



# Continuing progress of spike sorting in the era of big data

David Carlson<sup>1</sup> and Lawrence Carin<sup>2</sup>

Engineering efforts are currently attempting to build devices capable of collecting neural activity from one million neurons in the brain. Part of this effort focuses on developing dense multiple-electrode arrays, which require post-processing via 'spike sorting' to extract neural spike trains from the raw signal. Gathering information at this scale will facilitate fascinating science, but these dreams are only realizable if the spike sorting procedure and data pipeline are computationally scalable, at or superior to hand processing, and scientifically reproducible. These challenges are all being amplified as the data scale continues to increase. In this review, recent efforts to attack these challenges are discussed, which have primarily focused on increasing accuracy and reliability while being computationally scalable. These goals are addressed by adding additional stages to the data processing pipeline and using divide-and-conquer algorithmic approaches. These recent developments should prove useful to most research groups regardless of data scale, not just for cutting-edge devices.

## Addresses

<sup>1</sup> Civil and Environmental Engineering, Biostatistics and Bioinformatics, Duke University, United States

<sup>2</sup> Electrical and Computer Engineering, Duke University, United States

Corresponding author: Carlson, David ([david.carlson@duke.edu](mailto:david.carlson@duke.edu))

**Current Opinion in Neurobiology** 2019, 55:90–96

This review comes from a themed issue on **Machine learning, big data, and neuroscience**

Edited by **Maneesh Sahani** and **Jonathan Pillow**

<https://doi.org/10.1016/j.conb.2019.02.007>

0959-4388/© 2019 Elsevier Ltd. All rights reserved.

## Introduction

'Spike sorting' is the process of extracting neural spike trains from electrophysiological data recorded by electrodes implanted in extracellular brain tissue. This general procedure is now a classic technique in the field, with a literature dating back decades [1,2]. Because action potentials are considered to be a fundamental unit of information in the brain, these extracted neural spike trains are typically used in basic science applications. Despite the critical importance and ubiquity of this

preprocessing step for scientific analysis, the procedures for 'spike sorting' are still undergoing significant changes and still possess open problems.

Some of these recent developments have been pushed by the changing nature of data collection. Traditionally, multiple electrodes were implanted to maximize information and minimize redundancy, so electrodes were placed far apart [3] and the data from each electrode could be processed independently. A classic review paper covers much of the original developments on processing a single electrode [4], and this methodology largely extends to tetrodes. However, a single electrode is limited in its ability to distinguish between neurons, so there has been significant interest in creating dense Multi-Electrode Arrays (MEAs), where the signal-to-noise power is increased by observing neurons over multiple electrodes [5]. These dense MEAs require that data from all electrodes are processed simultaneously to account for the overlapping statistical properties of the spike trains. A review paper in 2012 in this same venue addressed the initial algorithmic developments on these devices as the field expanded from a small set of electrodes to a thousand neurons [6], and has been an object of continued discussion [7,8].

Because these dense MEAs are increasing in prevalence, such as the thousand-electrode Neuropixels probe [9], there has been a flurry of activity to address the algorithmic problems that arise from this drastic increase in scale. In addition to many of the classical problems in spike-sorting, such as low signal-to-noise and the fact that the number of measured neurons is unknown [4], computational challenges are now coming to the forefront. Current devices already record from tens of thousands of electrodes [10,11], and there are efforts to increase the number of measured neurons to the scale of millions<sup>3</sup>, further rendering spike sorting as the computational bottleneck in the data analysis pipeline.

As these data goals are realized, the quantity and quality of the data will be a blessing to the field if it can be adequately processed and interpreted. The spike sorting pipeline is crucial to these goals, and must be complementary to the data collection. There are several necessary conditions for the pipeline. First, it must require minimal human intervention. In many research groups,

<sup>3</sup> <https://www.darpa.mil/program/neural-engineering-system-design>.

combining automatic procedures with manual sorting or corrections is the *de facto* approach, but manual interventions are neither reliable [12] nor even feasible at the data scale being considered [13<sup>••</sup>]. Second, the pipeline needs to be scalable, as computationally heavy techniques for a single electrode do not scale to dense MEAs. Third, the pipeline needs to be reliable, such that results are similar from repeated runs or minor perturbations of the data. If the goal is reproducible science, it is critical to know which neural spike trains are trustworthy [14<sup>•</sup>].

There have been many manuscripts over the last several years that address the computational challenges and reliability of spike sorting in dense MEAs. Here, we discuss recent development, starting with how the methodology has changed even for a single electrode, and how these methods are being extended to dense MEAs. After discussing algorithmic advances, recent proposals for evaluating the pipelines in the absence of ground truth will be addressed.

### The modern spike sorting pipeline

Given the voltage recordings from all the electrodes, the spike sorting pipeline attempts to recover the spike trains of all the neurons with sufficient signal-to-noise power. However, it is unknown *a priori* how many neurons there are and what an action potential from a given neuron looks like. This turns into a ‘cocktail party’ problem, where we know there are many neurons active simultaneously and need to infer how many neurons and their unique signatures [6]. Because of the historical nature of the spike sorting problem, there are a multitude of methods attempting to solve this issue. However, a general structure for a multi-stage procedure has emerged that most recent algorithms follow. This has led to an increased focus on a modular pipeline [13<sup>••</sup>], where individual stages can be swapped to improve the overall performance.

The general procedure for a single electrode is outlined briefly in Figure 1. In general, the core algorithmic stages of the pipeline typically begin with appropriately pre-processed electrophysiological data, typically raw data from all electrodes after going through high-pass filtering, spatial whitening, and common average removal steps [15]. The goal is to recover the number of neurons, the shape of an action potential for each neuron, and the times that each neuron spikes from the preprocessed data. This is fundamentally an issue of demixing the time series (or source separation) into its component spike trains and background signal. Instead of directly addressing the demixing problem, these properties have been traditionally estimated by first running a detection algorithm on the data to recover snippets that indicate (unsorted) action potentials from all neurons (Figure 1(b)), then clustering action potentials based on their shape or extracted features (Figure 1(c)) [4].

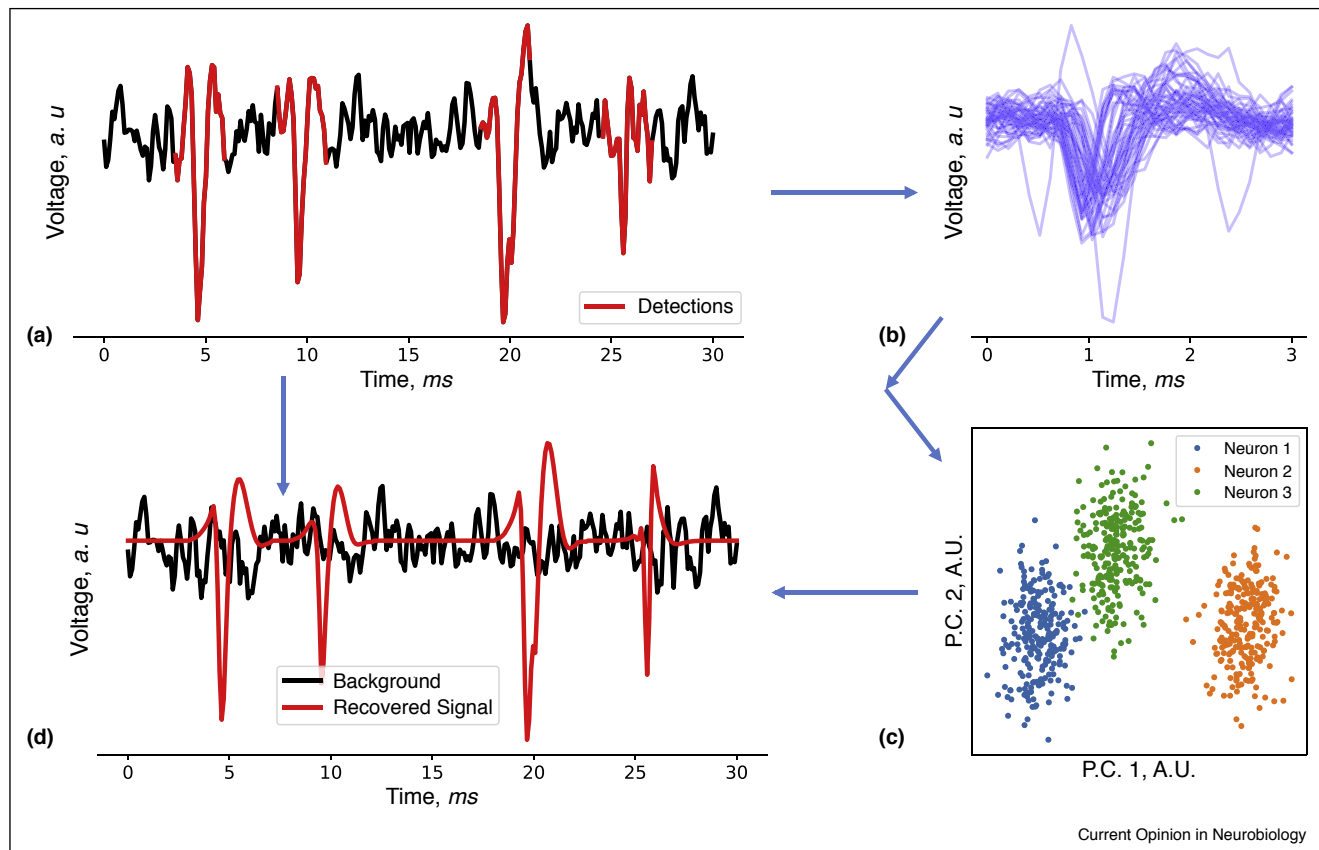
Each detected cluster nominally represents a single neuron’s activity. This clustering step can be tricky — *a priori* one does not know the appropriate number of neurons nor the appropriate features. This has received significant historical and continued attention, with strategies such as information metrics on Gaussian Mixture Models [4], Bayesian nonparametrics [16–18], spectral clustering [19], consensus-based clustering [20], superparamagnetic clustering [21], and fast peak search [22,23<sup>••</sup>] being considered amongst many others. Additionally, there has been significant work in adjusting these algorithms for long-term tracking of neural waveforms [24<sup>•</sup>,25], which necessitates accounting for both waveform [26–28] and electrode drift [29<sup>••</sup>].

In this traditional pipeline, the output from the clustering could be directly used to construct spike trains for each neuron. However, this clustering approach solves a surrogate problem to our fundamental goal of demixing or source separation of the complete time series into component neurons. The accuracy of these output spike trains is dependent not just on the quality of the feature extraction and clustering, but also on the quality of the detection. Furthermore, the feature-extraction and clustering steps are greatly hindered by the problem of ‘overlapping’ or ‘collided’ spikes [15,30], where two or more neurons fire action potentials in the same temporal and spatial vicinity. An example model and collision are visualized in Figure 2 (a). These overlapping spikes corrupt the feature space (Figure 2(c)), lowering the signal-to-noise power and interfering with effective clustering and the inference of the number of neurons. Therefore, clustering approaches are typically successful in identifying well-isolated spikes, but capturing overlapping waveforms requires directly addressing the demixing or source separation problem. Toward this end, there have been several recent proposals to use iterative methods based on template matching or deconvolution algorithms, where the original signal is modeled as the superposition of sparse spike trains and background noise [15,27,30–32,33<sup>••</sup>]. These methods have shown significant promise to recover overlapping waveforms and improve detection quality, and are increasingly being used in practice.

### Moving to dense MEAs

The surrogate problem solved by traditional pipelines (detection then clustering) struggles when overlapping waveforms increase. Because recently developed dense MEAs increase the signal-to-noise ratio (SNR) [34], more action potentials are detectable, increasing the fraction of (observable) overlapping waveforms. Capturing more low-SNR neurons with these devices require directly addressing the overlapping spike issue. Therefore, most modern algorithmic frameworks for dense MEAs consider a source separation model [13<sup>••</sup>,23<sup>••</sup>,31,33<sup>••</sup>,35–37,38<sup>•</sup>]. When observed electrophysiological data are demixed into component neurons, deconvolution as one of the

Figure 1



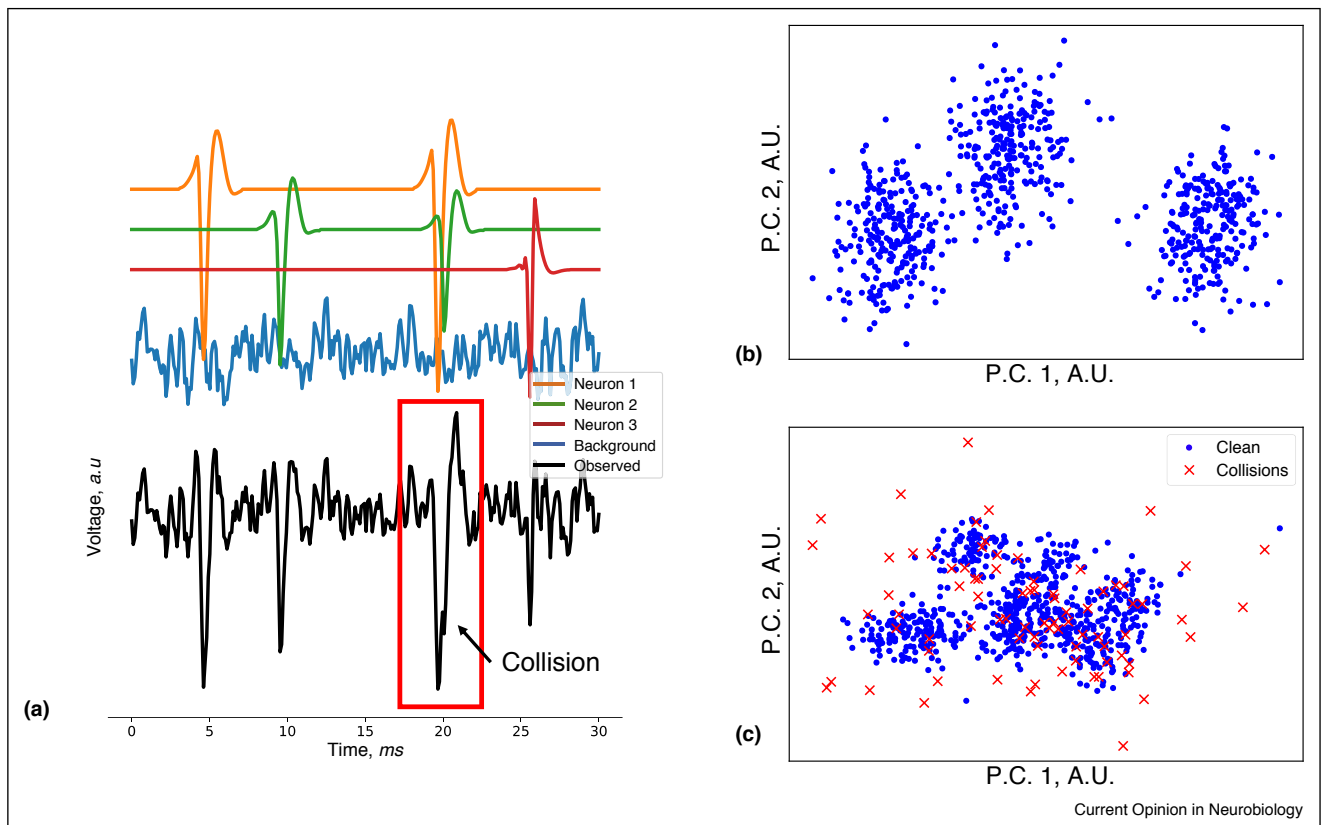
**(a)** An example electrophysiological signal that has gone through a low-pass filtered signal, constructed using the 'hybrid' strategy discussed in 'Evaluating and Benchmarking Algorithms'. Areas in red are detected waveforms, which are used to create detected snippets. **(b)** Snippets output from a detection algorithm. The raw detections contain waveforms that are not aligned and include outliers and collisions. **(c)** After the detected snippets have been cleaned, features are extracted and then clustered. **(d)** Using the results from the clustering, a deconvolution algorithm (or alternative) is used to recover the spike waveform signal and extract it from the background signal. The deconvolution is necessary to improve the quality of the detection.

core algorithmic steps. Unfortunately, deconvolution is a computationally expensive and memory-intensive operation; this is especially true if the algorithm is used naively over an expanding set of channels. One approach to address this problem is to first use a detection and clustering approach, and run deconvolution a single time to detect overlapping waveforms [13<sup>••</sup>,23<sup>••</sup>]. Alternatively, approximations can be made to speed up inference in deconvolution, such as in the popular Kilosort algorithm [33<sup>••</sup>]. Notably, if the decision to use a clustering approach before deconvolution is chosen, then the clustering algorithm only needs to recover the waveform shapes for the neurons, not sort every spike. This leads to an 'triage-sort-recover' strategy used in the Yet Another Spike Sorter (YASS) pipeline [13<sup>••</sup>], where atypical or collided waveforms are put aside to be recovered during deconvolution, and only clean waveforms are used to construct the feature space and cluster [13<sup>••</sup>]. Such an

approach can significantly improve the feature space, visualized in Figure 2(b).

A major area of ongoing research is to create computationally efficient approaches of the above pipelines to scale to dense MEAs [13<sup>••</sup>,23<sup>••</sup>,31,33<sup>••</sup>,35–37,38<sup>•</sup>], where many stages naively struggle to scale to increased data size. Succinctly, the number of neurons is expected to increase linearly with the number of electrodes, and the data/features for each electrode also increase linearly; thus, a naïve algorithm will typically scale quadratically in time (or worse) with respect to the number of channels. Therefore, it is critical to utilize spatial locality and computationally effective techniques in all stages of the pipeline. An increasingly common approach is to explicitly learn a 'masking' vector that denotes presence or absence on each electrode for each detection and neuron [13<sup>••</sup>,33<sup>••</sup>,37,38<sup>•</sup>], visualized in Figure 3. The

Figure 2



(a) The model of the waveforms consists of the superposition of spike trains from individual neurons and a background signal. When two or more neurons have a temporally (and spatially) overlapping action potentials, this is a 'collided' spike. (b) If only clean, well-aligned detected waveforms are used, PCA creates a clearly clusterable feature space. (c) Outliers are well-known to corrupt feature spaces. Here, we show the same system where both collisions and clean waveforms are used to compute the PCA feature extraction.

masking vector is determined by comparing the peak measurement on each channel for a detected spike to the standard deviation over all channels [37]. This masking allows implicit updates during algorithms, critically reducing the amount of data considered [37]. A similar approach can also be used in deconvolution, as is done in the Kilosort algorithm [33<sup>••</sup>]. An additional requirement is that the waveform should only be present on a single spatial neighborhood to disentangle simultaneously firing neurons from distinct locations, which can be enforced by processing only a local neighborhood of an electrode array (divide-and-conquer) [13<sup>••</sup>, 23<sup>••</sup>, 33<sup>••</sup>, 37].

A common manual post-processing step is to merge clusters that are believed to represent the same neuron, which can increase robustness in practice. These merges can happen because of a small difference in the feature space or an apparent temporal offset. This problem is exacerbated by the proliferation of neurons and how they are split in divide-and-conquer algorithmic approaches. Therefore, most pipelines include some *post-hoc* steps after clustering

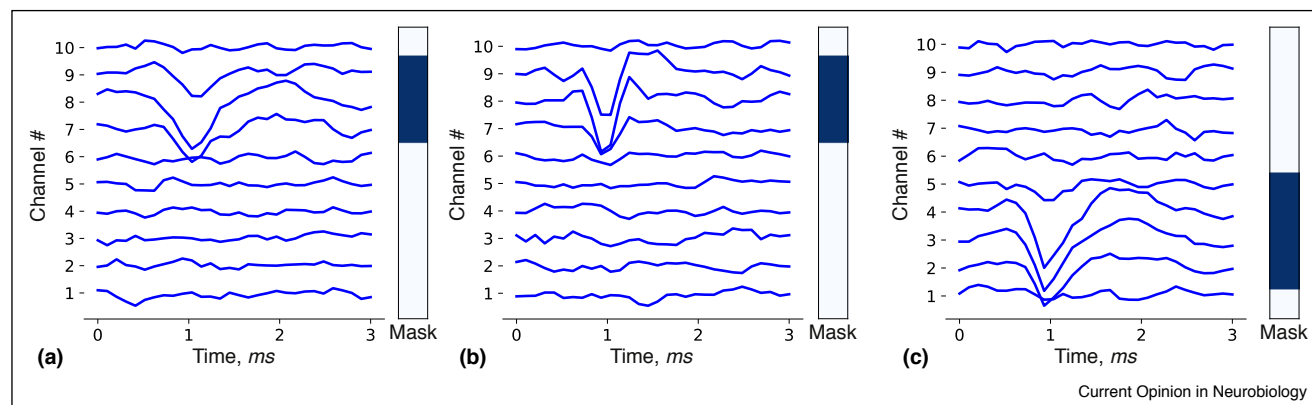
and deconvolution to merge related waveforms, which enhances performance [23<sup>••</sup>, 29<sup>••</sup>, 33<sup>••</sup>, 37, 38<sup>•</sup>, 39].

### Evaluating and benchmarking algorithms

The previously discussed pipelines share many commonalities, but ultimately differ in many ways. It is important to be able to evaluate the performance of the algorithms to choose between them and optimize to improve the pipeline's accuracy, consistency, and efficiency. Ideally, we would like to be able to compare a predicted spike train to ground truth. Unfortunately, due to the nature of the problem, ground truth data is hard to come by. Because of this limitation, many alternative strategies have been and continue to be developed to evaluate performance.

First, pipelines can be evaluated on partial ground truth. Specifically, individual neurons can be measured separately from the primary recording device, creating a dataset where one or more neurons has high quality ground truth spiking information. Previously, this type of result in a tetrode was

Figure 3



(a) An example waveform detection over a 10-channel linear array. The masking vector denotes which channel the waveform has non-trivial energy on, which must be accounted for in the modeling and pipeline. (b) A second example waveform. This waveform exists on the same channels as the first waveform, and must be accounted for in the modeling. (c) A third example waveform. Note that because this waveform does not exist on the same channels as the first two examples, it can be separated in the processing pipeline to take advantage of parallelization.

used to validate different algorithms [5], and there are a few such dense MEA datasets [23<sup>••</sup>,40–43]. Alternatively, it is possible to create a biophysically supported reference sort by using biophysical properties and hand-validation [13<sup>••</sup>,29<sup>••</sup>]. For example, in the primate retina the cell types and their responses to white noise are well-known [44]; therefore, a sorting result can be validated neuron-by-neuron for whether its stimuli responses match a known cell type. While such an approach has limitations, notably with neurons at low SNR, limited brain regions, potential unknown cell types, and potential inconsistencies of hand sorting, this type of benchmark is important to demonstrate that a given pipeline can replicate human expert performance [13<sup>••</sup>].

As a surrogate to ground truth data, one can use high-quality synthetically generated data from a biophysically based generator such as VisaPy [45], Neuron [46], or Neurocube [47]. Synthetic data can also be constructed by simulating a few synthetic spike trains and superimposing them into real datasets via a ‘hybrid’ approach, which is frequently used to perform evaluations [14<sup>•</sup>,33<sup>••</sup>,41]. Note that synthetic data do not have to perfectly simulate a biophysical system to be useful; if a pipeline struggles with controlled synthetic data, similar struggles should be expected with real data. Simulation techniques are constantly increasing in quality, which should yield improvements in the synthetic data and benchmarking in the coming years.

Instead of comparing how accurate a clustering is to ground truth, an alternative solution is to evaluate stability. These metrics attempt to answer the question of how much a sorting result will change under a minor perturbation of a dataset [14<sup>•</sup>], which can measure both the total stability of a

sort and the reproducibility of each individual detected neuron. Unstable results have questionable reproducibility, and scientific conclusions should proceed with care [48,49]. Only using clusters or neurons with high stability can increase the robustness of conclusions [13<sup>••</sup>,14<sup>•</sup>]. For example, if one is interested in precise stimuli responses of individual neurons, then only highly reliable neurons should be included. These highly reliable neurons can be chosen by only including units with stability metrics above a given threshold (for example, neurons that are 90% or more similar across different sorts) [14<sup>•</sup>], with the unfortunate downside that many potentially real neurons will be discarded. In lieu of stability metrics, per-cluster quality can be assessed by SNR [21], isolation [29<sup>••</sup>], 1-d projection distance [29<sup>••</sup>], Mahalanobis distance [18], and several others [50]. Biophysical properties, such as refractory period violations [15,50] or patterns in cross-correlograms indicative of colliding waveforms [30], are also helpful in identifying low-quality clusters. A critical consideration is how robust individual units need to be for inclusion in downstream analysis. When determining precise stimuli responses, a high threshold on reliability is needed; in contrast, accurate sorting does not appear to significantly enhance performance in brain–computer interfaces, were primarily only prediction matters [51].

Finally, slow algorithms are not used in practice, and many research groups do not have access to high-performance computing. Therefore, considerations on computational efficiency, memory usage, and scalability are important. These metrics are often addressed in manuscripts [13<sup>••</sup>,23<sup>••</sup>] in addition to real world considerations such as on-chip feasibility [52] or real-time sorting [27,31]. Already, hundreds of channels can be processed in real-time [13<sup>••</sup>,23<sup>••</sup>,33<sup>••</sup>].



## Discussion

Over the past several years, there has been a flurry of activity in spike sorting methodology, and improved tools are becoming increasingly available for use with publicly released code. In order to determine the best procedures, it is critical that data sharing and public validation increase along with code availability, which is the goal of ongoing projects [53<sup>\*</sup>]. All pipelines mentioned in this review performed the best on their datasets and benchmarks, with little overlap between datasets. These datasets come from different devices, brain regions, animal models, and behavioral tasks, so direct comparisons are difficult. Therefore, a crucial contribution to improving the spike-sorting pipeline will come from open competitions of greater size and variability of data. Ideally, this will facilitate learning how to best modify pipelines for each situation and data type. Furthermore, the goal of these algorithms is to be used in research, so it is important to have publicly available code and an active user community.

The pipelines discussed in this manuscript are already significantly better than the tools available even a few years ago, and should enhance electrophysiological analysis for all research groups. As the field develops these newer devices, there will be increasing computational challenges, with parallel processing and divide-and-conquer algorithms increasing in importance. However, while the current and future data scales present unsolved algorithmic problems, this size of data is enthralling for neuroscientists, allowing a greater lens to neural activity in the brain, more than motivating addressing these issues.

## Role of the funding source

The funders of the study had no role in study design, data collection, data analysis, data interpretation, or writing of the report. The corresponding author had full access to all the data in the study and had final responsibility for the decision to submit for publication.

## Conflict of interest statement

Nothing declared.

## Acknowledgements

We would like to thank Liam Paninski for helpful comments. D.C. received funding from the National Institutes of Health by grant R01MH099192-05S2.

## References and recommended reading

Papers of particular interest, published within the period of review, have been highlighted as:

- of special interest
- of outstanding interest

1. Schmidt EM: **Computer separation of multi-unit neuroelectric data: a review.** *J Neurosci Methods* 1984, **12**:95-111.
2. Yang X, Shamma SA: **A totally automated system for the detection and classification of neural spikes.** *IEEE Trans Biomed Eng* 1988, **35**:806-816.
3. Maynard EM, Nordhausen CT, Normann RA: **The Utah Intracortical Electrode Array: a recording structure for potential brain-computer interfaces.** *Electroencephalogr Clin Neurophysiol* 1997, **102**:228-239.
4. Lewicki MS: **A review of methods for spike sorting: the detection and classification of neural action potentials.** *Netw Comput Neural Syst* 1998, **9**:R53-R78.
5. Harris KD, Henze DA, Csicsvari J, Hirase H, Buzsáki G, Abeles M, Gerstein G, Abeles M, Goldstein M, Anderson T et al.: **Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements.** *J Neurophysiol* 2000, **84**:401-414.
6. Einevoll GT, Franke F, Hagen E, Pouzat C, Harris KD: **Towards reliable spike-train recordings from thousands of neurons with multielectrodes.** *Curr Opin Neurobiol* 2012, **22**:11-17.
7. Rey HG, Pedreira C, Quiroga RQ: **Past, present and future of spike sorting techniques.** *Brain Res Bull* 2015, **119**:106-117.
8. Lefebvre B, Yger P, Marre O: **Recent progress in multi-electrode spike sorting methods.** *J Physiol* 2016, **110**:327-335.
9. Steinmetz NA, Koch C, Harris KD, Carandini M: **Challenges and opportunities for large-scale electrophysiology with Neuropixels probes.** *Curr Opin Neurobiol* 2018, **50**:92-100.
10. Rios G, Lubenov EV, Chi D, Roukes ML, Siapas AG: **Nanofabricated neural probes for dense 3-D recordings of brain activity.** *Nano Lett* 2016, **16**:6857-6862.
11. Tsai D, Sawyer D, Bradd A, Yuste R, Shepard KL: **A very large-scale microelectrode array for cellular-resolution electrophysiology.** *Nat Commun* 2017, **8**:1802.
12. Wood F, Black MJ, Vargas-Irwin C, Fellows M, Donoghue JP: **On the variability of manual spike sorting.** *IEEE Trans Biomed Eng* 2004, **51**:912-918.
13. Lee J, Carlson D, Shokri H, Yao W, Goetz G, Hagen E, Batty E, Chichilnisky EJ, Einevoll G, Paninski L: **Yass: yet another spike sorter.** *Adv Neural Inf Process Syst* 2017, **30**:1-11.
- Develops a fast, multi-stage pipeline using a 'triage-cluster-recover' approach to effectively deal with overlapping waveforms that scales to hundreds of electrodes in real-time. Code is available in the YASS package for Python.
14. Barnett AH, Magland JF, Greengard LF: **Validation of neural spike sorting algorithms without ground-truth information.** *J Neurosci Methods* 2016, **264**:65-77.
- Proposes a suite of techniques to evaluate spike sorters without ground truth. Code is available in the valid spike package for Matlab.
15. Pillow JW, Shlens J, Chichilnisky EJ, Simoncelli EP: **A model-based spike sorting algorithm for removing correlation artifacts in multi-neuron recordings.** *PLoS One* 2013, **8**:e62123.
16. Wood F, Goldwater S, Black MJ: **A non-parametric Bayesian approach to spike sorting.** *Conf Proc IEEE Eng Med Biol Soc* 2006, **1**:1165-1168.
17. Wood F, Black MJ: **A nonparametric Bayesian alternative to spike sorting.** *J Neurosci Methods* 2008, **173**:1-12.
18. Carlson DE, Vogelstein JJT, Wu Q, Lian W, Zhou M, Stoetzner CRC, Kipke D, Weber D, Dunson DDB, Carin L: **Multichannel electrophysiological spike sorting via joint dictionary learning & mixture modeling.** *IEEE TBME* 2014, **61**:41-54.
19. Chah E, Hok V, Della-Chiesa A, Miller JJH, O'Mara SM, Reilly RB: **Automated spike sorting algorithm based on Laplacian eigenmaps and k-means clustering.** *J Neural Eng* 2011, **8**:016006.
20. Fournier J, Mueller CM, Shein-Idelson M, Hemberger M, Laurent G: **Consensus-based sorting of neuronal spike waveforms.** *PLoS One* 2016, **11**:e0160494.
21. Quiroga RQ, Nadasdy Z, Ben-Shaul Y: **Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering.** *Neural Comput* 2004, **16**:1661-1687.
22. Rodriguez A, Laio A: **Clustering by fast search and find of density peaks.** *Science (80-)* 2014, **344**:1492-1496.

23. Yger P, Spampinato GL, Esposito E, Lefebvre B, Deny S,
    - Gardella C, Stimberg M, Jetter F, Zeck G, Picaud S *et al.*: **A spike sorting toolbox for up to thousands of electrodes validated with ground truth recordings in vitro and in vivo.** *eLife* 2018, **7**:e34518.
 Develops a fast, multi-stage pipeline based on fast search methods for dense multi-electrode arrays. Code is available in the spiking-circus package for Python.
  24. Dhawale AK, Poddar R, Wolff SB, Normand VA, Kopelowitz E,
    - Ölveczky BP: **Automated long-term recording and analysis of neural activity in behaving animals.** *eLife* 2017, **6**.
 Shows impressive recent progress in the long-term tracking of individual neurons.
  25. Okun M, Lak A, Carandini M, Harris KD: **Long term recordings with immobile silicon probes in the mouse cortex.** *PLoS One* 2016, **11**:e0151180.
  26. Calabrese A, Paniski L: **Kalman filter mixture model for spike sorting of non-stationary data.** *J Neurosci Methods* 2011, **196**:159-169.
  27. Carlson D, Rao V, Vogelstein JT, Carin L: **Real-time inference for a gamma process model of neural spiking.** *NIPS*. 2013.
  28. Shan KQ, Lubenov EV, Siapas AG: **Model-based spike sorting with a mixture of drifting t-distributions.** *J Neurosci Methods* 2017, **288**:82-98.
  29. Chung JE, Magland JF, Barnett AH, Tolosa VM, Tooker AC, Lee KY,
    - Shah KG, Felix SH, Frank LM, Greengard LF: **A fully automated approach to spike sorting.** *Neuron* 2017, **95**:1381-1394.e6.
 A full pipeline to address spike sorting in dense MEAs that shows impressive performance and speed. Code is available in the Mountain-Sort package for Python, including visualization tools.
  30. Ekanadham C, Tranchina D, Simoncelli EP: **A unified framework and method for automatic neural spike identification.** *J Neuro Methods* 2014, **222**:47-55.
  31. Franke F, Natora M, Boucsein C, Munk MHJ, Obermayer K: **An online spike detection and spike classification algorithm capable of instantaneous resolution of overlapping spikes.** *J Comput Neurosci* 2010, **29**:127-148.
  32. Ekanadham C, Tranchina D, Simoncelli E: **A blind deconvolution method for neural spike identification.** *NIPS*. 2011.
  33. Pachitariu M, Steinmetz NA, Kadir SN, Carandini M, Harris KD:
    - **Fast and accurate spike sorting of high-channel count probes with KiloSort.** *NIPS*. 2016:4448-4456.
 A fast multi-channel deconvolution approach that has shown good collision recovery. Code is available in the Kilosort package in Matlab.
  34. Jun JJ, Steinmetz NA, Siegle JH, Denman DJ, Bauza M, Barbarits B, Lee AK, Anastassiou CA, Andrei A, Aydn Ç *et al.*: **Fully integrated silicon probes for high-density recording of neural activity.** *Nature* 2017, **551**:232-236.
  35. Muthmann J-O, Amin H, Sernagor E, Maccione A, Panas D, Berdondini L, Bhalla US, Hennig MH: **Spike detection for large neural populations using high density multielectrode arrays.** *Front Neuroinform* 2015, **9**:28.
  36. Mokri Y, Salazar RF, Goodell B, Baker J, Gray CM, Yen S-C: **Sorting overlapping spike waveforms from electrode and tetrode recordings.** *Front Neuroinform* 2017, **11**:53.
  37. Kadir SN, Goodman DFM, Harris KD: **High-dimensional cluster analysis with the masked EM algorithm.** *Neural Comput* 2014, **26**:2379-2394.
  38. Rossant C, Kadir SN, Goodman DFM, Schulman J, Hunter MLD, Saleem AB, Grosmark A, Belluscio M, Denfield GH, Ecker AS *et al.*: **Spike sorting for large, dense electrode arrays.** *Nat Neurosci* 2016, **19**:634-641.
- Full pipeline that demonstrated how a masking vector and localization can be used to speed inference. Code is available in the Phy package in Python.
39. Magland JF, Barnett AH: *Unimodal Clustering Using Isotonic Regression: ISO-SPLIT*. . arXiv Prepr arXiv150804841 2015.
  40. Neto JP, Lopes G, Frazão J, Nogueira J, Lacerda P, Baião P, Aarts A, Andrei A, Musa S, Fortunato E *et al.*: **Validating silicon polytodes with paired juxtacellular recordings: method and dataset.** *J Neurophysiol* 2016, **116**:892-903 bioRxiv.
  41. Spampinato GL, Esposito E, Yger P, Duebel J, Picaud S, Marre O: *Ground Truth Recordings for Validation of Spike Sorting Algorithms*. 2018 <http://dx.doi.org/10.5281/ZENODO.1205233>.
  42. Franke F, Pröpper R, Alle H, Meier P, Geiger JRP, Obermayer K, Munk MHJ: **Spike sorting of synchronous spikes from local neuron ensembles.** *J Neurophysiol* 2015, **114**:2535-2549.
  43. Allen BD: *Ground Truth in Ultra-Dense Neural Recording*. 2017.
  44. Dacey DM, Peterson BB, Robinson FR, Gamlin PD: **Fireworks in the primate retina: in vitro photodynamics reveals diverse LGN-projecting ganglion cell types.** *Neuron* 2003, **37**:15-27.
  45. Hagen E, Ness TV, Khosrowshahi A, Sørensen C, Fyhn M, Hafting T, Franke F, Einevoll GT: **ViSAPy: a Python tool for biophysics-based generation of virtual spiking activity for evaluation of spike-sorting algorithms.** *J Neuro Methods* 2015, **245**:182-204.
  46. Lytton WW, Seidenstein AH, Dura-Bernal S, McDougal RA, Schürmann F, Hines ML: **Simulation neurotechnologies for advancing brain research: parallelizing large networks in NEURON.** *Neural Comput* 2016, **28**:2063-2090.
  47. Martinez J, Pedreira C, Ison MJ, Quiñero R: **Realistic simulation of extracellular recordings.** *J Neurosci Methods* 2009, **184**:285-293.
  48. Todorova S, Sadtler P, Batista A, Chase S, Ventura V: **To sort or not to sort: the impact of spike-sorting on neural decoding performance.** *J Neural Eng* 2014, **11**:056005.
  49. Febinger HY, Dorval AD, Rolston JD: **A sordid affair: spike sorting and data reproducibility.** *Neurosurgery* 2018, **82**:N19-N20.
  50. Hill DN, Mehta SB, Kleinfeld D: **Quality metrics to accompany spike sorting of extracellular signals.** *J Neurosci* 2011, **31**:8699-8705.
  51. Fraser GW, Chase SM, Whitford A, Schwartz AB: **Control of a brain-computer interface without spike sorting.** *J Neural Eng* 2009, **6**:055004.
  52. Gibson S, Judy JW, Markovi D: **Spike sorting: the first step in decoding the brain.** *IEEE Signal Process Mag* 2012, **29**:124-143.
  53. Teeters JL, Godfrey K, Young R, Dang C, Friedsam C, Wark B,
    - Asari H, Peron S, Li N, Peyrache A *et al.*: **Neurodata without borders: creating a common data format for neurophysiology.** *Neuron* 2015, **88**:629-634.
 Discusses challenges in creating common standard and data processing pipelines in neuroscience, which can improve portability of both data and code.