

Spike sorting: new trends and challenges of the era of high-density probes

Alessio P Buccino, Samuel Garcia, Pierre Yger

▶ **To cite this version:**

Alessio P Buccino, Samuel Garcia, Pierre Yger. Spike sorting: new trends and challenges of the era of high-density probes. Progress in Biomedical Engineering, 2022, 4 (2), pp.022005. 10.1088/2516-1091/ac6b96 . inserm-03890947

HAL Id: inserm-03890947

https://inserm.hal.science/inserm-03890947

Submitted on 8 Dec 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

TOPICAL REVIEW • OPEN ACCESS

Spike sorting: new trends and challenges of the era of high-density probes

To cite this article: Alessio P Buccino *et al* 2022 *Prog. Biomed. Eng.* **4** 022005

View the article online for updates and enhancements.

You may also like

- Comparison of spike sorting and thresholding of voltage waveforms for intracortical brain–machine interface performance
Breanne P Christie, Derek M Tat, Zachary T Irwin et al.
- A data-driven spike sorting feature map for resolving spike overlap in the feature space
J Wouters, F Kloosterman and A Bertrand
- Noise-robust unsupervised spike sorting based on discriminative subspace learning with outlier handling
Mohammad Reza Keshtkaran and Zhi Yang

This content was downloaded from IP address 90.127.40.184 on 12/12/2022 at 10:55

IOP Publishing Prog. Biomed. Eng. 4 (2022) 022005 https://doi.org/10.1088/2516-1091/ac6b96

Progress in Biomedical Engineering

CrossMark

OPEN ACCESS

RECEIVED
7 January 2022

REVISED
29 March 2022

ACCEPTED FOR PUBLICATION
28 April 2022

PUBLISHED
17 May 2022

Original Content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence.

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

TOPICAL REVIEW

Spike sorting: new trends and challenges of the era of high-density probes

Alessio P Buccino[1,*], **Samuel Garcia**[2] and **Pierre Yger**[3]

[1] Department of Biosystems Science and Engineering, ETH Zurich, Switzerland
[2] Centre de Recherche en Neuroscience de Lyon, CNRS, Lyon, France
[3] Sorbonne Université, INSERM, CNRS, Institut de la Vision, F-75012 Paris, France
[*] Author to whom any correspondence should be addressed.

E-mail: alessio.buccino@bsse.ethz.ch

Keywords: spike sorting, extracellular electrophysiology, high-density neural probes

Abstract

Recording from a large neuronal population of neurons is a crucial challenge to unravel how information is processed by the brain. In this review, we highlight the recent advances made in the field of 'spike sorting', which is arguably a very essential processing step to extract neuronal activity from extracellular recordings. More specifically, we target the challenges faced by newly manufactured high-density multi-electrode array devices (HD-MEA), e.g. Neuropixels probes. Among them, we cover in depth the prominent problem of drifts (movements of the neurons with respect to the recording devices) and the current solutions to circumscribe it. In addition, we also review recent contributions making use of deep learning approaches for spike sorting, highlighting their advantages and disadvantages. Next, we highlight efforts and advances in unifying, validating, and benchmarking spike sorting tools. Finally, we discuss the spike sorting field in terms of its open and unsolved challenges, specifically regarding scalability and reproducibility. We conclude by providing our personal view on the future of spike sorting, calling for a community-based development and validation of spike sorting algorithms and fully automated, cloud-based spike sorting solutions for the neuroscience community.

1. Introduction

Recording simultaneously from a large population of neurons is a crucial challenge to unravel how information is processed by the brain. This is especially striking if one assumes that fine neuronal correlations, ubiquitous in the brain, are a crucial key to such an understanding [4, 23]. Nowadays, only calcium imaging and extracellular electrophysiology can perform simultaneous large-scale recordings from hundreds of neurons and, at a first glimpse, it may seem that extracellular electrophysiology suffers from fewer intrinsic limitations than calcium imaging [92]. First, genetically-encoded calcium indicators (GECIs) are only an indirect sensor of neuronal spiking activity and have a slow kinetic leading to a poor temporal resolution compared to electrophysiological recordings. Second, calcium imaging requires indicators to be genetically expressed in the target cells, which is not the case with extracellular recordings. Finally, the imaged neurons need to be visible under a microscope (by removing the skull), which makes imaging more invasive than implanting an electrode. Therefore, extracellular signals seem to be much more flexible candidates to extract relevant information from a large ensemble of neurons.

The above-mentioned advantages are even more relevant when one considers the emergence of newly manufactured high-density multi-electrode array (HD-MEA) devices, both for *in vitro* [7, 25] and *in vivo* applications, e.g. Neuropixels probes [40, 83]. These devices introduced a tremendous increase in the number of extracellular channels that can be recorded simultaneously (from tens to thousands) and their spatial density, resulting in the capability to record from hundreds, or even thousands of neurons at once. Such HD-MEA probes can record numerous cells either across layers [83], with flexible designs across different structures [17, 32], or by optimizing electrode switching to increase the yield [45].

© 2022 The Author(s). Published by IOP Publishing Ltd

The procedure of 'spike sorting' consists of various processing steps to extract single-neuron spiking activity from extracellular recordings. Mathematically, it belongs to the category of blind source separation problems, but despite the numerous possibilities given by these new HD extracellular recordings, the analysis of such data arguably creates new challenges [84]. Indeed, with the increased density of the recording channels, the probability of isolating neurons *via spike sorting* is also higher, and so is, as a direct consequence, the probability of having to deal with overlapping spikes (either in space or in time). These spike collisions are hard to resolve and complicate the spike sorting process [26]. In addition, the very dense spatial resolution of HD probes makes some known issues of spike sorting even more evident, especially regarding the prominent problem of drifts (movements of the tissue with respect to the recording devices). Finally, the amount of data generated by HD devices is now becoming so large that the need for fully automated and reproducible spike sorting solutions is pressing.

In this review, we will discuss these open issues in the field of spike sorting. On the algorithmic side, we will focus on describing the generic implementation principles that have been developed to handle drifts and reviewing recent contributions making use of deep learning approaches for spike sorting. Then we will discuss recent advances in benchmarking, validation, and comparison of available spike sorters. In particular, we will describe the `SpikeInterface` [12] and `SpikeForest` [50] projects, as common efforts to provide a unified framework for spike sorting and a quantitative comparison among a large number of well-established spike sorters on a large body of ground-truth datasets. Finally, we will discuss the open challenges of the spike sorting field and provide an overview on how, in our opinion, the future of spike sorting will look like.

2. The spike sorting pipeline

Spike sorting is a complex but necessary procedure to turn extracellular signals into spike trains. Over the years, several algorithms have been developed (see [15, 33, 46] for recent reviews), but despite mild subtleties and variations, they can all be described given the schematic shown in figure 1. In a nutshell, the generic workflow of spike sorting can thus be described as follows. First, the raw extracellular signals (panel 1) are preprocessed to include only the relevant information for spike sorting. Usually, preprocessing involves filtering via a band-pass filter to exclude the low frequency local field potential (LFP) contribution. In addition, a denoising step, such as whitening or common median reference, can also be used to reduce correlated noise among channels (panel 2). Once signals have been preprocessed, putative spike events are detected via threshold crossings (panel 3). Snippets of these putative spikes are then aligned in time (panel 4) and gathered so that they can all be projected into a lower dimensional space. This dimensionality reduction (or feature extraction) step is often achieved using principal component analysis (PCA), but other methods can be considered (e.g. independent component analysis or wavelet transform). In such a reduced space (panel 5), a clustering step splits the spike events into separate groups (see panel 6 and [90] for a review on all clustering algorithms used in the field). Every cluster centroid is a so-called 'template' (panel 7), i.e. the average spatio-temporal motif visible on the extracellular channels when a given neuron is firing an action potential. Once the clustering step is completed, either the spike sorting algorithm stops (and the units are thus the labels of the clusters), or an extra template-matching step is added (see [46] for a review) to reconstruct the signals as a linear sum of the discovered templates and obtain a more accurate estimation of the spike trains (panel 8). Note that this reconstruction creates a more accurate estimation of the spike trains both by resolving colliding spikes and improving detection of low signal-to-noise ratio (SNR) spikes.

While template-matching is probably one of the most significant recent addition to the classical pipeline of spike sorting, since it enhances considerably the performance of the reconstruction [26, 62], this is not the only attempted modification. Indeed, several methods have been recently introduced to enhance the classical view of spike sorting. A full review of these methods is out of the scope of this article, but among those, one could cite works trying to improve the selection of the features (panel 5 of figure 1) [16, 81], using Independent Component Analysis [11, 47] to identify the templates, tracking down the temporal evolution of the templates over time with [79, 86] with statistical assumptions, and consolidating the clustering step by consensus-based methods [12, 22].

3. Recent advances in spike sorting algorithms

3.1. Drift correction

One of the major disadvantages of extracellular recordings over calcium imaging is that the tissue in the vicinity of the electrodes remains *invisible*. Thus the number, the types, and more importantly the positions of the neurons in the surrounding of the neural probe is unknown. The latest is currently the most challenging problem in the field of spike sorting, since one wants to guarantee the identities of the cells over times. While recording, as shown in figure 2(A), it is possible for cells to drift over time and change their

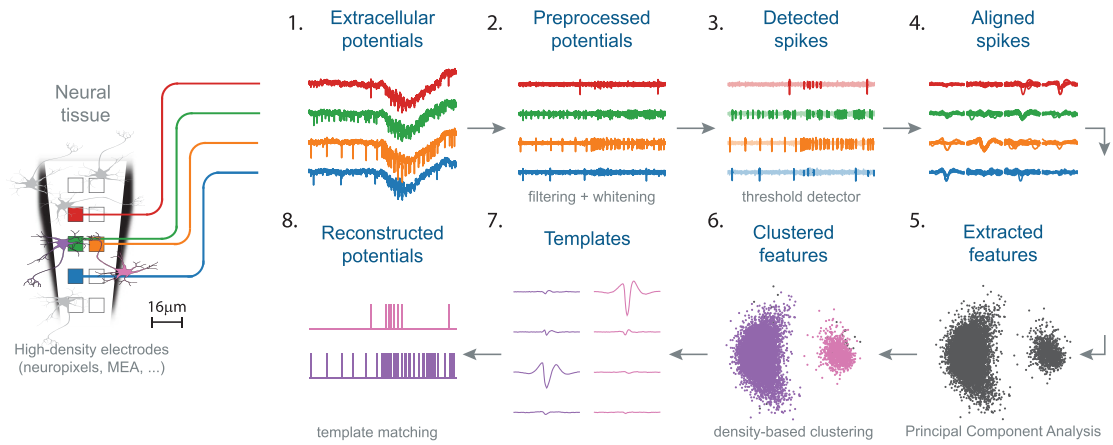


Figure 1. Overview of the generic spike sorting workflow. The spike sorting pipeline is composed of multiple processing steps. Raw extracellular potentials (panel 1) are preprocessed via band-pass filtering (panel 2) before detecting putative spikes (panel 3) as thresholds crossings. Snippets around these spike times are then aligned (panel 4) and projected into a lower dimensional space (panel 5) in order to be clustered as groups (panel 6). Each group is a putative unit with a given template (panel 7), and spikes can be obtained either from the clustering or via an additional template-matching procedure (panel 8).

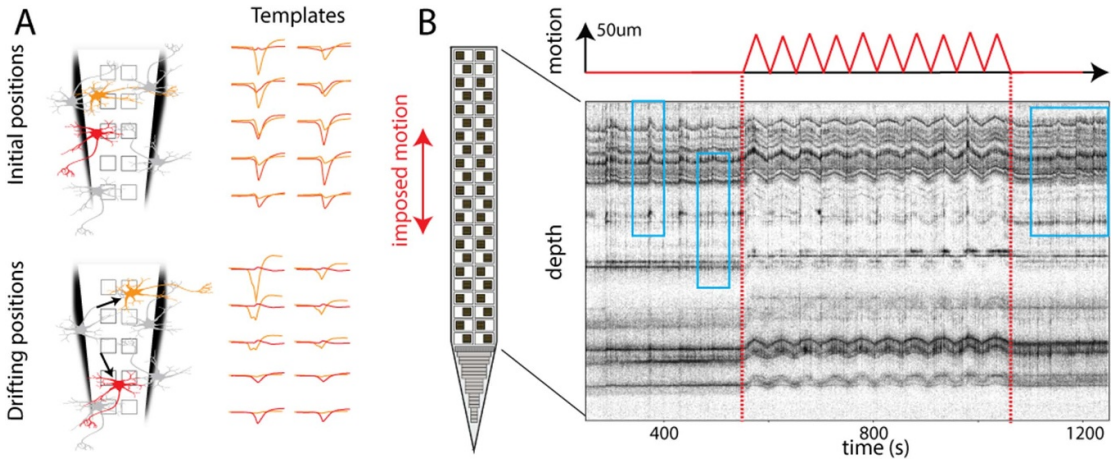


Figure 2. Neuronal drifts. (A) Illustration of the problems raised by neuronal drifts. Two neurons (red and orange) have two distinct waveforms, but after some non rigid drifts (movements), the templates are changed. (B) Illustration of the drift *in vivo*. A Neuropixels 1.0 probe in the cortex of the mice is moved up and down with an imposed triangular movement of amplitude 50 μm (top trace) [82, 83]—`p1` dataset. The row shows the depth of all the detected peaks (*positional* raster plot, see text) as function of the time. As one can see, when the probe starts moving (red dotted line), so does the depth of the peaks, since neurons are drifting across channels. While the drift here is imposed, spontaneous drifts can happen locally and non homogeneously over the whole probe (see inside light blue boxes before and after movements).

positions with respect to the recording probe. These drifts can introduce distortions of the waveforms recorded extracellularly. In order to ensure that units are recognized correctly over the course of a recording, one needs to track down these waveform changes over time. One way to visualize drifts is to represent the distribution of the spiking activity (e.g. the firing rates) as function of the electrode position over time. In figure 2(B), for example, one can see the activity of the neurons as function of time, recorded by a Neuropixels probe *in vivo* [83]. More precisely, the panel displays a *positional* raster plot, where the peak positions are estimated just after spike detection, before any spike sorting step. Usually, the position of the peaks is estimated as its center of mass, but more refined methods can be applied [9, 13, 74]. During the recording, the probe is moved along the vertical axis with a triangular movement of 50 μm amplitude. The oscillations visible in the raster plot here reflect the fact that neurons are moving up and down with respect to the recording channels.

The causes and the temporal extent of these drifts can be numerous. The main source is mechanical: *in vivo*, cells are likely to slowly drift from initial positions because of the pressure release in the tissue after the insertion of a probe. These drifts can have arbitrary time constants, and thus either be abrupt and discontinuous or slow and continuous. Another source of waveform distortion, which is not related to relative displacement, can be functional: if a pharmacological drug is added to the media, or if neurons are

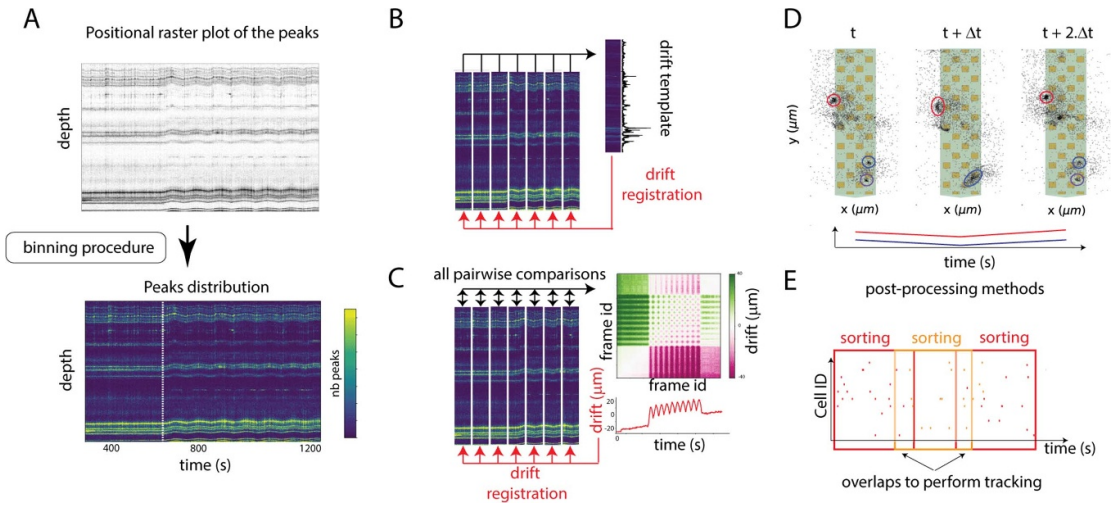


Figure 3. Strategies for drift correction. (A) Peaks position as function of the depth, estimated via the center of mass method, for the recording shown in figure 2. A binning procedure can be used to create an image of the peaks distribution, over space, as function of time. (B) Registration of the extracellular recordings via a common activity template, computed as the average activity profile over all chunks [83]. (C) Same as (B), but with a template computed in a decentralized manner [89]. (D) Illustration of the motion clouds method, trying to refine the estimation of the peaks' position before clustering. Spikes (black dots) during three different consecutive intervals of time δt are shown, superimposed on the probe layout. Red and blue circles show spikes form two different cells, drifting along the depth axis over time. Tracking these movements (bottom lines) can be used to resolve drifts. (E) Illustration of the post-processing methods. Several independent spike sorting procedures are launched, with overlapping time windows. Overlapping results can be used to track drifts over units.

forced to fire in a non-physiological manner (for example with optogenetics if sustained and/or too intense light stimulation [58, 71]). Such external perturbations are likely to distort the templates, in a possibly non-reversible manner, and thus are another real challenge for spike sorting.

There is yet no consensus on how to deal with drifts in extracellular recordings. However, before reviewing the different methods used by modern algorithms, it is worthwhile noticing that solutions can be grouped into two main categories: drifts can either be handled during the pre- or post-processing stages. Both approaches have advantages and disadvantages. On one hand, handling drifts at the pre-processing stage could ease the task of the spike sorting algorithms, at the cost of (possibly) distorting the input data (for example by introducing unwanted artifacts in the signals). On the other hand, handling drifts at the post-processing stage does not modify the input data, but it might result in erroneous results depending on the extent and nature of the drifts.

The basic idea behind correcting for drifts at the pre-processing stage is to mimic drift-correction procedures that have been applied in the field of calcium imaging [28], and thus perform a non-rigid spatial registration of the extracellular signals directly (*pre-registration*). While such approach, to date, has only been applied to Neuropixels devices to estimate the drifts along the depth dimension of the electrodes [83], in theory it could be generalized to drifts in several dimensions [9]. The algorithm starts from the *positional* raster plot, where each detected event is plotted against its estimated position; next, a binning procedure allows one to create an image of the drifts as function of time, i.e. a picture of the activity as function of the depth (see figure 3(A)). Each column of this image is termed activity histogram. Assuming the activity of neurons in the recording is stationary enough, the pre-registration method tries to find the (non) rigid transformations over time that would maintain the histogram of the peaks, for every chunk of duration d, as constant as possible.

The fine details of the registration might vary from algorithm to algorithm. For example, in [83], an average template is obtained as the mean of activity histograms computed by chunk (see figure 3(B)) and used as a reference for motion correction. However, this *average template* approach assumes that the spiking activity is relatively constant over the time course of the recording. To relax this assumption, a recent work has proposed a decentralized method [89] that computes all the pairwise displacements between chunks to directly estimate a motion signal to register the traces (see figure 3(C)). These pre-registration methods can be either rigid (motion is constant over depth) or non rigid (motion depends on the depth). The extension to the non-rigid case can be easily performed by considering only local spatial neighbourhoods along the depth axis. However, note that irregardless of the strategy, the registration procedure, acting at the pre-processing stage, is distorting the input data and, in doing so, might introduce unwanted artifacts that might affect, to some extent, the overall spike sorting performance. Nevertheless, pre-registration of the data

seems to be more robust than what was done in previous versions of KiloSort (v2.0), where chunks of data were grouped by their similarity and templates were able to to evolve through the detection process as a result of an optimization procedure.

A different approach to handle drifts is to try to perform spike localization and motion correction in order to ease the job of downstream clustering (figure 3(D)). An accurate estimate of spike position with respect to the probe layout can be used as an additional feature for clustering and reduce spurious noisy fluctuations in the feature space [9, 34, 39]. While the simplest estimation of spike positions can be done via the centers of mass of the detected spike waveforms, more refined methods using variational inference [38] or point-source models [9] have been proposed and can also provide and estimate of the spike depth (3D localization). Using estimated positions as a spike feature for the clustering step could enhance the the recognition and tracking of different clusters over time [9]. In addition, the estimated spike positions can be used as an alternative method for image-based registration approaches (figures 2(B)–(C)), using a point-cloud registration to estimate motion signals and correct for drifts.

Finally, the last option is to handle drifts at the post-processing stage (figure 2(E)). To do so, the approach followed by [18, 57] is to divide the recording into small and overlapping temporal intervals and to perform the sorting of each of these chunks individually. Using the information in the overlapping regions, one could then try to establish the juncture and track the cell identities over time despite small changes in waveforms. For example, if neuron A is firing during $[t_0, t_1]$ and neuron B is firing during $[t_2, t_3]$ with $t_1 \geqslant t_2$, then if spikes of A and B are the same in $[t_2, t_1]$, the two cells are likely to be the same. The main disadvantage of such an approach is that one needs to launch several instances of the spike sorting procedure, at the cost of slowing down the whole processing pipeline. In addition, the duration of the overlapping window is ad hoc, and should reflect a prior expectation of the temporal evolution of the drifts.

While most of the methods for drift correction were mainly developed for and applied to data from rodents, a recent study featuring Neuropixels recordings in human subjects [60] showed that, at this spatial scale, the drift problem is more prominent and it is strongly modulated by cardiac and breathing cycles. The relatively fast dynamics of these oscillations might make the methods described above, which rely on spike detection and binning to correct for drift, to fall short, mainly because there would not be enough spikes to robustly estimate drift in such short periods (e.g. 1 s). In order to correct for drifts, Paulk and colleagues [60] turned instead to LFP signals and manually drew and aligned signals based on vertical shifts of the LFP bands. LFPs, therefore, could potentially be the data source for automated drift correction methods that perform well across species and scales.

3.2. Deep learning approaches

The last few years have witnessed a boom in the efforts of using deep learning approaches to tackle the spike sorting problem [20, 44, 48, 64, 66, 76, 87, 95, 97, 98]. Deep learning methods, in fact, have proven so powerful and accurate in many complicated applications, ranging from image classification to natural language processing. It is therefore natural to turn to deep learning for such a complex task as spike sorting.

Deep learning (DL) refers to machine learning techniques using very *deep* artificial neural networks (ANN), with many more layers, parameters and artificial neurons. In addition to the higher complexity with respect to standard feed-forward ANN models, DL can also employ a wide variety of architectures to tackle different problems. *Convolutional neural networks* (CNNs), for example, make use of convolutional layers that combine the input features using convolutions in various dimensions (1D, 2D, and even 3D). They are commonly used in image classification tasks, where 2D convolutional layers allow the model to capture 2-dimensional features of the data [41]. Another class of DL models that have been applied to spike sorting are *long short-term memory* (LSTM) networks, which are a type of recurrent neural networks well suited to process time series data, as their recurrent structure enables them to keep a *memory* of the previous times [36]. A third DL concept that is worth introducing is *autoencoders* [5]. Autoencoders (AE) are a family of unsupervised models that aim at encoding the input data into a low-dimensional latent space. This is usually achieved by an *encoder* network, that projects the input data onto the latent space, and a *decoder* network that goes from the latent space to the output.

In the following, we review the recent contributions that used machine/deep learning strategies to tackle the spike sorting process. We identified four main strategies in which DL solutions have been employed for spike sorting (figure 4): (i) end-to-end solutions, (ii) DL approaches for different spike sorting steps, (iii) SNN-based solutions, and (iv) DL for automatic spike sorting curation.

3.2.1. End-to-end solutions

The first strategy to employ DL models for spike sorting is to build a so-called *end-to-end* solution. End-to-end solutions ideally take the raw (or filtered) traces as input to the DL model and output directly the spike trains of the different spike sorted neurons (figure 4(A)). A few different solutions have been proposed

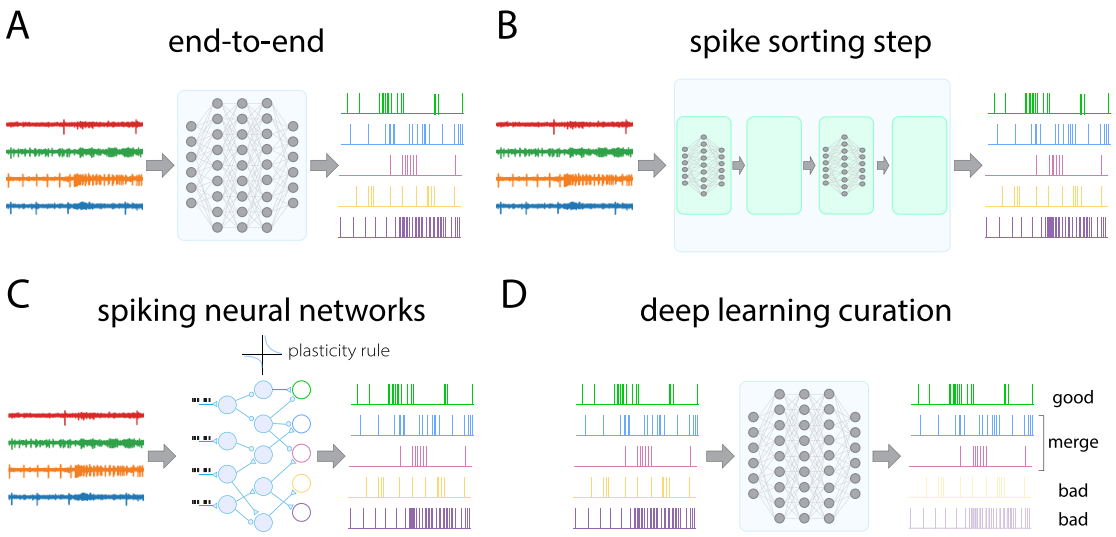


Figure 4. Deep learning strategies for spike sorting. (A) End-to-end solutions use DL network to predict spike times and labels from the signal traces. (B) Alternatively, different DL approaches can be used to for separate steps, such as denoising, spike detection, and feature extraction. (C) Unsupervised spiking neural networks (SNN) use spiking neuronal models and plasticity rules to learn to recognize different units in the recordings. (D) DL can also be used to automatically curate and clean the raw output of an automated spike sorter.

to this end. In the work from Racz *et al* [64], the authors implemented separate networks for spike detection and spike sorting for a 128-channel high-density device [21] using a combination of CNN and LSTM. Li *et al* [48] developed a similar approach focused on recordings from single channels. However, instead of splitting the detection and sorting phase, they used a single 1D CNN architecture to directly detect and sort spikes in the input recording. Racz *et al* [66] proposed a different architectures that uses a CNN+LSTM-based autoencoder architecture to find a latent space and dense output networks to perform the spike sorting task.

While end-to-end solutions arguably represent a *tempting* strategy to tackle the spike sorting problem, there are a few apparent limitations that need to be highlighted. Supervised learning methods, and in particular DL models with many parameters, require a very large amount of *labeled* or *ground-truth* data to be trained. In the three solution described above, the authors mainly used manually curated or synthetic recordings to train their models, and each model was trained separately on each recording. This allowed them to know *a priori* the number of output neurons to be used, for example, but this information is normally not known in advance when spike sorting, which is an *unsupervised* problem in nature. In addition, it is still unclear how end-to-end strategies could generalize over different recordings from different brain regions and states, as well as from different probe geometries. For example, in the cerebellum, Purkinje cells produce both simple and complex spikes, which have very different spike shapes. Complex spikes represent a well-known problem for spike sorting, so that specific cerebellum-specific spike sorters have been developed [77]. To tackle the same problem, Markanday and colleagues [51] implemented an end-to-end solution to detect complex spikes data manually labeled by experts.

3.2.2. Deep learning for spike sorting steps

A second strategy to use DL models for spike sorting is to utilize specific networks for different spike sorting steps, rather than as an end-to-end solution (figure 4(B)). As introduced in section 2, spike sorting methods can be split into several and relatively isolated steps, some of which may be tackled and improved with the aid of a DL approach.

3.2.2.1. Recording denoising

Extracellular recordings contain several sources of independent noise components, that may arise from the readout electronic circuit and also from the biological sample itself (e.g. the activity of far away neurons). Before further processing, it could be therefore useful to apply some denoising techniques, e.g. common average/median reference or whitening of the signals. Lecoq and colleagues [43] approached the problem of denoising by using an autoencoder approach to *interpolate* the input signal (*DeepInterpolation*). The idea is to train a deep network to predict the signal at time t using samples before and after t. This allowed the authors to remove independent noise sources in the input signals (the same approach was also applied to calcium imaging and fMRI data). When applied as a pre-processing step to extracellular recordings from a

Neuropixels probe, the spike sorting output (using Kilosort2 [59]), yielded a larger number of high quality units.

3.2.2.2. Spike detection and waveform denoising

A second processing step that is essential to spike sorting is spike detection. Normally, this step is performed channel-wise by simply detecting intervals of the signals that cross a certain threshold that is set based on some dispersion measures (e.g median absolute deviation [65]). One of the major problems of spike detection is the presence of colliding spiking events, that result in distorted waveforms. These collision events can strongly degrade downstream processing steps, especially clustering, and therefore need to be removed. In order to ameliorate the performance of spike detection, Lee and colleagues, in developing the `YASS` spike sorter [44], used a DL strategy for spike detection and waveform cleaning/denoising. First, the authors implemented a temporal and spatial CNN architecture to detect spikes in the signals, similarly to [64]. The detection network was trained using synthetic snippets on channel neighborhoods that either included a spike event (manually curated templates with additive noise) or did not (noise snippets). After a spike event was detected, a second denoising network was used to clean collision-corrupted spikes. In this case, the model was trained using simulated spike events corrupted by a collision to output the clean template. The combination of the spike detection and waveform denoising network showed to greatly improve the separation of different units in the clustering space. Note that this waveform denoising method was recently also applied to Neuropixels data to clean waveforms before the spike localization [9]. Saif *et al* [75] implemented a 1D-CNN model for single-channel data to remove noisy spike events before clustering. In this case, the model was trained on a large dataset of manually labeled data, in which experts flagged single spikes as being of neuronal origin or noise events.

3.2.2.3. Feature extraction

After spikes are detected, the aligned waveforms need to be projected onto a low-dimensional space before the clustering step; this *feature extraction* step is normally performed by means of PCA or wavelet decomposition. As briefly mentioned before, autoencoders have emerged as optimal candidates to find low-dimensional latent spaces and have therefore been proposed to reduce the dimensionality of waveforms. Wu and colleagues [97], for example, used deep compressive autoencoders to extract features from detected spikes and showed that they outperform several other dimensionality reduction techniques in compressing waveforms. Similarly, Eom *et al* [20] proposed an ensemble of autoencoders with different complexities to project waveforms onto a low-dimensional space and showed that their method provides a better separability of different units in the latent space in comparison to other dimensionality reduction strategies.

In addition to the use of autoencoders to extract features, Wouters *et al* [95] introduced a DL strategy to tackle the problem of spike collisions in spike sorting [26]. When two spikes are perfectly overlapping in time, the observed collision event will be a linear sum of the waveforms from the two neurons; if the feature extraction method is linear (e.g. PCA), the collision spikes from two neurons will form a third cluster with a centroid in correspondence to the sum of the two neurons' clusters. However, when spikes from two neurons have a small jitter in time, the linearity in the feature space is lost. The authors introduced a deep neural network for dimensionality reduction and trained it with synthetic data to enforce linearity in the feature space with respect to spike collisions even when a jitter between the colliding spikes is present. The promising results showed that the proposed feature map is capable of achieving a performance close to optimal template matching without a template-matching phase.

3.2.3. Spiking neural networks (SNN)-based solutions

Deep neural networks utilize a very simplistic model of artificial neurons, i.e. the *perceptrons* [68], which integrate their weighted input to produce an output. The output of an artificial neuron can be thought as a mean firing rate, as there is no concept of *spikes* in these kind of neural networks. Conversely, in spiking neural networks (SNN) the artificial neurons communicate via spikes and synaptic inputs. Leaky integrate-and-fire (LIF) models of single neurons temporally integrate the synaptic inputs received from other neurons and they produce a spike whenever their membrane potential crosses a threshold. By employing models of synaptic plasticity, SNNs have shown to be able to perform several unsupervised learning tasks [19, 30].

In the context of spike sorting, Werner *et al* [93] introduced a 2-layer SNN implementation capable of spike sorting extracellular data from a single channel in real-time. The proposed method first decomposed the input signal using bank filters whose output modulated the firing of the corresponding input neurons. The input neurons were fully connected to five output neurons (the number of output neurons needs to be greater than the number of expected units), which connected to each other with inhibitory synapses to

implement a winner-take-all approach (only one output neuron can fire at a time). Using spike-timing dependent plasticity (STDP), the SNN was able to learn to recognize spikes from different neurons in the recordings. Importantly, the spike trains generated by the output layer correspond directly to the sorted spike trains (figure 4(C)). In a follow-up study, Bernert and colleagues [8] used a different SNN-based solution to spike sort data both from single channel recordings and from tetrode data. The authors introduced an additional hidden layer that uses plasticity to learn different waveform fragments in the input signal. The output neurons learn to recognize combinations of different fragments as different units and to output the spike trains. Additionally, an attention neuron is employed to gate the learning only when an action potential is occurring.

SNN-based solutions represent an emerging strategy for spike sorting, especially given their intrinsic capability of functioning in real-time and of being implemented in neuromorphic hardware solutions [93]. This characteristics makes them optimal candidates for brain-machine interface applications, where the raw signals need to be processed and decoded in real-time. However, in the context of precise spike sorting for systems neuroscience, some limitations exist. Unsupervised learning rules for SNN (such as spike timing dependent plasticity) are very sensitive to initial conditions, and quite often unstable without metaplasticity (see [99] for review). Thus, very precise fine tuning of the parameters is often necessary. Moreover, questions such as how the extracellular traces should be turned into spikes, or what is the optimal size of the SNN required to capture all the spiking dynamics, still remain open.

3.2.4. Deep learning for spike sorting curation

The last strategy to utilize DL in the spike sorting process is to build models that automatically curate a spike sorting output. Differently from all the strategies described above, in this case the DL model will take a raw output from a spike sorter as input and output a curated version of the spike sorted units (figure 4(D)). Manual curation of spike sorting output is still a very common and required step to improve the results of automatic spike sorters before further analysis. Spike sorters, in fact, do make mistakes [12, 50] and an expert electrophysiologist can correct such mistakes by inspecting features of the spike sorted units. Several software tools (`Phy` [70], `Neuroscope` [31], `CellExplorer` [61], `JRClust` [39]) are used by the community to remove bad or noisy units, merge oversplit units, and split unit clusters (when overmerged or when contaminated by noise events). The former two curation actions (removing bad units and merging) could be well suited for a supervised DL approach, since they do not modify the underlying spike trains. However, this solution is still relatively unexplored. One option could be to gather all the annotated curation steps and train a network using these community-labeled data. However, since it is well known that manual curation is prone to bias [70, 94], a second or complementary option could be to use ground-truth datasets to train a network to perform the best curation steps given the ground-truth spiking activity.

4. Benchmarking spike sorting results

Spike sorting tools require extensive and quantitative validation to assess their performance. To provide such validation, one needs to utilize so-called ground-truth (GT) data, i.e. recordings in which the spiking activity of the underlying neurons is known. There exist three different strategies to obtain ground-truth recordings, which differ in their biophysical detail and amount of units available as ground truth (figure 5(A)).

4.1. Paired recordings

These GT data are obtained by recording the same neuron simultaneously with an extracellular probe and a second measuring modality, such as a glass micro-pipette inserted in the target neuron (intracellular) [91], attached to the cell membrane (juxtacellular) [56], or via whole-cell patch-clamp [2, 52, 100] (figure 5(1)). The second modality enables to record spikes of the target neuron unequivocally and reliably, so that they can be used as ground-truth. While paired recordings are the most realistic source of ground-truth data, the acquisition of such paired dataset is arguably challenging, especially *in vivo* [2, 52, 56]. A second limitation of this approach is its very low-yield: in most cases only a single ground-truth neuron is available for each dataset and this prevents a full characterization of the spike sorting performance. In addition, recording via a micro-pipette could be inherently biased in terms of the cell-type of the target neuron, as interneurons are less numerous and smaller than pyramidal cells, and therefore less likely to be targeted.

4.2. Hybrid recordings

A second type of GT data is constructed by adding artificial spiking activity to experimental recordings [70, 96] (figure 5(2)). The templates of the artificial neurons are usually extracted from previously sorted and curated data and are re-injected into the recording by adding modulations to resemble physiological

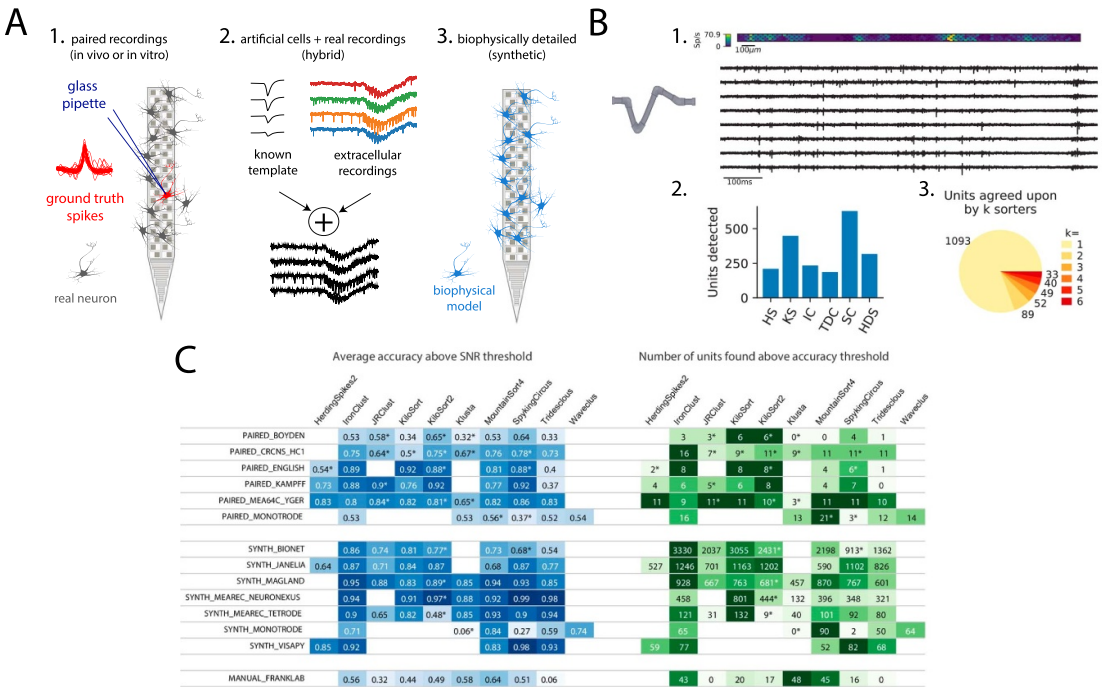


Figure 5. Spike sorting validation. (A) Different strategies for GT data: (1) paired extracellular and intra-/juxtacellular recordings; (2) hybrid recordings with injected artificial neurons; (3) fully synthetic recordings with biophysical models. (B) The `SpikeInterface` project showed how largely spike sorters disagree. (1) Snippet of a Neuropixels trace with its *activity map* on the top; (2) number of units found by six different sorters (HS—Herding Spikes, KS—Kilosort2, IC—IronClust, TDC—Tridesclous, SC—Spyking Circus, HDS—HDSort); (3) units in agreement with k different sorters. Reproduced from [12]. CC BY 4.0. (C) The `SpikeForest` project systematically benchmarks spike sorters against a large body of ground-truth data. Example of the result tables, with the accuracy of spike sorters on the left (blue) and the number of units found by each sorter on the right (green). Reproduced from [50]. CC BY 4.0.

variability, for example due to bursting [2]. While hybrid recordings provide a higher yield with respect to paired recordings, they still do not allow a full characterization of the spike sorting output. In practice, only a limited number of artificial units can be injected in the original recording in order to avoid an *overcrowded* spiking activity that could compromise the intrinsic performance of spike sorters.

4.3. Synthetic recordings

A third strategy to obtain GT data is to use simulations of extracellular signals and corresponding spiking activity [10, 14, 29, 39, 55] (figure 5(3)). In this case, the GT dataset allows for an exhaustive and full benchmark of spike sorting results, since all neurons making up the spiking activity are known. However, the debate on whether synthetic recordings are realistic enough to capture subtleties of experimental data is still open. Nevertheless, simulations can provide a powerful and highly controllable tool to develop and benchmark spike sorting tools. As an example, controlled simulations were used to specifically assess the performance of spike sorters against spike collisions, which are a well known challenge for accurate spike sorting [26].

With these different ground-truth recordings, one can start approaching the task of systematically benchmarking the performance of multiple spike sorting tools. However, this step is challenging for several reasons: first of all, in extracellular electrophysiology, there is a large variety of proprietary formats and acquisition systems; second, the community has produced several spike sorters, each coming with its own requirements in terms of input format, parameters, configuration files, and output format. `SpikeInterface` is an open-source software tool developed to unify existing technologies into a single framework [12]. Providing a standardized access to the multiple acquisition formats and spike sorters, `SpikeInterface` allows one to run and assess the performance of many sorters on the same recordings. Strikingly, when six modern spike sorters were run on the same Neuropixels dataset [80] (figure 5(B.1) shows a snippet of the traces) there was a large disagreement among them (figure 5(B.2) and (B.3)). Out of a total of ∼1400 units found, only 33 were in agreement between all sorters and 263 in agreement between at least two sorters [12]. Clearly, this highlights the need of more systematic benchmarking efforts in order to *trust* the results of automatic spike sorters.

In this direction, the SpikeForest project [50] attempted to gather a large body of ground-truth data, systematically run spike sorters on them, and report the results in an interactive and intuitive manner. The resulting website (https://spikeforest.flatironinstitute.org/) reports the performance tables (figure 5(C)) of spike sorting results from over 1.3 TB of ground-truth data coming from diverse sources and strategies (paired, hybrid, and synthetic recordings). Such effort provides unprecedented guidance to end-users, willing to explore and understand the performance of different tools on different types of recordings (e.g. tetrodes vs high-density probes) and it arguably represented a push for the spike sorting community, as now developers have a massive and curated ground-truth repository to test their tools, and new features, against. Moreover, the SpikeForest team had to overcome intrinsic software engineering challenges that we cover in more details in the next section.

5. Open challenges and the future of spike sorting

In this review, we covered the recent advances in the field of spike sorting, especially all the recent pre- and post-processing techniques applied by spike sorting algorithms to track the neurons over time when they are drifting with respect to the recording sites. We also gave an overview of all the attempts to use deep learning approaches in order to boost the performance of these algorithms, highlighting different strategies with their advantages and limitations. Finally, we covered the latest progress in generating ground-truth data for systematic benchmarking and validation of spike sorting tools. Despite the tangible advances in spike sorting development and validation efforts over the last few years, the neurotechnology *revolution* that we are experiencing with the advent and broad distribution of high-density probes is setting novel challenges for spike sorting pipelines.

The first new challenge is *scalability*. Since the first extracellular recording from a tungsten microelectrode [37] neural devices for both *in vivo* and *in vitro* applications have experienced a steady increase in the number of electrodes that can be recorded simultaneously (figure 6(A)). For *in vitro* applications, where probe size is less of a constraint than *in vivo*, micro-electrode array technologies have allowed to reach over 1000 simultaneously recorded electrodes [7, 25] packed at very high spatial densities. Recently, some prototypes have been proposed that can even record from over ∼20 000 electrodes simultaneously [101]. Neural devices with several hundreds of electrodes have also been developed for *in vivo* applications [3, 17, 21, 40, 83], with the Neuropixels technology being arguably a game changer in the field of systems neuroscience. Spike sorting algorithms need to be able to cope with this steadily increasing number of channels. A 1 hour Neuropixels 1.0 or 2.0 probe recording [40, 83], with their 384 channels sampled at 30 kHz and with 2 bytes per sample, will produce over ∼80 GB of unprocessed raw data. With a similar calculation, a 1 hour recording of the 20 000-channel device in [101] (sampled around 10 kHz) will give a 1 TB file to analyze. Spike sorters, therefore, need to implement algorithmic solutions with scalability, speed, and performance in mind, for example by leveraging multiprocessing, distributed, and GPU-accelerated solutions. In addition, spike sorting developers need to be aware of the trends in neurotechnology and try to predict what neural devices will look like in the next years to come[4]. Finally, it is also essential that the computational time of spike sorting is faster than 1x (i.e. it takes less for spike sorting than the duration of the recording), otherwise, as a field, we will never finish our spike sorting *pre*-processing and move on with downstream analysis.

A second challenge for spike sorting is *reproducibility*. Despite the more and more common usage of automatic tools for an initial spike sorting, a manual curation step to *clean* the spike sorting output is still very common, if not necessary. A number of tools have been developed to allow users to remove, merge, and split units [31, 39, 61, 69], but this manual curation is still very subjective and can provide different outputs [70, 94]. Clearly this hinders full reproducibility of the results and all downstream analysis that depends on spike sorting. In section 3.2.4 we discussed the possibility to use deep learning methods trained on ground-truth data to automatically curate spike sorting outputs, but this strategy is still relatively unexplored and it requires a large amount of high-quality ground-truth data and validation efforts. A second and probably more viable approach consists of using unsupervised quality metrics computed directly on the spike sorting output [6, 35]. These metrics are designed to assess the *goodness* of a sorted unit both using known biophysical properties (e.g. a neuron should not fire during its refractory period) and data-driven metrics, for example by measuring the isolation of a unit in the feature space. While a wide array of these metrics have been proposed, the community is still lacking a consensus on which metrics are informative and should be used to automatically clean spike sorting outputs. With the ever increasing effort of gathering ground-truth and curated datasets [50], it is of paramount importance for spike sorting developers and end-users to

[4] Strikingly, the Klusta spike sorter [70], developed in 2016 for 'large and dense' probes (up to 64 channels), has become outdated in less than 4 years.

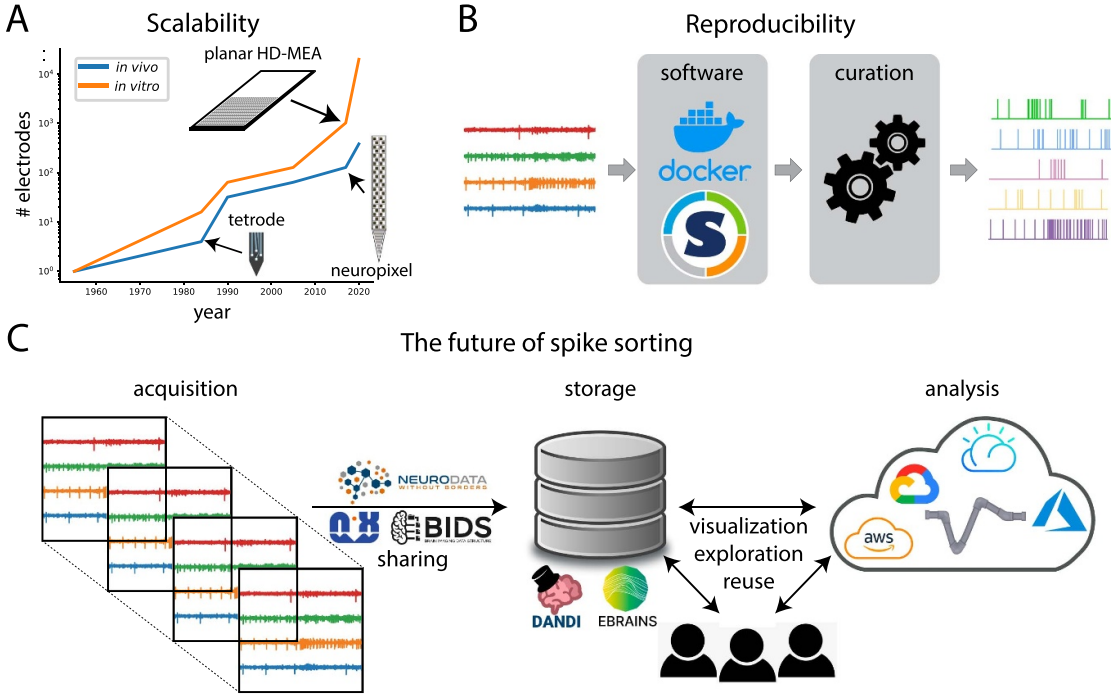


Figure 6. Challenges and future of spike sorting (A) Scalability. The constant increase of electrode counts both *in vivo* (blue) and *in vitro* (orange) represents a challenge for spike sorting tools. (B) The reproducibility challenge can be addressed by using containerized software (via Docker or Singularity) and automatic curation. (C) The future of spike sorting, in our view, is to make storage and processing remotely. After acquisition, recordings are converted to standardized formats, such as NWB, NIX, or BIDS, and uploaded to cloud-based archives, e.g. DANDI or EBRAINS. Analysis pipelines, including spike sorting, can be then deployed on cloud computing services.

collaborate in order to identify and converge on common curation strategies to reduce the burden and bias of manual curation. A third but computationally expensive option to avoid manual curation could be running several spike sorters on the same recording and aggregating the results in an *ensemble* fashion [12]. While this *ensemble* strategy produced curated outputs which largely overlapped with manually curated ones when applied to a neuropixels recording [12], its validation is still limited and more testing is required.

Reproducibility can also be affected by the intrinsic life cycle of spike sorting software tools, which can undergo many modifications and releases as bugs are fixed, methods are improved, and new features are added. While version controlled systems (VCS) can keep track of all these changes, from a user perspective it can be confusing and even complicated to know which exact version of a spike sorter was used on specific datasets and to report correctly. A possible solution to the *version* problem is to encapsulate spike sorters of different versions in containers, i.e. pieces of software that package all codes and dependencies for an application to run independent of the environment. Containers can be run through platforms such as `Docker` [54] and `Singularity` [42] and ensure reproducibility on the software side. The `SpikeForest` project first introduced containerized and version-tagged containers of 11 different spike sorters which we recently collected and distributed through `SpikeInterface`[5]. In addition to improving reproducibility of the results, containers can also ease the process of installing and setting up different spike sorters, which can be difficult as the number and complexity of dependencies increases. The combination of containerized solutions and automatic curation would enable a fully reproducible spike sorting pipeline, which does not require any human intervention (and possibly human bias) in the loop (figure 6(B)).

In addition to the above-mentioned scalability and reproducibility challenges, the main open challenge of spike sorting in general probably remains its accuracy and reliability. Spike sorters are prone to various types of errors, especially for low SNR units [50], and have been shown to largely disagree between each other [12]. Taken together, this inaccuracy requires users to carefully check and correct spike sorting results, e.g. via manual curation. Therefore, spike sorting algorithms still necessitate extensive development to increase their accuracy and reliability, specifically in light of some complex characteristics of extracellular recordings which are known to make spike sorting even harder. Among them, a major problem is how to automatically handle

[5] On GitHub https://github.com/SpikeInterface/spikeinterface-dockerfiles—and DockerHub https://hub.docker.com/orgs/spikeinterface.

waveforms modulation due to bursting, which causes an amplitude and shape distortion of the extracellular waveforms during the firing of consecutive spikes [2]. Several algorithmic attempts have been proposed to tackle the issue [63, 78], but to date there is no definitive solution implemented in the most commonly used spike sorting tools and correction of spike sorting errors due to bursting modulation is still mainly addressed via post-hoc manual curation. Spike collisions (spike events overlapping in space and/or time), which have been proven to degrade the spike sorting performance [26], represent a second well-known issue for spike sorting methods to address. Although recent spike sorting solutions based on template-matching appear to ameliorate the problem [26], automatic spike sorters are still more error-prone when spikes are colliding (especially from neurons with similar templates). Spike sorting becomes particularly challenging during network bursts, which are very common in *in vitro* cell cultures [88], due to the concomitant bursting modulation and high spike collision rates. Different brain samples (e.g. *in vivo*, brain slices, embryonic cell cultures, or induced pluripotent stem cell (IPSC)-derived cultures), brain areas, species, as well as brain states could largely affect spike sorting results, given the large diversity of the underlying neural signals. A one-spike-sorter-fits-all strategy might therefore be sub-optimal (see for example specific solutions developed for spike sorting data from the cerebellum [49, 77]) and the spike sorting step should be flexible enough to accommodate recordings with various subtleties and differences.

Finally, while we have mainly focused on offline applications, online spike sorting for HD probes has the potential to tremendously boost the possibilities for neuroprosthetics or brain-machines interfaces. While several attempts have been made [24, 73], all the aforementioned bottlenecks (drifts, bursts, collisions, scalability) are even more striking in an online context.

In addition to the increased number of electrodes and size of single files, the large availability and distribution of HD probes is producing, worldwide, an unprecedented amount of data. Current solutions utilized by single laboratories or institutions usually rely on local machines or institutional servers, which might fall short in terms of both storage capacity and computational load, given the sharp increase of number of electrodes and overall number of recordings. Additionally, there is still a large variation among groups in the way that spike sorting is carried out, both in terms of the chosen software and custom post-processing and manual curation steps. A possible solution forward is to move both data storage and spike sorting fully remotely, on cloud computing services [1, 67]. A remote and centralized solution for spike sorting could solve several issues that we have introduced so far. As new probes with even more electrodes are developed and more and more recordings are generated by the neuroscience community, cloud-based solutions (e.g. the DANDI archive https://gui.dandiarchive.org/#/ and the EBRAINS platform https://ebrains.eu/) could provide on-demand storage as required and would prevent individual institutions to constantly upgrade their servers and storage capabilities. In addition, when uploading data onto a remote server, standardized file formats could be enforced, e.g. NWB [72, 85], NIX [53], and BIDS [27], and as a by-product several standardized tools for data processing and visualization can be developed. Plus, data sharing and reuse would be drastically facilitated. Leveraging cloud computing providers, we could move the heavy computational burden of spike sorting remotely and use processing resources as needed. In this regard, the containerization of spike sorting tools is already an important step toward the deployment of spike sorting on the cloud. In addition, users could also benefit from efficient and cloud-based visualization tools from standardized APIs (e.g. `SpikeInterface`), which would facilitate data exploration and reuse. Another benefit of moving processing on the cloud is to allow a centralized and collaborative spike sorting development and deployment, instead of the current *individual* state where many different groups are aiming to develop and improve their own tools. Given the modular nature of the spike sorting processing, where several well identified and circumscribed steps are needed to go from extracellular traces to spike trains, it is very likely that different sorters are providing optimal solutions for different steps. Following this idea, the `SpikeInterface` project has started to break down different sorters and re-implement them in a modular fashion. By doing so, it calls upon the spike sorting developer community to collaborate in this effort and collaborate to discuss, define, implement, and test different *sorting components* available from multiple sorters[6]. In such a view, a modular and collaborative spike sorting development phase, where different steps of the pipeline are implemented in a standardized, efficient, accessible, and continuously tested framework by the entire spike sorting developer community, will represent a tangible progress in making spike sorting tools more reliable, efficient, and trustworthy.

Data availability statement

No new data were created or analysed in this study.

[6] https://spikeinterface.readthedocs.io/en/latest/module_sorting_components.html.

Acknowledgments

This work was supported by the ETH Zurich Postdoctoral Fellowship 19-2 FEL-17 (A P B). In addition, we would like to thank Julien Boussard for fruitful conversations on the spike localization and drift correction topic. We are thankful to Nick Steinmetz for acquiring and sharing the Neuropixels dataset with imposed drift movements that we used to generate figures 2 and 3. Finally, we wish to thank the spike sorting community for sharing and discussing exciting updates over the past year: Jeremy Magland, Liam Paninski, Matthias Hennig, Jeff Soules, Fernando Chaure, Hernan Rey, Julien Boussard, Erdem Varol, Cole Hurwitz, Ben Dichter, and Olivier Winter.

ORCID iDs

Alessio P Buccino https://orcid.org/0000-0003-3661-527X
Samuel Garcia https://orcid.org/0000-0001-6389-9779
Pierre Yger https://orcid.org/0000-0003-1376-5240

References

[1] Abe T *et al* 2021 Neuroscience cloud analysis as a service *bioRxiv Preprint* pp 2020–06
[2] Allen B D *et al* 2018 Automated *in vivo* patch-clamp evaluation of extracellular multielectrode array spike recording capability *J. Neurophysiol.* **120** 2182–200
[3] Angotzi G N, Boi F, Lecomte A, Miele E, Malerba M, Zucca S, Casile A and Berdondini L 2019 Sinaps: an implantable active pixel sensor cmos-probe for simultaneous large-scale neural recordings *Biosens. Bioelectron.* **126** 355–64
[4] Averbeck B B, Latham P E and Pouget A 2006 Neural correlations, population coding and computation *Nat. Rev. Neurosci.* **7** 358–66
[5] Baldi P 2012 Autoencoders, unsupervised learning and deep architectures *Proc. ICML Workshop on Unsupervised and Transfer Learning* (*JMLR Workshop and Conf. Proc.*) pp 37–49
[6] Barnett A H, Magland J F and Greengard L F 2016 Validation of neural spike sorting algorithms without ground-truth information *J. Neurosci. Methods* **264** 65–77
[7] Berdondini L, Imfeld K, Maccione A, Tedesco M, Neukom S, Koudelka-Hep M and Martinoia S 2009 Active pixel sensor array for high spatio-temporal resolution electrophysiological recordings from single cell to large scale neuronal networks *Lab Chip* **9** 2644–51
[8] Bernert M and Yvert B 2019 An attention-based spiking neural network for unsupervised spike-sorting *Int. J. Neural Syst.* **29** 1850059
[9] Boussard J, Varol E, Lee H D, Dethe N and Paninski L 2021 Three-dimensional spike localization and improved motion correction for neuropixels recordings *35th Conf. on Neural Information Processing Systems*
[10] Buccino A P and Einevoll G T 2021 Mearec: a fast and customizable testbench simulator for ground-truth extracellular spiking activity *Neuroinformatics* **19** 185–204
[11] Buccino A P, Hagen E, Einevoll G T, Häfliger P D and Cauwenberghs G 2018 Independent component analysis for fully automated multi-electrode array spike sorting *2018 40th Annual Int. Conf. IEEE Engineering in Medicine and Biology Society (EMBC)* (IEEE) pp 2627–30
[12] Buccino A P, Hurwitz C L, Garcia S, Magland J, Siegle J H, Hurwitz R and Hennig M H 2020 Spikeinterface, a unified framework for spike sorting *eLife* **9** e61834
[13] Buccino A P, Kordovan M, Ness T V, Merkt B, Häfliger P D, Fyhn M, Cauwenberghs G, Rotter S and Einevoll G T 2018 Combining biophysical modeling and deep learning for multielectrode array neuron localization and classification *J. Neurophysiol.* **120** 1212–32
[14] Camuñas-Mesa L A and Quiroga R Q 2013 A detailed and fast model of extracellular recordings *Neural Comput.* **25** 1191–212
[15] Carlson D and Carin L 2019 Continuing progress of spike sorting in the era of big data *Curr. Opin. Neurobiol.* **55** 90–96
[16] Chaure F J, Rey H G and Quian Quiroga R 2018 A novel and fully automatic spike-sorting implementation with variable number of features *J. Neurophysiol.* **120** 1859–71
[17] Chung J E *et al* 2019 High-density, long-lasting and multi-region electrophysiological recordings using polymer electrode arrays *Neuron* **101** 21–31
[18] Chung J E, Magland J F, Barnett A H, Tolosa V M, Tooker A C, Lee K Y, Shah K G, Felix S H, Frank L M, Greengard L F 2017 A fully automated approach to spike sorting *Neuron* **95** 1381–94
[19] Diehl P U and Cook M 2015 Unsupervised learning of digit recognition using spike-timing-dependent plasticity *Front. Comput. Neurosci.* **9** 99
[20] Eom J, Park I Y, Kim S, Jang H, Park S, Huh Y and Hwang D 2021 Deep-learned spike representations and sorting via an ensemble of auto-encoders *Neural Netw.* **134** 131–42
[21] Fiáth R, Raducanu B C, Musa S, Andrei A, Lopez C M, van Hoof C, Ruther P, Aarts A, Horváth D and Ulbert I 2018 A silicon-based neural probe with densely-packed low-impedance titanium nitride microelectrodes for ultrahigh-resolution *in vivo* recordings *Biosens. Bioelectron.* **106** 86–92
[22] Fournier J, Mueller C M, Shein-Idelson M, Hemberger M and Laurent G 2016 Consensus-based sorting of neuronal spike waveforms *PLoS One* **11** e0160494
[23] Franke F, Fiscella M, Sevelev M, Roska B, Hierlemann A and da Silveira R A 2016 Structures of neural correlation and how they favor coding *Neuron* **89** 409–22
[24] Franke F, Natora M, Boucsein C, Munk M H and Obermayer K 2010 An online spike detection and spike classification algorithm capable of instantaneous resolution of overlapping spikes *J. Comput. Neurosci.* **29** 127–48
[25] Frey U, Egert U, Heer F, Hafizovic S and Hierlemann A 2009 Microelectronic system for high-resolution mapping of extracellular electric fields applied to brain slices *Biosens. Bioelectron.* **24** 2191–8

[26] Garcia S, Buccino A P and Yger P 2021 How do spike collisions affect spike sorting performance? *bioRxiv Preprint* (https://doi.org/10.1101/2021.11.29.470450)
[27] Gorgolewski K J *et al* 2016 The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments *Sci. Data* **3** 160044
[28] Greenberg D S and Kerr J N 2009 Automated correction of fast motion artifacts for two-photon imaging of awake animals *J. Neurosci. Methods* **176** 1–15
[29] Hagen E, Ness T V, Khosrowshahi A, Sorensen C, Fyhn M, Hafting T, Franke F and Einevoll G T 2015 Visapy: a python tool for biophysics-based generation of virtual spiking activity for evaluation of spike-sorting algorithms *J. Neurosci. Methods* **245** 182–204
[30] Hazan H, Saunders D, Sanghavi D T, Siegelmann H and Kozma R 2018 Unsupervised learning with self-organizing spiking neural networks *2018 Int. Joint Conf. on Neural Networks (IJCNN)* (IEEE) pp 1–6
[31] Hazan L, Zugaro M and Buzsáki G 2006 Klusters, neuroscope, ndmanager: a free software suite for neurophysiological data processing and visualization *J. Neurosci. Methods* **155** 207–16
[32] He F, Lycke R, Ganji M, Xie C and Luan L 2020 Ultraflexible neural electrodes for long-lasting intracortical recording *iScience* **23** 101387
[33] Hennig M H, Hurwitz C and Sorbaro M 2019 Scaling spike detection and sorting for next-generation electrophysiology *In Vitro Neuronal Networks* (*Advances in Neurobiology* vol 22) (Cham: Springer) pp 171–84
[34] Hilgen G *et al* 2017 Unsupervised spike sorting for large-scale, high-density multielectrode arrays *Cell Rep.* **18** 2521–32
[35] Hill D N, Mehta S B and Kleinfeld D 2011 Quality metrics to accompany spike sorting of extracellular signals *J. Neurosci.* **31** 8699–705
[36] Hochreiter S and Schmidhuber J 1997 Long short-term memory *Neural Comput.* **9** 1735–80
[37] Hubel D H and Wiesel T N 1962 Receptive fields, binocular interaction and functional architecture in the cat's visual cortex *J. Physiol.* **160** 106–54
[38] Hurwitz C L, Xu K, Srivastava A, Buccino A P and Hennig M H 2019 Scalable spike source localization in extracellular recordings using amortized variational inference *NeurIPS Proc.* (available at: https://proceedings.neurips.cc/paper/2019/hash/f12f2b34a0c3174269c19e21c07dee68-Abstract.html)
[39] Jun J J, Mitelut C, Lai C, Gratiy S, Anastassiou C and Harris T D 2017 Real-time spike sorting platform for high-density extracellular probes with ground-truth validation and drift correction *bioRxiv Preprint* p 101030
[40] Jun J J *et al* 2017 Fully integrated silicon probes for high-density recording of neural activity *Nature* **551** 232–6
[41] Krizhevsky A, Sutskever I and Hinton G E 2012 Imagenet classification with deep convolutional neural networks *Advances in Neural Information Processing Systems* vol 25 pp 1097–105
[42] Kurtzer G M, Sochat V, Bauer M W and Gursoy A 2017 Singularity: scientific containers for mobility of compute *PLoS One* **12** e0177459
[43] Lecoq J, Oliver M, Siegle J H, Orlova N, Ledochowitsch P and Koch C 2021 Removing independent noise in systems neuroscience data using deepinterpolation *Nat. Methods* **18** 1401–8
[44] Lee J *et al* 2020 Yass: yet another spike sorter applied to large-scale multi-electrode array recordings in primate retina *bioRxiv Preprint* (https://doi.org/10.1101/2020.03.18.997924)
[45] Lee K H, Ni Y-L, Colonell J, Karsh B, Putzeys J, Pachitariu M, Harris T D and Meister M 2021 Electrode pooling can boost the yield of extracellular recordings with switchable silicon probes *Nat. Commun.* **12** 1–14
[46] Lefebvre B, Yger P and Marre O 2016 Recent progress in multi-electrode spike sorting methods *J. Physiol. Paris* **110** 327–35
[47] Leibig C, Wachtler T and Zeck G 2016 Unsupervised neural spike sorting for high-density microelectrode arrays with convolutive independent component analysis *J. Neurosci. Methods* **271** 1–13
[48] Li Z, Wang Y, Zhang N and Li X 2020 An accurate and robust method for spike sorting based on convolutional neural networks *Brain Sci.* **10** 835
[49] Llobet V, Wyngaard A and Barbour B 2022 Automatic post-processing and merging of multiple spike-sorting analyses with lussac *bioRxiv Preprint*
[50] Magland J, Jun J J, Lovero E, Morley A J, Hurwitz C L, Buccino A P, Garcia S and Barnett A H 2020 Spikeforest, reproducible web-facing ground-truth validation of automated neural spike sorters *eLife* **9** e55167
[51] Markanday A, Bellet J, Bellet M E, Inoue J, Hafed Z M and Thier P 2020 Using deep neural networks to detect complex spikes of cerebellar purkinje cells *J. Neurophysiol.* **123** 2217–34
[52] Marques-Smith A, Neto J P, Lopes G, Nogueira J, Calcaterra L, Frazão J, Kim D, Phillips M G, Dimitriadis G and Kampff A R 2018 Recording from the same neuron with high-density cmos probes and patch-clamp: a ground-truth dataset and an experiment in collaboration *bioRxiv Preprint* p 370080
[53] Martone M, Gerkin R, Moucek R, Das S, Goscinski W, Hellgren-Kotaleski J, Kennedy D, Leergaard T, Boline J and Abrams M 2020 Nix–neuroscience information exchange format *F1000Research* **9** 358
[54] Merkel D *et al* 2014 Docker: lightweight linux containers for consistent development and deployment *Linux J.* **2014** 2 (available at: www.linuxjournal.com/content/docker-lightweight-linux-containers-consistent-development-and-deployment)
[55] Mondragón-González S L and Burguière E 2017 Bio-inspired benchmark generator for extracellular multi-unit recordings *Sci. Rep.* **7** 1–13
[56] Neto J P *et al* 2016 Validating silicon polytrodes with paired juxtacellular recordings: method and dataset *J. Neurophysiol.* **116** 892–903
[57] Niediek J, Boström J, Elger C E, Mormann F and Menendez de la Prida L 2016 Reliable analysis of single-unit recordings from the human brain under noisy conditions: tracking neurons over hours *PLoS One* **11** e0166598
[58] Octeau J C, Gangwani M R, Allam S L, Tran D, Huang S, Hoang-Trong T M, Golshani P, Rumbell T H, Kozloski J R and Khakh B S 2019 Transient, consequential increases in extracellular potassium ions accompany channelrhodopsin2 excitation *Cell Rep.* **27** 2249–61
[59] Pachitariu M, Steinmetz N A, Kadir S N, Carandini M and Harris K D 2016 Fast and accurate spike sorting of high-channel count probes with kilosort *Advances in Neural Information Processing Systems* vol 29 pp 4448–56
[60] Paulk A C, Kfir Y, Khanna A R, Mustroph M L, Trautmann E M, Soper D J, Stavisky S D, Welkenhuysen M, Dutta B, Shenoy K V 2022 Large-scale neural recordings with single neuron resolution using neuropixels probes in human cortex *Technical Report* (Nature Publishing Group)
[61] Petersen P C, Siegle J H, Steinmetz N A, Mahallati S and Buzsáki G 2021 Cellexplorer: a framework for visualizing and characterizing single neurons *Neuron* **109** 3594–608

[62] Pillow J W, Shlens J, Chichilnisky E, Simoncelli E P and Krekelberg B 2013 A model-based spike sorting algorithm for removing correlation artifacts in multi-neuron recordings *PLoS One* **8** e62123
[63] Pouzat C, Delescluse M, Viot P and Diebolt J 2004 Improved spike-sorting by modeling firing statistics and burst-dependent spike amplitude attenuation: a markov chain monte carlo approach *J. Neurophysiol.* **91** 2910–28
[64] Rácz M, Liber C, Németh E, Fiáth R, Rokai J, Harmati I, Ulbert I and Márton G 2020 Spike detection and sorting with deep learning *J. Neural Eng.* **17** 016038
[65] Rey H G, Pedreira C and Quiroga R Q 2015 Past, present and future of spike sorting techniques *Brain Res. Bull.* **119** 106–17
[66] Rokai J, Rácz M, Fiáth R, Ulbert I and Márton G 2021 Elvisort: encoding latent variables for instant sorting, an artificial intelligence-based end-to-end solution *J. Neural Eng.* **18** 046033
[67] Rokem A, Dichter B, Holdgraf C and Ghosh S 2021 Pan-neuro: interactive computing at scale with brain datasets
[68] Rosenblatt F 1958 The perceptron: a probabilistic model for information storage and organization in the brain *Psychol. Rev.* **65** 386
[69] Rossant C and Harris K D 2013 Hardware-accelerated interactive data visualization for neuroscience in Python *Front. Neuroinform.* **7** 36
[70] Rossant C *et al* 2016 Spike sorting for large, dense electrode arrays *Nat. Neurosci.* **19** 634–41
[71] Roux L, Stark E, Sjulson L and Buzsáki G 2014 *In vivo* optogenetic identification and manipulation of gabaergic interneuron subtypes *Curr. Opin. Neurobiol.* **26** 88–95
[72] Rübel O, Tritt A J, Ly R, Dichter B K, Ghosh S S, Niu L, Soltesz I, Svoboda K, Frank L M and Bouchard K 2021 The neurodata without borders ecosystem for neurophysiological data science *bioRxiv Preprint* (https://doi.org/10.1101/2021.03.13.435173)
[73] Rutishauser U, Schuman E M and Mamelak A N 2006 Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, *in vivo J. Neurosci. Methods* **154** 204–24
[74] Ruz I D and Schultz S R 2014 Localising and classifying neurons from high density mea recordings *J. Neurosci. Methods* **233** 115–28
[75] Saif-ur Rehman M, Ali O, Dyck S, Lienkämper R, Metzler M, Parpaley Y, Wellmer J, Liu C, Lee B, Kellis S 2021 Spikedeep-classifier: a deep-learning based fully automatic offline spike sorting algorithm *J. Neural Eng.* **18** 016009
[76] Saif-ur Rehman M *et al* 2019 Spikedeeptector: a deep-learning based method for detection of neural spiking activity *J. Neural Eng.* **16** 056003
[77] Sedaghat-Nejad E, Fakharian M A, Pi J, Hage P, Kojima Y, Soetedjo R, Ohmae S, Medina J F and Shadmehr R 2021 P-sort: an open-source software for cerebellar neurophysiology *J. Neurophysiol.* **126** 1055–75
[78] Shabestari P S, Buccino A P, Kumar S S, Pedrocchi A and Hierlemann A 2021 A modulated template-matching approach to improve spike sorting of bursting neurons *2021 IEEE Biomedical Circuits and Conf. (BioCAS)* (IEEE) pp 1–4
[79] Shan K Q, Lubenov E V and Siapas A G 2017 Model-based spike sorting with a mixture of drifting t-distributions *J. Neurosci. Methods* **288** 82–98
[80] Siegle J H *et al* 2019 A survey of spiking activity reveals a functional hierarchy of mouse corticothalamic visual areas *bioRxiv Preprint* p 805010 (available at: www.nature.com/articles/s41586-020-03171-x)
[81] Souza B C, Lopes-dos Santos V, Bacelo J and Tort A B 2019 Spike sorting with gaussian mixture models *Sci. Rep.* **9** 1–14
[82] Steinmetz N 2021 Imposed motion datasets from Steinmetz et al Science 2021
[83] Steinmetz N A *et al* 2021 Neuropixels 2.0: a miniaturized high-density probe for stable, long-term brain recordings *Science* **372** 6539
[84] Steinmetz N A, Koch C, Harris K D and Carandini M 2018 Challenges and opportunities for large-scale electrophysiology with neuropixels probes *Curr. Opin. Neurobiol.* **50** 92–100
[85] Teeters J L *et al* 2015 Neurodata without borders: creating a common data format for neurophysiology *Neuron* **88** 629–34
[86] Toosi R, Akhaee M A and Dehaqani M-R A 2021 An automatic spike sorting algorithm based on adaptive spike detection and a mixture of skew-t distributions *Sci. Rep.* **11** 1–18
[87] Valencia D and Alimohammad A 2020 Neural spike sorting using binarized neural networks *IEEE Trans. Neural Syst. Rehabil. Eng.* **29** 206–14
[88] Van Pelt J, Vajda I, Wolters P S, Corner M A and Ramakers G J 2005 Dynamics and plasticity in developing neuronal networks *in vitro Prog. Brain Res.* **147** 171–88
[89] Varol E, Boussard J, Dethe N, Winter O, Urai A, Laboratory T I B, Churchland A, Steinmetz N and Paninski L 2021 Decentralized motion inference and registration of neuropixel data *ICASSP 2021-2021 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE) pp 1085–9
[90] Veerabhadrappa R, Ul Hassan M, Zhang J and Bhatti A 2020 Compatibility evaluation of clustering algorithms for contemporary extracellular neural spike sorting *Front. Syst. Neurosci.* **14** 34
[91] Wehr M, Pezaris J S and Sahani M 1999 Simultaneous paired intracellular and tetrode recordings for evaluating the performance of spike sorting algorithms *Neurocomputing* **26** 1061–8
[92] Wei Z, Lin B-J, Chen T-W, Daie K, Svoboda K, Druckmann S and Gutkin B S 2020 A comparison of neuronal population dynamics measured with calcium imaging and electrophysiology *PLoS Comput. Biol.* **16** e1008198
[93] Werner T, Vianello E, Bichler O, Garbin D, Cattaert D, Yvert B, De Salvo B and Perniola L 2016 Spiking neural networks based on oxram synapses for real-time unsupervised spike sorting *Front. Neurosci.* **10** 474
[94] Wood F, Black M J, Vargas-Irwin C, Fellows M and Donoghue J P 2004 On the variability of manual spike sorting *IEEE Trans. Biomed. Eng.* **51** 912–8
[95] Wouters J, Kloosterman F and Bertrand A 2021 A data-driven spike sorting feature map for resolving spike overlap in the feature space *J. Neural Eng.* **18** 0460a7
[96] Wouters J, Kloosterman F and Bertrand A 2021 Shybrid: a graphical tool for generating hybrid ground-truth spiking data for evaluating spike sorting performance *Neuroinformatics* **19** 141–58
[97] Wu T, Rátkai A, Schlett K, Grand L and Yang Z 2019 Learning to sort: few-shot spike sorting with adversarial representation learning *2019 41st Annual Int. Conf. IEEE Engineering in Medicine and Biology Society (EMBC)* (IEEE) pp 713–6
[98] Wu T, Zhao W, Keefer E and Yang Z 2018 Deep compressive autoencoder for action potential compression in large-scale neural recording *J. Neural Eng.* **15** 066019
[99] Yger P and Gilson M 2015 Models of metaplasticity: a review of concepts *Front. Comput. Neurosci.* **9** 138
[100] Yger P, Spampinato G L, Esposito E, Lefebvre B, Deny S, Gardella C, Stimberg M, Jetter F, Zeck G and Picaud S 2018 A spike sorting toolbox for up to thousands of electrodes validated with ground truth recordings *in vitro* and *in vivo eLife* **7** e34518
[101] Yuan X *et al* 2020 Versatile live-cell activity analysis platform for characterization of neuronal dynamics at single-cell and network level *Nat. Commun.* **11** 1–14