Politecnico di Torino

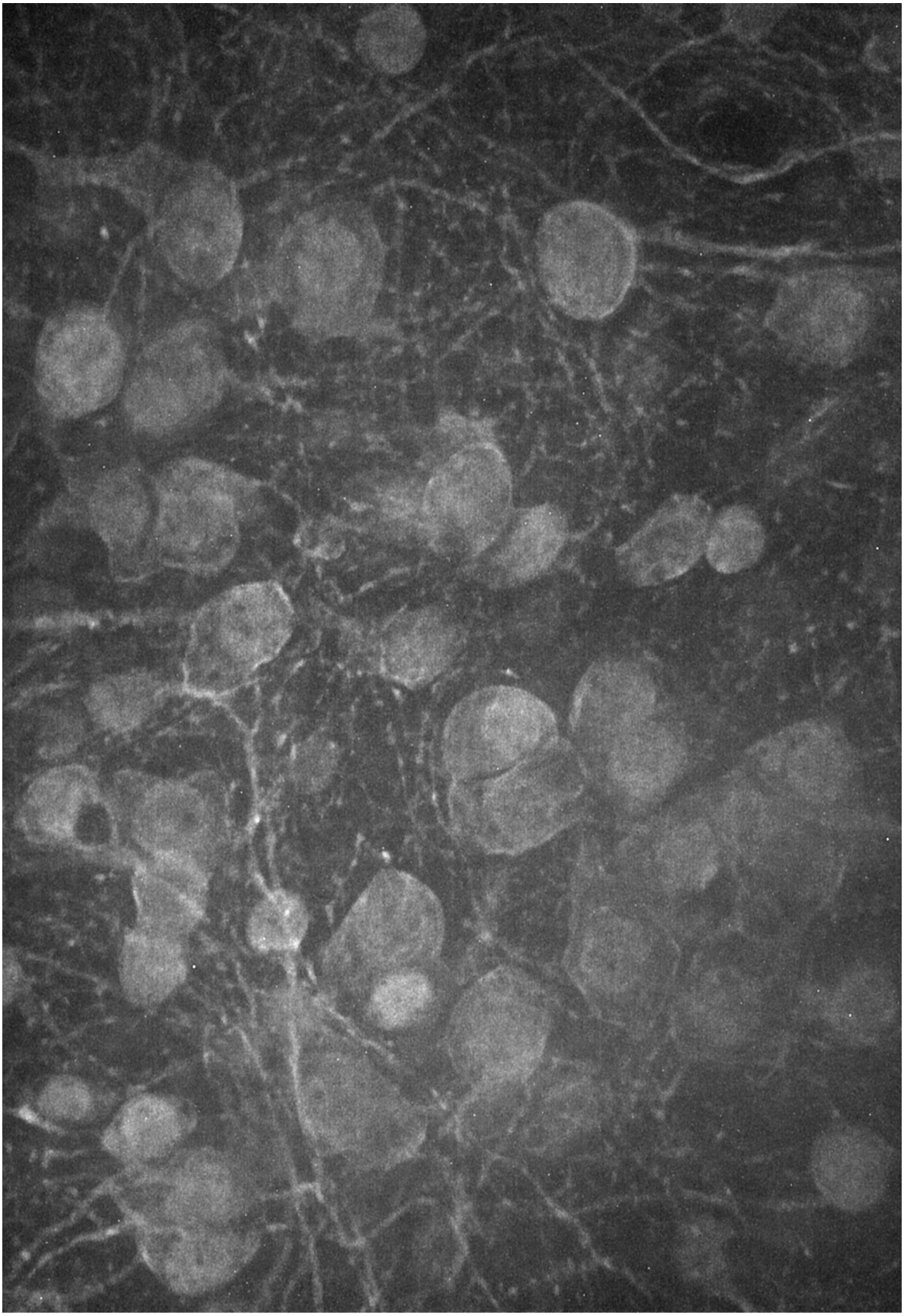
Physics of Complex Systems
A.a 2022/2023
Sessione di Laurea Aprile 2023

Neuronal classification based on high spatial and temporal resolution extracellular electrophysiological recordings performed using HD-MEAs

Advisor:
Andrea Gamba
Andreas Hierlemann

Candidate:
Francesco Modena

Contents



Acknowledgements

I would like to thank Prof. Andrea Antonio Gamba, for following me along this project and for his counsel, and Prof. Andreas Hierlemann, for allowing me to work in his wonderful laboratory and for giving me this amazing opportunity.

I would also like to thank greatly Dr. Sreedhar Saseendran Kumar, for being an amazing example of competence and passion, for guiding me in this work, always giving me the opportunity to figure solutions out on my own; and my thanks to Tobias Gaenswein, for helping me more times than I can recount, and for sharing his vast knowledge with me, along with not a few laughs, and to Dr. Julian Bartram, for sharing his data set with me.

Thank you to my family. To my mother, Marta, and my father, Andrea, for being role models of what and how I want to be in life, and giving me the instruments to carve my own path, alongside their footprints. Thanks to Alice and Marco, for both being examples in your own ways, for being so smart and talented and, honestly, for not leaving me anything to wish in a sister and brother. To my nonno Andrea, for infecting me with his love for science and mountains. And thank you to my nonno Giulio, for being my flatmate and always sharing his encyclopedias, and for all those times at the natural sciences museum.

Thank you Teresa, for being the first one I turn to with my problems and my joys, for your insight, your support and for all our conversations, which made me grow in ways I did not know possible. Thank you for the way in which you are going to hate me a little bit, for writing this here.

Thank you to my friends at home, for always having my back even when sprawled across Europe; to Matteo, Maria and Jessica, for always being there for me, growing together, and for inevitably being in every single acknowledgement I have written the past 15 years. And thank you to all my friends and colleagues at BEL, in particular Odysseas, Neethu and Amanzhol, for being truly remarkable human beings, for motivating me and making me laugh through hardship.

Thank you to all that I have not nominated, but have a safe spot in my heart.

0.1 Abbreviations

In the review, the following abbreviations will be used:

- PreS: presynaptic
- PostS: postsynaptic
- AP: action potential
- EAP: extracellular action potential
- AIS: axon initial segment
- GABA: $\gamma-$aminobutyric acid
- DIV: days in vitro
- HD-MEA: high-density microelectrode array
- EPSP: excitatory postsynaptic potential
- IPSP: inhibitory postsynaptic potential
- Pt-black: platinum black
- u: sorted unit, or putative neuron. The output of the spike-sorting procedure
- CV: coefficient of variation
- ISI: inter-spike interval
- FR: firing rate
- IBI: inter burst interval
- E: excitatory neurons
- I: inhibitory neurons
- ICCS: immunocyto-chemistry staining
- EDTC: ensemble decision tree classifier
- GBDT: gradient boosted decision tree classifier
- SVC: support vector machine classifier
- STP: spike-transmission probability
- CCG: cross-correlogram

0.2 Abstract

Strategies to navigate the complexity of the brain are vital for a bottom-up understanding of the function (and dysfunction) of neural circuits, and thus the brain itself. The first step towards reducing this complexity is to create a parts list of the individual elements comprising neural circuits. Identifying functionally distinct types of neurons enables the systematic analysis of their individual contributions to circuit function. Yet, reliable and high-throughput neuron type classification remains a challenge. Modern extracellular electrophysiological devices offer access to the activity of neural ensembles at high spatiotemporal resolution. In this study we asked if multi-scale features harvested from high-resolution extracellular electrophysiology enable reliable and high-throughput profiling of neurons into two broad functional classes: excitatory and inhibitory. We addressed this question using generic *in vitro* networks of rat primary dissociated hippocampal neurons grown on high-density microelectrode arrays. Using ground truth labels — assigned based on spike train correlations or molecular features— we assessed the feasibility of such a task. Eventually, we used the labelled data set to train classification models and evaluated the influence of individual features on overall performance, and what information this could give us regarding the underlying physiological behavior.

Chapter 1

Introduction

Understanding the principles of information processing in the brain is one of the primary challenges in neuroscience [1]. Information processing is widely thought to be an emergent property of the dynamics of large neuronal circuits [2]. Studying such a system requires simultaneous access to the electrophysiological readouts from a large number of neurons that make up such circuits. With technological advancements in extracellular electrophysiology, it is increasingly possible to record electrical activity from hundreds to thousands of neurons at once [3]. However, the high dimensional complexity of these readouts, and the structural heterogeneity of the neural circuits have been major challenges towards interpreting such data sets. If any improvements are to be made in this direction, they have to start from bottom up: by trying to reduce this complexity and remove degrees of freedom from the problem.

Neurons are the elementary computational units of the brain. The first and central step towards reducing complexity would be to identify functionally distinct types of neurons so that the working of a network of neurons can be simplified in terms of interactions between distinct neuronal sub-populations. Such simplifications are expected to enhance our understanding of the working principles of neuronal ciruits in the brain.

Self-organizing neuronal networks cultured *in vitro* are promising model systems to study a generic assembly of biophysically complex neurons. They can be grown on planar microelectrode arrays that offer stable, long-term, bi-directional (record and stimulate) access to the electrical activity of these neurons. In particular, the use of high-density microelectrode arrays (HD-MEAs) has great potential in this field because they enable the acquisition of spatially and temporally highly resolved information, covering, possibly, the whole neuronal network. However, it would be desirable to simplify such rich data sets, for example, by stratifying the recorded neurons by their functional classes, to improve their interpretability. The two main functional neuron classes that make up neuronal ensembles are *excitatory* and *inhibitory* neurons. Interactions between these classes of neurons are considered to be critical in shaping the activity of the network [3][4].

Currently, there are no methods to reliably infer the functional class of a neuron in *in*

vitro models based only on electrical readouts and activity patterns. In this study, we explored the feasibility of distinguishing neurons into excitatory and inhibitory classes based solely on multiscale features derived from extracellular readouts acquired using HD-MEAs. We used ground-truth data, in which neuronal class labels were assigned independently of electrophysiology, to benchmark the reliability of our classification.

The following sections will provide a short overview of the subject and electrophysiological theory behind the thesis. After an introduction on neurons and their features, we will talk about their classification: what do we mean when talking about cell types, how and why we should distinguish them and what problems arise in this endeavor. The discussion will then focus on the specific classes of interest to us, and why we chose to work with an *in vitro* model. The electrophysiological aspects of the thesis, and the tools to deal with them, will then be described, and a brief summary of the structure of the thesis will be presented.

1.1 Neurons are the fundamental units of the nervous system

Neurons are the principal cellular elements of the nervous system together with glial cells. They are functionally distinct and form the basic components of the electrical network of the nervous system that is at the core of all brain functions. While there is great anatomical variation among neurons in the nervous system, they share a general morphological form and the ability to respond in an electrical and ligand-dependent manner.

Neurons have four defined regions: soma (or cell body), dendrites, axon and synaptic terminals (Fig 1.1). The **soma**, which has a radius around 0.005 mm to 0.1 mm in mammals, contains the cell's nucleus and much of the genomic expression and synthetic machinery of the cell. Neurons are generally thought to have input and output poles. Elaborate branching tree-like extensions – called **dendrites** – arise from the soma and form the input pole of the neuron. They act as the receivers for signals arising from other cells of the nervous system. A unitary **axon** acts as the transmitter or output pole of the cell. It can vary in length from ca. 0.1 mm to more than 2 m and can have a diameter ranging between 0.22 µm and 20 µm. The axon conducts electrochemical signals termed *action potentials* away from the soma. Action potentials initiate in a specialized microdomain at the proximal end of the axon called the *axon initial segment* (AIS), and propagate at speeds ranging from 1 m/s to 100 m/s.

For a brief description of each of these neuronal compartments and its functional roles, please refer to Appendix A.1.

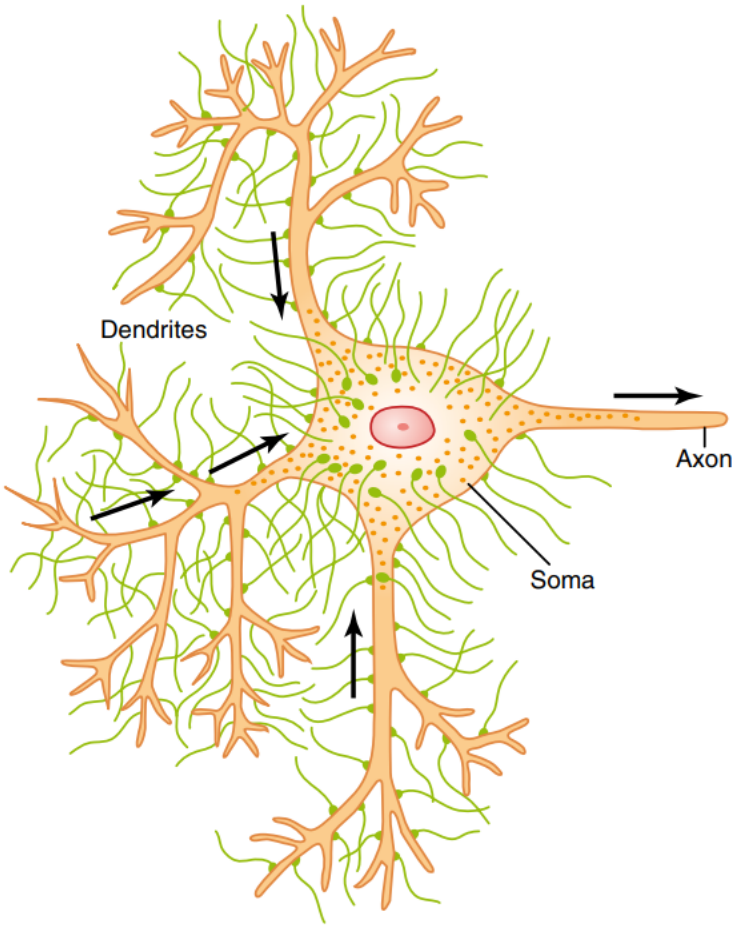


Figure 1.1: Sketch of a neuron and its four regions. In these figure we can see the four regions that compose a neuron: soma, dendrites, axon and synaptic terminals (green). The arrows represent the direction of electrical signalling: dendrites collect signals from other neurons and transmit them to the soma, where they are integrated. The resulting signal – the action potential – is transmitted away from the soma along the axon. Signals are expressed in the form of a variation in the membrane potential that travels along the neurons. Adapted from [5], pg. 547.

1.2 Motivation and challenges of neuronal classification

Different types of cells constitute the basic computational element of the brain, the neuronal ensemble. Having a better understanding of their distribution in the network and their reciprocal interactions would open the door to understanding the functioning of the much more complex machine, the brain itself. In this section, we first attempt to define the term 'cell types' and 'classes' in the context of neurophysiology. We then summarize the motivation for attempting this kind of classification, what techniques are available for such a task, and underline some of the challenges involved.

1.2.1 What are cell types?

Cell types are defined as families of cells that share a specific function, different from the ones of other families of cells [3]. As the function of a single neuron is often difficult to determine, and may only be identifiable in a neuronal circuit, we can define a neuronal type as a population of cells sharing homogeneously some properties, differing from other neurons [3]. The three main categories for these characteristics are morphological, physiological and molecular [6][7]. Morphological features such as dendritic and axonal branching, as well as soma size, can be used to determine morphological types. For physiological properties, firing rate (by firing we mean the action through which a neuron sends a signal to other neurons) and resting membrane potential are used as well as biophysical features. The protein and mRNA composition are the molecular features that are mainly used for the definition of types.

As a simplification of the neuronal zoo, classification also needs to be well organized with a hierarchical structure. This will allow us to analyze the structure of the brain on different scales, focusing on more or less specific, diversified neurons. In addition to this, it also allows for simpler updating of the list of classes and types, and uses relationships between types as a feature itself [3].

In the brain, the hierarchy of neurons goes from classes and subclasses, to types and then to subtypes, all defined according to the feature properties listed above. An example of this hierarchical diversification can be seen in Fig(1.3). The class is defined as a collection of types that share a common feature [4]. The first distinction that we can make is between projection neurons (or principal), which send their axons outside the structure where their soma is localized, and intrinsic neurons (or interneurons), with synapses only in the same structure as the soma.

We can see some examples of neuronal types and classes in the neocortex. In this structure, we have a larger population of principal neurons (80%) than intrinsic ones [4]. The main class of projection neurons in the neocortex is composed by the pyramidal cells. This name is used to describe a variety of principal neurons in all regions of the brain (we also have pyramidal hippocampal neurons), characterized by a large, triangular soma. We can see the morphology of pyramidal cells in the neocortex in Fig(1.4 A). Just below the pyramidal neurons, the same figure also shows some examples of the many diverse types of interneurons in the neocortex (Fig 1.4 B). In this case, we have the additional use of molecular features for the types characterization. It must also be noticed that the term pyramidal cell in cortex and hippocampus is often used as a synonym of excitatory neuron (as pyramidal cells in these structures are mainly excitatory), while interneurons are considered to be mainly inhibitory. In these structures, excitatory neurons are also refered to as **glutamatergic**, while inhibitory ones are **GABAergic** neurons (with a few exceptions) [3]. Another kind of binary diversification is between **excitatory** and **inhibitory** neurons, on which we will focus in this thesis. An example of hierarchical structure for classes and types can be seen in Fig(1.2), in which we have excitatory and inhibitory neurons as the largest classification scale in the neocortex.

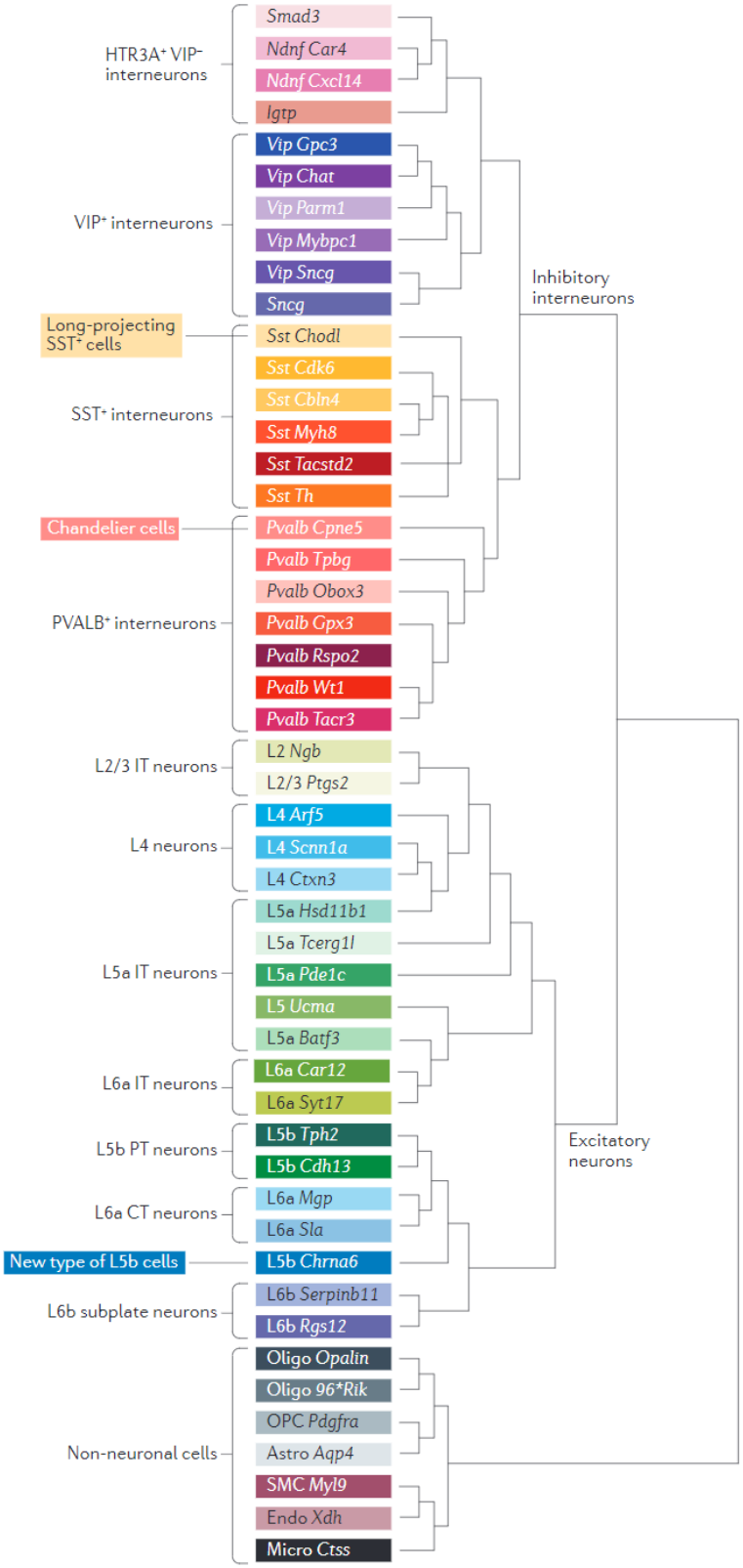


Figure 1.2: List of known types and classes known in the cerebral cortex. The types in the center (colored boxes) were determined via single cell transcriptomics. The types on the left are the corresponding types that were previously identified via morphological, connectivity and neurochemical properties. As we can see, all types belong to –apart for a few non-neuronal ones– either the pyramidal excitatory class (7 previously known types) or inhibitory interneuron class (4 previously known types). Adapted from [3].

A deeper outlook of the variety of types and subtypes of neocortical neurons can also be seen in Fig(1.2), in which we also have a few more interesting examples. Excitatory (glutamatergic) cortical neurons have five known subclasses, being: projecting layer 4 neurons that can be further separated in spiny stellate cells and star pyramidal cells depending on the presence of apical dendrites, corticocortical projection neurons, pyramidal tract neurons, corticothalamic projection neurons, and layer 6b subplate neurons. The four subclasses, each one of which can be subdivided in types, in the GABAergic class are named after their expressed neurochemical marker: somatostatin-expressing (SST+) cells, vasoactive intestinal peptide-expressing (VIP+) cells, parvalbumin expressing (PVALB+) cells and cells that express 5-hydroxytryptamine receptor 3A but lack VIP (HTR3A+VIP-).

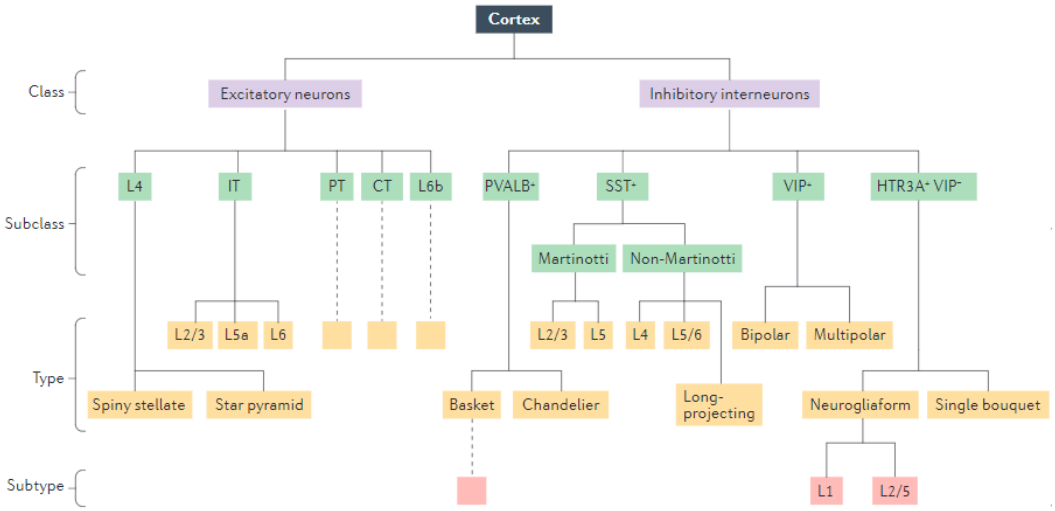


Figure 1.3: Hierarchy of cell classification in the cortex.

1.2.2 Multiple properties may be used to classify neurons

Neuronal classification can be performed via multiple characteristics, each one with its own techniques of acquisition. Here, we briefly list each one and determine the pros and cons of all of them and when they should be used. In doing so, we should keep in mind that the cell-classification efforts should be based on large data sets, so that small variations inside type definitions can be identified as such, and infrequent cell types can also be found. We will eventually see that the technique implemented in this project, extracellular electrophysiology, offers the most advantages among all the available ones.

The first attempt at classification was made by Cajal, using the Golgi stain technique. This way, he differentiated neurons based on their common morphological structure. Some examples of this kind of classification, taken from Cajal's paper, can be seen in Fig(1.4). As we have already said, the main morphological properties that were used are axonal and dendrite shapes and branching, together with spine density and spine shape. An example of how morphology can be used to distinguish between cell classes

and types can also be seen in Fig(1.5.a), in which the cells where imaged from brain slices after staining and show stereotypical shapes for their types [3].

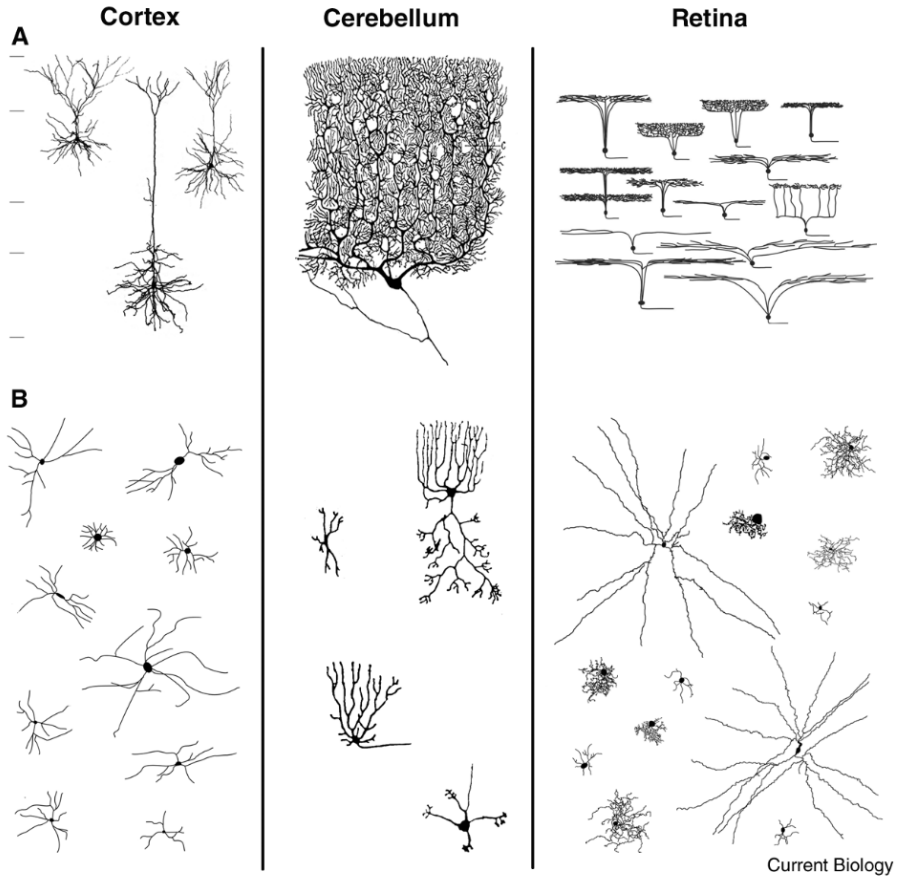


Figure 1.4: Morphologies of neurons in cortex, cerebellum and retina. Types in the cerebellum and retina are much better defined, thanks to their simple architecture. A. are principal neurons. B. are interneurons. Adapted from [4].

The morphology of the cell is directly related to the function of the cell. This means that this kind of classification is incredibly powerful at finding distinctions between cell phenotypes, and can be used to confirm or refute possible classifications from other methods. Morphological classification relies on techniques such as electron and light microscopy. They are still incredibly time-consuming and resource-consuming, which means that they have a low throughput of information. In addition, high-resolution imaging is usually done with high-energy microscopes, which could be detrimental to cell health [8][9].

Another vastly used method for cell classification is the molecular one, based on protein or mRNA profiling of the cells. The new methods that have been developed in this field have greatly improved the sensitivity of this kind of classification. Molecular classification is reliable, determines well defined, objective and quantifiable separations

between types and classes, and yields a high throughput of data. The most advanced technique in this field, single cell RNA sequencing, requires though for the cells to be dissociated and separated completely, which kills the cells in the process and makes it very invasive, disrupting further experiments on the cells. Examples of this kind of classification can be seen in both figure Fig(1.5.c) and Fig(1.2).

The third kind of classification technique is the biophysical one, which collects information about the activity of the cells to determine differences in behavior. These differences, when observed homogeneously in a large enough sample, can then be used to define class- or type-specific properties. The main techniques used for biophysical classification are intracellular recordings, via patch-clamp method, optical imaging with voltage sensitive dyes and extracellular recordings using MEAs. While patch-clamping offers a reliable way of observing the specific behavior of cells, it can work only on one cell at a time, and is invasive. The most cells recorded this way, using the same setup, was 12, very far from the kind of high-throughput analysis we need. Some classification examples based on stimulus-response properties recorded via patch-clamping can be seen in Fig(1.5.b) Optical imaging, be it with voltage or calcium indicators, offers high data throughput, and a direct window into the activity and behavior of the cell. It is still very expensive, requires a lot of resources and has comparably worse temporal resolution than modern extracellular electrophysiology. The use of HD-MEAs offers an incredible amount of data and information, is the least invasive process among all the ones listed (even optical imaging requires the addition of indicators, disrupting the normal activity), acts as a direct access to the function and activity of the cell and allows for the same cells to be measured multiple times, under the same or different conditions, before being disposed. Extracellular electrophysiological recordings have already been used vastly *in vivo* for classification; some examples in this regard are regular spiking, fast spiking, intrinsic burst and chattering neurons in the cat primary visual cortex [10], or the distinction between interneurons and pyramidal cells in the awake macaque cortex [11] and in the rat cortex [12], all of which were mainly based on firing rate, trough-to-peak interval and connectivity analysis from *in vivo* recordings.

As we have seen in this section, there are many techniques and properties that can be used for classifying neurons. Each of these has its pros and cons, but we know what we should look for in a classification technique: reliability, high data throughput, low invasiveness, low time and resource cost, and precision. Thanks to the advances in the field of HD-MEAs, extracellular electrophysiology has the most of these qualities among all the techniques.

1.2.3 What is the purpose of neuronal classification?

There are many reasons for the amount of interest behind neuronal cell-type identification and classification. First, as a tool to simplify the system. The brain is an incredibly complex machine, which for many years has been studied in terms of compartments: different regions of the brain have different functions, which concatenated together make up what we define as *thought*. Every characterization of neurons in types

or classes, with their attached functionalities, reduces this complexity, and makes the chaotic sea of information regarding the brain much more manageable. Thus, we need to produce a *parts list*, in order to consider neurons in terms of types instead of individuals, which would be far too complicated. Second, the study of diseases would also be improved: each disease affects different neurons in specific ways, and being able to discern between them while recording would help in figuring out the disease itself. As an example, amyotrophic lateral sclerosis only affects lower and upper motor neurons [3].

1.2.4 Neuron-type classification has many obstacles

There are many obstacles in general in the cell classification effort. The first one is different terms corresponding to specific levels of hierarchy, like "variety", "class" and "subtype", have been used indiscriminately over the years [4]. A common definition framework is thus necessary. In addition to this, some names have changed connotation during the years: the name pyramidal cell, for example, has changed over the years and is now also used to indicate cells that do not have a triangular soma or the characteristic branching pattern that first suggested the name [4]. Secondly, cell classification is also difficult to be done quantitatively: discontinuous variations along–possibly–multiple features are required to define a good classification metric [3]. This is also necessary not to risk splitting a single type, because of small variations within its population.

Apart from these, the most relevant problem is that in order to implement the best classification possible, we need to find a harmony between all the different techniques. It is often difficult to find a correspondence between classification in the morphological, molecular, and biophysical fields.

When different techniques identify the same classes and types, the classification is more robust and representative of the underlying diversity of cells. Some examples of coherent classification across properties can be seen in Fig(1.5), in which 5 types are shown, which were differentiable by morphology, biophysical properties and molecular signature. Another example are MSN neurons (medium spiny neurons), which are GABAergic and thus inhibitory. These also offer some insight on why multiple properties coming from different techniques are needed: MSN neurons are inhibitory, but spiny neurons are mostly pyramidal and thus excitatory. MSN constitutes a smaller subtype among the inhibitory class that shares morphological similarities with a largely excitatory morphological class. Another interesting example of cross-technique classification is that of parvalbumin expressing fast-spiking interneurons, in which both molecular marking and electrophysiological features are used to define a neuron type (which belongs to the GABAergic and inhibitory class).

Finally, the lack of translation of many classification metrics across model systems presents a considerable challenge. Dissociated neuronal networks *in vitro* cultured on microelectrode array systems are a popular model system to study a generic ensemble of neurons. These systems enable label-free simultaneous long-term recordings from populations of neurons at high temporal resolution. Additionally, the complementary

metal-oxide-semiconductor (CMOS) based high-density MEA (HD-MEA) systems developed in the BEL lab offer unprecedented spatial resolution that enables the mapping of neuronal activity ranging from subcellular compartments through individual neurons to entire networks [13][14].

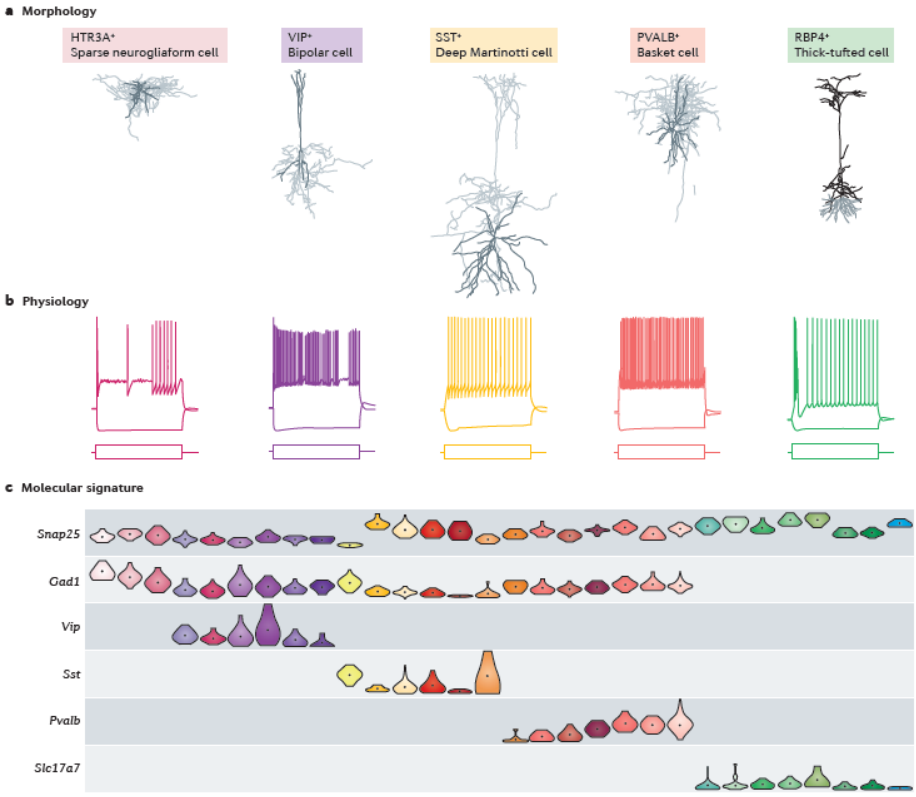


Figure 1.5: Neuron types classified via different properties. Neurons can be classified based on morphological, molecular and physiological features. Adapted from [3]. In the figure, we see five examples of cortical neurons that show characterizing differences through all of the different kind of features.

However, many of the classification metrics that offer reliable neuron class separability in *in vivo* animal models do not translate well to the *in vitro* culture context. Clear statistical parameters and feature combinations from these rich data sets that enable reliable distinction between even broad functional neuronal classes (e.g., excitatory vs. inhibitory neurons) are still lacking. Fast and high-throughput data-based neuron type inference would be invaluable as a scientific tool to study neuronal information processing and in applications, e.g., in functional biomarking of neurological disease models, drug screening and neurotoxicity studies.

The next section gives an overview of the main functional classes of neurons: excitatory and inhibitory neurons, and previous studies that have attempted to identify them in multiple model systems.

1.3 Distinguishing GABAergic inhibitory from glutamatergic excitatory neurons

The two broad functional neuron classes of the nervous system are glutamatergic excitatory neurons and GABAergic inhibitory neurons. The ability to identify these neuron classes would enable to study the distinct contribution of each subpopulation to the underlying computation of neuronal circuits.

Several synaptic neurotransmitters can act as activators or inhibitors on postsynaptic cells. Glutamate is the main excitatory transmitter in the brain, acting as an activator for channels in the postsynaptic neuron with a similar permeability for both K^+ and Na^+: through this process, it induces a depolarization of the postsynaptic membrane, which generates an excitatory postsynaptic potential (EPSP). $\gamma-$aminobutyric acid (GABA), on the other hand, binds to GABA-A and GABA-B receptors: the first results in an increase in the Cl^- influx, while the second results in a decrease in the Ca^{2+} influx and an increase in the K^+ efflux. This causes a hyperpolarization of the postsynaptic membrane, generating an inhibitory postsynaptic potential (IPSP). At the same time the binding of the GABA-B receptor leads to a K^+ efflux also in the presynaptic neurons, thereby inhibiting its ability to release neurotransmitters, in particular glutamate [15].

There are multiple studies that have reported reliable separability between neurons of these classes based on specific features observed in extracellular electrophysiological recordings *in vivo* [16][12][2][11][17][10]. This studies used extracellular features tied to both the spike shapes generated by each neuron and the corresponding time-series *in vivo* [12]. However, many of these metrics turn out to be unreliable indicators in *in vitro* models [18][19] and there is still no certainty about which features could enable a similar classification as in *in vivo*. Multiple studies have been investigated in this direction, some of which claimed to have found promising features [18], but direct confirmation has remained elusive and many of these studies have been refuted in later years [20].

Moreover, these studies were performed using low density microelectrode arrays. The low spatial density of these arrays provide less information about the underlying structure of the circuit and the neuron themselves. They also did not implement any advanced spike sorting technique in their analysis, but rather a less precise visual method, relying on low plating density and large pitch between electrodes to separate putative neurons. This in turn provided them with low precision in the distinction between units (where by unit we mean a putative neuron, a possible neuron identified from electrophysiological features and not via imaging), and thus their properties, on whose reliability the classification is fully based on. Due to the low spatial resolution of their arrays they could not design a series of multichannel unit properties, as their whole sorting technique was based on neurons only being detected from one electrode. We will report how multichannel features will prove to be vital for the efficiency of the classification. Their approach of classification was to find a separation between classes in one or two-dimensional feature distributions: no attempt was made to observe the

high dimensional structure of the data in two dimensions. Nor did they use any kind of machine learning based model, either supervised or unsupervised, or attempted drafting a labeled ground truth data set, which could then made available in the public domain. In this project, we aimed at improving all of these points, in which we identify possible reasons to the failure in classifying neurons *in vitro* up until now.

1.4 High density MEAs for extracellular electropyhsiology

The functioning of the brain cannot be directly deduced by the behavior of the single neurons: it is determined by the way these cells interact in a network, processing information as a series of complex machines interconnected with each other rather than the single gears. The best way to access information about this pathways, and the interactions between the neurons, is to observe their electrophysiological behavior. Neurons send and receive information in the form of an inflection of their membrane potential, with respect to the resting potential. This inflection, and the travelling signal produced, is called **action potential** (AP), and is discussed more in detail, along with the *resting potential*, in Appendix A.2, in which also existing models for this two properties are briefly seen. As we have anticipated in the previous section (1.2), to be able to classify neurons efficiently we must be able to collect data and information on the number scale of a whole network. For this, and for all the reasons listed in section 1.2, we chose to measure electrophysiological features from the extracellular environment, using the relatively new technology known as HD-MEAs.

Recordings of the AP are usually done either through the patch-clamp technique, or using HD-MEAs. Although the patch-clamp technique allows the measurement of intracellular voltages and transmembrane currents of the neuron at high fidelity, it is invasive and can only be applied to a small set of neurons at the same time.

For this reason, in later years, the use of HD-MEAs has become more and more prevalent in neuroscientific research. This technology enables long-term access to the whole network at the same time without damaging cells. It consists of tens of thousands of electrodes, regularly spaced on a small surface, that can record variations in electrical potentials in the conducting medium at high spatiotemporal resolution. By using these arrays to record neuronal networks, we are thus able to record the behavior of the whole network with minimal loss. This gives us a lens through which to observe the interactions at the origin of information processing. While MEAs have been in use in the field since 1970, the introduction of complementary-metal-oxide-semiconductor (CMOS) technology has enhanced greatly the resolution, both spatially and temporally, also increasing the sensing area and making them one of the protagonists in modern-era neuroscience. Modern CMOS-technology-based MEAs are widely used in this field, featuring thin metal electrodes arranged in an array with a pitch $< 30\mu m$. Even though these chips can be used both for recording and for stimulating neurons, we mainly focused on using them for measuring the extracellular potential, which is a direct con-

sequence of the action potential generated at the AIS and propagating through the axons.

1.5 Aim and structure of the thesis

One of the current main objectives of neuroscience is understanding the principles of information processing. The brain is a complex machine, and so are the neuronal circuits that compose it. As for any other machine, the first step to disentangle its functioning is to reduce its complexity, and this endeavor starts with the creation of a functional parts list of its smallest components. There is thus an urgent need to explore reliable data-based indicators of functional neuronal classes.

The aim of this master thesis was to determine a procedure for classification between excitatory and inhibitory neurons, based solely on their electrophysiological behavior recorded via HD-MEAs. The possibility of being able to discern between the two main neuronal classes through high throughput *in vitro* recordings would offer many advantages to the neuroscientific field, as described in 1.4 and 2.1.1.

In order to give an unambiguous proof that extracellular spikes can be used to define a meaningful diversification between the two main classes of neurons, the thesis included the following intermediate goals:

- Development of a full pipeline, going from the raw recordings through a spike-sorting method (Kilosort2.5 [21]), curation of the identified putative neurons (or units) and extraction of waveform and time-series related features and returning the high dimensional feature space which we designed as the core of the project.
- Creation of a labeled, ground truth data set, to check the efficiency of the classification and infer physiological information from it. Two procedures were developed to collect this labeled data set: one based on immunocytochemistry staining and a full imaging pipeline, the other on neuronal connectivity inference and spike transmission probability kindly provided by Dr. Julian Bartram [22].
- Development of a machine learning procedure to classify the neurons with respect to the collected data, based on either visual confirmation of a separation between data points (via also the use of dimensionality reduction machines), or a series of supervised classifiers, both linear and nonlinear, trained and tested on the labeled data sets.

Chapter 2

Materials and methods

In the following section, we describe the tools and techniques implemented in the project. First we focus on the cell cultures, why we chose *in vitro models* for the experiments, and how exactly the cultured networks were prepared. Then we describe the HD-MEA, and how these were used to record electrophysiological activity. The discussion moves then to spike sorting, how it detects putative neurons (to which we often refer to also as **unit**), and the electrophysiological features that we extracted for each one of these units. The immunocytochemistry-based imaging pipeline is considered after this, as well as the two procedures we established to obtain a labelled ground truth data set of units. Eventually, we also talk about the machine learning techniques implemented, both in terms of dimensionality reduction and supervised classifiers.

2.1 Primary dissociated rat hippocampal cultures

All of our recordings were performed *in vitro*, where we recorded the electrophysiological activity of primary dissociated rat hippocampal cultures. In the following, an overview of the reasons behind the choice of *in vitro* models, and the procedure for extracting and culturing our cells.

2.1.1 The choice of *in vitro* model system

There are numerous advantages to working *in vitro* with neurons, given that a robust recapitulation of the physiological state is achieved. It offers greater control over experimental conditions, more reliable repetitiveness of experiments, being able to affect via chemical agents the same population multiple times to observe different combinations of effects. The kind of stability that *in vitro* setups offer also gives us a chance for more delicate and dense measurements [13][14], like the ones needed for feature-wise classification necessary to break down information processing in the brain. The use of *in vitro* models offers a series of advantages [23] with respect to procedures performed *in vivo*:

- Environment control: temperature, pH level, nutrients, humidity, everything is under the direct influence and control of the researcher.
- Replicability: much easier to replicate under same conditions than *in vivo* studies.
- Reduction in variability: this one is particularly important. By working *in vivo*, we greatly reduce the complexity of the system itself. The innate variability of *in vivo* systems, which are affected by multiple factors not always under our control, like the stress levels of the animal, is partially removed *in vitro*. In *in vitro* systems, most of the factors that affect fluctuations in the system are under our control, or can anyway be observed and quantitatively accounted for.
- Time requirement: *in vitro* cultures are usually measurable in a matter of days or weeks, and can be measured multiple times.
- Quantity of consumables: the required quantity for reagents is much smaller than for an *in vivo* experiment.
- Reduction of animal use: the stress exerted on living animals is greatly reduced.

Obviously, the use of *in vitro* models also comes at a price, with some disadvantages with respect to studies done *in vivo*:

- Absence of periphery: many processes are determined by signals and influences coming from the periphery of the body.
- Difficult handling: *in vitro* systems are more fragile than *in vivo* ones, and require a lot of knowledge about handling and sterile working conditions, as they do not have innate systems to protect them from bacteria, fungi or viruses.
- Loss of structure: some structures, in particular mostly 3-dimensional ones, are usually lost when working *in vitro*.

The choice between the use of *in vivo* or *in vitro* systems has to be made depending on the specific study, weighing both pros and cons of both techniques to determine which one is going to serve better the objective of the researcher. For our study, we determined the pros of measurements done *in vitro* to far outweigh the cons. It was particularly relevant, in our choice, the possibility of using HD-MEAs for recordings, as they gave us exactly what we needed for the success of the project: long-term and simultaneous recordings, with very high throughput, and high spatiotemporal resolution [13].

2.1.2 Cell extraction and dissociation

The experimental protocols involving animal tissue harvesting were approved by the veterinary office of the Canton Basel-Stadt according to Swiss federal laws on animal welfare and were carried out in accordance with the approved guidelines. The hippocampi of E-18 Wistar rat embryos were extracted in ice cold HBBS (Gibco), then dissociated in trypsin with 0.25% EDTA (Gibco).

2.1.3 Cell plating

After sterilization, and just before cell application, 8 µL of 0.02 mg/mL laminin (Sigma-Aldrich) in Neurobasal medium (Gibco, Thermo Fisher Scientific) were added to the HD-MEAs, to improve and support growth of the cells and differentiation, and were incubated this way for 30 minutes at $37°C$. The chips were then seeded with populations of 10000, 20000 or 30000 cells over the sensing area of the array $\sim 8\,mm^2$, and incubated for 30 minutes at 37 °C before the addition of 1.5 mL of plating medium, whose stock consists of 450 mL Neurobasal (Gibco), 50 mL horse serum (HyClone, 1.25 mL Glutamax(Invitrogen), and 10 mL B-27 (Invitrogen). After ≈72 h we substituted half of the plating medium with growth medium, for which the stock solution consists of 450 mL D-MEM (Invitrogen), 50 mL horse serum(HyClone), 1.25 mL Glutamax (Invitrogen), and 5 mL sodium pyruvate (Invitrogen. The media change procedure was then repeated twice a week, at intervals of ∼96 h. All chips were covered with a lid and kept in closed Petri dishes, equipped with smaller Petri dishes filled with water to help maintain the necessary humidity, and kept in an incubator at 37 °C, 5% CO_2 and 20% O_2.

2.2 High density MEAs

In this project we recorded the electrophysiological behavior of neuronal cultures *in vitro* via an HD-MEA called MEA1k, featuring 26400 bidirectional electrodes, with an area of $5 \times 9\,\mu m^2$, and arranged on a $3.85 \times 2.10\,mm^2$ array with a pitch of 17.5 µm. Printed circuit boards (PCB) were connected through gold wires to the chips, while epoxy (Epo-Tek 353ND, 35ND-T, Epoxy Technology Inc., Billerica, MA, USA) was used to protect the wires from saline solutions like the culture medium. The process of adding and curing of epoxy to the chip is called *packaging*, and is done in order to protect the exposed wires from water and medium after the plating. A fully packaged chip can be seen in Fig(2.1).

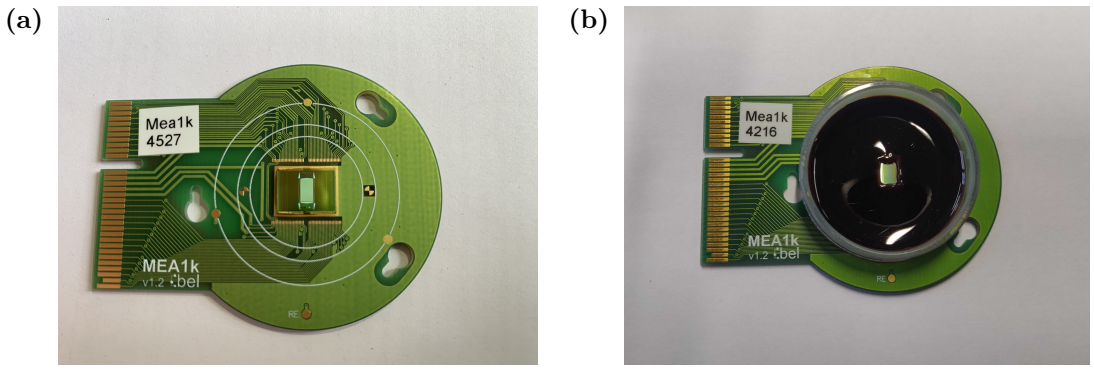


Figure 2.1: The CMOS high-density microelectrode array (HD-MEA), used in the project. (a) An HD-MEA before packaging. (b) A packaged HD-MEA: the dark circle is the epoxy, protecting the wires and contained by a plastic ring, while the rectangle in the middle is the sensing area of the electrode array.

The high spatial resolution of the array allowed us to reliably localize the neurons. The chip features the possibility of configuring 1024 readout channels, which can be used to record simultaneously with different possible configurations over the sensing area. Sampling is done at a 20 kHz frequency, with a power consumption of 75 mW (the readout noise was typically around $2.4\mu V_{rms}$ between 300 Hz and 10 kHz, and $5.4\,\mu V_{rms}$ between 1 Hz and 300 Hz). Before the actual cell plating chips were also sterilized by immersion in 70% ethanol for 40 min to an hour, after which they were rinsed with deionized water (DIW) 3 times.

2.2.1 Platinum black deposition

A platinum black (Pt-black) layer was deposited over the electrodes before sterilization. This was done to reduce the impedance of the electrode by increasing the sensing surface, thus improving the signal-to-noise ratio (SNR) of the recorded signal. We covered the HD-MEA array surface with a solution of chloroplatinic acid hexahydrate 7 mM, Sigma-Aldrich, St. Louis, MO, USA) and lead acetate (0.3 mM, Honeywell, Morris Plains, NJ, USA) in DIW, then we applied a 550 µA current to the array electrodes, while a Pt reference electrode was immersed in the solution, and gently scrubbed the sensing surface with a cotton-swab (or pipetted in and out the solution) to improve the adherence of Pt-black to the electrodes. The entire procedure lasted ≈1 min per HD-MEA, and once completed, we examined the chips under a microscope to assess the state of the deposition.

2.2.2 Recording

Recordings were performed using the MaxLab Software (MaxWell Biosystems, Zurich). While being recorded, the chips were kept inside incubators at 37 °C, 5% CO_2 and 20% O_2. Recordings were done via the 'Activity Scan' mode (*not* spikes only), at a sampling rate of 20 kHz and with configurations chosen to cover the entire electrode array. We used one of two configurations for the recordings: the first, was composed of 7 sparse steps (one out of two channels were selected, both column and row wise), the first of which covered in part the left side of the array and the right, as can be seen in Fig(2.2); the second was a custom configuration (7C), in which the first 6 steps were the same as for the 7X (but starting from the border of the array), and the last was set so that the routed electrodes only kept a space between two consecutive ones in the vertical direction. All steps were designed to route exactly 1020 electrodes each.

2.3 Spike sorting

The action potential is seen in the recording as a temporal and local deflection of the voltage, which takes the name of spike waveform. When multiple healthy neurons are close to the same electrode, as in our case, their spikes overlap and get entangled, making it difficult, together with consistent background noise, to distinguish spikes coming from different neurons. Spike sorting is a term used to describe the process

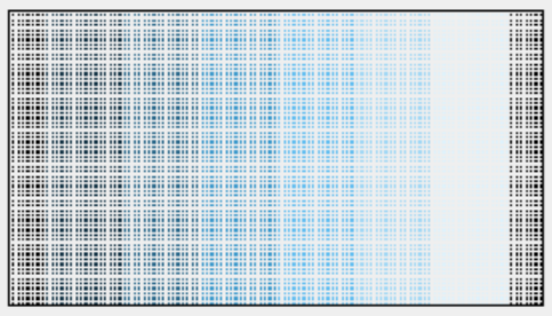


Figure 2.2: Configurations of the recording in the default 7 sparse setting. The image is a representation of the 7 different, consecutive configurations that composed a 7 sparse recording. The 7 configurations are characterized by a different coloring, for the electrodes that are activated respectively. After the first configuration (in black, half on the left and half on the right side) was done, the electrodes in the second configuration (dark blue) were routed and recorded, and so on until the last configuration (in white).

of identifying and assigning each spike to the correct neuron, thus returning what's defined as **spike train** (the time series of the spiking events for the specific unit during the recording), based on the features of the recorded waveforms. As a spike sorting machine we used Kilosort2.5 [21], a scalable and fast algorithm. Kilosort2.5 takes also in account the fact that our recordings are collected through electrodes that are closely spaced, which can record the same spike at different positions on the array, giving rise to specific spatial shapes of our waveforms, determined by the electrophysiological characteristics of the neuron and by its position relative to the electrodes. Spatial and temporal information are then used together, to assign spikes to putative neurons, or *units*. The data pre-processing and major computational steps involved in spike sorting are briefly outlined in the following subsections.

We take also this chance to clarify the distinction between **unit** (or putative neuron) and neuron: a unit is the output of a spike-sorting procedure, a putative neuron to which a series of spikes were attributed based solely on electrophysiological recordings. By neuron, instead, we meant the real cell, which can be observed via imaging. In order to be certain that our units actually correspond to neuron, we need to correlate with imaged cells, as we did in this study.

2.3.1 Pre-processing

The signals were filtered first with a high-pass filter at 300 Hz, to remove low-frequency fluctuations, then, from each sample the median of the signal across all channels was subtracted (common average referencing) to remove artifacts shared across recording sites. The data was then whitened across channels to remove correlated noise.

During the recording, some channels could be too far from any neuron to pick up meaningful waveforms and have a very low signal-to-noise ratio. In order to make the spike sorting procedure more reliable and ease the computational load, a masking procedure for these channels must be developed. In Kilosort2.5, this process was based

on the following observation: any mean spike waveform, (averaged over spike waveforms measured at different times on the same channel), can be described well by an SVD over the spatio-temporal waveform, with variable number of spatial or temporal components. Each spike was then represented by "private PCs".

2.3.2 Template matching

At the basis of the template matching framework was a generative model for the electrical voltage, based on the fact that electrical potentials coming from different sources sum approximately linearly in the extracellular medium. Each spike k is assumed to belong to one of N clusters, representing the number of neurons, and its cluster identity is defined as $\sigma(k)$. Each neuron n has a corresponding normalized waveform, described as the matric K_n, with dimensions equal to the number of routed channels by the number of time samples for each spike.

K_n is then approximated through a three dimensional SVD of the kind $K_n = U_n W_n$, so that K_n is deconstructed in three pairs of spatial and temporal functions U_n and W_n, with norm 1. The voltage at time t measured by channel i is modeled as:

$$V(i,t) = V_0(i,t) + \mathcal{N}(0,\epsilon) \tag{2.1}$$

where $\mathcal{N}(0,\epsilon)$ is a Gaussian noise with variance ϵ. We then have:

$$V_0(i,t) = \sum_{k,s(k)<t}^{s(k)\geq t-t_s} x_k K_{\sigma(k)}(i, t-s(k)) \tag{2.2}$$

where index k indicates spikes that overlap at timepoint t and x_k is the amplitude of spike k:

$$x_k \sim \mathcal{N}\left(\mu_{\sigma(k)}, \lambda\mu^2_{\sigma(k)}\right) \tag{2.3}$$

This last expression takes into account variability in the amplitude of spikes originating from the same neuron due to many physiological and structural reasons, such as adaptation and attenuation during a burst and drift of the neurons. Having a variance that scales with the square of the mean represents the fact that spikes coming from neurons closer to the channel vary in relative amplitude, not absolute. ϵ and λ are scaling hyperparameters.

From this model, the following cost function was defined:

$$\mathcal{L}(\mathrm{s},\mathrm{x},K,\sigma) = \|V - V_0\|^2 + \frac{\epsilon}{\lambda}\sum_k \left(\frac{x_k}{\mu_{\sigma k}} - 1\right)^2 \tag{2.4}$$

and was minimized with respect to spike times s, amplitudes x, cluster assignment σ and templates K. The sum on the right of this expression limited the number of spikes to be assigned amplitudes strongly deviating from the average of the cluster, scaled by $\frac{\epsilon}{\lambda} \in [1, 10]$.

2.3.3 Cost minimization step

After the templates were initialized, optimization was performed by alternating between a template matching step and an optimization of the template waveforms. Eventually, similar clusters of spikes get merged. It is known that the temporal density of spikes across the electrode array varies vastly, due to the locations of the neurons with respect to the channels. The best strategy to initialize the templates is to do it in such a way that finds a match between the number of initial clusters to the local spike density across the array. An initial detection of the spikes is done via an amplitude threshold, and each of these spikes is compared to a subset of prototypical spikes built progressively while going through the recording, with the only constraint that these initial prototypes have to be different enough. Of these prototypes, only the best N are kept, depending on how many matches they had with spikes in the recording. Then the spike prototypes were used for the initialization of a K-means clustering machine, scaled so that it does not rely on spike amplitudes (which might bring errors, as they depend on distances relative to the electrodes). The K-means algorithm aims at minimizing equation (2.3.2), in which the spike times are fixed, and the ones found through the amplitude detection.

These clusters were used for the initialization of the first set of templates K_n, which were considered to be fixed, along with the corresponding mean amplitudes, for the step in which the algorithm tries to find the optimal spike times, cluster assignment of said spikes and amplitudes. As said previously, the templates were obtained from the average waveform A_m through an SVD decomposition, and had a low-rank decomposition of the kind:

$$A_n \sim \mu_n K_n = \mu_n U_n W_n$$

In which $\|U_n W_n\| = 1$, U_n orthonormal and W_n orthogonal.

Spike times were optimized through a template matching machine: given template n, it first looked for times where the dot product between the raw data and the template waveform was large, and the amplitude of the spike was close to μ_n. Among all these dot products, only the local maxima were kept, and a sample window was imposed around each spike time, in which no other peak could be detected (to take into account the refractory period). As a final step, a *matching pursuit algorithm* was used to find overlapping spikes.

At each iteration, the spike times and the template waveforms K were re-estimated. The spike times were computed via (2.3.2), and the average waveform A_n for the cluster

n was updated using the following algorithm:

$$A_n^{\text{new}}\ (i,t_0) \leftarrow e^{-j_n/\eta} A_n^{\text{old}}\ (i,t_0) + \left(1 - e^{-j_n/\eta}\right) \sum_{k \in \text{ batch}}^{\sigma(k)=n} V\left(i, s(k) + t_0\right) \tag{2.5}$$

where j_n is the number of spikes that have been assigned to this cluster, and the averaging is weighted with respect to the old samples with an exponential, in which η is a memory constant. So A_n can be seen as an average waveform for the old samples. Template matching and the inference step were alternated until the cost function reached a plateau, after which, overlapping spikes were dealt with.

2.4 Feature extraction

For each recording, multiple features were computed. One of the main contributions of the project was the design of an exhaustive feature space. Many papers that tried to deal with the same task mainly focused on a single feature modality, either based on the time-series information of their network or on the waveform features, which are known to work well *in vivo*. In particular, Becchetti et al. [18] focused on time-series features because they surmised that waveform features were highly unreliable, being heavily dependent on the distance between the neuron and the recording electrode. Recent papers determined that this dependence reflects only on the amplitude of the waveform and not on its shape, so that dealing with normalized waveforms would be enough to be able to also use the waveform features, as Weir et al. did [19]. We thus decided that we would try to integrate as many features as possible, over different granularity and spatiotemporal scales, making use of dimensionality reduction machines to also interact and analyse higher-dimensional feature spaces, which might contain more information about the distribution of our neurons and about a possible threshold between different classes. Our feature space was organized as follows:

- Waveform features:
  - Cellular scale - measures averaged over multiple channels, and often represented as arrays instead of scalar values. We referred to them as *multichannel features*.
  - Subcellular scale - measures averaged over a single channel, what we called the best channel. We referred to them as *single channel features*. All single channel features can also computed over different channels to make a multichannel feature, given a coherent ordering of the channels.
- Time-series features:
  - Long scale - in the scale of minutes.
  - Short scale - in the scale of milliseconds.
  - Intermediate scale - in the scale of seconds, such as burst features.

- Curation features - features used to quantitatively assess the quality of the spike sorted units.

In the following sections, we describe how the individual features were computed and the constraints we imposed (if any).

2.4.1 Filtering

In order to compute many of the features, in particular those related to the waveform of the spikes, we had to filter the signal to remove the background noise. We used an *infinite impulse response* bandpass filter, with a lower frequency limit of 300 Hz and a higher frequency limit of 7 kHz. In our project, we chose a fifth-order filter, and applied it to individual cutouts of the recording.

2.4.2 Curation features

Before computing the features, the data-set had to be cleansed of all outlier units, which could either correspond to artifacts of the sorting procedure (such as multiple units which actually correspond to just one neuron, and vice-versa) or to non-healthy neurons. The outlier detection was performed by applying sequentially a series of filters, based on:

- Inter-spike interval (ISI) violation rate
- Average firing rate
- Maximum amplitude
- Signal-to-noise ratio

We first eliminated all units with an amplitude lower than 30 µV in their best channels. The **best channel** of a unit was defined as the one with the largest amplitude among the channels that comprised its spatial footprint. This first automatic curation was then followed by another much stricter one, based on the observation of the curation parameter distributions and the conserved units' behavior and described in detail in 3.1.1.

2.4.3 Waveform features

In order to get into the merit of the single waveform features, first we have to discuss how the waveforms themselves are computed. For each unit, or putative neuron, detected by Kilosort2.5, the so-called *best channel* was defined.

Among the files output by Kilosort2.5, `Templates.npy` contains the templates computed by the spike sorter for each unit: each template corresponded to a series of optimized and smoothed waveforms, one for each active channel in the recording. It was better not to use these templates for the evaluation of the waveform features: they were optimized through an unsupervised, recursive process to which we had no access,

and were deliberately created to be "vague" in their form, as they were then used in a template matching process for which they could not work if they were too specific and peculiar (as described in 2.3). The templates still retained most of the information about the average amplitude of the spikes recorded on that channel for that specific unit, so we could use them to look for the channel over which the largest template for the unit, meaning the template with the deepest dip, was computed. The largest spikes are usually detectable around the AIS of the neuron, which in 80% of the cases is close to the *soma* itself, so we assumed that the channel that recorded the largest spikes was directly under or in the vicinity of the main body of the neuron: this was what we called the **best channel**.

Once we identified the best channel for each unit, we extracted a cutout: we took the spiking times of a hundred random spikes assigned to the specific unit, and cut out of the signal recorded by the best channel 80 samples (corresponding to 4 ms for a 20 kHz sampling rate) around these spiking times. These cutouts were then filtered as described before, and averaged, to produce an average waveform for the unit. Most of the waveform features that we will describe later on were computed from these mean cutouts, which from now on will be referred to as **footprint waveforms**. Before proceeding, a bit more details about the spiking times chosen to compute the waveforms must be given. As will be shown later in the thesis, spikes during a burst tended to attenuate in amplitude, so that picking one of the later spikes in a burst could bring to an error in the averaged waveform, as we would have had to then take into account also the law by which they attenuate; to avoid doing this, the hundred random spikes were chosen only from a smaller subset containing isolated ones and spikes that were either first or second in a burst, with priority for the first ones (the second ones were included only if there were not enough spikes in the other two families, which rarely happened).

After computing footprint waveforms for all units, we compared them with the template waveform on the corresponding channels. In Fig(2.3) we show one such comparison, considered over the best channel for the unit.

While there seemed to be some agreement in the general form of the trough, there were quite a few differences, which could be explained. First, we observed that the dip was deeper in the footprint waveform than the template: this could be attributed to the filtering procedure that Kilosort2.5 involves in its computation, which could cause it to lose some of the amplitude of the spikes (their filtering procedure was stricter than ours, as Kilosort2.5 filtered between 350 Hz and 3.5 kHz, which was the expected range of frequencies for our spikes). As the filtering procedure corresponded to basically removing from the signal the Fourier components corresponding to frequencies out of the range, keeping a stricter range meant removing more components and, thus, attenuating the average amplitude of the signal. We also noticed that the form of the wave was different both before and after the depolarization step (corresponding to the dip in the potential): the template was smoother in these intervals, while the footprint waveform was more jagged. While the footprint waveform was the simple product of an averaging between different spikes, the template was an extremely optimized and unsupervised version of the waveform, which Kilosort2.5 used in a series of template matching steps:

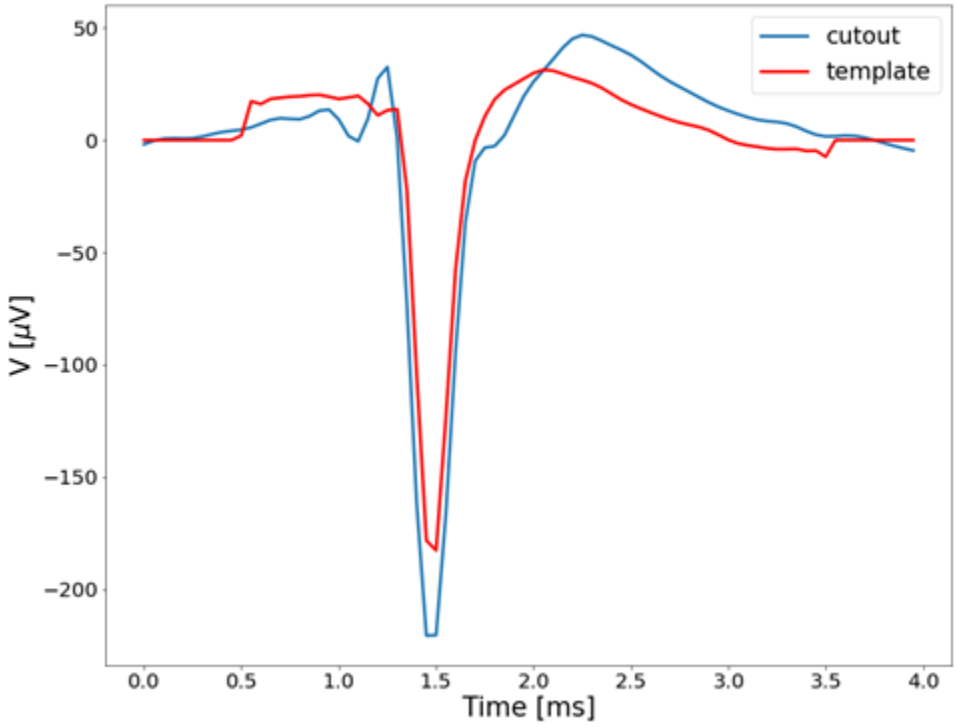


Figure 2.3: Comparison between a footprint waveform and the template over the best channel, for an example unit. The template waveform was unwhitened and rescaled to convert it to the appropriate units. While there is good agreement between the dips, the shapes differ quite a lot, especially before and after the depolarization step.

thus the template had to maintain a smoother and more "vague" look, in order for the algorithm to be able to find spikes which were 'similar' to it, which it could not do if the template were to be specific as the footprint waveform was. For these reasons, we preferred to use the footprint waveforms to compute the features, rather than the templates waveforms.

Another point of interest is then the **footprint**: for each unit, the footprint was defined as the averaged cutouts over the best channel and the set of nearest channels to the best one. The footprint gives insight about the morphology of the neuron, and about the reliability of putative neurons (Fig(2.3)). As we can see in this case, the unit should be considered to be not very reliable: having such small cutouts over channels that are directly adjacent to the best channels is very unlikely, as these should still be recording the same spikes as the best channel. Once we have the footprints for all our putative units, we can go ahead and compute waveform features.

Half-height width

The half height width was computed as the interval, in milliseconds, between the two sides of the depolarization spike at the half-height of the trough, as can be seen in Fig(2.4). The fact that our waveforms were actually discrete series of samples could present a problem in this instance: if we were to just choose the two samples nearest to the actual half-height on either sides to compute the distance, we would incur in possibly large errors. In order to avoid this approximation, we instead took the two samples on each sides for which the half-height fell in-between and performed a polynomial fit, taking into account these samples and the ones immediately before and after (Fig(2.4)). Once the fit was performed, we took the time coordinates of the nearest points generated by the fit as our times of reference to compute the half-height width, thus largely reducing the error (by two orders of magnitude, as the fit was operated over a linear space composed of a hundred points).

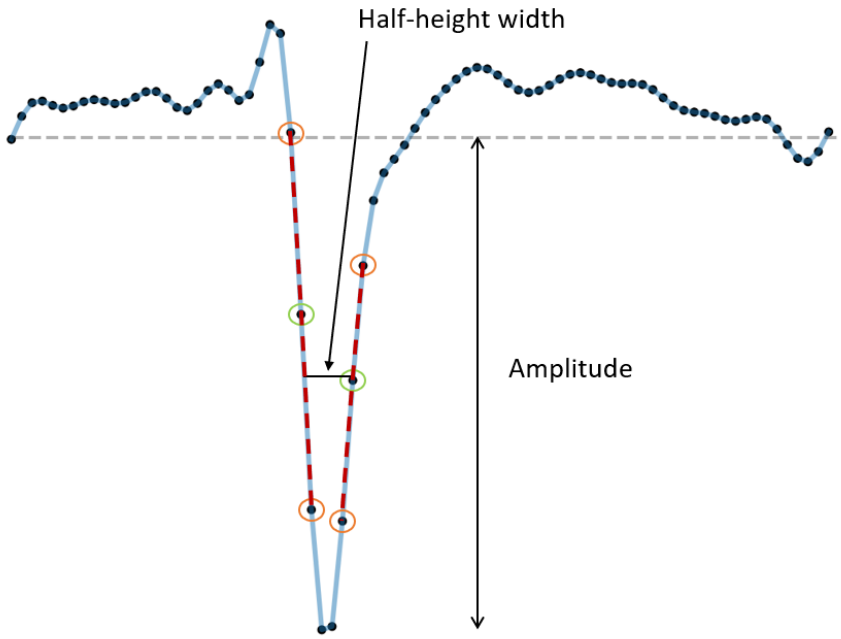


Figure 2.4: Visualization of the procedure to compute the half-height width. Once the two samples closest to the half amplitude were found (circled in green in the figure), a polynomial fit was applied on each slope (red dashed lines), using the above samples and the two closest ones (circled in orange). The half-height width was then determined by finding the distance between the two closest points in the respective fits.

Amplitude

The amplitude was the only waveform feature for which we used the template waveform returned by the spike-sorting procedure to compute. While the actual form of the template should not be used for characterization, as it is optimized via a "black

box" procedure, the amplitude of the template could be used (after unwhitening and appropriate rescaling), as it did not depend as much on the machine learning procedure operated by Kilosort2.5. The amplitude, however, could not be used as a single-channel feature (this distinction will be more clear once the concept of multichannel feature has been explained, in 2.4.6) for classification: spike amplitude was directly correlated with the distance between the neuron and the recording electrode, as it decayed exponentially with it [19]. It was difficult to account for this correlation without knowing the exact relative position of neurons and recording spikes, which would require us to have images with a much higher resolution and reliability than we can, together with very isolated neurons. Thus, we did not deem this measure to be reliable enough to be used in the classification analysis. As the amplitude of the templates determined by Kilosort2.5 was closer to the amplitude of the actual spikes (due to our filtering excluding more modes), we could use it as a curation feature, to remove the units with small templates over the best channel, which were likely to be artifacts.

Trough to peak distance

The trough to peak distance was one of the main features used to characterize the difference from extracellular recordings between excitatory and inhibitory neurons *in vivo* [24]. In order to compute it, we took for each unit the footprint waveform over the best channel and computed the distance between the trough of the spike and the following peak, as shown in Fig(2.5).

Depolarization and repolarization slopes

Depolarization and repolarization slopes were computed from the cutout over the best channel. For each unit, we took the midpoint between trough and peak of the waveform after the trough (the "half height" of the spike). We then computed the slopes using the tangent curve to the waveform at the midpoint before and after the trough, as shown in Fig(2.5).

2.4.4 Time-series features

The computation of time-series features was based on the*spike times* output by Kilosort2.5, which returned all the times corresponding to recorded spikes and the respective unit to which they were attributed.

Firing rate

The firing rate was computed as the total number of spikes attributed to a unit divided by the total duration of the recording:

$$FR = \frac{N_{spikes}}{T_{recording}} \tag{2.6}$$

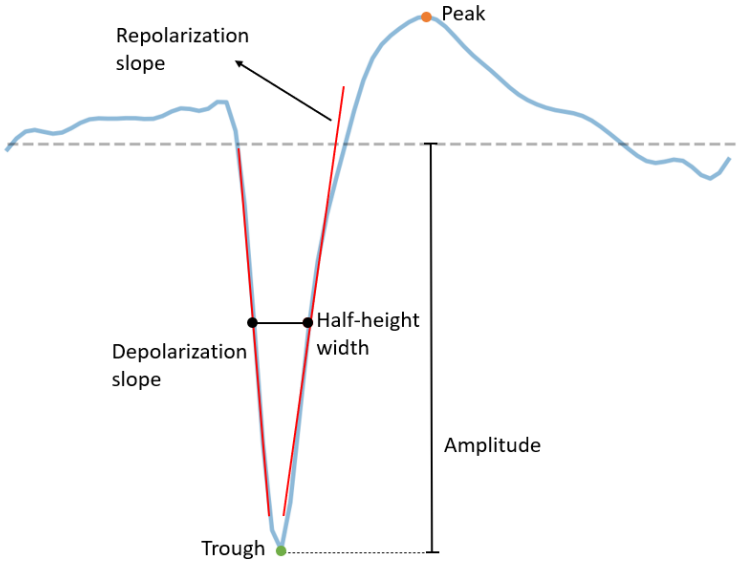


Figure 2.5: Image showing what the different waveform features computed are. As we can see both depolarization and repolarization slopes were calculated around the half-height of the trough, same as the width of the spike, and the amplitude. The trough-to-peak distance was just defined as the time interval between trough and peak.

Autocorrelation

The spike train autocorrelation is often used to determine the behavior of a neuron. The spike-train for one unit was separated into bins with a bin width equal to 0.5 ms; this binned array was then convolved with itself with a lag of 50 ms, which eventually resulted in the autocorrelogram (ACG) showed, for unit 1 in the STP labelled data set:

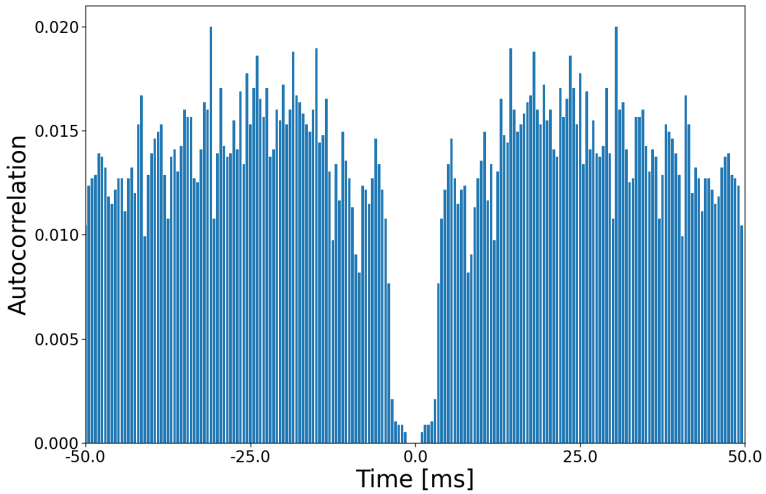


Figure 2.6: Autocorrelogram computed from the spike train of a labelled unit (E). The width of each bin was 0.5 ms. The central bin, which would have been equal to 1 and non-informative, was set to 0 intentionally.

After computing the ACG for each unit, we wanted to parameterize it. To do so, we fit the positive-lag portion of the autocorrelogram with a 3-exponential curve fit [24], as described by the equation 2.7:

$$ACG_{\text{fit}} = \max\left(c\left(e^{-\frac{x-t_{\text{refrac}}}{\tau_{\text{decay}}}} - de^{-\frac{x-t_{\text{refrac}}}{\tau_{\text{rise}}}}\right) + he^{-\frac{x-t_{\text{refrac}}}{\tau_{\text{burst}}}} + \text{rate}_{\text{asymptote}}, 0\right) \quad (2.7)$$

Where x was the autocorrelogram, t_{refrac} the refractory period, the single τ (respectively decay, rise and burst) were the characteristic times for the different sections of the autocorrelogram behavior, and the other parameters, such as the coefficients and the $\text{rate}_{\text{asymptote}}$ were added to enhance the quality of the fit. The three exponential curves used for the fit can be seen in Fig(2.7a). Fitting the positive-lag portion of Fig(2.6) with the equation in 2.7, we obtained the result shown in Fig(2.7b). As we can see from this figure, the fit described well the behavior of the ACG. Of the parameters described by the fit, we focused especially on τ_{decay} , τ_{rise} and τ_{burst}, which we used as features descriptive of the autocorrelogram behavior in the classification analysis.

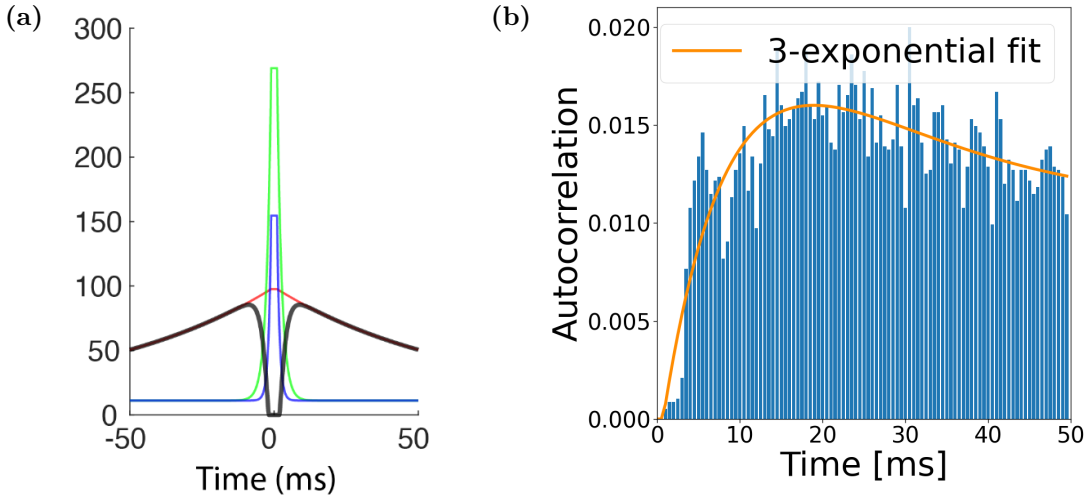


Figure 2.7: 3-exponential fit for autocorrelogram. (a) Image showing the three different exponentials that were used for the fit (red is decay, blue is burst and green is rise), adapted from [24]. (b) Result of the 3-exponential fit described in 2.7, applied to the positive-lag portion of Fig(2.6). As we can see, the curve determined by the fit (in orange) followed the behavior of the ACG. The average R^2 score for this fit, over all the autocorrelograms for the labelled data set, was ~ 0.8.

2.4.5 Burst features

In order to compute burst features, we first devised a method to detect bursts in our recordings. The algorithm designed worked in a few steps: first, it defined a threshold $ISI_{threshold}$ for the ISI of the specific unit which separated spikes belonging to bursts from those instead considered to be isolated, then it ran this threshold over the spike train of the unit to detect bursts.

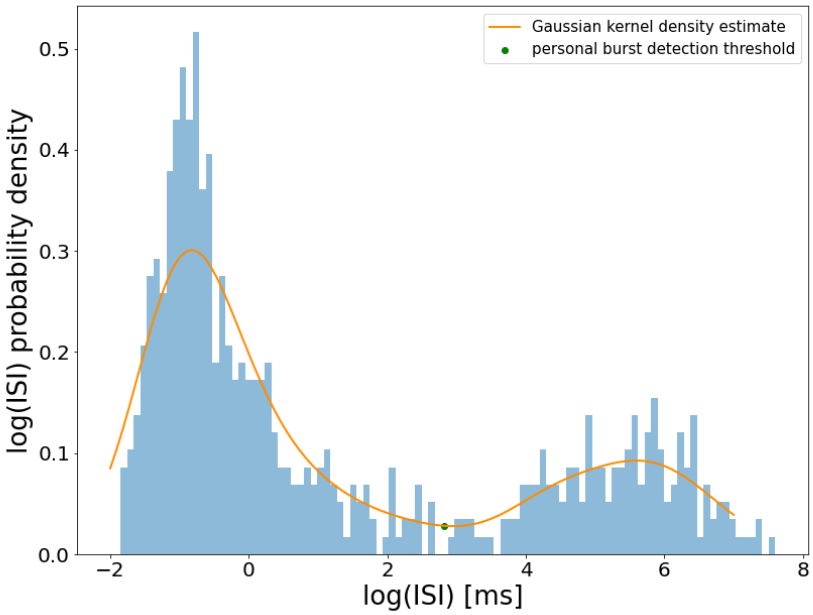


Figure 2.8: The distribution of the logarithm of the ISIs for a putative neuron (unit 0, MEA1k 4205). As we can see, the distribution was bimodal, and the threshold was found as the minimum (in green) of the corresponding Gaussian kernel density estimate (in orange). The threshold was denominated "personal" as it was the one specifically computed for the unit, and not one averaged over many units (which was the other alternative for the computation, which we rejected).

If the threshold found through the dynamic procedure for a putative unit resulted to be lower than 1 ms or larger than 100 ms, the threshold was hard set to 100 ms. This detector assumed that periods with high and low ISIs represented spikes that were either inside a burst or isolated [25]. A burst was detected if there were at least 5 consecutive spikes with ISI lower than $ISI_{threshold}$ and if the total duration of the burst was larger than 3 ms.

To calculate the $ISI^i_{threshold}$ for unit i, we computed the distribution of the logarithm of the $ISIs$ for i. As can be seen in Fig(2.8), this distribution typically had a bimodal behavior: spikes that belonged to the left mode, which corresponded to lower ISIs, were spikes that were present within bursts (phasic firing), while those that appeared in the right mode were isolated spikes (tonic firing), with higher ISI. The threshold was then the local minima between these two modes, computed after smoothing the distribution with a *Gaussian kernel density estimator* (with `bandwidth=0.2`, using the `scipy.stats.gaussian_kde` method in `Python`).

After detecting bursts, we computed for each unit burst features, such as inter-burst interval (IBI), burst average ISI (or Phasic ISI), burst average firing rate (or Phasic firing rate), average number of spikes per burst (to which we will simply refer to as "number of spikes" in the following sections) and amplitude attenuation of spikes within bursts.

Both burst average ISI and burst average FR were computed for each burst generated by a unit, and then the results belonging to the single bursts were averaged together. The IBI was computed by averaging the inter burst intervals (time interval between the end of a burst and the start of the successive one) between all bursts detected for a unit.

Attenuation

By attenuation, we referred to the dynamic changes of spike shapes (primarily amplitude) of a given unit, within bursts. Consecutive spikes inside a burst tended to become sequentially smaller, often following a behavior which we found to be exponential.

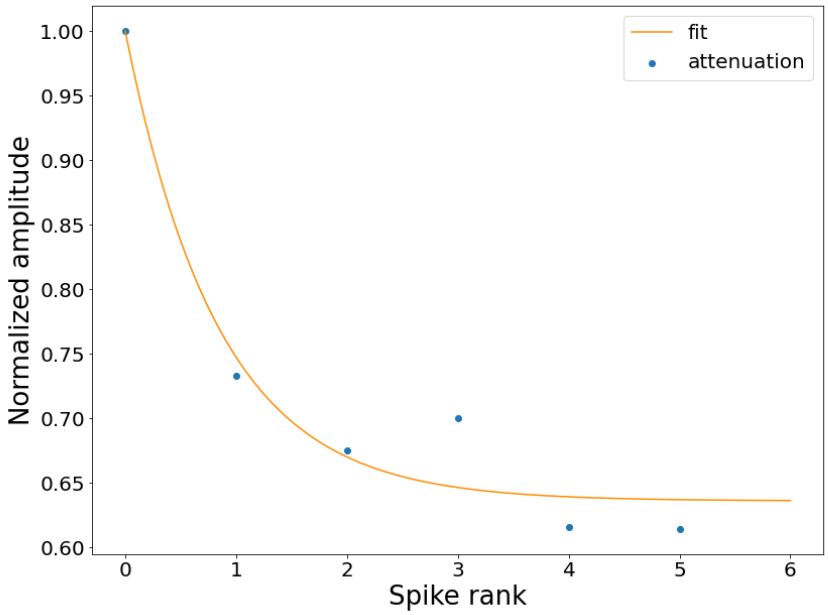


Figure 2.9: Average attenuation of spike amplitude along a burst for a putative neuron (unit 0, MEA1k 4205). In blue we have the average amplitudes for the different ranked spikes, normalized with respect to the largest one. In orange the exponential fit from which we extracted the exponent, as a parameter representing the behavior for the specific unit.

We wanted to represent this behavior as a single parameter in our feature space. To do so, we selected for each unit all the bursts containing at least 6 spikes, where the rank of the spike indicates its position within the burst (so the first spike in a burst has rank 1, the second, rank 2, and so on). We then averaged the amplitudes of all spikes having the same rank, and built a sequence of average amplitudes to represent the average attenuation for the unit, normalized by the largest average in the series. Eventually we performed an exponential fit, which can be seen in Fig(2.9), and the exponent (Att_{exp}) of the fit was used as the parameter to insert in the feature space. The only constraint imposed on the fit was that it had to pass through the first point of the sequence (so

the average amplitude or rank 1 spikes). As access to the raw data for STP ground truth data set (described in 2.6.1) was not possible, we couldn't compute this feature for the labelled units. For this reason, this feature was excluded in the end from the classification analysis.

2.4.6 Multichannel features

One of the most important features of the HD-MEA is the high spatial resolution: each array is constituted by 26,400 electrodes, with a pitch of 17.5 µm. We postulated that if our routed channels are close enough, the electrophysiological recordings could also carry information about morphological and functional identity of the cell. To convey this information in the form of features, we developed the concept of **multichannel features**. For each unit, we sorted the channels based on the amplitude of the templates computed over them by the spike sorter, in descending order. Each one of the waveform features was then computed for the footprint waveform computed over the first 5 channels ordered this way, for all the units. This way, all the waveform features were represented, in this multichannel space, as arrays with 5 elements, instead of just scalars as for the single channel features previously described.

For each unit, the 5 best channels were then ordered as follows: to each channel corresponded a template waveform, computed by the spike sorter as an optimized and processed average of all the spikes fired by the unit. For each one of these templates, we determined the time point corresponding to the peak of the signal (corresponding to the lowest point of the trough in our recordings). We then ordered the channels based on these time points, which represented the **latency** of the spikes for that channel and were tied to the propagation of the signal from the AIS, where it originated. The assumption was that the channel with the lowest latency, or the earliest spike, was the closest to AIS of the neuron, and increasing latency indicated removal from it. All the feature arrays were sorted according to this latency based ordering. In case two channels had the same latency, the order was then chosen randomly between the two.

After being ordered, the feature arrays were scaled between the minimum and the maximum values for each unit, following the equation:

$$\begin{aligned} \overline{X}_{std} &= \frac{\overline{X} - max(\overline{X})}{max(\overline{X}) - min(\overline{X})} \\ \overline{X}_{scaled} &= \overline{X}_{std}(max(\overline{X}) - min(\overline{X})) + min(\overline{X}) \end{aligned} \tag{2.8}$$

where $\overline{X}$ is a feature array. This was done in an effort to limit the influence of distance between the neuron and the recording channels. Scaling these features this way removed any dependence over the amplitude of the spikes, and only retained information about the effect distance from the AIS had on the features. All feature arrays were this way composed of values between 0 and 1, for all units.

2.5 Imaging pipeline

The fulcrum of the project was to collect a ground truth data set, containing cells labeled as either GABAergic (inhibitory) or Glutamatergic (excitatory). In order to do so, we stained our cultures via immunocytochemistry staining, with antibodies specific to the two classes, and collected the stained images via microscopy. These images were later processed to find the positions of the single neurons and create the data set.

2.5.1 Staining

In order to stain the cells, we used an immunostaining technique based on primary and secondary antibodies. First we fixed the cell cultures on the chips, using IC Fixation Buffer (FB) containing paraformaldehyde: we first cleaned the chips of their medium by rinsing once, slowly, with PBS, after which we removed the PBS and added the FB, which we left on for 10 minutes. After that we removed the FB and twice added PBS, removed it after 5 minutes and added it again. Eventually we added $1ml$ of PBS and removed it immediately, then added $1ml$ of Blocking Solution (BS) per chip. The reason why we added the BS was to block the specific sites to which the secondary antibody could inadvertently bind, ruining the staining process. The blocking solution was composed of 5 mL PBS (10X), 2.5 mL Triton, 0.5 mL sodium azide, 0.5 g bovine albumin serum, 5 mL normal donkey serum and DIW to reach a total of 50 mL. Once filled with BS, the chips were left for 45 minutes on an oscillating plane (5 oscillations/minute), after which the BS was removed. The chip was then filled with the primary antibodies, added to a so-called Antibody Solution (AB). The preparation for the AB was the same as for the BS, with 1.5 mL of normal donkey serum instead of 5 mL. The primary antibody solution for each chip was composed of: 1 mL AB, 5 µL Rabbit GAD65/67 for GABAergic cells, 25 µL Mouse HUC and 2 µL chicken NeuN. The chips were left on the oscillating plate for 2 h with this solution, which was then removed and substituted with the secondary staining (after cleaning the chips with PBS three times). HUC antibodies bind to antigens present exclusively in all neuronal cells; Glutamate decarboxylase 65/67 (GAD 65/67) is involved in gamma-aminobutyric acid (GABA) synthesis, which is an indicator of inhibitory neurons; NeuN is a neuron-specific nuclear protein, and Anti-NeuN stains exclusively neuronal cells in the central and peripheral nervous systems. The composition of the secondary staining solution was: 1 mL AB, 2 µL Donkey-anti-Rabbit (AlexaFluor, 488 nm), 2 µL Donkey-anti-Mouse (AlexaFluor, 568 nm), 2 µL Donkey-anti-Chicken (AlexaFluor, 647 nm) and 1 µL DAPI (405 nm) as a counterstain. Once again, the chips were left in the dark on the oscillating machine for 2 h, then cleaned of the secondary staining solution with PBS. Finally, 1 mL of PBS was added to each chip as to not let them dry up, and they were conserved in the fridge until we could put them under the microscope.

2.5.2 Microscopy

For the imaging we used a confocal microscope with spinning disk (W1 Upright Spinning Disk Confocal, Nikon Zstage + MCL Piezo), and a dipping lens with a magnification

of $20X$. The number of stages to cover the entire chip was variable between 28 and 40, depending on how close we wanted to be to the border of the array. For each stage multiple images for the different wavelengths were taken, for different z-positions around a single central z-position, which varied depending on the HD-MEA. For each stage and z positioning, we took images via lasers (488 nm for GAD65/67, 568 nm for HUC and 647 nm for NeuN as counterstaining). In order to capture the underlying electrode array, we used a 515 nm laser. The full images were stitched together using `Huygens Professional` (version=21.10; ScientificVolume Imaging, 10% overlap, circular vignetting correction mode).

2.5.3 Image processing

Each stage image collected by the confocal microscope was composed of 5 channels and around ~ 25 stacked images for each of them, corresponding to the different z-positions. The stacked images were collapsed into one (corresponding to a 2-dim matrix) by selecting for each pixel the maximum intensity one over all the z positions for that channel. Once the stage-wise images were collapsed, stitching of the stages was performed via `Huygens Professional`. Stitching was optimized over just two channels: the one capturing the electrode array, which we called "regular channel", and the one capturing either HUC staining or GAD65/67, called "irregular channel". The corresponding stitching template was then used to stitch over the other channels as well: this was done to increase performance time-wise. Coordinates of the four corner electrodes were extracted from the regular channel in the stitched image (the pixel at the intersection of the diagonals of the rectangular electrode was considered as position). These coordinates were then compared with those provided by the manufacturer, to construct a mapping between the positions of objects in the images and in the artificial frame of reference of the actual electrode array. This mapping took into account rotation of the electrode array in 3-dimensions, considering the axes passing through the lower left corner of the image as axes of rotation.
Cell segmentation was then performed using `Cellpose` [26]. Once the images corresponding to the HUC (all neuronal bodies, or somas) and the Gad 65/67 (only inhibitory cells) channels had been segmented, we removed from the images clusters of cells, considering cells with a respective distance of less than the diameter of a cell to be clustered. This was done because, in order to correlate imaged cells and spike-sorted units (which was our objective), we needed well distinguished neurons. We then subtracted from the HUC segmented image the inhibitory cells found in the GAD 65/67 channel, to obtain an image containing mainly excitatory neurons. Using the subtracted image and the GAD65/67 one, we could then pinpoint the positions of the routed electrodes standing directly below respectively excitatory and inhibitory cells, as described in section 2.6.2.

2.6 Ground truth data set

In order to perform classification between excitatory (E) and inhibitory (I) units, we needed a ground truth labelled data set to use both for the training and the testing of the different models. In order to establish this ground truth data set, two procedures were developed:

- Spike-transmission probability (STP) based labelling.
- Imaging-based (ICCS) labelling.

We can briefly see the methodology behind both of these procedures.

2.6.1 Spike-transmission probability

It is possible to establish the E or I identity of neurons based on their respective interactions, and the way they influence each other's firing pattern [27]. The method and data were the same used in [22], and were kindly shared by Dr. Julian Bartram.

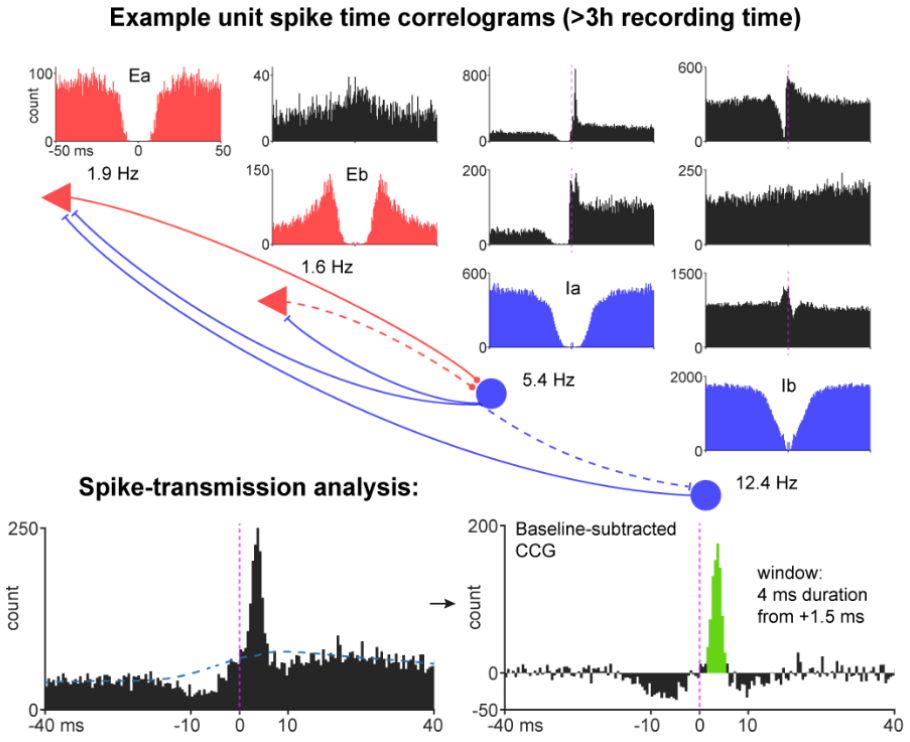


Figure 2.10: Visual description of the STP computation. (top) CCG between 4 different units (excitatory in red and inhibitory in blue), with respective putative interactions (the dashed lines represented weaker interactions, supposedly not direct). (bottom) On the let, the CCG between two units, with the slow baseline (dashed blue line). On the right, the histogram resulting from the subtraction of the baseline from the CCG. The bins in green, when summed, returned the STP value. Adapted from [22].

In order to determine if spike sorted units were either E or I, the concept of spike

transmission probability was used: if the average effect of a neuron on other neurons with which it had a direct synoptic connection was to promote their spiking, then it was excitatory; on the other hand, if it made their firing less likely, it was an inhibitory neuron. To check if the effect of a unit on another is excitatory or inhibitory, the cross-correlogram between the spike-trains coming from the two units could be used.

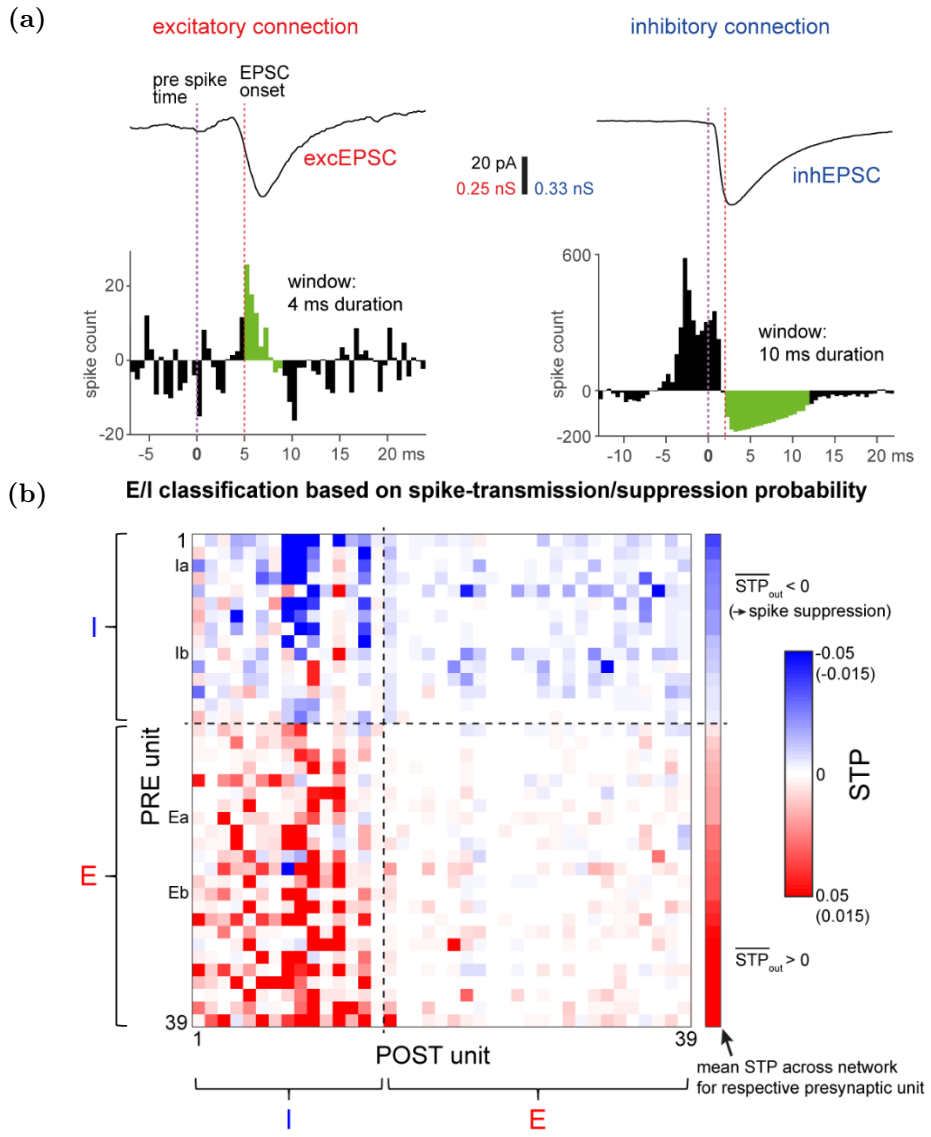


Figure 2.11: Difference in STP between excitatory and inhibitory units. (a) Results of the difference between the CCG and the baseline for an excitatory (on the left) and an inhibitory (on the right) unit. (b) The STP values for all considered interactions between pre- and postsynaptic units in a single recording. The mean STP for all presynaptic units is shown in the vertical bar to the right. Adapted from [22].

First, the CCG was computed between the two chosen units, using a recording of >3 h. In order to determine the post-synaptic effect of our target unit, a slow CCG baseline had to be extracted, representing the behavior we would expect to see between the

two units if there was no interaction at all. Extraction of this baseline was done by convolving the CCG with a partially hollow Gaussian kernel, with $std = 10$ and hollow fraction equal to 60% [27]. After subtracting this baseline from the CCG, the resulting bins were normalized by the number of spikes fired from the presynaptic unit (which was the one whose identity we were inspecting). The STP was then computed as the sum of bins in a 4 ms window starting at the 1.5 ms positive lag. To determine the class of a single unit, this procedure was repeated multiple times with different post-synaptic units: the results were then averaged to determine the STP.

In figure Fig(2.11) we can see a visual interpretation of the STP computation process. In Fig(2.10), we can see (top) the CCG for 4 different units (red are excitatory, blue are inhibitory) with their corresponding expected interactions represented as lines between either circles or triangles (dashed lines represent weaker interactions, probably not direct.) In the same figure we can see (bottom) how the STP for a single synaptic interaction was computed: first the baseline was computed for a single CCG (bottom-left of Fig(2.10)), represented here as a blue dashed line. This baseline was subtracted from the CCG, resulting in the figure on the bottom-right. The bins in green are the ones whose sum is equal to the STP. In Fig(2.11a), we see how different the CCG looked after baseline removal for putative inhibitory or excitatory pre-synaptic units. As can be observed, the difference between an excitatory and an inhibitory pre-synaptic unit was determined by the STP, given by the sum of the bins in green in the figure: if the presynaptic unit was excitatory, the STP had a positive value; if instead it was inhibitory, the computation returned a negative value. For the final classification of units as either E or I, we took as reference Fig(2.11b): in this matrix, the STP for all connections between 39 units were represented. The final STP value was computed as the average over all STP values calculated for one presynaptic unit (in the figure, this would correspond to the average between all values in a row). These mean values are represented in the vertical bar on the right of Fig(2.11b).

All the labelled units used in this report for the ground truth data set were computed with this procedure, and were the result of the work of Dr. Julian Bartram. All thanks and recognition go to the authors of [22]. Using this procedure, a labelled data set composed of 39 I and 54 E units was created.

2.6.2 Imaging-based labelling

In order to obtain a ground truth data set, with neurons labelled as either E or I, we designed a procedure based on immunocytochemistry staining and imaging. The procedure was the following for each HD-MEA: after the recording, the culture was fixed using PFA (paraformaldheyde), then stained following the procedure described in 2.5.1, for all neuronal cells and also, specifically, for GABAergic neurons. Afterwards, the array was imaged following the methodology described in 2.5.2. The images collected for a single HD-MEA, which ranged between 35–42 tiles that covered the entire array, were stitched together using `Huygens professional`, to form two images representing the population: one for the neuronal staining (image N), and one for the GABA staining

(image G). Cell segmentation was performed on each image using `Cellpose2.0` [26]. Using a custom code written in `Python`, the images were then filtered for clusters of cells, which we wouldn't have been able to use for the cell position identification and correlation that we needed for the labelling. The clustering filtering was simply set such that all cells that had a distance lower than the average diameter of a cell from the closest one were removed from the image. Subtracting from image N, all cells identified in image G, we were left with an image (image E) containing only non-GABAergic cells, which we assumed to be E cells. We then had two images containing cells belonging to either the E or the I classes. These cells had to then be localized on the array, and correlated with the units sorted using Kilosort2.5, to create a labelled data set.

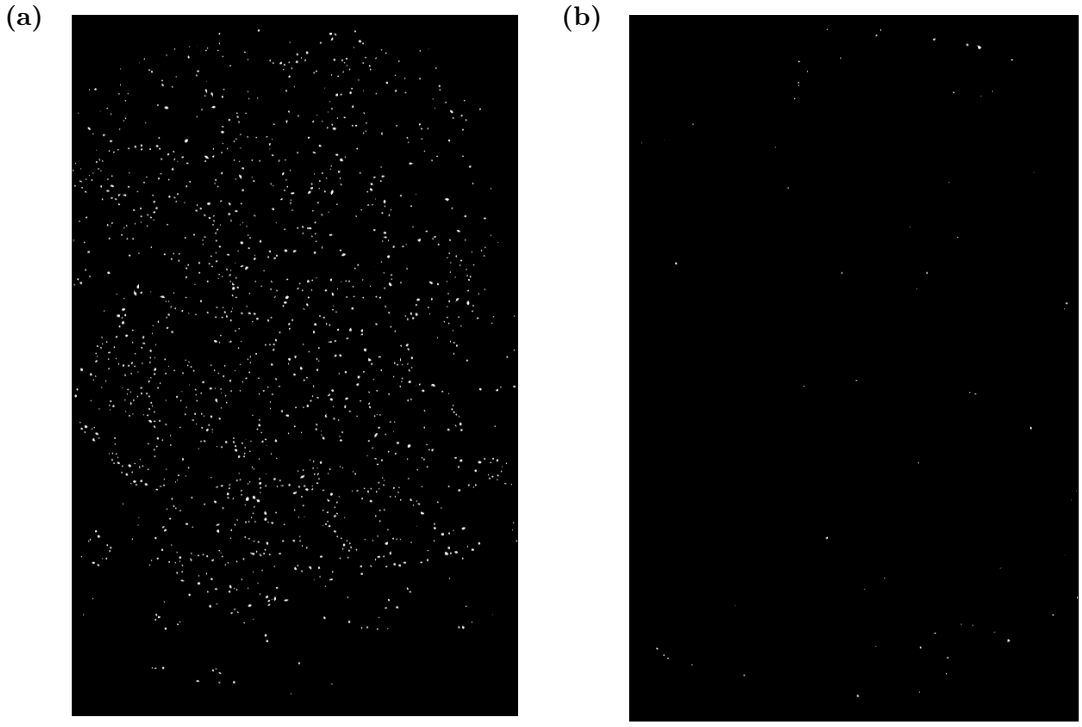


Figure 2.12: Masks derived from the images immunocyto-stained for all neurons. (a) and GABAergic neurons (b), via `Cellpose2.0`. The white dots are segmented cells. As we can see, the image on the left was far more populated than the one on the right, which could indicate either an absence of GABAergic cells in our population or a malfunction of the antibodies used for the staining.

In order to correlate the segmented cells to the sorted units, we first had to find the electrodes that were closest to the cells in the microscopy images. To do this we collected also a full image of the electrode array of the HD-MEA. Knowing the position of 3 corners of the array in the image, we could then recreate an artificial array, using the spacing given by the producers of the chips (17.5 µm) and the correct number of rows and columns, and fit it over the electrode array in the images. We can see the corresponding rotated and translated electrode array (in red) drawn over the imaged array for one of our chips (HD-MEA 4171) in Fig(2.14b). We can see in Fig(2.13) the way in which artificially computed electrodes fitted over the real, imaged counterpart.

In this figure (in which we have overlaid both the YFP image of the electrode array and the HUC image for the neurons, which can bee seen over the electrodes), the red dots represent the artificially computed electrodes.

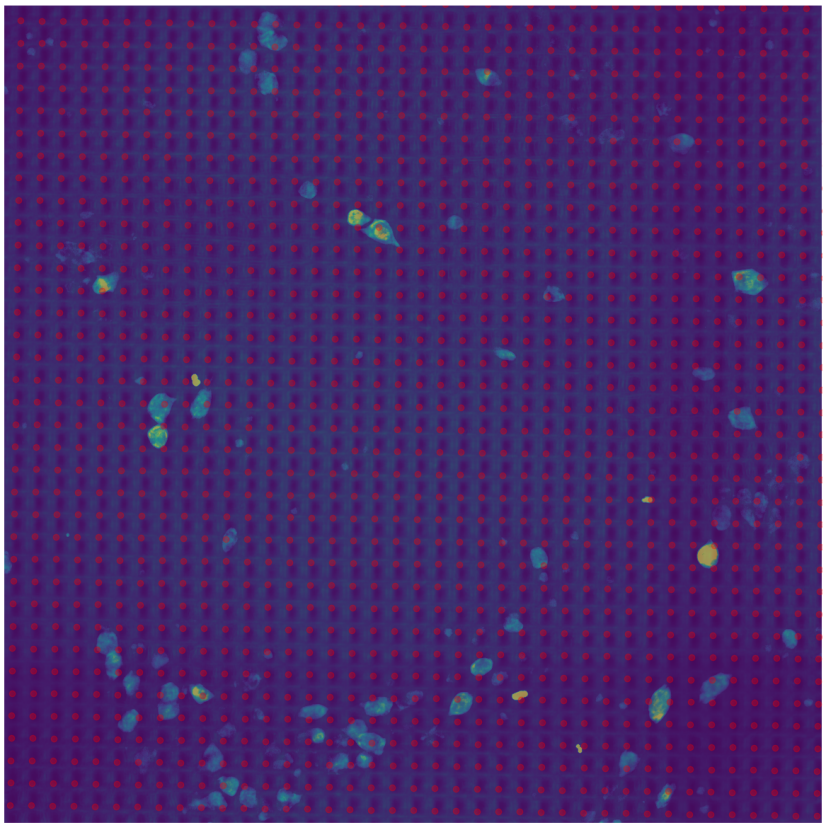


Figure 2.13: Artificial electrode grid overlayed on the imaged one. This image shows the way in which the artificially computed electrodes fitted over the real ones in the microscopy image. The real image shows both all the neurons (HUC staining) and the underlying array (captured via YFP imaging).

The artificial grid was translated and rotated in three dimensions to fit over the imaged one, by using three corners of the imaged array as reference. The rotation in three dimensions was needed because the frame for immobilizing the chip during imaging could have been rotated with respect to the objective, as well as the array itself could be slightly rotated over the chip. The array could also be tilted vertically, due to the gluing process that fixes it over the chip. By applying the composite transformation, taking as rotation axis on both planes the ones passing through the lower left corner of the array, we reduced the misplacement of the artificial electrodes with respect to the real ones. In the middle of the array, which was the most susceptible area to misplacement, we encountered at most a misplacement of $\sim$50 µm (less than two electrodes).

Once the artificial electrode grid had been generated, we traced the position of each channel (or routed electrode) on the image. For each cell belonging to either image E, for which the corresponding mask is shown in Fig(2.12a), or image G (Fig(2.12b)), after filtering, we traced the underlying electrodes. Electrodes were considered to be

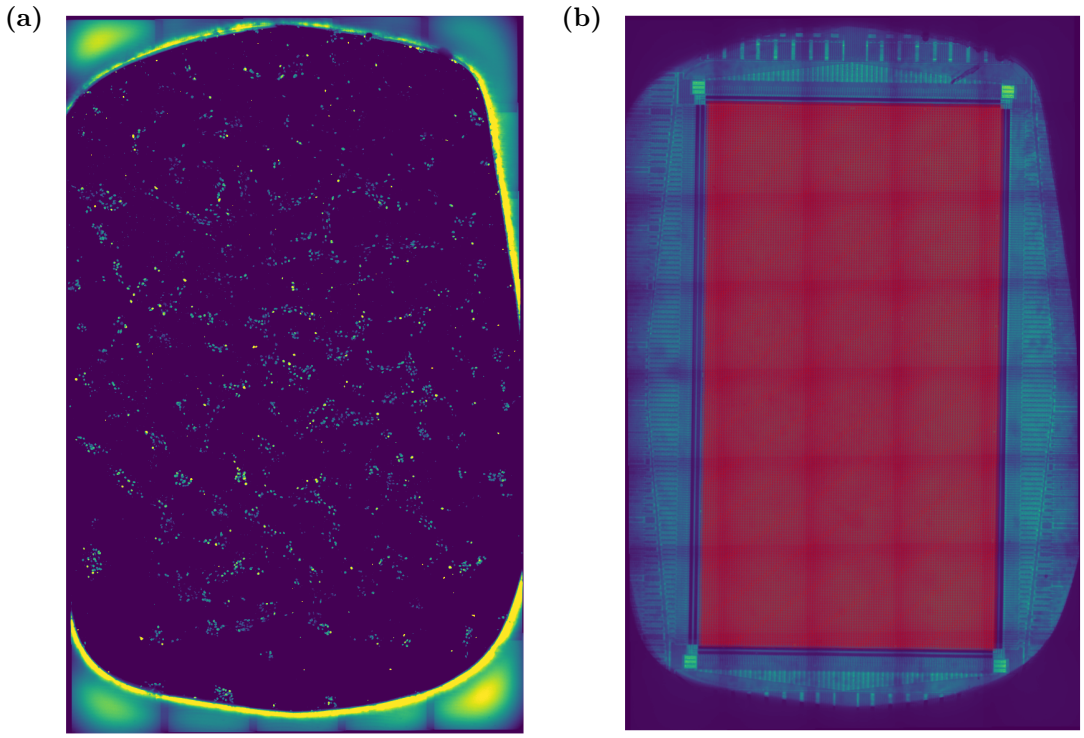


Figure 2.14: Immunocyto-stained images of our chips. (a) shows the microscopy image for the HUC-sensitive immunocyto-staining. This staining shows all the neurons, with no distinction between E and I neurons. The color coding is only representative of the intensity of the fluorescence measured by the microscope. (b) image showing how the full artificially computed electrode grid fitted over the real, imaged one. The underlying image is the microscopy image collected to visualize just the electrode array. In both (a) and (b), the lines in which the image becomes darker corresponds to the regions where different tiles were stitched together using `Huygens professional`.

"underlying" if any of the pixels that composed the surface of the electrode fell in the same space as the pixels of the cell in the mask.

We can use Fig(2.15) as a visual aid to better understand the full cell tracing procedure. Let us consider the cell closer to the middle of the image as our reference, which we will call cell R. Cell R is a cell that was identified to be E via the procedure described above. For each artificially computed electrode, we considered as the surface of the electrode for checking if it was beneath a cell a square $35\,\mu m X 35\,\mu m$, shown in the figure with a white outline. As we can see, cell R in the figure fell inside this area, so this was considered a directly underlying channel. If the cell fell inside the area of an electrode that was not routed during the recording, the directly adjacent routed electrodes were considered as indirectly underlying channels. All of the electrodes circled in white in Fig(2.15) were the electrodes that were identified as directly or indirectly underlying the cell in the middle, which was an E neuron in Image E. These electrodes constituted the list of underlying channels for cell R.

Once we had such a list for each cell identified in image E and image G, we proceeded with correlating these cells with the units sorted by `Kilosort2.5`. To do so, we also

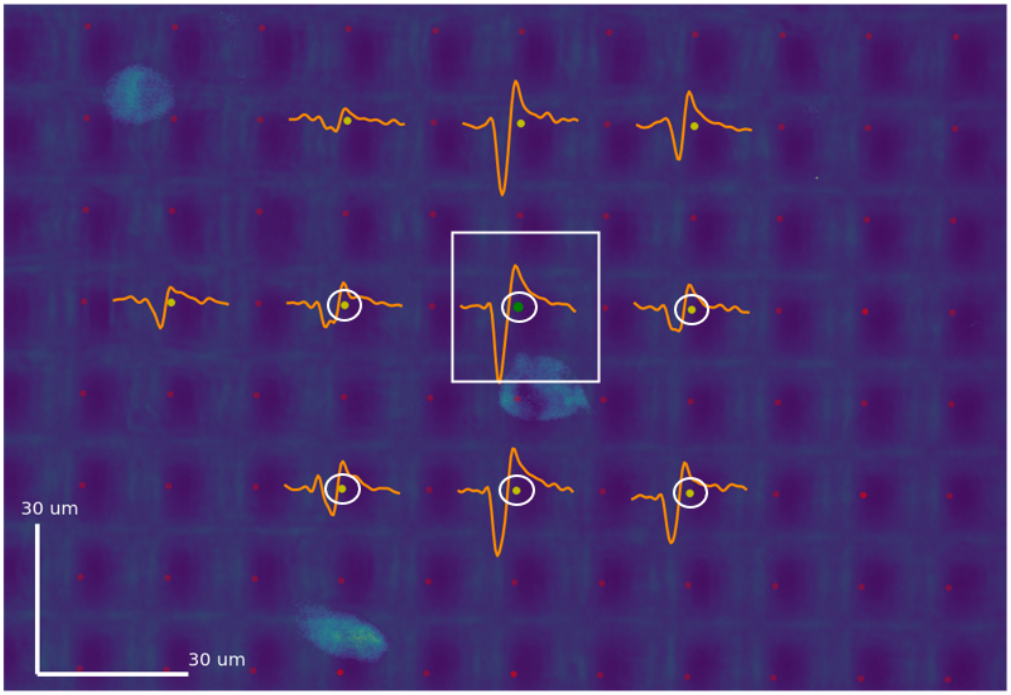


Figure 2.15: Image describing the procedure to assign immunocyto-stain determined labels to the spike-sorted units. The dots represent channels of the artificially computed array. The white square outlines the surface assigned to the channel in its middle (in green). A similar area was assigned to all channels in the array, and these were checked to see if any cell fell inside such area (such as for this case). Channels for which this condition was satisfied, or which were adjacent to non-routed electrodes for which it was, are circled in white. The green dot represents the best channel for a spike sorted unit, and the yellow ones the adjacent routed electrodes. The lines in orange are the footprint waveforms over the specific channels for the unit which had the highest level of correspondence with the cell in the middle, and to which its label was assigned.

built a list for each unit, composed of the identities (so the number that acts as ID for the channel) of the best channel for the unit, and all the channels adjacent to the best one. We then compared the channels contained in this list with the ones in the lists for cells in image E and image G. When two lists had a correspondence of more than 3 channels, then the label of the cell in the image (either E if in image E or I if in image G) was assigned to the unit with which this correspondence was found. We can see such a case in Fig(2.15): in this, the dot in green represents the best channel for a unit, while the yellow ones were the adjacent ones (all together they composed the list of channels for the comparison). All channels which have both a colored dot (either yellow or green) and are circled represent a correspondence between the unit and cell R. The label of cell R, in this case E, was then assigned to the corresponding unit (which in this case was u187 for the specific recording). In this case we had also an additional confirmation that our method for labelling units was working, as the best channel for u187 was standing almost directly beneath cell R. The waveforms shown in orange were the footprint waveforms computed over the single channels for u187, from

the raw recording. In the rare case in which multiple units shared channels with the same cell from image G or image E (rare as we first filtered the images for clusters), the label was assigned depending on the following hierarchical decision:

- the label was assigned to the unit for which the best channel was a directly underlying channel.
- if none of the units had a best channel which was directly underlying, the label was assigned to the one with the highest level of correspondence (largest number of channels shared between lists)
- if the correspondence level was the same for both units, the label was not assigned to either.

The same procedure was followed also in cases in which we had more than two units under scrutiny. As we can see from figure Fig(2.12b), this method presented an issue: not enough I cells were observed to make labelling possible. For this, two hypothesis were raised, still to be verified: either there was an issue with the immunocytochemistry staining (related to the concentrations of the agents, cross-talk or to the antibodies themselves), or no I cells were present on the chips to be observed. As the creation of a labelled data set did not work yet with this procedure, all the labelled units used in the following sections were determined using the spike transmission probability (STP) based method described in 2.6.1.

2.7 Machine learning techniques

The end result we were aiming for in this thesis was a classifier between E and I units, based on extracellular electrophysiological recordings. In order to obtain this result, we used a series of machine learning methods, to assess possible procedures for classifications. These consisted in two kinds of approach.The first was a visual one, in which we used dimensionality reduction machines to observe the distribution of the high-dimensional data in a lower, interpretable space. We implemented linear and nonlinear methods, in the hope of being able to detect via these a peculiar structure of the data in high dimension, which might represent a classification threshold. The second approach was more quantitative, and involved the use of supervised classifiers, both linear and nonlinear, to identify what feature space was going to be the best for the classification, and specifically which features.

2.7.1 Dimensionality reductions machines

As anticipated in the introduction, in a first step we tried exploring the data collected from our cell cultures via some machine learning techniques, to try and detect interesting underlying structures in the feature space that we could use to pinpoint a classification technique. As working in lower dimensionality features spaces hasn't shown any success in the same endeavor in the past [20], we decided to try and deal with many features at the same time, so in a higher dimensionality feature space or in its subspaces. To

efficiently perform analysis in higher dimensionality we resorted to dimensionality reduction machines, to try and reproduce the complex, high dimensional topology of the feature space in a number of dimensions which we could visualize and interact with. The main methods that we used were *PCA* and *UMAP*. We will briefly discuss the theory at the basis of each one of these methods, their weaknesses and strengths.

Principal Component Analysis

Principal component analysis (PCA) is a linear dimensionality reduction machine with high interpretability. It relies heavily on the varying behavior of the population. PCA can be used to reduce the dimensionality of any space by an arbitrary amount, usually dependent on the amount of variation we would like to preserve in the lower dimensional space.

The first step for PCA was to standardize the data: this was done because PCA uses the variation over the single features to design the reduced space. Standardization consisted simply in subtracting, from each feature, the corresponding mean and dividing by its standard deviation. Next, we computed the covariance matrix of our high-dimensional space, and calculated its eigenvalues and eigenvectors. The *principal components* were then identified by sorting the eigenvalues (and corresponding eigenvectors) in decreasing order. The principal components are new variables (or coordinates in the reduced space), constructed as a linear combination of the original variables.

Principal components are uncorrelated, and are computed in such a way that they contain most of the information just in the first few ones (the total number of components is the same as the dimension of the original space). The information is also referred to as *variational energy*, as it describes mostly the way in which the population varies across the different features (which is exactly what we need for this project). To retain this information, principal components are constructed such that the first ones contain information about the largest variance in the data set. The eigenvectors of the covariance matrix correspond exactly to the directions with the highest variance in the feature space. The eigenvalues represent the amount of variance described along the corresponding eigenvector, so we could create a rank of principal components based on the eigenvalues themselves. The variational energy contained in each eigenvector was then computed as the corresponding eigenvalue divided by the sum of all the eigenvalues. To obtain the final reduced feature space, each element of the standardized data set was projected into the space having as axes the principal components (so they were projected on the directions of maximum variance). In this project, PCA was performed via the `sklearn.decomposition.PCA` method in `Python`.

What made PCA powerful was its linearity: as the variables in the reduced space were just a linear combination of the original standardized ones, it was very easy to interpret the results of PCA. For example, if PCA underlined a separation between two modes, we could immediately tell what feature (or combination of features) was determining this distinction, just by backtracking the algorithm.

The weakness for PCA was related to the complexity of the original feature space:

often, if in the initial space the structural complexity of the data set was too high, or the data is distributed very homogeneously, each principal component accounted for just a small amount of information. If not enough information was contained in the first few features, so that it was directly observable by the user, we had to refer to other methods to extract classification and structures from the reduced space.

UMAP

The term UMAP stands for Uniform Manifold Approximation and Projection, and is used to represent a nonlinear dimensionality reduction method. UMAP works by first constructing a high-dimensional graph from the data, and then reduces it to a 2-dimensional version, optimized to be as structurally similar as possible [28]. In this high-dimensional graph, edges are constructed with weights that represent the probability that the corresponding points are connected. This probability of connection is established by extending a multidimensional sphere around each point, and connecting them when the spheres intersect. The choice for the radius of the sphere is chosen locally for each point by UMAP, based on the distance from the point and its n_{th} nearest neighbor. The probability of connection then decreases as the radius from the point grows, making up a "fuzzy ball" around each point (the full structure is called a *fuzzy simplicial complex*[28]). Thus, the graph is constructed, with the additional imposed feature that each point has to be connected to at least its nearest neighbors. The reduction to the lower dimensional space is then performed, optimized to maintain the same high dimensional structure. The most important hyperparameter to be tuned in UMAP is the number of neighbors (NN) to be considered when building the high dimensional graph: low NN will increase the focus of the reduction on the local interactions, while large NN will maintain the global structure more, at the cost of the local. Increasing or decreasing the NN is then comparable to increasing or decreasing the resolution of the reduction, at the cost of susceptibility of noise in the distribution (whereby noise we mean peculiar local structure non-representative of the actual distribution).

2.7.2 Supervised classifiers

Using the labelled data set given to us by Dr. Julian Bartram, we could train a series of supervised machine learning classifiers, both linear and non-linear. Each model was trained on all the possible different subspaces of the feature space (time-series features, single and multichannel waveform features and mixture of the last two with the first). As will be shown in the results section, the performance of each model was determined via a series of metrics. We computed the *Precision* of a classification as:

$$\text{Precision } = \frac{TP}{TP + FP} \tag{2.9}$$

Where TP and FP are, respectively, the counts of true positives (so correct classification) and false positive (incorrect classification). We also computed the *Accuracy*, as follows:

$$\text{Accuracy } = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.10}$$

In which TN and FN are true negatives and false negatives. As ours is a binary classification, positive and negative will just represent the excitatory and inhibitory classes. The last score computed was the *F1-score*, for which the formula is:

$$F1 \text{ score } = 2 * \frac{\text{Precision } * \text{ Recall}}{\text{Precision } + \text{ Recall}} \tag{2.11}$$

For which we also need to define the *Recall*:

$$\text{Recall } = \frac{TP}{TP + FN} \tag{2.12}$$

The F1-score is the most reliable source of information for the classification, in cases in which the training data set is skewered towards one of the two classes (as is our case, with 54 E and 39 I units). For the assessment of feature importance, one method was implemented for all classifiers: the *permutation feature importance*. The permutation importance is computed as the decrease in a certain classification score (in our case we chose the F1-score), with respect to permutation across all the data points of every feature in the feature space, in succession. This permutation is performed multiple times over the same feature, and the importance is computed as the average over all the implementations. All the classifiers then had specific procedure to compute feature importances, which are going to be described in the following sections.

It is particularly important, for nonlinear classifiers, the correct choice of the hyperparameters involved in the computation. The procedure via which these hyperparameters were optimized over the different feature spaces is described in Appendix(B)

Linear support vector machine

The linear support vector machine (LinSVM) is a linear classifier that aims at determining the *maximum-margin hyperplane* between the two different classes in the labelled data. This is defined as the hyperplane which is maximally distant from the nearest data points belonging to either class. This hyperplane is defined as the set of points that satisfies the condition:

$$\mathbf{w}^\top \mathbf{x}_i - b = 0 \tag{2.13}$$

where x are the points belonging to the hyperplane and w is the vector normal to the hyperplane, while b is the parameter that determines the offset of the hyperplane along w with respect to the origin. As we did not expect the data to be simply linearly separable, we used a version of the SVM based on a soft-margin procedure (all the computations were done via the `svm.LinearSVC` function in the `sklearn` package in `Python`), which uses a `hinge loss` function (2.14) to determine the best hyperplane.

$$\max\left(0, 1 - y_i\left(\mathbf{w}^\top \mathbf{x}_i - b\right)\right) \tag{2.14}$$

Logistic regression classifier

The logistic regression classifier is a linear classifier that, in the case of binary classification, works by determining the probability of a single data point of belonging to either class. The *log-odds* for each data point to belong to one of the two classes are computed, as linear combination of independent variables. The method consists in computing the cross-entropy of the predicted distribution with respect to the actual distribution of labels in our labelled data. The cross-entropy, also called negative loss, is then minimized, to determine the parameters of the logistic regression.

Kernel support vector machine

An alternative to the simple, even if powerful, linear support vector machine is the kernel support vector machine. This method is used in cases in which the performance of the LinSVC is low, and consists in performing the same procedure of the LinSVC with soft-margin, but with an additional dimension added to the feature space. A new feature is added to each data point, with a computation based on the real features, in the hopes that this additional dimension will disentangle the data and allow for the LinSVC to perform better. The function used to calculate the additional feature is called *kernel*, and many kernels are available for this computation. To be specific, in our attempt at classification we implemented a Gaussian Radial Basis Function (RBF) as our kernel, a powerful tool that improves classification but makes the classifier nonlinear. This kernel depends on two parameters, gamma and C, which were optimized via a grid search for each feature space used (for the optimization we used the `modelselection.GridSearchCV` in the `sklearn` package in `Python`).

Ensemble decision tree classifiers

Decision trees are a particularly powerful tool in the classification field, as they are "invariant under scaling and various other transformations of feature values, robust to inclusion of irrelevant features and produce inspectable models" [29]. A single decision tree works by selecting a random subset of m features from the total feature space (this is called "feature bagging"), and choosing among those the best feature and best threshold value to split the population in two classes, attempting at separating the labelled classes as well as possible according to some measures of impurity (in our case, the Gini index). The two separated families of samples form two new leaves of the tree, and the procedure is repeated for each node. This algorithm is repeated sequentially until we reach a point in which each leaf has data points belonging to only one class (minimal impurity) or until a depth (depth represents the number of sequential iterations of the algorithm computed) imposed by the user. The best kind of classification for a decision tree is obtained when the tree spontaneously reaches an iteration at which the impurity has a value of 0, but this kind of deep trees is very prone to overfitting. The use of ensemble decision trees classifiers (EDTC) makes up for this problem: many decision trees, or learners, are trained over the labelled data, and the classification will be determined as the majority vote by this ensemble of trees. In order to improve the performance of this EDTC, for each tree in the ensemble a

different random subset of the data set with replacement is chosen (this procedure is called "bootstrap aggregating"). The random choices in the bootstrap agggregation and in the feature bagging are what makes these classifiers nonlinear. The EDTC in this project was implemented using the `ensemble` module in the `sklearn` package in `Python`. This supervised classifier also had a specific procedure to compute feature importance, called *Gini importance* or *Mean Decrease in Impurity*: this is computed as the number of splits that use the specific feature over all trees, proportionally to the number of split samples and normalized over the number of splits.

Another model closely related to decision trees is the Gradient Boosted Model (GBM), which in this project was implemented via the use of the `lightGBM` package in `Python`. In this, each decision tree was built to perform better where the previous one made errors in the classification. Combining these trees, then, is bound to improve the performance. It is based on some loss function that the algorithm aims at minimizing: each new tree is then added to the model such that it reduces the loss function. To do so, the tree's parameters and subsets of features and samples are optimized, following this gradient descent procedure. The procedure stops after a set amount of iterations or when the loss function reaches a plateau and does not decrease anymore.

2.8 Full pipeline

We briefly summarize the full procedure for this project. Primary hippocampal rat neurons were cultured over an HD-MEA, and recorded after 14 DIV. Each recording was performed as a 7X-sparse configuration on `MaxLab Live`, so that for each HD-MEA the final output were 7 different raw files, corresponding to different areas of the electrode array and covering it fully. Each of these recordings was spike-sorted individually with `Kilosort2.5`, then all resulting units from a single HD-MEA were collected into a single data set. For each putative neuron output by the spike-sorting procedure we computed a series of features, both over a single channel or concatenated measures over different channels (multichannel features), and both for the waveforms and the time-series. The data sets also went through a rigorous process of curation, based on these features, to remove all putative neurons that would make our classification effort more difficult. This full procedure was repeated for all of our HD-MEAs.

At this point, we had two different pipelines working in parallel: an ICCS and a STP based one. For the ICCS pipeline we fixed the cells in the recorded HD-MEAs, immunocyto-stained them for all neuronal species and for inhibitory neurons specifically, and imaged them with a confocal microscope at 20X enhancement. The images for a single HD-MEA were stitched together with `Huygens Professional`, segmented with `Cellopse` and processed to find a correspondence between imaged neurons and spike-sorted putative neurons. Once these correlations were found, the corresponding labels from the images (either excitatory or inhibitory) were used to build a labelled, ground truth data set of computed features. The STP pipeline worked similarly, but used a different method, based on neuronal connectivity and electrophysiology, to establish the labels for the putative neurons (described in 2.6.1). We once again thank

Dr. Julian Bartram for allowing us to use his STP data set.

The labelled data sets were then explored visually, both with simple plotting and with the use of dimensionality reduction machines, to observe any peculiar structure in the data. Eventually, the labelled data sets were used to train and test different classification models, in an effort to successfully classify between E and I putative neurons. The importance of the features towards classification was also studied, to infer information regarding the physiological differences between the two classes *in vitro*.

Chapter 3

Results

In this chapter, we present the results of our efforts to determine one, or a series of extracellular electrophysiological features that could be used to classify neurons between putative excitatory (E) and inhibitory (I) classes. The first part of the chapter is dedicated to the curation procedure and feature space exploration, consisting of the visualization of the distributions of single or pairs of features [18][20]. This is followed by the analysis of high-dimensional feature spaces, containing most of the information gathered for our units, and reduced via dimensionality reduction machines to be observed and studied. The last part of the chapter is dedicated to the results of a series of supervised machine learning classifiers, linear and non-linear, trained and tested over the data set labelled based on spike-transmission probability (STP) estimates.

3.1 Data exploration

We started our data exploration by examining the curation features for our data. This helped us determine the best threshold values to impose for curation. Using strict enough threshold values, we could extract the best possible units from our data set, in order to achieve reliable classification.

The first step to discern a classification feature between E and I units (or putative neurons) was to explore the single feature distributions (or at most two-feature distributions). In case a separation between modes was observed in one of these distributions, we explored the specific feature to try and understand where this diversification might arise, given the physiological aspects of our systems. In this section, we started by studying exactly these kinds of distributions, initially over the unlabeled data set as well as the labelled one, but then focusing on the STP data only. The reason for this was that even if we were to observe a separation in clusters in the unlabeled data, we would have no way to tell for sure if it was representative of the two classes, an artifact of our data set or another behavior unrelated to the E and I classification. This could result in incorrect deductions, based on incomplete information, and we would anyway need to observe the same separation in the labeled data set to draw some conclusions.

In a second step, we explored higher dimensional feature spaces using dimensionality reduction machines, linear and nonlinear, to embed this spaces in 2-dimensional plots containing as much information as possible, which we could then observe and analyze one by one.

3.1.1 Semi-automatic curation check

In the first stretch of this project, in which we still did not have an established ground-truth data set, we tried exploring our data, to try and identify the features which would be most relevant for us. For this analysis, we built a full pipeline, going from the recorded raw data through Kilosort2.5 [21] for spike sorting, then the computation via a custom code of many features, regarding both waveforms and time-series, which are described in (2).

First, we had to deal with the curation of the putative units: that is, the removal of falsely detected neurons, resulting, typically, from spike sorting artifacts. After the standard automatic curation, described in Section(2.4.2), we started by checking the footprints of some randomly chosen units among those that were included by the curation, to assess if an ulterior curation was needed. From a "good unit", we expected a footprint like the one in Fig(3.1). There, templates and cutouts over the best channel as well as the closest routed electrode fit well each other, which is an indication that the spike sorting algorithm identified a real neuron, whose AIS was in the vicinity of the best channel (in this case, channel 996). The fact that the templates and cutouts over the other channels were all not too small is also the indicator of a good detection. In Fig(3.1) we have an example of a bad footprint: cutout and template over the best channel (134) are very different, and over the other channels we have cutouts with a very little spike or just noise. In this case, the unit was probably an artifact of the sorting technique, as it is very unlikely for routed channels near a neuron to not pick up anything when the neuron is spiking.

We also see that the waveform footprint over the best channel has a much deeper trough than the corresponding template. This usually suggests the splitting of a single recorded neuron in two different units (usually due to a strange firing pattern, or fluctuations in the recording). The missing sink in the best channel template, which is visible in the waveform footprint, must have been dealt with in another unit, localized close to the location of unit 65 (abbreviated as u65). Depending on how noisy the recording was, we had variable amounts of aberrations and spike-sorting artifacts like this in our results, some of which survived the first automatic curation.

One way of checking if this was indeed a split neuron would be to look at the cross-correlograms (CCG): if the spike sorting algorithm had split a single neuron in two putative units, we would find two units with strong correlation. Let us consider u65 as an example, of which we can see the footprint in Fig(3.2). The first step is to check which units share the most channels (surrounding the best channel) with it, which indicated units that were detected as close to one another. Over these channels, we computed the average similarity between the two units over the templates. If we take

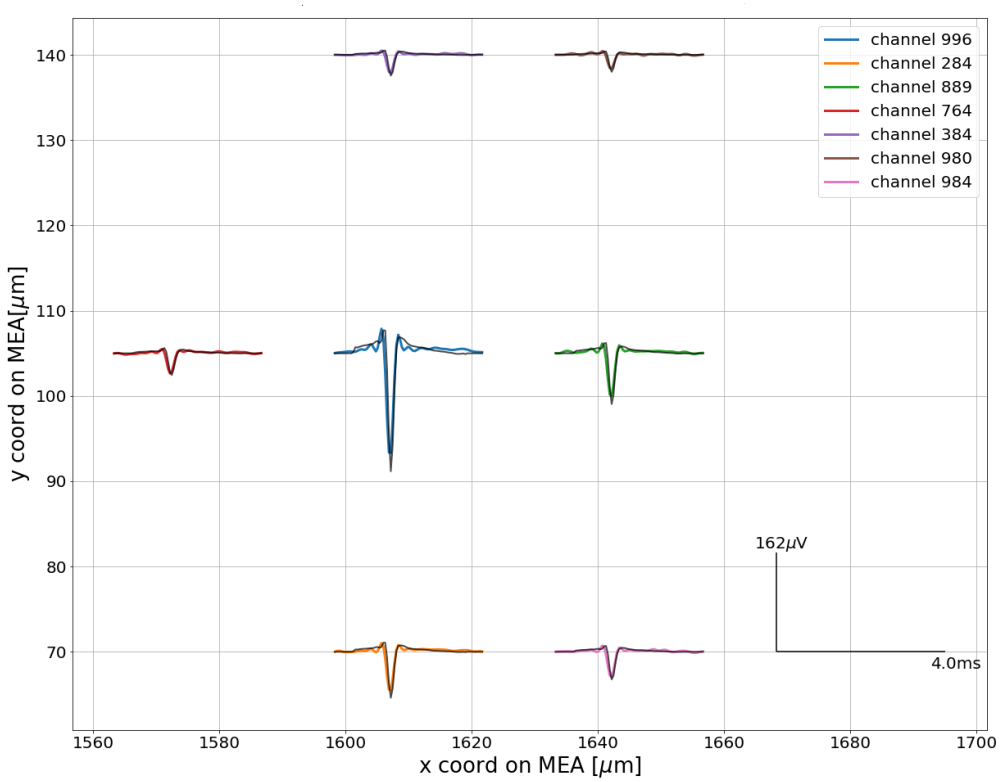


Figure 3.1: Spatiotemporal footprint of u515, (from MEA1k 4205). In black we represent the overlaid templates over the corresponding channels (which are chosen as the closes active channels to the best channel, which is the first in the legend list). The colored waveforms are the corresponding footprint waveforms.

all units with which u65 shares at least 7 channels, the one with the highest similarity in terms of templates was u64, with a similarity score of ~ 0.56 (computed as cosine similarity) and whose footprint is plotted in Fig(3.3). As we can see, this unit shows a better sink at channel 134 (its best channel) in terms of amplitude, which was the same channel at which we were missing part of the trough for u65 in Fig(3.2). This means, together with the similarity score, that it is a good candidate for a split neuron. We can then show the CCG corresponding to units u65 and u64, in Fig(3.4).

From the CCG in Fig(3.4) we could not discern if the two units were actually the same neuron: the autocorrelogram (ACG) for u64 had the shape we would expect for a correct unit, with higher bin values close to the 0 ms lag, but equal to 0 in the period exactly around the 0 ms lag, corresponding to the refractory period. On the other hand, the ACG for u65 had a shape determined by a low number of assigned spikes, with sparse bins homogeneously distributed and no clear pattern.

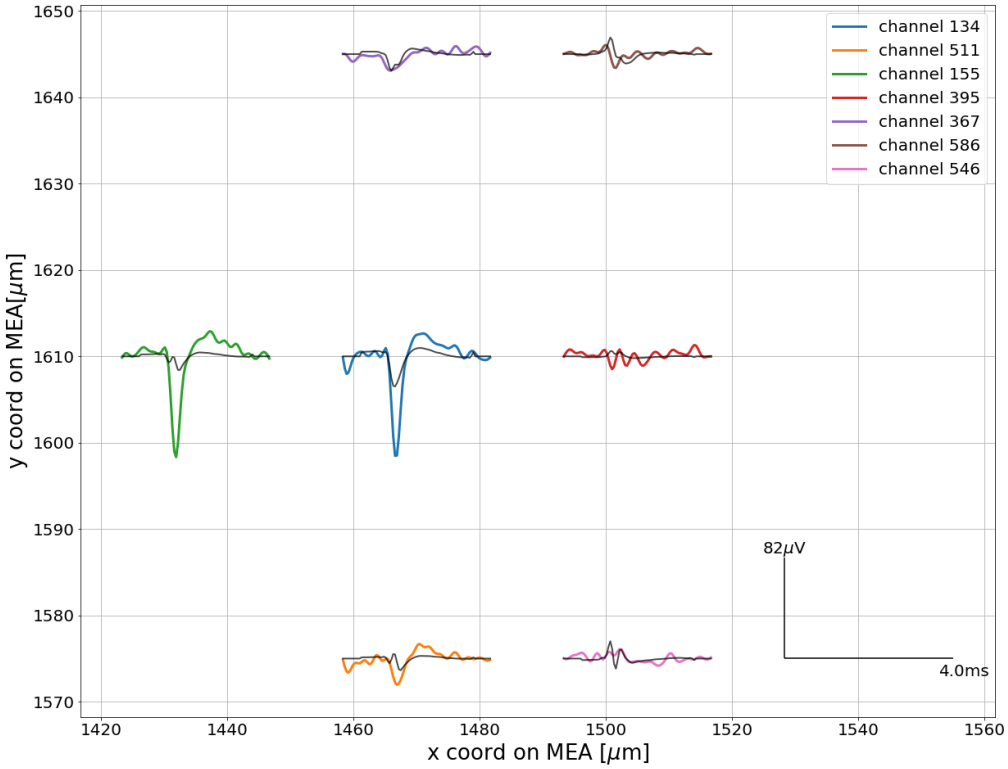
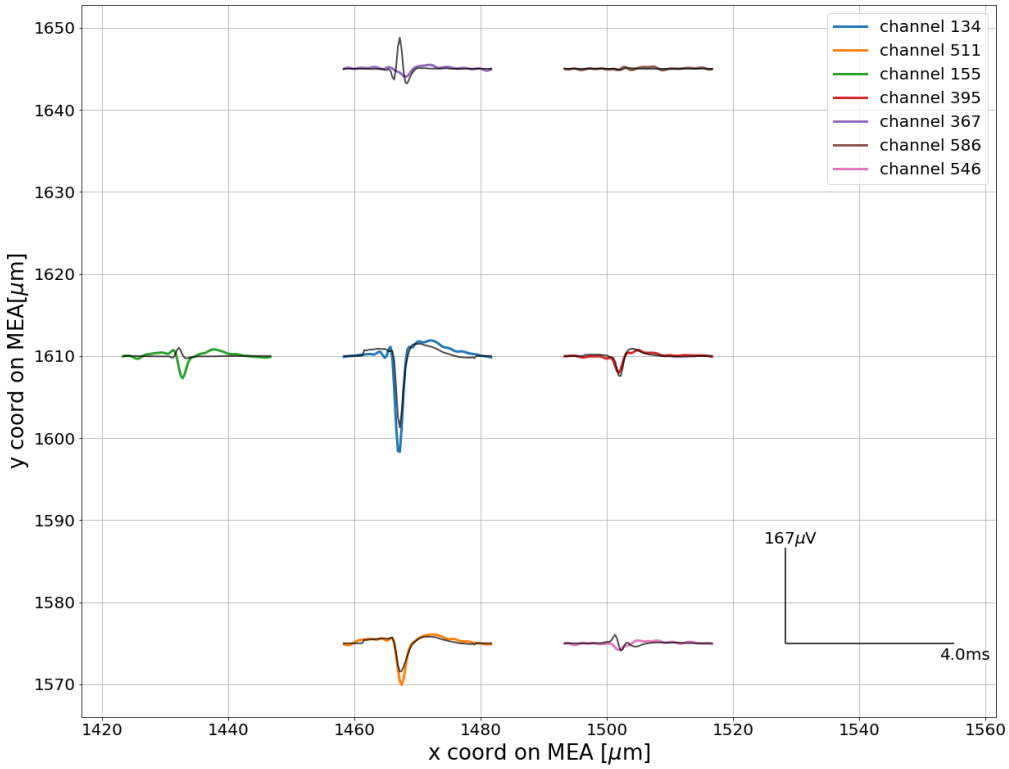


Figure 3.2: Spatiotemporal footprint of u65, (from MEA1k 4205). In black, we represent the overlaid templates over the corresponding channels (which are chosen as the closes active channels to the best channel, which is the first in the legend list). The colored waveforms are the corresponding footprint waveforms.

We then moved along the list of units which were similar to u65, for example, u77, which had a similarity score lower than ~ 0.56 (which was the similarity score between u65 and u64). Checking Fig(3.6), footprint waveforms and templates were similar for all channels but the best channel, in which we were seemingly missing part of the trough (as for u65). Once we checked the CCG in Fig(3.7) we saw that there was a high correlation between the two units, with a peak larger than 1 close to the $0ms$ lag. We could interpret all of these factors as the two units being actually a single putative neuron that was split erroneously during the spike-sorting. At this point we had three options: increase the thresholds for the curation, run one more time the function and see if one of the two units would disappear (likely u65, with a slightly higher threshold over the average firing rate), remove the worst unit (in this case u65) manually, or try merging the two units. Given the aim of the project, which required us to only maintain the best possible putative neurons to have clean features over which to build a ground truth data set, we avoided merging units unless the spike sorter only identified

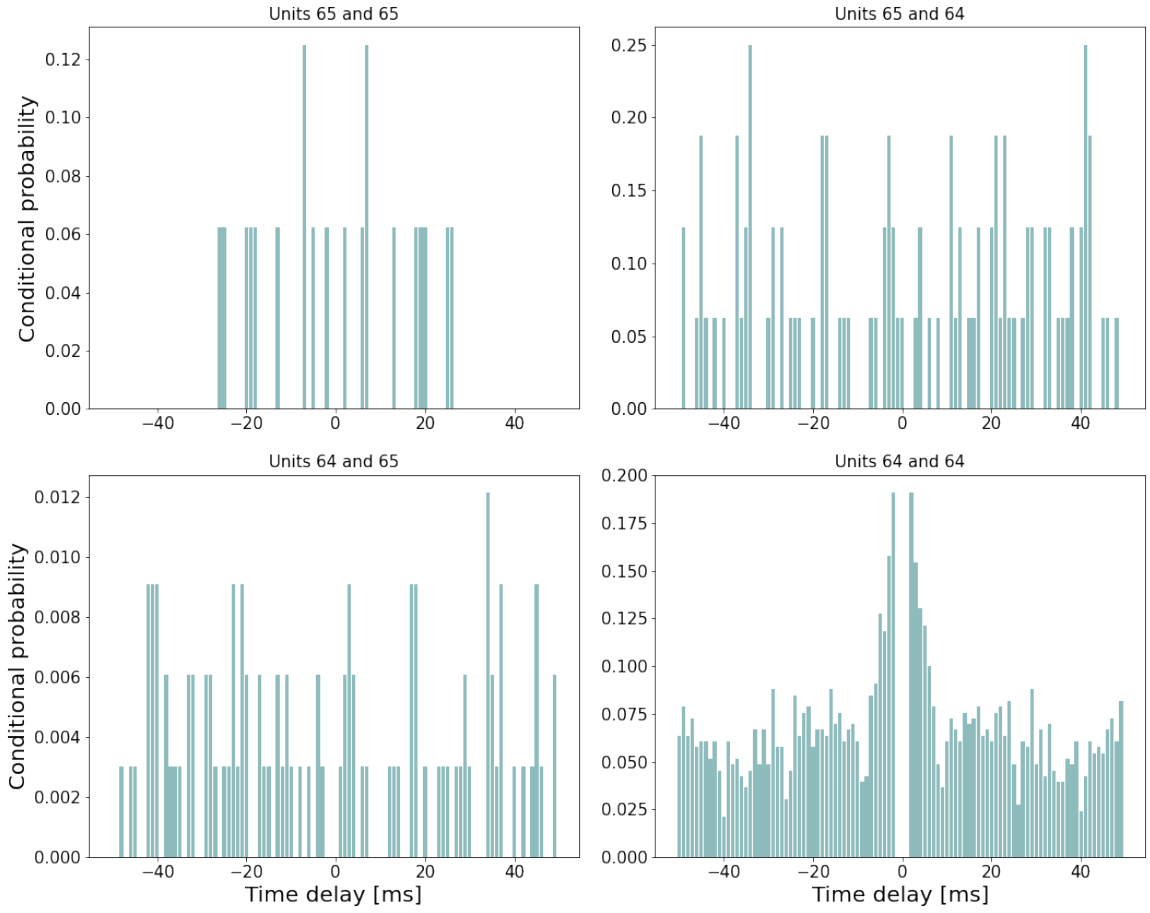


.

Figure 3.3: Spatiotemporal footprint of u64, (from MEA1k 4205). In black, we represent the overlaid templates over the corresponding channels (which are chosen as the closes active channels to the best channel, which is the first in the legend list). The colored waveforms are the corresponding footprint waveforms.

a few neurons. We preferred, in the case of a large number of putative neurons being sorted, to implement increasingly stricter curation, to maintain only the best units, with quality over quantity in mind. We thus observed the total distribution of our unlabeled data set over the curation features, shown in Fig(3.5), and established strict thresholds based on this observation. The orange vertical line in the figures represents the location of the threshold (either lower or higher) for each feature.

As for the inter spike interval (ISI) violation rate, we ignored all units with a value larger than 0.05 Hz, as can be seen in Fig(3.5a). This was due to the fact that we expected in our 20 minutes recordings, the units to have no more than 60 violations of the ISI. The value in this case could not be just 0 for the violation rate threshold: as we have seen, there are multiple reasons for which the spike-sorting algorithm might assign spikes even if they do not satisfy the ISI violation criteria (merging of neurons, overlapping of spikes, bursting and more). This first step removed around $\sim 8\%$ of the



.

Figure 3.4: Cross-correlograms between u65 and u64, (MEA1k 4205). Each bin has a width of $1ms$, so that the cross-correlogram spans a period of $100ms$ ($50ms$ per side) around each reference spike

total population.

As a second curation step, we imposed a threshold over the average firing rate of at least 0.04 Hz, shown in Fig(3.5b). We observed, over multiple recordings, that the spike-sorting procedure did not work reliably for units with less than 50 spikes assigned to them. These units ended up being in most cases just containers for noisy spike waveforms, which could not be assigned to already established units via the template matching. Given that we were dealing with 20 minutes recordings, the chosen threshold removed this kind of units which amounted to around $\sim$ 15% of the remaining population.

Finally, we focused on the signal-to-noise ratio (SNR) in Fig(3.5c), for which we imposed a threshold of at least 8. This was done based on the observation of the units that were placed under this value, their footprints and some of their features. $\sim$ 12.5% of the remaining population was left out this way, with the final curated data set being

composed of 768 units. Both u65 and u77 ended up being removed this way. Once again, even though numerous units were lost via this procedure, it was justified since we were much more interested in the quality of the units, rather than the quantity.

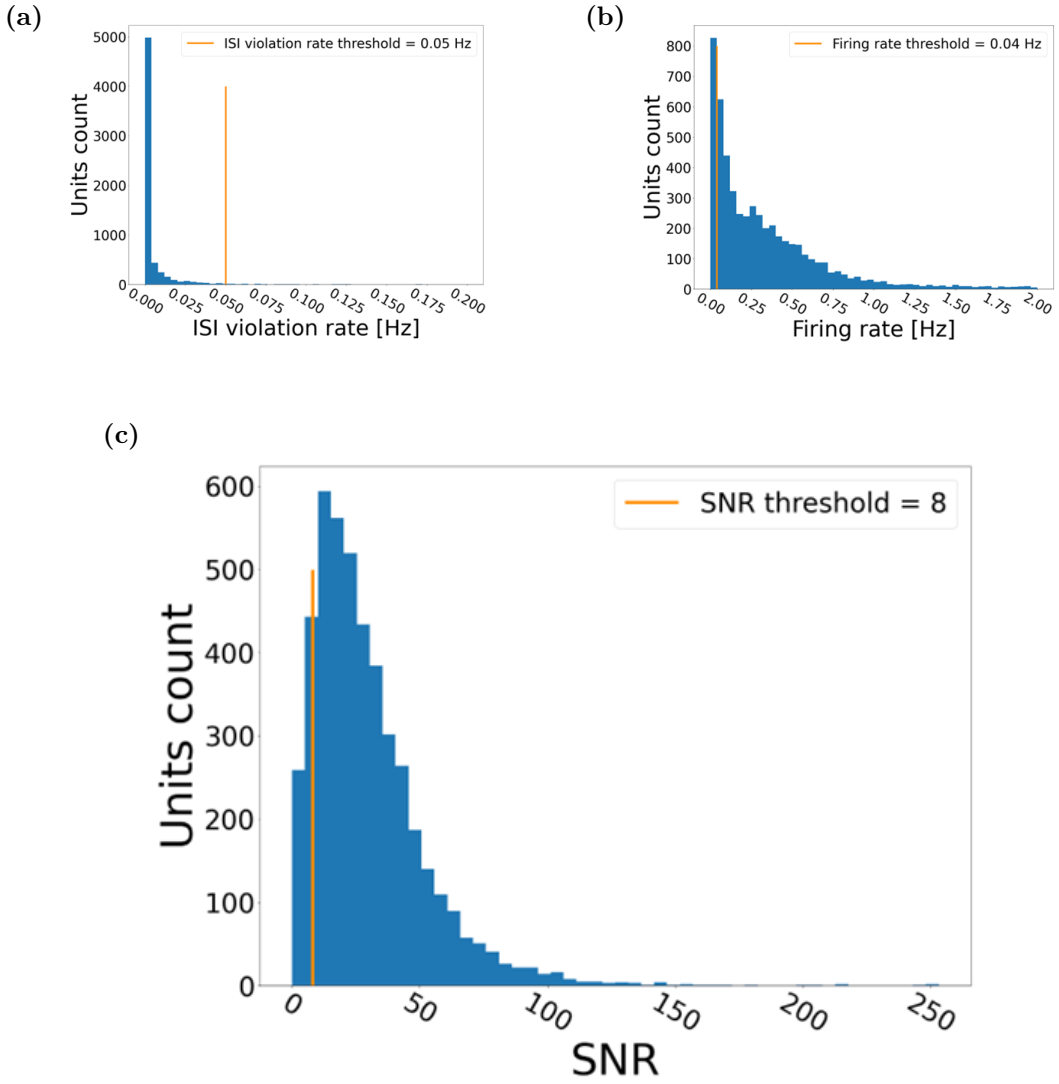


Figure 3.5: Feature distributions for the curation procedure. In orange, for each distribution, we have the chosen threshold for that curation parameter. (a) Imposing the upper threshold of 0.5 Hz over the ISI violation rate removed $\sim 8\%$ of the total population. (b) The 0.04 Hz lower threshold for the average firing rate (amounting to ~ 50 spikes) removed $\sim 15\%$ of the remaining units. (c) The final curation step removes $\sim 12.5\%$ of the remaining units, with an SNR lower threshold of 8. Eventually, the procedure leaves us with 768 curated units.

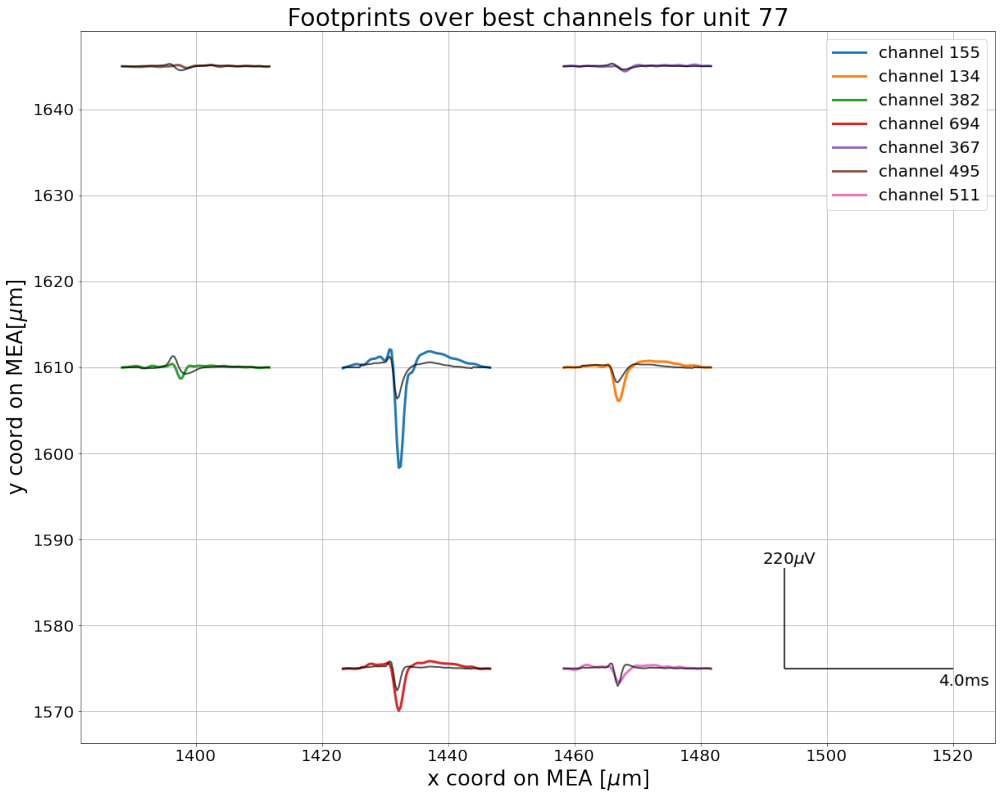


Figure 3.6: Spatiotemporal footprint of u77, (from MEA1k 4205). In black we show the template corresponding to each channel, while the colored waveforms are the footprint waveforms. The difference between the trough over channel 155 for the cutout and the template seems to indicate that the template is missing part of a potential sink, which is usually a tell for a split neuron.

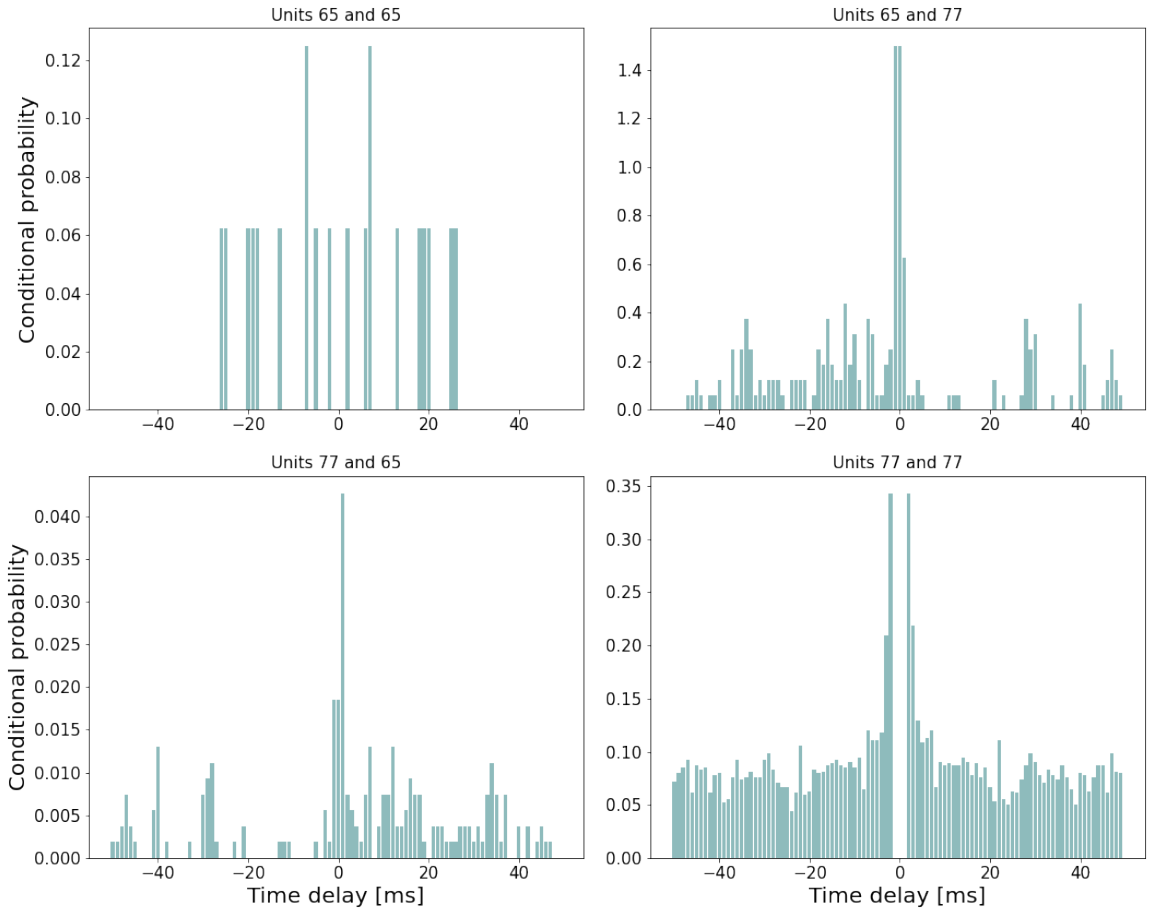


Figure 3.7: Cross-correlograms between u65 and u77, (MEA1k 4205). Each bin has a width of 1 ms, so that the cross-correlogram spans a period of 100 ms (50 ms per side) around each reference spike. From the top-right figure, we see a high correlation between the two units, with a correlation in the immediate vicinity of the reference spike that surpasses 1 (obviously this cannot be considered as a conditional probability anymore)

3.1.2 Low dimensionality feature distributions

In the process of establishing the best features to perform a classification, we first analyzed some of the distributions over the features that we computed. Without having a ground truth data set, this constituted the first attempt at classifying the neurons in a way that would then allow to infer the physiological differences between the two classes [18][20]. We separate the results regarding this first approach in either waveform features or time-series features. During the observation of the results, we will see that no distinction can be made between different modes, or families, with respect to distributions over only one feature. While at the start we also show distributions regarding the unlabelled data set, the majority of the focus for the exploration was be on the ICCS labelled one. This was because having labelled units gives us more insight about the possible significance of diversification in the distributions.

Waveform features

We tried to observe possible patterns or modes in the distributions corresponding to different features tied to the waveform of the putative neurons. If not specified otherwise, the waveforms considered for the computation of these features are the averaged cutouts over the best channel for the single unit: that is, the footprint waveform over the best channel. For the computation of the distributions we created a data set containing data coming from different chips. In (3.8) we can see the different features that we studied in relation to the footprint waveform of the single putative neurons.

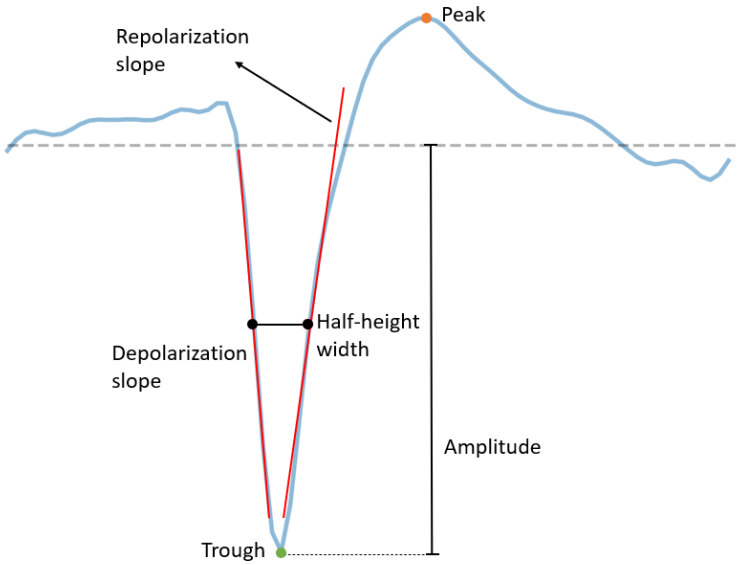


Figure 3.8: Image showing what the different waveform features computed were. As we can see both depolarization and repolarization slopes were calculated around the half-height of the trough (taken from the 0, here indicated by the dotted line), same as the width of the spike, and the amplitude, which was mainly considered as a parameter for curation, and was considered as the difference between the zero and the trough.

Starting with the depolarization slope, in Fig(3.9a) we can see the corresponding distribution of our units, which is largely peaked for values around $\sim -2.9\,\frac{\mathrm{mV}}{\mathrm{ms}}$. Let's focus our attention on the units that had values of the depolarization with absolute value larger than $10\,\frac{\mathrm{mV}}{\mathrm{ms}}$, composing the 5% of total population, which are shown in orange in the total distribution in Fig(3.9a) and whose focused distribution is in Fig(3.9b). As we can see, also from Fig(3.9d), these units showed no peculiar pattern in their distribution. Moreover, looking at Fig(3.9c), we see that these extreme-case units almost exclusively belonged to the ones recorded via the HD-MEA labelled as 4171, which was also the main source of our available data, with more than 3000 units surviving the semi automatic curation, and 575 the second round of curation. Taking into consideration the distribution represented in Fig(3.9a), we see that no separation between different modes could be observed. Therefore, the depolarization slope was not representative of a different behavior between excitatory and inhibitory neurons and could not be used as a feature by itself for classifying our units data set into different classes.

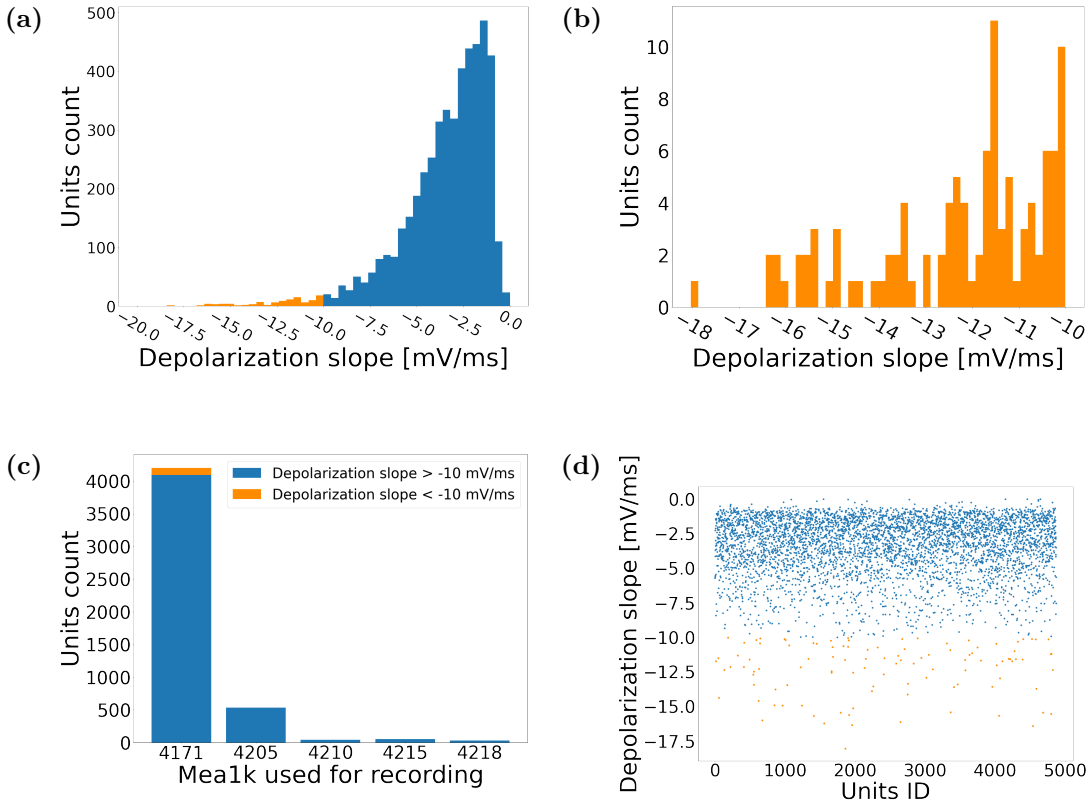


Figure 3.9: Figures relative to the depolarization slope distribution of all of the sorted units. The whole set of units, scattered among chips as visible in (c), was distributed with respect to the depolarization slope (a) with a peak around $\sim -2.9\,\frac{\mathrm{mV}}{\mathrm{ms}}$ and with no visible separation between families, as can also be seen from (d). In (a) in orange we see units falling below $-10\,\frac{\mathrm{mV}}{\mathrm{ms}}$, whose specific distribution is in (b) and (c) shows in orange that the chip that recorded most of them is 4171. The units count in (c) is relative to the total number of units before curation. In (d) we see the labelled units with the corresponding depolarization, the ones falling below $-10\,\frac{\mathrm{mV}}{\mathrm{ms}}$ in orange.

From Fig(3.10) we can see that the average footprint waveform over the best channel for units belonging to the population with highest depolarization slope (orange), with

absolute value larger than $10\,\frac{\text{mV}}{\text{ms}}$, had a much deeper trough than those that fell under this value (in blue). This could be expected: given that the half-height width of the spikes varied just slightly, and with it the depolarization time, the depth of the trough was mainly determined by the depolarization slope. Among the footprint waveforms with higher absolute value of the depolarization shown in the background of Fig(3.10) (thin orange lines), we see one with a huge peak before the depolarization. We will discuss this more in detail below, when talking about the repolarization.

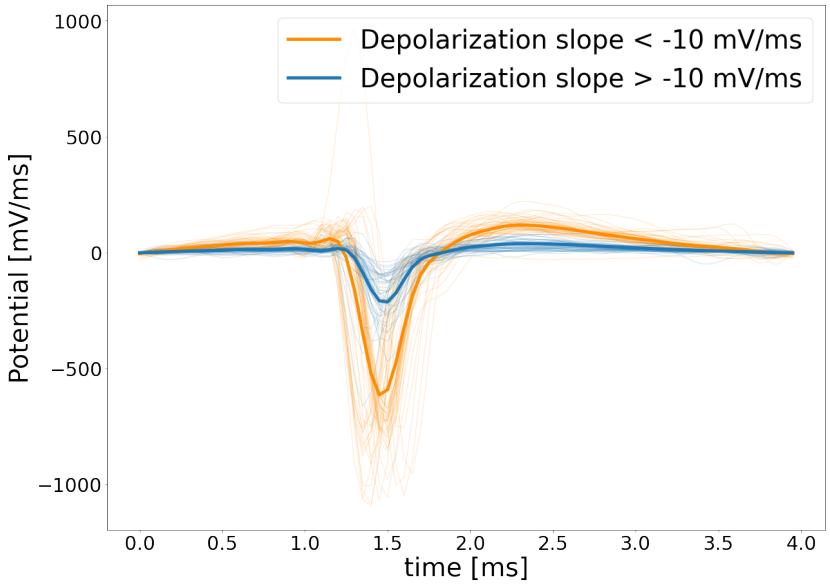


Figure 3.10: Comparison between waveform footprints, for units having depolarization slope larger or smaller than $-10\,\frac{\text{mV}}{\text{ms}}$, which represented the threshold for the bins with the highest population in the distribution in 3.9a. The average spike for units in the corresponding population (orange) has a smaller trough than the one for spikes outside of this population (blue), as to be expected.

Looking at the distribution according to the repolarization slope shown in Fig(3.11a), we can see that it was not a visibly bimodal distribution: we could thus deduce that it would not be possible to define a classification machine based on the repolarization slope as a single feature. In Fig(3.11c) we see that there was no pattern in the distribution. In this figure we also represented, in orange, the units belonging to the family that had depolarization slope with absolute value larger than $10\,\frac{\text{mV}}{\text{ms}}$: as we can see they all showed also a high repolarization slope, as to be expected, with the exception of a few on the rightmost side which had lower repolarization slope. We extracted a few of these apparent outliers, and plotted them in Fig(3.11b). We can observe that these spikes tended to have very large peaks before the trough, which is a kind of non-physiological behavior or a structure usually attributed to dendritic spikes, in which we have no interest, and a normal behavior after the trough. These outlying units were likely artifacts of the spike sorting, and represented only a very small part of the total population ($> 0.95\%$).

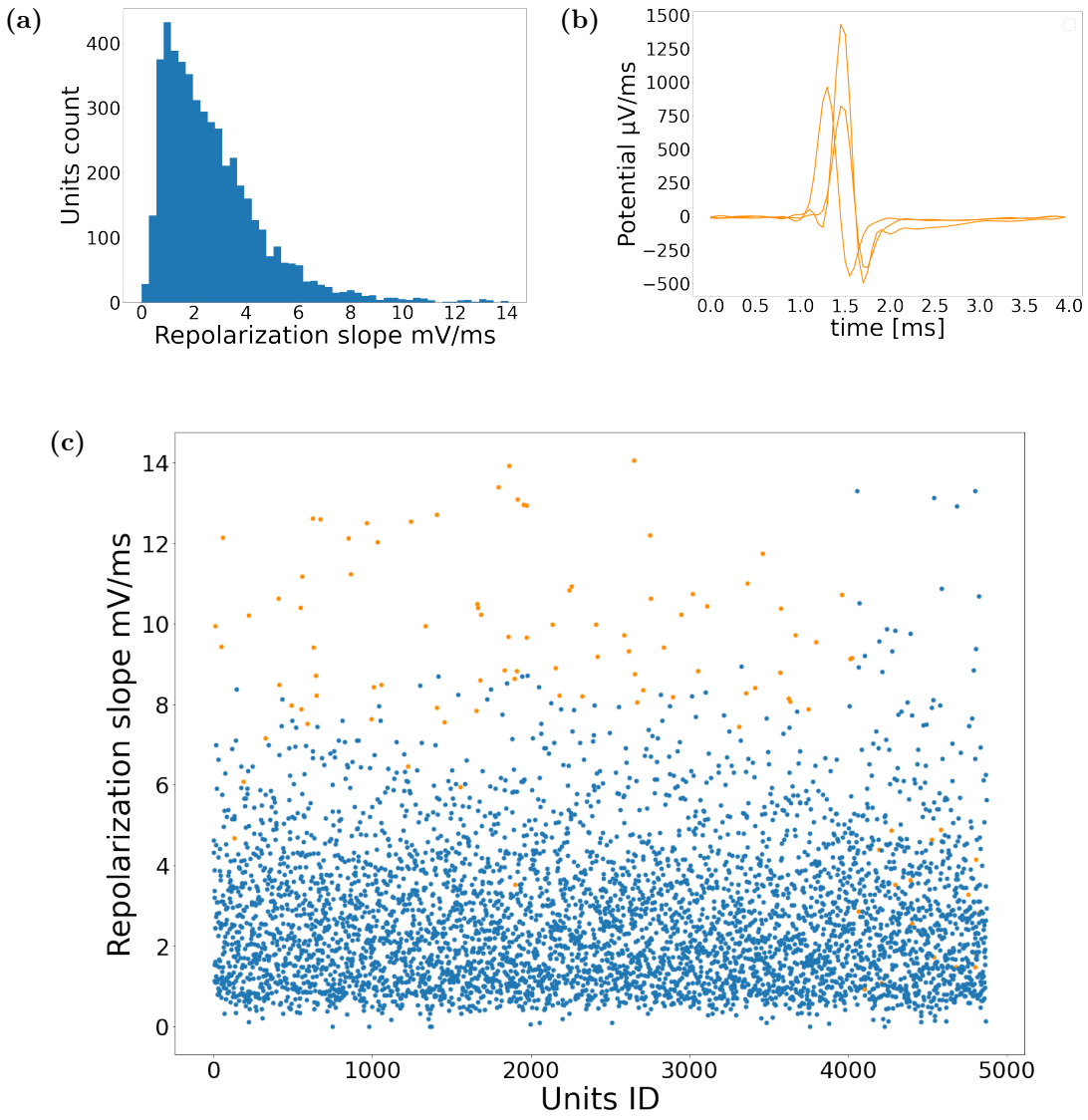


Figure 3.11: Figures relative to the repolarization slope distribution of all of the sorted units. In (a) we see the distribution of our data set with respect to the repolarization slope, which did not show any kind of bimodal behavior, nor any apparent outlier. In (c) we have the distribution of the single units, where the units considered specifically in the depolarization slope discussion and shown in Fig(3.9b) are in orange. Some of these units, on the right side of the plot, showed a strange behavior with smaller values of the repolarization slope than we would expect. We extracted three representative outliers and plotted them in (b), from which we could see that they had an high peak before the trough, considered as non-physiological or, at most, to be expected from a dendritic spike, in which we had no interest.

For both depolarization and repolarization slopes we checked the behavior of the ground truth data set established via spike transmission probability (STP) method, for which we knew whether each unit was an excitatory or inhibitory one. As can be seen in Fig(3.12), the ground truth data set confirmed that there was a strong correlation between depolarization and repolarization slopes. Neither of the features showed a well defined diversified distribution behavior between putative excitatory (red) and putative

inhibitory (light blue), as can be observed from the kernel density estimates on the top and right of the figure.

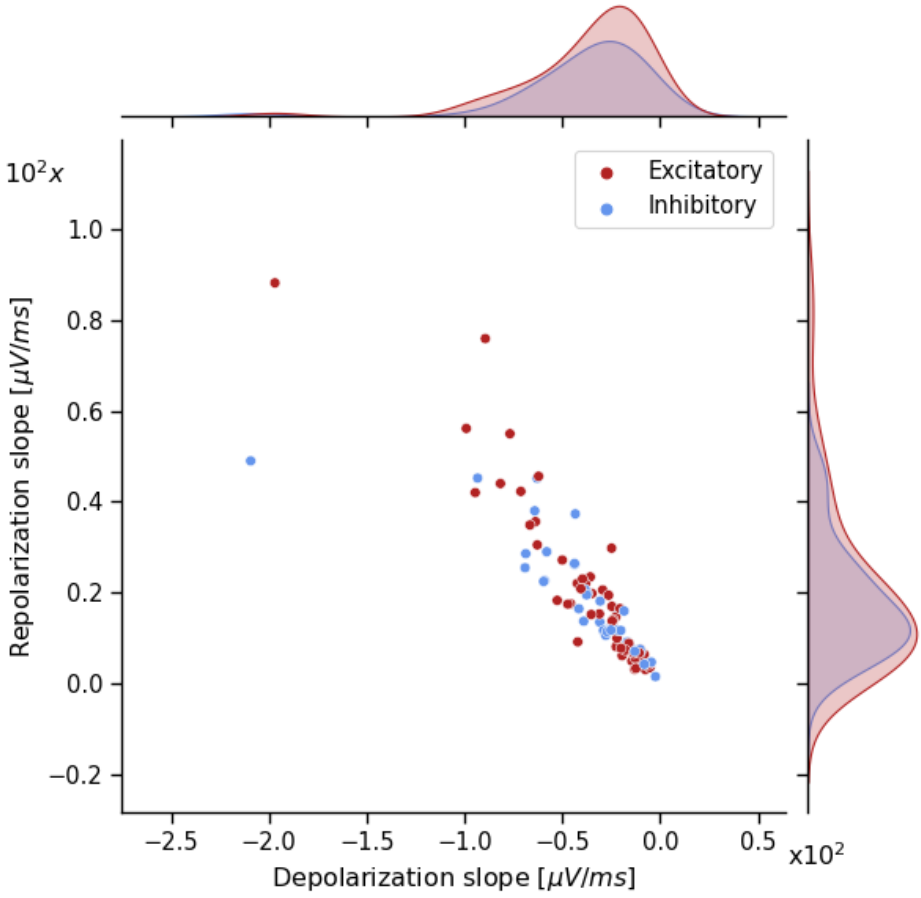


Figure 3.12: Depolarization and repolarization slopes combined distribution. We can see depolarization and repolarization slopes are strongly correlated in the immunocyto-stained ground truth data set (putative excitatory and inhibitory units in red and light blue, respectively). From the kernel density estimates on the top and right of the figure, we can also see clearly that there is no bimodal behavior to be observed in regards of the slopes for putative excitatory and inhibitory units.

From the behavior shown in Fig(3.12) we could deduce that no classification could be implemented between excitatory and inhibitory units depending on depolarization and repolarization slopes over a single waveform (best channel waveform).

We can see the way our units are distributed with respect to the half-height width in Fig(3.13a): units were considered as outliers if they had half-height width larger than -0.7 ms (small red bins in figure), and constituted only 0.5% of the total population. On the left Fig(3.13b), the distribution without the outliers: as we can see there was no bi-modality, so once again the half-height width did not constitute a good feature to establish a diversification criterion as a standalone. In Fig(3.13c) we can see how the units were distributed individually: in orange we have the outliers from the depolariza-

tion slope analysis (Fig(3.9b)), which did not stand out in terms of half-height width; in red we see the outliers of the half-height width defined previously.

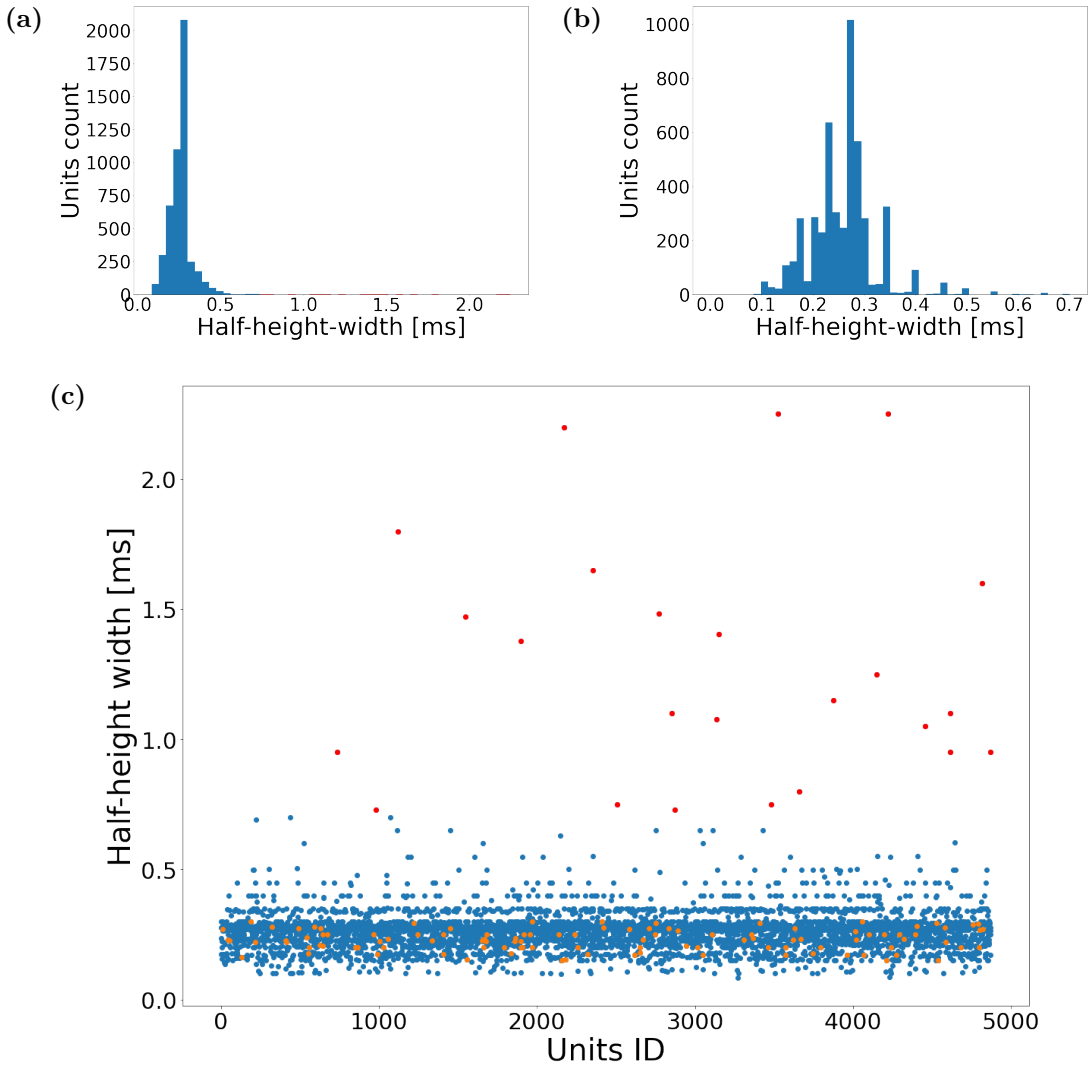


Figure 3.13: Figures relative to the half-height width distribution of all of the sorted units. In (a) the distribution over the half-height width for our whole population, where in red we represented bins with values higher than 0.7 ms. (b) shows how the distribution would look if units belonging to these bins, considered to be outliers, were removed. (c) shows how the single units were distributed, with units which were taken in consideration in the depolarization slope discussion (in orange) and the outliers for the current feature in red

We plotted the footprint waveforms over the best channel for a couple of these outliers which we reputed representative of the family, in Fig(3.14), to discuss the possible reasons why we had units with such a large half-height width. Both the footprint waveforms showed a highly non physiological behavior: Fig(3.14a) had low amplitude, even if above the curation threshold, and a too smooth and slow spike, while Fig(3.14b) did not show a well defined trough, and had a large positive spike. Both of these footprint waveforms belonged to putative neurons that could thus be considered as sorting artifacts, along with the other outliers described previously in this section, and

were removed from the data set for the high dimensional feature space exploration. We should remember that, due to the fact that we were aiming at a classification based also on these features, we were not afraid of losing bad units, which could make the unsupervised machine learning exploration much less reliable.

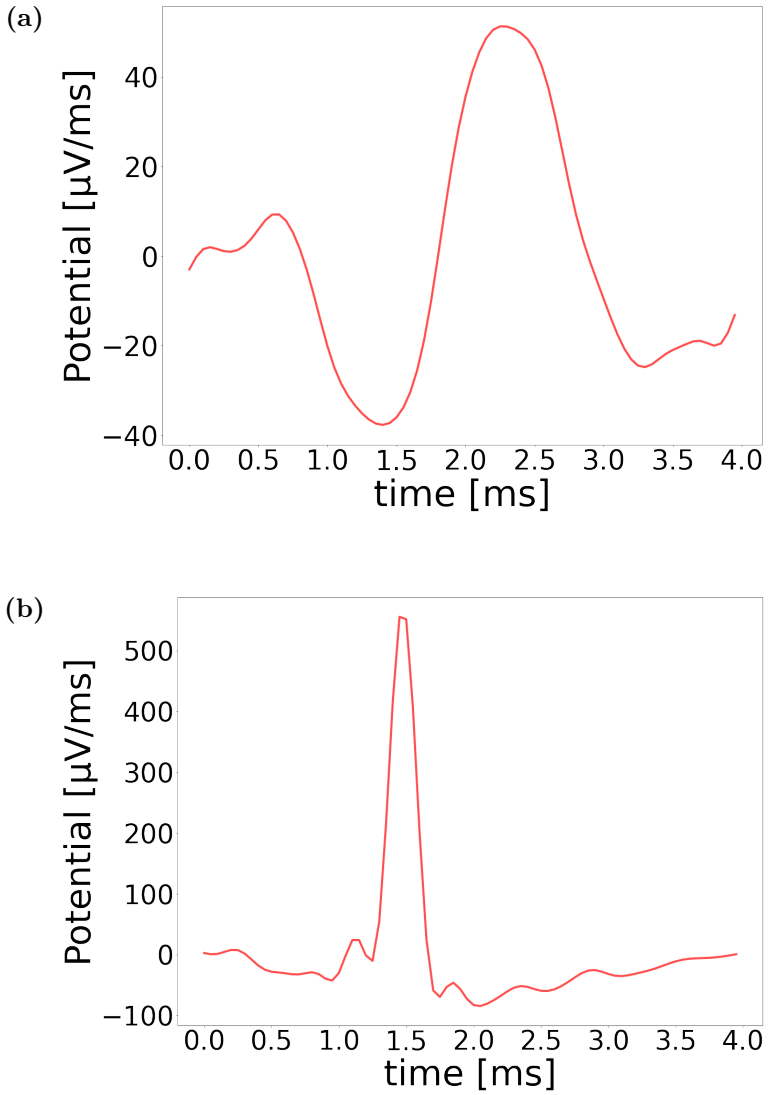


Figure 3.14: Waveform footprints for representative units belonging to the half-height width outliers family. Both footprint waveforms showed a strange behavior, very different from what we would expect from somatic or axonal spikes, and the corresponding units, together with the others belonging to this group, were removed from further analysis.

Additionally, we observed this feature, along with the trough-to-peak interval, for the STP ground truth data set, represented in Fig(3.15). While the two features don't show strong correlation, we see that for both trough-to-peak interval and half-height width the two classes show largely similar behaviors. The main exception is that the distribution for trough-to-peak interval is slightly more skewered toward larger values for excitatory units than inhibitory ones.

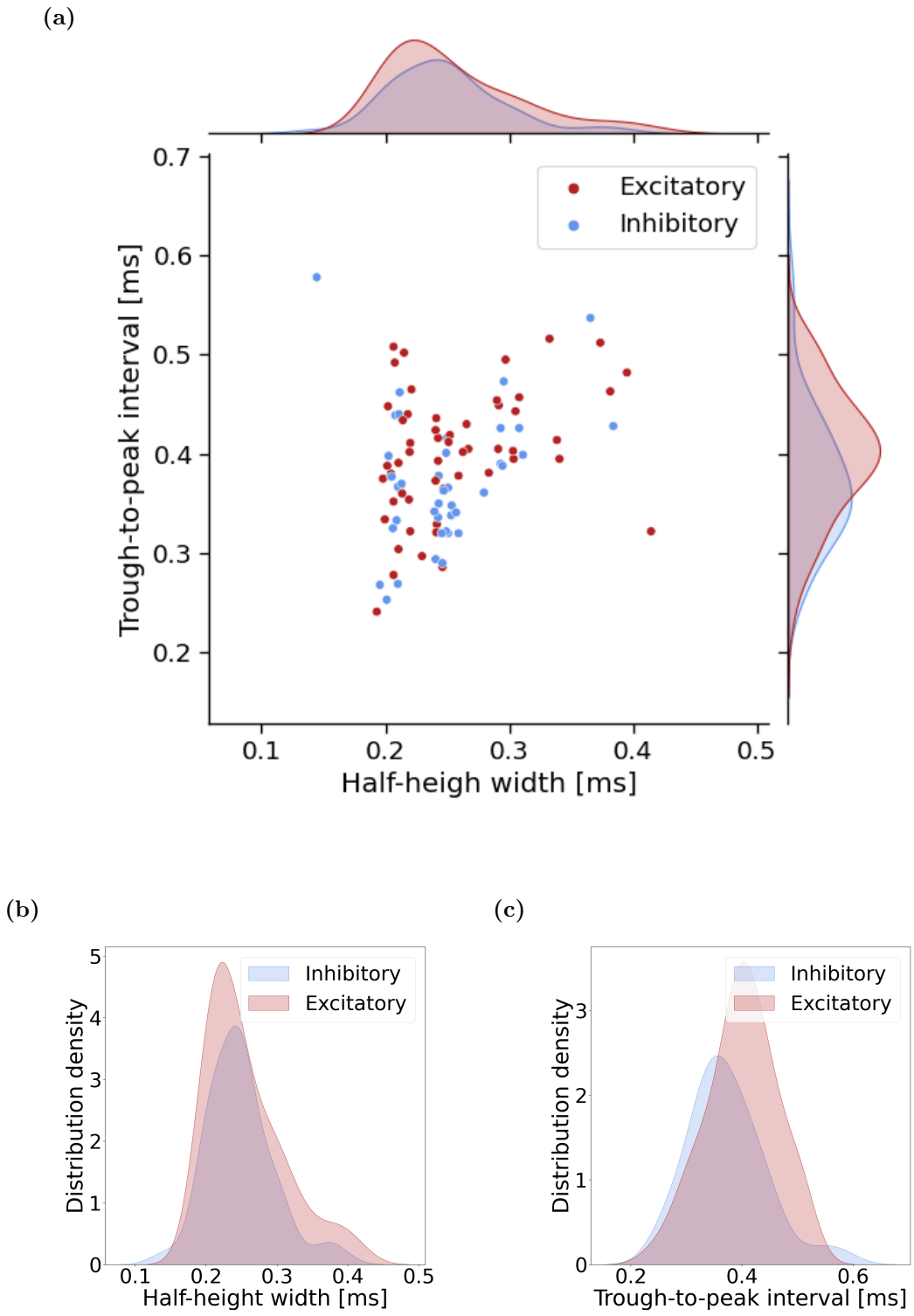


Figure 3.15: Distribution plots for E and I units relative to their half-height width and trough-to-peak interval. From (a) we see that there was not a clear correlation between the two features, and neither of them strongly separated the two classes. This is better seen in (b) and (c), where we saw that for both classes and features the density distribution were very similar. The only appreciable difference is that the distribution over the trough-to-peak was slightly more skewered towards larger values for the excitatory units than the inhibitory ones.

Time-series features

Along with waveform features, we also computed time-series related features, such as: average tonic ISI and firing rate, average phasic ISI and firing rate (were by phasic we mean the activity inside bursts, and tonic instead covers the whole spike train), inter-burst interval, number of spikes per burst, ans rise, burst and decay time constants for the autocorrelogram. No linear classification threshold was found over these features, neither in the low nor the high-dimensional features spaces. We computed multiple distribution plots in varying dimensions both for the STP ground truth data set and for the unlabeled data set. In no case were we able to determine a classification threshold between E/I putative neurons. In the following pages we will only show results from the labelled data set, as it is the one from which we could gather more information about the specific class distributions.

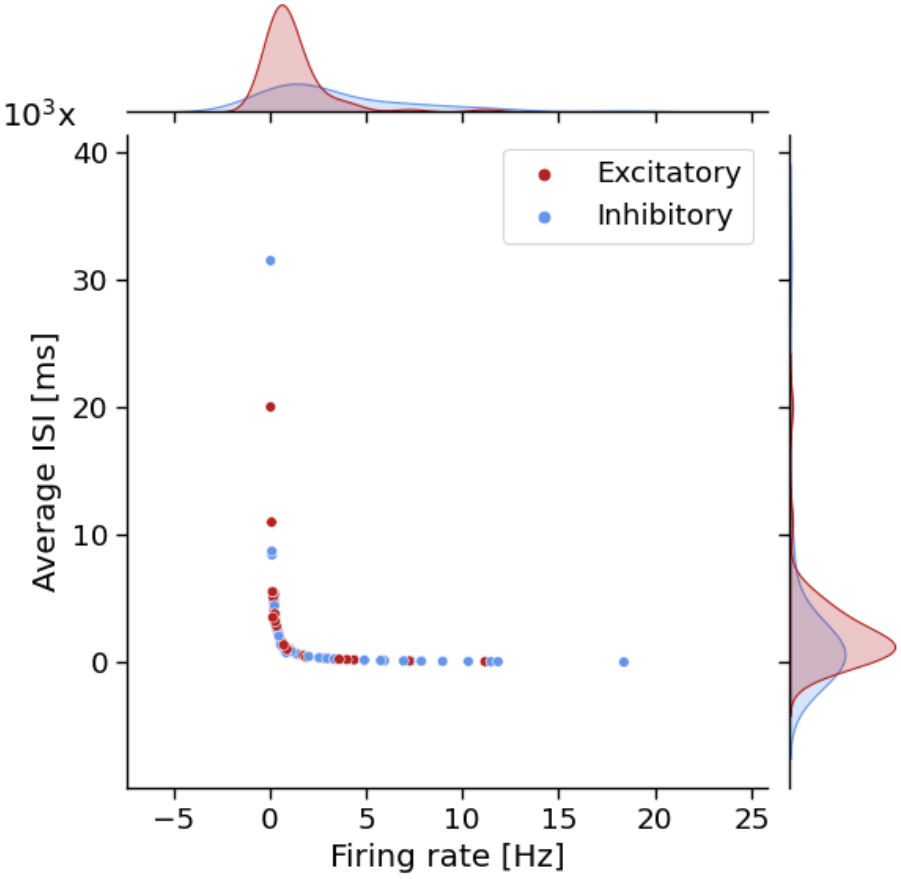


Figure 3.16: Distribution plots of E and I units with respect to total average firing rate and ISI. From the scatter plot we can easily see the hyperbolic relationship between the two featues, which was to be expected given the respective definition.

From Fig(3.16) we can see the characteristic hyperbolic relationship between firing rate and ISI for the labelled units, but observing both the scatter plot and the distribution

densities (on the top and right side of the figure), we see no clear separation between E and I classes. We also inspected the autocorrelogram for each unit. In order to reduce the autocorrelation behavior to a series of quantifiable parameter, we computed three different characteristic times for the 3-exponential fit of the autocorrelogram: rise, decay and burst times (respectively τ_r, τ_b and τ_d, described in 2.4.4) [30][31]. As these features are heavily correlated (~ 1 correlation across all of them), we only show the distribution of our units depending on one of them, in particular τ_d. As can be seen from Fig(3.17), the two population showed no differentiation when it came to autocorrelation behavior.

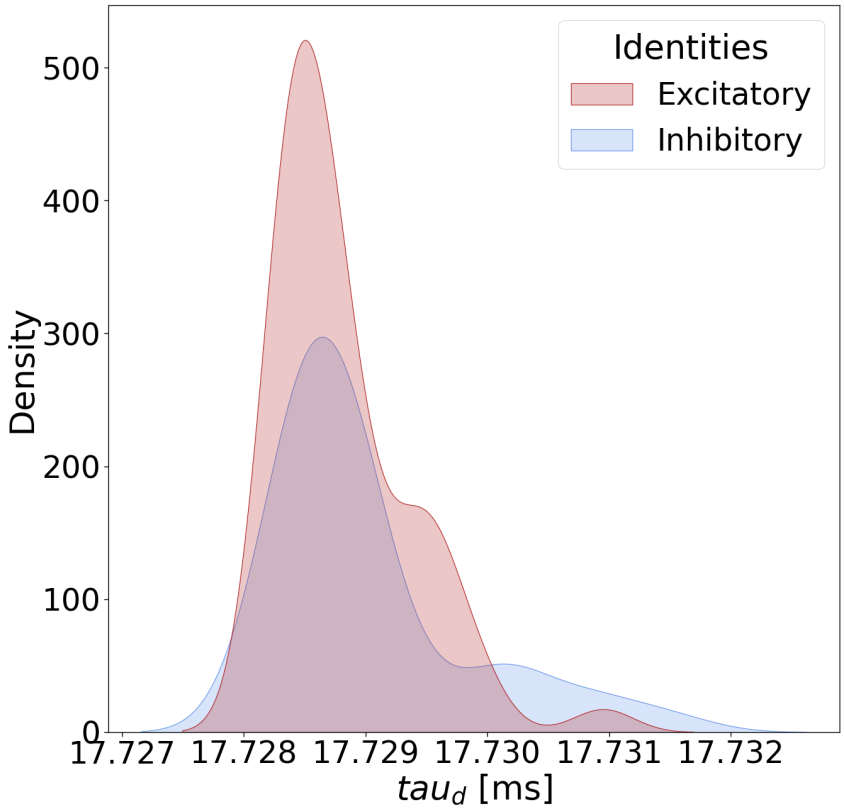


Figure 3.17: Distribution density for E and I units with respect to the τ_d parameter of the autocorrelogram. Only this parameter was shown as all of the characteristic times of the autocorrelation are linearly correlated. As such, one parameter is enough to convey the information needed.

We computed multiple features related to bursts, such as phasic firing rate, average phasic ISI, number of spikes per burst and IBI. Neuron can have very different behaviors inside and outside bursts, and we wanted to make sure we were characterizing the phasic behavior on its own. It's possible, by looking at Fig(3.18), to see that we could not determine a threshold to distinguish between E and I units depending on burst features

alone, in a low-dimension feature space environment.

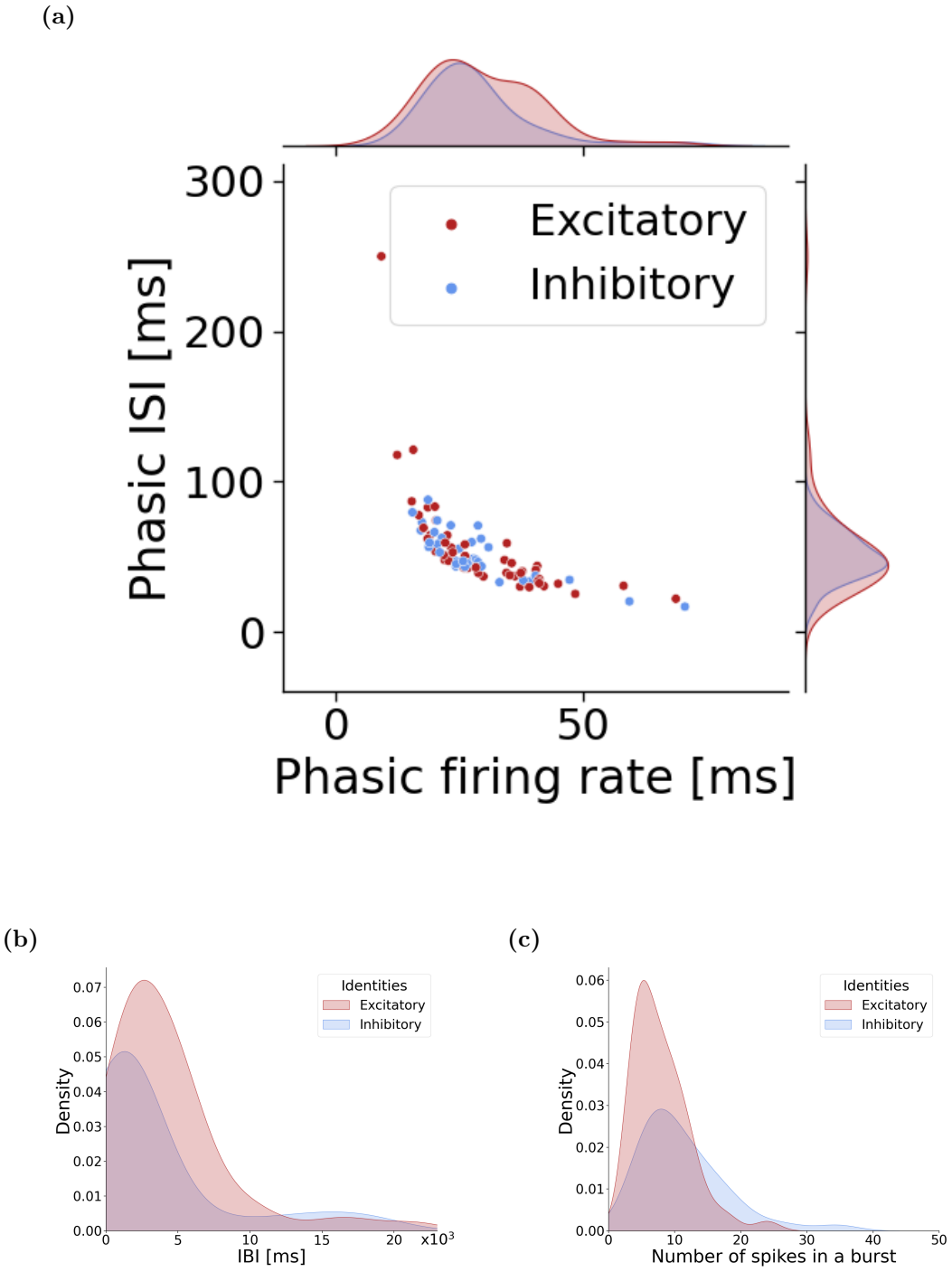


Figure 3.18: Collection of distributions for burst related features. No distinctive threshold could be determined from the computed features. As we can see, the distinctive hyperbolic relation between firing rate and ISI is still visible for phasic parameters.

3.1.3 High-dimension feature distributions

We have seen from the previous section that it's not possible to determine a classification threshold between E and I units over observation of low dimensionality feature distributions. Using algorithms such as PCA and UMAP, we projected higher-dimensional feature spaces on a 2-dimensional plot: our assumption was that E and I putative neurons could cluster in higher dimension, and using a dimensional reduction we could visualize this clustering in an embedded space with dimension equal to 2. In order to reduce the dimensionality, we used a linear and a nonlinear dimensionality reduction machines.

Linear dimensionality reduction

We performed a dimensionality reduction over different high-dimensional subspaces of the feature space, using principal component analysis. In figure Fig(3.19) PCA was applied separately to the waveform features computed on the footprint waveform for the best channel only (a), and to the time-series features (b). Neither subspaces contained enough variational information about the two classes for them to be separated in the embedded space. For the single channel waveform features, we had no separation in the embedded space between the two classes (represented in red and blue). The explained variance of the two principal components used for the plot was ~ 0.99, so almost all the information contained in this data set in the high dimensional space was transmitted to this 2-dim one. For the time-series features dimensionality reduction, we see that there was no clear separation between the two classes, even though I units seemed to be clustered more closely together than E units, which were instead diffused over the space. The explained variance ratio for the two principal components in this case was ~ 0.99.

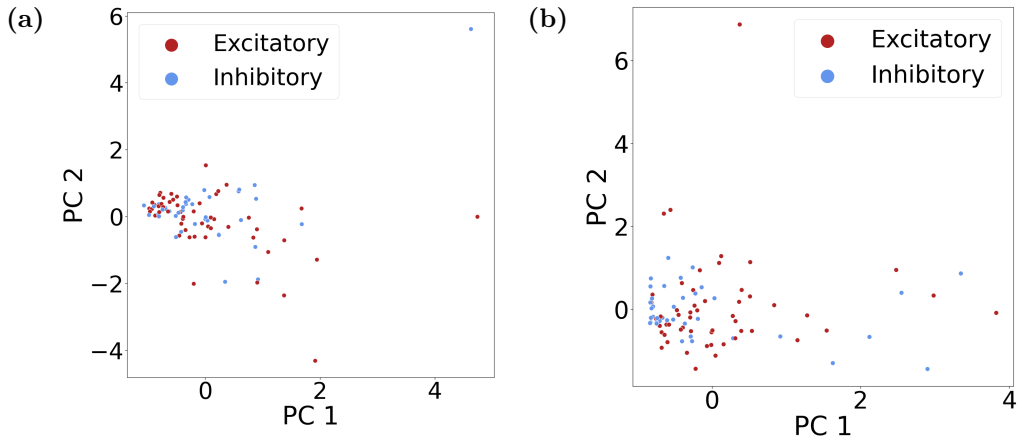


Figure 3.19: Principal component analysis applied to different subsets of our feature space. Between squared parenthesis we show the explained variance of the represented principal components (a) PCA applied to the waveform features of our putative labelled units, computed over the best channel only [0.99]. (b) PCA applied to all time-series and burst features [0.99]. As we can see, reducing linearly these features spaces to a 2-dim embedding space did not produce any separation in clustering between classes.

Both of the data sets reduced in Fig(3.19) were standardized before the dimensionality reduction. That is, their mean was set to 0 and variance to 1 over each feature. This was done to ensure that the principal component analysis was performed correctly, even with features that ranged very differently and have different units of measurement.

In Fig(3.20) we can see PCA applied to the sub-spaces of the feature space containing: single channel waveform features and time-series feature together Fig(3.20a), and burst related features alone Fig(3.20b). In both cases we see that no separation or specific clustering of the two classes could be discerned. Fig(3.20b) and Fig(3.19b) are very similar, due to the fact that they were computed over two subspaces that shared most of the features (the time-series space only has two, strongly correlated features more than the burst feature space), and PCA being a linear algorithm projected them via a similar transformation. This also gave us insight over the fact that the features that contained most variational information energy were those related with burst features, while tonic values were less relevant towards PCA.

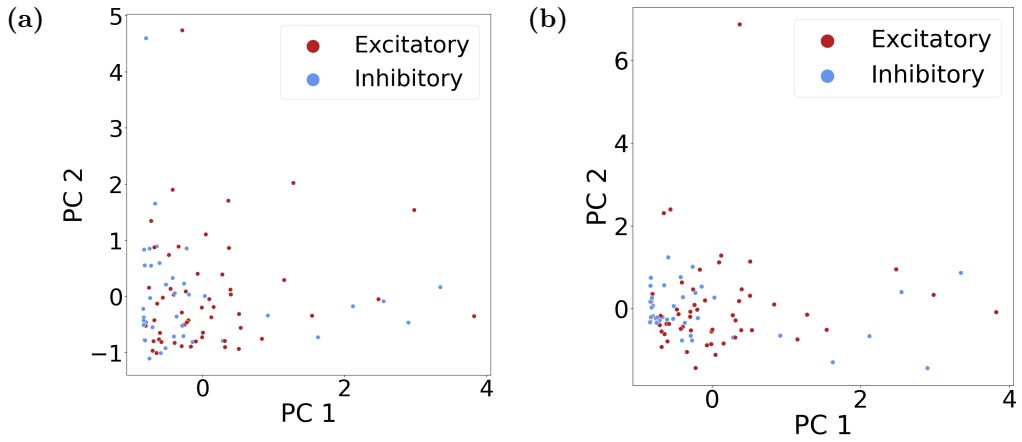


Figure 3.20: Principal component analysis applied to different subsets of our feature space. Between squared parenthesis the explained variance of the two principal components represented (a) PCA applied to the feature subspace containing waveform features computed only on the first channel and time-series features. [0.99] (b) PCA applied only to burst features [0.99]. As we can see, reducing linearly these features spaces to a 2-dim embedding space did not produce any separation in clustering between classes.

Eventually we applied PCA to the sub-spaces of the feature space with the highest dimension, or largest number of features Fig(3.21). In Fig(3.21a) the dimensionality reduction was applied to the multichannel waveform features, that is waveform features computed over 5 different channels among the best ones for each putative neuron (or unit) and ordered by latency. Each feature was scaled between the minimum and the maximum for each unit, to remove the bias due to the possible distance between channels and origin of the spikes that were used to compute the features. As we can see in this case the units were more spread out in the embedding space, but still no classification could be observed between them. With just two principal components (PC) in these cases we had ~ 0.38 explained variance ratio for the multichannel waveform features, which decreased to ~ 0.34 if we added the time-series features. In order for

the principal components of the two data set to explain at least 95% of the full variance of the data sets, we would have needed 9 PC for the multichannel waveform features and 13 if the time-series features were added.

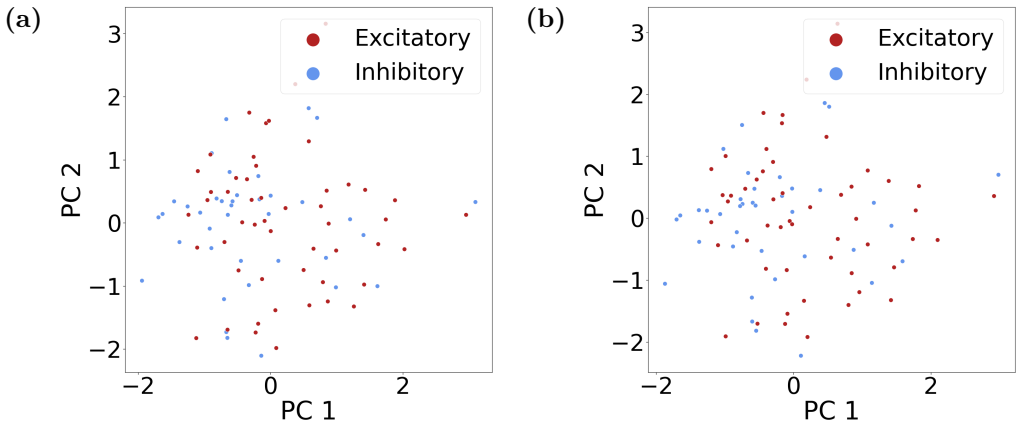


Figure 3.21: Principal component analysis applied to different subsets of our feature space. In the squared parenthesis, the explained variance of the principal components shown in the figures. (a) PCA applied to the feature subspace containing multichannel waveform features, computed over the 5 best channels and ordered depending on channel template latency [0.38]. (b) PCA applied to the subspace containing multichannel waveform features and all time-series features [0.34]. As we can see, even reducing linearly this high-dimensional feature space did not produce a good clustering between classes.

Nonlinear dimensionality reduction

We performed dimensionality reduction via UMAP (Uniform Manifold Approximation and Projection), a nonlinear algorithm which maintains the topology of the higher-dimensional space using a graph and edges between data points, instead of distance based metrics. This is particularly useful when operating on large dimensional spaces, which are subjected to the curse of dimensionality. If the data has a particular structure in the high dimensional space, this structure will be also represented in the 2-dimensional embedded space.

We first applied UMAP to the features space containing only waveform features computed over the best channel for each putative neuron (or unit), represented in Fig(3.22a). In Fig(3.22b) we can see the embedded space computed with UMAP for our labelled units based on their time-series features. Both data sets were standardized before applying UMAP. Neither of the figures showed any specific clustering of E and I units.

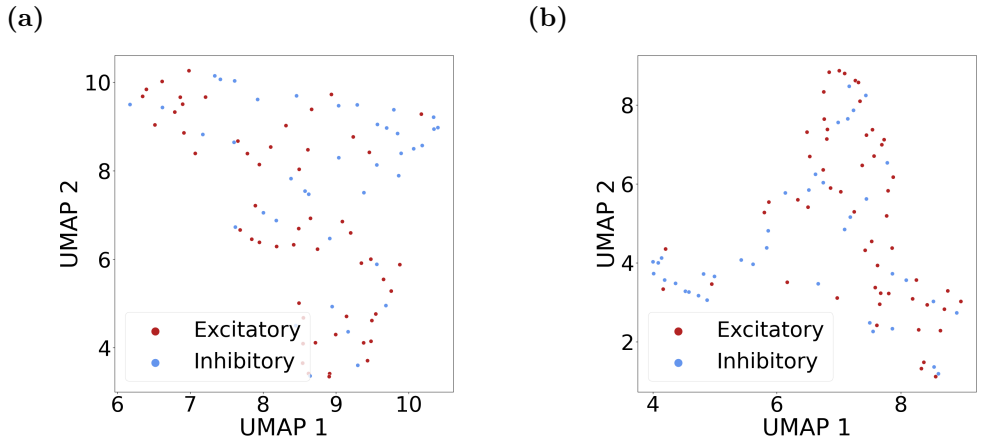


Figure 3.22: UMAP applied to subspaces of the features space for the labelled units. (a) UMAP applied to the feature space containing only waveform features computed over a single channel. (b) UMAP applied to the time-series features space. Even using a non-linear dimensionality reduction, which maintains the topological aspects of the higher dimensional feature space, no specific clustering was observed for the two classes.

In Fig(3.23a) we see the embedded space computed via UMAP, for the feature space composed by single channel waveform features and time-series features. No separation between the two classes was visible in this reduced space, and E and I units were equally distributed over it. Same considerations applied to the data set whose reduction is represented in Fig(3.23b), in which UMAP was applied to burst-related features only. To be noted that, while in the case of PCA the outlook of the population was preserved between the time-series features reduction in Fig(3.19b) and the burst features one in 3.20b, in the case of dimensionality reduction performed via UMAP this preservation was not sustained: this was due to the innate stochasticity and nonlinearity of UMAP as a process. If we look at Fig(3.22b) and Fig(3.23b), we can see that they had very different structures indeed.

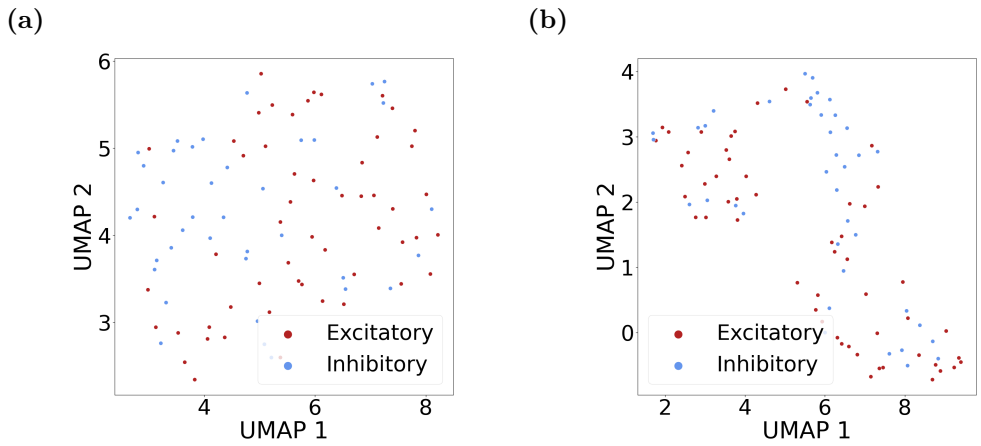


Figure 3.23: UMAP applied to subspaces of the features space for the labelled units. (a) UMAP applied to the subspace of the feature space containing all waveform features computed over the best channel for each labelled unit and all time-series related features. (b) UMAP applied to the feature subspace containing only burst-related features. No separation wass visible in these embedding spaces between the two different classes.

If there was a somewhat organized structure in the high dimensional space, such as a specific clustering of the units, UMAP would preserve this structure in the embedded space. When such a structure is not present, the stochastic nature of UMAP makes it so that each iteration will show a different distribution of the data points in the embedded space.

We then applied UMAP to higher-dimensional feature spaces: namely the ones containing multichannel waveform features, computed over different channels, and time series features in addition to these. In both cases, multichannel features where scaled between the minimum and the maximum for each putative neuron, to remove the bias due to the position of the neurons with respect to the recording channels. When applying UMAP, additional care had to be put on the preprocessing of the data, as UMAP was a strong tool and would represent any high-dimensional structure faithfully, even if it was an artifact or a bias-caused one. If, for example, we did not scale between the minimum and the maximum the multichannel amplitudes for each unit, UMAP would cluster together all units having recorded large amplitude spikes over their first channel, and the ones with a smaller amplitude in another cluster. Theses clusters would not be representative of the nature of the two classes or any physiological property of the neurons, as the amplitude was mainly influenced by the distance with respect to the recording electrode, for which there was still no model that can be used to account reliably for this behavior, without knowing the exact distance between neuron and electrode.

In Fig(3.24) we see the embedded space computed by UMAP over the multichannel waveform features: as for the other, low-dimensional spaces that we reduced, no clustering or separation was visible in the reduced space. Even by applying UMAP to the multichannel waveform features and the time series together, whose result can be seen in the reduced space in Fig(3.25), no classification seemed to be possible based on the topology of the high-dimensional space. This last data set was the one containing the

larger amount of information about our labelled units, and either it did not organize in class-specific structures in the high-dimensional space, or UMAP was not able to detect this structures.

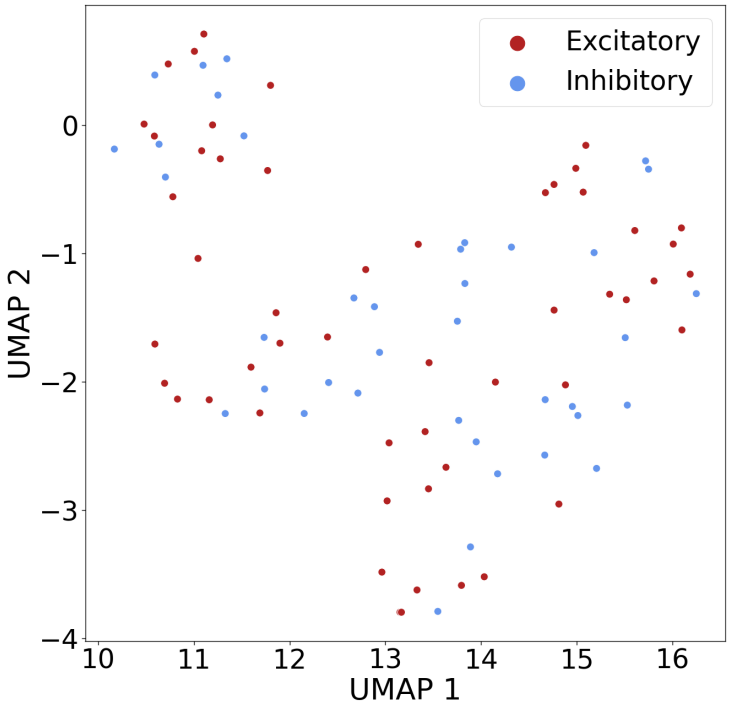


Figure 3.24: Dimensionality reduction performed via UMAP, of the high-dimensional feature space containing multichannel waveform features. These features were calculated for the footprint waveforms computed for the 5 best channels for each putative neuron, and ordered depending on the latency corresponding to the channel. For each unit and for each feature, the entries corresponding to the different channels were scaled between the maximum and the minimum value. As we can see, even in a reduced space that reproduced the high dimensional feature space topology, no separate clustering was possible. Multiple combinations of removal of different features were attempted, with no separation produced.

For each one of the plots shown in this section multiple different iteration were performed, with different sets of hyperparameters, namely the number of neighbors to be considered for the construction of the graph, the minimum distance in the embedded space and the metric to be used for the construction of the graph (both euclidean and manhattan distance were used, but above we only showed results for the manhattan distance), with no improvement towards the classification.

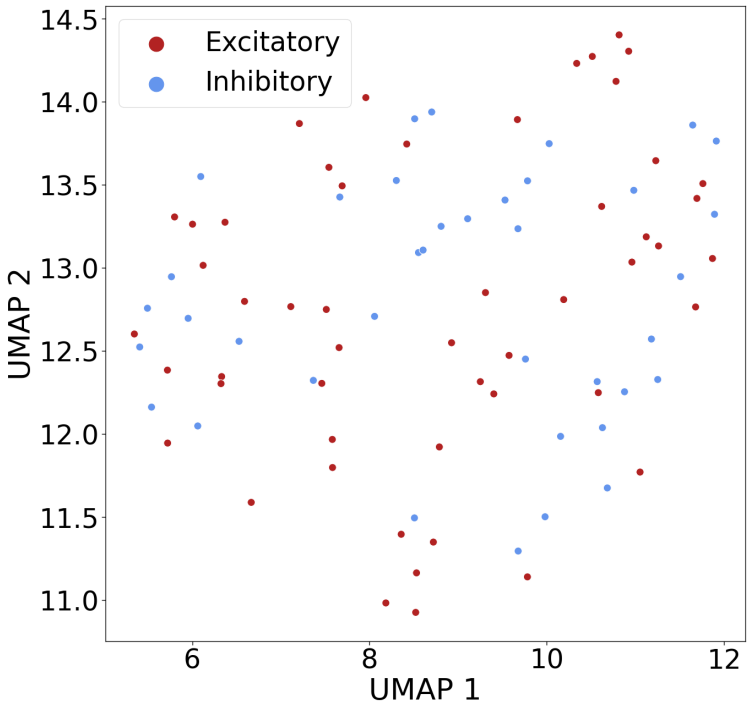


Figure 3.25: Dimensionality reduction performed via UMAP, of the high-dimensional feature space containing multichannel waveform features and all time-series related features. As we can see, even in a reduced space that reproduced the high dimensional feature space topology, no separate clustering was possible. Multiple combinations of removal of different features were attempted, with no separation produced.

3.2 Classification

After attempting classification via visual identification of structures or specific clustering in the data distribution, using dimensionality reduction machines, we used two different kinds of machine learning classifiers:

- Linear classifiers, such as linear SVM and logistic regression.
- Non-linear classifiers, such as kernel SVM, Ensemble decision tree classifier and gradient boosted tree classifier.

While most of the non-linear classifiers available require as many features as possible to work properly, linear classifiers are more susceptible to having too many variables involved. This is due to the fact that most linear classifiers involve some kind of distance metric to work, and thus are susceptible to the curse of dimensionality. For this reason, instead of using immediately all of the data available, we proceeded by starting with low-dimensional feature spaces for all classifiers (single channel waveform features, time-series features), and then passed onto trying to classify E and I through high-dimensional feature spaces (single channel waveform features plus time-series features, multichannel waveform features plus time-series features). In the following the corresponding classification results are going to be listed. Notice that burst features alone were not added to this part of the report, as the results were almost identical to that of the classification for all the time-series features.

Performance of the classifiers is going to be shown in form of confusion matrices. If not stated otherwise, the model was fitted and tested over the whole data set: as the amount of labelled data at our disposal at the moment of this report was scarce, splitting test and train data sets gave rise to a high degree of variability, depending on how the splitting was performed. The most relevant results, for this reason, are the metric scores listed in Tab(3.1), which were computed as average values of 5-fold cross validations of the models.

3.2.1 Linear classifiers

We first attempted classification via use of linear classifiers. Linear classifiers perform classification based on combining linearly the features contained in the data set, and extracting a decision function that assigns labels to the data based on some training data set. The reason for which linear classifiers are to be preferred with respect to nonlinear ones, if performance is similar, is that the decision function of linear classifiers is easily interpretable. The importance of the features towards classification is usually accessible in form of coefficients developed by the algorithm for the decision function. This makes linear classifiers a better choice for classification, if there is also an interest in inference through the results and in understanding the characteristic features of the different classes (as is our interest in this case).

The first linear classifier we used was the linear SVM. In Fig(3.26) we see the performance of the classification in form of confusion matrices. In Fig(3.26a) we can see the

confusion matrix relative to single channel waveform features. In this case the classifier was biased, with an acceptable level of precision only over E and high rate of misclassification for I units. In Fig(3.26b) we have the classification performed only over time-series features, in which the bias towards E units was even higher. In both of this classification attempts we saw the first weakness of these linear classification machines: in some instances, if the decision function could not find a way to separate linearly the classes, it just attempted at lowering the value of the cost function by classifying most units to the largest family (in this case, the E units one). It seems like the minima of the cost function did not coincide with a well performing classification attempt, but just with the least worst one.

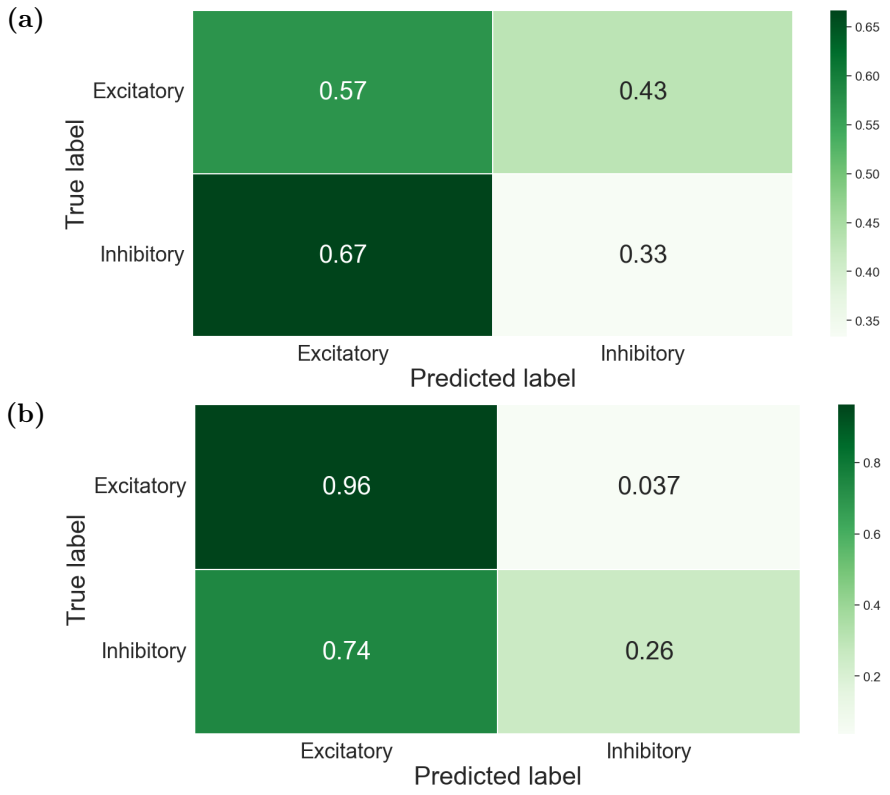


Figure 3.26: Linear support vector classifier applied to different subsets of the feature space. (a) Linear SVC applied to single channel waveform features ($C = 10$). Precision was higher for E than I units, as most I units were misclassified. (b) Linear SVC applied to time-series features ($C = 0.01$). Bias towards E units was very high, the classifier tended to classify most units as excitatory regardless of the training data set.

In Fig(3.27), the confusion matrices show reports of the performance of linear SVC applied to higher dimensional feature spaces, mixtures of both waveform and time-series features. The linear SVC applied to both single channel features and time-series features seemed to perform better than the previous case with separated feature spaces, as can be seen in Fig(3.27a). In Fig(3.27b), which shows the classification performance for multichannel waveform features and time-series it was instead even clearer the problem with this kind of linear classifiers if the dimensionality was too high. In this case, All of

the I units were assigned correctly, while more than half of the E units were misclassified as I themselves.

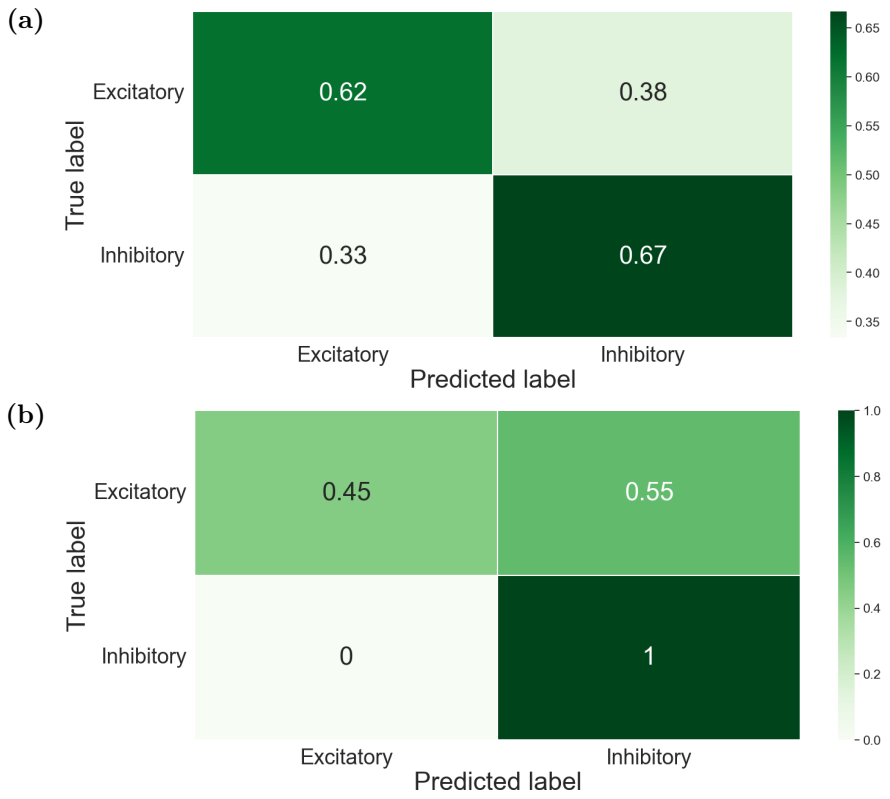


Figure 3.27: Linear support vector classifier applied to different, high-dimensional subsets of the feature space. (a) Linear SVC applied to single channel waveform features and time-series features together ($C = 1$). Precision is higher for I than E units. There is a certain level of balance in the accuracy over both classes (b) Linear SVC applied to multichannel waveform features plus time-series ($C = 10$). All of the I units in the test data set were classified correctly, but more than half the E units were misclassified.

We could examine the feature importance for the linear SVC applied to the single channel waveform features and time-series features, for which the average performance was better (3.27a). Here we computed importance via two different procedures: permutation importance and coefficients values. For permutation importance, the model was fitted and tested multiple times, while permuting the values of one feature among all putative neurons in the data set at each iteration. The importance was computed as the decrease in a certain classification performance score metric with respect to the unchanged case. In this instance the chosen metric was the F1-score, as our data set had a higher number of E units (54) with respect to the I (39), making the F1-score metric a better score than the accuracy.

In Fig(3.28) we see the permutation importances computed as described above. As we can see the most important feature in this case was the inter-burst interval, followed by the waveform slopes (they were highly correlated with each other, so it was understandable that the importance of one would imply that of also the other), and by

other burst related features. Other waveform features and tonic time-series features had zero importance, meaning that their being shuffled among units did not affect the performance of the classifier trained for Fig(3.27a).

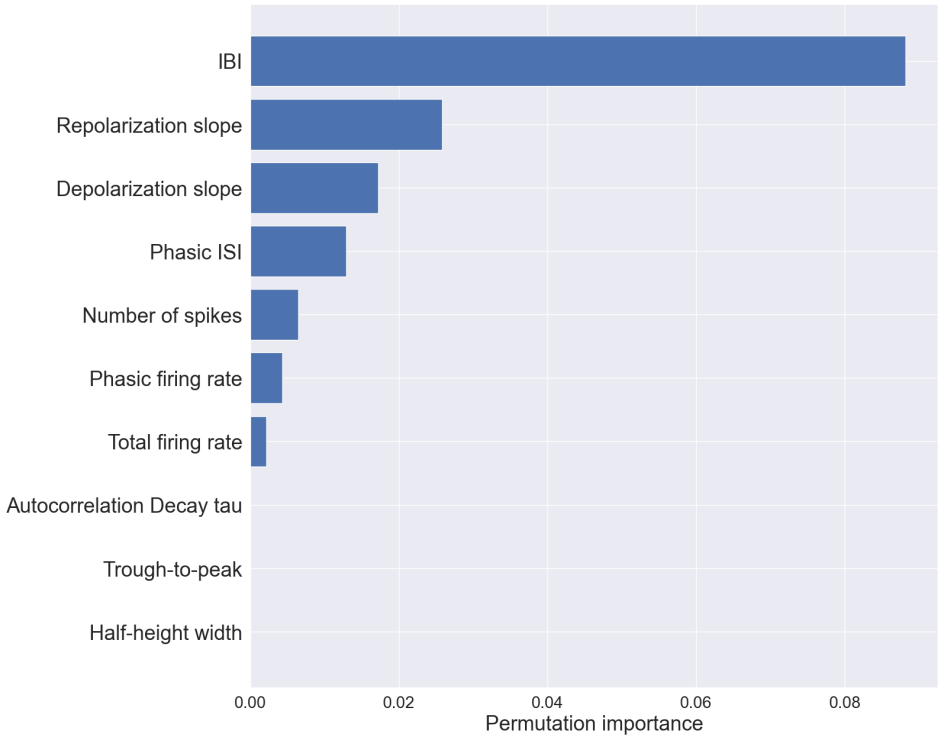


Figure 3.28: Permutation importance of single channel waveform features and time-series features, with respect to a linear SVC classifier.

Below in Fig(3.29) we have another set of feature importances, computed as the coefficients calculated by the linear SVC after being trained to classify for Fig(3.27a). These coefficients were directly tied to the classification function, and a coefficient with a larger absolute value indicated that the corresponding feature was more important towards classification. Even though the order and importances changed with respect to 3.28, the most important features remained those tied to phasic behavior of the units. From the perspective of coefficient importances, waveform features as a whole seemed to be less relevant than time-series related one, while the characteristic decay time of the autocorrelation gained importance, as well as the total firing rate computed for each unit. In this case, no feature had zero importance, meaning that no feature could be removed from the classifier while still maintaining the performance.

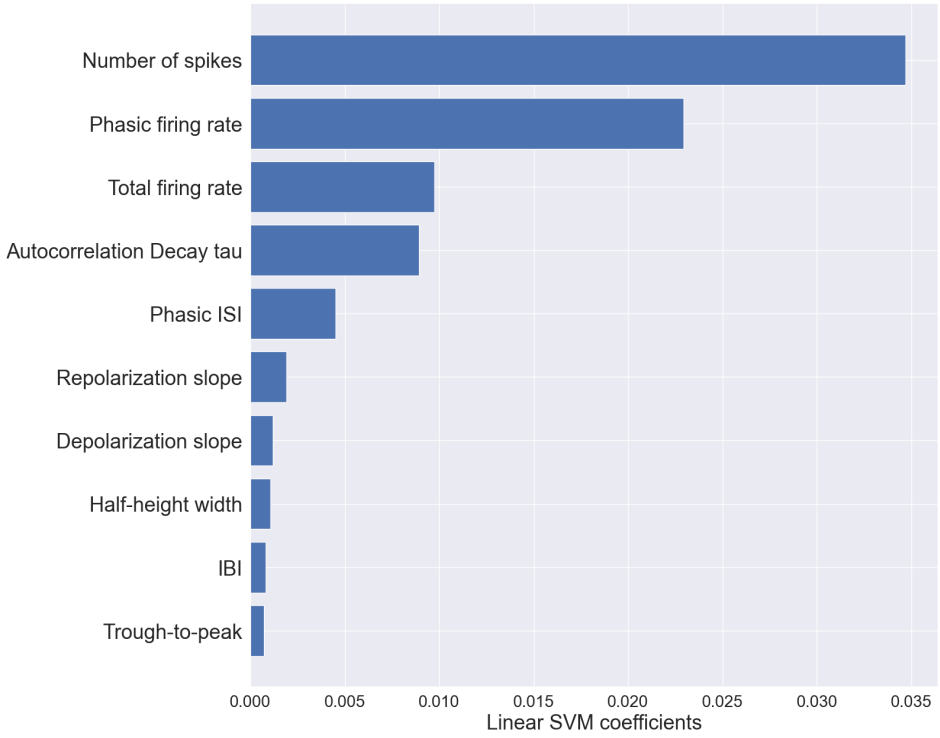


Figure 3.29: Importance of single channel waveform features and time-series features in terms of coefficient value, with respect to a linear SVC classifier.

As a second kind of linear classifier, we used logistic regression to classify between E and I units. The same feature sub-spaces were used, and the results are shown below. Classification performance was overall similar to the linear SVC, with a slightly more balanced accuracy for the best model (3.31a).

First we attempted classification using logistic regression over the two lower dimensional spaces containing single channel waveform features Fig(3.30a) and time-series features Fig(3.30b). As we can see, for single channel waveform features we obtained acceptable results only towards classification of E units, while the I units were assigned with equal probability to either class. For time-series only features we had instead an heavier bias towards E units, as most I units were classified as E and only 56% of E units were correctly classified. These results were overall similar to the ones for linear SVC, but we can see that in Fig(3.26b) the bias towards E was far more marked.

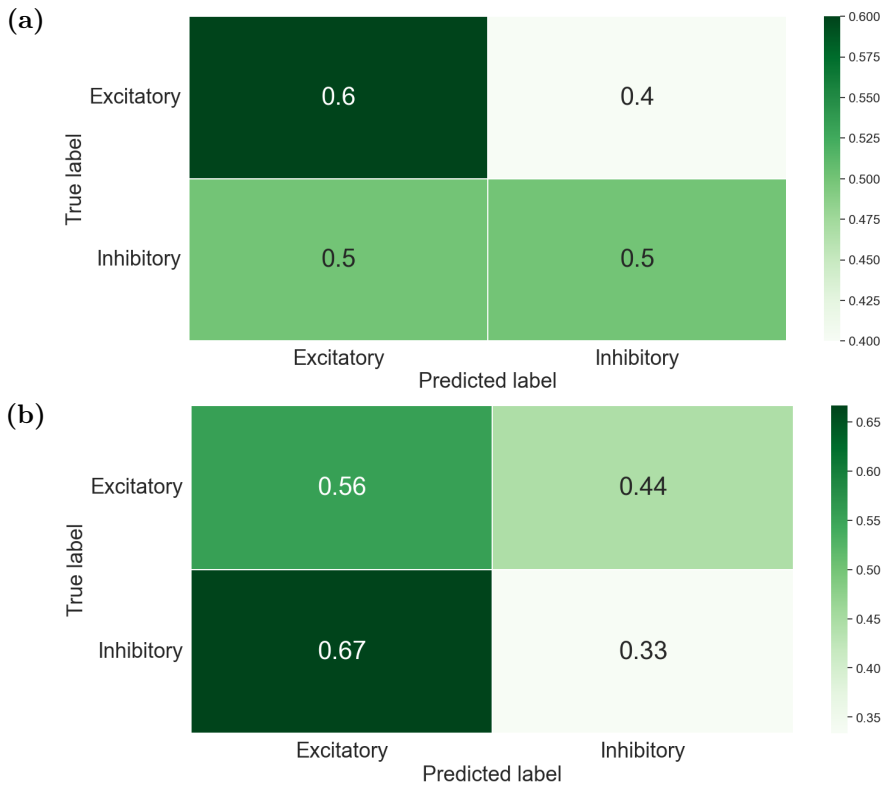


Figure 3.30: Logistic regression classifier applied to different subsets of the feature space. (a) Logistic regression applied to single channel waveform features. Precision was higher for E than I units, while classification for I units separated them equally in either classes. (b) Logistic regression applied to time-series features. We had a heavy bias towards E, as most of the I units are classified wrongly as E. Precision for E units is also quite low.

Next we proceeded, as with the previous method, to classify the units based on higher dimensional feature spaces. In Fig(3.31a) we see the best performing model, applied to a mixture of single channel waveform features and time-series features. In this case we had perfect balance between E and I units, no bias, and precision was pretty high for both (0.67 was the highest average accuracy seen up to now). Once we moved on to an even higher dimensional space, that of multichannel waveform features and time-series features, the performance deteriorated drastically. The bias towards E reinstated, precision was less than 0.5 for I and low for E as well. We could assume this to be due, once again, to the curse of dimensionality: to perform classification, logistic regression makes use of distance metrics (in particular euclidean distance), which performed badly in high dimensionality. The performance in this case was difficult to compare to the one of linear SVC, as in Fig(3.27b) the classifier was just classifying most units as I to reduce the cost function, without the actual training enhancing it.

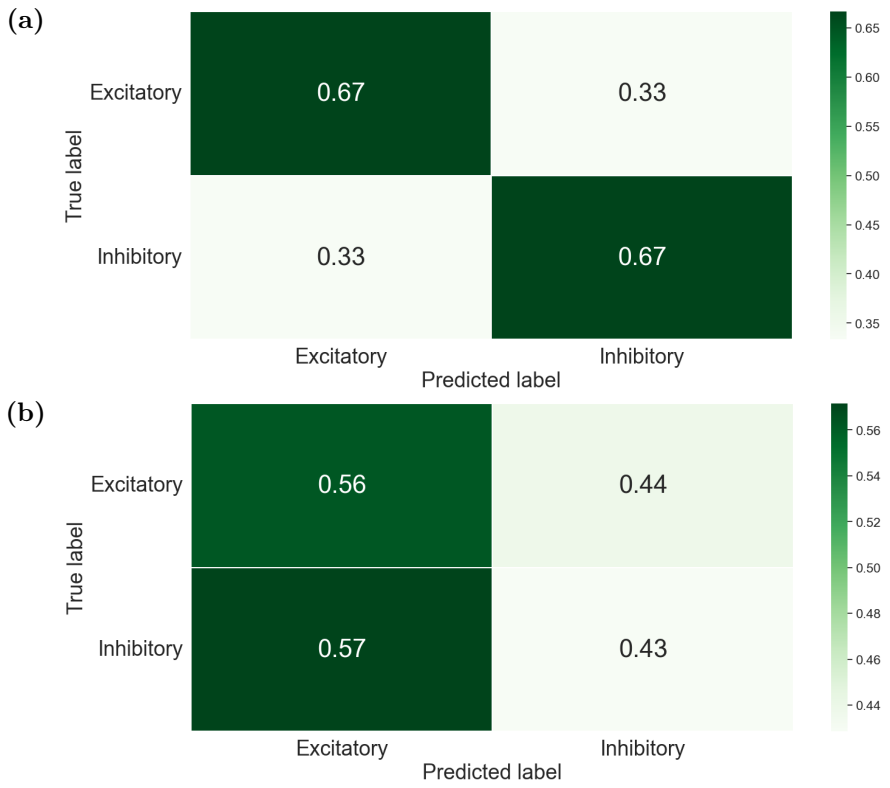


Figure 3.31: Logistic regression classifier applied to different, high-dimensional subsets of the feature space. (a) Logistic regression applied to single channel waveform features and time-series features together. We had an equilibrium between classification of E and I units, and precision was quite high (67%) for both (b) Logistic regression applied multichannel waveform features and time-series features together. The large increase in classifying parameters made classification worse for both classes, and more than half I units were misclassified.

Once again, we checked feature importance for the best performing logistic regression classifier, the one operating on the mixture space of single channel waveform features and time-series features. First we observed the permutation importance, in which multiple iterations of the training and testing of the model work in succession, permuting a different feature among units at each iteration. As we can see in Fig(3.32), the most important features were related to phasic behavior, followed by waveform features. This was consistent with what we observed for the linear SVC, and confirmed that the burst behavior of the units was tied with their class. As burst behavior was also expected to be related to neuronal function, this could correspond to a specific physiological function of E and I units.

As feature importance for logistic regression could be regarded in terms of model coefficients as well, we considered these values in 3.33. The most important feature was the number of spikes per burst, once again related to bursting behavior. Tonic activity or total activity time-series feature grew in relevance, such as total firing rate and characteristic decay time of the autocorrelation. All single channel waveform features had low importance.

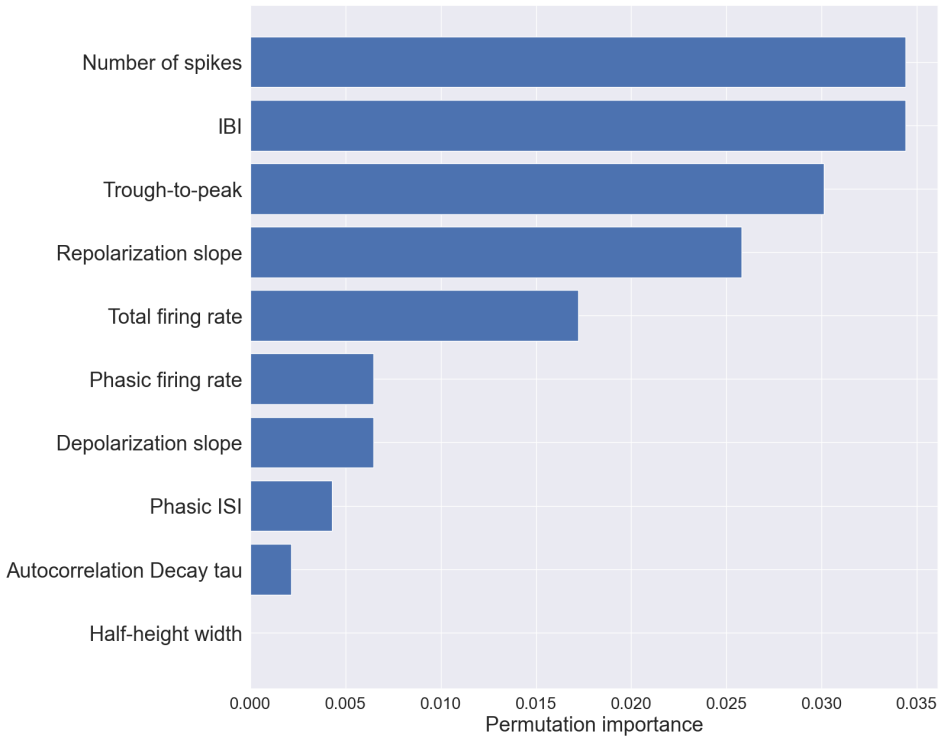


Figure 3.32: Permutation importance of single channel waveform features and time-series features, with respect to a logistic regression classifier.

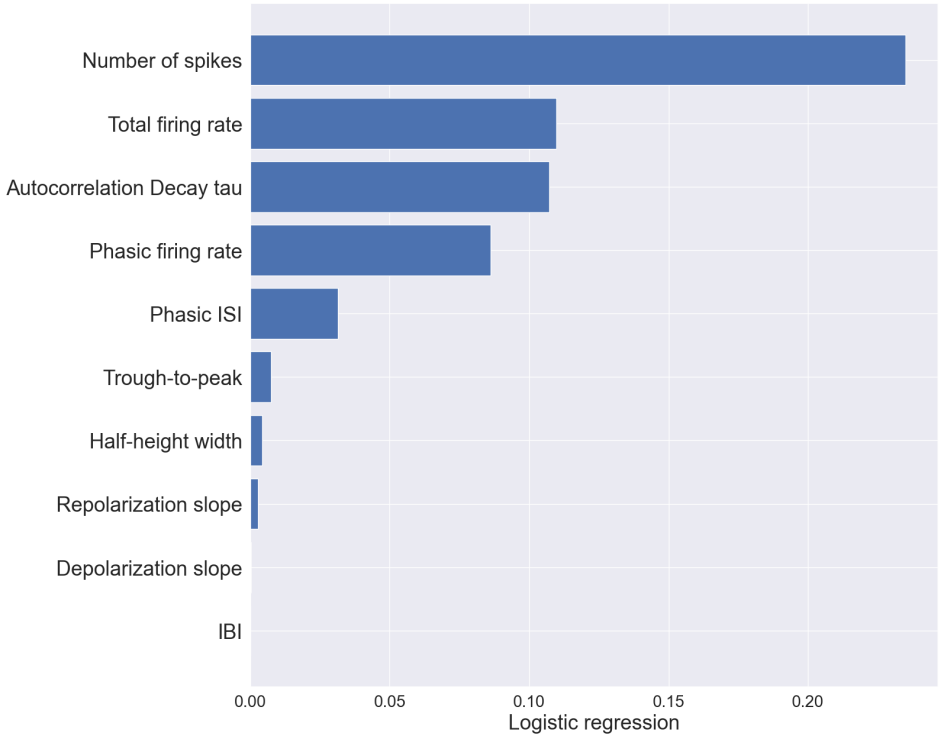


Figure 3.33: Importance in terms of value of the coefficients of single channel waveform features and time-series features, with respect to a logistic regression classifier.

3.2.2 Nonlinear classifiers

We applied nonlinear classifier to different subspaces of our feature space, expecting an improvement in performance, in particular for what concerned higher dimensional feature spaces such as multichannel waveform features plus time-series features. This improvement in performance was indeed observed, both over testing and training operated over the same data set and with cross-validation based metrics (can be seen in Tab(3.1)). When working with nonlinear classifiers, one must always be aware of the risk of overfitting, which is very high with these kind of models, and the lower level of interpretability. Most nonlinear models also require a very large amount of training and testing data to perform at the best: in particular the number of samples should always be much larger than the number of features involved. As this was not our case, we assumed that the classifiers could be improved vastly by expanding the available labeled data set.

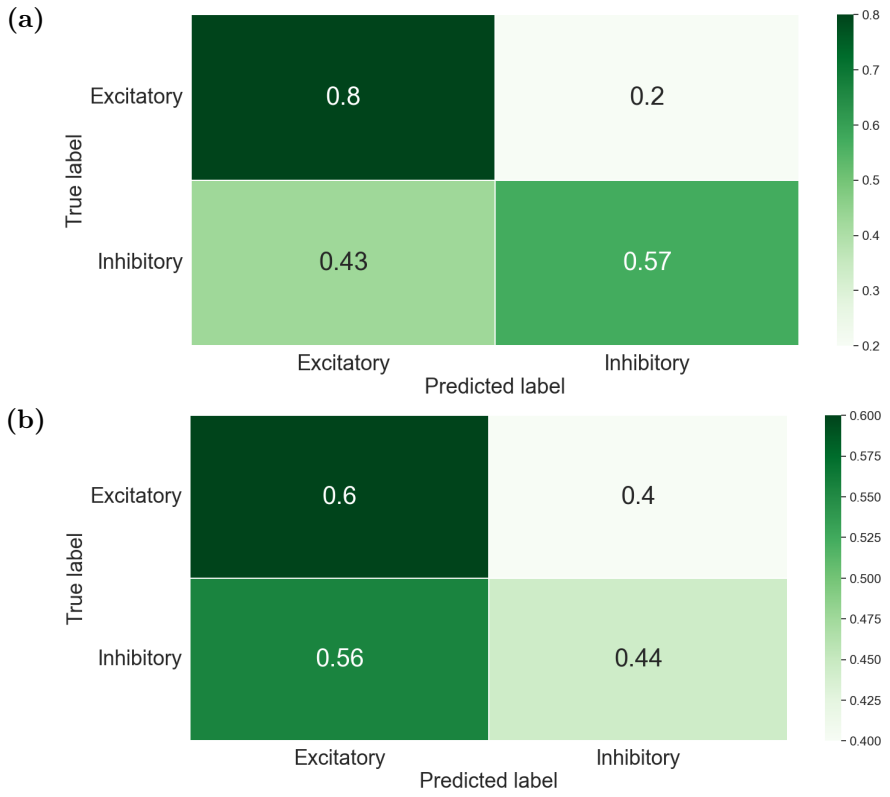


Figure 3.34: Confusion matrix for Kernel SVC with gaussian kernel applied to different subsets of the feature space. (a) Kernel SVC applied to single channel waveform features ($C = 20$), precision was higher for E units; for both classes the majority of the units were classified correctly. (b) Kernel SVC applied to time-series features ($C = 40$). The classification was biased towards E, with more than half of both E and I units being classified as E.

The first nonlinear classifier we used was a kernel SVC, with a radial basis function kernel. For each use of the kernel SVC, the optimal parameters for the model were determined via the use of the `GridSearchCV` function in the `sklearn.model_selection`

module in `Python`. A kernel SVC classifier uses a kernel function to expand the feature space, by adding a dimension via the kernel and correlated to the available variables. It had the potential of improving the performance with respect to a linear SVC, but as the underlying method was the same, we did not expect it to offer any improvement to the higher-dimensional feature spaces classification. In these cases we actually expected it to perform worse, as the addition of a dimension would only make the curse of dimensionality even more problematic.

First we used a kernel SVC over the single channel waveform features and time-series feature spaces, for which the performances in terms of confusion matrices can be seen in Fig(3.34). The best performance for this classifier was seen on the single channel waveform features data set, in which we obtained very high precision for E units. For time-series features the classification was biased towards E units, with more than half I units being misclassified.

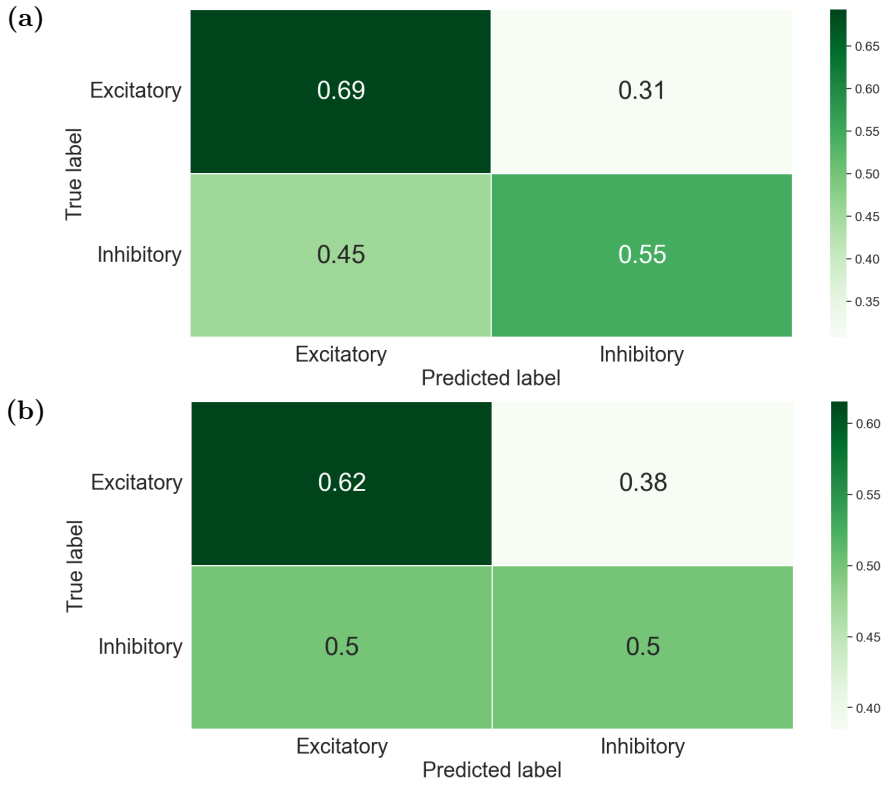


Figure 3.35: Confusion matrix for Kernel SVC with gaussian kernel applied to different, high-dimensional subsets of the feature space. (a) Kernel SVC applied to single channel waveform features and time-series features together ($C = 10$), precision was higher for E units; for I units, only 55% of the units were classified correctly. (b) Kernel SVC applied to multichannel waveform features and time-series features together ($C = 1$). I units were equally partitioned in both classes, while most (62%) of E units were classified correctly.

Applying kernel SVC to higher dimensionality feature spaces returned the expected results. In Fig(3.35a) the average accuracy of the classifier was lowered with respect to the linear SVC in Fig(3.27a), for the single channel waveform features plus time-series

features. In the same way, the accuracy over the multichannel waveform features plus time-series features in Fig(3.35b) decreased with respect to Fig(3.27b). This was to be expected, as the kernel SVC only added a feature to two spaces which were already high-dimensional.

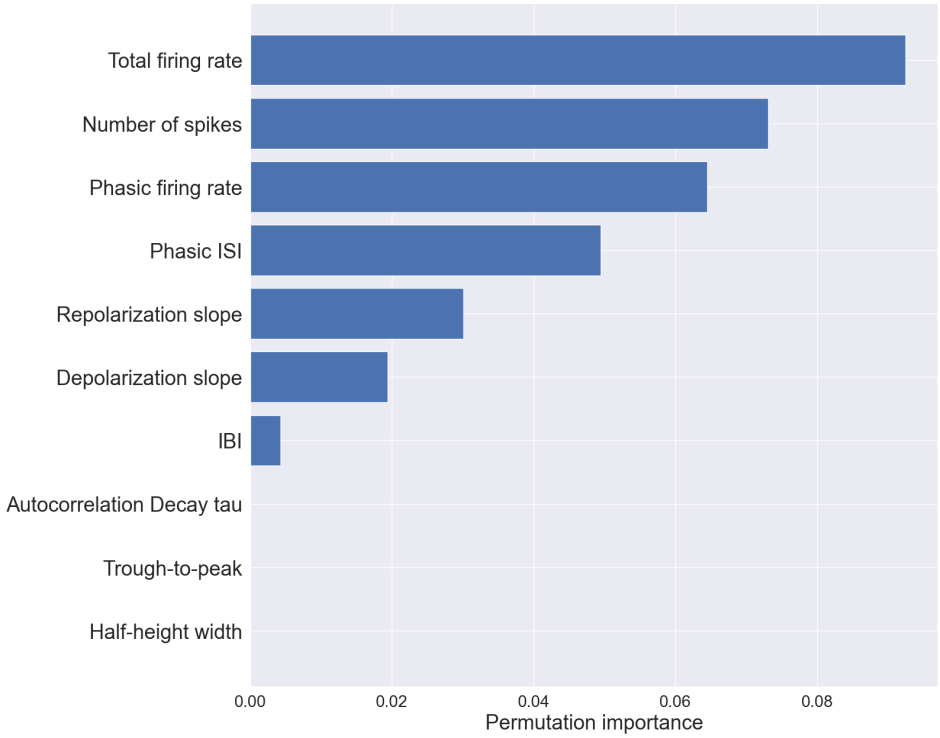


Figure 3.36: Permutation importance of single channel waveform features and time-series features towards classification of E/I units with kernel SVC, with gaussian kernel. Most important features were time-series related. In particular, among the most important we had most of the burst related features.

No importance in the form of coefficient could be computed for any of the nonlinear classifiers, so we only computed importance based on permutation of features. The feature importance represented in Fig(3.36) was computed for the classifier trained on single channel waveform features plus time-series features, which as we said decreased in accuracy with respect to the corresponding linear SVC one. We can see in this case that the feature with the highest importance was the total average firing rate, followed by burst related features. In the linear SVC classifier burst features were higher in the hierarchy of importance, while the total firing rate was lower. The same applied to the logistic regression classifier, and both classifiers performed better over this feature space than the kernel SVC one. We could then assume that the decrease in performance of the kernel SVC was due to the fact that it determined the total firing rate to be more important towards classification than burst related features, which instead seemed to be more specific to the two classes.

Next we used an ensemble decision tree classifier (EDTC). A decision tree classifier

creates a waterfall of feature-based decisions to separate samples from the two classes. Each decision generates a new branch with two leaves, each one of which will contain a percentage of samples from either classes. The aim of the decision tree is to make it so that each leaf of a branch has the largest amount of one unit and lower of the other, and vice versa. The algorithm stops when each leaf only has samples belonging to one or the other class. A single decision tree is not very powerful, which is why an ensemble model pools many of them together: each one is trained over a different subset of the data (bootstrapping), and the predicted classes are determined as an average of the predictions over many different decision trees thus trained differently.

We first applied an EDTC model to the lower dimensionality feature spaces, as before. For single channel waveform features (Fig(3.37a)) and time-series features (Fig(3.37b)) the performance of the classifier did not improve with respect to the linear classifier used previously or the kernel SVC, as EDTC, like all decision tree based models, need very high amounts of data and features both to work best.

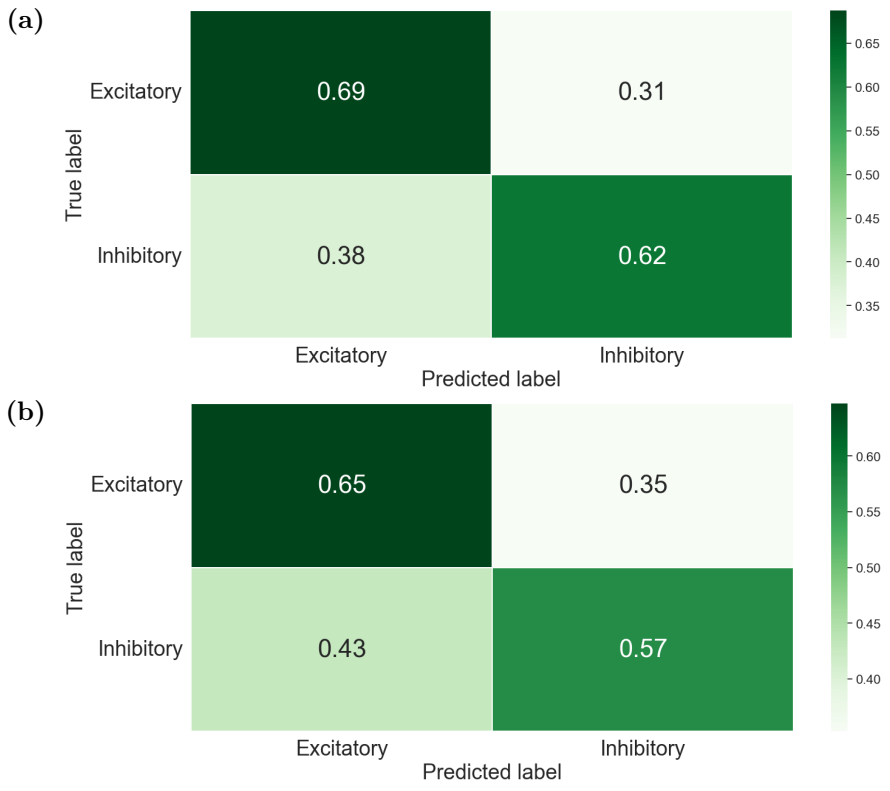


Figure 3.37: Confusion matrix for Ensemble Decision Tree Classifier (EDTC), applied to different subsets of the feature space. (a) EDTC applied to single channel waveform features. Most units were classified correctly for either classes, precision was more than 0.6 for both E and I. (b) EDTC applied to time-series features. More than half of the population for each class was classified correctly, with a slight bias towards E units.

For the single channel features and time-series features, the classifier worked as expected. As we can see in Fig(3.38), the classifier's performance did not improve in

this case, and it was still lower than that of linear SVC and logistic regression for this specific feature space.

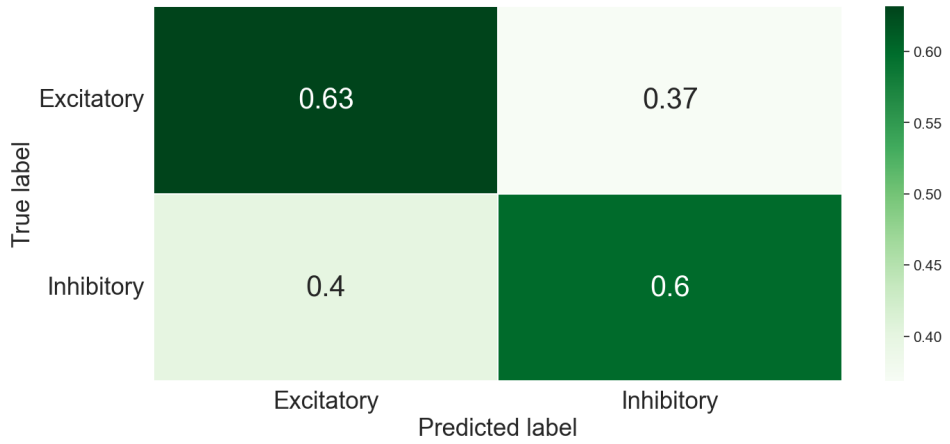


Figure 3.38: Confusion matrix for Ensemble Decision Tree Classifier, applied to the subspace composed by single channel waveform features and time-series features together (maximum depth = 25, minimum samples per leaf = 1, minimum samples per split = 10, number of trees = 100). More than 60% of each population is classified correctly, with a slightly larger precision for E units.

It was interesting to study the feature importance determined by EDTC over this set of features. Such importance, in the case of EDTC and all nonlinear classifiers, could only be computed as permutation importance. This was computed by implementing a series of training and testing of the classifier over the data set, permuting over all units one feature for each iteration, and studying how a specific score decreased with respect to the default situation. As our data set was composed of more E units than I, respectively 54 and 39, we chose as the score to be checked the F1-score. As we could observe, the importance of the features resembled strongly what we obtained also for previous models, both linear and nonlinear. Burst features such as the number of spikes per burst and the inter-burst interval (IBI) were the most relevant towards classification, followed by waveform features. Trough-to-peak interval is among the most utilized features used to distinguish between E and I units *in vivo*, and in the importance scale showed in Fig(3.39) it was in third place among all features *in vitro* in our case.

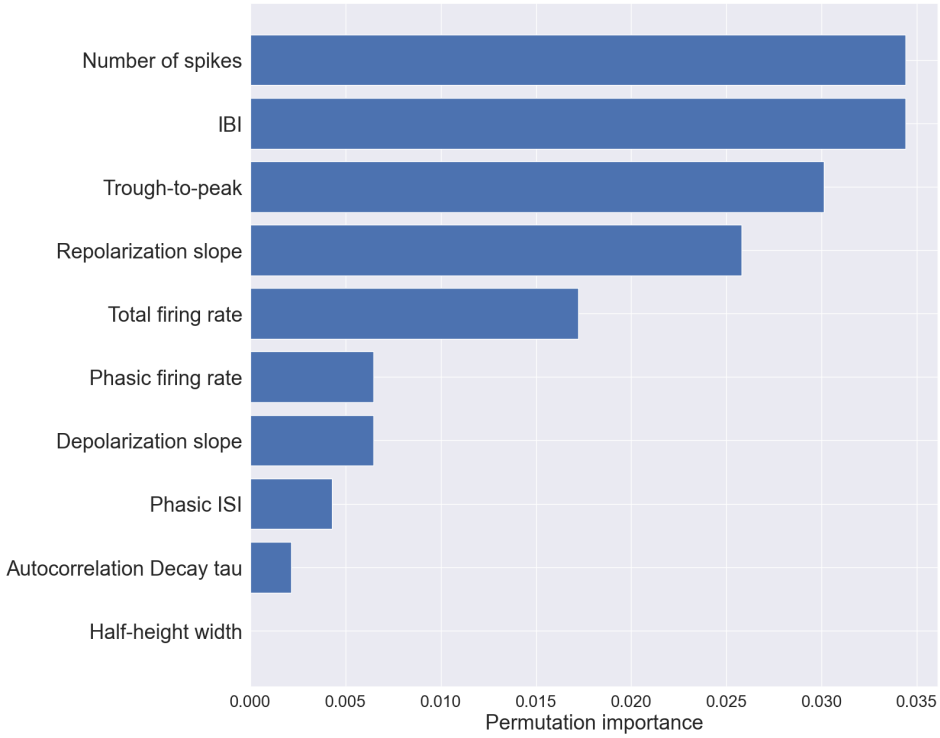


Figure 3.39: Permutation importance of single channel waveform features and time-series features, towards classification of E/I units with EDTC. The most important features resulted to be burst-related time-series features, followed by trough-to-peak interval, which is used successfully as a classification parameter *in vivo*.

The best classification results were obtained for the EDTC classifier applied to multi-channel waveform features and time-series features together, for which the results are presented in the form of a confusion matrix in Fig(3.40).

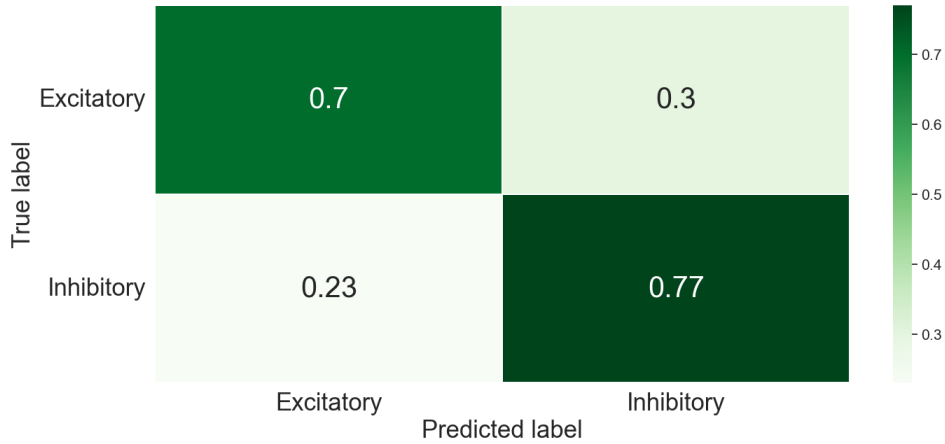


Figure 3.40: Confusion matrix for EDTC, applied to the subspace composed by multichannel waveform features and time-series features. More than 70% of each population was classified correctly, with a larger precision for I units than E. Among the classifiers and subspaces shown until this point, it was the first one to pass the 70% precision mark for both classes.

As we can see, precision was higher than 0.7 for both classes, and we had an average accuracy of ~ 0.74. This was the best performance we obtained in the classification effort, and was also confirmed through the cross-validation process whose results are shown in Tab(3.1)(in which the performance was actually better). It was of primary importance, then, to check the permutation importance of the features, to better understand what allowed for such an improvement in performance. The most relevant result we could see from Fig(3.41) was that burst related features were still among the most important (especially the IBI), and the most important waveform features did not all come from the same ranked channel.

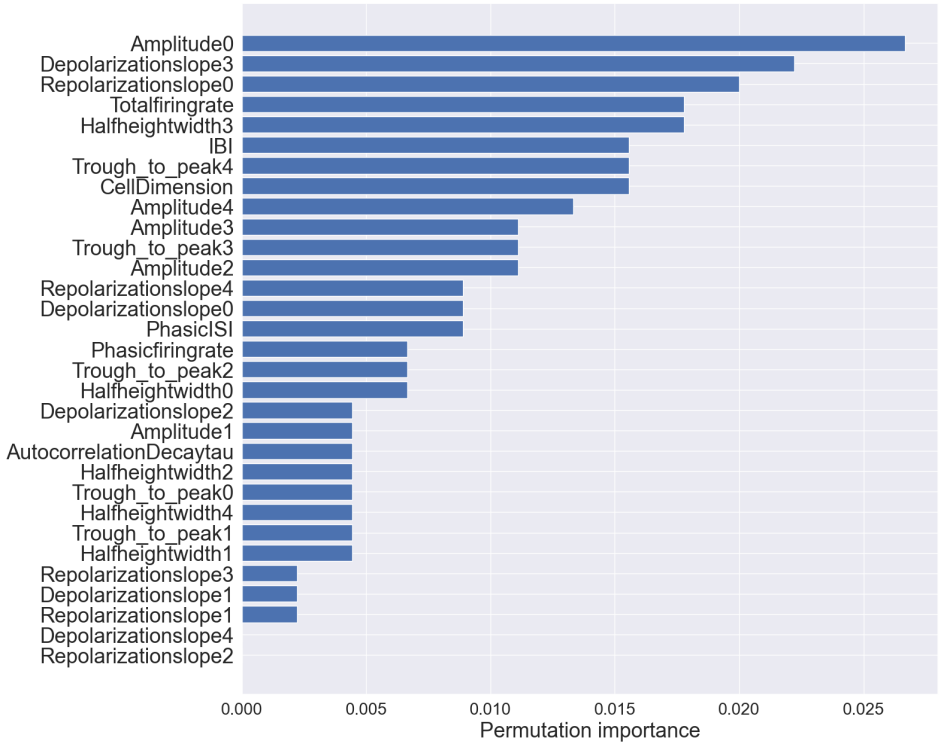


Figure 3.41: Permutation importance of multichannel waveform features and time-series features towards classification of E/I units with EDTC. As we can see, among the most important features we have waveform features coming from differently ranked channels, showing that the accuracy of the classifier is due to the more global view that this subspace offers. Using multichannel features we indirectly add the spatial dimension to the classification, which describes the underlying morphology of the neuron itself.

Relevance of waveform features was then directly tied to the way the signal evolved as it propagated further from the AIS. The assumption we made was that, by selecting features coming from different channels, we are also incorporating the morphology of the corresponding unit. This was also conveyed to the classifier via the cell dimension parameter, which placed quiet high in the importance score. Having a high spatial resolution and developing features based on spatial distribution was thus a powerful instrument towards classification, if we used a model that was able to perform well with a large number of features (as the EDTC). The confusion matrices represented

in Fig(3.40), Fig(3.37) and Fig(3.38) were computed by repeating the training and testing over 10 different computations, splitting and shuffling in a different, random way the data set and then computing the metric on the concatenated result. This was done in an effort to reduce overfitting, to which these models were prone, as the data set pool was a bit too small to really appreciate the classification power of this machine.

The last model we used was the gradient boosted model (GBDT), based on decision tree classifier, from the `LightGBM` library in `Python`. This model differed from the EDTC in the training of the different trees in the ensemble, which were trained sequentially rather than in parallel. At each step a series of decision trees was trained on the negative gradient of a loss function (in our case, the log-loss function, same used in the logistic regression), then a new set was trained based on the results of the first stage. We first tried to perform a classification using this method over the low dimensional feature spaces, comprised only of single channel waveform features and time-series features. As we can see from Fig(3.42), the classifier performed poorly over this low dimensional spaces: as for the EDTC, the GBDT classifier needed either many features or many data points to fully take advantage of the ensemble character of its procedure.

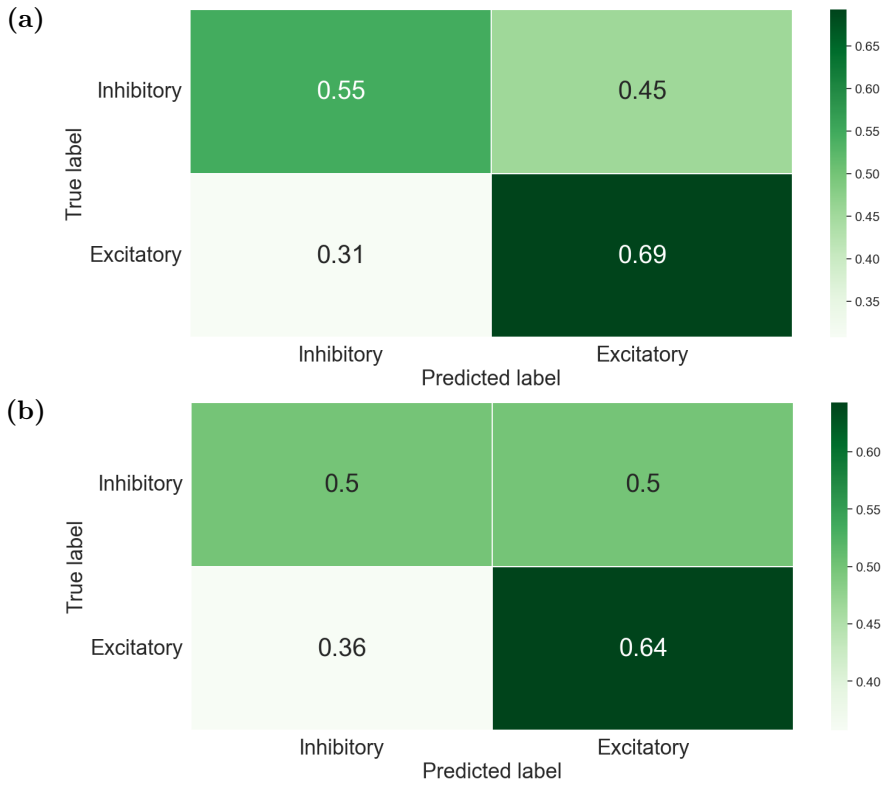


Figure 3.42: Confusion matrix for gradient boosted decision tree classifier (GBDT) applied to different subsets of the feature space. (a) GBDT applied to single channel waveform features. Classification was biased towards E, and for both classes precision was more than 0.5. (b) GBDT applied to time-series features. The same number of I units was classified in both classes, while precision for E was higher than 0.6.

Finally, we used a GBDT classifier on the multichannel waveform feature plus time-series features space, for which we obtained the results shown in figure 3.43: as we can see, accuracy and precision were consistent with what we obtained for the EDTC classifier applied on this space. The only real improvement with respect to the EDTC seemed to be a more balanced result, with a lower precision in the E and an higher one for the I, but the change was small. As for the EDTC model, confusion matrices for the GBDT were computed over 10 iterations, in between which the training and testing sets were reconstructed, each time with a different split shuffle but with the same stratification as the original data set. This way we reduced the risk of overfitting, which was one of the main problems when using this model, especially when working with a small data set like ours.

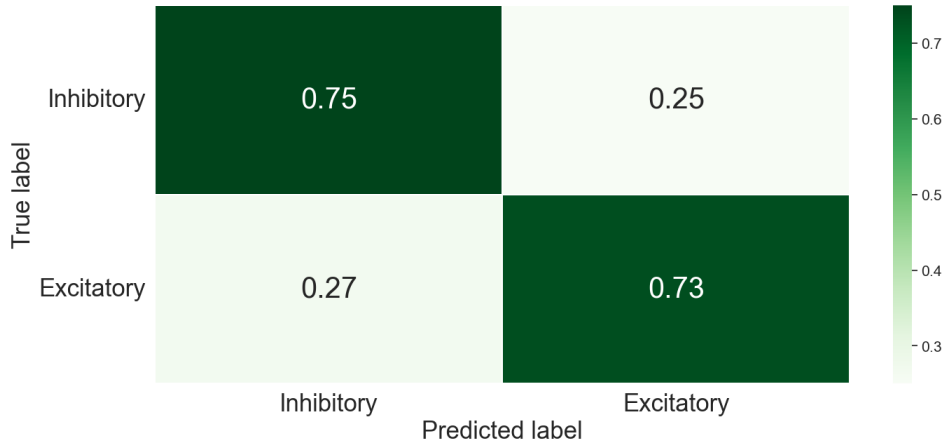


Figure 3.43: Confusion matrix for gradient boosted decision tree classifier, applied to multichannel waveform features and time-series features together. Precision was higher than 0.7 for both classes, and classification was more balanced between E and I than with the EDTC.

3.2.3 Classifiers comparison

We have seen in the previous section different models and data sets used for classification between E and I units, based on electrophysiological data only. The subsets of the feature space over which we attempted classification were:

- Single channel waveform features - feature related to the waveform of spikes generated from the unit. The features were computed over a cutout, created by averaging together all spikes generated by the unit and record by the best channel for that unit.
- Time-series features - features related to the time series alone, composed by all of the times of spikes assigned to that unit by the spike sorter. Sometimes, we used features related to spikes in bursts only: in this case, these were called burst-features.
- Multichannel waveform features - same features computed for the single channel waveform features. The footprint waveforms for the computation for these

Feature space	F1 score	Accuracy	Precision	Predicted fraction (E, I)
Single channel WF				
Linear SVC	(0.2 ± 0.2)	(0.47 ± 0.08)	(0.3 ± 0.3)	(0.01, 0.99)
Logistic regression	(0.48 ± 0.03)	(0.57 ± 0.03)	(0.58 ± 0.02)	(0.99, 0.01)
Kernel SVC	(0.6 ± 0.5)	(0.5 ± 0.1)	(0.5 ± 0.3)	(0.99, 0.01)
Ensemble tree classifier	(0.65 ± 0.06)	(0.59 ± 0.07)	(0.63 ± 0.03)	(0.98, 0.2)
Time-series features				
Linear SVC	(0.5 ± 0.4)	(0.6 ± 0.2)	(0.4 ± 0.4)	(0.01, 0.99)
Logistic regression	(0.5 ± 0.1)	(0.6 ± 0.1)	(0.6 ± 0.1)	(0.1, 0.9)
Kernel SVC	(0.67 ± 0.08)	(0.6 ± 0.06)	(0.64 ± 0.03)	(0.65, 0.35)
Ensemble tree classifier	(0.70 ± 0.05)	(0.65 ± 0.07)	(0.68 ± 0.07)	(0.26, 0.74)
Single channel WF plus time-series features				
Linear SVC	(0.41 ± 0.4)	(0.5 ± 0.2)	(0.5 ± 0.3)	(0.43, 0.57)
Logistic regression	(0.65 ± 0.12)	(0.6 ± 0.1)	(0.65 ± 0.01)	(0.33, 0.67)
Kernel SVC	(0.67 ± 0.08)	(0.6 ± 0.07)	(0.65 ± 0.03)	(0.73, 0.27)
Ensemble tree classifier	(0.71 ± 0.04)	(0.65 ± 0.07)	(0.67 ± 0.04)	(0.99 0.01)
Multichannel features plus time-series features				
Linear SVC	(0.4 ± 0.1)	(0.4 ± 0.2)	(0.4 ± 0.4)	(0.86, 0.14)
Logistic regression	(0.57 ± 0.1)	(0.6 ± 0.1)	(0.6 ± 0.1)	(0.68, 0.32)
Kernel SVC	(0.7 ± 0.2)	(0.68 ± 0.02)	(0.66 ± 0.04)	(0.5, 0.5)
Ensemble tree classifier	(0.77 ± 0.03)	(0.79 ± 0.01)	(0.8 ± 0.1)	(0.96, 0.04)

Table 3.1: Classification metrics for all classifiers used and all feature subspaces explored. The column on the right contains the class predictions for the corresponding models for the full unlabelled data set, trained on the full labelled one.

features were taken over multiple channels, so that each feature was an array of values, instead of just a scalar. Values in the array were then scaled between the minimum and the maximum, and ordered based on the latency of the average spike recorded over the specific channel.

Combinations of time-series and waveform features were also used, to work in higher dimensional spaces, and were actually the ones with the best results.

For classification, we used both linear and nonlinear models over each sub-space of the features space. In the previous section, when not indicated otherwise, the training was performed over a shuffled subset of the whole data set, while the testing and scores computation was done over the full data set, instead of just a small subset. This was done to reduce the variability given by different choices of training and testing data sets, and to have a larger pool of data for both the training and the testing, as the labelled data at our disposal represented only 93 units. All the classification metrics shown in Tab(3.1) were, instead, averaged over results coming from a 5-fold cross validation

computation of each model, over each feature space. Results for the GBDT and the EDTC were almost identical, and thus only those for the EDTC were represented in the table. As we could see, the results that we had in the previous section, in which metrics were computed using a different combination of training and testing data sets, are reiterated. Linear classifiers performed on par with nonlinear classifiers in the context of low-dimensional features spaces, such as single channel waveform features and time-series features. The disparity increased with the dimensionality of the space over which we are classifying, for which nonlinear classifiers performed largely better. The best result overall was obtained for the EDTC classifier applied over multichannel waveform features and time-series features together. In this case, F1-score, accuracy and average precision were all higher than 0.75. We were particularly interested in the F1-score, which was to be the preferred metric in the case of a slightly skewered population like ours, and for which the closest the value was to 1 the better the classification was deemed. F1-score for the EDTC over multichannel features and time-series features was the largest overall, equal to ~ 0.77. We noticed that it also maintained a very low variance over the different folds, and the same could be said for all metrics computed over EDTC with respect to the other classifiers. EDTC then also seemed to be a more stable classifier than either linear and other nonlinear ones.

Chapter 4

Discussion

In this project, we attempted to distinguish glutamatergic excitatory (E) from GABAergic inhibitory (I) neurons based only on extracellular electrophysiological features. New tools in the field of electrophysiological recordings, in particular HD-MEAs with their high spatiotemporal resolution, constitute a very high throughput source of neurophysiological information. There is a need to classify neurons into classes while they are active in a network, to gain a better understanding of the principles of information processing in biological neuronal networks. Being able to perform such a classification *in vitro* would make it possible to repeat experiments in a very controlled environment, under tunable conditions, thus reducing the innate complexity of biological systems (which is still too high *in vivo* to obtain conclusive results). Even though such classification has been shown to be possible *in vivo* [24][32][17][2][12], attempts *in vitro* have remained inconclusive [18][20].

The intent of this project was to use labelled data sets of E and I units, and their respective electrophysiological recordings, to discern the features that distinguish between the two classes, and subsequently draft a procedure to rapidly classify large amounts of recorded cells. To work towards this objective, we developed a pipeline to obtain a labelled data set, based on immunocytochemistry and imaging. This would allow us to use molecular features, which are difficult to observe but are, nonetheless, very precise, to trace classification features in an electrophysiological field, at high-throughput. The immunocytochemistry procedure was based on tracing down E and I neurons on the electrode array, to then correlate these cells with the units found via spike sorting. This procedure did not return the results we hoped for, as it was not possible from the images to identify any I neurons. We postulate that this could have been due to poor culture health during our experiments. In order to proceed with the classification effort, we decided then to use as labelled data set one which was established via spike transmission probability, by kind concession of Dr. Julian Bartram.

4.1 Feature distributions exploration

We first tried to establish a simple threshold between E and I units based on single features, or at most mixtures of two features. It is known that some extracellular electrophysiological features have been used successfully towards classification between E and I classes *in vivo* [24][12][33], such as trough-to-peak distance. We plotted the distributions corresponding to each one of these features for both our large non-labelled data set and for the labelled one. We hypothesized that if such canonical features alone were sufficient to enable reliable neuron class separability, we would observe distinct bimodal characteristics in the distribution of feature values in the unlabelled data set, or see a specific separation of the labelled one. However, no diversification of this sort was observed when using this low dimensional feature space exploration, as already described before [20].

Our next step was to explore higher-dimensional feature spaces. Using more of our features together, our interest was on possible multi-dimensional structures the data points might form. As we were not able to observe directly spaces with more than 3-dimensions, we used dimensionality reduction machines to reduce them to bi-dimensional embedding spaces. We used both linear and nonlinear dimensionality reduction machines towards this endeavor, being able this way to also use what we called multichannel features. These vectorized features, were collected over multiple channels for each unit. They took into account not just the morphology of the cell, but also the kinetics of the AP as it propagates away from the initiation zone.

PCA, which was our choice as a linear dimensionality reduction machine, was easily interpretable, but did not return any useful information about separation of the data set into classes. Regarding the higher dimensional feature spaces, comprised of both waveform features (single channel or multichannel) and time-series features, the explained variance of the first two principal components only amounted to < 0.4. For the lower dimensional feature spaces the explained variance was much higher, ~ 0.99, but with no separation between E and I. From these results we could conclude that not enough information was likely contained in low dimensional feature spaces, either just single channel waveform features or time-series features, for a classification to be performed.

Higher dimensional feature spaces, instead, while still not showing diversification, only explained a small portion of the variance when embedded via a dimensionality reduction, suggesting that there might be a separation hidden in more dimensions. We thus used UMAP, a nonlinear dimensionality reduction that aims at preserving the global and local structure of the data set manifold. UMAP is very efficient at maintaining any peculiar structure in the higher dimensional feature space, at the cost of direct interpretability. No separation could be observed in any of the UMAP reduced feature spaces, between labelled E and I units.

4.2 Classifiers

We next used supervised machine learning classifiers, trained and tested over our labelled data set. We performed training and testing using both linear (Linear SVM, logistic regression) and nonlinear (kernel SVM, EDTC, GBM) classifiers, and compared the performances. For each of these we also inspected the importance of each feature towards classification, to infer some information about what might physiologically diversify E and I neurons *in vitro*.

Table 3.1, summarizes the averaged scores for different metrics over a 5-fold cross validation. The performance for the various classification algorithms heavily depended on the training set (due to the small size of our labelled data set) for a single iteration; the k-fold cross-validations offer instead a reliable score estimation. As we can see, the performance of linear classifiers remains mostly stable over the different feature spaces, with the logistic regression always performing slightly better than linear SVC over most of them (the difference between the two is increased for single channel waveform features). Focusing on the F1 score, which, given the bias in our data set (54 E, 39 I units) was the best metric for performance evaluation, we found that both linear classifiers tended to fail for the largest feature space (multichannel features plus time-series features). This is especially evident if we compare it to the results from the nonlinear classifiers. Burst-related features also appear to be the most relevant ones in terms of feature importance for linear classifiers; this behavior is observed for both linear classifiers, and is stable for different choices of training and testing. It is also interesting to notice that single channel waveform features, which are historically better for discerning between E and I units in vivo, here are apparently the least important features towards classification. This might be due to the inability to replicate with 100% fidelity, the physiological environment *in vitro* (such as the absence of peripheral signalling, or the 3D dimensional structure).

Nonlinear classifiers showed lower performance in lower dimensional feature spaces (but still on par or higher than linear classifiers for all scores). The primary importance of burst related features is also maintained, with an increased relevance of waveform features as well. The best performing classifier, based on all tested scores, was the EDTC based on multichannel features and time-series feature together. For this classifier, we obtain the highest average F1 score, accuracy and precision, with an average precision of ~ 0.8 and an F1 score of ~ 0.77.

Observing the feature importance relative to this model and feature space (which are the same as for the GBM model), we see that the merit of this improvement in performance is mainly due to the use of multichannel features. Observing in particular Fig(3.41), we see that among the most important features we have multiple waveform features, which instead for in all the single channel feature spaces were the less relevant ones. In this case, the waveform features are the ones computed over differently ranked channels, with all channels chosen for the computation having similar relevance towards classification (no channel rank appears to be stably higher in the hierarchy). Among the important features we also have, as for all the models and feature spaces, burst related

features. The appearance of many differently ranked multichannel waveform features in the importance hierarchy, together with the vast improvement in performance with these features and the use of a nonlinear classifiers, are an interesting result.

The successful decision process involved in the training of the EDTC (or GBM) models is making use of the information coming from different channels, in order to perform classification. The channels were chosen based on the amplitude of the cutouts computed over them, which we said is dependent on the distance with respect to the origin of the channel itself. Based on this choice of the channels, and on the way we decided to order them (based on the latency of the spike), we indirectly added to the data set also information on the underlying morphology of the cell, and the behavior of the signal as it propagates away from the AIS (either along the dendrites or the axon). Morphology of E and I neurons is historically known to be different [16], so having insight over can be the decisive knowledge we needed to add to the model. The nonlinear models are able to pick up this hidden information in the feature space, and use it to better classify the E and I units.

Looking at Tab(3.1), we can also infer some information about our own cultures and measures. As we discussed in section (2.6.2), we were unable to determine labels for our units based on the imaging process. The reason for this was that we were unable, for any of our recorded cultures, to observe inhibitory cells in the images, after performing the ICCS (as can be seen in Fig(2.12b)). We have two possible interpretations to this absence of inhibitory cells in the GAD-stained images: either the used primary antibody was from a faulty lot; or inhibitory neurons in our cultures died (e.g. due to excitotoxicity) before we recorded and imaged. Tab(3.1) contains, in the rightmost column, the predicted labels over our curated, unlabelled units (in particular, the ratio of E and I units in the whole population). We can see that for our best performing model - the EDTC model trained over the whole labelled data set - the units classified as E in the best performing feature spaces (multichannel features or single channel features plus time-series) far outnumber the ones predicted as I. If we focus on EDTC applied over the multichannel waveform features and time-series features (which when tested had an average accuracy of ~ 0.8), the predicted E units are 0.96 of the total population, while I units only constitute 0.04 (this is even lower for the second best performing feature space, with only 0.01 ratio of I units). This, together with the peculiar behavior in the ICCS process which we already described, seems to support our hypothesis regarding the absence of I units in our cultures.

Chapter 5

Conclusion

The aim of this project was to record, curate and classify sorted putative units, based on their waveform features and time-series related features. Neuronal networks of primary rat hippocampal neurons were cultured, and their extracellular electrophysiological behavior recorded using high-density microelectrode arrays, or HD-MEAs. The raw recordings were then curated to eliminate artifacts and units that would have made our classification effort more difficult. From the remaining units, an assortment of features were extracted, over different temporal and spatial scales. Two different methods were then implemented, to create a labelled data set of excitatory and inhibitory putative units. One was based on immunocytochemistry staining, and aimed at imaging the glutamatergic (excitatory) and GABAergic (inhibitory) neurons, to be then traced on the electrode array and assigned to the corresponding sorted units.

This line of analysis did not return usable results due to issues in the immunocytostaining process, so we resorted to the second method: labels assigned via an electrophysiological connectivity study, in particular in the form of spike transmission probability. Thanks to this labelled ground truth data set, we were able to analyze features corresponding to the two different classes to find a separation between excitatory and inhibitory units based solely on this extracellular and high-throughput recordings. The first attempt at classification was performed via visualization of 1 (or at most 2) dimensional feature distributions, which did not return any useful information about separation between the two classes [18][20][19].

We postulated that the two classes might form some peculiar structures in the higher dimensional feature spaces, which we were not able to visualize and in which clustering machines would work inefficiently, because of the curse of dimensionality. We thus employed dimensionality reduction machines, both linear and nonlinear, to try and maintain any such high-dimensional structure in a 2-dimensional space, which could then be visualized. This procedure, once again, did not return any useful information regarding separation between the two classes: none of the plots exhibited a prominent diversification between the excitatory and inhibitory labelled units.

The last attempt at classification was performed via supervised machine learning classi-

fiers. For this step, we also implemented linear and nonlinear machines, with promising results, especially for nonlinear classifiers. We trained and tested our classifiers over the labelled data, and obtained the highest accuracy (~ 0.79) for the ensemble decision tree classifier, implemented over the multichannel waveform features and time-series related features together.

Observing the feature importance in our classification, we could also deduce some of the underlying behavior of the classifier, and why its performance over this high-dimensional feature space improved so much with respect to the other classifiers and spaces. While burst features were among the most important ones for all classifiers and different feature spaces used for training, waveform features were always the less relevant ones towards classification (that is, the most uniform across classes). This was the case until we introduced in the training data set the multichannel waveform features: in the EDTC applied to this feature space, which returns the best performance, waveform features were deemed the most important ones. Not just that, but the importance did not directly depend on the specific channel for which the features had been computed: different features, coming from differently ranked channels, shared the same amount of importance at the top of the relevance list.

By using these multichannel features, thanks to the high spatiotemporal resolution of our MEAs, we are very likely augmenting our feature space with spatial information about our neurons: their dimension, structure, the speed at which the signal is propagating through their morphological sections and how it is modified during this propagation. All of this information, not directly observable for us, is then captured by EDTC models, which are very sensitive to small but consistent variations in the feature space, and result in the best classification possible.

We can thus conclude that the ability to classify neurons based on their extracellular electrophysiological behavior is directly tied to the amount of information about the structure of the neuron itself we can either explicitly or implicitly recapitulate in our measurements, for example with the concept of multichannel features. This is only possible via the use of recording tools with superior spatial resolution, such as HD-MEAs, and powerful classification machines that are able to extract these underlying statistical dependencies from the data. We can also expect that the performance could be improved even further with higher density of electrodes, or higher number of recording channels, and with a larger labelled data set to train the supervised classifiers.

Data based methodological advances to systematically analyze information-dense complex data sets is a critical prerequisite to extract meaningful and neurobiologically relevant principles from extensive extracellular electrophysiological data. We would like to note that the analysis pipeline proposed in this work, and the techniques to systematically train, benchmark, and refine data-based inference engines (classifiers in our case) are contributions in this direction.

Appendix A

Neuron structure and electrophysiology

A.1 The compartments of the neuron

In the following subsection, we describe in more detail the different compartments of the neuron, their function and their morphology.

A.1.1 The main body of the neuron – Soma

The soma is the cell body of the neuron. It contains the nucleus and connects the dendrites to the axon initial segment. As all of the proteins necessary to the neuron are assembled in the soma, and can only travel a brief distance by diffusion from the nucleus, the axon contains microtubules associated motor units that transport the proteins to their destination (for example the synaptic terminal, which might be meters away from the nucleus). While most excitatory synapses land on dendritic spines, most inhibitory ones are directly connected to the soma (in the case of pyramidal neurons). The axon sprouts from the soma at the *axon hillock*, whose membrane contains many voltage-gated ion channels, and acts as the origin of the action potential.

A.1.2 How neurons receive signals – Dendrites

Dendrites are extensions of the neuron that propagate to the soma the electrochemical stimuli received from other neurons via synapses. Once a signal reaches a synaptic button, in the case of chemical synapses, the presynaptic axon or its telodendria release neurotransmitter, that traverse the synaptic cleft and attach to specific receptors on the membrane of the postsynaptic dendrites. This receptors, once activated, can induce either an excitatory or inhibitory response in the dendrite: contributions from each dendrite are integrated together in the soma, and if the total sum surpasses a specific threshold, an action potential is initiated in the AIS, and starts propagating into the axon (and back in the soma itself). Depending on their morphology, neurons can be

unipolar, bipolar or multipolar: unipolar neurons are characterized by a single stalk exiting the soma and separating into neuron and dendrites, bipolar have dendrites and axon on opposite side of the cell body and multipolar neuron have one axon and multiple dendritic trees. Pyramidal cells, which are the main excitatory neuron type in the cerebral cortex and hippocampus, belong to this last group, and are characterized by a high concentration of Na^+, Ca^{2+} and K^+ channels in their dendrites.

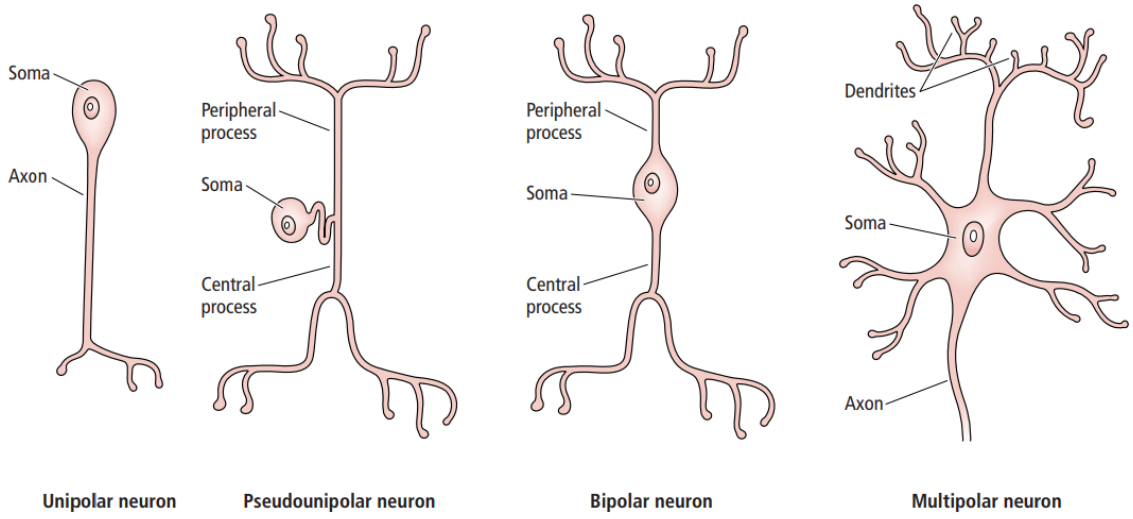


Figure A.1: The different morphologies of a neuron, depending on the number of dendrites and the respective positioning of axon and dendrites. Peripheral process is another term to describe dendrites, while central process describes axons. Adapted from [34], pg. 35.

A.1.3 How neurons send signals – Axon

Axons act as cables that propagate action potentials from the cell body towards other neuron. They have varying lengths (up to $2m$) and diameters. Axons are enveloped in a *myelin sheath*, which acts as an insulator and is composed of two types of glial cells: Schwann cells and oligodendrocytes. Along the axon there are gaps in the myelin sheath, called *nodes of Ranvier*. The insulation allows for a faster propagation of the AP, while the gaps serve the purpose of regenerating the action potential, which has lost some of its amplitude during the propagation. Of great relevance in the axon is the *axon initial segment* (AIS): this region at the origin of the axon separates the soma and the axon itself, and helps in the initialization of the action potential. It is approximately $20\mu m$ to $60\mu m$ in length and unmyelinated, and is characterized by an high concentration of voltage-gated sodium channels. It is characterized by three morphological features: $\sim 50nm$ undercoat, an absence of ribosomes and bundles of microtubules called fascicles [35]. An elevated concentration of proteins acts as a barrier between the cytoplasm of the soma and the AIS, lowering the speed of diffusion of membrane lipids and proteins. The morphological structure of the AIS can also change dynamically to respond to physiological and pathological changes, with variations that can last from seconds to entire days.

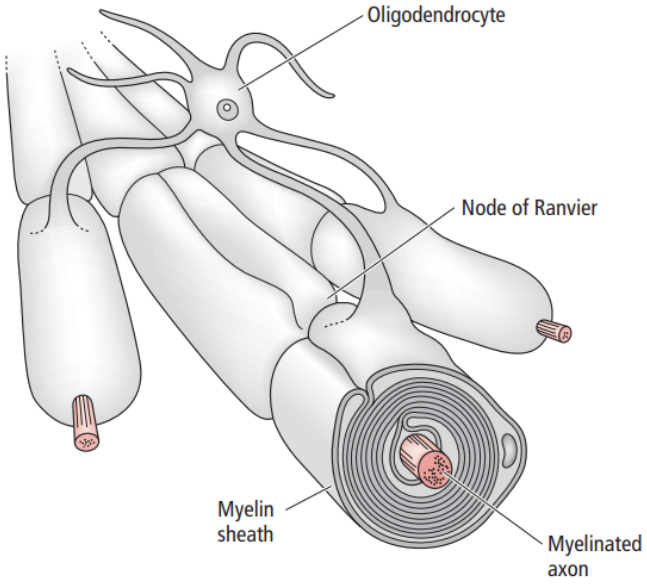


Figure A.2: Sketch of multiple axons running close to each other. As can be seen the oligodendrocyte is myelinating all of the axons, and one Node of Ranvier, where the depolarization happens again along the propagation, is visible over the central axon. Adapted from [34], pag 37.

A.1.4 How the signal is transmitted between neurons – Synapses

A synapse is a specialized structure which enables the communication of electrical or chemical signals from one neuron to the other. The signal is transmitted from the so called *presynaptic* cell, travels through the *synaptic cleft*, and enters the *postsynaptic* cell. Thus, a synapse has both pre- and postsynaptic sites, each with its own set of molecular machinery, to facilitate interneuronal communication.

Presynaptic structures tend to be localized along axon terminals where the axon divides into multiple thinner branches, whereas, postsynaptic structures are generally located along the dendritic tree. However, synaptic inputs may also be received on the soma or along the axon hillock as well.

Synapses could be of two main types: *electrical* or *chemical*. Electrical synapses maintain the continuity of the cytoplasm between pre- and postsynaptic cells, and facilitate rapid transmission of signals. At chemical synapses, there is no such continuity, instead, a complex chain of electrochemical transduction is initiated.

All of the signals incoming in the receiving part of the postsynaptic neuron cause variations of the transmembrane potential, which get integrated together and, if the potential surpasses a certain threshold, the AIS in the postsynaptic cell fires an action potential of its own.

Electrical activity in the presynaptic neuron leads to the release of chemicals called neurotransmitters from the presynaptic terminal that diffuse across the cleft and bind to receptors on the postsynaptic end and cause the direct or indirect opening of ion

channels in the postsynaptic cell membrane.

We focus mainly on chemical synapses, as the synaptic transmitters will be what makes it possible to apply the staining that we are going to need. The vesicles that diffuse the neurotransmitter in the presynaptic terminal are concentrated in *active zones*. An increase in Ca^{2+} levels at the presynaptic terminal, caused by the action potential and the consequent opening of specialized channels, results in the fusion of the vesicles with the membrane, causing the release of the neurotransmitter (the process is known as *exocytosis*). After traveling across the synaptic cleft, the transmitters bind to specific receptors in the postsynaptic cell membrane, activating them, and causing the opening or closing of ion channels.

All of these passages comport a delay in the transmission of the signal, in a range going from just $0.3ms$ to a few milliseconds. Chemical synapses also act as amplifiers of the transmitted signal: even a weak presynaptic signal opens up thousands of vesicles, increasing the chance of the generation of a postsynaptic AP. It is also important to note that the function of the neurotransmitters does not directly depend on the neurotransmitters themselves, but on the postsynaptic receptors and the channels that they activate: in general, the same neurotransmitter might excite some cells, while acting as an inhibitor for others.

A.2 Electrochemical signalling in neurons

The aim of the project is to determine a classification procedure between E and I neurons based on extracellular electrophysiological characteristics. The functioning of the brain is based on the interactions between neurons, which form locally structures which we can define as **neuronal circuits**. Understanding information processing in the brain then comes down to building a functional model for neuronal circuits and the pairwise interactions between neurons in them. We have already seen the morphological structures at the basis of neuron-to-neuron interactions (see section 1.1). We take a step back and study the signal that functions as the information carrier within neuronal circuits, the **action potential**.

A.2.1 The action potential

As we know, neurons communicate with each other, and thus process information as a network, via the generation, propagation and reception of electrical signals in the form of a sudden, large variation in the potential across the membrane of the neuron, between the cytoplasm and the extracellular fluid. This potential is commonly called *action potential* (AP), and was first studied in the squid giant axon by Hodgkin and Huxley in 1952, in a series of papers that earned them the Nobel Prize. The AP is produced by a sudden depolarization of the membrane from its equilibrium value, caused by a flux of ions across the membrane through specialized voltage-gated channels, which is translated in an *ionic current*. The AP has three phases: the *depolarization*, the

overshoot and the *repolarization* phase. Before and after the AP the membrane also goes through an *hypopolarization* and an *hyperpolarization* phase, respectively.

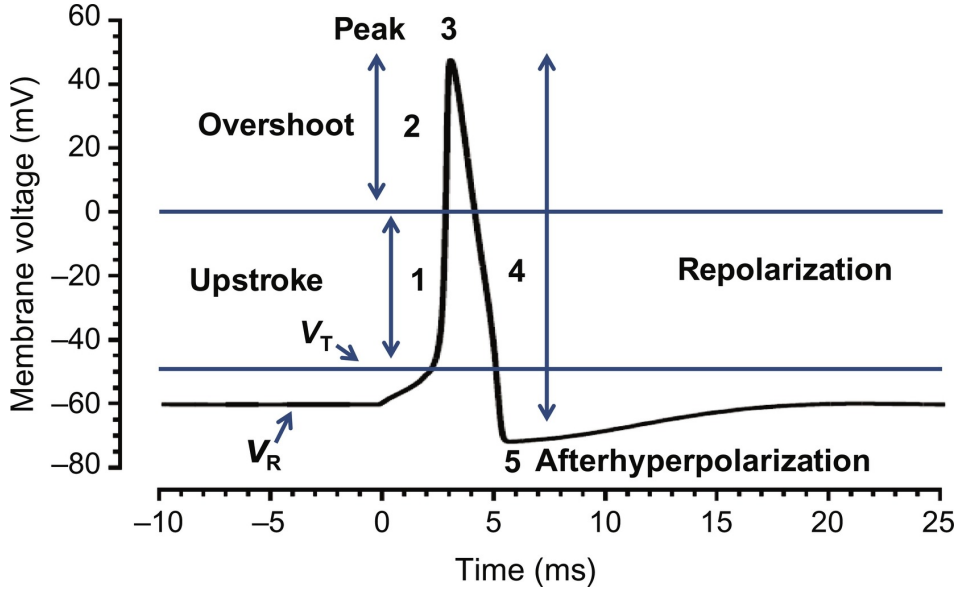


Figure A.3: The three main phases of an action potential: depolarization, overshoot and repolarization, preceded by an hipopolarization of the membrane and followed by a brief hyperpolarization. Adapted from [36].

APs differentiate via their amplitude, the duration and the ions involved in the depolarization/repolarization steps, and can be either Na^+-dependant (as for neuronal axons and soma, characterized by large amplitude and brief duration), Na^+/Ca^{2+}-dependant or Ca^{2+}-dependant [37]. We focus on ***Na*$^+$-dependant AP**: Na^+ channels are composed of 4 homologous domains, and act as a positive and fast feedback response to a membrane depolarization, via an influx of Na^+ ions. Once an incoming stimulus, usally corresponding to the integration of many stimuli coming from the dendrites, brings the membrane potential from its RMP (A.2.2) to a **threshold potential** usually around $-50mV$ to $-40mV$, the depolarization causes the opening of Na^+ voltage-gated channels (hypopolarization phase), following an *all-or-none law* in which all stimuli below threshold do not produce any effect on the postsynaptic neuron. The starting current of ions through the few open channels causes an increase in the depolarization (depolarization phase) due to the rapid influx of Na^+ caused by both electrical and chemical gradient, which in turns opens more channels until all of the channels in the AIS are open, at which point the cytoplams has been charged positively and V_m is closer to Nernst equilibrium potential for the sodium, $V_{Na} \simeq +61mV$ (overshoot, the potential gets circa to $V_m \simeq +50mV$). Vital characteristic of the Na^+ channels is the fact that, after a brief activation period, they inactivate, thus dropping the sodium permeability and decreasing quickly the depolarization (the peak of the AP corresponds to inactivation of all Na^+ channels): for the channels to reactivate again it takes a couple milliseconds, so that the current AP cannot reopen the channels that have determined its generation (repolarization phase).

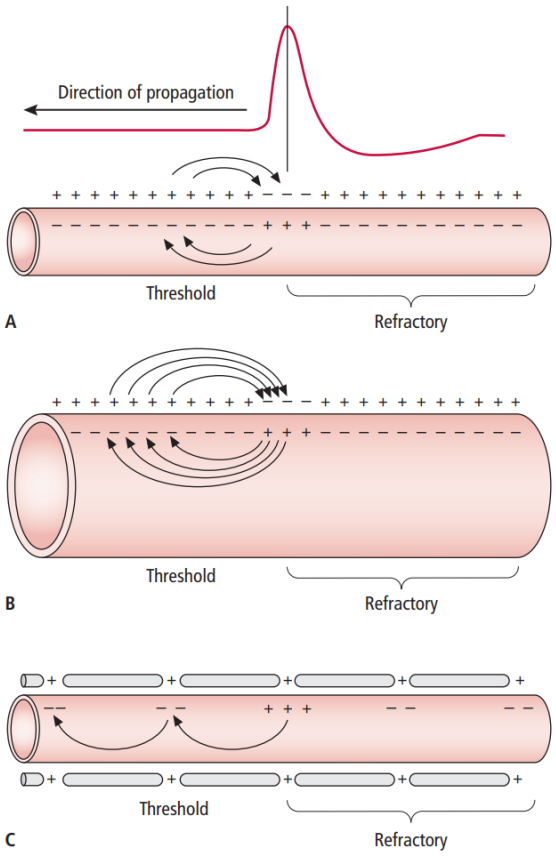


Figure A.4: Propagation of the action potential in the axon. As we can see, the action potential propagates via depolarizing sequentially the membrane of the axon (A), by increasing the membrane potential above the threshold and leaving behind regions of already open/inactive sodium channels, which determine the refreactory period and the inability of the AP to backpropagate (B). In (C) we see the effect of the myelin sheath and the Nodes of Ranvier, which makes it so that the depolarization only happens at specific and regularly spaced positions along the axon, increasing the speed of the propagation. Adapted from [34], pag 41.

Also K^+ voltage-gated channels play a role in the repolarization of the membrane, even though it's much less relevant in mammalian than in the giant squid studied by Hodgkin and Huxley. These channels can be either delayed rectifier, activating with a delay and slowly deactivating (they play a role in the repolarization), or fast activating and deactivating (only relevant for firing patterns). Membrane depolarization tends to open the delayed rectifier K^+ channels, causing an outflux of K^+ ions that speeds up even further the repolarization after the influx of Na^+ has depolarized the membrane. These channels open in a slower fashion than their Na^+ counterpart, and the K^+ outflux is pushed both by a concentration gradient and by the electrical one (once the V_m becomes positive). Before setting back to its resting potential, V_m always passes through a state in which it is more negative (hyperpolarization phase). The period immediately after the firing of an AP is known as *refractory*, and no AP can be generated during this period by the excitable cell (with the exception of its very end,

when a strong stimulus over the threshold might be able to induce a new triggering episode): during this time sodium channels are still either open from the old AP, and cannot thus open again to generate a new depolarization, or are inactivated and haven't had the time to regenerate and become active again.
As we said, the AP is actually not generated in the soma, but in the AIS: stimuli incoming are collected in the dendrites, integrated in the soma and passed to the AIS, where the triggering zone either fires a spike or not. This is because the EPSP needed to trigger an action potential in the AIS is just $10 - 20mV$, while in the soma it has to be as large as $30 - 40mV$. Once an AP is generated, it travels along the axon by sequentially depolarizing its membrane and following the same phases we just described, and thanks to the refractory period of each segment of the axonal membrane, it cannot travel backwards. The AP propagates much faster in a myelinated axon than an unmyelinated, as the myelin sheath insulates the axon allowing for saltatory conductance, with the sole exception of the Nodes of Ranvier, at which the depolarization happens and the AP gets "recharged". A larger diameter of the axon also causes a faster propagation. Once the AP reaches the terminal part of the axon, it induces a release of neurotransmitters from the presynaptic membrane, which then diffuse across the synaptic cleft to reach the receptors on the postsynaptic membrane. Here, depending on the neurotransmitters and their respective receptors, the postsynaptic neuron will experience either an excitatory or inhibitory effect from the presynaptic signal.

A.2.2 Resting potential

The distribution of various ionic species (in particular $N+$, Ca^{2+} and Cl^{-}) across the membrane produces a resting membrane potential (RMP), due also to the presence of specific ionic channels that allow free passage for the corresponding ions [36]. All of the ionic species have specific equilibrium potentials V_{eq}, which can be described by Nernst equation:

$$V_{eq,X} = \frac{RT}{z_X F} ln \frac{[X]_e}{[X]_i} \tag{A.1}$$

Where X is the ionic species, $[X]_i$ and $[X]_e$ are the internal and external concentration of such species, R is the universal gas constant, T the temperature in kelvin, F is Faraday's constant and z_X is the is the valence of the ions.
The equilibrium membrane potential can then be determined via the Goldman-Hodgkin-Katz equation:

$$V_m = \frac{RT}{F} \ln \left(\frac{p_K \left[K^+\right]_e + p_{Na} \left[Na^+\right]_e + p_{Cl}[Cl^-]_e}{p_K \left[K^+\right]_i + p_{Na} \left[Na^+\right]_i + p_{Cl}[Cl^-]_i} \right) \tag{A.2}$$

where p_X are the permeability values corresponding to the different ions, or via Millman equation:

$$V_m = \frac{g_K V_{eq,K} + g_{Na} V_{eq,Na} + g_{Cl} V_{eq,Cl}}{g_K + g_{Na} + g_{Cl}} \tag{A.3}$$

where g_X is the membrane conductance for the different species. ATPase pumps make sure that no ionic species is at its equilibrium potential, by pumping three Na^+ out for each two K^+ driven inside, thus fighting the spontaneous ionic current across free channels. The RMP is between -40-$-80mV$ with respect to the external fluid (for neurons it is about $-70mV$).

Appendix B

Classification optimization

The optimization of the classifiers, in particular the nonlinear ones, was implemented via the GridSearchCV method in the sklearn.model_selection package, in Python. This works by computing a k-fold classification (in our case the number of folds chosen was 5) for a range of values over each parameter, and choosing the set that returns the best classification quality metric, which we chose to be the F1-score. The optimized hyperparameters thus computed are shown below, in table (B.1).

Feature space	C	Maximum depth	Minimum samples per leaf	Minimum samples per split
Single channel WF				
Kernel SVC	20			
EDTC		8	2	2
Single channel WF plus time-series features				
Kernel SVC	10			
EDTC		25	1	10
Multichannel features plus time-series features				
Kernel SVC	10			
EDTC		25	2	2

Table B.1: Optimized hyperparameters for the nonlinear classifiers, with the feature spaces they were calculated upon. For all feature spaces, the number of estimators for the EDTC chosen via the optimization procedure was 100.

References

[1] Ralph Adolphs. The unsolved problems of neuroscience, 4 2015.

[2] Eric Kenji Lee, Hymavathy Balasubramanian, Alexandra Tsolias, Stephanie Anakwe, Maria Medalla, Krishna V. Shenoy, and Chandramouli Chandrasekaran. Non-linear dimensionality reduction on extracellular waveforms reveals cell type diversity in premotor cortex. *eLife*, 10:e67490, 8 2021.

[3] Hongkui Zeng and Joshua R. Sanes. Neuronal cell-type classification: Challenges, opportunities and the path forward. *Nature Reviews Neuroscience*, 18(9):530–546, 2017.

[4] Richard H. Masland. Neuronal cell types. *Neuron*, 2004.

[5] Arthur C. Guyton and John E. Hall. Textbook of Medical Physiology 12th-Ed. *Saunders Elsevier*, 2011.

[6] Giorgio A Ascoli, Lidia Alonso-Nanclares, and Stewart A Anderson. Petilla terminology: nomenclature of features of GABAergic interneurons of the cerebral cortex. *Nature reviews, Neuroscience*, 9:557–568, 7 2008.

[7] Gord Fishell and Nathaniel Heintz. The neuron identity problem: Form meets function, 10 2013.

[8] Sina Waldchen, Julian Lehmann, Teresa Klein, Sebastian Van De Linde, and Markus Sauer. Light-induced cell damage in live-cell super-resolution microscopy. *Scientific Reports*, 5, 10 2015.

[9] Jing Ge, David K. Wood, David M. Weingeist, Somsak Prasongtanakij, Panida Navasumrit, Mathuros Ruchirawat, and Bevin P. Engelward. Standard fluorescent imaging of live cells is highly genotoxic. *Cytometry Part A*, 83 A(6):552–560, 6 2013.

[10] Lionel G. Nowak, Rony Azouz, Maria V. Sanchez-Vives, Charles M. Gray, and David A. McCormick. Electrophysiological classes of cat primary visual cortical neurons in vivo as revealed by quantitative analyses. *Journal of Neurophysiology*, 89(3):1541–1566, 3 2003.

[11] Hugo Merchant, Thomas Naselaris, and Apostolos P. Georgopoulos. Dynamic sculpting of directional tuning in the primate motor cortex during three-dimensional reaching. *Journal of Neuroscience*, 28(37):9164–9172, 9 2008.

[12] Peter Barthó, Hajime Hirase, Lenaïc Monconduit, Michael Zugaro, Kenneth D. Harris, and György Buzsáki. Characterization of neocortical principal cells and interneurons by network interactions and extracellular features. *Journal of Neurophysiology*, 92(1):600–608, 7 2004.

[13] Vishalini Emmenegger, Marie Engelene J. Obien, Felix Franke, and Andreas Hierlemann. Technologies to study action potential propagation with a focus on HD-MEAs. *Frontiers in Cellular Neuroscience*, 13, 4 2019.

[14] Arne Schousboe. Advances in Neurobiology Volume 22 Series Editor. Technical report.

[15] Prasanna Tadi. Anthony S. de Leon. Biochemistry, gamma aminobutyric acid.

[16] Nathan W. Gouwens, Staci A. Sorensen, Jim Berg, Changkyu Lee, Tim Jarsky, Jonathan Ting, Susan M. Sunkin, David Feng, Costas A. Anastassiou, Eliza Barkan, Kris Bickley, Nicole Blesie, Thomas Braun, Krissy Brouner, Agata Budzillo, Shiella Caldejon, Tamara Casper, Dan Castelli, Peter Chong, Kirsten Crichton, Christine Cuhaciyan, Tanya L. Daigle, Rachel Dalley, Nick Dee, Tsega Desta, Song Lin Ding, Samuel Dingman, Alyse Doperalski, Nadezhda Dotson, Tom Egdorf, Michael Fisher, Rebecca A. de Frates, Emma Garren, Marissa Garwood, Amanda Gary, Nathalie Gaudreault, Keith Godfrey, Melissa Gorham, Hong Gu, Caroline Habel, Kristen Hadley, James Harrington, Julie A. Harris, Alex Henry, Di Jon Hill, Sam Josephsen, Sara Kebede, Lisa Kim, Matthew Kroll, Brian Lee, Tracy Lemon, Katherine E. Link, Xiaoxiao Liu, Brian Long, Rusty Mann, Medea McGraw, Stefan Mihalas, Alice Mukora, Gabe J. Murphy, Lindsay Ng, Kiet Ngo, Thuc Nghi Nguyen, Philip R. Nicovich, Aaron Oldre, Daniel Park, Sheana Parry, Jed Perkins, Lydia Potekhina, David Reid, Miranda Robertson, David Sandman, Martin Schroedter, Cliff Slaughterbeck, Gilberto Soler-Llavina, Josef Sulc, Aaron Szafer, Bosiljka Tasic, Naz Taskin, Corinne Teeter, Nivretta Thatra, Herman Tung, Wayne Wakeman, Grace Williams, Rob Young, Zhi Zhou, Colin Farrell, Hanchuan Peng, Michael J. Hawrylycz, Ed Lein, Lydia Ng, Anton Arkhipov, Amy Bernard, John W. Phillips, Hongkui Zeng, and Christof Koch. Classification of electrophysiological and morphological neuron types in the mouse visual cortex. *Nature Neuroscience*, 22(7):1182–1195, 7 2019.

[17] Roger N. Lemon, Stuart N. Baker, and Alexander Kraskov. Classification of Cortical Neurons by Spike Shape and the Identification of Pyramidal Neurons. *Cerebral Cortex (New York, NY)*, 31(11):5131, 11 2021.

[18] Andrea Becchetti, Francesca Gullo, Giuseppe Bruno, Elena Dossi, Marzia Lecchi, and Enzo Wanke. Exact distinction of excitatory and inhibitory neurons in neural networks: A study with GFP-GAD67 neurons optically and electrophysiologically

recognized on multielectrode arrays. *Frontiers in Neural Circuits*, (AUGUST 2012), 8 2012.

[19] Keiko Weir, Oriane Blanquie, Werner Kilb, Heiko J Luhmann, and Anne Sinning. Comparison of spike parameters from optically identified GABAergic and glutamatergic neurons in sparse cortical cultures. *Frontiers in cellular neuroscience*, 8(January):460, 2014.

[20] Keiko Weir, Oriane Blanquie, Werner Kilb, Heiko J. Luhmann, and Anne Sinning. Comparison of spike parameters from optically identified GABAergic and glutamatergic neurons in sparse cortical cultures. *Frontiers in Cellular Neuroscience*, 8(JAN), 1 2015.

[21] Marius Pachitariu, Nicholas Steinmetz, Shabnam Kadir, Matteo Carandini, and Kenneth D Harris. Kilosort: realtime spike-sorting for extracellular electrophysiology with hundreds of channels.

[22] Julian Bartram, Felix Franke, Sreedhar Saseendran Kumar, Alessio Paolo Buccino, Xiaohan Xue, Tobias Gänswein, Manuel Schröter, Taehoon Kim, Krishna Chaitanya Kasuba, and Andreas Hierlemann. A detailed input-output characterization of single neurons reveals the synaptic basis of spontaneous spiking in recurrent networks.

[23] et al Polikov V. In vitro models for neuroelectrodes: A paradigm for studying tissue–materials interactions in the brain. in: Reichert wm, editor. indwelling neural implants: Strategies for contending with the in vivo environment. chapter 4. boca raton (fl): Crc press/taylor and francis; 2008.

[24] Peter C. Petersen, Joshua H. Siegle, Nicholas A. Steinmetz, Sara Mahallati, and György Buzsáki. CellExplorer: A framework for visualizing and characterizing single neurons. *Neuron*, 109(22):3594–3608, 11 2021.

[25] Douglas J. Bakkum, Milos Radivojevic, Urs Frey, Felix Franke, Andreas Hierlemann, and Hirokazu Takahashi. Parameters for burst detection. *Frontiers in Computational Neuroscience*, 7(193), 2014.

[26] Carsen Stringer and Marius Pachitariu. Cellpose 2.0: how to train your own model.

[27] Daniel Fine English, Sam McKenzie, Talfan Evans, Kanghwan Kim, Euisik Yoon, and György Buzsáki. Pyramidal Cell-Interneuron Circuit Architecture and Dynamics in Hippocampal Networks. *Neuron*, 96(2):505–520, 10 2017.

[28] Leland McInnes, John Healy, and James Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. 2 2018.

[29] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Springer Series in Statistics The Elements of Statistical Learning Data Mining, Inference, and Prediction. Technical report.

[30] Anton Sirota, Sean Montgomery, Shigeyoshi Fujisawa, Yoshikazu Isomura, Michael Zugaro, and György Buzsáki. Entrainment of Neocortical Neurons and Gamma Oscillations by the Hippocampal Theta Rhythm. *Neuron*, 60(4):683–697, 11 2008.

[31] Yuta Senzai, Antonio Fernandez-Ruiz, and György Buzsáki. Layer-Specific Physiological Features and Interlaminar Interactions in the Primary Visual Cortex of the Mouse. *Neuron*, 101(3):500–513, 2 2019.

[32] Xiaoxuan Jia, Josh Siegle, Corbett Bennett, Sam Gale, Daniel R Denman, and Christof Koch. High-density extracellular probes reveal dendritic backpropagation and facilitate neuron classification 1 2.

[33] Henry Markram, Maria Toledo-Rodriguez, Yun Wang, Anirudh Gupta, Gilad Silberberg, and Caizhi Wu. Interneurons of the neocortical inhibitory system. *Nature Reviews Neuroscience*, 5(10):793–807, 2004.

[34] Maria A. Patestas and Leslie P. Gartner. A Textbook of Neuroanatomy. *Blackwell Publishing*, 2006.

[35] Christophe Leterrier. The axon initial segment: An updated viewpoint. *Journal of Neuroscience*, 38(9):2135–2145, 2 2018.

[36] Manoj Raghavan, Dominic Fee, and Paul E. Barkhaus. Generation and propagation of the action potential. In *Handbook of Clinical Neurology*, volume 160, pages 3–22. Elsevier B.V., 1 2019.

[37] Constance Hammond. The voltage-gated channels of Na+ action potentials. In *Cellular and Molecular Neurophysiology: Fourth Edition*, pages 55–91. Elsevier Inc., 1 2015.

[38] Marco Ballini, Jan Mueller, Paolo Livi, Yihui Chen, Urs Frey, Alexander Stettler, Amir Shadmani, Vijay Viswam, Ian L Jones, David Jaeckel, Milos Radivojevic, Marta K. Lewandowska, Wei Gong, Michele Fiscella, Douglas J. Bakkum, Flavio Heer, and Andreas Hierlemann. A 1024-Channel CMOS Microelectrode Array With 26,400 Electrodes for Recording and Stimulation of Electrogenic Cells In Vitro. *IEEE J Solid-State Circuits*, 49(11):2705–2719, 2017.

[39] Alessio P. Buccino, Torbjorn V. Ness, Gaute T. Einevoll, Gert Cauwenberghs, and Philipp D. Hafliger. A Deep Learning Approach for the Classification of Neuronal Cell Types. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2018-July(C):999–1002, 2018.

[40] Valentina Pasquale, Sergio Martinoia, and Michela Chiappalone. A self-adapting approach for the detection of bursts and network bursts in neuronal cultures. *Journal of Computational Neuroscience*, 29(1-2):213–229, 8 2010.

[41] Alessio Paolo Buccino, Xinyue Yuan, Vishalini Emmenegger, Xiaohan Xue, Tobias Gänswein, and Andreas Hierlemann. An automated method for precise axon reconstruction from recordings of high-density micro-electrode arrays. *Journal of Neural Engineering*, 19(2), 4 2022.

[42] Daniel A. Wagenaar, Jerome Pine, and Steve M. Potter. An extremely rich repertoire of bursting patterns during the development of cortical cultures. *BMC Neuroscience*, 7, 2 2006.

[43] Roger N. Lemon, Stuart N. Baker, and Alexander Kraskov. Classification of Cortical Neurons by Spike Shape and the Identification of Pyramidal Neurons. *Cerebral Cortex*, 31(11):5131–5138, 11 2021.

[44] Satoshi Katai, Keichiro Kato, Shunpei Unno, Youngnam Kang, Masanori Saruwatari, Naoki Ishikawa, Masato Inoue, and Akichika Mikami. Classification of extracellularly recorded neurons by their discharge patterns and their correlates with intracellularly identified neuronal types in the frontal cortex of behaving monkeys. *European Journal of Neuroscience*, 31(7):1322–1338, 4 2010.

[45] Francesco Masulli, Alfredo Petrosino, and Stefano Rovetta. Clustering High-Dimensional Data. Technical report, 2012.

[46] Maria Teleńczuk, Romain Brette, Alain Destexhe, and Bartosz Teleńczuk. Contribution of the axon initial segment to action potentials recorded extracellularly. *eNeuro*, 5(3), 5 2018.

[47] Hua An Tseng and Xue Han. Distinct Spiking Patterns of Excitatory and Inhibitory Neurons and LFP Oscillations in Prefrontal Cortex During Sensory Discrimination. *Frontiers in Physiology*, 12, 2 2021.

[48] Silvia Ronchi. ETH Library Electrical Stimulation and Functional Characterization of Neurons Using High-Density Microelectrode Arrays.

[49] Caterina Trainito, Constantin von Nicolai, Earl K. Miller, and Markus Siegel. Extracellular Spike Waveform Dissociates Four Functionally Distinct Cell Classes in Primate Cortex. *Current Biology*, 29(18):2973–2982, 9 2019.

[50] Kamil Rajdl, Petr Lansky, and Lubomir Kostal. Fano Factor: A Potentially Useful Information. *Frontiers in Computational Neuroscience*, 14, 11 2020.

[51] Huy Le, Beverly Peng, Janelle Uy, Daniel Carrillo, Yun Zhang, Brian D. Aevermann, and Richard H. Scheuermann. Machine learning for cell type classification from single nucleus RNA sequencing data. *PLoS ONE*, 17(9 September), 9 2022.

[52] Sam A. Booker and Imre Vida. Morphological diversity and connectivity of hippocampal interneurons, 9 2018.

[53] Carl Gold, Darrell A. Henze, Christof Koch, and György Buzsáki. On the origin of the extracellular action potential waveform: A modeling study. *Journal of Neurophysiology*, 95(5):3113–3128, 5 2006.

[54] Hernan Gonzalo Rey, Carlos Pedreira, and Rodrigo Quian Quiroga. Past, present and future of spike sorting techniques. *Brain Research Bulletin*, 119:106–117, 2015.

[55] Daniel N. Hill, Samar B. Mehta, and David Kleinfeld. Quality metrics to accompany spike sorting of extracellular signals. *Journal of Neuroscience*, 31(24):8699–8705, 6 2011.

[56] Alex Suarez-Perez, Gemma Gabriel, Beatriz Rebollo, Xavi Illa, Anton Guimerà-Brunet, Javier Hernández-Ferrer, Maria Teresa Martínez, Rosa Villa, and Maria V. Sanchez-Vives. Quantification of signal-to-noise ratio in cerebral cortex recordings using flexible MEAs with co-localized platinum black, carbon nanotubes, and gold electrodes. *Frontiers in Neuroscience*, 12(NOV), 11 2018.

[57] Henry Markram, Eilif Muller, Srikanth Ramaswamy, Michael W. Reimann, Marwan Abdellah, Carlos Aguado Sanchez, Anastasia Ailamaki, Lidia Alonso-Nanclares, Nicolas Antille, Selim Arsever, Guy Antoine Atenekeng Kahou, Thomas K. Berger, Ahmet Bilgili, Nenad Buncic, Athanassia Chalimourda, Giuseppe Chindemi, Jean Denis Courcol, Fabien Delalondre, Vincent Delattre, Shaul Druckmann, Raphael Dumusc, James Dynes, Stefan Eilemann, Eyal Gal, Michael Emiel Gevaert, Jean Pierre Ghobril, Albert Gidon, Joe W. Graham, Anirudh Gupta, Valentin Haenel, Etay Hay, Thomas Heinis, Juan B. Hernando, Michael Hines, Lida Kanari, Daniel Keller, John Kenyon, Georges Khazen, Yihwa Kim, James G. King, Zoltan Kisvarday, Pramod Kumbhar, Sébastien Lasserre, Jean Vincent Le Bé, Bruno R.C. Magalhães, Angel Merchán-Pérez, Julie Meystre, Benjamin Roy Morrice, Jeffrey Muller, Alberto Muñoz-Céspedes, Shruti Muralidhar, Keerthan Muthurasa, Daniel Nachbaur, Taylor H. Newton, Max Nolte, Aleksandr Ovcharenko, Juan Palacios, Luis Pastor, Rodrigo Perin, Rajnish Ranjan, Imad Riachi, José Rodrigo Rodríguez, Juan Luis Riquelme, Christian Rössert, Konstantinos Sfyrakis, Ying Shi, Julian C. Shillcock, Gilad Silberberg, Ricardo Silva, Farhan Tauheed, Martin Telefont, Maria Toledo-Rodriguez, Thomas Tränkler, Werner Van Geit, Jafet Villafranca Díaz, Richard Walker, Yun Wang, Stefano M. Zaninetta, Javier Defelipe, Sean L. Hill, Idan Segev, and Felix Schürmann. Reconstruction and Simulation of Neocortical Microcircuitry. *Cell*, 163(2):456–492, 10 2015.

[58] Amy Bernard, Staci A. Sorensen, and Ed S. Lein. Shifting the paradigm: new approaches for characterizing and classifying neurons, 10 2009.

[59] Francesca Gullo, Andrea Maffezzoli, Elena Dossi, and Enzo Wanke. Short-latency cross- and autocorrelation identify clusters of interacting cortical neurons recorded from multi-electrode array. *Journal of Neuroscience Methods*, 181(2):186–198, 7 2009.

[60] Lailun Nahar, Blake M. Delacroix, and Hyung W. Nam. The Role of Parvalbumin Interneurons in Neurotransmitter Balance and Neurological Disease, 6 2021.

[61] Attila I Gulyá, Manuel Megías, Zsuzsa Emri, and Tamá F Freund. Total Number and Ratio of Excitatory and Inhibitory Synapses Converging onto Single Interneurons of Different Types in the CA1 Area of the Rat Hippocampus. Technical report, 1999.

[62] Tatyana O. Sharpee. Toward functional classification of neuronal types, 2014.

[63] Maria Victoria Puig, Mika Ushimaru, and Yasuo Kawaguchi. Two distinct activity patterns of fast-spiking interneurons during neocortical UP states. Technical report, 2008.

[64] Rimjhim Tomar and Lubomir Kostal. Variability and Randomness of the Instantaneous Firing Rate. *Frontiers in Computational Neuroscience*, 15, 6 2021.