

t-SNE Visualization of Large-Scale Neural Recordings

George Dimitriadis*

g.dimitriadis@ucl.ac.uk

Joana P. Neto*

j.neto@ucl.ac.uk

Adam R. Kampff

a.kampff@ucl.ac.uk

Sainsbury Wellcome Centre, UCL, London W1T 4JG, U.K.

Electrophysiology is entering the era of big data. Multiple probes, each with hundreds to thousands of individual electrodes, are now capable of simultaneously recording from many brain regions. The major challenge confronting these new technologies is transforming the raw data into physiologically meaningful signals, that is, single unit spikes. Sorting the spike events of individual neurons from a spatiotemporally dense sampling of the extracellular electric field is a problem that has attracted much attention (Rey, Pedreira, & Quiari Quiroga, 2015; Rossant et al., 2016) but is still far from solved. Current methods still rely on human input and thus become unfeasible as the size of the data sets grows exponentially. Here we introduce the *t*-student stochastic neighbor embedding (t-SNE) dimensionality reduction method (Van der Maaten & Hinton, 2008) as a visualization tool in the spike sorting process. t-SNE embeds the *n*-dimensional extracellular spikes (*n* = number of features by which each spike is decomposed) into a low- (usually two-) dimensional space. We show that such embeddings, even starting from different feature spaces, form obvious clusters of spikes that can be easily visualized and manually delineated with a high degree of precision. We propose that these clusters represent single units and test this assertion by applying our algorithm on labeled data sets from both hybrid (Rossant et al., 2016) and paired juxtacellular/extracellular recordings (Neto et al., 2016). We have released a graphical user interface (GUI) written in Python as a tool for the manual clustering of the t-SNE embedded spikes and as a tool for an informed overview and fast manual curation of results from different clustering algorithms. Furthermore, the generated visualizations offer evidence in favor of the use of probes with higher density and smaller electrodes. They also graphically demonstrate the diverse nature of the sorting problem when spikes are recorded with different methods and arise from regions with different background spiking statistics.

*G.D. and J.N. contributed equally to this letter.

1 Introduction

It is neuroscience dogma that the brain's computational mechanics are implemented by the complex dynamics of its spiking neural networks. As a consequence, detailed knowledge of the spiking activity for as many neurons as possible during behavior is seen as essential to understand how the brain receives and transforms information. Electrophysiological methods that record spiking activity extracellularly have been one of the most significant tools for exploring the correlations between behavior and neural activity. Using these tools, there has been a constant drive to record from more neurons, for longer times, from a host of brain regions, in diverse physiological conditions, and from many different species. This trend was recently accelerated by new microfabricated recording probes that extend the standard single electrode and tetrode devices (Recce and O'Keefe, 1989) with integrated electronics to produce devices with thousands of recording sites (Alivisatos et al., 2013; Jun et al., 2017; Raducanu et al., 2017; Ruther & Paul, 2015). The new generation of recording tools brings with it the challenge of extracting meaningful physiological signals from the resulting (big) data sets. In the case of extracellular probe recordings, that usually means transforming the voltages measured at the electrode sites into spiking activity of the nearby neurons. The importance of accurate spike sorting stems from a number of ideas on how cell spiking contributes to brain functions. For example, competent sorting is required to test for sparse coding in memory function (Chaudhuri & Fiete, 2016) or to assess the diverse responses of neighboring cells, important in theories of concept (Rey, Ison et al., 2015) and place cells (Redish et al., 2001).

The original attempts to spike-sort greatly benefited from the development of the tetrode and its ability to simultaneously monitor the spiking signal of nearby neurons from multiple locations (i.e., four) (Fee, Mitra, & Kleinfeld, 1996; Gray, Maldonado, Wilson, & McNaughton, 1995; Wehr, Pezaris, & Sahani, 1999). It has since become clear that dense electrode configurations, in which that same neuron is detected by multiple electrodes, generally improve sorting (Buzsáki, 2004; Lewicki, 1998). This has driven the push for an increase in the electrode density of modern probe designs. Today, new methodologies have evolved to work with the next generation of multielectrode probes addressing the problem of exploding data set size and complexity (Rey, Pedreira, & Quian Quiroga, et al., 2015; Rossant et al., 2016). However, the basic idea of the spike-sorting pipeline remains the same (see Figure 1A). The (filtered) data go through a process of spike detection that has traditionally relied on thresholding the raw signal. The multiunit activity generated is then passed through a dimensionality-reduction method that transforms the space-time spike matrices into a smaller set of features. The most commonly used dimensionality-reduction techniques for offline sorting are principal component analysis (PCA) (Harris, Henze, Csicsvari, Hirase, & Buzsáki, 2000) and wavelet decomposition (Hulata,

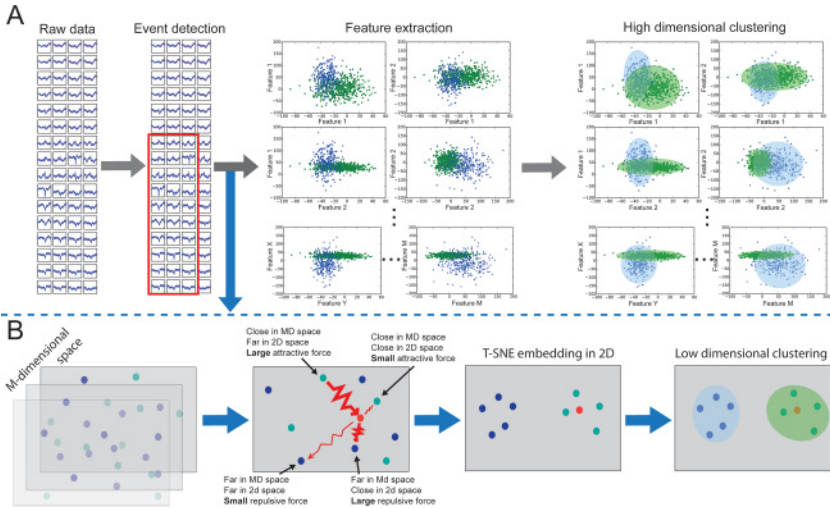


Figure 1: Graphical representation of (A) the current pipeline for spike sorting and (B) the way t-SNE can be added as a visualization tool in this pipeline.

Segev, & Ben-Jacob, 2002; Quiroga, Nadasdy, & Ben-Shaul, 2004; Takekawa, Isomura, & Fukai, 2010). For online sorting, geometric/spike shape methods (Gerstein & Clark, 1964; Lewicki, 1998) are mainly used. More recent approaches even combine the two offline methods to generate an optimum set of features for further analysis (Rey, Pedreira et al., 2015). Finally, a clustering method is employed to automatically group together the spikes from an isolated single unit in the high-dimensional space of the decomposed features. Techniques commonly used for this clustering are k-means (Wood, Black, Vargas-Irwin, Fellows, & Donoghue, 2004), mixtures of gaussians based on an expectation-minimization algorithm (Wood, Fellows, Donoghue, & Black, 2004), and template matching (Wang, Zhou, Chen, Zhang, & Liang, 2006; Zhang, Wu, Zhou, Liang, & Yuan, 2004). An overview of different techniques for detection, feature extraction and classification is given in Bestel, Daus, and Thielemann (2012). Methods that are currently under development follow a different route where the event detection, the feature extraction, and the clustering steps are realized in a single template matching step (Pachitariu, Steinmetz, Kadir, Carandini, & Harris 2016; Yger et al., 2018). These methods offer better parallelization capabilities and are proving very capable in handling millions of spikes arising from recordings of hundreds to thousands of channels.

In all cases, the automated clustering algorithms operate on a number of dimensions that scale linearly with the number of channels of the recording probe. For the more recent multichannel probes, this feature space usually

contains hundreds of dimensions. Such multidimensional spaces make either manual clustering, or the manual supervision and quality assurance of the automated algorithms' results, prohibitive. The t-SNE dimensionality-reduction technique was designed to reduce such multidimensional data sets to two or three dimensions in a way that visualizing them can offer meaningful insight into their original high-dimensional structure (Van der Maaten & Hinton, 2008). Embedding techniques like t-SNE transform the position of points in a high-dimensional space to positions in a lower- (usually 2-) dimensional space. This reduction transformation obviously requires that some information is lost. Each embedding technique decides which aspects of the original structure to keep and which to ignore. t-SNE focuses on ensuring that the local structure (i.e., the ordering of distances between nearby points) remains intact while it ignores the global structure (i.e., the large distances in the t-SNE space are not representative of the large distances in the original space). A good mental representation of how t-SNE achieves this is to think of all points as objects connected to each other with spring-like forces. In the original space, these forces are in equilibrium. When the points are transferred (randomly at first) into the 2D space, the forces between them start both pulling and pushing so that a new equilibrium might be reached (see Figure 1B). Points that are close in the original space are attracted to each other until they get roughly equally close in the 2D space. At the same time, points that are far away in the original space are repulsed by each other if they happen to find themselves close in the 2D space. This ability of the t-SNE algorithm to repulse points that are nearby in the 2D space but not in the original space offers a solution to the crowding problem of other embedding methods. This underlies the informative 2D plots that it generates.

In this work, we apply t-SNE to the spike-sorting process and generate 2D plots that show obvious clusters of spikes. We use two types of data to validate our technique. The first is a ground-truth data set that comes from paired recordings (Neto et al., 2016) with an extracellular and a juxtacellular probe, thus providing labels from the juxtacellularly recorded unit within the extracellular probe's spiking activity. The second type is a hybrid data set generated from the synthesis of real extracellular recorded data with manually superimposed spikes belonging to a number of single units (Rossant et al., 2016). In the following, we demonstrate that many of the t-SNE-generated low-dimensional clusters represent the activity of single units, while others group together spikes arising from a large number of putative units and likely noise. We develop a graphical user interface (GUI) that allows the fast visual identification of the single-unit clusters and report on how accurately the manually selected clusters represent the labeled single units. We then use the visual representations of spike clusters that t-SNE generates to offer an overview of how the sorting/clustering difficulty increases with decreasing electrode density. We enhance the results of this density effect analysis by showing the t-SNE embedding of

a juxtacellular-tetrode recording (Harris et al., 2000). This shows how a small number of electrodes present issues in distinguishing between different units. Utilizing the input agnostic nature of t-SNE, we use it to embed the results of a template-matching algorithm (kilosort) applied to the same ground-truth data set. We subsequently use the GUI to overview and manually correct kilosort's results. This shows that t-SNE's 2D embedding visualization makes digesting and curating the high-dimensional output of automated spike clustering algorithms a simple procedure. At the same time, this visualization provides a satisfying overview of the otherwise overwhelmingly large, high-dimensional data sets. Finally, to expand on this last point and demonstrate the scalability of the method, we show a t-SNE embedding from a data set of 1 million spikes collected from a state-of-the-art CMOS probe with 908 active electrodes on which the kilosort detected 579 putative units. We conclude with a discussion of possible extensions and future use cases of the t-SNE algorithm for sorting and visualization of large-scale spike recordings.

2 Methods

2.1 Data Sets. We used four data sets (conceptually split into two groups—one with ground-truth data sets and one with hybrid ones) to quantitatively test the efficacy of our methods and two separate ones to qualitatively expand on certain points that arose from the use of t-SNE on the first four data sets. The first group of two sets consisted of recordings from the anaesthetized rat's (motor) cortex (Neto et al., 2016). In these data sets the extracellular probe's recording was paired with a juxtacellular recording with a pipette. There were two different types of extracellular probes used in different sessions: a 32-channel staggered array (A1 × 32-Poly3-5mm-25s-177-CM32, NeuroNexus, U.S.A.) and a dense 128-channel matrix developed by the collaborative NeuroSeeker project (<http://www.neuroseeker.eu>). The 128-channel probe design is shown in Figure 4 and the 32 channel design in supplementary Figure 1. From the paired recording sessions available (www.kampff-lab.org/validating-electrodes), we selected one from the 32-channel probe (paired data 32: PD32) and one from the 128-channel probe (paired data 128: PD128). Those were the data sets in which the juxtacellularly recorded neuron was close enough to the extracellular probe to have its spikes easily detected on the extracellular recording without any spike-triggered averaging.

The PD128 set, when spikes were detected with $6.5\times$ standard deviations threshold, consisted of 128,820 spikes. Of these, 4420 were spikes belonging to the juxtacellularly recorded neuron (out of a total 4998 juxtacellularly recorded spikes). These are the spikes we used in the analysis shown in Figures 2 and 4. At 4.5 SD, it consisted of 255,026 spikes, of which 4775 were the juxtacellularly recorded ones. When the same data set was put through the kilosort algorithm (where no detection through thresholding is

actually used), it gave back 466,313 putative spikes out of which we defined as nonspike noise 139,073 (see section 3.4). That left 327,240 spikes (72,214 more than the 4.5 SD threshold detection). The PD32 set consisted of 73,814 spikes of which 331 belonged to the juxtacellularly recorded neuron (using an SD of 6.5 for the detection of extracellular spikes).

For the second group of data sets, we used two hybrid data sets provided to us by the lab of Kenneth Harris. Their construction was based on recordings of a 129-channel probe and the superposition of previously sorted spikes derived from separate recordings (hybrid spikes). The details of how these hybrid time series were constructed can be reviewed in Rossant et al. (2016). The first data set (HD1) had 86,271 spikes with 7 hybrid spike sets coming from different (labeled) neurons ranging in size from 432 to 26,043 spikes. The second data set (HD2) had 126,102 spikes, again with 7 groups of labeled hybrid spikes with group sizes between 442 and 24,987.

For further exploration and understanding of the results of the t-SNE algorithm on spike data, we used two more data sets. One was a juxtacellular-tetrode paired recording (PD4) from the data sets described in Harris et al. (2000): cell d5331. We used KlustaKwik to both detect and cluster the spikes in this data set (with a spike detection threshold of 4.5 SD). This provided a total of 4387 spikes, out of which 842 were also present on the juxtacellular probe (99.7% of the juxtacellular spikes). The KlustaKwik clustering algorithm generated three clusters from this data set running with default parameters and no further manual sorting.

The last data set (NS) was from an acute rat recording utilizing a new, very high density CMOS probe developed by the NeuroSeeker project (Raducanu et al., 2017). This NeuroSeeker probe has 1440 active electrodes, each $20 \times 20 \mu\text{m}$ square, with a $22.5 \mu\text{m}$ pitch, arranged in four columns spanning a total area of $8000 \times 100 \mu\text{m}^2$ (see Figure 6 for a schematic). The data set used here was from an auditory, hippocampal, medial geniculate nucleus implantation we conducted that used 908 active electrodes. For spike detection and initial spike sorting, we used the kilosort algorithm, which resulted (after cleaning of the templates that were obviously noise) in 1,091,229 spikes and 579 templates.

2.2 t-SNE Code. For all our experiments, we used an extended version of the GitHub available C++ t-SNE code by Van der Maaten (<https://github.com/lvdmaaten/bhtsne/>). The bhtsne implementation in this repository allows for faster computations and makes larger data sets feasible through the use of the Barnes-Hut (BH) algorithm (Van der Maaten, 2014). The BH algorithm groups together distant samples into a single average sample, minimizing the number of Euclidean distance computations the algorithm must perform. This step makes the calculation of the actual error minimization of the 2D landscape faster and tractable for large sample numbers (on the order of millions). Yet the calculation of the initial probabilities in the multidimensional space still requires the algorithm to calculate

the Euclidean distances of all sample pairs, adding, for large sample numbers, a significant amount to the algorithm's run time.

To speed up this part, we implemented the computation of the Euclidean distances between all sample pairs directly on the GPU. To do this, we used the GPU Euclidean distance implementation presented by Chang, Jones, Li, & Ouyang (2008). This allows us to run data sets of more than 10^5 of spikes on a gaming desktop (i7 CPU, Titan X GPU, 64GB RAM) in just a couple of hours (see supplementary Table 1 about times for different numbers of samples and perplexity settings). However, storing the distances for all sample pairs on RAM soon imposes a bottleneck on the number of samples that can be used. For N samples, one requires $4 * N * N$ (for 32-bit floats) bits of RAM, which translates to 40 GB for 10^5 spikes and to 1 TB for $0.5 * 10^6$ samples. To overcome this bottleneck, we added the possibility of calculating the pairwise distances in groups and keeping in memory only the ones that the BH algorithm will use (populating the tree structure used in the algorithm with GPU precalculated distances). This process requires a sorting step of the distances that increases the time required but allows much larger sample numbers to be used. For data sets that are prohibitively large and to show that online sorting of samples is possible, we extended the algorithm to be able to position samples on a t-SNE precomputed 2D landscape. This is done by measuring the Euclidean distance of each new sample to the samples already passed through the t-SNE algorithm. Then the extra samples' positions on the 2D landscape are calculated as the average of the positions of the closest five original samples for each new sample. For spike sorting, given that the number of spikes passed through the t-SNE algorithm offers a complete representation of the spiking units in the recording, we show that the algorithm correctly places the extra samples (see supplementary Figure 2).

We also extended the Python wrapper that came with the original code to accommodate the use of the extra parameters our C++ t-SNE function requires and to allow the user to choose between the use of the C++ executable or the CPU only SciPy implementation of t-SNE. The latter allows users without access to the CUDA library, or with hardware not capable of supporting our code, to still run a full t-SNE spike sorting session only on CPU and fully in Python (no C++ executable is called with this option). This option of course allows only a small number of spikes and incurs large run times.

2.3 Spike-Sorting Pipeline. We applied the t-SNE algorithm in two separate points of the spike-sorting pipeline (postspike detection and post-clustering) using two different spike representation feature sets as inputs to the algorithm. In the first instance (see the results in Figures 2, 3, and 4), the algorithm operated on the masked PCA components of the spikes detected as described in Rossant et al. (2016). In order to facilitate the input of these components to t-SNE, we wrote Python code that accepts the (masked or

not) PCA components of the detected spikes, transforms these into a data set ready for the t-SNE algorithm, and calls the executable on it. We have also tried using raw data or nonmasked PCA components as input to the t-SNE algorithm but found the masked PCAs to consistently outperform these other inputs (results not shown). The second input feature set to the algorithm (see the results in Figure 5) was the distance to templates generated by the kilosort algorithm. In this case, the t-SNE algorithm operated on feature vectors that measured how near or far each spike was to the set of spike templates that kilosort generated from the data.

We also developed a minimal, and easily extensible, GUI that allows users to visualize the results of the t-SNE algorithm and use this visualization to manually sort the detected spikes. The GUI offers a number of views and tools for manual spike sorting. These include a 2D scatter plot of the t-SNE results with several ways to select groups of spikes directly on the plot. A view to preview the selected group's average time traces for all channels. The presented data are not explicitly filtered, but the baseline is subtracted using the first 10 samples. An autocorrelogram view of the selected spikes with 1 ms resolution bins. A heat map view of the average difference between the minimum and the maximum values in the spikes' time window superimposed on the probe's layout diagram. A way to label a selected group and store it as a cluster in a pandas structure (saved to disk as a pickle). The GUI also allows selecting and previewing all plots for any cluster, deleting a cluster, and previewing (color-coding) all clusters on the t-SNE scatter plot. A number of input boxes and buttons also allow the merging and splitting of clusters, as well as the reassignment of spikes to different clusters.

The development of the GUI is based on the PyQtGraph Python library. This makes further development of good-quality views as desired by individual users relatively easy and fast. At the same time, the PyQtGraph library will struggle with a large number of points presented simultaneously on the t-SNE view. For the gaming desktop described, the easy-to-work-with limit was reached at about 1 million spikes.

The Python code design was informed by the need to keep the code simple and extendable. To achieve this, we chose to implement only functions without any obfuscating object orientation or passing data around data structures in more complicated ways other than function arguments and return statements. Individual functions are constructed to be self-standing and usable outside the context of the spike-sorting work flow the code was designed for. For example, the basic t-SNE functions can operate on any other data of a samples \times features dimensionality. The functions that produce the average spike time courses or the average heat maps of the probes can easily be used to generate plots outside the confines of the GUI.

2.4 t-SNE Parameters and Accuracy Measurements. For spike detection, we used a high- and low-detection threshold of 6.5 and 2 SD,

respectively. We tried out a large range of t-SNE parameters (perplexity, learning rate, theta, and number of iterations) in order to define the set that gave us good results but did not take too long to run. For all of the results presented here, the parameters used were perplexity 100, learning rate 200, and theta 0.2. The theta parameter defines the angle of the cone inside which all points are treated as a single average point by the BH algorithm. Smaller values mean that the algorithm averages fewer points together (i.e., only those that are far away from the central point). A value of 0.2 is considered an approximation closer to an exact solution. For the PD128 and the HD1 sets, we ran the algorithm for 2000 iterations, and for the PD32 and the HD2, 5000. Perplexities lower than 100 were shown to compromise the results (insofar as separation of clusters defined by visual inspection), while higher numbers (we tried up to 1000) would make no obvious difference other than adding to the run time of the algorithm. We have found that perplexity is a sample number-dependent measure, but for tens to hundreds of thousands of samples (as in all our data sets), the chosen number of 100 offered the best quality versus run time balance. The fact that over a certain value, perplexity did not seem to change the quality of the embedding, adds to the idea in the t-SNE literature that this is a stable parameter that can vary a lot without substantially influencing the results.

Having labeled data allowed us to measure the quality of the t-SNE clustering visualization as a tool for separating single units. Since the t-SNE algorithm itself does not cluster (i.e., label) the data but only offers a 2D embedding, we needed a way to label the spikes according to their position in that embedding. We chose to use the density-based spatial clustering of applications with noise (DBSCAN; Ester, Kriegel, Sander, & Xu, 1996) algorithm. This provides a nonparametric way to label the embedded spikes by clustering together samples that form denser groupings compared to their immediate environment. We found that DBSCAN's approach to clustering matched most closely the human intuition of neural units corresponding to separate groups of spikes in the 2D visualization of the t-SNE data.

Having established a method for labeling the t-SNE results, we then compared the generated labels with the ground-truth information from the juxtacellular recordings or the hybrid spike groups. We report here the results of three commonly used measures for such comparisons. The first is precision (or confidence or true positive accuracy)—the ratio of the true positive samples over all positively labeled samples. True positive are the spikes labeled by DBSCAN as part of a unit that also had either a juxtacellular spike correspondence or the correct hybrid label. Positive are all spikes defined by DBSCAN to belong to the specific single unit. The second is recall (or sensitivity or true positive rate)—the ratio of the true positive samples over all true samples. True are all spikes with a juxtacellular spike correspondence or a specific hybrid spike label. The third is the F-factor, which is defined as the harmonic mean of precision and recall (i.e., $2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall})$). We also calculated the receiver operating

characteristics (ROC) values for each label (either hybrid spike set or juxtacellular corresponding set) as a point on the plot of the true-positive rate versus the false-positive rate (see supplementary Figure 3). The false-positive rate is defined as the ratio of the false-positive samples over all the negative samples. False positive are the spikes defined by the DBSCAN as part of the label but not having a corresponding juxtacellular spike or a hybrid set label. The negative spikes are all spikes not having the specific juxtacellular or hybrid label.

3 Results

3.1 Paired Data. We ran the t-SNE algorithm (for a 2D embedding) on the full 128,820 spikes of the PD128 set; each spike was represented as a masked vector of 384 dimensions (128 channels * 3 largest PCA components per channel). The masking of the PCA components (i.e., modulated between 0 and their full value) is described in Rossant et al. (2016). The resultant embedding is shown in Figure 2A. The embedding generates a number of distinct groups of spikes and a number of more diffuse clouds with varying internal densities. A spike grouping at the bottom of the figure contains the majority of the extracellular spikes that correspond to juxtacellularly recorded spikes (i.e., ground-truth spikes from the isolated single cell). The figure shows that a small percentage of the spikes coming from the labeled cell are spread throughout the entire 2D space, while the labeled cluster also contains some spikes that are not generated by the labeled cell. For this set of ground-truth spikes, the precision, recall, and F-factor are 0.86, 0.83, and 0.84, respectively, which translates to 14% of the ground-truth spikes not being classified as part of the main cluster and to 17% of the main cluster spikes being misclassified as part of the ground truth. We note here that due to the clear isolation of this group of points from all others on the plot, the DBSCAN labeling of this cluster is identical to any naive user's manual clustering on the same plot. That means that the errors produced are not due to misplacement of fringe spikes due to DBSCAN's parameters (or different users' biases). Both the false-positive and false-negative errors are due to t-SNE positioning these spikes either fully inside the cluster without them belonging there or far away from it although they belonged there. One possibility for the false-negative errors (quantified by the recall measure) is that t-SNE, just like all other spike-sorting algorithms on which it relies, cannot differentiate overlapping spikes and falsely assigns them to separate spike groupings from the ones they should belong to.

The color coding of the samples with a corresponding juxtacellular spike shows that the embedded cluster's internal structure has a strong relationship with physical characteristics of the actual spikes—in this case, their peak amplitude. The t-SNE algorithm is designed to retain the local structure of the multidimensional space during its transformation into the 2D embedding space. That means that any correspondence between the

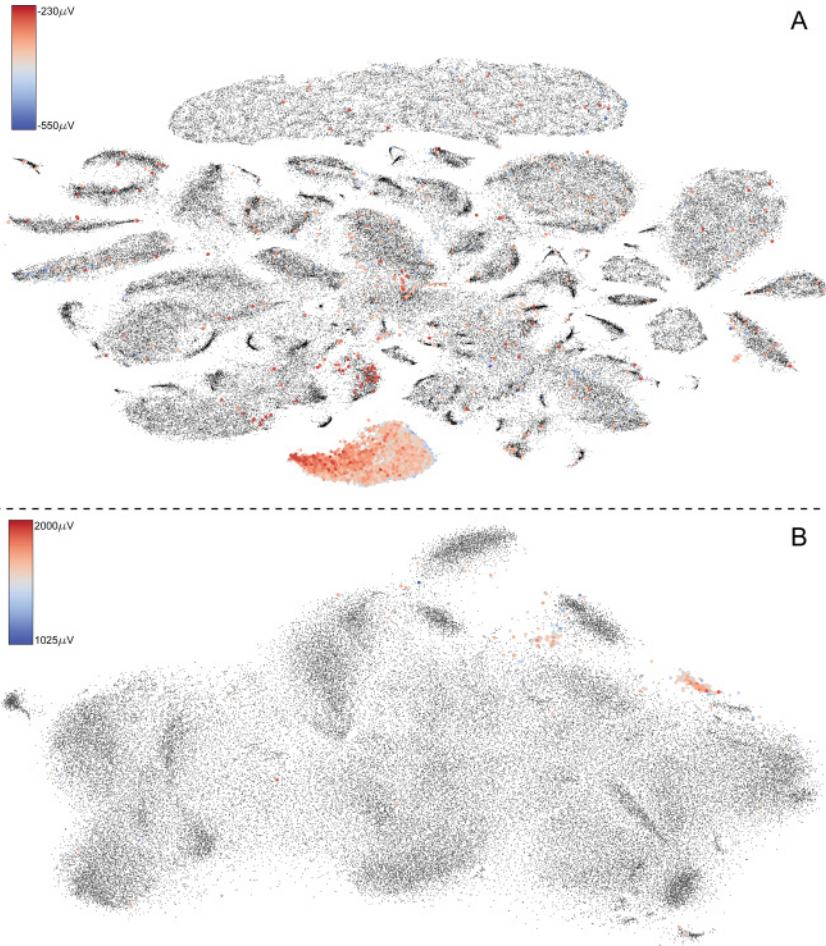


Figure 2: Results of t-SNE analysis on PD128 (A) and PD32 (B). The color coding represents the size of the juxtacellular spikes that corresponds to the extracellular ones on the t-SNE plots.

physical properties of the samples and their distances in the high-dimensional PCA space will be retained in the subsequent embedding, at least for the samples that are close together (i.e., the ones within a single cluster). The 2D embedding space is easy to visualize and thus allows quickly noticing such relationships. By simply changing the color-coding scheme to represent different properties, one can easily browse the relationships (or lack thereof) between any number of specific characteristics. For example, color-coding the spikes according to the time they appear in the

recording reveals that there is no correspondence between the size of spikes and the time they were recorded (results not shown).

Having used t-SNE to generate an embedding of the data set that is informative to the spike-sorting problem, we briefly explored the possibility that other embedding algorithms might offer an equal or even better result. We used two variants of the locally linear embedding (LLE)—standard (Roweis & Saul, 2000) and modified (Zhang & Wang, 2007)) methods—as well as an isomap (Tenenbaum, Silva, & Langford, 2000) and a spectral embedding (Belkin & Niyogi, 2003). We used all algorithms with their default parameters as set in the scikit-learn implementation. All four algorithms failed to run for the whole data set (128,000 spikes) since the scikit-learn implementation is a CPU only one that assumes data sizes in the low thousands range. We subsampled the data set so that all the algorithms could run (maximum allowed samples were 20,000 spikes) and run this with all four algorithms and the t-SNE algorithm. The results of this experiment, demonstrating the crowding problem that most of the embedding algorithms face can be seen in supplementary Figure 4.

The clearest picture of how t-SNE operates on the data can be gained from videos of the t-SNE process in which each frame is the result of progressive iterations of the embedding algorithm. We captured this process for the PD128 data set in supplementary video 1. Figure 2B shows the t-SNE embedding of the 73,814 spikes of the PD32 set. Here, the embedding generates a more homogeneous cloud with significantly fewer easy-to-delineate groupings of spikes. The precision, recall, and F-factor for the juxtacellularly labeled spikes in this case are 0.91, 0.65, and 0.76, respectively, for the t-SNE/DBSCAN-generated cluster with the most color-coded (juxtacellular) spikes. In this case, the cluster is a fairly homogeneous one (with only 9% of the spikes not belonging to the unit it represents), but it fails to capture a large percentage (35%) of spikes from the same unit that end up in other groups. Also, the cluster's internal structure shows no correspondence with the spikes' peak amplitude as measured by the juxtacellular electrode. We will propose (see section 3.3) that the drop in clustering performance between the P128 and P32 embeddings is mainly due to the lower sampling density of the extracellular space provided by the 32-channel probe.

3.2 Hybrid Data. We also used t-SNE to embed the two hybrid data sets described in section 2.1. The results for HD1 are in Figure 3A and for HD2 in Figure 3B. In this case, the output of the t-SNE algorithm provides a clean visualization of the known single-unit clusters. Most clusters are fully separated from the other clusters and contain a very small number of spikes that do not belong to their corresponding unit. The spikes shown in black in both sets are not hybrid spikes, but rather those that were in the original recording (i.e., not ground truth). For HD1, all clusters have precision, recall, and F-factor of 0.99. For the HD2 set, clusters 1, 4, 5, and 7 have precisions, recalls, and F-factors ranging between 0.95 and 0.99. Cluster 6

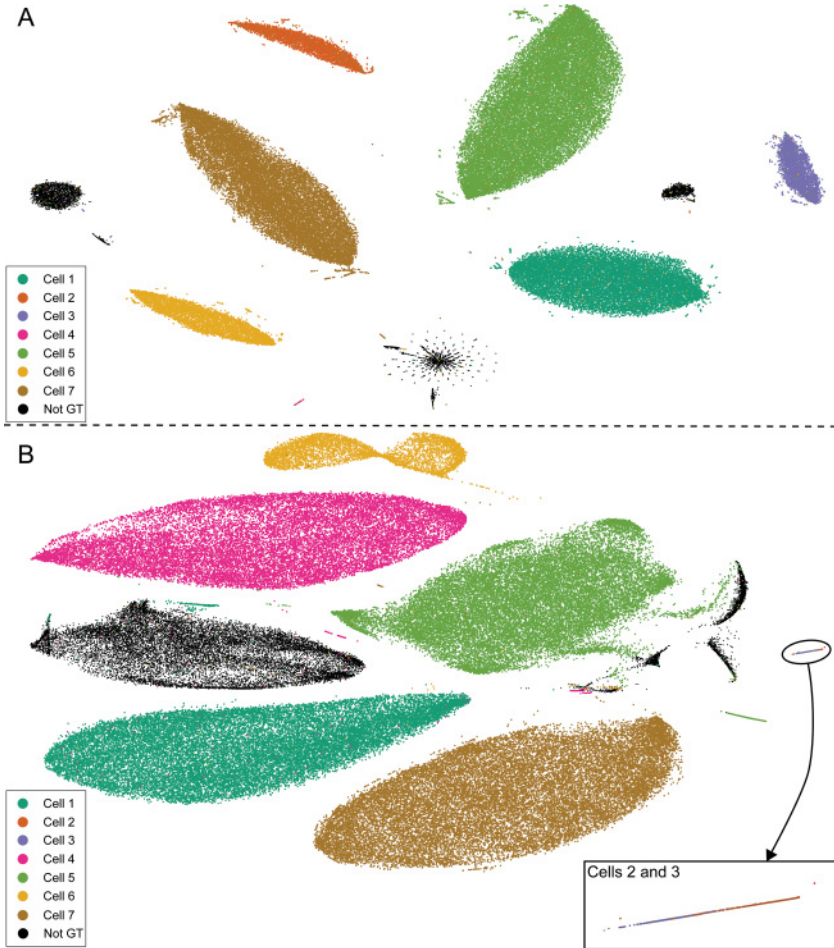


Figure 3: Results of the t-SNE analysis on the hybrid data, HD1 (A) and HD2 (B). The colors represent spikes that belong to preclassified units added to the data sets, while the black points represent spikes preexisting in the data sets and thus do not belong to known units. The inset in panel B shows a zoom-in of the embedding of units 2 and 3 that t-SNE does not manage to segregate.

has 0.91, 0.9, and 0.91, respectively. In the case of cluster 5, DBSCAN did not classify as part of the unit the spikes that expand away from the main group to the right of the cluster. Yet their proportion was small enough to keep the recall at 0.95. The same applied to cluster 1, which was missing the group of spikes showing at the top of the large not-ground-truth spike grouping (black group at the left of the plot labeled “Not GT”). That group

of spikes was also small enough in percentage that the cluster still achieved a recall of 0.96. t-SNE failed to fully separate clusters 2 and 3 (see the inset of Figure 3B).

In this case, the clusters' recalls were 0.99, but their precisions were 0.44 and 0.55 with F-factors of 0.61 and 0.71, respectively. The internal structure of their common spike group in the 2D embedding space shows that the similarity of their spikes changes in a uniform way and that some spikes from the two groups are quite similar, thus causing the embedding to overlap the two units.

As in the case of PD128, the separation of the spike groups on the t-SNE embeddings was so clear that the DBSCAN assignment of labels was robust to the DBSCAN's parameters. Given this separation of groups of spikes, it is also obvious that users trying to manually assign the spikes to different clusters based on the t-SNE plot will come to very similar conclusions with the automatic clustering shown here.

3.3 Probe Electrode Density and Clustering Quality. As the number of electrodes per probe increases, it is important to understand how the density of electrodes relates to an algorithm's ability to reliably extract and isolate single units. A major argument for the miniaturization of silicon probes was the idea that more densely spaced electrodes and probes with larger numbers of electrodes will increase spike-sorting quality by providing more features with which to classify spikes. Here we review this argument using t-SNE by performing embeddings of the P128 data set, originally captured with a density of 2050 electrodes/mm², at artificially reduced densities. This was achieved by removing more and more channels while keeping the total coverage intact and then performing a new t-SNE embedding with each sub-data-set. The results visually demonstrate how artificially decreasing the probe's density affects the sorting of the detected spikes.

We started from the PD128 set and used the manual sorting GUI we developed (see section 2) to define as many groups of spikes belonging to the same unit. The criteria used to delineate a number of spikes as a single unit were the following: the spikes had to be close in the t-SNE space and ideally belonging to one, obvious, unique grouping (e.g., large orange cluster at the right of the plot or yellow cluster at the bottom). If that was not the case, the grouping had to at least be contiguous (e.g., the dark red and green clusters at the far right and middle of the plot or the yellow and red clusters at the left and top of the plot). The average time course of the spike on all electrodes had to show a standard extracellularly measured action potential. The autocorrelogram of the spike times within a group had to be zero within 1 ms of time zero (i.e., no two spikes of the group could have been fired in a time interval shorter than 2 ms of one another). Also it had to show large numbers of hits outside the refractory period to show that the refractory period was not a result of just a small number of spikes. Finally, the

heat map of the group had to show a contiguous region of activation on the probe that was compact (just a few neighboring electrodes).

Following these criteria, we labeled 90 units in the PD128. The autocorrelograms of a few of those units (one being the unit of the juxtacellular spikes) are shown in supplementary Figure 5. Given the extent and density of electrodes, as well as the position of the probe within cortex, this is consistent with the number of neurons one might expect to detect in local vicinity (fewer than 50 microns; Neto et al., 2016). Figure 4A overlays the manually classified clusters directly on the t-SNE plot. There are three separate groupings of spikes that we have not labeled as a single unit (the grouping at the top of the plot and two on the right side of the plot between the orange, yellow, blue, and green units). t-SNE in this case had clustered together three groupings of multiunit spikes. These groupings also showed no internal grouping that could be assigned to a single unit. Figures 4B to 4F show the t-SNE results arising from the same data set with some electrodes removed. From panels B to F, the number of electrodes used were 64, 32, 22, 16, and 8, respectively (denoted as blue in the figure). The color coding in these subplots is the same as in Figure 4A and denotes the unit each spike belongs to as defined by the above procedure. Even at 64 channels, there is an obvious deterioration of the t-SNE clustering quality that progresses all the way to the eight-channel data set. This deterioration can be seen from the reduction of the number of easy-to-delineate spike groups, the increase of the number of spikes that form part of the larger amorphous clouds, and the mixing of the labeled spikes among the unit clusters and between the clusters and the undifferentiated cloud structures.

In order to enhance the argument toward probes with higher density and a larger number of electrodes, we also embedded using t-SNE the PD4 data set. Supplementary Figure 6 shows the t-SNE embedding of these spikes (using as features the first 10 principal components of each tetrode channel) color-coded with both the Juxta spikes (see Supplementary Figure 6A) and the KlustaKwik's cluster definitions (see Supplementary Figure 6B). Given that the extracellular probe in this case has only 4 channels one expects the number of units contributing to the extracellular signal to be around 8 (Pedreira, Martinez, Ison, & Quian Quiroga, 2012) with a theoretical maximum of around 20 (Buzsáki, 2004). The KlustaKwik algorithm in our case finds only 3 clusters, the same number as in the original work presenting these data (Harris et al., 2000). The smaller number of units found in this case might be due to the damaging effect of the juxtacellular electrode positioned next to the tetrode. We note also that in the only other paired tetrode, juxta study available (Wehr et al., 1999), the tetrode signal was decomposed in only 4 units. Supplementary Figure 6B shows that one of these clusters contains most of the Juxta spikes but also a large number of non-Juxta ones. t-SNE is doing a better job at identifying the Juxta spikes. More specifically, the spike group manually selected in the blue square in supplementary Figure 6A has 1070 spikes out of which 825 are Juxta spikes (out of 842 total

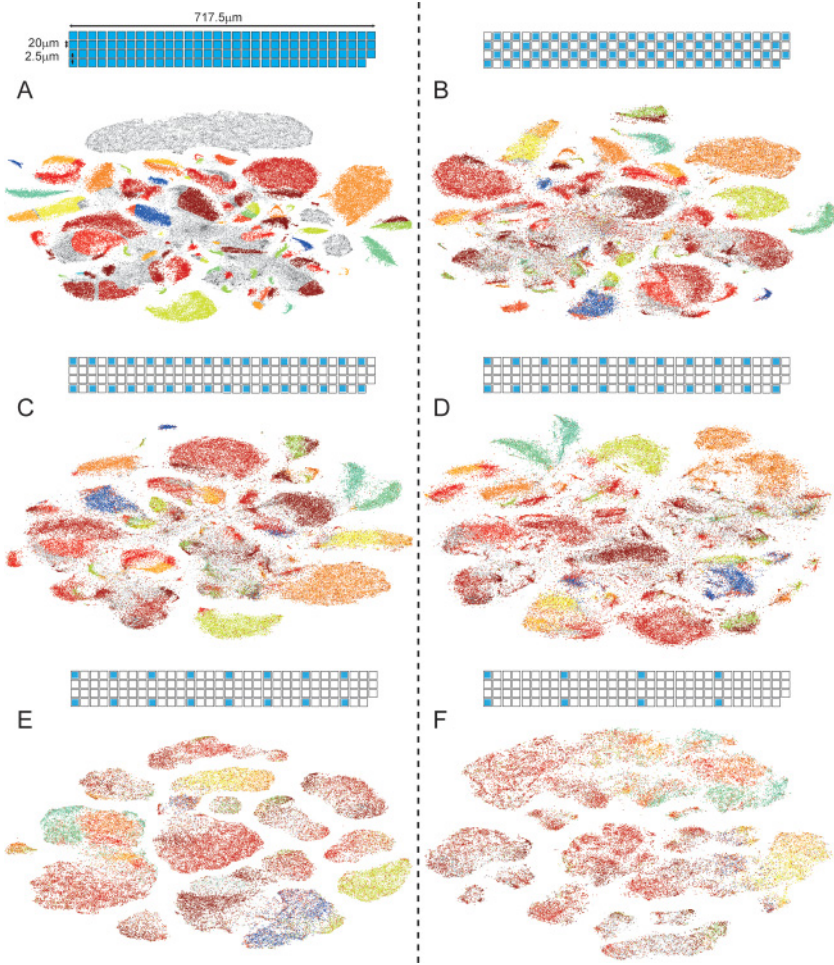


Figure 4: t-SNE embeddings of the same PD128 data set for a series of electrode densities (the electrode configuration is shown at the top of each subplot with blue being the used channels). The color code represents the manual sorting of the spikes done on the full (128 channels) t-SNE embedded data set. The way the different color groups start to break up and merge with each other over the consecutive electrode configurations visually depicts the increased difficulty of the spike-sorting problem for probes with reduced densities. For the last two densities (panels E and F), the electrodes are far enough from each other to ensure that each spike is seen by only one electrode. The quality of these embeddings makes immediately obvious the usefulness of the multiple spatially separated channels with information about each spike.

Juxta), while there are 17 Juxta spikes not found in that group. That gives a precision, a recall, and an F-factor of 0.98. Nonetheless, the autocorrelogram of supplementary Figure 6A (inset) shows that given no prior knowledge of the Juxta spikes, a manual curator would be hard-pressed to consider this grouping as a single unit (compare this to autocorrelogram a of supplementary Figure 5 also belonging to a Juxta spike group).

3.4 Intuitive Manual Curation of Automated Clustering. Template matching algorithms are proving to be fast alternatives to the more classical detect, embed, and classify spike-sorting pipeline, especially for probes with a large number of densely spaced electrodes. These algorithms iteratively generate templates of the spikes present and then use these templates to detect and classify the individual spikes. The result is a list of spike templates together with the Euclidean distances of each spike to each of these templates. Each spike is assigned to the template that matches it best (i.e., it has the smallest distance to). We report here how t-SNE embeds these template distance features and how we used our GUI to sanity-check the automatically generated results and get a broader overview of the clustering problem for a specific data set.

We ran the kilosort algorithm (Pachitariu et al., 2016) on the PD128 data set with a maximum number of templates set to 256 (double the number of channels on our probe). It generated 252 templates and detected 466,313 spikes. We then used the distances of each spike to all the templates as the t-SNE input feature space (a 252-dimensional one), setting to 0 all the distances except the nearest 16 templates. The result of this t-SNE embedding can be seen in Figure 5. The embedding shows three different types of spike groupings. The first are a number of very tight groupings (denoted from here on as “points”) where the spread of the entire group in both the x - and y -axis is at least three orders of magnitude smaller than the total spread of the embedding. The second is a linear grouping where there is an obvious extended axis (not necessary along the x - and y -axis of the full embedding). This axis shows a much larger spread than its perpendicular one, making the grouping appear very elongated (denoted from here on as “lines”). The third type of grouping has large spreads on both the x - and y -axis, no fewer than two orders of magnitude smaller than the total t-SNE spread (denoted here as “blobs”).

Using the manual clustering GUI, we could easily evaluate in which kilosort assigned templates the spikes in the different embedding groupings belonged and vice versa (i.e., which groupings held spikes from any single template). From this comparison, five general categories emerged. The first category involved groupings that were fully represented by a single template and whose heat map and autocorrelogram both indicated a single unit. There were 73 of these single units (SUs), out of which the majority (49) were point groupings, 18 were line groupings, and a small minority (7) were blobs. This category included 109,890 spikes. The second category

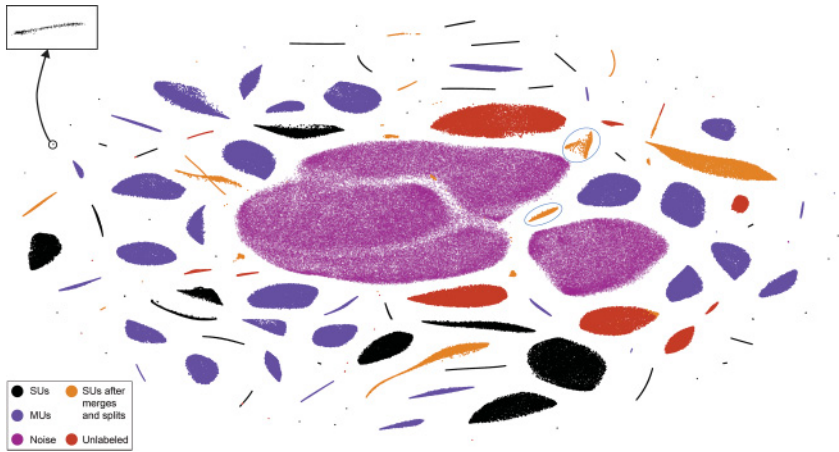


Figure 5: t-SNE embedding of the results of the kilosort algorithm (distances to templates) for the PD128 data set. Colors indicate the different categories that the spikes fall into after using our custom GUI to manually evaluate and correct as necessary kilosort's spike assignments to templates. The SU labeled groupings represent a single unit each. The MU labeled groupings represent a single template each but multiple units. The Noise labeled groupings represent noise that has been picked up by kilosort as spike templates. The SUs after merges and splits have a more complex representation, where each single unit can be either one or more groupings and have originally belonged to one or more templates. The Unlabeled groupings represent spikes that could not be assigned to either a single unit, a multiunit, or a noise group. The inset at the top of the figure depicts a zoom-in ($\times 30$ magnification) to a point grouping (SU) showing that these groupings actually comprise large numbers of spikes all embedded very close to one another. That compactness indicates a very large similarity in the template distance feature space and a large dissimilarity to any other spike, so it is not surprising that they are the groupings that kilosort has assigned to templates that represent single, easy-to-define units.

consisted of groupings that again were fully defined by a single template, but this time the autocorrelogram (and sometimes the heat map) indicated multiple units (MUs). There were 39 of these MUs, the majority of which (26) were blob groupings with only a small minority being either lines (7) or points (6). There were 134,288 spikes in this category of MUs. The third category was represented by the three largest, semiconnected blob groupings in the center of the embedding and the fourth largest blob on their right. Their spikes belonged to five templates (four for the central blob and one for the blob on the side), and each template had its spikes grouped together but with a large minority spread throughout other parts of the groupings. These five templates were the most numerous, and they all showed average

spike trains that were very small in amplitude (less than $100 \mu\text{V}$), did not resemble spike shapes, and were identical on all channels. We denoted this category as noise. It had 139,073 spikes.

In the fourth category we lumped together all the cases that after splitting or merging of templates or moving spikes from one template to another ended up with acceptable SUs (based on their average spike shape, their probe heat map, and their autocorrelogram). We arrived at 24 SUs with a total of 38,482 spikes. In these cases, the combined information of where the spike lay in the embedding and in which template it belonged to made it fairly straightforward to either appropriately merge or split templates or move spikes along templates. In some cases, a single SU would be represented by two separate t-SNE groupings. The two groupings embedded within blue ellipses in Figure 5 are an example. They represent a single unit after the merging of two kilosort templates (with spikes mixed over the two groupings). This unit happens to correspond to the juxtacellularly marked spikes. The unit has 4845 spikes in it. Out of those, the 4821 spikes' timestamps correspond to the 4998 juxtacellular time stamps within a jitter of 1 ms. That translates to a false-positive error of 0.5% and a false-negative error of 3.7%. In other cases, a single grouping would contain spikes assigned to two templates, and a merging indicated a single SU (like the orange line at the bottom center of the embedding), while in others, a single template would be represented by multiple groupings, each defining an SU, resulting in the template's splitting (like the two orange line groupings at the top center of the embedding). Finally, the fifth category involved all the spikes that we were unable to assign to any of the previous divisions (SUs, MUs, or noise). These were spikes that showed no obvious correlation between the embedding position and their template assignment. For example, the large red blob to the top and right of the large noise blob had 14,184 spikes that kilosort had assigned to 95 separate templates (most were templates with fewer than 10 spikes each). There was no internal structure to the blob; the spikes of each template appeared randomly spread throughout. The embedding of all these spikes and templates in a single grouping made it straightforward to visualize the situation and assign all the spikes to the unlabeled division. Supplementary Figure 7 also shows some representative autocorrelograms for the SU, MU, and SU after merges and splits, both depicting a case of a three templates merge and a case of a split of one template to two units.

3.5 Using t-SNE with Very High-Density Probes. The t-SNE embedding as a visualization tool for spike sorting is expected to operate best on data sets arising from probes with high densities and large numbers of electrodes. In order to test this hypothesis and further demonstrate the utility of t-SNE embedding on very large electrophysiology data sets, we applied the algorithm to the NS data (described in section 2.1). If one were to manually curate the results of the kilosort algorithm for this data set using

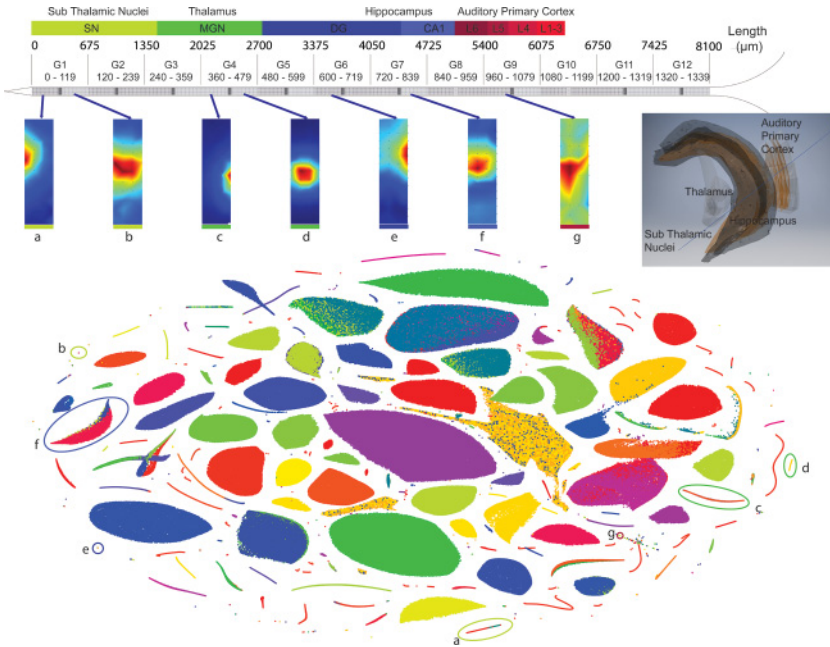


Figure 6: t-SNE embedding of the data from an acute recording of the NeuroSeeker CMOS probe. (Top) A graphical description of the 8 mm long probe with its 12 regions (G1–G12) of active electrodes. The colored bar on top of the probe denotes the brain regions that the different parts of the probe were inserted into (inset on the right shows a 3D representation of the probe insertion). Heat maps a to g under the probe show the peak-to-peak value of the average spike of the spike groups circled on the t-SNE plot with the corresponding lettered ellipse. These are selected from different parts of the probe, as shown by the arrows from the probe schematic to the heat maps. The number of active electrodes for each spike is usually between 10 and 50, with an average of about 40. The t-SNE plot is in this case color-coded with the kilosort templates. Spike groups a (a subthalamic unit) and f (a hippocampus unit) are two examples of merging on the t-SNE plot 2 and 4 kilosort templates, respectively, (autocorrelograms not shown). SN: substantia nigra; MGN: medial geniculate nucleus; DG: dentate gyrus; CA1: cornus ammonis 1; L1–L6: layers 1–6 of the primary auditory cortex.

existing tools, this would require looking through data residing in 2724 (908 electrodes * 3 pcs) dimensions. Figure 6 shows the result of the 2D t-SNE embedding of these data using as features the 579 distances of each spike to the kilosort templates.

The embedding shows a clear separation of spikes into groups with different geometries, very much like the embedding arising from the kilosort

templates of the PD128 data set. The more than doubling of the number of features (579 versus 252) and spikes (1,091,229 versus 466,313) and the eight times increase in number of electrodes (908 versus 128) compared to the PD128 kilosort data set have not diminished the algorithm's ability to generate intuitive groupings of spikes. Using that embedding, it becomes very easy to spot groups of spikes and templates that require further manual curation (like the a and f groups in Figure 6 that proved to require merging of their respective templates) or that needed to be removed from the data set (like the irregularly shaped group in the middle of the embedding comprising of a number of templates that proved to be noise not detected during the initial cleaning of the kilosort results).

4 Discussion

Spike sorting has experienced a dramatic evolution over the past 20 years. It started out as a set of techniques to discriminate 5 to 10 distinct units in a space of a few tens of features. Today, it includes strategies for labeling many tens to hundreds of units in large feature spaces. The number of distinct units is expected to soon reach well into the thousands. The emerging demands of these growing data sets have inspired the development of more capable automatic sorting algorithms. However, the manual overview of the spike-sorting process remains an essential step of the pipeline, albeit one that is becoming increasingly labor intensive and error prone. The problem of producing human-readable visualizations of structures that exist in large-dimensional spaces is, of course, not unique to spike sorting but is common in all data-intensive fields. One commonly employed strategy for working with such data is the use of nonlinear dimensionality-reduction methods that try to retain in their projections as much of the initial structure of the data as possible. The current state of the art in these methods is t-SNE, which manages to project onto two or three dimensions the multidimensional data in a way that preserves its local structure and makes the visualization of this structure both possible and intuitive. A large literature has evolved applying this technique to a diverse number of big data problems ranging from AI (Mnih et al., 2015) to genetics (Platzer, 2013; Xu, Jiang, Hu, & Li, 2014) and behavior (Berman, Choi, Bialek, & Shaevitz, 2014). We show here that in the case of spike sorting, using t-SNE on the results of either a masked PCA or template matching provides a visualization that offers a clear picture of how the individual spikes form groups in the high-dimensional feature space. These groups directly relate to the individual units that generate the spikes, which can now be visualized and curated in an intuitive manner or compared to the results of an automatic clustering method. Furthermore, t-SNE allows a visual inspection of the structural groups that different data sets of spikes contain, something that can be achieved neither by looking at 2D plots of pairs of features within a multidimensional space nor by any current clustering algorithm. Here

we demonstrate that t-SNE visualization immediately reveals differences between biological and hybrid data sets (see Figure 3), how spike-sorting challenges accrue when each spike is represented by features generated by fewer and fewer electrodes (see Figure 4), and how manual curation of the output from an automated template matching procedure becomes effortless in a low-dimensional space (see Figure 5). We also show that using the combined information of a spike-sorting algorithm and the position of the spikes on the t-SNE embedding is an intuitive and useful approach to manual curation of spike sorting, especially for data sets that arise from hundreds to thousands of electrodes, have hundreds to thousands of putative units, and have spike counts into the millions (see Figure 6), exactly at the regime that current methods of manual curation and data overview are failing.

The generality of the t-SNE method also makes it applicable to data sets that arise from diverse sources. The successful application of t-SNE to behavioral data (Berman et al., 2014) suggests the possibility of applying it to data sets with a combination of spike and other electrophysiological signals (e.g., local field potentials), as well as the behavioral features occurring during the recorded neural activity. It is thus possible that groupings in the t-SNE visualization of such a data set could provide informative clues as to the connection between different forms of brain activity and to behavior itself.

We are now working on using t-SNE embeddings into three-dimensional space to generate spatial (X,Y,Z) representations that can be explored in virtual reality, thus providing a new form of access to complex data sets (see supplementary video 2).

5 Additional Information

5.1 Software Access. A first version of the software used here can be downloaded through the following means. In a conda environment, do: `conda install -c georgedimitriadis t_sne_bhcuda`. That will install both the Python code and the CUDA executable and will work in either Windows or Linux OSes. You can grab the Python code from the PyPi repository: https://pypi.python.org/pypi/t_sne_bhcuda/0.2.1. You can get the C/CUDA code for the CUDA executable from the Github repository: https://github.com/georgedimitriadis/t_sne_bhcuda.

We are currently in the process of rewriting the code so that the CUDA part is embedded in the Python code. The release of this version 2 of the code will also be followed by the release of the manual curation GUI. For this and future developments, check the pypi and Github repositories or contact the authors.

5.2 Competing Interests. We declare no competing financial interests.

Acknowledgments

J. N. was supported by fellowship SFRH/BD/76004/2011 from Fundação para a Ciência e Tecnologia, Portugal. This work was supported by funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement 600925 and by the the Bial Foundation (grant 190/12). We thank the NeuroSeeker project. We also thank the designer of the spyking circus package, Pierre Yger, for his invaluable help in some of the analysis presented in this letter.

References

- Alivisatos, A. P., Andrews, A. M., Boyden, E. S., Chun, M., Church, G. M., Deisseroth, K., . . . Zhuang, X. (2013). Nanotools for neuroscience and brain activity Mapping. *ACS Nano*, *7*, 1850–1866.
- Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, *15*, 1373–1396.
- Berman, G. J., Choi, D. M., Bialek, W., & Shaevitz, J. W. (2014). Mapping the stereotyped behaviour of freely moving fruit flies. *J. R. Soc. Interface*, *11*, 20140672.
- Bestel, R., Daus, A. W., & Thielemann, C. (2012). A novel automated spike sorting algorithm with adaptable feature extraction. *J. Neurosci. Methods*, *211*, 168–178.
- Buzsáki, G. (2004). Large-scale recording of neuronal ensembles. *Nat. Neurosci.*, *7*, 446–451.
- Chang, D., Jones, N. A., Li, D., & Ouyang, M. (2008). Compute pairwise Euclidean distances of data points with GPUs. In *Proceedings of the IASTED International Symposium Computational Biology and Bioinformatics*. Calgary, AB: ACTA Press.
- Chaudhuri, R., & Fiete, I. (2016). Computational principles of memory. *Nat. Neurosci.*, *19*, 394–403.
- Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (pp. 226–231). Menlo Park, CA: AAAI Press.
- Fee, M. S., Mitra, P. P., & Kleinfeld, D. (1996). Automatic sorting of multiple unit neuronal signals in the presence of anisotropic and non-gaussian variability. *J. Neurosci. Methods*, *69*, 175–188.
- Gerstein, G. L., & Clark, W. A. (1964). Simultaneous studies of firing patterns in several neurons. *Science*, *143*, 1325–1327.
- Gray, C. M., Maldonado, P. E., Wilson, M., & McNaughton, B. (1995). Tetrodes markedly improve the reliability and yield of multiple single-unit isolation from multi-unit recordings in cat striate cortex. *J. Neurosci. Methods*, *63*, 43–54.
- Harris, K. D., Henze, D. A., Csicsvari, J., Hirase, H., & Buzsáki, G. (2000). Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. *J. Neurophysiol.*, *84*, 401–414.
- Hulata, E., Segev, R., & Ben-Jacob, E. (2002). A method for spike sorting and detection based on wavelet packets and Shannon's mutual information. *J. Neurosci. Methods*, *117*, 1–12.

- Jun, J. J., Steinmetz, N. A., Siegle, J. H., Denman, D. J., Bauza, M., Barbarits, B., . . . Harris, T. D. (2017). Fully integrated silicon probes for high-density recording of neural activity. *Nature*, *551*, 232.
- Lewicki, M. S. (1998). A review of methods for spike sorting: The detection and classification of neural action potentials. *Netw. Comput. Neural Syst.*, *9*, R53–R78.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., . . . Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*, 529–533.
- Neto, J. P., Lopes, G., Frazão, J., Nogueira, J., Lacerda, P., Baião, P., . . . Kampff, R. A. (2016). Validating silicon polytrodes with paired juxtacellular recordings: Method and dataset. *J. Neurophysiol.*, *116*, 892–903.
- Pachitariu, M., Steinmetz, N., Kadir, S., Carandini, M., & Harris, K. D. (2016). *Kilosort: Realtime spike-sorting for extracellular electrophysiology with hundreds of channels*. bioRxiv doi:<https://doi.org/10.1101/061481>
- Pedreira, C., Martinez, J., Ison, M. J., & Quian Quiroga, R. (2012). How many neurons can we see with current spike sorting algorithms? *J. Neurosci. Methods*, *211*, 58–65.
- Platzer, A. (2013). Visualization of SNPs with t-SNE. *PLoS One*, *8*, e56883.
- Quiroga, R. Q., Nadasdy, Z., & Ben-Shaul, Y. (2004). Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. *Neural Comput.*, *16*, 1661–1687.
- Raducanu, B. C., Yazicioglu, R. F., Lopez, C. M., Ballini, M., Putzeys, J., Wang, S., . . . Srinjoy, M. (2017). Time multiplexed active neural probe with 1356 parallel recording sites. *Sensors*, *17*, 2388.
- Recce, M. L., & O’Keefe, J. (1989). The tetrode: An improved technique for multi-unit extracellular recording. *Society for Neuroscience Abs*, p. 1250.
- Redish, A. D., Battaglia, F. P., Chawla, M. K., Ekstrom, A. D., Gerrard, J. L., Lipa, P., . . . Barnes, C. A. (2001). Independence of firing correlates of anatomically proximate hippocampal pyramidal cells. *J. Soc. Neurosci.*, *21*, RC134.
- Rey, H. G., Ison, M. J., Pedreira, C., Valentin, A., Alarcon, G., Selway, R., . . . Quian Quiroga, R. (2015). Single-cell recordings in the human medial temporal lobe. *J. Anat.*, *227*, 394–408.
- Rey, H. G., Pedreira, C., & Quian Quiroga, R. (2015). Past, present and future of spike sorting techniques. *Brain Res. Bull.*, *119*, 106–117.
- Rossant, C., Kadir, S. N., Goodman, D. F. M., Schulman, J., Hunter, M. L. D., Saleem, A. B., . . . Harris, K. D. (2016). Spike sorting for large, dense electrode arrays. *Nat. Neurosci.*, *19*, 634–641.
- Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*, 2323–2326.
- Ruther, P., & Paul, O. (2015). New approaches for CMOS-based devices for large-scale neural recording. *Curr. Opin. Neurobiol.*, *32*, 31–37.
- Takekawa, T., Isomura, Y., & Fukai, T. (2010). Accurate spike sorting for multi-unit recordings. *Eur. J. Neurosci.*, *31*, 263–272.
- Tenenbaum, J. B., Silva, V. de, & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, *290*, 2319–2323.
- Van der Maaten, L. (2014). Accelerating t-SNE using tree-based algorithms. *J. Mach. Learn. Res.*, *15*, 3221–3245.

- Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *J. Mach. Learn. Res.*, 9, 85.
- Wang, G.-L., Zhou, Y., Chen, A.-H., Zhang, P.-M., & Liang, P.-J. (2006). A robust method for spike sorting with automatic overlap decomposition. *IEEE Trans. Biomed. Eng.*, 53, 1195–1198.
- Wehr, M., Pezaris, J. S., & Sahani, M. (1999). Simultaneous paired intracellular and tetrode recordings for evaluating the performance of spike sorting algorithms. *Neurocomputing*, 26–27, 1061–1068.
- Wood, F., Black, M. J., Vargas-Irwin, C., Fellows, M., & Donoghue, J. P. (2004). On the variability of manual spike sorting. *IEEE Trans. on Biomed. Eng.*, 51, 912–918.
- Wood, F., Fellows, M., Donoghue, J. R., & Black, M. J. (2004). Automatic spike sorting for neural decoding. In *Proceedings of the 26th Annual International Conference of the IEEE on Engineering in Medicine and Biology Society* (pp. 4009–4012). Piscataway, NJ: IEEE.
- Xu, W., Jiang, X., Hu, X., & Li, G. (2014). Visualization of genetic disease-phenotype similarities by multiple maps t-SNE with Laplacian regularization. *BMC Med. Genomics*, 7, S1.
- Yger, P., Spampinato, G. L. B., Esposito, E., Lefebvre, B., Deny, S., Gardella, C., . . . Marre, O. (2018). A spike sorting toolbox for up to thousands of electrodes validated with ground truth recordings in vitro and in vivo, *ELife*, 7, e34518. <https://doi.org/10.7554/eLife.34518>.
- Zhang, Z., & Wang, J. (2007). MLL: Modified Locally Linear Embedding Using Multiple Weights. In P. B. Schölkopf, J. C. Platt, & T. Hoffman (Eds.), *Advances in neural information processing systems*, 19 (pp. 1593–1600). Cambridge, MA: MIT Press.
- Zhang, P.-M., Wu, J.-Y., Zhou, Y., Liang, P.-J., & Yuan, J.-Q. (2004). Spike sorting based on automatic template reconstruction with a partial solution to the overlapping problem. *J. Neurosci. Methods*, 135, 55–65.

Received May 20, 2017; accepted February 20, 2018.