POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

ETH zürich

Improving Spike Sorting Performance of Bursting Neurons

Tesi di Laurea Magistrale in Ingegneria Biomedica

In Collaboration With Bioengineering Laboratory, ETH Zürich

Payam Sadeghi Shabestari

Student ID: 927647
Advisors: Prof. Alessandra Pedrocchi, Dr. Alessio Buccino, Dr. Sreedhar Kumar
Co-advisors: Dr. Manuel Schröter, Prof. Andreas Hierlemann
Academic Year: 2021-22

To all people fled their home ...

Abstract

For the understanding of how neural networks grow, learn and are able to fulfill their impressive functions, a reliable way to monitor their activity is crucial. In order to do so, one of the oldest techniques applied, is extracellular recordings. The main reasons for its broad usage are probably the low cost and relative ease of application. Extracellular recordings provide the possibility to detect spikes communicated through the neural network. In extracellular neural electrophysiology, individual spikes have to be assigned to their cell of origin in a procedure called "spike sorting". Spike sorting is an unsupervised problem, since no ground-truth information is generally available. Here, we focus on improving spike sorting performance, particularly during periods of high synchronous activity or so-called "bursting". Bursting entails systematic changes in spike shapes and amplitudes and remains a challenge for current spike sorting schemes. First, we use realistic simulated bursting recordings of high-density micro-electrode arrays (HD-MEAs) and we present a fully automated algorithm based on template matching with a focus on recovering missed spikes during bursts. Moreover, we suggest a method to limit manual curation of spike sorting output of bursting neurons by automatically merging units that have been over split due to bursting modulation. To compare and benchmark spike sorting performance after applying our method, we use ground-truth information of simulated recordings. Second, we propose an alternative approach to deal with bursting behaviour in more general cases. We use both simulated data and experimental recordings from primary dissociated rat hippocampal cell cultures to validate and benchmark our proposed method. For this purpose, we use HD-MEA recording combined with simultaneous patch recording. In this case the patch recording served as a ground-truth information to assess spike sorting performance. We show that our approach can be effective in improving spike sorting performance during bursting. Our method yields a reliable way for analyzing electrophysiological data of bursting neurons.

Keywords: extracellular recording, bursting, spike sorting, template matching

Sommario

Per comprendere come le reti neurali crescono, apprendono e sono in grado di svolgere le loro straordinarie funzioni, è fondamentale un metodo affidabile per monitorare la loro attività. Una delle più antiche tecniche applicate è la registrazione extracellulare. Le ragioni principali del suo ampio utilizzo sono probabilmente il basso costo e la relativa facilità di applicazione. Le registrazioni extracellulari offrono la possibilità di rilevare potenziali di azione trasmessi attraverso la rete neurale. Nell'elettrofisiologia neurale extracellulare, i singoli "picchi" devono essere assegnati alla loro cellula di origine in una procedura chiamata "spike sorting". L'ordinamento dei picchi è un problema non supervisionato, perché generalmente non sono disponibili informazioni di riferimento, sia sperimentali che da procedura standard. Qui, ci concentriamo sul miglioramento delle prestazioni di "smistamento dei picchi", in particolare durante i periodi di elevata attività sincrona o il cosiddetto "bursting". Bursting comporta cambiamenti sistematici nelle forme e nelle ampiezze "dei picchi" e rimane una sfida per gli attuali schemi di "smistamento dei picchi". In primo luogo, utilizzato registrazioni "esplosive" simulate realistiche di array di microelettrodi ad alta densità (HD-MEA) e presentiamo un algoritmo completamente automatizzato basato sulla corrispondenza dei modelli con particolare attenzione al recupero dei "picchi" mancati durante le "esplosioni". Inoltre, suggeriamo un metodo per limitare la cura manuale dell'output di smistamento dei picchi di neuroni durante burst unendo automaticamente le unità divise. Per confrontare e verificarne le prestazioni di "ordinamento dei picchi" dopo aver applicato il nostro metodo, utilizziamo informazioni di riferimento delle registrazioni simulate. In secondo luogo, proponiamo un approccio alternativo per affrontare il "comportamento esplosivo" in casi più generali. Utilizziamo sia dati simulati che registrazioni sperimentali ottenute da colture dissociate di cellule primarie ippocampali di ratto di ratto per validare e confrontare il nostro metodo proposto. A tale scopo, utilizziamo la registrazione HD-MEA combinata con la registrazione simultanea da microlettrodo (patch clamp). In questo caso la registrazione "delle patch" è servita come informazione di riferimento. Dimonstrato che il nostro approccio può essere efficace nel migliorare le prestazioni di "smistamento dei picch"i durante "lo scoppio".

Parole chiave: registrazione extracellulare, bursting, spike sorting, template matching

Acknowledgements

The work presented in this thesis was conducted from November 2020 to August 2021 at BEL group in the Department of Biosystems Science and Engineering of ETH Zurich. I would like to thank Alessandra and Andreas to have given me the chance to start this journey. Thank you for believing in me and giving me all the support, I needed. Alessio, you have been a great supervisor. I would really like to thank you for allowing me to follow and pursue my interests, always checking that I was on track. Sreedhar, this thesis would not have been possible without your input, insightful feedbacks and supervision. And Manuel thank you for collecting data and your constructive recommendations on this project. Thank you guys, I'll see you down the road !!!

Basel, September 2021

Payam S. Shabestari

Contents

Abstract **i**

Sommario **iii**

Acknowledgements **v**

Contents **vii**

1 Introduction **1**
1.1 Bursting Neurons . . . 2
1.2 Intra and Extracellular Recordings . . . 3
1.3 Spike Sorting . . . 4
1.4 Template Matching . . . 6
1.5 Impact of Bursting on Spike Sorting . . . 8
1.6 Structure of Thesis . . . 11

2 Methods **13**
2.1 Simulated Recordings . . . 13
2.1.1 MEArec Workflow . . . 14
2.1.2 Simulated Bursting Recordings . . . 15
2.2 Spike Sorting Benchmark and Evaluation . . . 16
2.2.1 Matching Firing Events and Computing Agreement Score . . . 17
2.2.2 Matching Units . . . 17
2.2.3 Performance Metrics . . . 18
2.3 Strategies to Improve Spike Sorting Performance . . . 19
2.3.1 Modulated Template Matching . . . 20
2.3.2 Merging Procedure for Burst-related Oversplits . . . 23
2.4 Experimental Recordings . . . 24
2.5 Model-free Approach . . . 25

3 Results **27**
3.1 Modulated TM Recovers Missed Spikes 27
3.2 Merging Potentially Split Units Improved Agreement Scores 31
3.3 Model-free Approach Recovered Undetected Spikes 33

4 Discussion **37**
4.1 Future Developments 37
4.2 Outlook 39

Bibliography **41**

5 Appendices **47**

A Executive Summery **49**

B Conference Paper **61**

C Pseudo-algorithms **67**

D Matched Filtering **69**

List of Figures **71**

List of Tables **73**

1 | Introduction

"All outstanding work, in art as well as in science, results from immense zeal applied to a great idea."

-Santiago Ramon y Cajal

The brain is what it is because of the structural and functional properties of interconnected neurons. The brain's unique abilities, such as perceptions, memory, and learning are believed to emerge from the interactions of networks of neurons. A neuron or nerve cell is an electrically excitable cell that communicates with other cells via specialized connections called synapses. The membrane of the axon, soma and dendrites in a neuron cell contains voltage-gated ion channels that allow the neuron to generate and propagate an active electrical signal (an action potential). These signals are generated and propagated by charge-carrying ions including sodium (Na^+), potassium (K^+), chloride (Cl^-), and calcium (Ca^{2+}). Extracellularly, if an electrode is close enough to a neuron firing an action potential, we observe a fast transition in the recorded electric potential, that we refer to as *spike*.

Extracellular recordings have been historically performed with single microwires or bundles of microwires, capable of recording a few tens of neurons per experiment. Traditional microelectrode arrays (MEAs) are circuitless chips carrying arrangements of usually 60 electrodes that are used for multisite extracellular recording from electrogenic cells, such as neurons or heart cells, provided higher density and electrode counts than microwire technology [6, 53, 57]. They are used in the fields of neuroscience and biosensing, to study fundamentals of neuronal information processing, of learning processes, of aging and mental diseases, or to assess the behavior of electrogenic cells upon exposure to pharmacological agents or hazardous substances. The development and use of CMOS-based (Complementary Metal-Oxide-Semiconductor) devices can overcome limitations of passive MEAs, in particular with respect to performing measurements at high spatial and temporal resolution with excellent signal quality. Through on-chip signal multiplexing, a large number of electrodes can be addressed to achieve a high spatio-temporal resolution in recording and stimulation scenarios. Moreover, weak neuronal signals can be

conditioned right at the electrodes by dedicated circuitry units, which provide a large signal-to-noise ratio [61]. Finally, analog-to-digital conversion is performed on chip, so that only stable, digital signals are transmitted.
In addition to enabling researchers to record the activity of several hundred neurons, the high spatial resolution of the electrodes enables the observation of the same action potentials on many different recording contacts. The known spatial geometry of the recording sites provide a suitable approach to study neurons, from extracellular signals, even at the sub-cellular level.

1.1. Bursting Neurons

Neurons typically operate by firing single action potentials in relative isolation as discrete input postsynaptic potentials combine and drive the membrane potential across the threshold. The collective interaction of neurons in networks gives rise to various complex emergent phenomena; an important example is the coherent occurrence of closely spaced action potentials called network burst. Bursting is an extremely diverse phenomenon, where periods of rapid action potential firing are followed by quiescent periods much longer than typical inter-spike intervals [54]. Bursting can occur for many reasons, but neurons can be generally grouped as exhibiting input-driven or intrinsic bursting. Most cells will exhibit bursting if they are driven by a constant, subthreshold input and particular cells which are genotypically prone to bursting (called *bursters*) have complex feedback systems which will produce bursting patterns with less dependence on input and sometimes even in isolation [29, 31]. In neuronal networks in vivo, network bursts play crucial roles during development and in information processing and have also been linked to disorders like epilepsy [30, 34, 49]. In neural computation, there are many hypotheses on the importance of bursting activity. Some are listed below:

- Bursts are more reliable than single spikes in evoking responses in postsynaptic cells.
- Postsynaptic responses to a single presynaptic spike may fail (release does not occur), however in response to a bombardment of spikes, i.e., a burst, synaptic release is more likely [33].
- Bursts have higher signal-to-noise ratio than single spikes.
- Bursts encode different features of sensory input than single spikes [17, 38].
- Bursts have more informational content than single spikes when analyzed as unitary events [44].

- Bursts can be used for selective communication if the postsynaptic cells have sub-threshold oscillations of membrane potential [30].

Observed in a wide variety of experimental preparations, network bursts are thought to be universal to networked neuronal systems [8, 41]. Despite their significance, we do not yet fully understand the principles governing the initiation and maintenance of network bursts. Figure 1.1 shows intracellular recording of a bursting neuron and extracellular raster plot of a bursting network over 1000 electrodes.
How to detect bursts? ISI-threshold detectors consider that periods of low and high ISIs correspond to spikes occurring within and outside of bursts, respectively, and analyzing peaks in the probability distribution of the ISIs can identify appropriate thresholds which are commonly based on finding valleys in the distributions of plain ISIs or the logarithm of ISIs or from the discharge density [2, 32, 40, 54].

1.2. Intra and Extracellular Recordings

Two fundamentally different approaches to measure electrical signals from neurons have been established and extensively used to study brain function and underlying mechanisms: intracellular and extracellular recordings. In intracellular recordings, the voltage across the cell membrane is measured. Intracellular recordings are typically performed using the patch clamp technique [36], where a microelectrode is placed close to the neuronal membrane, and a high-resistance seal between the micropipette and the cell membrane is

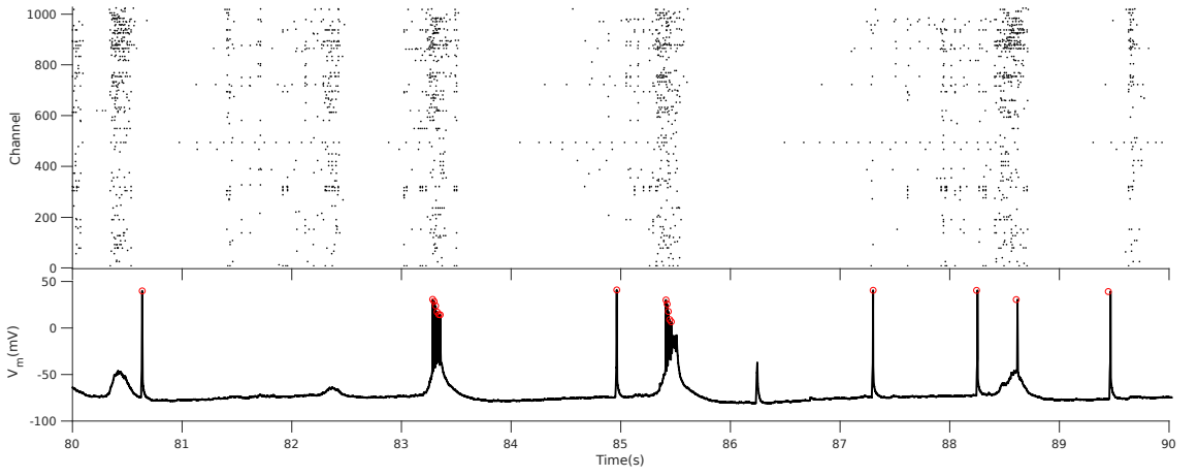

Figure 1.1: A raster plot of the extracellular spikes detected by an HD-MEA over 10 seconds showing the network burst phenomenon. The lower panel shows the membrane potential of a single current-clamped neuron recorded simultaneously from the same network. Individual spikes are marked in red.

formed by the applied negative pressure. By further applying pulses of negative pressure, the membrane between the micropipette and the cell is then disrupted and the pipette solution comes into contact with the intracellular volume [35]. Intracellular recordings feature an excellent signal-to-noise ratio (SNR) and can be used to record subthreshold signals, such as postsynaptic potentials. The method is, however, time consuming, laborious and low-throughput, as one individual cell is measured per micropipette. As it is an invasive method, the time of the experiment is limited to a few hours. Extracellular recordings, in contrast, represent a non-invasive method and can be carried out over long time scales. Multi electrode arrays (MEA) provide a label-free method to measure extracellular action potentials produced by electrogenic cells. The tissue is placed on top of the measurement electrodes and kept alive in a perfusion medium (see figure 1.2). When a cell fires an action potential, it causes a rapid influx of positively charged sodium ions into the cytosol, prompting a voltage drop at the measurement electrode. This is followed by a rapid outflux of positively charged potassium ions, which hyperpolarizes the cell and is measured as a voltage increase. Extracellular signals are a superposition of the activity of a number of neurons in the vicinity of the electrode. Given the high density multi electrode arrays (HD-MEA), each neuron is also recorded by multiple electrodes close to it. A further computational step is thus necessary to disentangle the recorded signals and assign single spikes to their putative neuronal sources. The combination of HD-MEA technology with the patch clamp technique can be used for both understanding and validating extracellular features and characteristics of extracellular data [23] and evaluation of spike sorting algorithms [21, 25].

1.3. Spike Sorting

The detection and classification of voltage deflections in extracellular recordings caused by action potentials of neurons, called spikes, is known as spike sorting. On single electrodes, spikes often differ in amplitude and waveform depending on the type, location, and morphology of the cells and the surrounding tissue. Spike sorting results and performance are very limited in such recordings, as spikes cannot be unambiguously assigned to the respective neurons. "Fortunately, with HD-MEAs extracellular action potentials can be detected on multiple electrodes at the same time. This gives us additional information on the location of cells and on the waveform shapes across a two-dimensional area". The assignment of spikes to neurons is only possible because the spike shapes of neurons differ due to their morphology, their spatial position with respect to the recording electrodes,

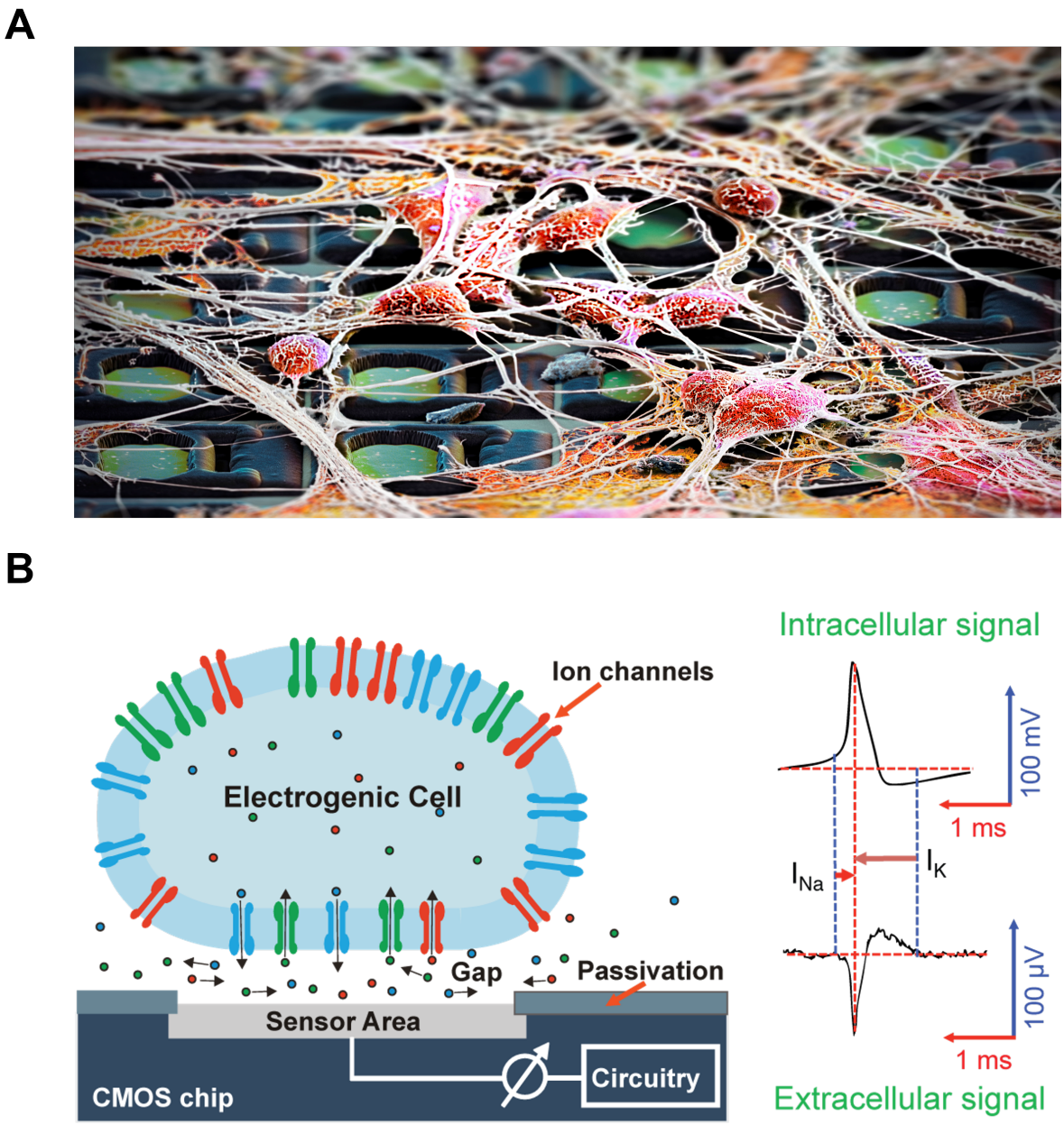

Figure 1.2: **A.** Neurons plated on a high-density microelectrode array. The small electrode size and the tight spacing between the 26400 electrodes enable functional extracellular electrophysiological characterization of neurons, from subcellular-resolution features, like axons and dendrites, through individual neuronal cells to entire networks[1]. **B.** Schematic of Neuron-electrode interface in a network of neurons for recording extracellular APs; Compact system chips comprise recording and stimulation electronics for bidirectional communication with neurons[2].

the intrinsic membrane properties of the neuron and the surrounding medium [7].

[1] *photo taken by Martin Oeggerli (https://www.micronaut.ch/)*
[2] *https://bsse.ethz.ch/bel/*

Moreover, at least to a first approximation, spikes from the same neuron have similar waveforms. Therefore, it is feasible to group extracellular spikes based on their waveform assuming that spikes within one group were emitted by the same neuron. Spike sorting algorithms despite mild variations, can all be described given the schematic shown in Figure 1.3. First, the raw extracellular signals are filtered to remove the irrelevant information for spike sorting. This step involves filtering via a band-pass filter to remove the frequencies related to Local Field Potential (LFP). For this purpose, a digital filter, e.g., between 300 Hz and 3000 Hz, is typically used [45]. Moreover, in many cases, before spike sorting, it is wise to re-reference the signals to reduce the common-mode noise from the recordings using common average reference (CAR) or common median reference (CMR) filtering (panel 1). After pre-processing, spikes are detected as threshold crossings and extracted from the data (panel 3). Since the detection threshold is related to the signal to noise ratio (SNR) of the recording, this certain threshold is set with respect to some dispersion measures like standard deviaton or median absolute deviation of the signal. After the spike detection phase, snippets of the spikes are aligned in time (panel 4). In the next step, spikes are projected into lower dimensional space (panel 5). One of the most common feature extraction and dimensionality reduction methods is principal component analysis (PCA). PCA gives an ordered set of orthogonal vectors that represent the directions of largest variations in the data and any waveform is represented as a linear combination of the principal components. This way, the dimensionality reduction is achieved by going from an M-dimensional space to a K-dimensional space, with $K << M$. However, there are alternative approaches such as Independent Component Analysis (ICA) transforms [50, 51] and wavelet transforms [27, 43, 52], used in spike sorting to project spikes into lower dimensions. In the next step, spikes are clustered based on the features (panel 6). An extra classification step could be applied to any spike sorter as a post-processing step to improve quality of sorting. This step is called template-matching (panel 7). One way to build such a classifier is to calculate the average of all elements (waveforms) for each cluster. This cluster center is called the template. Then by comparing each unclassified spike to each unit's template, we can assign each spike to a template which is most similar to it, using some appropriate similarity measure [16].

1.4. Template Matching

The first step of template matching is moving the template over the entire signal and calculating the similarity between the template and the covered window on the signal. In the spike sorting problem, templates are every cluster centroid i.e., the average spatio-temporal spikes on the extracellular channels. The template matching approach sub-

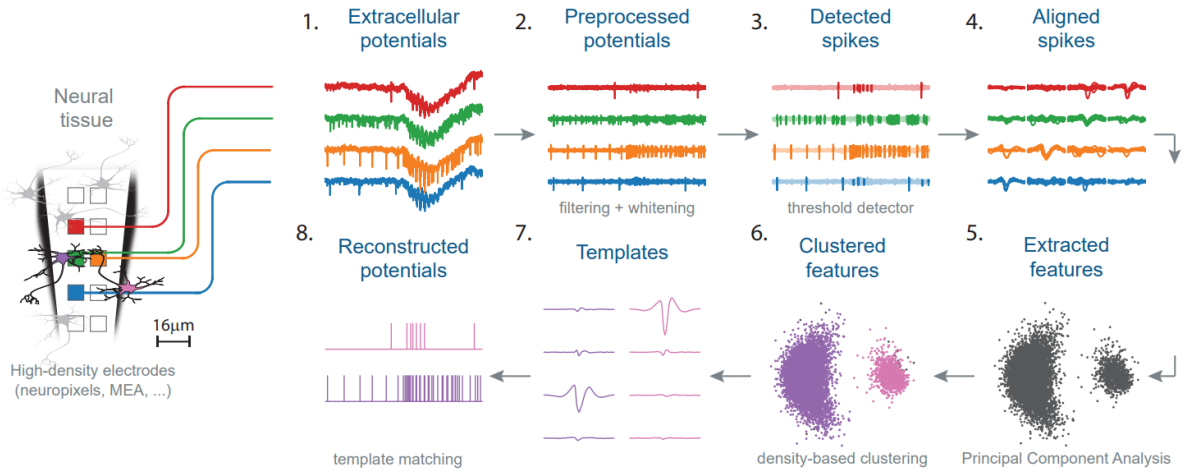

Figure 1.3: Overview of the generic spike sorting workflow [3]; After pre-processing of recorded extracellular signals by means of noise suppression, spikes are detected by thresholding over traces. After aligning detected spikes, it is compulsory step to assign each detected spike to a cluster, which is performed via selecting and extracting features from spikes. An optional step could be applied as computing similarity between any element and center of clusters, called template matching.

stantially improves detection of single-unit spiking over clustering alone, especially in the presence of spike collisions. Different approaches to solve this problem were proposed; from simple strategies like Euclidean distance with manual threshold [19] and convolutional methods to more complex strategies including filter-based methods [46]. In the Euclidean distance method, templates of each electrode are individually subtracted at each point from the data and the norm of the residual over all electrodes is computed. If this residual is close to zero, the template is assumed to be present in the data at the respective temporal position. In the convolutional template matching technique, for each multi-electrode template the data from every electrode is convolved with the respective single-electrode template individually and the results are summed as illustrated in Figure 1.4. In the matched-filter methods, matched filters are computed from the template and the noise covariance matrix, then filters are convolved with the traces instead of templates and the results are summed. For all three methods spikes would have to be detected in the template matching outputs by thresholding. However it was not thoroughly investigated yet what the best strategy is for template matching.

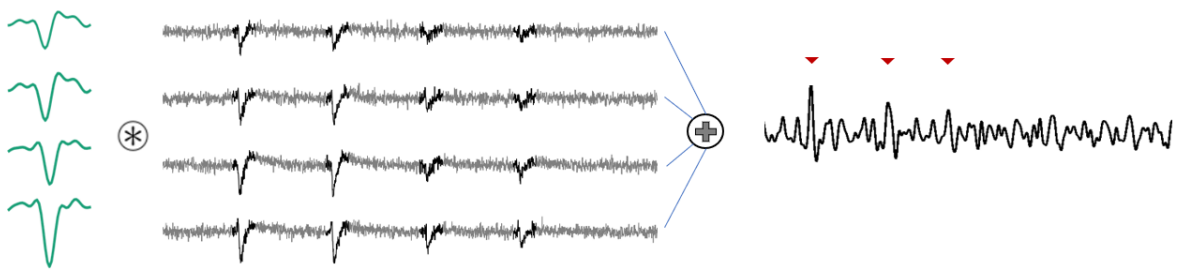

Figure 1.4: Illustration of template matching technique; The first column of panels shows the templates extracted from the initial spike sorting for each electrode individually, the second column shows the short piece of traces in 4 different channels and the last column the final multichannel template matching output. Spikes would have to be detected in the template matching outputs by thresholding (in this case on maxima of the output). For each channel the corresponding template is convolved with the extracellular trace probed by the electrode individually and the results are summed.

1.5. Impact of Bursting on Spike Sorting

In general, bursts can be generated as a result of the intrinsic properties of a single neuron, or as a result of local network activity [14]. Different transmembrane ionic currents may result in different theoretical mechanisms of burst generation, which strongly influence burst statistics (e.g., inter-spike-intervals (ISIs), spikes amplitude and burst duration) [62]. At least four different mechanisms of modulation of spiking within bursts are presented. However, many neurons have a mixture of these mechanisms [1, 22, 26, 55]. In all these cases, inter-spike-intervals (ISIs) and the amplitude of the spikes vary within the burst. However, the relationship between electrophysiology of a cell and its effect on bursting topology and statistics is not systematically understood yet.
How do intracellular and extracellular waveforms transform during bursting? To answer this question, we analysed variation of three different spike features within bursts for both extracellular and corresponding intracellular (ground-truth) recordings from primary dissociated rat hippocampal culture. These features include spike's amplitude, peak derivative and full width half maximum (FWHM). Peak derivative is the slope of a tangent line at the peak point and FWHM is defined as width of the spike at a level that is just half the maximum ordinate of the peak. Intracellular action potentials are detected by means of thresholding over zero milivolt. In Figure 1.5, APs within bursts are ranked (ordered in bursts) and cut out (panel A). Amplitudes (panel B), FWHMs (panel C) and peak derivatives (panel D) are computed for each spike and their corresponding distribution (violin plots) are plotted with respect to spike ranks. Figure 1.5 outlines that amplitude

and peak derivative of spikes moves toward smaller values (as expected) and FWHM of waveforms takes larger values as the burst proceeds.

To analyze extracellular spikes transformation, spikes are initially cut out with respect to intracellular ground-truth action potential, then channel with largest template size is selected, and the same aforementioned features variation are investigated. Figure 1.6.A, B and C depict the variation of extracellular spikes's amplitude, FWHM and slope within bursts respectively. As intracellular action potentials, amplitudes and slopes (absolute values) of the extracellular spikes are reduced as proceeding within bursts, however variation of full width half maximum of the spikes is not as large. To better visualize the spikes amplitude decrease within bursts, we averaged the spikes with respect to their ranks, then we applied a normalization with respect to first spike amplitude (largest spike), then a decaying exponential function is fit (see Figure 1.6.D).

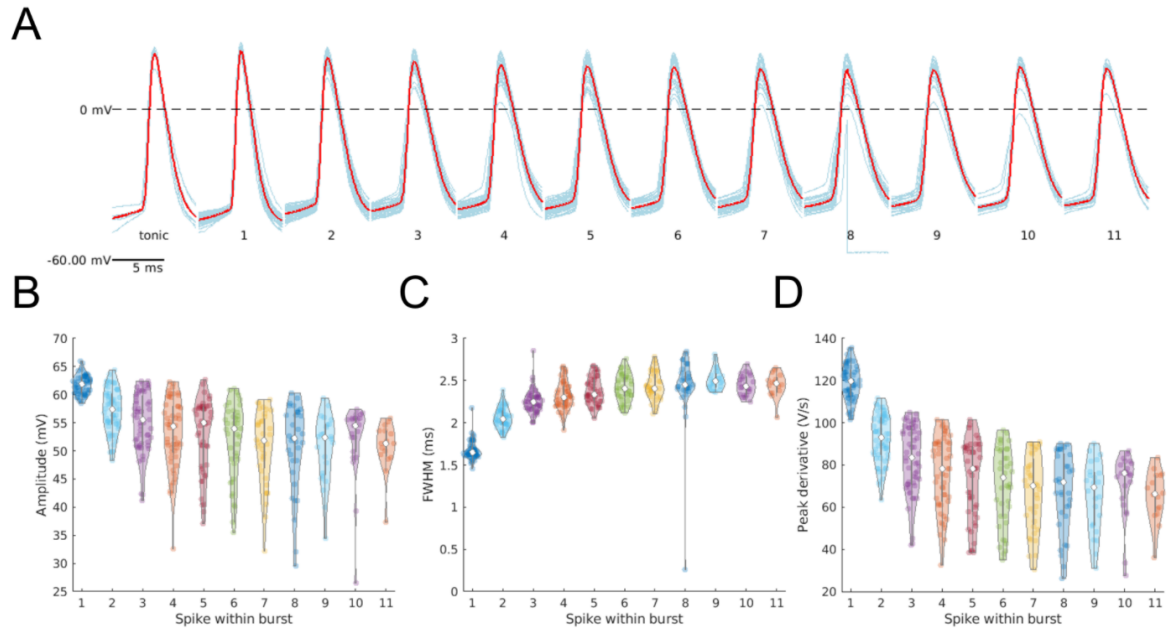

Figure 1.5: Intracellular waveform changes during bursting; **A**. Individual spike waveforms from all bursts are ranked and superimposed, illustrating the successive change in amplitude and shape of the spike waveform. Red waveforms represent the mean AP derived by averaging cyan action potentials. **B**. Distribution of spike amplitudes as a function of their rank, indicating a decay in proceeding spikes amplitude over bursts. **C**. Full width half maximum variation indicates that spikes are also modulated in shape during bursting. The latter spikes are stretched more with respect to earlier spikes. **D**. Distribution of peak derivatives over spike ranks is an indicator of decay in spikes amplitude within bursts.

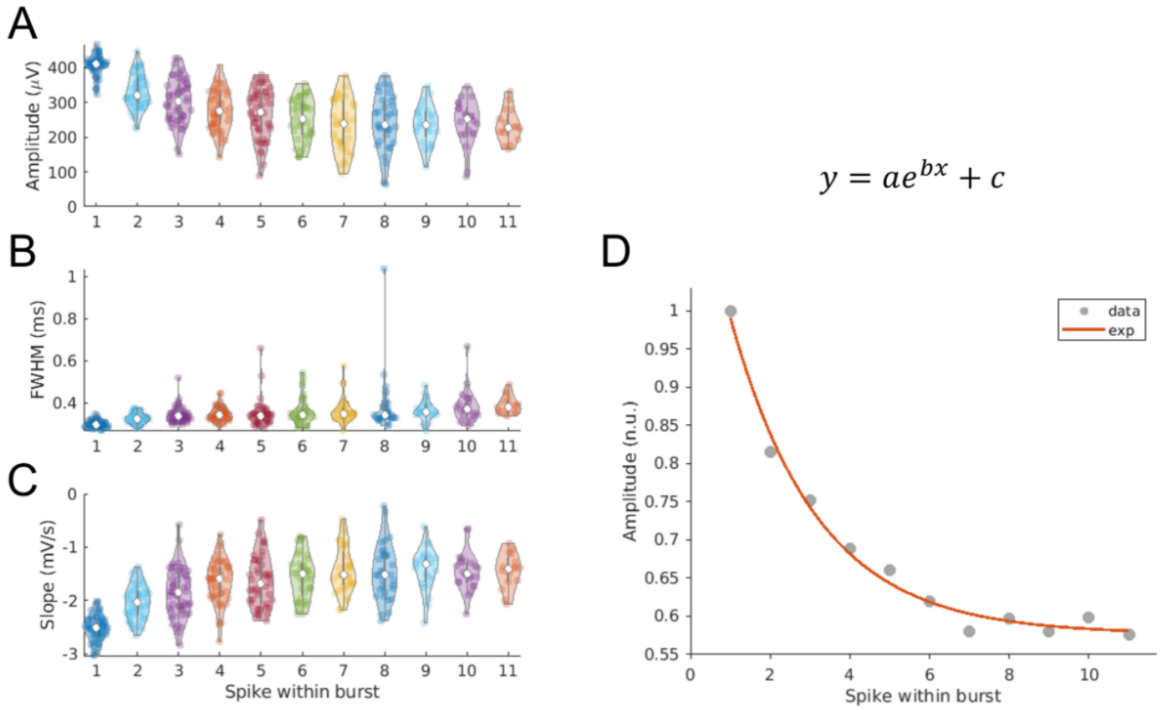

Figure 1.6: Extracellular waveform changes during bursting; **A**. Violin plots outlining distribution of extracellular spike amplitudes at each spike rank; the individual spike amplitudes are colored with the same color of distribution cloud and the mean value of the amplitudes are colored in white. Spikes size are reduced to approximately 60 percent of largest spike as proceeding in bursts. **B**. Full with half maximum (FWHM) of spikes at the channel with largest template is computed and their distribution as a function of spike's rank is plotted. The variation in mean values (white dots) of each distribution does not indicate a significant change in bursting spikes. **C**. Slopes of spikes between initial peak and the minimum value as an alternative indicator of spikes ampliude are computed for each spike. Mean values of violin plots (white dots) decrease in absolute value as the ranks increase. **D**. Normalized mean values of the amplitudes are plotted in grey dots and an exponential function (red line) is superimposed to better visualize amplitude decay of bursting spikes.

A decrease in spike amplitude that occurs during successive spikes in a burst is the principal factors contributing to the misclustering of bursting waveforms, since spike sorting algorithms assume that spikes of the same neuron should remain relatively stationary over time. In addition to the failure mode of misclustering, spikes that occur late in burst events potentially may fall below the threshold of detection and remain undetected (see Figure 1.7).

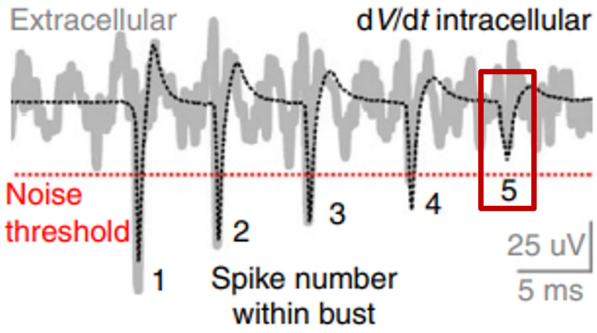

Figure 1.7: Extracellular trace of a burst with the superimposed first derivative of the corresponding intracellular waveform. The noise threshold used for spike detection is indicated by the red dotted line highlighting that the later spikes in a burst typically fall below the detection threshold [28].

An important challenge for spike sorting algorithms is the observed variability of spike amplitudes and shapes that depends on the recent firing history of the neuron [13]. A prominent example is neuronal bursting: patch-clamp experiments, combined extracellular-juxtacellular recordings [12] and computational studies [20] suggest that, during bursting, extracellular spikes become lower in amplitude (see Figure 1.6). If unaccounted, such amplitude changes may contribute to spikes of a certain unit being erroneously assigned to other units (false positives) or being missed altogether (false negatives). Our approach to address this problem will be in two phase: in the first phase we will use simulated realistic extracellular recordings, in which we have a full control over bursting and ground-truth data. In the second phase we will have a deeper insight over bursting and will introduce a modified approach for improving spike sorting of bursting neurons by using simultaneous combined intra and extracellular recordings.

1.6. Structure of Thesis

The thesis is organized as follows: in chapter 2, first we describe the workflow of extracellular recording simulator and the method used for simulating bursting recordings in 2.1. Next in section 2.2 we describe the approach for evaluating the spike sorting outputs, then we illustrate the algorithms for recovering missed and falsely assigned spikes by spike sorters within bursts in section 2.3. In sections 2.4 and 2.5 we introduce a modified approach for more advanced bursting activities using experimental recordings. We depict how it can improve the overall spike sorting performance in chapter 3. Finally, we contextualize this work by abstracting its potential and limitations and by indicating future directions in chapter 4 (Discussion).

2 | Methods

"Our brain simulates reality. So, our everyday experiences are a form of dreaming, which is to say, they are mental models, simulations, not the things they appear to be."

-Stephen LaBerge

For the performance assessment of a spike sorter or any post-processing method, it is necessary to have a benchmarking dataset in which the exact spike times of many neurons are known. In this chapter we describe how to simulate realistic extracellular bursting recordings with user defined spike times.

2.1. Simulated Recordings

Why we need simulated recordings? In paired recordings a single neuron is isolated using a high SNR (signal-to-noise) single cell approach, such as intracellular or juxtacellular recordings, that provides an accurate record of the target cell's spike times. Another recording device, e.g., a MEA probe, is then lowered into close vicinity of isolated neuron, then probe recording is analysed with a spike sorting algorithm. The initial problem with this approach is that spike sorting performance can be partially validated using the ground-truth spike times obtained from the intracellularly or juxtacellularly recorded cell, because only one ground-truth neuron is recorded at a time. Another problem with paired recordings is that they involve an expensive, time consuming procedure, which often fails, since it is very difficult to register the activity of one neuron on both recording devices. Therefore a precise mutual positioning of both recording devices is required for recordings to succeed. As opposed to paired approach, recordings generated from simulators provide a full ground-truth information used for validating spike sorter performance. Albeit, simulated recordings suffer from lack of the richness of real recordings but are more flexible in coping with changing recording settings; for example, variety of bursting modulations in bursting recordings.

2.1.1. MEArec Workflow

In order to generate multiple recordings with bursting units, we used the `MEArec` Python package [4]. Combination of `MEArec` and `MEAutility`[1] Python packages gives the user the possibility of dealing with various probe configurations including square multi-electrode arrays (MEA) and investigating the effect of these probe models on extracellular recordings. `MEArec` simulator makes possible to study and explore the generated recordings with full control, on various scenarios such as bursting activity, drifting, overlapping spikes and environmental noise. `MEArec` simulator works in two stages: templates generation and recordings generation as depicted in Figure 2.1.

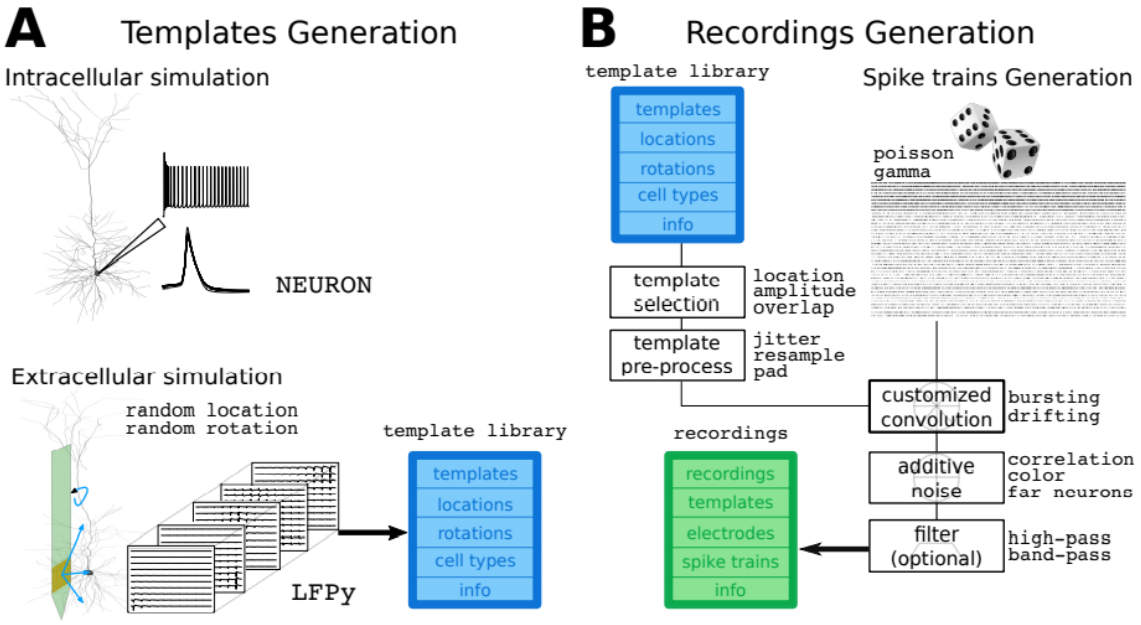

Figure 2.1: Overview of extracellular recording simulation by `MEArec`; **A**. Template generation phase; the multi-compartment model of neuron is run with intracellular simulations (only needs to run once), then the computed transmembrane currents are used to generate extracellular action potentials (EAPs) and finally templates are saved. **B**. Recording generation phase; templates from *template library* are processed and convolved with user-defined spike trains. In the case of bursting, convolution operation is customized with respect to selected values by user. Finally an uncorrelated Gaussian noise is added to the traces [4].

In the template generation phase, templates refer to the extracellular action potentials (EAPs) generated by a neuron on the recording sites. Templates are generated in two

[1] *https://meautility.readthedocs.io/en/latest/*

steps: In the first step, NEURON simulation environment is used for modeling individual neurons. Transmembrane currents for each cell model is computed and saved, so the currents do not need to be recomputed. Then, these transmembrane currents are used to simulate extracellular action potentials (EAPs) by using LFPy package[1]. LFPy is a Python module for calculation of extracellular potentials from multicompartment neuron models, which relies on the NEURON simulator. Cells are randomly placed (and optionally rotated) around the selected probe before the extracellular potentials are computed. In the second stage, noted as recording generation, recordings are generated by combining templates and user defined spike trains. After convolving selected templates and corresponding spike trains, temporal jitter is added to them to simulate the uncertainty of the spike event within the sampling period. Moreover, after convolution, additive gaussian noise is generated and added to the recordings. To simulate bursting neurons extracellular activity, MEArec allows user to select units (neurons) to burst, in which modulated convolution between templates and corresponding neurons spike train is done. This amplitude modulation is function of inter-spike-intervals and modulation value predefined by user [4].

2.1.2. Simulated Bursting Recordings

In order to generate bursting recordings, we first simulated sets of random bursting spike trains by selecting random values for ISIs (inter-spike-intervals), IBIs (inter-burst-intervals) and number of spikes in the bursts. Second, templates from biophysically realistic cell models were selected that did not overlap in space to limit the presence of spatiotemporal collisions (this was done by setting parameter $sync\_rate = 0$ in MEArec) and then convoluted with the spike trains by applying an amplitude modulation; The amplitude modulation consists of scaling the amplitude of each spike event with a modulation value. This procedure included a random and small Gaussian modulation to reproduce physiological variations and a bursting term, which was computed as:

$$\mu[n] = \left(\frac{\bar{t}_{isi}}{(n-1) \cdot t_{max}} \right)^{\lambda} + N(0, \sigma^2) \tag{2.1}$$

where $\mu[n]$ is the amplitude modulation factor of the n^{th} consecutive spike within a burst ($n \geq 2$), $\bar{t}_{isi}$ is the average inter-spike interval within a burst, t_{max} is the maximal duration of a single burst (used as a normalization factor), and λ is an exponent that regulates the degree of attenuation. The spike amplitudes generated by each neuron depend on the

[1] *https://lfpy.readthedocs.io/en/latest/*

elapsed time since the previous spike of this neuron and the measured spike amplitudes are corrupted by a Gaussian white noise, which sums linearly with the spikes and is statistically independent of them. An example of simulated bursting recording is shown in Figure 2.2. Figure 2.2.A displays generated spike trains for five neurons with random values for ISIs (inter-spike-intervals), IBIs (inter-burst-intervals) and number of spikes in the bursts. A square MEA with 15 micrometer inter electrode distance, used for probing is shown in Figure 2.2.B and Figure 2.2.C shows one burst from the generated recording in 4 channels with an exponent factor λ set to 0.2.

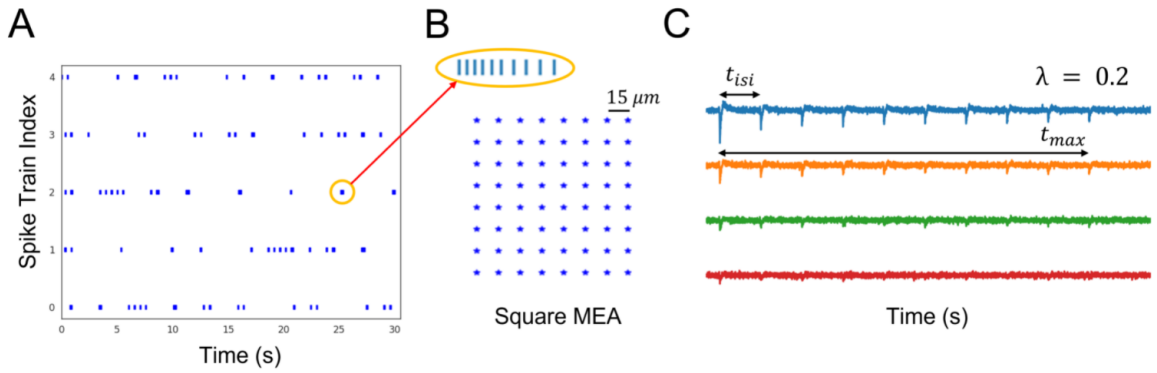

Figure 2.2: **A**. Generated bursting spike trains with for 5 individual neuron with duration of 30 seconds; ISIs (inter-spike-intervals), IBIs (inter-burst-intervals) and number of spikes in the bursts are selected randomly. Moreover, later ISIs in the bursts are elongated to mimic real bursting activity. **B**. Geometrical representation of 10*10 square multi-electrode arrays with inter probe distance of 15 μm. **C**. A simulated burst trace probed by MEA in 4 different channels; Spikes amplitude are modulated within burst by setting λ exponent factor to 0.2.

2.2. Spike Sorting Benchmark and Evaluation

Spike sorting algorithms are intrinsically unsupervised, since the spiking activity of underlying neurons-so-called ground-truth (GT) is not known. However there are approaches to compare output of two or multiple spike sorters [5]. In order to validate and benchmark a spike sorter or any post-processing step, one needs to use ground-truth data. There exist several experimental and computational strategies to yield this information, including **paired recordings**, in which a GT of only one neuron cell (intracellularly or juxtacellularly recorded cell) is known [37, 56]; **hybrid recordings** which augments artificial spiking activity to real experimental recording [47, 59] and **synthetic recordings** that utilize simulations based on cell models and spike trains and provide full benchmark of spike sorting results [4].

We compared spike sorting outputs with ground-truth spiking activity from the `MEArec` simulated recordings. The main workflow used to compute perfomance metrics and evaluate spike sorting performance is as following:

2.2.1. Matching Firing Events and Computing Agreement Score

To evaluate the performance of the spike sorting algorithms, the resulting sorted units were compared with a given "ground-truth". To quantify accordance with and deviations from the ground truth, we calculate the following numbers for all pairs of ground truth units and tested units: we first count how many events are matched within a small Δt tolerance. In another words, two spikes from two different spike trains are *matched* when they occur within a certain time window of each other. We set the window length Δt equal to $0.4ms$. The agreement score between two spike trains can be computed as:

$$agreement\_score = \frac{\#n_{matches}}{\#n_1 + \#n_2 - \#n_{matches}} \tag{2.2}$$

Where $\#n_{matches}$ is the number of *matched* spikes between the two spike trains (one from GT units and one from tested sorter units) and $\#n_1$ and $\#n_2$ are the number of spikes in the first (GT unit) and second spike train (tested unit), respectively. The agreement score can be seen as the ratio of two spike trains intersection and their union, which is also equivalent to the accuracy metric.

2.2.2. Matching Units

During this step, given the agreement score matrix each ground truth units can be matched to a tested unit. For matching, a minimum *match score* is used (we set *match score* to 0.1). If the *agreement score* is below this threshold, the possible match is discarded. There are two ways to perform the match: *Hungarian* and *best match*. With the *Hungarian* method, each tested unit from the sorting output is matched to at most a single ground-truth unit. With the *best match* method, a tested unit from the sorting output can be matched to multiple ground-truth units (above an agreement threshold).
For all pairs of ground-truth and sorted spike trains, we first counted how many events were matched within a time tolerance of 0.4 *ms*. We used a *best match* approach, in which each ground-truth unit was associated with a test unit that had the best (maximum) *agreement score*, independently of all other units. The advantage of using *best match* method is that, each ground-truth unit is matched totally independently from other units, therefore the accuracy score of a GT unit is totally independent from other

units. Furthermore, the *best match* approach can identify over-merged units, as multiple ground-truth units can be matched to the same sorted unit. Figure 2.3 displays an example of agreement score matrix of recording with 5 units spike sorted by *Tridesclous* [18] spike sorter and full ground-truth data from `MEArec` simulator. Agreement scores between any pairs between GT and tested sorter are computed, *best match* units are selected and depicted by green boxes with agreement score values at the center.

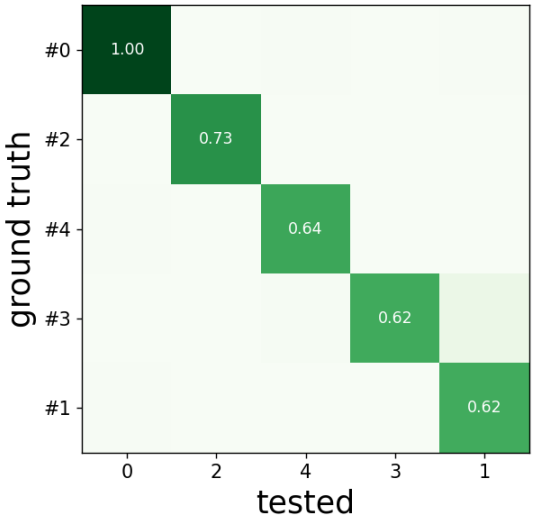

Figure 2.3: An example of the agreement matrix of recording with 5 units, computed between ground truth data from `MEArec` and *tridesclous* sorter output; in which only scores greater than 0.5 are displayed. Given the agreement score matrix each ground truth unit can be matched to a tested unit. For matching, agreement score should be greater than 0.1. If the agreement score is below this threshold, the possible match is discarded. *Best match* units are colored by green with agreement score values at the box center.

2.2.3. Performance Metrics

With the list of matched units, to quantify overall performance of spike sorter with respect to ground truth, first we define following numbers:

- **TP** true positives, i.e., correctly assigned spikes: a spike of a certain single unit was sorted correctly into best-matched unit.
- **FP** false positives (sorted), i.e., wrongly assigned (misclassified) spikes: a spike of a certain sorted unit does not correspond to a spike in a matched GT unit.

- **FN** false negatives, i.e., spikes wrongly left unsorted: a spike was given as element of a certain single unit but was left unsorted.
- **TN** true negatives, are not defined due to sparse nature of spike trains.

Thus, a 100% correct classification contains only TPs. For each recording sorted by spike sorter, we count the corresponding hits (TPs) and misses (FPs and FNs). We calculate the following measures:

$$accuracy = \frac{\#TP}{\#TP + \#FN + \#FP} \tag{2.3}$$

$$precision = \frac{\#TP}{\#TP + \#FP} \tag{2.4}$$

$$recall = \frac{\#TP}{\#TP + \#FN} \tag{2.5}$$

The accuracy describes how many spikes out of all spike events are correctly assigned. Precision illustrates proportion of truly sorted spikes in the list of all sorted spikes and recall is the ratio of spikes of the ground truth units that were correctly matched to spikes of matched sorted test units to the number of spikes that could have been sorted. However, there are other metrics such as *miss rate* $(\#FN/\#TP + \#FN)$ or *false discovery rate* $(\#FP/\#TP + \#FP)$, which represent the fraction of accesses that result in a miss and percentage of finding false positive from detected events, respectively.
All spike sortings and comparisons are performed using `SpikeInterface` platform [5]. Using `SpikeInterface`, it is possible to create a *"study folder"* in which, a systematic performance comparisons of several ground truth recordings with several sorters is done in one shot. An example of a simulated recording sorted by *"HDsort"* [11] and computed performance metrics is shown in table 2.1.

2.3. Strategies to Improve Spike Sorting Performance

In this section, first we propose an approach to recover spikes that are missed during initial spike sorting phase due to spike amplitude variation within bursts and second, we illustrate a criteria that automatically merge bursting units that are incorrectly split.

[1] *false discovery rate*
[2] *miss rate*

Performance Metrics

	sorter	gt unit id	accuracy	recall	precision	fdr[1]	mr[2]
rec1	HDsort	0	0.467	0.666	0.609	0.390	0.333
rec1	HDsort	1	0.176	0.220	0.470	0.529	0.779
rec1	HDsort	2	0.523	0.539	0.944	0.055	0.460
rec1	HDsort	3	0.495	0.637	0.689	0.310	0.362
rec1	HDsort	4	0.685	0.904	0.739	0.260	0.095

Table 2.1: Comparison results of one recording with 5 GT units sorted by "HDsort" spike sorter; Performance metrics (accuracy, recall, precision, false discovery rate and miss rate) computed on a study folder, containing multiple recordings spike sorted by multiple spike sorter by comparing spike sorters output and ground-truth data.

2.3.1. Modulated Template Matching

We implemented a post-processing step based on template-matching technique that limits number of undetected spikes by spike sorter introduced through spike amplitude variability [48]. The first step includes extracting waveforms by combining spike sorter output with the recording traces to compute unit templates (i.e., the average extracellular waveforms), then we utilize extracted template in the time windows that are prone to have missed spikes. Considering that the missed spikes within bursts have smaller amplitudes, we introduce a novel technique based on bayesian optimal template matching (BOTM) approach. Bayes optimal template matching (BOTM) computes matched filter outputs for the given template and adds constants that depend on the energy of the template and the probability that it occurs in the data to compute the final discriminant function [16]. For each unit, we chose the threshold ISI as the valley of unit's bimodal ISI distribution, considering that periods of low and high ISIs (inter-spike-intervals) correspond to spikes occurring within and outside of bursts, respectively. After detecting spikes in the burst, we computed $\bar{t}_{isi}$. The exponent $\hat{\lambda}_i$ was then estimated for each unit i using a least squares fit of unit spike amplitudes during bursting periods to the model in eq. 2.1. A positive value of the estimate was indicative of amplitude attenuation within bursts; conversely, we considered units with null or negative $\hat{\lambda}_i$ as non-bursting units.
For each unit with detected bursts, we first computed the spatiotemporal noise covariance matrix $\mathbf{C}$ in the recording using snippets of traces, where no spikes were detected [16, 42]. For each unit, only 7 channels with the largest amplitude were used for template matching. Unit templates were extracted in the same channels (from the spike sorting output).

Next we applied modulated BOTM to each isolated spike and the last detected spike in each burst. We used a scaled version of each template using $\hat{\mu}_i[n]$ for the n^{th} spike within a burst of unit i, computed using the corresponding estimated exponent $\hat{\lambda}_i$. A time window of $2 \cdot \bar{t}_{isi}$ was used for template matching if spikes were in bursts. For isolated spikes, the maximal ($\bar{t}_{isi}$) over all bursts was considered and n was set to 2. Over this window we computed the BOTM discriminant function D_i for unit i as follows:

$$D_i(t) = \hat{\mu}_i[n] \left(X(t)^T \mathbf{C}^{-1} \xi_i - \frac{\hat{\mu}_i[n]}{2} \xi_i^T \mathbf{C}^{-1} \xi_i \right)$$

where $X(t)$ is a column vector of size $n_s n_c$ made of concatenated data snippets of length n_s from n_c channels, $\mathbf{C}$ the spatiotemporal noise covariance matrix of size $n_s n_c \times n_s n_c$, ξ_i the template of the i^{th} unit, also a column vector of size $n_s n_c$, and $p(i)$, the prior probability of observing a spike of unit i. The following values were used in our study: $n_s = 190$, $n_c = 7$, for all units. Spikes were recovered by detecting peaks in the discriminant function. Peaks are detected by crossing the threshold equal to five times of mean absolute deviation over signal baseline ($baseline + 5 \times MAD$). If a new spike was detected, we proceeded iteratively (incrementing n) to look for further spikes until a window with no spikes was reached or n reached n_{max}. We set n_{max} to 5 to limit the number of false positives. In [16], these discriminant functions were computed continuously for the entire recording, and spike detection and classification were performed simultaneously. Whenever a $D(t)$ crossed a threshold, a spike event was detected. This spike was immediately assigned to the neuron whose discriminant function produced the largest peak within a short time window of the detection. For performance reasons, we simplified the process here by applying the matched filters only onto the previously detected spike waveforms. The template matching was, therefore, only used for spike detection, not for spike classification, and we did not attempt to resolve overlapping spikes. Figure 2.4 illustrates the proposed modulated BOTM algorithm applied as a post-processing step to the spike sorting output.

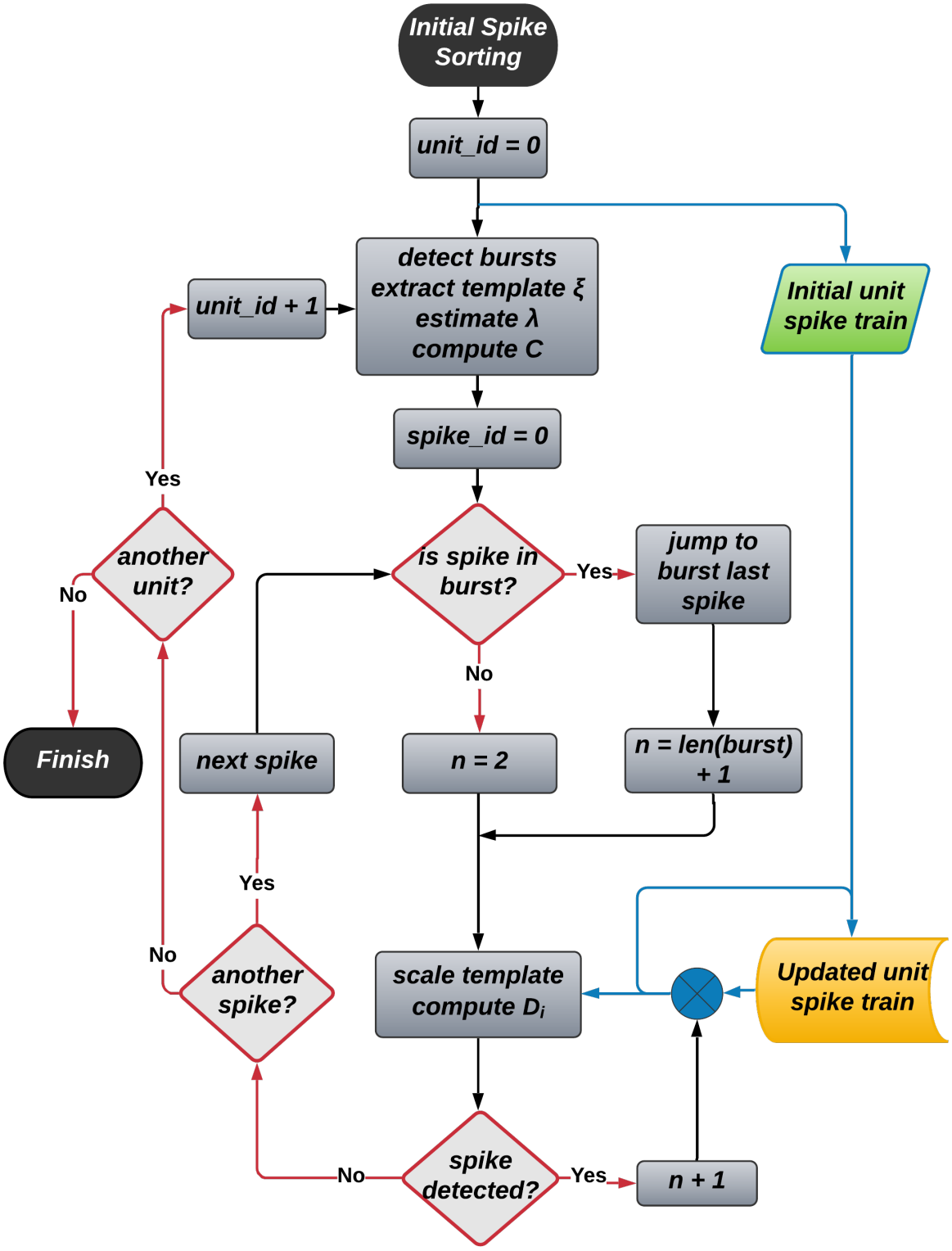

Figure 2.4: Block diagram of the proposed post-processing step to spike sorting; Modulated Bayes optimal template matching was applied in a small window beyond each isolated spikes and the trailing spike in each burst. If a new spike was detected, the initial unit spike train was correspondingly updated.

2.3.2. Merging Procedure for Burst-related Oversplits

Spike sorting often requires a manual curation step, in which the results are visually inspected by a human operator and spike clusters are either split or merged when necessary. For recordings with thousands of neurons this step becomes impractical and could be biased [12, 58]. A neuron can produce different spike waveforms within bursts of action potentials. This can cause a spike sorter to split the same neuron into multiple units. We addressed this problem by merging clusters based on the similarity of their templates and cross-correlograms. However, the quality of this merging depends on how strongly the waveforms change within a burst and, in turn, on how well the parameters in the merging step can be adapted to this change.

Most spike sorting algorithms may split one cell into several units due to strong bursting modulation. After running the entire algorithm, it is therefore necessary to merge together the units corresponding to the same cell. However, for hundreds or thousands of electrodes, going through all the pairs of units and merging them by hand would take a substantial amount of time. To overcome this problem, we designed a method to merge automatically split bursting units. Units that likely belong to the same bursting cell (and thus should be merged) have templates that look alike and in addition, the combined cross-correlogram between the two cell's spike trains shows a peak in a lag not equal to 0. However, while using this approach one would have to take into account that even distant neurons could be correlated. To automate the merging process entirely, we defined two simple rules to merge two units: first, their similarity should be above a threshold, since we expect high similarity in case of oversplit of bursting units because the modulation is mainly in amplitude. Similarity between two units template computed based on cosine distance as following:

$$\cos(\theta) = \frac{A.B}{\|A\| \, \|B\|} \tag{2.6}$$

Where A and B are templates of different units. Second, the peak value of cross-correlogram between the two units spike trains, must be greater than 0.95 quantile of cross-correlogram between number of generated surrogates of a spike trains by ISI shuffling. The surrogates are obtained by randomly sorting the ISIs of the corresponding units, which generates independent spike trains with same ISI distribution and spike count as in unit spike train, while destroying temporal dependencies and firing rate profile. Figure 2.5 shows two stages of similarity (panel A) and cross-correlograms (panel B) checks between splitted units.

We used 0.8 as a similarity threshold between templates and number of randomly generated surrogates for each unit is set to 20. We used `Elephant` (Electrophysiology Analysis

Toolkit) Python package [10] for generating spike train surrogates. Moreover, window and bins selected for computing cross-correlograms set as estimated maximum bursting duration and minimum average ISI between two units, respectively. Spike trains are merged (in the case of meeting all conditions) by merging close spikes ($thr = 0.5ms$) in to one, since some spikes could be assigned to two units.
An alternative to this approach could be applying a unimodality test [9], a manually curated merging step [60] or an adaptation of existing cluster comparison methods [25, 42, 47].

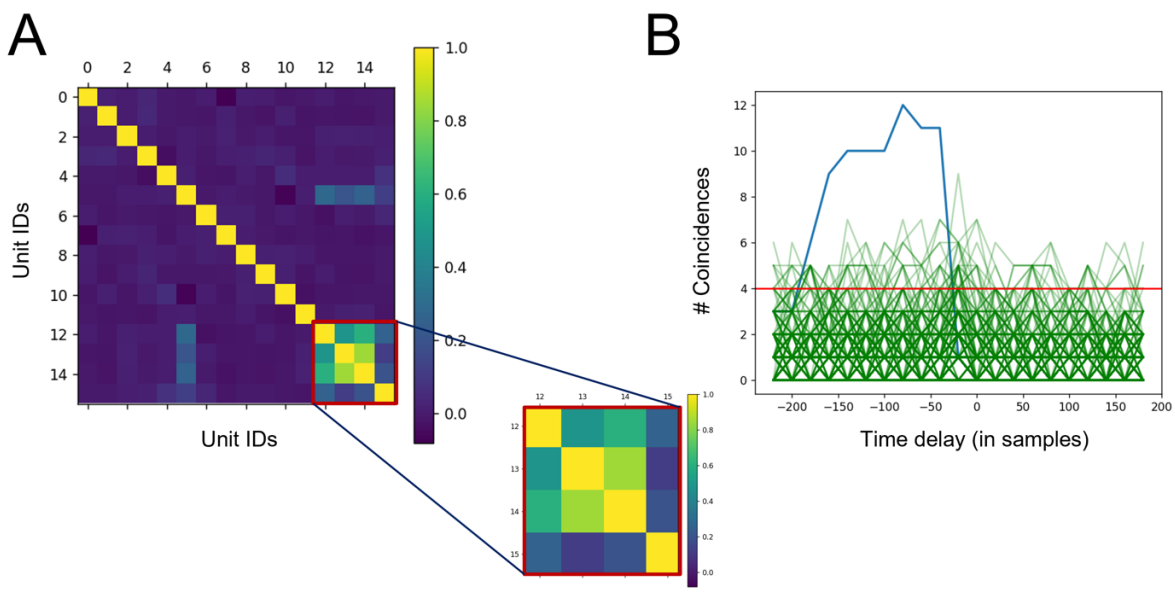

Figure 2.5: Two steps procedure for merging burst-related oversplits; **A.** Initial step is to check similarity via computing cosine distance between unit templates. Units with high similarity are selected. **B.** The cross-correlogram between spike trains of selected units are plotted for different values of time delay in blue. The cross-correlogram between 20 generated surrogates are colored in green. The threshold value is set to 0.95 quantile of cross-correlograms.

2.4. Experimental Recordings

The extracellular neuronal activity from primary dissociated rat hippocampal culture from many neurons was recorded for a duration of nearly 1000 seconds with the approximately 900 electrodes of HD-MEA, while, at the same time, the intracellular membrane potential of an individual patched neuron was measured using the whole-cell patch clamp technique. The synchronization of the intra and the extracellular signals was ensured by integrating the patch-clamp and the HD-MEA signals and recording them through the same data

acquisition system. Exploiting bimodal distribution of inter-spike-intervals, indicated that 98.6% of spikes were within bursts, which made it a suitable recording to benchmark our algorithm. Extracellular signal of an electrode during a burst event is shown in Figure 2.6. Notice that spike amplitudes showed a tendency to recover towards the end of the burst. Such a phenomenon would challenge our initially proposed post processing step. Since the spikes at the end of the bursts are more prone to be detected by spike sorter with respect to earlier spikes, which makes undetected spikes out of searching region in our proposed method. Therefore, here we propose an model-free approach based on modulated template matching (as previous one) to recover unsorted spikes.

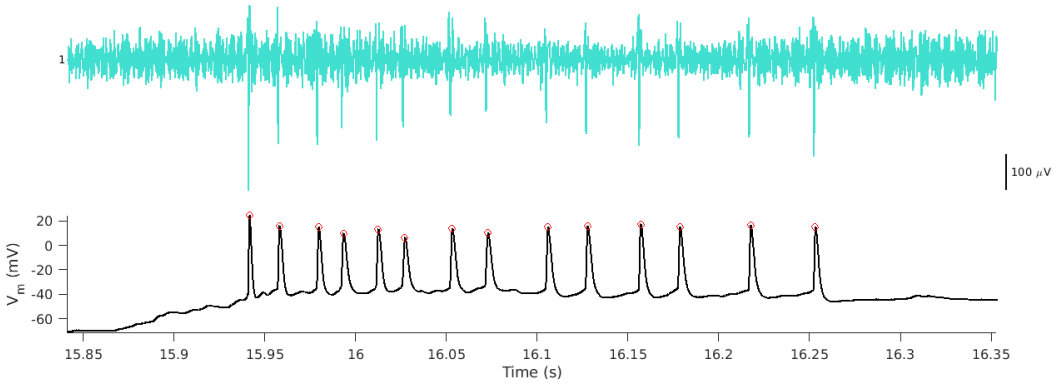

Figure 2.6: Upper panel shows extracellular signals recorded during a burst from an electrode in the vicinity of the soma. The lower panel shows the simultaneously acquired intracellular signal of patched neuron. Intracellular action potentials (APs) were detected by thresholding over 0 mv and are indicated by red circles. Extracellular spikes in bursts are initially attenuating in amplitude, but has a strong tendency to recover as the burst progresses.

2.5. Model-free Approach

In many bursts, complex amplitude adjustments are observed within the burst, particularly later on in the burst. Therefore, our proposed method is not able to capture these variations and will fail in recovering missed spikes. If we don't know the model for spike shape modulation or if the modulation is too complex, we can't use our initially model-based proposed method. In this case, using the model-free approaches would be better. In this section, we propose an model-free method to overcome these issues.

The algorithm is as following: first we select spikes from the initial spike sorting output that are not preceded by another spike within a short time window, this time window can be estimated as *argmax* of ISI histogram, then for each of the selected spikes (reference

spikes), modulated matched-filter template matching is performed in a small time window including the corresponding spike, then a unique threshold is generated as the fraction of the amplitude of the modulated template matching output at the matching point after signal baseline removal. Modulated template matching computes matched filter outputs for the given template as following:

$$D_i(t) = X(t)^T \mathbf{C}^{-1}(\mu\xi_i)$$

where $X(t)$ indicates a vector made by concatenating snippets of data from number of selected channels, $\mathbf{C}_i$ is the spatiotemporal noise covariance matrix computed on pieces of noise, that is, periods of traces where no spikes were detected. $\mathbf{C}_i$ is computed by random selecting of chunks of noise snippets on selected channels for each unit. μ indicates scaling factor used to make templates smaller in amplitude and ξ_i is the templates of the unit i on the selected channels. After setting a template matching output threshold for each selected spike, we apply modulated template matching in a large time window representing the estimated maximum burst duration, immediately after the corresponding spike and select the spikes whom cross the predefined threshold.

If a false positive spike from spike sorting output determined as a reference spike for threshold setting of template matching, it will result in lower template matching output and consequently lower threshold, which will result in finding new false positive spikes. Therefore, an extra optional step before modulated template matching is applied to reduce number of false positive spikes by thresholding the similarity between unit waveforms and corresponding template.

3 | Results

"Of course it is happening inside your head, Harry, but why on earth should that mean that it is not real?"

-Albus Dumbledore in Harry Potter and the Deathly Hallows

3.1. Modulated TM Recovers Missed Spikes

In order to check the performance of modulated TM, We simulated 5 recordings from different realizations of randomly generated bursting spike trains using `MEArec`. The decay λ associated for each neuron was randomly drawn from the interval $[0, 0.4]$, and each recording contained the activity of 5 neurons (25 neurons in total). Extracellular recordings were computed on a 100-channel microelectrode array in a 10×10 configuration with an electrode pitch of $15\,\mu m$ as explained in section 2.1. Then, spike sorting was performed on the recordings with four different spike sorters, namely "HerdingSpikes" [24], "Tridesclous" [18], "HDsort" [11] and "Klusta" [47]. All spike sorters and comparisons were run using the `SpikeInterface` framework [5]. We compared spike sorting outputs with ground-truth spiking activity from the `MEArec` simulated recordings (see section 2.2).

Post-processing by applying the proposed modulated TM method to the output of an initial spike sorting enabled the recovery of bursting spikes that were otherwise missed (false negatives). In Figure 3.1.A we show an example of a burst in 4 selected channels (signals of 4 electrodes). The waveforms in blue were correctly spike sorted by HDsort, while the green ones were only recovered after our post-processing step. The last spike (red), remained undetected since we had limited the recovery to $n_{max} = 5$ spikes. A comparison across many bursts is shown in Figure 3.1.B. Dots represent individual spikes within each burst. Spikes found by "HDsort" are shown in blue and the missed spikes in red. Spikes recovered after the modulated TM post-processing step are colored in green. The added TM step enabled the recovery of almost all initially missed spikes in the bursts.

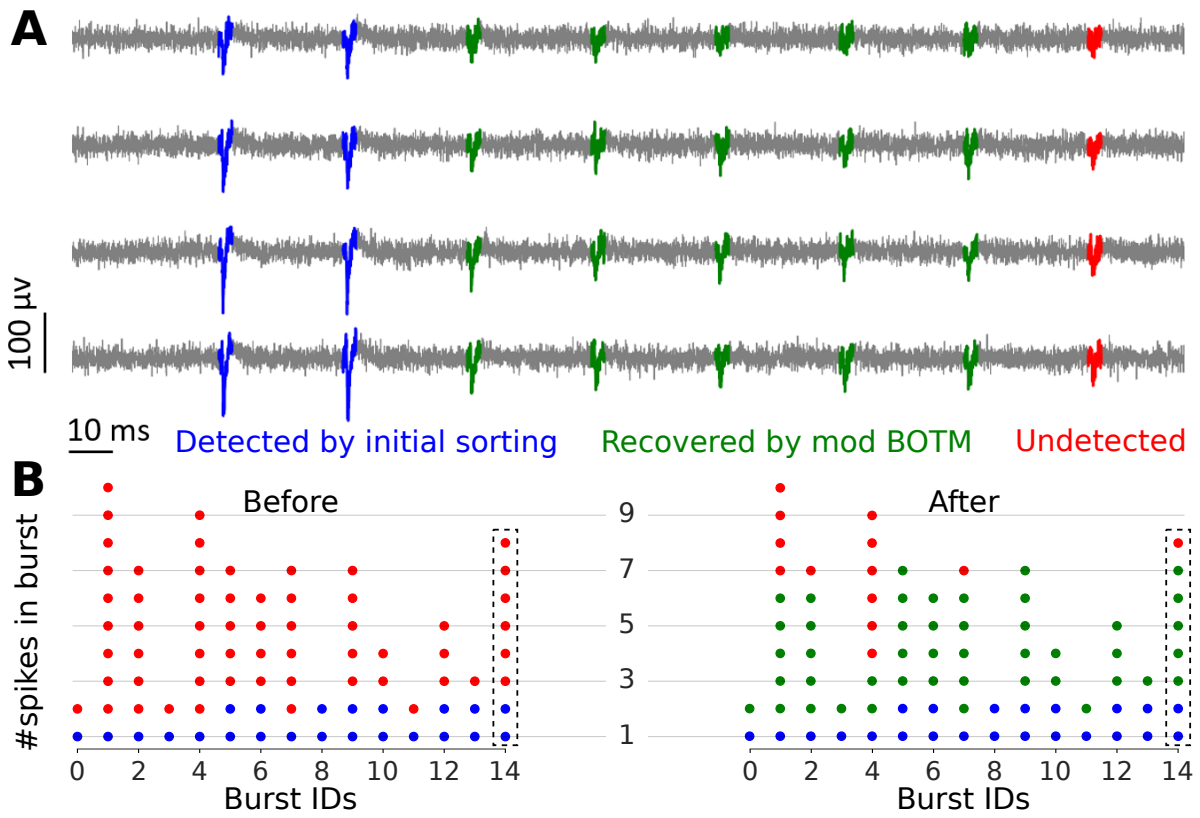

Figure 3.1: **A.** Data from a simulated dataset including 4 channels demonstrate the amplitude attenuation during bursting. Only two of the 8 spikes in this burst (Burst ID 14 in **B**) were detected during initial sorting ("HDsort", blue); the next 5 spikes were recovered after applying modulated TM to the "HDsort" output (green); the last spike in the burst remained undetected (red). **B.** Most of the spikes missed by the initial sorting were recovered during our post-processing step. Each column represents a burst and each dot a spike within it. Blue and red indicate that spikes were detected or not detected during the initial sorting (in this example, using "HDsort"). The spikes recovered during post-processing are indicated in green.

By looking at agreement-matrices of one simulated recording (*rec1*), spike sorted by *"Tridesclous"* before and after applying modulated TM, we observe that agreement score between ground-truth units and their corresponding best-match units from tested units increased due to spikes recovered in bursts after applying our post-processing step (see Figure 3.2). For instance, agreement scores for ground-truth units 0 and 1 are increased from 0.91 to 0.94 and from 0.74 to 0.87, respectively. However, for GT units 2 and 3, no elevation is observed, the possible reason is that, for these units only the first spikes in the bursts are detected by the spike sorter, therefore the algorithm classified these units as non-bursting units, which exempts algorithm to search for new spikes. Another feasible

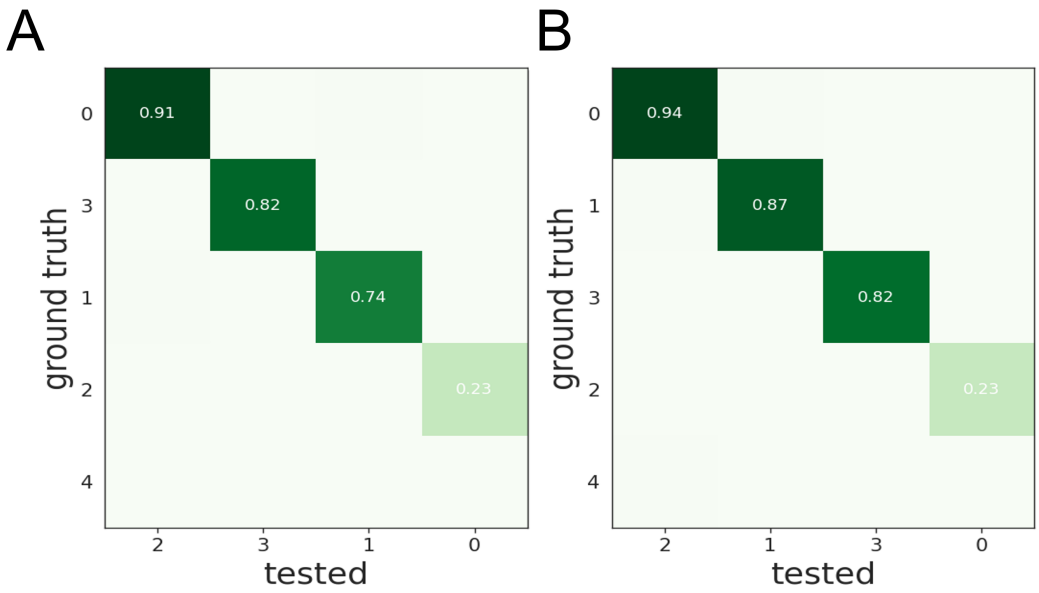

Figure 3.2: Agreement scores between best-match units from ground-truth and spike sorter are computed and shown in green boxes. A noticeable elevation in agreement score values after recovering spikes via modulated template matching (**B**) is observable with respect to values computed after initial spike sorting by *"Tridesclous"* (**A**). For ground-truth units 2 and 3, no new spikes (true positive or false positive) is detected, because initially sorted spikes in theses units are categorized as single spikes.

reason could be that, due to inaccurate estimation of λ, the discriminant signal didn't cross the detection threshold in spike presence windows.

We quantified the performance of our approach using the recall score (Figure 3.3). Units with a recall below 0.2 were not considered for the post-processing step. Across sorters, the vast majority of the units showed an increased recall after performing the additional TM step. We observed an overall increase in both accuracy and recall for all sorters (Figure 3.4.A and C). Since our algorithm looks for low-amplitude re-scaled spikes in bursts, there is the risk of finding false positive spikes. Such spurious spikes would lead to a decrease in the precision of the sorter. However, with the parameters we used, we did not observe an appreciable loss of precision except for a modest decrease for "HDsort" (Figure 3.4.B). In summary, the increase in recall in our data was much larger relative to the loss in precision, which steered the accuracy change towards positive values.

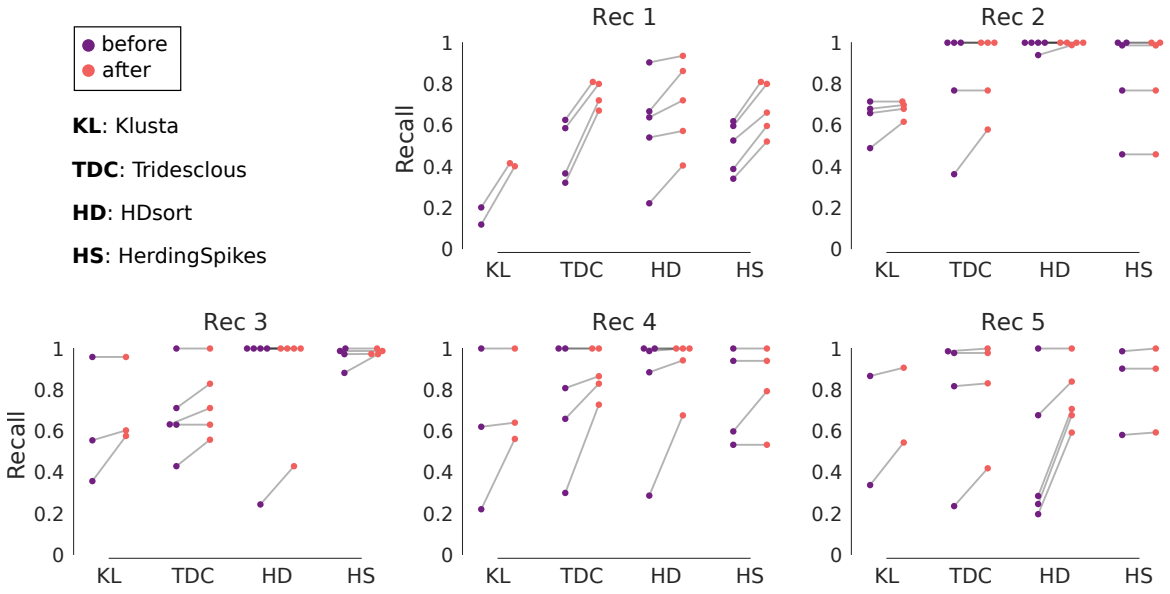

Figure 3.3: Recall (fraction of true spikes that were recovered), of unit assignments improved after post-processing using modulated TM. Recall scores are shown for 5 recordings with 4 different sorters, before and after post processing.

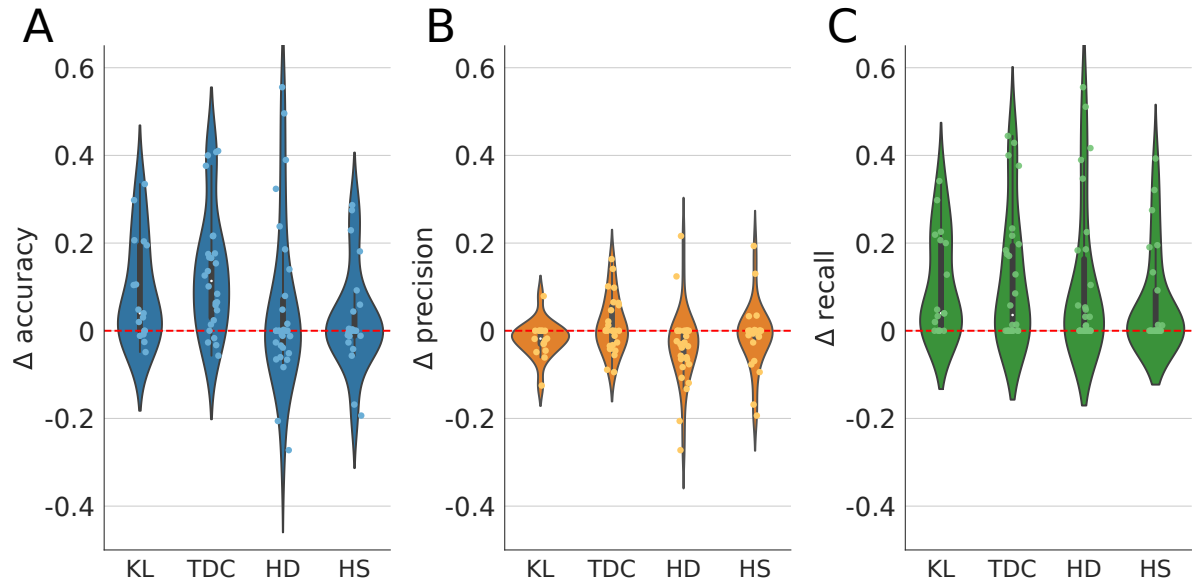

Figure 3.4: Distributions of changes in each sorter's performance measures before and after performing the modulated TM step. Units were pooled from all 5 recordings shown in Figure 3.3; overlaid dots correspond to individual units. **A.** Accuracy generally increased after post-processing, yet did not lead to an overall loss of spike sorting precision due to false positives (**B**). However, a modest decrease in precision was observed for a few units (especially for "HDsort"). **C.** Post-processing resulted in an appreciable overall improvement in recall scores.

In summary, we found that spike sorting accuracy and recall improved for most units after post processing with our approach. Moreover, we did not observe an appreciable loss of spike sorting precision as a result of the post-processing. Recall is the fraction of true spikes that were recovered. A consistent improvement of recall across all sorters suggests that our approach was successful in reducing the number of false negatives or missed spikes. However, the added sensitivity comes with the risk of having more false positives, which is reflected in the modest changes observed in precision and the slight drop in accuracy in a few units (especially for "HDsort"). No changes in accuracy, recall and precision were observed for many units. One possible reason is that if the detection and signal assignment to a unit was already highly accurate, the additional post-processing step was not expected to further improve the accuracy. In such cases, the accuracy remained the same or decreased slightly if a few false positives spikes were inadvertently detected. Alternatively, low SNR could yield units with low amplitudes, thereby limiting the ability of our approach to improve sorting performance.

3.2. Merging Potentially Split Units Improved Agreement Scores

To study the splitting of unit assignments during bursting, we simulated a recording with higher number of bursting units. Synthetic recordings, such as the recordings generated by `MEArec`, give us full access to ground-truth, which enable us to check bursting unit split, which is challenging to obtain by dual-recordings.

Using the `MEArec` package, we simulated an extracellular recording with 15 bursting units and a duration of 300 seconds. Random amplitude modulation factors (λ) were assigned to each unit from an interval between 0 and 0.4. Recording was then spike sorted using *Tridesclous*, then agreement score between tested and ground-truth units was computed. By checking agreement matrix between *Tridesclous* output and ground-truth units (see Figure 3.5.A), we found three *redundant* units; A *redundant* unit had a relatively high agreement score, but it was not a *best-match*, which means that it could either be an *oversplit* unit or a *duplicate* unit. After checking template similarities and the cross-correlograms, three pairs of units were merged (see section 2.3.2). Figure 3.5.B. shows agreement matrix after merging split units only for merged units; in the first case, units 0 and 13 from the spike sorter's output were merged and the agreement score between the *best-match* unit and unit 6 from ground-truth data increased from 0.58 to 0.97. In the second case, the agreement score between units 3 and 12 from ground-truth and sorter output respectively increased from 0.41 to 0.73 after merging units 1 and 12, and in the

third case, units 9 and 17 of spike sorter output merged in one unit and increased the *best-match* agreement score between unit 0 of ground-truth data and spike sorter from 0.45 to 0.8. Units 0 and 9 from tested sorter after merging turn to *well-detected* units. *Well-detected* unit is defined as a unit with agreement score higher than 0.8.

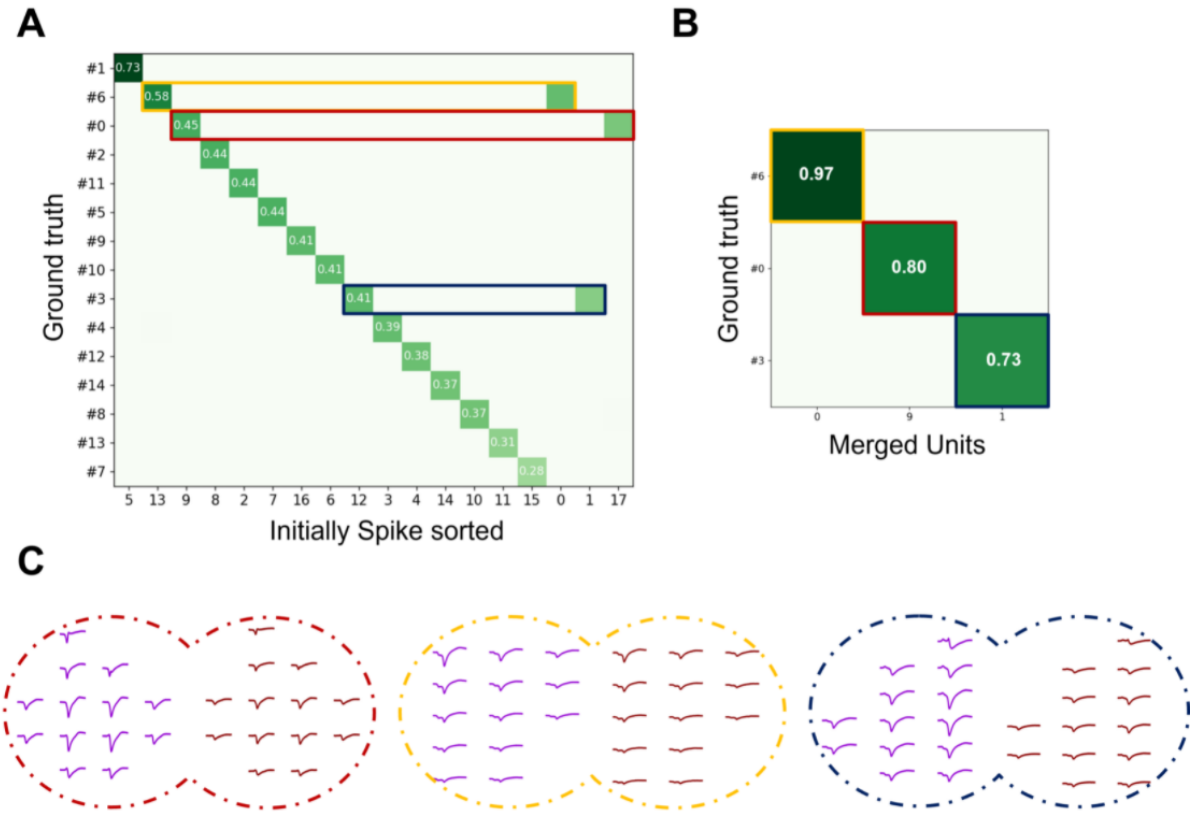

Figure 3.5: Agreement matrix between "Tridesclous" spike sorter output and ground-truth data from `MEArec`; **A**. Tested sorter split spikes in the bursts from units 0, 3 and 6 from ground-truth data in two separate units, which should be merged. **B**. After applying the algorithm for merging split units, units 1 and 12, 0 and 13, 9 and 17 are merged and agreement score (accuracy) between ground-truth and respective best-match units increased significantly and the overall performance of spike sorter is improved. **C**. Templates of merged assignment units in their 13 channels with largest templates are shown. Templates colored in violet represent earlier (larger) spikes sorted in the bursts, while templates colored in red are representative of later (smaller) spikes in the bursts. Split units have similar templates that are only modulated in amplitude.

3.3. Model-free Approach Recovered Undetected Spikes

We used a dual recordings (HD-MEA + Patch-clamp) and simulated recording to benchmark our algorithm (see 2.5). To test our algorithm, we performed spike sorting using different algorithms including "Kilosort2" [39], "HerdingSpikes" [24], "SpyKING Circus" [60] and "Tridesclous" [18] on HD-MEA recording, considering that for one cell (patch-clamped one) we know what should be the output of the spike sorting. Comparing spike trains of output units with intracellular spike train, within a small time tolerance of 1 ms and computing agreement score for each unit, we selected a unit with highest agreement score as *best-match* unit. We applied Modulated template matching on the *best-match* unit with the following parameters; 25 ms was set as a time window threshold for choosing selected spikes, time window to look for new spikes was set to 250 ms, 2/3 of the modulated template matching output of selected spike is selected as the threshold and scaling factor was set to 0.7. Figure 3.6 illustrates the application of the algorithm on an example burst with 14 spikes The waveforms colored in blue were correctly spike sorted by "HerdingSpikes" (including the yellow one), while the red ones remained undetected. The yellow waveform was chosen as the reference spike and modulated template matching was performed in a 250 ms time window immediately after the yellow waveform. Modulated template matching output is shown in seagreen in the bottom panel. 10 spikes including 7 undetected and 3 detected spikes were recovered by thresholding the template matching output.

To check the validity of our algorithm on the previously spike sorted simulated data, we applied our method on the recording with 15 bursting units (see section 3.2). After spike sorting, an automated procedure for merging split bursting units was performed, then followed by modulated template matching.. Figure 3.7 shows partition of 40 seconds of recording on 4 channels (panel B). Initial spikes in the bursts (colored in blue) were assigned to unit 0, and the smaller spikes (colored in cyan) are clustered in unit 13. These two units are merged after automatic merging procedure. Spikes colored in green are recovered after applying our alternate method. However, some spikes (colored in red) remained undetected or were assigned to wrong units.

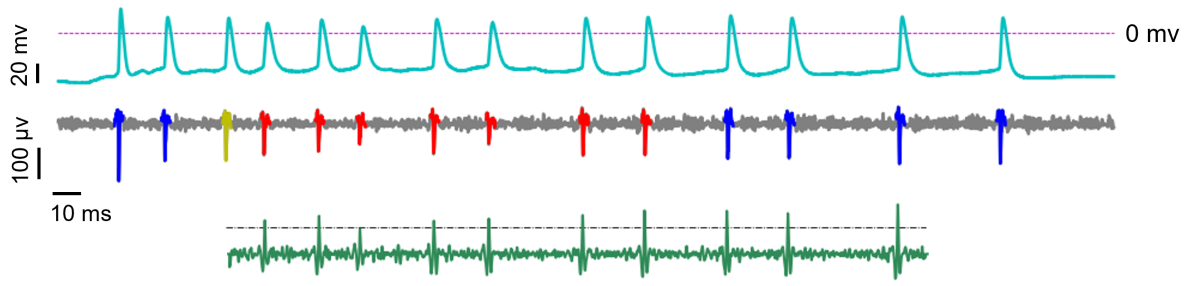

Figure 3.6: Detection of burst spikes by thresholding the output of modulated template matching on an example burst; Upper panel shows intracellular voltage trace zoomed in on one burst from a patched neuron of primary dissociated rat hippocampal culture; middle panel shows the burst recording of channel with largest template from HD-MEA recording. Blue spikes are truly assigned spikes (TPs) sorted by the spike sorter "Herdingspikes". Spikes from ground-truth data that were not detected by the spike sorter are colored in red and selected spikes from detected spikes for threshold setting colored by yellow. Bottom panel shows modulated template matching output applied after selected spike for a duration of 250 ms. Threshold is set as 2/3 of modulated template matching output of selected spike at the matching point.

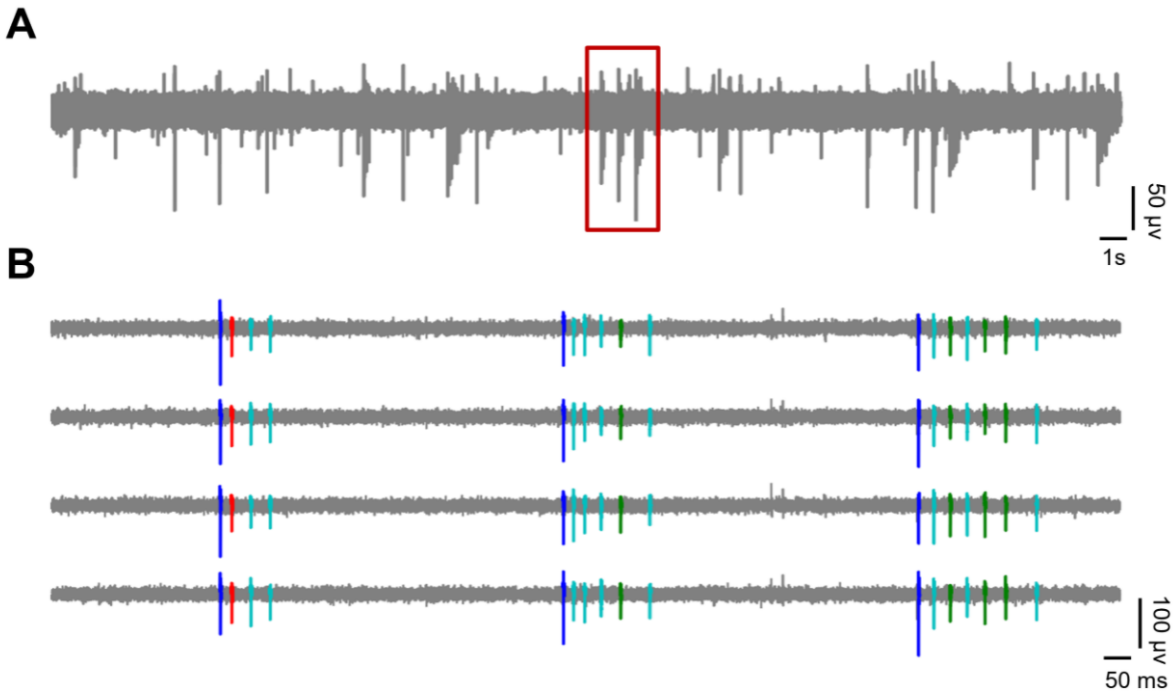

Figure 3.7: Partition of extracellular trace from simulated data containing 3 bursts is shown in 4 different channels. Detected spikes by initial spike sorting, spikes of merged units and recovered spikes by template matching are colored in blue, cyan and green, respectively. While, undetected spikes or falsely assigned spikes are colored in red.

The same performance metrics (accuracy, precision and recall) were computed for the *best-matched* unit. Figure 3.8 shows performance of our post-processing algorithm by comparing accuracy (panel A), precision (panel B) and recall (panel C), before and after applying it. It is important to mention that an extra post-processing step of thresholding similarity of spikes and template is performed for "Kilosort2", in which spikes with similarity (based on cosine distance) lower that $thr = 0.5$ with respect to unit template were discarded. This step is done to limit number of false positive spikes, since a false positive spike will result in finding new false positive spikes. The accuracy for all 4 spike sorters increased after post-processing, which means that number of recovered true positive spikes exceeds number of newly detected false positive spikes (if any). Precision value for "Kilosort2" increased dramatically after post-processing, which was mainly due to removing false positive spikes via similarity thresholding. For other sorters, there was no significant alteration after post-processing, because the initial precision values were close to the ideal value. Recall elevation after post-processing shows that number of true positive spikes increased, which tells us that our method were successful in detecting missed spikes.

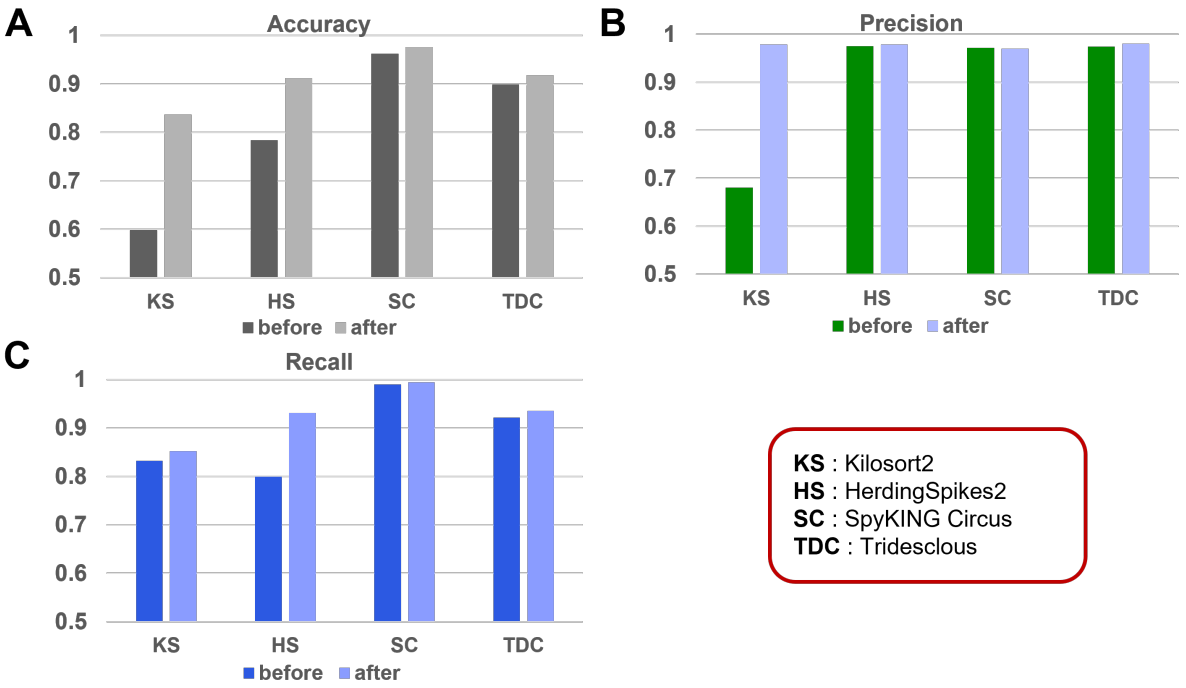

Figure 3.8: Performance metrics before and after applying modulated template matching step; An extra curation step after spike sorting was performed on "Kilosort2" to limit number of false positive spikes. **A**. Improvement in accuracy is indicative of increase in number of true positives in comparison with false positive spikes (if any). **B**. Except for "Kilosort2", the precision generally remained constant after post-processing. We did not observe an overall loss of spike sorting precision due to false positives. **C**. Post-processing resulted in an appreciable overall improvement in recall scores (fraction of true spikes that were recovered).

4 | Discussion

"We can pool information about experiences, but never the experiences themselves. From family to nation, every human group is a society of island universes."

-Aldous Huxley, The Doors of Perception

Here, we presented an approach to recover spikes that were missed by automatic spike sorters due to spike amplitude changes that accompany bursting. In our post-processing routine, we applied amplitude-modulated template matching locally, to recover potentially missed spikes. We tested the routine on simulated ground truth data. We found that spike sorting accuracy and recall improved for most units after post processing with our approach. Moreover, we did not observe an appreciable loss of spike sorting precision as a result of the post-processing. In the next phase, we introduced a novel approach which could be used in the more general bursting behaviours. We used combined HD-MEA and patch recordings to validate our proposed algorithm. By computing recall before and after applying our routine, we discovered that, our algorithm was successful in detecting new true positive spikes. However, accuracy and precision variation illustrated that, there were very few recovered false positive spikes. Furthermore, we have automated the process of merging spike trains. Hence, the post processing routines for bursting periods may be fully automated. We tested this method using simulated ground truth data. In these recordings, there is full access to each neuronal spike train. We could clearly merge all the split pairs.

4.1. Future Developments

The variability of spike shapes in addition to amplitude in bursting cells also poses technical challenge to the study of bursting from extracellular data, because spikes may be misclassified as belonging to a different cell. Therefore, one should consider more complicated features extracted from both amplitude and shape of the spikes to better study the bursting from extracellular recordings.

One would extend the approach to rescue spikes of units that are synchronously bursting,

where the spatiotemporal ovevrlap of spike waverforms may pose challenges for our algorithm [15]. In this thesis, we constrained the `MEArec` simulator so as to limit the presence of spatiotemporal collisions in the simulated data.
In the simulated data, the spike amplitude attenuation in bursts were similar for all channels, therefore we scaled the templates equally for the selected channels. However, in general this is not true, and the attenuation profile is spatially asymmetric. Attenuation profile could be modeled as a function of distance from *Soma* or the mean amplitudes of the first spikes in bursts (see Figure 4.1).

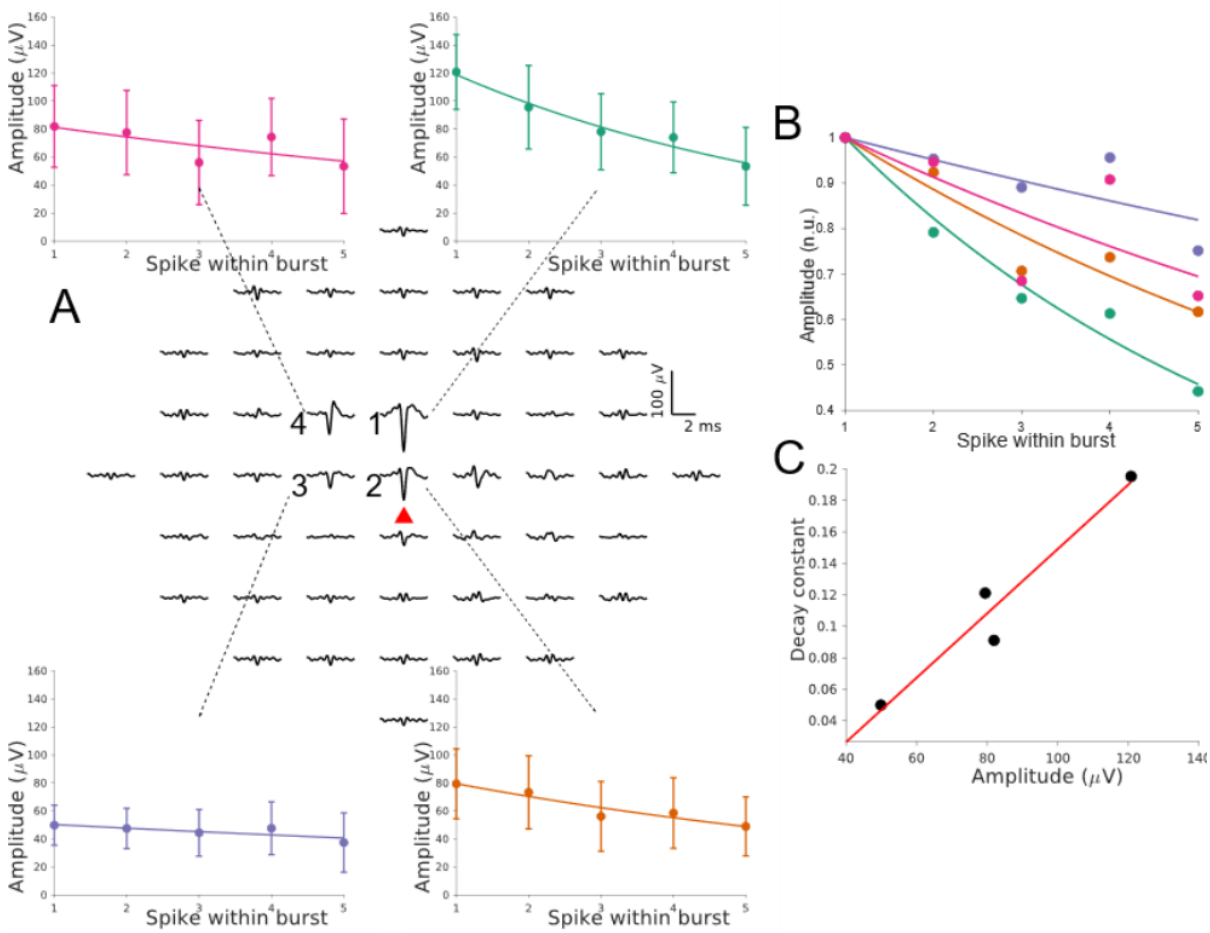

Figure 4.1: Amplitude attenuation is spatially asymmetric; **A**. Extracellular footprint obtained by averaging signal snippets around each intracellularly detected spike time of an individual neuron; spikes are ranked within bursts then the amplitude decay for channels around *soma* were plotted. **B**. Decay constants differ from very low values (almost constant) to high values for different channels, this value could be modeled as a function of probe distance from *soma*. **C**. Relationship between the decay constant and mean amplitudes of the first spikes in bursts outline that as the spike amplitude is larger (which mirrors electrode to *soma* distance), absolute value of attenuation is higher.

Despite computational approaches, one could use modeling criteria to better understand spike shape transformation during bursting regimes, and use it to mimic spike amplitude variation via modulated template matching. Beyond spike rank within the burst, other factors may also be important determinants of amplitude attenuation, especially as the burst progresses. The high frequency discharges within a burst can be thought of as a history dependent renewal process. Amplitude likely reflects the fraction of active voltage gated sodium channels available. This is likely dependent on the time to the previous spike discharge. Intuitively, thus, the inter-spike-interval could be a relevant co variate that could contribute to explain the observed variability in bursts. As an interesting aside, one could consider spike amplitude within the burst as a function of rank and the ISI (inter-spike-interval), which could be decomposed into two functions of single variables, assuming negligible interaction between these terms. Equation 4.1 could be a potential example of combining of two exponential models $g(.)$ (equation 4.2) and $h(.)$ (equation 4.3) representing rank and ISI dependency, respectively.

$$f(n, t_{isi}) = g(n) + h(t_{isi}) \tag{4.1}$$

$$g(n) = a_1 e^{b_1 n} + c_1 \tag{4.2}$$

$$h(t) = a_2(1 - e^{b_2 t}) + c_2 \tag{4.3}$$

4.2. Outlook

In conclusion, in this thesis, I presented and validated a novel methods to improve the quality of spike sorting when spike features are non-stationary and history dependent. The algorithm can be readily applied to the output of any generic automated spike-sorting procedure. Further studies are necessary to explore the optimal trade-off between precision and recall in our algorithms and importantly, better understanding of how extracellular spike features change during bursting will contribute to advance and refine our approach.

Bibliography

[1] B. Amini, J. Clark Jr, and C. Canavier. Calcium dynamics underlying pacemaker-like and burst firing oscillations in midbrain dopaminergic neurons: a computational study. *Journal of Neurophysiology*, 82(5):2249–2261, 1999.

[2] D. J. Bakkum, M. Radivojevic, U. Frey, F. Franke, A. Hierlemann, and H. Takahashi. Parameters for burst detection. *Frontiers in computational neuroscience*, 7:193, 2014.

[3] A. Buccino, S. Garcia, and P. Yger. Spike sorting: new trends and challenges of the era of high-density probes. 2022.

[4] A. P. Buccino and G. T. Einevoll. Mearec: a fast and customizable testbench simulator for ground-truth extracellular spiking activity. *Neuroinformatics*, 19(1):185–204, 2021.

[5] A. P. Buccino, C. L. Hurwitz, S. Garcia, et al. Spikeinterface, a unified framework for spike sorting. *Elife*, 9:e61834, 2020.

[6] G. Buzsáki. Large-scale recording of neuronal ensembles. *Nature neuroscience*, 7(5): 446–451, 2004.

[7] L. A. Camuñas-Mesa and R. Q. Quiroga. A detailed and fast model of extracellular recordings. *Neural computation*, 25(5):1191–1212, 2013.

[8] D. R. Chialvo. Emergent complex neural dynamics. *Nature physics*, 6(10):744–750, 2010.

[9] J. E. Chung, J. F. Magland, A. H. Barnett, V. M. Tolosa, A. C. Tooker, K. Y. Lee, K. G. Shah, S. H. Felix, L. M. Frank, and L. F. Greengard. A fully automated approach to spike sorting. *Neuron*, 95(6):1381–1394, 2017.

[10] M. Denker, A. Yegenoglu, and S. Grün. Collaborative HPC-enabled workflows on the HBP Collaboratory using the Elephant framework. In *Neuroinformatics 2018*, page P19, 2018. doi: 10.12751/incf.ni2018.0019. URL `https://abstracts.g-node.org/conference/NI2018/abstracts#/uuid/023bec4e-0c35-4563-81ce-2c6fac282abd`.

[11] R. Diggelmann, M. Fiscella, A. Hierlemann, and F. Franke. Automatic spike sorting for high-density microelectrode arrays. *Journal of neurophysiology*, 120(6):3155–3171, 2018.

[12] G. T. Einevoll, F. Franke, E. Hagen, C. Pouzat, and K. D. Harris. Towards reliable spike-train recordings from thousands of neurons with multielectrodes. *Current opinion in neurobiology*, 22(1):11–17, 2012.

[13] M. S. Fee, P. P. Mitra, and D. Kleinfeld. Variability of extracellular spike waveforms of cortical neurons. *Journal of neurophysiology*, 76(6):3823–3833, 1996.

[14] M. S. Fee, A. Kozhevnikov, R. Hahnloser, et al. Neural mechanisms of vocal sequence generation in the songbird. *Ann NY Acad Sci*, 1016(1), 2004.

[15] F. Franke, R. Pröpper, H. Alle, P. Meier, J. R. Geiger, K. Obermayer, and M. H. Munk. Spike sorting of synchronous spikes from local neuron ensembles. *Journal of neurophysiology*, 114(4):2535–2549, 2015.

[16] F. Franke, R. Q. Quiroga, A. Hierlemann, and K. Obermayer. Bayes optimal template matching for spike sorting–combining fisher discriminant analysis with optimal filtering. *Journal of computational neuroscience*, 38(3):439–459, 2015.

[17] F. Gabbiani, W. Metzner, R. Wessel, and C. Koch. From stimulus encoding to feature extraction in weakly electric fish. *Nature*, 384(6609):564–567, 1996.

[18] Garcia and Pouzat. tridesclous. `https://github.com/tridesclous/tridesclous`, 2015.

[19] Y. Ghanbari, L. Spence, and P. Papamichalis. A graph-laplacian-based feature extraction algorithm for neural spike sorting. In *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 3142–3145. IEEE, 2009.

[20] E. Hagen, T. V. Ness, A. Khosrowshahi, C. Sørensen, M. Fyhn, T. Hafting, F. Franke, and G. T. Einevoll. Visapy: a python tool for biophysics-based generation of virtual spiking activity for evaluation of spike-sorting algorithms. *Journal of neuroscience methods*, 245:182–204, 2015.

[21] K. D. Harris, D. A. Henze, J. Csicsvari, H. Hirase, and G. Buzsaki. Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. *Journal of neurophysiology*, 84(1):401–414, 2000.

[22] R. M. Harris-Warrick and R. E. Flamm. Multiple mechanisms of bursting in a conditional bursting neuron. *Journal of Neuroscience*, 7(7):2113–2128, 1987.

[23] D. A. Henze, Z. Borhegyi, J. Csicsvari, A. Mamiya, K. D. Harris, and G. Buzsaki. Intracellular features predicted by extracellular recordings in the hippocampus in vivo. *Journal of neurophysiology*, 84(1):390–400, 2000.

[24] G. Hilgen, M. Sorbaro, S. Pirmoradian, et al. Unsupervised spike sorting for large-scale, high-density multielectrode arrays. *Cell reports*, 18(10):2521–2532, 2017.

[25] D. N. Hill, S. B. Mehta, and D. Kleinfeld. Quality metrics to accompany spike sorting of extracellular signals. *Journal of Neuroscience*, 31(24):8699–8705, 2011.

[26] J. R. Huguenard and D. A. McCormick. Simulation of the currents involved in rhythmic oscillations in thalamic relay neurons. *Journal of neurophysiology*, 68(4): 1373–1383, 1992.

[27] E. Hulata, R. Segev, and E. Ben-Jacob. A method for spike sorting and detection based on wavelet packets and shannon's mutual information. *Journal of neuroscience methods*, 117(1):1–12, 2002.

[28] D. L. Hunt, C. Lai, R. D. Smith, A. K. Lee, T. D. Harris, and M. Barbic. Multimodal in vivo brain electrophysiology with integrated glass microelectrodes. *Nature biomedical engineering*, 3(9):741–753, 2019.

[29] E. M. Izhikevich. Neural excitability, spiking and bursting. *International journal of bifurcation and chaos*, 10(06):1171–1266, 2000.

[30] E. M. Izhikevich, N. S. Desai, E. C. Walcott, and F. C. Hoppensteadt. Bursts as a unit of neural information: selective communication via resonance. *Trends in neurosciences*, 26(3):161–167, 2003.

[31] J. Jefferys and H. Haas. Synchronized bursting of ca1 hippocampal pyramidal cells in the absence of synaptic transmission. *Nature*, 300(5891):448–450, 1982.

[32] Y. Kaneoke and J. Vitek. Burst and oscillation as disparate neuronal properties. *Journal of neuroscience methods*, 68(2):211–223, 1996.

[33] J. E. Lisman. Bursts as a unit of neural information: making unreliable synapses reliable. *Trends in neurosciences*, 20(1):38–43, 1997.

[34] D. A. McCormick and D. Contreras. On the cellular and network bases of epileptic seizures. *Annual review of physiology*, 63(1):815–846, 2001.

[35] A. Molleman. *Patch clamping: an introductory guide to patch clamp electrophysiology*. John Wiley & Sons, 2003.

[36] E. Neher and B. Sakmann. Single-channel currents recorded from membrane of denervated frog muscle fibres. *Nature*, 260(5554):799–802, 1976.

[37] J. P. Neto, G. Lopes, J. Frazão, J. Nogueira, P. Lacerda, P. Baião, A. Aarts, A. Andrei, S. Musa, E. Fortunato, et al. Validating silicon polytrodes with paired juxtacellular recordings: method and dataset. *Journal of neurophysiology*, 116(2):892–903, 2016.

[38] A.-M. M. Oswald, M. J. Chacron, B. Doiron, J. Bastian, and L. Maler. Parallel processing of sensory input by bursts and isolated spikes. *Journal of Neuroscience*, 24(18):4351–4362, 2004.

[39] M. Pachitariu, N. A. Steinmetz, S. N. Kadir, M. Carandini, and K. D. Harris. Fast and accurate spike sorting of high-channel count probes with kilosort. *Advances in neural information processing systems*, 29:4448–4456, 2016.

[40] V. Pasquale, S. Martinoia, and M. Chiappalone. A self-adapting approach for the detection of bursts and network bursts in neuronal cultures. *Journal of computational neuroscience*, 29(1):213–229, 2010.

[41] T. Petermann, T. C. Thiagarajan, M. A. Lebedev, M. A. Nicolelis, D. R. Chialvo, and D. Plenz. Spontaneous cortical activity in awake monkeys composed of neuronal avalanches. *Proceedings of the National Academy of Sciences*, 106(37):15921–15926, 2009.

[42] C. Pouzat, O. Mazor, and G. Laurent. Using noise signature to optimize spike-sorting and to assess neuronal classification quality. *Journal of neuroscience methods*, 122 (1):43–57, 2002.

[43] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul. Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. *Neural computation*, 16(8): 1661–1687, 2004.

[44] P. Reinagel, D. Godwin, S. M. Sherman, and C. Koch. Encoding of visual information by lgn bursts. *Journal of neurophysiology*, 81(5):2558–2569, 1999.

[45] H. G. Rey, C. Pedreira, and R. Q. Quiroga. Past, present and future of spike sorting techniques. *Brain research bulletin*, 119:106–117, 2015.

[46] W. M. Roberts and D. K. Hartline. Separation of multi-unit nerve impulse trains by a multi-channel linear filter algorithm. *Brain research*, 94(1):141–149, 1975.

[47] C. Rossant, S. N. Kadir, D. F. Goodman, et al. Spike sorting for large, dense electrode arrays. *Nature neuroscience*, 19(4):634–641, 2016.

[48] P. S. Shabestari, A. P. Buccino, S. S. Kumar, A. Pedrocchi, and A. Hierlemann. A modulated template-matching approach to improve spike sorting of bursting neurons. In *2021 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–4. IEEE, 2021.

[49] S. Tajima, T. Mita, D. J. Bakkum, H. Takahashi, and T. Toyoizumi. Locally embedded presages of global network bursts. *Proceedings of the National Academy of Sciences*, 114(36):9517–9522, 2017.

[50] S. Takahashi, Y. Sakurai, M. Tsukada, and Y. Anzai. Classification of neuronal activities from tetrode recordings using independent component analysis. *Neurocomputing*, 49(1-4):289–298, 2002.

[51] S. Takahashi, Y. Anzai, and Y. Sakurai. A new approach to spike sorting for multi-neuronal activities recorded with a tetrode—how ica can be practical. *Neuroscience research*, 46(3):265–272, 2003.

[52] T. Takekawa, Y. Isomura, and T. Fukai. Accurate spike sorting for multi-unit recordings. *European Journal of Neuroscience*, 31(2):263–272, 2010.

[53] J. Voigts, J. H. Siegle, D. L. Pritchett, and C. I. Moore. The flexdrive: an ultra-light implant for optical control and highly parallel chronic recording of neuronal ensembles in freely moving mice. *Frontiers in systems neuroscience*, 7:8, 2013.

[54] D. A. Wagenaar, J. Pine, and S. M. Potter. An extremely rich repertoire of bursting patterns during the development of cortical cultures. *BMC neuroscience*, 7(1):1–18, 2006.

[55] X.-J. Wang. Fast burst firing and short-term synaptic plasticity: a model of neocortical chattering neurons. *Neuroscience*, 89(2):347–362, 1999.

[56] M. Wehr, J. S. Pezaris, and M. Sahani. Simultaneous paired intracellular and tetrode recordings for evaluating the performance of spike sorting algorithms. *Neurocomputing*, 26:1061–1068, 1999.

[57] M. A. Wilson and B. L. McNaughton. Dynamics of the hippocampal ensemble code for space. *Science*, 261(5124):1055–1058, 1993.

[58] F. Wood, M. J. Black, C. Vargas-Irwin, M. Fellows, and J. P. Donoghue. On the variability of manual spike sorting. *IEEE Transactions on Biomedical Engineering*, 51(6):912–918, 2004.

[59] J. Wouters, F. Kloosterman, and A. Bertrand. Shybrid: A graphical tool for generating hybrid ground-truth spiking data for evaluating spike sorting performance. *Neuroinformatics*, 19(1):141–158, 2021.

[60] P. Yger, G. L. Spampinato, E. Esposito, B. Lefebvre, S. Deny, C. Gardella, M. Stimberg, F. Jetter, G. Zeck, S. Picaud, et al. A spike sorting toolbox for up to thousands of electrodes validated with ground truth recordings in vitro and in vivo. *Elife*, 7: e34518, 2018.

[61] X. Yuan, A. Hierlemann, and U. Frey. Extracellular recording of entire neural networks using a dual-mode microelectrode array with 19,584 electrodes and high snr. *IEEE Journal of Solid-State Circuits*, 2021.

[62] F. Zeldenrust, W. J. Wadman, and B. Englitz. Neural coding with bursts—current state and future perspectives. *Frontiers in computational neuroscience*, 12:48, 2018.

5 | Appendices

A | Executive Summery

Improving Spike Sorting Performance of Bursting Neurons

POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Improving Spike Sorting Performance of Bursting Neurons

Laurea Magistrale in Biomedical Engineering - Ingegneria Biomedica

Author: Payam Sadeghi Shabestari

Advisor: Dr. Alessio Buccino, Dr. Sreedhar Kumar

Co-advisors: Dr. Manuel Schröter, Prof. Alessandra Pedrocchi, Prof. Andreas Hierlemann

Academic year: 2021-2022

1. Introduction

Bursting activity is present in many cells of different nervous systems playing important roles in neural information processing [12]. A burst consists of a short, high frequency train of spikes, which could reliably cross a synapse, increasing the likelihood of eliciting a postsynaptic spike [4]. Therefore, detecting bursting activity is crucial for many neuroscientific studies.
Extracellular recording is an electrophysiology technique that uses an electrode inserted into living tissue to measure electrical activity coming from adjacent cells, usually neurons. Assigning of each extracellularly recorded putative spike to it's cell of origin is done through an unsupervised technique called spike sorting. However, an important challenge for spike sorting algorithms are nonstationarity of spikes waveform over time, since spike sorting algorithms assume that spikes of a very same neuron should remain almost constant over time. A prominent example of spike waveform transformation over time is bursting. Many experimental and computational studies [10] using various techniques including patch-clamp, combined extracellular-juxtacellular recordings [1] demonstrate that, during bursting, extracellular spikes become lower in amplitude. This transformation may lead in incorrect assigning of a spike (false positive) or not detecting a spike (false negative) due to it's small amplitude.
To overcome these two problems we proposed two different fully automatic approaches, which could be applied to any spike sorter output. The first method is based on *template-matching* which can recover spikes belonging to a burst from the output of existing spike sorting algorithms. Initially, we compute *templates*, i.e., the average extracellular waveforms, for each unit, then we apply template-matching in a small time window after detected spikes in bursts, considering that undetected spikes are smaller in amplitude. Therefore, we modulated the template amplitude in dependence on the spiking history. We introduced a novel modulated template matching based on Bayesian optimal template matching (BOTM) technique proposed by Franke et al. [8]. We proposed a second approach for merging burst-related oversplit units, which includes two steps of similarity check between unit templates and checking temporal dependencies between spike trains of split units.
To assess and validate our proposed methods, we used recordings with full access to ground-truth information. We used the `MEArec` Python

package [3], which is a customizable simulator for generating realistic extracellular recordings by using various cell models. `MEArec` enables us to simulate variety of bursting recordings with different modulation values. Having access to full ground-truth in terms of unit's spiking activity enable us to benchmark our algorithms.
We also proposed an alternative model-free post-processing approach to recover spikes undetected by initial spike sorting. This method could be used in the case of bursting with complex amplitude adjustment or when a priori information about burst model is not known. To validate the proposed model we used combined HD-MEA and patch recordings, in which intracellular spike train of patched cell serves as ground-truth.
The article is organized as follows: In section 2, first we describe the method we used for simulating bursting recordings using `MEArec` simulator (section 2.1). Then we introduce the algorithms for recovering missed and falsely assigned spikes in section 2.2 and 2.3, respectively. In section 2.4 we introduce a modified approach for more advanced bursting activities using experimental recordings. Next in section 2.5 we describe the approach for evaluating the spike sorting outputs. Results of how our algorithms can improve overall spike sorting performance are presented in section 3. Finally, a discussion over results, limitations, future directions and conclusion remarks are indicated in section 4.

2. Methods

2.1. *Simulated Bursting Recordings*

In order to generate different bursting recordings using `MEArec`, first we generated random bursting spike trains. These spike trains include random ISIs (inter-spike-intervals), IBIs (inter-burst-intervals) and random number of spikes in the bursts. `MEArec` exploits different biophysically detailed cell models to generate extracellular templates and then convolves the generated templates with user defined bursting spike trains. An amplitude modulation is applied to spikes within bursts, based on their rank in the burst, inter-spike-interval and burst duration. The equation 1 represents the bursting spike amplitude modulation in `MEArec` simulator.

$$\mu[n] = \left(\frac{\bar{t}_{isi}}{(n-1) \cdot t_{max}} \right)^{\lambda} + N(0, \sigma^2) \qquad (1)$$

where, $\mu[n]$ is the value which scales the amplitude of n^{th} spike within the burst. Note that the values of n are equal or greater than 2 ($n \geq 2$). λ is the value defined by user to regulate the amount of amplitude attenuation. As the value of λ is larger, the bursts are more modulated in terms of the spike's amplitude. $\bar{t}_{isi}$ is indicator of average inter-spike interval computed till the n^{th} spike within a burst, t_{max} is used as a normalization factor and estimated as maximal burst duration. An extra uncorrelated additive Gaussian noise with σ^2 variance is applied on the recordings to better mimic the realistic extracellular recordings.
We simulated 5 bursting recordings, by using generated bursting spike trains and random values for λ. Regularization factor λ is selected randomly from the interval $[0, 0.4]$ for each unit. Each simulated recording is extracellular activity of 5 neuron cells recorded by a 10×10 square microelectrode array (100 channel in total) with $15\,\mu$m inter electrode distance used for probing. Moreover we forced `MEArec` by setting $sync - rate = 0$ to limit spatiotemporal overlapping of spikes to better investigate the effect of bursting modulation in spike sorting performance. The flowchart of post-processing steps applied on simulated recordings are depict in Figure 1.

2.2. *Modulated BOTM*

In order to recover spikes within bursts that are missed during initial spike sorting step due to their small amplitude, we proposed a template-matching based approach for spike detection as following [15]: first we detect bursts from spike sorting output. By checking bi-modal distribution of ISIs (inter-spike-intervals) and selecting a critical ISI value in terms of local minimum, we could assign spikes with ISIs lower than critical value as bursting spikes. After detecting bursts, we estimated exponent $\hat{\lambda}_i$ for each sorted unit i by fitting equation 1 by means of minimizing sum of squared errors. If a positive value is estimated for $\hat{\lambda}_i$ we keep the unit as bursting, otherwise the unit is discarded from post-processing step. After estimating amplitude attenuation values and selecting bursting units, we

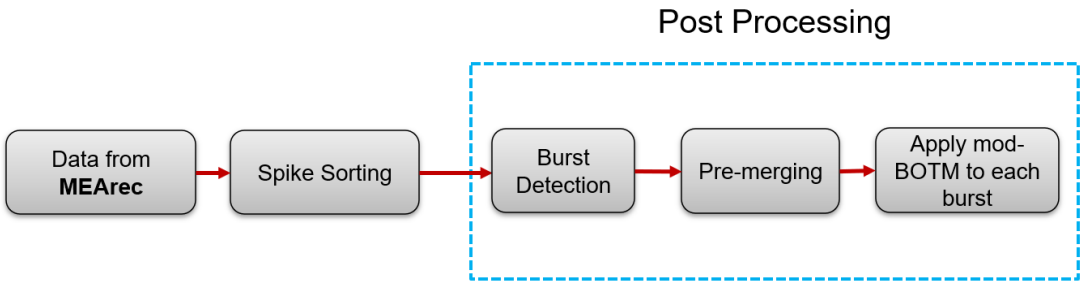

Figure 1: Workflow of applying post-processing steps after initial spike sorting for recovering false positive (pre-merging) and false negative spikes (modulated BOTM).

computed templates (spatiotemporal average of spike waveforms) for each unit in order to perform the modulated BOTM. Template matching is the process of moving the template over the entire signal and calculating the similarity between the template and the covered window on the signal. We applied modulated BOTM as following: we estimated spatiotemporal noise covariance matrix $\mathbf{C}$ for each unit, by randomly selecting extracellular traces in 7 channels with largest templates, where no spike was detected by spike sorter. Next, we applied modulated BOTM immediately after any single spikes and last detected spikes in the bursts, considering that for each unit, the template is scaled with respect to estimated value of $\hat{\mu}_i[n]$. Only 7 channels with maximum template size are selected, and templates are computed on only these channels. Modulated BOTM is performed on a small time window equal to $2 \cdot \bar{t}_{isi}$ immediately after the last detected spikes in the bursts. If a new spike is detected we incremented the n and performed the template matching iteratively. The process continued till a new spike is not detected. For isolated spikes, the window for performing the modulated BOTM is selected as the maximal ($\bar{t}_{isi}$) over all bursts and n was set to 2. Modulated BOTM output is computed trough following equation:

$$D_i(t) = \hat{\mu}_i[n] \left(X(t)^T \mathbf{C}^{-1} \xi_i - \frac{\hat{\mu}_i[n]}{2} \xi_i^T \mathbf{C}^{-1} \xi_i \right)$$

where $X(t)$ is a matrix with the size equal to length of the snippets n_s by number of selected channels n_c ($n_s \times n_c$). The template of the unit i is indicated by ξ_i and spatiotemporal noise covariance matrix is represented as $\mathbf{C}$ with the size of $n_s n_c \times n_s n_c$. The estimated factor for modulating template amplitude is represented as $\hat{\mu}_i[n]$. The following values were used for all units in our study: $n_s = 190$, $n_c = 7$. Spikes are detected by means of thresholding the template matching output. Threshold is set equal to five times of mean absolute deviation over signal baseline ($baseline + 5 \cdot MAD$). The flowchart of proposed modulated BOTM for recovering undetected spikes are shown in Figure 2.

2.3. *Merging Over-split Units*

Most spike sorting algorithms may split one cell into several units due to strong bursting modulation. After running the entire algorithm, it is

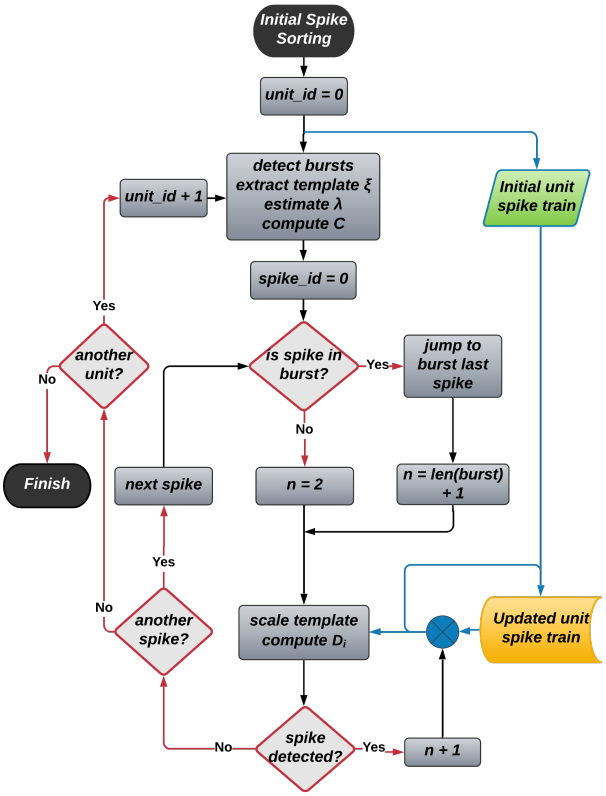

Figure 2: Flowchart of performing modulated BOTM after every last detected spikes in the bursts; Spike train is updated after every new spike is detected.

therefore necessary to merge together the units corresponding to the same cell. To automatically merge over-split units, we defined a method containing two steps; in the first steps similarity between sorted units are computed using cosine distance as following equation:

$$\cos(\theta) = \frac{A.B}{\|A\| \, \|B\|} \tag{2}$$

where A and B are sorted unit's templates. After selecting units with high similarity by means of thresholding, we checked the spike trains of corresponding units in the second step. In order to meet the second criteria (matching spike trains), the peak value of cross-correlogram between the two units spike trains, must be greater than 0.95 quantile of cross-correlogram between number of generated surrogates of a spike trains by ISI shuffling. The surrogates are obtained by randomly sorting the ISIs of the corresponding units, which generates independent spike trains with same ISI distribution and spike count as in unit spike train, while destroying temporal dependencies and firing rate profile. Following parameters used in this study: 0.8 set as a similarity threshold between templates and number of randomly generated surrogates is set to 20 for each unit. Cross-correlogram was computed in a window equal to estimated maximum bursting duration and bins of cross-correlogram was selected as minimum average inter-spike-interval between two units. After applying both steps, spike trains of selected units are merged by merging close spikes ($thr = 0.5ms$) into one, since some spikes could be assigned to two units. We used `Elephant` (Electrophysiology Analysis Toolkit) Python package [5] for generating spike train surrogates. Two steps of proposed algorithm for merging burst-related over split units are illustrated in Figure 3

2.4. *Model-free Approach*

Complex amplitude modulations are observed within many bursts, particularly later on the burst. This phenomena make our algorithm to fail in recovering undetected spikes. The solution in these cases (when we don't have the model of bursting) is to use model-free approach. We proposed a model-free method to recover undetected spikes as following: first we select potential spikes from spike sorting output that are not proceeded with other spikes in a small time window. We set the time window equal to $25ms$. Then, modulated template matching is performed in a small window including the potential spike in order to set threshold for template matching output. thresholds are set as 2/3 of the maximum value of discriminant function. Next, for each potential spike a modulated template matching is performed in a large time window estimated duration of burst. Modulated template matching is computed by using equation 3 and the time window size set equal to $250ms$. New spikes are selected whom cross the predefined threshold in template matching output.

$$D_i(t) = X(t)^T \mathbf{C}^{-1}(\mu\xi_i) \tag{3}$$

where $X(t)$ indicates a vector of concatenated snippets of data from number of selected channels, $\mathbf{C}_i$ is the spatiotemporal noise covariance matrix computed on pieces of noise, that is, periods of traces where no spikes were detected. $\mathbf{C}_i$ is computed by random selecting of chunks of noise snippets on selected channels for each unit. μ indicates scaling factor used to make templates smaller in amplitude and ξ_i is the templates of the unit i on the selected channels. An optional curation step could be applied before template matching to limit the number of false positive spikes. Since if a false positive spike is detected as a potential spike, it will result in false threshold selection and finding new false positive spikes.

To validate our method, we used HD-MEA recording of primary dissociated rat hippocampal culture, with simultaneous intracellular recording through patch-clamp. In this case, intracellular action potentials serve as a ground-truth information.

2.5. *Spike Sorting Evaluation and Benchmark*

Simulated bursting recordings are spike sorted with various spike sorters including, "HerdingSpikes" [11], "Tridesclous" [9], "HDsort" [6] and "Klusta" [14]. To assess the performance of our proposed methods, before and after their application, first we compared spike sorters output with `MEArec` ground-truth data. We compared spike trains of sorted units and ground-truth units and detected the *best-match* unit for each GT unit by selecting the maximum *agreement-score*. The *agreement-score* is computed via fol-

lowing equation.

$$agreement\_score = \frac{\#n_{matches}}{\#n_1 + \#n_2 - \#n_{matches}}$$

which computes the ratio of two spike trains intersection and their union. We computed three different performance metrics including accuracy, precision and recall as following, before and after post-processing step for *best-match* units:

$$\text{accuracy} = \frac{\#TP}{\#TP + \#FN + \#FP}$$

$$\text{precision} = \frac{\#TP}{\#TP + \#FP}$$

$$\text{recall} = \frac{\#TP}{\#TP + \#FN}$$

where TP are spikes that are correctly assigned to *best-match* unit, FN are spikes that remained unsorted and FP are spikes that are wrongly sorted to another unit. All evaluation process is performed using `SpikeInterface` Python package [2]. However, for evaluation of spike sorters performance of the experimental recording, intracellular action potentials used as ground-truth spikes.

3. Results

3.1. *Modulated TM Recovers Missed Spikes*

We applied modulated Bayesian optimal template matching (BOTM) as a post-processing step to the spike sorters output. This step could increase the overall performance of spike sorters by recovering missed spikes during initial spike sorting phase. We evaluated the spike sorting performance before and after performing modulated BOTM. An example of one burst unit with 8 spikes are plotted in Figure 4.A. Detected spikes by spike sorter are colored in blue, recovered spikes through modulated template matching are colored in green and undetected spikes are shown in red. Comparison across all bursts of the unit is shown in Figure 4.B, in which each dot is representative of a spike in a burst, blue dots shows spikes truly detected by "HDsort" spike sorter, recovered spikes through post-processing step are colored in green, and undetected spikes are shown as red dots. In order to limit number of newly detected false positive spikes due to small amplitude of scaled template, we set the $n_{max} = 5$.

The changes in recall values for 5 different simulated recordings are displayed in Figure 5. We quantified the performance of our approach using the recall score (Figure 5). Units with a recall below 0.2 were discarded. For the vast majority of the units the increase in recall after applying the modulated template matching is observable. Figure 6 indicates that as well as recall (Figure 6.C), the accuracy (Figure 6.A) of spike sorting is improved after performing our method. Since our algorithm works with small

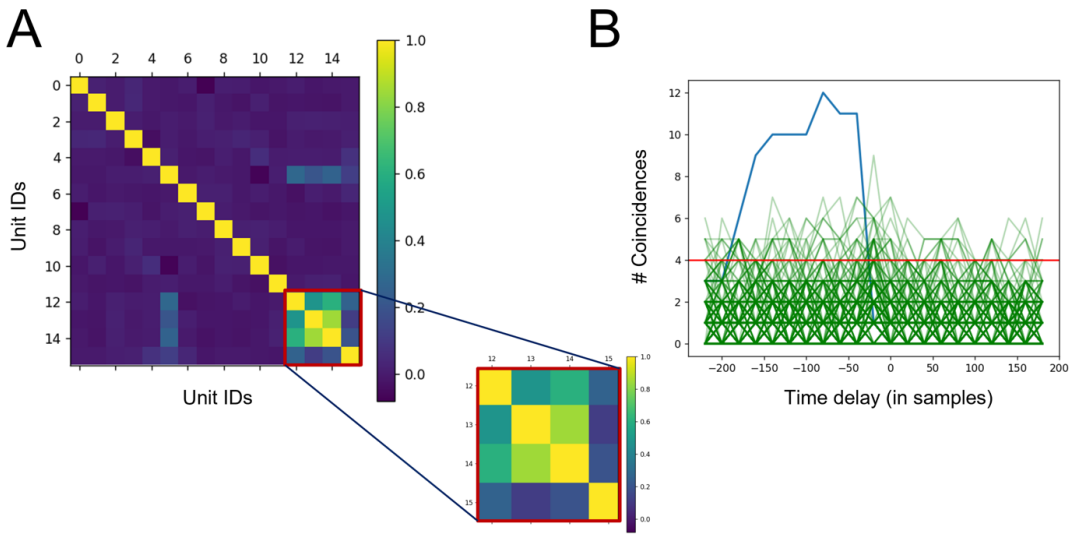

Figure 3: Steps of merging burst-related oversplits; Similarity check (A) and spike train cross-correlograms (B) **A**. Units with high template similarity are colored brighter. **B**. The cross-correlogram between two units and 20 generated surrogates are colored in blue and green, respectively. The threshold value is set to 0.95 quantile of cross-correlograms.

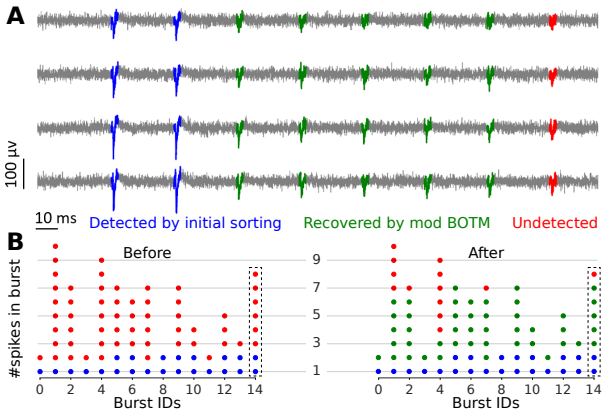

Figure 4: **A.** An example of simulated burst with 8 spikes (burst ID 14); recovered spikes via mod-BOTM are colored in green. Blue spikes were detected during initial sorting ("HDsort"). **B.** Each dot represents a spike within a burst. Most of the spikes missed by the initial sorting were recovered during our post-processing step.

version of template, it's prone to find some false positive spikes, which can explain the modest decrease in precision values (Figure 6.B). But in overall, the increase in recall was much significant than loss in precision, which is mirrored in accuracy increase.

3.2. *Merging Potentially Split Units Improved Agreement Scores*

To evaluate the efficiancy of our proposed method for merging burst-related oversplit units, we simulated an extracellular recording with 15 bursting units and a duration of 300 seconds, using the `MEArec` package. Random amplitude modulation factors (λ) were assigned to each unit from an interval between 0 and 0.4. Recording was then spike sorted using *Tridesclous* [9], then agreement score between tested and ground-truth units was computed. By checking agreement matrix between *Tridesclous* output and ground-truth units (see Figure 7.A), we found three units, which had a relatively high agreement score, but they were not *best-match* units, which means that they could be *oversplit* units. After checking template similarities and the cross-correlograms, three pairs of units were merged and the agreement scores between *best-match* and GT units are increased. Figure 7.B. shows agreement matrix after merging split units only for merged units.

3.3. *Model-free Approach Recovered Undetected Spikes*

To test our algorithm, we performed spike sorting using different algorithms including "Kilosort2" [13], "HerdingSpikes" [11], "SpyKING Circus" [16] and "Tridesclous" [9] on HD-MEA recording, considering that for one cell (patch-clamped one) we know what should be the output of the spike sorting. Comparing spike trains of output units with intracellular spike train, within a small time tolerance of 1 *ms* and computing agreement score for each unit, we selected a unit with highest agreement score as *best-match* unit. We applied Modulated template matching on the *best-match* unit. The scaling factor was set to 0.7. Figure 8 illustrates

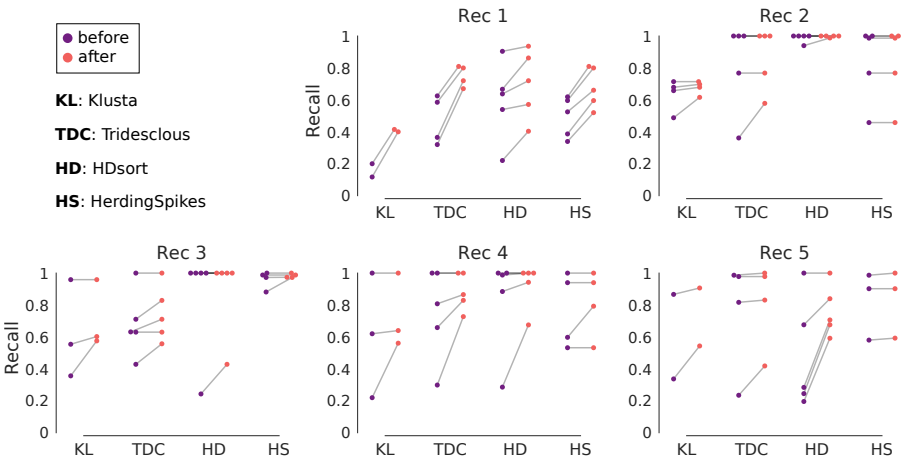

Figure 5: Recall scores computed and plotted for every best-match unit in the 5 simulated recordings spike sorted by 4 different spike sorters. Recall of unit assignments improved after post-processing using modulated BOTM.

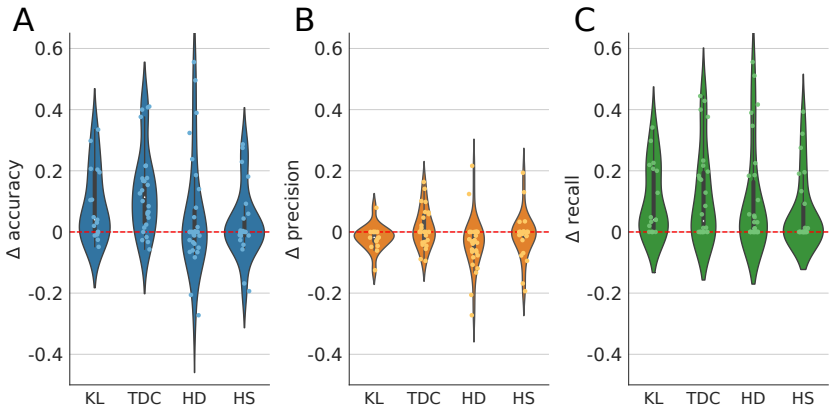

Figure 6: Violin plots of alteration in each spike sorter performance metrics before and after applying modulated BOTM; Each overlaid dot correspond to individual unit of spike sorter output. **A.** Accuracy generally increased after post-processing, yet did not lead to an overall loss of spike sorting precision due to false positives (**B**). However, a modest decrease in precision was observed for a few units (especially for "HDsort"). **C.** Post-processing resulted in an appreciable overall improvement in recall scores.

the application of the algorithm on an example burst with 14 spikes The waveforms colored in blue were correctly spike sorted by "HerdingSpikes" (including the yellow one), while the red ones remained undetected. The yellow waveform was chosen as the reference spike and modulated template matching was performed in a 250 ms time window immediately after the yellow waveform. Modulated template matching output is shown in seagreen in the bottom panel. 10 spikes including 7 undetected and 3 detected spikes were recovered by thresholding the template matching output. Figure 9 shows performance of our post-processing algorithm by comparing accuracy (panel A), precision (panel B) and recall (panel C), before and after applying it. It is important to mention that an extra post-processing step of thresholding similarity of spikes and template is performed for "Kilosort2", in which spikes with similarity (based on cosine distance) lower that $thr = 0.5$ with respect to unit template were discarded. This step is done to limit number of false positive spikes, since a false positive spike will result in finding new false positive spikes. The accuracy for all 4 spike sorters increased after post-processing, which means that number of recovered true positive spikes exceeds number of newly detected false positive spikes (if any). Precision value for "Kilosort2" increased dramatically after post-processing, which was mainly due to removing false positive spikes via similarity thresholding. For other sorters, there was no significant alteration after post-processing, because the initial precision values were close to the ideal value. Recall elevation after post-processing shows that number of true positive spikes increased, which tells us that our method were successful in detecting missed spikes.

4. Discussion and Conclusion

We proposed three methods for recovering missed spikes due to amplitude modulation in bursting recordings. Our routines comprised a template matching based methods to recover missed spikes in both model-based and model-free scenarios. We proposed a fully automatic approach to merge bursting units that were split due to amplitude variation within bursts. By evaluating the model-based method, we found that spike sorting accuracy and recall improved for most units after post processing with our approach. We have automated the process of merging spike trains through checking template similarities and cross-correlation between spike trains. Finally, we proposed a model-free method for recovering missed spikes. Our algorithm was successful in detecting new true positive spikes. However, accuracy and precision variation illustrated that, there were very few recovered false positive spikes. Further studies are necessary to explore the optimal trade-off between precision and recall in our algorithms. An important criteria which should be take into account in the future is synchronously bursting neurons, which pose challenges for our algorithms through spatiotemporal ovevrlap of spike waveforms [7]. In our algorithm, we scaled

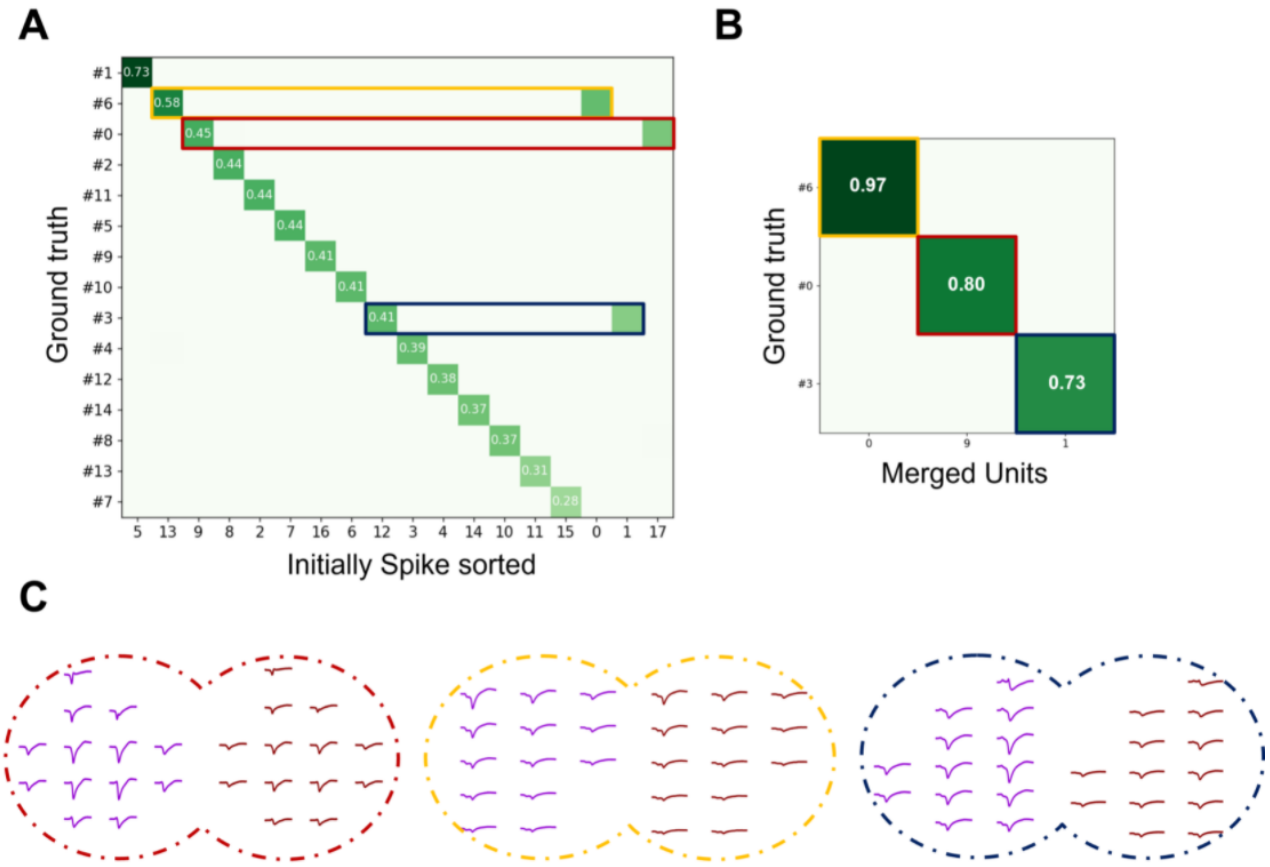

Figure 7: **A**. "Tridesclous" sorter split spikes in the bursts from units 0, 3 and 6 from ground-truth data in two separate units, which should be merged. **B**. After merging split units, units 1 and 12, 0 and 13, 9 and 17 are merged and respective agreement scores increased significantly. **C**. Templates (unit ID 13) colored in violet represent earlier (larger) spikes sorted in the bursts, while templates colored in red are representative of later (smaller) spikes in the bursts. Split units have similar templates that are only modulated in amplitude.

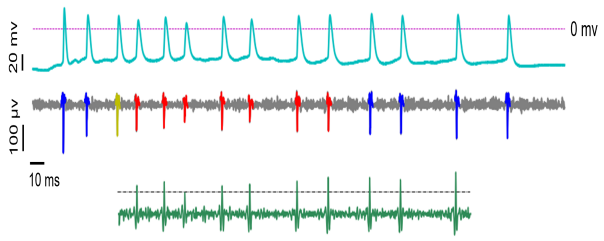

Figure 8: Upper panel shows intracellular trace of one burst; middle panel shows the burst recording of channel with largest template. Blue spikes are truly sorted spikes by the "Herdingspikes". Undetected spikes are colored in red and spikes selected for threshold setting colored by yellow. Bottom panel shows modulated template matching output applied after selected spike. Threshold is set as 2/3 of modulated template matching output of selected spike at the matching point.

the templates equally for the selected channels. However, in general the attenuation values for different channels are not symmetric and it could be modeled as a function of distance from *Soma*. Beyond spike rank within the burst, other factors may also be important determinants of amplitude attenuation, especially as the burst progresses. As a solution it is possible to consider spike amplitude variation within the burst as a function of spike rank and the ISI (inter-spike-interval) within bursts.

5. Acknowledgements

We thank our colleagues from ETH Zürich and Politecnico di Milano who provided insight and expertise that greatly assisted the research.

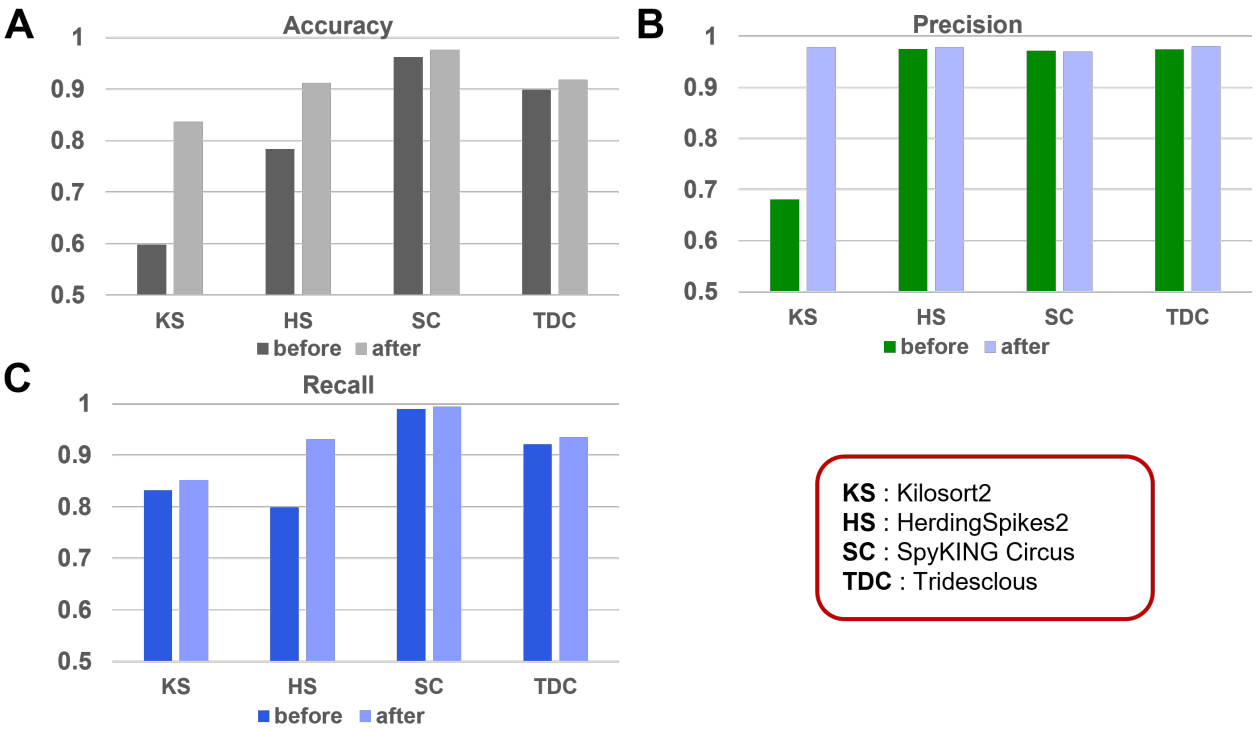

Figure 9: Improvement in accuracy is indicative of increase in number of true positives in comparison with false positive spikes (**A**). We did not observe an overall loss of spike sorting precision due to false positives (**B**). Appreciable overall improvement in recall scores are observed (**C**).

References

[1] Brian D Allen, Caroline Moore-Kochlacs, Jacob G Bernstein, et al. Automated in vivo patch-clamp evaluation of extracellular multielectrode array spike recording capability. *Journal of neurophysiology*, 120(5):2182–2200, 2018.

[2] Alessio P Buccino, Cole L Hurwitz, Samuel Garcia, et al. Spikeinterface, a unified framework for spike sorting. *Elife*, 9:e61834, 2020.

[3] Alessio Paolo Buccino and Gaute Tomas Einevoll. Mearec: a fast and customizable testbench simulator for ground-truth extracellular spiking activity. *Neuroinformatics*, 19(1):185–204, 2021.

[4] José Luis Carrillo-Medina and Roberto Latorre. Detection of activation sequences in spiking-bursting neurons by means of the recognition of intraburst neural signatures. *Scientific Reports*, 8(1):1–15, 2018.

[5] M. Denker, A. Yegenoglu, and S. Grün. Collaborative HPC-enabled workflows on the HBP Collaboratory using the Elephant framework. In *Neuroinformatics 2018*, page P19, 2018.

[6] Roland Diggelmann, Michele Fiscella, Andreas Hierlemann, and Felix Franke. Automatic spike sorting for high-density microelectrode arrays. *Journal of neurophysiology*, 120(6):3155–3171, 2018.

[7] Felix Franke, Robert Pröpper, Henrik Alle, et al. Spike sorting of synchronous spikes from local neuron ensembles. *Journal of neurophysiology*, 114(4):2535–2549, 2015.

[8] Felix Franke, Rodrigo Quian Quiroga, Andreas Hierlemann, and Klaus Obermayer. Bayes optimal template matching for spike sorting–combining fisher discriminant analysis with optimal filtering. *Journal of computational neuroscience*, 38(3):439–459, 2015.

[9] Garcia and Pouzat. tridesclous. `https://github.com/tridesclous/tridesclous`, 2015.

[10] Espen Hagen, Torbjørn V Ness, Amir Khosrowshahi, et al. Visapy: a python tool for

biophysics-based generation of virtual spiking activity for evaluation of spike-sorting algorithms. *Journal of neuroscience methods*, 245:182–204, 2015.

[11] Gerrit Hilgen, Martino Sorbaro, Sahar Pirmoradian, et al. Unsupervised spike sorting for large-scale, high-density multielectrode arrays. *Cell reports*, 18(10):2521–2532, 2017.

[12] John E Lisman. Bursts as a unit of neural information: making unreliable synapses reliable. *Trends in neurosciences*, 20(1):38–43, 1997.

[13] Marius Pachitariu, Nicholas A Steinmetz, Shabnam N Kadir, Matteo Carandini, and Kenneth D Harris. Fast and accurate spike sorting of high-channel count probes with kilosort. *Advances in neural information processing systems*, 29:4448–4456, 2016.

[14] Cyrille Rossant, Shabnam N Kadir, Dan FM Goodman, et al. Spike sorting for large, dense electrode arrays. *Nature neuroscience*, 19(4):634–641, 2016.

[15] Payam S Shabestari, Alessio P Buccino, Sreedhar S Kumar, Alessandra Pedrocchi, and Andreas Hierlemann. A modulated template-matching approach to improve spike sorting of bursting neurons. In *2021 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–4. IEEE, 2021.

[16] Pierre Yger, Giulia LB Spampinato, Elric Esposito, Baptiste Lefebvre, Stéphane Deny, Christophe Gardella, Marcel Stimberg, Florian Jetter, Guenther Zeck, Serge Picaud, et al. A spike sorting toolbox for up to thousands of electrodes validated with ground truth recordings in vitro and in vivo. *Elife*, 7:e34518, 2018.

B | Conference Paper

A Modulated Template-Matching Approach to Improve Spike Sorting of Bursting Neurons

A modulated template-matching approach to improve spike sorting of bursting neurons

Payam S. Shabestari*†, Alessio P. Buccino*, Sreedhar S. Kumar*,
Alessandra Pedrocchi†, and Andreas Hierlemann*
Email: psadeghi@student.ethz.ch, alessandra.pedrocchi@polimi.it, {andreas.hierlemann, sreedhar.kumar, alessio.buccino}@bsse.ethz.ch
*Department of Biosystems Science and Engineering, ETH Zurich, Basel, Switzerland
†Department of Electronics, Informatics, and Biomedical Engineering, Politecnico di Milano, Milan, Italy

2021 IEEE Biomedical Circuits and Systems Conference (BioCAS) | 978-1-7281-7204-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/BIOCAS49922.2021.9644995

***Abstract*—In extracellular neural electrophysiology, individual spikes have to be assigned to their cell of origin in a procedure called "spike sorting". Spike sorting is an unsupervised problem, since no ground-truth information is generally available. Here, we focus on improving spike sorting performance, particularly during periods of high synchronous activity or so-called "bursting". Bursting entails systematic changes in spike shapes and amplitudes and remains a challenge for current spike sorting schemes. We use realistic simulated bursting recordings of high-density micro-electrode arrays (HD-MEAs) and we present a fully automated algorithm based on template matching with a focus on recovering missed spikes during bursts. To compare and benchmark spike-sorting performance after applying our method, we used ground-truth information of simulated recordings. We show that our approach can be effective in improving spike sorting performance during bursting. Further validation with experimental recordings is necessary.**

I. Introduction

Extracellular voltage recordings are a common type of electrophysiological data in experimental and clinical neuroscience. The signals recorded by each electrode are, however, a superposition of the activity of a number of signal sources or neurons in the vicinity of an electrode. A computational step is thus necessary to disentangle the recorded signals and assign single spikes to their putative neuronal sources. This problem is called "spike sorting" [1]. Over the years, considerable efforts have been made to build algorithms that perform automated spikes sorting. Yet, significant challenges remain and need to be addressed before a generally applicable algorithmic solution will be available. An important challenge for spike sorting algorithms is the observed variability of spike amplitudes and shapes that depend on the recent firing history of the neuron [2], [3]. A prominent example is neuronal bursting: patch-clamp experiments, combined extracellular-juxtacellular recordings [4] and computational studies [5] suggest that, during bursting, extracellular spikes become lower in amplitude. If unaccounted, such amplitude changes may contribute to spikes of a certain unit being erroneously assigned to other units (false positives) or being missed altogether (false negatives).

In this study, we demonstrate a template matching-based post-processing step that limits sorting errors introduced through spike amplitude variability. We added a procedure in which spikes belonging to a burst can be recovered from the output of existing spike sorting algorithms. We utilized the *raw* output of an automated spike sorting method and combined it with the recording traces to compute unit *templates*, i.e., the average extracellular waveforms. To recover potentially missed spikes within bursts, we performed a further template matching step, but this time, focused on short time windows after isolated spikes and trailing spikes within bursts. In addition, we modulated the template amplitude in dependence on the spiking history. For this purpose, we introduced a novel modulated Bayes optimal template matching (BOTM) step. We used the `MEArec` Python package [6], which is a highly customizable simulator for generating ground-truth extracellular spiking activity, to generate a variety of simulated recordings exhibiting bursting modulation. Knowing the full ground-truth information in terms of spiking activity allowed us to validate the spike sorting output and to benchmark the algorithm.

The article is organized as follows: in Section II we describe the method used for simulating bursting recordings, then we illustrate the algorithm for recovering spikes within bursts. Finally we describe the approach for evaluating the modulated BOTM outputs; in Section III we present the results of applying modulated BOTM to spike sorters and how it can improve the overall spike sorting performance; finally, in Section IV, we contextualize this work by abstracting its potential and limitations and by indicating future directions.

II. Methods

A. Simulated datasets

In order to generate multiple recordings with bursting units, we used the `MEArec` Python package [6]. We first simulated sets of random bursting spike trains. Second, templates from biophysically realistic cell models were selected that did not overlap in space (this was done to limit the presence of spatiotemporal collisions) and then convoluted with the spike trains by applying an amplitude modulation. This procedure included a random and small Gaussian modulation to reproduce physiological variations and a bursting term, which was computed as:

978-1-7281-7204-0/21/$31.00 ©2021 IEEE

$$\mu[n] = \left(\frac{\bar{t}_{isi}}{(n-1) \cdot t_{max}} \right)^{\lambda} \quad (1)$$

where $\mu[n]$ is the amplitude modulation factor of the n^{th} consecutive spike within a burst ($n \geq 2$), $\bar{t}_{isi}$ is the average inter-spike interval within a burst, t_{max} is the maximal duration of a single burst (used as a normalization factor), and λ is an exponent that regulates the degree of attenuation. We simulated 5 recordings from different realizations of randomly generated bursty spike trains. λ was randomly drawn from the interval $[0, 0.4]$, and each recording contained the activity of 5 neurons (25 neurons in total). Extracellular recordings were computed on a 100-Channel microelectrode array in a 10×10 configuration with a 15 μm electrode center-to-center distance.

B. Modulated BOTM

Bayes optimal template matching computes matched filter outputs for the given template and adds constants that depend on the energy of the template and the probability that it occurs in the data to compute the final discriminant function [7].

A burst was defined as a group of spikes in which two or more consecutive inter-spike-intervals (ISIs) fell within a certain threshold. The threshold ISI for each unit was chosen as the valley of its bimodal ISI distribution. Using spikes in the detected bursts, we computed $\bar{t}_{isi}$. The exponent $\hat{\lambda}_i$ was then estimated for each unit i using a least squares fit of unit spike amplitudes during bursting periods to the model in eq. 1. A positive value of the estimate was indicative of amplitude attenuation within bursts; conversely, we considered units with null or negative $\hat{\lambda}_i$ as non-bursting units. The modulated BOTM was applied as follows: for each unit with detected bursts, we first computed the spatiotemporal noise covariance matrix $\mathbf{C}$ in the recording using snippets of traces, where no spikes were detected [7], [8]. For each unit, only 7 channels with the largest amplitude were used. Unit templates were extracted in the same channels (from the spike sorting output). Next we applied modulated BOTM to each isolated spike and the last detected spike in each burst. We used a scaled version of each template using $\hat{\mu}_i[n]$ for the n^{th} spike within a burst of unit i, computed using the corresponding estimated exponent $\hat{\lambda}_i$. A time window of $2 \cdot \bar{t}_{isi}$ was used for template matching if spikes were in bursts. For isolated spikes, the maximal $(\bar{t}_{isi})$ over all bursts was considered and n was set to 2. Over this window we computed the BOTM discriminant function D_i for unit i as follows:

$$D_i(t) = \hat{\mu}_i[n] \left(X(t)^T \mathbf{C}^{-1} \xi_i - \frac{\hat{\mu}_i[n]}{2} \xi_i^T \mathbf{C}^{-1} \xi_i \right) + ln(p(i))$$

where $X(t)$ is a column vector of size $n_s n_c$ made of concatenated data snippets of length n_s from n_c channels, $\mathbf{C}$ the spatiotemporal noise covariance matrix of size $n_s n_c \times n_s n_c$, ξ_i the template of the i^{th} unit, also a column vector of size $n_s n_c$, and $p(i)$, the prior probability of observing a spike of unit i. The following values were used in our study: $n_s = 190$, $n_c = 7$, and a constant prior $p(i) = 0.5$ for all units. Spikes were recovered by detecting peaks in

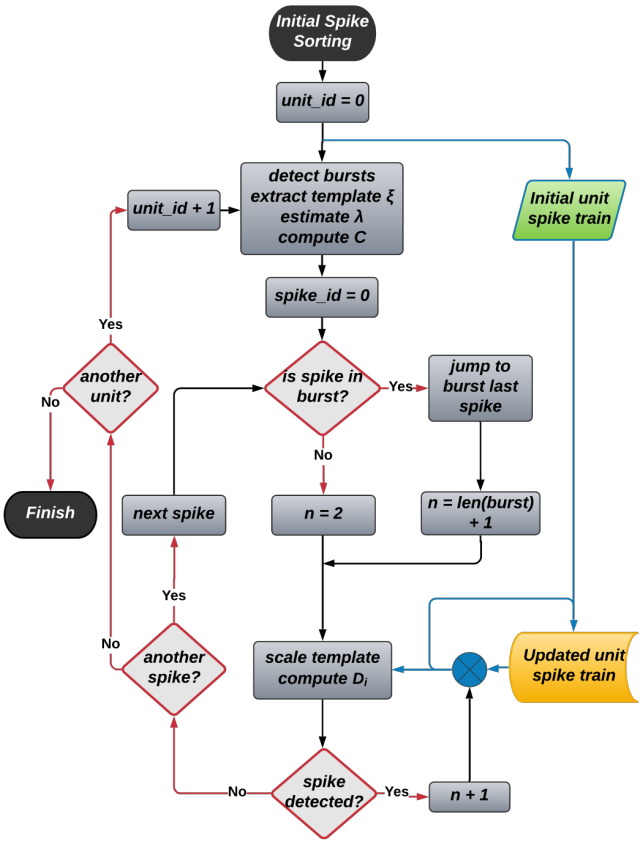

Fig. 1. Block diagram of the proposed post-processing step to spike sorting. Modulated Bayes optimal template matching was applied in a small window beyond each isolated spikes and the trailing spike in each burst. If a new spike was detected, the initial unit spike train was correspondingly updated.

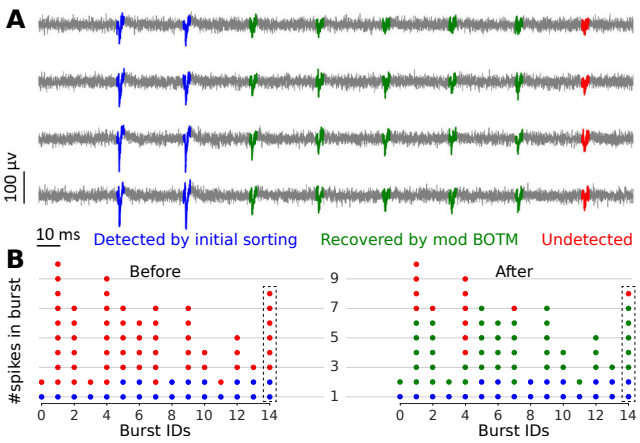

Fig. 2. **A.** Data from a simulated dataset including 4 channels demonstrate the amplitude attenuation during bursting. Only two of the 8 spikes in this burst (Burst ID 14 in **B**) were detected during initial sorting ("HDsort", blue); the next 5 spikes were recovered after applying modulated BOTM to the "HDsort" output (green); the last spike in the burst remained undetected (red). **B.** Most of the spikes missed by the initial sorting were recovered during our post-processing step. Each column represents a burst and each dot a spike within it. Blue and red indicate that spikes were detected or not detected during the initial sorting (in this example, using "HDsort"). The spikes recovered during post-processing are indicated in green.

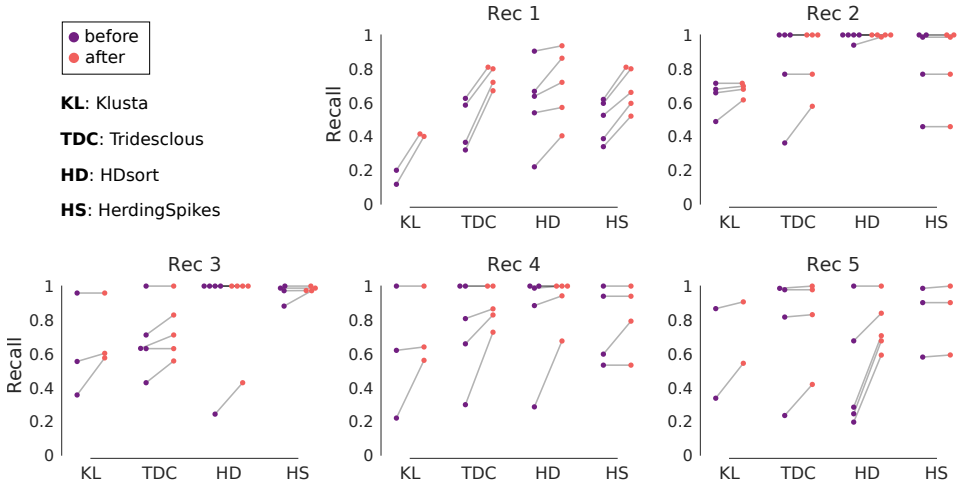

Fig. 3. Recall (fraction of true spikes that were recovered), of unit assignments improved after post-processing using modulated BOTM. Recall scores are shown for 5 recordings with 4 different sorters, before and after post processing.

the discriminant function. If a new spike was detected, we proceeded iteratively (incrementing n) to look for further spikes until a window with no spikes was reached or n reached n_{max}. We set n_{max} to 5 to limit the number of false positives. Figure 1 illustrates the proposed modulated BOTM algorithm applied as a post-processing step to the spike-sorting output.

C. Spike sorting benchmark and evaluation

Recordings were sorted with four different spike sorters, namely "HerdingSpikes" [9], "Tridesclous" [10], "HDsort" [11] and "Klusta" [12]. All spike sorters and comparisons were run using the SpikeInterface framework [13]. We compared spike sorting outputs with ground-truth spiking activity from the `MEArec` simulated recordings. Comparison to ground-truth data sets was used to benchmark spike sorting algorithms. For all pairs of ground-truth and sorted spike trains, we first counted how many events were matched within a time tolerance of 0.4 ms. We used a best-match approach, in which each ground-truth unit was associated with a test unit that had the best agreement score, independently of all others units. With the list of matched units we computed the following performance metrics for each ground-truth unit:

$$\text{accuracy} = \frac{\#TP}{\#TP + \#FN + \#FP}$$

$$\text{precision} = \frac{\#TP}{\#TP + \#FP}$$

$$\text{recall} = \frac{\#TP}{\#TP + \#FN}$$

where $\#TP$ indicates the number of true positives, i.e., spikes of the ground truth units that were correctly matched to spikes of matched sorted test units; $\#FN$ is the number of false negatives, which included spikes of ground-truth units, which were not assigned to matched sorted test units; and $\#FP$ is the number of false positives, which are spikes that were not included in the ground-truth spike trains and that were incorrectly assigned to matched sorted test units.

III. Results

Post-processing by applying the proposed modulated BOTM method to the output of an initial spike sorting enabled the recovery of bursting spikes that were otherwise missed (false negatives). In Figure 2A we show an example of a burst in 4 selected channels (signals of 4 electrodes). The waveforms in blue were correctly spike sorted by HDsort, while the green ones were only recovered after our post-processing step. The last spike (red), remained undetected since we had limited the recovery to $n_{max} = 5$ spikes. A comparison across many bursts is shown in Figure 2B. Dots represent individual spikes within each burst. Spikes found by HDsort are shown in blue and the missed spikes in red. Spikes recovered after the modulated BOTM post-processing step are colored in green. The added BOTM step enabled the recovery of almost all initially missed spikes in the bursts. We quantified the performance of our approach using the recall score (Figure 3). Units with a recall below 0.2 were discarded. Across sorters, the vast majority of the units showed an increased recall after adding the BOTM step.

We observed an overall increase in both accuracy and recall for all sorters (Figure 4A, C). Since our algorithm looks for low-amplitude re-scaled spikes in bursts, there is the risk of finding false positive spikes. Such spurious spikes would lead to a decrease in the precision of the sorter. However, with the parameters we used, we did not observe an appreciable loss of precision except for a modest decrease for "HDsort" (Figure 4B). In summary, the increase in recall in our data was much larger relative to the loss in precision, which steered the accuracy change towards positive values.

IV. Discussion and Conclusions

Here, we presented an approach to recover spikes that were missed by automatic spike sorters due to spike amplitude changes that accompany bursting. In our post-processing routine, we applied amplitude-modulated template matching locally, to recover potentially missing spikes. We tested the

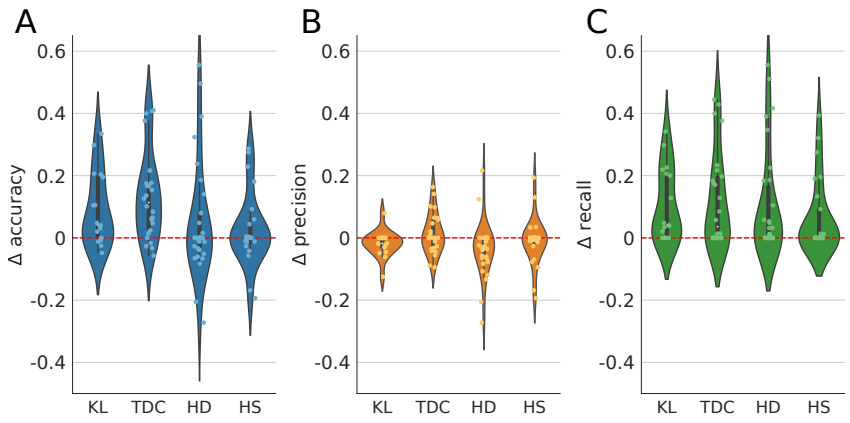

Fig. 4. Distributions of changes in each sorter's performance measures before and after performing the modulated BOTM step. Units were pooled from all 5 recordings shown in Figure 3; overlaid dots correspond to individual units. **A.** Accuracy generally increased after post-processing, yet did not lead to an overall loss of spike sorting precision due to false positives (**B**). However, a modest decrease in precision was observed for a few units (especially for "HDsort"). **C.** Post-processing resulted in an appreciable overall improvement in recall scores.

routine on simulated ground truth data. We found that spike sorting accuracy and recall improved for most units after post processing with our approach. Moreover, we did not observe an appreciable loss of spike sorting precision as a result of the post-processing.

Recall is the fraction of true spikes that were recovered. A consistent improvement of recall across all sorters suggests that our approach was successful in reducing the number of false negatives or missed spikes. However, the added sensitivity comes with the risk of having more false positives, which is reflected in the modest changes observed in precision and the slight drop in accuracy in a few units (especially for HDsort).

No changes in accuracy, recall and precision were observed for many units. One possible reason is that if the detection and signal assignment to a unit was already highly accurate, the additional post-processing step was not expected to further improve the accuracy. In such cases, the accuracy remained the same or decreased slightly if a few false positives spikes were inadvertently detected. Alternatively, low SNR could yield units with low amplitudes, thereby limiting the ability of our approach to improve sorting performance.

In this study we present and validate a novel method to improve the quality of spike sorting when spike features are non-stationary and history dependent. The algorithm can be readily applied to the output of any generic automated spike-sorting procedure. Further studies are necessary to explore the optimal trade-off between precision and recall in our algorithm. In future, we will extend the approach to rescue spikes of units that are synchronously bursting, where the spatiotemporal ovevrlap of spike waverforms may pose challenges for our algorithm [14]. Further, a better understanding of how extracellular spike features change during bursting will enable us to advance and refine our approach. To this end, ground truth validation studies with experimental data will be an important and necessary next step.

ACKNOWLEDGEMENTS

This work was supported by the ERC Advanced Grant 694829 "neuroXscales", and the ETH Zurich Postdoctoral Fellowship 19-2 FEL-17 (APB).

REFERENCES

[1] H. G. Rey, C. Pedreira, and R. Q. Quiroga, "Past, present and future of spike sorting techniques," *Brain research bulletin*, vol. 119, pp. 106–117, 2015.
[2] M. S. Fee, P. P. Mitra, and D. Kleinfeld, "Variability of extracellular spike waveforms of cortical neurons," *Journal of neurophysiology*, vol. 76, no. 6, pp. 3823–3833, 1996.
[3] G. T. Einevoll, F. Franke, E. Hagen, *et al.*, "Towards reliable spike-train recordings from thousands of neurons with multielectrodes," *Current opinion in neurobiology*, vol. 22, no. 1, pp. 11–17, 2012.
[4] B. D. Allen, C. Moore-Kochlacs, J. G. Bernstein, *et al.*, "Automated in vivo patch-clamp evaluation of extracellular multielectrode array spike recording capability," *Journal of neurophysiology*, vol. 120, no. 5, pp. 2182–2200, 2018.
[5] E. Hagen, T. V. Ness, A. Khosrowshahi, *et al.*, "Visapy: a python tool for biophysics-based generation of virtual spiking activity for evaluation of spike-sorting algorithms," *Journal of neuroscience methods*, vol. 245, pp. 182–204, 2015.
[6] A. P. Buccino and G. T. Einevoll, "Mearec: a fast and customizable testbench simulator for ground-truth extracellular spiking activity," *Neuroinformatics*, vol. 19, no. 1, pp. 185–204, 2021.
[7] F. Franke, R. Q. Quiroga, A. Hierlemann, and K. Obermayer, "Bayes optimal template matching for spike sorting–combining fisher discriminant analysis with optimal filtering," *Journal of computational neuroscience*, vol. 38, no. 3, pp. 439–459, 2015.
[8] C. Pouzat, O. Mazor, and G. Laurent, "Using noise signature to optimize spike-sorting and to assess neuronal classification quality," *Journal of neuroscience methods*, vol. 122, no. 1, pp. 43–57, 2002.
[9] G. Hilgen, M. Sorbaro, S. Pirmoradian, *et al.*, "Unsupervised spike sorting for large-scale, high-density multielectrode arrays," *Cell reports*, vol. 18, no. 10, pp. 2521–2532, 2017.
[10] Garcia and Pouzat, "tridesclous." https://github.com/tridesclous/tridesclous, 2015.
[11] R. Diggelmann, M. Fiscella, A. Hierlemann, and F. Franke, "Automatic spike sorting for high-density microelectrode arrays," *Journal of neurophysiology*, vol. 120, no. 6, pp. 3155–3171, 2018.
[12] C. Rossant, S. N. Kadir, D. F. Goodman, *et al.*, "Spike sorting for large, dense electrode arrays," *Nature neuroscience*, vol. 19, no. 4, pp. 634–641, 2016.
[13] A. P. Buccino, C. L. Hurwitz, S. Garcia, *et al.*, "Spikeinterface, a unified framework for spike sorting," *Elife*, vol. 9, p. e61834, 2020.
[14] F. Franke, R. Pröpper, H. Alle, *et al.*, "Spike sorting of synchronous spikes from local neuron ensembles," *Journal of neurophysiology*, vol. 114, no. 4, pp. 2535–2549, 2015.

C | Pseudo-algorithms

Pseudo-algorithms for applying both modulated template matching approaches on recordings are presented here.

Algorithm C.1 Modulated Template Matching 1

```
for sorter unit ids do
    Estimate modulation factor λ
    if λ > 0 then
        Detect bursts via critical ISI
        for bursts in unit id do
            looking-for-spikes = True
            n = 0
            while n < n_max and looking-for-spikes = True do
                Scale the unit template
                Compute BOTM output immediately after last detected spike in bursts
                Threshold crossing the BOTM output signal
                if threshold is not crossed then
                    looking-for-spikes = False
                end if
                n+ = 1
            end while
        end for
    end if
end for
Concatenate initial spike train with recovered spikes
```

Algorithm C.2 Modulated Template Matching 2

Initialize the parameters Δt, $\Delta t'$, $\Delta t''$, μ
for *all spikes* **do**
    Compute inter-spike-intervals (ISIs)
    if $ISI > \Delta t$ **then**
        Select spikes
    end if
end for
for *selected spikes* **do**
    Do scaled template matching ($\mu\xi$) on the selected spike
    Set threshold value as a fraction of MF output peak
    Apply MF with scaled template on the predefined window $\Delta t'$
    Select the spikes crossing the estimated threshold
end for
Merge initial spike train with recovered spikes by time tolerance $\Delta t''$

D | Matched Filtering

The aim is to find a filter f so that its response y_t to a noisy recording x_t is maximal if a given signal ξ is present and minimal otherwise. If the signal is present, considering η_t as the noise process, the recording could be written as:

$$x_t = \eta_t + \xi \tag{D.1}$$

The noise covariance matrix is computed as expected value of noise process product with itself, given by $C = E(\eta\eta^T)$. The filter output is usually given by the convolution between the filter and data, therefore, $y_t = f^T x_t$ where f^T stands for time reversed filter.
If the signal ξ is present $y_t = f^T(\eta_t + \xi_t)$ and if not $y_t = f^T\eta_t$. SNR (signal-to-noise ratio) after filtering is computed as:

$$SNR = \frac{|y_s|^2}{E(|y_n|^2)} = \frac{|f^T\xi|^2}{f^T C f} \tag{D.2}$$

By rewriting C as $C = UU^T$, the Cholesky factorization or matrix square root U and exploiting Cauchy-Schwarz inequality we obtain:

$$SNR = \frac{|(Uf)^T(U^{-1}\xi)|^2}{(Uf)^T(Uf)} \leq \frac{(Uf)^T(Uf)(U^{-1}\xi)^T(U^{-1}\xi)}{(Uf)^T(Uf)} \tag{D.3}$$

which can be simplified to:

$$SNR \leq (U^{-1}\xi)^T(U^{-1}\xi) = \xi^T C^{-1}\xi \tag{D.4}$$

The upper bound of the inequality can be achieved by choosing appropriate value for f as:

$$f = \frac{1}{\alpha} C^{-1}\xi \tag{D.5}$$

Setting α equal to $\sqrt{\xi^T C^{-1}\xi}$, the minimum power distortionless response beamforme (MPDR) and by adjusting α equal to $\xi^T C^{-1}\xi$ the minimum variance distortionless response beamformer (MVDR) is obtained.

List of Figures

1.1 Raster plot of bursting neurons . . . 3
1.2 Schematic and a photo of a Multi-electrode-array recording . . . 5
1.3 Overview of the generic spike sorting workflow . . . 7
1.4 Convolutional template matching technique . . . 8
1.5 Intracellular action potentials (APs) transform within bursts . . . 9
1.6 Extracellular waveforms transformation within bursts . . . 10
1.7 Undetected spikes in the burst . . . 11

2.1 Overview of extracellular recording simulation by `MEArec` . . . 14
2.2 An example of simulated burst . . . 16
2.3 Agreement matrix between ground-truth and tested units . . . 18
2.4 Flowchart of method for recovering missed spikes . . . 22
2.5 Steps of merging oversplit units . . . 24
2.6 An example of burst with complex amplitude modulation . . . 25

3.1 Recovered spikes in a burst trace . . . 28
3.2 Agreement matrix before and after post-processing step . . . 29
3.3 Recall change after post-processing . . . 30
3.4 Performance metrics before and after TM . . . 30
3.5 Merging split bursting units . . . 32
3.6 Modulated template matching approach for recover missed spikes . . . 34
3.7 Post-processing of simulated `MEArec` recording . . . 34
3.8 Performance metrics before and after applying modulated template matching on real recording . . . 36

4.1 Attenuation is spatially asymmetric . . . 38

List of Tables

2.1 Performance Metrics 20