization value Sync can take M possible values in $[0, 1]$ (with M being the total number of spikes), where $SYNC = 0$ means no coincidences exist, while for $Sync = 1$ all spikes are coinciding. The SPIKE-Synchronization distance $D_{sync} = 1 - Sync$ consequently has the same properties, but with inverted meaning of the values 0, 1. It is immediately clear that the SPIKE-Synchronization distance is not a metric, as one finds $D_{sync}(s_1, s_2) = 0$ even if $s_1 \neq s_2$. Furthermore, also transitivity is violated as one easily finds examples where $D_{sync}(s_1, s_2) = 0$ and $D_{sync}(s_2, s_3) = 0$, but $D_{sync}(s_1, s_3) > 0$.

EXPECTATION VALUE FOR POISSON SPIKE TRAINS WITH $r = \lambda_1/\lambda_2$ For spike trains with Poisson distribution that have rates λ_1 and λ_2 we again expect the SPIKE-Synchronization distance to only depend on the ratio $r = \lambda_1/\lambda_2$ and to be symmetric in r and r^{-1}. Indeed, an analytical calculation of the SPIKE-Synchronization distance of Poisson spike trains (see Appendix B in Mulansky et al., 2015) gives:

$$\langle D_{sync} \rangle = 1 - \frac{1}{r^{-1} + 2 + r}. \tag{42}$$

Also this dependence is visualized in Figure 3.2.

3.4 COMPARISON OF MEASURES

One of the main arguments for the use of time-scale-dependent measures of spike train (dis)similarity is their potential insight into the precision of the neuronal code (Victor and Purpura, 1996). This argument has recently been re-evaluated in Chicharro et al., 2011. According to this study the optimal time-scale obtained from the cluster analysis is far from being conclusive. Rather it results in a non-trivial way from the interplay of many different factors such as the distribution of the information contained in different parts of the response and the degree of redundancy between them. Despite these problems in the interpretation of the optimal timescale, the Victor–Purpura and the van Rossum distance are designed to test for neuronal codes ranging from a rate code to a coincidence detector. This is a level of generality somewhere in between two extremes. While some methods evaluate a very specific coding hypothesis (e.g., the classical correlation coefficient based on binning which focuses purely on coincidences), other methods are more general (e.g., the ISI- and the SPIKE-distances which are time-scale-adaptive and parameter-free). Measures on different ends of this scale are complementary in nature. If a particular coding scheme is assumed, specific measures are needed for a confirmatory analysis, otherwise more general measures are very well suited for an exploratory analysis (Kreuz et al., 2011).

4

LEVELS OF INFORMATION EXTRACTION

The ISI- and the SPIKE-distance combine a variety of properties that make them well suited for the application to real data. In particular, they are conceptually simple, computationally efficient, and easy to visualize in a time-resolved manner. By taking into account only the preceding and the following spike in each spike train, these distances rely on local information only. They are also time-scale-adaptive since the information used is not contained within a window of fixed size but rather within a time frame whose size depends on the local rate of each spike train.

Moreover, the sensitivity to spike timing and the instantaneous reliability achieved by the SPIKE-distance opens up many new possibilities in multi-neuron spike train analysis (Kreuz et al., 2013). These build upon the fact that there are several ways to extract information all of which we describe in the following. As an illustration we use the detailed analysis of an artificially generated spike train dataset with the SPIKE-distance (for the raster plot see upper part of Figure 4.1A).

Since the profile of SPIKE-synchronization is only defined at the times of the spikes and not at any time instants, for this measure only a few of these levels apply as we will explain in more detail below.

EXTENDED CAPTION OF FIGURE 4.1: The different levels of information extraction for the SPIKE-distance. A | Top: Spike raster plot of 20 artificially generated spike trains divided in 4 spike train groups of 5 spike trains each. The clustering behavior changes every 500 ms. Bottom: Dissimilarity profiles of the SPIKE-distance for the four spike train groups (thin color-coded lines) and for all spike trains (thick black line). The overall dissimilarity is defined as the temporal average of the dissimilarity

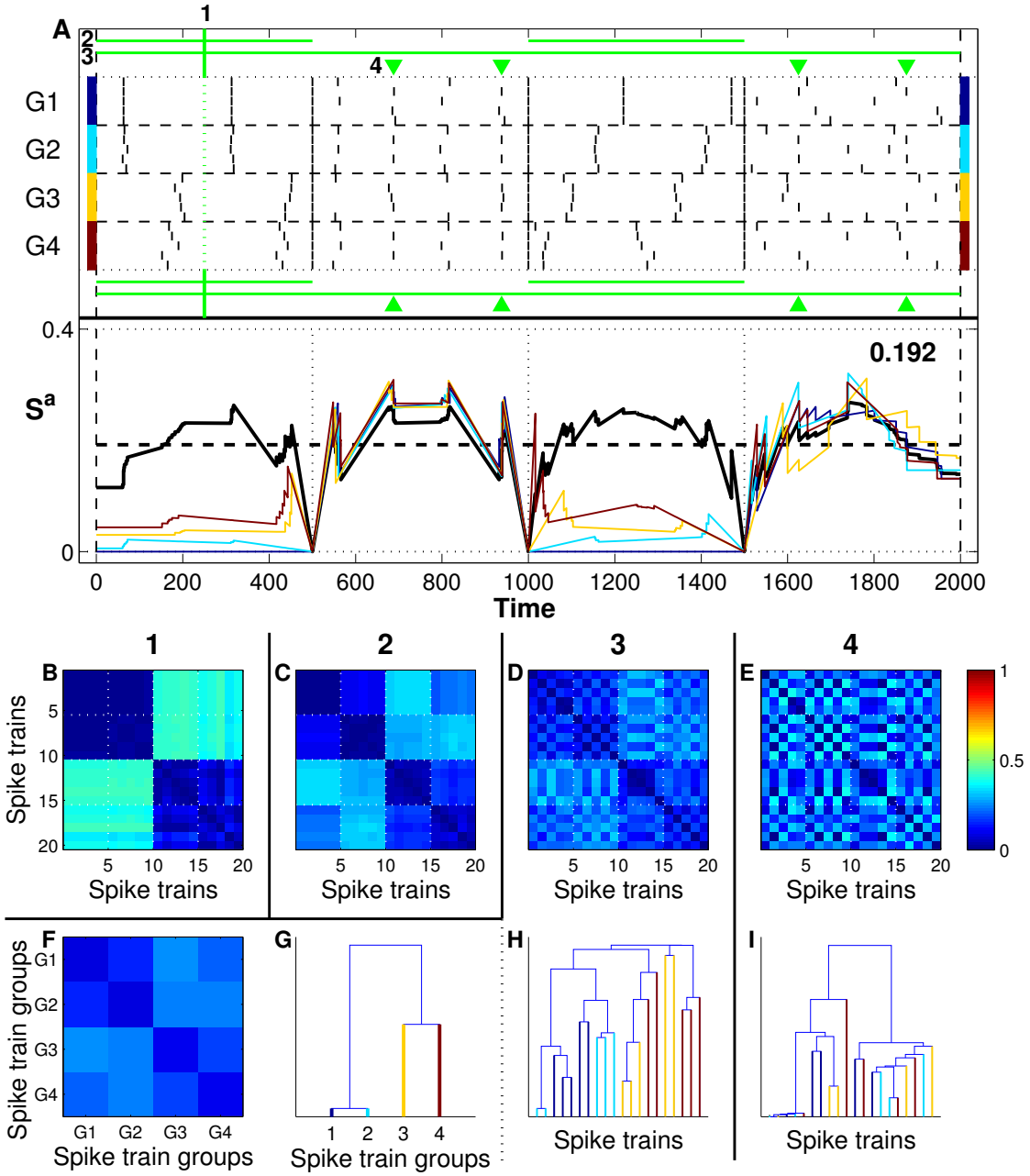


Figure 4.1: The different levels of information extraction for the SPIKE-distance. A | Top: Spike raster plot. Bottom: Dissimilarity profiles of the SPIKE-distance for the different spike train groups. B-E | Matrices of pairwise instantaneous dissimilarity values. F-G | Matrices of overall pairwise instantaneous dissimilarity values for all spike train groups (F) and the corresponding dendrogram (G). H-I | Dendrogram of spike train matrices in D and E. More detailed explanation in the paragraph below.

profile of all spike trains (0.192 in this case) and is marked by a dashed horizontal line. The green lines and symbols on top of the raster plot mark temporal instants and intervals results of which are detailed in the subplots below. B-E| Matrices of pairwise instantaneous dissimilarity values for a single time instant (mark 1, subplot B), for two selective averages (over non-consecutive intervals in mark 2, subplot C, and over the whole interval in mark 3, subplot D) and for a triggered average (mark 4, subplot E). F-G| For the overall average (mark 3) we also show the matrices of overall pairwise instantaneous dissimilarity values for the 4 spike train groups (F) and the corresponding dendrogram (G). H-I| Dendrogram of spike train matrices in D and E. Note that in contrast to the overall average (mark 3, subplot H) the triggered average (mark 4, subplot I) captures the local similarity between 5 of the spike trains.

4.1 FULL MATRIX AND CROSS SECTIONS

The starting point is the most detailed representation in which one instantaneous value is obtained for each pair of spike trains (see Equation 27). This representation could be viewed as a movie of a symmetric pairwise dissimilarity matrix in which each frame corresponds to one time instant (an example can be found in the supplementary material of Kreuz et al., 2013). For a movie of finite length the time axis necessarily has to be sampled but in principle this most detailed representation consists of an infinite number of values. However, since all dissimilarity profiles are piecewise linear there is a lot of redundancy.

One step towards a more compact and memory-efficient representation is to store all pairwise dissimilarity profiles in a matrix of size 'number of ISIs in the pooled spike train' $\times$ 'number of spike train pairs' ($\times 2$ for the SPIKE-distance, see Section 5.1.1). From this two-dimensional matrix it is possible to extract both kinds of cross sections. By selecting a pair of spike trains, one obtains the bivariate dissimilarity profile $S(t)$ for this pair of spike trains. Selecting a time instant t_s (and using linear interpolation for time instants in between spikes) yields an instantaneous matrix of pairwise spike train

dissimilarities $S_{mn}(t_s)$ (see Figure 4.1B). This matrix can be used to divide the spike trains into instantaneous clusters, that is, groups of spike trains with low intra-group and high inter-group dissimilarity.

For SPIKE-synchronization it is possible to select pairwise dissimilarity profiles but instantaneous matrices are not defined.

4.2 SPATIAL AND TEMPORAL AVERAGING

Another way to reduce the information of the dissimilarity matrix is averaging. There are two possibilities that commute: the spatial average over spike train pairs and the temporal average. Since the spatial average over spike train pairs can be done locally it yields a dissimilarity profile for the whole population. Examples for averages over four different spike train groups as well as over all spike trains are shown in the lower subplot of Figure 4.1A. Temporal averaging over certain intervals on the other hand leads to a bivariate distance matrix (see Figure 4.1C and D for examples of non-continuous and continuous intervals). In real data, these temporal intervals could be chosen to correspond to different external conditions such as normal vs. pathological, asleep vs. awake, target vs. non-target stimulus, or presence/absence of a certain channel blocker.

A combination of temporal and spatial averaging can be seen in Figure 4.1F. This dissimilarity matrix is obtained from the overall temporal average shown in Figure 4.1D by (spatially) averaging over the 16 submatrices and thus depicts the pairwise spike train group dissimilarity (4×4 instead of 20×20). Figure 4.1G shows the respective dendrogram. In applications to real data, these groups could be different neuronal populations or responses to different stimuli, depending on whether the spike trains were recorded simultaneously or successively. Finally, successive application of spatial average over all spike train pairs and temporal average over the whole interval results in just one distance value that describes the overall level of dissimilarity for the entire dataset. In Figure 4.1A this value is stated in the upper right of the lower subplot.

Both spatial and temporal averaging are well-defined for SPIKE-synchronization, and so is the overall value.

4.3 TRIGGERED AVERAGING

The fact that there are no limits to the temporal resolution allows further analyses such as internally or externally triggered temporal averaging. Here, the matrices are averaged over certain trigger time instants only. The idea is to check whether this triggered temporal average is significantly different from the global average since this would indicate that something peculiar is happening at these trigger instants.

The trigger times can either be obtained from external influences (such as the occurrence of certain features in a stimulus) or from internal conditions (such as the spike times of a certain spike train). External triggering is a standard tool to address questions of neural coding, for example, it can be used to evaluate the influence of localized stimulus features on the reliability of neurons under repeated stimulation. In multi-neuron data, internal triggering might help to uncover the connectivity in neural networks or to detect converging or diverging patterns of firing propagation.

An example is shown in Figure 4.1E. Here, neurons 2, 9, 14, and 19 follow the 7^{th} neuron during the 2^{nd} and the 4^{th} 500-ms subinterval. This can be revealed by triggering on the spike times of neuron 7 during these subintervals. The difference between the dissimilarity matrix of the full interval in Figure 4.1D and the triggered dissimilarity matrix of Figure 4.1E can be best seen by comparing the respective dendrograms in Figure 4.1H and Figure 4.1I.

Since instantaneous values are not defined, triggered averaging does not make any sense for SPIKE-synchronization.

Part III

RESULTS

5

SPIKY

In recent years, neuronal recordings have become both massively parallel (they provide data from large numbers of neurons) and more precise. However, if there would be a magic wand that could provide us with the ability to record simultaneously the whole brain, up to the neuronal level, at first, there would be no way to store these data, and later, there would be no available tool that we could use to analyze them, except maybe simple estimates of the mean firing rate over some part of the data. However, the disadvantage of having a measure that estimates the overall rate is that it is invariant to switching spikes between spike trains. It is thus unable to track any means of population coding, nor the effect of synchrony — since the effect should occur in all spike trains. Summing all together, it would be rather useful to identify similarity patterns with a very high temporal resolution and across different spatial scales. In recent years one important step towards this goal was made. Three time-resolved measures of spike train synchrony was proposed, the ISI-distance, the SPIKE-distance (Kreuz et al., 2013), and SPIKE synchronization (Kreuz et al., 2015).

With all three of these measures spike trains can be analyzed on different spatial and temporal scales, accordingly there are several levels of information extraction (Kreuz, 2012). In the most detailed representation one instantaneous value is obtained for each pair of spike train. The most condensed representation successive temporal and spatial averaging leads to one single distance value that describes the overall level of synchrony for a group of spike trains over a given time interval. In between these two extremes are spatial averages (dissimilarity profiles) and temporal averages (pairwise dissimilarity matrices). This variety of representations makes a straightforward implementation of the measures in one simple program/function unfeasible. Other

important goals are high computational speed, efficient memory management, and applicability to the extensive experimental datasets that have nowadays become increasingly common. What is needed is an intuitive and interactive tool for analyzing spike train data which is able to overcome all of these challenges.

Here we address this need and present the Graphical User Interface (GUI) SPIKY (Bozanic et al., 2014; Kreuz et al., 2015). Given a set of real or simulated spike train data (importable from many different formats), SPIKY calculates the time-resolved measures of choice and allows the user to switch between many different visualizations such as measure profiles, pairwise dissimilarity matrices, or hierarchical cluster trees. SPIKY also includes the possibility of generating movies which are very useful in order to track the varying patterns of (dis)similarity. SPIKY has been optimized with respect to both computation speed (by using C-based Matlab Executable (MEX)-files) and memory demand (by taking advantage of the piecewise linear nature of the dissimilarity profiles).

SPIKY also consists of a spike train generator which facilitates micro and macro data handling (so both single spikes and group of spikes (even from different spike trains) and an event detector which makes it capable of analyzing continuous data. One of the time-consuming parts of SPIKY that could not be optimized are the Matlab internal-function for plotting. Although data handling and Matlab plot function provide classy visualization for the SPIKY GUI, the analysis of large numbers of datasets and the estimation of significance levels can be done without this. The SPIKY package includes an additional complementary program, SPIKY_loop, aimed at these types of analysis. It gives the same results without losing extra time or memory for the steps that would be necessary for any user-friendly interface. Finally, another complementary program, SPIKY_loop_surro, is designed to evaluate the statistical significance of the results obtained for the original dataset by comparing them against the results of spike train surrogates generated from that dataset.

5.1 MEASURES AND IMPLEMENTATION

SPIKY implements four time-resolved measures, one is multivariate and three are bivariate (see Chapter 3). The multivariate measure, included for comparison, is the standard PSTH (see Section 3.1.1) which measures the overall firing rate. In this section we give an overview over the three bivariate measures, for more detailed illustrations please refer to Chapter 3. The ISI-distance (see Section 3.3.1.1, Kreuz et al., 2007), the SPIKE-distance (see Section 3.3.1.2, Kreuz et al., 2013), and the newly proposed SPIKE-Synchronization (see Section 3.3.2, Mulansky et al., 2015) based on event synchronization (Quian Quiroga et al., 2002) share several properties, however, there are also a few conceptual differences between the ISI- and the SPIKE-distance on the one hand, and SPIKE-Synchronization on the other hand.

All three measures rely on instantaneous values which are normalized between zero and one. The same holds true for the respective temporal averages, the distance values D_I and D_S and the SPIKE-Synchronization S_C. However, while the two distances are measures of dissimilarity which yield the value zero for identical spike trains, SPIKE-Synchronization is a measure of similarity with high values denoting similar spike trains.

While all three measures are time-resolved, the ISI-distance and the SPIKE-distance even have a continuous domain since there exists a unique definition of an instantaneous value ($I(t)$ and $S(t)$, respectively) for every single time instant. The resulting dissimilarity profiles are either piecewise constant (ISI-distance) or piecewise linear (SPIKE-distance). SPIKE-Synchronization is time-resolved as well, however, its domain is discrete since instantaneous values $C(t_k)$ are only defined at the times of the spikes. Incidentally, the same distinction holds true regarding the range of values that can be obtained for the measures: it is continuous for the two distances and discrete for SPIKE-Synchronization.

All three measures can also be applied to more than two spike trains (spike train number $N > 2$). For the ISI- and the SPIKE-distance this extension is simply the

average over all bivariate distances. Extending SPIKE-Synchronization is even more straightforward. Essentially, the same spike-based definition holds for both the bivariate and the multivariate case.

5.1.1 *Comparison with other implementations*

The very first all-in-one implementation of the ISI- and the SPIKE-distance[1] was published online at the time of the publication of Kreuz et al., 2013. In between this first and our current release, several other source codes written in various languages and for different platforms have been made available. The most prominent examples are the Python-Implementation of the SPIKE-distance courtesy of Jeremy Fix and available on his homepage[2], and the C++-Implementation of both ISI- and SPIKE-distance courtesy of Răzvan Florian and hosted on GitHub[3]. The SPIKE-distance was also implemented in the commercially distributed HRLAnalysis™ software suite Thibeault et al., 2014 designed for the analysis of large-scale spiking neural data. Note that all of these implementations are restricted to the dissimilarity profile and its temporal average (the overall dissimilarity). In contrast, SPIKY also allows the user to interactively access all the other different ways to extract information that were introduced in Chapter 4.

All of these codes for calculating the ISI- and the SPIKE-distance rely on equidistantly sampled dissimilarity profiles, and the same holds true for codes of event synchronization. Typically the precision is set to the sampling interval of the neuronal recording. Since the dissimilarity profiles have to be calculated and stored for each pair of spike trains, one obtains, for each measure, a matrix of order 'number of sampled time instants' × 'number of spike train pairs' (i.e., $\#(t_s) \times N(N-1)/2$). For small sampling intervals and large numbers of spike trains this leads to memory problems.

1 This subsection concerns only the ISI- and the SPIKE-distance. Since SPIKE-Synchronization is a new proposal, so there are no previous implementations.

2 http://jeremy.fix.free.fr/Softwares/spike.html

3 https://github.com/modulus-metric/spike-train-metrics

In SPIKY we use an optimized and more memory-efficient way of storing the results. We make use of the fact that the dissimilarity profile $I(t)$ of the ISI-distance is piecewise constant and the dissimilarity profile $S(t)$ of the SPIKE-distance is piecewise linear. Each constant/linear interval runs from one spike of the pooled spike train to the next. Thus, for each such interval (and for each pair of spike trains) we have to store only one value for the ISI-dissimilarity and two values for the SPIKE-dissimilarity, one at the beginning and one at the end of the interval (see Figure 5.1). The memory gain is proportional to the number of sample points per ISI in the pooled spike train and is typically much larger than one.

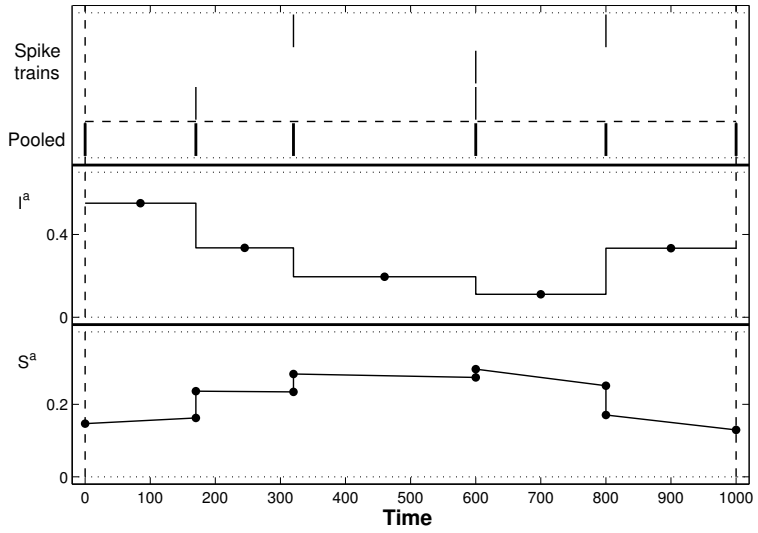


Figure 5.1: Illustration of the memory efficient storage of the dissimilarity profile for both the ISI- and the SPIKE-distance.

The dissimilarity profiles exhibit instantaneous jumps at the times of the spikes since this is where the lengths of the ISIs and the identity of the previous and the following spikes change abruptly or where a new coincidence is counted. For sampled dissimilarity profiles one has to 'cut the corners' of these instantaneous jumps. This leads to an estimation error which increases with the sampling interval. In contrast, within the new implementation each spike marks both the end of the previous and the beginning of the next interval and it becomes possible to store two dissimilarity

values for these points. This way the integration from one spike of the pooled spike train to the next can be performed over the full interval. Thus, besides being far more memory-efficient, the new implementation also computes the exact distance values without any spurious dependence on the sampling interval.

The third effect of the new implementation is a considerable speed-up. To show this (and to provide relative computational costs for the different measures) we here calculate the speed gain achieved by going from the old equidistantly sampled implementation to the new minimally sampled implementation of the ISI- and the SPIKE-distance.

As benchmark we use the comprehensive performance comparison carried out by Rusu and Florian, 2014. In this test the authors compared the performance of their newly proposed modulus-metric with the performance of previously proposed spike train distances including the ISI- and the SPIKE-distance. Like them we used two random spike trains with different numbers of spikes. However, since we were also interested in applications to larger datasets, we extended the maximum number of spikes from 500 to 10000 spikes. The firing rate is kept constant, hence the duration of the trial increases with the number of spikes. For a fair comparison we implemented all algorithms in C++ and ran them on an Intel i7-4700MQ CPU @ 2.4 GHz. All speed gains were averaged over 10,000 trials (Figure 5.2).

In a first step we replicated the results of Rusu and Florian, 2014 who had calculated the ISI- and SPIKE-distances using dissimilarity profiles that were equidistantly sampled with a fixed time step of dt = 1 ms. Rerunning the same algorithms on our computer, we could reproduce their results (for less than 500 spikes) within the same order of magnitude. Subsequently, we measured the running times using minimally sampled dissimilarity profiles and found that for both the ISI- and the SPIKE-distance the new implementation was considerably faster than the old implementation. As expected, the speed gain depends critically on the sampling rate used for the old implementation and is particularly large for densely sampled data (Figure 5.2).

Apart from minimal sampling, SPIKY includes another improvement which strongly increases the overall performance. The source codes published along with Kreuz et

al., 2013 were entirely written in Matlab. In contrast, the new SPIKY-implementation uses C-based MEX-files for the most time-consuming parts and this leads to another enormous performance boost (typically by a factor between 3 and 30).

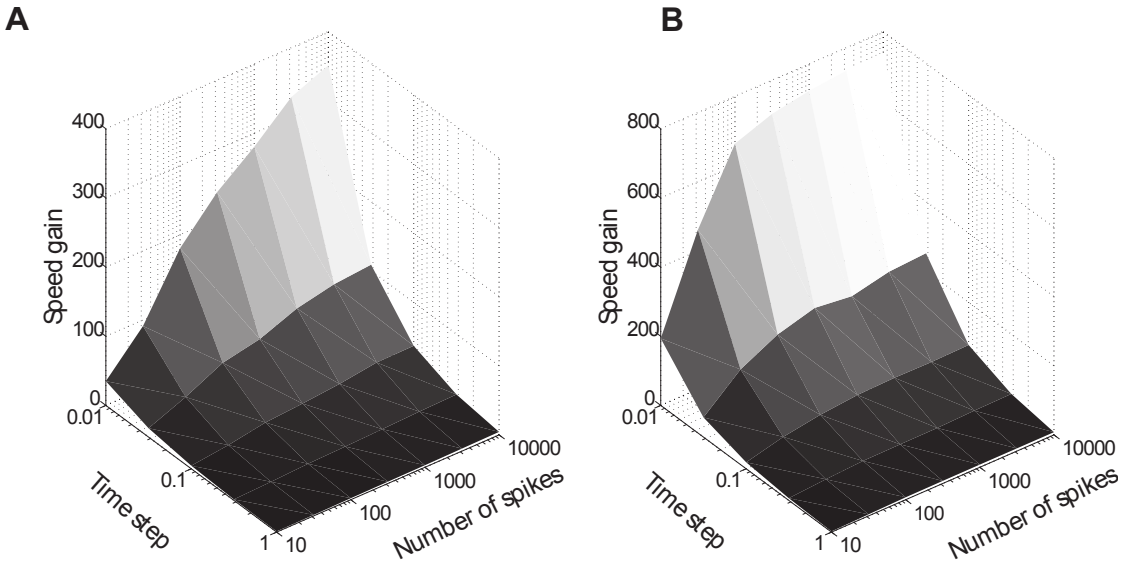


Figure 5.2: Speed gain achieved by the new (minimally sampled) implementation of A| ISI-distance and B| SPIKE-distance with respect to the old (equidistantly sampled) implementation in dependence on the number of spikes and on the time step used by the old implementation.

5.2 GRAPHICAL USER INTERFACE

SPIKY is a (GUI) for monitoring synchrony between artificially simulated or experimentally recorded neuronal spike trains or spike train turned continuous data. It contains PSTH and optimized implementations of the ISI-distance, the SPIKE-distance, and SPIKE-Synchronization (see Section 3.3). Moreover, two variants of SPIKE-distance[4], suitable for estimating synchrony online, are included (see Section 3.3.1.3 and Section 3.3.1.4). The source codes are written in Matlab (MathWorks Inc, Natick, MA,

4 The forward variant of the SPIKE-distance as well as a 'simulation' of the realtime SPIKE-distance.

USA) with the most time-consuming loops coded in MEX-files[5]. Consequently, SPIKY is not stand-alone but requires Matlab to run.

5.3 ACCESS TO SPIKY AND HOW TO GET STARTED

SPIKY is distributed under the BSD licence (Copyright (c) 2014, Thomas Kreuz, Nebojsa Bozanic. All rights reserved.). A zip-package containing all the necessary files can be accessed for free on the download page[6]. This package also contains a folder with documentation (such as a FAQ-file and an introduction to all individual elements of SPIKY and all individual files of the SPIKY-package). Further information and many demonstrations (both images and movies) can be found on the download page and on the SPIKY Facebook-page[7]. Both of these pages are used to announce updates and distribute the latest information about new features. They also provide an opportunity to give feedback and ask questions. Moreover, the Facebook-page includes various screen recordings with voice-over in which the user is guided step by step through some of the most important features of SPIKY. All of these movies can also be viewed on the SPIKY Youtube-channel[8].

After downloading SPIKY the user has to first extract the zip-package which leaves all files in one folder named 'SPIKY'. If the system has a suitable MEX-compiler installed, the MEX-files can be compiled from within this folder by running the m-file 'SPIKY_compile_MEX'. The program is started with the m-file 'SPIKY'.

When SPIKY is running, the user has the option to select to view short hints ('tooltips') when hovering above individual elements of the GUI. An overview of all the information contained in the hints can be found in the documentation file 'SPIKY-

5 In our case these are subroutines written in C/C++. However, as some users may not have access to a suitable C/C++ compiler, SPIKY contains the (slower) pure Matlab code as well.

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

7 https://www.facebook.com/SPIKYgui

8 https://www.youtube.com/user/SPIKYgui1

Elements'. Furthermore, at each step the suggested element for the next user action is highlighted by a bold font.

To get the user started quickly, SPIKY provides a few example datasets from previous publications. The most useful example is the 'Clustering' dataset which has already been used in Figure 5.4, then it will be shown in Figure 4.1, and as well it can be found in the supplementary movie of Kreuz et al., 2013. The best way to get acquainted with SPIKY is to advance from panel to panel by pressing the highlighted button. When the end is reached the user can reset and run the same example again while changing some parameters in order to see the consequences. Note that it is not necessary to set all the parameters each time when SPIKY is started. Rather, it is possible to use the file 'SPIKY_f_user_interface' to set and modify the spike train data as well as the parameters (again, the dataset 'Clustering' provides an example).

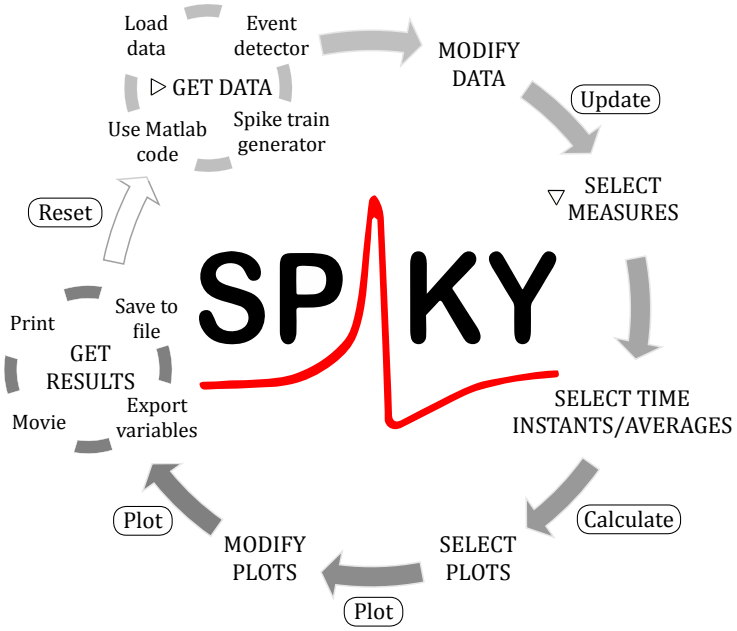


Figure 5.3: SPIKY-logo (center) and flowchart describing the workflow of SPIKY from the input of spike train data to the output of results. A typical SPIKY-session begins at the top with 'Get data', then goes clockwise and ends on the left with 'Get results'. [9]

5.4 STRUCTURE AND WORKFLOW OF SPIKY

Overall, SPIKY has a rather linear workflow, however it is much more interactive than previous implementations of the measures and there are many potential shortcuts and loops along the way. As shown in the SPIKY-flowchart in Figure 5.3, the general flow is clearly directed from the input of spike train data to the output of results. So the first step the user has to do is to give SPIKY spike train data (i.e. sequences of spike times) to work with. There are four possible ways to do this: one can make use of predefined examples, load data from a file or from the Matlab workspace, detect discrete events from continuous data, or employ the spike train generator (see Section 5.4.1 for more details on the SPIKY input).

Once the full dataset is available, modification is still possible. The user can restrict the analysis to a specific subset, e.g., select a smaller time window and/or a subset of spike trains. It is also possible to impose some external structure on the raster plot (spike trains vs time). For that, SPIKY allows the definition of two types of time markers (e.g. thick/thin markers for specific events such as seizure onset/offset in epilepsy, trigger onset/offset during stimulation etc.) and two types of spike train separators (e.g. a thick separator for neurons from the left vs. neurons from the right hemisphere and a thin separator for different regions within the two halves). The user can also define spike train groups. Depending on the setup these could be spike trains recorded in different brain regions or upon presentation of different stimuli. Figure 5.4 shows an example of a raster plot with annotations marking all these different elements.

After updating all of these data parameters, the next step is to select the measures to be calculated. Options include the PSTH as well as all measures described in detail in Chapter 3. In the same step the user can select successive frames for a temporal

9 For clarity we omitted two possible actions which can be performed from multiple positions on the workflow circle: From any point it is possible to reset to the very beginning ('reset' leads to the symbol ▷), and from any later point it is possible to jump back before the measure calculation ('reset with the same data' leads to the symbol ▽)

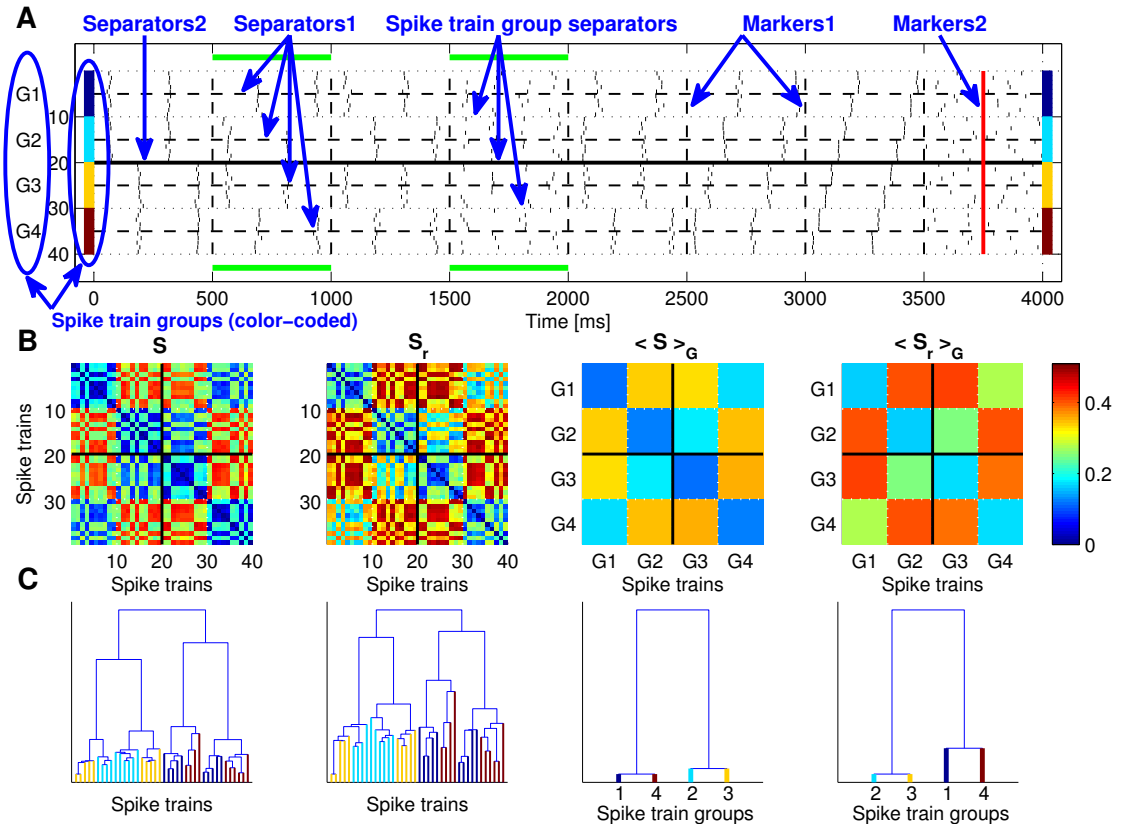


Figure 5.4: Annotated screenshot from a movie. A | Artificially generated spike trains. B | Dissimilarity matrices obtained by averaging over two separate time intervals for both the regular and the real-time SPIKE-distance as well as their averages over subgroups of spike trains (denoted by $< r >_G$). C | Corresponding dendrograms.

analysis of spike train patterns. These can be individual time instants for cross sections, temporal intervals for selective averages and sequences of time instants for triggered averages (see Chapter 4).

Now the actual calculation of the measures takes place. For reasonably sized datasets this should take at most a few seconds. Very large datasets (typically datasets containing hundreds of spike trains and/or hundred thousands of spikes) are divided in smaller subintervals and the calculation will be performed in a loop which might take longer. It is from this point on that SPIKY becomes truly interactive. Now the user can

switch between different representations of the results (such as dissimilarity profiles, dissimilarity matrices and dendrograms). Different matrices and dendrograms can be compared in the same figure or they can be viewed in sequence.

The presentation can be restricted to smaller time windows and/or subsets of spike trains, and temporal and spatial averaging (for example moving average, and average over spike train groups) can be performed. Spike trains or spike train groups can also be sorted according to the number of spikes (either within the whole spike train or within a certain interval) or according to the spike latency with respect to a specific time instant. The order can even depend on the result of the clustering analysis, i.e., spike trains belonging to the same cluster will appear next to each other.

Moreover, it is possible to add further figure elements such as spike number histograms, overall averages, or dissimilarity profiles for individual spike train groups. At this stage the user can also retrospectively change the appearance of all the individual elements of the figure (see Section 5.4.2 for more details). Finally, SPIKY allows to extract both data and results to the Matlab workspace for further analysis, and it is also possible to save individual figures as postscript-file or a sequence of figures as an 'avi'-movie (see Section 5.4.3 for more details on the SPIKY output).

5.4.1 *Input*

There are four different ways to input spike train data into SPIKY.

The first option is to select one of the predefined examples which are generated using Matlab-code. Initially these are the examples used in Kreuz et al., 2013 but one can also define new examples.

The second option is to load spike train data either from the Matlab workspace or from a file. Two different file formats are accepted, '.mat' and '.txt' (ASCII) files. For the mat-files SPIKY currently allows three different kinds of input formats (further formats can be added on demand):

- cell arrays (ca) with just the spike times. This is the preferred format used by SPIKY since it is most memory efficient. The two other formats will internally be converted into this format;
- rectangular matrices with each row being a spike train and zero padding (zp) in case of non-identical spike numbers ;
- matrices representing time bins where each zero/one (0/1) indicates the absence/presence of a spike.

In case of a mat-file or of the workspace, SPIKY looks for a variable called 'spikes'; if it cannot find it, the user has the option to select the variable name (or field name) which contains the spikes via an input mask which provides a hierarchical structure tree of all the variables and fields contained in the mat-file or in the workspace.

In the text format spike times should be written as a matrix with each row being one spike train. The SPIKY-package contains one example file for all four formats ('testdata_ca.mat', 'testdata_zp.mat', 'testdata_01.mat' and 'testdata.txt').

The third option is to use the *event detector* in order to detect discrete events in continuous data. There are many different possibilities of defining an event. A variety of standard events (such as local maxima and minima and threshold crossings) with a number of parameters are already included.

The fourth option is to create new spike train data via the *spike train generator*. After setting some defining variables (number of spike trains, start and end time, sampling interval) the user can build spike trains from predefined spike train patterns (such as periodic, splay, uniform or Poisson) and/or by manually adding, shifting and deleting individual spikes or groups of spikes.

5.4.2 *Figure-Layout*

SPIKY is designed so that it can directly generate figures suitable for publication. The user is given control over the appearance of every individual element (e.g. fonts, lines

etc.) in each type of figure. There are two ways to determine essential properties such as color, font size or line width. It is possible to change elements in the active figure while the program is already running. Context menus let the user edit the properties of individual elements or of all elements of a certain type. Conveniently, one can also use the file 'SPIKY_f_user_interface' to define the standard values for all the parameters that describe the principal layout of the figure.

If a figure contains more than one subplot (besides the subplot containing the spike rasterplot and/or the dissimilarity profiles, these are typically subplots with dissimilarity matrices and dendrograms), it is possible to change their position and size. One can edit all position variables together or change the x-position, the y-position, the width and the height individually. In case there are several dissimilarity matrices/dendrograms this can be done either for an individual matrix/dendrogram or for all of them at once.

5.4.3 *Output*

From within SPIKY it is possible to extract the spike trains and the results of the analysis (measure profiles, matrices, dendrograms) to the Matlab workspace for further processing. Results will be stored in variables such as 'SPIKY_spikes', 'SPIKY_profile_X_1', 'SPIKY_profile_Y_1', 'SPIKY_profile_name_1', 'SPIKY_matrix_1' and 'SPIKY_matrix_name_1'. In addition, the results obtained during an analysis will automatically be stored in the output structure 'SPIKY_results' which will have one field for each measure selected. Depending on the parameter selection within SPIKY, for each measure the structure can contain the following subfields that correspond to the different representations identified in Chapter 4:

- SPIKY_results.<Measure>.name: Name of selected measure

- SPIKY_results.<Measure>.distance: Level of dissimilarity over all spike trains and the whole interval. This is just one value, obtained by averaging over both spike trains and time
- SPIKY_results.<Measure>.matrix: Pairwise distance matrices, obtained by averaging over time
- SPIKY_results.<Measure>.time: Time-values of overall dissimilarity profile
- SPIKY_results.<Measure>.profile: Overall dissimilarity profile obtained by averaging over spike train pairs

Note that the dissimilarity profiles are not equidistantly sampled. Rather they are stored as memory-efficiently as possible which means just one value for each interval of the pooled spike train for the ISI- and two values for the SPIKE-distance. Since this format can be more difficult to process, SPIKY includes three functions: 'SPIKY_f_selective_averaging' for computing the selective average over time intervals, 'SPIKY_f_triggered_averaging' for calculating the triggered average over time instants, and 'SPIKY_f_average_pi' for averaging over many dissimilarity profiles. Furthermore, for the ISI-distance the function 'SPIKY_f_pico' can be used to obtain the average value as well as the x- and y-vectors for plotting.

Besides the standard way to work with Matlab-figures SPIKY also offers the opportunity to save each figure as a postscript-file. Finally, it is possible to save a sequence of figures as an 'avi'-movie.

5.5 GUI VS. LOOP

SPIKY was mainly designed to facilitate the detailed analysis of one dataset. It enables the user to switch between different representations (see Chapter 4) and to zoom in on both spatial and temporal features of interest. However, SPIKY is less convenient for the analysis of many different datasets when e.g. the statistics of a certain quantity

such as an average over a specific time interval should be evaluated over all available datasets (e.g. over all trials of a stimulus setup or for recordings of all subjects etc.) in some kind of loop. For these purposes the SPIKY-package contains a program called SPIKY_loop which is complementary to SPIKY. It is not a GUI but it should be simple enough (and plenty of examples are provided) to allow everyone to run the same kind of analysis for many different datasets and to evaluate and compare their 'SPIKY_results'. SPIKY_loop provides the full functionality of SPIKY and gives access to time instants, selective and triggered averages as well as averages over spike train groups.

So by combining these two programs it is possible to first use SPIKY for a rather exploratory but detailed analysis of a limited number of individual datasets and then use SPIKY_loop and its output structure 'SPIKY_loop_results' to verify whether any effect discovered on the example dataset is consistently present within all of the datasets.

Both SPIKY and SPIKY_loop require the storage of matrices of the order 'number of ISIs in the pooled spike train' $\times$ 'number of spike train pairs'. For very large datasets with many spike trains and/or spikes this can lead to memory problems. We addressed this issue by making the calculation sequential, i.e., by cutting the recording interval into smaller segments, and performing the averaging over all pairs of spike trains for each segment separately. In the end the dissimilarity profiles for the different segments (already averaged over pairs of spike trains) are concatenated, and its temporal average yields the distance value for the whole recording interval. During this sequential calculation SPIKY uses a waitbar (which displays the % completed) to continuously inform the user about the progress. This way, by trading memory against speed - running more loops with smaller matrices takes longer - SPIKY is able to deal with datasets containing more than one hundred spike trains and overall more than one million spikes.

5.6 SPIKE TRAIN SURROGATES AND SIGNIFICANCE

A very important issue that has not yet been addressed is statistical significance. Given a certain value of the SPIKE-distance, how can one judge whether it reflects a significant decrease or increase in spike train synchrony and does not just lie within the range of values obtained for random fluctuations? One answer to this question is the use of spike train surrogates (Kass et al., 2005, Grüen, 2009, Louis et al. 2010). The idea is to compare the results for the original dataset with the results obtained for spike train surrogates generated from that dataset. If the value obtained for the original lies outside the range of values for the surrogates this value can be assumed to be significant to a level defined by the number of surrogates used (e.g. $\alpha = 0.05$ for 19 surrogates or $\alpha = 0.001$ for 999 surrogates).

The SPIKY-package contains a program 'Spiky_loop_surro' which was designed to evaluate significance. At the moment it includes four different types of spike train surrogates. They differ in the properties that are preserved and maintain either the individual spike numbers (obtained by shuffling the spikes), the individual ISI distribution (obtained by shuffling the ISIs), the pooled spike train (obtained by shuffling spikes among the spike trains) or the PSTH. In the last case the spike train surrogates are obtained by means of inverse transform sampling (Ross, 1997), i.e., by resampling from the PSTH using its Cumulative Distribution Function (CDF). Other ways to calculate rate functions (e.g. based on kernels with different bandwidths) will be added in future releases.

5.7 DISCUSSION

5.7.1 *Summary*

In this chapter we presented SPIKY, up to date the only GUI which facilitates the application of methods of spike train synchrony to both simulated and real datasets.

Apart from the standard PSTH, SPIKY contains three parameter-free and time-resolved measures (see Section 3.3). These measures are complementary to each other since each one addresses a different specific aspect of spike train similarity. While the ISI-distance quantifies local dissimilarities based on covariances of the neurons' firing rate profiles, both the SPIKE-distance and SPIKE-Synchronization capture the relative timing of local spikes. However, whereas the SPIKE-distance weights and normalizes the differences between nearest neighbor spikes, SPIKE-Synchronization acts as a binary coincidence detector, i.e. there is a cutoff at the (adaptive) time lag relative to which two neighboring spikes are either considered coincident or not and all detailed information both within or outside this coincidence window is discarded.

All of these measures yield instantaneous values for each pair of spike trains, and thus there are many different possible representations of the results (see Chapter 4). Often the most informative representation might depend on the amount and type of spike train data and SPIKY can be used to reveal it via an exploratory and interactive analysis. SPIKY also allows to alter a given dataset before or after the actual analysis, e.g., to interactively select subintervals or spike train subsets, to define time markers and spike train separators, and to divide the dataset into different spike train groups.

In addition to the main GUI designed for the detailed analysis of one dataset, the SPIKY-package also includes two complementary programs. While SPIKY_loop aims at the grand average analysis of large numbers of datasets, SPIKY_loop_surro allows the estimation of significance levels. In all of these programs we use MEX-files, i.e. C-based Matlab executables for the more time-consuming parts and we exploit the piecewise linearity of the dissimilarity profiles, thereby guaranteeing high computation speed and memory efficiency.

5.7.2 *Other spike train analysis packages*

For completeness, here we provide a short description of software packages for spike train analysis that while not providing time-scale independent measures, implement some of the other methods of interest.

CHRONUX A Matlab toolbox (Bokil et al., 2010) which performs neuronal data analysis. It can be used for preprocessing as well as for filtering data. Cross correlation analysis (see Section 3.1.2) of spike trains is included as well.

SIGTOOL A Matlab package (Goldberg et al., 2009) that performs time frequency analysis of neuronal data and includes cross correlation (see Section 3.1.2).

SPIKE TRAIN ANALYSIS TOOLKIT A Matlab package (Lidierth, 2009) that provides information analysis of spike train data and includes the Victor-Purpura distance (see Section 3.2.1).

For more comprehensive information on software for spike train analysis please refer to Ince, 2013.

5.7.3 *Outlook*

Note that all three of these time-resolved measures of spike train synchrony used in SPIKY are based on some explicit notion of localness. Therefore, they all can be fooled by time lags, independent of the cause of these time lags. Thus, in a preprocessing step before the actual synchrony analysis any such time lag should be removed by suitably shifting the spike trains. This preprocessing step will now be addressed in Chapter 6.

6

LATENCY CORRECTION

Latency refers to a constant global shift of spike times between different spike trains. It is caused by various biophysical limitations and largely contributes to a spuriously low synchrony, as well as a false (de)correlation in the analysis of real data. Unlike standard methods which are time-scale dependent, we here propose the SPIKE-distance as a time-scale independent and parameter-free measure to estimate the latency in simulated and real data. Here we simulate a realistic setup with spike trains contaminated by a controlled level of noise.

6.1 INTRODUCTION

Latency can be characterized as a time lag between the presented stimulus and a recorded response. The latency is present in trial-to-trial single neuron responses as well as in simultaneously recorded populations of neurons (Nawrot et al., 2003). This is depicted in Figure 6.1 where an example of latency correction in real data is shown. Strong variations of these latencies can significantly change the spike train dissimilarity measures, since they are sensitive to exact spike timing. In principle, any kind of spike train synchrony analysis should only be applied to data that have already been latency-corrected. Establishing this correction as a mandatory preprocessing step would strongly improve the overall estimation of synchrony.

Note that latency is substantially different from neural noise. Noise in neural systems can be separated in several categories based on their sources. Most prominent are the following three: 1| Response of the single neuronal cell with its intrinsic variability. 2| Finite size effect of synapses' vesicles that makes them unreliable. 3| Neurons

are communicating between themselves, which means that variations is propagating which leads to the probably most influential type of noise (Lindner et al., 2004).

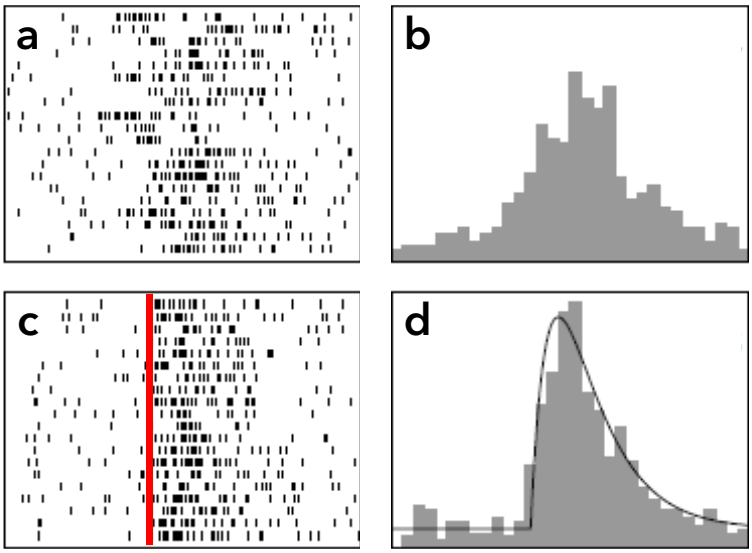


Figure 6.1: Example of trial-to-trial response latency and its elimination. a | Rasterplot before the correction. b | PSTH before the correction. c | Rasterplot after the correction. d | PSTH after the correction. Adapted from Figure 1, Nawrot et al., 2003

.

The standard methods for the estimation of the latency require one or more parameters which are typically estimated via some (un)supervised learning algorithm. Finding the optimal parameter(s) can be a very time-consuming task. The methods most employed for latency estimation are the Cross Correlation-based methods (see Section 3.1.2) applied to the estimated firing rates (see Section 3.1.1) (Nawrot et al., 2003; Bollimunta et al., 2007). Another approach is based on spike train distance such as the well known Victor-Purpura distance (see Section 3.2.1). Both methods require one preprocessing step. While for the Pearson Correlation coefficient one has to estimate the firing rate profile mostly via some kind of binning having a binsize as parameter, the Victor-Purpura distance relies on the cost parameter q. In this chapter, the SPIKE-distance is proposed as a latency estimator in a controlled setup which is used

to compare the performance of the SPIKE-distance with the one of the two standard methods.

6.2 EFFECTS OF JITTER AND LATENCY

Although the sources of noise in neuronal responses are diverse, ranging from axon jitter to vesicle release properties, they all have the effect of introducing uncertainty into any signal sent by the 'transmitting' neuron (Eliasmith and Anderson, 2004). A considerable amount of trial-by-trial variability is exhibited by both sensory neurons (Benda et al., 2007) and neurons in the cortex. This variability leads to spike time jitter, which also causes a variable shape of the neural activation in response to a stimulus (Vogels et al., 1989; Shadlen and Newsome, 1998). However, the most impactful form of variability concerns the temporal relation of the neural response with the time frame of the stimulus, i.e., the onset of the response to repeated presentations of the same stimulus (Figure 6.2). This latter type is more prominent in prefrontal, parietal, and motor areas (Schreiber et al., 2004). Although in sensory areas, the variability of the neural response is still significant, it is minor compared to other brain areas, since the responses of sensory neurons are often strongly locked to the stimulus.

Methods that are based on averaging across trials, (such as PSTH, see Section 3.1.1) can be adversely affected by such trial-to-trial variability (Figure 6.1). For example, as an external experimental parameter is varied, peaks of the PSTH become flattened and they may not any more be a good estimate, since the actual maximum firing time varies from trial to trial.

6.3 METHODS

The majority of methods published to date are based on parametric models and rely on the estimation of one or more parameters. Most of them, if not all, after starting

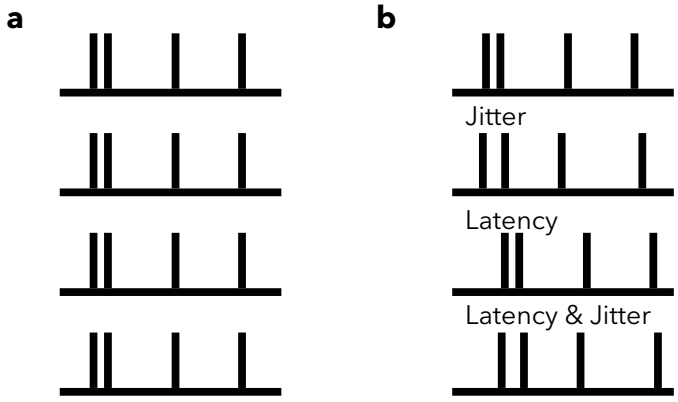


Figure 6.2: Reliability vs Latency vs Jitter. a | Spike trains are perfectly aligned. b | When spikes are displaced the neuron becomes imprecise. If the distribution of the displacement is close to a Gaussian with rather small standard deviation, one speaks of jitter. However, if the distribution is a delta pulse, that means that all spikes are shifted. This is called an onset latency. Adapted from Figure 1, Tiesinga et al., 2008.

with just spike trains, and use dynamic rate profiles for each trial. These rate profiles are obtained by convolving the spike trains with a kernel. Typically, a Bayesian interference framework is used to optimize the internal kernel parameters and this can significantly increase the computational efficiency (Nawrot et al., 2003). These rate profiles lead to estimates of the absolute and relative latencies. The absolute latency is tied to an optimal alignment (usually with the respect to some external trigger event) for trials, while relative latency estimation tend to yield pairwise correlations of continues firing rate profiles[1].

As stated, strictly speaking firing rates do not characterize synchrony, in contrast to spike train distances. Therefore, it make sense to use spike train distances for latency correction. SPIKE-distance is the perfect candidate since it is time-scale independent and sensitive to spike timing (see Section 3.3.1.2).

1 The absolute latency can also be quantified as trial-averaged relative latency.

In the following, we will investigate the performance of the SPIKE-distance in latency correction and compare it with both the Pearson Correlation coefficient and the Victor-Purpura distance.

6.3.1 *Setup*

The results of this chapter are based on a setup which consists of four steps: 1| Mimicking the recording of a neuron responses upon repeated presentation of the same stimulus. 2| Introducing two kinds of noise, spike time jitter and onset latency. 3| For each of these responses: Calculation of the Pearson Correlation coefficient of the estimated firing rates, the Victor-Purpura distance, and the SPIKE-distance. 4| Comparison of discrimination performance (here: Z-score) for all methods.

We mimicked the idea of reliability in the following way. The hypothesis is that once the same stimulus is presented repeatedly, we obtain the same or at least similar responses. We assume that the responses are reliable, just 'delayed'. However, due to the reasons detailed above they each might have a different latency. Additionally, we add Gaussian jitter noise with zero mean and standard deviation σ (see bottom right in Figure 6.3, Figure 6.4 and Figure 6.5). As we are trying to reconstruct the latency value τ, we have a completely arbitrary choice of setting this value and hence for simplicity we choose to set it to $\tau = 0$. Note that here we make standard assumptions about the kind of jitter present in biological systems. In particular, it is assumed that the noise is independent Gaussian with zero mean (Figure 6.3), and has the same variance for each neuron (Eliasmith and Anderson, 2004). There is some biological support for this assumption. For example, Rieke et al., 1997 note that in the case of frog calls 'the distribution of effective noise amplitudes is nearly Gaussian' (p. 184).

In the following benchmark we compare the performance of the Pearson Correlation coefficient and the Victor-Purpura distance with the performance of the SPIKE-distance. This analysis is based on two spike trains. First we generate a random Poisson spike train, and then we obtain the second spike train by distorting the first one

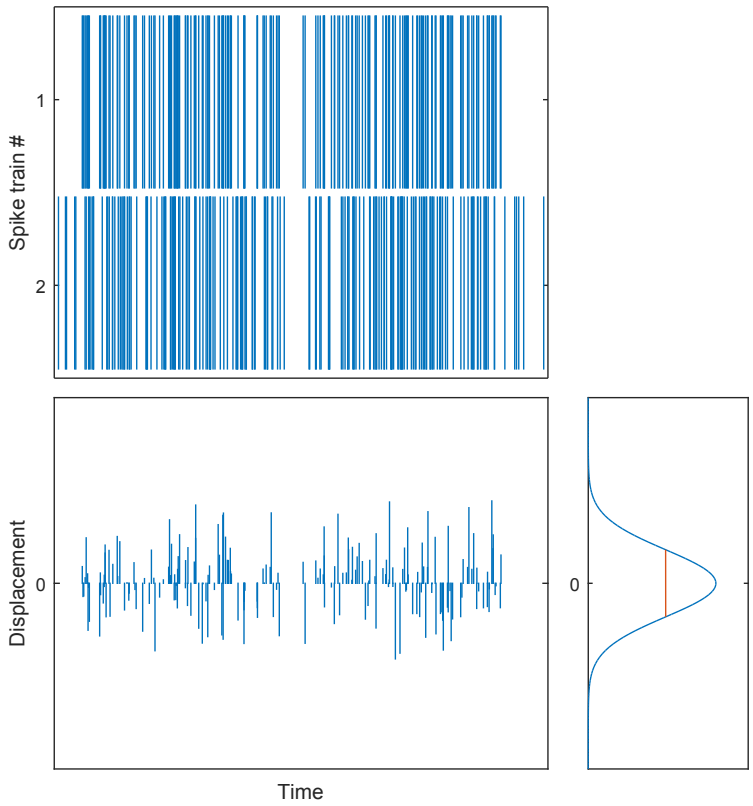


Figure 6.3: Top: Example with two spike trains. The second spike train is obtained from the first by adding Gaussian jitter of zero mean. Bottom left: Displacement (amplitude of jitter) for each spike in the second spike train. Bottom right: Distribution of displacements with mean value μ. The standard deviation is marked with a red line.

with controlled jitter. As explained above, this situation resembles for example a repeated recording of the same neuron with the same stimulus. We then try to identify the latency, which is zero in this case, and quantify how discriminative it is to the rest of the profile (for different values of latency). Except for the last example, the firing rate is kept constant. The jitter amplitude is measured relative to the average ISI. The computation was conducted in Matlab, with SPIKE-distance codes written in C/C++[2]

2 The new C-based MEX-files are available on our download page.

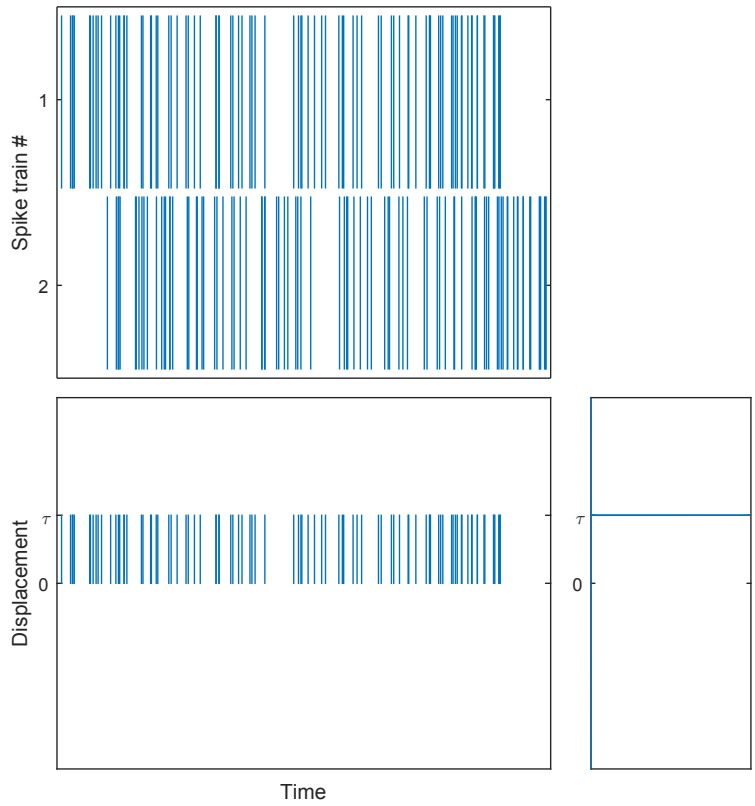


Figure 6.4: Top: Example with two spike trains. The second spike train is shifted with respect to the first by a latency of τ. Bottom left: Displacement (amplitude of jitter) for each spike in the second spike train. Bottom right: Distribution of displacements, i.e. Dirac delta in τ.

and compiled with MEX. The run was made on an Intel i7-4700MQ CPU @ 2.4 GHz. All results were averaged over 10,000 trials.

To avoid the problem of parameter optimization, see e.g. Nawrot et al., 2003 and Bollimunta et al., 2007, we simply covered a very wide range of parameters for both the Pearson Correlation coefficient and the Victor-Purpura distance and then for every measure and each level of jitter choose the optimal result.

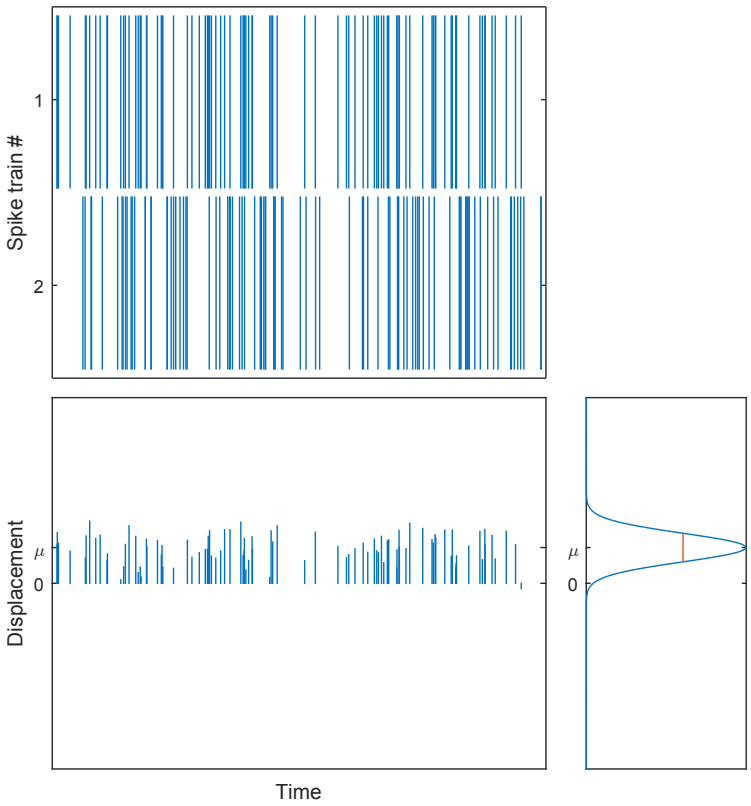
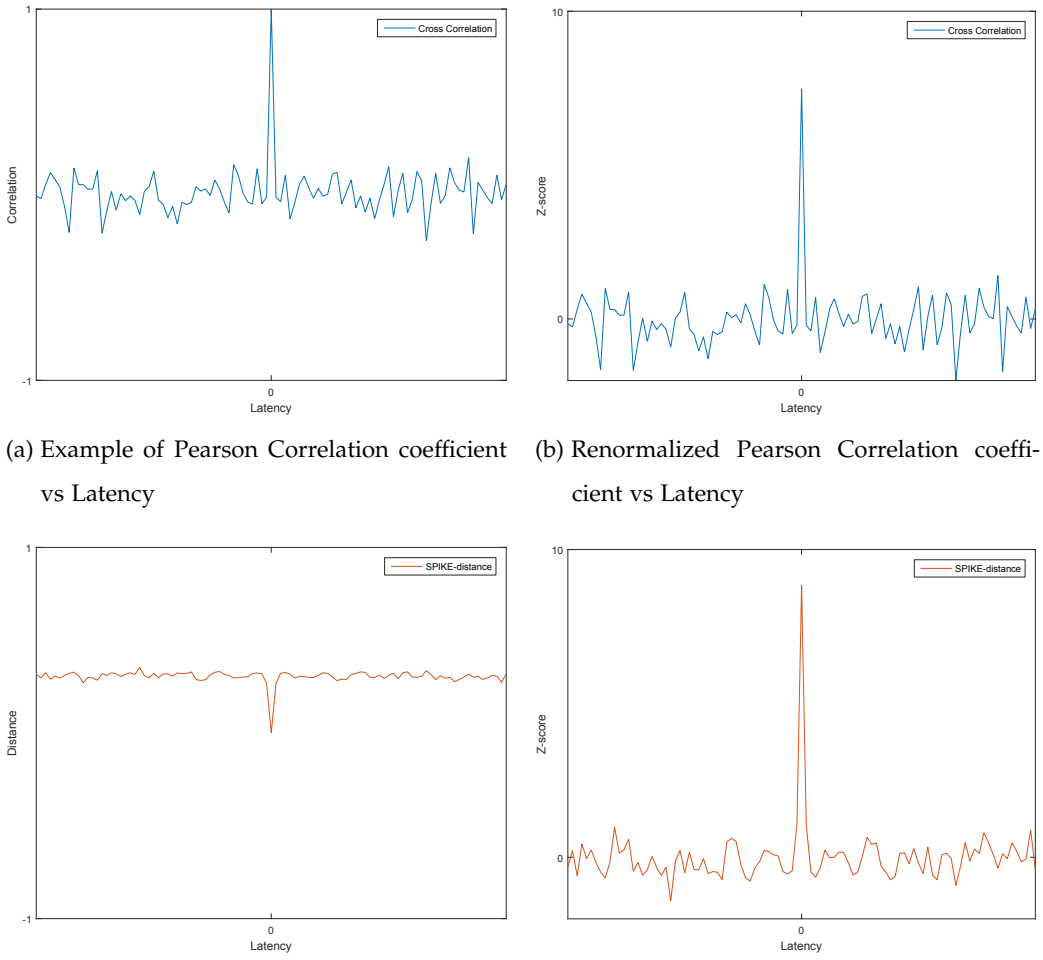


Figure 6.5: Top: Example of two spike trains. The second spike train is obtained from the first by adding Gaussian jitter of zero mean. The second spike train is also shifted by a latency of τ. Bottom left: Displacement (amplitude of jitter) for each spike of the second spike train. Bottom right: Distribution of displacements with mean value μ. The standard deviation is marked with a red line.

Finally, the profiles for all methods have a different range[3] and a different orientation[4]. In order to make all measures comparable, as a formal indicator of discrimination performance we use the normalized Z-score (see below). We want to see how discriminative the peak at time lag zero is compared to the values obtained for all other values of latency (Figure 6.6).

3 Range for: Pearson Correlation coefficient is $[-1, 1]$, SPIKE-distance $[0, 1)$, Victor-Purpura distance $[0, \infty)$.

4 Value for identical spike trains for the Pearson Correlation coefficient is 1 oriented upwards, for the Victor-Purpura distance 0 oriented downwards, and for the SPIKE-distance it is 0 oriented downwards.



(a) Example of Pearson Correlation coefficient vs Latency

(b) Renormalized Pearson Correlation coefficient vs Latency

(c) Example of SPIKE-distance vs latency

(d) Renormalized SPIKE-distance vs latency

Figure 6.6: Examples of Pearson Correlation coefficient and SPIKE-distance versus latency for two identical Poisson spike trains without any jitter. Note that when the latency is 0, the spike trains are identical, so the value of the Pearson Correlation coefficient is 1, and the value for the SPIKE-distance is 0. After the renormalization they both have values around 10. The values for positive and negative latencies fluctuate around the average value obtained for two Poisson spike trains with equal firing rates. This value is 0 for the Pearson Correlation coefficient, and 0.2945 for the SPIKE-distance (Kreuz et al., 2013). By definition of the z-score after the renormalization both profiles fluctuate around the new mean value of 0.

6.3.2 *Discrimination criterion*

The Z-score (or standard score) quantifies deviations from the mean in units of the standard deviation. Consequently, these Z-scores have a distribution with a mean of $\mu = 0$ and a standard deviation of $\sigma = 1$. The formula for calculating the Z-score is given below:

$$Z(x) = \frac{x - \mu(x)}{\sigma(x)} \tag{43}$$

6.4 RESULTS

In the following Section the performance of the SPIKE-distance in estimating the latency will be investigated, and compared against the two standard methods Pearson Correlation coefficient and Victor-Purpura distance. On the basis of simulated Poisson data it is clear that the mean discrimination parameter depends on the level of jitter and decreases for increasing jitter. This is seen in Figure 6.7 and Figure 6.8 where the amplitude of the Z-score at the latency value $\tau = 0$ is shown (compare Figure 6.6). As we increase jitter the peak at zero is becoming less and less discriminative. This means that the jitter of the spikes has become so strong that the structure is partially or completely lost. For noise levels around 2 all three methods are starting to slowly fail in recognizing any similarity.

The Pearson Correlation coefficient depends on the bin size parameter, which significantly affects the overall result. So for this measure in a first step the analysis is repeated over a whole range of bin sizes $[2^4, 2^{16}]$. Once the optimal bin size has been identified, the results obtained for this bin size are compared against the results obtained for the parameter-free SPIKE-distance.

On the other hand, the Victor-Purpura distance has a cost parameter, which also strongly influences the overall result. Here we vary this paramater on the range $[2^{-7}, 2^7]$.

Again the optimal value is taken and then it is compared against the SPIKE-distance (see Figure 6.7). As seen there, the SPIKE-distance shows a similar effectiveness as the two standard methods, without the drawback of requiring any parameter optimization.

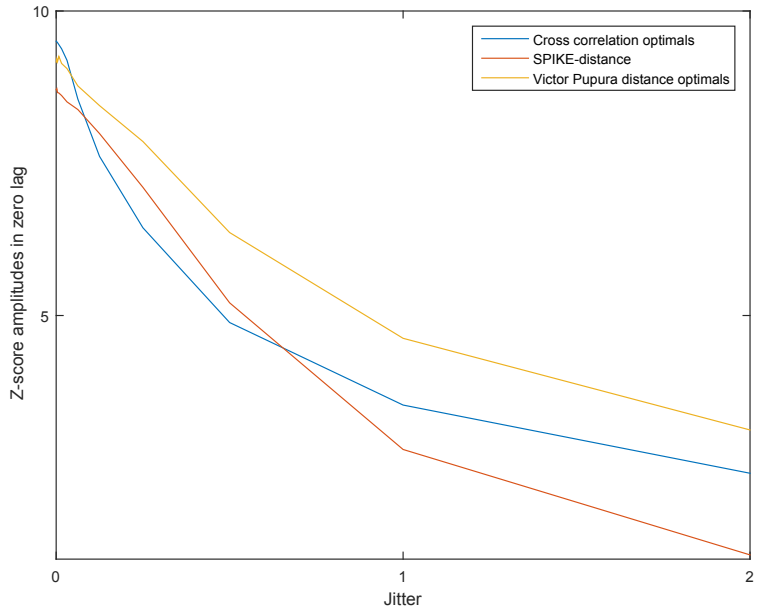


Figure 6.7: Comparison between SPIKE-distance and Victor-Purpura distance and Pearson Correlation coefficient both of which have been optimized for each level of jitter. Again the y-axis represents the amplitude of each method at zero time lag normalized by means of the z-score.

Although being parameter-free is a clear advantage for the SPIKE-distance, experimentalists often tend not to care about the complexity but the overall result. So that would mean that, qualitatively in Figure 6.7 the SPIKE-distance does not have any advantage.

On the other hand the Pearson Correlation coefficient and the Victor-Purpura distance and almost all other methods are time-scale dependent. We will now show that this can be a clear drawback when dealing with real data.

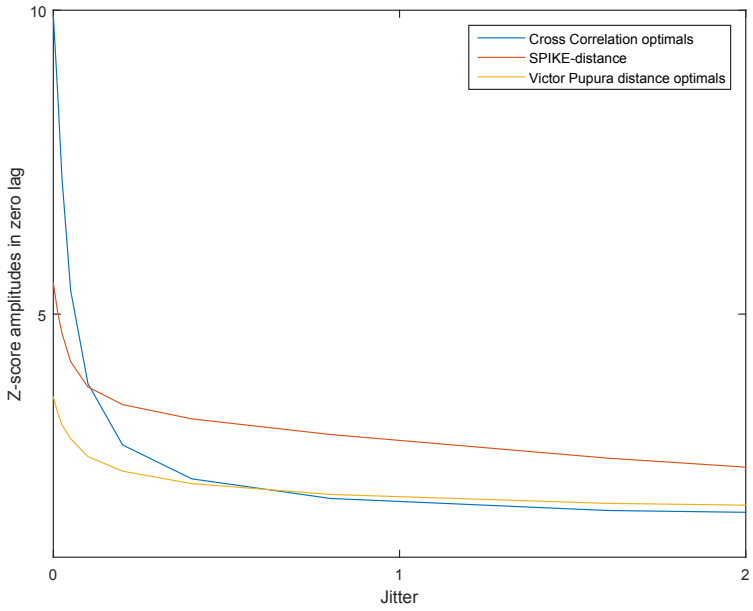


Figure 6.8: Latency estimation for spike trains with a dynamic firing rate.

In our setup above, we kept the firing rate constant. If we change this condition, we can start to see a vary large performance difference between SPIKE-distance and the other methods. We now divide the first spike train in two different parts with two distinctive firing rates. Specifically, we put the firing rate in the first half of the spike train to 0.01 and in the second to 100. Repeating the procedure as above, the SPIKE-distance now outperforms the Pearson Correlation coefficient (except for the very small values of jitter $[0 - 0.05]$), and the Victor-Purpura distance (for all levels of jitter) by a factor of two (see Figure 6.8). The reason why the SPIKE-distance method is superior in this case is the fact that these two standard methods are time-scale dependent. There is no single parameter value that is optimal for both time-scales. Therefore, in this simple example where we have spike trains with two different time-scales the advantage of using a time-scale independent method such as the SPIKE-distance becomes apparent (Bozanic et al.).

6.5 SUMMARY

In this Chapter we performed a detailed analysis on latency detection using three different methods. The Pearson Correlation coefficient, strongly affected by the bin size used for the estimation of the firing rate profile, and the Victor-Purpura distance, strongly affected by the cost parameter q defining the time-scale, were compared with the SPIKE-distance, a parameter free and time-scale independent alternative. In a first step we could show that for artificial data with constant firing rate all methods exhibit a very similar effectiveness (see Figure 6.7). However, for data with non-homogeneous firing rates, the SPIKE-distance significantly outperforms the other two methods (see Figure 6.8). A similar result is obtained when keeping the firing rate constant, but changing the jitter amplitude. Changing the jitter magnitude leads to the same problem as above, giving SPIKE-distance a clear advantage. This can be understood by noting that the optimal parameter value for the Pearson Correlation coefficient and the Victor-Purpura distance does not only depend on the firing rate, but also on the jitter amplitude. Therefore, the important ingredient is the ratio of the jitter amplitude to the average ISI (inverse firing rate).

Part IV

CONCLUSIONS

7

DISCUSSION

In order to understand the fundamental processes in the brain, one first needs to 'learn the language' of the brain. Since neurons communicate via sequences of stereotypical action potentials called spike trains, one very prominent way of doing this is spike train analysis. When dealing either with the activity of populations of neurons or with neuronal responses upon repeated stimulations in the context of neuronal coding, the essential notion of spike train synchrony appears naturally.

Thus, this thesis dealt with measures of spike train synchrony (or inversely spike train distances), i.e. estimators of the (dis)similarity between two or more spike trains. We focused mainly on three parameter-free and time-resolved measures (see Chapter 3) which are complementary to each other since each one addresses a different specific aspect of spike train similarity. While the ISI-distance is based on covariances of the neurons' firing rate, both the SPIKE-distance and SPIKE-synchronization capture the precise timing of local spikes. The SPIKE-distance weights and normalizes the differences between nearest neighbor spikes, while SPIKE-synchronization acts as a binary coincidence detector, i.e., there is a cutoff at the (adaptive) time lag relative to which two neighboring spikes are either considered coincident or not and all detailed information both within or outside this coincidence window is discarded.

For the first time we presented the mathematical properties of those three measures to provide all information necessary for their successful application to experimental or simulated data. Specifically, we were able to obtain the expectation values of the overall distances for random Poisson spike trains of arbitrary rates. This gives another, crucial point of reference for interpreting experimental and numerical results. For the ISI-distance as well as the SPIKE-Synchronization, we could calculate the expectation

values exactly. For the SPIKE-distance, the analytic result is intractable at this point and we present an empirical estimate that shows excellent agreement with numerical results. Note that for all three methods, the expectation value for two Poisson spike trains only depends on the ratio of the rates. This is an immediate consequence of the invariance of the measures under global rescaling of the time.

All of these measures yield instantaneous values for each pair of spike trains, and thus there are many different possible representations of the results (see Chapter 4). Often the most informative representation might depend on the amount and type of spike train data and this could be revealed best via an exploratory and interactive analysis.

Thus, in order to facilitate the application of time-scale independent and time-resolved methods of spike train synchrony to both simulated and real datasets, in the first main part of this thesis (Chapter 5) we provided a novel tool to a growing neuroscience community, the Matlab based graphical user interface SPIKY. We made an enormous effort to make SPIKY user-friendly for neuroscience experimentalists. For a given dataset SPIKY calculates the measures of choice and allows the user to switch between many different visualizations such as dissimilarity profiles, pairwise dissimilarity matrices, or hierarchical cluster trees (see Chapter 4). SPIKY allows to alter the dataset before or after the actual analysis, e.g., to interactively select subintervals or spike train subsets, to define time markers and spike train separators, and to divide the dataset into different spike train groups.

SPIKY also includes a spike train generator and an event detector which makes it capable of analyzing continuous data. Finally, the SPIKY-package includes complementary programs for the automatized analysis of a large number of datasets and for the evaluation of the statistical significance of the results. In all of these programs we use MEX-files, i.e. C-based Matlab executables for the more time-consuming parts and we exploit the piecewise linearity of the dissimilarity profiles, thereby guaranteeing high computation speed and memory efficiency. We invested a lot of effort on making

GUI user friendly. We have already received a lot of feedback on SPIKY, which means that it is already being used.

Since all three time-resolved measures of spike train synchrony are based on some explicit notion of localness and simultaneousness, all of them can be strongly affected by latency no matter whether this latency is caused by internal delay loops or by a common driver with different delays. In fact, most experimental results (e.g., in neuronal coding analysis) are seriously contaminated with different types of latencies, so any such latencies should be removed in a preprocessing step by suitably shifting the spike trains.

In the second main part of this thesis (Chapter 6), we performed a detailed analysis on latency detection using the SPIKE-distance, cross correlation and Victor Purpura distance. We demonstrated the advantages of the time-scale independent SPIKE-distance, which in spike trains with non-homogeneous firing rate clearly outperforms the other two measures. The reason is that in biologically inspired setups for both cross-correlation-like methods and Victor Purpura distance there might not be an optimal parameter that defines the time-scale.

8

OUTLOOK

The aim of all our research is to bring theory and data analysis closer to experimental data. The SPIKY and latency correction, and every correction and modification of any method, is done in order to work with real data. Here we discuss some of the potential directions for future real life applications.

One of the measure variants is the realtime SPIKE-distance. The present algorithm calculates its instantaneous dissimilarity values by making use of past information only but it does so in a speed-optimized and parallelized manner which would not be compatible with an actual realtime- implementation. However, such a realtime-implementation should actually be simpler and computationally (speed and memory) less problematic even for large numbers of neurons since the only information that would have to be stored at each time instant are the time stamps of the latest spikes of each spike train and their nearest neighbors in the other spike trains.

The graphical user interface SPIKY was primarily designed to analyze neuronal spike trains. But in principle it is also applicable to any other kind of discrete data which comes in the form of sequences of time stamps (such as times of bouncing basketballs or time-coded social interactions in a psychological test). Furthermore, in Kreuz et al., 2013 the SPIKE-distance was already applied not only to discrete data but also to an example of continuous data (EEG). To clear the way for such applications SPIKY includes an event detector which allows to bridge the gap between continuous data and our methods for discrete data.

We chose to write SPIKY in Matlab due to its popularity in the neuroscience community, ease of use, and the engaging visualization capabilities of its GUI design. Because of its high level parallelization, Matlab provides a powerful tool for processing vec-

torized data, but it also includes a well-developed MEX-interface for integrating C functions for performance enhancements. C functions do not only lead to an increase in performance, they can also easily be incorporated into other programming platforms. Indeed, we have already ported all four measures (the PSTH, the ISI-distance, the SPIKE-distance and SPIKE synchronization) as well as some of the additional SPIKY-functionality to Python as part of the PySpike[1] module, an open-source project hosted on Github. As in SPIKY, in PySpike the computation intensive functions are implemented as C backends.

Another potential direction would be to extend the methods from the quantification of synchrony within one population of two or more neurons to the quantification of synchrony between two neuronal populations. Similar extensions have been done for two other spike train distances (Aronov et al., 2003; Houghton and Sen, 2008), but both of these methods depend on not one but two parameters. So one particular challenge for a potential extension of our methods would be to make them parameter-free while maintaining their high temporal resolution.

There are still many open questions regarding neuronal synchrony. Among these questions are its role in dynamical diseases like epilepsy and its relevance for neural coding. While a thorough discussion of these issues is beyond the scope of this thesis, we argue that our time-resolved and time-scale independent measures will be a very useful tool for its investigation. If epileptic seizures and/or time-dependent stimulations lead to changes in spike train synchrony or spike train clustering, the methods presented here should be able to detect them.

1 https://github.com/mariomulansky/PySpike

BIBLIOGRAPHY

[1] D. Aronov, D. S. Reich, F. Mechler, and J. D. Victor. "Neural coding of spatial phase in V1 of the macaque monkey." In: *Journal of neurophysiology* 89.6 (2003), pp. 3304–3327 (cit. on p. 92).

[2] F. A. Azevedo, L. R. Carvalho, L. T. Grinberg, J. M. Farfel, R. E. Ferretti, R. E. Leite, R. Lent, S. Herculano-Houzel, et al. "Equal numbers of neuronal and non-neuronal cells make the human brain an isometrically scaled-up primate brain." In: *Journal of Comparative Neurology* 513.5 (2009), pp. 532–541 (cit. on p. 9).

[3] J. Benda, T. Gollisch, C. K. Machens, and A. V. Herz. "From response to stimulus: adaptive sampling in sensory physiology." In: *Current opinion in neurobiology* 17.4 (2007), pp. 430–436 (cit. on p. 73).

[4] H. Bokil, P. Andrews, J. E. Kulkarni, S. Mehta, and P. P. Mitra. "Chronux: a platform for analyzing neural signals." In: *Journal of neuroscience methods* 192.1 (2010), pp. 146–151 (cit. on p. 69).

[5] A. Bollimunta, K. H. Knuth, and M. Ding. "Trial-by-trial estimation of amplitude and latency variability in neuronal spike trains." In: *Journal of neuroscience methods* 160.1 (2007), pp. 163–170 (cit. on pp. 72, 77).

[6] M. R. Bower, M. Stead, F. B. Meyer, W. R. Marsh, and G. A. Worrell. "Spatiotemporal neuronal correlates of seizure generation in focal epilepsy." In: *Epilepsia* 53 (2012), p. 807 (cit. on p. 4).

[7] N. Bozanic, M. Mulansky, and T. Kreuz. "Latency correction using the time-scale independent SPIKE-distance" (cit. on pp. 7, 82).

[8] N. Bozanic, M. Mulansky, and T. Kreuz. "SPIKY." In: *Scholarpedia* 9.12 (2014), p. 32344 (cit. on pp. 6, 52).

[9] D. Chicharro, T. Kreuz, and R. G. Andrzejak. "What can spike train distances tell us about the neural code?" In: *Journal of neuroscience methods* 199.1 (2011), pp. 146–165 (cit. on pp. 20, 41).

[10] F. J. Damerau. "A technique for computer detection and correction of spelling errors." In: *Communications of the ACM* 7.3 (1964), pp. 171–176 (cit. on p. 20).

[11] M. Di Poppa and B. S. Gutkin. "Correlations in background activity control persistent state stability and allow execution of working memory tasks." In: *Front Comp Neurosci* 7 (2013), p. 139 (cit. on p. 5).

[12] C. Eliasmith and C. H. Anderson. *Neural engineering: Computation, representation, and dynamics in neurobiological systems*. MIT press, 2004 (cit. on pp. 9, 73, 75).

[13] W. Gerstner and W. M. Kistler. *Spiking neuron models*. Cambridge, UK: Cambridge Univ. Press, 2002 (cit. on p. 9).

[14] D. H. Goldberg, J. D. Victor, E. P. Gardner, and D. Gardner. "Spike train analysis toolkit: enabling wider application of information-theoretic techniques to neurophysiology." In: *Neuroinformatics* 7.3 (2009), pp. 165–178 (cit. on p. 69).

[15] S. Grüen. "Data-Driven Significance Estimation for Precise Spike Correlation." In: *J Neurophysiol* 101 (2009), pp. 1126–1140 (cit. on p. 67).

[16] A. Guyton and J. Hall. "Textbook of medical physiology." In: (2006) (cit. on p. 12).

[17] D Hebb. "The Organization of Behaviour: A Neuropsychological Theory." In: (1949) (cit. on p. 4).

[18] L. R. Hochberg, M. D. Serruya, G. M. Friehs, J. A. Mukand, M. Saleh, A. H. Caplan, A. Branner, D. Chen, R. D. Penn, and J. P. Donoghue. "Neuronal ensemble control of prosthetic devices by a human with tetraplegia." In: *Nature* 442 (2006), p. 164 (cit. on p. 23).

[19] C. Houghton and K. Sen. "A new multineuron spike train metric." In: *Neural Computation* 20 (2008), p. 1495 (cit. on p. 92).

[20] R. A. Ince. "Open-source software for studying neural codes." In: *Principles of Neural Coding eds. Quiroga R. Q. and Panzeri S.* (2013), pp. 597–604 (cit. on p. 69).

[21] E. M. Izhikevich. *Dynamical systems in neuroscience*. MIT press, 2007 (cit. on p. 4).

[22] R. Jolivet, R. Kobayashi, A. Rauch, R. Naud, S. Shinomoto, and W. Gerstner. "A benchmark test for a quantitative assessment of simple neuron models." In: *J Neurosci Methods* 169 (2008), pp. 417–424 (cit. on p. 17).

[23] E. R. Kandel, J. H. Schwartz, T. M. Jessell, et al. *Principles of neural science*. Vol. 4. McGraw-Hill New York, 2000 (cit. on p. 9).

[24] R. E. Kass, V. Ventura, and E. N. Brown. "Statistical issues in the Analysis of Neuronal Data." In: *J Neurophysiol* 94 (2005), pp. 8–25 (cit. on p. 67).

[25] T. Kreuz. "SPIKE-distance." In: *Scholarpedia* 7 (12) (2012), p. 30652 (cit. on pp. 6, 29, 51).

[26] T. Kreuz, J. S. Haas, A. Morelli, H. D. I. Abarbanel, and A. Politi. "Measuring spike train synchrony." In: *J Neurosci Methods* 165 (2007), p. 151 (cit. on pp. 5, 25, 35, 53).

[27] 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." In: *Phys D* 225 (2007), p. 29 (cit. on p. 21).

[28] T. Kreuz, D. Chicharro, R. G. Andrzejak, J. S. Haas, and H. D. I. Abarbanel. "Measuring multiple spike train synchrony." In: *J Neurosci Methods* 183 (2009), p. 287 (cit. on p. 25).

[29] T. Kreuz, D. Chicharro, M. Greschner, and R. G. Andrzejak. "Time-resolved and time-scale adaptive measures of spike train synchrony." In: *J Neurosci Methods* 195 (2011), p. 92 (cit. on pp. 29, 39, 41).

[30] T. Kreuz, D. Chicharro, C. Houghton, R. G. Andrzejak, and F. Mormann. "Monitoring spike train synchrony." In: *J Neurophysiology* 109 (2013), p. 1457 (cit. on pp. 5, 21, 23, 29, 32, 43, 45, 51, 53, 54, 56, 57, 59, 62, 79, 91).

[31] T. Kreuz, M. Mulansky, and N. Bozanic. "SPIKY: A graphical user interface for monitoring spike train synchrony." In: *Journal of neurophysiology* 113.9 (2015), pp. 3432–3445 (cit. on pp. 6, 21, 33, 34, 51, 52).

[32] A. Kumar, S. Rotter, and A. Aertsen. "Spiking activity propagation in neuronal networks: reconciling different perspectives on neural coding." In: *Nature Rev Neurosci* 11 (2010), pp. 615–627 (cit. on p. 3).

[33] V. I. Levenshtein. "Binary codes capable of correcting deletions, insertions, and reversals." In: *Soviet physics doklady*. Vol. 10. 8. 1966, pp. 707–710 (cit. on p. 20).

[34] M. S. Lewicki. "A review of methods for spike sorting: the detection and classification of neural action potentials." In: *Network: Computation in Neural Systems* 9.4 (1998), R53–R78 (cit. on p. 13).

[35] M. Lidierth. "sigTOOL: a MATLAB-based environment for sharing laboratory-developed software to analyze biological signals." In: *Journal of neuroscience methods* 178.1 (2009), pp. 188–196 (cit. on p. 69).

[36] B. Lindner, J. Garcıa-Ojalvo, A. Neiman, and L. Schimansky-Geier. "Effects of noise in excitable systems." In: *Physics Reports* 392.6 (2004), pp. 321–424 (cit. on p. 72).

[37] S. Louis, G. L. Gerstein, S. Grüen, and M. Diesmann. "Surrogate spike train generation through dithering in operational time." In: *Front Comp Neurosci* 4 (2010), p. 127 (cit. on p. 67).

[38] D. Lyttle and J.-M. Fellous. "A new similarity measure for spike trains: sensitivity to bursts and periods of inhibition." In: *Journal of neuroscience methods* 199.2 (2011), pp. 296–309 (cit. on p. 27).

[39] Z. Mainen and T. J. Sejnowski. "Reliability of spike timing in neocortical neurons." In: *Science* 268 (1995), p. 1503 (cit. on pp. 4, 17).

[40] J. Martínez and R. Q. Quiroga. "Spike Sorting." In: *Principles of Neural Coding eds. Quiroga R. Q. and Panzeri S.* (2013), pp. 61–74 (cit. on p. 13).

[41] P. Mitra and H. Bokil. *Observed brain dynamics*. Oxford University Press, 2007 (cit. on p. 18).

[42] F. Mormann, R. G. Andrzejak, C. E. Elger, and K. Lehnertz. "Seizure prediction: the long and winding road." In: *Brain* 130 (2007), p. 314 (cit. on p. 23).

[43] M. Mulansky, N. Bozanic, A. Sburlea, and T. Kreuz. "A guide to time-resolved and parameter-free measures of spike train synchrony." In: *International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*. 2015, pp. 1–8 (cit. on pp. 23, 24, 28, 33, 40, 53).

[44] M. Munz, D. Gobert, A. Schohl, J. Poquérusse, K. Podgorski, P. Spratt, and E. S. Ruthazer. "Rapid Hebbian axonal remodeling mediated by visual stimulation." In: *Science* 344.6186 (2014), pp. 904–909 (cit. on p. 4).

[45] M. P. Nawrot, A. Aertsen, and S. Rotter. "Elimination of response latency variability in neuronal spike trains." In: *Biological cybernetics* 88.5 (2003), pp. 321–334 (cit. on pp. 71, 72, 74, 77).

[46] S. Nirenberg and J. D. Victor. "Analyzing the activity of large populations of neurons: how tractable is the problem?" In: *Curr Opin Neurobiol* 17 (2007), p. 397 (cit. on p. 3).

[47] T. S. Otis and M. V. Sofroniew. "Glia get excited." In: *Nature neuroscience* 11.4 (2008), pp. 379–380 (cit. on p. 9).

[48] A. R. Paiva, I. Park, and J. C. Príncipe. "A comparison of binless spike train measures." In: *Neural Computing and Applications* 19.3 (2010), pp. 405–419 (cit. on p. 20).

[49] B. Pakkenberg, D. Pelvig, L. Marner, M. J. Bundgaard, H. J. G. Gundersen, J. R. Nyengaard, and L. Regeur. "Aging and the human neocortex." In: *Experimental gerontology* 38.1 (2003), pp. 95–99 (cit. on p. 9).

[50] A. Papoutsi, K. Sidiropoulou, V. Cutsuridis, and P. Poirazi. "Induction and modulation of persistent activity in a layer V PFC microcircuit model." In: *Front Neural Circuits* 7 (2013), p. 161 (cit. on p. 5).

[51] R. Quian Quiroga, T. Kreuz, and P. Grassberger. "Event Synchronization: A simple and fast method to measure synchronicity and time delay patterns." In: *Phys. Rev. E* 66 (2002), p. 041904 (cit. on pp. 4, 5, 35, 53).

[52] R. Quian Quiroga, A. Kraskov, T. Kreuz, and P. Grassberger. "Performance of different synchronization measures in real data: A case study on electroencephalographic signals." In: *Phys. Rev. E* 65 (2002), p. 041903 (cit. on p. 21).

[53] A. D. Reyes. "Synchrony-dependent propagation of firing rate in iteratively constructed networks in vitro." In: *Nature Neurosci* 6 (2003), p. 593 (cit. on p. 17).

[54] F. Rieke, D. Warland, R de Ruyter van Steveninck, and W. Bialek. "Spikes." In: *Cambridge, MA: Massachusetts Institute of Technology* 3 (1997) (cit. on p. 75).

[55] S. M. Ross. "Introduction to probability models." In: (1997) (cit. on p. 67).

[56] C. V. Rusu and R. V. Florian. "A new class of metrics for spike trains." In: *Neural Computation* 26.2 (2014), pp. 306–348 (cit. on p. 56).

[57] P. Sacré and R. Sepulchre. "Sensitivity Analysis of Oscillator Models in the Space of Phase-Response Curves: Oscillators As Open Systems." In: *Control Systems, IEEE* 34 (2014), pp. 50–74 (cit. on p. 5).

[58] 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." In: *IEEE Signal Processing* 25 (2008), p. 29 (cit. on p. 23).

[59] B. Schrauwen and J. Van Campenhout. "Linking non-binned spike train kernels to several existing spike train metrics." In: *Neurocomputing* 70.7 (2007), pp. 1247–1253 (cit. on p. 20).

[60] S. Schreiber, J. M. Fellous, J. H. Whitmer, P. H. E. Tiesinga, and T. J. Sejnowski. "A new correlation-based measure of spike timing reliability." In: *Neurocomputing* 52 (2003), p. 925 (cit. on p. 20).

[61] S. Schreiber, J.-M. Fellous, P. Tiesinga, and T. J. Sejnowski. "Influence of ionic conductances on spike timing reliability of cortical neurons for suprathreshold rhythmic inputs." In: *Journal of neurophysiology* 91.1 (2004), pp. 194–205 (cit. on p. 73).

[62] M. N. Shadlen and W. T. Newsome. "The variable discharge of cortical neurons: implications for connectivity, computation, and information coding." In: *The Journal of neuroscience* 18.10 (1998), pp. 3870–3896 (cit. on p. 73).

[63] J. Shlens, F. Rieke, and E. L. Chichilnisky. "Synchronized firing in the retina." In: *Curr Opin Neurobiol* 18 (2008), p. 396 (cit. on p. 3).

[64] C. M. Thibeault, M. J. O'Brien, and N. Srinivasa. "Analyzing large-scale spiking neural data with HRLAnalysis?" In: *Front. Neuroinform.* 8 (2014), p. 17 (cit. on p. 54).

[65] P. H. E. Tiesinga, J. M. Fellous, and T. J. Sejnowski. "Regulation of spike timing in visual cortical circuits." In: *Nature Reviews Neuroscience* 9 (2008), p. 97 (cit. on p. 3).

[66] P. Tiesinga, J.-M. Fellous, and T. J. Sejnowski. "Regulation of spike timing in visual cortical circuits." In: *Nature reviews neuroscience* 9.2 (2008), pp. 97–107 (cit. on p. 74).

[67] W. Truccolo, J. A. Donoghue, L. R. Hochberg, E. N. Eskandar, J. R. Madsen, W. S. Anderson, E. N. Brown, E. Halgren, and S. S. Cash. "Single-neuron dynamics in human focal epilepsy." In: *Nature Neurosci* 14 (2011), p. 635 (cit. on p. 4).

[68] J. D. Victor and K. P. Purpura. "Nature and precision of Temporal Coding in Visual Cortex: A Metric-Space Analysis." In: *J Neurophysiol* 76 (1996), p. 1310 (cit. on pp. 19–21, 41).

[69] J. D. Victor and K. P. Purpura. "Metric-space analysis of spike trains: theory, algorithms and application." In: *Network: Comput. Neural Syst* 8 (1997), p. 127 (cit. on pp. 19, 21).

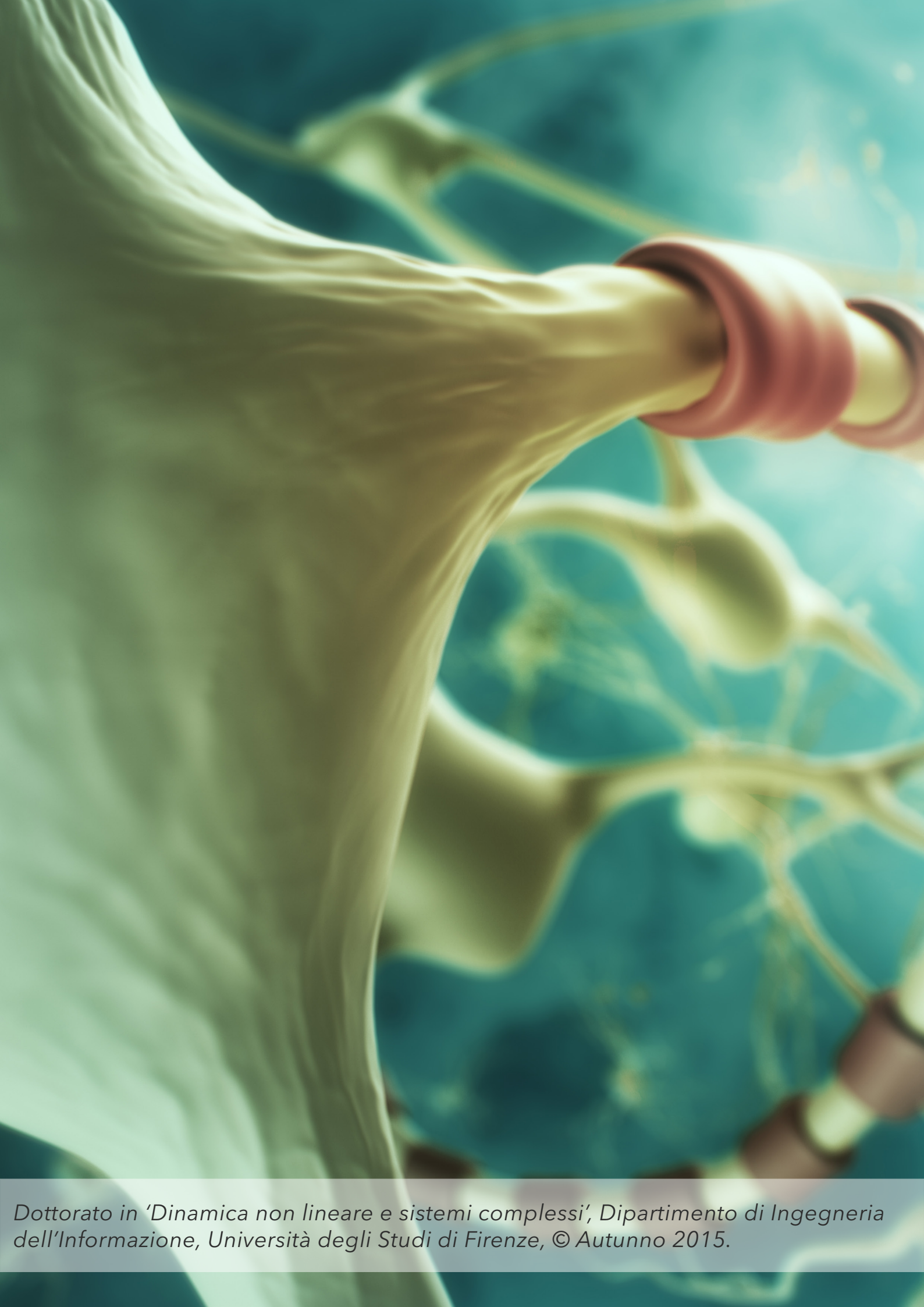
[70] J. D. Victor. "Spike train metrics." In: *Current opinion in neurobiology* 15.5 (2005), pp. 585–592 (cit. on pp. 4, 19).

[71] R. Vogels, W. Spileers, and G. A. Orban. "The response variability of striate cortical neurons in the behaving monkey." In: *Experimental brain research* 77.2 (1989), pp. 432–436 (cit. on p. 73).

[72] M. C. W. van Rossum. "A novel spike distance." In: *Neural Computation* 13 (2001), p. 751 (cit. on p. 20).

The PhD candidate's thesis was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. `classicthesis` is available for both LaTeX and LyX:

https://bitbucket.org/amiede/classicthesis/



Dottorato in 'Dinamica non lineare e sistemi complessi', Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Firenze, © Autunno 2015.