Towards Reliable Spike Sorting Evaluation Without Ground Truth Data

Robyn Greene

MInf Project (Part 1) Report

Master of Informatics
School of Informatics
University of Edinburgh

2021

Abstract

How do brains work? Computational neuroscience sits at the heart of this very question. Cognitive processes are underpinned by complex dynamics of networks of neurons in the brain. Recent recording technology now enables researchers to detect activity from thousands of neurons with immense precision. Whilst recording technologies evolve rapidly, analysis techniques struggle to keep up with the increasing quantity and complexity of the resultant data. This discrepancy, between the potential of these recording devices and their accompanying data processing techniques needs to be bridged. Spike sorting algorithms have been developed as an attempt to satisfy this need. These algorithms must successfully detect firing behaviour from signals in raw data, then assign each signal to the cell which produced it. Many spike sorting algorithms exist, but show little similarity in their results for one dataset [17, 1, 20, 27, 8, 7, 6]. Simultaneously, many studies which rely on extracellular recordings make use of only one such algorithm.

Confirming the results of spike sorting algorithms requires use of complicated and expensive experimental procedures. So called "paired recordings" can provide ground truth data, which are rarely available in practice. Evaluating these algorithms in the absence of paired recordings remains a challenge. Whilst various methods have been suggested, there remains little consensus on the best way to assess the accuracy of results. Without established methods by which to evaluate spike sorting results, experiments which use extracellular data are forced to rely on ill-validated algorithms. This data is used to draw ambitious conclusions about the role of the brain in cognitive processes. As the validity of these claims hinges on the correctness of spike sorting algorithms, it is imperative (to the reliability of such studies) that robust evaluation techniques are developed.

This project examines existing evaluation methods and results in the field of electrophysiology in order to suggest a more holistic approach to evaluation. This project demonstrates that many evaluation methods are not consistently indicative of accuracy. By evaluating the usefulness of eleven common performance metrics, a predictive model was designed and implemented to demonstrate one possible method by which to evaluate individual spike sorting outputs without requiring the use of multiple sorters. This model combines the analysis of metrics with a promising existent evaluation technique, the "consensus based method". In doing so, this project aims to contribute to the reliability of studies which make use of extracellular recording devices.

Acknowledgements

I would like to thank my supervisor, Dr Matthias Hennig, for his incredible patience and guidance throughout this year. I could not have persisted without his continual support and encouragement.

I would also like to thank my mother, Jo Greene, for providing stability in this recent and unprecedented pandemic.

Lastly, special thanks to my partner, Maya Khela, for emotionally supporting me through this period.

Table of Contents

Chapter 1

Introduction

When a neuron fires, a sudden spike in voltage can be recorded by a microelectrode, placed sufficiently nearby. One electrode may detect firing activity from multiple neurons therefore a process of "spike sorting" is required to establish the underlying correspondence between recorded voltage spikes and neural activity. Attempts to automate this process take the form of algorithms, "spike sorters", which employ various methods from the field of machine learning to first detect firing events using local potential changes detected in the space between neurons (hence "extracellular"), then associate each event with a specific spatio-temporal location in order to infer which neurons are firing [5]. These inferred neurons are referred to as "units" [14].

Whilst many spike sorters have been developed to execute this process, little agreement has been found between results of different methods [6, 3]. This leads to widely varying interpretations about which neurons are firing depending on which algorithm is used. To add to this problem, many labs choose one spike sorting algorithm and use this for most or all experiments [6]. It is often considered sufficient to run one spike sorter and simply trust the results.

To confirm the accuracy of results from spike sorters so called "ground truth data" is required alongside the extracellular recordings. Such ground truth data is both difficult and expensive to collect and is therefore rarely available [19]. In the absence of ground truth data, there are still no clear standards for evaluating the accuracy of the results of such algorithms on a given data set [5].

One commonly used method to evaluate results is by way of metrics; attempts to quantify the "goodness" of a unit by considering (for example) the ratio of signal-to-noise (SNR) in a signal, or the geometric and temporal separation of spikes from one another [5, 14].

Another possible method for evaluation is the so called "consensus method", suggested by Buccino et al. [6]. This method relies on the comparison of multiple sorter outputs and considers the agreement between results to be indicative of accuracy.

This project builds on the results found by Buccino et al. by using agreement as an approximation of accuracy in order to address the possibility that metrics can be used

to establish whether spike sorter outputs should be trusted. In doing so, this project aims to provide viable methods to evaluate spike sorting outputs in the absence of ground truth data. The overall goals of this project are to:

- Firstly, corroborate results found by Buccino et al. [6].
- Secondly, consider which sorters provide the most reliable results.
- Third, consider whether common metrics provide a good indication of whether sorted results should be trusted.
- Fourth, investigate practical solutions for establishing the trustworthiness of results, in the absence of multiple sorting outputs and ground truth data.

To this end, one dataset was chosen from a recent study which used extracellular neural recordings [10]. The following contributions were made:

- By making use of the tools provided by the SpikeInterface framework, neural recordings were analysed using various sorting algorithms [6, 3].
- In order to incorporate various modern spike sorters, some of the original analysis pipeline was adapted.
- The relationship between metrics and cross-sorter agreement was investigated.
- The relative reliability of sorting methods was examined.
- A predictive model was designed and implemented to establish the accuracy of spike sorter outputs in the absence of ground truth data or multiple sorting algorithms.

Chapter 2

Background

2.1 Extracellular neural recordings

Neurons mainly communicate with each other via "spikes" - all or nothing firing events which each last around 1ms. When an electrode is placed closed to a firing neuron, it is able to detect these spikes by recording the changes in extracellular potential resulting from ionic currents flowing through cell membranes during firing [15].

Performing this kind of recording involves the placement of one or more electrodes into brain tissue (in the space between neurons, hence "extracellular") with the aim of registering the activity of nearby neurons. Extracellular electrode recordings with multiple channels boast a range of advantages over other neural recording methods.

Firstly, this recording method boasts higher temporal resolution than other similar techniques. For example, two-photon calcium imaging, which is an invasive technique, allowing for multiple cells to be recorded at once with high spatial resolution, but suffers poor temporal resolution [16].

Crucially, given enough electrodes, this method allows for simultaneous measurements to be made from multiple cells, thus providing insight into the workings of networks of neurons [14]. Because neurons work as a system, accurate simultaneous recordings of many neurons are necessary in order to explain the neuron interactions which underpin cognitive processes [11, 25]. Additionally, it is possible to use this technique both outside a living organism ("in vitro") and by directly recording from a living organism ("in vivo"). Furthermore, since recordings can be made without interfering with neural function accurate results from such recording devices have the potential to provide insight into the workings of intact neural circuits in awake animals [12]. These advantages, when combined, provide a powerful basis for understanding the brain mechanisms underlying animal behaviour.

Developments in extracellular recording technology motivate the need to develop a common reference framework to quickly assess the performance of algorithms which process this kind of data [19]. Nonetheless, data processing techniques and evaluation remain a subject of debate [6, 14, 5].

Initially, in vivo recordings were done using only single electrodes at a time. The frustration with this method is that single electrode recording devices record the activity of very few neurons, and do not have the capacity to record from multiple locations in the brain. Recordings of this kind therefore struggle to support claims about the interaction between networks of neurons [11, 18]. Later recording devices therefore focused on recording larger populations of neurons simultaneously. In particular, the development of the tetrode (a four-electrode recording device) satiated the need to record from many neurons, detected from many locations at once. This device remains a popular recording device and has undergone significant validation, often by use of procedures that record intracellularly and extracellularly simultaneously (so-called "paired recordings", further discussed in section 2.3.1) [11, 25, 13]. Overall, the accuracy and explanatory power of tetrode recordings motivates the need for methods which can reliably process the data produced.

Further developments in recording technology have since emerged and continue to develop at a greater pace than their accompanying processing technologies can cope with. Of particular interest is the high density micro-electrode array (HDMEA), which is capable of recording approximately 1000 sites (often referred to as "channels") at once (one such probe is the NeuroPixel probe) [17]. HDMEAs provide a substantial amount of data, therefore it is essential that automated methods are capable of processing this data reliably and at scale. Without trusted methods for extracting information from these recordings, it is impossible to use these devices to their full potential.

In some sense, tetrodes provide a "middle ground" between the limited single electrode and the HDMEA. Tetrodes, unlike single electrode devices, provide the capacity to record from multiple locations. They therefore produce data that face similar challenges in analysis compared with more recent recording devices, albeit at a smaller scale. When compared with MEAs however, the number of neurons identified in a recording is generally much lower. This makes it possible to visually inspect the recorded data and facilitates the comparisons made in this project.

This project discusses results from tetrode recordings for this reason. Whilst the difficulties faced in using tetrode data are comparable to the difficulties faced by HDMEAs these challenges do not simply "scale up" when applied in HDMEAs. In some sense, HDMEAs provide more information as more electrodes can report activity from the same neuron. Nonetheless, with more channels (ergo more data) the computational challenge of processing such data increases. That is, the results presented here cannot necessarily easily extend to these new devices, and these challenges to scaling are presented in detail in Hennig's paper "Scaling Spike Detection and Sorting for Next Generation Electrophysiology" [12]. The hope is that the analysis performed in this project can offer a starting point for analyses which can extend to these more complex recording devices.

2.2 Spike sorting - methods and challenges

In electrode recording devices, one channel may detect signals which originate from multiple neurons. Additionally, signals from one neuron may be picked up by many

channels. Therefore, a process is required to infer the location of the neurons which are firing given the multiple points in space and time from which channels may detect the consequent signal. This process is referred to as "spike sorting".

One way to conceptualise the problem of spike sorting is the analogy of a cocktail party: given the conversations occurring at the party, the task is to isolate each of the speakers [12]. For tetrode data, consider the same task, but with several microphones detecting the sound at a much larger party. For HDMEAs, consider the task again but with more microphones situated around a football stadium of speakers.

In essence, this is a clustering problem, and various methods utilise different machine learning techniques in order to infer which clusters are present and to further infer which neuron is producing each spike train. The term "unit" in this context is used to differentiate the inferred location and behaviour of a neuron based on spike sorting, and the underlying behaviour of the neuron in question [19, 14]. That is, a unit is a model of a neuron. The measure of success, then, is whether each unit identified by a sorter actually corresponds to a neuron firing in the brain. Established methods to evaluate sorting are discussed in section 2.3.

For the purposes of this project, the terms "pre-processing" and "post-processing" are used to differentiate between the analysis steps to process raw data into spike sorted data, and the subsequent process of analysing and interpreting neuron interactions. The process of spike sorting typically occurs among several related steps in the pre-processing stage of analysing recordings.

In Hill's 2011 paper "Quality Metrics to Accompany Spike Sorting of Extracellular Signals", the process of spike sorting is broken down into two steps: Extraction of spike waveforms and clustering of waveforms into groups that represent the activity of single neurons [14]. This can be further broken down as follows:

- **Filtering:** Raw data from extracellular recording devices contain signals from noise as well as signals from local field potentials (LFPs), which both need to be filtered out [19]. Generally speaking, very high frequency signals are attributable to noise, whilst very low frequency signals are associated with LFPs. To eliminate these signals, a bandpass filter is often used, usually in the range 300-3000Hz. The signal that remains provides the activity from a few neurons close to the electrode plus background activity from neurons far from the tip of the electrode, leaving signals which are most useful for spike sorting [19].
- **Spike detection:** Spikes are identified based on the signals that remain. Usually, this is done by setting some amplitude threshold (often automatically and dynamically based on the signal-to-noise ratio observed in the data) and detecting spikes with amplitudes that exceed that threshold [19].

  Specific strategies for spike detection vary between sorting algorithms, and constitute one source of differing results among spike sorters. As a simple example, a low amplitude threshold leads to more spikes being detected, which may then lead to a larger number of units identified compared to a sorter which uses a higher threshold. Choice of threshold mechanism is therefore a trade-off between false positives (units identified which do not correspond to neuron activ-

ity) and false negatives (neuron activity which is not identified) [14].

- **Feature extraction and clustering:** This step involves attempting to differentiate spikes which were caused by different neurons. This is a nontrivial problem since signals from multiple neurons may have the same amplitude and occur nearby to each other in time and space. Neuron firing events do, however, have a "signature" - certain features (for example, amplitude and shape) specific to the neuron that produces them, which can be used in this process [19, 12]. By extracting features (often by principal component analysis) this signature can be used to differentiate spikes caused by different neurons [19, 14]. The feature space established in the previous step can then be used to cluster detected spikes into groups ("spike trains") and associate each cluster with a single unit.

  Approaches to assigning spikes to particular units can be split into two categories: density-based approaches and template-based approaches. In density-based approaches, dimensionality reduction is used alongside clustering techniques to first estimate the source of the signal and then cluster the signals together. By contrast, template matching approaches either extract or learn template spike shapes based on the spatio-temporal footprints of individual events. These templates are then used, alongside distance metrics, to assign the detected waveforms to the relevant unit [19, 12]. Figure 2.1 provides an overview of each of these processes.

In this project, 5 common sorting algorithms are evaluated: IronClust [17], Kilosort2 [1], Klusta[20], MountainSort4 [8] and WaveClus [7]. The most notable difference between these sorters is in their approaches to the "feature extraction and clustering" step described above. IronClust, Klusta and MountainSort4 use density-based clustering methods whereas Kilosort2 and WaveClus take a template matching approach.

Sorting algorithms are often designed with specific types of recording device in mind. It is worth noting that whilst some sorters are designed with a particular type of use case in mind, little work has been done on establishing criteria for the most accurate sorter in any given set of experimental conditions (for example the probe used, brain region recorded or experiment type) [16].

One analyses which does address the comparison of spike sorters is the meta-analysis performed by Magland et al. in their study "SpikeForest, reproducible web-facing ground-truth validation of automated neural spike sorters". Magland et al's study benchmarked 10 automated spike sorting algorithms by using a range of recordings combined with simulated ground truth data [16]. Whilst this project does not have access to ground truth data, a number of their findings relate to this analysis. Specifically, Klusta was identified as being less accurate than other sorters in most cases [16]. Additionally, MountainSort4 was found to be among the top performing sorters (in terms of accuracy to ground truth data) for low-channel-count datasets such as the tetrode data used here [16].

The data used in this project was originally sorted using Mountainsort3 [2, 10]. MountainSort4 is the updated version of this sorter [8]. Therefore, it is expected that the results from this sorter will echo those of the original analysis.

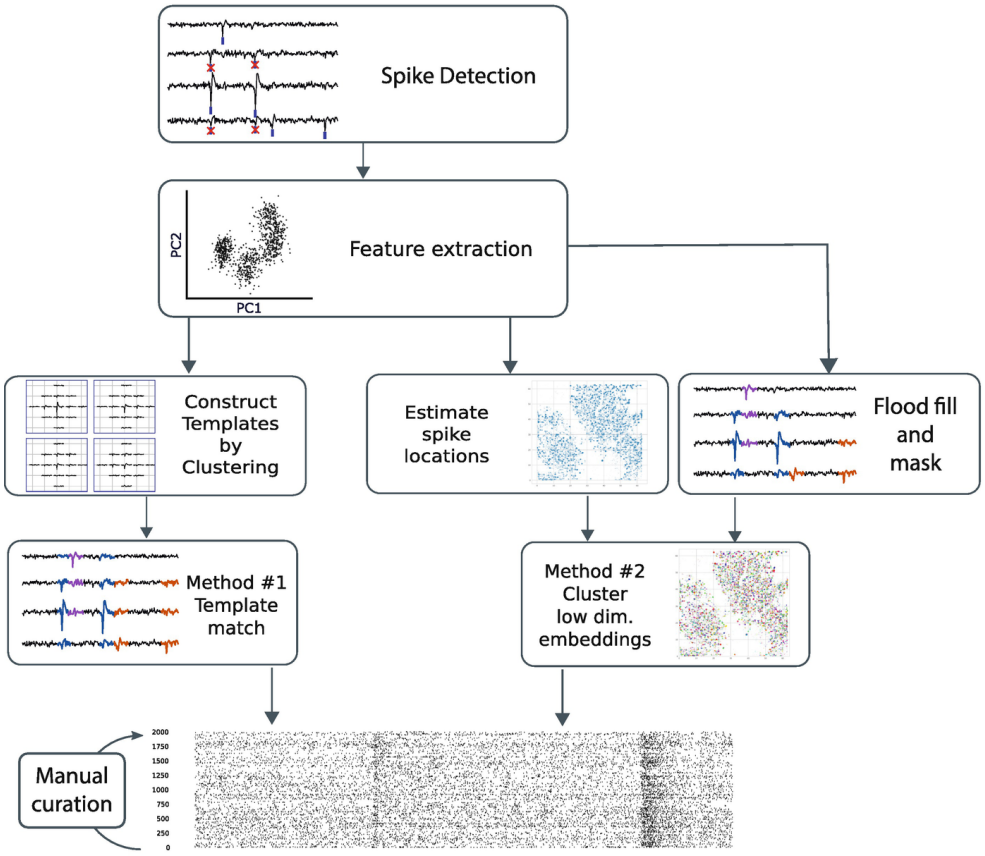

Figure 2.1: Spike sorting techniques. After detection and feature extraction, either template matching or density based methods are used to identify spike trains which may then be manually curated. Figure by Hennig et al. [12].

2.3 Evaluating sorting methods

Analysis of spikes can provide powerful results and evidence for studies in neuroscience, therefore the accuracy of this process is vital to ensuring reliability of inferences made [12]. In studies regarding neural mechanisms underlying organism behaviour, electrode data may be used to establish whether a neuron is firing in response to something in particular [19]. Interpretations based on inadequate spike sorting can therefore lead to erroneous claims about the neural coding behind behaviour [14, 24]. Establishing valid interpretations about the relationship between neuron firing behaviour and cognitive processes therefore hinges on the accuracy of spike sorting methods.

To complicate this issue, use of multiple sorters is time-consuming and complex, therefore many labs choose to perform analyses based on the outputs of only one sorting algorithm. [1] [6, 16].

Additionally, different sorting algorithms produce wildly varying results and algo-

[1] Although recently there have been attempts to streamline the process of using multiple sorters, most notably the SpikeInterface project described in section 2.3.4

rithms often yield many false positives [6]. Therefore, it is imperative that units which are likely to be false positives can be identified and eliminated from analysis. Without reliable methods by which to remove false positives many analyses will erroneously interpret units as real neuron activity in the brain, obscuring invaluable data from real neurons. Therefore, much of the task of evaluating sorting outputs centres around the task of identifying false positives.

Despite the obvious importance of this issue, there are still no clear standards for how spike sorting should be performed and evaluated [6, 14, 5]. Without established and tested evaluation methods, the reproducibility of results gathered in this way is challenged [6].

Various approaches for evaluation are possible. These include ground truth comparison, manual curation of results, performance metrics and agreement. The rest of this section addresses each of these methods and their limitations.

2.3.1 Ground truth data

Ground truth data provides the gold standard for evaluation for spike sorting results. In this context, ground truth data takes the form of "paired recordings". These are recordings in which intracellular recordings and extracellular recordings are performed simultaneously to validate the firing behaviour observed [25]. Such recordings are difficult to attain and are limited in the number of neurons which can be confirmed [19]. In most lab experiments, ground truth data is not regularly available since it is expensive and time consuming to produce. Therefore other methods must be relied upon to evaluate spike sorted results.

Evaluating results from spike sorters without ground truth data is therefore an ill-posed problem. In the absence of confirmation for unit correspondence with neurons, it is impossible to make a claim about the neurons' behaviour with certainty. Therefore there is a need to develop reliable methods to determine the accuracy of sorting outputs in the absence of ground truth data.

2.3.2 Manual curation

Manual curation relies on experts evaluating units in order to determine whether they reflect real neurons. This was a common component of the spike sorting process until recently, although there has been demand for fully automated processes since at least 2000 [11]. Manual intervention is still sometimes included as a final stage of spike sorting (for example Klusta, which is included in this analysis) [20]. Other sorting algorithms are intended to be fully automatic [8].

Fully automated spike sorting algorithms were developed to remove the requirement for time-consuming manual curation. Demand for full automation increases as better recording technology is developed and the amount of data produced by a single recording grows [16, 6].

Aside from saving time, the motivation for efficient automatic spike sorting algorithms also derives from the reliability of the sorting outputs themselves, since manual

curation is subject to inherent operator bias and is therefore difficult to standardise [6, 26, 20].

Manual curation is sometimes used as a benchmark for comparison with automated spike sorter performance [6]. Clearly, manual curation does not provide a consistent measure of performance. Hence, this technique cannot be solely relied upon for the evaluation of spike sorter outputs.

2.3.3 Metrics

Metrics are intended to give some quantification of the likelihood that a given unit corresponds to a real neuron in the brain. Finding useful metrics in the absence of ground truth data has proven challenging, and many such measures exist [14, 8, 5]. Most metrics attempt to describe one of the following:

- **Isolation:** These are measures relating to the quality of the clusters associated with firing events ascribed to a particular neuron [5].
- **Physiological factors:** The metrics use knowledge of the underlying physiological processes to detect errors in results. For example, neurons have "refractory periods"; periods of time after a firing event in which further spikes should not be present. Spikes detected during this time period may indicate a false positive unit.

A selection of these metrics have been included in this analysis (see section 3.4 for details).

Metrics are, in general, a common method for evaluating and curating results in the absence of ground truth [5, 6]. This project demonstrates that whilst some metrics may indeed indicate accurate units, others do not seem to hold much explanatory power.

2.3.4 Agreement

When multiple sorting algorithms are used on the same data, the overlap in their results can highlight the true positive units found. Use of agreement between sorting outputs as a tool for evaluation and curation is a relatively new proposal suggested by Buccino et al.. In their paper "SpikeInterface, a unified framework for spike sorting", six sorting algorithms were analysed and their outputs evaluated [6]. Their results indicated that different sorting algorithms identified wildly different numbers of units, with little consensus as to which units were present in the recording. Kilosort2 in particular was highlighted as producing many false positives, a result which was also found by this project (see section 4.1). Further, by using ground truth data and comparison with manually curated results, they were able to demonstrate that units which were found by multiple sorters were generally true positives and that units which were not found by multiple sorters ("low agreement units") were likely false positives.

These results together lead to the "consensus based method" for evaluating spike sorted outputs. This is the idea that agreement across sorters gives a reasonable indication of which units in a sorting are true positives [6]. This project relies on these results by

considering agreement to be a reasonable substitute for ground truth data. By combining the quantitative approach of metrics with the consensus based method this project aims to build on the above results.

Motivated by their results, as well as the fact that many labs use only one spike sorting algorithm, SpikeInterface was developed as a framework for sorting which provides various libraries to streamline spike sorting analysis pipelines [6, 16, 3]. SpikeInterface was used in this project because it allowed convenient use and comparison of multiple sorters. Additionally, the framework provided a number of additional benefits such as the capability to compute metrics for units and calculate agreement among sorters. Specific use of this resource is detailed in section 3.3.

2.4 Data

The data used in this project originated from the experiments conducted by Gerlei et al. in their paper "Grid cells are modulated by local head direction" [10]. The experiment involved recording from the medial entorhinal cortex region in the brains of mice as they roamed freely around a $1m^2$ open field arena. Recordings were taken from a total of 16 mice, with data from 8 mice ultimately being reported in the study. This data was appealing in part because the recordings were extensive, with an average duration of 25.1 minutes each ($\pm$3.8 sd) [10].

There were two reasons why this data was chosen for the project. Firstly, the data was recorded using tetrodes, which was appropriate for the purposes of this analysis (a discussion of this can be found in section 2.1). Secondly, a total of 208 recordings were available for analysis. The analysis presented in this project makes use of a small subset of these recordings, with the aim of extending it to the full data set next year. By using more samples, conclusions drawn will be more reliable by virtue of being tested on more data.

The key challenge presented by this particular dataset was that no ground truth data was available. Comparison with ground truth data would provide confirmation of claims made about which spikes relate to real neurons. Without this data conclusions about sorting results cannot be confirmed, as explained in section 2.3.1. In this way, the data used here is fairly typical of studies using extracellular recordings but with the benefit of comprising a large number of samples which can be used for analysis.

Chapter 3

Methods

3.1 Recordings used here

Data for this analysis consists of four recordings, which are taken as representative of the full dataset (described in section 2.4). Recordings from the original data set are as follows:

- Recording 0: M0_2017-11-14_16-54-12_of
- Recording 1: M10_2018-03-12_17-08-23_of
- Recording 2: M0_2017-11-08_15-21-09_of
- Recording 3: M10_2018-03-15_13-59-05_of

The full recording names given above correspond to the names as per the original data. The data from each recording consists of raw tetrode recordings from each of the 16 channels (four per tetrode), as well as a text file containing identifiers for the dead channels (discussed in section 3.3).

3.2 Adaptation of original pipeline

The original pipeline was used as a starting point for analysis. This pipeline was designed for use on a specific machine. Therefore, the initial stage of this project involved adapting the pipeline for use on another machine. Additionally, with a view to generalise the pipeline for use with many modern sorters using SpikeInterface, the pipeline needed to be adapted to incorporate this more general mechanism for sorting raw data.

Essentially, the goal was to sort the data using SpikeInterface and provide the output of each sorter as input to the original post-processing pipeline. This necessitated adapting the pipeline needed to be neatly split between pre-processing and post-processing of results such that sorting outputs could be provided as input to the post-processing portion of the original analysis code. At first glance, it would seem sufficient to simply replace the snippet of the code in which the sorting is performed.

This approach would have been time consuming and error prone since sorters vary wildly in usage and are therefore difficult to automate [6]. Without experience in using each of the sorters it would have been difficult to diagnose errors or establish reasonable parameter values for each individual method. SpikeInterface (described in section 2.3.4) provided an ideal solution to this problem, by streamlining the sorting process to a few lines of code which could easily be generalised over multiple sorters using inbuilt default settings. The pre-processing steps were successfully replicated using SpikeInterface (implementation details are given in section 3.3). Given a recording folder spike sorted outputs could be produced for all five sorters.

Whilst the SpikeInterface framework provided a more tested and streamlined way to incorporate many sorters, incorporating the output into the original pipeline proved difficult. The original pipeline used the MountainSort3 sorting algorithm [2]. This sorter can produce outputs to complement the main "firings.mda" output file. These supplementary output files were relied upon in later stages of the post-processing pipeline. The sorters used in this project have no similar mechanism for producing these files, so the options were to either replicate them manually or exclude them from the analysis.

In some cases, it was possible to replicate the steps taken in the original pipeline. For example, dead channels were removed by the same mechanism as in the original pipeline by using a dedicated dead channel file to identify and remove malfunctioning or very noisy channels.

The geometric location of the electrodes relative to each other was not directly available from the data and therefore needed to be inferred or excluded from the analysis. As sorting algorithms use the geometric location of electrodes when assigning events to specific units, excluding this information would render the sorting results meaningless. This information was therefore inferred on the basis of parameter files in the original data set as well as designs of the recording device used.

Additionally, a "filt.mda" file was expected by the post-processing pipeline. This file is an output produced by MountainSort3, which includes a representation of the data after the filtering stage of pre-processing. Whilst some detail on filtering was available in the original paper, it was insufficient to replicate the "filt.mda" file using updated sorting packages [10]. As a compromise between similarity to the original pipeline and generality for multiple sorters, the filtered data used was bandpass filtered (see section 3.3 for details). The post-sorting portion of the pipeline will be incorporated into this project next year, therefore the effect of these substitutions is not yet well known.

3.3 Data pre-processing and sorting

The pre-processing stage involved a series of steps: convert data to MDA format; group channels by tetrode; set tetrode locations and remove dead channels.

As per the original pipeline, raw Openephys data was converted to MDA format [23, 8]. This was done by making use of the SpikeInterface function `OpenEphysRecordingExtractor()` which reads the recording from Openephys format, then writes the resultant recordingExtractor object to MDA format by use of the

folllowing function: `MdaRecordingExtractor.write_recording()`. In the original pipeline, this was done for compatibility with the MountainSort3 sorter. When using SpikeInterface, converting the data to MDA format before sorting was found to produce faster run times than sorting data directly from Openephys format.

Channels were then grouped by tetrode (represented by colour in Figure 3.1). This was done by use of the spikeextractor module from SpikeInterface, specifically the function `set_channel_groups()`. This step is also in agreement with the original pipeline and is necessary for sorting algorithms to properly interpret data.

Channel locations were set using `set_channel_locations()` from the same module, which provided the sorting algorithms with information regarding the arrangement of tetrodes relative to each other.

Dead channels were then removed from the recording, as per those identified in the original analysis. This was done using toolkit from SpikeInterface with the `remove_bad_channels()` function. In Figure 3.1, the uppermost channel is a dead channel, and has been removed.

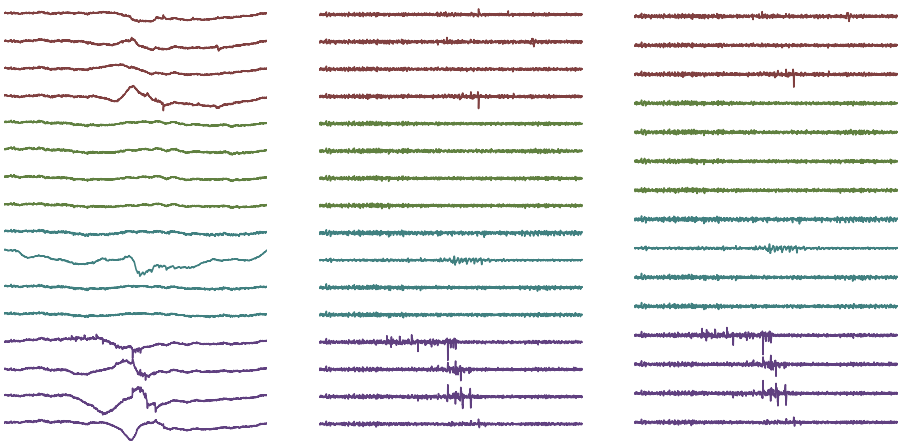

Figure 3.1: Left: raw data snippet. Center: data snippet after application of a bandpass filter. Right: Filtered data with dead channels removed.

A representative 0.2s snippet of the raw data can be seen in Figure 3.1. On the left is a snippet from the raw data. The image in the centre shows the effect of bandpass filtering the data (using the `bandpass_filter()` from spiketoolkit). On the right, the snippet is displayed after the dead channel has been removed and the filter has been applied. The filtering stage was included in this image to demonstrate the transformation that occurs at the filtering stage of sorting, described in section 2.2. As sorting algorithms, in practice, perform filtering themselves, this step is not actually performed prior to calling a sorting algorithm, and is included in Figure 3.1 for illustrative purposes.

It is worth noting that ground truth comparisons are used extensively in sorting algorithm development as well as evaluation (this was a major focal point for SpikeForest project) [16]. For this reason, spike sorters used in this project make use of these

well validated default parameters, which were readily available through SpikeInterface when calling sorting algorithms. Recordings were each sorted by all five sorting algorithms and the results stored. Sorting algorithms were called by use of the `run_sorter()` function from the spikesorters module of SpikeInterface. Given the name of a sorting algorithm and a recording extractor, this resulted in a "firings.mda" file, which contained information on units identified by the sorter.

The result of these steps were a set of sorting outputs ("firings.mda" files), one for each sorter. This process was repeated for all four recordings.

3.4 Metrics

Eleven performance metrics were tested in this project: firing rate, presence ratio, inter-spike-interval (ISI) violations, amplitude cutoff, sinal-to-noise ratio (SNR), silhouette score, isolation distance, L-ratio, NN hit rate, NN miss rate and D-prime.

- **Firing rate** is the average firing rate of the detected unit.
- **Presence ratio** refers to the fraction of time in which spikes are present.
- Neurons have a refractory period after a spiking event during which they cannot spike again. **Inter-spike-interval violations** refers to the rate of refractory period violations. Since refractory period violations in a unit can indicate errors, this metric is commonly used to detect false positives [14].
- **Amplitude cutoff** provides an estimate of the miss rate which is based on the observed amplitudes.
- **Signal-to-noise ratio (SNR)** is expected to be higher in units for which spikes are not heavily obscured by noise. This is a common metric, and Buccino et al. concluded that SNR corresponds to agreement among sorters [6].
- **Silhouette score** is the ratio of cohesiveness of a cluster and separation from other clusters. The cohesiveness of a cluster is measured using the distance between member spikes, and separation from other clusters is measured as the distance to non-member spikes [6]. A cohesive and well separated cluster will therefore have a low silhouette score.
- **Isolation distance** measures the smallest ellipsoid which contains all spikes from a given cluster as well as an equal number of spikes from other clusters, where the ellipsoid is centred round the given cluster [6]. This value should be higher in more reasonable units as this represents a more isolated cluster.
- **L-ratio** is another metric which relates to cluster quality. A low L-ratio value indicates a cluster that is well separated from other spikes on that tetrode [22, 21]. Therefore it is expected that L-ratio will be lower in "good" units.
- **NN hit rate** and **NN miss rate** refer to the nearest neighbours method, which in this context considers the class of nearby points as an indication of cluster quality [28]. Both metrics provide a non-parametric measure to estimate unit

contamination [6, 8]. In essence, this is another separation based metric aiming to describe something similar to isolation distance [1].

- Finally, **D-prime** uses linear discriminant analysis to estimate the classification accuracy between two units.[6]. As a measure of accuracy, this metric is expected to be higher in true positive units.

Given each recording and its corresponding sorting outputs, SpikeInterface was used to calculate metrics associated with each unit.

First, recordingExtractor objects were created for each recording, along with sortingExtractor objects for each sorted output. This was done by making use of the extractor module of SpikeInterface. Specifically, recording extractors were loaded from "raw.mda" files using `MdaRecordingExtractor()`. SortingExtractor objects were then loaded from the "firings.mda" files using `MdaSortingExtractor()`.
These Extractor objects allowed for convenient access of information about the recordings and sortings.

The spiketoolkits module from SpikeInterface was then used to calculate metrics using `validation.compute_quality_metrics()`. The function returned a dataframe containing values for all the metrics for each unit produced by a particular sorting output. In practice, these dataframes were combined for all recordings and all sorters.

3.5 Agreement calculations

To identify a match between two spike trains (spiking activity associated with a unit) an agreement score is calculated as follows: first a number of "matches" between the two spike trains is calculated by identifying the number of pairs of spikes from the first and second train which occur within a similar time frame. Again, the default value was used, which here is 0.4ms. An agreement score, s, is then calculated:

$$s = \frac{n_m}{n_1 + n_2 - n_m} \tag{3.1}$$

where n_m is the number of matched spikes between the two spike trains and n_1 and n_2 is the number of spikes identified in the first and second spike trains respectively. For the purposes of this project, "matched units" is used to refer to units which were found by two or more sorters while "unmatched units" is used to refer to units which were found by only one sorter.

Agreement is therefore not explicitly linked to any metric listed here and can be considered a separate measure of spike sorter performance. In this project, based on the results found by Buccino et al., unit agreement is assumed to be an indicator of a true positive unit [6]. Specifically, if a unit is found by more than one sorter it is assumed to be a true positive unit (as explained in section 2.3.4).

[1] indeed Chung et al. used these measures as a representation of isolation in their paper presenting the MountainSort4 method [8]

Agreement between units from different sorters was calculated using the SpikeInterface framework. For each recording, all sorting outputs were converted from "firings.mda" files to sortingExtractor objects using `MdaSortingExtractor()`. All five SortingExtractor objects were stored in a list and a MultiSortingComparison object was produced by using `compare_multiple_sorters()`. This MultiSortingComparison object was then used to identify each matched unit, given the agreement criteria described above.

This information was incorporated into the dataframe described in section 3.4 such that matched units could be identified and associated with that unit's metrics.

Chapter 4

Results and analysis

4.1 Sorting methods disagree on the number of units present

After sorting using all five sorting methods, the number of units identified in each recording was counted and can be seen in Figure 4.1.

Kilosort2 identified the largest number of units in all recordings with a mean value 66.5 (sd = 1.67). Klusta produced an average of 45.75 units, and had the largest variance in number of units identified (sd = 13.79). MountainSort4 and WaveClus performed similarly with means 17.75 (sd = 1.48) and 13.25 (sd = 2.28) respectively. IronClust identified the fewest units of all sorters, with an average of only 11 (sd = 2.12).

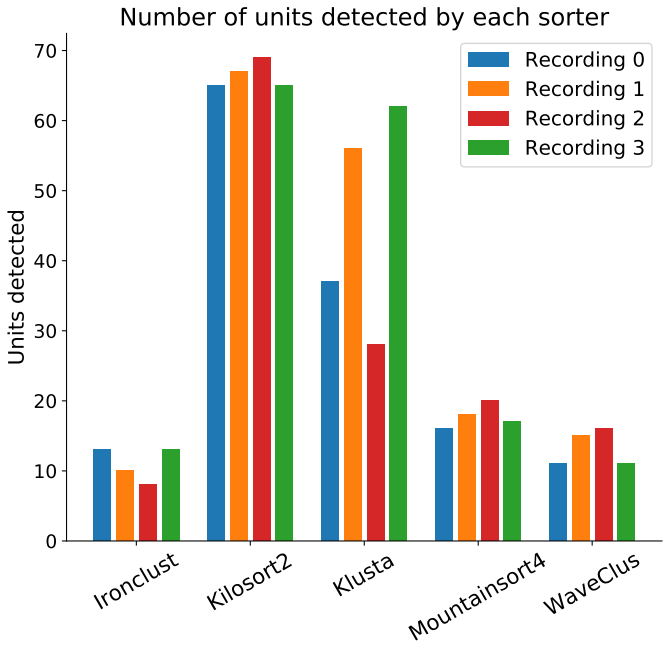

Figure 4.1: Bar chart displaying the number of units identified in each recording by each sorter.

Since each recording consists of data taken from only 4 tetrodes, the quantity of units identified by Kilosort2 and Klusta seem particularly unreasonable. Each tetrode can be expected to record from approximately 3 neurons, so no more than 12 units are expected. Klusta showed little consistency with other sorters in terms of the relationship between the number of units identified between recordings. However, since this sorter identified more units in each recording than any of IronClust, MountainSort4, or WaveClus, it is possible that all units identified in the other sorters were also identified by Klusta. This would be reflected in a high agreement between Klusta and the three sorters with the lowest numbers of identified units. Klusta can be somewhat excused for producing so many units, as the method is intended to include a manual curation step.

Kilosort2, MountainSort4 and WaveClus showed agreement in the number of units in each recording relative to the others. Across the three sorters, Recordings 0 and 3 were both identified as having the fewest units, with recording 1 producing a few more, and recording 2 producing the most. The pattern seen among these three sorters calls into question whether these sorters share this behaviour as a result of high agreement - perhaps this pattern arises because similar units were identified. One may consider the possibility that different sorters have different sensitivity, leading to this outcome. This seems unlikely to be the cause in this case, given the implausible number of units that were detected by some sorters. The more likely explanation for this is that a large number of units are false positives. This result is further addressed in the following section.

Of the two sorters that overlap between this project and the analysis used in Buccino et al.'s analysis, a similar relationship between sorters was observed [6]. Specifically, in that analysis, Kilosort2 found 446 units, whilst IronClust identified only 233. That analysis identified a far larger number of units due to the specific recording device used, but the relationship between Kilosort2 and IronClust is seen here nonetheless.

Results seen here agree with those reported by Buccino et al.. Specifically, there was high variation in the number of units identified by different sorters, and Kilosort2 produced a large number of units (suggesting many false positives).

4.2 Unit agreement varies by sorter

SpikeInterface was used to compare sorter outputs and find matching units in each recording. In total, 8 units were found by at least two sorters ("matched units").

4.2.1 Sorting outputs demonstrate low agreement

In recording 0, no matched units were identified (that is, sorters completely disagree on which units are present). Three units from recording 1 were identified by multiple sorters: units 2, 7 and 38. Unit 2 was identified by three sorters, unit 7 by all 5 sorters and unit 38 by 3 sorters. The waveforms from unit 7 in each sorting output are displayed in Figure 4.2.

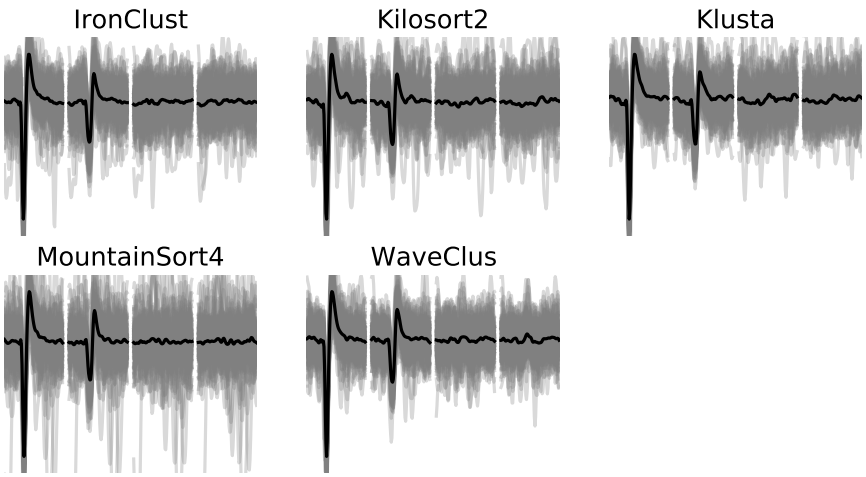

Figure 4.2: Waveforms for one unit found by all sorters in recording 1.

In recording 2, three units were identified by multiple sorters: units 1 (4 sorters), 8 (2 sorters) and 18 (2 sorters).

In recording 3, 2 units were identified by multiple sorters: unit 9 (4 sorters) and unit 11 (4 sorters). In both cases the unit was picked up by all sorters except for WaveClus. The waveforms for unit 9 are given in Figure 4.3.

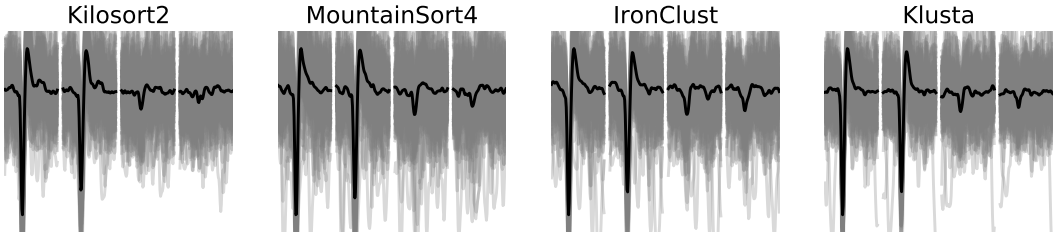

Figure 4.3: Waveforms for one unit found by four sorters in recording 3.

Visually, the matched units' waveforms look similar across each of the sorters which identified them. All matched units follow expected forms - that is, they are plausibly reflective of actual spiking behaviour from a neuron.

There were many units which were only found by one sorter. Some of these units may plausibly be reflective of real neurons, but many are clearly false positives. A handful of unmatched units are shown in Figure 4.4. In some cases it is clear that a unit is not to be trusted. But visually inspecting each unit is time consuming and does not always provide a clear picture about whether the unit corresponds to a neuron.

This raises the question of how to determine, given a limited (possibly only 1) sorting, which units are likely to be true positives. The rest of this section examines the relationship between sorter algorithm used and unit agreement.

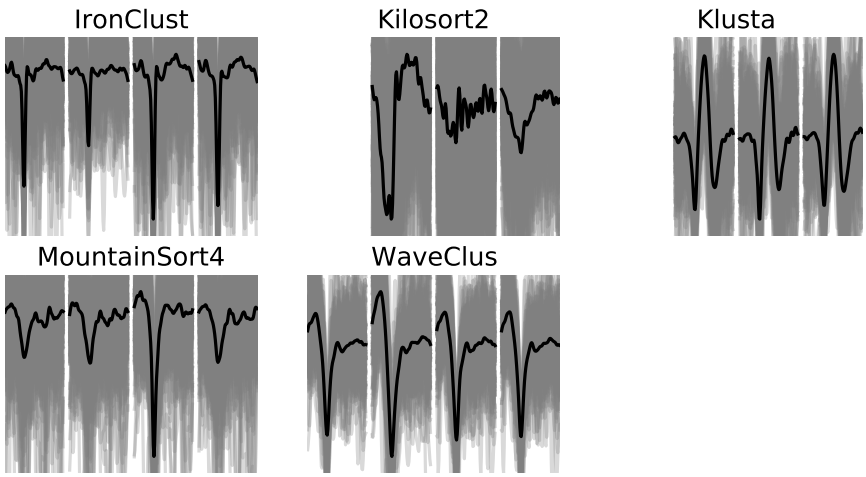

Figure 4.4: Selection of unmatched units from each sorter. Units were selected at random from the pool of unmatched units, and do not correspond to the same neuron.

For each recording, the set of units found was split into two subsets: matched and unmatched units. The "matched" subset contained units (from all sorting outputs) which were found by at least two sorting algorithms. The "unmatched" subset contained units which were found by only one sorting algorithm.

Figure 4.5 (left) shows the total number of matched and unmatched units across all sorters for each recording. Matched units are shown in red, whilst all other units are displayed in blue.

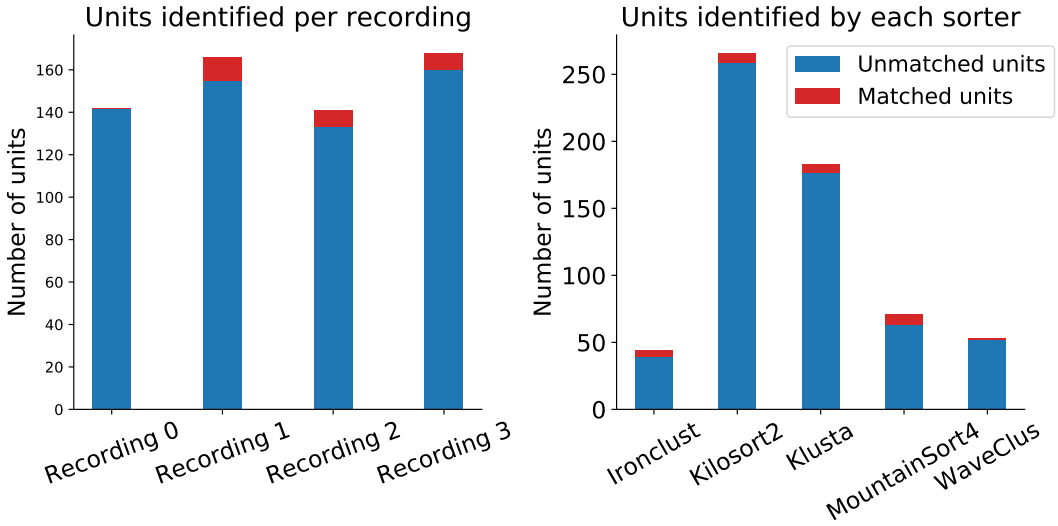

Figure 4.5: Left: bar chart demonstrating the total number of units identified in each recording, matched units are represented in red. Right: bar chart demonstrating the total number of units identified by each sorter, again matched units are represented in red.

Across all sorters, 142 units were found in recording 0, of which all were unmatched.

166 units were found in recording 1, of which 11 matched (6.63%). 141 units were identified in recording 2, of which 8 units matched (5.67%). 8 matched units were found in recording 3, of a total of 168 units (4.76%).

There is a "double counting" issue when agreement is compared in this way. Since matched units are seen over a number of sorters, they each produce multiple entries in the matched category for that recording. One may instead choose to count matched units only once - in which case agreement is reduced to only 3 matched units in recordings 1 and 3, and only 2 matched units in recording 3. However, the "duplicate" counts of units still differ in their metrics, and constitute multiple estimates of a neuron (albeit the same neuron) therefore all entries for a matched unit are included in the analysis.

Results here provide a discouraging indication that the initial hypothesis 4.1, that Kilosort2, MountainSort4 and WaveClus find many of the same units, was false. Additionally, the completely distinct results produced between sorters for recording 0 further emphasise the need for unified evaluation methods among sorters.

In Buccino et al's study, agreement between sorters was examined on a dataset produced with high density recording devices. Their analysis included HerdingSpikes2, SpyKINGCircus, Tridesclous, and HDSort which are not included in this analysis. Their analysis did also include Kilosort2 and IronClust, which are used in this analysis. Whilst the data and specific sorting algorithms used in that paper differed from this project, a similar general result of low agreement between sorters was found [6].

Given that Buccino et al.'s analysis was dealing with a far larger number of electrodes per recording, it is expected that there are far more units in general identified. 2031 units were identified across their 6 sorters, in which only 263 units are agreed by two or more sorters. Agreement criteria was the same as that used in this project (50% spike train match). Overall 12.95% of units were agreed by 2 or more sorters; only slightly more than that of this analysis of only 5 sorters.

By comparison, in this analysis, across the 4 recordings, there were on average 4.27% matched units. Even with 5 sorting algorithms, only a small number of matched units are identified. This echoes the results presented by Buccino et al. insofar as both analyses saw very low agreement across sorters.

In the scenario in which only one sorting algorithm can be executed, which one should be used? The next section addresses the hypothesis that sorters do not contribute to the set of matched units equally. In doing this, one sorter is identified as producing more reliable results.

4.2.2 MountainSort4 identifies all matched units

To examine the question of which sorters tend to produce more matched units, the data was next split based on sorting algorithm used rather than recording. Figure 4.5 (right) shows matched and unmatched units as counted across all recordings split by sorter. The "double counting" problem mentioned above does not apply when looking at agreement by sorter - there are 8 agreed upon units across all recordings, and agreement is measured in terms of the number of these units contributed to by each

sorter. Thus, if a unit is seen by other sorters, it is nonetheless counted only once for that sorter. The body of matched units for a given sorter is therefore a representation of that sorter's contribution to the total set of matched units.

MountainSort4 shows the highest number of units in agreement with other sorters, with all of the 8 matched units being found by this sorter. MountainSort4 identified 71 units in total, meaning 11.27% of the units identified by this method were matched units.

IronClust achieved a higher ratio of matched units than all other sorting methods (11.36% of the 44 units identified were matched units). This high ratio, comes at the cost of lower recall, as only 5 of the 8 matched units were found by IronClust.

Kilosort2 showed promising results on first glance, as 7 of the 8 matched units were found by this sorter. Having said that, Kilosort2 did identify a large number of units overall (266) meaning the ratio of matched units was only 2.63%. Klusta identified 183 units of which 6 were matched, giving a ratio of only 3.28%. Klusta showed similar behaviour to Kilosort2 in the sense that both found a reasonable number of matched units relative to others, but did so at the cost of generating a high number of false positive units.

WaveClus produced disappointing results, with only one unit in agreement with any other sorter (of a total of 53 units, 1.89% of units agree). Given that IronClust managed to identify 5 matched units from fewer units overall, it is disappointing to see that WaveClus demonstrated such low agreement with other sorters in terms of both ratio and count of matched units. The one matched unit that was identified by WaveClus is the one unit across all recordings which was found by all sorters.

In answer to the question of which sorter could be considered most trustworthy, MountainSort4 seems the best candidate based on these results. Not only was this the only sorter to identify all 8 units, the ratio of matched to unmatched units was among the best seen in terms of ratio of matched units. This echoes the results found by Magland et al., mentioned in section 2.2 [16].

Looking forward to future work, it would be useful to examine whether this finding generalises to the full dataset. If MountainSort4 is able to consistently find all matched units, this sorter may prove the best choice in the scenario that only one sorter can be used. Given suitable curation methods, for example that described in section 4.4, this would allow for the identification of all matched units on the basis of only one sorting; effectively allowing the consensus method to be used without the overhead of executing multiple sorters.

4.3 Metrics relate to unit agreement

This leads to the main hypothesis of this project: Unit quality metrics provide information which can be used to differentiate matched units from unmatched units.

To address this hypothesis, the Mann-Whitney U test was used to establish whether there is a statistically significant difference in metrics between matched and unmatched units. This test determines whether the probability of a matched sample having a "bet-

ter" metric than an unmatched sample is higher than the probability of an unmatched sample having a better metric than a matched sample [9]. Whether a higher or lower value is considered "better" depends on the metric in question, as laid out in the section 3.4.

There are two reasons why this test is appropriate for providing initial insight into the relevance of metrics to agreement. Firstly, as can be seen in Figure 4.6, distributions for each metric cannot reasonably be described as approximately Gaussian. Secondly, as demonstrated in the section 4.2, there was very low agreement across sorters. This led to a small sample size for the matched unit population, therefore a non-parametric test is most suitable.

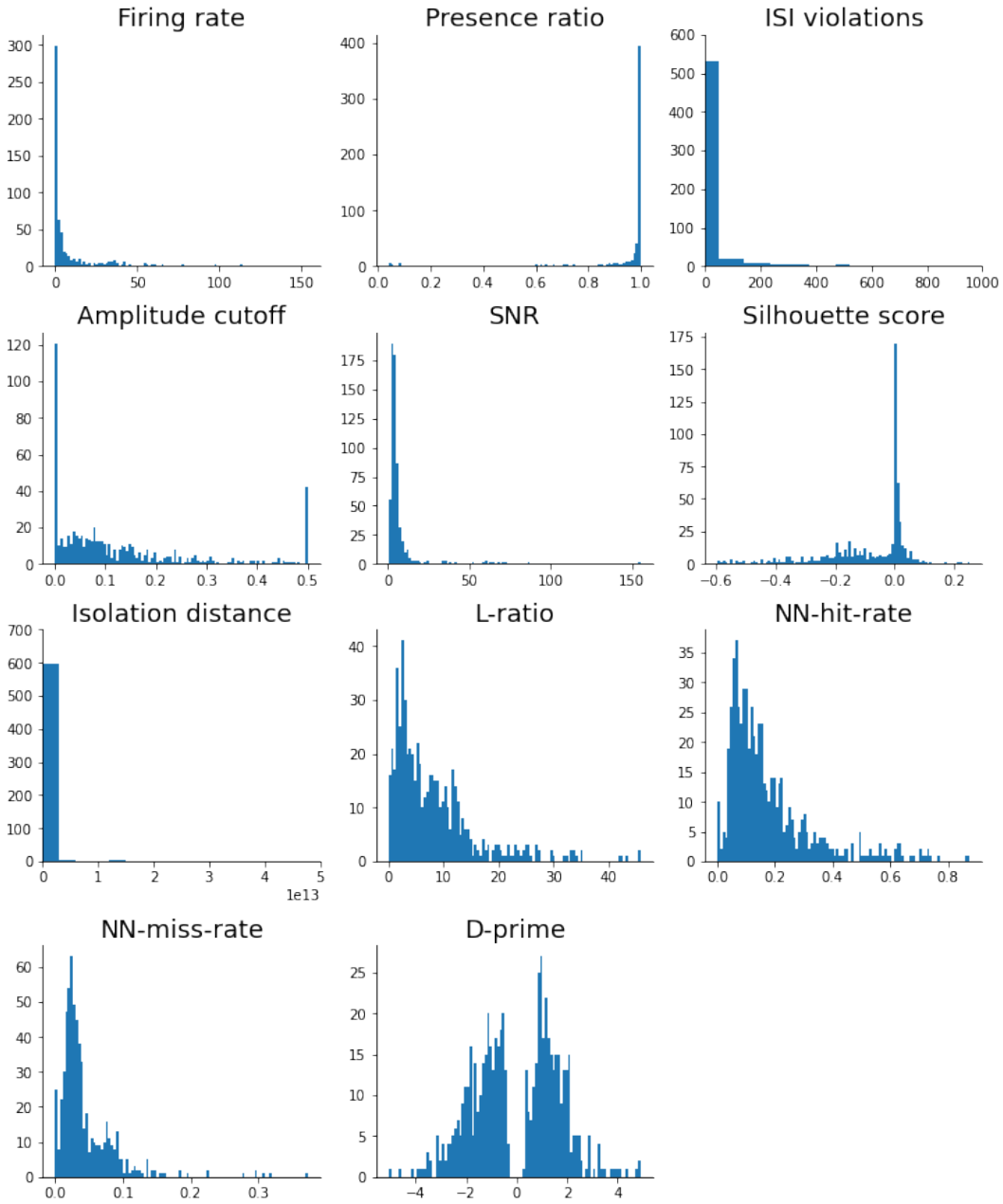

Figure 4.6: Histograms displaying the distribution for individual metrics.

Matched and non-matched unit groups are here considered to be independent groups with independent observations in each group. That is, no (recording, sorter, unit) tuple is represented more than once in the dataset. A significance level of 5% is used throughout. For brevity, individual hypotheses will not be formally stated for each metric, as all follow the same general form:

- H_0: The distributions for the matched and unmatched populations are equal (in terms of central tendency).
- H_1: The distributions for the matched and unmatched populations are not equal.

4.3.1 Mann-Whitney U test on pooled data

As shown in table 4.1, at a 5% significance level there are only two metrics which are not do not produce significant results: nearest neighbour hit rate and D-prime. Box plots for each of the significant metrics are displayed in Figure 4.7.

Metric	Mann-Whitney U test score
	Pooled recordings
Firing rate	**P<0.001, (U= 2344.0)**
Presence ratio	**P<0.001, (U=4968.0)**
ISI violations	**P<0.001, (U=2432.5)**
Amplitude cutoff	**P= 0.002, (U= 5356.0)**
Silhouette score	**P<0.001, (U= 4517.0)**
Isolation distance	**P<0.001, (U= 4166.0)**
L-ratio	**P<0.001, (U= 2270.0)**
NN miss rate	**P<0.001, (U= 3102.0)**
NN hit rate	P= 0.208, (U= 7228.0)
SNR	**P= 0.001, (U= 5250.0)**
D-prime	P= 0.192, (U= 7177.0)

Table 4.1: Mann-Whitney U test P values and test scores for pooled recordings. Significant results are displayed in bold font.

These results seem generally in agreement with expectations. Presence ratio is generally 1 in matched units, with lower values predominantly seen in the unmatched population. ISI violations are less common in the matched population. Isolation distance is higher in the matched population. This seems reasonable given that such units should be easier to identify, because they present well separated clusters. L-ratio sits lower in the matched population, again indicating that matched units score "better".

Silhouette scores for matched units sit very close to 0, whilst unmatched units take on a wide array of values. As an estimate of the miss rate, it is unsurprising that matched units generally take on a lower amplitude cutoff value. Finally, nearest neighbour miss rate does produce a significant P-value, but the distributions are quite similar visually, giving a mixed indication of the relevance of this metric to unit agreement.

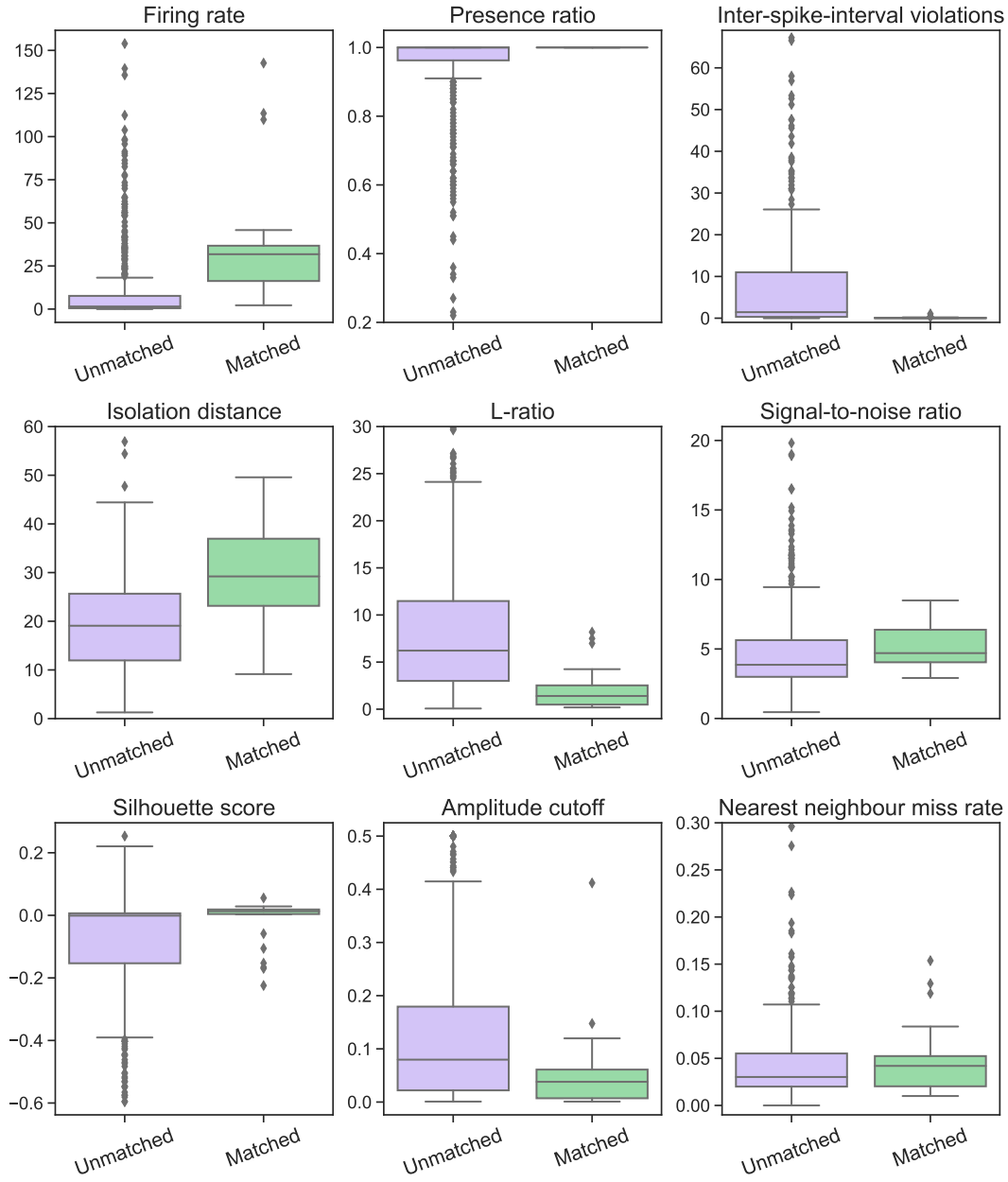

Figure 4.7: Box plots showing distributions for each metric. Data is here pooled across sortings and recordings.

Here several metrics have been highlighted as relevant to the agreement outcome of a unit. The remainder of this section examines these results in more detail in order to gain more insight into the relationship between individual metrics and agreement outcomes, and establish whether this relationship applies to different data and different sorting algorithms.

4.3.2 Mann-Whitney U test on individual recordings

Ideally, the relationship between metric and agreement outcome should generalise well over different recordings. With a small sample set such as this, it is difficult to "hold

out" data to test conclusions. In the absence of additional samples to test conclusions, this paper aims to maximise robustness of general claims by examining influences on metrics in more detail. To that end, the same test was performed on each test individually. Results from this test are summarised in table 4.2 and 4.3.

Metric	Mann-Whitney U test score Results		
	Recording 1	Recording 2	Recording 3
Firing rate	**P<0.001**	**P<0.001**	**P<0.001**
Presence ratio	**P=0.017**	**P=0.005**	**P=0.006**
ISI violations	**P<0.001**	**P=0.004**	**P<0.001**
Amplitude cutoff	**P=0.004**	P=0.093	P=0.241
Silhouette score	**P=0.003**	**P=0.032**	P=0.053
Isolation distance	**P=0.001**	**P=0.005**	**P=0.022**
L-ratio	**P<0.001**	**P<0.001**	**P=0.001**
NN miss rate	**P=0.001**	**P<0.001**	**P=0.022**
NN hit rate	P=0.222	**P<0.001**	P=0.189
SNR	P=0.069	**P=0.030**	P=0.051
D-prime	P=0.129	P=0.356	P=0.097

Table 4.2: Mann-Whitney U test P-values for individual recordings. Significant results are displayed in bold font.

Metric	Mann-Whitney U test score values		
	Recording 1	Recording 2	Recording 3
Firing rate	**U=308.0**	**U=161.0**	**U=167.0**
Presence ratio	**U=594.0**	**U=264.0**	**U=332.0**
ISI violations	**U=305.0**	**U=230.5**	**U=131.0**
Amplitude cutoff	**U=447.0**	U=383.0	U=545.0
Silhouette score	**U=429.0**	**U=324.0**	U=422.0
Isolation distance	**U=357.0**	**U=243.0**	**U=370.0**
L-ratio	**U=213.0**	**U=109.0**	**U=239.0**
NN miss rate	**U=378.5**	**U=119.5**	**U=370.0**
NN hit rate	U=734.0	**U=157.0**	U=521.0
SNR	U=623.0	**U=320.0**	U=420.0
D-prime	U=678.0	U=490.0	U=465.0

Table 4.3: Mann-Whitney U test scores for individual recordings. Significant results are displayed in bold font.

Firing rate, presence ratio, ISI violations, isolation distance, L-ratio and nearest neighbour miss rate produced significant results in all three recordings in addition to the pooled data. Despite showing significance when using pooled data, SNR, silhouette score and amplitude cutoff did not produce a significant result in at least one of the three recordings. Results from individual recordings echoed those of the pooled data insofar as both tests failed to produce a significant results for either nearest neighbour hit rate or D-prime.

The firing rate is significant in all three recordings at a 5% significance level (and indeed at a 1% significance level). As can be seen in Figure 4.8, the firing rate is generally higher in units which are matched. This is in agreement with the relationship seen in the pooled data.

Presence ratio showed consistency with results from the pooled data, with matched units taking values close to 1 in matched units, but with more variation in values for unmatched units.

The isolation distance metric also showed consistent results across the three recordings individually and pooled, with matched units typically displaying higher values (better isolation).

As can be seen in Figure 4.8, matched units tend to have ISI violation values at or close to 0, whereas unmatched units generally show ISI violations in the range $[0, 20]$. This is in alignment with previous results as well as expectations.

L-ratio is consistently lower in units that did have matches, as can be seen in Figure 4.8. This is expected. A low L-ratio value indicates a well separated cluster, therefore it seems plausible that such units are easier for sorting algorithms to identify.

Nearest neighbour miss rate was a particularly interesting example. Despite producing significant results in all tests, closer inspection of the recording specific data reveals that there is no consistent relationship between the metric values and the agreement score. As can be seen in Figure 4.8, the relationship between the means in each population differed across recordings. In recordings 1 and 3, nearest neighbour miss rate was lower in the unmatched population. In recording 2 mean nearest neighbour miss rate was lower in the matched population. Visually, the distributions show significant overlap in all three recordings. This gives a mixed picture of the relevance of nearest neighbour miss rate as an indicator of likely unit agreement. A similar metric, nearest neighbour hit rate, was only found to be significant in recording 2.

Of all the metrics tested in this project, D-prime was the only metric that failed to produce significant results in any recording. It is surprising that SNR was not found to be significant in this data, as this is a widely used metric across sorting methods [14, 19]. In Buccino et. al's study, a positive correspondence between SNR and high agreement units was identified [6]. Results found here are therefore counterintuitive - high SNR units should be easier to detect and therefore more consistently found by sorters.

Results presented here are encouraging in that some of the metrics used in this project may be useful in determining whether a unit will be found by many sorters. In particular firing rate, presence ratio, ISI violations, isolation distance and L-ratio demonstrate clear results. Metrics which showed significant and consistent relationships with unit agreement across both pooled and individual recordings suggest that these metrics not only relate to unit agreement, but do so in a way that applies to individual recordings as well as generalising over recordings.

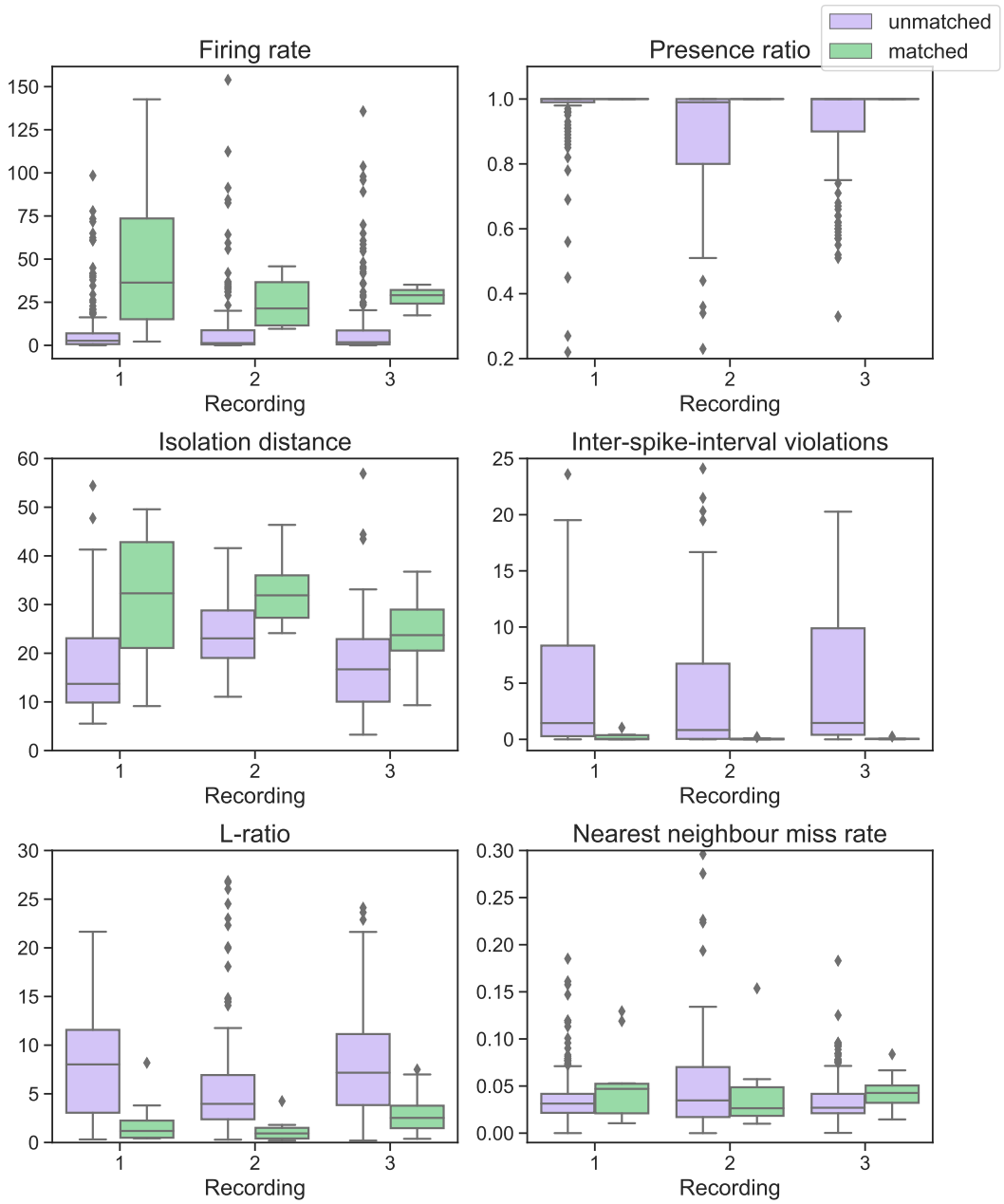

Figure 4.8: Box plots showing distributions for each significant metric. Data is here pooled across sortings but split by recording.

4.3.3 Welch's t-test on individual recordings

Welch's t-test with unequal variance is recommended for reliability of results when a Mann-Whitney U test produces significant results [9]. Therefore, using only the metrics highlighted above, a single tailed Welch's t-test was performed for each recording using a 5% significance level.

Where the Mann-Whitney U test addresses the hypothesis that one distribution is greater than the other, Welch's t-test addresses the hypothesis that the mean of each the matched and unmatched populations differ. Results are displayed in tables 4.4 and

4.5.

Metric	t-test score - P-values		
	Recording 1	Recording 2	Recording 3
Firing rate	**P=0.022**	**P= 0.033**	**P<0.001**
Presence ratio	**P<0.001**	**P<0.001**	**P<0.001**
ISI violations	P=0.114	P=0.156	**P=0.045**
Isolation distance	P=0.286	P= 0.172	P=0.088
L-ratio	**P<0.001**	**P<0.001**	**P=0.001**

Table 4.4: Welch's t-test P values for individual recordings. Significant results are displayed in bold font.

Metric	t-test score - t-values		
	Recording 1	Recording 2	Recording 3
Firing rate	**t=2.703**	**t=2.503**	**t=5.519**
Presence ratio	**t=3.705**	**t=6.394**	**t=6.483**
ISI violations	t=-1.588	t=-1.426	**t=-2.024**
Isolation distance	t=-1.070	t=-1.372	t=-1.714
L-ratio	**t=-7.121**	**t=-6.618**	**t=-4.702**

Table 4.5: Welch's t-test test scores for individual recordings. Significant results are displayed in bold font.

Under this test, firing rate, presence ratio and L-ratio were all found to be significant in all three recordings, indicating that the mean values for these populations differ significantly. This further supports the claim that these three metrics are relevant to unit agreement.

Inter-spike-interval violations were only found to be significant in recording 3. Isolation distance was not found to be significant in any of the recordings. As can be seen in Figure 4.6, both ISI violations and isolation distances can not reasonably be described as Gaussian, which may be the cause of this failure to produce significant outcomes.

4.3.4 Mann-Whitney U test on individual sorting methods

The SpikeForest project conducted similar work to use quality metrics as predictors of accuracy [16]. Their analysis, which focused on SNR, mean firing rate and ISI violations, showed that the relationship between metrics and accuracy was itself highly dependent on the sorter used. Specifically, SNR and ISI violations were both found to be predictive of accuracy when using IronClust (firing rate was not predictive). When using Kilosort2 and MountainSort4 however, firing rate is the only predictive metric of the three. In their study, a linear predictor using the three metrics was found to produce the best prediction of accuracy.

To investigate the relationship between sorters, metrics and accuracy, the Mann-Whitney

U test was again used to establish whether metrics differed between matched and unmatched populations.

Metrics which had failed to produce significant results in previous tests were excluded from this stage of the analysis since generality across recordings is essential to the purposes of this project. Additionally, since WaveClus only produced one agreed unit, it was excluded from this test.

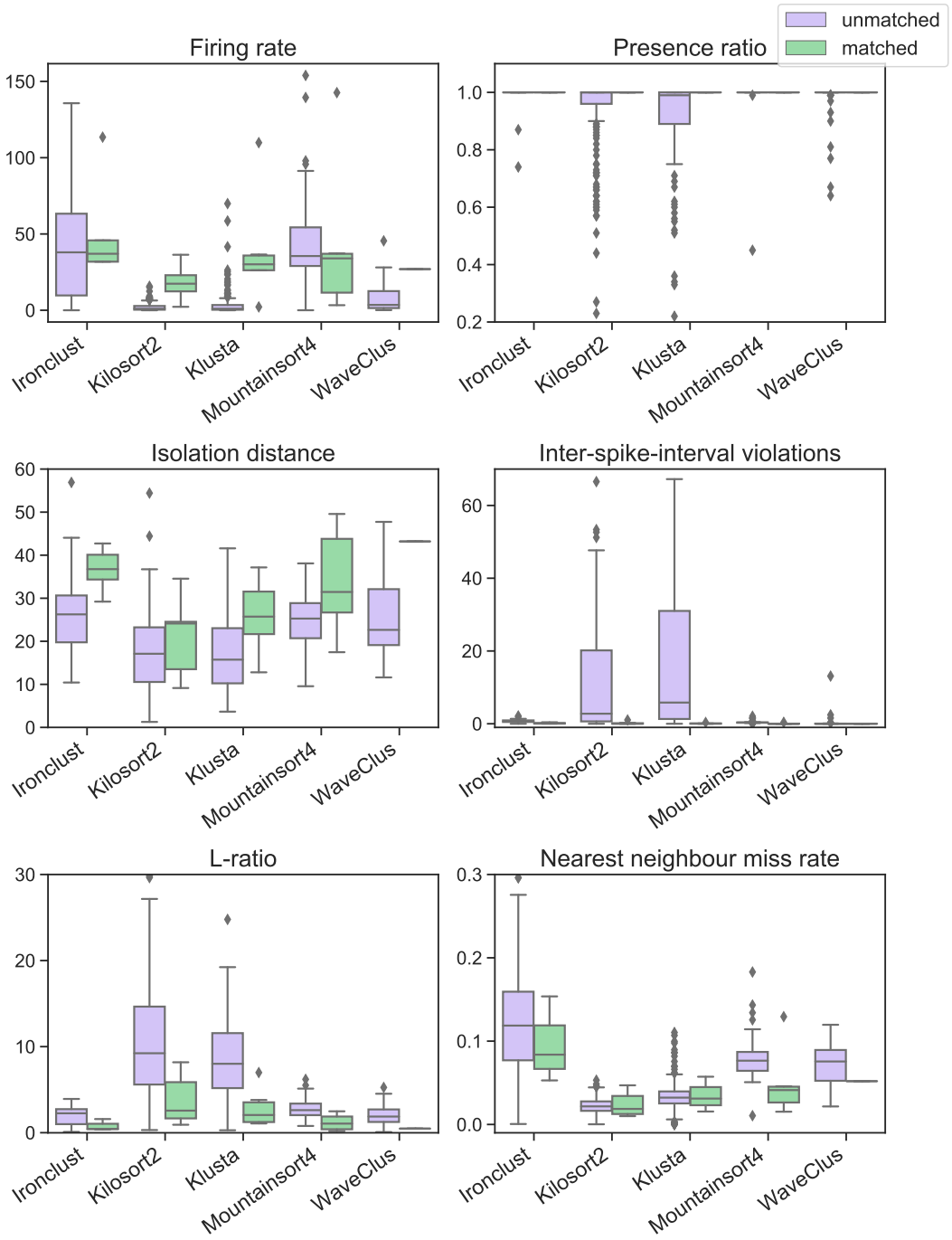

Figure 4.9: Box plots showing distributions for each significant metric. Data is here pooled across recordings but split by sorting algorithm used.

Of the five metrics tested, only ISI violations and L-ratio proved statistically significant across all sorters. ISI violations were found to be significant for IronClust (P=0.005, U=27.0), Kilosort2 (P<0.001, U=217.0), Klusta (P<0.001, U=104.0) and MountainSort4 (P<0.001, U=104.0). Lower values were consistently seen in the matched population compared to the unmatched population. L-ratio was found to be significant for IronClust (P=0.006, U=29.0), Kilosort2 (P=0.001, U=277.0), Klusta (P<0.001, U=106.0) and MountainSort4 (P<0.001, U=66.0). Again, lower values were seen consistently in the matched population compared with the unmatched population.

Firing rate was found to be significant for both Kilosort2 (P<0.001, U=82.0) and Klusta (P<0.001, U=65.0). In both cases, matched units tended to have higher firing rates than unmatched units. Firing rate was not found to be significant in IronClust or MountainSort4. Despite the lack of significance in the latter two sorters, it is interesting that matched units seem to generally take lower values than unmatched units. Nonetheless, given the lack of significance here it is difficult to draw any conclusions on the basis of visual inspection alone.

Presence ratio was found to be significant for Kilosort2 (P=0.019, U=539.0) and Klusta (P=0.009, U=243.0). This was due to substantial numbers of unmatched units displaying values well below 1 (see Figure 4.9) - more so than any of the other sorters. This metric was not found to be significant in IronClust or MountainSort4.

Isolation distance was found to be significant for IronClust (P=0.021, U=42.0), Klusta (P=0.020, U=268.0) and MountainSort4 (P=0.020, U=138.0). Matched units tended to have larger values for this metric. Isolation distance was not found to be significant for Kilosort2.

SNR was also tested on IronClust, Kilosort2 and Klusta out of interest in Magland et al.'s results. It was found to be significant for IronClust (P=0.016, U=39.0) and Kilosort2 (P=0.005, U=389.0) but not Klusta. Pleasingly, the results of this test for IronClust echo those found by Magland et al.. ISI and SNR differentiate populations well, whilst firing rate does not produce significant results [16]. Kilosort2 and Klusta also showed significance in firing rate, but additionally SNR was significant for Kilosort2, and both Kilosort2 and Klusta produced significant results for ISI violations.

Overall, ISI violations and L-ratio were the only metrics which produced significant results across all tests. By testing on each recording, each sorter, and using pooled results, these metrics have demonstrated consistent relationships between matched and unmatched populations. With this in mind, these metrics are here considered to be the most useful in determining metric agreement.

4.4 Metrics predict unit agreement

Up to this point, metrics have been addressed individually, but there is scope for combining metrics to give a more comprehensive description of units. Ideally, metrics should be used to classify points based as either likely to be matched, or unlikely to be matched. Given this, metrics used in such a model should have a consistent relationship with agreement outcome. For this reason, ISI violations and L-ratio were used to

produce a baseline model. Development of such a predictor would allow for elimination of at least some false positives, in the absence of multiple sortings or ground truth data. The rest of this section describes one possible baseline approach which produces promising results.

In the later stage of this project, it was possible to attain a larger body of spike sorted data. Nine recordings were used, producing a total of 1583 units, of which 73 were matched units. Since both ISI violations and L-ratio appeared to be approximately log-normal distributed, values for both variables were log-transformed to produce more plausibly Gaussian distributed data (see Figure 4.10). After removing invalid inputs and outliers, this left 1401 units for classification, of which 59 were matched units.

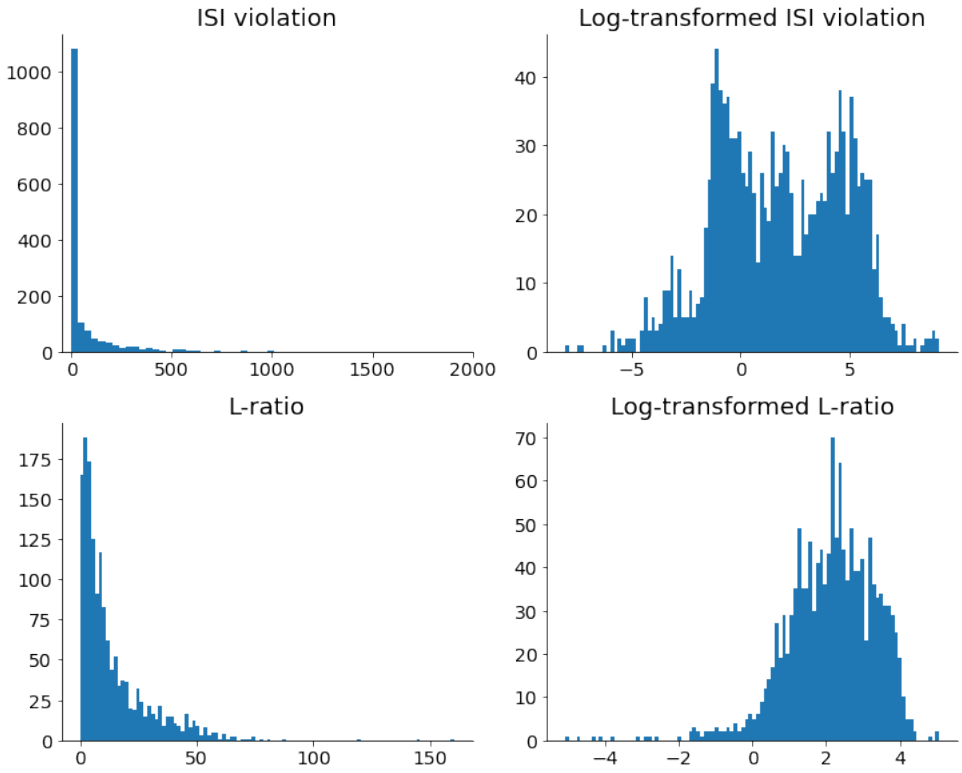

Figure 4.10: Histograms for metric values before and after log-transformation. Upper left: ISI violations. Upper right: log-transformed ISI violations. Lower left: L-ratio. Lower right: log-transformed L-ratio.

A Gaussian Bayes model was used. This type of model has the benefit of providing a probability that a given unit will be assigned to each class, in addition to a prediction [4]. This is particularly useful in cases such as these where the value of false positives and false negatives is particular to the model user's purposes. For example, if results from a sorting output need to be curated, but some false positives are acceptable, the user can access the probability of each unit being assigned to the "matched" class and elect to keep all units which have a probability of being matched equal to 0.3 or more. By contrast, in settings where false positives are unacceptable, one may choose to keep units which have a higher probability of being matched.

Another reason why unequal priors are a suitable choice in this case are that recall is here prioritised over precision. After curation (that is, keeping all data which was classified as likely to be matched) a few remaining false positives are not as problematic as elimination of true positive units. This is because a unit which is a true positive, but is classified as unmatched would reduce the already small sample size of true units. By contrast, a false positive which remains after curation is undesirable, but not so catastrophic. Unless all units are falsely classified as matched units, the number of false positives in the curated data will still be lower than that of the data before curation.

To train the model, data was shuffled randomly and split to train and test sets using a ratio of 80:20. Gaussians were then fit to the training set, one for the matched data and one for the unmatched data. The respective probability distributions were then used to calculate the probability of some test point belonging to each distribution according to Bayes rule.

To predict, the probability of each test point belonging to each classifier was calculated and the higher of the two was used to assign the test point to a class. By comparing results directly, the model uses equal prior probabilities for each class. The choice to use equal priors will be examined in more detail in work conducted next year. For a baseline model, this choice was made to provide initial insight into the viability of this kind of predictor.

Figure 4.11 displays the results of fitting such Gaussians. On the left, the model is fitted using the naive Bayes assumption, which asserts that features are independent. On the right, the model is fitted without such an assumption. That is, in the second model, covariance between features is included.

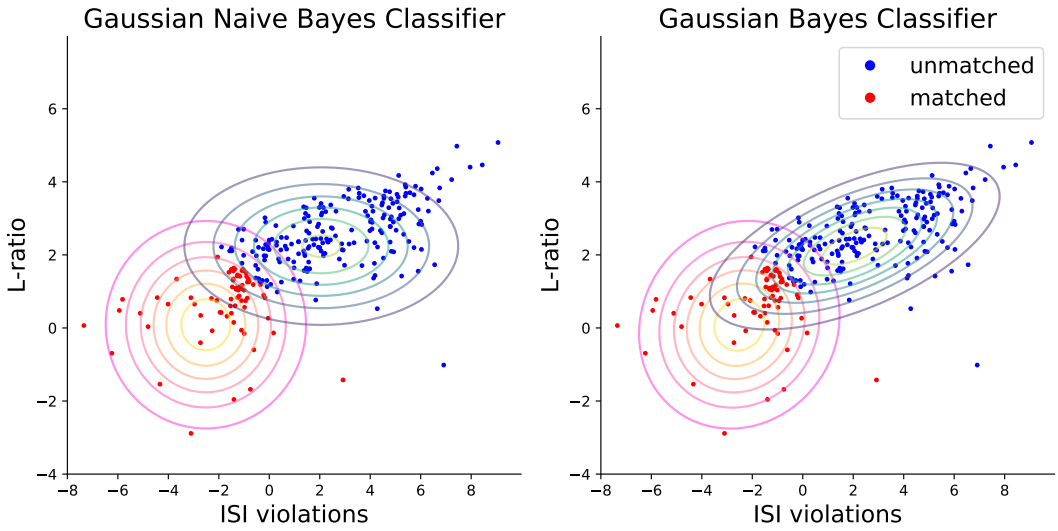

Figure 4.11: Left: Contour plot showing Gaussians fit to both unmatched and matched training populations under the naive Bayes assumption. Right: Contours shown for Gaussians fit to training data without the naive Bayes assumption. Test data are coloured based on predicted value. Contour plots are coloured blue for the unmatched population Gaussian and pink for the matched population Gaussian.

Even in the naive case, the model performs with an accuracy of 77.9% on the test set, correctly identifying 10 of the 11 matched units. Model 2 performs with an accuracy of 84.0% on the test set, correctly identifying all of the matched units in the test set. Figure 4.12 shows heat maps for the classification of units in each model.

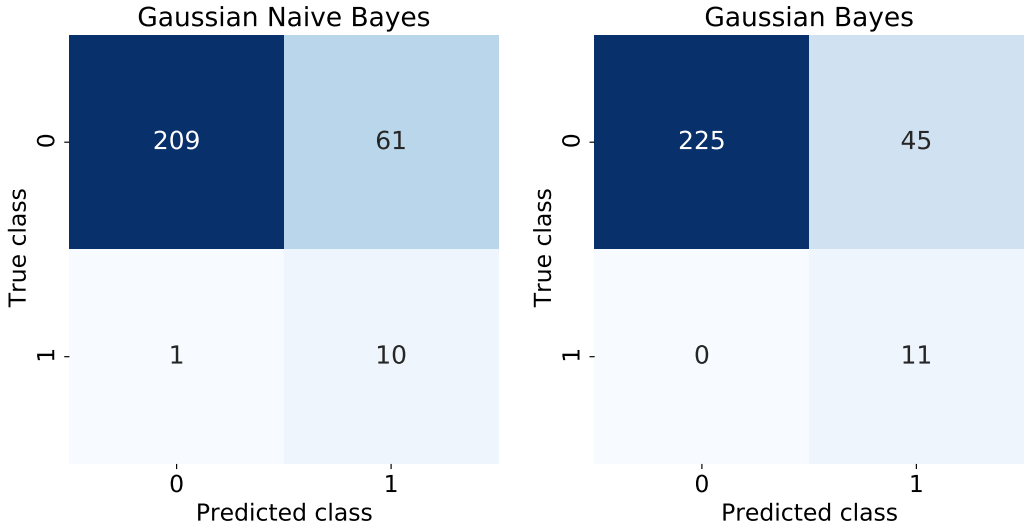

Figure 4.12: Left: Heat map for classifications made by Gaussian Naive Bayes model. Right: Heat map for classifications made by Gaussian Bayes model. Values represent the number of units in each category, as predicted by the two models.

This predictor presents one way to reduce a large number of units, many of which are inevitably false positives, to a more manageable number of units without discarding invaluable true positive unit samples.

Chapter 5

Conclusions

This project has examined the relationship between choice of spike sorting algorithm and accuracy of units identified. This analysis corroborated several results from previous studies. Firstly, little agreement was found between sorting outputs. Secondly, Kilosort2 was found to produce many false positive units. Kilosort2 and Klusta produced many more units than could be biologically plausibly reflective of real neuron activity detected by a tetrode. Thirdly, MountainSort4 was identified as the most reliable sorter, having contributed to all matched units across recordings without producing an implausible number of units overall.

The relationship between individual metrics and agreement was then evaluated. Metrics differed significantly between matched and unmatched unit populations. Whilst many metrics were generally better in the matched population, some metrics produced surprising results. Notably SNR was not consistently found to be significant, and D-prime produced no significant results. Some metrics were found to have significant differences between matched and unmatched populations over all sorters and recordings, demonstrating that some metrics may be useful even when the result is generalised over recordings. Additionally, this suggests that these metrics may be useful when generalising over a greater number of sorting algorithms.

Choice of sorting algorithm proved to influence the importance of metrics, echoing results found in previous studies. Finally, a baseline prediction model was presented using ISI violations and L-ratio. As good results have been obtained from this small dataset, containing just nine recordings, it is expected that further improvements can be made to this model next year. This model is able to predict unit accuracy based on metrics alone and does not require multiple sortings or ground truth data. The impact of such a model presents an easy-to-use method that proves more effective and less expensive than previous evaluation tools.

Chapter 6

Future work

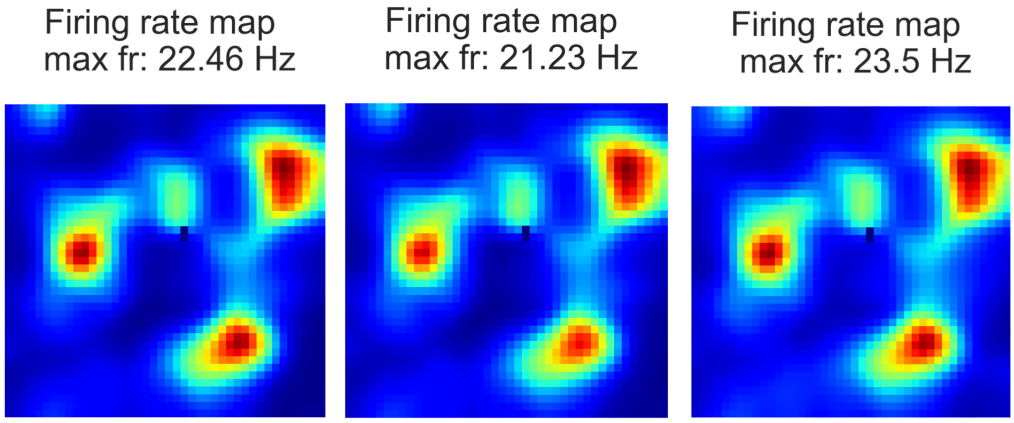

Figure 6.1: Firing rate maps for one neuron from three different sorters. Colour represents firing activity at a given point on the mouse's trajectory. Red is used to represent high firing rate, whereas blue is used to indicate low firing rate. Left: Mountainsort4. Centre: Kilosort2. Right: SpyKINGCircus [27].

The focus of next year will be to use the tools and results presented here to conduct analysis on the full dataset. Specifically, the predictor model presented here will be developed in order to allow identification of true positive units on the basis of only one sorting. This will involve extending the model to make use of the full dataset, and conduct further testing and model improvements. Successful development of such a model will allow for false positive units to be extracted without relying on the use of multiple sorters or ground truth data.

In particular, having identified MountainSort4 as the most reliable sorter in this analysis, the model presented here will be used to extract true positive units from the full set of recordings based on only this sorting algorithm. Using these units, spatial properties of the neurons will be computed and compared. Using the original post-processing pipeline, full analysis will be performed on these neurons. This will provide an interpretation of these units, similar to those used in the original analysis presented by Gerlei et al. [10].

Figure 6.1 demonstrates the final output of the full analysis pipeline for one matched unit, using three different sorters: MountainSort4, KiloSort2 and SpyKINGCircus[1] [27].

The overall goal of future work will therefore be to examine the overall impact that sorting algorithm choice has on final interpretation of results from extracellular recordings.

[1]SpyKINGCircus was not included in this year's analysis, but can be used within the analysis pipeline.

Bibliography

[1] Kilosort2. https://github.com/MouseLand/Kilosort2. Accessed: 2021-04-12.

[2] MountainSort3. https://github.com/flatironinstitute/mountainsort. Accessed: 2021-04-12.

[3] SpikeInterface. https://github.com/SpikeInterface. Accessed: 2021-04-12.

[4] Luc devroye, laszlo gyorfi and gabor lugosi, a probabilistic theory of pattern recognition (springer, new york, 1996) 636 pages. *Discrete Applied Mathematics*, 73(2):192–194, 1997.

[5] Alex H. Barnett, Jeremy F. Magland, and Leslie F. Greengard. Validation of neural spike sorting algorithms without ground-truth information. *Journal of Neuroscience Methods*, 264:65–77, 2016.

[6] Alessio P. Buccino, Cole L. Hurwitz, Jeremy Magland, Samuel Garcia, Joshua H. Siegle, Roger Hurwitz, and Matthias H. Hennig. Spikeinterface, a unified framework for spike sorting. *bioRxiv*, 2019.

[7] Fernando J. Chaure, Hernan G. Rey, and Rodrigo Quian Quiroga. A novel and fully automatic spike-sorting implementation with variable number of features. *Journal of Neurophysiology*, 120(4):1859–1871, 2018.

[8] Jason E. Chung, Jeremy F. Magland, Alex H. Barnett, Vanessa M. Tolosa, Angela C. Tooker, Kye Y. Lee, Kedar G. Shah, Sarah H. Felix, Loren M. Frank, and Leslie F. Greengard. A fully automated approach to spike sorting. *Neuron*, 95(6):1381–1394.e6, 2017.

[9] W. J. Conover. *Practical nonparametric statistics*. Wiley series in probability and statistics. Applied probability and statistics. Wiley, second edition, 1980.

[10] Klara Gerlei, Jessica Passlack, Ian Hawes, Brianna Vandrey, Holly Stevens, Ioannis Papastathopoulos, and Matthew F. Nolan. Grid cells are modulated by local head direction. *Nature Communications*, 11(1), 2020.

[11] Kenneth D. Harris, Darrell A. Henze, Jozsef Csicsvari, Hajime Hirase, and György Buzsáki. Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. *Journal of Neurophysiology*, 84(1):401–414, 2000.

[12] Matthias H. Hennig, Cole Hurwitz, and Martino Sorbaro. Scaling spike detection and sorting for next-generation electrophysiology. *Advances in neurobiology*, 22:171–184, 2019.

[13] Darrell A. Henze, Zsolt Borhegyi, Jozsef Csicsvari, Akira Mamiya, Kenneth D. Harris, and György Buzsáki. Intracellular features predicted by extracellular recordings in the hippocampus in vivo. *Journal of Neurophysiology*, 84(1):390–400, 2000.

[14] 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, 2011.

[15] Cole L. Hurwitz, Kai Xu, Akash Srivastava, Alessio Paolo Buccino, and Matthias Hennig. Scalable spike source localization in extracellular recordings using amortized variational inference. *bioRxiv*, 2019.

[16] Magland Jeremy, James J. Jun, Elizabeth Lovero, Alexander J. Morley, Cole L. Hurwitz, Alessio P. Buccino, Samuel Garcia, and Alex H. Barnett. Spikeforest, reproducible web-facing ground-truth validation of automated neural spike sorters. *eLife*, 9, 2020. Copyright - © 2020, Magland et al. This work is published under http://creativecommons.org/licenses/by/4.0/ (the "License"). Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License; Last updated - 2020-06-01.

[17] James J. Jun, Catalin Mitelut, Chongxi Lai, Sergey L. Gratiy, Costas A. Anastassiou, and Timothy D. Harris. Real-time spike sorting platform for high-density extracellular probes with ground-truth validation and drift correction. *bioRxiv*, 2017.

[18] Ryan C. Kelly, Matthew A. Smith, Jason M. Samonds, Adam Kohn, A. B. Bonds, J. Anthony Movshon, and Tai Sing Lee. Comparison of recordings from microelectrode arrays and single electrodes in the visual cortex. *Journal of Neuroscience*, 27(2):261–264, 2007.

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

[20] Cyrille Rossant, Shabnam N Kadir, Dan F M Goodman, John Schulman, Maximilian L D Hunter, Aman B Saleem, Andres Grosmark, Mariano Belluscio, George H Denfield, Alexander S Ecker, and et al. Spike sorting for large, dense electrode arrays. *Nature Neuroscience*, 19(4):634–641, 2016.

[21] N. Schmitzer-Torbert, J. Jackson, D. Henze, K. Harris, and A.D. Redish. Quantitative measures of cluster quality for use in extracellular recordings. *Neuroscience*, 131(1):1–11, 2005.

[22] Neil Schmitzer-Torbert and A. David Redish. Neuronal activity in the rodent dorsal striatum in sequential navigation: Separation of spatial and reward responses on the multiple t task. *Journal of Neurophysiology*, 91(5):2259–2272, 2004.

[23] Joshua H Siegle, Aarón Cuevas López, Yogi A Patel, Kirill Abramov, Shay Ohayon, and Jakob Voigts. Open ephys: an open-source, plugin-based platform for multichannel electrophysiology. *Journal of Neural Engineering*, 14(4):045003, 2017.

[24] Valérie Ventura and Richard C. Gerkin. Accurately estimating neuronal correlation requires a new spike-sorting paradigm. *Proceedings of the National Academy of Sciences*, 109(19):7230–7235, 2012.

[25] M. Wehr, J.S. Pezaris, and M. Sahani. Simultaneous paired intracellular and tetrode recordings for evaluating the performance of spike sorting algorithms. *Neurocomputing*, 26-27:1061–1068, 1999.

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

[27] Pierre Yger, Giulia L. B Spampinato, Elric Esposito, Baptiste Lefebvre, Stéphane Deny, Christophe Gardella, Marcel Stimberg, Florian Jetter, Guenther Zeck, Serge Picaud, Jens Duebel, and Olivier Marre. A spike sorting toolbox for up to thousands of electrodes validated with ground truth recordings in vitro and in vivo. *eLife*, 7:e34518, 2018.

[28] Zhongheng Zhang. Introduction to machine learning: K-nearest neighbors. *Annals of translational medicine*, 4(11):218, 2016.