Notes on metrics and datasets for community spike sorting validation

Alex Barnett, Flatiron Institute

March 23, 2018

Abstract

We sketch and extend some ideas from the discussion in the last part of the Janelia spike sorting meeting of 2/22/18. In an attempt to get the ball rolling, this will include some subjective opinions. This could form the basis of a white paper. The community should suggest changes.

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 front-end (possibly also with an API).

The main goal is an objective accuracy comparison of current spike sorting codes. This should indicate for the e-phys community the best code to use and its expected accuracy (or distribution of accuracies across units), both of which may depend on contex (probe type, in/ex vivo, whether you have GPU, etc).

A meta-goal is to gather information about which quality metrics that are computable without ground-truth most closely match the ground-truth accuracy metric.

Curating and gathering such datasets and metrics is a community effort. Please give feedback and/or additions to this document.

1.1 Abbreviations used

GT	ground-truth(ed)
AC	auto-correlation of firing events for a single sorted unit
CC	cross-correlation of firings between different sorted units

2 Considerations

One issue is whether to upload *sorted data* or *algorithms*; the consensus seemed to be both. We have already wrapped several popular algorithms to run in the MountainLab framework, and will continue in this mode initially. We can run algorithms from a github repo with a dockerfile, which could become a standard submission format. Being careful about version numbers will be important as groups start to update their algorithms.

Held-out or *hidden* data in the style of the NetFlix Challenge might be useful. We are not yet planning to do this.

All algorithms benchmarked have to be *full automatic*, come with some *meta-data* such the parameter set, version number, etc. Algorithms should be cleanly separated from any visualization/GUI tools.

One cannot proceed without a standard *interface* between algorithms and e-phys datasets. We suggest that the interface be:

INPUT:	Raw voltage as a matrix of size *number of channels* by *number of time-points*, stored in eg 16-bit signed interger format, with all channels stored contiguously for each timepoint (ie column-major order).
OUTPUT:	For each event, a firing time (either in seconds or in time samples of the input) and label (unit assignment). This could take the form of a 2 by *number of events found* matrix.

The advantage is that no inner workings of algorithms are touched; there are algorithms such as ICA that have no pipeline stages in common with other algorithms. (Benchmarking sub-stages is a separate task that we do not tackle here.) Writing wrappers and format-converters is easy, and should not slow down the sorter unless the (large) input data is converted. All quantities of interest to display in a web-based exploration of the sorting result (peak channel, mean waveform, etc) can be derived from the union of above input and output data; some of these are expensive but can be computed after algorithm runs and cached.

A disadvantage of the interface is that it does not allow for probabilistic outputs; however, there are currently few if any spike sorters that produce such outputs. A viable patch to such a probabilistic output is to draw 20 independent samples from it and compute all metrics for each sample, giving distributions of each metric.

3 Datasets

It was agreed that a variety of brain regions, probes, and types of data are needed. We discussed some of the following datasets. 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. 32-channel and 128-channel with single juxtacellular, in vivo rat cortex. Only 1-2 datasets of each type have a close-enough unit to feasibly sort.
   - Yger et al. '18. 252 channel w/ loose patch, ex vivo mouse retina. Subset at `https://dio.org/10.5281/zenodo.1205233`
   - Franke et al '15 [4].
     URL for dataset unknown.
   - Boyden, possibly (James?)
   - Brendon Watson's recent thread at `https://groups.google.com/forum/#!topic/klustaviewas/pfVC-CMSCTs` mentions upcoming: neuropixels GT data from Kampff; data from Dan English.

[1] It was noted that there are no ground-truthed recordings in awake, let alone behaving, animals. Please correct me if I got this wrong.

2. **Hydrid datasets.**

   Good features: they embody correct noise model.

   Bad features: biased towards already-sorted units.

   - Steinmetz, Harris et al, 2016. `http://phy.cortexlab.net/data/sortingComparison/datasets/`
   - A hybrid dataset can be created from sorted units with any dataset; this idea is also described as the "spike addition metric" in [1] ...

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

   Good features: can be abundant, allow arbitrary electrode design and drift simulation.

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

4 Metrics

These are metrics that would be reported for each algorithm-dataset pair. These 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 ordering is implied.

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

  1. If the dataset has a single ground-truth unit (eg via juxta or intra-cellular recording), false negative fraction and false positive fraction for the single sorted unit which best matches the GT. An allowed time-error must be chosen, eg ± 1 ms. If multiple GT units, the best-permuted confusion matrix between the GT and sorted output must be found. [2]

- **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 [5].
  2. Existence of unphysical notch in CC. This is useful for seeing missed spikes due to collisions; see eg [3]. 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 [5].
  3. Highly-one-sided CC, indicating a bursting pair (or triplet, etc), that has been mistakenly split. Of course, one-sided CCs can occur legitimately due to synaptic coupling; we seek expertise on this.
  4. 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.
  5. Disappearance of a unit over time, as an indicator of failure to handle drift.

[2] GT should be in the form of a vector of firing times and labels. Sometimes human work is needed to extract this set from eg an intracellular recording.

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

  These can be applied independently to each unit. Crucially, they can all be computed from only the input and output data of Section (2).

  1. Peak amplitude of template, divided by RMS signal. Ie, peak SNR.
  2. Noise-overlap [2]. Is cluster-shape agnostic. [3]
  3. Isolation [2]. Is cluster-shape agnostic.
  4. 1D projections (eg, amplitude histograms) as used in Pachitariu et al [6]
  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 [5]
  7. Various stability metrics requiring reruns of the sorter [1].
  8. Community-supplied quality metrics, that could be uploaded automatically as scripts acting on the data...

- **Other metrics** and meta-data.

  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. RAM usage.
  3. Disk usage (temporary intermediate files).
  4. Subjective user reports on ease of installation and use.

Since algorithm comparison is also needed, Class IV could be metrics of similarity of two sorting outputs. We suggest that these should be derived from the *best-permuted confusion matrix* [1].

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.

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

- `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 To do in this document

Give math symbols to objects and definitions of various metrics.
Finish in-silico dataset list.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.