Journal of Neuroscience Methods 264 (2016) 65–77

Contents lists available at ScienceDirect

Journal of Neuroscience Methods

journal homepage: www.elsevier.com/locate/jneumeth

Validation of neural spike sorting algorithms without ground-truth information

Alex H. Barnett[a], Jeremy F. Magland[b], Leslie F. Greengard[c]

[a] Simons Center for Data Analysis, and Department of Mathematics, Dartmouth College, United States
[b] Simons Center for Data Analysis, and Department of Radiology, University of Pennsylvania, United States
[c] Simons Center for Data Analysis, and Courant Institute, New York University, United States

HIGHLIGHTS

- We present per-neuron validation metrics for automatic spike sorting algorithms.
- The metrics measure stability under perturbations consistent with those in the data.
- A standardized interface assesses any algorithm, independent of its internal workings.
- We illustrate and test the metrics on *in vivo* and *ex vivo* recordings with overlapping spikes.

ARTICLE INFO

Article history:
Received 28 August 2015
Received in revised form 18 February 2016
Accepted 26 February 2016
Available online 28 February 2016

Keywords:
Validation
Automatic
Spike sorting
Algorithms
Stability

ABSTRACT

Background: The throughput of electrophysiological recording is growing rapidly, allowing thousands of simultaneous channels, and there is a growing variety of spike sorting algorithms designed to extract neural firing events from such data. This creates an urgent need for standardized, automatic evaluation of the quality of neural units output by such algorithms.
New method: We introduce a suite of validation metrics that assess the credibility of a given automatic spike sorting algorithm applied to a given dataset. By rerunning the spike sorter two or more times, the metrics measure stability under various perturbations consistent with variations in the data itself, making no assumptions about the internal workings of the algorithm, and minimal assumptions about the noise.
Results: We illustrate the new metrics on standard sorting algorithms applied to both *in vivo* and *ex vivo* recordings, including a time series with overlapping spikes. We compare the metrics to existing quality measures, and to ground-truth accuracy in simulated time series. We provide a software implementation.
Comparison with existing methods: Metrics have until now relied on ground-truth, simulated data, internal algorithm variables (e.g. cluster separation), or refractory violations. By contrast, by standardizing the interface, our metrics assess the reliability of *any* automatic algorithm without reference to internal variables (e.g. feature space) or physiological criteria.
Conclusions: Stability is a prerequisite for reproducibility of results. Such metrics could reduce the significant human labor currently spent on validation, and should form an essential part of large-scale automated spike sorting and systematic benchmarking of algorithms.

1. Introduction

One of the most powerful and widely used methods for studying neuronal activity *in vivo* (for example, in behaving animals) or *ex vivo* (for example, in extracted retinal preparations) is direct electrical recording. Using either single electrodes, tetrodes, or high-density multielectrode arrays, the experimental data consist of voltage patterns measured by sensors at the electrode tips, which are typically in the extracellular space. The strongest signals arise from an unknown but modest number of neurons in the immediate vicinity of each sensor, superimposed on a background consisting substantially of signals due to more and more distant, electrically shielded neurons (Buzsáki et al., 2012). *Spike sorting* is the name given to algorithms that detect distinct firing events and associate those events with specific neurons. The input consists of voltage traces from one or more electrodes and the output is a list of

E-mail address: ahb@math.dartmouth.edu (A.H. Barnett).

http://dx.doi.org/10.1016/j.jneumeth.2016.02.022
0165-0270/

individual neurons that have been identified, as well as the spiking (firing) times for each. The literature on this subject is vast, and a very partial list of references includes (Einevoll et al., 2012; Fee et al., 1996; Gibson et al., 2012; Harris et al., 2000; Lewicki, 1998; Quiroga, 2007, 2012).

In this environment, a typical data processing pipeline consists of (1) filtering the signal to remove high frequency noise and low frequency baseline drift, (2) spike detection, windowing and alignment, and (3) clustering based on the spike shape. The latter step, which is typically the core of the spike sorting algorithm, is predicated on the critical assumption that the electrical signals from distinct neurons have distinct "shapes," discussed in more detail below. Finally, once the algorithm has assigned a neuron identity to each spike, various metrics are employed to assess the quality of the classification. In many experiments, the signals are, in fact, *nonstationary*. That is, the voltage pattern due to the firing of each neuron may vary over time. For the sake of simplicity, we will restrict our attention here to stationary data, but note that the framework for quality assessment that we introduce below does not depend on stationarity.

Unfortunately, despite major progress in algorithms and software, spike sorting remains a labor-intensive task with a substantial manual component, both in the clustering and quality assessment phases. Furthermore, in the last two decades, the number of channels from which it is becoming possible to record simultaneously has grown from one (or a few) to thousands (Eversmann et al., 2003; Berdondini et al., 2009; Einevoll et al., 2012; Li et al., 2015; Rossant et al., 2015; Müller et al., 2015). As a result, it has become an urgent matter to accelerate the data analysis, to automate the clustering algorithms, and to develop statistical protocols that can provide robust estimates of the accuracy of the neuronal identification.

Despite some important work on validation, discussed in the next section, however, there are currently no established standards in the community either for estimating errors or for assessing the reproducibility of the output of spike sorting software. Moreover, as noted in Hill et al. (2011), Neymotin et al. (2011), Pouzat et al. (2002), and Schmitzer-Torbert et al. (2005), it is of particular importance to be able to assess the fidelity of the output for each of the identified neurons separately. This is vital information in subsequent modeling, since the error rate from the spike sorting phase may permit the inclusion of some neurons and the exclusion of others when making inferences about neural circuitry, depending on the sensitivity of the question being asked.

In this paper, we concentrate on the problem of quality assessment, and propose a framework for estimating confidence in the results obtained from a *black box* spike sorting algorithm, in the absence of ground truth data such as an intra-cellular reference electrode. Since rigorous estimates of accuracy are not obtainable in this environment, we develop statistical measures of stability instead, drawing on ideas from bootstrapping and cross-validation in statistics and machine learning (Rand, 1971; Yu, 2013; Zaki and Meira, 2014; Lange et al., 2004; von and Tishby, 2009). One of the important features of our framework is that quality estimation is carried out in an *algorithm-agnostic* fashion, not as an internal part of the spike sorting algorithm itself. This requires a standard interface for the data that is compatible with all (or most) algorithms, and the ability to invoke the spike sorting procedure without manual intervention. Our validation scheme, in fact, needs to be able to re-run the spike sorting software two or more times; see Fig. 1.

We hope that the stability metrics and neuron-by-neuron confidence measures described here will serve, in part, as a step toward systematizing the external evaluation of spike sorting algorithms and, in part, as motivation to fully automate the spike sorting process. Indeed, we see this as crucial to progress in the analysis of increasingly large-scale electrophysiology data. Finally, we should note that we do not claim to present a novel algorithm for spike sorting—we will illustrate our metrics using conventional clustering and time series analysis tools.

The structure of this paper is as follows. In the remainder of the introduction we overview some existing approaches to validation (Section 1.1), then describe (Section 1.2) the two versions of spike sorting that we consider: the simpler sorting of individual spikes or "clips" (which have already been detected and aligned), and the more realistic sorting of a full time series. Standardizing the *interfaces* to these two algorithms is crucial for wide applicability of our metrics. Section 2 presents four schemes of increasing complexity for validating spike sorting on clips, and illustrates them on a standard PCA/k-means sorting algorithm applied to *in vivo* rodent data. Then Section 3 presents two schemes for validating the spike sorting of time series, and illustrates them on *ex vivo* monkey retina data. In Section 3.4 we synthesize time series in order to show how the metrics compare to ground-truth accuracy. We conclude with a summary and discussion in Section 4. Some methodological and mathematical details are gathered in the two appendices.

1.1. Prior work on quality metrics

As noted above, there has been substantial progress made over the last decade or so in developing methods for assessing the output of existing spike sorting methods. Broadly speaking, these consist of (a) statistical or information-theoretic measures of "isolation" of the clusters of firing events associated with each neuron and (b) physiological criteria for detecting errors such as refractory period violations. False positives are defined as firing events that are associated with neuron j, but should have been grouped with some other neuron i, with $i \neq j$. False negatives are defined as firing events that should have been associated with neuron j, but were either discarded or grouped with some other neuron. We do not seek to review these approaches here, and refer the reader to Hill et al. (2011), Neymotin et al. (2011), Pouzat et al. (2002), and Schmitzer-Torbert et al. (2005). While the establishment of such particular metrics is important, those of type (a) have until this point relied on accessing *internal* data (e.g., PCA components) that are particular to certain types of algorithms while excluding many other successful approaches (e.g., ICA, model-based fitting, and optimal filters).

Spike sorting algorithms can, of course, be run on simulated data, where ground truth is available. Thus, the development of more and more realistic forward models for neuronal geometry, neuronal firing, and instrumentation noise is an important goal (Camuñas-Mesa and Quiroga, 2013; Hagen et al., 2015). Recently another approach has emerged, called "surrogate" spikes (Marre et al., 2012) or "hybrid datasets" (Rossant et al., 2015), consisting of the original voltage recordings, to which are added new spikes (known waveforms extracted either from different electrodes in the same recording (Marre et al., 2012), or from a different recording with the same equipment (Rossant et al., 2015)), at known times. Running the spike sorting procedure again on the new dataset provides a powerful validation test, particularly since it permits testing fidelity in the presence of overlapping spikes (Prentice et al., 2011; Franke et al., 2015a), which tend to result in errors using simple clustering schemes.

Despite all of these advances, however, most algorithms still involve manual intervention either at late stages of the clustering process or in quality assessment. While there have been some attempts to validate algorithms with human-operated steps (Prentice et al., 2011), we believe that limiting human intervention to an early, exploratory phase, followed by a fully automated execution of the spike sorting software will be essential for establishing standards of reproducibility and quality, particularly with high-density multi-electrode arrays.

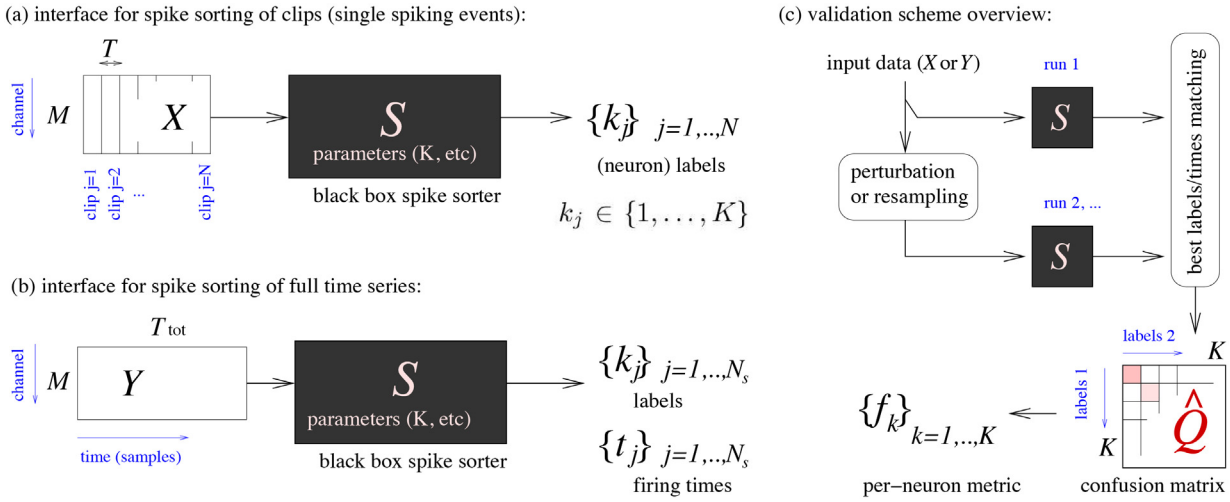

Fig. 1. Interfaces and validation of automatic spike sorting algorithms. The algorithm S to be validated is inside the "black box", and is only accessible via the interface (a) or (b). In (a) the spike sorter assigns the jth clip a label (neural unit) k_j from 1 to K. In (b) the spike sorter acts on a whole time series Y and returns a number N_s of firing events (determined by the algorithm) described by labels k_j and firing times t_j in $[0, T_{\text{tot}}]$. (c) Overview of the validation scheme, applying to both clip-based and time series based algorithms, requiring the ability to rerun the algorithm S as needed, and outputting the stability metric f_k.

The related, and broader, problem of quality metrics in clustering algorithms has received much recent attention (Lange et al., 2004; Zaki and Meira, 2014; von and Tishby, 2009; Hennig, 2007). However, many such metrics are not directly applicable to spike sorting, since they require access to internal variables that may be present in some algorithms and not others, and since they usually seek to assess the *overall* quality of the clustering (e.g. see (Rand, 1971); for an exception, see (Hennig, 2007)). In contrast, with spike sorting the concept of "overall quality" is not useful: there are often high-amplitude spikes that cluster well, plus a "tail" of spikes of decreasing amplitudes that cluster arbitrarily poorly, becoming indistinguishable from measurement noise. Thus, as realized by many researchers (e.g. (Schmitzer-Torbert et al., 2005; Pillow et al., 2013)), per-neuron metrics (as opposed to overall performance metrics) are crucial.

1.2. Set-up of the problem and validation schemes

We consider *interfaces* to two spike sorting tasks (Fig. 1(a) and (b)). The first is a simple classification of isolated events, which will serve to introduce ideas, whereas the second is a rather general spike sorting interface:

1. The sorter classifies *clips*, i.e. short time-windows, each presumed to contain a single spiking event. We assume that the clips have already been extracted (by some detection process) and aligned (so that events of the same type occur at the same time within the clip). For each clip, an integer label is produced. More formally, let $X := \{x_{mtj}\}_{m=1,\dots,M,\ \ t=1,\dots,T,\ \ j=1,\dots,N}$ be a three-dimensional array with N clips, each of duration T time samples and M channels (electrodes). The sorting algorithm S performs the map
$$S(X) = \{k_j\}_{j=1}^{N} =: \mathbf{k} \tag{1}$$
assigning a label (neuron identity) $k_j \in \{1, \dots, K\}$ to the jth clip. K may be a fixed internal parameter of the algorithm, or may be a variable to be determined by the algorithm.
2. The sorter identifies all firing times and labels within a time-series $Y = \{y_{mt}\}_{m=1,\dots,M,\ \ t=0,\dots,T_{\text{tot}}-1}$, i.e.
$$S(Y) = \{t_j, k_j\}_{j=1}^{N_s} \tag{2}$$
where N_s is the number of spikes found (determined by the algorithm), while $t_j \in [0, T_{\text{tot}}]$ and $k_j \in \{1, \dots, K\}$ are their real-valued firing times and integer labels. In contrast to the clip-based interface, this allows S to handle overlapping spiking events, which is crucial for complete event detection when firing rates are high (Prentice et al., 2011; Pillow et al., 2013; Ekanadham et al., 2013).

The former interface captures the clustering task common to much software (Lewicki, 1998; Harris et al., 2000; Rossant et al., 2015). The latter encompasses essentially all types of automatic sorting algorithms,[1] including those based on template matching (Prentice et al., 2011; Marre et al., 2012; Pillow et al., 2013; Ekanadham et al., 2013; Franke et al., 2015b), ICA (Takahashi et al., 2002; Moore-Kochlacs et al., 2014) or filters (Franke et al., 2010), that have no simple detection step. Note that in neither interface are waveform shapes output, since these can be reconstructed from the inputs and outputs. We require that any algorithm parameters (thresholds, clustering settings, electrode information, etc) are *fixed*, and inaccessible via the validation interface.

Our key output comparison tool is the following matrix (see Fig. 1(c)). Given two size-N lists of labels, $k_j \in \{1, \dots, K\}, j = 1, \dots, N$, and $l_j \in \{1, \dots, L\}, j = 1, \dots, N$, the *confusion matrix* (or contingency table (Zaki and Meira, 2014, Ch. 17)) is a rectangular matrix Q whose k, l entry is the number of spikes labeled k in the first list and l in the second, i.e.
$$Q_{k,l} := \#\{j : k_j = k, l_j = l\}. \tag{3}$$
Since usually the ordering of the labels is arbitrary, we seek the "best" permutation of one of the sets of labels, in the sense of maximizing the sum of diagonal entries of the permuted Q matrix. This best-permuted confusion matrix $\hat{Q}$ is defined by
$$\hat{Q}_{k,l} := Q_{k,\hat{\pi}(l)}, \quad \text{where} \quad \hat{\pi} = \arg\max_{\pi \in S_L} \sum_{k=1}^{\min(K,L)} Q_{k,\pi(k)} \tag{4}$$
where S_L is the set of permutations of L elements. Large diagonal entries mean that the two sets of labels are consistent; off-diagonal

[1] Probabilistic (fuzzy) algorithms that output a probability density over spike parameters such as (Wood and Black, 2008; Carlson et al., 2013) could be adapted to our interface by selecting as output either the mode of the density, or a random sample from it.

entries indicate the amount and type of variation between the two labelings. The assignment problem of finding $\hat{\pi}$ can be solved in polynomial time by applying Kuhn's "Hungarian algorithm" (Kuhn, 1955) to $-Q$. When firing times are also present, we will need an extended confusion matrix, which we postpone until Section 3.

2. Clip-based validation schemes

In this section, we present some clip-based validation schemes, proceeding from simple to more sophisticated. Since there is no ground truth data, we must focus on the idea of *stability*. If a neuron is not stable, then is has a low probability of being accurate. We illustrate the schemes on a set of 6901 upsampled and peak-aligned clips extracted from an *in vivo* rat motor cortex recording, as described in Appendix A.1.

2.1. A standard clip-based spike sorting algorithm

Since our goal is to validate existing algorithms rather than present new ones, we use as our default a standard spike-sorting algorithm that we refer to as PCA/k-means++, as follows. Clips are organized into a matrix $A \in \mathbb{R}^{MT\times N}$ such that each column contains the time-series for all channels for one clip. Dimension reduction (from dimension $MT = 1470$ to dimension $N_{\text{fea}} = 10$ by default) is then done by replacing each column by its first N_{fea} PCA components.[2] Then, treating each column (clip) as a point in $\mathbb{R}^{N_{\text{fea}}}$, k-means++ (Arthur and Vassilvitskii, 2007) is used for clustering, which requires a user-specified K. We remind the reader that k-means++ is a variant of k-means using a certain favorable random initialization of the centroids. The converged clustering produced often depends on initialization, i.e. the k-means iteration is not often able to find the global minimum of the sum of squared distances from points to their assigned centroids (this is in general *NP*-hard (Arthur and Vassilvitskii, 2007)). Thus, as is standard, we repeat k-means++ r times and choose the repeat with the minimum sum of squared distances. The result is a label k_j for each clip. The labels are permuted so that the l_2-norms of the estimated spike waveforms, i.e. $(\sum_{mt}[W^{(k)}_{mt}]^2)^{1/2}$, decrease with increasing k. For this the estimated spike waveforms $W^{(k)}$ are found by simple averaging over all clips which have been assigned the same label k. More formally,

$$W^{(k)}_{mt} = \frac{1}{n_k}\sum_{j:k_j=k} x_{mtj}, \quad m = 1,\dots,M, \quad t = 1,\dots,T, \quad k = 1,\dots,K \tag{5}$$

where

$$n_k = n_k(\mathbf{k}) := \#\{j : k_j = k\} \tag{6}$$

is the kth population in the labeling $\mathbf{k}$.

Fig. 2(a) and (b) shows the result of this sorting algorithm, for $K = 8$ neurons and $r = 100$ repeats, running on the clip dataset. Note that we do not claim that this is an optimal algorithm; our goal is merely to use it to illustrate the presented validation schemes.

2.2. A stability metric based on rerunning

Our goal is to validate a spike sorting algorithm acting on a particular dataset of clips. If the spike sorter is non-deterministic, then comparing two runs on exactly the same clip data (with different random seeds) gives a baseline measure of reproducibility for each label. Let $\{k_j\} = \mathbf{k}$ and $\{l_j\} = \mathbf{l}$ be the sets of labels returned from two such runs, with K and L label types respectively (in this work usually $L = K$). Let $\hat{Q}_{k,l}$ be the elements of their best-permuted confusion matrix (4). Then define for each label k the *stability*,[3]

$$f_k := \frac{2\hat{Q}_{k,k}}{n_k(\mathbf{k}) + n_k(\mathbf{l})}, \quad k = 1,\dots,\min(K,L) \tag{7}$$

Samples of f_k are now computed by performing this process for many independent pairs of runs, with the same K, and the mean $\bar{f}_k$ is reported. We use the notation f^{rerun}_k to indicate the "rerun" stability metric. Note that the pairs of runs used to sample the metric should not be confused with the r "repeats" that are internal to this particular sorting algorithm S.

We illustrate this for our default sorting algorithm (fixing $K = 8$, taking the best of $r = 10$ k-means++ repeats) in Fig. 2(c), which shows 20 independent samples of f^{rerun}_k with quantiles and means. This shows that the first four spike types are stable (as expected from the large waveform norms in panel (a) and well-separated clusters in (b)), while the last four are less so. The last ($k = 8$) is the least stable, and from its tiny waveform it might be judged not to represent a single neural unit.

Remark 1. We argue that 20 samples of any metric f_k are more than sufficient for validation, because the resulting standard deviation of the estimator of the mean is then somewhat smaller than the width of the underlying distribution of f_k. If a neuron has a wide distribution of f_k, it is unstable, and high accuracy in estimating $\bar{f}_k$ is not needed. This follows wisdom from the Monte Carlo literature (MacKay, 1998, Sec. 7.5).

However, as Fig. 3 shows, as a larger number of repeats r is used, the stability for each label $k = 1,\dots,K$ grows. By $r = 100$ all stabilities have converged to very close to 1, i.e. there is very little variation in labeling due to random initialization. This is because picking the best of many repeats finds a sum of squares distances to centroids that is closer to the global minimum. The algorithm has effectively become deterministic, hence the rerun metric is uninformative. Since we wish to validate deterministic spike sorting methods (e.g., ones based on hierarchical or density-based clustering (Zaki and Meira, 2014, Ch. 14–15)), it is clear that one must introduce variation into the input X, which we now do.

2.3. A cross-validation metric using subsampling

In machine learning it is common to use cross-validation (CV) to assess an algorithm. The simplest version splits the data into training and test subsets, and uses the performance on the test set to estimate the generalizability of a classifier built using the training set. Although CV requires an observed dependent variable, a similar idea can be applied to spike sorting evaluation even in the absence of ground truth information. We divide the N clips randomly into three similar-sized subsets I, II, and III, and train a classifier C_{I} with only the data I, and a classifier C_{II} with only the data II. Finally we compute the best-permuted confusion matrix (4) between the labels produced by classifying (labeling) the data III using C_{I} and using C_{II}, then from this compute stabilities f^{CV}_k for each spike type, as before using (7). Other more elaborate variants involving m-fold CV are clearly possible; we do not test them here.

We now describe the classifier. Our clip-based interface Fig. 1(a) does not include a classifier, yet we can build a simple one by taking the label of the mean waveform (5) that is closest (in some metric)

[2] Numerically, A is replaced by the first N_{fea} columns of V^TA, where V is the matrix whose columns are the eigenvectors of AA^T ordered by descending eigenvalue.

[3] This is similar to the "F-measure" of a clustering (Zaki and Meira, 2014, Eq. (17.1)), or harmonic mean of the precision and recall, relative to a ground truth clustering, with the difference that a fixed matching is enforced between the two sets of labels.

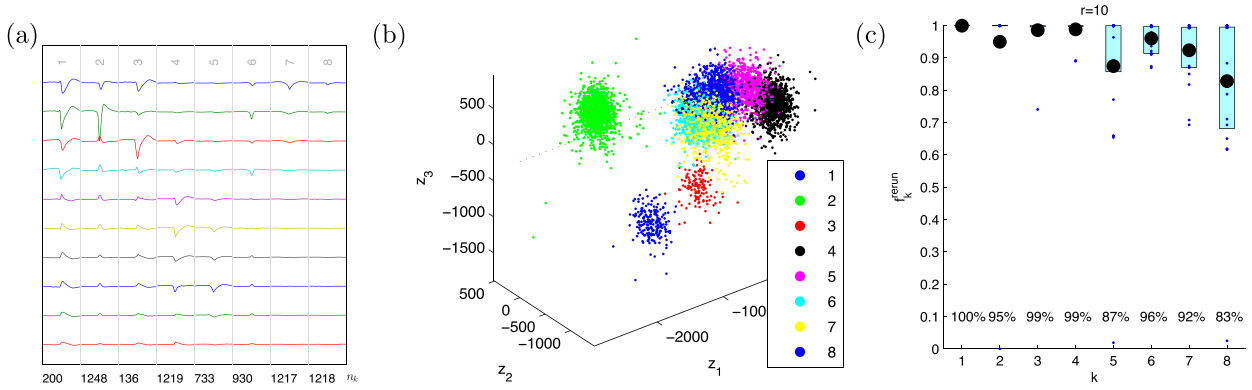

Fig. 2. Spike-sorting of clips extracted from a rat motor cortex recording, using PCA/k-means++ (see Section 2) with $K=8$ and best of $r=100$ repeats. (a) Average waveforms $W^{(k)}$ for $k=1,\ldots,K$, (see (5)). Each column is a single waveform k, with its firing population n_k noted below. Each of the horizontal traces is one of the $M=10$ channels. The upwards-spiking channels are a result of spatial pre-whitening (Appendix A.1). (b) The labeling, each shown as a point in the feature space of the first three PCA components. Colors indicate label k_j as in the inset. (c) 20 samples of the "rerun" stability metric of Section 2.2, for PCA/k-means++ with $K=8$ and $r=10$. The large dot (and percentage below) shows the mean $\tilde{f}_k^{\text{rerun}}$, the small dots show all samples of f_k^{rerun}, and the bar shows their 25–75% quantiles. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

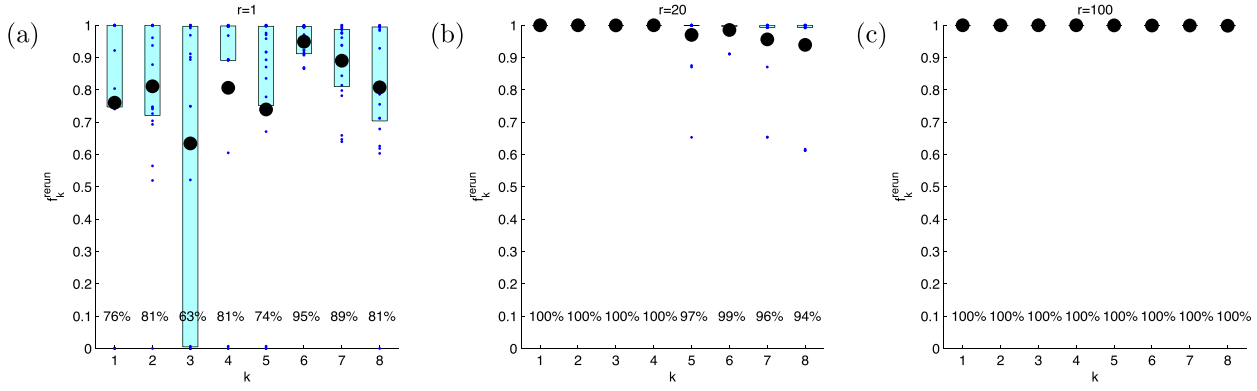

Fig. 3. The "rerun" stability metric f_k^{rerun} for PCA/k-means++ with $K=8$ fixed, for various r (best of r repeats of k-means++). (a) $r=1$, (b) $r=20$, (c) $r=100$. Plots should be interpreted as in Fig. 2(c). This shows convergence of all label stabilities to 1 (hence an uninformative metric) as r increases.

to each clip in question. That is, given a set of clips X which gives spike-sorted labels $\mathbf{k}=S(X)$, the estimated waveforms $W^{(k)}$, $k=1,\ldots,K$ are computed via (5); they form the information needed for the classifier. Applying the classifier to new clips $X' := \{x'_{mtj}\}$ gives new labels according to

$$k'_j = \underset{k\in\{1,\ldots,K\}}{\arg\min} \sum_{mt} \left(x'_{mtj} - W^{(k)}_{mt}\right)^2,$$

where for simplicity the l_2 norm has been chosen as the distance metric.[4] Finally, since the waveform norm ordering of Section 2.1 may vary for independent subsamplings, then in order to collect meaningful per-label statistics of f_k^{CV} we must permute the labelings from later repetitions to best match the first. For this we apply the Hungarian algorithm to the matrix of squared distances (l_2-norms) between all pairs of mean waveforms in the repetitions in question.

Fig. 4(a) shows this stability metric (for 20 independent 3-way splits) for the PCA/k-means++ spike sorter with $K=8$ and $r=100$. Notice that the last four spike types have stabilities in the range 70–90%, in contrast to Fig. 3(c) for which they were all 100%. This shows that the three-way data subsampling introduces variation that detects unstable label assignments, even for an essentially deterministic spike sorter.

We now show that the 3-way CV metric is able to detect an erroneously split cluster. For this we generate a simple "toy" dataset comprising just the $N=1584$ clips with labels $k_j=1$, 2 or 3 from the default $K=8$ and $r=100$ spike sorting (Fig. 2). Thus there are three very well-separated true clusters in feature space with a clear ground-truth; see Fig. 4(b). For this toy dataset we pick a standard sorting algorithm of PCA/k-means++ with $K=4$ (chosen to be erroneously large), $r=100$, and $N_{\text{fea}}=2$ (to aid visualization). Running this algorithm on the toy data gives the labeling in two dimensions of feature space shown in Fig. 4(b), and waveforms of Fig. 4(c): the largest true cluster is erroneously split into two nearly-equal pieces, which are given new labels $k=2$, 3 (hence the similar pair of waveforms in (c)). There are many ways to summarize accuracy[5] here (Zaki and Meira, 2014, Sec. 17.3.1): filling the confusion matrix (4) of size 3×4 between truth and the $K=4$ clustering gives the F-measures (7) for the three true labels as $f_2=0.701$, $f_1=f_3=1$. Running the CV metric on this $K=4$ sorting algorithm results in the stabilities in Fig. 4(d): a mean around 87% for $k=2$, 3 (somewhat higher than measures of accuracy), and 100% for the other (unsplit) clusters. The point of our framework is that this conclusion is reached via *only* the standard interface S, i.e. without

[4] This is appropriate for an iid Gaussian noise model; a more realistic noise model is described in Appendix A.2. Also see Prentice et al. (2011).

[5] All four clusters have precision 1, but $k=2$, 3 have recall around 0.5.

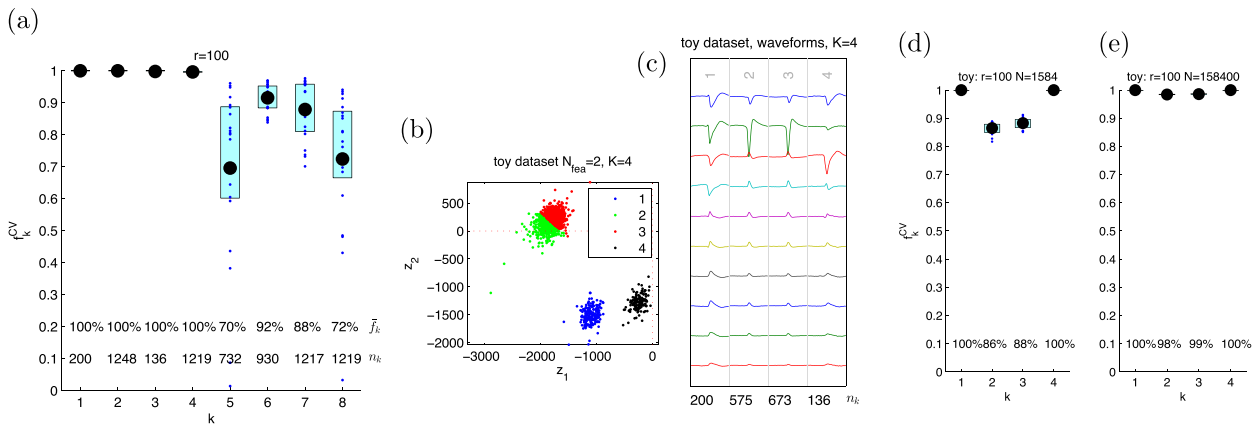

Fig. 4. Three-way cross-validation stability metric for PCA/k-means++ (best of $r = 100$ repeats). (a) Stabilities for $K = 8$, and populations from a single run of the sorter. (b) The two PCA components of the "toy" data (extracted from the labels $k \leq 3$) with $N = 1584$, labeled with PCA/k-means++ with $K = 4$ and $N_{\text{fea}} = 2$. (c) Estimated waveforms from the run shown in (b). (d) Stability metrics for the $K = 4$ labels on the data from (b).(e) Same for an expanded dataset with a 100 times larger N.

access to the feature vectors shown in (b), nor of course the ground truth.

However, the variation in labeling induced by subsampling is small compared to the known accuracy (one reason being that a small N_{fea} raises the stability), and furthermore dies away entirely as the number of clips grows, $N \to \infty$. We illustrate this in Fig. 4(e), which shows the 3-way CV metric for the same algorithm as (d) but running on a set of clips 100 times larger, generated as described in Appendix A.2. The stabilities of the $k = 2, 3$ labels of the split cluster are now around 98.5%, hardly an indication of instability. This is consistent with a central limit theorem proven (Shamir and Tishby, 2009) for stability metrics in a wide class of clustering algorithms, when their global optimum is unique, implying in our setting the convergence

$$1 - f_k^{\text{CV}} = \mathcal{O}(N^{-1/2}), \quad \text{for each} \quad k.$$

Indeed, increasing N by a factor of 100 decreased the deviation of f_k from unity by around a factor of 10. We expect the prefactor for this deviation to be smaller when the split cluster is more *non-spherical*: resampling induces high instability in splitting a spherical cluster, but in k-means a non-spherical cluster (as in Fig. 4(b)) tends to split consistently along a hyperplane perpendicular to its longest axis.

This unfortunate phenomenon that, regardless of the apparent quality of the labeling, stability tends to unity as N grows has been known in the clustering literature for some time (Krieger and Green, 1999). Renormalizing a global stability metric by $\sqrt{N}$ is possible (von and Tishby, 2009, Sec. 3.1.1), but it is not clear how to do this when a *per-neuron* metric is needed. On the other hand, restricting to small-N subsamples purely to induce instability has the major disadvantage that the spike sorter is never validated on a dataset of the desired size.

Since we seek a scheme which can validate spike sorting applied to arbitrarily large datasets, we must move beyond subsampling, and explore perturbing the clips themselves.

2.4. Metrics based on data perturbations without subsampling

Fig. 4(b)–(e) shows that for large N, stability under subsampling fails to indicate a highly erroneous split cluster. By examining feature space (Fig. 4(b)) it is clear that a decision boundary of k-means passes through a high-density region. Can one create a metric that quantifies this undesirable property without access to feature space, using only the standard interface? We now present two such metrics, both of which involve perturbing the data.

2.4.1. Self-blurring

Firstly we describe a method that we name *self-blurring* for reasons that will become clear. Let $\mathbf{k} = S(X)$ be the output of the sorter on the full set of clips, and $\{W^{(k)}\}_{k=1,\ldots,K}$ be the mean waveforms found via (5). Let π be a permutation of the N clips that randomizes *only within each label class*. Precisely, if $J_k := \{j : k_j = k\}$ is the index set of the kth label type, and π_k is a random permutation of n_k elements, then the overall permutation π is defined by

$$\pi(J_k(i)) = J_k(\pi_k(i)), \qquad i = 1, \ldots, n_k$$

holding for each label type $k = 1, \ldots, K$. From this we create a perturbed clip dataset $\tilde{X}$ with elements

$$\tilde{x}_{mtj} = x_{mtj} + \gamma(x_{mt,\pi(j)} - W^{(k_j)}_{mt}), \quad m = 1, \ldots, M, \quad t = 1, \ldots, T, \quad j = 1, \ldots, N, \tag{8}$$

where $\gamma > 0$ is a parameter controlling the size of the perturbation. Running the sorter on this new data gives $\tilde{\mathbf{k}} = S(\tilde{X})$, then we compute the best-permuted confusion matrix (4) between labels $\mathbf{k}$ and $\tilde{\mathbf{k}}$, and from it obtain the per-label stabilities f_k^{blur} via (7).

The bracketed expression in (8) can be interpreted as a random sample from the "noise distribution" of the k_jth cluster in clip signal space, relative to its mean waveform (centroid). So, with $\gamma = 1$, the cluster distribution about its centroid is convolved (blurred) with itself. The result is that, even if a decision boundary remains fixed, if there is density near this boundary then points are carried across it by the convolution, thus change label, and are registered as instability. The new data $\tilde{X}$ is consistent with the original X in the sense that it differs only by additive noise of exactly the type observed within each cluster. No assumption on the form of the noise model or noise level is made.

Since $\mathbb{R}^{MT}$ (signal space) is hard to visualize, we again "peek under the hood" of our PCA/k-means++ algorithm, by in Fig. 5(b) showing the result of self-blurring on the 2D feature vector points. Notice that there is significant overlap of red and green points, and rotation of one of the decision lines. The self-convolution also results in larger clusters, which may cause label exchange between clusters that were distinct (e.g. see the tails of the lower two clusters). For an isolated Gaussian cluster, its size (noise level) grows by a factor $\sqrt{2}$. Fig. 5(e) shows the distribution of f_k^{blur} for 20 samples of $\tilde{X}$. The mean around 75% for $k = 2, 3$, being similar to the actual F-measure accuracy of 70%, is a clear indicator of instability. For the dataset of size $100N$ this also holds; thus, we have cured the large-N vanishing of instability present with CV.

We show an idealized example where points are drawn from a symmetric distribution $p(z)$ in a 1D variable z, erroneously split

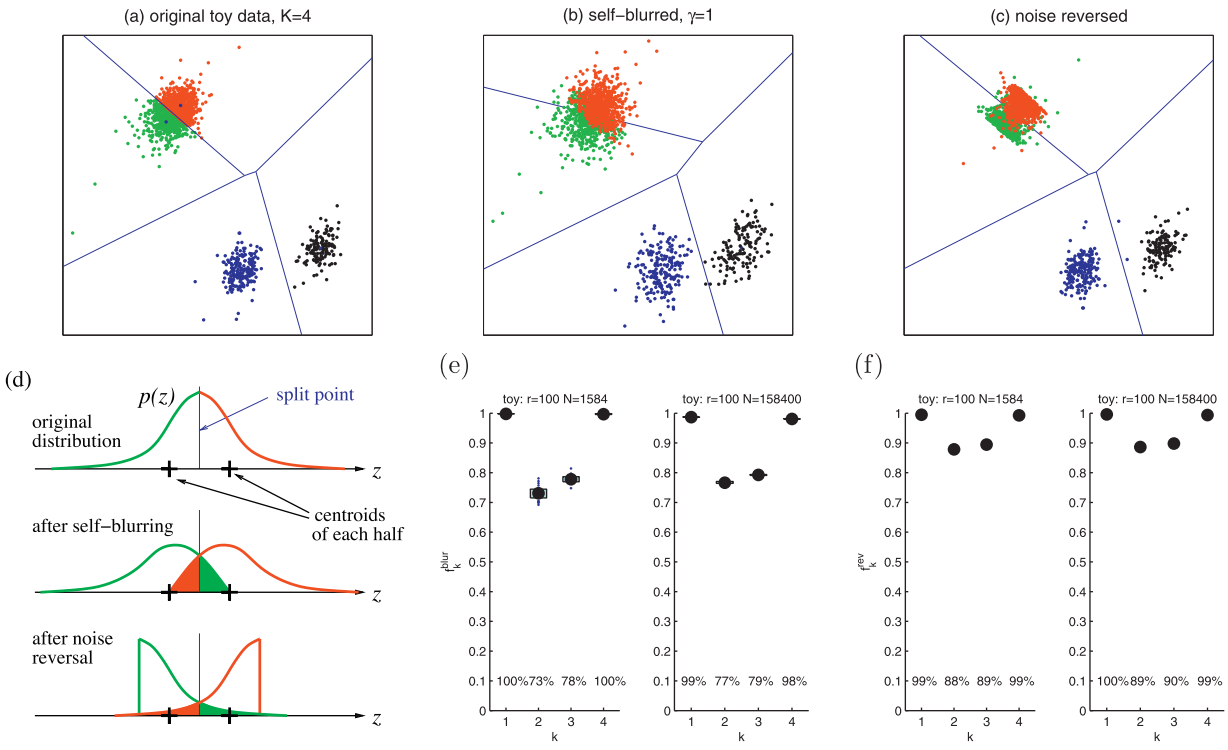

Fig. 5. Data perturbation validation metrics illustrated on the "toy" clips (from labels $k \leq 3$), sorted with PCA/k-means++ with $K = 4$, $r = 100$, $N_{\text{fea}} = 2$. (a)–(c) Shows the 2D PCA feature space. (a) Original points labeled by color as in Fig. 4(b), showing k-means centroids and decision boundaries. (b) Points after self-blurring and (c) After noise-reversal; colors show the *original* labelings, to highlight movement, and new decision boundaries. (d) Idealized 1D picture: only the shaded parts of the distribution change labeling. Stability metrics are shown for (e) self-blurring and (f) noise-reversal, for the original toy data size N, and a 100 times larger N. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

into two clusters, in the top two pictures in Fig. 5(d). The decision boundary is unchanged by self-blurring. The mass of each half of the distribution that changes label is shaded; this controls the size of $1 - f_1^{\text{blur}}$. If this distribution were a Gaussian, and $\gamma = 1$, we can solve analytically (see Appendix B) that $f_1^{\text{blur}} = 1 - [\text{erf}((2\pi)^{-1/2})]^2 \approx 0.817$. The same holds for a multivariate Gaussian in signal space split along a plane of symmetry. The reported stabilities in Fig. 5(e) also are similar to this value, even though clusters are not expected to be Gaussian (Fee et al., 1996).

Later we will vary γ: a smaller γ tends to increase all reported stabilities, but also causes less inflation of clusters (i.e. less increase in noise level), which may better reflect an algorithm's stability for close, but distinct, clusters.

2.4.2. Noise-reversal

Our second perturbation method has a similar flavor: using the above idea of noise (deviation) relative to a cluster centroid, we simply negate the noise. That is, we replace (8) by

$$\tilde{x}_{mtj} = 2W_{mt}^{(k_j)} - x_{mtj}, \quad m = 1, \ldots, M, \quad t = 1, \ldots, T, \quad j = 1, \ldots, N, \tag{9}$$

and proceed exactly as above to get f_k^{rev}. Since this is a deterministic procedure, there is no need to resample; only two spike sorter runs are needed (the original and the noise-reversed). This is shown in Fig. 5(c), and lower picture of (d): each cluster is inverted about its centroid. Label exchange occurs in a split cluster because the tails of the distributions are made to point inwards, and bleed across the boundary. The results for this toy dataset appear in Fig. 5(e): the erroneously-split labeling induces around 89% stability for both the original N and a 100 times larger N. This is more stable than both self-blurring and the true accuracy—a disadvantage—however its advantages are that the clusters are not artificially enlarged, and that there are no free parameters and no resampling.

In the idealized 1D Gaussian case, and the multivariate Gaussian split along a symmetry plane, analytically (see Appendix B) we get $f_1^{\text{rev}} = \text{erf}(2/\sqrt{\pi}) \approx 0.889$, which is very close to the observed values, and explains why stabilities are expected to be closer to 1 than for self-blurring with $\gamma = 1$.

2.4.3. Perturbation metric tests for the default algorithm

We return to the full clips data from Section 2.1 with PCA/k-means++ with the default $N_{\text{fea}} = 10$ and $r = 100$. Fig. 6(a) compares the self-blurring and noise-reversal metrics for $K = 8$: we see that noise-reversal has all stabilities roughly 3 times closer to unity than self-blurring at $\gamma = 1$, but otherwise similar relative stabilities vs neuron label k.

Since we cannot presume that $K = 8$ is optimal for extracting the largest number of stable neurons, in Fig. 6(b) and (c) we show the mean stabilities varying $K = 2, \ldots, 12$. Both show that the first four neurons are stable for most values $K \geq 4$, and show that one or two others are also somewhat stable for $K \geq 8$ (these are neurons $k = 7, 8$ for $K \geq 10$). Notice that, for noise-reversal, no stabilities drop below 0.75, i.e. the stability range is compressed towards unity.

We now focus on the case $K = 10$, where neuron $k = 5$ shows a self-blurring stability close to zero. Fig. 6(d) shows the mean waveforms for this K; indeed $k = 4, 5$ have very similar waveforms so would probably be deemed erroneously split in practice. The (best-permuted) confusion matrix $\hat{Q}$ is a useful diagnostic: as the dark vertical "domino" in Fig. 6(e) shows, $k = 4$ and 5 have been merged by self-blurring. Off-diagonal entries such as $\hat{Q}_{6,9}$ show that $k = 6, 9$ are prone to mixing, which is consistent with their visually similar waveforms. The point is that our metrics quantify such judgments, without a human having to examine whether waveforms

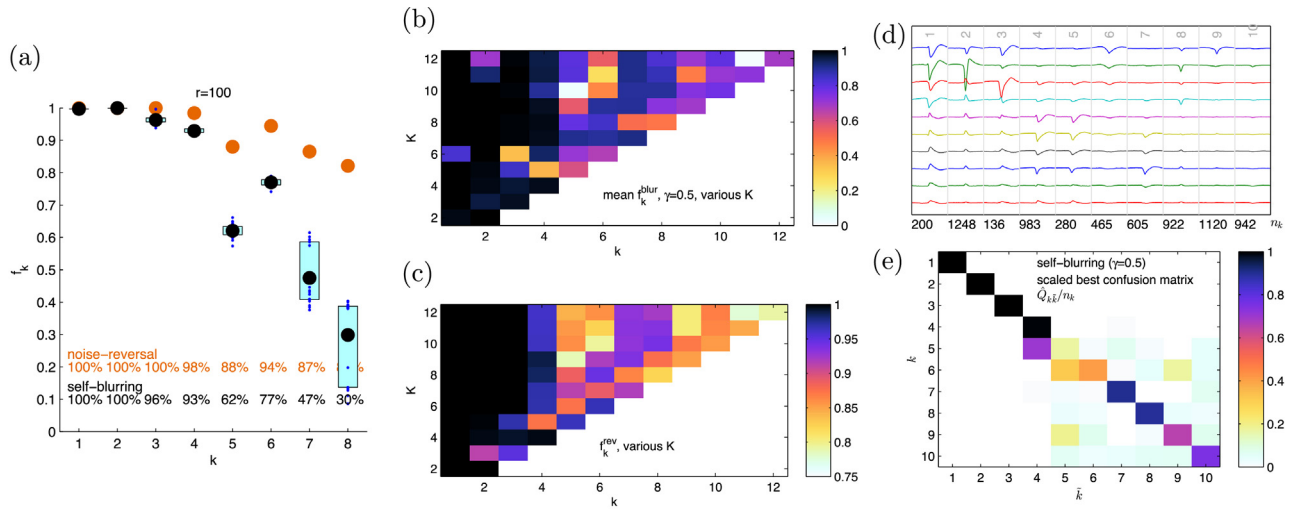

Fig. 6. Perturbation metrics for default clips dataset with PCA/k-means++ with $r = 100$. (a) Self-blurring ($\gamma = 1$, shown in black) and noise-reversal (shown in orange) metrics for $K = 8$. (b) Self-blurring ($\gamma = 0.5$) metric $\tilde{f}^{\text{blur}}_k$ for various $K = 2, \ldots, 12$ (each row a different K). (c) Same for noise-reversal metric f^{rev}_k. (d) Mean waveforms for $K = 10$, and (e) self-blurring scaled best confusion matrix $\hat{Q}_{k,\tilde{k}}/n_k$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

are "close" given the particular noise (and firing variation) of the experiment.

Finally, we observe that for $K = 4$, all supposed neurons are very stable (under both metrics), even though, with our knowledge of the larger number of distinct waveforms that can be extracted at greater K, it is obviously lacking in accuracy. This illustrates a general point that *stability is necessary but not sufficient for accuracy*.

Remark 2. Although we illustrated the self-blurring and noise-reversal metrics with a label assignment (clustering) scheme involving hard decision boundaries (hyperplanes for k-means), we propose a broader view: since both metrics involve perturbing the clip data in a manner consistent with the noise, any reproducible neural unit output by a spike sorting algorithm should be stable under these metrics.

3. Time-series based validation schemes

We now turn to the second, more general, interface of Section 1.2, where the spike sorter extracts both times t_j and labels k_j from a time series Y of multi-channel recorded data. To compare two outputs of such a sorter, say $\{t_j, k_j\}_{j=1}^{N_s}$ and $\{t'_j, k'_j\}_{j=1}^{N'_s}$, where $k_j \in \{1, \ldots, K\}$, $k'_j \in \{1, \ldots, K'\}$, and K and K' are the two (not necessarily equal) numbers of types of neurons found, one must match spikes from the first run to those in the second. A simple criterion is that matched firing times be within ϵ of each other; we fix $\epsilon = 0.5$ ms in our tests. Specifically, let a matching μ be a one-to-one map[6] from a subset of $\{1, \ldots, N_s\}$ to an (equal-sized) subset of $\{1, \ldots, N'_s\}$ such that

$$|t_j - t'_{\mu(j)}| \le \epsilon, \quad \text{for all } j \text{ matched in the map.} \tag{10}$$

Given any such μ, the confusion matrix is defined by

$$Q^{(\mu)}_{k,k'} = \#\{j : \mu(j) \text{exists}, k_j = k, \quad k'_{\mu(j)} = k'\}, \quad \text{for} \quad 1 \le k \le K, \quad 1 \le k' \le K'. \tag{11}$$

Since there may be unmatched spikes j in the first run (for which we write $\mu(j) = \emptyset$), and unmatched spikes j' in the second run (for which we write $\mu^{-1}(j') = \emptyset$), a row and column is appended for these cases to $Q^{(\mu)}$, i.e.,

$$Q^{(\mu)}_{k,\emptyset} = \#\{j : \mu(j) = \emptyset, k_j = k\}, \quad \text{for} \quad 1 \le k \le K, \tag{12}$$

$$Q^{(\mu)}_{\emptyset,k'} = \#\{j : \mu^{-1}(j') = \emptyset, k'_{j'} = k'\}, \quad \text{for} \quad 1 \le k' \le K', \tag{13}$$

and set $Q^{(\mu)}_{\emptyset,\emptyset} = 0$. The $(K+1)$-by-$(K'+1)$ matrix $Q^{(\mu)}$ defined by (11)–(13) we call the *extended confusion matrix*; see Fig. 7(a).

Since a sorting algorithm may arbitrarily permute the neuron labels, as before we need to find the "best" confusion matrix. However, now one must search over permutations π of the second label *and* over matchings μ, i.e. find

$$\hat{Q}_{k,k'} := Q^{(\hat{\mu})}_{k,\hat{\pi}(k')}, \quad \text{where} \quad \{\hat{\mu}, \hat{\pi}\} = \underset{\mu \in \mathcal{M}, \pi \in S_{K'}}{\arg\max} \sum_{k=1}^{\min(K,K')} Q^{(\mu)}_{k,\pi(k)}, \tag{14}$$

where $\mathcal{M}$ is the set of all maps μ satisfying (10). In practice we find that a simple three-pass algorithm performs well at typical firing rates (optimizing π using a greedy-in-time-difference algorithm for matching, then with π fixed, optimizing μ via two passes of greedy matching). We do not know the complexity of finding the exact solution.

3.1. A simple time-series spike sorting algorithm

As an illustrative spike sorter we will use the following model-based fitting algorithm which handles overlapping spikes (the full description we leave for a future publication). We assume that the time series Y has been low-pass filtered. Firstly, upsampled clips are extracted from Y using the triggering and upsampling procedure of Appendix A.1, with a $-100\,\mu$V threshold. Secondly, these clips are fed into the clip-based spike sorter of Section 2.1, using the best of $r = 100$ repetitions of k-means++, with a preset number of clusters K. Thirdly, a set of upsampled mean waveform shapes $\{W^{(k)}\}_{k=1}^{K}$ is found by applying (5) to the resulting clips and labels. These waveforms are ordered by descending l_2-norm. Finally, these waveforms are matched to the complete time series Y via a greedy fitting algorithm (Prentice et al., 2011; Pillow et al., 2013). Specifically, the change in l_2-norm of the residual due to subtraction of each (downsampled, fixed-amplitude) waveform at each time-shift was computed, and firing events t_j were declared at sufficiently negative local minima (wth respect to time shift) of this function,

[6] μ can also be expressed as a subgraph of the complete bipartite graph between N_s nodes and N'_s nodes, whose vertex degrees are at most one.

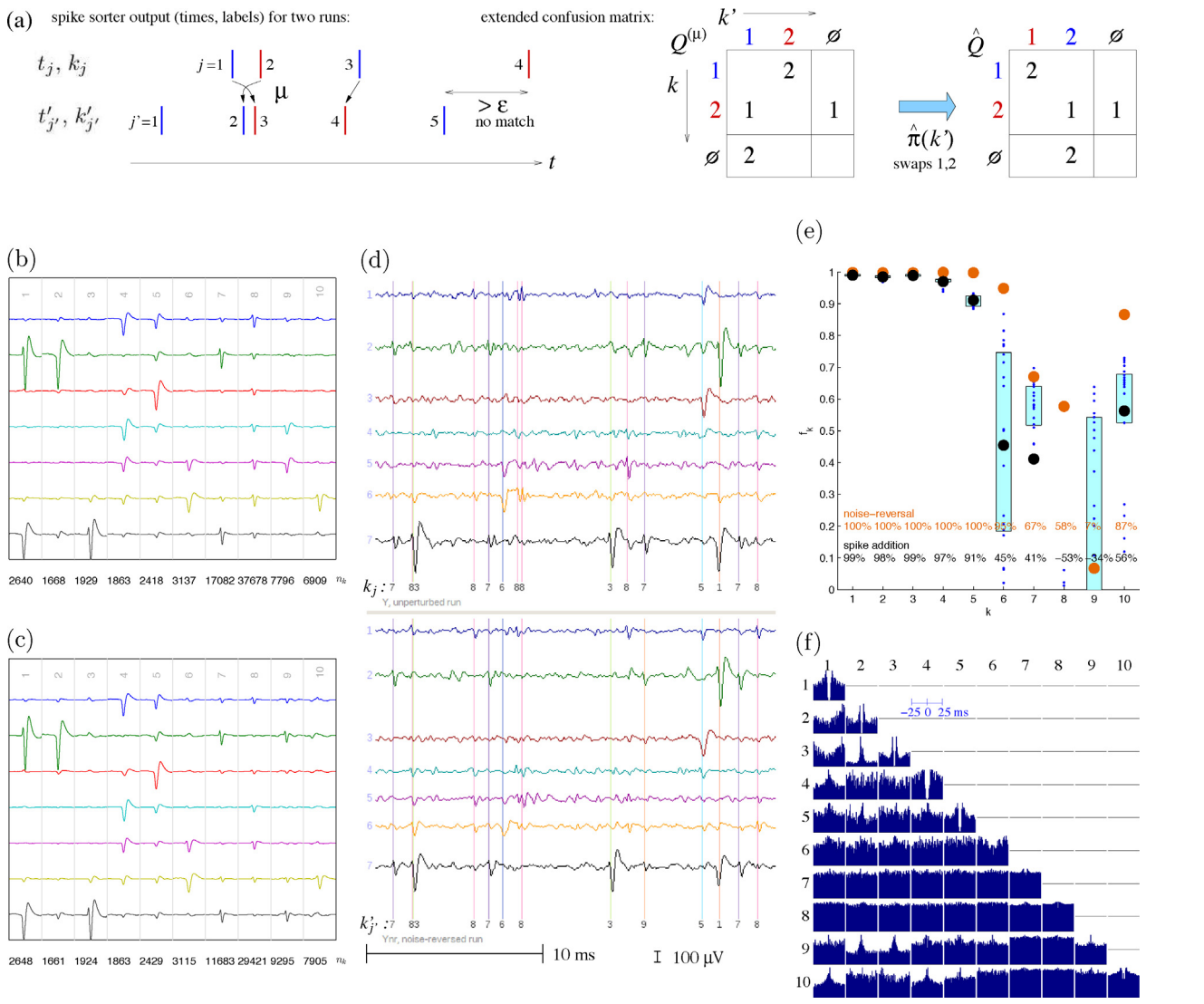

Fig. 7. Time-series stability metric results. (a) Toy example ($K=2$) of a matching μ between one spike sorter output t_j, k_j and another $t'_{j'}$, $k'_{j'}$ (blue shows label $k=1$ and red label $k=2$). The resulting extended confusion matrix and its "best" permutation $\hat{Q}$ are shown. The matching μ shown is the best matching $\hat{\mu}$ even though the nearby pair is swapped in time. (b)–(f) Show results for the spike sorter of Section 3.1 with $K=10$. (b) Waveform shapes $\{V^{(k)}\}_{k=1}^{K}$ and populations from the unperturbed run, and (c) from the noise-reversed run. (d) Input time series data Y, and noise-reversed time series $\tilde{Y}$, showing firing times and (best-permuted) labels from each run. (e) Stability metrics f_k for noise-reversal (shown in orange), and for the spike addition metric (black). (f) Spike firing cross-correlations of spike sorter output t_j, k_j. Refractory dips (of at least ±3 ms) are complete for $k=1,\ldots,5$ and partial for $k=6$, 10, matching the metrics in (e). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

that were also global minima over all waveforms with time shifts up to ±0.5 ms from t_j. The corresponding k_j was the label of the waveform responsible for this largest decrease in residual. This greedy sweep through the time series was repeated until no further change occurred, allowing closely overlapping spikes to be resolved. Note that, as with many modern algorithms (Takahashi et al., 2002; Franke et al., 2010, 2015a; Ekanadham et al., 2013; Moore-Kochlacs et al., 2014; Pillow et al., 2013) the main fitting stage involves no concept of "clips", nor event windows, nor feature space.

As in Section 2.2, a simple rerunning validation metric would be possible for time series. However, we present the following two metrics as much more realistic indicators of stability.

3.2. Noise-reversal

The noise-reversal idea of Section 2.4.2 generalizes to time series, handling overlapping spikes naturally, as follows. Firstly, the sorter runs on the unperturbed data to give $S(Y) = \{t_j, k_j\}_{j=1}^{N_s}$. Since the interface does not output waveforms, they are next estimated by averaging[7] windows from Y centered on the output firing times t_j for the appropriate label, e.g.

$$V_{mt}^{(k)} = \frac{1}{n_k} \sum_{j:k_j=k} y_{m,t_j+t-T/2}, \quad m = 1,\ldots,M, \quad t = 1,\ldots,T. \tag{15}$$

The "forward model" generating a (noise-free) time series from a new set of firing times $\mathbf{s} := \{s_j\}_{j=1}^{N}$ and labels $\mathbf{l} := \{l_j\}_{j=1}^{N}$ is then $F(\mathbf{s},\mathbf{l})$, with entries

$$F_{mt} = F(\mathbf{s},\mathbf{l})_{mt} := \sum_{j=1}^{N} V_{m,t-s_j+T/2}^{(l_j)}, \quad m = 1,\ldots,M, \quad t = 1,\ldots,T_{\text{tot}}. \tag{16}$$

[7] In the case of high firing rates, a least-squares solution of a large linear system may provide an improved estimation (Pillow et al., 2013; Ekanadham et al., 2013).

(We in fact use variants of (15) and (16) which achieve sub-sample firing time accuracy by upsampling the waveforms $V^{(k)}$ as in Appendix A.1, and building downsampling into F.) Using this, a perturbed time series (analogous to (9)) is generated,

$$\tilde{Y} = 2F(\mathbf{t}, \mathbf{k}) - Y, \tag{17}$$

and sorted to give $S(\tilde{Y}) = \{t'_{j'}, k'_{j'}\}_{j'=1}^{N'_s}$. Finally, the best extended confusion matrix $\hat{Q}$ between the two sets of outputs is found as in (11)–(14), and the usual stability metric f_k computed via (7).

Fig. 7 illustrates this metric for spike sorting filtered time series data Y taken from an *ex vivo* retinal multi-electrode array recording described in Appendix A.3, with $K = 10$ neurons assumed. This dataset appears to have a large fraction of overlapping events: spike sorting as in Section 3.1 results in around 10% of firing events falling within 0.1 ms of another event. The upper half of Fig. 7(d) shows a short time window from Y, while the lower half shows the resulting $\tilde{Y}$. Note that some of the sorted times and labels have changed, while between firing events the signal is *negated*. The orange dots in Fig. 7(e) show the resulting f_k^{rev} values: the five neurons with largest waveform norms are quite stable, while $k = 6$ and 10 are marginally stable (recalling from Section 2.4.2 that 90% is hardly a stable noise-reversal metric), and $k = 7, 8, 9$ are unstable and—judging by their huge n_k—massively overfit. This matches well the per-neuron physiological validation based on clarity of refractory dips in the firing time auto-correlations in Fig. 7(f).

It is worth examining why $k = 9$ has almost zero stability. The cause is apparent in panels (b) and (c), which show the waveforms $V^{(k)}$ estimated via (15) from the unperturbed and noise-reversed runs (the latter in "best permuted" ordering $\hat{\pi}$). The two sets of waveforms are very similar apart from the possibly distinct neuron $k = 9$ in (b) which has disappeared in (c) due to the splitting of $k = 7$. We find that such clustering instabilities are common, and vary even with rerunning the sorter with a different random seed in k-means++. Our metric quantifies these instabilities as induced by variations consistent with any even-symmetric noise model.

3.3. Auxiliary spike addition

The final stability metric we present perturbs the time series by linearly adding new spikes at known random firing times and then measuring their accuracy upon sorting; this gauges performance in the context of overlapping spikes—by creating and assessing new overlaps—while respecting the linearity of the physical model for extracellular signals (Buzsáki et al., 2012).

Let the first unperturbed sorter run produce times and labels $S(Y) = \{t_j, k_j\}_{j=1}^{N_s}$, from which a forward model is built, as above via (15)–(16). An estimate for the firing rate of the kth neuron is n_k/T_{tot}, where n_k is given by (6). One generates an auxiliary set of events $\{s_j, l_j\}_{j=1}^{N_a}$ as the union of K independent random Poissonian spike trains, with rates $\beta n_k/T_{\text{tot}}$, $k = 1, \ldots, K$, where β is a rate scaling parameter. If β is too small, not much is learned from the perturbation; if too large, the total firing rates are made unrealistically large. We fix $\beta = 0.25$. The perturbed time series is then

$$\tilde{Y} = Y + F(\mathbf{s}, \mathbf{l}), \tag{18}$$

which is sorted to give $S(\tilde{Y}) = \{t'_{j'}, k'_{j'}\}_{j'=1}^{N'_s}$. The best extended confusion matrix $\hat{Q}$ is now found (as in (11)–(14)) between the total of $N_s + N_a$ events $\mathbf{t} \cup \mathbf{s}$, $\mathbf{k} \cup \mathbf{l}$, and the second run outputs $\mathbf{t}'$, $\mathbf{k}'$. To assess the *change* in $\hat{Q}$ induced by adding spikes, we compute the matrix

$$\hat{Q}^{\text{add}} := \hat{Q} - \text{diag}\{n_1, \ldots, n_K, 0\}$$

where diag denotes the diagonal matrix with diagonal elements as listed. Finally, f_k^{add} is computed as in (7) using $\hat{Q}^{\text{add}}$. One may interpret this diagonal subtraction as follows: if all the new spikes were correctly sorted, and all the original spikes sorted as in the first run, $\hat{Q}^{\text{add}}$ would be diagonal with entries given by the added populations for each label, so $f_k^{\text{add}} = 1$ for all k. Thus low f_k^{add} values indicate either lack of sorting accuracy for the added spikes, or induced instability (e.g. false positives or negatives) in re-sorting the original spikes. Either of these is a warning that a putative neuron is less believable.

Remark 3. Marre et al. (2012) and Rossant et al. (2015) use similar but distinct validation metrics: they add spikes with known firing times from either a spatially translated neuron, or a "donor" neuron from a different recording, then assess accuracy for this neuron. While useful as overall measures of reliability, these methods require dataset-specific adaptations, and it is not clear that they validate any of the particular neurons *in the original data*. In contrast, the metric we propose here validates each neuron as sorted in the context of the dataset of interest (including situations with overlapping spikes). Our scheme is reminiscent of simulation-based overlapping spike validation (Prentice et al., 2011; Franke et al., 2015a), but is performed "on the fly" and through the standardized sorter interface alone.

Fig. 7(e) shows 20 independent samples of f_k^{add} (i.e., 20 realizations of the Poisson spike trains), with the black dots showing the means. Only the first five neurons have stabilities above 60%. The overall pattern is similar to noise-reversal, but with a lower overall calibration of the metric, and it matches well the independent validation via cross-correlations in Fig. 7(f). Note that if adding a certain type of spike induces changes in the sorting of the (more numerous) existing spikes, then huge drops in stability, including highly negative f_k^{add}, are possible, as visible in Fig. 7(e), $k = 8, 9$. Indeed, from its waveform and huge population in Fig. 7(b), $k = 8$ would likely be classified as the "noise cluster" (see e.g. (Wild et al., 2012)) by an experienced human.

3.4. Stability and ground-truth accuracy in simulated data

It is natural to ask: what do the above stability metrics convey about sorting accuracy? To assess this, we synthesize time series data with known ground-truth firing times and labels, and from these compare per-neuron stability and accuracy. Specifically, we fix the $K = 10$ mean waveforms in Fig. 7(b), use the forward model (16) (again with up- and down-sampling for sub-sample precision in firing times), Poissonian firing statistics with mean rates as in Fig. 7(b), with each firing event scaled in amplitude by an independent Gaussian variable of mean 1 and standard deviation 0.2. Finally, we add Gaussian white noise of standard deviation $\eta = 20\,\mu$V. Our goal is to provide noise and variation comparable to those in recordings; obviously more detailed simulations are possible (Camuñas-Mesa and Quiroga, 2013; Hagen et al., 2015).

Accuracy a_k for a given synthesized time series Y ("realization") is defined to be f_k as in (7), using $\hat{Q}$ as the best confusion matrix between the ground-truth labeling and the sorter output. We use the sorter in Section 3.1 with the correct $K = 10$, and find that accuracy of some labels k varies wildly between sortings, even for fixed data Y, because the sorter is non-deterministic (e.g. see $k = 9$ in Fig. 8(a) and (b), for which the distribution of a_9 clusters at 0 and 1). There is also variation due to the choice of realization Y. Thus the meaningful quantity to study is the *mean accuracy* $\bar{a}_k$, which we estimate using 100 realizations. Likewise, we define $\bar{f}_k$ as the mean of the noise-reversal metric (using 50 realizations), or of the spike addition stability metric (using 10 realizations). For per-neuron comparisons, we must permute all labels to match the ground-truth neuron k using the best confusion matrices.

The results are shown in Fig. 8(a) and (b): both stability metrics have a strong correlation with mean accuracy, especially so for spike addition. The first five neurons have stabilities and accuracies

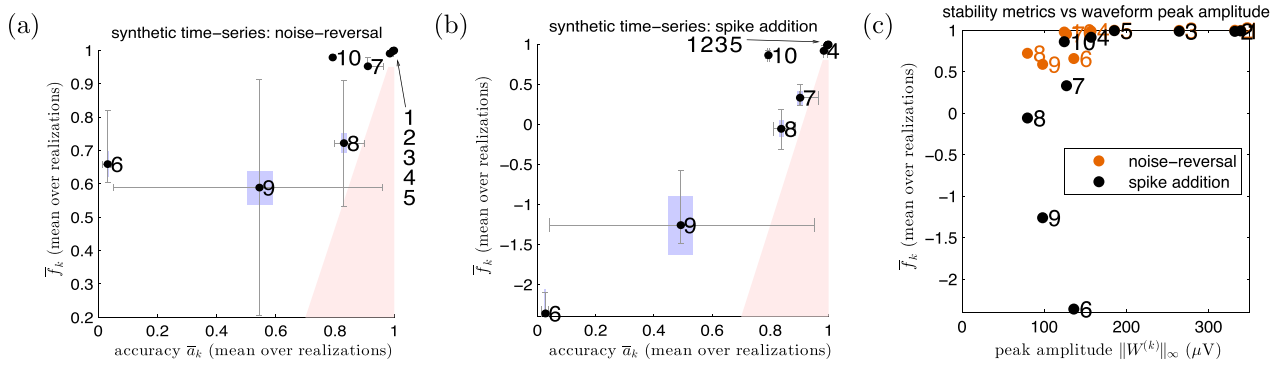

Fig. 8. Mean results over an ensemble of synthetic time series with Poissonian firing, amplitude variation, and additive noise. In all three panels, numbers indicate neuron label k. (a) Noise-reversal metric vs accuracy. (b) Spike addition metric vs accuracy. Blue rectangles show standard errors in the estimates of the means $\bar{a}_k$ and $\bar{f}_k$, the grey errorbars the 25% to 75% quantile range in the samples of a_k and f_k, and the pink triangle an "accurate but unstable" region. (c) Stability metrics vs peak absolute waveform amplitude. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Summary of the proposed validation metrics for clip-based and time series spike sorting algorithms.

Interface	Metric	Section	Advantages	Disadvantages
Clip based	Rerun	Section 2.2	Simplicity	Useless if sorter is deterministic
	3-way CV	Section 2.3	Simplicity	Classifier needed; not useful for large N
	Self-blurring	Section 2.4.1	Adjustable parameter γ	Raises noise level (can underestimate accuracy)
	Noise-reversal	Section 2.4.2	Only needs 2 runs; no extra noise	Stability range close to 1
Time series	Noise-reversal	Section 3.2	Only needs 2 runs; no extra noise	Stability range close to 1; reverses noise polarity
	Spike addition	Section 3.3	Accuracy in context; no extra noise	Firing rate slightly higher; no built-in shape variation

close to 100%. The other five neurons illustrate the idea that an unstable neuron cannot be accurate: they avoid the "accurate but unstable" region (shown in pink). Note that neuron $k = 10$ is quite stable under either metric, but is only 80% accurate, illustrating that stability is only an (approximate) *upper* bound on mean accuracy.

Finally, we address the issue of whether our stability metrics could be replaced with simpler measures. One such heuristic measure that is compatible with the black-box sorter interface is the peak amplitude of the resulting mean waveforms $W^{(k)}$. To investigate this, Fig. 8(c) plots the noise-reversal and spike addition metrics vs the peak amplitude of $W^{(k)}$, for the synthetic time series. Above 180 μV, amplitude is predictive of high stability (and accuracy), but below this stability (and accuracy) appears essentially uncorrelated with amplitude. Thus, amplitude is not a very useful validation metric. We do not compare against metrics such as Fisher discriminants (Pouzat et al., 2002; Hill et al., 2011), since, as discussed in the introduction, they require access to internal algorithm-dependent variables such as feature space.

4. Discussion and conclusions

We have proposed several schemes for the validation of automatic spike sorting algorithms, each of which outputs a per-neuron *stability metric* in the context of the dataset of interest, without ground truth information or physiological criteria, and with few assumptions about the noise model. Our key contribution is a common framework (Fig. 1) where this validation occurs only through a simple standardized interface to the sorter, which is treated as a *black box* that can be invoked multiple times. Indeed, since many automatic sorting algorithms are stochastic, such a metric cannot be defined without averaging over sorting runs. We see this as essential to automated validation and benchmarking of a growing variety of spike sorting algorithms and codes. By contrast, most existing validation metrics rely on accessing internal parameters (e.g. feature spaces) peculiar to certain algorithms. We also envisage such metrics as the first filter in a laboratory pipeline in which—due to the increasing scale of multi-electrode recordings—it will be impossible for a human operator to make decisions about most neurons. While stability cannot guarantee accuracy or single-unit activity (e.g. see Sections 2.4.3 and 3.4), instability is a useful warning of probable inaccuracy or multi-unit activity. Only neurons that pass the stability test should then be further validated based on e.g. shape (Tankus et al., 2009), refractory gaps (Hill et al., 2011) and/or receptive fields (Litke et al., 2004; Marre et al., 2012; Pillow et al., 2013).

Table 1 summarizes the six schemes: the first four use a simple sorting interface that classifies "clips" (spiking events), while the remaining two use a more general interface that accesses the full time series and allows handling of overlapping spikes. All schemes resample or perturb the data in a realistic way, then measure how close the resulting neuron-to-neuron *best confusion matrix* $\hat{Q}$ is to being diagonal; its off-diagonal structure also indicates which neurons are coupled by instabilities. Although our demonstrations used standard sorting algorithms with a specified number of neurons K, the metrics would also apply to variable K with minor extra bookkeeping. The first two schemes (rerunning and cross-validation) are conceptually simple but are shown to have severe limitations. The next two (self-blurring and noise-reversal for clip sorting) seem useful even in large-scale settings. However, we expect that the last two (noise-reversal and spike addition for time series) will be the most useful; unlike metrics relying on cluster distributions in feature space (Pouzat et al., 2002; Hill et al., 2011), these two metrics also naturally handle overlapping spikes. Each metric carries its own *calibration*: for instance, fixing the data and sorter, for noise-reversal f_k is much closer to 1 than for spike addition. We used simulated time series to show that these two metrics correlate well with *mean* ground-truth accuracy. We emphasize that, although a particular sorting run may by chance be accurate, mean quantities are the correct measures. A future task is to study this with more realistic noise and correlated firing models, and for intracellular ground-truth recordings.

Although very general, our schemes make certain assumptions. Noise-reversal assumes sign-symmetry in the noise distribution

(which may not be satisfied for "noise" due to distant spikes or during bursting), whereas spike addition as presented does not account for realistic firing variation in the added spikes; the latter could be achieved in a model-free way by adding random *spike instances* (as in self-blurring) instead of mean waveforms.

Several other extensions are worth exploring. (1) The last two validation schemes essentially involve reverse-engineering a forward model (16) from a single spike sorter run. This model could be improved in various ways, by making use of firing amplitude information (if provided by the sorter), or, in nonstationary settings (waveform drift), constructing waveform shapes from only a local moving window. (2) The random known times in spike addition could be chosen to respect the refractory periods of the existing spikes, avoiding rare but unrealistic self-overlaps. Added spikes could also be chosen in a bursting sequence that tests the sorter's accuracy in this difficult setting. (3) A variant of self-blurring that estimates $\partial f_k^{\mathrm{blur}}/\partial\gamma|_{\gamma=0}$ could avoid the problem of noise level increase (cluster inflation) at larger γ. (4) Extending the metrics from per-neuron to per-firing-event would allow laboratories to filter out unstable events without discarding an entire neuron.

Documented code which implements the validation metrics and sorting algorithms used is freely available (in MATLAB with a MEX interface to C/OpenMP) at the following URL: https://github.com/ahbarnett/validspike.

Acknowledgments

We are very grateful for the Chichilnisky and Buzsáki labs for supplying us with test datasets. We have benefited from comments by the anonymous reviewers, and from many helpful discussions with EJ Chichilnisky, György Buzsáki, Adrien Peyrache, Brendon Watson, Bin Yu, Eero Simoncelli, Mitya Chklovskii, and Eftychios Pnevmatikakis.

Appendix A. Provenance of test datasets

A.1. Clips dataset

We started with an electrical recording, supplied to us by the Buzsáki Lab at NYU Medical Center, taken from the motor cortex of a freely behaving rat at 20 kHz sampling rate with a NeuroNexus "Buzsaki64sp" probe (Watson and Buzsáki, 2015). We took 3×10^6 consecutive time samples from a single shank with $M=10$ channels. The following procedure to extract clips combines standard techniques in the literature.

We first high-pass filtered at 300 Hz, using the FFT and a frequency filter function $a(f)=(1+\tanh[(f-300)/100])/2$, where $f\geq 0$ is the frequency in Hz. This gives a M-by-3×10^6 matrix Y, which was then spatially prewhitened by replacing by QY, where Q is R^{-1} but with each row normalized to unit l_2 norm. $R^TR=C$ is the Cholesky factorization of C, an estimate of the $M\times M$ channel-wise cross-correlation matrix. C itself was estimated using 10^4 randomly-chosen "noise clips" (time-series segments of duration 4 ms for which no channel exceeded a threshold of 158 μV). This prewhitening significantly improved the noise level, and helped remove common-mode events. Events were triggered by the minimum across channels passing below −120 μV (around 4 times the estimated noise standard deviation); events where this trigger lasted longer than 1 ms, or where another trigger occurred within 2 ms either side of the triggering event, were discarded as unlikely to be due to a single spiking event. This gave 6901 clips of length 60 samples each.

Clips were then upsampled by a factor of 3, using interpolation from the sampling grid by the Hann-windowed sinc kernel (Blanche and Swindale, 2006) of width $\tau=5$ samples,

$$f(t)=\begin{cases}1, & t=0,\\ 0, & |t|>\tau,\\ \dfrac{\sin(\pi t)}{\pi t}\cos^2\!\left(\dfrac{\pi t}{2\tau}\right), & \text{otherwise},\end{cases}$$

where t is given in the original sample units. The kernel width reduced the clip duration from 3 ms to 2.45 ms. Finally, the upsampled clips were aligned by time translation by an integer number of upsampled grid points until the minimum across channels lies at the central grid point. When translated, values either side of the data were padded with zeros (on average less than one padded zero per clip per channel was needed). The upsampling allows alignment to be more accurate and improves cluster quality (Blanche and Swindale, 2006). The result was a M-by-T-by-N array of clips X with $M=10$, $T=147$, and $N=6901$.

A.2. Larger clips dataset

In Sections 2.3 and 2.4 datasets comprising a larger number of clips N were needed. Since we did not have stationary time series containing such large numbers of clips, we generated them from the above clips as follows. Let F be the factor to grow the number of clips. Each clip was duplicated F times, then to each channel of each new clip was added time-correlated Gaussian noise with variance η^2, where $\eta=30$ μV is an estimate of the noise standard deviation from pre-whitened "noise clips", as above. The autocorrelation function in time was taken as $C(t)=e^{-t/\tau}$, with $\tau=0.5$ ms, giving a rough approximation to the observed noise autocorrelation. As is standard, the time-correlated noise samples were generated by applying the Cholesky factor of the time-sample autocorrelation matrix to an iid Gaussian signal of the desired length.

A.3. Time-series dataset

Our time-series data Y comes from $M=7$ adjacent electrodes (spatially forming a hexagon around a central electrode) within a 512-electrode array (Litke et al., 2004) recording at a 20 kHz sample rate from the spontaneous activation of an *ex vivo* monkey retina. The recording length was 2 minutes ($T_{\mathrm{tot}}=2.4\times10^6$ samples), and mean neuron spiking rates are of order 20 Hz. This was supplied to us by the Chichilnisky Lab at Stanford. The raw data was then high-pass filtered at 300 Hz as in Appendix A.1. Since there was little noise correlation between channels, no spatial prewhitening was done.

Appendix B. Analytic stability for a Gaussian erroneously split in two

When the N clips are drawn from some underlying probability distribution function (pdf) p in signal or feature space, we have analytic results as $N\to\infty$ for the stability of a single cluster when split into two symmetrically by a decision hyperplane (e.g. by k-means with $K=2$). Consider as in Fig. 5(d), the 1D case $z\in\mathbb{R}$, with $p(z)$ symmetric about the splitting point $z=0$ (without loss of generality). The centroid of the positive half of the pdf is $c=2\int_0^\infty zp(z)dz$. The stability (of either of the two labels) is the mass that does not change label, which, specializing to $\gamma=1$, is the chance that the sum of two independent positive samples from p exceeds c,

$$f_1^{\mathrm{blur}}=1-4\int_0^c\int_0^{c-y}p(x)p(y)dx\,dy. \tag{19}$$

Choosing a Gaussian $p(z) = (1/\sqrt{\pi})e^{-z^2}$ for which $c = 1/\sqrt{\pi}$, the second term in (19) is the mass in a diamond region which, by the rotational invariance of $p(x)p(y)$, may be rotated by $\pi/4$ to become the square $|x|, |y| < c/\sqrt{2}$. Thus

$$f_1^{\mathrm{blur}} = 1 - \left[2\int_0^{c/\sqrt{2}} p(x)dx\right]^2 = 1 - [\mathrm{erf}(1/\sqrt{2\pi})]^2 = 0.817\ldots,$$

where the usual error function is defined in Olver et al. (2010; 7.2.1).

For noise-reversal, stability is simply the probability that a single sample has $|z| < 2c$,

$$f_1^{\mathrm{rev}} = \mathrm{erf}(2c) = \mathrm{erf}(2/\sqrt{\pi}) \approx 0.889\ldots$$

Both of these cases apply to a multivariate Gaussian split on a symmetry hyperplane, by taking z as the coordinate normal to the hyperplane.

References

Arthur D, Vassilvitskii S. k-means++: the advantages of careful seeding. In: Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms; 2007.

Berdondini L, Imfeld K, Maccione A, Tedesco M, Neukom S, Koudelka-Hep M, Martinoia S. Active pixel sensor array for high spatio-temporal resolution electrophysiological recordings from single cell to large scale neuronal networks. Lab Chip 2009;9:2644–51.

Blanche TJ, Swindale NV. Nyquist interpolation improves neuron yield in multiunit recordings. J. Neurosci. Methods 2006;155:81–91.

Buzsáki G, Anastassiou CA, Koch C. The origin of extracellular fields and currents – EEG, ECoG, LFP and spikes. Nat. Rev. Neurosci 2012;13:407–20.

Camuñas-Mesa LA, Quiroga RQ. A detailed and fast model of extracellular recordings. Neural Comput 2013;25:1191–212.

Carlson DE, Vogelstein JT, Wu Q, Lian W, Zhou M, Stoetzner CR, Kipke D, Weber D, Dunson DB, Carin L. Multichannel electrophysiology spike sorting via joint dictionary learning & mixture modeling. IEEE Trans. Biomed. Eng 2013.

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(1):11–7.

Ekanadham C, Tranchina D, Simoncelli EP. A unified framework and method for automatic neural spike identification. J. Neurosci. Methods 2013;222:47–55.

Eversmann B, Jenkner M, Hofmann F, Paulus C, Brederlow R, Holzapfl B, Fromherz P, Merz M, Brenner M, Schreiter M, Gabl R, Plehnert K, Steinhauser M, Eckstein G, Schmitt-Landsiedel D, Thewes R. A 128 × 128 CMOS biosensor array for extracellular recording of neural activity. IEEE J. Solid-State Circuits 2003;38(12):2306–17.

Fee MS, Mitra PP, Kleinfeld D. Automatic sorting of multiple unit neuronal signals in the presence of anisotropic and non-Gaussian variability. J. Neurosci. Methods 1996;69:175–88.

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;29:127–48.

Franke F, Pröpper R, Alle H, Meier P, Geiger JR, Obermayer K, Munk MH. Spike sorting of synchronous spikes from local neuron ensembles. J. Neurophysiol 2015a;114:2535–49.

Franke F, Quiroga RQ, Hierlemann A, Obermayer K. Bayes optimal template matching for spike sorting – combining Fisher discriminant analysis with optimal filtering. J. Comput. Neurosci 2015b;38:439–59.

Gibson S, Judy JW, Marković D. Spike sorting: the first step in decoding the brain. In: IEEE Signal Proc. Mag.; 2012 January. p. 124–43.

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. Neurosci. Methods 2015;245:182–204.

Harris KD, H. D.A., Csicsvari J, Hirase H, Buzsaki G. Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. J. Neurophysiol 2000;84:401–14.

Hennig C. Cluster-wise assessment of cluster stability. Comput. Stat. Data An 2007;52(1):258–71.

Hill DN, Mehta SB, Kleinfeld D. Quality metrics to accompany spike sorting of extracellular signals. J. Neurosci 2011;31(24):8699–705.

Krieger AM, Green PE. A cautionary note on using internal cross validation to select the number of clusters. Psychometrika 1999;64(3):341–53.

Kuhn HW. The Hungarian method for the assignment problem. Naval Res. Logist. Quart 1955;2:83–97.

Lange T, Roth V, Braun ML, Buhmann JM. Stability-based validation of clustering solutions. Neural Comput 2004;16(6):1299–323.

Lewicki MS. A review of methods for spike sorting: the detection and classification of neural action potentials. Netw. Comput. Neural Syst 1998;9:R53–78.

Li PH, Gauthier JL, Schiff M, Sher A, Ahn D, Field GD, Greschner M, Callaway EM, Litke AM, Chichilnisky EJ. Anatomical identification of extracellularly recorded cells in large-scale multielectrode recordings. J. Neurosci 2015;35(11):4663–75.

Litke AM, Bezayiff N, Chichilnisky EJ, Cunningham W, Dabrowski W, Grillo AA, Grivich M, Grybos P, Hottowy P, Kachiguine S, Kalmar RS, Mathieson K, D. P. D., Rahman M, Sher A. What does the eye tell the brain? Development of a system for the large scale recording of retinal output activity. IEEE Trans. Nucl. Sci 2004;51(4):1434–40.

MacKay DJC. Introduction to Monte Carlo methods. In: Jordan MI, editor. Learning in Graphical Models, NATO Science Series. Kluwer Academic Press; 1998. p. 175–204.

Marre O, Amodei D, Deshmukh N, Sadeghi K, Soo F, Holy TE, M. J.B. II. Mapping a complete neural population in the retina. J. Neurosci 2012;32(43):14859–73.

Moore-Kochlacs C, Scholvin J, Kinney JP, Bernstein JG, Yoon Y, Arfin SK, Kopell N, Boyden ES. Principles of high-fidelity, high-density 3-d neural recording. BMC Neurosci 2014;15(Suppl 1):P122.

Müller J, Ballini M, Livi P, Chen Y, Radivojevic M, Shadmani A, Viswam V, Jones IL, Fiscella M, Diggelmann R, Stettler A, Frey U, Bakkuma DJ, Hierlemann A. High-resolution CMOS MEA platform to study neurons at subcellular, cellular, and network levels. Lab Chip 2015;15:2767–80.

Neymotin SA, Lytton WW, Olypher AV, Fenton AA. Measuring the quality of neuronal identification in ensemble recordings. J. Neurosci 2011;31(45):16398–409.

Olver FWJ, Lozier DW, Boisvert RF, Clark CW, editors. NIST Handbook of Mathematical Functions. Cambridge University Press; 2010. http://dlmf.nist.gov.

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(5):e62123.

Pouzat C, Mazor O, Laurent G. Using noise signature to optimize spike-sorting and to assess neuronal classification quality. J. Neurosci. Methods 2002;122:43–57.

Prentice JS, Homann J, Simmons KD, Tkačik G, Balasubramanian V, Nelson PC. Fast, scalable, Bayesian spike identification and multi-electrode arrays. PLoS ONE 2011;6(7):e19884.

Quiroga RQ. Spike sorting. Scholarpedia 2007;2(12):3583.

Quiroga RQ. Spike sorting. Curr. Biol 2012;22(2):R45–6.

Rand WM. Objective criteria for the evaluation of clustering methods. J. Am. Stat. Assoc 1971;66:846–50.

Rossant C, Kadir S, Goodman DFM, Schulman J, Belluscio M, Buzsaki G, Harris KD. Spike sorting for large, dense electrode arrays; 2015, http://dx.doi.org/10.1101/015198.

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

Shamir O, Tishby N. On the reliability of clustering stability in the large sample regime. Advances in neural information processing systems (NIPS); 2009. p. 1465–72.

Takahashi S, Sakurai Y, Tsukada M, Anzai Y. Classification of neuronal activities from tetrode recordings using independent component analysis. Neurocomputing 2002;49:289–98.

Tankus A, Yeshurun Y, Fried I. An automatic measure for classifying clusters of suspected spikes into single cells versus multiunits. J. Neural. Eng 2009;6(5):056001.

von Luxburg U. Clustering stability: an overview. Found. Trends Mach. Learn 2009;2(3):235–74.

Watson BO, Buzsáki G. Personal communication; 2015.

Wild J, Prekopcsak Z, Sieger T, Novak D, Jech R. Performance comparison of extracellular spike sorting algorithms for single-channel recordings. J. Neurosci. Methods 2012;203(2):369–76.

Wood F, Black MJ. A non-parametric Bayesian alternative to spike sorting. J. Neurosci. Methods 2008;173:1–12.

Yu B. Stability. Bernoulli 2013;19:1484–500.

Zaki MJ, Meira W Jr. Data Mining and Analysis: Fundamental Concepts and Algorithms. New York, NY: Cambridge University Press; 2014.