Spike sorting pipeline for the International Brain Laboratory

International Brain Laboratory[*], **Kush Banga**[1], **Julien Boussard**[3], **Gaëlle A Chapuis**[5], **Mayo Faulkner**[1], **Kenneth D Harris**[1], **Julia M. Huntenburg**[6], **Cole Hurwitz**[3], **Hyun Dong Lee**[3], **Liam Paninski**[3], **Cyrille Rossant**[1], **Noam Roth**[4], **Nicholas A. Steinmetz**[4], **Charlie Windolf**[3], **Olivier Winter**[2*]

***For correspondence:** olivier.winter@internationalbrainlab.org (Olivier Winter)

[1]University College London, UK; [2]Champalimaud Foundation, Lisbon, Portugal; [3]Columbia University, NY, USA; [4]University of Washington, WA, USA; [5]University of Geneva, Switzerland; [6]Max Planck Institute for Biological Cybernetics, Tuebingen, Germany

May 7, 2022

Abstract The International Brain Laboratory (IBL) is in the process of collecting thousands of Neuropixels recordings across a dozen experimental labs. While the Neuropixels technology is powerful, it is still fairly young. As such, we have found that existing analysis pipelines were not sufficiently robust to reliably and reproducibly process this scale of data. Here, we document our current processing pipeline, including several modifications that enable more scalable and robust processing. We also document our newly developed tools for visualizing and quality-scoring the output of each stage of the processing pipeline. We close by discussing known issues in our pipeline that present opportunities for further improvements.

Compression

The raw electrophysiological recordings were initially saved in a flat uncompressed binary format, representing a storage of 1.3 GB per minute (384 channels at 30 kHz and 16 bits per sample). To save disk space and achieve better transfer speeds, we developed a simple lossless compression tool, *mtscomp* (***International Brain Laboratory, 2021b***), which achieves a compression ratio between 2x and 3x.

The algorithm uses chunks of recording, usually one second, applies a temporal derivative and compresses using an off the shelf algorithm (zlib). The compressed chunks are appended to a binary file and the byte offsets within the file stored in a metadata file. This allows random access to a part of the recording by jumping to the appropriate byte offset within the file.

For convenience our reading packages (***International Brain Laboratory, 2021a***) provide a seamless integration with the compressed files and can be access as a conventional numpy array.

Raw data quality metrics

To quickly assess the quality of an electrophysiology recording, we calculate the root mean square (RMS) and the power spectral density (PSD) in both the action potential (AP) and low-field potential (LFP) band for each channel.

To get an estimate of the quality early on, we reduce the data size by extracting evenly spread samples (1 second every 300 seconds). We then compute the RMS on this reduced data before and after “destriping" (detailed in the Preprocessing section below). We calculate the median RMS per channel as well as the 10th and 90th percentile of this median across the channels. These values can provide a first indication that there are issues with the raw data, e.g. noisy or dead channels, and for whether these issues would be resolved by destriping. After this initial pass, we compute the RMS on the full AP data before and after RMS. We then plot the entire timeseries for visual inspection (Figure 1A). For the smaller LFP data, the RMS is computed directly on the full raw time series and then plotted for visual inspection (Figure 1B).

The PSD is computed using Welch's method (***Welch, 1967***) for both the LFP and the destriped AP data. The PSD can be used, for example, to find anomalous channels in the recording (cf. Channel selection below). We also perform a simple spike detection on the samples of the AP data used for the initial RMS calculation. These spikes can be overlaid on the RMS or raw data to get a first idea if any spikes are detectable from the data and if issues in the RMS appear to translate to issues in spike detection.

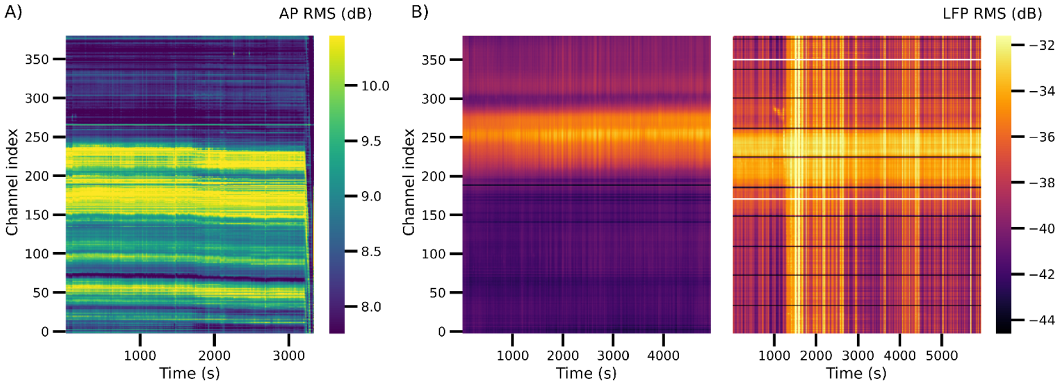

Figure 1. RMS plots used to assess recording quality. For visual inspection, RMS is calculated in a sliding window and plotted for each channel. (A) Example of an AP band RMS plot for one session after destriping. The recording quality is generally good, the channel signals are mostly stationary across time. However, the RMS plot also highlights that a spatial drift of the probe has occurred over time, notably from around 1800 seconds. At the end of the recording, the signal drifts completely, as if the probe was removed from the brain. (B) Two examples of LFP band RMS plots from two different sessions/animals. The LFP data is not destriped. The left plot shows a good recording. The signal is stationary within channels. The amplitude difference across channels can potentially be used to align the signal to brain regions. The right plot shows clear issues with this recording. Vertical stripes indicate electrical noise. Horizontal lines show flat reference channels of the 3A probe (***Jun et al., 2017***), and noisy channels that should be excluded.

Metric name	**Definition and units**	**Dimension**
`Spectral Density AP`	Power spectral density V^2/Hz	channels x frequencies
`Spectral Density LFP`	Power spectral density V^2/Hz	channels x frequencies
`Spike Rate`	Spikes per seconds with a simple threshold detection	Number of channels
`Channel labels`	Faulty channel detection: ok, dead, noisy or outside of the brain	channels
`RMS AP-band`	Median RMS, with 1 and 9th deciles, before and after pre-processing,	channels x 6
`RMS LFP-band`	Median RMS, with 1 and 9th deciles, before and after pre-processing,	channels x 6

The table summarizes the raw electrophysiology data metrics that are computed and stored.

Preprocessing

In many cases, we encounter line noise due to voltage leakage on the probe. This translates into large “stripes" of noise spanning the whole probe. While our first recommendation is to try to mitigate this noise during acquisition by grounding the apparatus, it is often not possible to achieve complete isolation and we observe transients on the order of 0.1 to 0.5 mV. To reduce the impact of these noise “stripes", we introduce three main pre-processing steps:

1. Correction for “sample shift" along the length of the probe by aligning the samples with a frequency domain approach.
2. Automatic detection, rejection and interpolation of failing channels.
3. Application of a spatial “de-striping" filter.

Each step is explained in more detail in the following sections.

High-pass filter

To account for the direct current (DC) offset of each channel, we apply a low-cut filter in the time domain, i.e., a third order Butterworth filter with a 300 Hz corner frequency.

Sample shift

On a Neuropixels 1.0 (phase 3A and 3B) probe, the analog to digital converter (ADC) operates per blocks of 32 channels. As there are 384 active channels on a probe, there are 12 ADC blocks. Within each block, the 32 channels are sampled sequentially over a duration of one sample; therefore, there is a small lag of 1/12th of a sample between consecutive channels. Figure 2A shows sample lag for each channel of the probe. It is worth nothing that the scanning rate is always 1/12th of a sample and is thus dependent on the sampling rate. For example, on the LFP band, the shift is still 1/12th of a sample even if the sampling rate is decreased from 30 kHz to 2.5 kHz.

Whether or not accounting for a time shift on the order of 30 μs is relevant for neural analysis, we determined that not accounting for it at the spatial filtering stage results in artefacts. Figure 2 illustrates the importance of re-phasing the channels before applying any kind of spatial filtering as artefacts due to discontinuities are reduced.

An efficient and accurate implementation of a sub-sample time shift is to apply it in the in frequency domain by applying a linear phase shift as a function of frequency.

$$\mathcal{F}\{x(t-t_0)\} = \exp(-j2\pi f t_0)X(f). \tag{1}$$

Vectorized implementation examples are available in Python (***International Brain Laboratory, 2021a***) and Matlab.

Channel selection

Anomalous channels break the spatial consistency of both signal and noise. For the vertical stripes we are targeting, not removing faulty channels can reduce the effectiveness of the algorithm at best, and at worst introduce artefacts. Selecting the “bad" channels is a two-step process. First, we detect anomalous channels. Then, we spatially interpolate over the bad channels using voltages from neighbouring channels.

Detecting anomalous channels

We distinguish between 3 main types of anomalous channels:

- Dead channels where the amplitude is abnormally low.
- Noisy channels where the amplitude is abnormally high.
- Channels outside of the brain.

An absolute voltage threshold on the RMS amplitude does not provide sufficiently robust detection of bad channels. This is because the recorded voltage amplitude depends on the brain

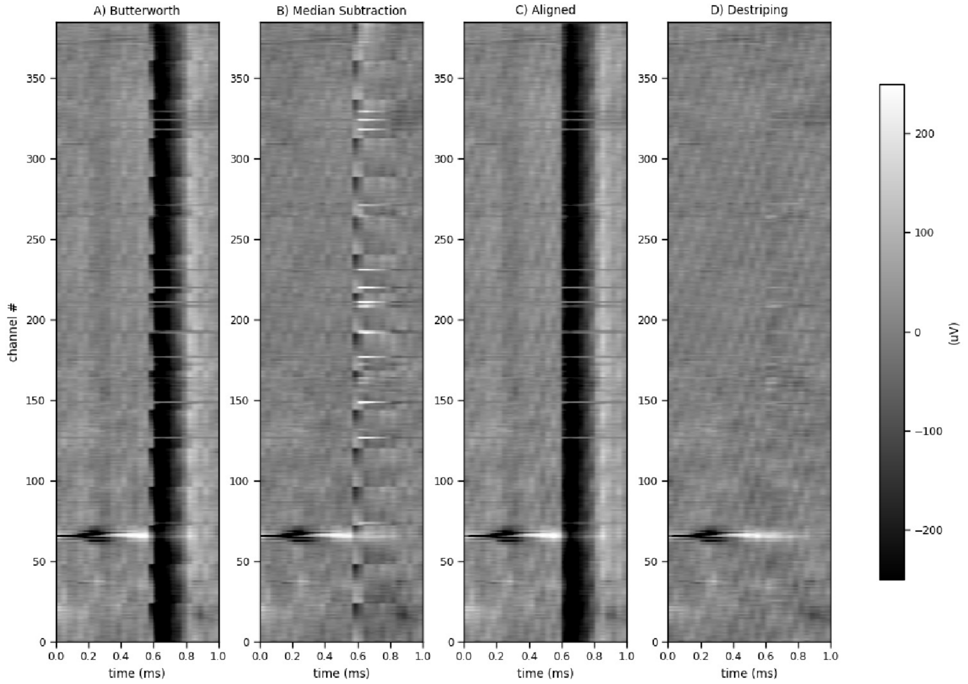

Figure 2. Effect of channel time-alignment on spatial filtering. 1ms of a voltage display with a high amplitude stripe (-400 uV) and a neural spike at channel 75 (-100 uV). (A) High-pass filtered, at this time scale the sampling time shifts are clearly visible (B) After median subtraction artefacts remain as channels are un-aligned. (C) Frequency domain re-alignment of channels. (D) Spatial filtering applied with aligned channels mitigates artefacts.

location and the neural activity. Instead, for low amplitude channels, we computed the zero lag cross-correlation peak with the common median reference. This zero-lag cross-correlation is normalized by the energy of the reference trace. To spot bad channels, we subtract the channels trend (a median filter with n=11 points) from the cross correlation which we refer to as the channel "similarity" (shown in Figure 3). A conventional channel will score around 0 and anomalously low amplitude channels are then tagged if under a threshold of -0.5.

For noisy channels, we use a frequency domain approach. While a conventional neurophysiological voltage power spectrum will tend to decay as a function of frequency, faulty channels exhibit a flatter spectrum. A good way to detect these channels was to apply an absolute threshold on the average power spectral density (PSD) above 0.8 Nyquist. For the action potential band (AP), our suggested threshold is 0.02 $\mu V^2\,Hz^{-1}$ and for the LFP band, our suggested threshold is 1.5 $\mu V^2\,Hz^{-1}$. Figure 3 shows an example on the AP band where 2 channels exhibit unnatural values for conventional neurophysiology.

For detecting channels outside of the brain, we use a similar method to the similarity thresholding, but we instead computed cross-correlations with the median of all channels instead of just the neighbouring channels. Only consecutive channels at the top of the probe below the threshold are excluded from further analysis.

Interpolating anomalous channels

For noisy and dead channels, we replace the voltages with a simple spatial interpolation using the voltages on neighbouring channels. The interpolation is a weighted sum of neighbouring channels where the weights decay as a function of distance. The weight of every channel is determined according to the equation below where the offset refers to the distance to the channel to be inter-

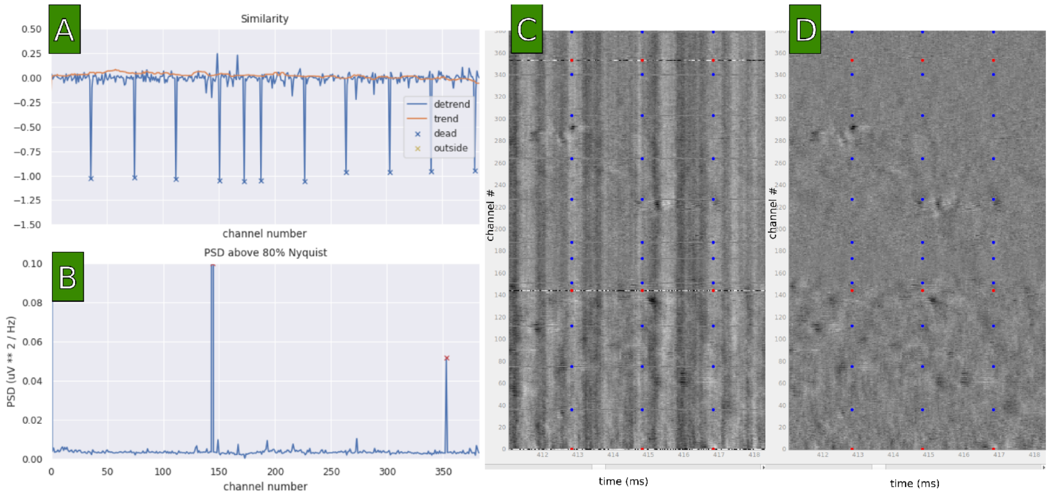

Figure 3. Bad channel tagging and interpolation. Noisy channels are labeled with red dots, dead channels with blue dots. (A) Similarity measure with neighbouring channels; dead channels score low on this metric. (B) Power spectral density above 80% of Nyquist frequency, 12 kHz. The noisy channels score very high on this metric. (C) Density display of corresponding voltage traces after high-pass filtering, showing the same anomalous channels. (D) Same voltage traces after interpolation over the removed bad channels.

polated:

$$w(\text{offset}) = \exp -\left(\frac{\text{offset}}{d}\right)^{p}. \tag{2}$$

The parameters d and p were set to 20 μm and 1.3, respectively. This is a similar algorithm to the probe drift correction method in the original Kilosort 2.5 algorithm (***Steinmetz et al., 2021***). Figure 3 shows the interpolation of noisy and dead channels.

High pass spatial filter

A common average referencing (CAR) filter removes the spatial DC component. We found this to be inadequate for performing destriping in many cases since the “stripe" was not completely constant across the full probe. Instead, we slightly extended the cut-band from the 0 spatial frequency to the 0.01 Nyquist corner frequency. (We used a Butterworth order 3 filter here.) Figures 4 and 5 illustrate the benefits of opening the cut band to low-spatial filtering versus removing only the common median. From this figure, however, it is clear that further improvements are likely possible. Figure 6 also illustrates the benefits of destriping the raw data before running spike detection and spatial localization.

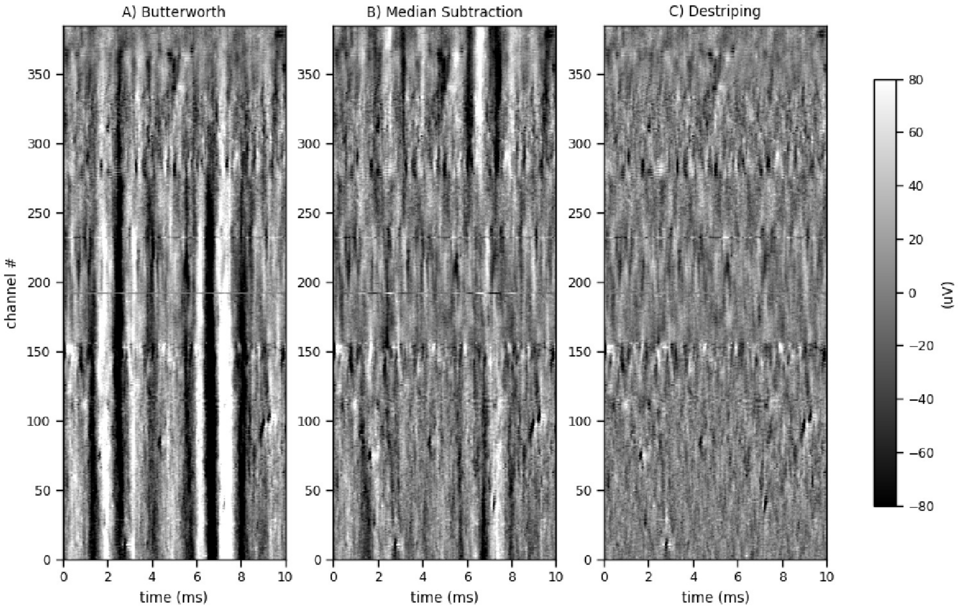

Figure 4. 100 ms of a voltage display illustrating spatial filtering advantages. (A) High-pass filtered. (B) After median subtraction. (C) After spatial destriping filtering. In this case striping artefacts span the lower 3/4 of the probes but do not extend to the surface. Subtracting the median results in artefacts at the upper and lower extremities of the probe.

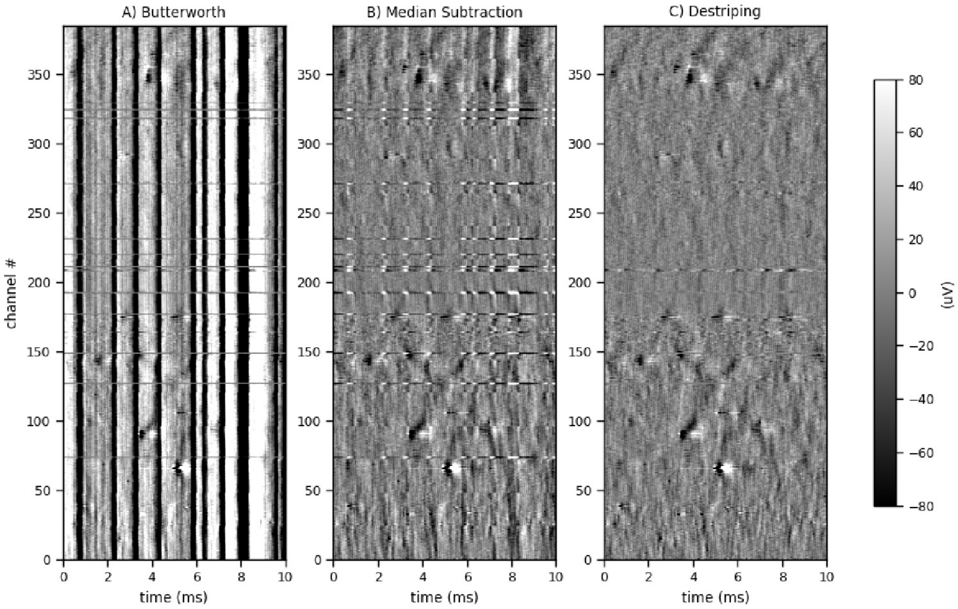

Figure 5. 100 ms of a voltage display illustrating high noise conditions. Conventions as in Figure 4. In this case, although destriping is an improvement over median subraction, artefacts remain and may contaminate the spike sorting output; thus there is still room to improve the processing chain in this case.

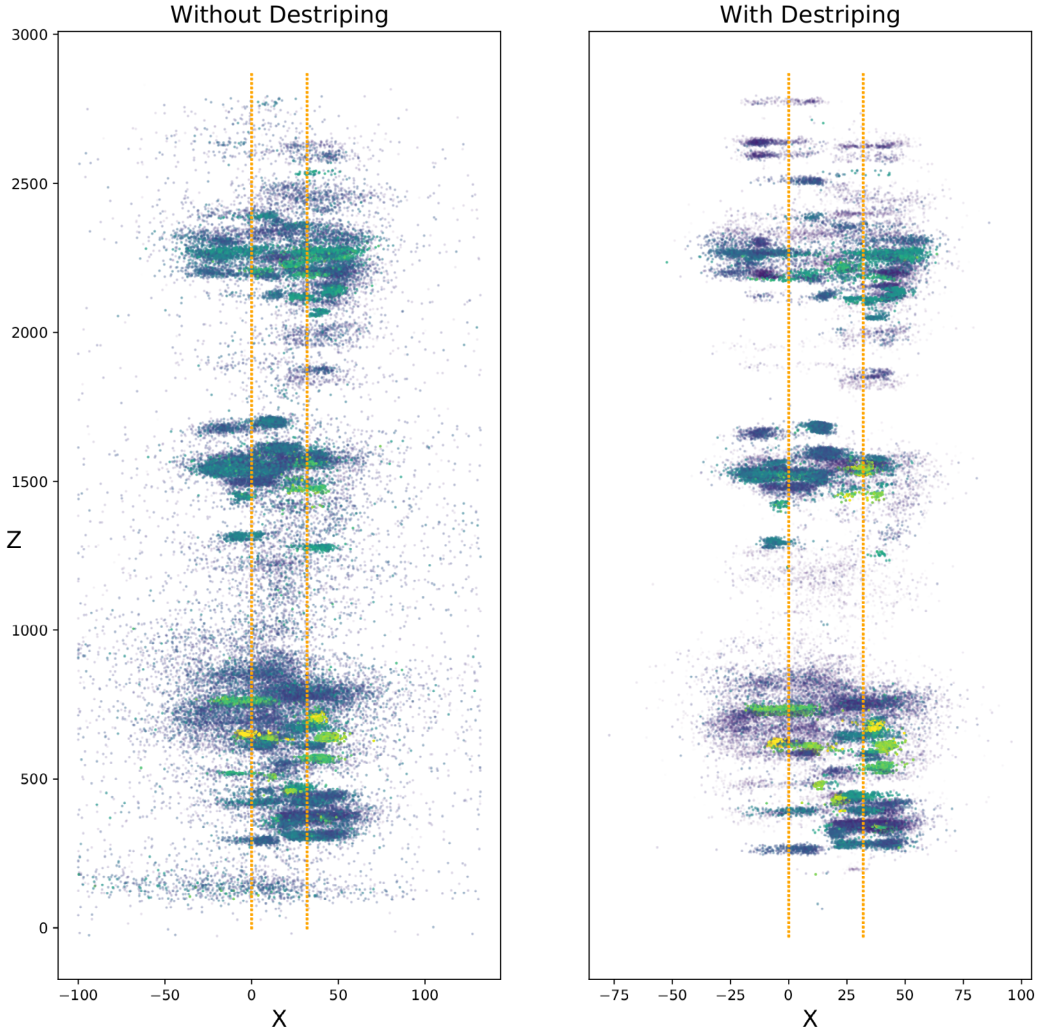

Figure 6. Effect of destriping on spike localizations. We ran spike detection on 50 seconds of a NP2.0 recording, before localizing the detected spikes using the method developed by ***Boussard et al.*** (***2021***). The two panels show the 2D-localization along the Neuropixels probe of detected spikes for the same dataset without destriping (Left) and with destriping (Right). The orange squares indicate the NP2.0 recording channels. The colors of the dots correspond to standardized amplitudes of the spikes, with dark colors (blue) corresponding to low-amplitude spikes, and light colors (yellow) corresponding to high-amplitude spikes. When not destriping the data, many artefacts are detected as spikes and localized all around the probe. This creates the “blue haze" visible in the left panel. This effect is ameliorated by destriping.

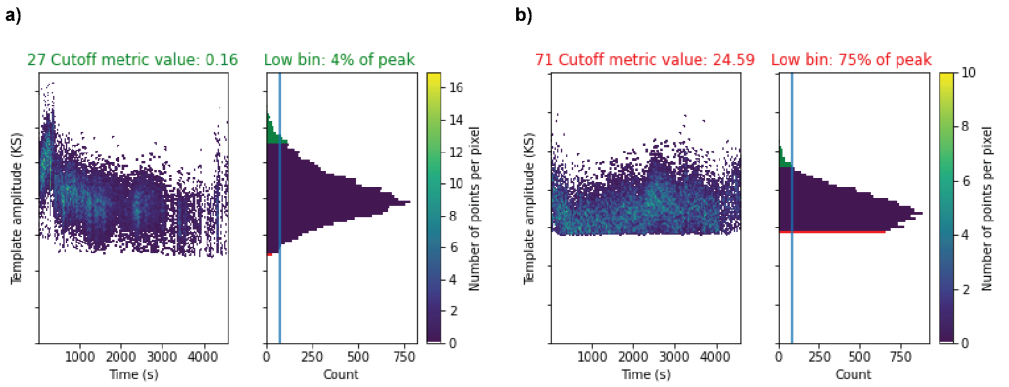

Figure 7. Illustration of the noise cut-off metric for a passing unit a) and a failing unit b). For each, the left panel shows the template amplitudes of the spikes as a function of time across the session, while the right panel shows the amplitude distribution. The metric is computed by comparing the height of the lowest histogram bin (red) with the bins in the high quantile of the distribution (green). The noise cut-off value is how many standard deviations the low bin falls outside of the mean number of spikes in the high quantile. For the unit a) the full distribution is captured while for b) the spike detection stops abruptly, potentially leading to biased analysis due to missing spikes. To pass, the computed cut-off metric value of a unit must be less than 5 (unit a: 0.16; unit b: 24.59), and the value of the lowest bin (highlighted in red) must be less than 10% of the height of the peak histogram bin (indicated by the vertical blue line; unit a: 4% of peak; unit b: 75% of peak).

Drift registration, clustering, and spike deconvolution

After the preprocessing steps above, we apply registration, clustering, and spike deconvolution code from Pykilosort 2.5 (***Steinmetz et al., 2021***; ***International Brain Laboratory et al., 2021***).

Single cluster quality metrics

A variety of single unit metrics are computed on each output cluster, largely based on the Allen Institute original implementation (***Siegle et al., 2017***) and later re-implemented in SpikeInterface (***Buccino et al., 2020***). Notable improvements relate to the refractory period violations and to the noise cut-off estimations, described further below. An illustration of the cut-off metric is illustrated in Figure 7.

Out of the set of computed metrics, we currently use three metrics to determine whether a single cluster will be used in downstream analysis. A neuron must pass all three metrics in order to be included in downstream analyses.

First, we compute a false positive estimate (slidingRP_viol). The Neuropixels data collected by the IBL was recorded across hundreds of brain regions, many of which have electrophysiological properties which have not been fully characterized. One example is the length of a neuron's refractory period; this parameter is often assumed to be 2 ms (***Siegle et al., 2017***; ***Hill et al., 2011***), but can vary across brain areas (***Bar-Gad et al., 2001***; ***Sukiban et al., 2019***). We thus developed a metric which estimates whether a neuron is contaminated by refractory period violations (indicating potential overmerge problems in the clustering step) without assuming the length of the refractory period. For each of many possible refractory period lengths (ranging from 0.5 ms to 10 ms, in 0.25 ms bins), we compute the number of spikes (refractory period violations) that would correspond to some maximum acceptable amount of contamination (chosen as 10%). We then compute the likelihood of observing fewer than this number of spikes in that refractory period under the assumption of Poisson spiking. For a neuron to pass this metric, this likelihood, or the confidence that our neuron is less than 10% contaminated, must be larger than 90% for any one of the possible refractory period lengths. This metric rejects neurons with short true refractory periods when firing rates are low, as we cannot be statistically confident that the lack of contamination did not arise by chance. As the true refractory period increases, neurons with low contamination begin to pass the metric. Thus, the metric has advantages for assessing neurons with both brief and long true refractory periods.

Next, we compute a noise cut-off estimate ('noise_cutoff'); this metric estimates whether an amplitude distribution is cut off by thresholding in the deconvolution step (thus leading to a large fraction of missed spikes). To do so, we compare the lowest bin of the histogram (the number of neurons with the lowest amplitudes), to the bins in the highest quantile of the distribution (defined as the top 1/4 of bins higher than the peak of the distribution.) Specifically, we compute how many standard deviations the height of the low bin falls outside of the mean of the height of the bins in the high quantile. For a neuron to pass this metric, this value must be less than 5 standard deviations, and the height of the lowest bin must be less than 10% of the height of the peak histogram bin.

Finally, we compute the median of the amplitudes. For a neuron to pass this metric, the median of the amplitudes must be larger than 50 uV.

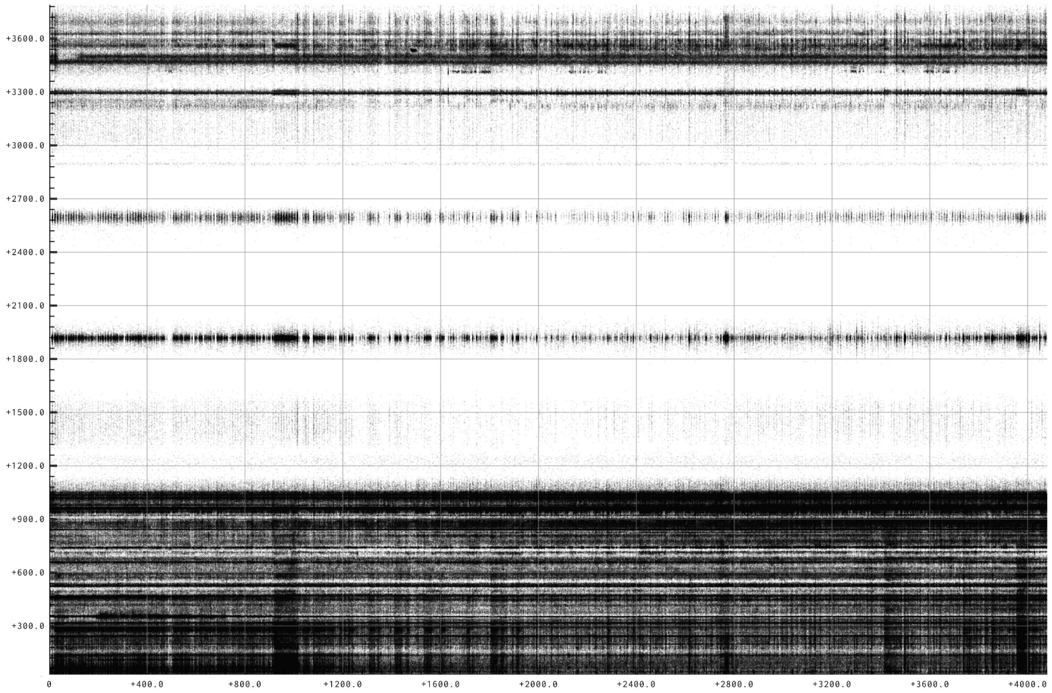

Figure 8. Display of a drift raster plot which the user can pan and zoom interactively. More than 10 millions points can be shown with good performance on a mid-entry GPU. Since the spike points have a low transparency value, the GPU rendering pipeline computes on hardware a heatmap of the spike density as a function of time and probe depth. The marker size and transparency can dynamically increase in real time at high zoom levels to improve visualization of individual spikes.

Visualization

High performance scientific visualization

For visualization, we use new GPU-based technologies that allow for fast display of large datasets. These technologies can operate on either the desktop (Vulkan, glfw) or on the web (Vulkan, WebSocket) via a custom distributed rendering architecture that avoids large data transfers between the server and the client. This allows for visualizing millions of spikes at multiple resolutions in a responsive manner. Figure 8 shows 5+ million spikes on a drift plot (***Rossant et al., 2021***; ***Rossant and International Brain Laboratory, 2021***).

We have implemented a web application prototype allowing users to visualize raster plots and raw data interactively. Users can view an interactive raster plot, zoom in, and dynamically select a time interval to visualize the corresponding raw data. This tool is used internally to investigate the efficacy of spike sorting algorithms and better understand limitations of spike detection and clustering.

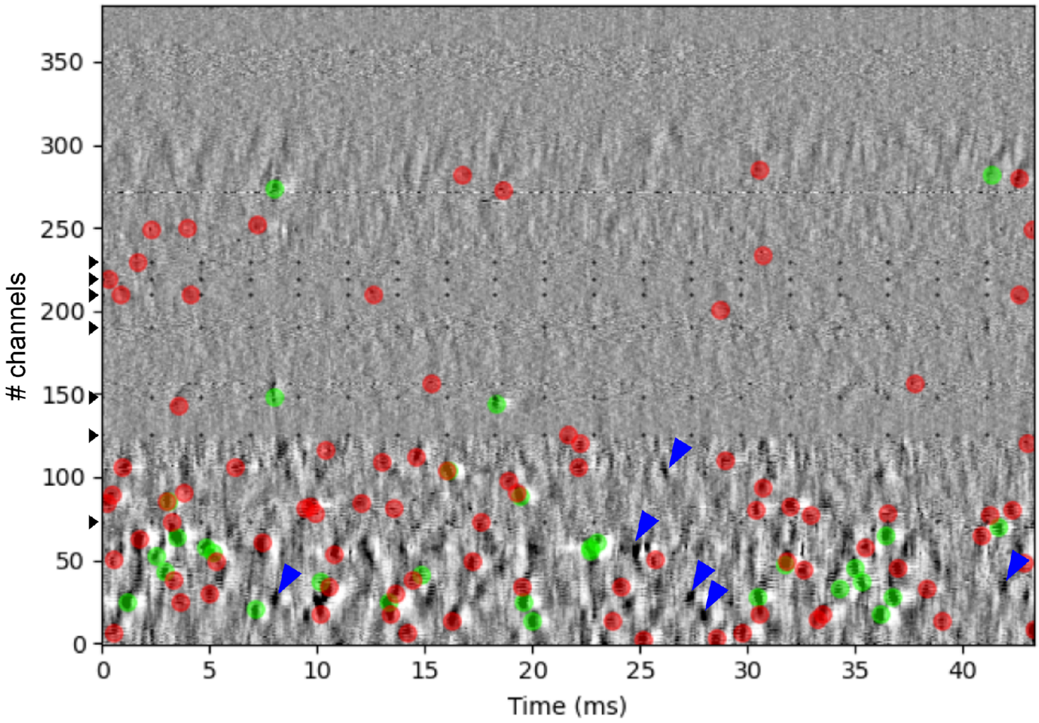

Figure 9. Missing spikes. Destriped raw voltage traces recorded on all of the channels over a short period of time (gray image, white/black boundaries set at ±50 uV), with overlaid spikes detected with our current spike sorting algorithm (green: spikes assigned to "good" clusters, red: spike assigned to "bad" clusters). In this view, a spike can be seen as a deflection from the gray color, spanning typically multiple channels at a given time. Examples of putative missed spikes are marked with blue arrows. "Bad" channels detected in the preprocessing step are marked with black dots and arrows on the left (N=7).

Caveats / known issues

Despite the significant improvements described above, a number of known issues remain; any downstream users of this data should keep these issues in mind when performing their own analyses. We expect future iterations of improved spike sorting packages to ameliorate at least some of these issues, and we plan to update our spike sorting output periodically as new techniques become available.

Under-detection of spikes

First, we see many remaining detection failures, where spikes are clearly visible in the raw data but are not detected by the sorter (Figure 9).

Non-stationary spike trains

Second, even after registration is applied, some sharp "glitches" remain visible (Figure 10), likely due to sudden motion of the probe relative to the brain. We also see larger-scale non-stationarities visible in some datasets (Figure 11). In both cases, additional data curation is likely needed before more detailed analysis of e.g. neural population dynamics or correlation structure can be pursued. Zooming in, Figure 12 shows two units displaying highly non-stationary behavior; again, such non-stationary units should be handled carefully in the context of downstream analyses.

Missing units and under/overmerges

Finally, our current pipeline still clearly misses many cells, oversplits many units, and overmerges some units. Figure 13 provides a useful overview illustration. Here we visualize the localized spikes

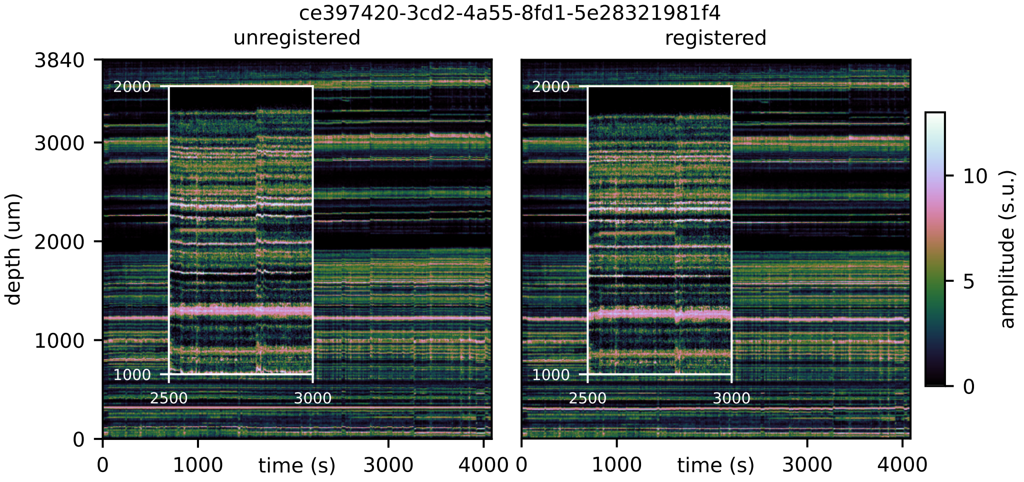

Figure 10. Motion artifacts and registration. After running detection on a full recording, the localization method of ***Boussard et al.*** (***2021***) was run (left) and registered (right) by the nonrigid decentralized method of ***Varol et al.*** (***2021***). Inset panels show sharp discontinuities and remaining challenges for registration. Activity detected at each location is colored by the spiking amplitude; green corresponds to small spikes, while pink and white correspond to larger spikes.

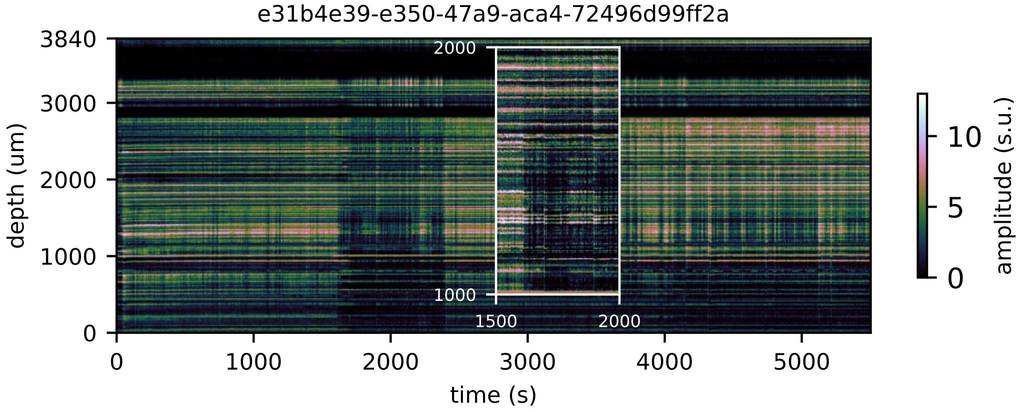

Figure 11. Nonstationarity in spiking activity. Conventions as in previous figure. Here we see a dramatic decrease in spiking activity visible across the full probe, lasting for minutes.

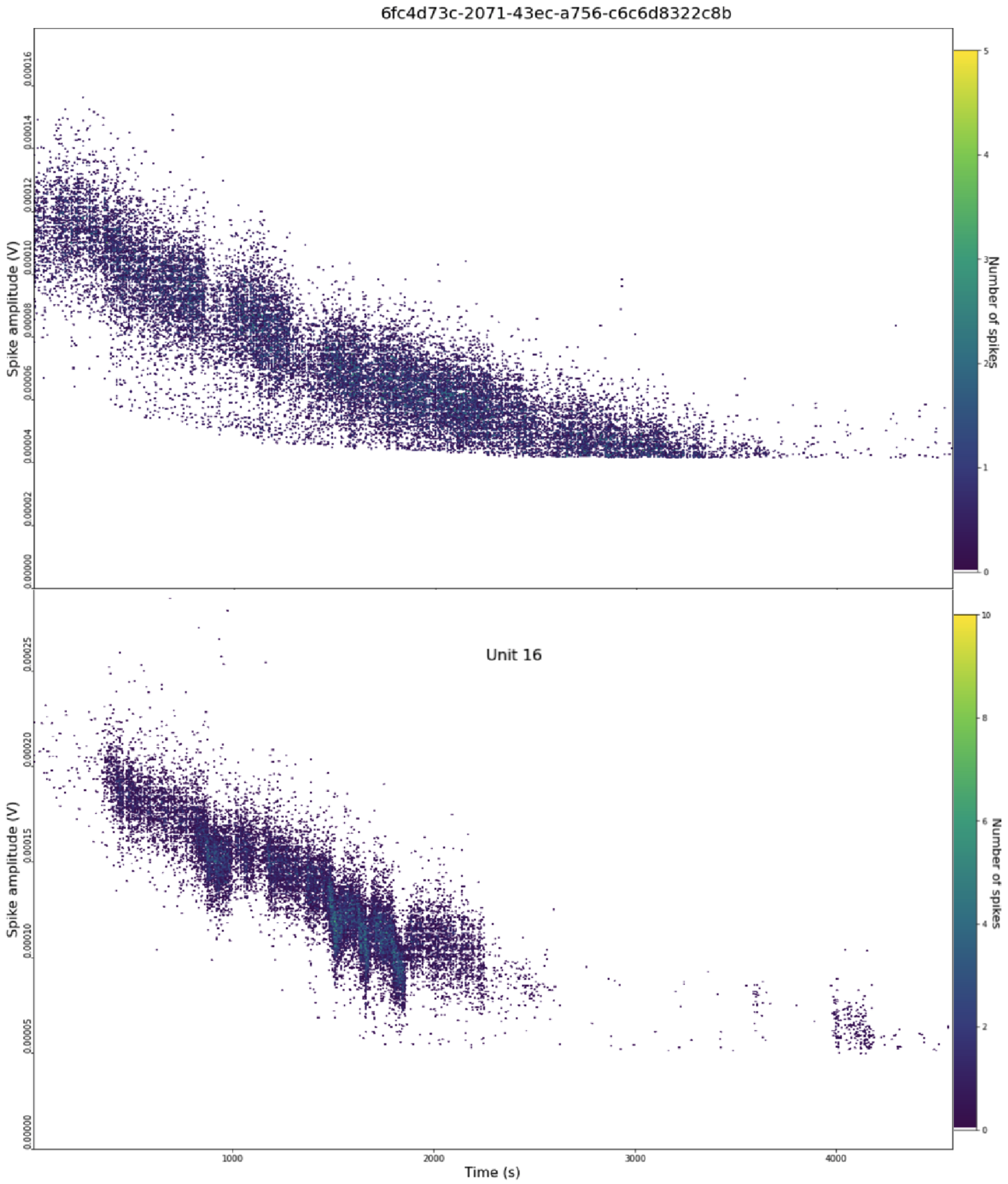

Figure 12. Two example highly-nonstationary units from recording 6fc4d73c-2071-43ec-a756-c6c6d8322c8b passing the quality metrics. The two scatter plots show the units' registered amplitudes across time, colored by density of points. The amplitudes seem to decrease until they are below the recording threshold. This might be the result of a dying neuron, or due to gradual drift away from the probe over time. Although these units are labeled as “good," it is important to develop a metric to flag them as they may introduce spurious correlations in the data.

harvested from five minutes of data (colored dots), and superimpose the locations (and amplitudes) of the templates estimated by the sorter (red x's indicate “good" units, and blue x's indicate “bad" units). A number of clear, isolated clusters are visible in the data without corresponding red x's: these spikes will be lost to downstream analyses restricted to “good" units. Conversely, many blue x's appear far from any visible clear clusters of spikes, indicating that these “bad" units may be corrupted by noise. Figure 13 also illustrates examples where clear units are missed because they are labelled as “bad" by our quality metrics.

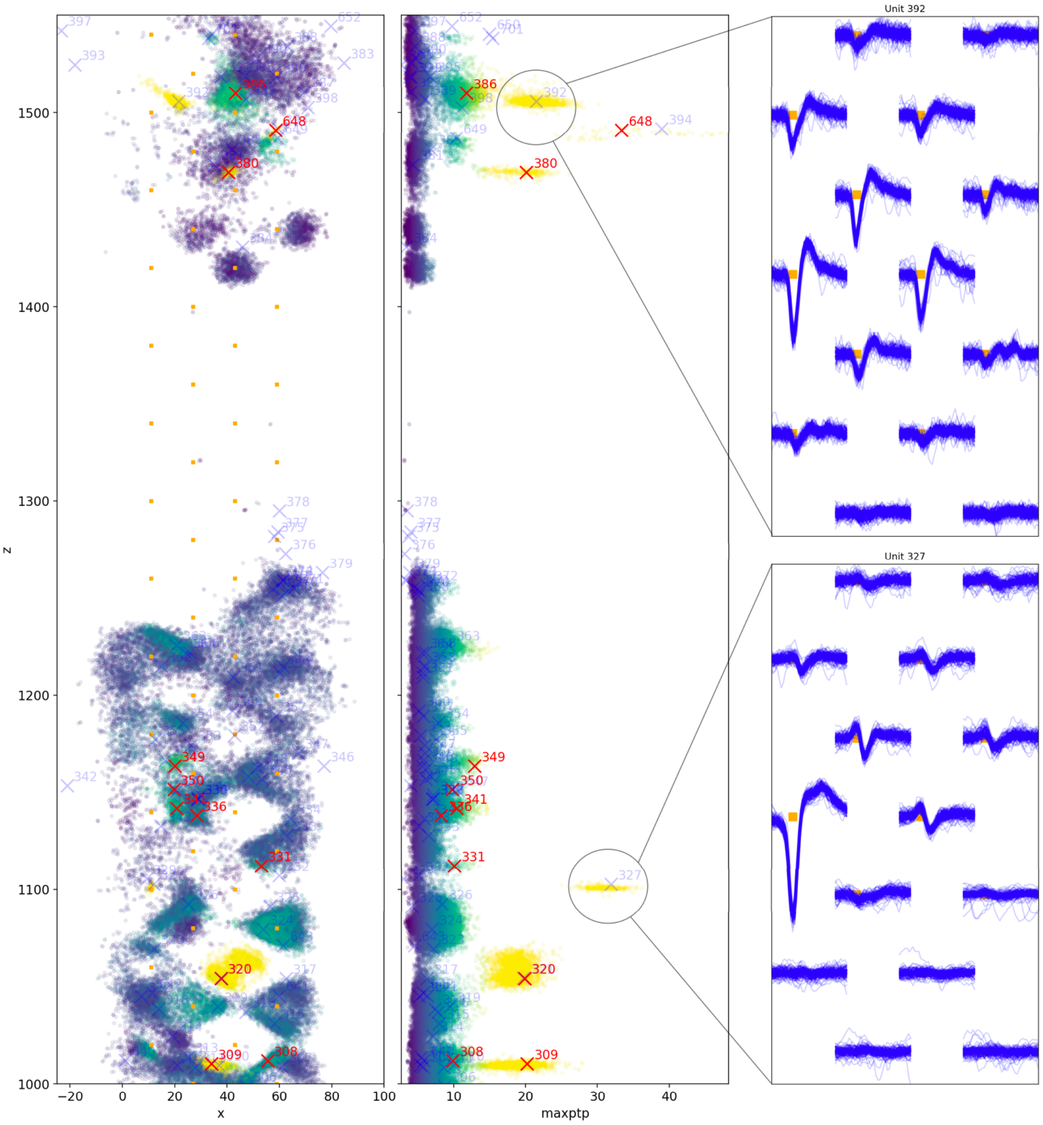

Figure 13. Example waveforms and locations of excluded units. We ran spike detection on five minutes of a NP1.0 recording. After triaging 20% of the detected spikes, we localized both the remaining detected spikes and the templates extracted from our existing pipeline, using the method developed by ***Boussard et al.*** (***2021***). All spikes are colored by their peak-to-peak and the templates are colored by whether or not they passed all of our single unit quality metrics (i.e. red=passed and blue=failed). The first panel shows 2D locations of the detected spikes and our estimated templates while the second panel shows the max peak-to-peak of the detected spikes and templates along with their inferred depths. In the third panel, we show examples of two units that were excluded from our analysis after failing our single unit quality metrics. While both units have high peak-to-peak and clear spatiotemporal footprints on the array, both failed the slidingRP_viol quality metric and were excluded for our analysis.

We also find that some “good" units are split into two units, most likely due to motion during the recording. Figure 14 illustrates an example taken from from a NP1.0 recording sorted with our processing pipeline. As can be seen in the figure, unit 438 is likely split into unit 439 and unit 682 is likely split into unit 562 due to motion at around 1000-2000 seconds into the recording. Despite these likely oversplits, units 438, 439, and 682 are labelled as good units in our analysis since our quality metrics are insensitive to oversplits. This results in oversplit units being included in our downstream single unit analysis; this issue will clearly need to be addressed in future data

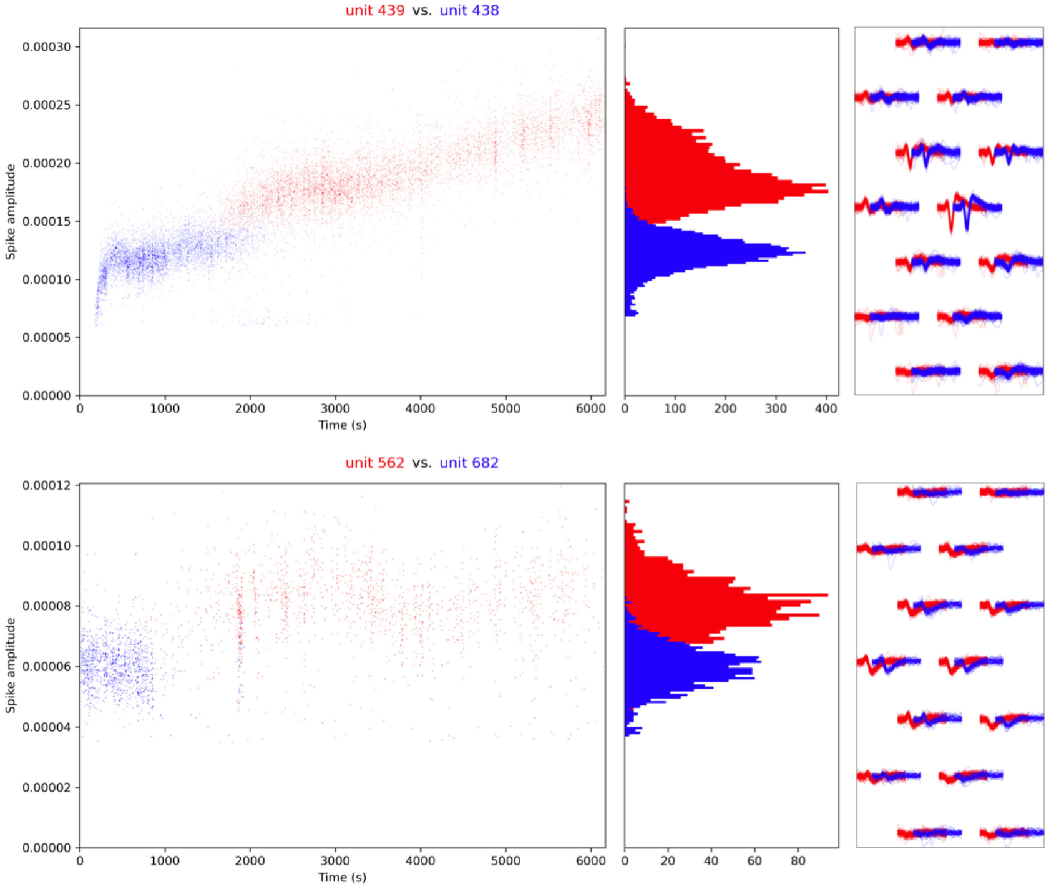

Figure 14. Amplitude and waveform plots for likely oversplit units. These data were taken from a NP1.0 recording sorted with our current processing pipeline. The far left panel includes two scatter plots that show the units' amplitudes across time, colored by unit identity. The middle panel shows the amplitude distributions, colored by unit identity. The far right panel shows the waveforms of the two units extracted from 5 minutes of the recording where the two units overlap, colored by unit identity. As can be seen in these panels, the units show a clear split due to drift that occurs from around 1000 to 2000 seconds into the recording.

releases.

Finally, we also find some examples of “good" units that are actually overmerged, i.e., two or more units combined into one unit. Figure 15 illustrates an example from the same NP1.0 recording which we used to visualize the above oversplit units. In this figure, we see that unit 648 likely consists of at least two units with different amplitudes. Again, our current quality metrics are insensitive to this bimodality and unit 648 is included in our downstream single unit analysis.

Operational challenges at the IBL scale

Projects like the IBL brain-wide recording, representing thousands of recordings across a dozen labs and spanning several years, pose several challenges.

Running a spike sorting algorithm on a thousand recordings will find some edge cases and performance issues that have not been encountered before. For this extensive work has been done to stabilize CPU and GPU memory leaks for the clustering phase, as large recordings ran frequently out of memory.

Then, spike sorting algorithms are still evolving continuously, which poses the challenges of data continuity and versioning. A pre-requisite to manage this is implement software engineering best practices for the spike sorting algorithm. We perform testing at several levels: unit tests were

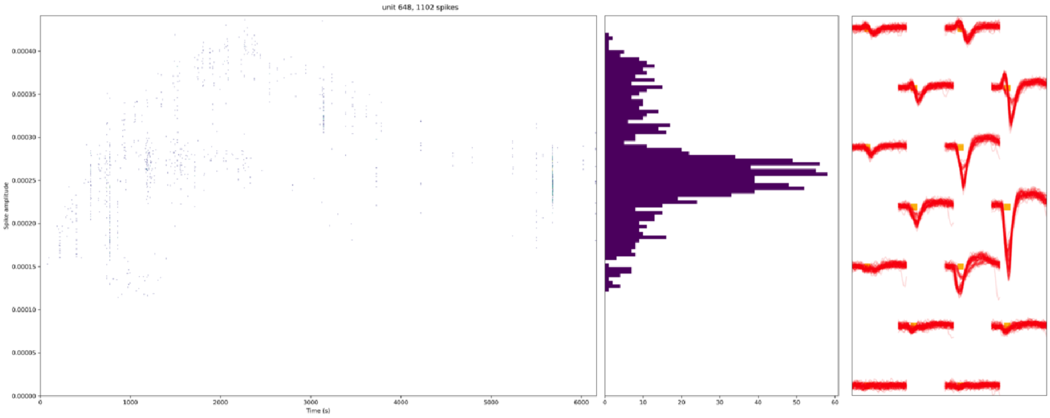

Figure 15. Amplitude and waveform plot for a likely overmerged unit. These data were taken from a NP1.0 recording sorted with our current processing pipeline. The left panel shows a scatter plot of unit 648's amplitudes across time, colored by density of points. The right shows the waveforms of unit 648 extracted from a 5 minute snippet of the recording. As can be seen in these two panels, unit 648 likely consists of at least two units with different amplitudes.

added to enforce behaviour at the small-scale function or method level, typically for GPU functions. Integration tests using benchmark datasets are used to make sure the whole pipeline has an appropriate output and behaviour.

Finally, output datasets should be tagged with the spike sorter version used. At IBL the version is stored in a dataset table in a relational database, and several versions of spike sorting may be available for a given spike sorting run (***International Brain Laboratory et al., 2020***). The versions are also stored in a companion log file with the spike sorting datasets.

References

Bar-Gad, I., Ritov, Y., and Bergman, H. (2001). The neuronal refractory period causes a short-term peak in the autocorrelation function. *Journal of Neuroscience Methods*, 104(2):155–163.

Boussard, J., Varol, E., Lee, H. D., Dethe, N., and Paninski, L. (2021). Three-dimensional spike localization and improved motion correction for neuropixels recordings. *Neurips*.

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.

Hill, D. N., Mehta, S. B., and Kleinfeld, D. (2011). Quality metrics to accompany spike sorting of extracellular signals. *Journal of Neuroscience*, 31(24):8699–8705.

International Brain Laboratory, T. (2021a). ibl-neuropixel. https://github.com/int-brain-lab/ibl-neuropixel.

International Brain Laboratory, T. (2021b). mtscomp. https://github.com/int-brain-lab/mtscomp.

International Brain Laboratory, T., Banga, K., Faulkner, M., Rossant, C., and Winter, O. (2021). pykilosort. https://github.com/int-brain-lab/pykilosort.

International Brain Laboratory, T., Bonacchi, N., Chapuis, G., Churchland, A., Harris, K. D., Hunter, M., Rossant, C., Sasaki, M., Shen, S., Steinmetz, N. A., Walker, E. Y., Winter, O., and Wells, M. (2020). Data architecture for a large-scale neuroscience collaboration. *bioRxiv*.

Jun, J. J., Steinmetz, N. A., Siegle, J. H., Denman, D. J., Bauza, M., Barbarits, B., Lee, A. K., Anastassiou, C. A., Andrei, A., Aydın, Ç., et al. (2017). Fully integrated silicon probes for high-density recording of neural activity. *Nature*, 551(7679):232–236.

Rossant, C. and International Brain Laboratory, T. (2021). Datoviz. https://datoviz.org/.

Rossant, C., Rougier, N. P., et al. (2021). High-performance interactive scientific visualization with datoviz via the vulkan low-level gpu api. *Authorea Preprints*.

Siegle, J. H., López, A. C., Patel, Y. A., Abramov, K., Ohayon, S., and Voigts, J. (2017). Open ephys: an open-source, plugin-based platform for multichannel electrophysiology. *Journal of neural engineering*, 14(4):045003.

Steinmetz, N. A., Aydin, C., Lebedeva, A., Okun, M., Pachitariu, M., Bauza, M., Beau, M., Bhagat, J., Böhm, C., Broux, M., et al. (2021). Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings. *Science*, 372(6539):eabf4588.

Sukiban, J., Voges, N., Dembek, T. A., Pauli, R., Visser-Vandewalle, V., Denker, M., Weber, I., Timmermann, L., and Grün, S. (2019). Evaluation of spike sorting algorithms: Application to human subthalamic nucleus recordings and simulations. *Neuroscience*, 414:168–185.

Varol, E., Boussard, J., Dethe, N., Winter, O., Urai, A., Brain Laboratory, T. I., Churchland, A., Steinmetz, N., and Paninski, L. (2021). Decentralized motion inference and registration of neuropixel data. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1085–1089.

Welch, P. (1967). The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73.