

Spike sorting for large, dense electrode arrays

Cyrille Rossant^{1,2,3}, Shabnam N. Kadir^{1,2,3}, Dan F. M. Goodman⁴, John Schulman⁵, Maximilian L.D. Hunter^{2,3}, Aman B. Saleem⁶, Andres Grosmark⁷, Mariano Belluscio⁷, George H. Denfield⁸, Alexander S. Ecker⁸, Andreas S. Tolias⁸, Samuel Solomon⁹, Gyorgy Buzsaki⁷, Matteo Carandini⁶, Kenneth D. Harris^{2,3,10}

1 – Equal contribution.

2 – UCL Department of Neuroscience, Physiology and Pharmacology, London, UK

3 – UCL Institute of Neurology, London, UK

4 – Department of Electrical and Electronic Engineering, Imperial College, London, UK

5 – Department of Electrical Engineering and Computer Sciences, UC Berkeley, Berkeley, CA

6 – UCL Institute of Ophthalmology, London, UK

7 – NYU Neuroscience Institute, Langone Medical Center, New York, NY

8 – Department of Neuroscience, Baylor College of Medicine, Houston, TX

9 – UCL Institute of Behavioural Neuroscience, Department of Experimental Psychology, London, UK

10 – Correspondence: kenneth.harris@ucl.ac.uk

Abstract

Developments in microfabrication technology have enabled the production of neural electrode arrays with hundreds of closely-spaced recording sites, and electrodes with thousands of sites are currently under development. These probes in principle allow the simultaneous recording of very large numbers of neurons. However, use of this technology requires the development of techniques for decoding the spike times of the recorded neurons, from the raw data captured from the probes. Here, we present a set of novel tools to solve this problem, implemented in a suite of practical, user-friendly, open-source software. We validate these methods on data from the cortex, hippocampus, and thalamus of rat, mouse, macaque, and marmoset, demonstrating error rates as low as 5%.

Introduction

One of the most powerful techniques for neuronal population recording is extracellular electrophysiology using microfabricated electrode arrays¹⁻³. Advances in microfabrication have continuously increased the number of recording sites available on neural probes, and the number of recordable neurons is further increased by having closely spaced recording sites. Indeed, while a single sharp electrode can provide good isolation of one or two neurons, placing as few as four recording sites together in a “tetraode” can reveal the firing patterns of 10-20 simultaneously recorded cells⁴⁻⁷. This increase is possible because each recorded neuron produces extracellular action potential waveforms (“spikes”) with a characteristic spatiotemporal profile across the recording sites⁸⁻¹⁰. The process of using these waveforms to decipher the firing times of the recorded neurons is known as “spike sorting”^{11,12}.

38 Spike sorting, as currently applied in nearly all labs using extracellular recordings, involves a manual
39 operator. While some labs use a fully manual system, lower error rates can be achieved with a semi-
40 automated process⁸, consisting of four steps. First, spikes are detected, typically by high-pass filtering
41 and thresholding. Second, each spike waveform is summarized by a compact “feature vector”, typically
42 by principal component analysis. Third, these vectors are divided into groups corresponding to putative
43 neurons using cluster analysis. Finally, the results are manually curated, to adjust any errors made by
44 automatic algorithms¹³. This last step is necessary because although fully automatic spike sorting would
45 be a powerful tool, the output of current algorithms cannot be accepted without human verification. A
46 similar situation arises in many fields of data-intensive science: in electron microscopic connectomics,
47 for example, automatic methods can only be used under the supervision of human operators¹⁴.

48 For tetrode data this semi-automatic process performs well, reaching error rates of 5% or lower, as
49 assessed by ground truth data obtained with simultaneous intracellular recording⁸. However, spike
50 sorting methods developed for tetrodes do not work for a newer generation of larger electrode arrays¹⁵,
51 ¹⁶. This failure occurs for two reasons. First, the automated component can fail in high dimensions, for
52 example due to the “curse of dimensionality” that affects cluster analysis in high-dimensional spaces¹⁷.
53 Second and perhaps more critically, the process of manual curation -- while manageable with low-count
54 probes -- cannot scale to the high-count case without software that guides the operator to only those
55 decisions that cannot be made reliably by a computer. While many different methods for spike sorting
56 have been proposed (e.g. refs. ¹⁸⁻²⁴), no method has yet solved these problems robustly enough to be
57 widely adopted by the experimental community.

58 Here we describe a system for the spike sorting of high-channel count electrode data, implemented in a
59 suite of freely available software. While the spike sorting problem has attracted considerable theoretical
60 research, our goal was to produce a practical system that can be immediately used by working
61 neurophysiologists. The ability to process large datasets (millions of spikes in hundreds of dimensions) in
62 reasonable human and computer time was deemed essential; error rates comparable to those of
63 commonly-used tetrode methods were deemed acceptable. We tested the software on data recorded
64 from rat neocortex with 32-site shank electrodes, as well as data from other species and brain regions.
65 While traditional methods performed extremely poorly on this data, the new algorithms gave close to
66 theoretically optimal performance. The techniques and software have been developed in a community-
67 led manner, through extensive feedback from a user base of over 320 scientists in 50 neurophysiology
68 labs. The software is downloadable and documented at <http://cortexlab.net/tools>, and is supported by a
69 highly active user-group mailing list, klustaviewas@groups.google.com.

70 Results

71 Our spike sorting pipeline involves three steps: (1) spike detection and feature extraction, (2) cluster
72 analysis, and (3) manual curation. We describe these steps in order.

73 Spike Detection

74 The first step of the pipeline is spike detection and feature extraction, implemented by the program
75 *SpikeDetekt*.

76 The primary difference between spike detection for high count silicon probes and for tetrodes is that
77 temporally overlapping spikes are extremely common in the former. This phenomenon can be seen by

78 examining of a segment of raw data recorded with high count probes (**Fig. 1**). The spikes seen in these
79 data are diverse, with some detected on only one or two channels, and others spanning large numbers
80 of channels, as expected of pyramidal cells whose apical dendrites are aligned parallel to the shank²⁵. In
81 these data, simultaneous firing of multiple neurons is common. However, simultaneously firing neurons
82 are usually detected on distinct sets of channels.

83 To deal with the problem of temporally overlapping spikes, we therefore sought to detect spikes as local
84 spatiotemporal events (**Fig. 2**). This step requires knowledge of the probe geometry, which is specified
85 by the user in the form of an “adjacency graph” (**Fig. 2a**). We illustrate the spike detection process with
86 reference to a small segment of data containing two temporally overlapping but spatially separated
87 spikes (**Fig. 2b**).

88 The first stage of the algorithm is high-pass filtering the raw data to remove the slow local field potential
89 signal (Butterworth in forward-backward mode; **Fig. 2c**). Next, spikes are detected using a double-
90 threshold flood fill algorithm (**Fig. 2d,e**). Specifically, spikes are detected as spatiotemporally connected
91 components, in which the filtered signal exceeds a “weak threshold” θ_w for every point, and in which at
92 least one point exceeds a “strong threshold” θ_s (optimal values for these parameters were found to be 4
93 and 2 times the standard deviation of the filtered signal, as described below). Two points are considered
94 neighboring if they are on a single channel and separated by one time sample, or at a single timepoint
95 on channels joined by the adjacency graph; this allows the algorithm to work with probes of any
96 geometry, not just linear ones. The dual-threshold approach avoids spurious detection of small noise
97 events, since isolated islands in which only the weak threshold is exceeded are not retained. Conversely,
98 spikes will not be erroneously split due to noise, as areas joined by weak threshold crossings are
99 merged.

100 After detection, spikes are temporally realigned to subsample resolution, to the center of mass of the
101 spike’s suprathreshold components, weighted by a power parameter p (see Methods). Visual inspection
102 showed that spike times detected with this method correspond closely to those that would be assigned
103 by a human operator (**Fig. 2e**).

104 The waveforms of each spike are summarized by two vectors. First, a “feature vector” is found by
105 principal component analysis of the realigned waveforms on each channel (3 principal components were
106 kept for the current analysis). All channels are used in computing the feature vector; thus our two
107 example spikes have similar feature vectors, as their central times are similar (**Fig. 2f**). Second, a “mask
108 vector” is computed from the peak spike amplitude on each detected channel, rescaled and clipped so
109 channels outside the connected component have mask 0, and channels with amplitude above θ_s have
110 mask 1. The mask vector allows temporally overlapping spikes to be clustered as separate cells. Indeed,
111 although the feature vectors of our two example spikes were very similar, their mask vectors are
112 completely different (**Fig. 2g**).

113 Performance Validation and parameter optimization

114 To quantify the performance and optimize the parameters of this algorithm requires “ground truth”:
115 knowledge of when the recorded neurons actually fired. We created a simulated ground truth dataset
116 by repeatedly adding the spikes of a “donor cell” identified in one recording, to a second “acceptor”
117 recording made with same probe; since the extracellular medium is a linear conductor²⁶, addition of
118 spike waveforms serves as a sufficient model for overlapping spikes. To evaluate the performance of the

119 system, we chose 10 donor cells with a variety of amplitudes and waveform distributions (**Fig. 3a**), using
120 recordings from rat cortex with a 32-channel probe shank. To model the variability of waveforms
121 produced by a single neuron due to phenomena such as bursting²⁷⁻²⁹, we scaled each spike to a random
122 amplitude in a range that varied by a factor of 2 (see Methods). We refer to the spikes added to the
123 acceptor dataset as “hybrid spikes”, and the result as a “hybrid dataset”.

124 To evaluate spike detection performance, we used a heuristic criterion to identify which spikes detected
125 by the algorithm corresponded to which hybrid spikes (see Methods). We measured performance as a
126 function of three algorithm parameters (θ_w , θ_s , and p), using four performance statistics.

127 The first statistic was the fraction of hybrid spikes detected (**Fig. 3b**). This showed a strong dependence
128 on the thresholds: values of θ_s above 4 times standard deviation (4 SD) resulted in poor detection,
129 particularly for low-amplitude cells. The dependence of performance on θ_w was more complex: poor
130 performance resulted not just from overly high values (>2.5 SD), but also overly low values (<2 SD).
131 Examination of example errors (not shown) indicated that overly low values of θ_w led to inappropriate
132 merging of temporally overlapping but spatially separated spikes, while overly high values led to
133 artificial splitting of single spikes.

134 The second statistic was the total number of detection events (**Fig. 3c**). Because this includes noise
135 events as well as true spikes of the hybrid and background cells, this number should be as small as
136 possible provided the fraction correctly detected remains high. We found that this statistic most
137 critically depended on the strong threshold, increasing markedly for values below 4SD.

138 The third statistic was timing jitter: the standard deviation of the difference between the detected and
139 actual times of each hybrid spike (**Fig. 3d**). Jitter was in all cases less than one sample, and improved for
140 larger values of θ_s and θ_w , indicating that spike times are best estimated from a minority of larger
141 amplitude spikes. For all hybrid cells, jitter was worse for $p < 1$; for low amplitude cells it showed a
142 further worsening for $p > 2$, reflecting noise introduced by overweighting of peak amplitude times.

143 The final statistic was mask accuracy (**Fig. 3e**), which measures how closely the detected mask vectors
144 match those expected from the ground truth (see Methods). This showed strongest dependence on θ_w
145 with a peak around 2 SD, and less pronounced dependence on θ_s peaking around 5 SD.

146 We conclude that close to optimal performance can be obtained using a strong threshold of 4 SD, a
147 weak threshold of 2 SD and a power weight of 2. Furthermore, using these parameters yields around
148 95% correctly detected spikes, and spike timing jitter of 0.5 samples.

149 Cluster Analysis

150
151 The second step of our spike sorting pipeline is automatic cluster analysis, implemented in the program
152 *KlustaKwik*.

153 For tetrode data, we previously found that cluster analysis using a mixture of Gaussians fit gave close to
154 optimal performance⁸. This approach cannot be directly ported to high-channel-count data for two
155 reasons. The first is the “curse of dimensionality”: in high dimensions, noise measured on the large
156 number of uninformative channels will swamp signals measured on the smaller number of informative
157 channels. Second, because temporally overlapping spikes have similar feature vectors (Fig. 2F), further
158 information such as the mask vectors must be used to distinguish these spikes.

159 To solve this problem, we designed a novel method, the “masked EM algorithm”³⁰. This algorithm fits
160 the data as a mixture of Gaussians, but with each feature vector replaced by a virtual ensemble in which
161 features with masks near zero are replaced by a noise distribution (see Methods). Channels with low
162 mask values are thus “disenfranchised”, and do not contribute to cluster assignment; the probabilistic
163 nature of this disenfranchisement means false clusters are not created when amplitudes cross an
164 arbitrary threshold. The computational complexity of this algorithm is better than that of the traditional
165 EM algorithm, scaling with the mean number of unmasked channels per spike (which does not increase
166 for larger arrays), rather than the total number of channels.

167 To evaluate the performance of this algorithm, we used the hybrid datasets described above. For each
168 dataset, we identified the cluster containing most hybrid spikes and computed the false discovery rate
169 (fraction of spikes in the cluster that were not hybrids), and the true positive rate (fraction of all hybrid
170 spikes assigned to the cluster). To estimate the theoretical optimum performance that could be
171 expected, we used the Best Ellipsoid Error Rate (BEER) measure⁸, which fits a quadratic decision
172 boundary using ground truth data, and evaluates its performance with cross-validation, varying the
173 parameters of the classifier to obtain an ROC curve showing optimal performance.

174 The masked EM algorithm’s performance on an example hybrid dataset was close to the optimum
175 estimated by the BEER measure but the classical EM algorithm’s performance was poor, with error
176 rates typically exceeding 50% (**Fig. 4a**). Across all hybrid datasets, we found no significant difference
177 between the total error of the masked EM algorithm and theoretical optimal performance ($p = 0.8$, t-
178 test), but a significant difference between the performance of the Classical and Masked EM algorithms
179 ($p = 0.005$, t-test; **Fig. 4b**). To ensure the poor performance of the classical EM algorithm did not simply
180 reflect incorrect parameter choice, we reran it for multiple values of the penalty parameter (which
181 determines the number of clusters found), but this could not improve Classical EM performance. This
182 analysis also demonstrated that the error rates of the masked EM algorithm were largely independent
183 of the penalty parameter; using a value corresponding to the Bayesian Information Criterion seems a
184 good option for penalty choice, as it led to a reasonably small number of clusters without compromising
185 error rates (**Fig. 4c,d**).

186 We conclude that the performance of the Masked EM algorithm is close to optimal for this clustering
187 problem, yielding false positive and false discovery rates both of the order 5%.

188 Manual Curation

189 The final step of the spike sorting pipeline is manual verification and adjustment of cluster assignments,
190 which are implemented in the program *KlustaViewa*.

191 Although semi-automatic clustering provides more consistency and lower error rates than fully manual
192 spike sorting⁸, further manual corrections are typically required, such as merging of clusters split due to
193 electrode drift, bursting, or other reasons²⁷⁻²⁹. These waveform shifts are hard to model and correct
194 mathematically, but can usually be identified by inspection of waveforms, auto- and cross-correlograms,
195 and cluster shapes. It is essential that this step be done with a minimum of human operator time, a
196 particularly acute problem with the very large numbers of neurons recorded by large dense electrode
197 arrays. Specifically, if N clusters are produced automatically, it is impractical for a human operator to
198 inspect all order N^2 potential merges.

199 We addressed this problem using a semi-automatic “Wizard,” that reduces the number of potential
200 merges to order N . The Wizard works by presenting the operator with pairs of potentially mergeable
201 clusters, ordered by a measure of pairwise cluster similarity. Because the Wizard is used iteratively, this
202 measure must be computable in a fraction of a second, even for datasets containing millions of spikes.
203 Thus, only metrics based on summary statistics of each cluster, rather than individual points, are
204 suitable. We evaluated several candidate similarity measures. The Kullback-Leibler divergence between
205 two Gaussian distributions was unsuitable as it overweighted differences in covariance matrix relative to
206 differences in the mean. However, good performance was obtained using a single step of the masked
207 EM algorithm to compute the similarity of the mean of one cluster to each of the others (**Fig. 5a**). To
208 verify the accuracy of this measure, we simulated automatic clustering errors by splitting the ground
209 truth clusters in the hybrid datasets into two subclusters containing high and low amplitude spikes. In all
210 cases, the similarity measure correctly identified the other half of the artificially split cluster (**Fig. 5b**).

211 The manual stage can take several hours of operator time, and human error is lowest during the start of
212 this period. The Wizard therefore iteratively presents the operator with decisions that can be made
213 quickly, with the most important decisions presented first. The Wizard iterates through all clusters
214 starting with the best currently unsorted spikes. The remaining clusters are ordered by similarity to the
215 best unsorted cluster, and the decision of whether to merge, split, or delete each merge candidate is in
216 turn made by the operator (**Fig. 5c,d**). Once satisfied that no more potential merges exist for the
217 currently best unsorted cluster, the operator either accepts it as a well-isolated neuron, or rejects it as
218 multiunit activity or noise, and the top-level iteration begins again.

219 Although the Wizard guides the operator through the decision process, the operator at all times has free
220 access to all data required to make rapid decisions, provided by KlustaViewa’s user-friendly and easily-
221 navigable graphical user interface (**Figure 6**). Using this software, the time taken for manual curation
222 scales linearly with the number of clusters, with a scaling factor that varies between operators and is
223 generally about 1 minute per cluster, regardless of probe size. This software therefore allows for
224 thorough manual curation of a dense-array recording in a few hours.

225 We assessed the performance of 8 human operators (5 experienced spike-sorters, 3 novices) using this
226 system (**Fig. 7a**). First, we asked whether the operators would correctly fix a misclustering that was
227 produced by the masked EM algorithm in simulation of electrode drift (described further below). All
228 experienced operators, and all but one of the novices did this correctly. Second we asked how
229 consistent the results of these operators would be on the same dataset (**Fig. 7b-d**). We separately
230 assessed consistency on spikes that all operators had identified be in “good” clusters, on spikes that at
231 least one operator had identified to be in a good cluster, and on all remaining spikes. Similarity was
232 assessed with the Fowlkes-Mallows index³¹, which gives a score between 1 for complete agreement, and
233 0 for complete disagreement. For all operators apart from one of the novices, consistency was
234 extremely high for those spikes identified as good by at least one operator (**Fig. 7e,f**); nevertheless the
235 judgement of whether a cluster should be considered well-isolated varied between operators (**Fig. 7g**).
236 We conclude that experienced operators are likely to make accurate and consistent judgements on
237 cluster merging identification, but that the judgement on which clusters to term “good” is inconsistent;
238 we therefore recommend that quantitative metrics^{32, 33} be used to determine isolation quality.

239 Additional tests

240 We used the system described above to answer several additional questions regarding the process of
241 spike sorting, and the design of electrodes.

242 First, we used our simulated ground truth dataset to ask how spike sorting performance would change
243 for different electrode designs. We considered two cases. In the first (“site thinning”; **Supplementary**
244 **Figs. 1 and 2**), the electrode was made less dense by omitting alternating channels on both sides. We
245 evaluated the performance of spike detection and clustering using the same hybrid spikes described
246 earlier, but only on this subset of channels (the adjacency graph was modified to join any two channels
247 that both connected to a missing channel). Spike detection was strongly impacted, with correct
248 detection rates dropping to an average of below 80% (Supplementary Fig. 1). Clustering performance
249 was also impacted, as assessed both by the theoretical optimum, and by the Masked EM algorithm.
250 While some cells saw little decrease in clustering performance (typically those found on multiple
251 channels), others were strongly impacted by both metrics (Supplementary Figure 2). We conclude that
252 performance in rat cortex decreases substantially for site spacing larger than the 40 μ m same-side site spacing
253 of these test probes.

254 Next, we simulated removing one side of the probe (**Supplementary Figs. 3 and 4**). Of the 10 hybrid cells
255 analyzed, 6 were only detectable on one of the probe’s two sides, while the other 4 could be detected
256 on both sides to a greater or lesser extent (**Supplementary Table 1**). The effect of side removal was
257 different to that of site thinning. The performance of each unit’s “preferred side” was comparable to
258 that of the full probe. However, for the 4 units that were visible on both sides of the probe, performance
259 on the “unpreferred side” was substantially worse than performance on the full probe, as assessed both
260 by theoretical optimum performance and the actual results of the masked EM algorithm. We conclude
261 that in staggered probes, the probe’s two sides function largely independently: the primary benefit of
262 two-sided shanks is not to increase the isolation quality of a cell already well isolated on one side of the
263 probe, but to record from a larger number of units.

264 Next, we asked whether similar performance to that seen in neocortex could also be obtained in other
265 brain structures and species. We first generated an additional 5 hybrid cells using 10-site recordings
266 from rat CA1 (**Supplementary Figs. 5 and 6**). Good performance was again obtained; furthermore, the
267 spike detection parameters found to be optimal in cortical data were also optimal in CA1 data. We then
268 ran the same code on high-count data collected from a wider range of preparations: V1 of awake mouse
269 and awake macaque monkey (**Supplementary Figs. 7-9**), and LGN thalamus of anesthetized marmoset
270 (**Supplementary Fig. 10**). Additional confidence in the method was provided both by further analyses of
271 hybrid data (**Supplementary Fig. 11**) and by the observation of sharp orientation-tuned responses
272 (Supplementary Fig. 7c-l), including amongst cells of apparently similar waveforms that were
273 nevertheless separated by the spike sorting procedure (Supplementary figure 7m).

274 Next, we asked how well the system would deal with non-stationarity in spike amplitudes. Such non-
275 stationarity can occur both because of electrode drift, and also because of activity-related changes in
276 spike amplitude such as after bursts or prolonged periods of firing²⁷. Examination of data from acute
277 recordings (where electrode drift is often stronger than with chronic probes), showed that the algorithm
278 often tracked drift successfully, but in other cases split the spikes of a single drift cell into multiple
279 clusters requiring manual merging (**Supplementary Fig. 12**).

280 To simulate nonstationarity, we constructed 6 hybrid datasets in which spike amplitude drifted
281 throughout the recording as a geometric random walk (**Supplementary Fig. 13**). Spike detection was
282 hardly impacted by this nonstationarity (**Supplementary Fig. 14**). For clustering, only one of the 6 drift
283 hybrid datasets required manual curation, and once this was performed, accuracy of the masked EM
284 algorithm was comparable to the theoretical optimum (**Supplementary Fig. 15**). A different type of
285 nonstationarity, in which the hybrid cell simply stopped firing halfway through the recording, also had
286 no effects on performance ($p=0.75$; two-sample t-test on total errors; **Supplementary Fig. 16**). As an
287 important task is often to track cells between recordings made over multiple days – i.e. where drift
288 occurs in non-recorded periods – we also asked whether the Wizard’s similarity metric might be used for
289 this purpose. Although ground truth data was not available, a conservative criterion gave encouraging
290 results, as indicated by the similarities of the autocorrelograms of the units associated to each other
291 (**Supplementary Fig. 17**).

292 A strategy sometimes used to deal with nonstationarity is to include time as an additional feature in the
293 cluster analysis algorithm, in principle allowing the algorithm to track slow changes in amplitude. To our
294 surprise, we found that this actually worsened clustering performance, which could not always be
295 overcome by manual curation (Supplementary Fig. 15). We conclude that nonstationarity (at least of the
296 type modelled here) does not present a serious problem to automatic sorting performance if time is not
297 added as an additional feature, and if manual curation is performed when required.

298 Discussion

299 We have produced a software suite for spike sorting of data from large, dense electrode arrays. Analysis
300 of simulated ground-truth data indicated that error rates of this approach are frequently of the order
301 5%.

302 A critical step in this system, and all others currently in wide use for *in vivo* data, is manual curation.
303 Extracellular array recordings are subject to numerous sources of error including electrode drift,
304 overlapping spikes, and the fact that neuronal spike waveforms are not constant, but change according
305 to firing patterns including but not limited to bursting²⁷⁻²⁹. While most working neurophysiologists have
306 a good understanding of these potential artifacts, formalizing this knowledge into a reliable
307 mathematical model has proved challenging. Because spike sorting errors could lead to erroneous
308 scientific conclusions²⁹, it remains essential that a scientist is able to inspect the results produced by an
309 automatic algorithm, then correct or discard its results. We found that experienced operators tended to
310 make similar judgements during the manual curation process, but that their judgements of which units
311 were well-isolated were subjective. Fortunately, quantitative criteria exist for assessing the quality of
312 unit isolation^{32,33}, and we therefore recommend that these be used, rather than human judgements,
313 when deciding which cells to include in further scientific analysis.

314 The current performance of the system is sufficient for practical analysis of data produced by current,
315 commercially-available silicon probes. Nevertheless, there remain areas for further improvement. The
316 first of these concerns execution time. KlustaKwik is several orders of magnitude faster than standard
317 mixture of Gaussians fitting; nevertheless, when running on large datasets, it can take hours or even
318 days to complete on a standard single-core machine. Hardware acceleration such as GPUs³⁴ or cloud
319 computing³⁵ may speed up this analysis stage, as may alternative cluster analysis algorithms that
320 exclude the most computationally expensive step of covariance matrix estimation (e.g. Refs. ^{36,37}). Faster

321 versions of the code presented here, currently under development, are available at
322 <https://github.com/kwikteam/klustakwik2> and <https://github.com/kwikteam/phy>. A second opportunity
323 for improvement regards the detection of spatiotemporally overlapping spikes. While the current
324 algorithm can detect the majority of temporally overlapping spikes, which occur on distinct sets of
325 channels, it cannot resolve spikes that overlap in both space and time. Template-matching algorithms
326 have solved this problem in the case of *in vitro* retinal array data^{38,39}, but these data are much less noisy
327 than *in vivo* brain recordings. While recent research suggests that certain forms of template matching
328 may succeed at least for tetrode data *in vivo*^{18,21}, such methods are not at present widely applied to *in*
329 *vivo* recordings, and numerous challenges need to be overcome, most critically regarding the manual
330 curation step. The platform we have described here constitutes both a practical solution to today's spike
331 sorting challenges, and also a framework from which to develop solutions for future generations of
332 electrodes containing thousands of channels.

333 Contributions

334 C.R., D.F.M.G., S.N.K. and J.S. wrote SpikeDetekt. K.D.H., S.N.K., and D.F.M.G. designed the Masked EM
335 algorithm and wrote KlustaKwik. C.R. and M.L.D.H. wrote KlustaViewa. C.R. wrote Galry. S.N.K. analyzed
336 algorithm performance. Rat data were recorded by A.G., M.B. and G.B.. Mouse data were recorded by
337 A.S. and M.C.. Marmoset data were recorded by S.S. The procedure for non-chronic laminar recordings
338 with Neuronexus Vector probes in awake, behaving macaques was developed by G.H.D., A.S.E., A.S.T.,
339 who also collected the data. K.D.H., S.N.K., and C.R. wrote the manuscript with inputs from all authors.

340 Acknowledgements

341 We thank the 200+ members of the klustaviewas@groups.google.com mailing list for their feedback,
342 bug reports, and suggestions. This work was supported by EPSRC (K015141, I005102, KDH) and the
343 Wellcome Trust (95668, 100154, KDH, MC).

344 References

- 345 1. Buzsaki, G. Large-scale recording of neuronal ensembles. *Nat Neurosci* **7**, 446-451 (2004).
- 346 2. Wise, K.D. & Najafi, K. Microfabrication techniques for integrated sensors and microsystems.
347 *Science* **254**, 1335-1342 (1991).
- 348 3. Csicsvari, J. *et al.* Massively parallel recording of unit and local field potentials with silicon-based
349 electrodes. *Journal of Neurophysiology* **90**, 1314-1323 (2003).
- 350 4. McNaughton, B.L., O'Keefe, J. & Barnes, C.A. The stereotrode: a new technique for simultaneous
351 isolation of several single units in the central nervous system from multiple unit records. *J*
352 *Neurosci Methods* **8**, 391-397 (1983).
- 353 5. Gray, C.M., Maldonado, P.E., Wilson, M. & McNaughton, B. Tetrodes markedly improve the
354 reliability and yield of multiple single-unit isolation from multi-unit recordings in cat striate
355 cortex. *J Neurosci Methods* **63**, 43-54 (1995).
- 356 6. Wilson, M.A. & McNaughton, B.L. Dynamics of the hippocampal ensemble code for space.
357 *Science* **261**, 1055-1058 (1993).
- 358 7. Recce, M. & O'Keefe, J. The tetrode: a new technique for multi-unit extracellular recording. *Soc*
359 *Neurosci Abstr* **15**, 1250 (1989).

- 360 8. Harris, K.D., Henze, D.A., Csicsvari, J., Hirase, H. & Buzsaki, G. Accuracy of tetrode spike
361 separation as determined by simultaneous intracellular and extracellular measurements.
362 *J. Neurophysiol.* **84**, 401-414 (2000).
- 363 9. Henze, D.A. *et al.* Intracellular features predicted by extracellular recordings in the hippocampus
364 In vivo. *J. Neurophysiol.* **84**, 390-400 (2000).
- 365 10. Gold, C., Henze, D.A., Koch, C. & Buzsaki, G. On the origin of the extracellular action potential
366 waveform: A modeling study. *J Neurophysiol* **95**, 3113-3128 (2006).
- 367 11. Einevoll, G.T., Franke, F., Hagen, E., Pouzat, C. & Harris, K.D. Towards reliable spike-train
368 recordings from thousands of neurons with multielectrodes. *Curr Opin Neurobiol* **22**, 11-17
369 (2012).
- 370 12. Lewicki, M.S. A review of methods for spike sorting: the detection and classification of neural
371 action potentials. *Network* **9**, R53--R78 (1998).
- 372 13. Hazan, L., Zugaro, M. & Buzsaki, G. Klusters, NeuroScope, NDManager: a free software suite for
373 neurophysiological data processing and visualization. *J Neurosci Methods* **155**, 207-216 (2006).
- 374 14. Briggman, K.L., Helmstaedter, M. & Denk, W. Wiring specificity in the direction-selectivity circuit
375 of the retina. *Nature* **471**, 183-188 (2011).
- 376 15. Berenyi, A. *et al.* Large-scale, high-density (up to 512 channels) recording of local circuits in
377 behaving animals. *J Neurophysiol* **111**, 1132-1149 (2014).
- 378 16. Du, J., Blanche, T.J., Harrison, R.R., Lester, H.A. & Masmanidis, S.C. Multiplexed, high density
379 electrophysiology with nanofabricated neural probes. *PLoS One* **6**, e26204 (2011).
- 380 17. Bouveyron, C. & Brunet-Saumard, C. Model-based clustering of high-dimensional data: A review.
381 *Comput Stat Data An* **71**, 52-78 (2014).
- 382 18. Ekanadham, C., Tranchina, D. & Simoncelli, E.P. A unified framework and method for automatic
383 neural spike identification. *J Neurosci Methods* **222**, 47-55 (2014).
- 384 19. Carlson, D.E. *et al.* Multichannel electrophysiological spike sorting via joint dictionary learning
385 and mixture modeling. *IEEE Trans Biomed Eng* **61**, 41-54 (2014).
- 386 20. Calabrese, A. & Paninski, L. Kalman filter mixture model for spike sorting of non-stationary data.
387 *J Neurosci Methods* **196**, 159-169 (2011).
- 388 21. Franke, F., Natora, M., Boucsein, C., Munk, M.H. & Obermayer, K. An online spike detection and
389 spike classification algorithm capable of instantaneous resolution of overlapping spikes. *J*
390 *Comput Neurosci* **29**, 127-148 (2010).
- 391 22. Quiroga, R.Q., Nadasdy, Z. & Ben-Shaul, Y. Unsupervised spike detection and sorting with
392 wavelets and superparamagnetic clustering. *Neural Comput* **16**, 1661-1687 (2004).
- 393 23. Swindale, N.V. & Spacek, M.A. Spike sorting for polytrodes: a divide and conquer approach.
394 *Front Syst Neurosci* **8**, 6 (2014).
- 395 24. Swindale, N.V. & Spacek, M.A. Spike detection methods for polytrodes and high density
396 microelectrode arrays. *J Comput Neurosci* (2014).
- 397 25. Buzsaki, G. & Kandel, A. Somadendritic backpropagation of action potentials in cortical
398 pyramidal cells of the awake rat. *J Neurophysiol* **79**, 1587-1591 (1998).
- 399 26. Logothetis, N.K., Kayser, C. & Oeltermann, A. In vivo measurement of cortical impedance
400 spectrum in monkeys: implications for signal propagation. *Neuron* **55**, 809-823 (2007).
- 401 27. Harris, K.D., Hirase, H., Leinekugel, X., Henze, D.A. & Buzsaki, G. Temporal interaction between
402 single spikes and complex spike bursts in hippocampal pyramidal cells. *Neuron* **32**, 141-149
403 (2001).
- 404 28. Quirk, M.C., Blum, K.I. & Wilson, M.A. Experience-Dependent Changes in Extracellular Spike
405 Amplitude May Reflect Regulation of Dendritic Action Potential Back-Propagation in Rat
406 Hippocampal Pyramidal Cells. *J. Neurosci.* **21**, 240-248 (2001).

407 29. Quirk, M.C. & Wilson, M.A. Interaction between spike waveform classification and temporal
408 sequence detection. *J. Neurosci. Methods* **94**, 41-52 (1999).

409 30. Kadir, S.N., Goodman, D.F. & Harris, K.D. High-Dimensional Cluster Analysis with the Masked EM
410 Algorithm. *Neural Comput*, 1-16 (2014).

411 31. Fowlkes, E.B. & Mallows, C.L. A Method for Comparing 2 Hierarchical Clusterings. *J Am Stat*
412 *Assoc* **78**, 553-569 (1983).

413 32. Schmitzer-Torbert, N., Jackson, J., Henze, D., Harris, K. & Redish, A.D. Quantitative measures of
414 cluster quality for use in extracellular recordings. *Neuroscience* **131**, 1-11 (2005).

415 33. Hill, D.N., Mehta, S.B. & Kleinfeld, D. Quality metrics to accompany spike sorting of extracellular
416 signals. *J Neurosci* **31**, 8699-8705 (2011).

417 34. Owens, J.D. *et al.* GPU computing. *Proceedings of the IEEE* **96**, 879-899 (2008).

418 35. Freeman, J. *et al.* Mapping brain activity at scale with cluster computing. *Nature methods* **11**,
419 941-950 (2014).

420 36. Comaniciu, D. & Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE T*
421 *Pattern Anal* **24**, 603-619 (2002).

422 37. Rodriguez, A. & Laio, A. Machine learning. Clustering by fast search and find of density peaks.
423 *Science* **344**, 1492-1496 (2014).

424 38. Marre, O. *et al.* Mapping a complete neural population in the retina. *J Neurosci* **32**, 14859-14873
425 (2012).

426 39. Pillow, J.W., Shlens, J., Chichilnisky, E.J. & Simoncelli, E.P. A model-based spike sorting algorithm
427 for removing correlation artifacts in multi-neuron recordings. *PLoS One* **8**, e62123 (2013).

428

429

430

431

432

433

434

435

436

437

438

439 Figure Legends

440

441 **Figure 1: High-count silicon probe recording.**

442 (a), Layout of the 32-site electrode array used to collect test data. (b), Short segment of data recorded in
443 rat neocortex with this array. Color of traces indicates recording from the corresponding colored site in
444 (a). Black rectangles highlight action potential waveforms; note the frequent occurrence of temporally
445 overlapping spikes on separate recording channels.

446

447 **Figure 2: Local spike detection algorithm.**

448 (a), Adjacency graph for the 32-channel probe. (b), Segment of raw data showing two simultaneous
449 action potentials on spatially separated channels (scale bars indicate 0.5mV / 10 samples). (c), High-pass
450 filtered data shown in pseudocolor format (units of standard deviation). Vertical lines on the colorbar
451 indicate strong and weak thresholds, θ_s and θ_w (respectively 4 and 2 times standard deviation). (d),
452 Gray-scale representation showing samples which cross the weak threshold (gray), and the strong
453 threshold (white). (e), Results of two-threshold flood fill algorithm, showing connected components
454 corresponding to the two spikes in orange and brown. Note that isolated weak threshold crossings
455 resulting from noise are removed. White lines indicate alignment times of the two spikes. (f),
456 Pseudocolor representation of feature vectors for the two detected spikes (top and bottom). Each set of
457 three dots represents three principal components computed for the corresponding channel (arbitrary
458 units). Note the similarity of the feature vectors for these two simultaneous spikes (top and bottom). (g),
459 Mask vectors obtained for the two detected spikes (top and bottom; 0 represents completely masked, 1
460 completely unmasked). Unlike the feature vectors, the mask vectors for the two spikes differ. Each set of
461 three dots represents the three identical components of the mask vector for the corresponding channel.

462

463 **Figure 3: Evaluation of spike detection performance.**

464 (a), Waveforms of the 10 donor cells used to test spike detection performance, in order of increasing
465 peak amplitude (left to right). (b), Fraction of correctly detected spikes as a function of strong threshold
466 θ_s (left), weak threshold θ_w (center), and power parameter p (right). Colored lines indicate performance
467 for the correspondingly colored donor cell waveform shown in A; black line indicates mean over all
468 donor cells. (c-e), Dependence of the total number of detected events, timing jitter, and mask accuracy
469 on the same three parameters.

470

471 **Figure 4: Evaluation of automatic clustering performance.**

472 (a), Receiver-Operating Characteristic (ROC) Curve showing the performance of the Masked EM
473 algorithm (blue) and Classical EM algorithm (red) on one of the 10 hybrid datasets; each dot represents
474 performance for a different value of the penalty parameter. The cyan curve shows a theoretical upper

475 bound for performance, the best ellipsoid error rate (BEER) measure obtained by cross-validated
476 supervised learning. **(b)**, Mean and standard error of the total error (false discovery plus false positive)
477 over all 10 hybrid datasets for theoretical optimum (BEER measure), Masked EM and Classical EM
478 algorithms. For each dataset and measure, the parameter setting leading to best performance was used.
479 **(c)**, Effect of varying the penalty parameter (as a multiple of the AIC penalty) on the total error for both
480 algorithms. The dotted line indicates the parameter value corresponding to BIC. Note that the Masked
481 EM algorithm performed well for all penalty values. **(d)**, The number of clusters returned by the Masked
482 EM algorithm as a function of the penalty parameter.

483

484 **Figure 5: The “Wizard” for computer-guided manual correction.**

485 **(a)**, Illustration of the measure used to quantify cluster similarity. p_{ij} represents the posterior
486 probability with which the EM algorithm would assign of the mean of cluster i to cluster j . **(b)**, To test
487 this measure, the clusters corresponding to hybrid spikes were artificially cut into halves of high and low
488 amplitude. In each case, the similarity measure identified the second half as the closest merge
489 candidate. **(c)**, The Wizard identifies the best unsorted cluster as the one with highest quality (top), and
490 finds the closest match to it using the similarity matrix. **(d)**, The Wizard algorithm. The best unsorted
491 cluster and closest match are identified. The operator can choose merge the closest match into the best
492 unsorted, ignore the closest match, or delete it by marking it as multiunit activity or noise; the wizard
493 then presents the next closest match to the operator (blue arrows). After a sufficient number of
494 matches have been presented, the operator can decide that no further potential matches could have
495 come from the same neuron, and either accept the best unsorted cluster as a well-isolated neuron, or
496 delete it as multiunit activity or noise. The wizard then finds the next best unsorted cluster to present to
497 the operator (orange arrows).

498 **Figure 6: Screenshot of the KlustaViewa graphical user interface.**

499 In order to make the decisions presented by the Wizard, the operator has access to information
500 including waveforms (center panel; gray waveforms correspond to masked channels), principal
501 component features (top right), auto- and cross-correlograms (bottom right), and an automatically
502 computed similarity metric for each pair of clusters (inset). To enable rapid navigation, all views are
503 integrated; for example, clicking on a particular channel in the Waveform View will update other views
504 to show the selected channels or clusters.

505 **Figure 7: Consistency of manual curation across operators.**

506 **(a)**, Performance of 8 human operators (5 experts, 3 novices) on a “drifty” hybrid cell requiring manual
507 curation (see supplementary figure 13b). A tick indicates correct merging of the split hybrid cell, a cross
508 indicates this merge was not performed. **(b-d)**, consistency of assignments of multiple operators over all
509 cells in this dataset. Each submatrix shows the conditional probability of the first operator’s cluster
510 assignments given the assignments of the second operator (color scale at bottom of **(d)**). **(b)**, consistency
511 of cluster assignments for spikes marked as well-isolated by all operators; **(c)**, consistency of cluster
512 assignments for spikes marked as well-isolated by at least one operator; **(d)**, consistency of whether
513 spikes were marked as well-isolated by different operators. **(e-g)**: Operator consistency for the analyses
514 of **(b-d)** was quantified using the Fowlkes-Mallows index, for which 1 represents complete agreement

515 and 0 complete disagreement. Note that while cluster assignments were highly consistent between all
516 expert operators, the operators were often inconsistent in their judgements of which units were well-
517 isolated.

518 Methods

519 A supplementary Methods checklist is available.

520 Test data

521 To test the algorithm, we created simulated ground truth data using a method termed “hybrid
522 datasets”. The primary raw data used to construct this ground truth (shown in the main text figures)
523 consisted of two separate recordings from somatosensory cortex (−3.8 mm from bregma, 3 mm lateral
524 to midline, 1mm depth) of sleeping adult rats, using silicon probes with 32 non-activated platinum-
525 plated recording sites of size 10x16 μm arranged in a staggered shank configuration (vertical spacing 20
526 μm between adjacent sites on opposite sides of the shank, 40 μm between adjacent sites on the same side),
527 mounted on a home-made microdrive. Ground and reference electrodes were stainless steel screws
528 over the cerebellum. Data was continuously recorded wideband (1Hz-Nyquist), at a sampling rate of 20
529 kHz. During the recording session, the signals were amplified (1000x), bandpass filtered (1 to 5000 Hz),
530 and acquired continuously at 20 kHz on a 128-channel DataMax system (16-bit resolution; RC
531 Electronics). All protocols were approved by the Institutional Animal Care and Use Committee of Rutgers
532 University.

533 To perform additional tests (supplementary figures 5-12), we analyzed data collected in additional brain
534 structures and species. Data was collected from the septal third of hippocampal CA1 region in male rats
535 using 10-site silicon probes using the same methods as above. All protocols were approved by the
536 Institutional Animal Care and Use Committee of Rutgers University. To obtain recordings in mouse V1,
537 mice were implanted with a custom-built head post and recording chamber (4 mm inner diameter)
538 under isoflurane anesthesia. After several days acclimatization to head-fixation, animals were
539 anesthetized under isoflurane and a ~1 mm craniotomy was performed over area V1 one day prior to
540 the first recording (see Refs. ^{40, 41} for further details). Data were recorded with an acutely-inserted 32-
541 site Neuronexus Edge probe (20 micron spacing). Experiments were conducted according to the UK
542 Animals (Scientific Procedures) Act, 1986 under personal and project licenses issued by the Home Office
543 following ethical review. Non-chronic recordings were obtained from cortical area V1 of two awake,
544 behaving, adult male rhesus monkeys (*macaca mulatta*) using Neuronexus Poly2 and custom-designed
545 Edge (60 micron spacing) Vector probes. Animals were first implanted with scleral search coils and fit
546 with a custom-built titanium head post and recording chamber (see Refs. ^{42, 43} for further details).
547 Subsequently, a 2-3mm diameter trephination was performed through which daily penetrations would
548 be made. Data were acquired as broad-band signals (0.5–16 kHz, sampled at 32 kHz), digitized at 24-bits
549 using PXI-4498 cards (National Instruments, Austin, TX). All procedures were conducted in accordance
550 with the ethical guidelines of the National Institutes of Health and were approved by the Baylor College
551 of Medicine IACUC. To obtain recordings from dorsal lateral geniculate nucleus (LGN) of sufentanil-
552 anaesthetised adult male marmoset monkey (*Callithrix jacchus*), a craniotomy was made over the right
553 LGN and a Neuronexus A16x2 probe (500μm probe separation, 50μm spacing between contact points
554 on each shank) was lowered into LGN and allowed to settle for at least 30 minutes before recording.
555 Data were band-pass filtered (0.3–5kHz, sampled at 24kHz), and digitized by a Tucker-Davis

556 Technologies RZ2 real time processor (see Ref. ⁴⁴ for further details). All procedures were approved by
557 the University of Sydney Animal Ethics Committee and conform to Australian National Health and
558 Medical Research Council (NHMRC) policies on the use of animals in neuroscience research.

559 Hybrid datasets

560 To create the hybrid datasets, we first completed a full spike sorting of each dataset, including manual
561 verification. Five clusters were chosen from each dataset, corresponding to neurons spanning the range
562 of amplitudes and channel distributions observed in the data (Figure 3A). The mean unfiltered waveform
563 of each neuron was computed, its mean was subtracted, and its value at each end was set to exactly
564 zero by tapering with a Hamming function. These “donor waveforms” were added at prescribed times to
565 the raw unfiltered data of the other “acceptor” recording. To simulate amplitude variability, we linearly
566 scaled each added waveform by a random factor chosen from the range $[\sqrt{2}/2, \sqrt{2}]$ causing amplitudes
567 to vary by a factor of two, which suffices to capture the variability typical of bursting neurons ²⁷. The
568 interspike intervals typical of bursting neurons were not simulated as this does not affect the spike
569 detection or clustering process; instead, hybrid spikes were added regularly at rates in the range 2-4
570 spikes per second. To ensure that the simulated data tested the ability of our software to realign spikes
571 to subsample resolution, each added spike was shifted by a random subsample offset using cubic spline
572 interpolation. For simulations of drift cells, amplitude was as geometric random walk (i.e. the
573 exponential of a Brownian random walk), which was then normalized so that the mean amplitude
574 remained the same as its non-drifty counterpart.

575 File format

576 To implement the software, we designed an HDF5-based file format to store raw data, intermediate
577 analysis results (such as extracted spike waveforms and feature vectors), as well as final data such as
578 spike times and cluster assignments ⁴⁵. The format makes use of HDF5 links to allow a single, small file
579 (the “.kwik file”) containing all data required for scientific analysis (e.g. spike times, cluster assignments,
580 unit isolation quality measures). Bulky raw data and intermediate processing steps such as feature
581 vectors are stored in separate files (the “.kwd” and “.kwx” files). This “detachable” format is designed
582 for data sharing applications, allowing users to download as much data as required for their needs. A full
583 specification of the format can be found at <https://phycortexlab.net/format>.

584 SpikeDetekt

585 Spike detection was implemented by SpikeDetekt, a custom program written in Python 2.7 using the
586 packages NumPy, SciPy, and PyTables.

587 The first step of the program is to filter the raw voltage trace data to remove the low-frequency local
588 field potential (LFP). This is achieved with a 3rd order Butterworth filter used in the forward-backward
589 mode to ensure zero phase distortion. Filter parameters can be specified by the user; for the analyses
590 described here we used a band-pass filter of 500 Hz to $0.95 \cdot \text{Nyquist}$.

591 The second step is threshold determination. Spike detection thresholds are specified as multiples of the
592 standard deviation of the filtered signal; at the option of the user, a single threshold is used for all
593 channels in order to avoid emphasizing noise from low-amplitude channels. To boost execution speed
594 while minimizing the chance of biased estimates, the standard deviation is estimated from five data
595 chunks of length 1 second each, picked randomly from throughout the recording. The standard

596 deviation is computed with a robust estimator, $\text{median}(|V|)/.6745$, to avoid contamination by spike
 597 waveforms.

598 The next step is spike detection. The spike detection code operates on consecutive chunks of data (1s
 599 length) for memory efficiency. Spatiotemporally connected regions of weak threshold crossing are
 600 detected using a non-recursive flood fill algorithm, with spatial continuity defined using a user-specified
 601 adjacency graph. Only connected components for which at least one point exceeds the strong threshold
 602 are kept for further analysis.

603 Spike alignment is computed based on a scaled and clipped transformation of the filtered
 604 voltage $V(t, c)$:

$$\psi(t, c) = \min\left(\frac{-V(t, c) - \theta_w}{\theta_s - \theta_w}, 1\right)$$

605 Note that $\psi(t, c)$ can never be negative within a spike, as the floodfill algorithm only finds points for
 606 which $-V(t, c) > \theta_w$. The center time for each spike S is computed as

$$\bar{t}_S = \frac{\sum_{(t,c) \in S} t \psi(t, c)^p}{\sum_{(t,c) \in S} \psi(t, c)^p}$$

607 where $(t, c) \in S$ denotes the set of times and channels, for all points assigned to this spike by the
 608 floodfill algorithm. If $p = 1$, this formula measures the spike's center of mass; if $p = \infty$, it measures the
 609 time of the spike peak.

610 Spikes were realigned on \bar{t}_S to subsample resolution using cubic spline interpolation (note that the
 611 center time will, in general, not be an integer number of samples). Feature vectors are computed for
 612 each channel separately by principal component analysis; the number of features per channel is a user
 613 settable parameter, with default value 3. Finally, mask vectors are computed for each spike S as zero for
 614 channels not appearing in the connected component, and as the maximum scaled waveform for all
 615 channels inside the component:

$$m_{c,S} = \max_{t:(t,c) \in S} \psi(t, c)$$

616 To evaluate the performance of SpikeDetekt, required identifying which detected spikes correspond to
 617 ground truth spikes. This was done with a dual criterion: the difference between the detected time and
 618 ground truth needed to be less than 2 samples, and the detected mask vector \mathbf{m}_S needed to have a
 619 similarity to the ground truth mask vector \mathbf{m}_G of at least 0.8, defined by the mask similarity measure

$$\frac{\mathbf{m}_S \cdot \mathbf{m}_G}{|\mathbf{m}_S| |\mathbf{m}_G|}$$

620 Note that mask similarity cannot exceed 1, by the Cauchy-Schwartz inequality. The validity of this
 621 criterion was verified by showing that detected spike timing jitter rapidly increased for similarity
 622 threshold for values less than 0.8, but was insensitive to threshold value above 0.8. Once the detected
 623 spikes corresponding to ground truth had been identified, the four measures in figure 3 were computed.
 624 This analysis used the Python library Joblib to prevent unnecessary recomputation.

625 KlustaKwik

626 Automatic clustering was performed by KlustaKwik, a custom program written in C++. The first version
627 of this program was designed for tetrode data, implemented a hard EM algorithm for maximum-
628 likelihood fitting of a mixture of arbitrary-covariance Gaussians, and was released in 2000 but not
629 specifically described in a published manuscript. Here, we have implemented several modifications of
630 this software to enable automatic sorting of high-count probe data. The program now implements a
631 novel “masked EM algorithm”³⁰ designed for high-dimensional classification, as well as other features
632 such as cache optimization resulting in a speed increase of over 10,000%.

633 The masked EM algorithm takes as input both feature vectors and mask vectors. It works by fitting a
634 mixture of Gaussians to a virtual dataset in which each feature vector is replaced by a probability
635 distribution:

$$\tilde{x}_{n,S} \sim \begin{cases} x_{n,S} & \text{prob } m_{n,S} \\ N(\nu_n, \sigma_S^2) & \text{prob } 1 - m_{n,S} \end{cases}$$

636 Here, $x_{n,S}$ represents the n^{th} component of the feature vector for spike S ; $m_{n,S}$ represents the n^{th}
637 component of the mask vector for spike S ; and $N(\nu_n, \sigma_S^2)$ denotes a univariate Gaussian distribution
638 with mean and variance equal to those of the subthreshold noise distribution of the n^{th} feature.

639 The masked EM algorithm consists of alternation of an “E step” in which each spike is assigned to the
640 cluster for which it has highest posterior probability, and an “M step” in which the means and
641 covariances of each cluster are estimated. We have derived analytic formulas for the expectation of the
642 cluster assignment probability used in the E-step, and the cluster mean and variance used in the M step,
643 over the virtual probability distribution $\tilde{x}_{n,i}$ ³⁰. Thus, explicit sampling from the virtual distribution does
644 not need to be performed; furthermore, these expectations can be computed much faster than those of
645 the full EM algorithm as they scale with the square of the number of unmasked features, rather than the
646 square of the total number of features.

647 KlustaKwik automatically determines the number of clusters that best fit the data, determined using a
648 penalty function that encodes a preference for fits with smaller numbers of clusters. We have found a
649 modification of the Bayesian Information Criterion to deal with masked data works well in practice³⁰.
650 Because the algorithm allows for dynamic splitting and merging of clusters during the fitting process, a
651 search for the optimal number of clusters can be achieved in a single run of the algorithm. We have
652 found that starting the algorithm from an initial clustering determined heuristically from the mask
653 vectors avoids the problem of local maxima, and allows good results to be obtained from a single run.

654 KlustaViewa

655 Manual correction of automatic clustering is performed with KlustaViewa, a custom program written in
656 Python 2.7. The manual stage requires interactive visualization of very large numbers of data points, for
657 which existing libraries such as matplotlib were not suitable. We therefore designed a new Python
658 library for rapid interactive data visualization named Galry⁴⁶. Galry leverages the computational power
659 of modern graphics processing units³⁴ through the OpenGL graphics library⁴⁷. High performance is
660 achieved by porting most visualization computations to the GPU using custom shaders, and by
661 minimizing the number of OpenGL API calls through batch rendering techniques.

662 To ensure rapid adoption by the experimental community, we designed KlustaViewa's user interface by
 663 the integrating novel features necessary for high-count probes into a user interface as similar as possible
 664 to existing manual spike sorting environments such as Klusters¹³. In addition to data views familiar from
 665 previous spike sorting systems (such as waveform, auto- and cross-correlograms, and similarity matrix),
 666 we implemented several new features. The most important of these is the Wizard (described in the
 667 main text), that automatically leads the user through the manual verification and merging process, while
 668 always allowing the user free access to all of the views familiar from standard spike sorting systems. In
 669 addition, a number of enhancements were designed specifically to make the sorting of high-count probe
 670 data tractable. These include features to allow display of masking information; rapid and automatic
 671 display of the channels relevant to selected clusters; transient color brushing⁴⁸; and automatic
 672 downsampling to ensure low latency display when dealing with very large datasets.

673 The Wizard is based on a metric of similarity for each pair of clusters. This was computed by running a
 674 single step from the EM algorithm to compute the posterior probability for assigning the mean of cluster
 675 i to cluster j :

$$p_{ij} = \frac{w_j N(\mu_i | \mu_j; C_j)}{\sum_k w_k N(\mu_i | \mu_k; C_k)}$$

676 Here w_j represents the weight of cluster j (i.e. the fraction of points already assigned to this cluster); μ_j
 677 and C_j represent its mean and covariance as computed by the M-step of the masked EM algorithm. The
 678 quality of each cluster j was defined as the diagonal element p_{jj} , i.e. the posterior probability for
 679 classifying cluster j 's mean as coming from cluster j itself. A high value for p_{jj} therefore indicates that
 680 cluster j has no close neighbors.

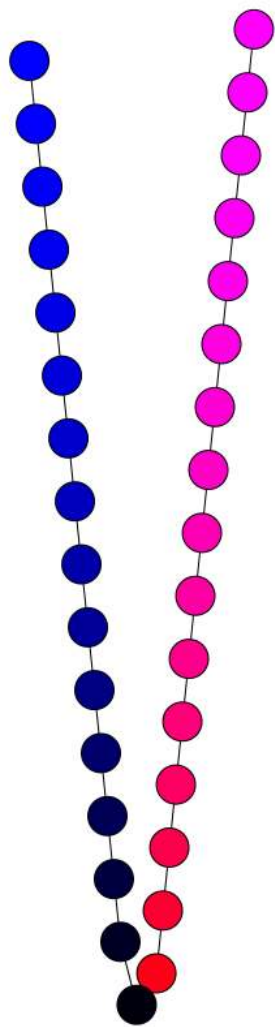
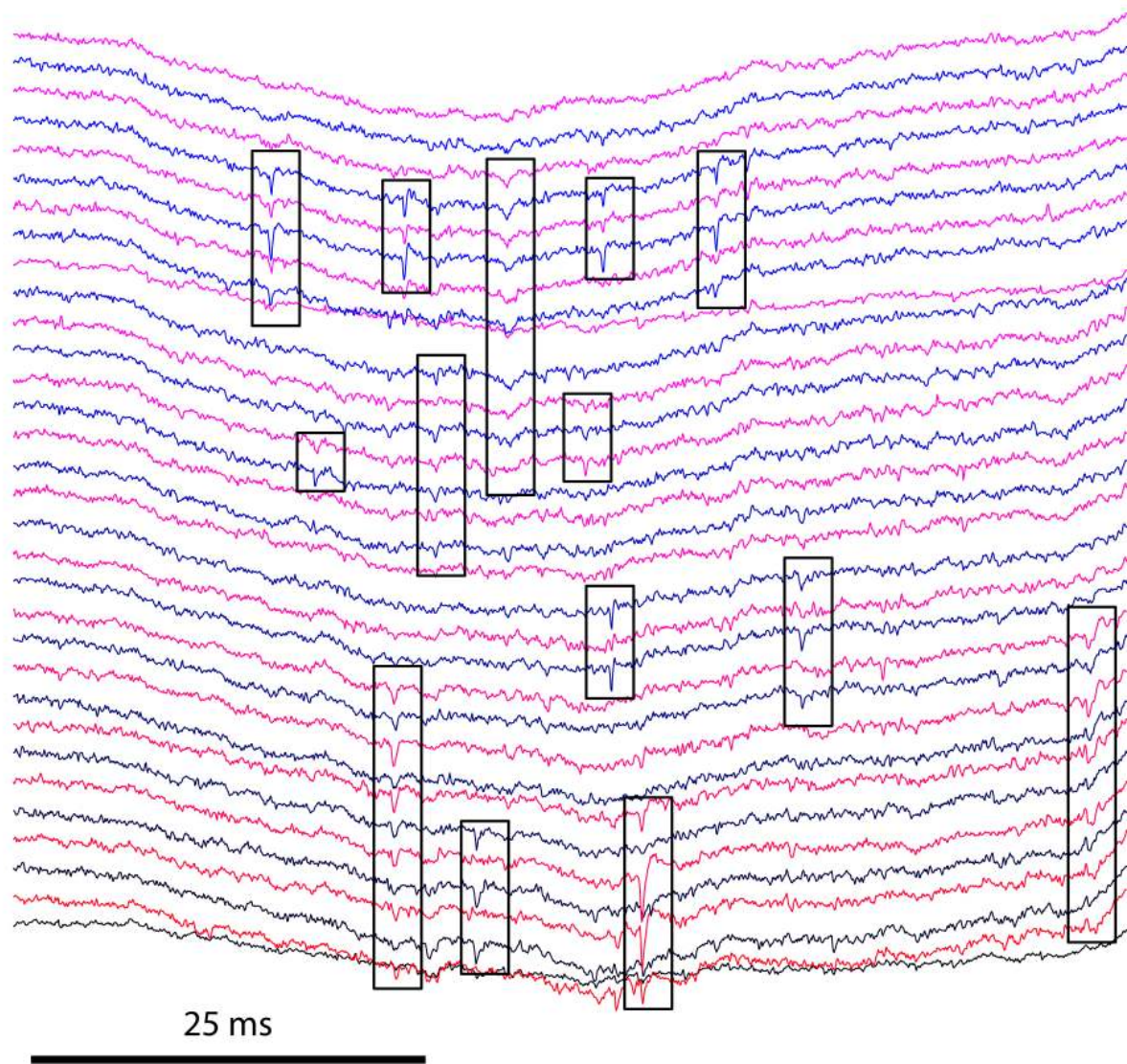
681 The difference between two clusterings C, C' , consisting of K and K' clusters, respectively, and
 682 confusion matrix entries, $n_{kk'}$ where measured using the Fowlkes-Mallows³¹ index, $\sqrt{W_1 W_2}$, where:

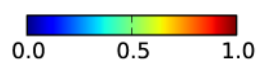
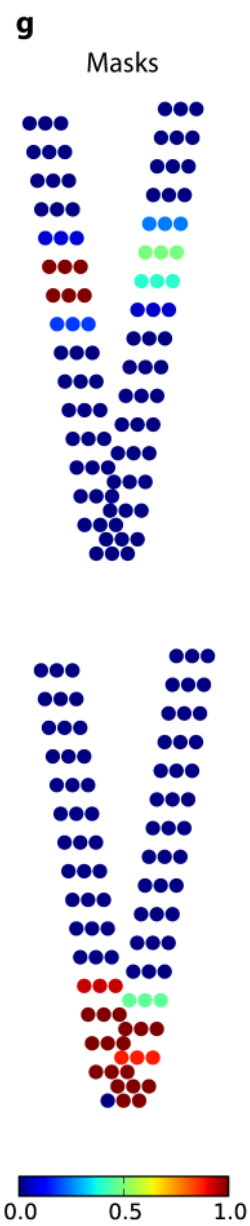
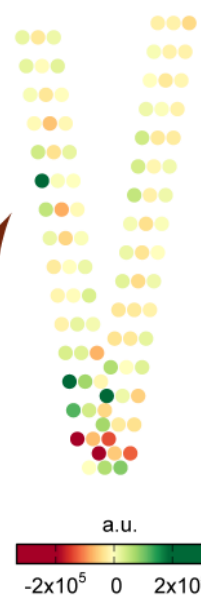
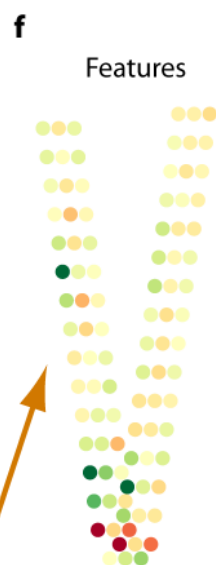
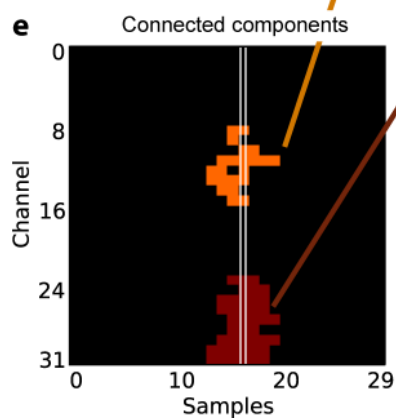
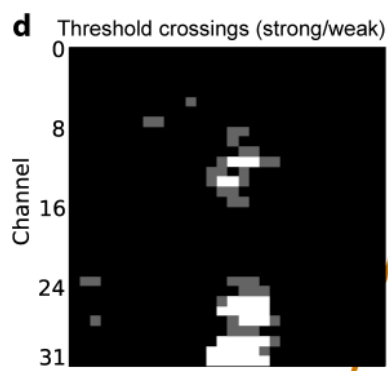
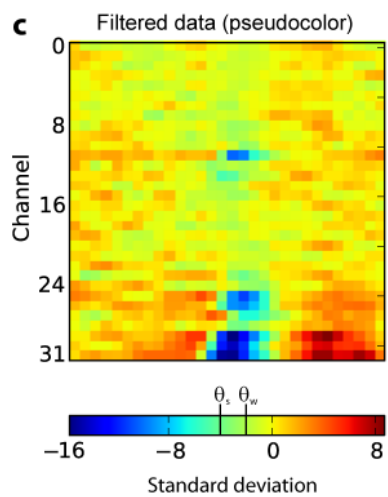
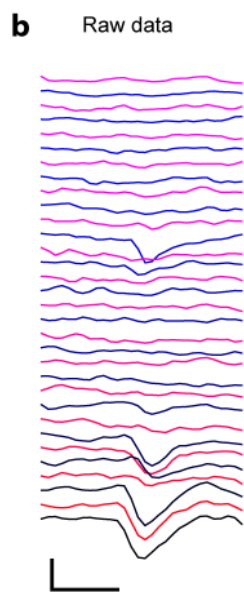
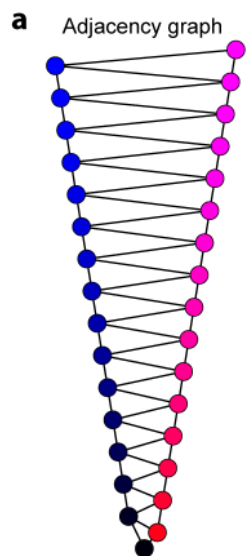
$$W_1(C, C') = \frac{\sum_{k,k'} n_{kk'}(n_{kk'} - 1)/2}{\sum_k n_k(n_k - 1)/2}, \quad W_2(C, C') = \frac{\sum_{k,k'} n_{kk'}(n_{kk'} - 1)/2}{\sum_{k'} n'_{k'}(n'_{k'} - 1)/2}$$

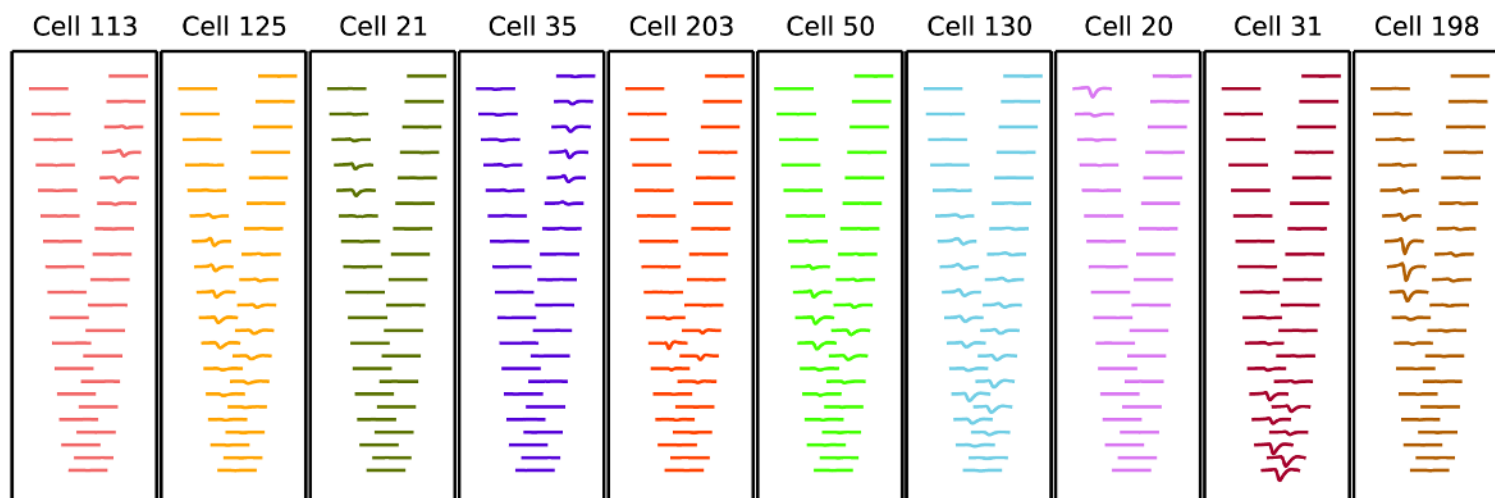
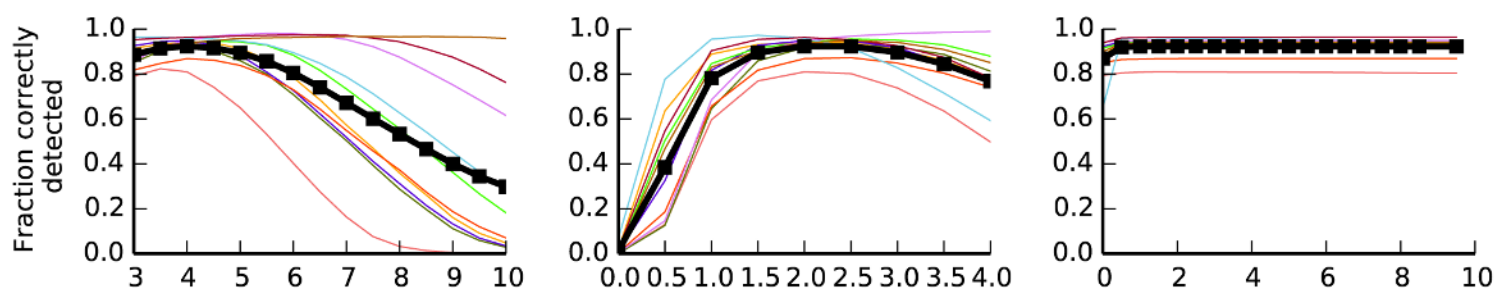
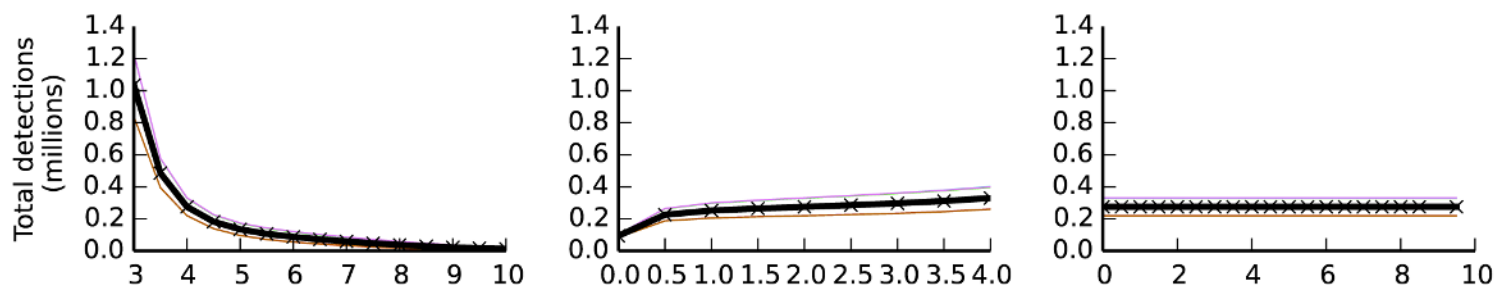
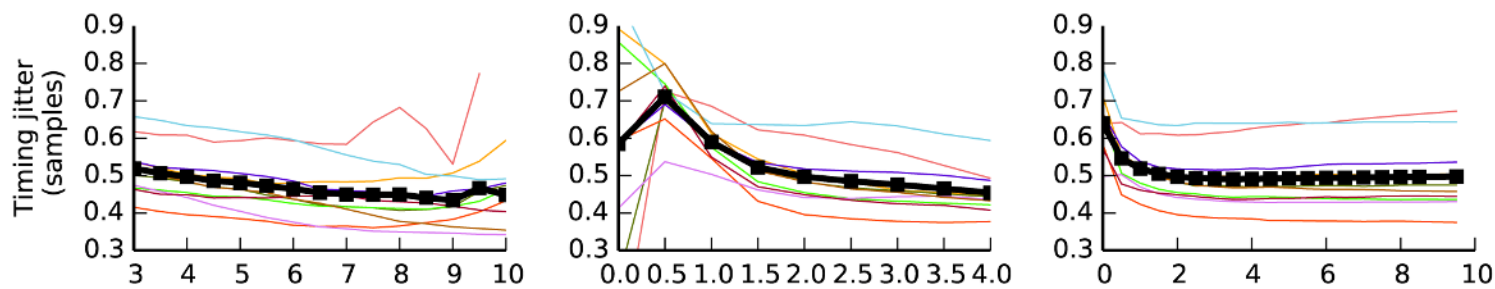
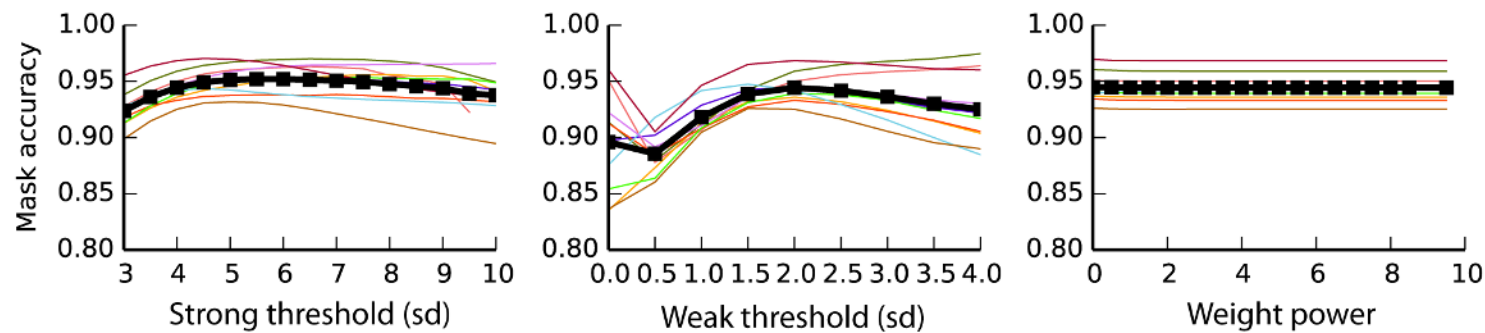
683 $n_{k'} = \sum_k n_{kk'}$, $n'_{k'} = \sum_k n_{kk'}$, $k = 1, \dots, K, k' = 1, \dots, K'$. W_1 is the probability that a pair of
 684 points which are in the same cluster under the clustering C is also in the same cluster in C' . W_2 is the
 685 same with the two clusterings interchanged. The Fowlkes-Mallows index symmetrizes these two
 686 asymmetric quantities by taking their geometric mean.

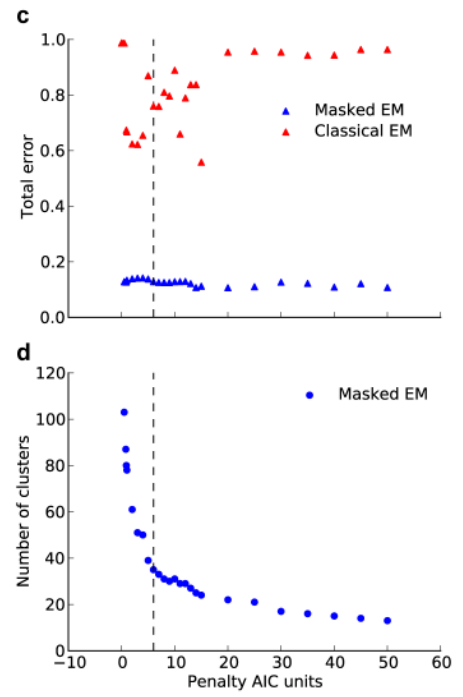
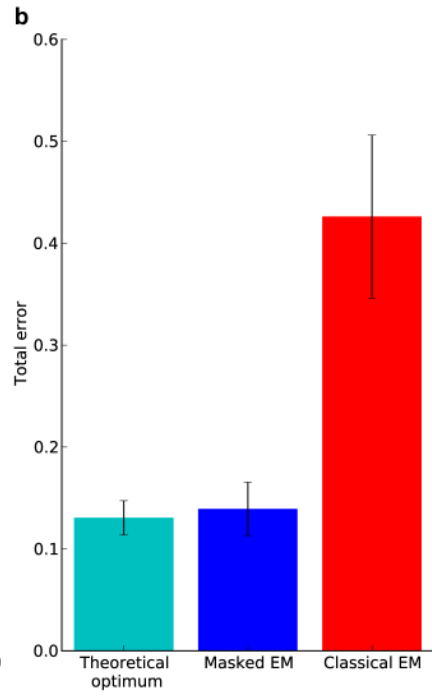
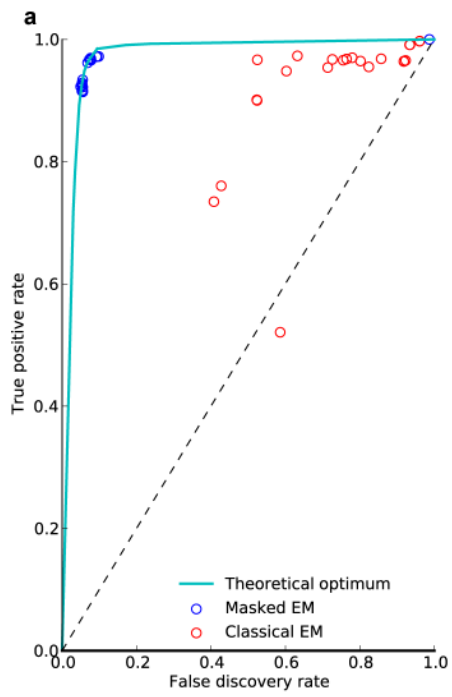
- 687 40. Saleem, A.B., Ayaz, A., Jeffery, K.J., Harris, K.D. & Carandini, M. Integration of visual motion and
 688 locomotion in mouse visual cortex. *Nat Neurosci* **16**, 1864-1869 (2013).
- 689 41. Ayaz, A., Saleem, A.B., Scholvinck, M.L. & Carandini, M. Locomotion controls spatial integration
 690 in mouse visual cortex. *Curr Biol* **23**, 890-894 (2013).
- 691 42. Ecker, A.S. *et al.* State dependence of noise correlations in macaque primary visual cortex.
 692 *Neuron* **82**, 235-248 (2014).
- 693 43. Ecker, A.S. *et al.* Decorrelated neuronal firing in cortical microcircuits. *Science* **327**, 584-587
 694 (2010).
- 695 44. Zeater, N., Cheong, S.K., Solomon, S.G., Dreher, B., Martin, P.R. Binocular responses in the
 696 primate lateral geniculate nucleus. *Curr Biol* **25**, 3190-3195 (2015).
- 697 45. The HDF Group. Hierarchical Data Format, version 5. <http://www.hdfgroup.org/HDF5>. (1997-
 698 2014).

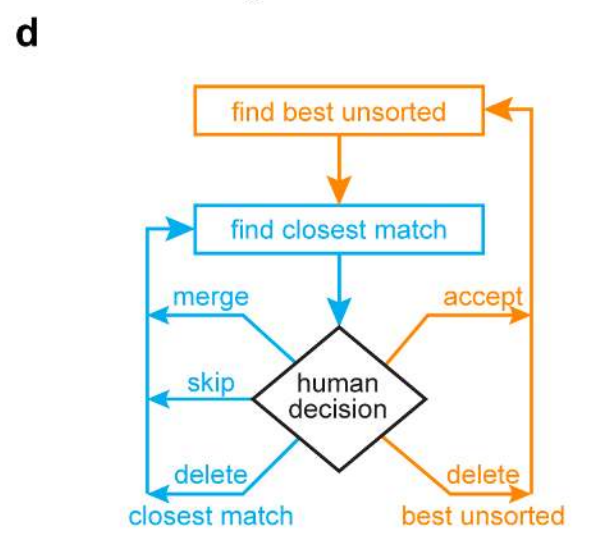
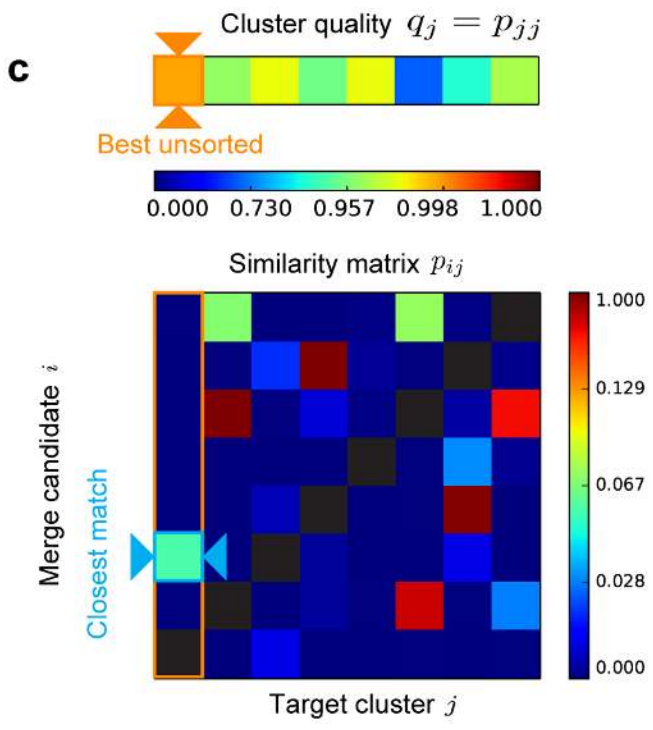
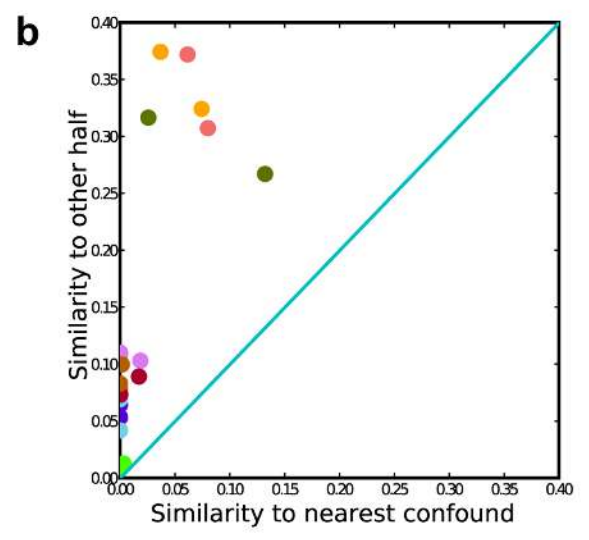
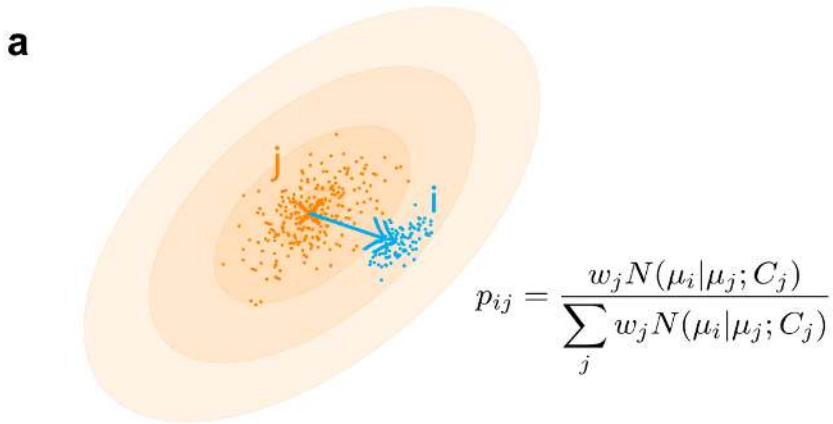
- 699 46. Rossant, C. & Harris, K.D. Hardware-accelerated interactive data visualization for neuroscience
700 in Python. *Frontiers in neuroinformatics* **7**, 36 (2013).
- 701 47. Shreiner, D., Sellers, G., Kessenich, J.M., Licea-Kane, B. & Khronos OpenGL ARB Working Group.
702 *OpenGL programming guide : the official guide to learning OpenGL, version 4.3*, Edn. Eighth
703 edition.
- 704 48. Swayne, D.F., Cook, D. & Buja, A. XGobi: Interactive dynamic data visualization in the X Window
705 System. *J Comput Graph Stat* **7**, 113-130 (1998).
- 706

a**b**



a**b****c****d****e**





ProjectionView

26 27

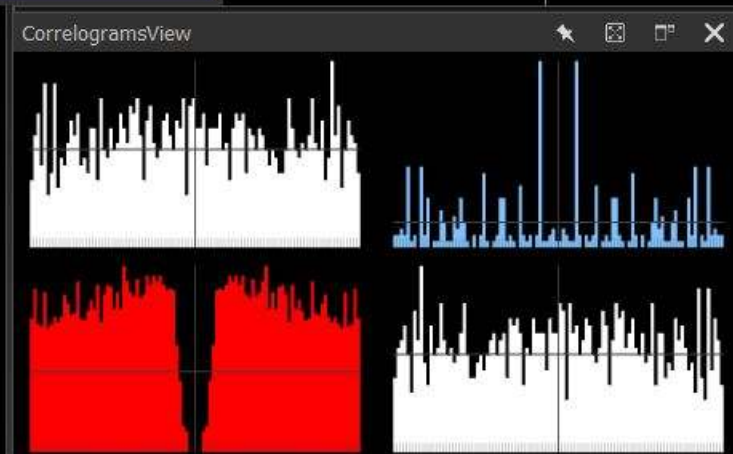
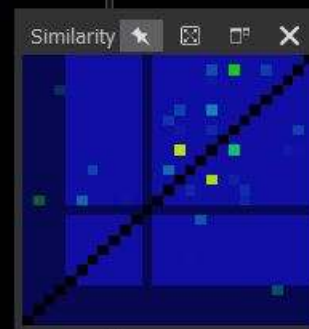
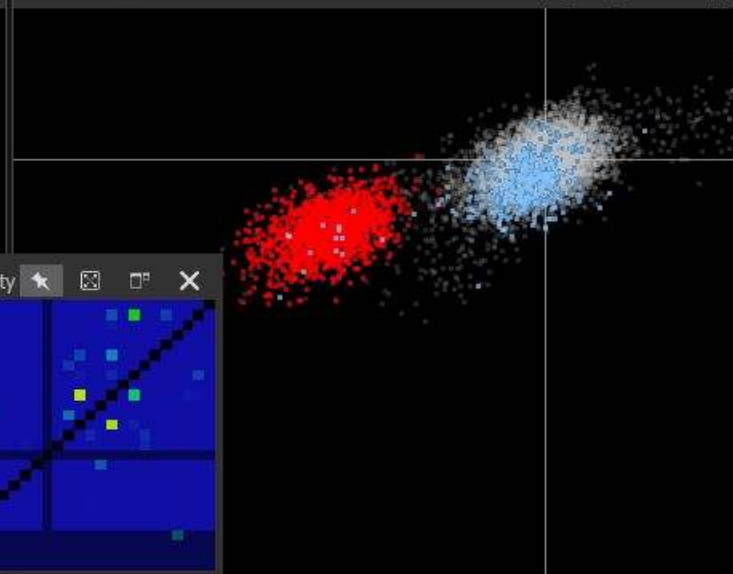
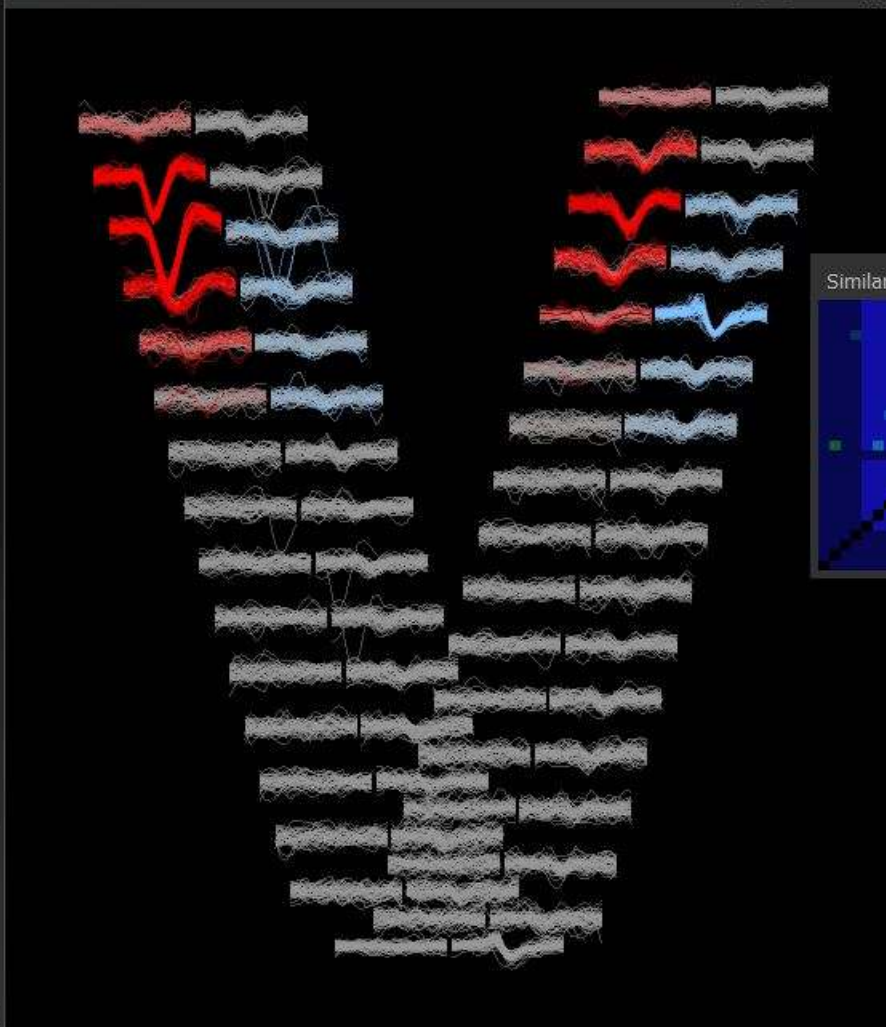
PC1 PC2 PC3 PC1 PC2 PC3

WaveformView

FeatureView

ClusterView

Cluster	Quality	Spikes	Color
MUA		876	
0	0.9994	69	
1	0.9999	550	
2	1.0000	168	
3	0.9753	88	
11	1.0000	1	
Good		5701	
4	1.0000	1496	
5	1.0000	449	
6	0.9991	383	
7	0.9995	291	
8	0.9998	128	
9	0.9997	460	
10	0.9872	920	
12	0.9347	1342	
26	1.0000	232	
Unsorted		3423	
13	0.9892	757	
14	0.9281	483	
15	0.9748	659	
16	0.9993	192	
17	0.8879	346	
18	0.9990	154	



a Experts Novices
 I O N A B | H C R
 ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✗

