Supporting Information

Friedman et al. 10.1073/pnas.1503940112

SI Materials and Methods

Although we expect our algorithm to be readily adaptable to a broad range of data, we have optimized it for use on tetrode data collected from the rat striatum and cortex. The electrophysiological data acquisition software (Neuralynx Cheetah) records 1-ms time windows triggered when the voltage crosses a threshold. In our MATLAB implementation of this algorithm, specific parameters (e.g., types of interpolation) can be entered as options in a configuration file. This approach gives the user flexibility in optimizing the algorithm for specific datasets.

Our algorithm can be divided into three main parts (Fig. 1*C*): (*i*) in preprocessing, the candidate spike data are prepared and the SNR is enhanced; (*ii*) in the core algorithm, a transformed feature space is constructed, and clusters and subclusters are found and refined. The core algorithm and part of the preprocessing are iterated on the unsorted data until a criterion is met; (*iii*) in postprocessing, nearly simultaneous candidate spikes removed during preprocessing are assigned to clusters, and cluster quality is evaluated.

Preprocessing. Our algorithm begins with several steps of preprocessing, including stationarity, spike amplitude SNR, and local density filtering, to increase the SNR and to enhance the effectiveness of the core clustering algorithm (Fig. 1 *C* and *D*). The spike amplitude SNR and local density filters are applied to the unsorted data at the beginning of each iteration of the algorithm except for the final one.

Initial Spike Selection. During data acquisition, when a 1-ms recording window is triggered by a threshold crossing, additional nearly simultaneous candidate spikes may also be recorded within the same time window. Our algorithm temporarily removes such spikes. The temporarily removed spikes are ultimately added back to clusters during postprocessing, after all of the iterations of the algorithm are completed (Fig. S1*D*).

Stationarity Filter. The algorithm removes candidate spikes in time bins containing a number of spikes more than five SDs above the mean; this removes noise consisting of rapid voltage deflections, which probably occur as a result of mechanical problems (e.g., a transient loose connection or bumping of the tetrode microdrive assembly and headstage preamplifier). This noise usually appears along the diagonal in peak vs. peak plots (Fig. S2).

Core Algorithm. The core algorithm consists of three main parts: (*i*) initial feature space construction, dimension evaluation, and spatial transformation, in which a transformed feature space is constructed by scaling dimensions (features) according to their importance; (*ii*) recursive clustering; and (*iii*) rebuilding the cluster from the core (Fig. 1 *C* and *D*). The SNR and local density filters and the core algorithm are iterated to find additional clusters. After each iteration, the resulting clusters are removed from the candidate spike data. As described in *Spike Amplitude SNR Filter* of the main text, the SNR criterion level is decreased in each iteration. In the final iteration, the algorithm omits the SNR and local density filters to look for valid clusters that might have been filtered out. Evaluation of dimensions according to their contribution to separability is an essential part of our algorithm. Evaluation refers to the determination of the relative importance of selected dimensions.

Initial Feature Space Construction. The algorithm constructs a feature space of 13 dimensions (Fig. 1*C*): four peak voltage dimensions (one for each tetrode channel), eight PC dimensions (PC1 and PC2 of the candidate spike waveforms recorded for each of the four tetrode channels), and three peak PC dimensions (PC1, PC2, and PC3 of the peak voltages on the four channels). The peak voltages on the four channels together can be viewed as a signature reflecting, in part, the location of the neuron relative to the four wires (Fig. 1 *A* and *B*).

As a final step, each dimension (peak voltage, PC, or peak PC) is independently z-score normalized. In most cases, when performing PCA, we apply a mean subtraction that focuses it on the variability in the shape of the waveform rather than on vertical shifts of the entire waveform. The initial space for each of the four channels consists of *n* dimensions, where each dimension is the voltage at one time point of the interpolated, peak-aligned, and padded waveform within the 1.25-ms time window. The waveform recorded in each channel is subtracted by the mean of the points within that waveform. This procedure contrasts with the built-in MATLAB function pca, which would subtract the waveform in each channel by the mean of all waveforms (over time) in that channel. This MATLAB version of PCA would be vulnerable to variability involving vertical shifts of the whole waveform. We attribute most of this type of variability to a background of multiunit activity. As a result of this variability, the MATLAB version of PCA would not be focused on the variability in waveform shape that is relevant to clustering, whereas our version of PCA is able to focus on this variability.

We use the MATLAB version of PCA only for calculating peak PC. In this case, each channel contributes only one dimension to the initial 4D space—namely, its peak voltage. Because the vertical shifts are expected to be different for the four channels, subtracting the mean peak voltage of the four channels would not be helpful.

Dimension Evaluation. MPC, an index of cluster validity, is an adaptation of the partition coefficient defined by Bezdek (1, 2) that reduces its monotonic tendency with respect to c (3). The MPC value m is defined as

$$m = 1 - \frac{c}{c-1}(1 - V_{PC}),$$

where V_{PC} is the partition coefficient defined by Bezdek:

$$V_{PC} = \frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n}u_{ij}^{2}.$$

Here, n is the number of vectors (e.g., spikes) in the multidimensional space, c is the number of clusters, and u_{ij} refers to the entries in the partition matrix U. For our case, u_{ij} gives the degree of membership of spike j in cluster i on a scale from 0 to 1.

Fuzzy C-Means Clustering. Having prepared a suitable space, the algorithm runs FCM clustering multiple times on the distribution in this space, assuming different numbers of clusters in each run. The correct number of clusters is the number that maximizes the MPC value (Fig. S4). Instead of randomly initializing FCM, we use a heuristic that deterministically produces the initial centers of each cluster. Our heuristic does not guarantee a global minimum for the FCM objective function.

Recursive Clustering. Following the first round of clustering, our algorithm identifies clusters within clusters by applying the same methods used in the initial clustering. PCA is applied to just the candidate spikes within the cluster. Dimensions are again selected and evaluated. If one or more dimensions meet the criteria of being potentially separable and having nonzero dimensional importance, the algorithm applies the same clustering method to the spikes within the cluster; it repeats this subclustering recursively until there are no dimensions that meet the criteria.

Finding Cluster Cores. The clusters found by the algorithm up to this point do not have the best possible boundaries, due to the removal of valid spikes by the strong filters and the limitations of the FCM algorithm. The algorithm reduces each cluster to a core that normally contains the 30% of the cluster's spikes closest to the centroid. The algorithm uses Euclidean distances for this purpose, because it is less affected by errors in FCM clustering.

Rebuilding Clusters from Cores. Along each dimension, the algorithm looks for a valley in the distribution of Mahalanobis distances and includes spikes in the cluster that are nearer to the cluster core than the valley. However, if a cluster is very small, up to 60% of the spikes closest to the centroid may be used, according to this formula in MATLAB:

$$\begin{aligned}\text{upper\_bound} = \min(&\text{round}(0.6\ {}^{*}\ \text{num\_spikes}),\\ &\times \max(200, \text{num\_core\_spikes}));\end{aligned}$$

where upper_bound is the number of spikes in the core, num_spikes is the number of spikes in the cluster, and num_core_spikes is 30% of num_spikes.

Removing Bad Clusters. Each dimension is tested individually. A cluster is removed if it is evaluated as potentially separable along any dimension according to the following test (Fig. S5*A*): Let v be any local minimum in the projection of the distribution onto a dimension. Let p be the largest local maximum on either side of v. If $v < \frac{3}{4}p$ for any v, the data are potentially separable along this dimension. For larger clusters (>10,000 spikes), we consider any local minimum not too near the peak, in a peak radius of 10 indexes (Fig. S5 *B* and *C*), to make the cluster bad.

Assigning Nearly Simultaneous Spikes to Clusters. Spike waveforms of different neurons that were recorded by the same tetrode within the same 1-ms window were temporarily removed. The waveforms are separated from each other, and each waveform is given an appropriate timestamp. Each waveform is then assigned to a cluster based on its peak voltages recorded on the four tetrode channels.

1. Bezdek JC (1981) Pattern recognition with fuzzy objective function algorithms. *Advanced Applications in Pattern Recognition* (Plenum, New York), pp 1–13.

2. Wang WN, Zhang YJ (2007) On fuzzy cluster validity indices. *Fuzzy Sets Syst* 158(19): 2095–2117.

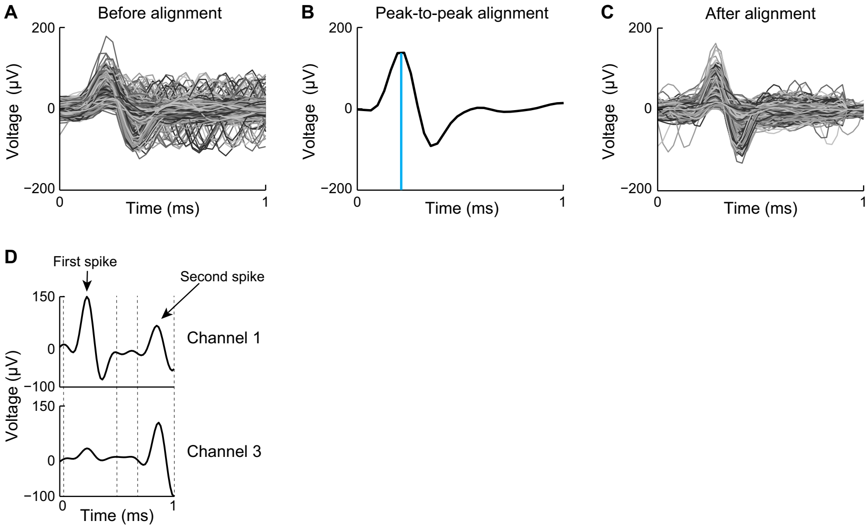

Fig. S1. Spike alignment and nearly simultaneous spike separation stationarity filter. (*A*) Raw data containing misaligned spikes. (*B*) Spikes undergo peak-to-peak alignment. (*C*) The data, after alignment, have a simplified structure that can now be combined in a unified waveform. (*D*) The unified waveform shows two spikes recorded in the 1-ms interval. The dashed lines indicate the beginning and end of each spike.

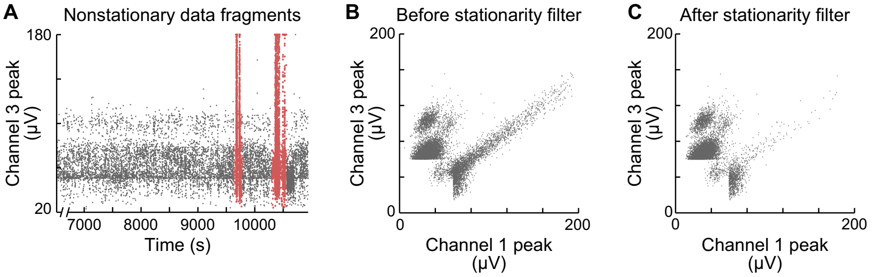

Fig. S2. Stationarity filter. (*A*) Nonstationary time periods are shown in red during a sample recording period. (*B*) A 2D projection of peaks exhibiting, along the diagonal, nonstationary noise. (*C*) The same 2D projection after removing the nonstationary recording fragment.

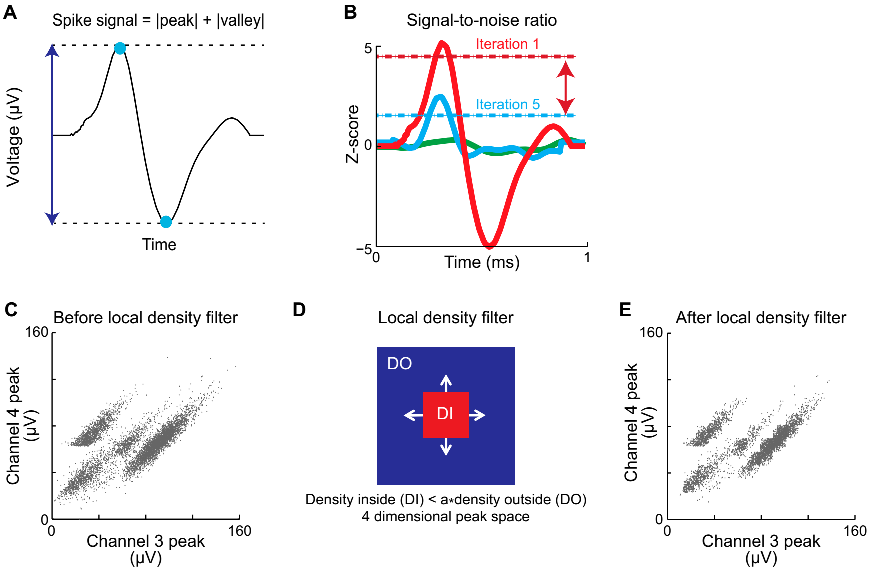

Fig. S3. SNR filter and local density filter. (*A*) The peaks and valleys of waveform, shown here as blue dots on a representative waveform, are identified in recorded spikes. (*B*) Three spikes with different SNRs. The red spike was found in iteration 1, and the blue spike was found in iteration 5. The green spike was identified as a multiunit cluster in iteration 5. (*C*) A 2D projection of a recorded dataset. (*D*) A conceptualization of a local density filter. The densities inside of the bin (DI) and outside of the bin (DO) are calculated. If the density inside of the bin is significantly lower than outside density, the spikes inside the bin are removed. (*E*) A 2D projection after the application of the local density filter to the data shown in *C*.

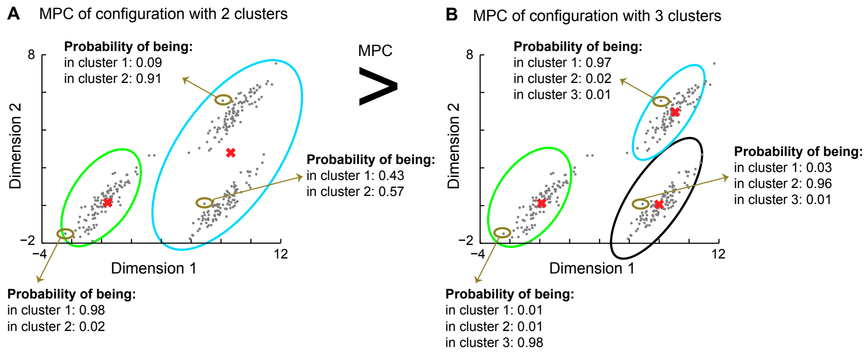

Fig. S4. FCM and MPC procedures. The dataset is clustered, using FCM, to yield different numbers of clusters. Ellipses are drawn around the cluster borders, and the each cluster center is marked with a red X. Due to the FCM approach, each point has a probability of inclusion in each of the clusters. MPC then ranks each clustering configuration, giving preference to cluster configurations that maximize each point's affinity to one cluster. The clusters provided by an FCM algorithm are shown for two clusters in *A* and for three clusters in *B*.

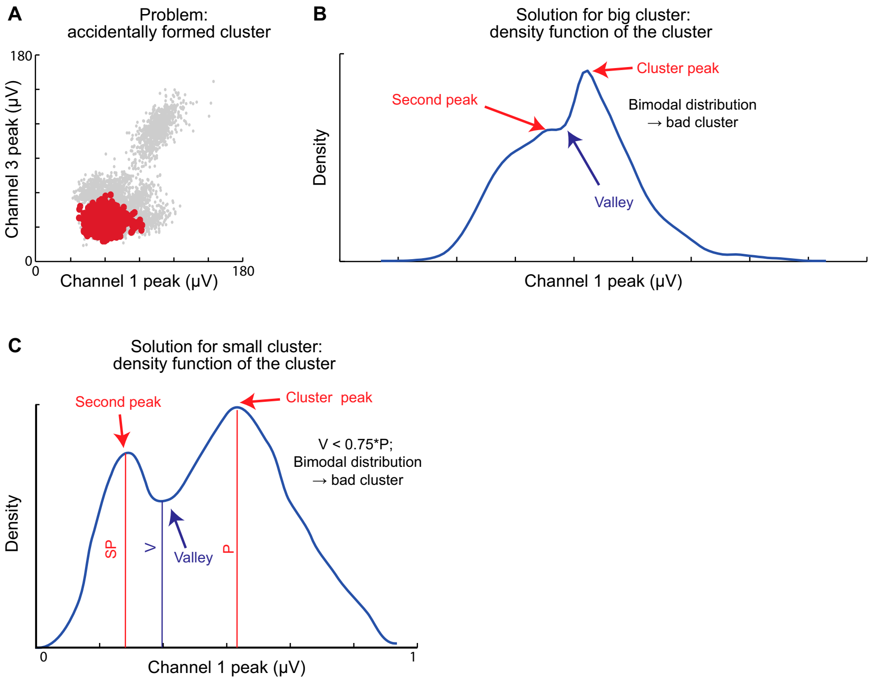

Fig. S5. Cluster lacking a normal distribution. (*A*) An example of a cluster (red) identified in the midst of multiunit activity (gray). (*B*) The distribution of this cluster is not normal. There is an additional second peak identified in the distribution. (*C*) Another example of a bimodal cluster density distribution.

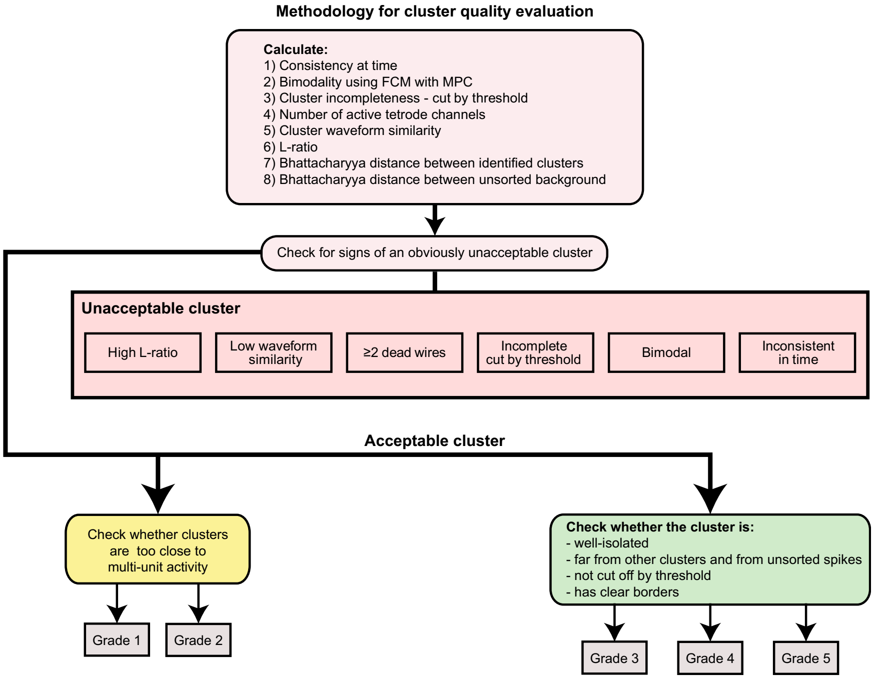
Methodology for cluster quality evaluation
Calculate:
1) Consistency at time
2) Bimodality using FCM with MPC
3) Cluster incompleteness - cut by threshold
4) Number of active tetrode channels
5) Cluster waveform similarity
6) L-ratio
7) Bhattacharyya distance between identified clusters
8) Bhattacharyya distance between unsorted background
Check for signs of an obviously unacceptable cluster
Unacceptable cluster
High L-ratio
Low waveform similarity
≥2 dead wires
Incomplete cut by threshold
Bimodal
Inconsistent in time
Acceptable cluster
Check whether clusters are too close to multi-unit activity
Grade 1
Grade 2
Check whether the cluster is:
- well-isolated
- far from other clusters and from unsorted spikes
- not cut off by threshold
- has clear borders
Grade 3
Grade 4
Grade 5

Fig. S6. Methodology for evaluation of cluster quality. After cluster attributes are calculated, unacceptable clusters are identified. Of the acceptable clusters, clusters with multiunit activity receive grades of 1 and 2, and separable clusters receive grades of 3, 4, and 5.

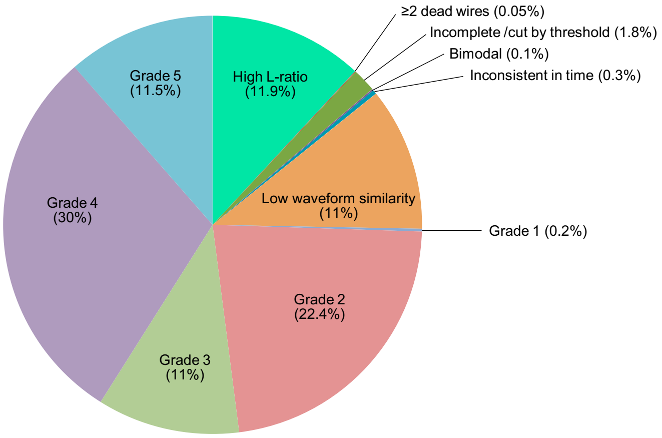

Fig. S7. Cluster grade distribution for spike sorting performed by experts. The percentage of identified clusters is shown. Categories shown on the upper right of graph represent different types of artifacts. Grades 1 and 2 correspond to multiunit activity. Grades 3, 4, and 5 are separable recordings.

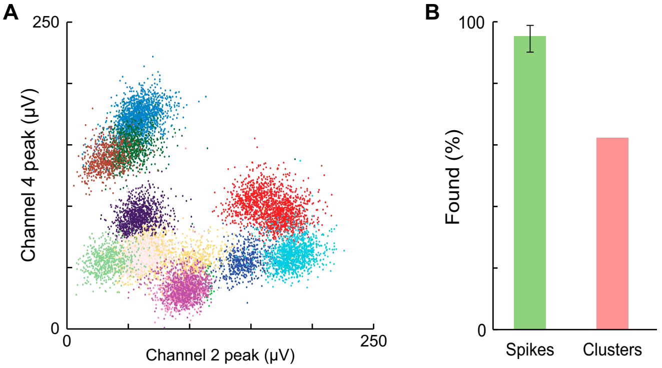

Fig. S8. Simulation-generated data containing 16 clusters with high level of overlap, similar to those shown in Fig. 6*H*. (*A*) An example of simulation data in which 16 clusters were partially identified by the algorithm. (*B*) The percentage of simulation spikes assigned to correct clusters by the algorithm (green) and percentage of simulation clusters accurately identified by the algorithm (red). Error bar represents SDs across simulation clusters.