Proposal for web-based spike sorting validation

Alex Barnett, Jeremy Magland, and James Jun

Flatiron Institute, April 10, 2018

Abstract

We lay out some choices for metrics and datasets for a community effort to compare and validate the major spike sorting packages, hosted at Flatiron. This includes summarizing and extending some ideas from the discussion at the Janelia spike sorting meeting of 3/22/18. It also includes discussion and some subjective opinions. This document will be open to suggestions from the electrophysiology community.

1 Background and goals

We are in the process of hosting at Flatiron a web-based platform where a variety of spike-sorting packages are run on a set of community-approved ground-truthed datasets, and their performance metrics made publicly available online via an interactive graphical front-end, and probably also via an API.

The main goal is an objective accuracy comparison of the major current spike sorting codes. This should indicate for the electrophysiology community the best code to use, and its expected accuracy (or distribution of accuracies across units), both of which may depend on context (probe type, in/ex vivo, etc). A best code choice may also depend on your computer resources (RAM, GPU, etc).

A meta-goal is to gather information about which *quality metrics*, meaning metrics computable without ground-truth data, most closely indicate ground-truth accuracy when it is available.

Curating and gathering such datasets and metrics is a community effort. Please give feedback and/or additions to this document. There are three ways to comment (in decending order of convenience for us): 1) post a git issue to this repository; 2) use the gitter chat room for this repository; 3) make a pull request with changes to this document.

possibly via margin comments like this

1.1 Abbreviations

GT	ground-truth(ed)
AC	auto-correlation of firing events for a single sorted unit
CC	cross-correlation of firings between different sorted units
(BP)CM	(best-permuted) confusion matrix [2]
e-phys	electrophysiological

1.2 Mathematical symbols

M	number of channels (electrodes)
T	number of time points (samples)
X	M-by-T matrix of voltages at each time point, with elements x_{mn}
L	number of firing events output by a sorter
K	number of units found by a spike sorter
t_j	time of the jth firing event
k_j	label (ie, unit number or classification) of the jth firing event, in the range $1, \ldots, K$
m_j	peak electrode number (in the range 1 to M) of the jth firing event
Q_{kl}	confusion matrix element, equal to the number of time-matching events labeled k in sorting 1 and l in sorting 2

2 Data and web interfaces

One cannot proceed without a standard *interface* between algorithms (codes) and e-phys datasets (recorded inputs and sorted outputs). Deciding on a commonly agreed-upon file format for e-phys datasets and the corresponding metadata is not an easy task. However, this is simplified when we split it into two levels: the "mathematical" interface (ie the list of inputs and outputs), distinct from the *implementation* of this interface (ie particular file formats).

We propose that the mathematical interface for spike sorting be (see notation in Sec. 1.2):

Inputs:	X: Raw (preferably unfiltered) voltage recording, as a matrix of size M (number of channels) by T (number of time-points).
Output:	Lists of firing times t_j and labels (unit or cluster assignments) k_j for each found event $j = 1, \ldots, L$. Time is given in units of samples; the labels are integers between 1 and K. We choose to require a third output: the list of electrode numbers m_j on which each event peaks (integers between 1 and M). Events $j = 1, \ldots, L$ need not be ordered in time, but should obviously be consistently ordered in the above three lists.

Notes:

1. The minimal interface to a spike sorter, that allows any kind of algorithm (including ones with very different internal mechanics such as ICA that do not have a detection stage), is input X and outputs $\{t_j\}$ and $\{k_j\}$ [2]. All quantities of interest to display in a web-based exploration of the sorting result—in particular mean spike waveforms—can be derived from the union of this input and output data.

   Benchmarking stages such as detection was discussed, but is a separate task that we do not tackle here.

2. For convenience of validation in the many-electrode setting, we also propose to require the third output $\{m_j\}$. While this output could easily be generated from the other two, by computing the mean waveforms and then finding their maximum over time and channel, most algorithms already provide this data, and it is useful to avoid its recomputation. We will provide a utility that computes it as part of wrapping algorithms that do not provide it.

3. Probabilistic outputs were discussed at the meeting; however, there are currently few if any spike sorters that produce such outputs. The above interface could also handle a sorter giving probabilistic outputs by drawing ~ 20 independent samples from it and recomputing all metrics for each sample, giving distributions of each metric.

2.1 Formats for input dataset

The raw voltages X (multi-channel timeseries arrays) should be stored in simply binary format, with channels stored contiguously for each timepoint, ie column-major order for the matrix X. (Various formats were discussed at Janelia, but it was agreed that avoiding time-consuming conversions between formats was essential.) The accompanying metadata that the sorter has access to will specify M the number of channels, the sample rate (Hz), the duration T, and the raw datatype (e.g. `int16`, `float32`, etc). The metadata should also contain a list of two-dimension coordinates of the electrodes, preferably in μm units. We are aware that many "raw" voltages are already filtered.

We are not sure whether filtering information should be part of the metadata that the sorter sees

We plan to write wrappers to convert other binary formats to the above. We may need to accommodate the sitation where a large timeseries is stored in a sequence of files to be concatenated.

2.2 Format for sorted output

We propose that the above three lists be output in a single 3-by-L matrix:

- row 1: the peak electrodes m_j
- row 2: the firing times t_j, in units of samples where the first sample is 1 and the last T. These should allow real numbers (and should be double-precision since $T > 10^8$ timepoints is common).
- row 3: the labels k_j

Since the middle row requires a 64-bit format, we propose that the other two rows also be 64-bit double-precision, which will be rounded to integers.

For sorters that do not produce this output, we plan to write simple format converter utilities.

2.3 Hosting of ground-truthed datasets

We already have in place a mechanism for uploading and sharing datasets. We propose to host standard datasets, in two categories: official datasets that metrics are computed on (see the next section), and unofficial test datasets that are needed for development and testing. Each new candidate for a standard dataset should first be uploaded, viewed and tested before being included in the standard set.

2.4 Hosting of sorting algorithms and outputs

One discussion at the meeting was whether to upload *algorithms* (codes) or *sorted data*; the consensus at the meeting seemed to be both. (Algorithms are harder to upload, but allow the website to show more complete information such as runtime, RAM usage, etc; they also reduce the potential for hand-tweaking of results.)

Algorithms. If a sorting algorithm is wrapped and registered as a MountainLab processor (see `https://github.com/flatironinstitute/mountainlab-js`), then the execution of that algorithm against the standard datasets may be automatically included in a nighly spike sorting run. We have wrapped several major algorithms this way.

All algorithms benchmarked have to be *fully automatic*, come with some *meta-data* such the parameter set, version number, etc. They should be cleanly separated from any visualization/curation/GUI tools. Any curation or post-processing (eg based on their internal quality metrics) should be automatic and included with the code.

All wrapped procedures would ideally operate on data in the standard agreed-upon format and output results in the standard format described above. However, we will also provide tools for converting between formats to accommodate different file conventions.

Sorting outputs. We also plan to allow direct uploading of spike sorting results in the format described above.

Other ideas discussed at the meeting that we do not have plans to implement yet include: 1) held-out or *hidden* data in the style of the NetFlix Challenge. 2) running codes from an arbitrary github repo with a dockerfile (which will allow groups to automatically update their algorithms).

3 Datasets

It was agreed that a variety of brain regions, recording conditions, probes, and types of data are needed. Here are some datasets, most of which were discussed at the meeting. They fall into three categories.

1. **Recordings with ground-truth.**

   Good features: gold-standard. Bad features: very small sample size, not in awake animals or various regions yet. [1]

   - Neto, Kampff et al. '16. [10] 32-channel and 128-channel (20 μm pitch) with single juxtacellular GT, in vivo rat, anesthetized, motor, sensory or parietal cortex, 30 kHz, roughly 10 min long. `http://www.kampff-lab.org/validating-electrodes/`

     We propose to include:

     `2014-11-25_Pair 3.0` : 32-channel, GT 52 μm from electrode, $N_{\text{true}} = 347$.

     `2015_09_03_Cell.9.0` : 128-channel, GT 29 μm from electrode (this is a bursting pair as discussed in [3]), $N_{\text{true}} = 4895$.

     These appear to be the only datasets where the GT unit can be viably sorted. The GT units are very easy to sort, hence this might not be useful in differentiating algorithms.

   - Yger et al. '18. [13] 252-channel (16×16 array, 30 μm), w/ loose patch, in vitro, mouse retina. Around 20 recordings of typical length 5 min, each with one GT unit with typically 500-5000 firings. Varying SNR of the GT units.

     `http://www.yger.net/software/ground-truth-recordings/`

     Subset at `https://dio.org/10.5281/zenodo.1205233`

     To do: decide if all or a subset of datasets should be included.

     The paper discusses at least 37 neurons; unsure how many GT are available.

   - Franke et al. '15 [5]. Tetrode with *dual* patch-clamp GT units, ex vivo rat cortex. 33.3kHz. Includes 25000 spikes with overlaps of less than 1.5 ms induced by current injections, designed to test overlapping (colliding) spikes. To do: write to authors and ask for data.

   - Bryan Allen [1] and Caroline Moore-Kochlachs [9] dataset from Boyden lab. Dense array, 11 μm.

   - Simultaneous calcium imaging and e-phys (Yuste and Shepard labs at Columbia; in progress; James Jun will have access to this). Provides low time accuracy in the spontaneous firing case, for approximately a dozen units in a single experiment. Optical excitation may remove time uncertainty.

[1] It was noted that there are almost no ground-truthed recordings in awake, let alone behaving, animals. An exception is the recent thesis of Brian Allen [1] in which the animal is only lightly anesthetized or awake.

- Brendon Watson's recent thread at `https://groups.google.com/forum/#!topic/klustaviewas/pfVC-CMSCTs` mentions upcoming data from Dan English.

2. **Hydrid datasets.**

   Good features: they embody the correct noise model.

   Bad features: biased towards already-sorted unit populations (which selects for, eg, high firing rate, manual selection)

   - Steinmetz, Harris et al, 2016. `http://phy.cortexlab.net/data/sortingComparison/datasets/`
   - A hybrid dataset can be created by adding additional firings of sorted units to any dataset (this idea is also described as the "spike addition metric" in [2]).

3. **Recordings without ground truth.**

   Good features: abundant, possible to create partial GT by hiding a subset of electrodes.

   Bad features: no GT.

   - Litke, Chichilnisky et al, 2004 [8]. 512-channel, hexagonal array, at least 2 hrs available. monkey retina, in vitro, 20 kHz. Used in YASS testing; have to check how much can be public.
   - Josh Siegle (Allen Institute), 2018. Six simultaneous neuropixels probes each with $\sim$ 150 channels within the brain. Length unknown.
   - Tetrodes and 18-channel polymer probes from Jason Chung et al. [3]. The tetrodes are hippocampal in behaving rats, and have three independent human sortings available as a vague GT. To do: check if data can be released.

   It is worth mentioning that.

4. **Simulated (in-silico) datasets.**

   Good features: abundant number of GT units, allows arbitrary electrode design and drift simulation.

   Bad features: unvalidated themselves, noise models too clean, no artifacts, no detailed modeling of electrode effects, harder to simulate non-rigid drift.

   - BioNET (Allen Institute) simulations, by C. Mitelut.

     We suggest a single non-drifting and a single drifting dataset, with neuropixels probe geometry. Exist as 4-minute segments.

     These are expensive to run.
   - `ViSAPy` simulations, Hagen et al. 2015 [6].

     The repo contains scripts for 16-channel polytrodes, etc.

     `https://github.com/espenhgn/ViSAPy`

     Yger et al. [13] uses this code but doesn't report data. We have not yet used it.

Notes:

1. Retinal data is quite different from cortical: retinal has much lower noise, almost no drift, partial validation by planar coverage of certain cell types (eg ON/OFF parasol), and axonal spikes that are non-local (propagate across the entire array).

2. We exclude other well-known datasets (eg Harris et al. tetrode from 2000, Martinez et al. synthetic from 2009, Cmunas-Mesa et al Neurocube simulator from 2013) that are either too small or have been superceded.

4 Metrics

We list metrics that could be reported for each algorithm-dataset pair. Most are just sketched, pending formulae, then scripts that implement them. A possible classification of metrics is as follows. The three classes I, II, and III have descending order of confidence. Within a class no rank ordering is implied. We first give an extensive list; below we propose a concrete subset.

- **Accuracy vs ground-truth** (Class I).

  We first choose $\tau \approx 1$ms as the allowable spike time error; we have not found its choice makes much difference (timing only becomes an issue at the 0.1 ms or less level).

  GT should be in the form of a vector of firing times and labels. Usually human work is needed to extract this set by thresholding eg an intracellular or juxtacellular recording; often there is a subjective choice of threshold.

  1. False negative fraction. For a given GT unit, the smallest value over the sorted units of the ratio: number of spikes missed over true number of spikes. Is in range $[0, 1]$, with zero best.
  2. False positive fraction. For a given GT unit, the smallest value over the sorted units of the ratio: number of spikes not matching GT over number of sorted spikes. Is in range $[0, 1]$, with zero best.
  3. Overall error rate (inaccuracy). This combines the above: number of missed plus number of false positives, all divided by the union of true and sorted spikes. This metric is similar but not identical to $1 - f_k$ in [2]. Is in range $[0, 1]$, with zero best.

- **Biophysical metrics** (Class II).

  1. rate of refractory violations (ISI, or AC plots), separately for each unit. Requires choice of ISI lower cut-off, eg 2 ms. Called $\mathbf{f}_1^p$ in [7].
  2. Existence of unphysical notch in CC. This is useful for seeing missed spikes due to collisions; see eg [4]. We need to choose a notch width. This could be computed faster than the entire set of CCs. Related to $\mathbf{f}_3^n$ of [7].
  3. Highly-one-sided CC, indicating a bursting pair (or triplet, etc), that has been mistakenly split. Parameters need to be fixed here. Of course, one-sided CCs can occur legitimately due to synaptic coupling; we seek expertise on this.
  4. "Burstiness", assessed by off-diagonal return map (scatterplot of successive ISI for a putative unit, as in JRclust).
  5. Refractory gap in CC between a pair that matches the AC of each in the pair; this indicates a false split, due to eg drift.
  6. Disappearance of a unit over time, as an indicator of failure to handle drift.

7. Validation by spatial localization of place cell firings [3] (special to hippocampus of behaving rodent).

- **Quality metrics** not requiring ground-truth (Class III).

  These are "surrogate" metrics for quality that can be quoted for each unit. We don't yet know which are the best indicators of (in)accuracy. Crucially, they can (and must) all be able to be computed from only the input and output data of Section (2.1), and are independent of the sorting algorithm.

  1. Peak amplitude of mean template, divided by RMS noise level, after filtering. Ie, peak SNR. This is the max over time and electrode number. Noise level (σ) may be robustly estimated using median absolute deviation (MAD) computed from filtered data, scaled to a Gaussian equivalent σ [12, (3.1)]. This requires a standard definition of filter parameters purely for the computation of this quality metric. (We would exclude spatial whitening from this.)
  2. Noise-overlap [3]. Is cluster-shape agnostic. [2]
  3. Isolation [3]. Is cluster-shape agnostic.
  4. 1D projections (eg, amplitude histograms) as used in Pachitariu et al [11]
  5. Mahalanobis-style estimates of false-pos and false-neg rates based on a Gaussian clusters assumption. Requires (re)computing a local feature space.
  6. Various other quality metrics from Hill et al [7]
  7. Various stability metrics requiring reruns of the sorter [2].
  8. Community-supplied quality metrics, that could be uploaded automatically as scripts acting on the data...

- **Other metrics** and meta-data (Class IV)

  1. Algorithm run-time in seconds. Runs our framework performs would all be on the same machine. This may need a CPU-only and CPU+GPU category.
  2. Peak RAM usage.
  3. Peak disk usage (temporary intermediate files).
  4. Subjective user opinions on ease of installation, use, parameter adjustment effort, and scalability for large datasets.

Since algorithm comparison is also needed, Class V could be metrics of similarity of two sorting outputs: this we propose to be based on the *best-permuted confusion matrix* (BPCM) as described in [2, 3], which is diagonal if two sortings match. The website should be able to produce and display graphically the BPCM between any two algorithms run on the same data.

Note: we decided to avoid the computation of confusion matrices for the accuracy vs GT; they are not relevant unless there are a large number of GT units to match.

4.1 Initial implementation plan

In the first released website we expect to implement from the above list the following. Class I: 1,2,3. Class II: 1. Class III: 1,2,3. Class IV: 1.

[2] This replaced the idea of amplitude histogram separation from detection threshold.

5 Prior work and influences

We are influenced by several previous versions of online comparison tools, including:

- `http://neurofinder.codeneuro.org/`
  J. Freeman's calcium-imaging algorithm comparison. Intuitive interface (click and mouse-over for more detail), submission info, gitter chatroom, contest, cash prizes. We may or may not want the competitive aspect of leaderboard style. No data or outputs are visible, just scores.

- `http://spikefinder.codeneuro.org/`
  P. Berens, spikes from calcium fluorescence curves, similar to above.

- `http://spike.g-node.org/`
  In this now-defunct 2011-2012 project the user uploads sorted data, which is compared against a *hidden* ground truth sorting and optionally published. Layout is a little non-obvious. Tags and owners of algorithms are good.

- `http://phy.cortexlab.net/data/sortingComparison/`
  N. Steinmetz comparison of several algorithms on hybrid data.

- `http://www.spikesortingtest.com/`
  C. Mitelut's comparison site. Good collection of recent simulated data, but no algorithms, not yet used by others. Metrics: purity and completeness.

- `http://simonster.github.io/SpikeSortingSoftware/`
  Large but incomplete list of codes and their features.

6 Web and graphical interface

To do: describe web interface and which summary plots can be brought up.

7 To do

Decide whether filtered data is accessible, or if filtering is always taken to be internal to an algorithm.
Write formulae, then scripts to compute the initial quality metrics from only X, $\{t_j\}$, $\{k_j\}$ above.
Go through datasets and be more specific about which to include, and their length and size (GB).
Ask re Franke data.
Quantify and collect the simulated datasets.

References

[1] B. D. Allen. Ground truth in ultra-dense neural recording, 2016. Ph.D thesis, MIT, `http://hdl.handle.net/1721.1/109655`.

[2] A. H. Barnett, J. F. Magland, and L. F. Greengard. Validation of neural spike sorting algorithms without ground-truth information. *J. Neurosci. Methods*, 364:65–77, 2016.

[3] J. E. Chung, J. F. Magland, A. H. Barnett, V. M. Tolosa, A. C. Tooker, K. Y. Lee, K. G. Shah, S. H. Felix, L. M. Frank, and L. F. Greengard. A fully automated approach to spike sorting. *NEURON*, 95(6):1381–1394, 2017.

[4] C. Ekanadham, D. Tranchina, and E. P. Simoncelli. A unified framework and method for automatic neural spike identification. *J. Neurosci. Methods*, 222:47–55, 2013.

[5] F. Franke, R. Pröpper, H. Alle, P. Meier, J. R. Geiger, K. Obermayer, and M. H. Munk. Spike sorting of synchronous spikes from local neuron ensembles. *J. Neurophysiol.*, 114:2535–49, 2015.

[6] E. Hagen, T. V. Ness, A. Khosrowshahi, C. Sørensen, M. Fyhn, T. Hafting, F. Franke, and G. T. Einevoll. ViSAPy: A Python tool for biophysics-based generation of virtual spiking activity for evaluation of spike-sorting algorithms. *J. Neurosci. Methods*, 245:182–204, 2015.

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

[8] A. M. Litke, N. Bezayiff, E. J. Chichilnisky, W. Cunningham, W. Dabrowski, A. A. Grillo, M. Grivich, P. Grybos, P. Hottowy, S. Kachiguine, R. S. Kalmar, K. Mathieson, D. P. D, M. Rahman, and A. Sher. What does the eye tell the brain? Development of a system for the large scale recording of retinal output activity. *IEEE Trans. Nucl. Sci.*, 51(4):1434–1440, 2004.

[9] C. E. Moore-Kochkacs. Extracellular electrophysiology with close-packed recording sites: spike sorting and characterization, 2016. Ph.D thesis, BU, `https://open.bu.edu/bitstream/handle/2144/19751/MooreKochlacs_bu_0017E_12440.pdf`.

[10] J. P. Neto, G. Lpoes, J. Frazão, J. Nogueira, P. Lacerda, P. Baião, A. Aarts, A. Andrei, S. Musa, E. Fortunato, P. Barquinha, and A. R. Kampff. Validating silicon polytrodes with paired juxtacellular recordings: method and dataset, 2016. in press, *J. Neurophysiology*.

[11] M. Pachitariu, N. Steinmetz, S. Kadir, M. Carandini, and K. D. Harris. Kilosort: realtime spike-sorting for extracellular electrophysiology with hundreds of channels, 2016. bioRxiv 061481.

[12] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul. Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. *Neural Computation*, 16:1661–1687, 2004.

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