Power Pixels: a turnkey pipeline for processing of Neuropixel recordings

Guido T. Meijer[1] and Francesco P. Battaglia[1]

https://doi.org/10.1101/2025.06.27.661890

Abstract

There are many open-source tools available for the processing of neuronal data acquired using Neuropixels probes. Each of these tools, focuses on a part of the process from raw data to single neuron activity. For example, SpikeInterface is an incredibly useful Python module for pre-processing and spike sorting of individual recordings. However, there are more steps in between raw data and spikes, such as synchronization of spike times between probes and histological reconstruction of probe insertions. Therefore, we developed Power Pixels, combining the functionality of several packages into one integrated pipeline, which may be run in any lab workflow. It includes pre-processing, spike sorting, neuron-level quality control metrics, synchronization between multiple probes, compression of raw data, and ephys-to-histology alignment. Integrating all these steps into one pipeline greatly simplifies Neuropixels data processing, especially for novel users who might struggle to find their way around all the available code and tools.

[1]Donders Center for Neuroscience, Radboud University, Nijmegen, The Netherlands

Correspondence
guido.meijer@donders.ru.nl

Introduction

The processing of electrophysiological recordings, from raw data to spike-sorted neurons in defined brain regions, is a very labor-intensive task for many researchers. This in particular with the advent of high-density silicon probe, such as Neuropixels (Jun et al., 2017). Large collaborations, like the International Brain Laboratory (IBL), have developed fully integrated end-to-end pipelines, this takes the workload away from the researchers and allows them to spend more time doing experiments, analysis and writing. These pipelines, however, are embedded in the database architecture used by the collaboration and do not run out-of-the-box on recordings made by external researchers. (IBL et al., 2022). Even when the code base is open-source, it's challenging for external researchers to apply the code to their data because the code is written for large-scale deployment instead of running a single local recording session. Furthermore, collaborations might have access to advanced equipment which is typically not available to individual labs.

Advances are made to streamline and standardize processing and spike sorting of electrophysiological recordings, most notably by SpikeInterface (Buccino et al., 2020). SpikeInterface is a powerful tool which is aimed at the first stages of processing: preprocessing, spike sorting, and curation of the spike sorting output. To get a full end-to-end pipeline, however, several additional steps are required, namely: synchronization between multiple probes, histological reconstruction of probe tracts, and alignment of histology to electrophysiological markers. We present here the Power Pixels pipeline, covering all these steps and additionally calculates several different neuron-level Quality Control (QC) metrics which can help, or even completely replace, manual curation of spike sorting output. In short, the Power Pixels pipeline combines SpikeInterface, the IBL pipeline, and probe tracing software into one end-to-end pipeline from raw recorded data to spike sorted neurons in defined anatomical regions in the brain.

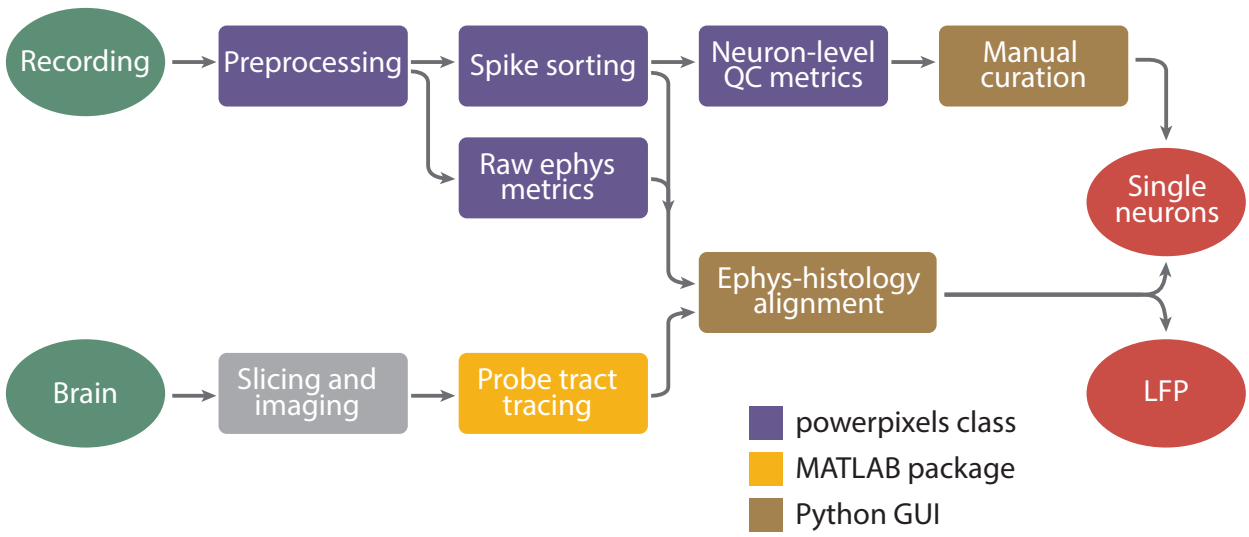

Figure 1 – Process diagram of the pipeline. The inputs are the raw recording and the perfused brain and the outputs are single neurons and LFP power, in defined brain regions.

Methods

The Power Pixels pipeline combines processing steps from SpikeInterface, elements from the IBL pipeline, and a standalone MATLAB package into one end-to-end pipeline (Figure 1). The pipeline takes raw data electrophysiological time series as input and generates curated spike-sorted neurons, along with the anatomical location of the recorded neurons as output. The backbone of the pipeline is the *powerpixels* Python class that contains several functions related to the different steps of the pipeline (Table 1). To establish in which brain region each recording channel ended up, histological reconstruction of the probe tracts is done using the MATLAB package Universal Probe Finder (Montijn and Heimel, 2022). Finally, the electrophysiological signatures along the probe are aligned with the inferred brain regions from the probe tract using the Ephys-Histology Alignment GUI developed by the IBL.

Preprocessing

Long-shanked, high-density silicon probes, such as Neuropixels, pose specific challenges for data preprocessing and the achievement of high-quality spike sorting. The Power Pixel pipeline recapitulates the steps in the pipeline used by the International Brain Laboratory (IBL et al., 2022).

Optional manual step: notch filters. Recordings sometimes have high frequency noise in specific frequency bands. To check if this is the case the user can use the *visualize_preprocessing.ipynb* Jupyter notebook to load in the raw data, this notebook will perform a power spectral analysis in which the user can see whether high frequency noise is present as peaks in the power spectrum. The user can then decide to include one or several notch filters, the notebook contains interactive plots which show the effect of these filters. To apply the notch filters the user can modify the example *notch_filter.json* file from the repository and copy it to the folder containing the raw data. The pipeline will read in the JSON file an automatically apply the notch filters at the specified frequencies.

Inter-sample shift correction. On a Neuropixel probe, the 384 active channels are not sampled exactly simultaneously. During one sample acquisition, data from 12 channels banks are acquired in succession, resulting in a shift of 1/12th of a sampling interval. At a sampling rate of 30 kS/s the inter-sample shift amounts to $\sim$ 30 microseconds. These shifts, albeit minuscule, can cause artifacts while applying a common-average reference over the entire probe. Therefore, the first step in the pipeline is to correct for these shifts.

Detect and remove channels. Before performing any common average referencing the channels that are noisy, or outside of the brain, should be removed. This prevents these signals from contaminating the reference signal. First, the channels outside of the brain are detected and removed; subsequently, the noisy channels are detected and interpolated over neighboring channels. This is done to ensure spatial homogeneity which is required for the destriping step in the next step.

Artifact removal. Electrical artifacts can have many different sources, but they share one feature: their signal is the same over nearby channels. Single-shank probes, however, are so long that electrical artifacts can have slightly different amplitudes at the tip versus at the base of the probe. Therefore, performing a common average reference, whereby the median of all channels

powerpixels	**Description**	**By**
.restructure_files	Move files and rename folders	N/A
.nidq_synchronization	Create nidq synchronization files	IBL pipeline
.preprocessing	Preprocess raw data	SpikeInterface & IBL
.spikesorting	Spike sort recording	Kilosort & SpikeInterface
.neuron_metrics	Compute neuron-level QC metrics	SpikeInterface
.raw_ephys_qc	Compute raw ephys QC metrics	IBL
.convert_to_alf	Convert Kilosort output to ONE data standard	IBL
.automatic_curation	Curate units into single and MUA	IBL & SpikeInterface
.probe_synchronization	Synchronize spike times to the nidq clock	IBL
.compress_raw_data	Compress raw .bin files	IBL

Table 1 – Description of the functions of the *powerpixels* class.

is subtracted from each channel, is often not sufficient to filter out the artifact. Instead, the pipeline employs a spatial filtering method developed by the IBL called *destriping*. This method takes into account that artifacts can vary smoothly over the length of the probe. In contrast to single-shank probes, four-shank probes cluster all their active sites at the tips of the shanks. Because the active part of each shank is limited in length, there is no need for destriping, and a common median reference is used.

Spike sorting

The use of Kilosort (any version) is recommended for spike sorting because subsequent steps of the pipeline are specifically written to ingest the output of Kilosort (Pachitariu et al., 2024, 2016). For example, the conversion script which converts the sorting output into a standardized data format, which can be read in by the alignment GUI, is written specifically for the Kilosort output format. Files with the recommended parameters for Kilosort 2.5, 3, & 4, and pykilosort are included in the pipeline repository but can be changed if necessary. Spike sorting is run by SpikeInterface which allows seamless integration in the pipeline, MATLAB-based sorters (such as Kilosort 2.5 and 3) are deployed in a Docker container, alternatively the user can install Python-based Kilosort 4 locally.

Neuron-level QC metrics

Modern spike sorters, like Kilosort, tend to be overzealous when detecting units. Only a subset of the detected units are well isolated single units, with the majority of detected units being multi-unit activity or noise. Therefore, after spike sorting has detected putative single units, quality metrics are calculated for these units. These metrics give an indication of how clean the unit is, for example, by computing the number of inter-spike interval violations. On the other hand, metrics can also be used to characterize units, for example by looking at the spike width to classify units as putative narrow-spiking interneurons.

Manual and automatic curation of spike sorting output

To make a final decision on which units constitute single neurons, most electrophysiologists manually curate their data. This entails visual inspection of the waveforms, interspike interval distribution, spike amplitudes, etc. Power Pixels computes many of these quality metrics to provide

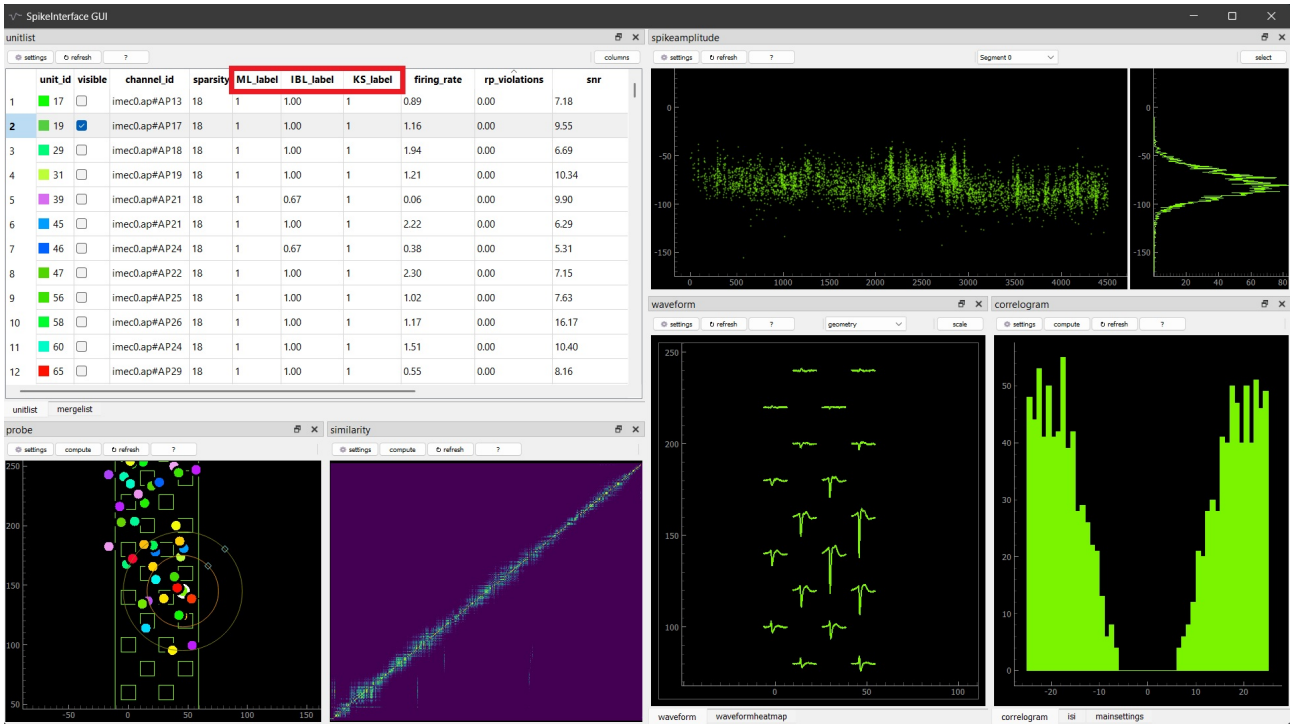

Figure 2 – The manual curation GUI of SpikeInterface (developed by Samuel Garcia) with added predictions of which units are well isolated single units from various sources. **ML_label**: prediction from machine learning model UnitRefine (1 = single neuron, 0 = multi-unit activity). **IBL_label**: neuron-level quality control from the IBL (1 = single neuron, <1 = multi-unit activity). **KS_label**: quality indicator from Kilosort itself (1 = single neuron, 0 = multi-unit activity).

the user with as much information as possible upon which to make this decision. The curation GUI of SpikeInterface is used to this purpose. Besides manual curation, however, there are now more and more algorithms which aim to automate this process. The pipeline runs two of these algorithms: one developed by the IBL (IBL et al., 2022) and UnitRefine: a machine learning algorithm trained on spike sorted sessions of other Neuropixels recordings (Jain et al., 2025). The predictions of these algorithms are compiled, together with the indication from Kilosort, and added to the SpikeInterface manual curation GUI (Figure 2). The user can use this information to make a better choice as to which units are good single neurons. Alternatively, one can use the predictions of one of these automatic curation algorithms and skip the manual curation step altogether.

Raw electrophysiology metrics

Besides spiking activity, other information can be extracted from the raw electrophysiological traces of each channel. From the low-frequency component of the raw voltage trace, the local field potential (LFP), the power in different frequency bands is computed using spectral decomposition. Furthermore, the root-mean square (RMS) of the high-frequency component is computed.

Histology

Another important part of the Power Pixels pipeline is to determine which brain region each recording channel is sampled from. To this end, the experimenter should label their probes with a

fluorescent dye (typically Di-I [V22888, ThermoFisher Scientific]) before insertion into the brain. After multiple recording penetrations are carried out, the experimenter will typically perfuse the animal, extract the brain, slice the brain, mount the brain slices on slides, and image these slides with a fluorescence microscope. This results in a collection of coronal slices in which the fluorescence tract of the probe is visible. The next step is to map the slices to the Allen Brain atlas and trace the fluorescence tract in these registered slices.

To our knowledge, there is no Python package available that can be used to do this. The Python-based option used by the IBL called lasagna, only works by registering the full 3D brain stack to the Allen Atlas. This requires whole-brain imaging using a serial-sectioning two-photon scanner or a light sheet fluorescence microscope. Both devices are highly expensive and will not be available to the typical experimenter. Therefore, the pipeline relies on Universal Probe Finder, a MATLAB package specifically designed to manually align histological slices to the Allen Brain Atlas and trace probe tracts there (Montijn and Heimel, 2022). After the experimenter has done this, the pipeline includes a conversion function (*convert_histology_to_alignment_GUI.m*) to convert the coordinates of the probe trajectories from Universal Probe Finder to a format that can be read by the Ephys-Histology Alignment GUI developed by the IBL.

Ephys-Histology alignment

Probe tracts in the brain can be well defined in terms of their insertion vector using the method described above. However, their precise depth cannot be reliably determined solely on the basis of the fluorescence tract. This is for several reasons: (1) the fluorescent dye diffuses at the tip making it hard to determine where the tip exactly was, (2) the experimenter might pull back the probe slightly before recording to increase recording stability, and (3) the force exerted on the brain by the probe can result in non-homogeneous compression of the brain tissue. Therefore, to establish which recording channel was in which brain region, the brain regions along the insertion vector need to be aligned with the electrophysiological markers recorded by the probe. For example, LFP power is high in the dentate gyrus but low in fiber bundles. Also, cross-correlations between neurons are expected to be higher within brain regions compared to across. To do this alignment, the ephys-histology alignment GUI developed by the IBL will be used (documentation can be found here). The GUI allows the experimenter to move, stretch and squeeze the brain region until they fit with the recorded electrophysiological markers.

Synchronization

When using multiple Neuropixel probes and / or when using a BNC breakout board (to record timestamps from behavioral events), the spike times need to be synchronized between the probes and with the event times. This is because each probe, and the BNC breakout board, has its own internal clock. These clocks can show small shifts over time which need to be corrected. To this end, the Neuropixels PXI basestation generates a 1s square wave. This synchronization signal is automatically routed to the headstages of all the probes, the (optional) BNC breakout board should receive the square wave by connecting the SMA connector of the PXI to one of the digital inputs of the breakout board (Figure 3). Alternatively, one can generate a custom synchronization pulse using an Arduino and use that for synchronization. For more details and instructions see the documentation here and here. The pipeline uses code from the IBL to synchronize these data streams and adjust the detected spike times accordingly.

A

```
{
    "SYSTEM": "3B",
    "SYNC_WIRING_DIGITAL": {
        "P0.0": "imec_sync",
        "P0.1": "lick_detector",
        "P0.2": "camera"
    },
    "SYNC_WIRING_ANALOG": {
        "AI0": "breathing_sensor"
    }
}
```

B

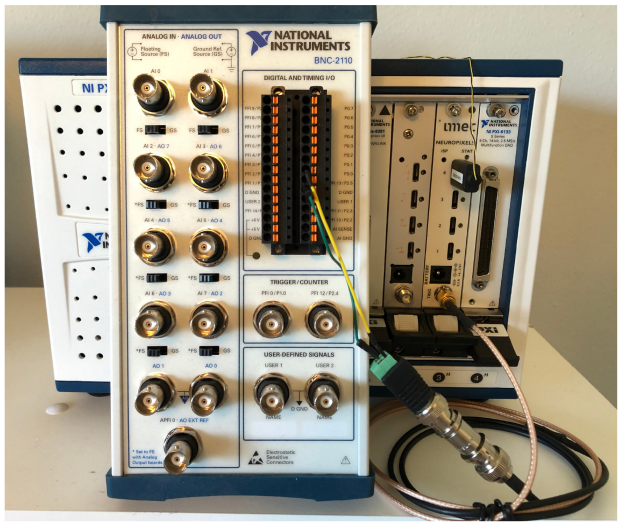

Figure 3 – (A) Example JSON file describing the wiring of the BNC breakout box. In this example the 1s square wave pulse from the PXI chassis is relayed to the first digital channel (P0.0). The next two digital channels are used to record timestamps of licks and camera frames. The first analog channel is used to record a continuous signal from a breathing sensor. (B) A picture of the BNC breakout box (BNC-2110, National Instruments) which receives the synchronization pulse from the PXI chassis as an input in the first digital channel (P0.0). Picture credits Open Ephys.

Data standard

The output of the pipeline is in the ALF data standard, which is the standard used by the International Brain Laboratory. The advantage of this conversion is that the data can be easily shared using the Open Neurophysiology Environment (ONE). Furthermore, having the data in this format allows one to use all the functions from the Brainbox library, which is part of ibllib. These functions include everything from plotting of peri-stimulus time histograms to decoding. If the experimenter prefers the Neurodata Without Borders (NWB) format, conversion functions are readily available on Github. For a detailed explanation of all the dataset types found in the output folder refer to this table.

Conclusion

The Power Pixels pipeline provides researchers the same integrated experience, while processing Neuropixel recordings, as those working in large collaborations. It uses SpikeInterface and the IBL codebase to mimic the tried-and-tested pipeline used by the International Brain Laboratory. This pipeline is especially useful for researchers who are new to Neuropixel recordings and might not have extensive insight into which choices to make during the processing of these recordings. After initial setup has been done the pipeline runs almost completely automatically through all the preprocessing steps and the spike sorting. The only preprocessing step which requires manual intervention is determining whether high frequency noise is present in the recording which needs to be filtered out with targeted notch filters. The biggest manual effort is tracing the probe tracts through the histological brain slices and subsequently aligning electrophysiological features to the brain regions determined by histology. To conclude, the Power Pixels pipeline requires minimal human intervention to go from raw data to spike sorted single neurons in specified brain regions.

Code availability

The Power Pixels pipeline code is available on Github: https://github.com/NeuroNetMem/PowerPixelsPipeline/

Acknowledgments

We thank all the developers and contributors to the following open-source projects: SpikeInterface (special thanks to Alessio Buccino and Samuel Garcia), The International Brain Laboratory (special thanks to Oliver Winter and Mayo Faulkner), Universal Probe Finder (Jorrit Montijn), and Kilosort. This work was supported by the Dutch Research Council (NWO) through grant number OCENW.XL21.XL21.069.

Conflict of interest disclosure

The author of this preprint declares to have no financial conflict of interest with the content of this article.

References

Buccino AP, Hurwitz CL, Garcia S, Magland J, Siegle JH, Hurwitz R, Hennig MH (2020). SpikeInterface, a unified framework for spike sorting. *eLife* **9**. Ed. by Laura L Colgin, Sonja Grün, and Fabian Kloosterman, e61834. https://doi.org/10.7554/eLife.61834.

IBL, Banga K, Boussard J, Chapius G, Faulkner M, Harris KD, Huntenburg J, Hurwitz C, Dong Lee H, Paninski L, Rossant C, Roth N, Steinmetz NA, Windolf C, Winter O (2022). Spike sorting pipeline for the International Brain Laboratory. online resource. https://doi.org/10.6084/m9.figshare.19705522.v3.

Jain A, Greene R, Halcrow C, Swann JA, Kleinjohann A, Spurio F, Graff S, Kampa B, Gall J, Grün S, Buccino A, Hennig MH, Musall S (2025). UnitRefine: A Community Toolbox for Automated Spike Sorting Curation. Pages: 2025.03.30.645770 Section: New Results. https://doi.org/10.1101/2025.03.30.645770.

Jun JJ, Steinmetz NA, Siegle JH, Denman DJ, Bauza M, Barbarits B, Lee AK, Anastassiou CA, Andrei A, Aydın Ç, Barbic M, Blanche TJ, Bonin V, Couto J, Dutta B, Gratiy SL, Gutnisky DA, Häusser M, Karsh B, Ledochowitsch P, et al. (2017). Fully integrated silicon probes for high-density recording of neural activity. *Nature* **551**, 232–236. https://doi.org/10.1038/nature24636.

Montijn JS, Heimel JA (2022). A universal pipeline for the alignment of electrode tracks with slice histology and electrophysiological data. https://doi.org/10.1101/2022.06.20.496782.

Pachitariu M, Sridhar S, Pennington J, Stringer C (2024). Spike sorting with Kilosort4. *Nature Methods* **21**. Publisher: Nature Publishing Group, 914–921. https://doi.org/10.1038/s41592-024-02232-7.

Pachitariu M, Steinmetz N, Kadir S, Carandini M, D HK (2016). Kilosort: realtime spike-sorting for extracellular electrophysiology with hundreds of channels. https://doi.org/10.1101/061481.