# Automatic post-processing and merging of multiple spike-sorting analyses with *Lussac*

Victor Llobet, Aurélien Wyngaard* and Boris Barbour†

Institut de Biologie de l'ENS (IBENS), Département de biologie, École normale supérieure, CNRS, INSERM, Université PSL, 75005 Paris, France.

February 7, 2022

## Abstract

**The increasing site counts of multielectrodes used in extracellular recording preclude traditional manual spike-sorting and have led to the development of several new software suites with increased automation. It remains difficult, however, to find parameters enabling optimal processing of signals from all neuronal types simultaneously. Here, we describe a procedure for automatically post-processing and merging the outputs of multiple spike-sorting analyses, enabling the accumulation of results optimised for different activity features. This approach was motivated by and first applied to the analysis of cerebellar activity, in which the complex spikes challenge standard analyses. The core of our modular procedure involves the identification of cluster-fusion candidates and assurance that only fusions that increase a clearly defined cluster-quality metric are performed; important subsidiary steps that have also been automated include spike alignment and duplicate removal. The procedure was validated by recovery of injected spike waveforms. In tests combining analyses from Kilosort and MountainSort, our procedure yielded a 50 % increase in the detected number of Purkinje cell clusters with both simple and complex spikes. A further validation used the synthetic *SpikeForest* dataset approximating neocortical activity, in which our package also improved performance. Combining multiple analyses therefore offers a**

[*] Victor Llobet and Aurélien Wyngaard made equivalent contributions to this work.

[†] Corresponding author: Boris Barbour, Institut de Biologie de l'ENS (IBENS), 46 rue d'Ulm, 75005 Paris, France.

**general method for improving spike-sorting, and furthermore *Lussac* reduces manual input and increases the objectivity of analyses.**

# 1 Introduction

Pioneering multielectrode recordings (McNaughton et al. 1983; Wilson and McNaughton 1993; O'Keefe and Recce 1993) were analysed using a semi-automatic method of 'cluster-cutting' (Gray et al. 1995). A small number of spike waveform parameters, such as peak amplitudes on each electrode, would be selected and detected spikes projected onto these dimensions pairwise. Clusters would be defined manually and verified using the rate of detections during the refractory period, which should be zero in the absence of any contamination.

For several years, improvement of the techniques focused on the generality of the automatic clustering, for instance taking into account amplitude variations, the statistics of firing and accommodating complex cluster shapes; this background is reviewed in Einevoll et al. (2012).

The advent of highly-integrated silicon probes forced development in another direction. The large volumes of data required above all increasing automation. Several very capable software suites have been developed, including the Klusta suite (Rossant et al. 2016), Kilosort (Pachitariu et al. 2016), MountainSort (Chung et al. 2017), Spyking Circus (Yger et al. 2018), Tridesclous (Pouzat and Garcia n.d.) and YASS (Lee et al. 2020). Typically, these must exploit the locality of extracellular signals and project spike waveforms in a reduced dimensional space, where the dimensions are the principal components of spike waveforms. In most suites, spikes are then redetected and sorted using template matching of the cluster waveforms.

The software presented here arose from the requirements of spike-sorting cerebellar cortical activity. Few laboratories perform multielectrode recordings of cerebellar activity (de Solages et al. 2008; Gao et al. 2012; Tsutsumi et al. 2020; Sedaghat-Nejad et al. 2021). Most prefer carefully targeted single-unit recordings, which therefore are generally performed acutely. Chronic recordings tend to have lower signal-to-noise ratios than such acute recordings and it is not possible to isolate a single unit. Additional challenges of the analysis of cerebellar multielectrode recordings include the high-frequency of activity in the Purkinje cell layer and the presence of complex spikes. For several reasons, the latter in particular tend to slip through the cracks with standard configurations of spike-sorting software, as explained by Sedaghat-Nejad et al. (2021). Firstly, the filtering and windowing may need to be adjusted to ensure that low-frequency components of the complex spike are not attenuated or truncated (Zur and Joshua 2019). Secondly, the variable waveform complicates detection and alignment. Finally, their low firing frequency compared to that of simple spikes (roughly a 100-fold difference) means they may only weakly influence the principal component decompositions; in consequence, they are often poorly represented during the sorting, which would tend to degrade the clustering of their waveforms. An alternative method to detecting complex spikes using deep learning has also been explored (Markanday et al. 2019).

As part of a series of experiments aiming to perform longitudinal chronic multielectrode recordings of Purkinje cells during eyeblink conditioning, we sought to optimise and automate the sorting of complex spikes. Our overall approach was to perform then merge multiple parallel analyses adapted to complex and simple spikes; we combined the outputs from multiple runs of Kilosort and MountainSort. To achieve this, we automated cluster fusions within a decision framework ensuring that cluster quality could only be increased. This in turn required jitter correction both within and between clusters and elimination of duplicate spikes. This resulted in a pipeline in which all post-processing was fully automated. We verified the efficacy of our optimisations by injecting spike waveforms into the traces and quantifying their recovery. For cerebellar recordings our package improves the spike-sorting analyses of two standard packages.

Having established the utility of our software for analysing cerebellar activity, we tested it on the synthetic SpikeForest datasets approximating activity from other brain regions (Magland et al. 2020). Here, too, our software outperformed the constituent Kilosort and MountainSort analyses. Our software is called *Lussac* and is available under a free-software licence.

## 2 Materials and Methods

The bulk of the techniques implemented will be explained in later sections, to avoid repetition and to maintain a logical narrative.

### 2.1 Data

The data presented here were obtained from chronic recordings in experiments studying longitudinal changes of cerebellar cortical and particularly Purkinje cell activity during eyeblink conditioning. The data and recording methods are described in detail in the corresponding paper (Llobet et al, in preparation). In brief, 64-site multielectrodes (Cambridge Neurotech) were implanted in mouse cerebellum and advanced towards the putative eyeblink region until cells responding strongly to the air puff directed at the ipsilateral eye were recorded. Continuous recordings throughout learning and extinction sessions (per day, 3 sessions of 30 minutes and 75 trials) were obtained. The mice were walking continuously on a wheel; the possible influence of the locomotion on the cerebellar activity should be borne in mind. However, the overall statistics of firing are unlikely to be strongly influenced by the brief eyeblink conditioning trials. The recordings were band-pass filtered (0.1 Hz–7.5 kHz) at acquisition and before spike sorting the instantaneous median of all 64 channels was subtracted point-wise from each channel; this attenuated common local field potentials, interference and movement artefacts. The data will be made available as part of Llobet et al (in preparation).

We also validated our software using the synthetic 'cortical' dataset described in Jun et al. (2017) and made available as part of the SpikeForest project (Magland et al. 2020).

### 2.2 Software

We used Kilsort 2 (which performed better than the newer versions) and MountainSort 3 and 4. We implemented a small patch in MountainSort 4 enabling the use of an asymmetric window around detected spikes in order better to accommodate the long waveform of complex spikes without introducing unnecessary noise through signal-free portions of the trace preceding detected spikes.

Our software package, implemented in Python 3.8, accepts multiple input files in the 'PHY' format (`https://phy.readthedocs.io/en/latest/`) or can schedule analyses via SpikeInterface (Buccino et al. 2020). The operation of the package is controlled by parameters that are read from a simple configuration file edited by the user. The package, called *Lussac*[1], is publicly available under a free-software licence at `https://github.com/BarbourLab/lussac`.

### 2.3 Filtering

Unless otherwise stated, offline filtering employed a 2nd-order Butterworth filter of the stated corner frequency/frequencies, applied forwards and backwards to eliminate any time shift.

### 2.4 Wavelet decompositions

The continuous wavelet decompositions of Fig. 1 were produced using the `scipy.signal.cwt` python library. The 'morlet 2' wavelet family was used.

## 3 Spatiotemporal profiles of simple and complex spikes

Before describing the specific adaptations implemented for sorting complex spikes, we illustrate their properties in comparison with simple spikes. The

[1] An appropriate interpretation as a backronym would be **L**ots of **u**nique **s**pike-**s**orting **a**nalyses **c**ombined.

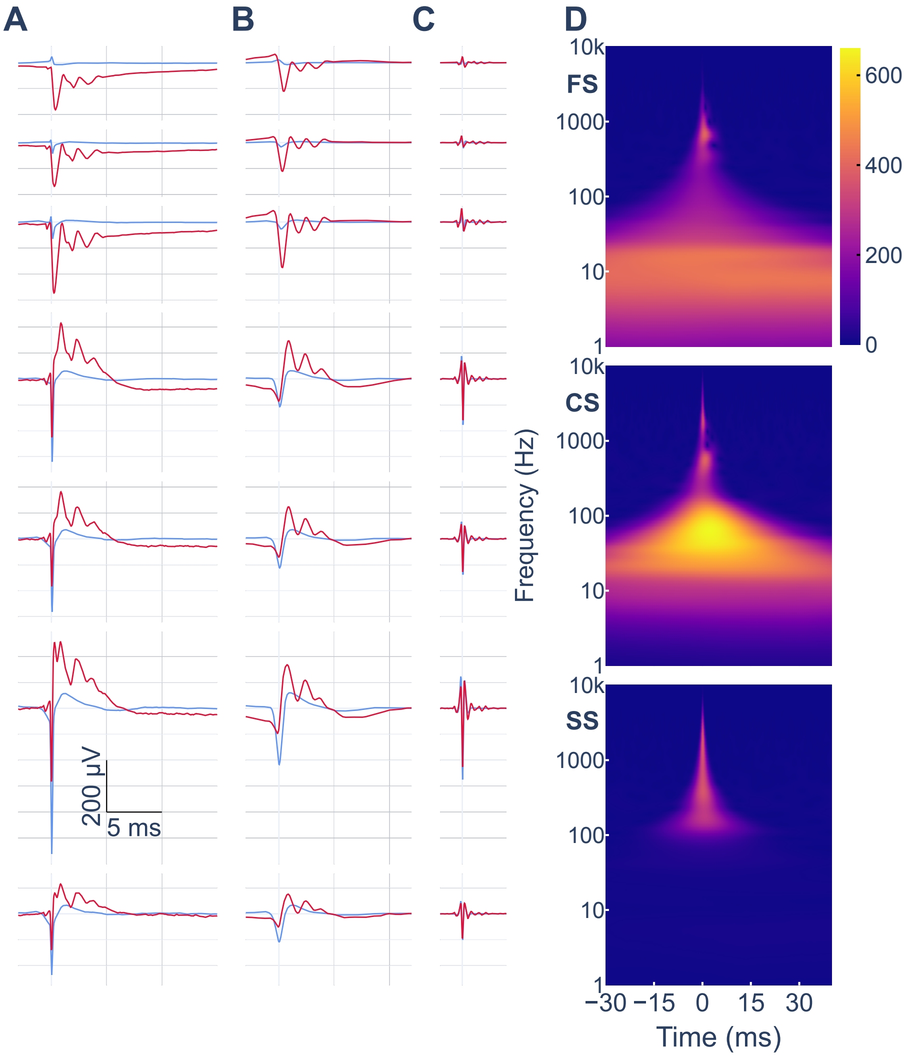

Figure 1: Temporal and frequency domain decompositions of simple and complex spike waveforms. Specimen average simple (blue) and complex (red) spike waveforms before filtering (**A**) and band-pass filtered at 60–1000 Hz (**B**) or 1500–9000 Hz (**C**); recorded from sites on a single electrode shank, with sites arranged in two columns separated by 25 $\mu$m and staggered by 12.5 $\mu$m vertically, with dorsal sites at the top of the figure. **D**. Wavelet spectrograms of the dendritic (top) and somatic (middle) complex spike waveforms, and the somatic simple spike (bottom). The units of spectral density are arbitrary. FS fat spike, CS complex spike, SS simple spike.

extracellular signals of simple and complex spikes are generated through different mechanisms. For simple spikes, the main dipole is generated by the action potential itself in the axon initial segment and soma. Inward membrane current in these compartments generates an extracellular negativity and returns via the proximal dendrites; generating a more diffuse positivity in that region. For complex spikes, the driving sink is the large, distributed synaptic current entering the proximal dendrites (where the field it generates is called a 'fat spike'; Gao et al. 2012) and it appears to return near the soma (Voorhoeve 1967; Han et al. 2020), probably through a somatic potassium conductance; there is also evidence for an expected passive return through spiny branchlets, which are not innervated by the climbing fibre. Upon these synaptically-generated dipoles are superimposed a small number of somatic action potentials, the first of which resembles the simple spike, while the remainder appear as spikelets, may and may involve both dendritic calcium spikes and axonal action potentials (Monsivais 2005; Otsu et al. 2014).

The waveforms of simple and complex spikes both depend upon the position of the electrode with respect to the Purkinje cell. Examples are shown in Fig. 1A. The forms of the complex spike are quite variable. However, despite the variation of signal sign, the components of the different dipoles will always have similar frequency components, simply because every inward current must always be balanced by an outward current to satisfy conservation of charge. The relative constancy of the frequency components (as opposed to the amplitude or sign) can be seen in the wavelet decompositions (spectograms; Fig. 1D).

The frequency components can also be observed in the temporal domain. Thus, in Fig. 1B,C, we illustrate simple and complex spike waveforms after bandpass filtering with low- and high-frequency ranges. Although the high-frequency components of the simple spike can also be found in the complex spike, their amplitude is strongly attenuated at positions removed from the soma. In contrast, the slowest frequency components are both exclusive to the complex spike and have a reasonable amplitude at most recording locations, although the underlying dipole is negative near the dendrites and positive near the soma.

The nearly ubiquitous slow frequency components exclusive to complex spikes constitute a key characteristic for detecting them and for discriminating them from simple spikes, as has already been implemented (de Solages et al. 2008) and characterised (Zur and Joshua 2019).

## 4 Parallel analyses and process automation

With standard configurations, most spike-sorting software works well for simple spikes but less so for complex spikes. As explained in the previous section, a distinguishing feature of the complex spike is a low-frequency, long-lasting signal component; indeed it is sometimes the only visible signal. There can also be several spikelets following the first spike, which resembles the associated simple spike. Optimal signal processing for complex spikes would therefore involve lower corner frequencies for the high-pass filter (possibly also for the low-pass filter), and longer peri-spike windows. However, these changes would all decrease the signal-to-noise ratio for simple spike processing. Furthermore, many algorithms cluster spikes by projecting them onto data-derived principal components. Depending on the details of the procedure, complex spikes, which fire at much lower frequencies than simple spikes, may barely influence the largest principal components and in consequence be poorly represented and less easily separable during the clustering.

Instead of trying to find a compromise set of parameters that would nevertheless risk being sub-optimal for both simple and complex spikes, we decided to adopt a strategy of performing parallel analyses optimised separately for simple and complex spikes. This necessarily entailed developing procedures for merging the clusters arising from different analyses, but the procedure was also of potential benefit within analyses. Complex spikes were sometimes detected with noticeable jitter (more an issue with MountainSort, because Kilosort performs template matching) and we found that a useful preliminary to cluster fusion was (re)alignment of spike waveforms within a cluster and of cluster averages. An additional objective was to automate the entire post-processing procedure. This optimises researcher time while reinforcing objectivity and reproducibility of the analyses.

We shall now outline (Fig. 2) the fully automated, modular pipeline for cleaning and merging multiple analyses including from different analysis suites. Each step will be described in detail in the subsequent sections.

We used multiple runs of MountainSort 3 and 4 (with a minor modification allowing for longer and asymmetrical vectors around detected spikes, to accommodate complex spikes, see Methods) and Kilosort 2 with varying parameter sets. The key parameters that we varied were the filtering band, the detection threshold and the extent of the vector around each spike . The filtering band favoured complex spikes (low frequencies) or simple spikes (high frequencies). A high detection threshold ensured that high signal-to-noise units were detected without contamination, while lower thresholds enabled more inclusive detection of units with lower signal-to-noise ratios.

Potential simple spike and complex spike clusters were then identified, largely on the basis of their firing frequencies, refractory periods and waveforms. Complex spikes were realigned within clusters. Clusters of all spike types were aligned to facilitate comparisons, however, it was also necessary to align pairs of clusters mutually when evaluating waveform similarity, comparing correlograms, removing duplicates and counting coincident spikes. Within each analysis, candidate clusters for fusion were identified on the

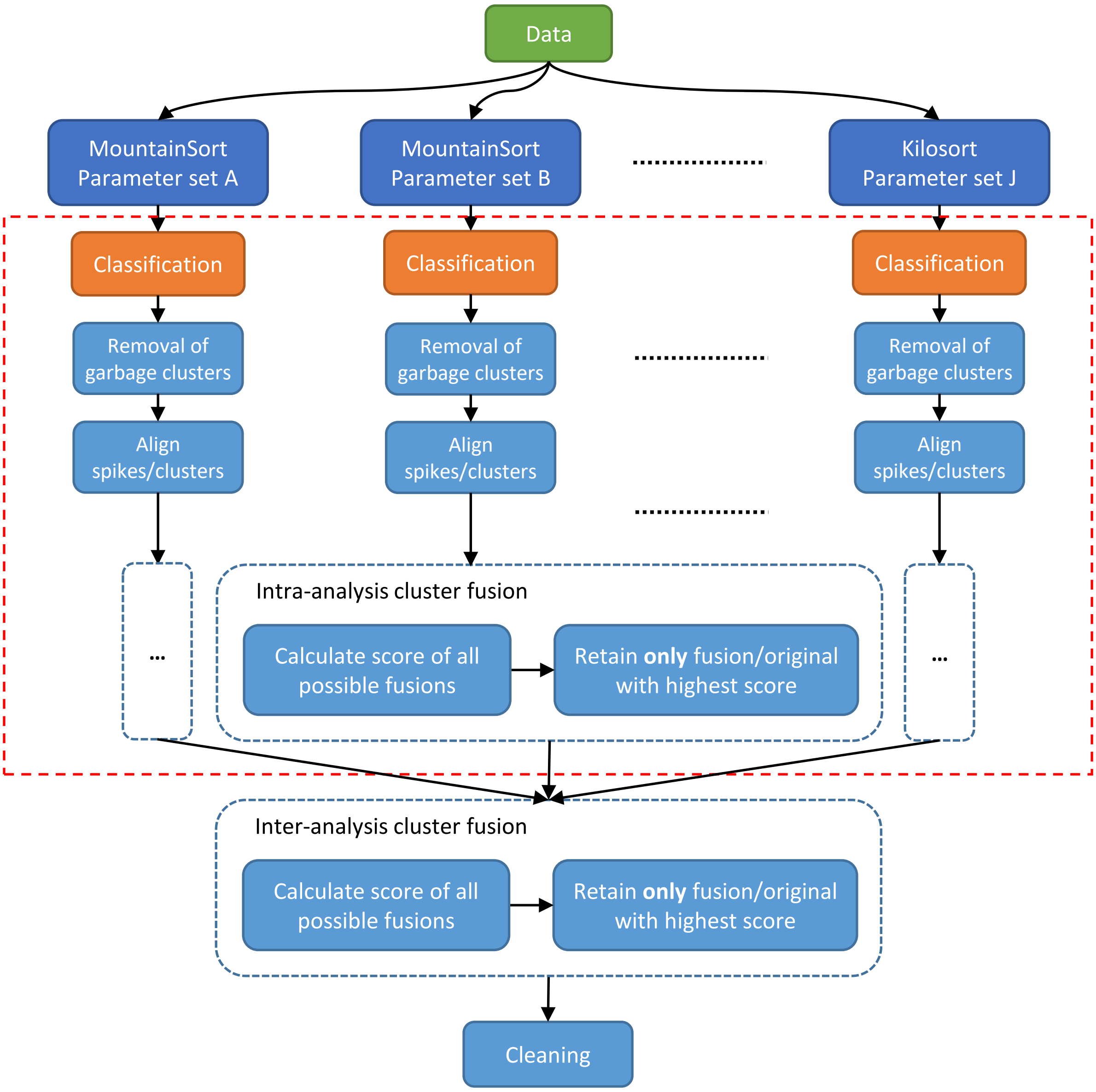
Data
MountainSort Parameter set A
MountainSort Parameter set B
Kilosort Parameter set J
Classification
Removal of garbage clusters
Align spikes/clusters
...
Intra-analysis cluster fusion
Calculate score of all possible fusions
Retain **only** fusion/original with highest score
Inter-analysis cluster fusion
Calculate score of all possible fusions
Retain **only** fusion/original with highest score
Cleaning

Figure 2: Flowchart outline of the automatic pipeline. Multiple analyses using the same spike-sorting suite with different parameters and/or different suites can be combined; in the example analysis presented below, we combined 5 MountainSort and 5 Kilosort analyses. The red, dashed box represents automation of what are often manual post-processing steps.

basis of waveform similarity and similarity of auto- and cross-correlograms. Clusters were merged if, after elimination of duplicate spikes, a higher-quality cluster resulted from the merge, based upon a score incorporating firing frequency and estimated contamination. To combine analyses, candidate clusters for fusion were identified on the basis of a high proportion of coincident spikes and then, as for within-analysis fusions, according to waveform similarity, similarity of auto- and cross-correlograms and quality score. At various stages of the analysis, low-quality clusters were removed to reduce calculation time. However, the most stringent selection was reserved until the end of the pipeline.

## 4.1 Cluster quality

A key criterion in curating the data and guiding decisions on cluster fusion is the quality of a cluster. The main factors determining cluster quality relate to its member spikes: the number of spikes correctly detected and attributed to the cluster, and the degree to which the cluster has been contaminated by spikes from other neurones or by noise events. An additional quality issue is the jitter of spike alignment. Although some information about cluster quality can be obtained by calculating distances with neighbouring clusters, which may for instance show that they overlap, these considerations generally inform the clustering process. Here we focus on the classic method of evaluating violations of the refractory period, which is an intrinsic estimator of quality that does not require inclusion of information about other clusters.

As a preliminary, it should be noted that absolute and sometimes relative refractory periods can be shorter than a millisecond at physiological temperature and, in some neurones, longer intervals are in practice never invaded, presumably because sufficient excitatory drive is not available. Thus, in the cerebellum, molecular layer interneurones and mossy fibres can display inter-spike intervals shorter than a millisecond, but the intervals between Purkinje cell simple spikes are rarely shorter than 2 ms. Although inferior olivary neurones can burst at high frequencies, the spikes of a burst contribute to a single complex spike of the Purkinje cell. Complex spikes are only rarely observed at intervals shorter than about 30 ms. For simplicity, we restrict our estimates of contamination to fixed time windows, for complex and for non-complex spikes. In an abuse of terminology, we shall continue to call these intervals normally devoid of spikes the refractory period. The measurements censored a small window close to zero, because detection of spikes within this window was sometimes unreliable and their processing was also inconsistent between software packages. Thus, for non-complex spikes, we measured refractory-period violations at times of 0.4–0.9 ms following the reference spike. The upper limit of 0.9 ms was imposed by mossy fibre activity, whose firing frequency could exceed 1 kHz. For complex spikes, contamination was evaluated over a window of 1.5–30 ms, because only very low levels of activity were observed within it for the complex spikes with the highest signal-to-noise ratios.

Two contamination models representing extremes of a continuum can be considered. In one, a single unit is responsible for all of the contaminating spikes. In this case, only intervals between the original and contaminating unit give rise to violations of the refractory period, because intervals between spikes of the contaminating neurone will also respect their refractory period. In the alternative, contaminating spikes arise from numerous other neurones or simply represent noise. In this extreme, any contaminating event could violate any refractory period and we assume that the violations of refractory periods of all spikes in the cluster are random.

We define cluster contamination $C$ as the fraction of contaminating spikes (or noise events):

$$C = \frac{n_c}{N}$$

where $n_c$ is the number of contaminating spikes, $N$ the total number of spikes and $n_t$ the number of true spikes; $N = n_t + n_c$. We assume that firing statistics are stationary.

We define the duration of the recording to be $T$ and the refractory period to be $t_r$, where both of these durations should be understood to exclude the censored intervals $\pm t_c$ surrounding detected spikes (here, 1.5 ms for complex spike analyses and 0.4 ms for other spikes). Thus, if the uncensored total time is $T'$, the uncensored refractory time $t'_r$ and the censored time (for one spike) is $t_c$, then $t_r = t'_r - t_c$ and $T = T' - 2Nt_c$. (Additional, small potential corrections involving $t_c$ have been neglected. Here and below, primes are used simply to distinguish similar quantities calculated under different assumptions.) Then, for a recording lasting time $T$, the expected number of contaminant spikes falling within the duration of refractory period $t_r$ of a given reference true spike is $n_c t_r / T$, assuming independence of the timing of true and contaminating spikes (and assuming throughout that spike numbers are sufficiently large such that accounting for the reference spike is unnecessary). Their total number, summed over all true spikes, would thus be $n_t n_c t_r / T$. By a symmetrical argument, each of the true spikes is expected to give rise to an equivalent number of violations of the refractory periods of the contaminant spikes, giving rise to the factor of two in the total number of refractory period violations

$$n'_v = \frac{2 n_t n_c t_r}{T}. \tag{1}$$

This is equivalent to the result obtained by Hill et al. (2011) on the basis that random 'rogue spikes' only interact with true spikes. In fact, this result seems more consistent with the case of a single neurone with its own refractory period (assumed also to be at least $t_r$) being responsible for the contaminating spikes. No additional violations would be expected, because that neurone will not violate its own refractory period. However, if contaminating spikes do occur randomly (because of noise or the contribution of many different units), it is necessary to include contaminant-to-contaminant violations. By a similar argument to that above, their number will be

$$\frac{n_c^2 t_r}{T}$$

and the total number of violations becomes

$$n_v = \frac{(2n_t + n_c)n_c t_r}{T}. \tag{2}$$

We can then solve each equation for the fraction of contaminant spikes under the two hypotheses, giving, for the case of random contaminating spikes

$$C = 1 - \sqrt{1 - \frac{n_v T}{N^2 t_r}} \tag{3}$$

where a quadratic root has been eliminated because the $n_c$ cannot exceed $N$, and, for the case of a single contaminating unit

$$C' = \frac{1}{2}\left(1 - \sqrt{1 - \frac{2n_v T}{N^2 t_r}}\right). \tag{4}$$

In this equation, we selected the smaller root since, by definition, the contaminant neurone is considered to be the one with the smaller number of spikes in the cluster.

The equations for $C$ and $C'$ may not have a real solution if $n_v$ is large; this can arise if the activities of reference and contaminant spikes are strongly correlated, thus causing a departure from the assumption of independent firing, and/or as a result of stochastic variations. In such cases - almost invariably removed by a quality threshold - we simply follow the convention of setting $C = 1$.

We shall use the estimate for random contamination of Eq. (3), because we consider this model more plausible. If the assumption of randomness is not satisfied, this choice will lead to an underestimation of the contamination. If the contamination is low, the two estimates are similar but the possible underestimate grows with the contamination. If $C = 0.2$, $C' = 0.24$, while if $C = 0.25$, $C' = 0.32$. The largest meaningful value of $C'$ is 0.5, at which point $C = 0.29$.

The estimated contamination varies through stochastic variations of $n_v$ and coefficient of variation of the contamination (sd/mean) can be approximated by $1/\sqrt{(n_v)}$, which decreases as $t_r$, $T$ or the firing frequency increase.

Correlated activity between the reference and contaminating spikes falling within refractory period leads to error in the estimated contamination, in the sense that a positive correlation causes an overestimate of contamination. This clearly occurs for complex spikes contaminated by neighbouring complex spikes. Because complex spikes systematically display synchronicity, any contamination is likely to be overestimated several fold. This is why different quality thresholds will be used below for complex spikes. In general, both positive and negative correlations between neurones can occur within refractory periods. Overestimates can arise from (positively) correlated activity, for instance through the action of gap junctions as described. Conversely, neurones whose activities are anti-correlated during, as might occur during some oscillations, would tend to cause underestimations of contamination. However, more important than the absolute quality value for our procedure is whether the quality increases or decreases in a possible fusion.

In addition to excluding spikes from other units, high cluster quality requires detecting most spikes from a unit, i.e. not to miss spikes. This quality is most simply captured indirectly—good clusters will have high frequencies yet low levels of contamination. We therefore combined these two factors into a single quality-metric

$$Q = f(1 - (k + 1)C) \tag{5}$$

where $f$ is the firing frequency of the cluster, $C$ is the contamination (see above) and $k$ a user-selected constant. $k$ can be understood as a ratio of true spikes to contaminants, such that a cluster resulting from a fusion is only considered to have a higher quality than a source cluster nested within it if the non-shared spikes contain at least $k$ times as many true spikes as contaminants, i.e.

$$\Delta n_t > k\Delta n_c, \tag{6}$$

where $\Delta n_t$ and $\Delta n_c$ are differences between the merged cluster and a source cluster. Throughout this paper, we have used $k = 2.5$, which means that the increase in the number of true spikes must exceed by that factor the increase in the number of contaminants resulting from any fusion.

## 4.2 Cluster classification

Complex spikes clusters required specific additional post-processing steps compared to other spikes (including simple spikes), so the clusters were classified as complex spikes or non-complex spikes.. The expected characteristics of complex spikes and those of other neurones were drawn from the literature but also, in a slightly circular manner, from examination of units recorded with high signal-to-noise ratios, which are assumed to be little affected by contamination or loss of spikes.

Complex spikes are widely reported to have frequencies around 1 Hz. We identified as potential complex spikes units with firing frequencies between 0.2 and 5.0 Hz and with fewer than 5 % of inter-spike intervals between 5 and 35 ms. The lower frequencies accommodate incomplete and split clusters while the upper frequency bound could be set inclusively because few other units had such low frequencies and discharge characteristics; it allows for the correct classification as complex spikes clusters that have been erroneously merged, simplifying their subsequent removal on the grounds of low quality. Non-complex spikes (including simple spikes) were therefore categorised by a complementary frequency range (1–250 Hz). The relatively low frequency limits were to allow retention of split clusters for future merging.

After this classification step, a non-stringent contamination limit of 35 % was applied to remove very low-quality non-complex-spike clusters. The contamination limit was inclusive at this stage in anticipation of stricter quality control at the end of the pipeline.

### 4.3 Spike alignment

Detection of complex spikes can be subject to substantial jitter, as a result of the low-pass filtering applied to optimise their detection and also because spikelets are sometimes detected. We found it necessary to align spikes within some complex spike units in order to facilitate comparisons of cluster waveforms, enable accurate detection of coincident/duplicate spikes and also to improve cross-correlogram accuracy. The alignment procedure was implemented and applied systematically to all complex spike clusters. (It could be applied to non-complex-spikes, but with additional computing cost for little gain.)

The overall approach was to shift each spike to a best-matching position with respect to a template, and the procedure outlined below is illustrated in Fig. 4. First, all spikes were band-pass filtered at 300–4000 Hz. The filtering reduced the influence of noise on the matching calculations. The matching was performed by finding the smallest shift between the test and reference waveforms for which a maximum of the scalar product exceeded a threshold. Multiple shifted scalar products could be efficiently calculated using a cross-correlation function, so the peak of that was taken. Calculations were performed on the 5 channels with the largest peak. The threshold was set adaptively, as follows.

1. A reference waveform was identified (this will be described below) and its scalar product with itself was calculated (‘reference scalar product’).

2. The cross-correlation peaks between all of the test waveforms and the reference were calculated.

3. Large values were winsorised to (i.e. replaced by) the reference scalar product.

4. The mean absolute deviation (MAD) of the winsorised cross-correlation peaks was calculated.

5. The threshold was set at (reference scalar product) - 5.18 MAD of these peaks.

The intuitive explanations for this approach are that the scalar product strongly favours coincidence of large peaks, exactly what is desired for spike alignment. The scalar product of the reference with itself represents the value expected for a noiseless ideal case. For this reason their values are winsorised when setting the threshold. Exploration with real data suggests that a threshold at about 5.2 MAD below the ideal will be exceeded by most spikes in the presence of typical noise. The choice of the first maximum implemented a preference for small shifts, thereby minimising disruption to the features that caused the clustering algorithm to form the cluster initially.

Reference waveforms were ‘bootstrapped’ in stages. First, a random sample of 150 spikes was selected from the cluster. For every spike, the optimum waveform alignment with each of the others was calculated according to the above procedure and the absolute values of the shifts to optimum alignment with the other waveforms were summed. The waveform with the smallest sum was chosen as the seed reference. The remaining spikes of the sample were aligned to it and the median waveform became the definitive reference. All spikes were then aligned to that reference and the cluster average waveform recalculated.

The alignment procedure is illustrated in Fig. 3.

Some clusters are naturally well aligned and for them the procedure produces little change (Fig. 4A). For others a larger effect is observed (Fig. 4B).

Finally, both complex spike and non-complex-spike clusters were shifted so that the peaks of their average waveforms aligned at time zero.

### 4.4 Cluster fusion

Many spike-sorting packages tend to over cluster. This is of course preferable to under clustering, because it is much easier to merge clusters during post-processing than it is to split them. We sought to automate the merging of clusters, both to relieve normal post-processing but also, more fundamentally, to enable the merging of multiple analyses.

Clusters were merged automatically at two levels: first within analyses, in which case units had been erroneously split, and secondly between analyses, when clusters from the same unit have many coincident spikes. (Coincident spikes can also occur within analyses, depending on the clustering

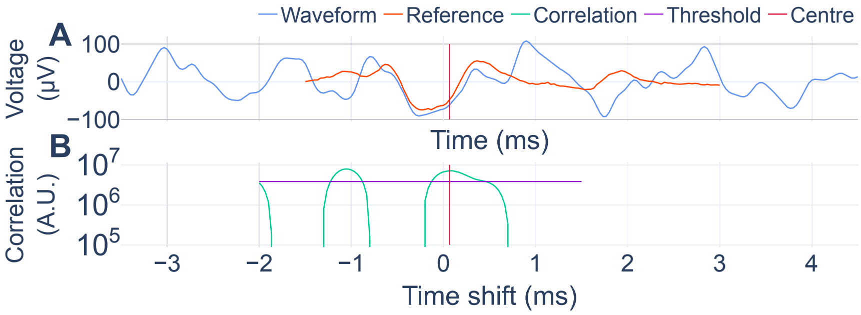

Figure 3: Illustration of spike alignment method. **A**. The bootstrapped reference waveform (orange) is shown at one shift relative to a single complex spike (blue). The cluster zero time is indicated by the red vertical line. **B**. The cross-correlogram (green) between reference and spike as a function of time shift (of the reference). The maximum (vertical red line) exceeding the threshold (horizontal mauve line) nearest zero shift was the selected alignment.

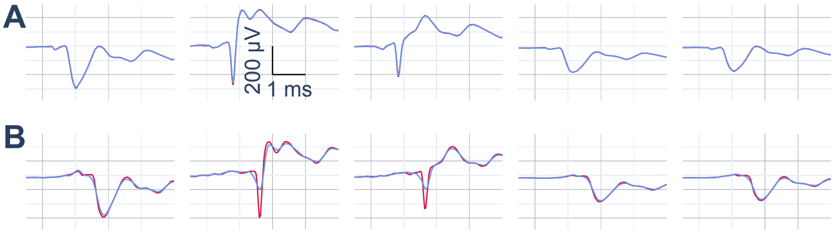

Figure 4: Spike alignment. Examples of complex spike cluster mean waveforms before (blue) and after (red) alignment, in which little change was produced (**A**) and in which a much sharper main spike was revealed (**B**).

algorithm.) The methods for merging clusters were similar for the two situations, with one specific difference. Thus, cluster candidates for fusion were identified on the basis of pairwise determinations of the similarity of waveforms, the similarity of their auto-correlograms with their cross-correlogram, and, only for inter-analysis fusions, on the basis of the numbers of coincident spikes. Candidates for fusion were only merged if a cluster of higher quality than all the source clusters was generated. These steps will now be detailed.

For inter-analysis fusions only, a proportion of 40 % simultaneous spikes were required for the clusters to be considered for fusion, where spikes were considered simultaneous if they occurred within an interval of 0.5 ms for complex spikes or 0.14 ms for any other spike.

To evaluate the similarity of the waveforms of two clusters, each average cluster waveform ($n \leq 10000$, 5 common channels with the largest absolute signal sum) was band-pass filtered (300–6000 Hz for simple spikes and 150–3000 Hz for complex spikes) they were then aligned (as above). The difference $\Delta$ of two waveforms $a$ and $b$ was determined as the sum of absolute differences of the aligned channels. This value was then normalised by the sum of the absolute values of both waveforms, which would compensate for variability that increases with waveform amplitude. Thus

$$\Delta = \frac{\sum\limits_{t,c} |V^a_{t,c} - V^b_{t,c}|}{\sum\limits_{t,c} |V^a_{t,c}| + |V^b_{t,c}|}, \tag{7}$$

where the sums are taken over time points $t$, channels $c$.

Examples of aligned waveforms are shown in Fig. 5, where complex and non-complex-spikes judged to have arisen from the same or different neurones are shown.

Auto- and cross-correlograms offer powerful diagnostics for identifying clusters belonging to the same unit. A particularly useful property is that the shape of an auto-correlogram is unchanged (up to noise effects) if spikes are removed from the cluster at random. Thus, the correlograms of two daughter clusters generated by a random split of a parent cluster will have similar auto-correlograms. In this situation, if follows that the cross-correlogram of two clusters representing the same unit should be similar to each of their auto-correlograms, which should also be similar. Therefore, the final criterion we used to identify fusion candidates was the similarity of each auto-correlogram of a cluster pair with their cross-correlogram (the similarity of the auto-correlograms being implicit), as follows.

The correlograms were calculated and any central peak due to duplicate spikes zeroed (within ±5 ms for complex spikes and within ±0.2 ms for simple spikes). The correlograms were low-pass filtered at 10 Hz for complex spikes and 800 Hz for non-complex spikes, then normalised by the average rate within a window of ±400 ms for complex spikes and ±75 ms for other

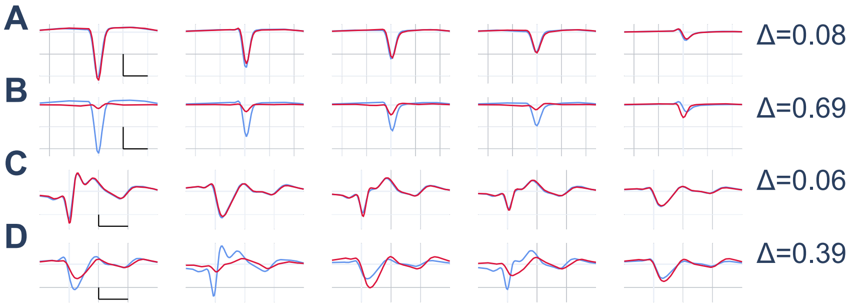

Figure 5: Examples of cluster mean simple (**A**, **B**) and complex (**C**, **D**) spike waveforms considered to be similar (A, C) and different (B, D). Complex spikes were filtered at 150–3000 Hz and simple spikes at 300–6000 Hz. Scale bars: 200 $\mu$V/0.5 ms (A, B) and 100 $\mu$V/1 ms (C, D). The waveform differences ($\Delta$, see Text) are indicated.

spikes. The comparison of correlograms was restricted to a region around the central dip in order to maximise the sensitivity of the difference measure (comparing long, normalised baselines would reduce the sensitivity). This region was fixed at ±40 ms for complex spikes. Because non-complex spike auto-correlograms are quite heterogeneous, the comparison region was delimited adaptively as follows. For each auto-correlogram, we determined the most central minima of the second derivative that exceeded half of the average correlogram value. We next calculated a single window from a weighted combination of the positions of these minima, which was then employed for the three comparisons, as indicated by the vertical dashed lines in Fig. 6. The common limits were $\pm\bar{l} = \pm(n_1 l_1 + n_2 l_2)/(n_1 + n_2)$ where $l_1, l_2$ are the half-widths of the comparison windows for the two auto-correlograms and $n_1$ and $n_2$ are the numbers of spikes in each cluster. The differences of the auto-correlograms and the cross-correlogram were calculated analogously as the sums of absolute differences within the windows defined above. The two differences were combined in a weighted average: $\bar{d} = (n_1 d_1 + n_2 d_2)/(n_1 + n_2)$ where $d_1, d_2$ are the summed differences between each auto-correlogram and the cross-correlogram.

Fig. 7 illustrates as scatter plots the distributions of waveform and correlogram difference measures for complex and simple spikes from one recording session. Three groups of points tend to be observed: fusion candidates, which are delimited in the figure, a group of moderately similar clusters that are mostly recorded on the same electrode shank, and highly different clusters mostly recorded on different electrode shanks.

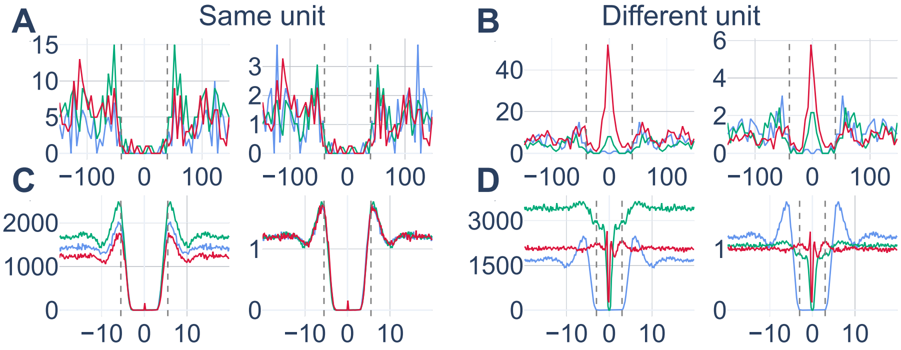

Figure 6: Examples of correlogram comparisons for fusion-eligible and ineligible complex and simple spike units. Similar (**A**, **C**) and different (**B**, **D**) auto- (blue, green) and cross- (red) correlograms; raw (left, units are counts) and normalised (right) correlograms are shown in each panel. Horizontal scale: time in ms. Dashed vertical lines indicate central peaks of the second derivative of the cross-correlogram (see Text), delimiting the regions in which correlograms were compared.

| | Intra-analysis | | Inter-analysis | |
|---|---|---|---|---|
| | CS | non-CS | CS | non-CS |
| Fraction coincident spikes | | | $\geq 0.40$ | $\geq 0.40$ |
| Waveform difference | $< 0.13$ | $< 0.50$ | $< 0.13$ | $< 0.20$ |
| Sum of correlogram differences | $< 0.35$ | $< 0.16$ | $< 0.16$ | $< 0.14$ |

Table 1: Criteria for cluster fusion eligibility.

In all possible pairwise comparisons, the number of common spikes (for inter-analysis fusions), the waveform differences and the correlogram differences were combined as indicated in Table 1 to determine which clusters were eligible for fusion. The different thresholds for complex spikes and others reflect an adaptation for the low firing rates of the former, which render the correlograms relatively noisy. Thus, when comparing complex spikes, a relatively greater weight is given to the difference of waveforms, while for non-complex spikes, the correlogram difference was considered a stronger diagnostic. These analyses provided a list of all cluster pairs eligible for fusion. Using the pairwise eligibility, the clusters were partitioned into groups eligible for fusion. Thus, if clusters A and B were eligible for fusion and also clusters B and C, then A, B and C were all grouped together for potential

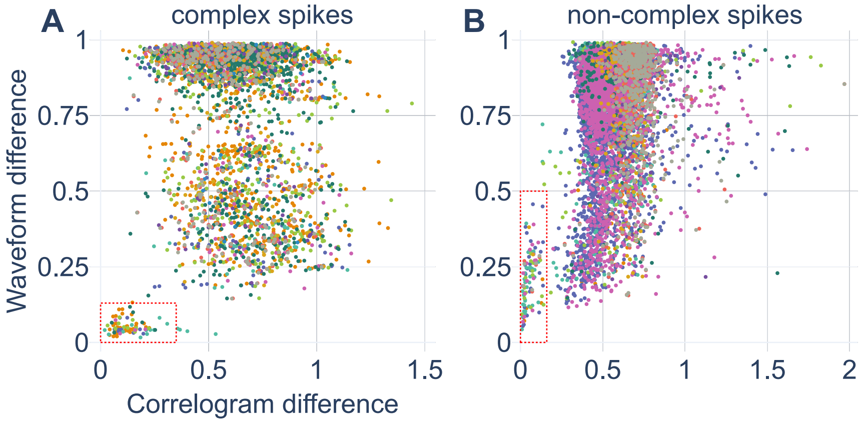

Figure 7: Scatter plots illustrating all pairwise correlogram and waveform differences from 10 analyses (each a different colour) from one recording session for complex (**A**) and simple (**B**) spikes (from different animals); in both cases the acceptance rectangle is outlined by a dashed red line. About 35 % of clusters were merged for both spike classes, with a large majority occurring in the post-processing of Mountainsort analyses. The much larger numbers of points outside the eligibility boxes results from the dependence of the number of comparisons on the square of the number of clusters.

fusion. Within eligible groups, all fusions of pairs and triples were tested, and the merged (or single) cluster with the highest quality was retained, using the metric of Eq. 5. Before calculating cluster quality, duplicate spikes were removed from the merged cluster by the simple method of retaining only a single non-complex spike within any interval of 0.4 ms and a single complex spike in any interval of 1.5 ms. (During intra-analysis fusions, the spike that best fitted the cluster mean was retained; during inter-analysis fusions, the first spike of the coincident pair was kept.)

This procedure of testing multiple fusions seeking the maximal quality score is susceptible to overestimating the quality because of noise in its estimation. Below we validate the procedure using datasets for which a ground truth is available, showing that the procedure is nevertheless able to improve the quality of the results.

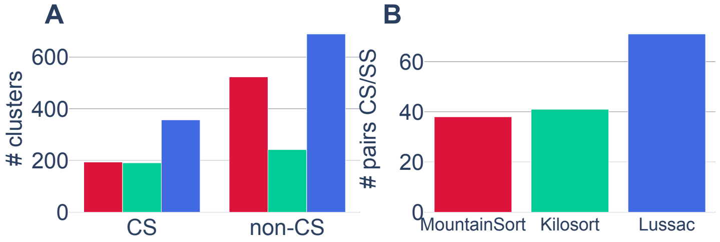

Figure 8: Performance of the multi-analysis pipeline. **A**. Numbers of complex ('CS') and non-complex ('non-CS') spike clusters detected using compromise parameters for MountainSort (red), Kilosort (green) and *Lussac* (blue). **B**. Number of Purkinje cells with both simple and complex spikes for the different packages.

## 5 Algorithm performance and validation by spike injection

We compared the performance of our multi-analysis procedure with single analyses of MountainSort and Kilosort, each with a parameter set hand-optimised for detection of both complex and non-complex spikes. All analyses underwent the same post-processing pipeline described above, and final clusters were selected using a contamination threshold of 20 % (a threshold that eliminates a significant proportion of molecular layer interneurones, whose signal-to-noise ratios are lower and whose contamination levels correspondingly higher). Purkinje cells with both simple and complex spikes were identified on the basis of a profound pause: the frequency of simple spikes between 0 and 8 ms after the complex spike was below 50 % of the frequency in the 20 ms before. We first counted the number of clusters of each type detected. We also counted the number of Purkinje cell clusters with both simple and complex spikes. The results are graphed in Fig. 8. *Lussac* provided an approximately 50 % improvement in the number of simple/complex spike pairs over the single analyses.

We also verified the overall accuracy of the spike sorting for native and combined analyses by injecting spikes into the existing traces and observing how they were detected. We injected unfiltered mean spike waveforms from existing clusters generated by the *Lussac* pipeline, avoiding known existing spike times from the same cluster and their refractory periods, and reanalysed the data. We compared the results with the *Lussac* package with those from the MountainSort and Kilosort analyses. This injection analysis was performed for 271 complex spike and 192 non-complex-spike clusters, with a number of spikes equal to 10 % (up to 1000) of the total number in each source cluster total being injected.

At reanalysis, a spike was considered coincident with an injected spike if they coincided within $\pm t_C = \pm 116\,\mu$s for non-complex spikes and within $\pm 350\,\mu$s for complex spikes (the specific values chosen relate to multiples of the sampling interval). The apparent proportion of recovered spikes was thus $N_C/N_I$, the number of coincident spikes divided by the number injected. However, contaminant spikes within a cluster could coincide with injected spikes by chance. We applied a crude correction for this contribution (greater accuracy was not necessary), by using the total number $N$ of spikes in a cluster at reanalysis as the upper bound for the number of contaminant spikes and normalising the result. Thus the (lower limit) for the proportion of recovered spikes was

$$\theta = \frac{N_C - 2NN_I t_C/T}{N_I(1 - 2NN_I t_C/T)} \tag{8}$$

Then, if $\theta$ exceeded 0.2 for a single cluster, this was identified as the cluster matching that from which the spikes were injected. If no or more than one cluster exceeded this threshold, we considered the injected spikes not to have been recovered.

Examples of injected spikes are shown in Fig. 9A,B. Evaluating the performance of recovery requires incorporation of information about selectivity and sensitivity. True positives were quantified as the proportion of injected spikes recovered in the cluster, while false positives were quantified using the estimate of the contamination of the full cluster. These were combined analogously to the quality score in Eq. 5,

$$Q_I = \frac{N_C(1 - (k+1)C)}{N_I}, \tag{9}$$

where $Q_I$ is the quality score for injections, $C$ is the contamination estimate as above and the constant $k$ also retains its meaning. The score tends to be lower for clusters with high signal-to-noise ratios, as can be seen in Fig. 9C–F, and was consistently higher for the *Lussac* package than the individual sorting softwares. It is worth recalling that for complex spikes the contamination is overestimated because of their correlated firing, causing the quality score of all complex spike clusters to be underestimated.

We carried out a more detailed analysis of the performance of the different packages on the injected spikes (Fig. 10). This shows that *Lussac* recovered more spikes from more clusters, with on average less contamination and an overall better quality score than both Mountainsort and Kilosort, for both complex and non-complex spikes.

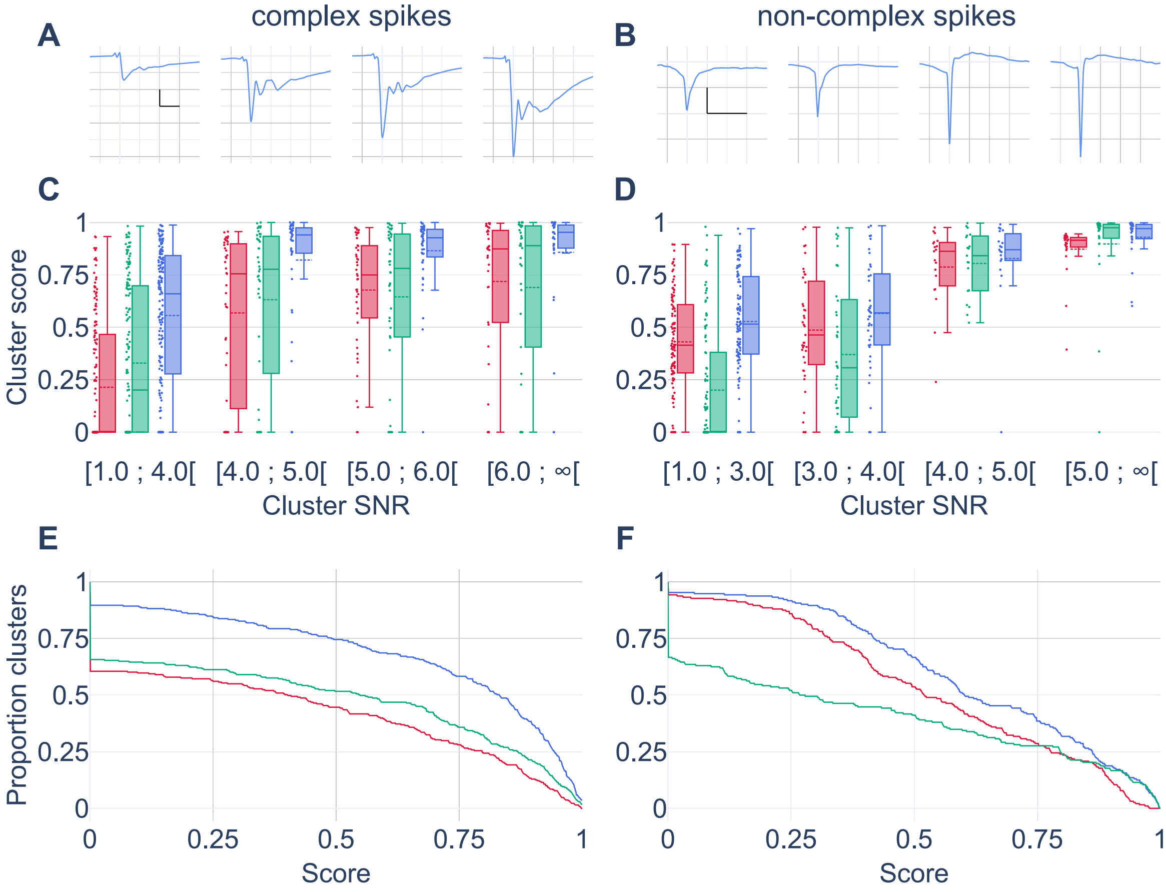

Figure 9: Validation of sorting by spike injection. **A**. Specimen complex spike waveforms before injection, with signal-to-noise ratios corresponding to the classes below in panel C. **B**. Simple spikes before injection, as for A. Scale bars in A and B: $100\,\mu\mathrm{V}/2\,\mathrm{ms}$. **C**. Quality score of injected complex spikes (see Eq. 9) for clusters in different signal-to-noise classes. MountainSort, red; Kilosort, green; *Lussac*, blue. Box plots (median, solid line; mean, dashed line) and individual cluster values (circles). **D**. Quality score of injected simple spikes. Same format as C. **E**. Fraction of injected complex spike clusters exceeding the quality score on the x-axis. MountainSort, red; Kilosort, green; *Lussac*, blue. **F**. Fraction of injected simple spike clusters exceeding the quality score on the x-axis. Format as for E. In C–F, negative cluster quality scores have been set to zero.

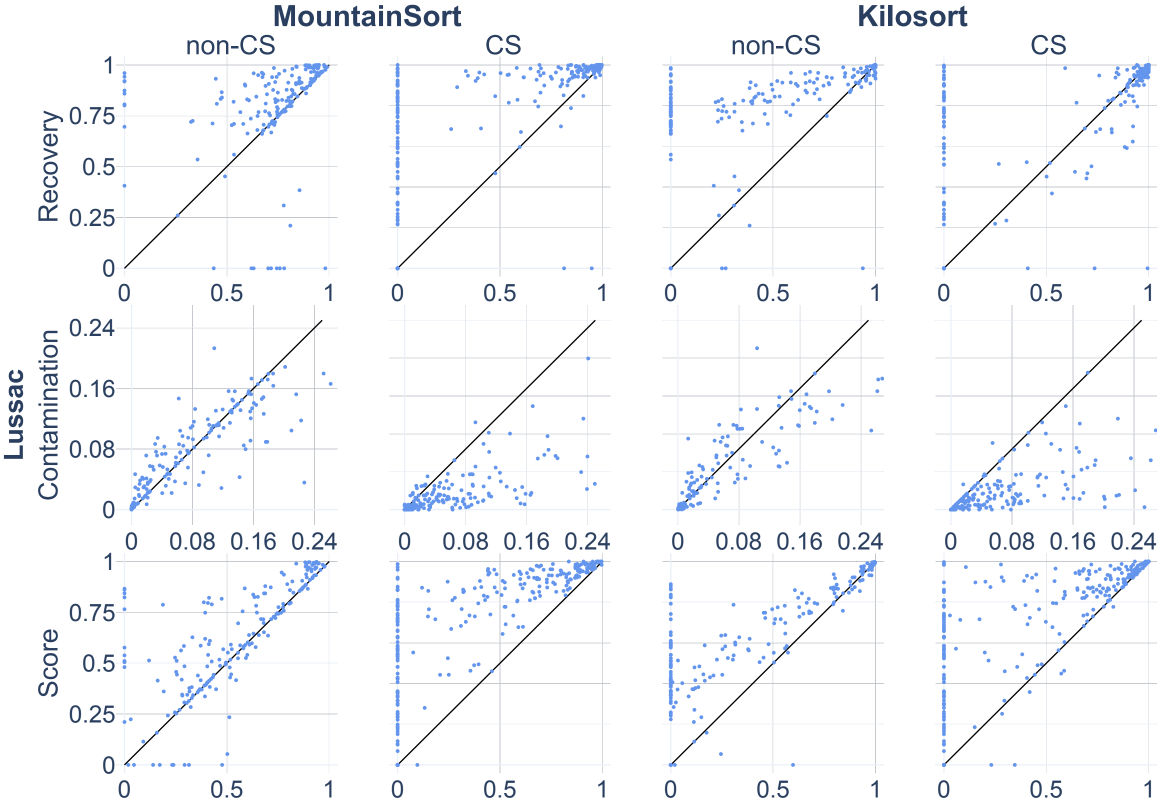

Figure 10: Recovery characteristics of injected spikes. Grid comparing the proportions of injected spikes recovered per cluster (zero if the cluster was not uniquely identified) (top row), the estimated contamination $C$ of clusters containing recovered spikes (middle row; not shown if the cluster was not found in one or both of the analyses) and quality score (bottom row), for *Lussac* (y-axes) vs. Mountainsort and Kilosort (x-axes). There are columns for both complex and non-complex spikes. In each plot the diagonal line indicates $Y = x$. *Lussac* is favoured by points above the diagonal in the top and bottom rows, and below the diagonal in the middle row.

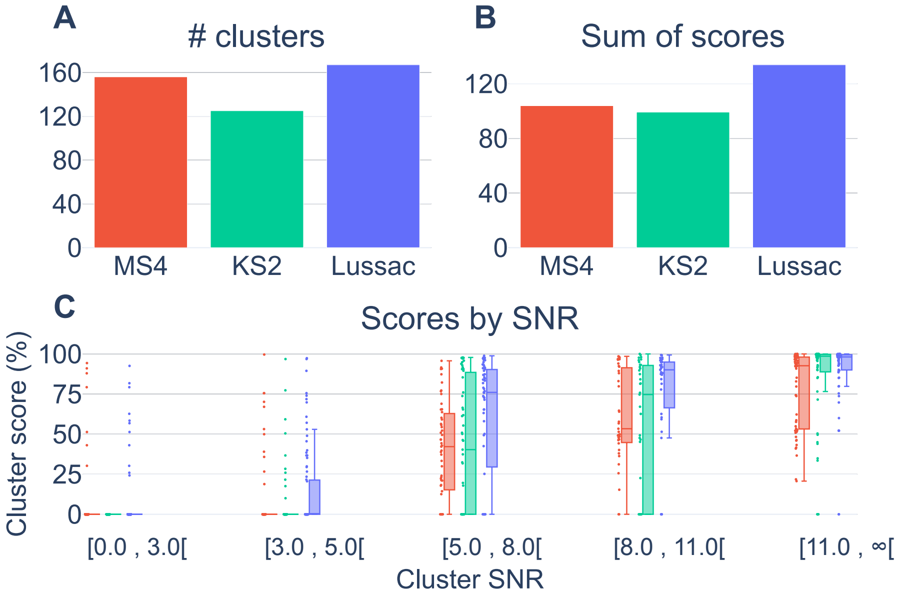

Figure 11: Validation against SpikeForest simulated datasets. **A**. Numbers of clusters detected by MountainSort 4, Kilosort 2 and *Lussac*. **B**. Sums of cluster quality scores for the same packages. **C**. Box plot of quality scores for clusters grouped by signal-to-noise ratio for the same packages.

## 6 Validation using SpikeForest simulated datasets

The successful validation of our package using injected spikes is subject to some circularity, both because we chose to inject spikes that had been correctly detected by our package, which would tend to confer an advantage for recovering the same spikes, and because we evaluated performance using the same quality score guiding cluster fusions and retention. In order to validate our package in a way that avoided this issue, we sought a dataset including a ground truth. We decided to use a synthetic data set generated by simulation of cortical pyramidal cells and internrueones (Jun et al. 2017) and made available through the SpikeForest project (Magland et al. 2020). An additional advantage was that these simulations aimed to reproduce activity from a brain region other than the cerebellum.

Because every spike time is known in the simulated dataset, we were able to determine the absolute quality according to the score of (9). As shown in Fig. 11, our package outperformed both MountainSort and Kilosort in terms of the number of clusters detected (Fig. 11A) and cluster scores (Fig. 11B). The breakdown of the score results depending on signal-to-noise ratio (Fig. 11C) shows that some of this performance arises, as expected, through better treatment of units with lower signal-to-noise ratios.

## 7 Discussion

From a starting point of trying to improve the treatment of cerebellar complex spikes in standard spike-sorting analysis packages, we have developed a principled method for combining analyses from different packages and using different parameters. As part of this development, we have fully automated the basic post-processing procedures of combining over-split clusters and applying a quality threshold to the results.

A comparative evaluation of our procedures with single analyses from MountainSort and Kilosort demonstrated that the performance of our package was never worse and, in particular for complex spike clusters with lower signal-to-noise ratios, sometimes markedly better. The strictly positive improvement is guaranteed by the use of the final quality metric to guide cluster selection and fusion during the merges—actions are only undertaken if they improve quality. Thus, insofar as our quality metric is valid, our method should never make things worse.

There was, however, no guarantee that merging analyses would improve results. The improvement observed unsurprisingly depended on the signal-to-noise ratio of the unit. All analyses methods yielded similar good results for units with a strong signal, although *Lussac* was generally still able to eke out a small improvement. At lower signal-to-noise ratios, however, the difference was more marked. We believe several different mechanisms contribute. The most obvious is that the ability to merge analyses offers the freedom to optimise parameters for different spike types. The cerebellar complex spike is a fairly extreme example of an unusual spike waveform for which current software has not been explicitly optimised. One key element of optimisation is the ability to vary filtering and window lengths to accommodate the long-lasting components of the complex spike waveform.

A partially related issue is the ability to vary detection threshold. A high threshold will detect only the strongest units and will isolate them from noise, ensuring a very low contamination; we believe this characterises the behaviour of Kilosort with standard parameters, which include a relatively high detection threshold. Lower thresholds will detect smaller signals, but will equally allow noise events to increase contamination of clusters that would have remained less contaminated with a higher threshold. The ability to combine analyses with low and high thresholds offers the possibility of obtaining the best of both worlds: retaining pure clusters of strong signals detected with a high threshold, but also including clusters of weaker signals detected with a lower threshold.

An unexpected discovery arising from the merging of multiple analyses was the detection of what appeared to be incorrectly merged activity from multiple neurones, a phenomenon that arises when different neurones have similar waveforms and anti-correlated activity, which reduces the rate of refractory period violations, giving the appearance of a well isolated cluster. Identification of such over-merged clusters requires multiple analyses, and is based on the presence of clusters in different analyses sharing a significant proportion of coincident spikes while the clusters have differing autocorrelograms, so must partially represent different neurones. We plan to implement the detection and removal of such over-merged clusters in a future version of our software.

Many current spike-sorting packages, including MountainSort and Kilosort are not deterministic. This may arise from the (very reasonable) acceleration technique of performing some steps of the analysis on random spike samples of controlled size. However, we can observe that this randomness manifests in stochastic detection of entire clusters at lower signal-to-noise ratios. The procedure of combining multiple analyses enables these intermittent detections to be accumulated, resulting in a higher probability of detection.

The use of multiple analyses forced us to automate what were traditionally manual post-processing steps. Depending on the (ever-increasing) size of the dataset, this automation could represent a significant gain in researcher efficiency, but also in objectivity and reproducibility.

For our initial objective of improving treatment of cerebellar complex spikes, our package is beneficial, a finding validated by spike injection. When interpreting the absolute quality metrics of the complex spike clusters, it should be borne in mind that their contamination tends to be overestimated (because of their correlated firing). For non-complex spikes, the gains from merging analyses were more modest but still appeared worthwhile. Moreover, as shown by the validation using the simulated datasets from SpikeForest, combining multiple analyses also seems to offer a significant improvement for activity recorded from other brain regions.

Merging analyses offers a simple method for improving spike sorting that does not require the use of a specific sorting algorithm. The approach is especially beneficial if the recordings include units with different waveform characteristics. Using one simple configuration file, the package offers a turnkey approach to automating both post-processing and the merging of multiple analyses. The package can in principle be applied to recordings from any brain region, with the only requirement being interoperability with SpikeInterface or the supply of sorting outputs in the PHY format. Finally, the flexible architecture of the package will facilitate its extension using custom modules.

## 8 Acknowledgements

This work was supported by the Fondation pour la Recherche Médicale through grant 'Equipe FRM DEQ20160334927'. This work has received support under the program 'Investissements d'Avenir' launched by the French Government and implemented by the ANR with the references ANR–10–LABX–54 MEMOLIFE and ANR–10–IDEX–0001–02 PSL* Université Paris.

We thank the SpikeInterface team for very helpful discussions and for implementing suggested extensions. We are also grateful to Guillaume Dugué for helpful discussions.

## References

(1) Buccino, A. P., Hurwitz, C. L., Garcia, S., Magland, J., Siegle, J. H., Hurwitz, R. and Hennig, M. H. (2020). SpikeInterface, a unified framework for spike sorting. *eLife 9*, e61834, DOI: 10.7554/eLife.61834.

(2) Chung, J. E., Magland, J. F., Barnett, A. H., Tolosa, V. M., Tooker, A. C., Lee, K. Y., Shah, K. G., Felix, S. H., Frank, L. M. and Greengard, L. F. (2017). A Fully Automated Approach to Spike Sorting. *Neuron 95*, 1381–1394.e6, DOI: 10.1016/j.neuron.2017.08.030.

(3) de Solages, C., Szapiro, G., Brunel, N., Hakim, V., Isope, P., Buisseret, P., Rousseau, C., Barbour, B. and Léna, C. (2008). High-Frequency Organization and Synchrony of Activity in the Purkinje Cell Layer of the Cerebellum. *Neuron 58*, 775–788, DOI: 10.1016/j.neuron.2008.05.008.

(4) Einevoll, G. T., Franke, F., Hagen, E., Pouzat, C. and Harris, K. D. (2012). Towards reliable spike-train recordings from thousands of neurons with multielectrodes. *Current Opinion in Neurobiology 22*, 11–17, DOI: 10.1016/j.conb.2011.10.001.

(5) Gao, H., Solages, C. d. and Lena, C. (2012). Tetrode recordings in the cerebellar cortex. *Journal of Physiology-Paris 106*, 128–136, DOI: 10.1016/j.jphysparis.2011.10.005.

(6) Gray, C. M., Maldonado, P. E., Wilson, M. and McNaughton, B. (1995). Tetrodes markedly improve the reliability and yield of multiple single-unit isolation from multi-unit recordings in cat striate cortex. *Journal of Neuroscience Methods 63*, 43–54, DOI: 10.1016/0165-0270(95)00085-2.

(7) Han, K.-S., Chen, C. H., Khan, M. M., Guo, C. and Regehr, W. G. (2020). Climbing fiber synapses rapidly and transiently inhibit neighboring Purkinje cells via ephaptic coupling. *Nature Neuroscience 23*, 1399–1409, DOI: 10.1038/s41593-020-0701-z.

(8) Hill, D. N., Mehta, S. B. and Kleinfeld, D. (2011). Quality Metrics to Accompany Spike Sorting of Extracellular Signals. *Journal of Neuroscience 31*, 8699–8705, DOI: 10.1523/JNEUROSCI.0971-11.2011.

(9) Jun, J. J., Mitelut, C., Lai, C., Gratiy, S. L., Anastassiou, C. A. and Harris, T. D. Real-time spike sorting platform for high-density extracellular probes with ground-truth validation and drift correction, preprint, Neuroscience, 2017, DOI: 10.1101/101030.

(10) Lee, J. et al. (2020). YASS: Yet Another Spike Sorter applied to large-scale multi-electrode array recordings in primate retina. *bioRxiv*, 2020.03.18.997924, DOI: 10.1101/2020.03.18.997924.

(11) Magland, J., Jun, J. J., Lovero, E., Morley, A. J., Hurwitz, C. L., Buccino, A. P., Garcia, S. and Barnett, A. H. (2020). SpikeForest, reproducible web-facing ground-truth validation of automated neural spike sorters. *eLife 9*, e55167, DOI: 10.7554/eLife.55167.

(12) Markanday, A., Bellet, J., Bellet, M. E., Hafed, Z. M. and Thier, P. (2019). Using deep neural networks to detect complex spikes of cerebellar Purkinje Cells. *bioRxiv*, DOI: 10.1101/600536.

(13) McNaughton, B. L., O'Keefe, J. and Barnes, C. A. (1983). The stereotrode: A new technique for simultaneous isolation of several single units in the central nervous system from multiple unit records. *Journal of Neuroscience Methods 8*, 391–397, DOI: 10.1016/0165-0270(83)90097-3.

(14) Monsivais, P. (2005). Determinants of Action Potential Propagation in Cerebellar Purkinje Cell Axons. *Journal of Neuroscience 25*, 464–472, DOI: 10.1523/JNEUROSCI.3871-04.2005.

(15) O'Keefe, J. and Recce, M. L. (1993). Phase relationship between hippocampal place units and the EEG theta rhythm. *Hippocampus 3*, 317–330, DOI: 10.1002/hipo.450030307.

(16) Otsu, Y., Marcaggi, P., Feltz, A., Isope, P., Kollo, M., Nusser, Z., Mathieu, B., Kano, M., Tsujita, M., Sakimura, K. and Dieudonné, S. (2014). Activity-Dependent Gating of Calcium Spikes by A-type K+ Channels Controls Climbing Fiber Signaling in Purkinje Cell Dendrites. *Neuron 84*, 137–151, DOI: 10.1016/j.neuron.2014.08.035.

(17) Pachitariu, M., Steinmetz, N., Kadir, S., Carandini, M. and Kenneth D., H. (2016). Kilosort: realtime spike-sorting for extracellular electrophysiology with hundreds of channels. *bioRxiv*, 061481, DOI: 10.1101/061481.

(18) Pouzat, C. and Garcia, S. Tridesclous https://github.com/tridesclous/tridesclous.

(19) Rossant, C., Kadir, S. N., Goodman, D. F. M., Schulman, J., Hunter, M. L. D., Saleem, A. B., Grosmark, A., Belluscio, M., Denfield, G. H., Ecker, A. S., Tolias, A. S., Solomon, S., Buzsáki, G., Carandini, M. and Harris, K. D. (2016). Spike sorting for large, dense electrode arrays. *Nature Neuroscience 19*, 634–641, DOI: 10.1038/nn.4268.

(20) Sedaghat-Nejad, E., Fakharian, M. A., Pi, J., Hage, P., Kojima, Y., Soetedjo, R., Ohmae, S., Medina, J. F. and Shadmehr, R. (2021). P-sort: an open-source software for cerebellar neurophysiology. *bioRxiv*, 2021.03.16.435644, DOI: 10.1101/2021.03.16.435644.

(21) Tsutsumi, S., Chadney, O., Yiu, T.-L., Bäumler, E., Faraggiana, L., Beau, M. and Häusser, M. (2020). Purkinje Cell Activity Determines the Timing of Sensory-Evoked Motor Initiation. *Cell Reports 33*, 108537, DOI: 10.1016/j.celrep.2020.108537.

(22) Voorhoeve, P. In *Progress in Brain Research*; Elsevier: 1967; Vol. 25, pp 268–281, DOI: 10.1016/S0079-6123(08)60968-1.

(23) Wilson, M. and McNaughton, B. (1993). Dynamics of the hippocampal ensemble code for space. *Science 261*, 1055–1058, DOI: 10.1126/science.8351520.

(24) Yger, P., Spampinato, G. L., Esposito, E., Lefebvre, B., Deny, S., Gardella, C., Stimberg, M., Jetter, F., Zeck, G., Picaud, S., Duebel, J. and Marre, O. (2018). A spike sorting toolbox for up to thousands of electrodes validated with ground truth recordings in vitro and in vivo. *eLife 7*, e34518, DOI: 10.7554/eLife.34518.

(25) Zur, G. and Joshua, M. (2019). Using extracellular low frequency signals to improve the spike sorting of cerebellar complex spikes. *Journal of Neuroscience Methods*, 108423, DOI: 10.1016/j.jneumeth.2019.108423.