Detection and Statistical Evaluation of Spike Patterns in Parallel Electrophysiological Recordings

Pietro Quaglio

Schlüsseltechnologien / Key Technologies
Band / Volume 217
ISBN 978-3-95806-468-3

Mitglied der Helmholtz-Gemeinschaft

JÜLICH
Forschungszentrum

Schriften des Forschungszentrums Jülich
Reihe Schlüsseltechnologien / Key Technologies Band / Volume 217

Forschungszentrum Jülich GmbH
Institute of Neuroscience and Medicine
Computational and Systems Neuroscience (INM-6/IAS-6)

Detection and Statistical Evaluation of Spike Patterns in Parallel Electrophysiological Recordings

Pietro Quaglio

Schriften des Forschungszentrums Jülich
Reihe Schlüsseltechnologien / Key Technologies Band / Volume 217

ISSN 1866-1807 ISBN 978-3-95806-468-3

Bibliografische Information der Deutschen Nationalbibliothek.
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der
Deutschen Nationalbibliografie; detaillierte Bibliografische Daten
sind im Internet über http://dnb.d-nb.de abrufbar.

Herausgeber und Vertrieb:	Forschungszentrum Jülich GmbH Zentralbibliothek, Verlag 52425 Jülich Tel.: +49 2461 61-5368 Fax: +49 2461 61-6103 zb-publikation@fz-juelich.de www.fz-juelich.de/zb
Umschlaggestaltung:	Grafische Medien, Forschungszentrum Jülich GmbH Titelbild: ©ktsdesign/stock.adobe.com
Druck:	Grafische Medien, Forschungszentrum Jülich GmbH
Copyright:	Forschungszentrum Jülich 2020

Schriften des Forschungszentrums Jülich
Reihe Schlüsseltechnologien / Key Technologies, Band / Volume 217

D 82 (Diss. RWTH Aachen University, 2020)

ISSN 1866-1807
ISBN 978-3-95806-468-3

Summary

The computational processes deployed by the brain to represent, process and transmit information are largely unknown. Cell assemblies (highly inter-connected groups of neurons) have been hypothesized to be the building block of the computational processes in the cerebral network. The coordination of spikes emission among neurons at millisecond temporal scale is one of the possible mechanisms of information coding and a signature of assembly activation. In particular, specific temporally precise spike sequences in the input can reliably cause a spike emission in a post-synaptic neuron. Evidences of coordination of the spiking activity at milliseconds precision have been collected in the past, yet such studies present two main limitations: in most cases they consider few neurons recorded in parallel and the correlation analysis are limited to spike synchronicity.

Recent developments of the recording devices overcome the first limitation. Modern electrophysiological technologies enable to obtain the spiking activity of hundreds of neurons in parallel, a number which is destined to grow. The size of the current available data requires optimized computational analysis technique and sophisticated statistical approaches.

In this work we address the second limitation, developing a method to detect spatio-temporal patterns of spikes in large parallel recordings. In particular we extend the Spike Pattern Detection and Evaluation (SPADE) method, originally limited to synchronous patterns detection, to search for any repeated sequence of spikes. SPADE can be summarized in two steps: a) extraction of all the repeated spike sequences using the frequent item-set mining framework, b) statistical evaluation of the significance of the mined sequences in respect to the null hypothesis of independent spike emissions in time. We extensively refined and validated the method using ground-truth artificial data designed to resemble experimental data to test the statistical performances of the method. We then made the python implementation of SPADE publicly available online as a submodule of the Electorphysiological Analysis Toolkit (Elephant).

We applied SPADE to in-vivo parallel recordings of neuronal activity in the motor area of two macaque monkeys performing a reach-to-grasp task, finding a large number of significant spike patterns. We then investigated the statistical features of the detected patterns in terms of neuronal composition, temporal occurrences and relation to behavior. Most of the patterns occur during the reach movement of the task and they are formed by two to four different neurons. Furthermore the neurons forming the patterns differ for different grip types, hinting to a high specificity of the patterns to the different behavioral contexts.

In the last part of this work we compare SPADE to other existing methods in the context of a more general review of methods for the analysis of correlations in parallel spike trains. In particular we argue for the importance of a thorough comparison of the different methods and for the integration different methodologies that highlight different aspects of the correlation structure of the data.

In summary we show that SPADE robustly detects and selects significant precise spike sequences and that multiple significant patterns repeat during the execution of a reach to grasp task. Nevertheless the spatio-temporal patterns alone do not guarantee a complete description of the correlation structure of the data, hence we present and compare alternative correlation analysis methods for parallel spike trains.

Zusammenfassung

Die rechnerischen Prozesse, die das Gehirn zur Darstellung, Verarbeitung und Übertragung von Informationen einsetzt, sind weitgehend unbekannt. Es wird angenommen, dass Zellverbände (hochgradig miteinander verbundene Neuronengruppen) die Bausteine der rechnerischen Prozesse im zerebralen Netzwerk sind. Einer der möglichen Mechanismen der Informationscodierung ist die Koordination von Emission von Aktionspotentialen (Spikes) zwischen Neuronen in Millisekundenpräzision was eine Signatur von der Aktivierung von Zellverbänden ist. Insbesondere, spezifische zeitlich präzise Spikesequenzen im Input, knnen zuverlässig eine Spike-Emission in einem postsynaptischen Neuron verursachen. Evidenz für die Koordination der Spikeaktivität in Millisekundenpräzision wurde bereits in der Vergangenheit gesammelt, dennoch weisen solche Studien zwei Hauptbeschränkungen auf: in den meisten Fällen berücksichtigen sie nur wenige parallel aufgezeichnete Neuronen und die Korrelationsanalyse ist auf Synchronizität beschränkt.

Jüngste Entwicklungen der Aufzeichnungsmethodik überwinden die erste Beschränkung. Moderne elektrophysiologische Technologien ermöglichen das aufnehmen der neuronaler Aktivität von Hunderten von Neuronen in parallel, eine Zahl, die dazu bestimmt ist zu wachsen. Die Gröe der bereits derzeit verfgbaren Daten erfordert eine optimierte rechnerische analyse und anspruchsvolle statistische Ansätze.

In dieser Arbeit befassen wir uns mit der zweiten Einschränkung, indem wir eine Methode entwickeln, um räumlich-zeitliche Muster von Spikes in groen parallelen Aufnahmen zu erkennen. Insbesondere erweitern wir die Spike Pattern Detection and Evaluation Methode (SPADE), die ursprünglich auf die Erkennung synchroner Muster beschränkt war, um nach einer beliebigen wiederholten Sequenz von Spikes zu suchen. SPADE kann in zwei Schritten zusammengefasst werden: a) Extraktion aller wiederholten Spike-Sequenzen unter Verwendung des Frequent-Item-Set-Mining-Frameworks, b) statistische Auswertung der Signifikanz der gefunden Sequenzen in Bezug auf die Nullhypothese unabhängiger zeitlicher Spikeemissionen. Wir haben das Verfahren umfassend verfeinert und validiert, wobei künstliche Daten verwendet wurden, die den experimentellen Daten ähneln, um die statistischen Leistungen der Methode zu testen. Anschlieend haben wir die Python-Implementierung von SPADE öffentlich als Submodul des Elektorphysiological Analysis Toolkit (Elephant) online zur Verfügung gestellt.

Wir haben SPADE auf in-vivo Aufzeichnungen paralleler neuronaler Aktivität im motorischen Areal von zwei Makaken angewandt, die einen reach-to-grasp task durchführten, und eine groe Anzahl signifikanter Spikemuster gefunden. Anschlieend untersuchten wir die statistischen Merkmale der erkannten Muster in Bezug auf die neuronale Zusammensetzung, zeitliches Vorkommen und die Beziehung zum Verhalten der Makaken. Die meisten Muster treten während der Greifbewegung auf und setzen sich aus zwei bis vier verschiedenen Neuronen zusammen. Darüber hinaus unterscheiden sich die Neuronen in den Zellverbänden für die verschiedenen Grifftypen, was auf eine hohe Spezifität der Muster für die verschiedenen Verhaltenskontexte hinweist.

Im letzten Teil dieser Arbeit vergleichen wir SPADE mit anderen existierenden Methoden im Rahmen einer allgemeineren Evaluierung von Methoden für die Korrelationsanalyse von parallelen Spiketrains. Insbesondere argumentieren wir für die Wichtigkeit eines grndlichen Vergleichs der verschiedenen Methoden und für die Integration verschiedener Methoden, die verschiedene Aspekte der Korrelationsstruktur der Daten hervorheben.

Zusammenfassend zeigen wir, dass SPADE es ermöglicht, signifikante präzise Spike-

Sequenzen robust zu erkennen und auszuwählen, und dass sich mehrere signifikante Muster während der Ausfhrung eines reach-to-grasp tasks wiederholen. Dennoch garantieren die räumlich-zeitliche Muster allein keine vollständige Beschreibung der Korrelationsstruktur der Daten, daher präsentieren und vergleichen wir alternative Korrelationsanalysen für parallele Spikedaten.

Author's list of publications

The work presented in this thesis is in part based on the following publications:

Synchronous Spike Patterns in Macaque Motor Cortex during an Instructed-Delay Reach-to-Grasp Task
by Emiliano Torre, ***Pietro Quaglio****, Michael Denker, Thomas Brochier, Alexa Riehle and Sonja Grün*

Published in Journal of Neuroscience (2016).

Parts of this work enter in chapter 3.

The authors had the following contributions:
ET, PQ, MD, TB, AR, and SG designed research; ET, PQ, TB, and AR performed research; ET, PQ, and MD contributed unpublished analytic tools; ET and PQ analyzed data; all the authors contributed to the writing of the manuscript.

Exploring the Usefulness of Formal Concept Analysis for Robust Detection of Spatio-Temporal Spike Patterns in Massively Parallel Spike Trains
by Alper Yegenoglu,* ***Pietro Quaglio****, Emiliano Torre, Sonja Grün and Dominik Endres*
* Equal contribution

Published in Graph-Based Representation and Reasoning, Lecture Notes in Artificial Intelligence (2016).

Parts of this work enter in chapter 2.

The authors had the following contributions:
SG, DE, and ET designed the work. PQ, AY, ET, DE, SG designed the analysis and statistical tools. PQ, AY, ET, DE wrote the analysis software and performed the experiments; all the authors contributed to the writing of the manuscript.

Detection and Evaluation of Spatio-Temporal Spike Patterns in Massively Parallel Spike Train Data with SPADE
by ***Pietro Quaglio****, Alper Yegenoglu*, Emiliano Torre, Dominik Endres and Sonja Grün*
* Equal contribution

Published in Frontiers of Neuroscience (2017).

Parts of this work enter in chapter 3.

The authors had the following contributions:

SG, DE, and ET designed the work. PQ, AY, ET, DE, SG designed the analysis and statistical tools. PQ, AY, ET, DE wrote the analysis software and performed the experiments; all the authors contributed to the writing of the manuscript.

Methods for Identification of Spike Patterns in Massively Parallel Spike Trains
*by **Pietro Quaglio**, Vahid Rostami, Emiliano Torre and Sonja Grün*

Published in Biological Cybernetics (2018).

Parts of this work enter in chapter 5.

The authors had the following contributions:
PQ, VR, ET, and SG designed the work. All the authors contributed to the writing of the manuscript.

A Collaborative Simulation-Analysis Workflow for Computational Neuroscience Using HPC
*by Johanna Senk, Alper Yegenoglu, Olivier Amblet, Yury Brukau, Andrew Davison, David Roland Lester, Anna Lührs, **Pietro Quaglio**, Vahid Rostami, Andrew Rowley, Bernd Schuller, Alan Barry Stokes, Sacha Jennifer van Albada, Daniel Zielasko, Markus Diesmann, Benjamin Weyers, Michael Denker, Sonja Grün*

Published in Jülich Aachen Research Alliance (JARA) High-Performance Computing Symposium (2016).

Not reported in this thesis.

The authors had the following contributions:
PQ was involved in the development and test of the visualization software (NeuronalCorrelationAnalyzer) for visualization of correlation graphs and contributed to the writing of the manuscript.

Reproducible Neural Network Simulations: Model Validation on the Level of Network Activity Data
*by Robin Gutzen, Michael von Papen, Guido Trensch, **Pietro Quaglio**, Sonja Grün and Michael Denker*

Accepted by Frontiers in Neuroinformatics (2018).

Not reported in this thesis.

The authors had the following contributions:
RG, MP, GT, SG, and MD designed the study. RG and PQ performed the analysis. GT performed the simulations and implemented the model. RG, MP and PQ wrote the software for performing the validations. RG, MP, PQ, GT, SG, and MD contributed to writing of manuscript.

Ich erkläre eidesstattlich, dass ich die Dissertation selbstständig verfasst und alle in Anspruch genommenen Hilfen in der Dissertation angegeben habe.

Unterschrift

Contents

Chapter 1

Introduction

This work introduces a new statistical method for the detection of Spatio-Temporal Patterns (STPs) in Massively Parallel Spike Trains (MPST). The method is first extensively validated on artificial data and then applied to electrophysiological recordings of MPST from the motor cortex of a behaving Macaque Monkey.

In the last chapter we tackle the problem of how to compare results of different statistical techniques applied to electophysiological recordings, focusing in particular on methods for detection of spike synchronization and more general spatio-temporal spike patterns.

This chapter introduces the motivation for investigating the presence of spike patterns in modern electrophysiological recordings and for the necessity of a standardized approach to the comparison of different techniques in data analysis.

1.1 Neuron and spikes

The cerebral cortex is an highly interwoven network. Neuronal cells (schematic representation in Figure 1.1A) are the fundamental building blocks of the cortical network. The whole cortex contains billions of neurons connected by thousands of billions of synapses. Each of them can receive inputs from more than 10000 neurons and project to 10000 other neurons (Braitenberg and Schüz, 1991). Neurons are excitable cells that transmit information via electrical impulses, known as action potentials or spikes. The spikes are generated in the cell body (soma) and they travel down along the cell axon to the synapses connecting the cell to its post-synaptic neurons. The input spikes generate a post-synaptic current (PSCs). In the post-synaptic cells all the PSCs are summed and if sufficient inputs in a sufficiently short time arrive to the cell, they trigger a spike, leading to the propagation of the information. In the next paragraph we illustrate this electrical micro-dynamic in more details.

The properties of cell's membrane enable to keep a voltage gradient between the inside and the outside (extracellular medium) of the neuron. This gradient is called membrane potential. In absence of electrical inputs the membrane has a low permeability to exchange of ions (e.g. Na^+, K^+, Cl^-). Consequently, the higher concentration of positive-charged ions extra-cellular medium, in respect to the inside of the cell, leads to a constant negative difference called resting potential. The permeability of the membrane potential can change via the opening of specific protein structures called ion-channels, which are selective to a specific ion type that can pass through. Two important classes of channels are the voltage-gated channels, whose opening probability depends on the membrane potential itself, and the chemically-gated channels, opened by chemicals called neurotransmitters. Input currents to a neuron cause the opening of specific ion channels, allowing the passage of specific ions. This dynamic (excitatory post-synaptic potential, ePSP) changes the voltage gradient, depolarizing the cell towards smaller negative values or even hyper-polarizing it to positive values. In absence of new inputs the membrane potential will gradually decrease again towards the resting potential. Instead, if the increase of the membrane potential is sufficient to reach a characteristic level called firing threshold (e.g due to a large number of incoming spikes in a short time), the voltage-gated channels open. The opening of these channels allows a much faster influx of positively charged ions and consequently to a rapid and strong depolarization. A spike consists of such a fast depolarization which is propagated through the axon to other neurons. Figure 1.1B shows in details the membrane potential trace during a stereotypical spike. The spike can be decomposed in three successive phases: i) when the membrane potential reaches the firing threshold, the Na^+ voltage channels open leading to a fast hyper-polarization, ii) after the potential reached a maximum, the Na^+ channels close and the K^+ channels open and the efflux of K^+ causes a fast negative excursion of the membrane potential, iii) after reaching a negative peak also K^+ channels close and the resting potential is restored. This last phase, which lasts for about 1 ms, is also called refractory period, since during this time span the neuron is not excitable and cannot emit new spikes.

The focus of this study is not the microscopic dynamics of spike emission but rather the interactions between multiple neurons that propagate and process information. The single spikes are the "atoms" of the neural code and can be metaphorically considered as the single letter used by neurons, all together forming the alphabet used by the brain. We are rather interested in the next level of complexity of the brain language, i.e. what could be considered as the words and the sentences that the neurons use to encode the information. In the next section we introduce two different hypotheses about the mechanisms (in terms of our metaphor the syntax forming the brain's language) that the cortical network can implement to propagate information using sequences of spikes.

The electrical excursion (spike wave) of the membrane potential during a spike emission is stereotypical for each neuron (e.g. the spike amplitude and duration is the same for each spike emission). When interested in the macroscopic interactions between neurons and not to the details of the electro-chemical dynamics regulating spike emission, it is then possible to reduce the neuronal activity to a sequence of events, each related to the time of emission of one spike. Mathematically such sequence of events can be described as a point process, often referred to with the term spike train. To reduce each spike wave to a single point in time it is possible to consider the time of one specific feature of the complete electrical trace (e.g. the positive peak, correspondent to the Na^+ closing or the negative one, corresponding to the K^+ channels closing).

1.2 Spike patterns: temporal and rate coding

The discovery of spikes (Adrian, 1926b) and of the microscopic mechanisms that regulate the electrophysiology of neurons, led to the modern interpretation of a spike-based information process implemented in the brain. Nevertheless the exact scheme that regulates this processes is widely debated and most of the efforts deployed in the analysis

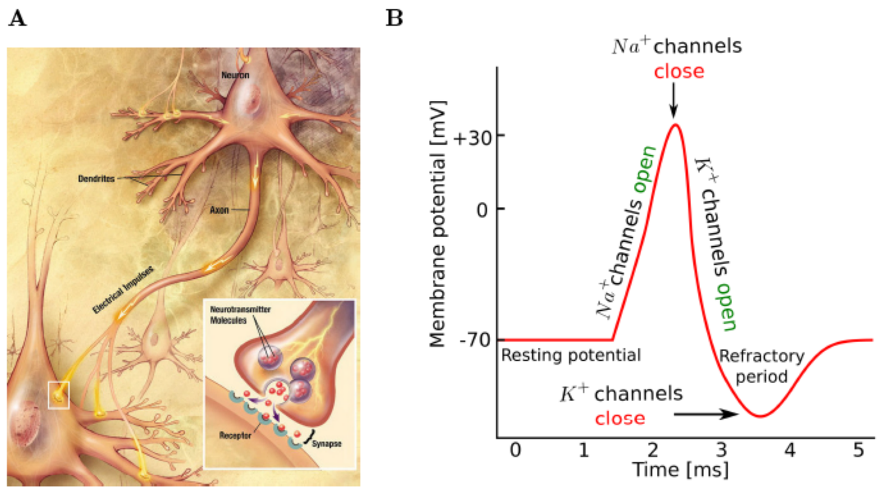

Figure 1.1: **Sketch of a biological neuron and an action potential. A)** Structure of a typical neuron and a chemical synapse (source: US National Institutes of Health, National Institute on Aging). **B)** Stereotypical shape of the action potential. Synaptic inputs to the neuron raise the membrane potential from the resting potential. This rise happens because of the opening of Na^+ channels and it continues until the potential reaches a maximum value. Then the Na^+ channels get inactivated. At this point the K^+ channels open and restore the membrane potential.

of electrophysiological recordings aim to provide new hints towards the understanding of the brain syntax. Two fundamental properties of neurons have to be taken into account to formulate a hypothesis about the computational scheme implemented by the cerebral cortex.

The first is the leakiness of the neuronal membrane. As already explained in the previous section, in absence of inputs the membrane potential tends to decrease towards the resting potential. In terms of computation this implies that to propagate the information and generate a spike in a post-synaptic neuron, it is necessary that pre-synaptic neurons emit multiple spikes in a short amount of time. In particular synchronous ePSPs maximize the probability of spikes emission.

The second property was already hypothesized by Hebb (1949a). In his work Hebb (1949a) hypothesized that temporally precise causal relations between spikes emissions can shape (reinforce or depress) the connections between neurons. This hypothesis, known as Hebbian Rule, has been experimentally confirmed by the discovery of spike-time dependent plasticity (STDP) (Bi and Poo, 1998). STDP is a mechanism of synaptic adaptation, which consists in potentiation (Bliss and Lomo, 1973; Lømo, 2003) and depotentiation (Stent, 1973; Massey and Bashir, 2007) of synaptic strength depending on the input-output temporal relation between two connected neurons. In particular, if a spike in the pre-synaptic neuron precedes by few milliseconds the emission of a spike in the post-synaptic cell (causal relation) then the synaptic connection is strengthened, vice versa if a spike in the post-synaptic cell precedes one in the pre-synaptic one the synapse is weakened. The sign (positive corresponding to potentiation and negative corresponding to depotentiation) and the amplitude of the adaptation is highly dependent on small variation of the delays between pre- and post-synaptic spikes.

Hebb derived by his rule a more elaborate learning scheme and introduced the concept of cell assembly. Cell assemblies are defined as groups of highly interconnected neurons (generated via the potentiation of specific synaptic pathways) that are hypothesized to be the building block of the information process in the brain. Such theory is also known as Hebbian learning and anticipated modern biological computation theories decades ago. In particular, in the last decades two prominent hypothesis on how spikes are organized to encode the information emerged (for a review see Brette, 2015): rate coding and temporal coding. The fundamental difference is that in the rate framework the information contained in the precise spike times is irrelevant and the information is rather encoded in the average firing rates (e.g. average number of spikes in time intervals), while at the opposite in the temporal coding the information is carried by the exact time emission of the spikes.

1.2.1 Rate coding

Firing rate modulation has been one of the earliest observation after the discovery of spikes (Adrian, 1926a). In particular it has been observed across different cortical areas and species that specific neurons increase their firing rate as a function of different stimuli or task performed (e.g. Georgopoulos et al., 1986; Lamme and Spekreijse, 1998; Roelfsema et al., 2004). This property is known as receptive field, since neurons appear to be tuned to a preferred stimulus and/or action (e.g. Figure 1.2 shows the tuning of to specific direction of movement in the motor area of a macaque monkey). In the context of the theories about information propagation in the brain such observation led to the rate code hypothesis. A

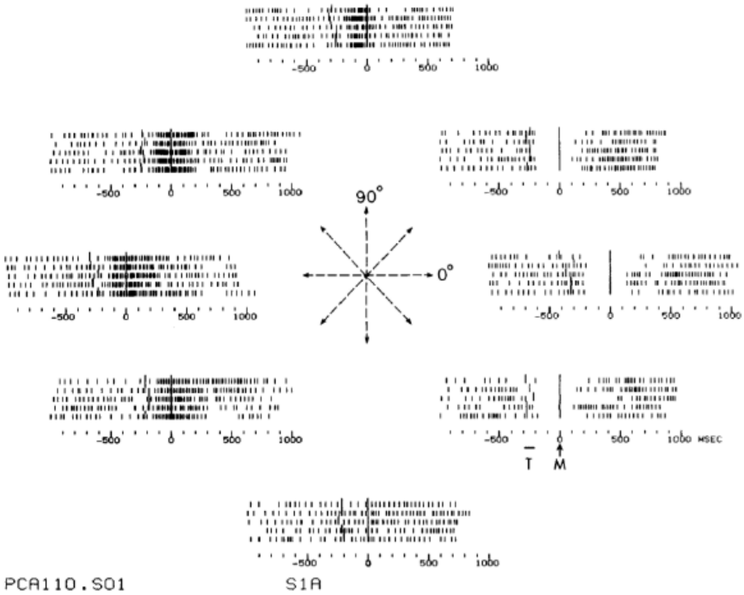

Figure 1.2: **Tuning of motor neurons in a Macaque Monkey.** Intracellular recording of the spiking activity of a motor cortical cell in a macaque monkey performing a center-out movement in six direction. All the spikes emitted from the cell are showed for different trials in which the monkey moved a cursor in one of the 6 different direction. The time is aligned with the movement on-set. The neuron show a preferred direction for which it increase the rate of spike emission. Vice-versa for the opposite direction the firing rate decrease around the movement onset. (Figure modified from Georgopoulos et al. (1982)).

neuron processes its input by integrating the incoming spikes in temporal windows of tens to hundreds of milliseconds. Hence the information is encoded in rate modulations rather than in the precise occurrences of the single spikes. This makes the computation robust to noise and random fluctuations of the membrane potential (Shadlen and Newsome, 1994, 1995; London et al., 2010). On the other hand several studies also suggest that this computational scheme would be too slow to propagate information (Gautrais and Thorpe, 1998; VanRullen et al., 2005) and too limited to provide a unique representation of the huge variety of sensory stimuli that has to be encoded (Gerstner et al., 1997).

1.2.2 Temporal coding

In contrast to rate coding, the temporal code is relying on the fact that the neurons form an interwoven and complex network that can operate with a high temporal precision, producing highly correlated and structured spike outputs. In particular, it is well possible that their connections are organized through synaptic modulations (STDP) such that neurons operate as coincidence detectors (e.g. a spike emission in a postsynaptic neurons is causally related to the emission of multiple spikes from a specific set of presynaptic neurons, arriving synchronously to the postsynaptic neuron) (Abeles, 1982). Furthermore neurons have been shown to propagate more reliably synchronous inputs rather than asynchronous, due to the leakiness of their memebrane (Kandel et al. (1991), chapter 12). Experimental evidences for the existence of neurons that work as coincidence detectors have been collected in several studies in the last decades (Roy and Alloway, 2001; Bender et al., 2006; Fino et al., 2010; Perez-Orive et al., 2004; Hong et al., 2012). Building on the concept of coincidence detectors, for the temporal code hypothesis the information is encoded in structured packet of synchronous spikes that propagate through the network. Detailed and formalized models that can implement the temporal coding scheme have been proposed and one of the most prominent is the synfire chain model (Abeles, 1991; Diesmann et al., 1999; Ikegaya et al., 2004; Hosaka et al., 2008). Spike synchronization has been found in recordings of parallel spike train in a number of different species and brain areas, such as auditory cortex (Seki and Eggermont, 2003; Carr, 2004; Eggermont, 2015), retina (Van Rullen and Thorpe, 2001; Hu and Bloomfield, 2003; Shlens et al., 2006; Pillow et al., 2008), visual cortex (von der Malsburg, 1986; Engel et al., 1992; van der Togt et al., 2006; Berger et al., 2007; Smith and Kohn, 2008; Martin and von der Heydt, 2015), motor cortex (Riehle et al., 1997b) (Figure 1.3), (Baker et al., 2001; Shimazaki et al., 2012; Torre et al., 2016b), somatosensory cortex (Steinmetz et al., 2000; Reed et al., 2008; Harvey et al., 2013), hippocampus (Sakurai, 1996; Diba et al., 2014), frontal and prefrontal cortex (Vaadia et al., 1995; Sakurai and Takahashi, 2006; Fujisawa et al., 2008; Pipa and Munk, 2011). This studies showed that the occurrences of patterns of synchronous spikes is related to behavior, supporting the idea that synchrony is used to encode and propagate sensory inputs and outputs.

Testing the hypothesis of rate and temporal code presented from the very beginning highly challenging problems. If on the one hand rate changes and spike synchronization in relation to different stimuli have both not only been observed in data since decades, but can also be considered a trivial expectation for the dynamics of a neural network. However, disentangling the two options is as necessary as non-trivial when one wants to test the different coding hypotheses. The first challenge is given by the fact that the two observations can reciprocally be a byproduct of the other: rate changes could be a consequence of spike synchronization and vice versa increasing rates cause an increase of spike synchronization. In the first scenario it could be that the receptive field that is commonly considered as a primary feature of single neurons could be the byproduct of a network effect (e.g. precise synchronization of multiple neurons) that involve other neurons that are not recorded. On the other hand increased firing rate of multiple neurons increases trivially the chance of synchronous spike emission in such neurons. Furthermore not only it is difficult to distinguish the two phenomena, but also they might well be both implemented in the cortical network and be used for information processing. In a simpler scenario it is possible that different brain regions of different neurons implement different coding frameworks, but it could also be that the two coexists in the same area. Hence it is crucial to be able to distinguish between possible correlation on the different time scales. For this reason it is crucial to develop sophisticated and ad hoc techniques to detect and evaluate the statistical significance of correlations of the spiking activity of neurons recorded in parallel and on the different time scales. In particular to test the temporal hypothesis it is not sufficient to detect synchronous spiking activity, but it is

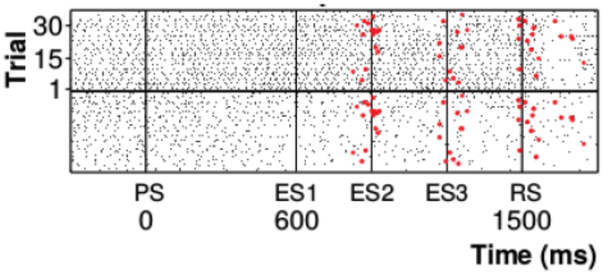

Figure 1.3: **Increased synchronization of two cell in correspondence of a motor cue.** Intracellular simultaneous recordings of two motor cell for different trials of a reach and touch task in the motor area of a macaque monkey. In red are highlighted synchronous spikes which occurred more often than what expected by chance synchronization. The monkey was trained in previous trials to expect the cue to start the movement in 3 different time point (ES1, ES2, ES3), preceding the actual time (RS) of the visual cue that signals the monkey to reach and touch a screen. The fact that synchronous events are aligned with two of these expect cues and with the actual go signal suggests that they are signature of the preparatory process for the movement. (Figure modified from Riehle et al. (1997b)).

necessary to test for their significance in respect to the firing rate dynamic (e.g. firing rate modulations). In particular in the last chapter, we will introduce different methods that aim to analyze MPST to test the temporal coding hypothesis distinguishing chance by statistically significant correlations. In this work we refer to precise temporal correlation to indicate synchronization of individual spikes (e.g. on millisecond time scale) and to firing rate correlation for average spike counts correlations (e.g tens of milliseconds time scales).

1.3 Beyond synchronization: Spatio-Temporal Patterns

So far, we have purposely avoided to include in our argument another fundamental property of the cortical network: the axonal conduction delays. Different experiments showed a large variability of spike propagation speed in the mammalian neocortex (Ferster and Lindström, 1983; Swadlow, 1988, 1994; Salami et al., 2003). In particular the axonal delays can be as small as 0.1 ms and as large as 44 ms (Figure 1.4). Such an experimental observation poses a natural question: why would the brain maintain such different but reliably precise spike propagation delays? A possible answer is given in the context of the temporal coding: the different delays are used to increase computational performance. While in the case of same delays, a given set of neurons can form one unique patterns and the information of a spike pattern is completely and only contained in the neuronal composition of the pattern (which neurons emits a spike synchronously), including in the same coding scheme different propagation delays enable a set of neurons to form a multitude of patterns depending on the order in which they spike (e.g. the same neurons spiking in different spiking sequences may activate different post-synaptic cells).

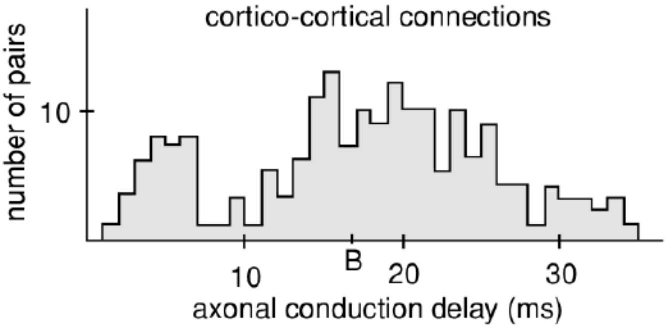

Figure 1.4: **Distribution of experimentally measured conduction delays of cortical axons running through the corpus callosum** (Figure modified from Izhikevich (2006)).

To better understand this concept it is possible to think of a simple example (see Figure 1.5) in which we have two pre-synaptic neurons (A,B) projecting to two different post-synaptic neurons (C, D). Lets assume that the synchronous input of two spikes is sufficient to elicit a new spike in the post-synaptic neurons. In the case of identical delays between pre- and post-synaptic neurons, every time that neurons A and B spike synchronously, both neurons C and D will emit a spike. With such scheme the two post-synaptic neurons are replicas and it is not possible to use them to encode different information. In order to use them as a classifier (e.g. distinguishing between two different inputs given that one between either C emit a spike or D emit a spike) we would need to add at least 2 more neurons E,F. If each of them is respectively connected to only one of neuron C and neuron D it is then possible to elicit a spike only in one of the two post synaptic neurons. In such a way it would be possible to represent three different inputs using three different possible synchronous patterns: the pair A,B elicits a spike in both C and D, the pair A,E leads only neuron C to spike while the pair B,F causes a spike only in neuron D. In other words, this simple network is capable to implement the AND/OR operations.
Yet, the same computation is possible also using only two pre-synaptic neurons if the connectivity allows for heterogeneous conduction speeds. In such a case, it is possible to organize the spike emission times in different temporal sequences in order to elicit a spike in both post-synaptic neurons or, alternatively, in only one of the two. For instance we can consider the case in which a spike emitted by A needs 2 milliseconds to reach neuron C and 4 milliseconds to neuron D, while for the connections between B and C and B and D have the opposite propagation delays (B to C 4 milliseconds and B to D 2 milliseconds). In such case it is possible to cause a spike only in C whenever A emits a spike 2 milliseconds before B and viceversa D emits a spike whenever a spike form B is followed by a spike in A after 2 milliseconds. In order to have the AND condition in which both C and D emit a spike synchronously it sufficient that A and B both emit two spikes each, one 2 milliseconds after the other. This is just a biologically unrealistic toy-example and many other network implementations of an AND/OR function exist, nevertheless it

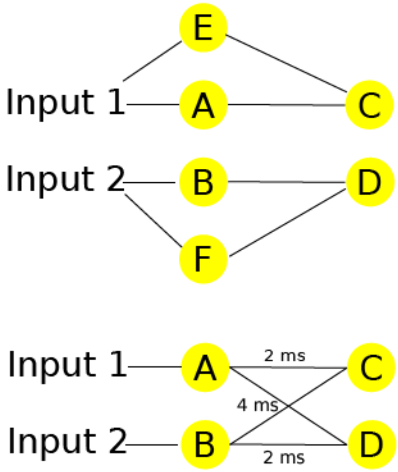

Figure 1.5: **Sketch of an example of two functionally equivalent networks:** one with a larger number of neurons (top) and the second with propagation delays (bottom)

makes evident how "storing" the information also in the connectivity delays may reduce the number of neurons necessary to deliver the activity to the relevant network.
In such a context the temporal coding for the propagation of information would rely on repeated precise sequences of spikes (Spatio-Temporal Patterns, STPs), rather than synchronous spike emissions only.

1.4 Detection Spatio-Temporal Patterns in Massively Parallel Spike Trains

Several studies (e.g. Abeles, 1991; Bienenstock, 1995; Diesmann et al., 1999; Izhikevich, 2006) studied the formation of spike patterns in network models, showing that these can be reliably propagated through the network and could emerge from the development of specific connectivity patterns due to synaptic plasticity.
Other works showed evidences of the occurrences of precise STPs in electrophysiological recordings (e.g. Prut et al., 1998; Takahashi et al., 2015) and introduced statistical analyses to detect spike patterns (e.g. Shimazaki et al., 2012; Torre et al., 2016a). The main challenges posed by the detection of spatio-temporal patterns in parallel spike trains are the following:

- the computational cost of the extraction from the data of all repeated spike sequences, which number grows exponentially with the number of parallel neurons
- the multiple testing problem caused by the large number of possible tests to perform in order to assess the significance of the patterns

For such reasons most of the methods introduced in the past consider only the interaction between few neurons or focuses on specific temporal structures which are not generalizable

to arbitrary sequences of spikes.
In this work we introduce a method for detection and statistical evaluation of spike patterns in massively parallel spike trains. In particular, we extend the Spike Pattern Detection and Evaluation (SPADE), method that Torre et al. (2013) developed for detection of synchronous spike patterns, to arbitrary sequences of spikes. We validate the method in multiple scenarios with artificial data and we then apply the method to two datasets of parallel spike trains recorded in vivo in the motor cortex of a macaque monkey performing a reach-to-grasp task.

1.5 Reproducibility and Comparability

The continuous growth of available data and analysis methodologies should allow a constant increase in the understanding of the fundamental cognitive mechanisms. Nevertheless, the efficacy of the research depends on the achievement of two fundamental standards: reproducibility and comparability.
The minimal requirement to validate a scientific study is its reproducibility. A results that cannot be reproduced and consequently validated is not beneficial to the scientific field development. The complexity of the experimental setups and analysis methodologies adopted in contemporary neuroscience poses the challenge of preserving the reproducibility of the research (e.g. Denker and Grün, 2015; Manninen et al., 2018). In this study we address such problem providing and publishing an implementation of the novel method in the context of the Elephant Python package (http://python-elephant.org) and publishing online the entire workflow for the analysis of the data.
The reproducibility of the research is not the only requisite for the advancement of a certain field of studies. It is also crucial to have the possibility to compare results among different datasets and methodology applied. In the second of this two scenarios it is firstly necessary to compare different assumptions and limitations of the analysis tools used. In this work we address such challenge by developing a comparative review of the available methods for detection of synchronous and spatio-temporal patterns in massively parallel spike trains.

Chapter 2

Detection of Spatio-Temporal Spike Patterns in Massively Parallel Spike Trains using Formal Concept Analysis

2.1 Introduction

In this chapter we present an approach to detect spatio-temporal patterns (STPs) of precisely timed spikes occurring in parallel spike trains.

Torre et al. (2013) addressed this problem by introducing an efficient method (Spike Pattern Detection and Evaluations, in short SPADE) to assess the presence of patterns of synchronous spikes in massively parallel spike trains that occur inexcess to the expectation. In a first step this method exploits to this aim frequent itemset mining techniques (Borgelt, 2012), and in a second step assesses the statistical significance of detected patterns. However, it can only detect synchronous spike patterns, and not more general STPs in which the spikes belonging to a pattern follow each other in a temporal sequence.

We develop a novel approach designed to find and evaluate such types of STPs. Compared to the analysis of synchronous spike patterns, this poses a number of additional challenges. One challenge lies in the increased number of patterns to look for. Adding the temporal dimension yields a number of possible patterns which is orders of magnitude larger. The occurrences of each of these patterns have to be counted, and non-chance patterns have to be differentiated from chance patterns based on properties such as the number of composing spikes or the number of pattern repetitions. Another challenge lies in the decreased contrast between these STPs and background activity, due to the fact that the events forming the STPs are here individual spike scattered across multiple points in time rather than synchronous spikes, leading to a higher-dimensional space in which patterns are represented. These leads to a large increase in the number of required computations, that prevents the practical application of the method in its original form. We deal with the first challenge by considering only patterns that actually happened, rather than all possible patterns, making use of the mining techniques provided by the formalism of the Formal Concept Analysis (FCA) (Ganter and Wille, 1999). More precisely, as we describe in section 2.2.1, we study patterns that form the intent of formal concepts,

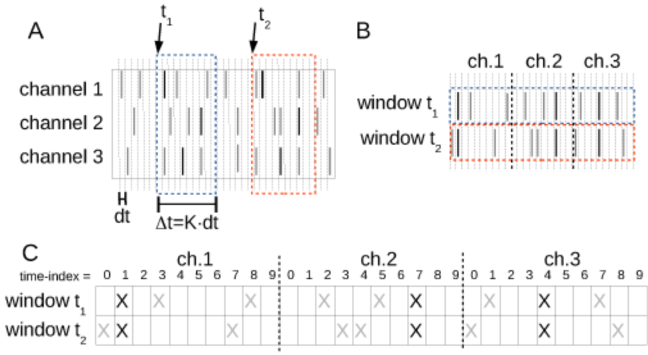

Figure 2.1: **Constructing a formal context from spike recordings in parallel channels.** **A:** we analyze parallel neuronal recordings in multiple channels by discretizing time into bins of duration dt and chopping time windows of length $\Delta t = K \cdot dt$ out of the recorded data stream. The gray spikes are background spikes, the black spikes are members of an STP. **B:** we concatenate the recorded spike trains within a window horizontally. **C:** incidence table representation of panel B. The objects of this formal context are the time windows, indexed by their starting time. The attributes are the spike time-indexes relative to the window start, combined with the channel identities.

repeated in multiple position of a sliding window. We address the second challenge by evaluating the intensional stability (Kuznetsov, 2007) of these concepts as an indicator for the non-randomness of a concept, as explained in detail in section 2.2.1.

In section 2.3.2 we validate the proposed method on test data consisting of parallel spike trains which comprise different independent background activity, each including different known statistical properties of parallel spike trains (e.g. firing rate changes), where patterns arise purely by chance, and multiple occurrences of an STP injected in the independent data. We investigate the performance of the new method in terms of true positive (TP) and false positive (FP) detections of STPs and quantify their precision and recall properties. We also demonstrate how the computational load of the method varies for different data and analysis parameters.

2.2 Methods

2.2.1 FCA on spike data

Regarding FCA, we use the standard definitions (Ganter and Wille, 1999): a formal context is a triple $K = (G, M, I)$ comprising a set of formal objects G, a set of formal attributes M, and a binary relation I between the objects in G and the attributes in M. If $(g, m) \in I$ with $g \in G$, $m \in M$, we also write gIm. We denote the set of all attributes shared by a set of objects $A \subseteq G$ as $A' = \{m \in M | \forall g \in A : gIm\}$ and likewise the set of all objects which have all the attributes in $B \subseteq M$ as $B \subseteq M$: $B' = \{g \in G | \forall m \in B : gIm\}$.

A formal concept is a tuple (A, B) with extent $A \subseteq G$ and intent $B \subseteq M$ such that $A' = B$ and $B' = A$. Let $\mathfrak{B}(K)$ be the set of all concepts of K. Concepts are (partially) ordered under set inclusion on the extents: $\forall (A_1, B_1), (A_2, B_2) \in \mathfrak{B}(K) : (A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2$. $(\mathfrak{B}(K), \leq)$ is a complete lattice (Ganter and Wille, 1999).

In our application, the objects are time windows within which spiking activity of neurons was observed. The example in Figure 2.1A depicts two such windows labeled by their starting times $t_1$and t_2. The attributes are tuples (channel-number, time-index) of the channel (or neuron) number from which a given spike was recorded, and of the spike time index relative to the window onset. Figure 2.1 illustrates the process of computing the relation I from spike data. Suppose we record spike trains (temporal sequences of spikes) from 3 channels, each channel recording the spikes of one neuron. We discretize the time axis into contiguous bins of duration dt. Then, we slide a time window of duration $\Delta t = K \cdot dt$ across these data, in increments of dt. dt is chosen depending on the resolution of the recording device and on the analysis needs, K is selected based on the expected maximal duration of an STP. We set $dt = 1\,\mathrm{ms}$ throughout this paper, which ensures that there is at most one spike from the same neuron in each bin. The value of K is discussed below. The contents of each window across all channels are then concatenated horizontally (see panel B). Lastly, spikes are converted to crosses, yielding the familiar incidence-table representation of I (panel C), to which we apply FCA. We use a pure Python implementation of the fast-FCA algorithm (Lindig, 2000). This algorithm creates the concepts in an order that embeds the usual concept order, which simplifies the subsequent evaluation of stability.

Figure 2.1A also shows by black ticks an STP. In this example, these spikes correspond to the attribute set $B = \{(1,1), (2,7), (3,4)\}$. $\mathfrak{B}(K)$ contains a concept whose intent consists of these spikes only, plus concepts whose intent comprises time-shifted versions of these spikes at all possible times within Δt. Any of these concepts corresponds to the STP we are interested in. We arbitrarily choose the one where the first spike is aligned with the window onset. However, typically such an STP does not appear in isolation, but is embedded in background spiking activity (gray spikes). Hence, after the application of FCA, there will be many concepts which are due to these background processes, and which we wish to separate from the STP concept. We experiment with conceptual stability analysis for this purpose (Kuznetsov, 2007). Specifically, we compute an intensional stability index of concepts $(A, B) \in \mathfrak{B}(K)$ (Roth et al., 2008) by

$$\sigma(A, B) = \frac{|\{C \subseteq A | C' = B\}|}{2^{|A|}} \tag{2.1}$$

with the algorithm described in that paper, and filter out all concepts whose stability index is lower than a pre-fixed threshold (see Section 2.3.1). Furthermore, we filter out concepts that are time-shifted superconcepts of concepts whose stability index is higher than the threshold.

2.2.2 Ground truth data generation

To test the performance of FCA in detecting STPs we use artificial ground truth data with controllable pattern occurrences and pattern size. We generate parallel Poisson processes to simulate independent background activity. We also generate data sets which contain STPs by injecting spike patterns into independent background activity, cf. (Grün et al., 2008; Berger et al., 2010a). The background rates are chosen to comply with the firing rates of experimental neurons, while the type of spike patterns resembles that which would be observed in data from synfire-like networks (Abeles, 1991; Bienenstock, 1995; Izhikevich, 2006).

For the background activity we generate independent spike trains as realizations of 50 parallel independent Poisson processes of duration $T = 1$ sec. The theoretical firing rates $r \in \{5\,\mathrm{Hz}, 10\,\mathrm{Hz}, 15\,\mathrm{Hz}\}$ used to draw the processes are identical for all neurons. We generate 100 realizations for each value of r, yielding a total of 300 different background activity-only data sets. Such independent data sets are used to study the occurrence of false positives (FP), i.e. cases where patterns are detected although there were none injected.

In a second type of data set, which we use for the evaluation of true positives (TPs) and false negatives (FNs), we generate c occurrences of an STP composed of a sequence of z spikes from different neurons. To this end, we randomly select c time points $t_1, \ldots, t_c$ in the interval $[0, T - z \cdot 5\,\mathrm{ms}]$. These times correspond to the times of the first spike of the patterns for each of the c repetitions of the pattern. Each following spike of the STP is injected 5 ms after the preceding one and into a different neuron. Therefore, the total duration of such patterns is $(z-1) \cdot 5\,\mathrm{ms}$. The pattern is chosen with regular spike delays for convenience, but this choice is not relevant for their detectability. We vary both z and c in the range $\{3, 4, \ldots, 10\}$. For each of the 64 combinations of these two parameters we generate 100 different realizations of the patterns, which we inject in an independent background activity data set generated as explained above, for each rate level r. This yields a total of $64 \cdot 100 \cdot 3 = 19200$ data sets, each containing a total of 50 neurons, z of which are involved in an STP. We then extract all concepts from each data set using FCA as explained in section 2.2.1. Since the maximum pattern length is equal to 45 ms for $z = 10$, we fix the length K of the sliding window to 50 ms, so that the longest pattern is covered by one single window. For a data set containing a pattern with parameters (z, c), we define as a TP the correct detection of the concept (A, B), such that $A = \{t_1, \ldots, t_c\}$ and $B = \{(1, 0), (2, 5), \ldots, (z, (z-1) \cdot 5)\}$. Any other detected concept is considered as an FP. Since in each of the data sets only one pattern type is injected, the number c_{TP} of TPs per data set is either 0 or 1. Reciprocally, the number of false negatives is $c_{\mathrm{FN}} = 1 - c_{\mathrm{TP}}$. For a given parameter tuple (r, z, c), we determine the fraction of the 100 data sets in which we find an FP or an FN, providing us with the FP rate and the FN rate, respectively.

2.3 Results

2.3.1 Independent data

We start evaluating the performance of the method by analyzing the occurrence of FPs in the background-only data sets. Concepts detected in these data sets are thus chance occurrences of specific spike sequences and are considered as FPs. The two parameters by which we characterize a concept (A, B) are its extent size $|A|$ and its intent size $|B|$. In terms of the spike data the extent size corresponds to the number of windows in which a specific sequence occurred, and the intent size to the number of spikes composing the sequence.

Figure 2.2 shows in gray code the number of concepts detected in independent data as a function of their intent size (horizontal axis) and extent size (vertical axis). We call this type of display 'pattern spectrum' in line with (Gerstein et al., 2012) and (Torre et al., 2013). Concepts with an extent or an intent of size 1 are not counted or displayed,

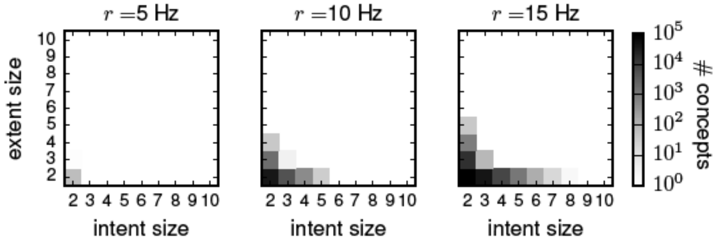

Figure 2.2: **Pattern spectra of independent background-only data:** each panel shows the concept counts as a function of their intent size $|B|$ (horizontal axis) and of their extent size $|A|$ (vertical axis) detected in 100 independent data sets, composed of 50 parallel spike trains without any pattern injection mimicking background noise, for different firing rates (from left to right: $5, 10, 15$ Hz). The number of counts is given in gray colors, as indicated by the color bar (white corresponds to 0, black to 10^5 counts).

since single spike occurrences and non-repeating sequences are not considered as potential STPs. Each panel shows the pattern spectrum for a different neuronal firing rate r. The area of non-zero entries and thus the total number of concepts increases with the firing rate of the neurons (see also Figure 2.3), which is expected since more spike sequences may occur by chance at higher firing rates due to the increased number of spikes. Most detected concepts have extent size of $|A| = 2$ or intent size of $|B| = 2$.

As mentioned above, we aim to select the true positive STPs by applying a stability analysis to the concepts. We wish to determine a stability threshold value such that the concepts that occur by chance are discarded. A suitable threshold value needs to fulfill the following constraints: 1) avoid detection of FPs, 2) have a high degree of TP detection. The histograms in Figure 2.3 (left to right) correspond to the pattern spectra shown in Figure 2.2 (left to right). The gray parts of the stacked bars show the counts of concepts whose intent contains only 2 spikes ($|B| = 2$) or whose extent contains only two windows ($|A| = 2$). Vice versa, the black parts correspond to the concept count for $|A| > 2$ and $|B| > 2$. Concepts with stability larger than or equal to 0.6 have always either intent or extent size equal to 2. This fact provides us with a suitable criteria for classifying all concepts which have either intent or extent size 2, or stability ≤ 0.6 as chance patterns.

2.3.2 Performance of pattern detection

Now that we have defined a suitable criteria to reject all the chance patterns composed only of spikes of the background activity (intent> 2, extent> 2, stability> 0.6), we can evaluate the performance of the method for the detection of STPs injected into artificial data, as described in section 2.2.2. Figure 2.4, top row shows the number c_{FP} of FPs for data with injected patterns, and background rates $r = 5, 10, 15\,\mathrm{Hz}$ from the left to the right panel. Each panel shows, in shades of gray, the number of data sets which contain at least one FP as a function of the number z of spikes (horizontal axis) and of the number c of occurrences of the pattern injected in the data sets (vertical axis). Because 100 simulations were carried out for each parameter set (z, r, c), the value of each entry ranges from 0 to 100. The total number of FPs is low (usually 3 or lower) but increases both with the firing rate, as we can expect from the results of the previous section, and

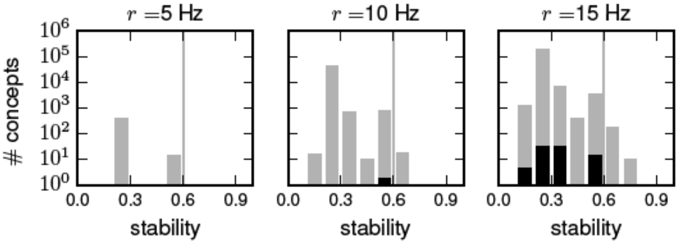

Figure 2.3: **Stability histogram of the concepts in independent background-only data:** each of the three panels shows the histogram of the number of concepts found in independent background data, for different stability values. The data analyzed are the same as shown in Figure 2.2, with the firing rates of the neurons changing from 5 to 15 Hz in the panels from left to right. The height of each bar in the histograms represents the number of concepts whose stability is in the range indicated on the horizontal axis (bin width: 0.1). The black bars (that is, the black parts of the stacked bars) show the counts of concepts with both extent and intent size larger than 2, whereas the gray bars show their counts when one of the two size values is equal to 2. The thin gray line indicates a threshold for the stability above which no black bars are visible, indicating that patterns with this or higher stability values have extent or intent size (or both) 2.

with the pattern parameters z and c. The reason for the latter is that the larger and more frequent the injected pattern is, the more likely is the repetition of spurious sequences of spikes which by chance occur with the injected pattern and which thus result in a concept with a stability index > 0.6.

In the second row of Figure 2.4 we show equivalent diagrams with FN counts (c_{FN}). Each entry shows the number of data sets (out of 100 realizations) in which the injected pattern is not detected. For each rate level, the number of FNs shows a sharp decrease at a particular number of pattern injections c, independent of the pattern size z (e.g. for $r = 5\,\mathrm{Hz}$ at $c = 4$). This border increases to higher c levels for higher firing rates (from left to right). The number of FNs increases with the rates due to an increased probability of spurious superpatterns of the injected STP, causing the correct STP to be ignored by our current concept filtering procedure in favour of the superpatterns. The independence from the pattern size is a feature of intensional stability, combined with our data generation process: all windows containing the target STP are guaranteed to contain all its constituting spikes (plus possibly additional noise spikes), no matter how large z is. For all studied firing rates, patterns occurring just 3 times are never detected, but all patterns occurring 5 or more times are correctly detected, which is a consequence of our choice of the stability threshold.

In order to get further insight in the performance of the our approach, we compute for each data set with parameters (r, z, c) its precision and recall as defined in Olson and Delen (2008):

$$\text{precision} = \frac{c_{\mathrm{TP}}}{c_{\mathrm{TP}} + c_{\mathrm{FP}}},\ \text{recall} = \frac{c_{\mathrm{TP}}}{c_{\mathrm{TP}} + c_{\mathrm{FN}}}$$

We calculate the two measures separately for the same data sets used for the FP/FN evaluation. Figure 2.4, third row shows the average precision and recall as a function

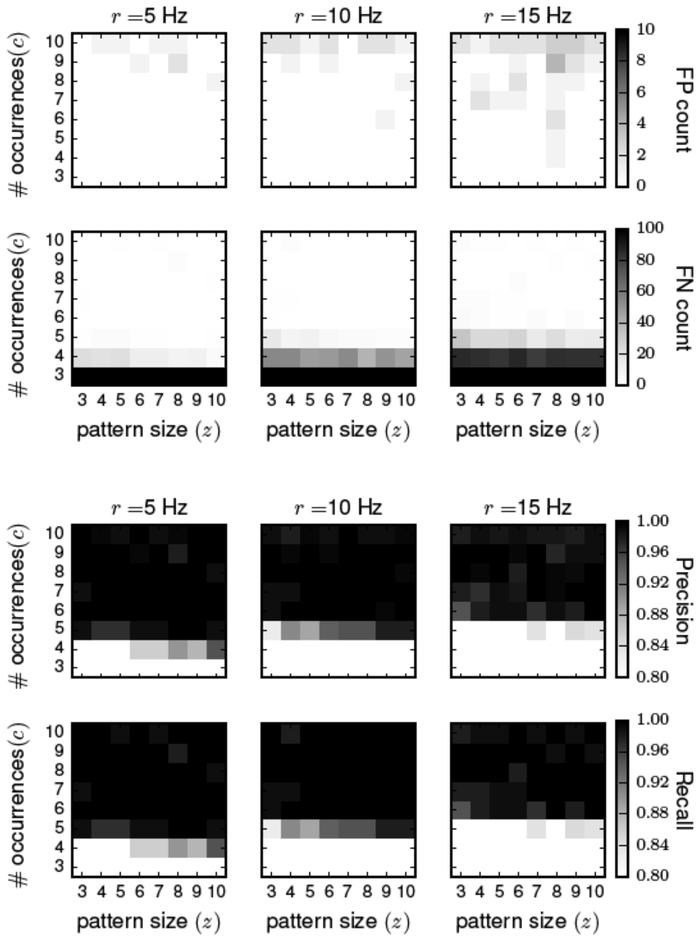

Figure 2.4: **Performance of the method for various data parameters. Top half: FPs and FNs in data with injected STPs:** each panel shows the number of data sets which contain at least one FP or one FN after filtering the concepts using a stability threshold of 0.6 and a lower threshold value of 2 for both intent and extent size. The firing rates are varied from left to right from $r = 5$ Hz to $r = 15$ Hz. For each display we varied along the horizontal and vertical axis the size z and the number of injections c of the injected pattern, respectively. **Bottom half: Precision and recall** corresponding to the FP/FN counts shown in the top half. The color bar for precision and recall ranges from 0.8 to 1.0, whereas all values below ≤ 0.8 are set to white.

of the parameters (r, z, c). Since we are particularly interested in the high performance regime, we emphasize this range by adjusting the color code such that it covers only the interval $[0.8, 1.0]$, and we set all values ≤ 0.8 to white. Black corresponds to a value of 1.0. The performances expressed by either measure are very similar, which is due to the fact that in the high precision/high recall regime, both c_{FP} and c_{FN} are very small. Both are close to 0 for low numbers of pattern occurrences c, but drastically increase for data sets with 5 or more injections, and are basically independent of the pattern size z. This is due to the behavior of the FNs.

The acceptable tradeoff between FPs and FNs depends on the hypothesis to be tested. Since we wish to show the existence of STPs, we prefer to choose a conservative threshold guaranteeing virtually no FPs, even if that incurs some FNs.

2.3.3 Runtime behavior of the FCA Algorithm

We test the runtime behavior of our method on data sets containing independent spike data without injected STPs. The presence of a few STPs would not noticeably affect the overall computational effort. Our goal is to determine whether our current implementation, which also uses the SciPy toolbox Jones et al. (2001), is fast enough to enable the analysis of typical experimental data sets from multi-channel recordings. Since the total runtime of FCA would in general increase with the context fill ratio, we generate the data sets with a fairly high baseline firing rate of 15 Hz. We simulate 50 neurons in parallel, a number comparable to the size of modern multi-electrode recordings. We would like to be able to analyze up to 30 repeated trials of the same experiment, in chunks of 500 ms. Hence, the average number of spikes in the data is 11250. Note, however, that this number is likely to grow as experimental techniques advance.

For the profiling results shown in Figure 2.5, we use a cluster with nodes consisting of 2$\times$ Intel Xeon E5-2680v3 processors with 2.5 GHz processing speed and 8×16 GB DDR4 RAM, and report the time taken for context construction (circles), FCA (stars) and stability (diamonds) as a function of the total number of spikes in the data set. The number of spikes is the actual number taken from the data set used for each run. Most of the time is consumed by the computation of stability, consistent with the scaling analysis presented by Roth et al. (2008): exact stability evaluation time grows quadratically with the number of concepts. The curves in Figure 2.5 are fits performed with a linear function for context construction, a quadratic function for FCA and thus a quartic fit for the stability. Exact stability evaluation is feasible within a day up to about 4000 spikes, corresponding to 12 experimental trials with the above-mentioned features (number of neurons, trial duration, average neuronal firing rates). Extrapolation of the runtime to 30 trials based on the quartic fit yields ≈ 45 days.

2.4 Discussion

Information in the cerebral cortex has been hypothesized to be encoded and processed in terms of spatio-temporal patterns (STPs) of spikes generated from different neurons. Under this hypothesis, an STP is assumed to be the signature of active cell assemblies through which temporal sequences of spikes propagate at millisecond precision. Various network models, such as the synfire chain Abeles (1991) and synfire braid Bienenstock (1995); Izhikevich (2006) exist, which process information by propagation of STPs. These

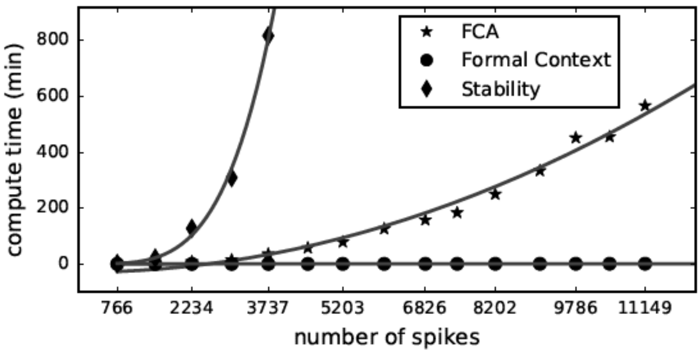

Figure 2.5: **Profiling results on synthetic data: nine data sets with different number of spikes were evaluated:** The compute time to calculate the FCA (asterix), formal contexts (filled circle), stability (diamond) and the corresponding fitted curves on each data set are shown. For details, see text.

models are compatible with known features of biological networks, in particular in terms of neuronal connectivities. Nevertheless, they remain of a speculative nature because their existence is hard to prove. Earlier work provided evidence for the occurrence of STPs in data from small numbers of simultaneously recorded neurons Nadasdy et al. (1999); Prut et al. (1998). However, these methods do not scale to massively parallel spike trains as recorded nowadays on a regular basis Riehle et al. (2013); Schwarz et al. (2014). We started to exploit a new approach based on formal concept analysis (FCA, Ganter and Wille (1999)) to mine parallel spike train data for STPs involving any possible subgroup of neurons of an observed population. We use FCA for an extensive search for potential STPs, which are expressed as formal concepts. To construct the formal context a window of pre-defined duration is attached to each time step. Thus, potential STPs longer than the analysis window are not detected. Only concepts that repeat at least three times are further considered. STPs found in parallel spike trains may either occur by chance or be generated by an underlying network process, which may reoccur and thus cause repeated occurrences of the patterns. To disentangle chance from real patterns we used the stability measure introduced in Kuznetsov (2007) and explored which threshold on the stability would serve this purpose.

We tested the approach based on ground-truth data which we generated by stochastic simulations of 50 parallel spike trains. We used independent spike trains to explore the detection rate of mere chance patterns, i.e. false positives (FPs). We found that the FP rate is generally low, but increases with the neuronal firing rate (varied from 5 to 15 Hz). Most FP patterns have either low intent size (= 2) or low extent size (= 2). The stability distribution of these FPs revealed that chance patterns with extent and intent size both larger than 2 always had stability lower than 0.6. In the light of these considerations, we minimized the risk of FPs by adopting a criteria which classifies as non-chance patterns only the STPs which occur at least 3 times and are composed of at least 3 spikes, as already done for instance in Prut et al. (1998), and additionally have stability 0.6 or larger. We then further generated data that contained STPs injected into independent background activity to investigate the amount of FPs that occurred in STP data as a result of the overlap of the injected patterns with background spikes, as well as the amount of false negatives (FNs). The rates for the background activity in these data varied again from 5 to 15 Hz, while the STPs involved 3 to 10 neurons and were injected 3 to 10 times in the data, at random time points. It turned out that FPs of this type were rare (usually less than 3 out of 100 simulations) and were always superpatterns of the injected STP. No FPs disjoint from the injected STP, i.e. composed of background spikes only, occurred. Instead, the FN rate decreased abruptly for a rate-dependent specific number of occurrences. The reason is that with more occurrences the stability of the pattern increases. The number of occurrences representing the border between high and low FN levels did not depend on the size z of the injected STP. We could draw analogous conclusions from the evaluation of the precision and recall of the method. Both quantities were low (< 0.8) for a low number of pattern occurrences, but drastically increased for data sets containing patterns with 5 or more injections. Besides, both were almost independent of the pattern size z, which is due to the behavior of the FNs mentioned above. We experimented also with larger values for the stability threshold than 0.6 (up to a threshold of 0.8). Higher thresholds yielded a lower amount of FPs at the expense of higher FN levels. In the light of this trade-off, 0.6 seemed to be a suitable threshold for stability in our settings.

For potential applications of the method to experimental data, a suitable value for the stability threshold remains to be determined. Several Monte Carlo techniques exist that use independent surrogates of the original data in order to derive the statistical significance of correlations present in the original data Grün (2009). The independent surrogates are generated from the original data by intentionally destroying the precise timing of spikes (and thus STPs as well) while preserving other features of the data (e.g. firing rates or spiking regularity) as much as possible Louis et al. (2010c). Using such surrogate data could be an effective way to determine typical stability values of chance patterns and thus to set a suitable stability threshold. This approach however remains to be investigated. Additionally, we plan to investigate the relation between the approach suggested here to detect STPs and methods we introduced e.g. in Grün et al. (2008); Torre et al. (2013) to detect patterns of synchronous spikes on the basis of their statistical significance. Both the stability value proposed here and statistical p-values used in other studies are meant as measures to distinguish chance and non-chance STPs. Suitable thresholds for these values must account for typical features of spiking activity, such as regularity of inter-spike intervals, temporal modulation of neuronal firing rates, rate heterogeneity across neurons, and so on.

Finally, another aspect needs to be solved before our approach becomes applicable to data of larger size than considered here (e.g. larger number of neurons, higher firing rates, longer recording time). Namely, our runtime analysis revealed that context construction and FCA are reasonably fast, but the time taken to compute the pattern stability scales as a quartic function of the number of spikes. Various steps could be undertaken to improve the computational performance of the method: a more efficient implementation of the FCA algorithm we used Lindig (2000), the use of a faster algorithm Kuznetsov and Obiedkov (2002); Andrews (2009), a parallel FCA algorithm Krajca and Vychodil (2009), or an approximate rather than exact evaluation of pattern stability Babin and Kuznetsov (2012).

Chapter 3

Extending SPADE to Spatio-Temporal Spike Patterns

3.1 Introduction

In this chapter, we provide an extension of SPADE (Torre et al., 2013) to spatio-temporal patterns. In chapter 2, we addressed this problem by a new approach based on Formal Concept Analysis (FCA; see Ganter and Wille, 1999) to efficiently find STPs in MPST data, count their occurrences, and evaluate their stability (see Kuznetsov, 2007) as a measure for non-randomness. We successfully applied the method to artificial test datasets of moderate size, e.g., 50 spike trains at 15 Hz for 1 s. The analysis was computationally too expensive to be applied on larger data sets as often realtistically encountered in neurophysiological experiments (e.g. Brochier et al., 2018).

Here, we improve the methodology introduced in the previous chapter in three respects. First, we use frequent item set mining on a suitably restructured format of the data as an equivalent but computationally more efficient alternative to currently available FCA algorithms. This shift of paradigm makes the method equivalent to SPADE from a procedural point of view. Second, we approximate exact stability with the Monte-Carlo approach suggested by Babin and Kuznetsov (2012), which reduces the cost of stability computation (previously the runtime bottleneck) by several orders of magnitude. Thus, one may compute different types of pattern stability and develop different criteria to filter patterns on the basis of these types improving further the performance of pattern detection. Third, we extend to STPs the evaluation of pattern significance originally introduced in Torre et al. (2013) and compare it with the approach based on pattern stability.

Section 3.2 presents the various steps of our novel method and links them to previous methods, mainly to the work we presented in chapter 2 and in Torre et al. (2013). Section 3.3 compares the performance of the stability-based (see section 2.2.1) and significance-based (SPADE) approaches for patterns filtering, and provides selection criteria for candidate patterns. We demonstrate the efficacy of the extended SPADE method in detecting STPs, while largely avoiding false positive detections in simulated MPST with different features typical for electrophysiological data, in particular different forms of non-stationarity such as firing rates varying over time and across neurons. Finally, section 3.4 discusses the advantages of SPADE over existing techniques for the analysis of correlations in MPST and proposes future studies.

3.2 Methods

The problem we are concerned with in this work is the extraction of spatio-temporal spike patterns in massively parallel spike trains and the classification of these STPs into those that occur reliably and those which do not, i.e. non-chance vs chance events. Here we first review a state-of-the-art method based on Formal Concept Analysis (FCA) which we introduced in chapter 1 to address this problem, and then we improve this method in various respects. The following sections describe the three main steps of the novel method, namely pattern extraction (section3.2.1), identification of reliable patterns by means of various stability measures (section 3.2.2), and statistical assessment of pattern significance (section 3.2.3).

3.2.1 Extracting non trivial patterns from large-size data

In our setting, an STP is defined as a pattern of spikes, emitted from a given collection of neurons, that have the same temporal relationship with each other across different occurrences. In chapter 1 we introduced how Formal Concept Analysis (FCA) can be used for extracting closed frequent patterns from large data sets, adapting the methodology to parallel spike train data, here we relate the Frequent Item set Mining (FIM) to FCA.

FP-growth as an equivalent alternative to fast-FCA

In chapter 1, we employed our pure Python implementation of the fast-FCA algorithm (Lindig, 2000). This algorithm creates the concepts in an order which simplifies the subsequent exact evaluation of stability used for isolating stable patterns from noise (see below). Unfortunately, the runtime of the computation of exact stability (see Roth et al. 2008) scales roughly with the fourth power of the number of spikes, as illustrated in Figure 3.1. This leads to a slow computation preventing the application to data sets of several tens of neurons. In chapter 1 we extrapolated the computation time to > 60 days on a data set of 15 s duration composed of simultaneous recordings of 100 neurons. However, when one does not compute stability, or computes it approximately rather than exactly (see below), the concept order is not needed, as explained in Babin and Kuznetsov (2012). Modern FCA algorithms exist that only compute the concepts and are, therefore, considerably faster, such as In-Close (Andrews, 2009) which is currently the fastest one to our knowledge. Unfortunately, at the time of writing, the state-of-the-art C implementation of In-Close exited because of a memory overflow when we input our data. The C implementation solving this problem will be provided soon (Kodoga, personal communication).

Instead, we explored another option for implementing a faster search for concepts. We exploited a known correspondence between FCA and (closed) frequent item set analysis (Zaki and Ogihara, 1998; Pisková and Horváth, 2013): formal objects can be mapped onto transactions, formal attributes onto items, intents onto closed itemsets and extent sizes onto supports, see also table 3.1. This allows us to compute the concepts based on the FP-growth algorithm known in the data mining community (Han et al., 2004). FP-growth is a frequent item set mining algorithm widely used to mine closed frequent item sets in large data sets. Specifically, we use a C implementation of FP-growth (Borgelt, 2012) to mine closed frequent patterns. Picado-Muiño et al. (2013) already used the algorithm to mine patterns of synchronous spikes in MPST data. The re-formatting of the data used here ('attribute scaling' in the terminology of FCA, see (Ganter and Wille, 1999)) allows us to extend the application of the FP-growth algorithm to the search for more general spike patterns, i.e. STPs.

Closed vs non-chance patterns

By use of fast-FCA/FP-growth algorithms, formal concepts / closed frequent item sets can be efficiently collected. Closed patterns can be understood as patterns, which are not trivial subsets of other patterns in the data, and which, therefore, may convey information not stored in any superset of them. However, not all closed patterns are necessarily of interest. Indeed, virtually any data set of simultaneous point processes contains closed patterns, even when the processes are completely independent of each other. Thus, many (usually the large majority) of closed patterns are merely chance events. A critical task that remains to be solved is, therefore, that of identifying non-chance patterns among the multitude of closed ones.

Several approaches can be taken to draw this assessment, depending on the features of the patterns used to discriminate interesting from non-interesting STPs. Simple pattern filtering criteria are often based on pattern size (intent size) and pattern occurrence count (extent size), see e.g. (Prut et al., 1998). These criteria are motivated by the following observations. First, the larger a pattern is the less likely it will occur by chance if spikes are independent events (Grün et al., 2002b). Likewise, under this independence assumption, the probability that a chance pattern has a given occurrence count decreases with increasing count (Grün et al., 2008). We found that rejecting all patterns with less than three spikes or less than three occurrences massively reduces the false positive pattern detection in our data. Another classical approach in the FCA community is to evaluate the (intensional) stability of a pattern (Kuznetsov, 2007), which loosely speaking can be understood as the tendency of a pattern to be an intent among any subset of windows where the pattern occurred. Stable patterns are of interest because they are unlikely to be produced by independent processes or neurons. Another approach consists in evaluating the statistical significance of the patterns found and in retaining only those patterns, which are not to be expected in the data given other statistics, such as the firing rates of the individual neurons (Grün et al., 2002a). This approach is common when testing the two alternative hypotheses of temporal coding (based on millisecond precise coordination of spikes among cell assemblies) versus rate coding (based on temporally less precise spike coordination and characterized by the rate profiles of the individual neurons). The next two sections are dedicated to reviewing the approach based on stability computation and the other based on evaluation of statistical significance and to integrate them into the analysis framework derived so far.

3.2.2 Filtering patterns by stability

As shown in chapter 1 conceptual stability (Kuznetsov, 2007) is a potential tool for separating chance patterns from non-chance STPs. In this section, we introduce the definition of extensional conceptual stability, in respect to the intensional stability used in chapter 1, and we illustrate advantages and computational issues thereof. We also show how recently developed efficient Monte-Carlo techniques can be used to approximate stability and, thus, make it applicable to large-size data.

Entity	Description
Spatio-temporal pattern (STP)	Precise temporal sequence of spikes repeated more often than expected under the hypothesis of independent firing
Item set / Attribute set	Set of spikes that form an STP
Pattern size / Attribute set size	Number of spikes forming an STP
Occurrence times / Object set	Set of time points at which an STP repeatedly occurs
Support / Object set size	Number of occurrences of an STP
Closed item set / Intent of formal concept	STP which is maximal in time and space, i.e. no larger set of time windows contains the STP, and no spike could be added to the STP without having to give up at least one occurrence time window. Formally defined as the pair of STP intent and STP extent. Extracted from parallel spike train data by the FP-growth/fast-FCA algorithm.
Support of closed item set / Extent size	Number of occurrences of a STP which is maximal in time and space
Closed frequent item set / Intent of frequent formal concept	STP which is maximal in time and space, and occurs at least a given number of times.
Signature (z, c)	Pair of parameters (z =pattern size, c =support), that characterize each concept and that are tested for significance with PSF
Stability	Measure that quantifies how reliably a pattern repeats identically across all its repetitions
P-value spectrum	Matrix whose entries z, c contain the p-values of pattern signatures (z, c), evaluated by PSF

Table 3.1: **Summary of terms often used in this chapter.** When two alternative terms appear in the left column, the first one is from frequent item set mining terminology and the second one from formal concept analysis terminology.

Intensional stability

Given a formal context $\mathbb{C}$ and the set $\mathfrak{B}(\mathbb{C})$ of all of its concepts, the intensional stability of a concept $(A, B) \in \mathfrak{B}(\mathbb{C})$, as already introduced in chapter 1, is defined

$$\sigma(A, B) = \frac{|\{C \subseteq A | C' = B\}|}{2^{|A|}} \tag{3.1}$$

where C' is the set of all attributes shared by the objects in C (see also Chapter 1). In words, the stability of an intent is the fraction of subsets of the extent (set of the indices of the starting points of time windows), which share exactly the attributes of the intent (pattern). This can be viewed as a kind of cross-validation (Kuznetsov, 2007): a pattern has a high stability index if it is found as a concept intent in many time windows. In the previous chapter, we computed the stability with the exact algorithm of Roth et al. (2008) and kept only those concepts whose stability exceeded a threshold. However, we found that this exact algorithm is too slow for application to large data sets due to its quadratic runtime scaling in the number of concepts.

Approximation of stability

Hence, we employ a fast Monte-Carlo approximation of the stability suggested by Babin and Kuznetsov (2012). Instead of iterating through all subsets of an extent in the numerator of the right hand side of 3.1 and checking whether the attributes shared by a given subset equal the intent, one performs this check only on a fixed number Z of randomly drawn extent subsets. The denominator is then replaced by Z.

Extensional stability

By definition, intensional stability only accounts for the occurrence count of a pattern and not for its size. Therefore, its value is unaffected (see chapter 1) by the number of spikes forming the pattern. This behavior is evident in the statistical evaluation results shown below. This feature is independent of the approach (exact or approximated) used to compute the stability. However, pattern size should play a role in determining whether a pattern is to be retained as a true pattern or rejected as a chance event. Indeed, more independent events (spikes) are less likely to re-occur in a specific temporal sequence by chance than fewer events. To account for this fact, we introduce here a filtering rule based on extensional stability, which accounts for the pattern composition (size) rather than the pattern occurrence count. Formally, extensional stability of a concept (A, B) is defined by exchanging extent and intent on the right hand side of 3.1:

$$\sigma(A, B) = \frac{|\{C \subseteq B | C' = A\}|}{2^{|B|}}.$$

Extensional stability can be calculated - as intensional stability - either in exact form or by approximation. A new filtering criterion can be devised for closed patterns based on extensional stability by retaining only those patterns whose extensional stability exceeds some pre-defined threshold.

Choice of stability threshold

An issue that remains to be addressed is how to set the stability threshold(s) used to distinguish STPs from chance patterns. In chapter 1, we set the threshold for intensional stability to 0.6 following an heuristic approach, as this choice turned out to provide a good balance between FPs and FNs on a broad range of simulated data with various parameters. Real data, however, may need different thresholds depending on their size (number of neurons and/or duration), the firing statistics and other features of the spike trains. Because we are interested in using stability as a measure to determine which patterns are more stable than one would expect a chance pattern to be, an appropriate threshold should be such that the stability of patterns found in independent data, i.e. of chance events, would not cross the threshold. We, therefore, propose here to estimate the appropriate stability threshold from independent surrogates of the original data via the following Monte-Carlo approach.

First, surrogates of the original data that contain only chance patterns need to be generated in such a way that other features of the data characterizing the null hypothesis of independence (importantly, the firing rate profiles) are preserved. A variety of techniques exists to this end (see Grün, 2009; Louis et al., 2010b). Among them, we opt for spike dithering, which moves each spike by a random amount (up to a few ms) around its original position. STPs occurring above chance level, if existing are, thus, destroyed, while firing rates - which are defined on a larger time scale - are almost unaffected. Second, we extract patterns from the surrogate data by use of FP-growth, compute their stability, and thereby derive the distribution of pattern stability under the null hypothesis. The stability threshold is finally set to a chosen upper quantile of the null distribution. In our settings, a single surrogate data set contains always several thousands of chance patterns and is, therefore, sufficient to obtain close estimates of small quantiles of the null distribution. We separately derive the thresholds θ_{int} and θ_{ext} for intensional and extensional stability, respectively.

3.2.3 Filtering patterns by statistical significance

An alternative to stability-based filtering to identify non-chance patterns among the closed frequent patterns extracted by FP-growth is to test the statistical significance of STPs directly. The null hypothesis of the test here is that the spike trains are mutually independent and no patterns exist in the data except for chance ones. The alternative hypothesis states that some patterns indeed occur too many times to be considered as chance events.

Testing the statistical significance of all closed frequent patterns one by one is not an option in applications to large-size data such as MPST data from tens or hundreds of neurons recorded simultaneously. Indeed, the immense amount of occurring patterns and, therefore, of tests to be performed raises severe multiple testing issues. We addressed this problem in the context of testing for synchronous spike patterns in Torre et al. (2013), where we developed an alternative statistical approach, here named SPADE (Synchronous PAttern Detection and Evaluation), that allows us to avoid such massive multiple testing. In that publication, we employed FP-growth to extract synchronous patterns, as we have done here for the more general case of spatio-temporal spike patterns. Thus, we aim to employ the statistical framework of SPADE to test for STPs. In the following we summarize the various steps of the SPADE analysis to assess pattern significance.

Pattern Spectrum Filtering

The first component of SPADE for assessing the significance of closed frequent patterns found by FP-growth is Pattern Spectrum Filtering (PSF). Instead of testing individually each of the thousands of closed frequent patterns, statistical significance is assessed for patterns of same size z and same number of occurrences c, i.e. for each pattern signature $(z,\, c)$. The probability of having a pattern with signature $(z,\, c)$ under the null hypothesis $\mathcal{H}_0$ of independence is evaluated via a Monte-Carlo technique on surrogate data which are generated from the original data by dithering. By repeated generation of surrogates and counting closed frequent item sets we implement the null-hypothesis of independence. SPADE then determines the p-value of each signature $(z,\, c)$ as the fraction of surrogates that contain closed frequent item sets with that signature, based on a large total number K of surrogates. Here, differently from what was necessary for the choice of the stability threshold, a large number of surrogates is needed, because each one contributes with only one instance (pattern of a certain signature $(z,\, c)$ present or not) to the Monte Carlo sampling. Since multiple tests are performed (one per signature found in the original data), we correct the significance level α using the false discovery rate (FDR) correction (Benjamini and Hochberg, 1995). All patterns mined in the original data with signatures $(z,\, c)$ that have a p-value smaller than the FDR corrected threshold are classified as statistically significant.

Pattern set reduction

In Torre et al. (2013) we showed that the presence of repeated occurrences of a real pattern A tends to increase the significance of patterns resulting from the chance overlap of pattern A with background activity. In other words, PSF correctly rejects FPs entirely composed by chance patterns, but, in the presence of a real pattern, it tends to overestimate the significance of patterns resulting from chance overlap with background spikes. The reason is that the size z and/or the occurrence count c of these patterns are indeed not entirely due to chance, but are boosted by the presence of the real pattern beyond the chance level that PSF determined under $\mathcal{H}_0$.

Pattern Set Reduction (PSR), the last step of the SPADE analysis, aims at removing these FPs by testing the patterns filtered with PSF reciprocally for conditional significance. When testing for a pattern A given a sub-pattern B of A (such that $z_A > z_B$ and $c_A < c_B$), PSR re-assesses the significance of A through the p-value of the signature $(z_{A|B} = z_A - z_B + h,\, c_A)$ already stored in the p-value spectrum previously computed by PSF. $z_{A|B}$ is a smaller value than z_A, penalized by the presence of B. Similarly, B is re-tested conditioning on the presence of A by replacing its occurrence count c_B with $c_{B|A} = c_B - c_A + k$. h and k are correction factors accounting for the fact that the p-values of $(z_{A|B},\, c_B)$ and $(z_A,\, c_{B|A})$ are taken from the original p-value spectrum, which is calculated over all time bins rather than over the time bins only where A and B occur. In our study we set $h = 0$, $k = 2$, which proved to be a good heuristical choice in the validation of SPADE (see also Torre et al., 2013). If only $(z_{A|B},\, c_A)$ is significant, the method retains A and discards B, and vice versa if only $(z_B, c_{B|A})$ is significant. If both $(z_{A|B},\, c_A)$ and $(z_{B|A},\, c_B)$ are significant, both patterns are kept. If neither signature is significant, in light of the fact that PSF returned both and, therefore, at least one of the two patterns should be a true positive, PSR retains the pattern covering the largest number of spikes, i.e. the patterns with the largest $z \times c$ score. For patterns A and B that only partially overlap ($A \cap B \neq \emptyset$,

$A \not\subset B$ and $B \not\subset A$) the conditional tests are performed over the conditioning pattern $A \cap B$.

3.3 Results

We presented above two different techniques to distinguish between chance patterns and selected STPs, based on stability measures and based on statistical significance of signatures (SPADE), respectively. Both of them take as input the concepts mined with FP-growth and return those that are statistically surprising because the assessed feature (stability or signature) is significantly larger for these patterns than for chance patterns. In this section we illustrate how each of the two methods performs, both in terms of computational time and of false positives and false negatives.

3.3.1 Computational efficiency

We first compare the computational efficiency of the components introdueced already in chapter 1 (fast-FCA, exact stability) to the proposed components introduced in the section above (FP-growth, approximate stability, PSF). Figure 3.1 shows the runtimes of these components applied to simulations of 50 parallel, mutually independent Poisson spike trains with a firing rate of 15 Hz each. The runtime of these various analyses components is evaluated on 10 data sets of different number of spikes, achieved by data sets of different duration, increasing from 1 to 10 s in steps of 1 s. The measured runtimes are marked by symbols, and their fitting curves are shown as solid lines in the same color. The profiling results were obtained on a compute cluster with 32 nodes, each consisting of a 2 $\times$ Intel Xeon E5 processor with 2.5 GHz processing speed and 8 $\times$ 16 GB DDR4 RAM.

In chapter 1, we made use of the fast-FCA algorithm introduced by Lindig (2000) after pre-processing the data as described in Chapter 1. The runtime behavior of the fast-FCA algorithm implemented in Python (same as used in chapter 1) and shown in Figure 3.1 (red) is fitted by a function which is quadratic in the number of spikes. Based on this function we extrapolate the runtime of larger data sets, in particular to the typical experimental data we aim to analyze, i.e. 100 parallel neurons with an average firing rate of 15 Hz of each neuron, recorded for 15 seconds and, thus, containing a total of 22500 spikes on average. Mining the concepts in a data set of this size with this implementation of fast-FCA would take about 68 days of compute time. FP-growth (brown) is significantly faster and exhibits a significantly slower and linear trend. For a data set of the same size the runtime is 4.5h instead. Thus, the speed up gained by using FP-growth instead of FCA enables the extraction of non-trivial patterns also from large-size data that were beyond the reach of our previous approach. Therefore, we decide to base our analysis on FP-growth.

The second step of our analysis is the computation of the stability (intensional and extensional stability) of all non-trivial patterns extracted by FP-growth, to filter out non trivial patterns (section 3.2.2). The stability can either be computed exactly or can be approximated by a Monte-Carlo approach (see section 3.2.2). We show here only the result for the intensional stability, since the runtime for the calculation of extensional stability is approximately the same (not shown here). The runtime necessary to derive the stability as described in section 3.2.2 is the sum of the time required to compute the stability on the original, empirical data set for each pattern and the time needed to compute its significance threshold. The latter requires the generation of a surrogate data set, the extraction of closed patterns, and the computation of their stability. Thus, the total runtime of our stability-based STP detection approach (see section 3.2.2) is twice the time needed for the calculation of the stability, plus twice the time consumed by FCA (see section 3.2.2). As shown in Figure 3.1, the computation of the exact stability (green) dominates the total runtime, increasing quartically with the number of spikes in the data. The approximate stability (aquamarine), in contrast, has a runtime which is several orders of magnitude smaller and shows a linear trend.

Overall, replacing FCA by FP-growth and replacing exact stability by approximate stability yields a compute time, which is about three orders of magnitude smaller and, thus, enables applications to data of unprecedented size.

The third and last step of the method that needs to be investigated in terms of runtime is the calculation of the statistical significance of the patterns by means of pattern spectrum filtering (PSF) and pattern set reduction (PSR). To derive the p-value for PSF for each signature, we need to generate at least 1000 surrogate data, each of which requires to be analyzed by FP-growth to extract closed frequent patterns in order to build up the statistics for each signature. Therefore, PSF is quite compute time intensive (Figure 3.1, blue - for a data set of 22500 spikes it would take about 1366 days) if processed in a serial way as shown here. Trivial parallelization (i.e. parts of the analysis can be run independent from each other to save compute time) of the analysis program can be applied to PSF, which absorbs the majority of the computational load of the method and reduces severely the computational time. To this end, the FIM analysis on different surrogates can be distributed over different computing cores and run in parallel. The independent results (closed frequent patterns of each surrogate data set) are finally collected to compute the p-value spectrum. The PSR runtime is negligible (not shown here) since it is linear and it is applied only to significant patterns which is typically a small number as compared to the mined concepts. It does not directly depend on the total number of spikes but on the actual significant patterns. For this reason we do not consider it as a computational component that might determine the computational feasibility of SPADE.

3.3.2 Stochastic models for validation

The increased computational performance achieved by combining FP-growth and approximate stability calculation enables the application to larger data sets than previously possible. We are, therefore, interested in generating ground truth artificial data with comparable size and properties of data typically obtained in electrophysiological recordings. To this end, we follow the same approach taken in chapter 1 and generate data consisting of a superposition of independent background activity and repeated STPs. The background activity is modeled by a set of $N = 100$ parallel independent Poisson processes, each having a firing rate r which may be stationary or variable over time, and identical or different across neurons, and lasts for a total period $T = 1\,\mathrm{s}$. An STP occurrence is modeled as a temporal sequence of z spikes from the first z neurons (without loss of generality) and with a constant time lag of 5 ms between successive spikes. Multiple occurrences of the STP are realized by injecting the sequence at multiple, random times within the simulation interval $[0, T]$.

We first consider stationary data with three different constant firing rates $r \in \{15\,\mathrm{Hz}, 20\,\mathrm{Hz}, 25\,\mathrm{Hz}\}$ for each of the 100 neurons. Then, we test the performance of the method for a variety

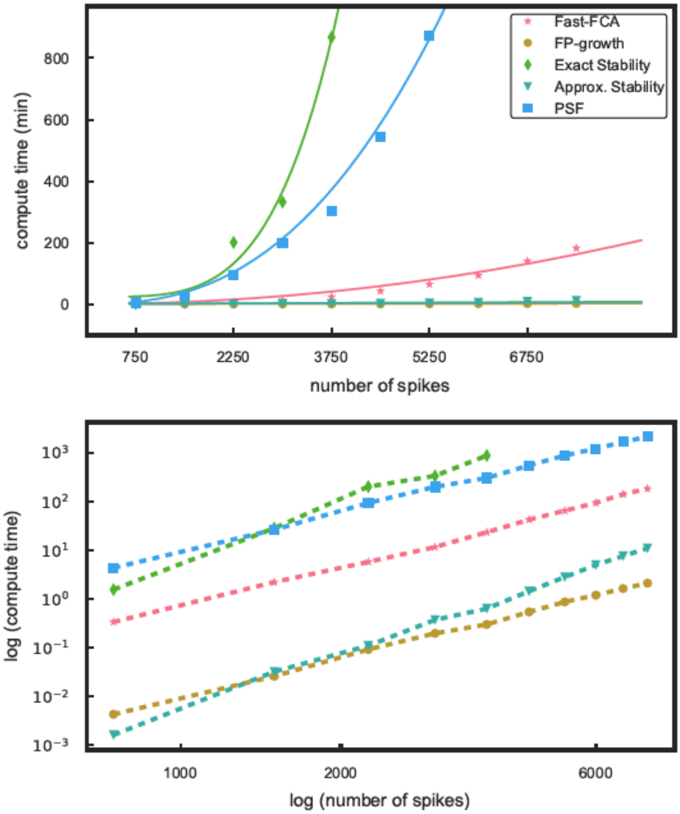

Figure 3.1: **Profiling results for different components of the methods.** In the top panel the runtimes as a function of the number of spikes in a data set are shown for pattern mining using fast-FCA (red, asterix) and FP-growth (brown, filled circles). We also compare the run times for the stability analyses, exact stability (green, diamonds) and approximate (aquamarine, triangles) and of PSF (blue, squares). The solid lines are fitted functions quantifying their characteristics. The bottom panel shows the same data in log-log scaling: the computational times follow approximately different power laws.

of non-stationary data sets that mimic typical statistical features of experimental data and have the tendency to generate FPs in correlation analyses. In particular, we analyze artificial data that include three different types of rate non-stationarity or variability (see Figure 3.2):

1. Non-stationary firing rates over time by means of a sudden rate jump, coherent across all neurons (Figure 3.2, top panel): all neurons have the same firing rate, equal to 10 Hz in the intervals $[0\,\mathrm{s}, 0.6\,\mathrm{s}]$ and $[0.7\,\mathrm{s}, 1\,\mathrm{s}]$, and 60 Hz in the interval $[0.6\,\mathrm{s}, 0.7\,\mathrm{s}]$;

2. Heterogeneity of the firing rates across neurons (Figure 3.2, middle panel): firing rates are stationary over time but different across neurons, and increase from 5 Hz (for the first neuron) to 25 Hz (for the last neuron) in steps of 0.2 Hz. The spike trains in which the patterns are later injected are randomly selected;

3. Short lasting, simultaneous, sequential rate jumps of subsets of neurons (Figure 3.2, bottom panel): the 100 neurons are grouped into 20 subsets of 5 neurons each. At two different time onsets (50 ms and 550 ms), the first group instantly changes its firing rate from a baseline level of 14 Hz to 100 Hz for an interval of 5 ms. When group i goes back to baseline level group $i+1$ experiences the same rate jump (5 ms later), $i = 1, \ldots, 19$.

Models 1 and 2 were already used in Torre et al. (2013) to explore the sensitivity of SPADE to rate variability. The third model was introduced in Torre et al. (2016a) to validate another method, called ASSET, designed for the analysis of sequences of synchronous spike events in massively parallel spike train data.

In total, we use 6 different models of background activity, three with different levels of stationary rates and three with variable rates across times or neurons. We then vary, for each of these models, the number z of neurons involved in an injected STP and the number c of its repetitions from 3 to 10 in steps of 1, for a total of $(number\ of\ models \times z \times c) = (6 \times 8 \times 8) = 384$ parameter combinations. For each choice, we determine the performance of our approaches in terms of the average number of false positives (FPs) and false negatives (FNs), defined below, obtained over 100 stochastic realizations of the respective background model, yielding a total of 38400 data sets to analyze.

3.3.3 False positives and false negatives

In pattern discovery, different definitions of false positives (FPs) and false negatives (FNs) are possible. The identification of the exact injected pattern is a clear example of correct identification (true positive, TP), while the identification of a pattern being completely disjoint from the injected pattern is a clear FP result. Similarly, the complete non-detection of an inserted pattern or subsets of it is an unambiguous FN outcome. Cases in between are less clear and need to be defined. For instance, the identification of a pattern whose spikes form a subset of the real pattern may be considered, depending on the portion of spikes of the found pattern relative to the true pattern, sufficient for a correct identification (TP). Here we adopt the most strict definition of a TP, i.e. classifying a found pattern as a TP if it consists of all and only the spikes forming each occurrence of the injected pattern. Otherwise, the pattern is detected as a FP and the absence of TPs yields an FN outcome.

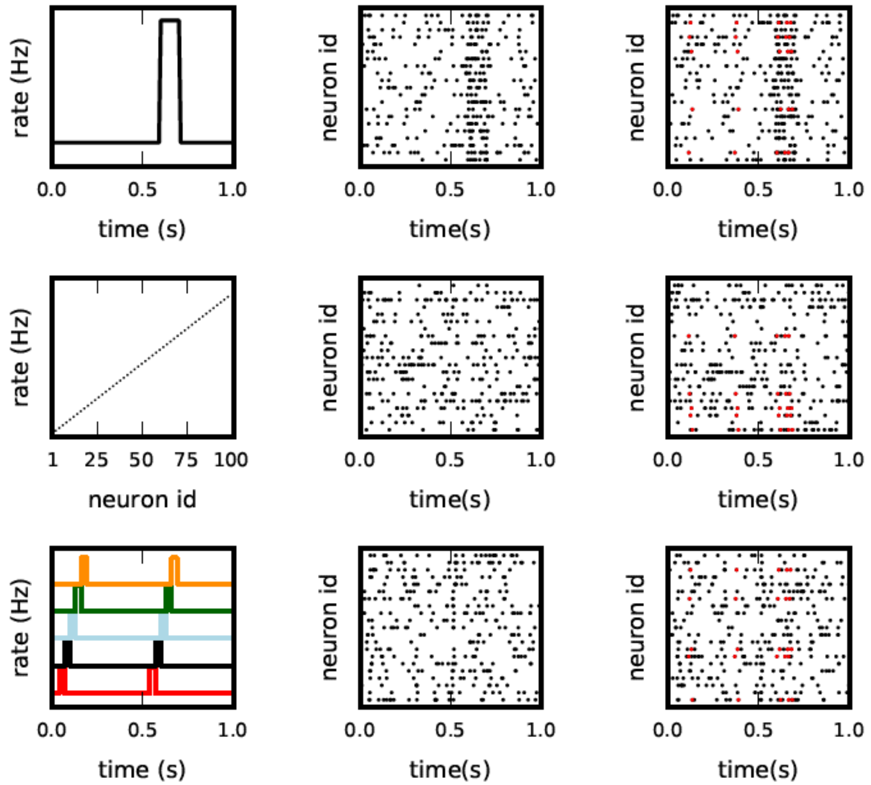

Figure 3.2: **Different background models for non-stationary and inhomogeneous data.** Each row illustrates one model of the test data: top: co-varying firing rates with a large rate step of all neurons; middle: inhomogeneous but stationary rate of each neuron; bottom: coherent short rate changes in subsets of neurons at consecutive time points. The columns show from left to right: the underlying rate profiles, an exemplifying raster plot of the spiking activity (one dot per spike), and a raster plot of the respective background activity enriched with $c = 5$ injections of an pattern of size $z = 5$ (spikes in red).

We compute two types of FPs. One is based on purely independent data, i.e. we only realize the respective background model without pattern injection. This provides us the FP level which is purely resulting from the stochasticity of the processes. The second type of FPs we are exploring are STPs that were detected but were not the injected STPs. This is relevant if we want to make sure that injected patterns are not forming new patterns with the background activity. The results for the latter ones are shown in figures below, the former ones will be just mentioned.

For evaluation of FNs as a function of the signature (z,c) of the injected pattern, we vary the parameters z, c between $3, \ldots, 10$ to get all combinations. For each signature we perform R realizations of a given background rate model and insert a pattern of size z and with c occurrences. We evaluate in how many of the R realizations we detected the injected pattern. The resulting FN rate, i.e. the fraction of realizations in which we did not detect the pattern divided by R, is entered in a matrix at the signature z (x-axis) and c (y-axis). By varying the signatures and performing this procedure again we fill the FN matrix.

For evaluation of the FP matrix, we use the same data as for the FN evaluation. For each signature, we count the number of realizations in which one or more patterns are detected as significant that are not identical to the pattern injected. The ratio of the realizations for which this occurred divided by the number of realizations R is entered at the signature of the injected pattern.

In the next sections, we test the performance of our approaches in terms of FPs and FNs on our artificial, simulated test data.

3.3.4 Performance of approximate stability

In order to quantify the error introduced by the approximated stability (section 3.2.2), we compute the exact and the approximate intensional stability for all patterns extracted by the mining technique (FP-growth) from synthetic data. We set the number Z of subsets used for the Monte-Carlo approximation of the stability to 500, while for the computation of the exact stability all possible subsets are used. The data, already used in chapter 1, comprised parallel spike trains from 50 neurons firing independently of each other at a constant of rate $r = 15\,\mathrm{Hz}$ each, for a total duration $T = 1\,\mathrm{s}$. In addition, we also generate data sets containing in addition an injected STP. The STP consists of $z = 8$ spikes from 8 different neurons, falling within a window of duration $w = 50\,\mathrm{ms}$. The STP is injected $c = 9$ times in the data, at random positions in the simulation period $[0, T]$. We define the approximation error as the absolute difference between the exact and the approximated stability values, both computed for each pattern extracted by FCA. The distribution of the errors greater than 0 is illustrated in Figure 3.3 (gray bars), with an average error of $1.888 * 10^{-3}$ and a maximum error of 0.14 (i.e. 14% of the max. stability value). However, no errors at all (black bar in Figure 3.3) occur for the majority (282510 out of 283451, i.e. 99.67%) of the patterns. The results indicate that approximating the intensional stability is a suitable alternative to the computationally unaffordable calculation of exact stability and, thus, allows one to apply approximate stability of data sets of more than 50 neurons.

Furthermore, we test whether or not the (small) error introduced by approximated stability does affect the performance of STP detection. To that end, we compare the results gained using exact stability and approximate stability to select significant concepts applied to the very same data. The data are also identical to the data analyzed in the previous

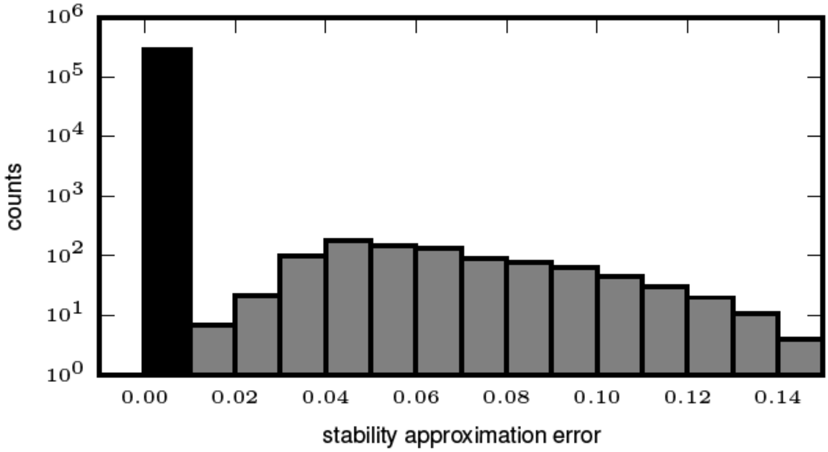

Figure 3.3: **Error of approximate calculation of intensional stability**. The histogram shows the number of patterns (in logarithmic scale) as a function of the absolute difference between the approximated and the exact intensional stability of one simulated data set. The leftmost black bar represents the number of patterns for which no difference occurs for the exact and the approximated stability which is the majority of the patterns (99.67%). The largest absolute errors (rightmost bar) are in the range of $0.13 - 0.14$ and occur for 4 out of 283451 total patterns. The average error (including also the error equal to 0 , black bar) is 2.189×10^{-4}.

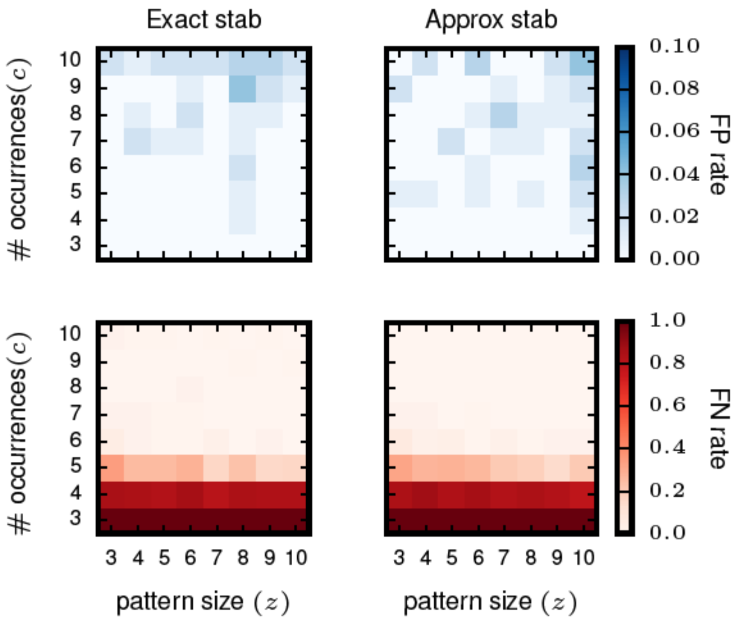

Figure 3.4: **Comparison of pattern selection based on exact and approximate stability.** The left column shows the FP matrix (top) and the FN matrix (bottom) for FCA analysis followed by exact stability filtering. The right column shows FP and FN after application of the approximate stability filter instead. For both approaches the threshold for the stability was > 0.6 for filtering the patterns. Each element of the matrices contains the FP or FN rate for a particular signature (z, c) of pattern size z (horizontal axis) and occurrence count c (vertical axis). The data analyzed are Poisson data ($N = 50$ neurons with stationary rate $r = 15$ Hz) with STPs injected with the respective signatures. $R = 100$ simulations are performed for FP / FN extraction.

chapter composed of 50 neurons, simulated for $1s$ with injected patterns with parameters z, c varying between 3 and 10, with 100 realizations for each parameter combination. Significantly stable concepts are detected if their stability crosses the threshold of $\theta_{\text{int}} = 0.6$, i.e. the same as used in chapter 1. Figure 3.4 shows the results in terms of number of realizations returning FPs (top) and FNs (bottom) out of 100 simulations for each signature (z, c). The left column shows the results using exact stability, the right column using the approximated stability. The performances of both methods, both in terms of FPs and FNs, are qualitatively identical (maximum of the absolute value difference of the two matrices smaller than 0.1 for the FP-rate as well smaller than 0.5 for the FN-rate and randomly distributed across the matrix entries), showing that the error introduced by the Monte-Carlo approximation of the stability is negligible.

3.3.5 Validation on artificial data

We assess and compare now the performance in terms of FPs and FNs of the two variants of our analysis method, i.e. one that filters patterns on the basis of their stability and the other based on significance evaluation. We use the test data described in section 3.3.2, i.e. data with a certain type of background activity (three different types with stationary rates, and three types with time-varying or inhomogeneous firing rates), and combined with injected STPs of a certain signature (z, c). For each data model, we generate $R = 100$ realizations (data sets), analyze each of them for the occurrence of STPs surviving the filtering process.

Stability based filter results for stationary data

We first examine the performance of pattern filtering based on intensional (or extensional) stability. After choosing one of the two measures, this approach classifies patterns found by FP-growth as stable (and, thus, retains them as reliably reoccurring patterns) if their stability exceeds a pre-determined threshold θ. As explained in section 3.2.2, we derive θ as a chosen quantile of the null distribution of stability values, obtained from independent data. We set the overall significance level to $\alpha = 0.01$ and set θ to the percentile corresponding to the Bonferroni corrected level $\alpha_{\text{corr}} = \frac{\alpha}{\text{total number of concepts tested}}$.

To obtain stable patterns by application of the stability evaluation we make use of surrogate data, i.e. independent data generated by dithering (see section 3.2.2) from the original data, to derive the null distribution and, thus, the stability threshold θ. For our extensive validation of data containing injected STPs we would have to derive the stability threshold for each of the total $100 \times 6 \times 8 \times 8 = 38400$ (see 3.3.2) data sets. To avoid such massive computations, we make use of the assumption that the few additional spikes injected by insertion of STPs do not change the null distribution of the stability values under the hypothesis of independence. Therefore, to evaluate FPs and FNs across all these scenarios, we derive a single stability threshold θ for all models with the same background rate as follows: we generate 100 data sets with independent background activity according to the rate model, and derive θ as the 95% quantile of the empirical distribution of pattern stability values in this case where no patterns were inserted. This threshold is then used for the assessment of FPs and FNs in all 64 models with the same background rate but containing STPs of different size and occurrence count. This approach, was already used in (Torre et al., 2013) and was shown to be appropriate in chapter 1, since the distribution

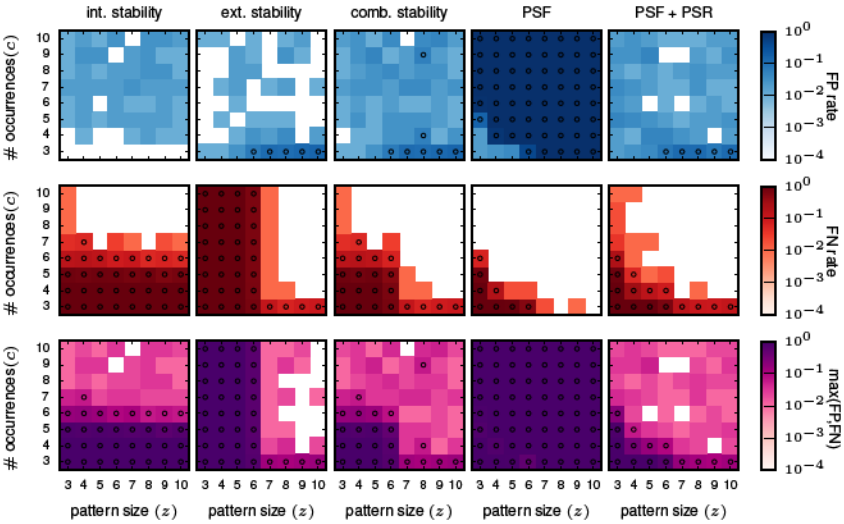

Figure 3.5: **Performance in terms of FPs and FNs for stationary data.** The results shown here are all for the data model in which all neurons have a stationary firing rate of 25 Hz. All data sets consist of 100 neurons simulated for 1 s. The parameters used for FP-growth are a bin size of 1 ms and a window length of 50 ms. We show the results for all types of injected patterns entered at their respective signature (64 signatures, all possible combinations for the size z (x-axis) and the number of occurrences c (y-axis)). Size and number of occurrences are varied between 3 and 10. For each pattern signature we perform and analyze 100 realizations. Each matrix element (signature) shows the fraction of realizations for which the filtered results contain one or more FPs (top row) / FNs (middle row). The bottom row shows the maximum rate of either the FPs or the FNs. *First column:* Results of the intensional stability filter, using a significance level of $\alpha = 0.01$ and Bonferroni corrected, yielding a stability threshold of $\theta_{int} \approx 0.55$. *Second column:* Results of the extensional stability filter and stability threshold of $\theta_{ext} \approx 0.8$ resulting from the same significance level as for the intensional stability. *Third column:* Results of the combined stability filter, where the Bonferroni correction was adjusted by a factor 2 to account for each concept being tested twice (once for the extensional and once for the intensional stability). *Fourth column:* Results of PSF. Significance level $\alpha = 0.01$, corrected with the FDR criterion. *Fifth column:* Results of PSF+PSR. Significance level $\alpha = 0.01$, corrected with the FDR criterion. The PSR parameters are set to $h = 0$, $k = 2$.

of θ is not affected by the insertion of a few STP spikes. In addition, this approach makes our validation considerably (64 times) faster.

Performance of intensional and extensional stability

We first analyze stationary data with constant firing rates, $r \in \{15\,\mathrm{Hz}, 20\,\mathrm{Hz}, 25\,\mathrm{Hz}\}$ containing injected STPs. We filter the patterns found by FP-growth on the basis of either their intensional or their extensional stability, and calculate the corresponding significance thresholds θ_{int} and θ_{ext} as explained above. In the following, only the results for $r = 25\,\mathrm{Hz}$ are shown, as those for $r = 15\,\mathrm{Hz}$ and $r = 20\,\mathrm{Hz}$ are comparable and lead to identical conclusions. The FPs on purely independent data sets (only background activity) as defined in section 3.3.3 has a FP rate of 0.01, i.e. in only 1% of the total realizations of the independent data FPs are detected (not shown).

Then we evaluate the performances for data with injected patterns. In the columns on the left of Figure 3.5, we show the FP and FN matrix for the analysis using intensional and extensional stability, respectively. Each entry in the matrices corresponds to one signature (z, c) of the injected pattern, and the color-coded value represents the FP (top panels) or FN (middle panels) rate. The FP rates for both, the intensional (left column) and the extensional stability (second column from left), are low, but somewhat higher (some times at 0.05, marked by circles in Figure 3.5) than for FPs in purely independent data.

The FN rates (Figure 3.5, middle row) are large (often close to maximum) for about half of the matrix elements for both stability measures but differ in respect to the signatures at which they occur. For intensional stability, the FNs are large for the whole range of z , but only for small c (about 3 - 6), and decrease abruptly for larger c. In contrast, when filtering with extensional stability, the FNs are high for all c, but only for small z (about 3-6) and decrease for larger z. These results are not unexpected since, as explained in section 3.2.2, intensional stability is almost exclusively affected by the number of occurrences of a pattern, while extensional stability is emphasizing the number of spikes forming a pattern. Nevertheless, these are undesired results, since in independent data chance patterns decay with their number of occurrence and the pattern size (see e.g. Torre et al., 2013)) and, thus, we expect that the border of selected patterns should also decay as a function of the combination of both parameters.

Combined stability

Aiming at a method whose FNs decay with the size z and the occurrence count c of the patterns, we combine the filtering criteria based on intensional and extensional stability. This approach keeps all concepts whose intensional or extensional stability value is larger than the respective thresholds. This procedure applied to independent data leads to a maximum FP rate equal to 0.02 (not shown), and can be explained by the application of two tests on the data. The results for the data sets containing the STPs are shown in the third column from left in Figure 3.5. The FP rate is again smaller than 0.05 for most of the entries, and higher than 0.05 for entries in the right bottom corner of the matrix, which result from data sets that contain patterns of large size occurring only 3 times and easily combine with background spikes and, therefore, become significant. The FNs instead decrease gradually, both as a function of pattern size z and of the pattern count c. Thus, the combined filter retains large patterns due to their high extensional stability, and patterns with a large occurrence count due their high intensional stability.

Significance based filter results for stationary data

For further comparison we analyze the same data as above using FP-growth followed by PSF and PSR. We set the significance level for the PSF to $\alpha = 0.01$, corrected by the number of different pattern signatures in the data using FDR correction. Note that when setting the threshold for extensional and intensional stability we use the more conservative Bonferroni correction instead, because FDR did not provide an adequate compensation (i.e., it leads to a large number of FPs, not shown here).

The FP rate of independent data is as for the stability filtering smaller or equal to 0.01 (not shown). In the independent data, PSF alone suffices to achieve this performance. PSR is not necessary, because the probability that completely chance patterns exceed the PSF significance threshold and overlap is close to 0. In contrast, PSR is a critical step for data containing non-chance patterns, where it is designed to remove the false positives found by PSF due to the overlap of the true patterns with the background activity. Without correcting for overlapping patterns the results show very high FP rates for any combination of size and number of occurrences of the patterns (Figure 3.5, fourth column, top). Using PSR the FP rates for data with injected patterns (Figure 3.5, right column, top) are similarly low as for the other approaches, i.e. at the level of 0.05. Some FPs have a FP rate larger than 0.05, and occur for data containing injected patterns with large size and small number of occurrences ($c = 3$) . As for the other methods, these FPs are due to combinations of injected patterns and background activity. The FNs decay as a function of z and c, but much faster than the combined stability approach (Figure 3.5, right column, middle). The PSR moderately increases the number of FNs for patterns with few neurons occurring often or composed by many spikes and occurring few times. This is due to the fact that these are the two conditions in which it is more likely to have one or more spikes of the noise background that overlap by chance with the injected pattern forming a larger, more significant pattern. Such chance overlap may cause the rejection of the injected pattern in the PSR.

Counting the number of STPs obtained after each step of the of the method clarifies the impact of that step on the overall performance. While these numbers change in magnitude depending on the parameters (z, c) of the injected pattern, their proportion between different steps of the method was very similar across different values of (z, c). We can, therefore, illustrate the results for one specific configuration ($z = 10, c = 10$). Each step is maximally effective if the number of STPs it keeps as non-chance patterns is 1 and if this pattern is the single injected pattern. For $z = c = 10$, we obtained on average: a) 1089.92 STPs after FIM (which retains all frequent closed patterns), b) 2 STPs after either the combined or the intensional-only filtering, and 1 STP after extensional-only filtering, c) 24.41 STPs after PSF, and d) 1.02 STPs after PSF+PSR (almost exclusively the injected pattern).

In conclusion, based on the validation of stationary data, the approach based on PSF and PSR, i.e. the SPADE method extended to spatio-temporal patterns, performs best and has the highest detection power as compared to the methods based on stability analysis. Even the combined stability analysis that uses the two stability measures independently has a smaller range of signatures with low FNs and improves only if at least one of the patterns' parameters z or c is large enough. In contrast, SPADE is also sensitive to the total number of spikes ($z \times c$) in the patterns. Thus, both approaches (stability or significance based) show the desired feature of a) small number of FPs, b) decreasing number of FNs for increasing (z, c). However, SPADE produces a smaller total number of FNs for comparable FP rate and, thus, has the best performance. We finally note that the patterns found by extensional stability filtering were almost always found by PSF+PSR too (which is slightly less conservative, and which was less prone to false negatives). Thus, combinations of these two selection criteria do not seem a valid option here. For instance, retaining all patterns found by any of the two criteria would be often identical to accepting the results from PSR+PSR. Retaining only patterns found by both would most of the times be equivalent to accepting the results from extensional stability. Both options would additionally sum the computational costs of the two methodologies.

Performance of SPADE in the presence of multiple STPs

The experiments illustrated so far were performed on data containing a single true STP, which the method was able to find with high reliability. Real data, however, are likely to contain STPs from more than one group of neurons. Experimental studies (e.g. Torre et al., 2016b; Riehle et al., 1997b) revealed, for example, an abundance of synchronous spike patterns arising during task execution. Torre et al. (2016b) used the original version of SPADE, demonstrating its ability to retrieve multiple synchronous patterns, when present. To demonstrate that our extended method can achieve the same for STPs, we investigated an additional scenario with data containing two different types of injected STPs. Both STPs had size $z = 5$, were injected $c = 10$ times, and had an inter-spike delay of 5 ms. We generated 100 realizations of this model. At each iteration, the neurons involved in each STP were selected randomly, but such that they would not form two identical sets. We obtained no FPs and no FNs. For the realizations where the two patterns overlapped, PSR successfully retrieved them, while correctly rejecting their intersection as a FP.

3.3.6 Validation of SPADE on inhomogeneous data

From the validation on stationary data we conclude that SPADE performs better than filtering methods based on stability measures. Therefore, we now concentrate on the SPADE method only. We aim at evaluating the performance of SPADE on data that mimic more closely features of experimental data, such as non-stationarity of firing rates in time and inhomogeneous firing rates across neurons. In particular, we study three cases (firing rate co-modulation, rate hetereogeneity across neurons, and rate change propagation, as introduced in section 3.3.2) that are known to be potentially strong generators of false positives for correlation analysis methods for (see e.g. Grün et al., 2003; Grün, 2009; Louis et al., 2010b; Torre et al., 2016a). The results are shown in Figure 3.6. The FP rate of the analysis of data with injected pattern is generally low (less than 0.05), for all three types of data. For firing rate co-modulation and rate inhomogeneity the FP rate is often virtually zero (white squares) but is somewhat more homogeneously at 0.05 for rate change propagation. This is somewhat surprising that such successive increases of firing rate, occurring repeatedly, do not elicit a higher level of FPs. The FNs decay with z and c in all three scenarios and fastest for the rate propagation model. Interestingly, these results are not worse than for the stationary data, meaning that SPADE can deal well with data that contain features that are typically generating FPs. In conclusion, SPADE can tolerate coherent rate non-stationarities and inhomogeneous rates and, thus, is qualified to be applied to experimental data.

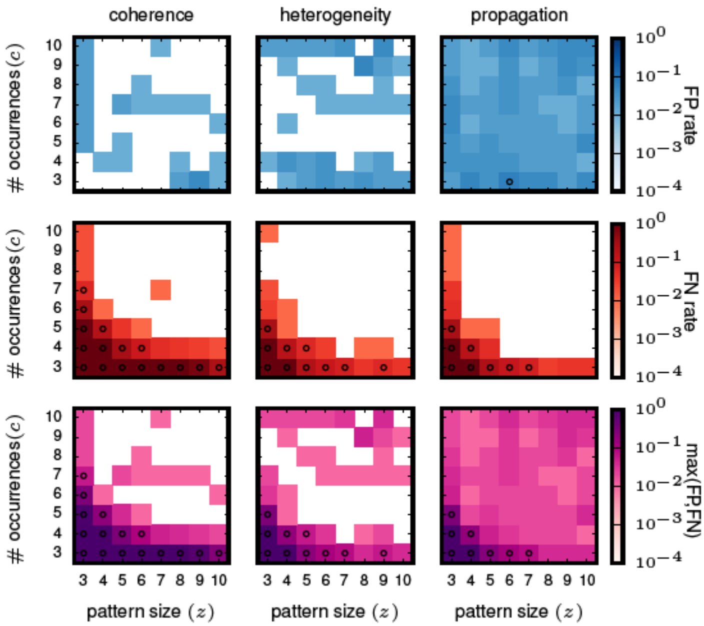

Figure 3.6: **Performance of SPADE on non-stationary and inhomogeneous data.** SPADE is applied to stochastic simulations of different rate data models, left column: rate coherence, middle column: rate heterogeneity, right column: rate propagation. For the FP rate (top row) and FN rate evaluation (middle row), patterns of a given size z and number of occurrences c are inserted into the same background rates (given by the data model). In the bottom row, the maximum of FP or FN of each (z,c) signature is shown. The same settings are used for FP-growth and SPADE as in Figure 3.5.

3.3.7 Summary of the Validation Results

Our validations highlight the following aspects:

- fast-FCA and FP-growth lead to identical mining results, however, the better computational performance of FP-growth allows one to mine concepts of real-sized data (100 or more neurons recorded over several seconds).

- Stability (approximate) filtering and significance filtering (the combination of PSF and PSR, i.e. SPADE) are efficient statistical techniques to reject chance patterns in independent (STP-free) data, as they all exhibit a small FP rate ($< 1\%$ for PSF+PSR, intensional stability and extensional stability; $< 2\%$ for the combined stability filter).

- FPs on data with injected patterns shows that all methods perform about the same with FPs on the 5% level.

- Significance filtering (SPADE) is the technique that best detects injected STPs in the data, exhibiting a FN rate below 5% for patterns with lower size or occurrence counts than stability filtering.

- These considerations for SPADE also hold for data with highly variable firing rates over time and across neurons, which suggests that the combination of FP-growth, PSF and PSR is the best technique to detect STP in real recordings.

- At the same time the stability based analyses, which have an equally low FP rate, although less sensitive to the presence of STPs, is applicable to particularly long recordings for which PSF is computationally not feasible.

3.4 Discussion

The ever growing number of neurons that modern electrophysiological techniques allow to record in parallel provides access to the coordinated spiking activity of neuronal populations of unprecedented size. The investigation of millisecond-precise spatio-temporal spike patterns (STPs) in large scale recordings becomes, therefore, possible. However, suitable analysis techniques have been lacking so far due to the exponential growth of the number of STPs in such large data, which yields severe computational and multiple testing issues.

Here we addressed this problem by introducing a method, named SPADE (Spike Pattern Detection and Evaluation), that extracts STPs from massively parallel spike train data and assesses their statistical significance under the hypothesis of spike independence. SPADE builds on and brings together two techniques that we had previously introduced for the identification of STPs in massively parallel spike trains and for the statistical evaluation of patterns of synchronous spikes (Torre et al., 2013). The latter avoided the computational and multiple testing issues that usually prevent applying such analyses to large data sets. The underlying pattern mining algorithm FP-growth, however, was implemented such that the technique was applicable for the discovery of synchronous spike patterns only. A restructuring of the input data format ('attribute scaling' in the language of FCA, see Ganter and Wille (1999)) allowed us now to use FP-growth (or similar frequent pattern mining techniques, see e.g. Borgelt, 2012) to extract more general STPs.

Thus, FP-growth served here the same purpose that fast-FCA served in chapter 1, that is, counting the occurrences of non trivial repeating patterns (there named "intents"). As known from the literature (Zaki and Ogihara, 1998) FIM and FCA yield results that can be mapped one to one onto each other: they extract closed frequent itemsets / formal concept intents including their occurrence count. Our implementation of FP-growth, however, proved to be much faster than the state-of-the-art implementations of the FCA algorithms available to our knowledge. Soon, a C implementation of the FCA's In-Close algorithm (Andrews, 2009) will be made available by S. Andrews, N. Kodoga and colleagues, which may provide a mathematically equivalent but computationally even faster algorithm to mine re-occurring STPs.

SPADE assesses the significance of the patterns identified by FP-growth or equivalent algorithms via the same analysis steps as in Torre et al. (2013). First, pattern spectrum filtering (PSF) is used to determine the p-value of signatures (z, c), i.e. pairs of pattern size z and occurrence count c, and retains patterns with significant signatures only. The number of different signatures in data is typically orders of magnitude smaller than the number of different patterns. Thus, testing for the signatures reduces the multiple testing problem to a size that can be handled with standard statistical corrections, such as false discovery rate. Then, pattern set reduction (PSR) is applied to test all patterns identified by PSF, conditioned on the presence of any other pattern in the remaining list. This allows one to distinguish, among overlapping patterns, the genuine ones from those that can be explained as a chance overlap of real patterns with background spikes. Validation on test data generated by different stochastic models of STPs injected into background activity demonstrated the ability of the method to discriminate real and chance STPs, ensuring low false negative (FN) and low false positive (FP) levels despite the large number of STPs to test (up to millions). For example, in simulated data consisting of 100 neurons spiking independently at an average rate of 15 Hz each for a period of 1 second, an injected STP was successfully isolated from the background activity as soon as it involved at least 5 neurons and it repeated 3 times, or it involved as low as 3 neurons and repeated 5 or more times. The method showed high power (FN rate lower than 5%) and reliability (FP rate lower than 5%) in different scenarios replicating various features of the firing rates often observed in experimental data, which typically represent strong generators of FPs (Louis et al., 2010b). These include abrupt and coherent rate changes over time, largely different firing rates across neurons, sudden rate changes propagating from one group of neurons to the other. Our method performs well in all of these scenarios.

Besides qualifying STPs as excess patterns on the basis of their statistical significance, we additionally explored various ways to extract them from background activity on the basis of their extensional or intensional stability. In FCA terminology, the extent of a concept in our context (frequent closed pattern) is the set of windows the patterns falls into. The intent of a concept is its composition, i.e. the neuron index and time index (within the window) of each composing spike. Intensional stability quantifies the tendency of a pattern occurring in a set of windows to be the largest pattern common to those windows (or any subset thereof). Low intensional stability indicates that the intersection of any number of those windows tends to contain supersets of that pattern and, therefore, that the pattern occurrences may have been the result of intersections of fewer occurrences of different larger patterns. Similarly, extensional stability quantifies the tendency of a set of windows to contain a subpattern of the pattern which comprises its intent, such that the subpattern is not found in any window that does not contain the pattern. Intensional and extensional stability are used as indicators of how reliably the pattern can be considered as a genuine event, rather than the sum of occurrences of larger patterns or the superposition of smaller patterns occurring in the same time segments, respectively. In chapter 1 we explored the use of intensional stability to isolate reliably re-occurring STPs from high background activity. The exact computation of the stability of each pattern, however, is computationally very demanding, and was possible only on data comprising a maximum of 50 neurons simulated for a few seconds. Here we adopted the Monte-Carlo based approximation of stability introduced by Babin and Kuznetsov (2012), which allowed us to speed up the computation by several orders of magnitude while introducing negligible errors and, thus, enabled the application of intensional as well as extensional stability based pattern filtering to larger data. In particular, we computed a statistical threshold for both intensional and extensional pattern stability, using independent surrogate data, and we filtered out patterns whose (intensional, extensional, or both) stability values were lower than the respective thresholds. Compared to the previous approach based on pattern significance, all of these stability-based criteria were computationally less demanding but yielded increased FNs, especially when the injected STP to be retrieved had low pattern size. Nevertheless, other combinations of these approaches may be envisioned in the future to improve the performance of the method even further if even more spike trains become available in parallel.

Existing methods for the identification of repeating STPs are either not applicable to data sets of large size, or limit the search to patterns with specific features (of fixed, usually small size, or exhibiting specific inter-spikes intervals such as synchronous patterns). SPADE does not suffer from these limitations thanks to a combination of attribute scaling, fast frequent item set mining, and a hierarchy of tests of pattern significance, which avoids severe multiple testing. Its extensive validation ensures its reliability in applications to real data, as well as to simulated data resulting from network models. SPADE can be applied not only to spike data, but also to any data consisting of parallel point processes, such as discretized calcium imaging data (e.g. Roth et al., 2012), discretized voltage-sensitive dye imaging data (Ayzenshtat et al., 2010) or discretized MEG recordings (e.g. see Tal and Abeles, 2016). In the work of Ayzenshtat et al. (2010) and Tal and Abeles (2016) the authors defined and extracted special events from their analog recordings (voltage-sensitive dye imaging and MEG, respectively) and reduced them to point events. In the published work they analyzed subsets of the resulting parallel point processes for pair-wise of triple-wise correlations or spatio-temporal patterns, respectively, and identified those in relation to the behavior. With SPADE the complete data sets from these recordings (i.e. massively parallel point processes) could be analyzed. The analysis time scale is thereby not restricted to milliseconds (as employed here) but can be freely adjusted depending on the research question.

Chapter 4

Detection and Analysis of Spatio-Temporal Patterns of Multiple Time Durations - SPADE extended

4.1 Introduction

In the previous chapters we introduced SPADE, a method for detection of Spatio-Temporal Patterns in parallel spike trains. The method consists of two main steps: a) a mining technique applied to the spike trains to detect all repeated STPs, b) statistical evaluation of the found STPs.

In the previous chapter we implicitly assumed that the patterns have the same duration (i.e. the time from first to last spike in the pattern). In this chapter we show that patterns of different durations have different distributions under the null hypothesis of independent spiking. Hence, in order to have an unbiased test for patterns with different durations, it is necessary to revise the statistical evaluation of SPADE by testing separately patterns of different durations. Using artificial ground truth data for the validation of the new statistical procedure, we show that the correction adopted in the testing procedure increases the statistical power without affecting the robustness of the method.

We then apply the extended version of SPADE to experimental data, in order to verify the hypothetical presence and computational role of STPs in cortical spiking activity. In particular we analyze two in-vivo recording sessions of massively parallel spike trains from motor cortex of two macaque monkeys performing a reach-to-grasp task (Brochier et al., 2018; Riehle et al., 2013). The monkeys were trained to execute four different motor behaviors, given by the combinations of two different grips and two different force levels. We analyzed separately the data of the different behavioral types to assess whether different spike patterns occur in different behavioral contexts, with the question in mind if we find indications of different active assemblies in different behavioral context.

In section 4.2 we introduce the new statistical testing procedure adapted to consider patterns with different durations. In section 4.3 we report validation results and results from the analysis of experimental parallel spike trains recordings.

4.2 Extension of SPADE

In section 4.2.1 we present the problem of considering data containing patterns of different durations. In addition we also discuss an alternative cause for false detection, i.e. coherent rate changes of neurons, on different time-scales. Consequently we propose an improvement of the statistical testing in SPADE to overcome such problems. In section 4.2.2 we introduce the artificial datasets that we use to evaluate the statistical performance of the new statistical test.

4.2.1 3-dimensional Pattern Spectrum

In chapter 3 we presented the SPADE method for detection of Spatio-Temporal patterns. In particular, that the Pattern Spectrum Filtering (PSF) is used to discriminate significant spatio-temporal patterns from chance occurrences, by avoiding massive multiple testing. If all individual patterns would be tested on their own separately, we would run into that problem. Instead of testing the significance of each closed frequent itemset, detected by FIM, all pattern candidates with same size z and number of occurrences c are pooled together and tested together. Thus, each pattern is characterized only by its signature, defined as the pair (z, c). The probability to observe any pattern with signature (z, c) is then derived using a Monte-Carlo technique based on surrogates. This approach was already introduced for synchronous patterns by Torre et al. (2013) and was also introduced for testing spatio-temporal patterns. Nevertheless, since multiple signatures are tested, we correct the significance threshold using a False Discovery Rate Correction.

We have shown that that SPADE is statistically robust and discards efficiently chance STPs if applied to artificial data . Yet so far we did not consider the case of spatio-temporal patterns of different durations in the data. In such a case the PSF pools patterns of different durations in one signature. Thus, for example, it is possible that the very same p-value is assigned to synchronous patterns or patterns with long temporal extension. The problem with such an approach is that, even in case of independent spiking, patterns with different durations have different probability to occur, because of the different number of possible spike combinations for different durations. It is possible to consider the minimal example of two Poisson Processes (X_1, X_2), each with one single spike, respectively at time t_1 and t_2, in the time domain $[0, T]$. Because of the poissonianity of the processes, the two spike times t_1 and t_2 are two uniform random variable in the interval $[0, T]$; hence it is possible to compute the probability of the two spikes being synchronous given one binsize dt. This probability is equal to the probability that the two spikes are in the same bin:

$$\mathbb{P}\left(\left\lfloor\frac{t_1}{dt}\right\rfloor = \left\lfloor\frac{t_2}{dt}\right\rfloor\right) = \sum_{k=1}^{\frac{T}{dt}} \mathbb{P}(k \cdot dt < t_1 < (k+1) \cdot dt)\mathbb{P}(k \cdot dt < t_2 < (k+1) \cdot dt) =$$

$$= \sum_{k=1}^{\frac{T}{dt}} \left(\frac{dt}{T}\right)^2 = \frac{T}{dt} \cdot \left(\frac{dt}{T}\right)^2 = \frac{dt}{T}$$

We can then consider the probability that they form a pattern of duration equal to d bins, with $0 < d < \frac{T}{dt}$. In this case, we have the combination of two possibilities, that t_1 precedes t_2 of d bins, or vice versa, t_2 is d bins before t_1:

$$\mathbb{P}\left(\left\{\left\lfloor\frac{t_1}{dt}\right\rfloor + d = \left\lfloor\frac{t_2}{dt}\right\rfloor\right\} \cup \left\{\left\lfloor\frac{t_1}{dt}\right\rfloor - d = \left\lfloor\frac{t_2}{dt}\right\rfloor\right\}\right) =$$

$$= \sum_{k=1}^{\frac{T}{dt}-d} \mathbb{P}((k+d)dt < t_1 < (k+d+1)dt)\mathbb{P}(k \cdot dt < t_2 < (k+1) \cdot dt) +$$

$$+ \sum_{k=d}^{\frac{T}{dt}} \mathbb{P}((k-d) \cdot dt < t_1 < (k-d+1) \cdot dt)\mathbb{P}(kdt < t_2 < (k+1)dt) =$$

$$= \sum_{k=1}^{\frac{T}{dt}-d} \left(\frac{dt}{T}\right)^2 + \sum_{k=d}^{\frac{T}{dt}} \left(\frac{dt}{T}\right)^2 = \left(\frac{T}{dt} - d\right) \cdot \left(\frac{dt}{T}\right)^2 + \left(\frac{T}{dt} - d\right) \cdot \left(\frac{dt}{T}\right)^2 = 2 \cdot \left(\frac{T}{dt} - d\right) \cdot \left(\frac{dt}{T}\right)^2$$

This simple calculation shows that even in the most simple case of two single Poisson spikes two patterns with the same signature ($z = 2, c = 1$) but different durations have different probability to occur. The example here considered can be easily extended to the case where we have multiple spikes, for which the combination of different patterns for each duration is even larger.

Furthermore we will show that the p-values obtained using the pattern spectrum, as defined in chapter 3, depend on the window length used for the pattern detection. In particular, if we consider two window lengths K_1 and K_2, such that $K_1 < K_2$, the p-value assigned to a certain signature (z, c) is different. Indeed for the smaller window we count all the chance patterns in the surrogates with z spikes and occurring c times that have duration $d \leq K_1$. For the second window the count includes also the patterns with the same signature but which duration is such that $K_1 \leq d \leq K_2$. Since more and more chance patterns are pooled together with increasing window lengths, the p-value assigned to a certain signature is larger when using a larger window. This implies that results obtained analyzing the same data, but using different window lengths are not comparable. In particular not considering the different pattern durations in the testing introduces a bias for short patterns, which are more likely to be significant with short windows, while no longer when pooled together with longer patterns using a longer window for the detection.

However, there is also another issue that effects the statistical evaluation of the STPs: delayed co-nonstationary firing rates. For example, in the case in which two neurons increase their firing rate with a constant delay, a large number of chance patterns in the same temporal delay range will be present in the surrogate data as well since they are designed to preserve the firing rate profiles. This results in a large p-value, i.e. non-significant, also for patterns with different duration but same signature. For example we can consider the case in which two neurons increase their firing rate with a constant delay l, such that l is smaller of the window used for pattern detection K. Using the current version of the p-value spectrum, a pattern in order to be significant needs to occur more than the most frequent pattern of the same size in the surrogate data. In such particular case many chance patterns of durations l would occur because of the delayed firing rate increase. Hence, also patterns with a different duration d (e.g. synchronous patterns with

$d = 0$) would need to occur more than the number of chance patterns of duration l found in the surrogate. This makes the test over-conservative, because patterns with duration d may occur less likely than pattern with duration l in the surrogates.

In order to overcome these limitations, we here propose an alternative definition of the pattern spectrum. We replace the original pattern signature (z, c) with the a triplet (z, c, d) signature, with the duration d defined as the number of bins between the first and the last spike of the pattern. We then derive the p-values independently for each 3-dimensional signature, again with the same bootstrap surrogate sampling as done for the 2-dimensional (chapter 3). This solution largely increases the number of tests to be performed and consequently may cause an increase in the number of false positives. In order to approximately obtain the same significance level as before, we apply the Holm-Bonferroni multi-comparison correction (Holm, 1979), which is more conservative than the False Discovery Rate (FDR) correction (Benjamini and Hochberg, 1995) used for the 2-d spectrum.

4.2.2 Artificial Data

In order to expose the problems with the previous definition of the significance test based on the 2-d spectrum, and to show how using the 3-dimensional signature amends them, we consider here two types of simulated datasets. Both types consist of $N = 100$ parallel processes.

The first is composed of stationary independent (see Figure 4.1) Poisson processes of constant firing rate $r = 15\,\mathrm{Hz}$ lasting in total $T = 10\,\mathrm{s}$. Into these processes we inject a total of 5 patterns with identical size $z = 3$ but of different durations ($d = 0\,\mathrm{ms}, 2\,\mathrm{ms}, 6\,\mathrm{ms}, 8\,\mathrm{ms}, 12\,\mathrm{ms}$, respectively). We then derive the minimal number of occurrences of a pattern of size 3 in order to be significant using a significance level equal to 0.05 and a binsize equal to $dt = 1\,\mathrm{ms}$. Given our selection of firing rate and time domain, such number of occurrences is equal to 4. We will show that with the original 2-d p-value spectrum all the patterns are not significant for the window length $K = 13$, which is larger than the maximal duration and should allow to detect all the patterns. At the same time for shorter window lengths the patterns are significant, but not all are detected because having a duration longer than the selected window length. On the other hand with the 3d p-value spectrum all patterns injected are always significant, as expected given the Poisson distribution, since the window length does not affect the computation of p-values. In particular using the single window length $K = 13$ it is possible to retrieve all the injected patterns.

For the second dataset (Figure 4.2) we use two different groups of non-stationary Poisson processes, now with time domain $T = 5\,\mathrm{s}$. The first 50 Poisson Processes have a firing rate equal to $5\,\mathrm{Hz}$ in the intervals $[0\,\mathrm{s}, 1\,\mathrm{s}]$ and $[1.05\,\mathrm{s}, 5\,\mathrm{s}]$, while in the $50\,\mathrm{ms}$ gap the processes have a coherent increase of the firing rate to $60\,\mathrm{Hz}$. The second half of the population has the same firing rate profile but the increase is delayed by $50\,\mathrm{ms}$ (firing rate equal to $60\,\mathrm{Hz}$ in the interval $[1.05\,\mathrm{s}, 1.1\,\mathrm{s}]$). We then inject one single synchronous pattern of size $z = 3$, occurring $c = 4$ times. In order to emphasize the potential effect of a correlation of the rate changes onto the p-values of the patterns we choose the quantity c such that the pattern should become significant (0.05 quantile of the Poisson distribution). The use of the 3d-spectrum enables to include the effect of rate correlations in the null hypothesis only for the patterns with a duration equal to the delay in the rate changes,

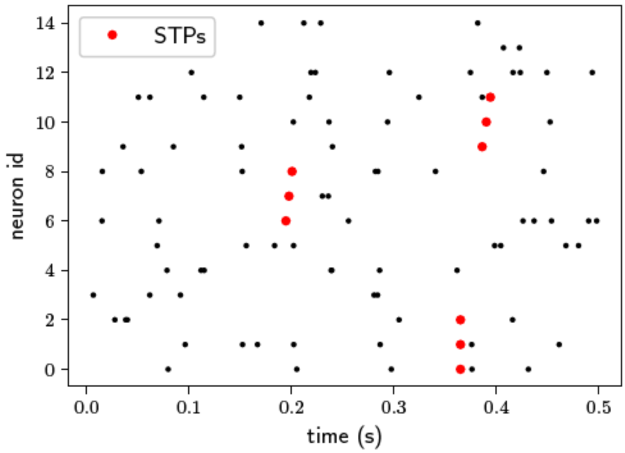

Figure 4.1: **Stationary artificial data with multiple patterns.** Each row corresponds to one of the parallel simulated Poisson processes, each dot corresponds to one spike. The several sequences of red dots show the injection of the same injected pattern. Here only 0.5 s and the first 15 out of the 100 processes are shown.

allowing the synchronous pattern which is not affected by the delayed rate changes to be correctly detected as significant.

Additionally, we analyze also 4 (stationary firing rates equal to 25 Hz, coherent increase of firing rate for all the neurons, heterogeneous stationary firing rates, propagation of firing rate changes) of the datasets presented in the previous chapter (section 3.3.2), in order to assess the statistical performance of the PSF using the 3-d spectrum (in terms of false positives and false negatives).

4.2.3 Analysis of artificial data

In the following we compare results of the analysis of artificial data using SPADE in the 2d and the 3d version. For doing that we created multiple STPs with different pattern durations (section 4.2.2). For both types of analysis we used window lengths K of $(1, 4, 7, 10, 13)$ bins. Other parameters, ($dt = 1$ ms, $\alpha = 0.05$) were not changed and are identical to ones used in the previous chapter. Since in this example application we do not focus on the statistical power of the method, we use the FDR correction for both (2-d and 3-d spectrum) as also previously done when applying SPADE.

In Figure 4.3 we show the number of patterns detected using the two different spectra. For the 2-d spectrum (top panel) all patterns are correctly detected for window lengths 1, 4 and 7. For window lengths 10 and 12 no significant patterns are found, although there were injected in the data. Including the pattern duration in the pattern signature (3-d spectrum bottom panel) enables to correctly detect all the injected patterns for all the different window lengths. In particular using a window length of 13 ms, that is larger than the longest duration of all the injected patterns (here 12 ms), makes the detection of all

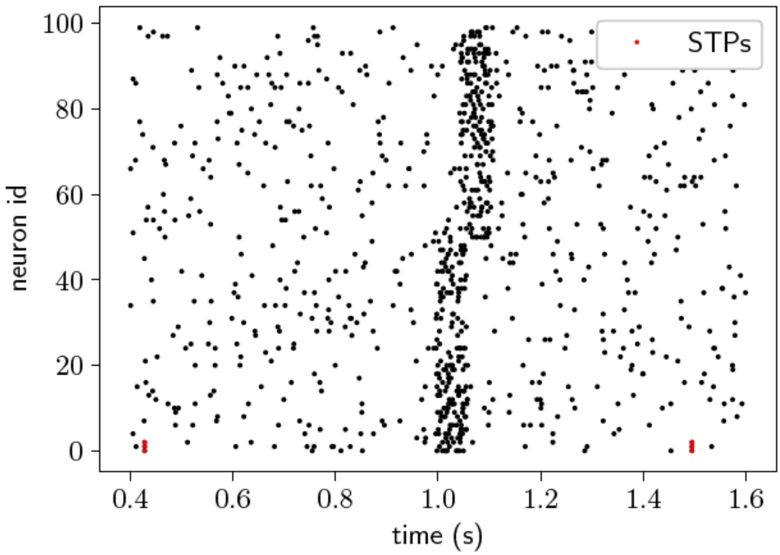

Figure 4.2: **Non-stationary artificial data raster plot.** Each row correspond to one of the parallel simulated Poisson processes, each dot is a spike, each line shows the spike trains of different neurons. The red dots denote the injected patterns. Here only the first 1 s of data are showed.

patterns possible.

The difference of the results using the 2d and the 3d spectra is explained by the p-values of patterns of size 3 shown in Figure 4.4. Indeed, in the 2-d spectrum, for window lengths larger than 10, patterns occurring 4 times do not become significant, while they do for shorter window lengths. The reason is that the p-values computed with the 2-d spectrum depend on the window length (see section 4.2.1). In particular smaller window have smaller p-values, simply because only shorter patterns can be extracted and less patterns are pooled together in the same entry of the spectrum. Vice-versa including the pattern duration as an additional dimension in the p-value spectrum enables to remove the dependency from the window length, causing the patterns to be significant also for longer windows.

For the analysis of the second data set we consider the case in which the firing rates are covarying, and contain per neuron one firing rate step (see 4.2). From studies on spike synchrony (e.g. Grün et al., 2002b; Louis et al., 2010c)we know that such coherent rate changes are potential generators of false positive results. Thus we here use data that are also potential false positive generators. However, since we deal here with spatio-temporal patterns, we insert two coherent rates steps with a certain delay. First we study here the potential effect of false negatives when delayed rate changes prevent the detection of the inserted patterns with the 2-d spectrum. In the next section we study also false positives. We set here the window lenght to 100 ms, because this is the temporal range of the two coherent rate steps and thus will generate chance STPs on that time scale. The other parameters are the same as used for the previous data set ($dt = 1\,\mathrm{ms}$, $\alpha = 0.05$), and we analyze this data also with the 2-d and 3-d spectrum for comparison. Indeed the 2-d pattern spectrum resulted in a non-significant entry for the signature $(z, c) = (3, 4)$ (for all the pattern durations) which should include our injected patterns. It does not become significant, because of the many chance STPs that are due to the delayed coherent rate steps. On the other hand, in the analysis using the 3d-spectrum, patterns of size $z = 3$ and $c = 4$ occurrences are not significant for patterns of a duration of 50 ms - the delay of the firing changes. Instead, the signature of the injected pattern $(z, c, d) = (3, 4, 0)$ is significant, and is not affected by the rate changes.

4.2.4 FP/FN performance

In order the statistical performance of the 3-d spectrum as compared to the 2d-spectrum we reanalyze the data that were already analyzed with the 2-d spectrum in the previous chapter (section 3.3.2). See the previous chapter for the definition of False Positive (FP) rate and False Negative (FN) rate. The FP rate for data with no patterns injected is below 0.05 (not shown). Figure 4.5 shows the FP and FN results for all configurations of pattern parameters ($z = 3, \ldots, 10, c = 3, \ldots, 10$), each column for one of the four datasets already used in the previous for the validation of the 2-d spectrum. The FP rate is below 0.05 for most of the pattern configurations. On the other hand, most of the patterns occurring more that 4 times are correctly identified, i.e. with low FN rates (second row). In conclusion, the 3-d spectrum have analogous statistical performance to the 2-d p-value spectrum, but it enables to test independently for patterns of different durations.

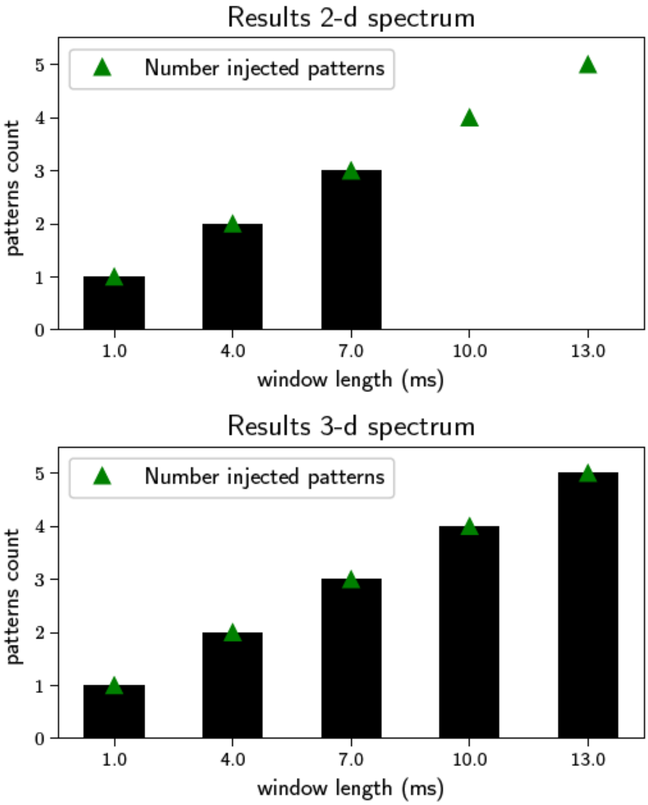

Figure 4.3: **Count of STPs.** The histograms show the number of detected significant STPs (y-axis) as a function of window length (x-axis). The green triangles mark the number of injected patterns with durations equal or shorter than the correspondent window length. The top panel shows the results when using the 2d-spectrum, the lower panel shows for the same data the results when using the 3d-spectrum. Other parameters: $dt = 1$ms, $\alpha = 0.05$, 5000 surrogates, FDR correction.

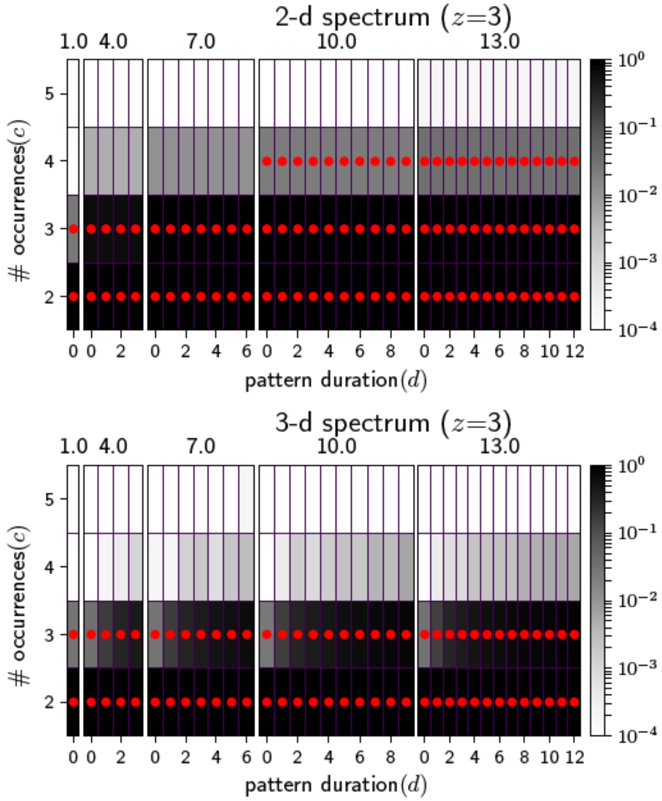

Figure 4.4: **P-value spectra.** P-value spectra computed for PSF for pattern size $z = 3$ and different number of occurrences c and pattern durations d. Non significant p-values are marked with a red star ($dt = 1ms$, $\alpha = 0.05$, 5000 surrogates and FDR correction)

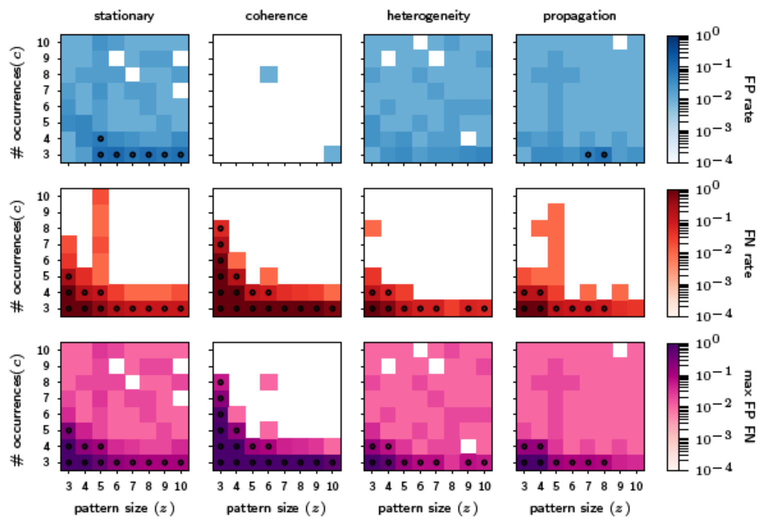

Figure 4.5: **Performance of SPADE using the 3-d spectrum.** SPADE is applied to stochastic simulations of different rate data models, left column: constant homogeneous firing rate (25 Hz) for all the neurons, second column: rate coherence, third column: rate heterogeneity, right column: rate propagation. For the FP rate (top row) and FN rate evaluation (middle row), patterns of a given size z and number of occurrences c are inserted into the same background rates (given by the data model). In the bottom row, the maximum of FP or FN of each (z, c) signature is shown.

4.3 Analysis of Experimental Data

We learned from the previous sections that the SPADE analysis with the 3d-spectrum performs much better as the 2d-spectrum version. The extension of the 2d-spectrum by the axis of pattern durations makes the analysis much more reliable in detecting patterns (less FNs). Here we show example applications of the 3d-version of SPADE on experimental data sets from the Reach-To-Grasp experiment (see section 4.3.1).

4.3.1 Reach-to-Grasp Data

Experimental protocol and experimental setup. We will analyze massively parallel spike train data from electrophysiological recordings from motor cortex of two different monkeys (*macaca mulatta*). The monkeys were trained to perform an instructed-delay reach-to-grasp task during the recordings. A detailed description of settings, setup and data preprocessing can be found in Brochier et al. (2018), Riehle et al. (2013), and Torre et al. (2016b). In brief, the experiment design was: the monkey was indicated to grasp and pull a cubic object horizontally, via two possible grips, either a full-hand side grip (SG) or a two-fingers precision grip (PG). Furthermore, the grip had to be performed with either of two different forces, either low ($1N$) or high ($2N$), respectively, indicated as low force (LF) and high force (HF). Both grip types and force types were indicated to the monkey with a 5-LED system situated above the to-be-pulled object. In order to get the reward, the monkey had to perform the reach and grasp and pull the object for $500ms$ and hold it within a given range. A scheme of the trial can be found in Figure 4.6. The time schedule of the experiment was: $400ms$ after the monkey self-initiated a new trial (trial start, TS), the LEDs system shortly lighted up in order to attract the attention of the monkey (WS). After $400ms$ a first visual cue, lasting $300ms$, appeared to inform the monkey about which grip to perform. After $1000ms$, both the force type and the GO signal were indicated to the monkey with a second visual cue. Thus, four possible trial types were designed: PGHF, PGLF, SGHF, SGLF. Each of them had the same probability to be requested, in a sequence of approximately 130 successful trial executions constituting a recording session (15 minutes). The stimuli, switch release and other behavioral data were also stored for offline analysis.

The recordings were performed by a 10-by-10-electrode Utah array (Blackrock Microsystems, Salt Lake City, UT, USA), positioned partly across the dorsal premotor (PMd) and primary motor (M1) cortex. The electrode lengths were $1.5mm$, thus approximately reaching layer 5. The inter-electrode distance was $400\mu m$.

For this study we selected two recording sessions, that are published online and described in detail in Brochier et al. (2018). The two datasets have been recorded from two different monkeys, monkey L and monkey N, respectively. In the two sessions 93 single units (SUAs) were recorded from monkey L, and 156 SUAs from monkey N. The spike sorting to extract the SUAs was performed offline using Plexon Offline Sorter (version 3.3.3). For any further details we refer to Brochier et al. (2018).

Definition of Epochs We aim at getting insight in the dynamics of pattern occurrence in relation to behavior. Typically a sliding window analysis is adopted for such a purpose (e.g. Grün et al., 2002b). However, since this would be too computationally too expensive we decided for a quasi-time resolved analysis by defining windows that capture behaviorally

Figure 4.6: **Reach-to-grasp experimental protocol.** The trial start (TS) was self-initiated by the monkey by pressing the home switch. A warning signal (WS) prepared the monkey for a visual cue (C-ON until C-OFF) providing the instruction about the grip type to use: SG or PG. 1 s later a second visual cue turned on which at the same time indicated the GO signal for movement initiation, specifying the force needed to pull the object (HF or LF). The movement onset is marked by the switch release (SR). After object touch (OT) the monkey pulled the object and had to hold it for 500 ms in a narrow position window until he was rewarded (RW) by a drop of juice. The timing of the behavioral events SR, OT and RW, which follow the GO signal, varied across trials depending on the monkey's reaction times and movement speed.

epoch name	**trigger**	t_{pre}	t_{post}
start	WS	250 ms	250 ms
cue	C-ON	250 ms	250 ms
early delay	C-OFF	0 ms	500 ms
late delay	GO	500 ms	0 ms
movement	SR	200 ms	300 ms
hold	RW	500 ms	0 ms

Table 4.1: **Definition of trial epochs.** The table summarizes the six different epochs defined for the analysis. Each epoch is a 500 ms time window starting a time t_{pre} before a trigger and ending a time t_{post} after that trigger ($t_{pre} + t_{post} = 500\,\text{ms}$).

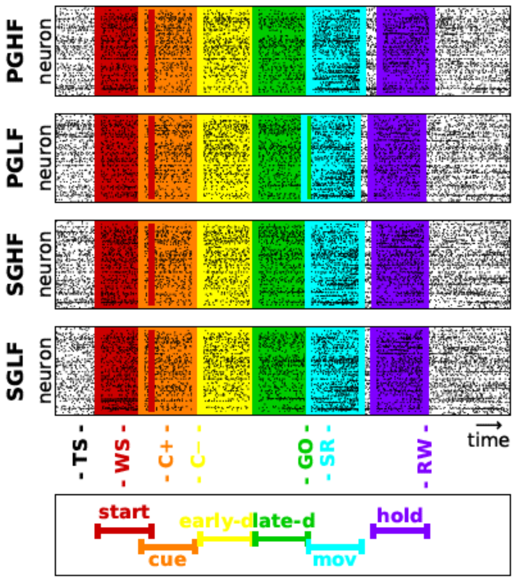

Figure 4.7: **Trial types and behavioral epochs.** Each panel shows the simultaneous spiking activity of all neurons (vertical axis) over time (horizontal axis), for four example trials (one per panel) of different behavioral types, from a representative session of monkey N. Each dot indicates a spike. The trials are aligned to trial start (TS). The six colored windows represent the position of the six epochs in a trial. The trigger associated to each epoch and the corresponding epoch are shown at the bottom (see 4.1 for details). The movement (green) and hold (yellow) epochs are centered around triggers that change the time of occurrence from trial to trial depending on the reaction time (time from GO to SR) of the monkey.

relevant epochs during the trial. Thus we divide each trial into six epochs of equal duration of $500ms$, identified by triggers and behavioral contexts. The epochs of one selected trial are illustrated as colored windows in Figure 4.7, and outlined in Table 4.1: *start* represents the time interval around the warning signal (WS) indicating to the monkey the beginning of a new trial; *cue* is centered around the first visual cue, informing the monkey which of the two grips to use; *early delay* and *late delay* relate, respectively, the first and the second half of the preparatory period, spanning in total the time between the first and the second visual cue (GO signal); *movement* covers the time during movement; *hold* indicates the epoch in which the object had to be hold until the reward was delivered.

Since we are interested in associating spatio-temporal patterns also to different movement behavior and its preparation, we divided each session into the different trial types (PG-LF, PG-HF, SG-LF and SG-HF). Each of these trials were divided into the 6 behavioral epochs mentioned before, such that we end up with 24 datasets (6 epochs $\times$ 4 trial types). Each dataset is composed by concatenated data pieces of the same epoch in the respective trial type, which then enter the analysis.

4.3.2 Parameters of the SPADE Analysis

For the analysis of the experimental data we set the SPADE parameters as follows: binsize $dt = 3\,\text{ms}$, sliding window duration $K = 20$ bins, 5000 surrogates, dithered in the interval $[-15, 15]\,\text{ms}$, and a significant level of $\alpha = 0.05$ corrected by the Holm-Bonferroni correction (Holm, 1979).

Reducing compute time

The extension of SPADE to STPs leads to a considerable compute time increase, even when using parallel computing on a compute cluster. For example, the analysis of one data session of 24 data epochs of 500ms each, and about 30 trials each, hits the computation wall-time of 48 hours on a compute cluster with 32 nodes, each consisting of a 2 $\times$ Intel Xeon E5 processor with 2.5 GHz processing speed and 8 $\times$ 16 GB DDR4 RAM. We were able to reduce this time considerably (to about 24 hours) by restricting the analysis to a selected set of patterns. In particular it is possible to derive a priori a minimal number of occurrences of a pattern to be considered in the analysis, reducing the number of patterns to mine and consequently the computational time. To derive such a minimal number we apply two criteria, that both have to be fulfilled. The first is that any pattern has to occur a global minimal number of occurrences, independently of its size. We request in particular that a pattern occurs in a minimal fixed portion of the trials, i.e. here where we have about 30 trials per trial type, we fix the minimal number of occurrences to 10. The idea behind that is that if a particular assembly is active in relation to a particular behavior, it should occur at least in 1 every 3 trials.

The second criteria is to derive analytically a minimal number of times $\hat{c}$ for any given pattern size z, assuming Poisson distribution and constant firing rates. It is then possible to ensure that the signature $(z, \hat{c})$ is not significant under the assumption of independent Poisson spiking and constant firing rate λ, given a significance level α and a binsize dt. Here we fixed the firing rate λ as the 90-th percentile of the distribution of firing rates of all the neurons in each behavioral context of the recordings. The probability to have one specific pattern occurrence in one window, for the Poisson and independence assumption is given by:

$$p = (\lambda \cdot dt)^z$$

The total number of possible combinations of z spikes from N neurons forming a pattern in a window of length K bins is given by:

$$\Gamma = \frac{K!}{(K-z-1)!} \cdot \binom{N}{z}$$

Hence, the minimal number of occurrences $\hat{c}$ can be obtained solving the following equation:

$$\frac{\alpha}{\Gamma} = \binom{\frac{T}{dt}}{\hat{c}} \cdot p^{\hat{c}} \cdot (1-p)^{\frac{T}{dt}-\hat{c}}$$

Furthermore, according to the first criteria, we do not consider patterns for which $\hat{c}$ is smaller than the $\frac{1}{3}$ of the number of trials, having about 30 trials per trial type here we considered only $\hat{c}$, such that $\hat{c} > 10$. In conclusion in our analysis we consider only patterns, which the number of occurrences c is such that $c \geq \hat{c} \geq 10$.

4.3.3 Results

We analyzed the two recording sessions of the Reach-to-Grasp experiment (described in section 4.3.1), using SPADE with the 3-d p-value spectrum. Figure 4.8 summarizes the features of the detected significant patterns for the two monkeys. Panel A shows the number of significant patterns found in each of the 6 epochs (along the x-axis) and 4 trial types (coded in color). Most of the patterns for both monkeys occur during the movement epoch. This finding is coherent with the results of the synchronous pattern analysis Torre et al. (2016b). In monkey Na larger number of patterns occurred, which is not too surprising since a larger number of single units are available for monkey N (96 for monkey L and 156 for monkey N).

In the panels below of the same figure only patterns of the movement epoch are further considered. Panel B shows the number of patterns of different sizes. Most of the patterns involve two neurons only, but we detect also a considerable number of patterns of size 3 and 4. The patterns exhibit a variety of different lags on different time scales (Panel C), but their number decreases with increasing lag.

The pattern occurrences in the two monkeys have different temporal dynamics. Panel D shows the counts of spikes involved in one or more patterns in respect to the time in the movement epoch (time axis is aligned to the movement on-set). For monkey N patterns occur after movement, while for monkey L they occur primarily before movement onset. This may reflect an anticipation of the movement in monkey L, which is also consistent with the higher number of patterns found during the late delay epoch. Furthermore, the average reaction time is shorter for monkey L than for monkey N (see Brochier et al., 2018).

Next, we investigated the neural compositions of the patterns and their spatial distribution on the Utah recording array. Figure 4.9 shows sketches of the electrode array, each for a different trial type. Each square of the array represents one of the 100 recording electrodes, organized in 10x10 grid of the Utah array. For the exact position of the

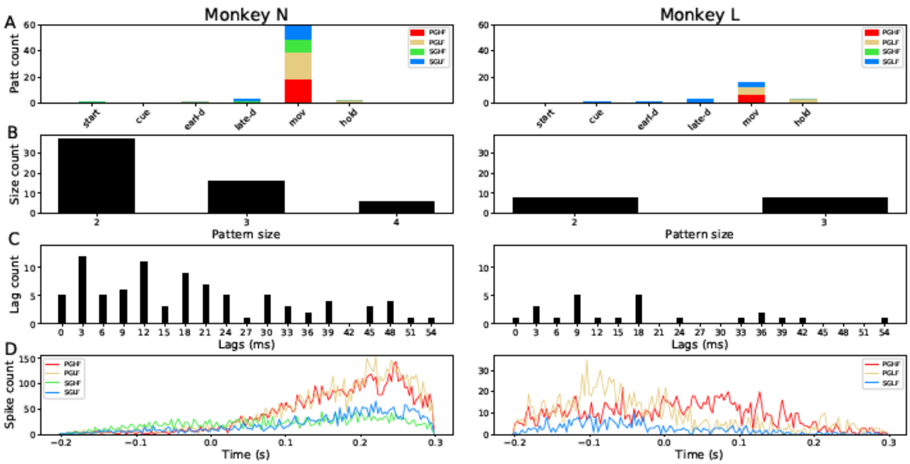

Figure 4.8: **Features of found significant patterns: A)** shows the number of significant patterns found in each of the 6 epochs (horizontal axis) and 4 trial types (color coded). B), C) and D) show only results of the patterns found during the movement period: **B)** histogram of the number of patterns with a specific pattern size, **C)** histogram of the number patterns of a particular lag, **D)** PSTHs of spikes involved in a pattern in a particular trial type (aligned on movement onset, SR).

arrays on the cortex surface we refer to Brochier et al. (2018) and Torre et al. (2016b).. The colors indicate the number of patterns a neuron recorded from a particular electrode participated to (no patterns formed by different neurons recorded on the same electrode were found and no patterns are detected for monkey L in trial type SGHF). Interestingly, for both monkeys we find that only a few neurons that participate in patterns (4/156 for monkey N and 4/96 for monkey L). For monkey N, the contributing neurons are identical for PGHF and PGLF, but with slightly different counts. The neuron involvements differ slightly for SGHF and SGLF: 5 neurons stay identical, one changes. However, there is also a strong overlap of neurons (3) that are active also in patterns occurring in SG. For monkey L there is a larger difference across trial types: the neurons involved in patterns do overlap between PGHF and PGLF, but additional ones are involved during PGHF. In SGHF no patterns occur, but in SGLF, 2 neurons overlap with the PG patterns, but there are additional neurons involved. Thus, the neurons involved in patterns are specific for the different grip modes but not for the force levels, yet there are neurons which participate in the patterns regardless of the trial types.

Currently, we cannot link the positions of the neurons involved to functional aspects of the behavior.

4.4 Software and Workflow

Reproducibility is an essential element for rigorous scientific research. To this aim we make all the software used in this study publicly available. This includes also the workflow

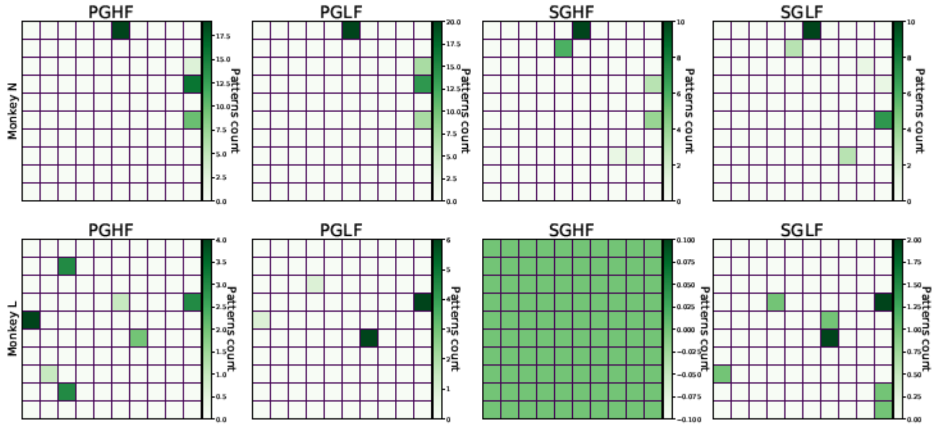

Figure 4.9: **Pattern counts on the Utah array.** Each panel shows the spatially resolved count of pattern participation of the neurons at the respective electrodes to patterns during movement, for the 4 different trial types. Each square in each of the panel corresponds to the position of one electrode on the 10x10 Utah electrode. The number of patterns in which the neurons are involved are color coded.

necessary to reproduce results and figures presented here.

The SPADE method, with all the different possible variants developed in this work (e.g. computation of pattern stability, 2-d and 3-d p-values spectra) has beed integrated by me into the Electrophysiology Analysis Toolkit[1](Elephant, RRID:SCR_003833), an open source toolbox providing data analysis tools for electrophysiology data, developed in our lab.

Typically, software on its own is not sufficient to reproduce a scientific results. The knowledge about the whole analysis workflow (e.g. parameters configuration, input-output relations, etc.) is in most cases crucial to reproduce a result. For keeping track of these workflows several provenance tracking and workflow managment tools emerged in the past years for research and analysis. In this study we make use of three of them to publish our analysis and keep track of the analysis workflow. We make use of the Snakemake[2] workflow management system for the definition and implementation of the analysis workflow. Figure 4.10 illustrates the Directed Acyclic Graph (DAG) representation of the workflow used to generate and analyze the artificial data. The code necessary to reproduce the entire study, including both SPADE (`https://github.com/NeuralEnsemble/elephant`) and the analysis workflow (`https://github.com/INM-6/SPADE_analysis`), is available on the web-based hosting service for version control GitHub[3]. The code necessary to simulate artificial data used in this study is included in the repository together with the Snakemake workflow. The data of the Reach-to-Grasp experiment analyzed here were published (Brochier et al.,

[1] `http://python-elephant.org`
[2] `https://snakemake.readthedocs.io`
[3] `https://github.com/`

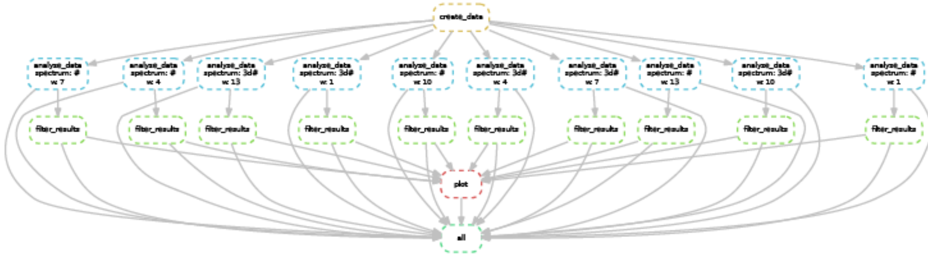

Figure 4.10: **The Directed Acyclic Graph (DAG) of the SPADE workflow to analyze the first artificial dataset.** Snakemake allows to obtain and visualize a diagram of the entire workflow and parameters used. Each node of the graph corresponds to one step of the analysis workflow, the connecting links represent the input-output relations. The steps that can be run in parallel form one layer of the graph. When one step requires external parameters, the values that each of them assumes are reported explicitly in the DAG (e.g. the rule `analyze_data` requires the parameter of the dimensionality of the spectrum to use and the window length for the FIM analysis).

2018)and thus are publicly available (at G-Node[4]) and are accompanied by meta-data and loading scripts (`https://web.gin.g-node.org/INT/multielectrode_grasp`).

4.5 Discussion

In this chapter three results concerning SPADE were shown: a) that is it necessary that patterns of different durations are evaluated for significance separately and why; b) we applied the new method to experimental data, and c) we found STPs that are specific to the behavior. In the following we discuss these issues separately.

STPs of different durations (time between first and last spike of the sequence) need to be analyzed for their significance separately, since otherwise the significance of STPs of larger durations are underestimated. We showed that the reason is that shorter patterns of the same number of spikes are by chance so many more that longer patterns may be overseen. In order to correct for that, we extend the p-value spectrum (introduced in previous chapter) to a third dimension, i.e. the pattern duration. Patterns of same size, number of occurrences and duration are now pooled into corresponding bins before to be tested for significance, while in the 2-d spectrum all the patterns with same size and number of occurrences were tested together regardless their duration. We present the application to artificial data using the 3-d p-values spectrum, and show that it amends the missed detection of injected patterns when using the 2-d spectrum. The 3-d spectrum ensures a larger statistical power than the 2-d p-value spectrum. In order to ensure that the new testing procedure does not affect the robustness of the method, we also validate its performance in terms of false positives and false negatives. We found the necessity of using a different multiple testing correction (Holm-Bonferroni rather than FDR) in order to compensate for the larger amount of tests to be performed. In conclusion, the performances of the revised pattern spectrum filtering are similar to the ones of the 2-d

[4] `https://web.gin.g-node.org/`

spectrum in the case of a single pattern injection, but improved considerably for detection of multiple patterns with different durations.

Next we applied SPADE, with the 3-d spectrum testing, to two experimental datasets. We considered two recording sessions of massively parallel spike trains, obtained from the pre-/motor area of two macaque monkeys performing a reach-to-grasp task (Riehle et al., 2013). The data were separated into trials of different behavioral conditions (PGHF, PGLF, SGHF, SGLF) and performed for each of them a quasi time-resolved analysis to capture dynamic aspects of pattern occurrence. We found that significant patterns primarily occur during the reach and grasp movement.

To our surprise (see also discussion in Torre et al. (2016b)) the size of the patterns range between 2 and 4 spikes, less than 5% of the total number of observed neurons. They exhibit a variety of temporal lags. .Different patterns, both in terms of neuronal composition and spatial arrangement of the involved neurons, appear to characterize the different grips (precise and side grip). In addition, some neurons participate in all patterns of a trial type, and even their individual spikes participate in different patterns. The same neurons also appear in different patterns across trial types.

In a next step we aim to analyze many more sessions of the same experiment, to test if the same observations of pattern features occur there as well. In addition, we aim to interpret the spatial arrangement of the neurons involved in patterns in terms of the functional map in pre-/motor cortex (Riehle et al., 2013). A further direction of research will be to relate pattern spikes to frequency and phase of the local field potentials in the spirit of Denker et al. (2011).

The entire analysis of both artificial and experimental data is publicly available online[5], as well as the SPADE implementation in python[6].

[5] `https://github.com/INM-6/SPADE_analysis`

[6] `https://github.com/NeuralEnsemble/elephant`

Chapter 5

Review of Methods for Identification of Spike Patterns in Massively Parallel Spike Trains

5.1 Introduction

In the previous chapters SPADE has been extended for detection higher order correlation expressed by Spatio-Temporal Patterns with an explicit temporal dimension. However, over the last years also several other methods to detect correlations in MPST have been developed. Each of them has specific assumptions and aim to highlight different aspect of the correlated activity. In this chapter we review such analysis approaches. We omit methods that are either not suitable for MPST data, or that reduce their attention to patterns driven by external stimuli. We identify in particular two classes of methods for the analysis of temporally precise spike correlations. The first class consists of methods that analyze what we call population correlation, i.e. correlation that manifests at the level of the (full) population of neurons being examined, and does not (necessarily) involve specific cell assemblies. The second class consists of methods designed to identify specific cell assemblies that produce specific types of STPs. In total, we discuss and compare nine methods (four of the first class, five of the second class).

The outline of the chapter is as follows. Section 5.2 introduces different models of correlations in parallel spike trains. Section 5.3 describes the methods for correlation analysis considered here, with a focus on clarifying their assumptions and detailing for which type of correlation they were designed to detect. Section 5.4 compares the considered methods in terms of their sensitivity to the different correlation models, and discusses their ability to reconstruct (entirely or partially) those correlation structures. A perspective on new research avenues that the combination of different methods opens is given in section 5.5. For consistency we also describe again briefly the functioning of SPADE, in order to compare it with the other methods.

5.2 Models for parallel correlated spike trains

Temporal coding has been associated to different (but not necessarily incompatible) forms of spike correlation at fine temporal scale, i.e. with ms precision. These can range from synchronization of always different cell groups, to spike sequences from specific neurons in a specific temporal order, to sequences of synchronous activity. Each method considered in this paper was designed to determine the presence of one such correlation structure in MPST data. Hence, it is first necessary to introduce the respective correlation models, and to highlight their similarities and differences. This section presents five different types of fine temporally correlated spiking activities that have been associated to mechanisms of temporal coding in the literature, either in theoretical or in experimental studies. Additionally to the heuristic description provided in this section, in section 5.2.6 we define formally a Point Process framework that can be used to model and generate artificial data for the different correlation structures here introduced.

5.2.1 Population synchronization

Population synchronization refers to spiking activity where some of (or all) the neurons observed emit synchronous spikes, repeatedly over time. The neurons involved are not hypothesized to be always the same, although they may. For this reason, methods designed to detect the presence of synchronization at the population level do not need to look for and to assess the statistical significance of a multitude of different spike patterns.

This fact per se does not exclude the presence of specific cell assemblies in the data being recorded. A neural network model that contains cell assemblies and may or may not produce repeated spike patterns, depending on the model parameters, is the synfire chain. A synfire chain is a network with a high convergent and divergent connectivity from one layer of neurons to the next (Abeles, 1991). The network exhibits synchronous spiking activity that, triggered by stimulation of the first layer, propagates through the next layers. The propagation is robust to noise (Diesmann et al., 1999). However, the latter study also showed that the composition of the active neurons may vary at each run, depending on the layer connectivity and its strength. If so, recordings from neurons in the same layer would contain different, although possibly overlapping, synchronous spike patterns (see Figure 5.1C). From a statistical perspective, a probabilistic model of parallel spike trains able to generate different but overlapping synchronous spike patterns, and often used for method validation to generate ground truth data, is the multiple interaction model (Kuhn et al., 2002, 2003).

5.2.2 Pairwise synchronization

In the 1960s researchers first started to look into correlations between spike trains with the idea that correlated neurons reflect functional correlation. Gerstein and Clark (1964) and Perkel et al. (1967) developed the cross-correlation analysis to detect correlations between two parallel spike trains beyond trivial effects like stimulus dependent rate increase. Many other studies then followed, a large collection of which is found in the book by Eggermont (1990).

In pairwise synchronization pairs of neurons synchronize their spikes independent of each other. Thus, higher-order correlations are absent. Patterns of size 3 or larger are still possible, however, only as the result of chance simultaneous spike emissions from individual neurons or neuron pairs. This type of spiking activity is shown in Figure 5.1A.

In studies concerned with the analysis of spike correlations, there was and there still is a focus on pairwise analysis in the field (Riehle et al., 1997a; Kilavik et al., 2009; Vaadia et al., 1995; Zandvakili and Kohn, 2015). The reason is not that the theory would predict pairwise correlations only (see, e.g., Abeles, 1982), but rather the simplicity of such analyses over that of higher-order correlations. Nevertheless, the study of purely pairwise correlations may reveal, in MPST data, interesting dependence structures that hint to larger interacting groups of neurons, cross-area interactions or spatial interactions. Some of the methods reviewed in this paper analyze pairwise correlations for statistical significance, and then group them into larger groups of interacting neurons.

5.2.3 Synchronous Spike Patterns

A neuron receiving synchronous synaptic inputs is more likely to emit a spike than asynchronously arriving inputs, as predicted by theory (Abeles, 1982; König et al., 1996; Fries, 2005; Schultze-Kraft et al., 2013) and shown in experiments (Ashida et al., 2016). This observation led to hypothesize that neurons synchronize their activities beyond pairs. To investigate this hypothesis in real data by statistical testing, as well as to generate synthetic data for method validation, probabilistic models of parallel spike trains including higher than pairwise synchronization were formulated. Two examples are the single interaction model by Kuhn et al. (2002), and the maximum entropy model by Schneidman et al. (2003). A realization of the single interaction process where a synchronous spike pattern has multiple occurrences is shown in Figure 5.1B.

5.2.4 Spatio-temporal Patterns

Spike synchrony can be generalized by adding a temporal dimension to the correlation: the neurons involved in the coordinated activity do not necessarily spike synchronously, but in specific temporal sequences with fixed (up to a given precision) delays between consecutive spikes (see Figure 5.1E). This type of activity is generally referred to as a spatio-temporal pattern (STP; Prut et al., 1998).

STPs may be the results of the large variability of conduction delays observed in cortical network (see e.g. Swadlow, 1994), and may arise in different network models. For instance, a synfire chain produced STPs where neurons in the same layer of the chain fire synchronously, while neurons belonging to different layers fire at fixed delays. If one would record only one neuron per layer, the STP would reduce to asynchronous spikes with fixed delays. Another model that generates STPs is the synfire braid (Bienenstock, 1995), also called polychrony model (Izhikevich, 2006). It is a generalization of the synfire chain, in which spikes produced in one layer arrive at the next layer at different times due to different propagation delays. Various methods have been developed to extract STPs from a small number of parallel spike trains (Dayhoff and Gerstein, 1983; Prut et al., 1998; Abeles and Gerstein, 1988) , and these methods retrieved statistically significant STPs in experimental data (see, e.g., Prut et al., 1998).

5.2.5 Sequences of synchronous spike events

A specific type of temporal correlation that features spike synchronization and temporal propagation is represented by sequences of synchronous events (SSEs). These consist of multiple synchronous events, each involving a specific group of neurons, occurring at a fixed temporal delay one after another. Parallel recordings from multiple layers of an active synfire chain would for instance exhibit such spike patterns (Schrader et al., 2008; Gerstein et al., 2012). The sets of neurons involved in different synchronous events may or may not overlap. A realization of one specific SSE occurring two times is shown in Figure 5.1E.

5.2.6 Point Processes models for Correlated Spike Trains

The correlated parallel spike trains models described in section 5.2 can be formalized as Marked Point Processes (MPP). We can consider an MPP $M(t)$ where the k-th event is marked with a random variable P_k, which corresponds to one subset of the set $\{1, \dots, n\}$, where n is to the total number of neurons. P_k consists of the set of indexes of the neurons that fires simultaneously at that point in time. The firing times of the i-th neuron can be described by the counting process $X_i(t) = N_i(t) + M_i(t)$ where $N_i(t)$ counts the individual spikes of neuron i occurring with background rate λ_i and $M_i(t) = M(t) \cdot \mathbb{1}_{\{i \in P_k\}}$ gives the spikes due to synchronous activity. If $M(t)$ and all $N_i(t)$ are independent Poisson processes then the population process $Z(t) = \sum_{i=1}^{n} X_i(t)$ (sum of the spikes of all the neurons) is a compound Poisson process CPP$(N(t), A_k)$, where $N(t)$ counts the total number of events until time t and A_k indicates the number of synchronous spikes in the k-th event, that is, either one if the event is a single neuron firing and otherwise the size of the corresponding set S_k . The complexity of the process is the size of the largest possible set.

Using this framework, it is possible to derive a formal model for each of the correlation structures introduced in section 5.2.

The population synchronization corresponds exactly to the model just introduced, where P_k can correspond to any possible subset of the set $\{1, \dots, n\}$. The pairwise synchronization can be modeled by considering P_k assuming all and only values in the set of possible pairs (i, j) with $i, j \in \{1, \dots, n\}$. For the synchronous pattern P_k is fixed for every event and it represents the set of the indexes of the neurons involved in the pattern. For the spatio-temporal pattern we need to additionally define a set of delays $\{\delta_0, \dots, \delta_\xi\}$ that represents the lags between the first and each of the other spikes forming the pattern, such that $M_i(t) = (M(t) + \delta_i) \cdot \mathbb{1}_{\{i \in A_k\}}$. Similarly for the SSEs we can define the neuronal sets forming the successive layers of synchronous spikes $\{L_1, \dots, L_l\}$ and their respective delays $\{\delta_0, \dots, \delta_l\}$, such that $P_k = \{L_1 \cup \dots \cup L_l\}$in order to define the sequences by $M_i(t) = (M(t) + \delta_j) \cdot \mathbb{1}_{\{i \in L_j\}}$.

5.3 Higher-order correlation analysis methods

In this section we summarize existing statistical methods for the detection of higher-order correlations in MPST data. We give a short description of these methods, highlighting their features and limitations, in particular with regard to how they deal with different properties of uncorrelated background activity in the data. For details we refer to the original publications.

Generally, two classes of methods can be distinguished, which investigate different aspects of spike correlation. The first classaim to identify the correlation order (number of neurons involved) rather than the identity of the neurons involved. Thus, each correlated event may involve a random subset of neurons or may be composed of a specific, always identical group. We refer to the correlation type underlying this analysis class as *population synchronization*. The other class of analysis methods assumes a correlation model in which the correlated neurons form stereotypical synchronous spike events or temporal sequences

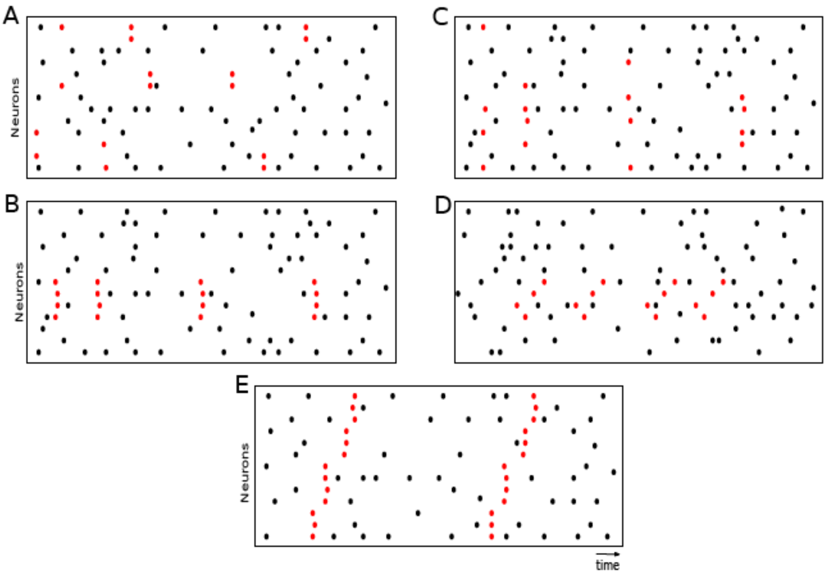

Figure 5.1: **Raster plots of different correlation types.** Each panel shows the spiking activity of parallel spike trains (one neuron per row) over time (horizontal axis). Each dot represents a spike; the red dots in particular represent spike belonging to a spike pattern. Different panels refer to different forms of temporal spike correlation. **A)** Pairwise correlation model. The population contains 6 pairs of synchronized neurons (the latter indexed from bottom to top): $(1, 2)$, $(1, 3)$, $(2, 4)$, $(8, 9)$, $(8, 14)$, $(13, 14)$. **B)** Synchronous spike patterns. Neurons $4, 5, 6, 7$ are repeatedly involved in the pattern. **C)** Differently from the spike patterns in panel A, the neurons involved in each synchronous event are randomly selected and change from one event to the next. **D)** Spatio-temporal patterns. The red spikes correspond to occurrences of an STP. The neurons involved in the patterns are $4, 5, 6, 7$, as in panel A, but their spikes occur now in a fixed temporal succession with fixed delays. **E)** Sequences of synchronous spike events. Two occurrences of the same SSE are shown. Here all observed neurons are involved, and groups of 4-4-4-3 synchronously firing neurons fire in short succession.

of spikes. We refer to these events as *spike patterns.* The aim of these methods is to retrieve the neuronal composition and the occurrence times of the spike patterns.

5.3.1 Methods to detect population synchronization

One of the challenges in the statistical assessment of synchronous spike events in MPST data is posed by the exponential growth of the number of possible patterns with the number of neurons being considered. However, this problem can be simplified if the research interest lies solely on assessing the presence and the order of excess (i.e. above-chance) synchronization, without resolving the specific neuron identities involved. Also, the data may contain patterns of synchronous spikes that change their neuronal composition each time, so that the correlation is distributed possibly across the full population being observed. We refer to synchrony which does not (or which is not assumed to) involve specific subgroups of neurons in the observed population as population synchronization.

Most methods for population synchronization analysis reduce the spike data to the number of active neurons (i.e. spikes) observed at any time bin. A spike train i is fully described by its spike times and, . given a time discretization in small temporal bins, we can define the population histogram as the count of spikes that occurred in the same time bin. The maximum possible count fo the histogram is thus the number N of neurons. The first three methods presented here are based on statistics derived from the population histogram. They were developed in succession, each to overcome the limitations of the previous one. The first method, the Complexity Distribution (CD) analysis (Grün et al., 2008), proposes a simple statistical approach purely based on the distribution of the entries of the population histogram. It compares such an empirically derived distribution to that expected from neurons firing independently to determine the presence of excess synchronization. The second method, the CUmulant Based Inference of Correlation (CuBIC, Staude et al., 2010b), derives the null distribution analytically under more specific assumptions about the data, and infers the minimum correlation order existent in the data. The third method, the Population Unitary Event (PUE, Rostami (2017)) analysis, works under the same assumptions as CuBIC, but uses a different test statistic which enhances the statistical power of the test, thereby requiring samples of smaller size for a correct identification of excess synchrony and thus also enabling a time resolved analysis.

The fourth method, called here the correlation information index (CII), is an approach originally suggested by Schneidman et al. (2006) as a way to condense the information delivered by maximum entropy models built on parallel spike train data to a single scalar. The method accounts for the neuronal identity of each spike in the observed synchronous patterns, and builds a full probabilistic model of those. This model is used to obtain a single scalar, the CII, that quantifies the amount of surplus of information contained in the data which is delivered by correlations of a given order.

Complexity Distribution (CD)

The value taken by each entry in the population histogram is called the bin complexity. Each synchronous spike event increases the empirical complexity in the bin of it's occurrence as compared to the scenario of independent spiking. Therefore, it also increases the value of the empirical complexity distribution at that complexity value. Grün et al. (2008) developed a method that tests for spike train independence based on the difference

	Method	Target correlations	Null model	Alternative model	Ref
Population Synchronization	CD	Population Synchronization	Independent spike trains	Population synchronization	Grün et al. (2008) Louis et al. (2010a)
	CUBIC	Population Synchronization	v Population Synchronization of order ξ	Population synchronization of order $\xi + 1$	Staude et al. (2010b) Staude et al. (2010a)
	PUE	Population synchronization	Population synchronization of order ξ	Population synchronization of order $\xi + 1$	Rostami (2017)
	CII	Population Synchronization	Maximum entropy model of order ξ	Maximum entropy model of order $\xi + 1$	Schneidman et al. (2003)Schneidman et al. (2006)
Spike Patterns	MEM	Synchronous Spike Patterns	Maximum entropy model of order ξ	Synchronous spike patterns	Schneidman et al. (2003)Schneidman et al. (2006) Shimazaki et al. (2012)Kelly and Kass (2012)
	GIC	Synchronous Spike Patterns	Independent	Pairwise synchrony	Berger et al. (2010a)
	SPADE	Synchronous and Spatio-temporal Spike Patterns	Independent	Synchronous pattern, Spatio-temporal patterns	Borgelt (2012) Torre et al. (2013) Quaglio et al. (2017)
	CAD	Synchronous and Spatio-temporal Spike Patterns	Poisson Independent	Synchronous pattern, Spatio-temporal patterns	Russo and Durstewitz (2017)
	ASSET	Sequences of Synchronous Events	Poisson Independent	SSEs	Schrader et al. (2008) Gerstein et al. (2012) Torre et al. (2016a)

Table 5.1: **Table of analysis methods, their assumptions, and related references.** The table summarizes the methods that we discuss here and their assumed data models (column 2 from left, all introduced in Section 2). Columns 3 and 4 describe the assumed null and the alternative hypothesis, respectively. Column 5 lists the publications in which each method has been introduced or further developed.

between the empirical complexity distribution and the null distribution. Excess synchrony causes the difference between the two distributions to have a positive bump at larger complexities. Due to the conservation of the total probability mass, a negative bump appears at lower complexities. Depending on the assumptions about the spiking behavior of each neuron, the null distribution may be available analytically (e.g. by assuming that the spike trains are stationary Poisson processes) or be approximated by Monte Carlo surrogate techniques (see Grün, 2009; Louis et al., 2010c,a), and will in general depend on the statistics of each spike train as well as on the chosen bin size.

An example of artificial test data is illustrated in Figure 5.2, modified from Grün et al. (2008). Panel A, top, shows data from a stochastic simulation of 100 parallel spike trains, 80 of which are independent Poisson. The first 20 neurons exhibit, in addition to independent spiking activity, also synchronous firing events. The synchronous events are hardly visible by eye in the raster plots if the neuron ids on the vertical axis are sorted randomly (Panel A, middle), but can be retrieved in the population histogram (Panel A, bottom; bin size: 1 ms), although with a loss of information about the involved neurons. Panel B shows the empirical complexity distribution (top), the null distribution computed by randomizing the spike times of each neuron (middle), and the difference between the two distributions (bottom). The latter contains a visible bump centered at complexity $\hat{\xi} = 22$. Importantly, the bump is right-skewed and is centered to the right of the true synchronization order $\xi = 20$. The reason for the offset in the peak is that the inserted synchronous events of fixed size ξ overlap with background activity from the other neurons, resulting in a higher total complexity. The bin width w determines the statistics of the random component of the total count.

Under the assumption that all spike trains are Poisson processes with identical firing rates, the null distribution can be computed analytically based on combinations of Binomial distributions (Grün et al., 2008; Figure 5.2B, solid). Otherwise it can be computed by surrogates, e.g. by spike time dithering (Figure 5.2B, dots). Confidence intervals are computed analogously, and allow to accept or reject the null hypothesis of independence (Louis et al., 2010a). Varying the bin size enables to determine the temporal jitter inherent to the synchronous events (for details, see Louis et al., 2010c).

CUmulant Based Inference of Correlation (CuBIC)

The complexity distribution method discussed above visualizes correlations among parallel spike trains. The CUmulant Based Inference of Correlation (CuBIC; Staude et al., 2010b) advances this technique by relaxing the hypothesis of independence and testing for the presence of correlations of progressively higher-order, given those of lower order observed in the data.

CuBIC comprises the following steps. Starting from $\xi = 1$ (spike train independence), it assesses whether peaks in the complexity distribution of the data could be explained entirely by assuming correlations of order at most ξ. If that is not the case, the method accepts the alternative hypothesis that correlations of order $\xi + 1$ or higher must exist. It then tests for correlations of order $\xi + 1$ against those of order $\xi + 2$ or higher, and so on. The procedure stops as soon as a value $\hat{\xi}$ is accepted. $\hat{\xi}$ is interpreted as the minimum order of population synchronization that has to be assumed to explain the observed amount of synchronous events in the data. The sequence $(p_1, \ldots, p_{\hat{\xi}})$ of test p-values is guaranteed to increase, because synchronous events of higher complexities correspond to higher expected

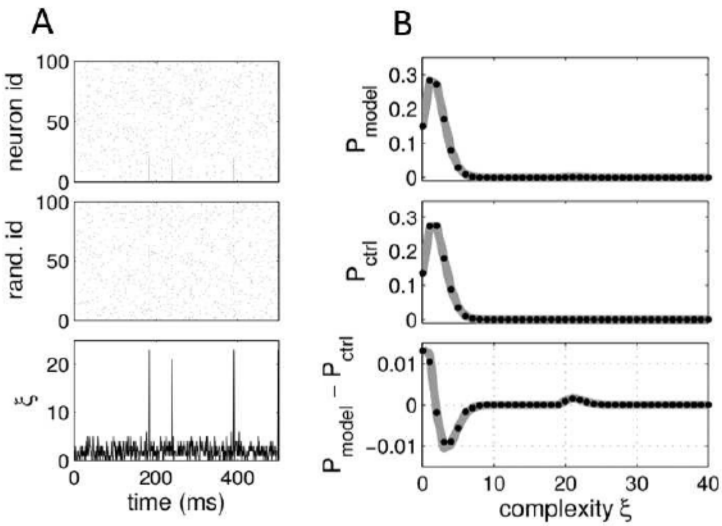

Figure 5.2: **Complexity distribution based correlation identification. A)** *Top:* Parallel spike trains comprising a synchronous spike events among the first 20 neurons, firing in synchrony with a rate of $\lambda_c = 5$ 1/s, plus 80 independent neurons. *Middle:* randomization of the neuron ids (vertical axis) of the top panel. *Bottom*: population histogram of the data (bin width: $w = 1$ ms). **B)** *Top:* Complexity distribution of the data in A. *Middle*: null distribution obtained analytically (solid line) or by surrogates through spike time randomization in time (dots). *Bottom*: difference between the observed and the null complexity distributions. Figure modified with permissions from Grün et al. (2008).

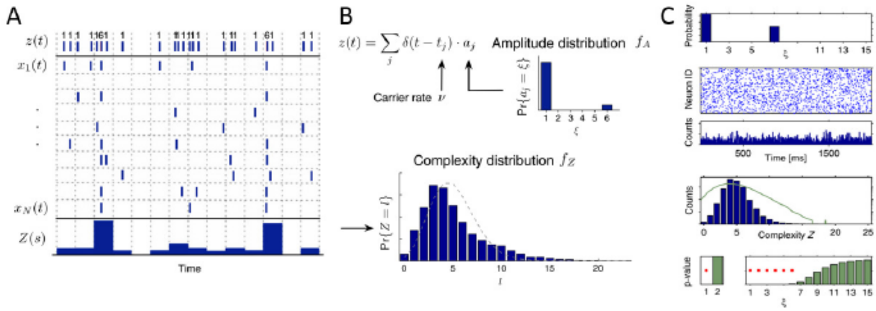

Figure 5.3: **CuBIC analysis. A-B)** Illustration of the generation of correlated parallel spike trains using a marked point process. This process is the null model used to test $\xi = 6$. Spikes are assumed to be copied from a hidden process $z(t)$, (A, top) consisting of spike times t_i drawn from a Poisson process and associated labels a_j drawn from the amplitude distribution f_A (panel B, top). Each spike t_i in the hidden process is copied into a_j spike trains, randomly selected each time from the full population $x_1, \ldots, x_N$. The population histogram Z (A, bottom) is computed by segmenting the time axis into consecutive bins of a few ms. The complexity distribution f_Z (B, bottom panel) is derived from Z. **C)**Application of CuBIC to simulated correlated data. The Figure shows, from top to bottom: the amplitude distribution used to generate the correlated data, the raster plot of the generated data, the derived population histogram, the empirical complexity distribution (blue) and its logarithmic transform (green), the test p-values for different orders of correlation tested by CuBIC. Figure modified with permissions from Staude et al. (2010b).

correlations, and thus eventually exceeds the selected significance threshold α, terminating the procedure (see last row of panel C in Figure 5.3).

The test p-value is obtained analytically in the limit of a large number L of i.i.d. observations (time bins), and by assuming that all spike trains are Poisson processes. In particular, the case $\xi = 1$ corresponds to the assumption that the spike trains are independent. The case $\xi > 1$ corresponds to the assumption that up to ξ neurons synchronize their spikes with positive probability. The probability of synchronous events of a given size is modelled by the so-called amplitude distribution (see panel B in 5.3). The analytical formulation makes CuBIC computationally inexpensive, but requires L to be large enough (according to Staude et al., 2010b, $L \geq 10^5$ bins) to get reliable results. The analysis is therefore limited to applications of relatively long and stationary data. The length of the data required does not enable the method to reveal changes of the correlation order over time. While the original publication developed the method for stationary data, generalizations had been later on provided for populations of spike trains with specific firing rate distributions, such as Gamma or uniform distributions (Staude et al., 2010a) or non-stationary processes (Reimer et al., 2012).

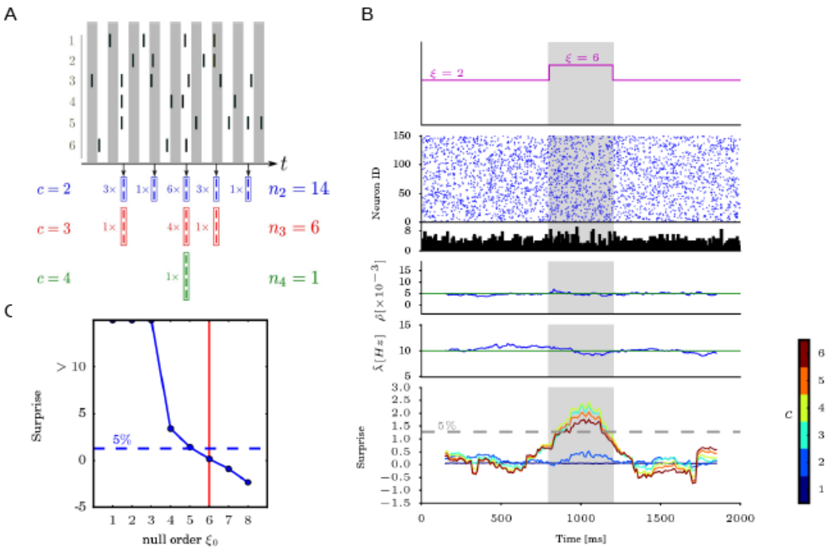

Figure 5.4: **Population Unitary Event analysis. A)** Top: raster plot of 6 neurons firing over time. Bottom: illustrative example of the test statistic n_c, showing for given values of c ($c = 2$ blue, $c = 3$ red, $c = 4$ green) and for each bin of the population histogram containing $Z_k \geq 2$ spikes, the number $\binom{Z_k}{c}$ of patterns of size c that can be extracted from the Z_k spikes. **B)** Time resolved PUE analysis applied to simulated data. The simulated data consist of $N = 150$ parallel spike trains with duration $T = 2000\,\text{ms}$, generated as a realization of a correlated Poisson process of order $\xi = 6$ in the time window $[800\,\text{ms}, 1200\,\text{ms}]$ (indicated in gray), and of order $\xi = 2$ elsewhere. The firing rate of each individual spike train is set to $10\,\text{Hz}$ and the pairwise correlation coefficient to 0.005. The data are analyzed with PUE varying the hyperparameter c from 1 to 6. From top to bottom: Time course of synchrony order ξ, raster plot of the data, population histogram (1 ms bin size), cross-neuron average of empirical pairwise correlation coefficients calculated over a 300 ms sliding window, average firing rate estimated over the same sliding window, and surprise measure of the PUE statistic using different values of the parameter c. The surprise is calculated for each time window with null order $\xi_0 = 2$. Different colors correspond to different values of c. The gray dashed line indicates the 5% significance level. **C)** Estimation of the synchronization order ξ in the central analysis window (highlighted in gray in panel B), for a null order ξ_0 increasing from 1 to 8. The data are obtained by concatenating 15 model realizations generated as explained in B, and mimicking identically distributed experimental trials. The blue dashed line indicates the 5% significance level and the vertical red line shows the true synchronization order $\xi = 6$. Figure modified with permissions from Rostami (2017).

Population Unitary Event (PUE)

As mentioned in the previous section, CuBIC is limited in its application to long stretches of stationary data. However, experimental results regarding pairwise correlation analysis using the unitary events analysis method (Riehle et al., 1997b; Kilavik et al., 2009) revealed that excess synchronization may appear dynamically and is related to behavior. Thus, time resolved analysis methods for detection higher-order correlation are required. The population unitary event (PUE) analysis method is designed to enable that. The test statistic of PUE is the number of synchronous spike events of a given size c observed in the data, which is extracted from the population histogram. For bins containing spike counts Z, we consider the total number of possible constellations of c from Z, thus $\binom{Z_k}{c}$ per bin k. Thus in a total of L bins we derive the number of synchronous spike events of size c according to $n_c = \sum_{k=1}^{L} \left(\begin{array}{c} Z_k \\ c \end{array} \right) = \sum_{k=1}^{L} \frac{Z_k!}{c!(Z_k-c)!}$. For example, as shown in Figure 5.3.1A, n_2, n_3 and n_4 are the total number of distinct pairwise, triple-wise and quadruple-wise synchronous spike events present in the data.

The PUE analysis exploits the following framework for testing the significance of the empirical test statistic observed in the empirical data. The null hypothesis of the PUE method is defined by a presumed order of synchrony among the spike trains, i.e. the null order ξ_0, and assumes that the order of synchrony among the given spike trains is at most the null order ξ_0. The null distribution and the associated test p-value are computed numerically by a Monte Carlo simulation by realizing a marked Poisson process (see Ehm et al., 2007; Staude et al., 2010b) used to model a multidimensional correlated Poisson process (as also assumed and introduced in CuBIC, section 5.3.1). The parameters for the null-model are adapted by the firing rate and the pairwise correlation parameters extracted from the data (see details in Rostami, 2017; Staude et al., 2010b).

The analysis can be performed in a time-resolved fashion by sliding a window through the data in steps of a few time bins, and by analyzing each time window separately. As shown in Figure 5.3.1B, the surprise measure, defined as a logarithmic transformation of the p-value (Palm, 1981), becomes significant when the analysis window overlaps with the synchronization period. This enables a time resolved analysis which is able to discover changes in the correlation order over time.

When multiple experimental trials are available, the PUE method may pool data from different trials to achieve increased statistical power, under the assumption of cross-trial stationarity. Figure 5.3.1C shows an example where the order of synchrony is inferred by PUE using all15 trials. By computing the surprise as a function of the null order ξ_0, the estimate $\hat{\xi}$ of true order of synchrony in the data can be obtained as the lowest value of the null order ξ_0 for which the surprise measure is not significant. PUE has higher statistical power (and therefore needs less evidence) than CuBIC to detect existing correlations in data.

Correlation information index (CII)

Maximum entropy models (MEMs) have been introduced to evaluate the occurrence probability of each synchronous spike pattern (seen as a binary sequence of on/off states) given the observed firing rates, pairwise correlations, and possibly higher-order moments of a population of observed neurons. Once a maximum entropy distribution accounting for all and only the observed correlations up to a given order ξ is inferred from data (see section

5.3.2 for more details), the amount of information delivered by such correlations can be quantified as follows.

The larger is the order ξ of the moments one accounts for to construct the maximum entropy distribution, the smaller is the total entropy $\mathcal{H}_\xi$ of the maximum entropy distribution (that is, its uncertainty). At one extreme ($\xi = 1$, only average firing rates being considered) one gets the uniform distribution, where the probability of a state is proportional to the product of the firing rates of the "on" neurons. At the other extreme ($\xi = N$, where N is the number of neurons) one gets the empirical distribution. The entropy $\mathcal{H}_\xi$ of the maximum entropy distribution constrained on all moments up to order ξ decreases, for ξ increasing from $\xi = 1$ to $\xi = N$, from $\mathcal{H}_1$ to $\mathcal{H}_N$. The difference $\mathcal{H}_1 - \mathcal{H}_\xi$ quantifies the reduction of the entropy (i.e., of the uncertainty about all possible states) due to the knowledge of all correlations of order 2 to ξ, that is, the amount of information conveyed by those correlations. The difference $\mathcal{H}_1 - \mathcal{H}_N$ quantifies the total information delivered by correlations of any order. Thus, the ratio

$$R_\xi = \frac{\mathcal{H}_1 - \mathcal{H}_\xi}{\mathcal{H}_1 - \mathcal{H}_N}$$

characterizes the portion of the total correlation information delivered by correlations of order 2 to ξ. This measure is called here the correlation information index (CII). R_2 was suggested by Schneidman et al. (2003) to assess whether or not triple- or higher-order correlations play a role in information processing in the nervous system. Schneidman et al. (2006), Shlens et al. (2006) and Tang et al. (2008), among others, applied this measure to data from the retina and from various cortical areas, reporting values ranging from 0.85 to over 0.95. Based on these high values, they concluded that higher-order correlations were negligible in the examined data. It should be noted, nevertheless, that even for extremely high values of R_2 ($R_2 > 0.99$), highly statistically significant spike patterns of 3 or more neurons may be present in the data (Torre, 2016). In addition, Roudi et al. (2009) showed in a theoretical study that conclusions obtained from MEMs built on data from few neurons cannot be extrapolated to larger samples of parallel spike data. Nevertheless, this approach may be helpful to quantify the amount of information present in parallel spike train data which is delivered by correlations of a certain order.

5.3.2 Methods for spike pattern detection

The second group of methods covered in this review is designed to detect groups of neurons involved in millisecond-precise spiking patterns. These methods achieve this goal by detecting spike patterns that repeat sufficiently many times to be classified as non-chance patterns. Non-chance patterns are considered a signature of assembly activation Abeles (1991), and have been associated to behavior in several experimental studies (Prut et al., 1998). The large number of possible patterns in large scale recordings often poses non-trivial computational and statistical problems. To get a flavor of this problem, consider a population of N neurons recorded in parallel. These neurons may organize their activity in up to 2^N different patterns of synchronous spikes, which is close to 10^{30} for $N = 100$, as regularly available in modern extracellular recordings. This number increases by orders of magnitude if arbitrary STPs, and not only synchronous events, are considered. Without any previous knowledge about the neurons possibly involved in the correlation, a blind search for patterns occurring more than expected under some null hypothesis has to be performed, accounting for all these possibilities. The computational burden may be excessive (even allocation of the occurrence counts of all possible patterns to memory may be impossible). Besides, testing all patterns individually for statistical significance would yield insurmountable multiple testing issues. Finally, the amount of data needed to collect adequate statistical evidence would be immense, and most likely unavailable. The methods considered here have been developed to address these issues. We specifically restrict our attention to methods that can be applied to large scale recordings, and whose ability to discover existing patterns has been demonstrated on simulated data. Also, we disregard those methods that search only for patterns temporally locked to some stimulation. A recent review of the latter can be found in Levakova et al. (2015).

Maximum entropy models (MEM)

As mentioned already in section 5.3.1, MEMs provide the possibility to assess the likelihood of specific spike patterns based not only on the average neuronal firing rates, but also on the observed second and higher-order correlations. As shown in Figure 5.3.2, a MEM of order ξ converts spike trains to binary sequences by binning, computes the average zero-lag correlations up to order ξ (the vector of average firing rates, the matrix of second order correlation coefficients, the tensor of third order correlations, and so on), and then provides an analytical estimate of the p-value of any spike pattern under these constraints (and under the additional assumption that the spike trains are Poisson). A joint distribution of N binary states (on/off neurons) is fully specified if and only if all multivariate moments up to order N are given. MEMs specify only the correlations up to an order $\xi < N$, and then determine the maximum entropy (the least assertive) distribution among all the distributions compatible with the given constraints (see Jaynes, 1957).

By constraining the distribution to correlations up to a given order, the presence of "genuine" higher-order correlations (that is, of correlations that are not expected based solely on the observed lower order correlations) can be ascertained. The analytical treatment provides an efficient way to analyze data from relatively large parallel recordings. This methodology has been used in a number of studies to search for statistically significant synchronous spike patterns, constraining on the observed average neuronal firing rates and average pairwise correlations ($\xi = 2$) (see, e.g., Schneidman et al., 2006; Tkacik et al., 2006; Tang et al., 2008). Shimazaki et al. (2012) extended the method to account for time varying interactions. Kass et al. (2011) and Kelly and Kass (2012) incorporated in the null hypothesis history effects that make the spike trains deviate from the Poisson assumption.

Despite these efforts, a number of short-comings limits the applicability of MEMs to MPST data. First, the maximum entropy distribution among a large number of neurons is computationally demanding to evaluate due to the large number of parameters to be determined. This is even the more so if non-stationarities are taken into account, which is necessary in most applications to avoid biased statistics. Second, evaluating the p-value of each pattern individually leads in MPST data to multiple testing issues, resulting in excessive false positives (or false negatives after standard statistical corrections like e.g. the Bonferroni correction). Third, Rostami et al. (2017) studied in detail the aptness of MEMs in application to MPST data and showed that MEMs predict a bimodal distribution for the population-averaged activity, when it is applied to typical experimental recordings of 150 or more neurons. Thus the MEM distribution is not uniquely predicted, but switches

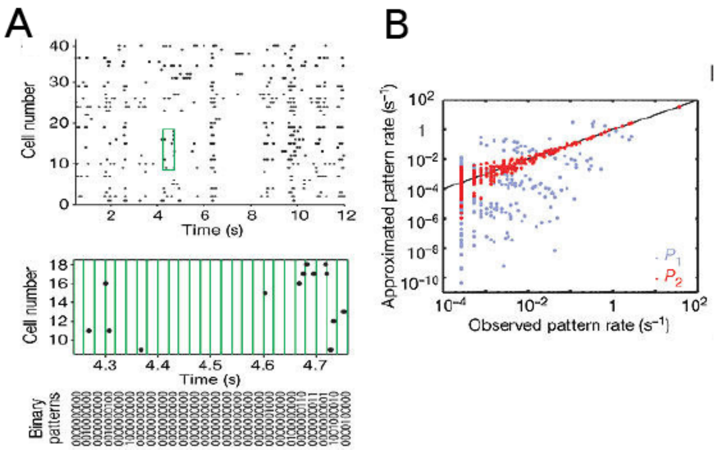

Figure 5.5: **Maximum entropy models. A)** A segment of the simultaneous responses of 40 ganglion cells in the salamander retina to a natural movie clip (top panel). Discretization of parallel spike trains into binary patterns is shown below (green). The binary vectors describe the joint activity patterns of the cells at a given time point. For clarity, 10 out of 40 cells are shown (bottom panel). **B)** Using the same group of 10 cells from panel **A**, the rate of occurrence of each firing pattern predicted from a maximum entropy model P_2 that takes into account all pairwise correlations is plotted against the pattern rate measured in the recorded data (red dots). For comparison, the independent model P_1 is also plotted in grey. The black line indicates equality. Figure modified with permissions from Schneidman et al. (2006).

between different states of activities for long data sets. For these reasons, the MEM model does not easily scale to data of large populations of neurons. Nevertheless, for smaller data sets (up to $30-40$ parallel recordings) MEMs provide a valuable tool to analyze genuine higher-order synchronous events.

Neuronal cliques and Groups of Intra-Correlated cliques (GIC)

A first approach to analyze MPST data for the presence of cell assemblies of possibly large size involved in correlated activity is the Groups of Intracorrelated Cliques (GIC) analysis, developed by Berger et al. (2007). The method first determines pairs of correlated neurons using the cross-correlation histogram (CCH; Perkel et al., 1967), then groups overlapping pairs into larger groups which possibly indicate higher-order interactions.

The CCH between a reference and a target neuron is a histogram whose entries count, for any positive or negative temporal delay Δt, the number of spikes that the target neuron emits with delay Δt from any one spike of the reference neuron. If the target neuron tends to fire with delay Δt before the reference (Δt negative) or after it (Δt positive), a

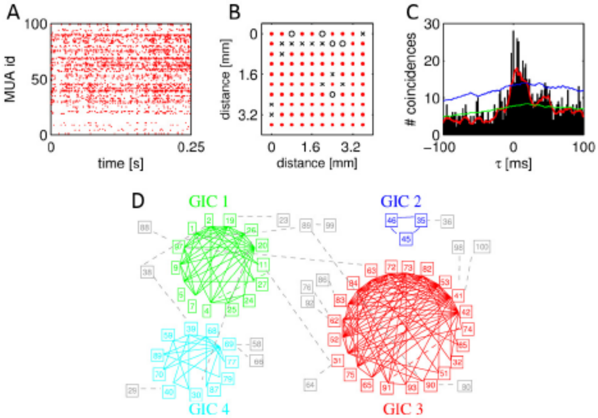

Figure 5.6: **Cliques of pairwise correlated spike data. A)** Raster display of 84 simultaneously recorded multi-unit (MUA) spike trains, i.e. spikes of the same train were not sorted into single neurons. Some of the electrodes did not record any data, thus the corresponding line is empty. **B)** Arrangement of the 100 electrode recording array (Utah array). Empty circles mark electrodes that were not connected, electrodes marked with a cross did not work. The rest (red dots) indicate working electrodes, from which the data in panel A were recorded from. The non-diagonal next electrode distance was $L = 400\,\mu m$. **C.** Example CCH of two multi-unit spike trains. Black: cross-correlation of the recorded data, bin width: 1ms, red line: smoothed CCH (rectangular kernel of 10ms width). The green line shows the bin-wise average CCHs of the surrogate data (100 repetitions) generated by spike dithering with ± 35 ms, and smoothed as the original CCH. It represent the expected CCH under the null hypothesis. The blue line indicates $+2$ std of the bin-wise entries of the surrogate CCHs. A pair of MUAs is considered significantly correlated if the smoothed original CCH (red) exceeded at or around $\tau = 0$ this significance level. **D)** Clusters of cliques of significantly correlated pairs of MUAs. A significantly correlated MUA pair is represented by two nodes (each MUA id is shown in the respective rectangle). Groups of 3 or more all-to-all correlated MUAs are clustered into cliques. Cliques sharing at least one node are further combined into a group of intracorrelated cliques (GIC). The resulting four clusters are marked in different colors (red, green, cyan and blue). MUAs not fulfilling these criteria are marked by gray squares, connected by dashed lines to the other MUA they are correlated with. Figure modified with permissions from Berger et al. (2007).

peak in the CCH arises, centered at Δt. Other effects not related to correlated activity, such as firing rate variability and high regularity of the individual spike trains, may also cause peaks or oscillations in the CCH. Unbiased predictors of the CCH under the null hypothesis of spike train independence that account for these factors have been developed based on data surrogates (Louis et al., 2010c). For instance, a predictor accounting for both rate changes and spike regularity can be computed using a Monte Carlo approach as the average CCH obtained from surrogates of the original data generated by spike dithering. Confidence intervals can be obtained analogously.

Possible interactions among more than two spike trains are then obtained combining the information provided by the CCHs between all pairs. The proposed method works in three steps. Statistically significant pairwise correlations are determined on the basis of suitable predictors (for synchrony: at time lag $\Delta t = 0$, or slightly larger to account for jitter). Second, cliques of all-to-all correlated pairs are collected, and all cliques above a preselected minimum size (e.g., all cliques of 3 or more neurons) are retained. Third, cliques sharing at least one neuron are merged into a single GIC.

Berger et al. (2007) applied this procedure to MPST data collected from cat V1 during visual stimulation with full field flash stimuli, and found four spatially clustered, distinct GICs comprising 3 to 21 neurons (Figure 5.3.2D, each GIC shown in a different color). These GICs also formed clusters in cortical space and were speculated to reflect activity from underlying connectivity forming orientation columns as was shown by optical imaging (e.g. Hübener et al., 1997).

The method relies on the computation of the CCHs between all pairs of investigated neuronal activities and the evaluation of their statistical significance. The first amounts to $\binom{N}{2}$ pairs for N neurons, a number that grows quadratically with N. Testing each CCH for significance using a Monte Carlo approach further requires the computation of up to hundreds of surrogate CCHs. The computational burden may become unaffordable without resorting on compute clusters. For this reason, Berger et al. (2010a) worked out a pre-processing approach that excludes from the analysis individual neurons contributing weakly to synchronous events. The pre-processing step was used effectively on the same data and verified the original analysis, however at considerably reduced computational cost.

GICs formed by three or more neurons may be evidence for, but not necessarily imply, the presence of higher-than-pairwise correlation. The method does not test for genuine higher-order correlations (i.e., correlations that remain statistically significant when conditioning on correlations of lower order). The corresponding model of spiking activity is the pairwise correlated point process described in section 5.2.2. On the other hand, higher-order correlations in the data may be, but not necessarily are, found as GICs.

Cell Assembly Detection (CAD)

Russo and Durstewitz (2017) recently introduced a different method to tackle the multiple testing problem arising in the search of repeated spike patterns in MPST data. The authors suggested an agglomerative algorithm (which we refer to here as Cell Assembly Detection, or CAD) that is composed of two recursive steps: a) a statistical test for pairwise correlations, and b) a clustering procedure that agglomerates pairwise interactions into patterns of larger size. A very similar idea was introduced by Gerstein et al. (1978).

In step a), spike trains are segmented in small time bins of width w. Then, for each pair (A,B) of spike trains, the algorithm counts the number $n_{AB,\bar{l}}$ of times that one spike of spike train A is followed by a spike of spike train B after $\bar{l}$ bins. The lag $\bar{l}$ is chosen to maximize the observed joint spike count $n_{AB,\bar{l}}$. Under the null hypothesis that the spike trains are realizations of independent Poisson processes, the method then derives the null distribution of the statistic

$$n_{ABBA,\bar{l}} = n_{AB,\bar{l}} - n_{AB,-l^*},$$

where l^* is an arbitrary reference lag for which $n_{AB,\bar{l}} \geq n_{AB,l^*}$. Considering $n_{ABBA,\bar{l}}$ instead of $n_{AB,\bar{l}}$ is necessary to compensate for bias due to firing rate non-stationarity (see Figure 5.7). If $n_{ABBA,\bar{l}}$ deviates significantly from 0, then the spike train pair AB is considered to be part of the same spike pattern. The advantage of this approach is that it avoids high computational cost by deriving all p-values analytically. However, this strategy heavily relies on the assumption of Poissonianity - which may not be a feature of the data and may lead to false positives (e.g. see Pipa et al., 2013). Also, $\binom{N}{2}$ statistical tests are performed at this step in the presence of N spike trains, leading to a moderate multiple testing issue.

In step b), larger spike patterns are obtained by recursively testing patterns previously formed with any other neuron, i.e. triplets are formed by testing each single significant pair AB with any other unit C using the same framework introduced for pairs. In order to make use of the null distribution derived for pairwise testing, all spikes of A with lag $\bar{l}_{AB}$ are considered to form a new artificial unit $(AB, \bar{l}_{AB})$, representing then the pattern occurrences. The test is then performed on the pair $((AB, \bar{l}_{AB})C, \bar{l}_{(AB)C})$. By proceeding iteratively with this agglomerative procedure, the algorithm extends from pairs to patterns of any size. Thus, this approach does not explicitly test for higher-order correlations, which leads to a lower statistical power than methods testing directly for higher-order correlations (see section 5.5).

CAD can detect not only STPs, but also correlations of average spike counts (e.g. firing rate modulation). To do so the method allows the user to increase the bin size w, such that more than one spike is contained in a bin. For example in case that neuron A shows repeated increase in the firing rate, followed by an increase in neuron B after l bins (e.g. correlated non-stationary firing rates) appearing as spike count correlations in $n_{ABBA,\bar{l}}$. In particular, it is possible – in the case of multiple spikes in the same bin – to decompose each process in a sum of binary processes and to successively assess their significance using the same framework previously introduced. For additional details we refer to the original publication. Thus CAD is not limited to detect fine temporal spike pattern, but is also capable to detect correlations on a larger time scale.

Spike Patterns Detection and Evaluation (SPADE)

Spike synchrony (see section 5.2.3) or spatio-temporal spike patterns (section 5.2.4) in MPST data can be effectively detected by the Spike PAttern Detection and Evaluation (SPADE) analysis method (see Torre et al., 2013, and Quaglio et al., 2017, respectively). SPADE comprises three steps: a) a data mining procedure to efficiently extract repeating synchronous spike patterns that are suitable candidates to be significant patterns, b) statistical testing to assess the significance of the mined pattern candidates, and c) assessment the conditional significance of each pattern retained after step b), given any other found pattern overlapping with it; the last step is needed to reject patterns that are due to chance overlap of real patterns with background activity.

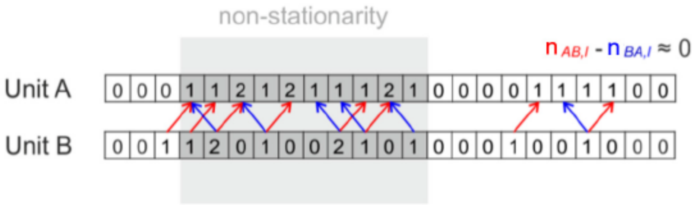

Figure 5.7: **CAD pairwise test:** Sketch of the statistical pairwise test of the CAD method. The count $n_{AB,l}$ of spikes with with a lag l is tested if it is significantly larger than the count at a reference lag. Here the reference lag is chosen equal to $-l$, which correspond to the count $n_{AB,-l} = n_{BA,l}$. Under the null hypothesis of independent Poisson processes the observable $n_{ABBA,l} := n_{AB,l} - n_{BA,l}$ has average equal to 0 also in case of firing rate non stationary firing rate. Figure modified with permissions from Russo and Durstewitz (2017).

Step a) is accomplished by Frequent Itemset Mining (FIM, Zaki and Ogihara 1998 or equivalently Formal Concept Analysis Ganter and Wille 1999; Pisková and Horváth 2013). Time is discretized into consecutive bins of duration w, and the sets of neurons emitting a spike in each bin are collected (see Figure 5.8A). The activity of a synchronous cell assembly immersed in a larger population of recorded neurons (e.g., neurons 1, 3 and 4) typically appears as a set of spikes falling in the same time bin, together with additional spikes emitted by other neurons and falling in the same bin by chance. Revealing active synchronous cell assemblies thus requires to assess the statistical significance of all subsets of all transactions. However, for N neurons the latter may be as many as 2^N different patterns, yielding severe computational and statistical issues. Of interest among these patterns are those which are *frequent*, i.e. occur at least a minimum number of times (in our case, 2 times), and which are *closed*, i.e., do not always occur as a subset of the same super-pattern. All patterns which are not frequent or not closed may be discarded under the rationale that they are either too sporadic, or trivially explained by larger patterns in the data. Figure 5.8A shows infrequent (black), frequent but not closed (blue) and frequent and closed (red) patterns extracted from the transactions in panel A. The latter are typically a small fraction of the total patterns. Therefore, testing them only for significance drastically reduces the computational burden and the multiple testing problem, without causing any information loss. FIM provides a class of efficient algorithms to collect closed frequent patterns in data of large size.

Similar approaches based on different, more heuristical data mining frameworks had been developed in previous work. See in particular Abeles and Gerstein (1988) and Gansel and Singer (2012) for two different algorithms to pre-filter patterns based on their neuronal composition. For an application of the former to MEG data, see Tal and Abeles (2016). These methods, however, do not guarantee that the filtered patterns are all closed (that is, all non-trivial) patterns in the data, thereby possibly leading to a loss of information. Also, neither of the two methodologies is accompanied by an approach to test for the statistical significance of the filtered patterns designed for MPST data.

Step b) of SPADE, called pattern spectrum filtering (PSF; see Figure 5.8B), assesses the statistical significance of each closed frequent pattern (typically thousands or more in MPST data) on the basis of the pattern size z (i.e. the number of neurons forming the pattern) and of the occurrence count c (i.e. the number of times the pattern occurs), irrespective of the specific neuronal composition of the pattern. The pair (z, c) is called the pattern *signature*. Because the number of different pattern signatures is orders of magnitude smaller than the total number of different patterns, this pooling strategy avoids the multiple testing issue that would arise from testing each closed frequent pattern individually. PSF computes the p-value of each observed signature based on surrogate data that preserve the marginal properties of the original spike trains such as the inter-spike intervals and the firing rate profiles (see Pipa et al., 2008; Louis et al., 2010c).

The presence of a real synchronous spike pattern in data tends to increase the occurrence count, and therefore the significance, of other patterns that result form a chance overlap of the pattern's spikes with background activity. Step c) of SPADE, called pattern set reduction (PSR) (see Figure 5.8C), detects and removes these false positives by assessing the conditional significance of all patterns found after step b) given any other overlapping one.

The present study and related publications (Yegenoglu et al., 2016; Quaglio et al., 2017) extended SPADE to detect arbitrary STPs (defined in section 5.2.4). STPs spanning a maximum number of K bins (for synchrony: $K = 1$) can be similarly defined as subsets of transactions constructed as follows. A window of K bins is slid through the data over time in steps of 1 bin (Figure 5.9A). Each window position corresponds to a transaction whose elements (items) are pairs (i, j), one pair per spike in the window, i represents the id of the neuron that emitted the spike, while j represents the relative location of the spike inside the window ($j = 1, \ldots, K$) (Figure 5.9B-C). Data formatted in transactions this way can be screened for closed frequent STPs by FIM (equivalently, FCA) algorithms. The evaluation of the statistical significance of closed frequent STPs requires the same steps as for synchronous patterns, namely PSF and PSR. The only difference is that the p-value spectrum can be extend to a third dimension (pattern duration d) to differentiate patterns with different time scales. Other approaches that filter patterns based on their stability (loosely speaking, the tendency of a pattern to reoccur identically) rather than on statistical significance were also investigated in Yegenoglu et al. (2016), but had a higher computational cost or yielded a lower performance.

Analysis of Sequences of Synchronous EvenTs (ASSET)

Sequences of synchronous spike events (SSEs) constitute one type of coordinated spiking where synchrony propagates from one group of neurons to the next in a temporally precise manner. The synfire chain was proposed as one potential model for such kind of network processing. Torre et al. (2016a) introduced the Analysis of Sequences of Synchronous EvenTs (ASSET) to reveal this type of correlated activity in MPST data. The method builds on the work of Schrader et al. (2008), extending it by introducing statistical tests and thereby allowing for a fully automated analysis.

First, time is segmented into consecutive bins of length w (see 5.10A, left). Second, any two time bins are compared for the number of neurons that spike in these two bins, i.e. the intersection of the two sets. The results of all these comparisons form the intersection matrix I such that the comparison of bin i and j is entered in the matrix element $I_{i,j}$. Synchronous events composed of the same (or many overlapping) neurons lead to a larger

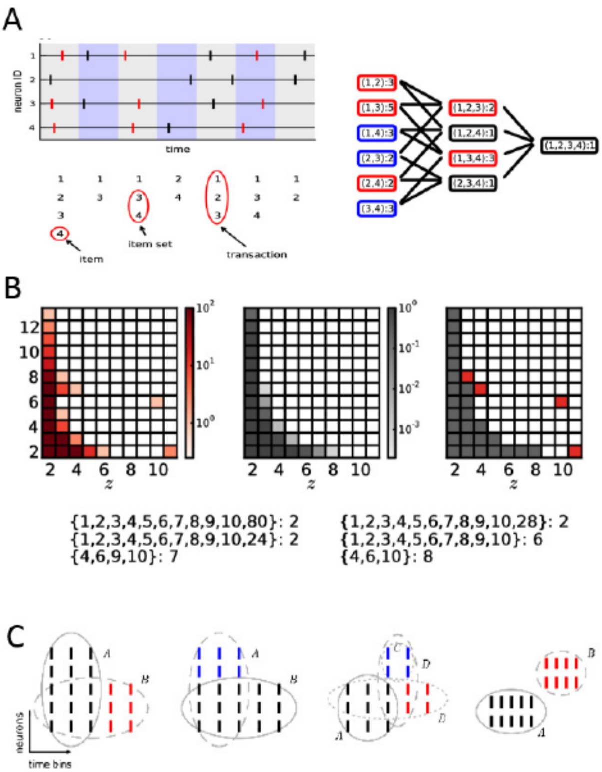

Figure 5.8: **SPADE analysis. A)** Sketch of the discretization of the parallel spike data into binned spike trains. The set of neuron ids ("items") spiking in each bin form a "transaction". The subsets extracted from each transaction, or "item sets", represent all observed synchronous spike patterns present in the data. The FIM data mining step organizes the item sets in a search tree and eventually returns all closed frequent item sets (right panel, circled in red), discarding the infrequent (black) and non closed (blue) ones. **B)** Significance evaluation. Illustration of assessment of closed frequent patterns for statistical significance of simulated data consisting of a synchronous pattern of size $z = 10$ occurring $c = 6$ times and embedded in a population with 90 additional independent spike trains). From left to right: pattern spectrum of the number of patterns for each signature (z, c) found in data; p-value spectrum of each signature under the null hypothesis computed over statistically independent surrogates of the original data; significant (red) and non significant (gray) signatures in the original data (significance threshold: $\alpha = 0.01$, corrected for multiple tests by false discovery rate correction). **C)** Patterns found as statistically significant after PSF (lower lists in B) are tested for reciprocal conditional significance. Conditionally significant patterns are retained (here, the true pattern $1, 2, \ldots 10$ occurring 6 times), the others are discarded as chance overlap of the significant ones with the background activity. Figure modified with permissions from Torre et al. (2013).

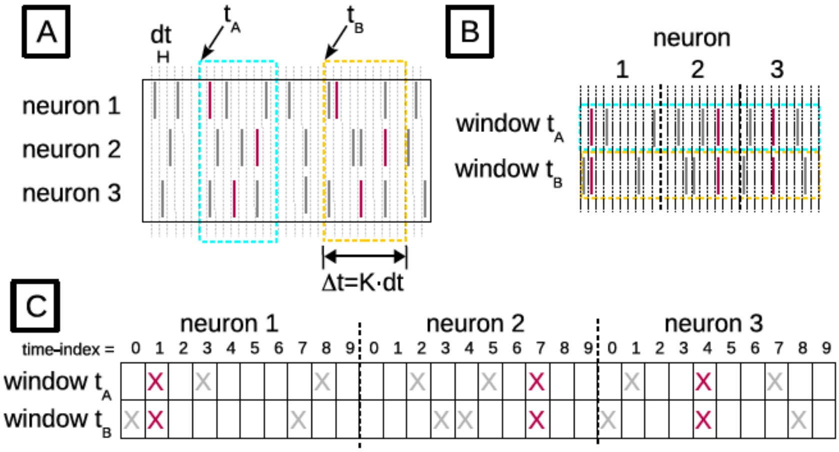

Figure 5.9: **Detection of spatio-temporal spike patterns. A)** Construction of a transaction data base. Spike trains are binned, and a window of length K bins is slid in time in steps of 1 bin. For window positions which start with a spike, the spikes falling into the window are collected. These are transformed in time such that the spikes per neuron are concatenated to a vector such that a list of pairs (i, j) of spike id i and relative spike time j, $j = 1, \ldots, K$ are formed. **B)** Transformed spiking activities from two window positions concatenated to parallel binary sequences enabling to search STPs by detection of synchronous entries as shown in **C)**. Figure modified with permissions from Quaglio et al. (2017).

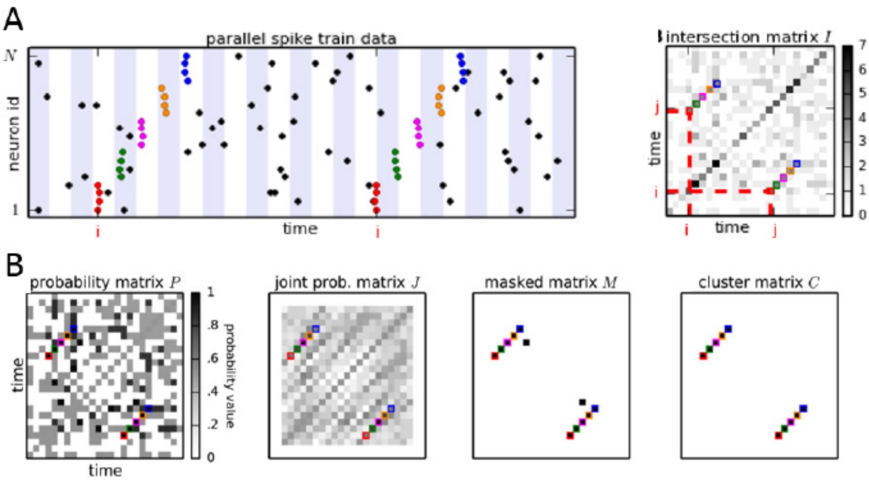

Figure 5.10: ASSET analysis. **A)** Sketch of a raster plot of parallel spike trains of multiple neurons (vertical axis) over time (horizontal axis). Dots in each row correspond to the spike times of one neuron. Time is discretized into adjacent bins (marked by white and blue shaded backgrounds) to define synchronous events. Synchronous spikes forming an SSE repeating twice are indicated by colored dots (one color per event). On the right: Intersection matrix I. Each matrix entry $I_{i,j}$ (values encoded by gray levels) contains the degree of overlap of neurons active in time bins b_i and b_j. **B)** Significance evaluation of repeating SSEs. Left: The cumulative probability $P_{i,j}$ calculated for each entry $I_{i,,j}$ analytically under the null hypothesis H_0 that the spike trains are independent and marginally Poisson. Second from left: The l largest neighbors of $I_{i,,j}$in a rectangular area extending along the 45 degree direction are isolated by means of a kernel and their joint cumulative probability is assigned to the joint probability matrix J at position $J_{i,j}$. Third from left: For a chosen significance threshold α_1 for the probability of individual entries $P_{i,j}$ and a significance threshold α_2 for the joint probability of the neighbors of entries $J_{i,j}$ each entry of I for which $P_{i,j} > \alpha_1$ and $J_{i,j} > \alpha_2$ is classified as statistically significant. Significant entries of I are retained in the binary masked matrix $M_{i,j}$, which takes value 1 at positions (i, j) where I is statistically significant and 0 elsewhere. B, right: 1-valued entries in M falling close-by are clustered together (or discarded as isolated chance events) by means of a DBSCAN algorithm, which thus isolates diagonal structures. Figure modified with permissions from Torre et al. (2016a).

value of $I_{i,,j}$ compared to independent data, i.e. chance overlap.

An SSE composed of (largely) the same neurons occurring two times in the data yields one diagonal structure of large entries in the intersection matrix I. Thus, a diagonal structure in the intersection matrix indicates the occurrence of a repeated SSE. ASSET detects and isolates diagonal structures in the intersection matrix by a statistical procedure. The method first transforms the intersection matrix I into a *probability matrix* P (Figure 5.10B, left) defined such that P_{ij} represents the probability for I_{ij} to be at most the observed value, under the null hypothesis of spike train independence. P_{ij} is obtained analytically or by Monte Carlo simulation. Values of I_{ij} larger than expected correspond to values of P_{ij} closer to 1. P is further transformed into a *joint probability matrix* J whose entries J_{ij} represent the joint probability of overlaps all the intersections I_{hk}, where the bins h, k form a neighborhood of (i, j) (Figure 5.10B, second from left). Diagonal structures in I due to a repeated SSEs lead to highly significant values both in P and in J. Individual, isolated repeated synchronous events yield a statistically significant entry in P but not in J. In light of these considerations, a *masked matrix* M is constructed, whose entries take binary values: $M_{ij} = 1$ if both P_{ij} and J_{ij} are statistically significant, $M_{ij} = 0$ otherwise (Figure 5.10B, third from right). Finally, close-by one-valued entries in the masked matrix are clustered together in the cluster matrix C of diagonal structures. This step allows to identify the diagonal structures as individual entities, and to discard spurious isolated entries in M (Figure 5.10B, right).

ASSET is robust to firing rate variability over time and across neurons, as well as to the presence of spike correlations different from SSEs (see Torre et al., 2016a). Furthermore, simulations of large balanced neuronal networks were used to demonstrate that the method is able to successfully discover SSEs resulting from the repeated synfire chain activation.

5.4 Method comparison

In the previous sections we gave an overview of nine methods for the analysis of temporally precise spike correlations in MPST data. We also illustrated the different ways these methods deal with the combinatorial and statistical challenges that characterize such an analysis. The various methods aim to reveal different types of correlated spiking activity. To this end, they rely on different statistics.

In the upcoming subsections we give a comparative overlook of the applicability of these methods to data characterized by different correlation structures. In particular, we discuss the sensitivity of each method to correlations of a different type than the one it was designed to detect. A natural question here is whether a method designed to analysis a particular correlation structure may still provide partial information about different types of correlated spiking. If so, analyzing a data set with different methods may provide the researcher with a richer picture of the possibly present correlations, and even aid a correct interpretation of the results. In the following we discuss how the introduced methods react to different correlation structures. 5.2 summarizes the results.

5.4.1 Population Synchronization

In the case when synchronous spike events involve different neurons at each occurrence time, no particular spike pattern reoccurs. CuBIC and PUE find the minimum order of excess synchronous events to be assumed in the data. The test statistics are based on the

Method	Pairwise Correlation	Synchronous Patterns	Population Synchronization	STPs	SSEs
Complexity distribution	(✓)	(✓)	✓		(✓)
CuBIC	(✓)	(✓)	✓		(✓)
PUE	(✓)	(✓)	✓		(✓)
Max. Entropy Information	✓		✓*		
Max. Entropy Models	✓	✓*			
Cliques	✓	(✓)	(✓)		(✓)
SPADE	(✓)	✓	(✓)	✓	✓
CAD	(✓)	✓	(✓)	✓	✓
ASSET					✓

Table 5.2: **Table of methods and stochastic models.** The table summarizes the ability of each method to retrieve correlations represented by different models. ✓: the method is designed to detect that particular model and the output matches perfectly and describes completely the correlation structure of the data. (✓) : The method was designed for a different correlation model, but it is still possible to get partial information about the correlation structure of the data. ✓*: The method is in principle applicable, but in practice affected by computational and/or multiple testing issues when used on MPST data; the results may lead to misinterpret the correlation structure due to lack of information about it. For the remaining entries the method does not provide sufficient information to reconstruct the correlation structure.

complexity distribution, which does not include information about the neuronal composition of each synchronous event. PUE can additionally be performed in a time resolved fashion, and therefore may discover time varying correlation orders. The CII approach quantifies the amount of information about the probability distribution of synchronous spike patterns that is delivered by correlations of a given order or lower, out of the total information delivered by all correlations. Thus, it may also be used to detect the maximum order of correlation in the data to account for a given percent (e.g., 99%) of such information. In practice though, CII is computationally intensive and typically cannot be used for MPST data to discriminate beyond second versus higher-order correlations.

The methods designed to detect specific groups of correlated neurons (MEMs, GIC, SPADE, CAD and ASSET), instead, are generally blind or weakly sensitive to population correlations. If the data are long enough and the population synchronization involves by chance the same spike patterns repeatedly, some of these methods may be able to classify such patterns as statistically significant. This, however, will provide only partial information about the true underlying correlation structure.

5.4.2 Pairwise Synchronization

The goal of the analysis of a data set containing pairwise synchronization consists in finding all the pairs of neurons involved in above-chance synchronous firing. In this scenario, CuBIC and PUE are expected to return the minimum order of correlation $\hat{\xi}$ necessary to explain the data, i.e. $\hat{\xi} = 2$. This holds also true for the case of overlapping pairs of correlated neurons. However, if the total amount of synchronous spike pairs present in the data is not high enough, these methods may report spike train independence instead. However, since the identity of the neurons involved in synchronous firing is not resolved, the specific correlated pairs are not found. CII instead takes values very close to 1, thus highlighting the absence of higher-order correlations. In the presence of time varying spike train statistics, CuBIC is prone to report higher values of $\hat{\xi}$ because the method assumes stationary conditions. The PUE analysis and the CII, instead, can account for time varying rates (the former by a time-resolved analysis). For the CII approach, however, this comes at a significantly increased computational cost.

Among the considered methods for detection of cell assemblies, GIC and CAD directly evaluate the statistical significance of each pair of synchronous firing neurons. Testing only for pairwise interactions makes these methods particularly efficient (high statistical power, relatively low computational burden). GIC can also cope well with time varying firing rates, as suitable CCH predictors (surrogates) exist for this case (Louis et al., 2010b). However, it may fail in properly characterizing time varying pairwise correlations, since it relies on the cross-correlogram, which is a time average measure. CAD instead detect all the occurrences of the synchronous activation, allowing the reconstruction of the exact temporal evolution of the pairs synchronizations. Furthermore, the more relevant difference between GIC and CAD is that the first groups together pairs or neurons which are mutually correlated, while CAD, if the single occurrences of synchronization involves only pairs of neurons, does not group them, but it returns patterns formed by individual pairs. SPADE is designed to detect specific sets of correlated neurons, including pairwise synchronization. It's statistical power, however, is lower than that of GIC and CAD for pairwise synchronization. Indeed, SPADE first tests for pattern significance on the basis of the pattern size and occurrence count, irrespective of the neuronal composition. Thus, a specific pair has to exhibit a larger number of synchronous events to be detected as significant, compared to direct statistical tests.

Finally, ASSET cannot retrieve correlated pairs of neurons, because they do not produce repeated SSEs.

5.4.3 Synchronous Spike Patterns

In the presence of synchronous spike patterns of size larger than 2, the optimal pattern detection would be a complete description of the correlated set of neurons: the neuron identities of the neurons in a synchronous event and their occurrence times.

In this scenario, methods to characterize population correlations (CuBIC, PUE, CII) generally tend to underestimate the correlation order in the data. The number of synchronous events of size ξ or larger needed for these methods to report a minimum order ξ of population correlation is much larger than the number of occurrences needed for a single pattern of size ξ to become statistically significant. Unless several patterns of size ξ exist, and their overall count is large enough, population methods will report correlations of lower order. This is particularly true for the CII index for $\xi = 2$, which has been shown to take values very close to 1 (meaning that correlations of order 3 or higher contribute negligibly to the total information about the probability distribution of synchronous spike patterns) also when highly significant synchronous spike patterns of much larger size are present in the data (Torre, 2016).

The GIC and CAD analysis may detect some of (but typically not all) the neurons forming a synchronous spike pattern of size larger than 2. The occurrences of the full pattern increase to some extent the peak in cross-correlations of the pairs contained in it, possibly leading to statistical significance for some of them. Only if all pairs become statistically significant, though, the two methods are guaranteed to further group them together and to reconstruct thereby the full pattern. This is typically not the case for patterns of larger size, since those typically exhibit lower occurrence counts in experimental data (see Torre et al., 2016b). An advantage of CAD, is the limited computational cost required to carry out the full analysis, due to the analytical formulation of the null distribution. Additionally for CAD the detected group form a pattern which occurs multiple times with the same neural composition, while with GIC it is not possible to distinguish between actual spike pattern and a groups of neurons that are mutually but independently correlated pairs.

MEM and SPADE are designed specifically to reliably detect reoccurring synchronous spike patterns, and therefore perform optimally in this scenario. MEM provides in addition a generative probabilistic model of the spiking activity, which allows for resampling. In addition, it allows one in principle to include correlations of any order among the neurons, as well as history effects that make the spike trains non-Poisson. On the down side, determining the model parameters becomes increasingly computationally demanding as more of such features are included. Also, testing for the statistical significance of each observed pattern runs into the multiple testing problem, effectively limiting the applicability of MEM to data with at most a few dozens of neurons. SPADE instead only indirectly conditions on existing correlations as it tests for the conditional significance of a pattern with a statistically significant signature given any other patterns overlapping with it. The method is also designed to drastically reduce the multiple testing issue. Importantly, it is very sensitive to synchronous events of large size, which need only few repetitions to reach the significance threshold. In contrast, low order events need to occur more times to be identified as statistically significant (see Torre et al., 2013, Figure 2). A downside compared to MEM is that SPADE solely assesses pattern significance, and does not provide a probabilistic model of the spiking activity.

ASSET, finally, does not detect isolated synchronous spike patterns (i.e., patterns not forming fixed, repeating sequences). The reason is that these events only produce isolated high-valued entries in the intersection matrix, but no diagonal structures.

5.4.4 Spatio-temporal patterns

STPs are the generalization of synchronous patterns to the case when neurons fire in a fixed temporal order (yielding a synchronous spike pattern in the special case when the delays are 0). The general definition of STPs also includes SSEs as a special case. Methods designed to detect population synchronization (such as CuBIC, PUE, CII), as well as methods limited to the detection of spike synchrony (GIC, MEMs), are not sensitive to STPs (except, of course, for synchronous spike patterns). Methods like SPADE and CAD are able to identify STPs of the general type. Specifically, SPADE allows to correctly identify and statistically test for any repeating spike sequence with pre-assigned maximum time lag. No additional assumptions are made on the structure of the pattern. The same holds for CAD, where also the maximum time lag is fixed before the analysis and thus limited to the maximum allowed delay.

Finally, ASSET is only able to identify STPs of the SSE type, a special case which is discussed next.

5.4.5 Sequences of synchronous spike events

An SSE consists of multiple synchronous events which occur at specific, fixed delays after one other. The presence of a reoccurring SSE (for instance due to the activation of an active synfire chain, see section 5.2.5) thus increases the overall amount of synchronization observed in data. If the SSE comprises sufficiently many events or these events involve sufficiently many spikes, population correlation methods could therefore detect the presence of synchrony. If the size of all synchronous events in the SSE is the same, say ξ, CuBIC and PUE would ideally return synchronization order ξ in the data. If instead the different synchronous events in the SSE have different size, they should return the maximum size. In both cases, however, both methods will typically return a lower correlation order. Furthermore, neither of the two methods identifies the neuronal composition of the events or their temporal structure. CII, instead, will report an information index R_2 very close to 1 if all events in the SSE comprise two spikes, and lower than if larger events are present. Computing indices R_ξ for $\xi > 2$ may help highlighting the existence of higher-order correlations, but it is computationally demanding. Besides, it would not provide a description of the complex correlation structure.

The GIC analysis could theoretically reconstruct the individual events forming an SSE. For this to be possible, the SSE has to occur sufficiently many times such that zero-delay pairwise correlations among all pairs of neurons involved in the same synchronous event become statistically significant. The method would then further group the overlapping pairs together, thus reconstructing each synchronous event separately. Besides that, even in this optimal scenario the synchronous events would be found in isolation, and further work would be needed to group them together into an SSE.

Since SSEs are a special case of STPs, they can be fully reconstructed with SPADE or CAD, if they occur sufficiently many times and if the total time span of one occurrence is shorter than the chosen analysis window. The number of occurrences needed for significance drops exponentially fast with the total number of involved neurons(see Torre et al., 2013). Thus, for SSEs involving sufficiently many neurons, even just a few repetitions are sufficient for detection by SPADE.

Finally, ASSET is specifically designed to detect SSEs occurring at least two times in the data. Unlike SPADE and CAD, the method accounts for their precise temporal structure (synchronous events and delays between them) to assess their significance. Specifically, ASSET computes the p-value of the SSEs as the joint probability of having synchronous events of the observed size in sequence. SPADE instead computes the probability of having any STP of different composition comprising the same number of spikes. For this reason, the statistical power of ASSET for SSEs occurring two times is higher than that of SPADE. This allows ASSET to retrieve SSEs composed of fewer neurons than SPADE is able to discover. SPADE does, on the other hand, more easily detect SSEs occurring more than 2 times, because it collects evidence from all pattern occurrences. ASSET, instead, evaluates by default only the significance of pairs of SSE occurrences, unless intersection tensors of higher dimension are built (see Gerstein et al., 2012, for dimension 3), which is possible but computationally demanding.

5.5 Discussion and Conclusions

In this manuscript we discussed methods which enable the analysis of massively parallel spike trains (the spiking activity of tens to hundred(s) of neurons recorded in parallel) for fine temporal correlations in the ms precision range. The common aim of such analyses is to identify spiking activity indicative of the presence of active cell assemblies (Hebb, 1949b), defined as groups of neurons that form building blocks for information processing in the cortex. Discovering and differentiating various types of temporally precise spike patterns in experimental spike data may be critical in understanding debated mechanisms of computations in the brain.

While no existing analysis method is able alone to distinguish among the different types of spike patterns discussed in the literature, combining the information delivered by different methods may provide a better strategy. Therefore, we suggest to apply multiple methods, in a particular sequence to approach unknown data. First, one would like to explore if there are at all indications for correlated activity. For doing that data can first be analyzed with computationally efficient methods, such as the complexity distribution (Grün et al., 2008) or other 'scanning' methods (e.g. Berger et al. (2010b)). If the complexity distribution provides no indication for the presence of higher-order correlations, pairwise or low order correlations or spatio-temporal patterns may still exist since the method is not sensitive for them. However, when correlations are found with the complexity distribution, or the maximum entropy methods, the sole interpretation is 'the data contain higher-order synchrony correlation', or in case of the application of CuBIC 'the data contain HOC exceeding order X'. Only SPADE, CAD or ASSET allow to identify higher-order spike correlations including temporal delay between the spikes and they identify the neurons involved in. If such spatio-temporal patterns are found, their spatial occurrence on the recording array (e.g. Utah array) may be identified (e.g. Torre et al., 2016b). With additional knowledge on the detailed position of the array on the cortex potentially involved local areas and the propagation direction may be uncovered. If on the other hand recordings are performed directly from different areas, e.g. as in Zandvakili and Kohn (2015), ASSET may uncover the propagation of sequences of synchronous activity from area to area.

Depending on the protocol of the experiment and the behavioral design, data can be split into different trials or segments that allow different interpretation. If for example data are split and pooled according different behavioral conditions, the analysis of the two with the same method (e.g. SPADE) may result in the presence of different spike patterns, which may be interpreted as 'in behavior A a different assembly was activated than in behavior B'. Even more informative are time-resolved analysis approaches which can identify dynamically occurring spike patterns, as done in Riehle et al. (1997b); Kilavik et al. (2009) using the UE analysis. PUE, as a further development of the CuBIC analysis, enables also such a time-resolved analysis due to the low computational requirement. Other methods that have a higher computational load, such as SPADE or ASSET, can be applied in a pseudo time-resolved fashion by segmenting the full data into epochs of interest and pooling across trials. Different significant spike patterns may occur in different epochs or experimental condition, which may be interpreted as 'different cell assemblies are activated in different behavioral contexts' (for an application of SPADE, see Torre et al. (2016b)).

Experimental data typically exhibit various types of variability - non-stationary firing rates, rate inhomogeneity across neurons or trials, and inter-spike intervals being more or less irregular than a Poisson process are common observations. These features need to be included in the null-hypothesis to avoid false positive findings (Grün et al., 2003; Pipa et al., 2013; Grün et al., 2002b). However, an analytical description of the null-hypothesis is for most of the cases mathematically not possible, or difficult in practice (for instance, parameters such as instantaneous firing rates cannot be well estimated from data; for a review see Grün (2009)). Surrogate data, i.e. modifications of the original data obtained by destroying the aspect that is tested for, e.g. fine temporal correlations, provide a practical alternative solution (see Louis et al. (2010c); Grün (2009), Platkiewicz et al. (2017)). For most of the methods discussed here, surrogates are used to derive the null-distribution(s) in the presence of such non-stationarities. The downside is that this approach leads typically to a higher computational load.

The temporal resolution (binning) chosen for the analyses is a matter of choice, and may also be varied as a parameter for finding the relevant time scale. Furthermore, the discussed methods can be applied to data not consisting of parallel spike trains, such as continuous signals, as long as they can be reduced topoint processes, and then to binary sequences by binning. . This approach is common for calcium imaging data, which are typically reduced to events in time of the potential underlying spikes(Grewe et al., 2010). The time resolution is much lower than of electrophysiologically recorded spike data. However, the result is then a matter the interpretation. Another example are spike-like signals in MEG recordings (Abeles, 2014). These were reduced in (Tal and Abeles, 2018) to point processes, and can then be treated as binary processes and analyzed by the methods discussed in this review.

We compared the methods with respect to the correlation model they are designed for, and their abilities to detect other correlation structures. A quantitative comparison of the methods would likely provide more insights. However, we learned from previous pairwise comparisons of some of such methods (e.g. CAD and SPADE, Stella (2017), FIM and the accretion algorithm, Picado-Muiño et al. (2013)) there are very few parameter configurations (e.g. temporal resolution, number of occurrences and size of the patterns or total length of the data) for which the performances are practically comparable. Moreover, the problem is not only about parameter configurations, but it is about the mathematical formulation, the different and not "hierarchical" definitions of correlated activity, which make a quantitative comparison difficult. A practical aspect for the difficulty of such comparisons is the fact that the various approaches are typically implemented in different software. A first step for an improvement of the situation would be a common software platform or even a common toolbox, as e.g. Elephant[1].

However, one may not forget that the number of neurons recorded in parallel are still small compared to the number of neurons contained in the tissue under observation. For example, the number of neurons contained in a piece of cortex covered by e.g. a 100 electrode Utah array (Blackrock Microsystems, Utah, USA) (4x4 mm^2) are about 10^5–10^6. Thus sampling 100 or 200 neurons from the tissue is still sparse compared to the number of neurons therein. In addition, we still do not know how cell assemblies are spatially embedded. Thus, unfortunately, it is very likely that we still miss neurons from active assemblies. For improving this situation a further increase of the number of neurons in parallel recorded should be aimed at and technically seems soon possible. This provides new opportunities to study large networks in even more details but will also require further extensions and developments of analysis methods.

[1]http://neuralensemble.org/elephant/

Chapter 6

Summary and Discussion

In this thesis we deal with the challenge of the detection of spatio-temporal patterns (STPs) in parallel electrophysiological spike recordings. In this context, we refer to spike patterns which consist of temporally precise (within a few ms) repeated spike sequences. Such patterns have been hypothesized to carry information in the brain and to be a signature of active cell assemblies (Hebb, 1949a; Gerstein and Kirkland, 2001; Harris, 2005). The analysis of parallel recordings for the detection of spike patterns presents challenging difficulties, both, in terms of data mining and statistical analysis. The number of possible combinations of spikes forming an STP scales exponentially, both, in respect to the number of simultaneously recorded neurons and to the maximal temporal duration of them. We developed a method that is capable of robustly detecting significant occurrences of precise spike patterns in massively parallel recordings. In particular we extended the Spike Pattern Detection and Evaluation (SPADE, Torre et al. (2013)) analysis. This method, originally designed to detect synchronous spike patterns, was here extended to detect general spike patterns with arbitrary inter-spikes lags. SPADE consists of two steps, addressing separately the challenges of the pattern analysis: a) it deploys Frequent Itemset Mining (FIM) to detect all the repeated spike patterns, b) it combines a bootstrap technique with a dimensionality reduction approach to test for the significance of the detected pattern candidates.

In more details, in the first chapter of this thesis we introduce the theoretical formalism of Formal Concept Analysis (FCA, Ganter and Wille (1999)) and show that such a framework is equivalent to Frequent Itemset Mining (Zaki and Ogihara, 1998; Pisková and Horváth, 2013), previously adapted by Picado-Muiño et al. (2013) for spiking data and used in SPADE to detect synchronous spike activity. Then we presented how FCA (or equivalently FIM) can be also used to detect repeated spatio-temporal spike patterns, extracted by using a temporal sliding window. Furthermore, it is possible to use the stability measure to discriminate between chance and relevant patterns. This measure was already introduced before as a metric which measures the amount of self-consistency of a Formal Concept and assigns a value between 0 and 1 to each mined concept (Kuznetsov, 2007). Here we showed that repeated spike sequences have a stability close to 1 and used the stability measure to discriminate relevant patterns from background noise. The performance of the approach was tested on simulated parallel point processes and showed that the main limitation of this approach is the considerable computational time required for the calculation of the stability measure. Indeed it is possible using FCA to select spike patterns forming stable concepts but only for a limited number of neurons (here only 50 parallel spike trains were considered) without incurring in an untreatable explosion of the time required for computing the stability. Furthermore, we showed that the stability measure does not depend on the pattern size. As a consequence, large patterns occurring a few times, and having a small stability value, were erroneously discarded thereby causing a large number of False Negatives. Part of this work was published in Yegenoglu et al. (2016).

In the second chapter we describe how we overcame the problem of the large computational time required for the computation of the stability measure, namely by replacing it with a computationally optimized approximation (Babin and Kuznetsov, 2012). We were able to show that the use of the approximated stability does not affect the performance of selecting the injected patterns (TPs). By speeding up the computation of the stability by the approximated stability, we were then able to compute a second stability measure, i.e. extensional stability, which depends on the pattern size and enables the detection of large patterns that occur only a few times.

Additionally we completed the extension of SPADE to STPs by adapting the statistical evaluation of the patterns deployed in the original publication (Torre et al., 2013). As described in the first chapter, the STPs are detected by a sliding window approach, and therein the patterns are mined by FIM/FCA. The output of the mining detection procedure is then analyzed for significance by the Pattern Spectrum Filtering (PSF), followed by the Pattern Set Reduction (PSR) approach, as done for synchronous patterns by the original SPADE method (Torre et al., 2013). The PSF deals with the multiple testing problem by grouping the patterns according to their size (number of spikes) and frequency (number of occurrences). Then a bootstrap technique is used to evaluate the statistical significance of each group of patterns. The PSR is an additional test designed to reject spurious patterns which are not conditionally significant on the ones that are already identified as significant. Thereby the significance of each pattern that resulted from PSF as significant is tested again, conditioned on the presence of other significant patterns.

The stability filtering and PSF followed by PSR were then compared using ground truth data, i.e. simulated data containing STPs and a variety of different statistical features of typical experimental data, such as different degrees of firing rates and correlated changes of firing rates. We showed that the combination of PSF and PSR performs better than the stability filtering, in particular we showed that PSF-PSR has a higher statistical power as a function of pattern size and frequency as compared to the stability measure. Yet the computational time required for the approximation of the stability is smaller than the one required for the bootstrap technique deployed by the PSF. In the case of very large datasets or the non-availability of a compute cluster for parallelizing the bootstrap analysis, one can resort to the stability approach to select the relevant patterns, at the cost of missing some of the statistically significant patterns (larger number of False Negatives) . These results were published in Quaglio et al. (2017).

A limitation of SPADE as introduced in chapter 1 and 2 arises when patterns, that are considered in the analysis, have different durations, defined as the time between the first and the last spike of the pattern. The pattern spectrum pools together patterns with the same size and same number of occurrences independently from their durations. However, within a longer time window may occur many more patterns by chance (more combinations possible) than in shorter windows. Therefore longer patterns require a different null-hypothesis than shorter ones, and thus should not be pooled for the significance estimation.

In chapter 3 we illustrated this limitation based on artificial data in which multiple patterns of different durations were embedded in the background activity. As a result, patterns of longer durations were not detected when pooled and evaluated together with shorter patterns. Therefore we proposed for improvement to extend the Pattern Spectrum to a third dimension, i.e. for different pattern durations. By this, patterns of different durations are tested separately from patterns of same size and number of occurrences. This extension to a 3d-pattern spectrum enables a statistical evaluation that does not depend on the length of the sliding window and enhanced the sensitivity of SPADE considerably. In order to avoid the increase of False Positive we used additionally a more conservative multiple testing correction (Holm, 1979).

The extended SPADE method, presented here for the first time, has still a few limitations that should be improved in the future. A first issue is the detection of spatio-temporal patterns in discretized time (small time bins), in order to apply FIM for the detection of repeated STPs. The capability of detecting STPs in continuous time would enable to consider also patterns which do not repeat with exact lags but with some variable time jitter. Detecting the patterns in continuous time would present very complex technical challenges and would require a re-implementation of the mining algorithm (as done for synchronous patterns in Borgelt and Picado-Muiño, 2014). Another limitation is the requirement of the exact repetition of the STPs in terms of neuronal compositions (e.g. FIM, in order to detect a pattern repetition, requires that all the spikes are present in every occurrence). Allowing also for partial activation of an assembly (thus only part of the pattern would be detectable), would enable us to detect more potential patterns. Such change would require to develop a new ad hoc mining technique to replace FIM for detection of incomplete pattern occurrences.

From the computational perspective most of the compute time is taken by the significance analysis based on the surrogates. If these could be replaced by an analytical expression of the null-hypothesis it would provide a significant speed up of the whole procedure. Obtaining such a null-distribution in a closed form is far from trivial and requires to make specific and strict assumptions in terms of the underlying statistical model of the spiking activity, i.e. the type of process and its time dependent changes (e.g. Poisson process, firing rate profiles). Such assumptions would limit the applicability of the method to data that fulfill such assumptions, which is not always the case for typical experimental data. In contrast the surrogate bootstrap method, deployed by SPADE, does not assume a specific spiking model or firing rate, but was designed such that it harms as little as possible the original processes (Grün, 2009; Louis et al., 2010c). All the observations here listed concern exceptional cases of spike patterns that SPADE does not take into account, but do not prejudicate the validity of the current results.

Once the development of the SPADE method was finalized, we applied the analysis to parallel spike recordings from pre- and motor cortex of two macaque monkeys performing a reach-to-grasp task (data published in Brochier et al., 2018). In each of the trials the monkeys received the instruction which (of two) grip to use to grasp the object after having reached for it. The amount of force needed to pull the object is indicated at the beginning of the trial by the cue signal (Riehle et al., 2013). In chapter 3 we present the results of such analysis. We found that SPADE detects precise STPs occurring a significant number of times and repeat over trials of the same behavioral type. Most of the patterns occur during the reaching movement and involve (are of size) 2, 3 or 4 neurons. The patterns show a variety of lags and durations, while the neurons forming the patterns appear to be a relatively small subset of all the recorded units (less then 10 out of the about 100 recorded neurons in all trial conditions and the two monkeys). Different patterns, both in terms of neuronal composition and delays, occur for different grips and thus are behavior specific (Quaglio et al., 2018a). The work is currently extended to analyze many sessions of each monkey to enable a quantitative description of the results over sessions and monkey, as for example performed for synchronous pattern in (Torre et al., 2016b). We further aim to relate the patterns to other electrophysiological signals of a different scale, such as the Local Field Potential (LFP). In former studies we and others found evidence that spikes and excess spike synchrony lock preferentially to the phase of the LFP (Denker et al., 2011; Kim et al., 2011; Takahashi et al., 2015).

By applying SPADE to the reach-to-grasp data, we found significant repetitions of a variety of precise STPs across different trials. Such a result provides evidence for the temporal coding hypothesis. Our analysis extend the previous observations of behaviorally relevant synchronous spike patterns in the recordings from the same experiment (Torre et al., 2016b) to spatio-temporal spike patterns. However, we are surprised that only a very small number of neurons are comprising significant STPs (typically 4 out of 160 recorded neurons = 1/40). The reason for recording massively parallel spike trains was to increase the probability to find neurons involved in correlated activity, and that the number of neurons involved in such activity would increase. Such an observation may have two reasons. First, the recorded neurons (100 - 160) are a small sample in respect to all the neurons in the motor cortex below the array (about 10^4). In addition, it is still not clear how a cell assembly is distributed in space. Further, we have to assume that our data contain only recordings from one cortical layer. With a length of 1.5 mm of the electrodes implanted, we can assume the recordings are in intermediate cortical layers from deep layer III to layer V (Brochier and Riehle, personal communication). It may well be that there are large assemblies, however spread over most of the motor cortex, and we only record from a few neurons that are involved in the processing of a particular movement. The second reason is that our analysis is very conservative - most of the actually repeated spatio-temporal patterns have been discarded because they did not occur often enough to become significant. The reasons are: a) in the analysis we require that a repeating pattern occurs at least in a third of the trials to be behaviorally relevant, b) The significance tests deployed by SPADE are very conservative as also shown in the performance tests on artificial data, in particular for patterns with a size of 3-4 spikes as found here. We choose such an approach that guarantees that the significant patterns cannot be explained by a correlation of firing rate modulations, but discards repeated patterns which do not occur often enough to be surprising. Nevertheless, a non-significant number of occurrences does not imply that a pattern is not relevant for the information processing in the brain. Third, we exclude auto-patterns, i.e. an individual neuron may not contribute with more than one spike in a pattern. The reason is that the spike trains in motor cortex have a tendency to be rather regular, and thus may lead to false positive chance patterns. This limitation may be considerable and may be an explanation why we have so many patterns in which the same neurons are involved.

Some years ago, when only a few neurons could be recorded simultaneously, pairwise correlations were found (e.g. as Unitary Events in Riehle et al., 1997a; Maldonado et al., 2008; Kilavik et al., 2009; Shimazaki et al., 2012) in about 1/3 of the recorded data sets. Now we find in massively parallel recordings synchrony or STP patterns in each of the recordings. Thus it seems more likely to record from cell assemblies by the use of more electrodes in parallel.

In this work we were interested in assessing the presence of significant precise patterns in the context of detecting patterns beyond the effect of rate changes and covariation, a concept that was also followed in the Unitary Events analysis (Grün et al., 2002a,b). Nonetheless, if one would not be interested in significant patterns only, but in any precise STP, for instance to correlate their occurrences to the behavior, it would still be possible to use SPADE for the detection without any statistical evaluation of the patterns. In terms of data analysis the natural continuation of this work is to extend the application of SPADE to more data in order to corroborate the results of this thesis, both, to more recording sessions of the reach-to-grasp experiment and to recordings in different experiments in different labs.

Many other methods have been developed for the analysis of spike correlations in massively parallel recordings. However, they consider different statistical aspects as a correlation and assume different models underlying the correlations, (e.g. Staude et al., 2010b). Thus they make use of different statistical measures. In order to get an understanding in how far these methods are complementary or detect similar correlations, we focused in the last chapter on reviewing a selection of such methods (Grün et al., 2008; Staude et al., 2010b; Rostami, 2017; Schneidman et al., 2003; Berger et al., 2010a; Russo and Durstewitz, 2017; Torre et al., 2016a; Quaglio et al., 2017), including SPADE. The methods and their performances are compared qualitatively, in particular in respect to which type of correlation structure they detect (e.g. pairwise correlation, sparse higher order correlations, spike patterns, etc.). We limited ourselves to a qualitative analysis because the first condition necessary for a quantitative analysis is the rigorous standardized implementation of each method and this is not yet the case for all of them. This work was published as Quaglio et al. (2018b).

In our comparative review we argue that SPADE is the only method of the ones considered that is capable to detect precise spatio-temporal patterns and test directly for the significance of such higher-order correlations. Indeed, all other methods considered in our comparison either do not resolve the neuronal composition of the correlation (Grün et al., 2008; Staude et al., 2010b; Rostami, 2017; Schneidman et al., 2003) or are built on pairwise testing (Berger et al., 2010a; Russo and Durstewitz, 2017). Nevertheless, an increasing interest in the detection of spike patterns led recently to developments of new methodologies for such analyses (e.g Watanabe et al., 2017; Peter et al., 2017; Kreuz et al., 2017; Grossberger et al., 2018; Mackevicius et al., 2018). All these methods have different assumptions as well as formalisms which require a thorough comparison both in terms of performance as well as compatibility of the object of the analysis (e.g. temporal scale used for the definition of the patterns). As already mentioned the first necessary condition for a quantitative analysis of the different methodologies is their standardization in terms of implementation and accessibility. To enable also others for comparison of methods we made the documented implementation of SPADE publicly available (`https://github.com/NeuralEnsemble/elephant`), as well as all the artificial and experimental data analyzed in this thesis (`https://web.gin.g-node.org/INT/multielectrode_grasp` and `https://github.com/INM-6/SPADE_analysis`).

Acknowledgments

I would like to express my gratitude to Prof. Sonja Grün that introduced me to the field of computational neuroscience and gave me the opportunity to develop my thesis at INM6 in the Jülich Forschungszentrum. Her valuable supervision was indispensable to accomplish this work.

I would also like to thank Prof. Markus Diesmann and Prof. Sonja Grün, who, as directors of the INM6, provided me with an livable scientifically stimulating environment where to work for the past four years.

My gratitude goes also to Dr. Emiliano Torre, Dr. Michael Denker, Prof. Dominik Endres, Alper Yegenoglu and Alessandra Stella with whom I had the pleasure to collaborate directly, their work and contribution were fundamental for the development of my thesis.

I also want to thank all the people I had the pleasure to share my time and ideas during all these years at the INM6. They are too many to list here, but in particular I want to cite Philipp Weidel, Robin Pauli and Michael Denker who helped revising the text of this thesis.

Bibliography

Abeles, M. (1982). Role of cortical neuron: integrator or coincidence detector? *Israel Journal of Medical Sciences*, 18:83–92.

Abeles, M. (1991). *Corticonics: Neural Circuits of the Cerebral Cortex*. Cambridge University Press, Cambridge, 1st edition.

Abeles, M. (2014). Revealing instances of coordination among multiple cortical areas. *Biological cybernetics*, 108(5):665–675.

Abeles, M. and Gerstein, G. L. (1988). Detecting Spatiotemporal Firing Patterns Among Simultaneously Recorded Single Neurons. *Journal of Neurophysiology*, 60(3):909–924.

Adrian, E. (1926a). The impulses produced by sensory nerve-endings: Part 4. impulses from pain receptors. *Journal of Physiology*, 62(1):33–51.

Adrian, E. (1926b). The impulses produced by sensory nerve endings: Part i. *Journal of Physiology*, 61(1):49–72.

Andrews, S. (2009). In close, a fast algorithm for computing formal concepts. In *Seventeenth International Conference on Conceptual Structures*.

Ashida, G., Kretzberg, J., and Tollin, D. J. (2016). Roles for coincidence detection in coding amplitude-modulated sounds. *PLoS computational biology*, 12(6):e1004997.

Ayzenshtat, I., Meirovithz, E., Edelman, H., Werner-Reiss, U., Bienenstock, E., Abeles, M., and Slovin, H. (2010). Precise spatiotemporal patterns among visual cortical areas and their relation to visual stimulus processing. *The Journal of Neuroscience*, 30(33):11232–11245.

Babin, M. A. and Kuznetsov, S. O. (2012). Approximating concept stability. In *Formal Concept Analysis - 10th International Conference, ICFCA 2012, Leuven, Belgium, May 7-10, 2012. Proceedings*, pages 7–15.

Baker, S. N., Spinks, R., Jackson, A., and Lemon, R. N. (2001). Synchronization in monkey motor cortex during a precision grip task. I. Task-dependent modulation in single-unit synchrony. *Journal of Neurophysiology*, 85(2):869–85.

Bender, V. A., Bender, K. J., Brasier, D. J., and Feldman, D. E. (2006). Two coincidence detectors for spike timing-dependent plasticity in somatosensory cortex. *The Journal of Neuroscience*, 26(16):4166–77.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):pp. 289–300.

Berger, D., Borgelt, C., Louis, S., Morrison, A., and Grün, S. (2010a). Efficient identification of assembly neurons withinmassively parallel spike trains. *Computational Intelligence and Neuroscience*, 2010.

Berger, D., Warren, D., Normann, R., Arieli, A., and Grün, S. (2007). Spatially organized spike correlation in cat visual cortex. *Neurocomputing*, 70(10-12):2112–2116.

Berger, T. K., Silberberg, G., Perin, R., and Markram, H. (2010b). Brief bursts self-inhibit and correlate the pyramidal network. *PLoS Biology*, 8(9):e1000473.

Bi, G.-q. and Poo, M.-m. (1998). Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of neuroscience*, 18(24):10464–10472.

Bienenstock, E. (1995). A model of neocortex. *Network: Computation in neural systems*, 6(2):179–224.

Bliss, T. V. P. and Lomo, T. (1973). Long-lasting potentation of synaptic transmission in the dendate area of anaesthetized rabbit following stimulation of the perforant path. *Journal of Physiology*, 232:331–356.

Borgelt, C. (2012). Frequent item set mining. In *Wiley Interdisciplinary Reviews (WIREs): Data Mining and Knowledge Discovery*, volume 2, pages 437–456. J. Wiley & Sons, Chichester, United Kingdom.

Borgelt, C. and Picado-Muiño, D. (2014). Simple pattern spectrum estimation for fast pattern filtering with coconad. In Blockeel, H., van Leeuwen, M., and Vinciotti, V., editors, *Advances in Intelligent DataAnalysis XIII*, volume 8819, pages 37–48. Springer International Publishing.

Braitenberg, V. and Schüz, A. (1991). *Anatomy of the Cortex: Statistics and Geometry.* Springer-Verlag, Berlin, Heidelberg, New York.

Brette, R. (2015). Philosophy of the spike: rate-based vs. spike-based theories of the brain. *Frontiers in systems neuroscience*, 9:151.

Brochier, T., Zehl, L., Hao, Y., Duret, M., Sprenger, J., Denker, M., Grün, S., and Riehle, A. (2018). Massively parallel recordings in macaque motor cortex during an instructed delayed reach-to-grasp task. *Scientific data*, 5:180055.

Carr, C. E. (2004). Timing is everything: organization of timing circuits in auditory and electrical sensory systems. *Journal of Comparative Neurology*, 472:131–133.

Dayhoff, J. E. and Gerstein, G. L. (1983). Favored patterns in spike trains. I. detection. *Journal of Neurophysiology*, 49(6):1334–1348.

Denker, M. and Grün, S. (2015). Designing workflows for the reproducible analysis of electrophysiological data. In *International Workshop on Brain-Inspired Computing*, pages 58–72. Springer.

Denker, M., Roux, S., Lindén, H., Diesmann, M., Riehle, A., and Grün, S. (2011). The local field potential reflects surplus spike synchrony. *Cerebral Cortex*, 21:2681–2695.

Diba, K., Amarasingham, A., Mizuseki, K., and Buzsàki, G. (2014). Millisecond timescale synchrony among hippocampal neurons. *The Journal of Neuroscience*, 34(45):14984–14994.

Diesmann, M., Gewaltig, M.-O., and Aertsen, A. (1999). Stable propagation of synchronous spiking in cortical neural networks. *Nature*, 402(6761):529–533.

Eggermont, J. (2015). Animal models of auditory temporal processing. *International Journal of Psychophysiology*, 95(2):202–215.

Eggermont, J. J. (1990). *The Correlative Brain*, volume 16 of *Studies of Brain Function*. Springer-Verlag-Verlag, Berlin.

Ehm, W., Staude, B., and Rotter, S. (2007). Decomposition of neuronal assembly activity via empirical de-poissonization. *Electron. J. Statist.*, 1:473–495.

Engel, A. K., König, P., Schillen, T. B., and Singer, W. (1992). Temporal coding in the visual cortex: new vistas on integration in the nervous system. *Trends in Neurosciences*, 15(6):218–226.

Ferster, D. and Lindström, S. (1983). An intracellular analysis of geniculo-cortical connectivity in area 17 of the cat. 342:181–215.

Fino, E., Paille, V., Cui, Y., Morera-Herreras, T., Deniau, J., and Venance, L. (2010). Distinct coincidence detectors govern the corticostriatal spike timing-dependent plasticity. *Journal of Physiology*, 588(16):3045–3062.

Fries, P. (2005). A mechanism for cognitive dynamics: neuronal communication through neuronal coherence. *Trends Cogn Sci*, 9(10):474–480.

Fujisawa, S., Amarasingham, A., Harrison, M. T., and Buzsáki, G. (2008). Behavior-dependent short-term assembly dynamics in the medial prefrontal cortex. *Nature Neuroscience*, 11(7):823–833.

Gansel, K. S. and Singer, W. (2012). Detecting multineuronal temporal patterns in parallel spike trains. *Frontiers in Neuroinformatics*, 6:18.

Ganter, B. and Wille, R. (1999). *Formal concept analysis: mathematical foundations*. Springer Science & Business Media.

Gautrais, J. and Thorpe, S. (1998). Rate coding versus temporal order coding: a theoretical approach. *Biosystems*, 48:57–65.

Georgopoulos, A., Kalaska, J., Caminiti, R., and J.T., M. (1982). On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. 11(2):1527–1537.

Georgopoulos, A. P., Schwartz, A. B., and Kettner, R. E. (1986). Neuronal population coding of movement direction. 233:1416–1419.

Gerstein, G. and Clark, W. (1964). Simultaneous studies of firing patterns in several neurons. *Science*, 143(3612):1325–1327.

Gerstein, G. and Kirkland, K. (2001). Neural assemblies: technical issues, analysis, and modeling. 6–7(14):589–598.

Gerstein, G. L., Perkel, D. H., and Subramanian, K. N. (1978). Identification of functionally related neural assemblies. 140:43–62.

Gerstein, G. L., Williams, E. R., Diesmann, M., Grün, S., and Trengove, C. (2012). Detecting synfire chains in parallel spike data. *Journal of Neuroscience Methods*, 206(1):54–64.

Gerstner, W., Kreiter, A. K., Markram, H., and Herz, V. M. (1997). Neural codes: Firing rates and beyond. *Proceedings of the National Academy of Sciences*, 94:12740–12741.

Grewe, B. F., Langer, D., Kasper, H., Kampa, B. M., and Helmchen, F. (2010). High-speed in vivo calcium imaging reveals neuronal network activity with near-millisecond precision. *Nature methods*, 7(5):399.

Grossberger, L., Battaglia, F. P., and Vinck, M. (2018). Unsupervised clustering of temporal patterns in high-dimensional neuronal ensembles using a novel dissimilarity measure. *bioRxiv*, page 252791.

Grün, S. (2009). Data-driven significance estimation of precise spike correlation. *Journal of Neurophysiology*, 101(3):1126–1140.

Grün, S., Abeles, M., and Diesmann, M. (2008). Impact of higher-order correlations on coincidence distributions of massively parallel data. In *Lecture Notes in Computer Science, 'Dynamic Brain - from Neural Spikes to Behaviors'*, volume 5286, pages 96–114.

Grün, S., Diesmann, M., and Aertsen, A. (2002a). 'Unitary Events' in multiple single-neuron spiking activity. I. Detection and significance. *Neural Computation*, 14(1):43–80.

Grün, S., Diesmann, M., and Aertsen, A. (2002b). 'Unitary Events' in multiple single-neuron spiking activity. II. Non-Stationary data. *Neural Computation*, 14(1):81–119.

Grün, S., Riehle, A., and Diesmann, M. (2003). Effect of cross-trial nonstationarity on joint-spike events. *Biological Cybernetics*, 88(5):335–351.

Han, J., Pei, J., Yin, Y., and Mao, R. (2004). Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data mining and knowledge discovery*, 8(1):53–87.

Harris, K. (2005). Neural signatures of cell assembly organization. *Nature Reviews Neuroscience*, 5(6):339–407.

Harvey, M. A., Saal, H. P., III, J. F. D., and Bensmaia, S. J. (2013). Multiplexing stimulus information through rate and temporal codes in primate somatosensory cortex. *PLoS Biology*, 11(5):e1001558.

Hebb, D. O. (1949a). *The organization of behavior: A neuropsychological theory.* John Wiley & Sons, New York.

Hebb, D. O. (1949b). *The organization of behavior: A neuropsychological theory.* John Wiley & Sons, New York.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70.

Hong, S., Ratté, S., Prescott, S. A., and De Schutter, E. (2012). Single neuron firing properties impact correlation-based population coding. *The Journal of Neuroscience*, 32(4):1413–1428.

Hosaka, R., Araki, O., and Ikeguchi, T. (2008). STDP provides the substrate for igniting synfire chains by spatiotemporal input patterns. *Neural Computation*, 20:415–435.

Hu, E. and Bloomfield, S. (2003). Gap junctional coupling underlies the short-latency spike synchrony of retinal alpha ganglion cells. *The Journal of Neuroscience*, 23(17):6768–77.

Hübener, M., Shoham, D., Grinvald, A., and Bonhoeffer, T. (1997). Spatial relationships among three columnar systems in cat area 17. *Journal of Neuroscience*, 17(23):9270–9284.

Ikegaya, Y., Aaron, G., Cossart, R., Aronov, D., Lampl, I., Ferster, D., and Yuste, R. (2004). Synfire chains and cortical songs: temporal modules of cortical activity. *Science*, 304(5670):559–564.

Izhikevich, E. M. (2006). Polychronization: Computation with spikes. *Neural Computation*, 18:245–282.

Jaynes, E. T. (1957). Information theory and statistical mechanics. *The Physical Review*, 106(4):620–630.

Jones, E., Oliphant, T., Peterson, P., et al. (2001). SciPy: Open source scientific tools for Python. [Online; accessed 2016-01-25].

Kandel, E. R., Schwartz, J. H., and Jessell, T. M. (1991). *Principles of neural science.* McGraw-Hill, New York, 3 edition.

Kass, R. E., Kelly, R. C., and Loh, W.-L. (2011). Assessment of synchrony in multiple neural spike trains using loglinear point process models. *The Annals of Applied Statistics*, 5(2B):1262–1292.

Kelly, R. C. and Kass, R. E. (2012). A framework for evaluating pairwise and multiway synchrony among stimulus-driven neurons. *Journal of Neural Computation*, 24(8):2007–2032.

Kilavik, B. E., Roux, S., Ponce-Alvarez, A., Confais, J., Grün, S., and Riehle, A. (2009). Long-term modifications in motor cortical dynamics induced by intensive practice. *The Journal of Neuroscience*, 29:12653–12663.

Kim, S., Putrino, D., Ghosh, S., and Brown, E. N. (2011). A granger causality measure for point process models of ensemble neural spiking activity. *PLoS computational biology*, 7(3):e1001110.

König, P., Engel, A. K., and Singer, W. (1996). Integrator or coincidence detector? The role of the cortical neuron revisited. *Trends in Neurosciences*, 19(4):130–137.

Krajca, P. and Vychodil, V. (2009). Distributed algorithm for computing formal concepts using map-reduce framework. In *Proceedings of the 8th International Symposium on Intelligent Data Analysis: Advances in Intelligent Data Analysis VIII*, IDA '09, pages 333–344, Berlin, Heidelberg. Springer-Verlag.

Kreuz, T., Satuvuori, E., Pofahl, M., and Mulansky, M. (2017). Leaders and followers: quantifying consistency in spatio-temporal propagation patterns. *New Journal of Physics*, 19(4):043028.

Kuhn, A., Aertsen, A., and Rotter, S. (2003). Higher-order statistics of input ensembles and the response of simple model neurons. *Neural Computation*, 1(15):67–101.

Kuhn, A., Rotter, S., and Aertsen, A. (2002). Correlated input spike trains and their effects on the response of the leaky integrate-and-fire neuron. *Neurocomputing*, 44–46:121–126.

Kuznetsov, S. O. (2007). On stability of a formal concept. *Annals of Mathematics and Artificial Intelligence*, 49(1-4):101–115.

Kuznetsov, S. O. and Obiedkov, S. (2002). Comparing performance of algorithms for generating concept lattices. *Journal of Experimental and Theoretical Artificial Intelligence*, 14:189–216.

Lamme, V. and Spekreijse, H. (1998). Neuronal synchrony does not represent texture segregation. *Nature*, 396(6709):362–366.

Levakova, M., Tamborrino, M., Ditlevsen, S., and Lansky, P. (2015). A review of the methods for neuronal response latency estimation. *Biosystems*, 136:23–34.

Lindig, C. (2000). Fast concept analysis. In *Working with Conceptual Structures - Contributions to ICCS 2000*, pages 152–161. Shaker Verlag.

Lømo, T. (2003). The discovery of long-term potentiation. *Philosophical Transaction of the Royal Society London, Serie B.*, 358:617–620.

London, M., Roth, A., Beeren, L., Häusser, M., and Latham, P. E. (2010). Sensitivity to perturbations in vivo implies high noise and suggests rate coding in cortex. *Nature Reviews Neuroscience*, 466(1):123–128.

Louis, S., Borgelt, C., and Grün, S. (2010a). Complexity distribution as a measure for assembly size and temporal precision. *NeuralNetworks*, 23:705 – 712.

Louis, S., Borgelt, C., and Grün, S. (2010b). Generation and selection of surrogate methods for correlation analysis. *Analysis of Parallel Spike Trains*, pages 359–382.

Louis, S., Gerstein, G. L., Grün, S., and Diesmann, M. (2010c). Surrogate spike train generation through dithering in operational time. *Frontiers in computational neuroscience*, 4(127).

Mackevicius, E. L., Bahle, A. H., Williams, A. H., Gu, S., Denissenko, N. I., Goldman, M. S., and Fee, M. S. (2018). Unsupervised discovery of temporal sequences in high-dimensional datasets, with applications to neuroscience. *bioRxiv*, page 273128.

Maldonado, P., Babul, C., Singer, W., Rodriguez, E., Berger, D., and Grün, S. (2008). Synchronization of neuronal responses in primary visual cortex of monkeys viewing natural images. *Journal of Neurophysiology*, 100(3):1523–1532.

Manninen, T., Aćimović, J., Havela, R., Teppola, H., and Linne, M.-L. (2018). Challenges in reproducibility, replicability, and comparability of computational models and tools for neuronal and glial networks, cells, and subcellular structures. *Frontiers in neuroinformatics*, 12.

Martin, A. B. and von der Heydt, R. (2015). Spike synchrony reveals emergence of proto-objects in visual cortex. *The Journal of Neuroscience*, 37(15):6860–6870.

Massey, P. and Bashir, Z. (2007). Long-term depression: multiple forms and implications for brain function. *Trends in Neurosciences*, 30(4):176–184.

Nadasdy, Z., Hirase, H., Czurko, A., Csicsvari, J., and Buzsaki, G. (1999). Replay and time compression of recurring spike sequences in the hippocampus. *The Journal of Neuroscience*, 19(21):9497–9507.

Olson, D. L. and Delen, D. (2008). *Advanced data mining techniques*. Springer Science & Business Media.

Palm, G. (1981). Evidence, information and surprise. *Biological Cybernetics*, 42:57–68.

Perez-Orive, J., Bazhenov, M., and Laurent, G. (2004). Intrinsic and circuit properties favor coincidence detection for decoding oscillatory input. *The Journal of Neuroscience*, 24(26):6037–47.

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

Peter, S., Kirschbaum, E., Both, M., Campbell, L., Harvey, B., Heins, C., Durstewitz, D., Diego, F., and Hamprecht, F. A. (2017). Sparse convolutional coding for neuronal assembly detection. In *Advances in Neural Information Processing Systems*, pages 3675–3685.

Picado-Muiño, D., Borgelt, C., Berger, D., Gerstein, G. L., and Grün, S. (2013). Finding neural assemblies with frequent item set mining. *Front. Neuroinform.*, 7(9). doi: 10.3389/fninf.2013.00009.

Pillow, J. W., Shlens, J., Paninski, L., Sher, A., Litke, A. M., Chichilnisky, E. J., and Simoncelli, E. P. (2008). Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454:995–999.

Pipa, G., Grün, S., and van Vreeswijk, C. (2013). Impact of spike train autostructure on probability distribution of joint spike events. *Neural Computation*, 25(5):1123–63.

Pipa, G. and Munk, M. H. J. (2011). Higher order spike synchrony in prefrontal cortex during visual memory. *Frontiers in Computational Neuroscience*, 5(23).

Pipa, G., Wheeler, D. W., Singer, W., and Nikolic, D. (2008). Neuroxidence: reliable and efficient analysis of an excess or deficiency of joint-spike events. *Journal of Neuroscience Methods*, 25:64 – 88.

Pisková, L. and Horváth, T. (2013). Comparing performance of formal concept analysis and closed frequent itemset mining algorithms on real data. In *CLA*, pages 299–304.

Platkiewicz, J., Stark, E., and Amarasingham, A. (2017). Spike-centered jitter can mistake temporal structure. *Neural computation*, 29(3):783–803.

Prut, Y., Vaadia, E., Bergman, H., Haalman, I., Hamutal, S., and Abeles, M. (1998). Spatiotemporal structure of cortical activity: Properties and behavioral relevance. *Journal of Neurophysiology*, 79(6):2857–2874.

Quaglio, P., A., Y., Torre, E., and Grün, S. (2018a). Detection of spatio-temporal spike patterns in motor cortex during a reach-to-grasp task. In *27th Annual Computational Neuroscience Meeting*.

Quaglio, P., Rostami, V., Torre, E., and Grün, S. (2018b). Methods for identification of spike patterns in massively parallel spike trains. *Biological cybernetics*, 112(1-2):57–80.

Quaglio, P., Yegenoglu, A., Torre, E., Endres, D. M., and Grün, S. (2017). Detection and evaluation of spatio-temporal spike patterns in massively parallel spike train data with spade. *Frontiers in Computational Neuroscience*, 11:41.

Reed, J., Pouget, P., Qi, H., Zhou, Z., Bernard, M., Burish, M., Haitas, J., Bonds, A., and JH., K. (2008). Widespread spatial integration in primary somatosensory cortex. *Proceedings of the National Academy of Sciences of the United States of America*, 105(29):10233–7.

Reimer, I., Staude, B., Ehm, W., and Rotter, S. (2012). Modeling and analyzing higher-order correlations in non-poissonian spike trains. *Journal of Neuroscience Methods*, 208(1):18–33.

Riehle, A., Grün, S., Diesmann, M., and Aertsen, A. (1997a). Spike synchronization and rate modulation differentially involved in motor cortical function. *Science*, 278(5345):1950–1953.

Riehle, A., Kornblum, S., and Requin, J. (1997b). Neuronal correlates of sensorimotor association in stimulus-response compatibility. *Journal of Experimental Psychology: Human Perception and Performance*, 23:1708–1726.

Riehle, A., Wirtssohn, S., Grün, S., and Brochier, T. (2013). Mapping the spatio-temporal structure of motor cortical lfp and spiking activities during reach-to-grasp movements. *Frontiers in Neural Circuits*, 7:48.

Roelfsema, P. R., Lamme, V. A. F., and Spekreijse, H. (2004). Synchrony and covariation of firing rates in the primary visual cortex during contour grouping. *Nature Neuroscience*, 7(9):982–991.

Rostami, V. (2017). *Statistical analysis tools for assessing the functional relevance of higher-order correlations in massively parallel spike trains.* PhD thesis, RWTH Aachen University.

Rostami, V., Mana, P. P., Grün, S., and Helias, M. (2017). Bistability, non-ergodicity, and inhibition in pairwise maximum-entropy models. *PLOS Computational Biology*, 13(10):e1005762.

Roth, C., Obiedkov, S. A., and Kourie, D. G. (2008). On succinct representation of knowledge community taxonomies with formal concept analysis. *International Journal of Foundation of Computer Science*, 19(2):383–404.

Roth, M. M., Helmchen, F., and Kampa, B. M. (2012). Distinct functional properties of primary and posteromedial visual area of mouse neocortex. *The Journal of neuroscience*, 32(28):9716–9726.

Roudi, Y., Nirenberg, S., and Latham, P. E. (2009). Pairwise maximum entropy models for studying large biological systems: when they can work and when they can't. *Plos Computational Biology*, 5(5):e1000380.

Roy, S. A. and Alloway, K. D. (2001). Coincidence detection or temporal integration? what the neurons in somatosensory cortex are doing. *The Journal of Neuroscience*, 21(7):2462–2473.

Russo, E. and Durstewitz, D. (2017). Cell assemblies at multiple time scales with arbitrary lag constellations. *eLife*, 6:e19428.

Sakurai, Y. (1996). Hippocampal and neocortical cell assemblies encode memory processes for different types of stimuli in the rat. *The Journal of Neuroscience*, 16(8):2809–19.

Sakurai, Y. and Takahashi, S. (2006). Dynamic synchrony of firing in the monkey prefrontal cortex during working-memory tasks. *The Journal of Neuroscience*, 6(40):10141–10153.

Salami, M., Itami, C., Tsumoto, T., and Kimura, F. (2003). Change of conduction velocity by regional myelination yields constant latency irrespective of distance between thalamus and cortex. *Proceedings of the National Academy of Sciences*, 100(10):6174–6179.

Schneidman, E., Berry, M. J., Segev, R., and Bialek, W. (2006). Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature*, 440:1007–1012.

Schneidman, E., Still, S., II, M. J. B., and Bialek, W. (2003). Network information and connected correlations. *Phisical Review Letters*, 91(23):238701.

Schrader, S., Grün, S., Diesmann, M., and Gerstein, G. (2008). Detecting synfire chain activity using massively parallel spike train recording. *Journal of Neurophysiology*, 100:2165–2176.

Schultze-Kraft, M., Diesmann, M., Grün, S., and Helias, M. (2013). Noise suppression and surplus synchrony by coincidence detection. *PLoS Computational Biology*, 9(4):e1002904.

Schwarz, D. A., Lebedev, M. A., Hanson, T. L., Dimitrov, D. F., Lehew, G., Meloy, J., Rajangam, S., Subramanian, V., Ifft, P. J., Li, Z., Ramakrishnan, A., Tate, A., Zhuang, K. Z., and Nicolelis, M. A. L. (2014). Chronic, wireless recordings of large-scale brain activity in freely moving rhesus monkeys. *Nature Methods*, 11:670–676.

Seki, S. and Eggermont, J. J. (2003). Changes in spontaneous firing rate and neural synchrony in cat primary auditory cortex after localized tone-induced hearing loss. *Hearing Research*, 180(1-2):28–38.

Shadlen, M. N. and Newsome, W. T. (1994). Noise, neural codes and cortical organization. *Current Opinion in Neurobiology*, 4(4):569–579.

Shadlen, M. N. and Newsome, W. T. (1995). Is there a signal in the noise? *Current Opinion in Neurobiology*, 5:248–250.

Shimazaki, H., Amari, S.-i., Brown, E. N., and Grün, S. (2012). State-space analysis of time-varying higher-order spike correlation for multiple neural spike train data. *Plos Computational Biology*, 8(3):e1002385.

Shlens, J., Field, G. D., Gauthier, J. L., Grivich, M. I., Petrusca, D., Sher, A., Litke, A. M., and Chichilnisky, E. (2006). The structure of multi-neuron firing patterns in primate retina. *The Journal of Neuroscience*, 26(32):8254–8266.

Smith, M. A. and Kohn, A. (2008). Spatial and temporal scales of neuronal correlation in primary visual cortex. *The Journal of Neuroscience*, 28(48):12591–12603.

Staude, B., Grün, S., and Rotter, S. (2010a). Higher-order correlations in non-stationary parallel spike trains: statistical modeling and inference. *Frontiers in Computational Neuroscience*, 4:16.

Staude, B., Rotter, S., and Grün, S. (2010b). Cubic: cumulant based inference of higher-order correlations in massively parallel spike trains. *Journal of Computational Neuroscience*, 29:327 – 350.

Steinmetz, P., Roy, A., Fitzgerald, P., Hsiao, S., Johnson, K., and Niebur, E. (2000). Attention modulates synchronized neuronal firing in primate somatosensory cortex. *Nature*, 404(6774):187–190.

Stella, A. (2017). Comparison of statistical methods for spatio-temporal patterns detection in multivariate point processes: an application to neuroscience. Master's thesis, University of Turin.

Stent, G. (1973). A physiological mechanism for Hebb's postulate of learning. *Proceedings of the National Academy of Sciences of the United States of America*, 70(4):997–1001.

Swadlow, H. A. (1988). Efferent neurons and suspected interneurons in binocular visual cortex of the awake rabbit: Receptive fields and binocular properties. 59(4):1162–1187.

Swadlow, H. A. (1994). Efferent neurons and suspected interneurons in motor cortex of the awake rabbit: axonal properties, sensory receptive fields, and subthreshold synaptic inputs. *Journal of neurophysiology*, 71(2):437–453.

Takahashi, K., Kim, S., Coleman, T., Brown, K., Suminski, A., Best, M., and Hatsopoulos, N. (2015). Large-scale spatiotemporal spike patterning consistent with wave propagation in motor cortex. *Nature Communications*, 6:7169.

Tal, I. and Abeles, M. (2016). Temporal accuracy of human cortico-cortical interactions. *Journal of neurophysiology*, 115(4):1810–1820.

Tal, I. and Abeles, M. (2018). Imaging the spatiotemporal dynamics of cognitive processes at high temporal resolution. *Neural computation*, (Early Access):1–21.

Tang, A., Jackson, D., Hobbs, J., Chen, W., Smith, J. L., Patel, H., Pietro, A., Petrusca, D., Grivich, M. I., Sher, A., Hottowy, P., Dabrowski, W., Litke, A. M., and Beggs, J. M. (2008). A maximum entropy model applied to spatial and temporal correlations from cortical networks in vitro. *The Journal of Neuroscience*, 28(2):505–518.

Tkacik, G., Schneidman, E., II, M. J. B., and Bialek, W. (2006). Ising models for networks of real neurons. *arXiv:q-bio.NC*.

Torre, E. (2016). *Statistical analysis of synchrony and synchrony propagation in massively parallel spike trains*. PhD thesis, RWTH AAchen.

Torre, E., Canova, C., Denker, M., Gerstein, G., Helias, M., and Grün, S. (2016a). ASSET: Analysis of sequences of synchronous events in massively parallel spike trains. *PLOS Computational Biology*, 12(7):e1004939.

Torre, E., Picado-Muiño, D., Denker, M., Borgelt, C., and Grün, S. (2013). Statistical evaluation of synchronous spike patterns extracted by frequent item set mining. *Frontiers in Computational Neuroscience*, 7:115.

Torre, E., Quaglio, P., Denker, M., Brochier, T., Riehle, A., and Grün, S. (2016b). Synchronous spike patterns in macaque motor cortex during an instructed-delay reach-to-grasp task. *Journal of Neuroscience*, 36(32):8329–8340.

Vaadia, E., Haalman, I., Abeles, M., Bergman, H., Prut, Y., Slovin, H., and Aertsen, A. (1995). Dynamics of neuronal interactions in monkey cortex in relation to behavioural events. *Nature*, 373(6514):515–518.

van der Togt, C., Kalitzin, S., Spekreijse, H., Lamme, V., and Supèr, H. (2006). Synchrony dynamics in monkey V1 predict success in visual detection. *Cerebral Cortex*, 16(1):136–148.

Van Rullen, R. and Thorpe, S. (2001). Rate coding versus temporal order coding: what the retinal ganglion cells tell the visual cortex. *Neural Computation*, 13(6):1255–83.

VanRullen, R., Guyonneau, R., and Thorpe, S. J. (2005). Spike times make sense. *Trends in Neurosciences*, 28(1):1–4.

von der Malsburg, C. (1986). Am I thinking assemblies? In Palm, G. and Aertsen, A., editors, *Brain Theory*, pages 161–176. Springer-Verlag, Berlin.

Watanabe, K., Haga, T., Euston, D. R., Tatsuno, M., and Fukai, T. (2017). Unsupervised detection of cell-assembly sequences with edit similarity score. *bioRxiv*, page 202655.

Yegenoglu, A., Quaglio, P., Torre, E., Grün, S., and Endres, D. (2016). Exploring the usefulness of formal concept analysis for robust detection of spatio-temporal spike patterns in massively parallel spike trains. In *Graph-Based Representation and Reasoning*, pages 3–16. Springer Nature.

Zaki, M. J. and Ogihara, M. (1998). Theoretical foundations of association rules. In *In 3rd ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*.

Zandvakili, A. and Kohn, A. (2015). Coordinated neuronal activity enhances corticocortical communication. *Neuron*, 87(4):827–839.

Band / Volume 204
Spin-orbitronics at the nanoscale: From analytical models to real materials
J. Bouaziz (2019), 228 pp
ISBN: 978-3-95806-429-4

Band / Volume 205
Advanced methods for atomic scale spin simulations and application to localized magnetic states
G. P. Müller (2019), xx, 194 pp
ISBN: 978-3-95806-432-4

Band / Volume 206
Different growth modes of molecular adsorbate systems and 2D materials investigated by low-energy electron microscopy
J. E. Felter (2019), vi, 114, XXXIV pp
ISBN: 978-3-95806-434-8

Band / Volume 207
NADPH-related studies performed with a SoxR-based biosensor in *Escherichia coli*
A. Spielmann (2019), IV, 73 pp
ISBN: 978-3-95806-438-6

Band / Volume 208
Chemisorption aromatischer Moleküle auf Übergangsmetalloberflächen: Bildung molekularer Hybridmagnete
S. Schleicher (2019), 109 pp
ISBN: 978-3-95806-442-3

Band / Volume 209
Regulatory interactions between *Corynebacterium glutamicum* and its prophages
M. Hünnefeld (2019), IV, 209 pp
ISBN: 978-3-95806-445-4

Band / Volume 210
Quantum Technology
Lecture Notes of the 51st IFF Spring School 2020
23 March – 03 April 2020, Jülich, Germany
ed. by H. Bluhm, T. Calarco, D. DiVincenzo (2020), ca. 700 pp
ISBN: 978-3-95806-449-2

Band / Volume 211
Interaction of physical fields with nanostructured materials
(2020), 255 pp
ISBN: 978-3-95806-450-8

Band / Volume 212
First-principles study of collective spin excitations in noncollinear magnets
F.J. dos Santos (2020), 270 pp
ISBN: 978-3-95806-459-1

Band / Volume 213
Direct measurement of anisotropic resistivity in thin films using a 4-probe STM
T. Flatten (2020), viii, 129 pp
ISBN: 978-3-95806-460-7

Band / Volume 214
The guided self-assembly of magnetic nanoparticles into two- and three- dimensional nanostructures using patterned substrates
W. Ji (2020), VI, 140 pp
ISBN: 978-3-95806-462-1

Band / Volume 215
Molecular layer deposition and protein interface patterning for guided cell growth
M. Glass (2020), iv, 81 pp
ISBN: 978-3-95806-463-8

Band / Volume 216
Development of a surface acoustic wave sensor for in situ detection of molecules
D. Finck (2020), 63 pp
ISBN: 978-3-95806-464-5

Band / Volume 217
Detection and Statistical Evaluation of Spike Patterns in Parallel Electrophysiological Recordings
P. Quaglio (2020), 128 pp
ISBN: 978-3-95806-468-3

Weitere ***Schriften des Verlags im Forschungszentrum Jülich*** unter
http://wwwzb1.fz-juelich.de/verlagextern1/index.asp

Schlüsseltechnologien / Key Technologies
Band / Volume 217
ISBN 978-3-95806-468-3

Mitglied der Helmholtz-Gemeinschaft

JÜLICH
Forschungszentrum