

# CellExplorer: A framework for visualizing and characterizing single neurons

## Highlights

- An open-source framework for single-cell characterization and visualization
- A processing module that calculates a set of standardized physiological metrics
- A graphical interface to explore computed features at the speed of a mouse click

## Authors

Peter C. Petersen, Joshua H. Siegle, Nicholas A. Steinmetz, Sara Mahallati, György Buzsáki

## Correspondence

petersen.peter@gmail.com (P.C.P.),  
gyorgy.buzsaki@nyulangone.org (G.B.)

## In brief

Petersen et al. present a framework allowing users to process, curate, and relate their data to a public collection of neurons. It aims to link genetically identified cell types and physiological properties of neurons collected across laboratories with the potential to lead to an interlaboratory standard of single-cell metrics.

NeuroResource

# CellExplorer: A framework for visualizing and characterizing single neurons

Peter C. Petersen,<sup>1,\*</sup> Joshua H. Siegle,<sup>4</sup> Nicholas A. Steinmetz,<sup>5</sup> Sara Mahallati,<sup>6</sup> and György Buzsáki<sup>1,2,3,7,\*</sup>

<sup>1</sup>Neuroscience Institute, Langone Medical Center, New York University, New York, NY 10016, USA

<sup>2</sup>Department of Neurology, Langone Medical Center, New York University, New York, NY 10016, USA

<sup>3</sup>Center for Neural Science, New York University, New York, NY 10003, USA

<sup>4</sup>MindScope Program, Allen Institute, 615 Westlake Avenue North, Seattle, WA 98109, USA

<sup>5</sup>Department of Biological Structure, University of Washington, Seattle, WA 98195, USA

<sup>6</sup>Institute of Biomedical Engineering, Krembil Research Institute, University of Toronto, Toronto, ON M5T 1M8, Canada

<sup>7</sup>Lead contact

\*Correspondence: [petersen.peter@gmail.com](mailto:petersen.peter@gmail.com) (P.C.P.), [gyorgy.buzsaki@nyulangone.org](mailto:gyorgy.buzsaki@nyulangone.org) (G.B.)

<https://doi.org/10.1016/j.neuron.2021.09.002>

## SUMMARY

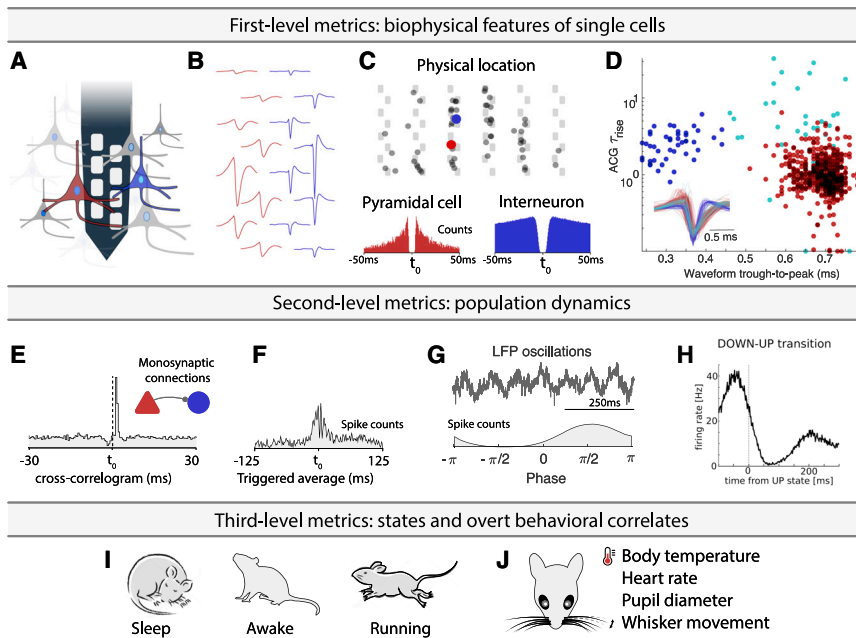
The large diversity of neuron types provides the means by which cortical circuits perform complex operations. Neuron can be described by biophysical and molecular characteristics, afferent inputs, and neuron targets. To quantify, visualize, and standardize those features, we developed the open-source, MATLAB-based framework CellExplorer. It consists of three components: a processing module, a flexible data structure, and a powerful graphical interface. The processing module calculates standardized physiological metrics, performs neuron-type classification, finds putative monosynaptic connections, and saves them to a standardized, yet flexible, machine-readable format. The graphical interface makes it possible to explore the computed features at the speed of a mouse click. The framework allows users to process, curate, and relate their data to a growing public collection of neurons. CellExplorer can link genetically identified cell types to physiological properties of neurons collected across laboratories and potentially lead to interlaboratory standards of single-cell metrics.

## INTRODUCTION

Discovering novel mechanisms in brain circuits requires high-resolution monitoring of the constituent neurons and an understanding of the nature of their interactions. Large-scale, extracellular electrophysiology aims to establish the relationship between neuronal firing and behavioral or cognitive variables to provide insights about the computational role of neurons and neuronal assemblies (Barlow, 1972; Buzsáki, 2004; Steinmetz et al., 2019). Exploiting the power of correlations between neuronal firing and behavioral variables requires multi-level characterization of single neurons and their interactions. Simultaneous recordings from many neurons, preferably identified by optogenetic and other methods, make it possible to build an extensive list of neuron features and their assigned “cell type” properties (Figure 1). Identification and manipulation of different neuron types in the behaving animal is a prerequisite for deciphering the neuron role in circuit dynamics and behavior. However, currently, a gap exists between neuron classification schemes based on molecular and physiological methods (Gouwens et al., 2019; Jia et al., 2019; Kepecs and Fishell, 2014; Klausberger and Somogyi, 2008; McBain and Fisahn, 2001; Roux and Buzsáki, 2015; Rudy et al., 2011), largely because of vast differences in processing data across laboratories and

even within the same laboratory. Ideally, acquired data must be findable, accessible, interoperable, and reusable (FAIR; Wilkinson et al., 2016). That requires an agreed-upon platform for data and metadata curation that allows the sharing of datasets across laboratories for cross-examination and the building of “big data” from experiments collected in multiple laboratories. Using an easily interpretable and standardized data format will facilitate neuroscientists in effectively and transparently communicating their experiments (Sejnowski et al., 2014; Teeters et al., 2015; Bouchard et al., 2016; Martone et al., 2020).

Properties of neurons can be described at multiple levels of complexity. The first level is a description of their biophysical characteristics. This level includes waveform features (Figure 1B), their position relative to the recording sites, and other units and metrics related to firing patterns: interspike interval statistics, ACGs, and derived metrics (Figure 1C). These first-level features can be used for a first-order separation of single neurons into putative major classes, typically excitatory and inhibitory cells (Figure 1D). Single-neuron properties can be related to genetically identified neuron classes with optogenetics and other more-direct methods, such as juxtacellular and intracellular recordings (Boyden et al., 2005; Klausberger and Somogyi, 2008; Rudy et al., 2011; Buzsáki et al., 2015; Roux and Buzsáki, 2015; Lima et al., 2009). Antidromic and unit local field potential



**Figure 1. Experimental, paradigm-independent characterization of single neurons**

(A) Using high-density silicon probes or multiple tetrodes (shown is a single shank with eight recording sites), dozens to hundreds of neurons can be recorded simultaneously.

(B) Spikes are extracted from the recorded traces and assigned to individual neurons through spike-sorting algorithms (the average waveforms across multiple channels from two putative cells).

(C) The relative positions of each neuron are determined through trilateration. Top: neurons projected on a silicon probe with six shanks and a staggered poly-2 electrode layout. Bottom panels: autocorrelograms (ACGs) are used to characterize the neurons (a bursting pyramidal cell with a wide waveform in red; a fast-spiking interneuron with a narrow waveform in blue).

(D) Neuron-type classification based on first-order biophysical parameters, such as spike-waveform width (trough-to-peak) and the temporal scale of the rising phase of the ACGs ( $\tau_{rise}$ ). Optogenetics and other direct identification methods can further ground units to neuron types.

(E) Interactions between neurons are characterized by their cross-correlograms and mono-synaptic connections (determined via spike transmission probabilities).

(F) Event-related histogram.

(G) Relating spikes to LFP patterns.

(H) Relating spikes to brain-state changes.

(I and J) Spike pattern correlations with brain states and overt behaviors. Only a few possible examples are shown.

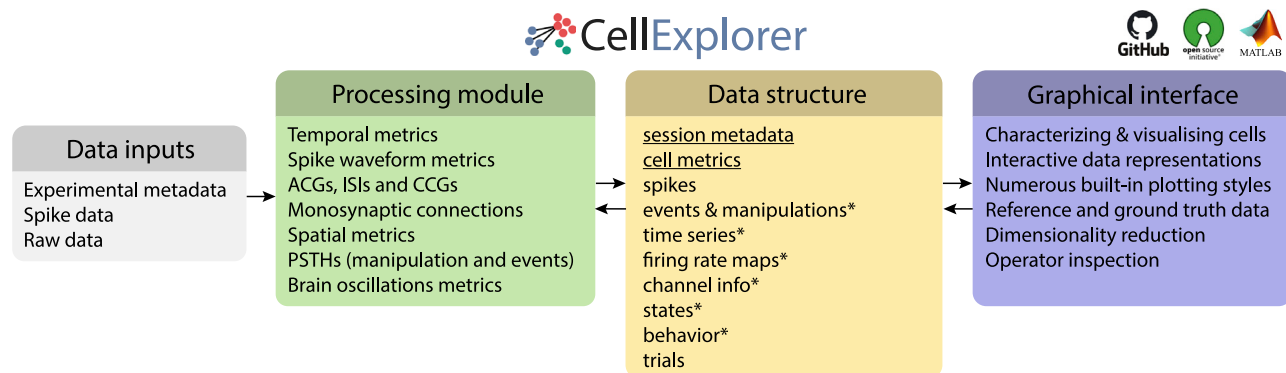
See also Table S1.

(LFP)-coupling techniques provide further assignment of single neurons to cortical regions, layers, and target projections (Bishop et al., 1962; Zhang et al., 2013; Cioocchi et al., 2015; Senzai et al., 2019; Shamash et al., 2018). The second level relates properties of single neurons to other neurons. Examples include cross-correlations and putative monosynaptic connections between excitatory and inhibitory neurons derived from spike transmission probabilities (Figure 1E), pairwise and population synchrony, and relationship to multiple oscillatory and irregular LFPs (e.g., rhythmic patterns and up-down transitions in the cortex). These metrics can be expanded for specific brain regions and questions. The third-level metrics of single-unit activity include the relationship between neuronal firing patterns and brain states (e.g., non-rapid-eye-movement [non-REM], REM, and awake states; Figure 1H) and overt behavioral correlates (Figure 1I). These include, but are not restricted to, arousal states (sleep states and waking), spontaneous motor patterns, movement pattern changes, locomotion speed, head turns, whisker movements, respiration, heart rate, body temperature, pupil diameter, and other autonomic parameters (McGinley et al., 2015; Stringer et al., 2019).

These three levels provide generic features of neuronal activity common to all experimental paradigms in the same species and, therefore, are communicable across different experiments and laboratories, leading to joint databases and standardized metrics across different laboratories. Next, these three-level, paradigm-independent features can be contrasted and compared with experiment-unique manipulations and higher-level corre-

lates of spiking activity. Because these latter correlations are often paradigm specific and differ across laboratories, the three-level analysis can safeguard against mistakenly assigning cognitive and other roles of neuronal spiking when spike pattern changes can be explained by measurable, overt behavioral correlates. However, even if all of the above information is available separately, factoring out critical variables and their combinations is possible only when the multitudes of single-neuron characteristics can be compared flexibly.

The above workflow is similar across many physiological experiments. In the analysis of various features, an often-asked question is how one particular feature of unit firing relates to the many other metrics calculated by the experimenter. This is typically done by identifying some unexpected firing patterns in a set of neurons and independently analyzing features one by one to find common features or to exclude potential artifactual explanations. Whether testing a specific hypothesis or mining the ever-growing number of publicly available datasets, this process can be advanced by user-friendly processing pipelines, standardization of data formats, and highly flexible visualization methods. To provide the needed flexibility and to facilitate new ways of data mining neurophysiological data, we developed the open-source framework CellExplorer to characterize and classify single-neuron features from multi-site extracellular recordings. CellExplorer consists of a pipeline for extracting and calculating physiological features, a flexible data format, and a powerful graphical interface that allows for fast, manual curation and feature exploration. We demonstrate its utility through



**Figure 2. Three-component framework**

A single, extensive processing module (green); standardized, yet flexible, data structure (yellow); and a graphical interface (purple). Data inputs are compatible with most existing spike-sorting algorithms (gray). The data structure joins the processing module with the graphical interface (\* signifies data containers). CellExplorer is open source, built in MATLAB, and available on GitHub. See also [Figures S1, S2, and S3](#) and [Table S2](#).

multiple examples and explain its user-friendly operation through detailed tutorials and video illustrations.

## RESULTS

The CellExplorer architecture and operation consist of three main parts: a processing module for feature extraction, a graphical interface for manual curation and exploration, and a standardized, yet flexible, data structure ([Figure 2](#)). A step-by-step tutorial is available in the [Supplemental information](#), and more tutorials are available online ([Video S1](#)). Flow charts are available in [Figure S1](#). The first step in running the pipeline is defining the data input.

### Data input

When running the pipeline, relevant metadata describing the spike format, raw data, and experimental metadata must be defined ([Figure 2](#)). All experimental metadata (session-level) are handled in a single MATLAB structure, with an optional graphical interface for inspection and manual entry ([Figure S2](#)). The platform supports several spike-sorting data formats, including Neurosuite, Phy, KiloSort, SpyKING Circus, Wave\_Clus, MClust, AllenSDK, Neurodata Without Borders (NWB), Alyx Files (ALF), MountainSort, and IronClust ([Chung et al., 2017](#); [Hazan et al., 2006](#); [Pachitariu et al., 2016](#); [Quiroga et al., 2004](#); [Schmitzer-Torbert et al., 2005](#); [Yger et al., 2018](#)). The wide-band, recorded, “raw” data are critical for comparing derived metrics across laboratories because preprocessing pipelines vary widely and depend on equipment type and filter settings (described here: <https://cellexplorer.org/datastructure/standard-cell-metrics/#waveform-based-metrics>). The hardware used should always be specified because it can affect the waveforms of the processed units (e.g., filter characteristics and bandwidth) and compromise the separability of units on waveform characteristics.

### Processing module

From the input data, the processing module will generate cell metrics corresponding to the three-level description of neuronal

firing ([Figure 1](#)) and their relationship to experiment-specific behaviors ([Table S1](#) contains a representative list of metrics for illustration; the full list is available at <https://cellexplorer.org>). The processing module comprises a single MATLAB script, `ProcessCellMetrics.m`, which computes metrics using a modular structure. The first-level description provides waveform features (filtered and wideband) and temporal features: interspike interval statistics (ISIs) and autocorrelograms (ACGs). Next, the unit parameters are used for the initial classification of single neurons into broad default classes: putative pyramidal cells, narrow waveform interneurons, and wide waveform interneurons. In experiments with silicon probes, the physical position, relative to recording sites, is also determined using spike-amplitude trilateration ([Petersen and Berg, 2016](#); [Csicsvari et al., 2003](#)). The user can generate a probe layout and save that to the data path. The processing module (`ProcessCellMetrics`) will then detect and import the layout (see the channel map tutorial: <https://cellexplorer.org/tutorials/channel-maps-tutorial/#channel-maps-tutorial>). The main MATLAB functions are described in [Table S2](#).

The second level relates single-neuron spikes to the activity of other neurons and population patterns. These metrics include spike cross-correlograms (CCGs), quantitative identification of putative monosynaptic connections, and phase relationships to various LFP patterns and to unit-population patterns. Monosynaptic connections, in turn, can be used to identify putative excitatory and inhibitory neurons, and that information can be used to refine the primary unit classification ([Figure 1E](#); [Barthó et al., 2004](#); [English et al., 2017](#)). All parameters can be customized according to the needs of each experimental paradigm ([Table S1](#); <https://cellexplorer.org/datastructure/standard-cell-metrics>).

The third-level metrics are used to assess the relationship between firing patterns of neurons and overt behaviors, including immobility, locomotion, and running speed. First- to third-level metrics can further be supported by other more-direct methods, which can bind physiological parameters to genetically identified neuron groups ([Boyden et al., 2005](#); [Buzsáki et al., 2015](#); [Roux](#)



and Buzsáki, 2015). Because these three-level metrics of single-unit features are generalizable, they can be readily compared with similar analyses across laboratories, independent of paradigm-specific features. Toward these goals, the processing module automatically generates all cell metrics in a standardized fashion.

Features related to any behavioral paradigms can also be computed, including manipulations (e.g., post-stimulus time histograms [PSTHs]), behavioral tracking (spatial firing rate maps), and task-related trial-wise response curves (e.g., response to a sensory cue).

### Data structure

The data structure of CellExplorer (the format is documented online and summarized in [Figures 2](#) and [S3](#)) is organized in data containers and MATLAB structured arrays (structs), which functionally separate different data content, making them both easily interpretable (human readable), machine-readable, expandable, and flexible. That structure is derived from Buzcode (a MATLAB-based data format for electrophysiological recordings and toolsets developed communally in the Buzsáki laboratory; <https://github.com/buzsakilab/buzcode>), Neurosuite ([neurosuite.sourceforge.net](https://sourceforge.net/projects/neurosuite)), and the Freely Moving Animal (FMA) Toolbox ([fmatoolbox.sourceforge.net](https://sourceforge.net/projects/fmatoolbox)). Using a data format in MATLAB's native mat files (binary MATLAB files storing workspace variables) provides greater flexibility for day-to-day analysis—where the codebase can change rapidly with user-dependent requirements for saving derived data—than some of the recently developed data standards, such as NWB (Teeters et al., 2015), which uses a single HDF5 container per session. Instead, scripts are available for translating standard fields of the data containers into NWB, including spikes, behaviors, and events. The processing module also supports NWB as an input format.

The three most-relevant structures are the session metadata struct, the spikes struct, and the cell\_metrics struct.

### Session struct

The session metadata struct contains all session-level experimental metadata ([Figure S1](#)). A session is defined as a set of data typically recorded within the same day, in the same subject (also commonly referred to as a single dataset). The session struct has a modular structure (example modules: general, animal subject, extracellular, and brain regions), which makes it flexible, expandable, and interpretable, and it offers a single structure, capable of handling a wide range of types of metadata related to extracellular data collection and processing. A metadata graphical user interface (GUI; `gui_session.m`; [Figure S2](#)) allows for intuitive metadata entry and inspection, and a template script (`sessionTemplate.m`) can assist in both importing existing experimental metadata and generating relevant fields. Well-curated and documented metadata are crucial for reproducibility and are always need to be linked to the electrophysiological data. See <https://cellexplorer.org/datastructure/data-structure-and-format/> for more information.

### Spikes struct

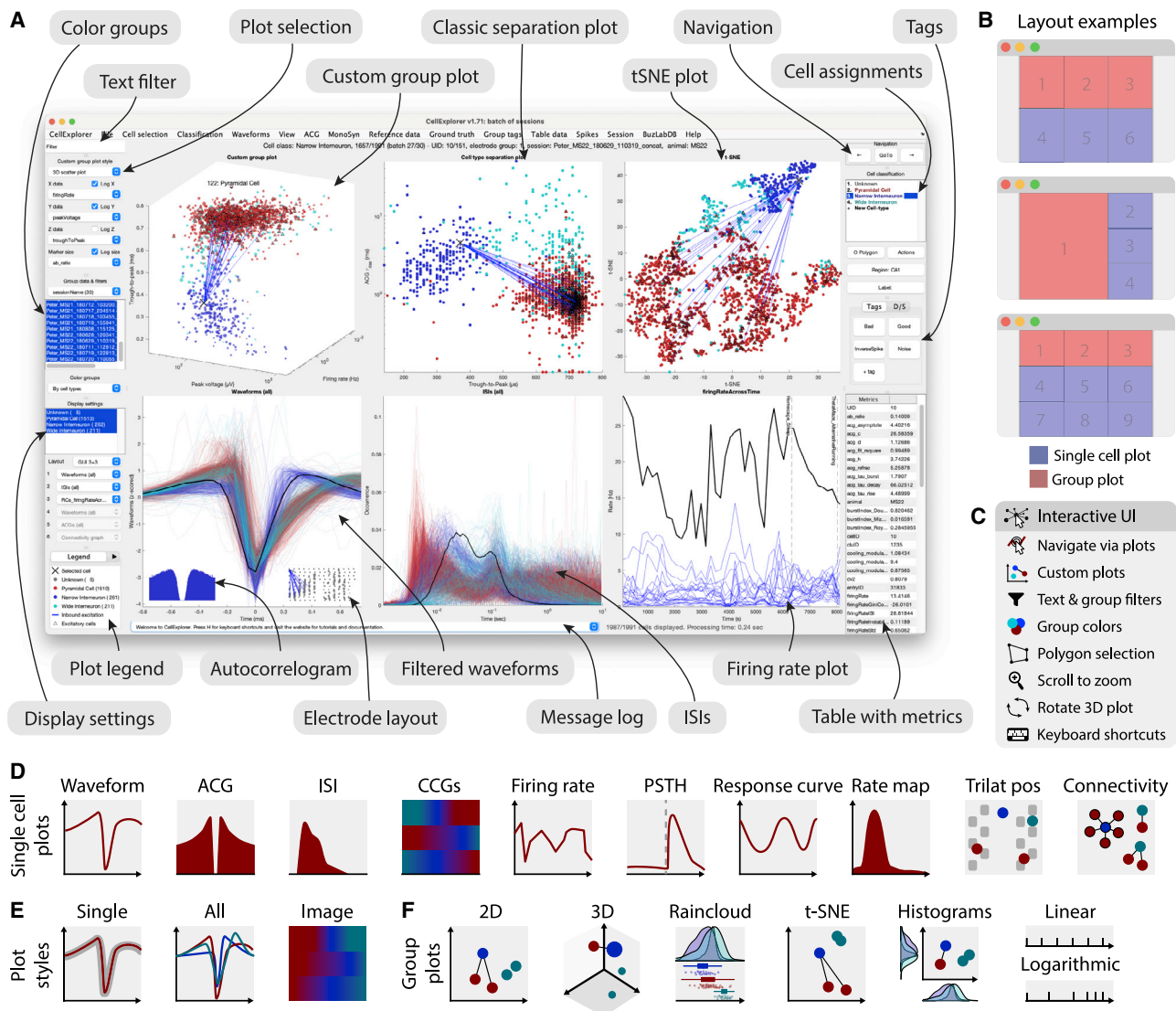
The spike struct contains spike times and cluster IDs of all spikes. It also contains all basic spike-derived fields, such as the average spike waveform of each cell and the peak voltages. These fields are expandable.

### Cell\_metrics struct

The cell\_metrics struct is a modular structure containing all cell metrics calculated in the processing module. It consists of three types of data fields for handling the diverse types of data: numeric-double, character-cells, and structs. Single-value metrics (e.g., peak voltage, firing rate, and putative cell type) are stored in numeric double or character cells. Time series (e.g., waveforms), group data (e.g., synaptic connections and user-defined tags), and session parameters are stored in predefined struct modules. This structure makes the content machine readable, including user-defined metrics, and provides expandability and flexibility (users can add their own metrics) and maintains compatibility with the graphical interface. The single struct allows for processing multiple sessions together in the graphical interface (batch processing) and is convenient for sharing with collaborators and the broader scientific community in publications (see [supplemental information](#) and [Table S1](#) for a detailed description and <https://cellexplorer.org/datastructure/standard-cell-metrics/>). To address cross-platform compatibility, we have also provided two other cell-metrics formats: NWB and JavaScript Object Notation (JSON) files. CellExplorer can save the cell metrics to these formats and load them back into the default MATLAB struct format (please see the NWB tutorial for further info: <https://cellexplorer.org/tutorials/nwb-tutorial/>).

### The CellExplorer graphical interface

The most important component of the framework is the user-friendly graphical interface for single cells, called CellExplorer ([Figure 3](#); [Video S1](#)), which allows for characterization and exploration of all single-unit metrics through a rich set of high-quality, built-in, interactive plots; neuron grouping; cross-level pointers; and filters. The user can interactively select plots and metrics to show in a highly dynamic manner. In the typical layout of CellExplorer with two rows of plots ([Figures 3A](#) and [3B](#), top panel), the top row consists of population-level representations, and the bottom row consists of single-cell features. Individual neurons can be selected from any plot, and the other features of the selected neurons will be automatically updated. The user can zoom and pan by scrolling and dragging any plot ([Figure 3C](#)). A middle mouse click links to the selected neuron and a right mouse click selects the neuron(s) from any of the plots for further actions. These selected groups can be displayed alone or highlighted and superimposed against data in the same session, multiple sessions, or the entire database. Clusters of neurons of interest can be selected by drawing polygons with the mouse cursor, and the other features of the selected groups are shown separately through group actions. Multiple group selections are also possible for both visualization and statistical comparison. Flexibility is assisted by side panels on both sides of the graphs. The left side panel contains options for the custom group plots, color groups, display settings, selection of single-cell plots, and legends. The right panel contains single-cell actions, including navigation elements, cell-assignment actions, tags, and a table with the metrics. The left side panel also includes a text field for custom filters (e.g., numeric and string filters). Below the graphs is a message log, keeping track of user actions. CellExplorer also has a built-in track record of all



**Figure 3. Graphical interface**

(A) The interface consists of nine to nine main plots; in which, the top row is dedicated to population-level representations of the neurons. Other plots are selectable and customizable for individual neuron (e.g., single waveforms, ACGs, ISIs, CCGs, PSTHs, response curves, and firing-rate maps). The surrounding interface consists of panels placed on either side of the graphs. The left side displays settings and population settings, including a custom-plot panel, color-group panel, display-settings panel, and legends. The right side displays single-cell dimensions, including a navigation panel, neuron-assignment panel, tags, and a table with metrics. In addition, there is a text filter and a message log.

(B) Layout examples highlighting three configurations with one to three group plots and three to six single-neuron plots.

(C) The interface has many interactive elements, including navigation and selection from plots (left mouse click links to selected cell, and right mouse click selects the neuron from all the plots), visualization of monosynaptic connections, various data-plotting styles (more than 30+ unique plots are built in), supports custom plots; plotting filters can be applied by text or selection, keyboard shortcuts, zooming any plot by mouse-scrolling and polygon selection of neurons.

(D) Single-cell plot options: waveform, ACG, inter-spike interval (ISI), firing rate across time, post-stimulus time histogram (PSTH), response curve, spatial firing rate maps, trilaterated neuronal position relative to recording sites, and monosynaptic connectivity graph.

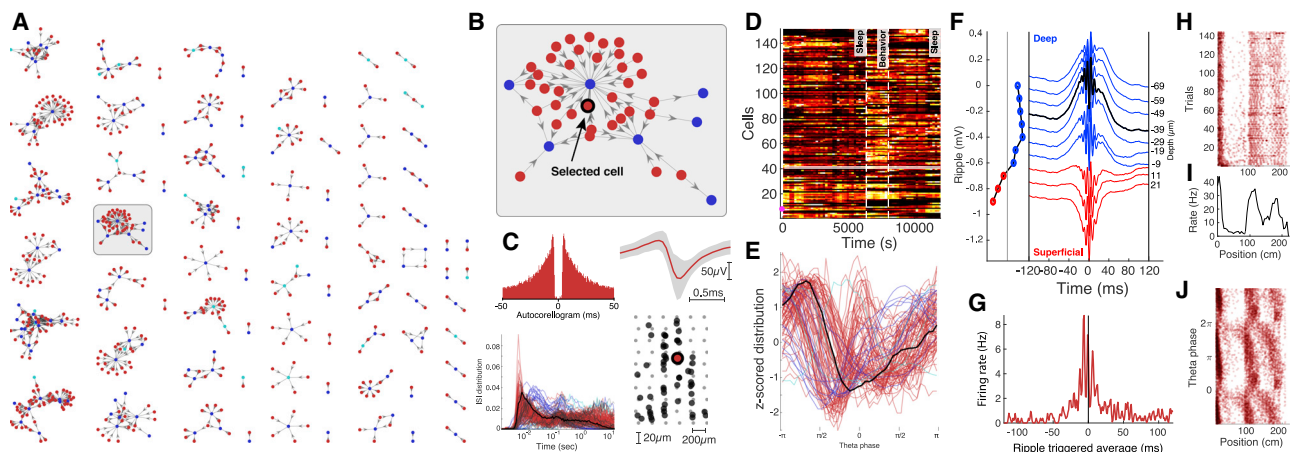
(E) Most single-cell plots have three representations: individual single-cell representation, single cell together with the entire population with absolute amplitude, and a normalized image representation (color map).

(F) Group plotting options: 2D, 3D, raincloud plot, t-SNE, and double histogram. Each dimension can be plotted on linear or logarithmic axes.

See also [Figures S4, S5, and S6](#) and [Video S1](#).

user actions, which can be undone in a stepwise manner. The group plotting options include a two-dimensional (2D)-representation, a 3D-representation, a double histogram, a raincloud plot (Allen et al., 2019), and a customizable dimensionality-reduction

plot (T-distributed stochastic neighbor embedding [t-SNE], principal component analysis [PCA], and uniform manifold approximation and projection [UMAP]; van der Maaten and Hinton, 2008; see group plot options in [Figure S4](#)). Axis scaling can be



**Figure 4. Data exploration example**

(A) Connectivity graph with monosynaptic modules found across multiple datasets. Neurons are color coded by their putative cell types (pyramidal cells in red, narrow interneurons in blue, and wide interneurons in cyan).  
 (B) Highlighted monosynaptic module with single pyramidal cell highlighted (arrow).  
 (C) First-level metrics: auto-correlogram, average waveform (top row; gray area signifies the noise level of the waveforms), ISI distributions, with the selected neuron in black, and the physical location of the neurons relative to the multi-shank silicon probe.  
 (D) Firing rate across time for the population; each neuron is normalized to its peak rate. The session consists of three behavioral epochs: pre-behavior sleep, behavior (track running), and post-behavior sleep (boundaries shown with dashed lines).  
 (E) Theta phase distribution for all neurons recorded in the same session (red, pyramidal cells; blue, interneurons) during locomotion with the selected neuron highlighted (black line).  
 (F) Average ripple waveform for the electrode sites on a single shank. The site of the selected neuron is highlighted (dashed black line). The polarity of the average sharp wave is used to determine the position of the neuron relative to the pyramidal layer in CA1.  
 (G) Ripple wave-triggered PSTH for the selected neuron aligned to the ripple peak.  
 (H) Trial-wise raster for the selected neuron in a maze.  
 (I) The average firing rate of the neuron across trials.  
 (J) Spike raster showing the theta-phase relationship to the spatial location of the animal.

either linear or logarithmic (Figure 3F). Many of the cell features have multiple visualizations and metrics. Let's take the spike waveform, as an example. It can be represented by (1) a single, average, filtered/raw waveform from the channel with the largest amplitude; (2) in comparison to the Z scored/absolute waveforms of the population; (3) in comparison to group averages (e.g., cell types); (4) waveforms across the probe: as a probe-layout-projected representation, as an image representation, and as a spike-amplitude distribution; (5) single-cell metrics: trough-to-peak (a measure of the waveform width), an AB ratio (a measure of the asymmetry of the waveform), peak voltage (the amplitude of the waveform), and peak channel; and (6) the temporal aspects of the spike amplitude across the entire recording, as a stability measure, which can be visualized through spike rasters (see a subset of the plotting options in Figure S5).

Examples of the flexible operation of the graphical interface module are illustrated in Figure 4 and are described in more detail in Video S1. Here, we begin with motifs of monosynaptically connected clusters of neurons from the hippocampal CA1 area, as determined by the processing module (Figure 4A). An example sub-network of connected neurons is highlighted in Figure 4B, with a selected single neuron to be characterized (arrow). Selected first- to third-level metrics of the neuron are displayed in Figures 4C–4G. In several panels, the metrics of the selected neuron are shown against other neurons from the same dataset.

A middle mouse click on any neuron will update all the panels for that particular cell, allowing quick screening and qualitative evaluation of multiple features. Neurons of interest can be marked for further quantitative comparisons. Next, first- to third-level metrics can be compared with paradigm-specific features of the selected neuron(s). For example, in the case of the hippocampal neurons, place field, trial-by-trial variability of firing patterns, travel direction firing specificity, spike-phase precession relative to theta-oscillation cycles, and multiple other features are possible. During the data-mining process, unexpected features and outliers may be noted, instabilities of neurons ("drifts") can be recognized, and artifacts can be identified visually. Such experimenter-supervised judgments are also essential for evaluating the quality of the quantified data processing.

## Performance

CellExplorer takes advantage of MATLAB's lowest-level graphical plotting methods and has optimized and enhanced plotting algorithms to make the user interface (UI) responsive. Benchmarks were performed to characterize the performance of the CellExplorer GUI (Figure S6), which consists of multiple graphical elements with up to nine individual, simultaneous visualizations; panels; and interface elements; all of which are updated as the user navigates the single-cell data.

Most of the individual single-cell plots (Figures 3D and S5) takes less than 80 ms to display (tested on an iMac from 2017

with a 4.2 GHz quad-core Intel i7 with 32 GB of random access memory [RAM]), even when representing thousands of cells, but a subset of the visualizations increases substantially with the number of cells (the trace presentations with many data points per cell; e.g., ACGs, ISIs, waveforms, and theta phase; [Figure S6A](#)). However, by introducing an overall display cap of 2,000 simultaneously random cells in these representations, all of the visualizations take less than 80 ms, except the connectivity graph, in which all connections are always shown. Updating the whole UI ranged from 180 ms to 300 ms, with the 2,000 cells cap, for four and nine subpanels, respectively ([Figure S6B](#)). These benchmarks capture CellExplorer's performance on real data and its scaling on larger dataset yet reflect, as well, the graphical performance of MATLAB, with built-in graphical hardware support and the hardware configuration.

### Putative cell-type characterization

Currently, the processing pipeline classifies putative cell types based on two parameters, the width of the spike waveform (measured by the waveform “trough-to-peak”) and the burstiness of spiking (measured via the rise time of the ACG [ $\tau_{\text{rise}}$ ]; [Petersen and Buzsáki, 2020](#); [Senzai and Buzsáki, 2017](#)). The ACG of each cell (from  $-50$  ms to  $50$  ms) is parameterized using a triple exponential fit ([Figure S7](#)). One exponential fits the rise time of the ACGs, and two further exponentials fit the burst-related fast decay and the slower decaying part of the ACG. Using the waveform and burstiness criteria, units are tentatively segregated to narrow waveform (trough-to-peak  $\leq 450$   $\mu$ s), wide waveform (trough-to-peak  $> 450$   $\mu$ s and  $\tau_{\text{rise}} > 6$  ms), putative interneurons, and the rest, as pyramidal cells ([Figure 5A](#)). These boundaries were optimized separately on data from the hippocampus and visual cortex and can be adjusted for neurons in other brain regions when running the processing module. In a separate step, monosynaptic excitatory and inhibitory connections, identified based on their short-time CCGs ([Barthó et al., 2004](#); [English et al., 2017](#)), can serve to verify the goodness of the first-order unit classification. In the dataset shown in [Figure 5](#), 39% of the pyramidal cells were determined to be excitatory ([Figure S8A](#)) and 5% of the narrow interneurons as inhibitory cells. There was a high degree of excitatory convergence on putative, narrow interneurons ([Figure S8B](#)). Population-average ACGs and waveforms are shown in [Figure 5C](#). A t-SNE dimensionality reduction further supported the reliability of the neuron separation ([Figure 5D](#)), and an agglomerative clustering determined similar clusters ([Figure 5E](#)). When more ground-truth data become available, the above classification scheme and boundaries will inevitably change ([Figure 5F](#)). The reliability of any unit-behavior correlation can be compared with cluster quality metrics ([Figures 5G and 6](#); [Schmitzer-Torbert et al., 2005](#)), increasing (or decreasing) the confidence in the validity of the correlation in question.

### Value of large inter-laboratory datasets

Although progress in discovery science often depends on an investigator-unique approach to novel insights, standardization of data processing and screening is essential in fields in which big-data generation is achieved through collaborative efforts. This applies to the current effort to quantitatively relate physi-

ology-based and genetically classified cell types ([Klausberger and Somogyi, 2008](#); [McBain and Fisahn, 2001](#); [Rudy et al., 2011](#)). In each experiment, typically only one or a few neuron types can be identified. However, combining datasets from numerous experiments and different laboratories can generate physiological metrics, grounded by other “ground-truth” data.

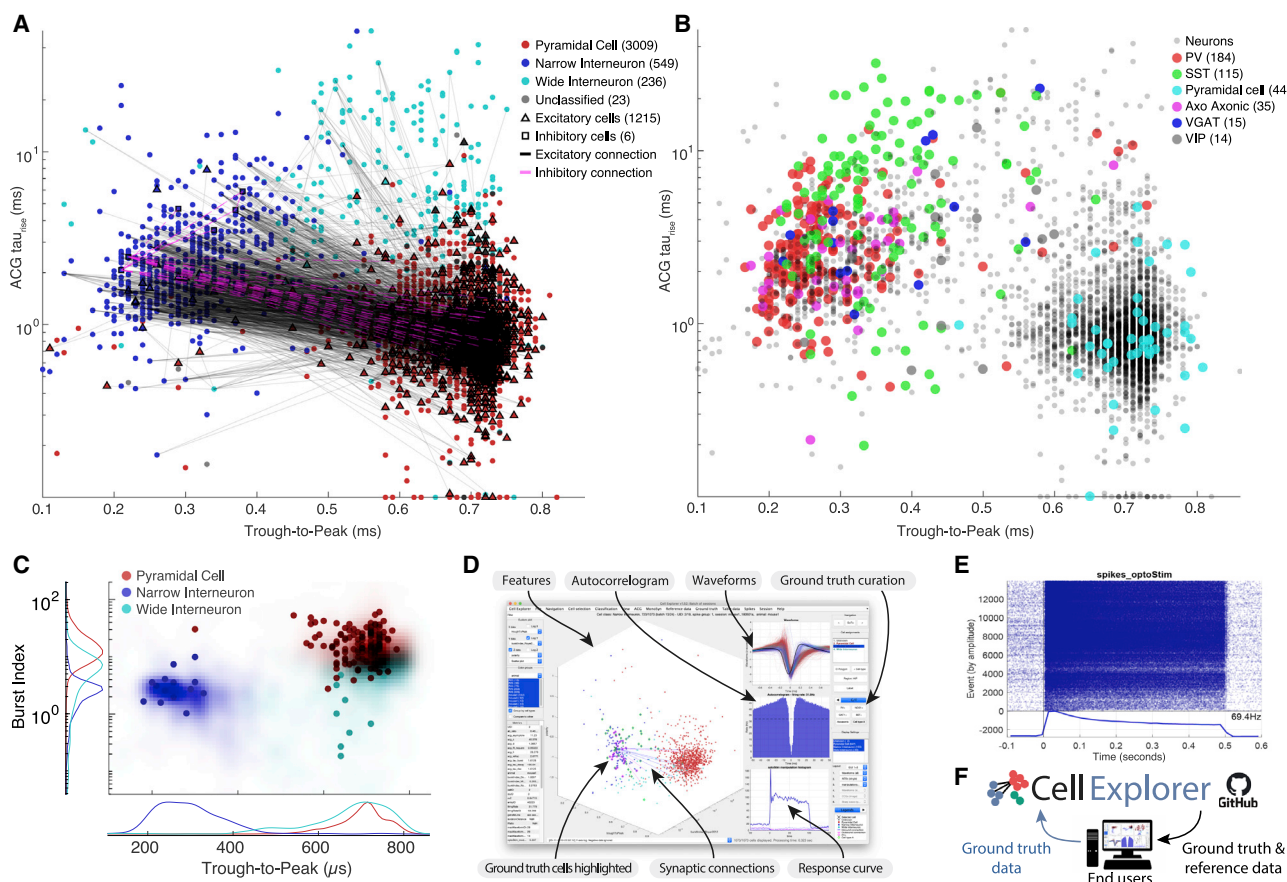
[Figure 5](#) also serves to illustrate the feasibility and utility of community-based approach. First- to third-level metrics of neurons recorded from the same brain region and layer can be combined from multiple experiments and laboratories and contrasted to the data quality of the units recorded in a single session. An ever-growing dataset allows for more-reliable modality separation and characterization of neuron types. For example, the initial divisions of neurons into putative pyramidal cells and narrow and wide interneurons can be further refined by quantifying monosynaptic connections, increasing confidence of pyramidal-cell-interneuron separation as well as identifying subsets of the unclassified group as interneurons ([Figure 5A](#)) ([Mizuseki et al., 2011](#); [Petersen and Buzsáki, 2020](#); [Peyrache et al., 2015](#); [Stark et al., 2013](#)).

Single neurons identified by opto-tagging, juxtacellular, or other direct means ([Ciocchi et al., 2015](#); [Klausberger and Somogyi, 2008](#); [Royer et al., 2012](#); [Senzai et al., 2019](#); [Stark et al., 2012](#); [Zhang et al., 2013](#); [Roux and Buzsáki, 2015](#)) can be used to link first- to third-level features of initially classified neurons to genetically defined neuron types ([Figures 5A–5E](#)). Optogenetic methods can be supplemented by other more-direct methods, such as simultaneous juxtacellular-extracellular or intracellular-extracellular recordings ([Figure 5F](#); [Harris et al., 2000](#); [Neto et al., 2016](#)). Having access to these ground-truth labels may offer further support for the validity of physiological classification. An expected outcome of such a growing dataset, containing ground-truth-verified neurons, is trained models for classifying diverse neuron types based on physiological metrics alone. This is especially important for recordings in model organisms for which genetic manipulations are less tractable than in mice. Opto-tagged neurons can be analyzed separately in CellExplorer ([Figures S8G and S8H](#)). Further manual curation can be done when accessing the neuron's other characteristics, including waveforms, firing rates, and connectivity. Communal contribution of ground-truth data to CellExplorer is possible through the public GitHub repository ([Figure S8I](#); visit <https://cellexplorer.org> for tutorials and further details).

Using many shared datasets, brain regions, different electrode types, and other features can begin to be compared efficiently, using t-SNE plots ([Figure 5E](#)). Such representations can highlight inconsistencies and differences across recording sessions, identify important regional and layer-specific differences, and alert for interspecies characteristics ([Figure 8](#)).

CellExplorer uses and shares data through our laboratory databank (<https://buzsakilab.com/wp/database>; [Petersen et al., 2020](#)). To demonstrate the value of inter-laboratory comparison, we processed datasets from the CA1 region of the hippocampus ([Figure 6A](#)) and visual cortex ([Figure 6B](#)) in freely moving mice ([Senzai et al., 2019](#); [Petersen and Buzsáki, 2020](#); [Petersen et al., 2020](#)) and comparable data from head-fixed mice from





**Figure 5. Community-based collaborations allow for improved single-neuron characterization**

(A) Distribution of putative cell types (3,657 cells), including their projections, determined via spike-transmission CCG curves (Petersen and Buzsáki, 2020; Petersen et al., 2020). Excitatory and inhibitory cells determined from monosynaptic connections are highlighted with black triangles and magenta squares respectively. The marginal distributions are shown both as counts and probability distributions.

(B) Example ACGs for the three cell types and the ACG fit (black line).

(C) Top row: average peak-normalized ACGs of the three cell types. Bottom row: average waveform for the three cell types (Z scored).

(D) t-SNE representation of the same cell population.

(E) Agglomerative clusters of data with two (top) and three (bottom) clusters.

(F) 407 optogenetically identified neurons, including PV (184), SST (115), pyramidal cells (44), axo-axonic (35), VGAT (15) and VIP cells (14) projected onto the same population of neurons as in (A) (Sources: Allen Institute and Buzsáki lab; English et al., 2017; Senzai et al., 2019; Siegle et al., 2021).

(G) Isolation distance in cluster space for the population shown in (A).

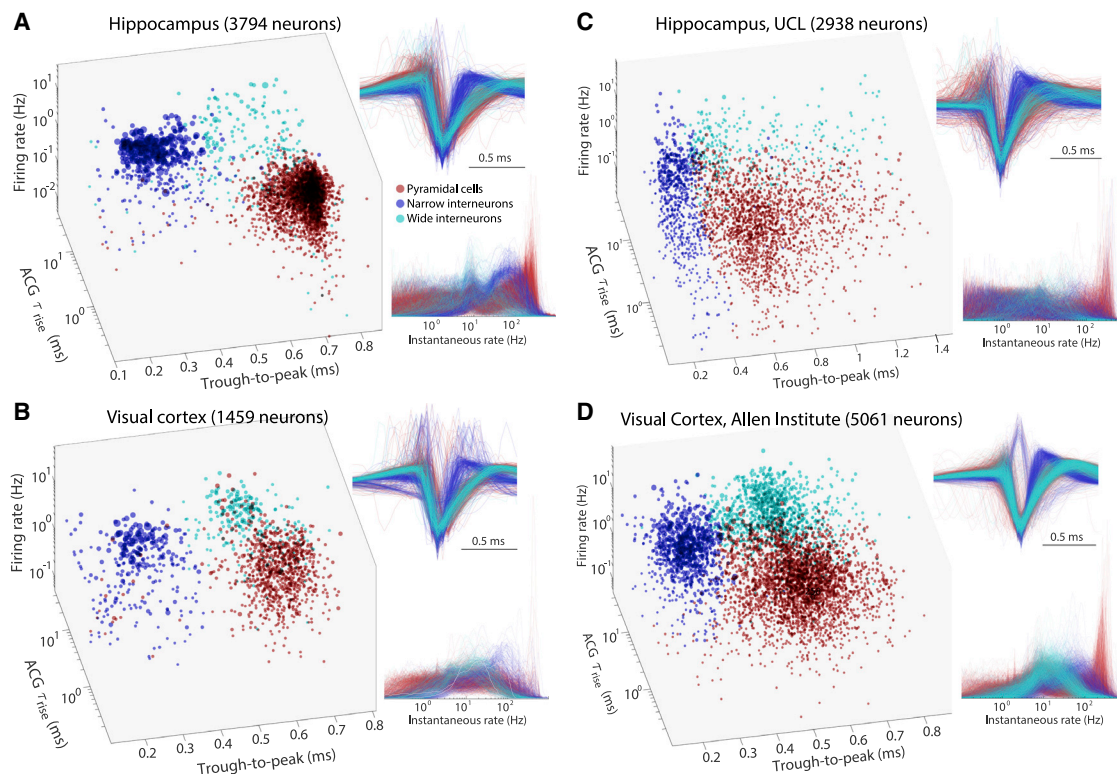
See also Figures S7 and S8.

two large public datasets from the University College London (UCL; Figure 6C; Steinmetz et al., 2019) and the Allen Institute (Figure 6D; (Siegle et al., 2021)). Through our database and CellExplorer, we currently share more than 79,000 processed neurons publicly. Datasets can be downloaded directly in CellExplorer and used as reference data or explored directly. The infrastructure is designed toward continually growing the public datasets and ground-truth data for discovery science, cross-laboratory interactions, and reproducibility control. Processing data collected in different laboratories and under different investigators by the same program(s) will allow investigators to standardize protocols and achieve greater reliability of inter-laboratory experiments between neuronal firing patterns and their behavioral, cognitive correlates.

## NeuroScope2—a data viewer for raw and processed data

To take advantage of the data types of CellExplorer and to back-project the results to the recorded data, we also built a data viewer, NeuroScope2 (Figure 7). It maintains many of the original functions of NeuroScope (Hazan et al., 2006) and enhances its performance to explore existing data, stream data being collected, and handle multiple data streams simultaneously (e.g., digital or analog traces, together with processed ephys data). Because NeuroScope2 is written in MATLAB, it is hackable, adaptable, and easily expandable. It functions fully within the data types of CellExplorer, using the session struct for metadata and supports the data containers of CellExplorer. NeuroScope2 can show the processed spikes and take





**Figure 6. Comparison of initial neuron classification by CellExplorer on large-scale datasets from three different laboratories**

(A–D) Data from hippocampus (Petersen and Buzsáki, 2020) (A). Data from visual cortex (Senzai et al., 2019) (B). Hippocampal and visual neurons selected from the UCL dataset (Steinmetz et al., 2019) (C). Visual cortex cells from the Allen Institute (Siegle et al., 2021) (D). Right panels across (A)–(D): Z scored waveforms across all neurons (top) and distribution of instantaneous rates (1/interspike intervals) across all neurons. (A) and (B) are based on long home cage (sleep) data (several hours), whereas (C) and (D) data are from short (~30 min) sessions in head-fixed, task-performing mice. See also Figure S8. Red, pyramidal cells; blue, narrow waveform interneurons; cyan, wide waveform interneurons.

advantage of the cell metrics to filter, group, and color single units, as well as show behavioral, time series, states, and event data (Figure 7B).

The interface is user friendly, with a single side panel for accessing most functions. Users can zoom, navigate, measure, highlight, and select traces directly with the mouse cursor, making manual inspection intuitive and efficient. NeuroScope2 can also perform basic data processing on the viewed traces, e.g., filter the bandpass, perform temporal smoothing, generate single-channel spectrogram, perform current source density (CSD) analysis, and detect spikes and events.

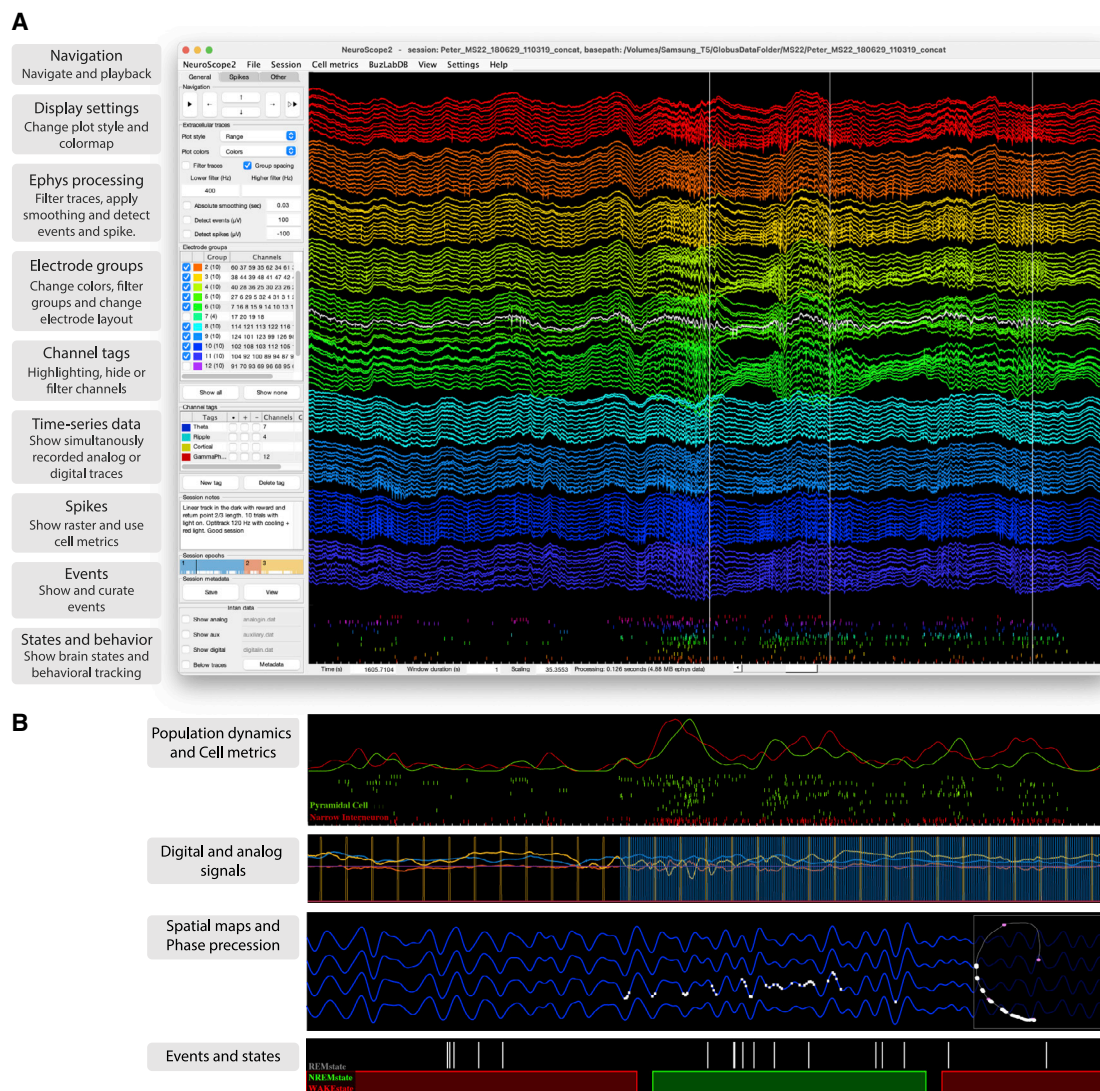
### Expandability of CellExplorer

CellExplorer comes with a long list of predefined metrics, advanced plots, ground-truth data, and classification schemes, allowing for standardized and efficient processing and visualization. Users can add their own metrics (numeric, strings, or other more-complex data) to the cell metrics, which will be available in CellExplorer (see tutorial in the [supplemental information](https://cellexplorer.org/datastructure/expandability/) and website for further details: <https://cellexplorer.org/datastructure/expandability/>). Further expandability and customization are achieved via MATLAB package folders (“+folders”). Users can add their own custom calculations to the pipeline, implement custom classification

schemes, provide preferences, create their own CellExplorer plots, and incorporate their ground-truth data. Relevant tutorials, example code, and templates are included online.

### DISCUSSION

We have developed CellExplorer, an open-source, MATLAB-based resource for characterizing single neurons and neuron types based on their biophysical features for collaborative analysis of data collected within and across laboratories. The CellExplorer platform enables visualization and analysis for users without the need to write code. Its modular format allows for fast and flexible comparisons of a large set of preprocessed physiological characteristics of single neurons and their interactions with other neurons, as well as their correlation with experimental variables. The code is publicly available on GitHub for users to download and to use the same standardized processing module on their local computers (Windows, OS X, and Linux). CellExplorer offers step-by-step online tutorials for first-time users. It is linked to the Allen Institute reference atlas to relate recording sites with structures and layers (Chon et al., 2019; Wang et al., 2020; <https://atlas.brain-map.org/>) with potential for expansion to other online resources that provide annotated data on putative neuron types.



**Figure 7. NeuroScope2—a data viewer for raw and processed extracellular data acquired using multisite silicon probes, tetrodes, or single wires**

NeuroScope2 is written in MATLAB, maintaining many of the original features of NeuroScope ([neurosuite.sourceforge.net](https://neurosuite.sourceforge.net)) but with many enhancements, and NeuroScope2 is faster. It is easy to hack or modify and supports and relies on the data types of CellExplorer.

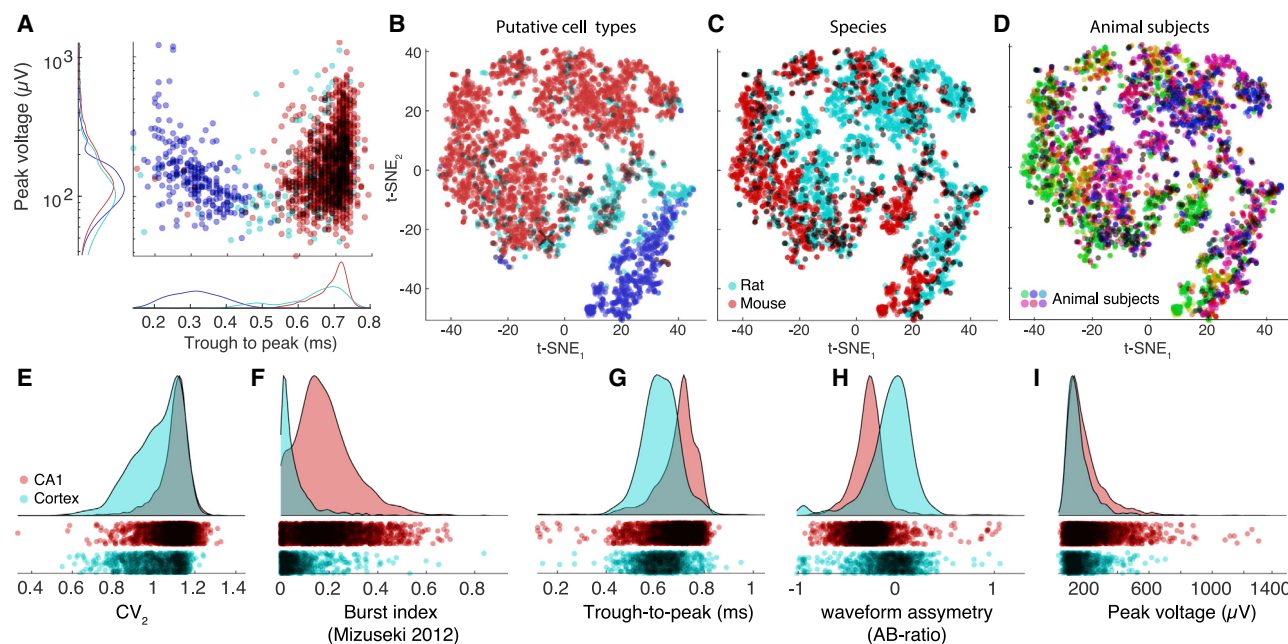
(A) Screenshot of the graphical interface, showing a 128-channel recording from the rat hippocampus (window duration = 1 s). Each colored groups of traces are from the same silicon probe shank. The three vertical lines are detected temporal events (sharp-wave ripples; the detection-channel is highlighted in white). The rasters below the traces are the spikes from curated single units. The left side panel consists of three tabs: general (panels: navigation, ephys traces, electrode groups, channel tags, session notes and epochs, and Intan time-series), spikes (panels: spikes, cell metrics, and population dynamics), and other (panels: events, states, time-series, and behavioral data).

(B) Various visualizations with NeuroScope2. Top row: population-average curves and spiking dynamics of the same population of cells as in (A) but color coded and grouped using the putative cell type determined via CellExplorer (window duration: 1 s). Second row: two digital transistor-transistor logic (TTL) pulses and 3D accelerometer data (mounted on the animal's head). Digital data captured using the Intan acquisition system (the TTL pulses are emitted by a 10-Hz camera and a 120-Hz behavioral tracking systems; window duration: 3 s). Third row: Ephys traces filtered in the theta band, with spikes of a single place cell projected on the same trace (white points; window duration: 3 s), the right square shows the animal's 2-dimensional spatial trajectory (gray line), and the white points indicate the spatial location of the place field. Bottom row: event raster and states data (window duration: 50 s).

### Multiple-level characterization and classification of single neurons

To correctly interpret neuron firing-behavior/cognition relationships, numerous controls are needed to rule out or reduce the potential contribution of spurious variables. The processing

module generates a battery of useful metrics for that purpose. In addition to the first-level description of the biophysical characteristics of single neurons, it computes brain-state-dependent firing rates, interspike interval variation, and relationships between single neurons and spiking activity of the population and



**Figure 8. Exploration and comparison of metrics and cells across species, subjects, and brain regions**

(A) Distributions of spike amplitudes and waveform width (quantified by the trough to peak metrics) for the three groups from multiple CA1 datasets. Note inverse relationship between spike amplitude and waveform for putative interneurons.

(B–D) t-SNE representations of putative cell types (B), species (C) (rat, and mouse in magenta and red, respectively), and subjects (D) (colors scaled across subjects) for hippocampal neurons.

(E–I) Comparison of spike features of neurons recorded from CA1 pyramidal cells and visual cortex pyramidal cells. Significant differences are observed across several basic metrics, including CV<sub>2</sub> (E), burst index (F), trough-to-peak (G), waveform asymmetry (H), and waveform peak voltage (I).

LFP (second level). When behavioral data are also available, it can describe the relationship between single neuron-firing patterns and routine behavioral parameters, such as immobility, walking, respiration, and pupil diameter (third level) or other task-independent measures, whose incidence or magnitude may change in the task. The third-level metrics can help avoid, e.g., inappropriately attributing spiking activity to high-level phenomena, such as learning, perception, or decision making, which are often linked to overt movement and autonomic changes. Because these three-level metrics are independent of particular experimental paradigms, they can be used as benchmarks for assessing consistencies across experiments performed by different investigators in the same laboratory or across laboratories (Figure 6). Concatenating datasets obtained from the same brain regions and layers will create a continuously growing data bank. In turn, these datasets make it possible to identify and quantify reliable boundaries among putative clusters and suggest inclusion and exclusion of parameters for a more-refined separation of putative neuronal classes. Sets from different brain regions can be readily compared to identify salient differences.

Although several statistical tests are available in CellExplorer, it is not meant to substitute for rigorous quantification. Instead, it is designed as a tool for flexible and fast comparison of parameters, facilitating visualization, interpretation, and discovery. It is a complementary approach to dimensionality-reduction and population-analysis methods. Because assemblies of neurons

consist of highly unequal partners (Buzsáki and Mizuseki, 2014), knowledge about the neuron-specific contributions to population measures is critical in many situations (Nicolelis and Lebedev, 2009). Such inequality may stem from unknowingly lumping neurons of different classes together into a single type and because even members of the same type belong to broad and skewed distribution, which may contribute to different aspects of the experiment (Grosmark and Buzsáki, 2016).

#### Public cell metrics built from optotagged cells and reference sessions

Various classification schemes have been developed to assign extracellular spikes to putative pyramidal cells, interneurons, and their putative subtypes, based on a variety of physiological criteria. These include waveform features, firing rate statistics in different brain states, embeddedness in various population activities, firing patterns characterized by their ACGs, and putative monosynaptic connections to other neurons (Barthó et al., 2004; Csicsvari et al., 1999; Fujisawa et al., 2008; Mizuseki et al., 2009; Okun et al., 2015; Sirota et al., 2008). Increasingly larger datasets will likely improve such physiology-based classification. However, the ground truth for these classifying methods is largely missing. There is a lack of agreement about neuronal “types” across laboratories, and even for data collected in the same laboratory, different criteria are applied in different experiments. Optogenetic tagging (Boyden et al., 2005) offers such grounding by connecting putative subtypes based on



physiologically distinct features to their molecular identities. Because, in a single animal, only one or a few neuron types can be tagged optogenetically or identified by other direct methods (Fosque et al., 2015; Klausberger and Somogyi, 2008), refinement of a library of physiological parameters should be conducted iteratively, so that, in subsequent experiments, the various neuron types can be recognized reliably by using solely physiological criteria (English et al., 2017; Royer et al., 2012; Senzai and Buzsáki, 2017, 2017; Roux and Buzsáki, 2015). In turn, knowledge about the molecular identity of the different neuronal components of a circuit can considerably improve the interpretation of correlational observations provided by large-scale extracellular recordings.

CellExplorer provides the end user with access to processed cell metrics of a wide range of opto-tagged cells and reference data (~79,000 neurons) collected in our database. These features can serve as benchmarks for comparison with data collected in any other laboratory and can assist with the initial neuron classification into the broad groups of pyramidal cells and interneurons; many of which are identified physiologically by their monosynaptic connections. The derived features also offer normative information about spike characteristics, firing rates, and spike dynamics, as captured in Figure 8. The ground-truth neurons are included in the CellExplorer GitHub repository and can be loaded from CellExplorer, while the reference data must be downloaded from our webshare.

### Comparison with other frameworks and tools

Several community efforts exist for machine- and human-readable databases of various morphological and transcriptomic features of neurons (Ascoli et al., 2007; Cembrowski et al., 2016; Sanchez-Aguilera et al., 2021; Tecuatl et al., 2021; Wheeler et al., 2015; neuromorpho.org, <https://hippocampome.org/php/index.php>), which served as the inspiration for our efforts. However, post-spike sorting tools are rare. NeuroExplorer (<http://www.neuroexplorer.com/>; Nex Technologies, USA) is the largest commercial solution, but costs thousands of dollars for a license; it is written in C but supports MATLAB and python code. Brainstorm (Tadel et al., 2011), another application for analysis of brain recordings, recently began supporting ephys data through the IN-Brainstorm expansion (Nasiotis et al., 2019). However, Brainstorm is mostly focused on noninvasive techniques, and the application has limited tools for analyzing single cells. The user interfaces of CellExplorer could potentially be integrated into Brainstorm. FieldTrip (Oostenveld et al., 2011), is another very popular, open-source MATLAB software toolbox for magnetoencephalography (MEG), electroencephalography (EEG), and intracranial EEG (iEEG) analysis but has a limited scope for single-cell analysis.

CellExplorer is open source, with all code, data, and documentation available online. We chose MATLAB, a commercial solution as a platform because it allowed us to build CellExplorer on existing MATLAB tools and historically popular toolboxes and also because MATLAB is a widely used coding platform in many electrophysiological laboratories in academia. An alternative for us was to build CellExplorer around NWB (Teeters et al., 2015). NWB has numerous advantages but provides limited flexibility

and is not ideal for a day-to-day data-analysis format with high demand for flexibility. Therefore, other approaches that enhance flexibility and alternative options are warranted. CellExplorer supports reading spike data from NWB files, thus allows users to combine the advantages of both platforms, including CellExplorer's machine-readable code. CellExplorer also supports saving the metrics to a NWB 2.0 file (Rübel et al., 2019; Teeters et al., 2015) and loading the NWB file back into the default MATLAB cell\_metrics struct. A dedicated NWB tutorial is available here: <https://cellexplorer.org/tutorials/nwb-tutorial/>. Handling the cell metrics in simpler JSON files is also supported. CellExplorer could be translated to an open platform like python or Julia, pending demand.

### Outlook and future directions

Through community efforts, there are many future potential directions for the use of CellExplorer. CellExplorer allows for user-friendly submission of ground-truth (e.g., opto-tagged) data that can be shared with the community via the GitHub repository. To date, CellExplorer hosts the largest collection of publicly available, characterized, single neurons. It also has the largest collection of opto-tagged cells, which will help link physiological markers and characteristics to genetic traits. Through implementation of publicly shared data by contributing neuroscientists, its scope can be expanded further for better coverage of several brain regions, species, and behaviors.

### Development and availability

Development takes place in a public-code repository at <https://github.com/petersenpeter/CellExplorer>. All examples in this article have been calculated with the pipeline and plotted with CellExplorer. Extensive documentation, including installation instructions, tutorials, and a description of all metrics and their calculations, is available at <https://cellexplorer.org> and hosted at the GitHub repository. CellExplorer is available for MATLAB 2017B and forward and for the operating systems Windows, OS X, and Linux. Compiled versions of the graphical interfaces (CellExplorer, NeuroScope2, and the session GUI) are available at the CellExplorer website for usage on computer systems without a MATLAB license. More information can be found at <https://cellexplorer.org>. All data presented are available from <https://buzsakilab.com/wp/database/> (Petersen et al., 2020). We pledge to continue to support CellExplorer and are eager to incorporate data generated by other laboratories. CellExplorer should be viewed as a small, but necessary, step toward FAIR practice in collaborative neuroscience and the emerging novel technical platforms that will facilitate data sharing and interlaboratory collective research.

### Tutorials

Two tutorials are available in the STAR Methods section: a general tutorial on the full pipeline, and a tutorial on how to add your own metrics. There are many additional, detailed tutorials online, which cover the generation of the metadata struct, the manual curation process, generating spike raster plots, connections, performing opto-tagging, using ground-truth data, exporting figure, and many other topics.

Tutorials are available at <https://cellexplorer.org/tutorials/tutorials>.

## STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- **KEY RESOURCES TABLE**
- **RESOURCE AVAILABILITY**
  - Lead contact
  - Materials availability
  - Data and code availability
- **EXPERIMENTAL MODEL AND SUBJECT DETAILS**
- **METHOD DETAILS**
  - CellExplorer layout and preferences saved between sessions
  - General tutorial
  - Expandability tutorial: add you own custom metrics
  - Add a string metric to your cell\_metrics
  - Add numeric values to your cell metrics

## SUPPLEMENTAL INFORMATION

Supplemental information can be found online at <https://doi.org/10.1016/j.neuron.2021.09.002>.

## ACKNOWLEDGMENTS

We thank Sam McKenzie, Mihály Vöröslakos, Michelle Hernandez, and Thomas Hainmueller for helping with documentation or implementation of CellExplorer code. Further, we thank Sam McKenzie, Daniel English, Roman Huszar, and Yuta Senzai for providing ground-truth datasets. Finally, thanks go to Thomas Hainmueller, Manuel Valero, Antonio Hernandez-Ruiz, Viktor Varga, and Omid Yaghmazadeh for comments on the manuscript. J.H.S. thanks the Allen Institute founder, Paul G. Allen, for his vision, encouragement, and support. Supported by NIH grants U19 NS107616, U19 NS104590, and R01 MH122391, the Kavli Foundation, and The Lundbeck Foundation.

## AUTHOR CONTRIBUTIONS

P.C.P. and G.B. conceived the study and designed the framework. P.C.P. and S.M. wrote the software. P.C.P., J.H.S., and N.A.S. contributed data. P.C.P. analyzed data. G.B. and P.C.P. wrote the manuscript with input from all authors.

## DECLARATION OF INTERESTS

The authors declare no conflicting interests.

Received: April 24, 2021

Revised: July 19, 2021

Accepted: August 30, 2021

Published: September 29, 2021

## REFERENCES

Allen, M., Poggiali, D., Whitaker, K., Marshall, T.R., and Kievit, R.A. (2019). Raincloud plots: a multi-platform tool for robust data visualization. *Wellcome Open Res.* 4, 63.

Ascoli, G.A., Donohue, D.E., and Halavi, M. (2007). NeuroMorpho.Org: a central resource for neuronal morphologies. *J. Neurosci.* 27, 9247–9251.

Barlow, H.B. (1972). Single units and sensation: a neuron doctrine for perceptual psychology? *Perception* 1, 371–394.

Barthó, P., Hirase, H., Monconduit, L., Zugaro, M., Harris, K.D., and Buzsáki, G. (2004). Characterization of neocortical principal cells and interneurons by network interactions and extracellular features. *J. Neurophysiol.* 92, 600–608.

Bishop, P.O., Burke, W., and Davis, R. (1962). Single-unit recording from antidromically activated optic radiation neurones. *J. Physiol.* 162, 432–450.

Bouchard, K.E., Aimone, J.B., Chun, M., Dean, T., Denker, M., Diesmann, M., Donofrio, D.D., Frank, L.M., Kasthuri, N., Koch, C., et al. (2016). High-Performance computing in neuroscience for data-driven discovery, integration, and dissemination. *Neuron* 92, 628–631.

Boyden, E.S., Zhang, F., Bamberg, E., Nagel, G., and Deisseroth, K. (2005). Millisecond-timescale, genetically targeted optical control of neural activity. *Nat. Neurosci.* 8, 1263–1268.

Buzsáki, G. (2004). Large-scale recording of neuronal ensembles. *Nat. Neurosci.* 7, 446–451.

Buzsáki, G., and Mizuseki, K. (2014). The log-dynamic brain: how skewed distributions affect network operations. *Nat. Rev. Neurosci.* 15, 264–278.

Buzsáki, G., Stark, E., Berényi, A., Khodagholy, D., Kipke, D.R., Yoon, E., and Wise, K.D. (2015). Tools for probing local circuits: high-density silicon probes combined with optogenetics. *Neuron* 86, 92–105.

Cembrowski, M.S., Wang, L., Sugino, K., Shields, B.C., and Spruston, N. (2016). Hipposeq: a comprehensive RNA-seq database of gene expression in hippocampal principal neurons. *eLife* 5, e14997.

Chon, U., Vanselow, D.J., Cheng, K.C., and Kim, Y. (2019). Enhanced and unified anatomical labeling for a common mouse brain atlas. *Nat. Commun.* 10, 5067.

Chung, J.E., Magland, J.F., Barnett, A.H., Tolosa, V.M., Tooker, A.C., Lee, K.Y., Shah, K.G., Felix, S.H., Frank, L.M., and Greengard, L.F. (2017). A fully automated approach to spike sorting. *Neuron* 95, 1381–1394.e6.

Ciocchi, S., Passecker, J., Malagon-Vina, H., Mikus, N., and Klausberger, T. (2015). Brain computation. Selective information routing by ventral hippocampal CA1 projection neurons. *Science* 348, 560–563.

Csicsvari, J., Henze, D.A., Jamieson, B., Harris, K.D., Sirota, A., Barthó, P., Wise, K.D., and Buzsáki, G. (2003). Massively parallel recording of unit and local field potentials with silicon-based electrodes. *J. Neurophysiol.* 90, 1314–1323.

Csicsvari, J., Hirase, H., Czurkó, A., Mamiya, A., and Buzsáki, G. (1999). Fast network oscillations in the hippocampal CA1 region of the behaving rat. *J. Neurosci.* 19, RC20, RC20.

English, D.F., McKenzie, S., Evans, T., Kim, K., Yoon, E., and Buzsáki, G. (2017). Pyramidal cell-interneuron circuit architecture and dynamics in hippocampal networks. *Neuron* 96, 505–520.e7.

Fosque, B.F., Sun, Y., Dana, H., Yang, C.-T., Ohyama, T., Tadross, M.R., Patel, R., Zlatic, M., Kim, D.S., Ahrens, M.B., et al. (2015). Neural circuits: labeling of active neural circuits in vivo with designed calcium integrators. *Science* 347, 755–760.

Fujisawa, S., Amarasingham, A., Harrison, M.T., and Buzsáki, G. (2008). Behavior-dependent short-term assembly dynamics in the medial prefrontal cortex. *Nat. Neurosci.* 11, 823–833.

Gouwens, N.W., Sorensen, S.A., Berg, J., Lee, C., Jarsky, T., Ting, J., Sunkin, S.M., Feng, D., Anastassiou, C.A., Barkan, E., et al. (2019). Classification of electrophysiological and morphological neuron types in the mouse visual cortex. *Nat. Neurosci.* 22, 1182–1195.

Grosmark, A.D., and Buzsáki, G. (2016). Diversity in neural firing dynamics supports both rigid and learned hippocampal sequences. *Science* 351, 1440–1443.

Harris, K.D., Henze, D.A., Csicsvari, J., Hirase, H., and Buzsáki, G. (2000). Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. *J. Neurophysiol.* 84, 401–414.

Hazan, L., Zugaro, M., and Buzsáki, G. (2006). Klusters, NeuroScope, NDManager: a free software suite for neurophysiological data processing and visualization. *J. Neurosci. Methods* 155, 207–216.



- Jia, X., Siegle, J.H., Bennett, C., Gale, S.D., Denman, D.J., Koch, C., and Olsen, S.R. (2019). High-density extracellular probes reveal dendritic backpropagation and facilitate neuron classification. *J. Neurophysiol.* 121, 1831–1847.
- Kepecs, A., and Fishell, G. (2014). Interneuron cell types are fit to function. *Nature* 505, 318–326.
- Klausberger, T., and Somogyi, P. (2008). Neuronal diversity and temporal dynamics: the unity of hippocampal circuit operations. *Science* 321, 53–57.
- Lima, S.Q., Hromádka, T., Znamenskiy, P., and Zador, A.M. (2009). PINP: a new method of tagging neuronal populations for identification during in vivo electrophysiological recording. *PLoS One* 4, e6099.
- Martone, M., Gerkin, R., Moucek, R., Das, S., Goscinski, W., Hellgren-Kotaleski, J., Kennedy, D., Leergaard, T., Boline, J., and Abrams, M. (2020). NIX—neuroscience information exchange format. *F1000Res.* 9, 358. <https://doi.org/10.7490/f1000research.1117858.1>.
- McBain, C.J., and Fisahn, A. (2001). Interneurons unbound. *Nat. Rev. Neurosci.* 2, 11–23.
- McGinley, M.J., David, S.V., and McCormick, D.A. (2015). Cortical membrane potential signature of optimal states for sensory signal detection. *Neuron* 87, 179–192.
- Mizuseki, K., Sirota, A., Pastalkova, E., and Buzsáki, G. (2009). Theta oscillations provide temporal windows for local circuit computation in the entorhinal-hippocampal loop. *Neuron* 64, 267–280.
- Mizuseki, K., Diba, K., Pastalkova, E., and Buzsáki, G. (2011). Hippocampal CA1 pyramidal cells form functionally distinct sublayers. *Nat. Neurosci.* 14, 1174–1181.
- Nasiotis, K., Cousineau, M., Tadel, F., Peyrache, A., Leahy, R.M., Pack, C.C., and Baillet, S. (2019). Integrated open-source software for multiscale electrophysiology. *Sci. Data* 6, 231.
- Neto, J.P., Lopes, G., Frazão, J., Nogueira, J., Lacerda, P., Baião, P., Aarts, A., Andrei, A., Musa, S., Fortunato, E., et al. (2016). Validating silicon polytrodes with paired juxtacellular recordings: method and dataset. *J. Neurophysiol.* 116, 892–903.
- Nicolelis, M.A.L., and Lebedev, M.A. (2009). Principles of neural ensemble physiology underlying the operation of brain-machine interfaces. *Nat. Rev. Neurosci.* 10, 530–540.
- Okun, M., Steinmetz, N., Cossell, L., Iacuruso, M.F., Ko, H., Barthó, P., Moore, T., Hofer, S.B., Mircs-Flogel, T.D., Carandini, M., and Harris, K.D. (2015). Diverse coupling of neurons to populations in sensory cortex. *Nature* 521, 511–515.
- Oostenveld, R., Fries, P., Maris, E., and Schoffelen, J.-M. (2011). FieldTrip: Open source software for advanced analysis of MEG, EEG, and invasive electrophysiological data. *Comput. Intell. Neurosci.* 2011, e156869.
- Pachitariu, M., Steinmetz, N.A., Kadir, S.N., Carandini, M., and Harris, K.D. (2016). Fast and accurate spike sorting of high-channel count probes with KiloSort. In *Advances in Neural Information Processing Systems* 29, D.D. Lee, M. Sugiyama, U.V. Luxburg, I. Guyon, and R. Garnett, eds. (Curran Associates), pp. 4448–4456.
- Petersen, P.C., and Berg, R.W. (2016). Lognormal firing rate distribution reveals prominent fluctuation-driven regime in spinal motor networks. *eLife* 5, e18805.
- Petersen, P.C., and Buzsáki, G. (2020). Cooling of medial septum reveals theta phase lag coordination of hippocampal cell assemblies. *Neuron* 107, 731–744.e3.
- Petersen, P.C., Hernandez, M., and Buzsáki, G. (2018). Public electrophysiological datasets collected in the Buzsáki Lab, or. <http://zenodo.org/record/3629881>.
- Petersen, P.C., Hernandez, M., and Buzsáki, G. (2020). The Buzsáki Lab Databank—public electrophysiological datasets from awake animals. <https://buzsakilab.com/wp/public-data/>.
- Peyrache, A., Lacroix, M.M., Petersen, P.C., and Buzsáki, G. (2015). Internally organized mechanisms of the head direction sense. *Nat. Neurosci.* 18, 569–575.
- Quiroga, R.Q., Nadasdy, Z., and Ben-Shaul, Y. (2004). Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. *Neural Comput.* 16, 1661–1687.
- Roux, L., and Buzsáki, G. (2015). Tasks for inhibitory interneurons in intact brain circuits. *Neuropharmacology* 88, 10–23.
- Royer, S., Zemelman, B.V., Losonczy, A., Kim, J., Chance, F., Magee, J.C., and Buzsáki, G. (2012). Control of timing, rate and bursts of hippocampal place cells by dendritic and somatic inhibition. *Nat. Neurosci.* 15, 769–775.
- Rübel, O., Tritt, A., Dichter, B., Braun, T., Cain, N., Clack, N., Davidson, T.J., Dougherty, M., Fillion-Robin, J.-C., Graddis, N., et al. (2019). NWB:N 2.0: an accessible data standard for neurophysiology. *BioRxiv*. <https://doi.org/10.1101/523035>.
- Rudy, B., Fishell, G., Lee, S., and Hjerling-Leffler, J. (2011). Three groups of interneurons account for nearly 100% of neocortical GABAergic neurons. *Dev. Neurobiol.* 71, 45–61.
- Sanchez-Aguilera, A., Wheeler, D.W., Jurado-Parras, T., Valero, M., Nokia, M.S., Cid, E., Fernandez-Lamo, I., Sutton, N., García-Rincón, D., de la Prida, L.M., and Ascoli, G.A. (2021). An update to hippocampome.org by integrating single-cell phenotypes with circuit function in vivo. *PLoS Biol.* 19, e3001213.
- Schmitzer-Torbert, N., Jackson, J., Henze, D., Harris, K., and Redish, A.D. (2005). Quantitative measures of cluster quality for use in extracellular recordings. *Neuroscience* 131, 1–11.
- Sejnowski, T.J., Churchland, P.S., and Movshon, J.A. (2014). Putting big data to good use in neuroscience. *Nat. Neurosci.* 17, 1440–1441.
- Senzai, Y., and Buzsáki, G. (2017). Physiological properties and behavioral correlates of hippocampal granule cells and mossy cells. *Neuron* 93, 691–704.e5.
- Senzai, Y., Fernandez-Ruiz, A., and Buzsáki, G. (2019). Layer-specific physiological features and interlaminar interactions in the primary visual cortex of the mouse. *Neuron* 101, 500–513.e5.
- Shamash, P., Carandini, M., Harris, K., and Steinmetz, N. (2018). A tool for analyzing electrode tracks from slice histology. *BioRxiv*. <https://doi.org/10.1101/447995>.
- Siegle, J.H., Jia, X., Durand, S., Gale, S., Bennett, C., Graddis, N., Heller, G., Ramirez, T.K., Choi, H., Luviano, J.A., et al. (2021). Survey of spiking in the mouse visual system reveals functional hierarchy. *Nature* 592, 86–92.
- Sirota, A., Montgomery, S., Fujisawa, S., Isomura, Y., Zugaro, M., and Buzsáki, G. (2008). Entrainment of neocortical neurons and gamma oscillations by the hippocampal theta rhythm. *Neuron* 60, 683–697.
- Stark, E., Koos, T., and Buzsáki, G. (2012). Diode probes for spatiotemporal optical control of multiple neurons in freely moving animals. *J. Neurophysiol.* 108, 349–363.
- Stark, E., Eichler, R., Roux, L., Fujisawa, S., Rotstein, H.G., and Buzsáki, G. (2013). Inhibition-induced theta resonance in cortical circuits. *Neuron* 80, 1263–1276.
- Steinmetz, N.A., Zátka-Haas, P., Carandini, M., and Harris, K.D. (2019). Distributed coding of choice, action and engagement across the mouse brain. *Nature* 576, 266–273.
- Stringer, C., Pachitariu, M., Steinmetz, N., Reddy, C.B., Carandini, M., and Harris, K.D. (2019). Spontaneous behaviors drive multidimensional, brainwide activity. *Science* 364, 255.
- Tadel, F., Baillet, S., Mosher, J.C., Pantazis, D., and Leahy, R.M. (2011). Brainstorm: a user-friendly application for MEG/EEG analysis. *Comput. Intell. Neurosci.* 2011, 879716.
- Tecuati, C., Wheeler, D.W., and Ascoli, G.A. (2021). A method for estimating the potential synaptic connections between axons and dendrites from 2D neuronal images. *Bio Protoc.* 11, e4073.
- Teeters, J.L., Godfrey, K., Young, R., Dang, C., Friedsam, C., Wark, B., Asari, H., Peron, S., Li, N., Peyrache, A., et al. (2015). Neurodata without borders: creating a common data format for neurophysiology. *Neuron* 88, 629–634.
- van der Maaten, L., and Hinton, G. (2008). Visualizing data using t-SNE. *J. Mach. Learn. Res.* 9, 2579–2605.

Wang, Q., Ding, S.-L., Li, Y., Royall, J., Feng, D., Lesnar, P., Graddis, N., Naeemi, M., Facer, B., Ho, A., et al. (2020). The Allen mouse brain common coordinate framework: a 3D reference atlas. *Cell* **181**, 936–953.e20.

Wheeler, D.W., White, C.M., Rees, C.L., Komendantov, A.O., Hamilton, D.J., and Ascoli, G.A. (2015). Hippocampome.org: a knowledge base of neuron types in the rodent hippocampus. *eLife* **4**, e09960.

Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.W., da Silva Santos, L.B., Bourne, P.E.,

et al. (2016). The FAIR guiding principles for scientific data management and stewardship. *Sci. Data* **3**, 160018.

Yger, P., Spampinato, G.L., Esposito, E., Lefebvre, B., Deny, S., Gardella, C., Stimberg, M., Jetter, F., Zeck, G., Picaud, S., et al. (2018). A spike sorting toolbox for up to thousands of electrodes validated with ground truth recordings in vitro and in vivo. *eLife* **7**, e34518.

Zhang, S.-J., Ye, J., Miao, C., Tsao, A., Cerniauskas, I., Ledergerber, D., Moser, M.-B., and Moser, E.I. (2013). Optogenetic dissection of entorhinal-hippocampal functional connectivity. *Science* **340**, 1232627.

## STAR★METHODS

### KEY RESOURCES TABLE

| REAGENT or RESOURCE     | SOURCE                     | IDENTIFIER  |
|-------------------------|----------------------------|---|
| Deposited data          |                            |   |
| Hippocampal dataset     | Petersen and Buzsáki, 2020 | <a href="https://buzsakilab.com/wp/projects/entry/4919/">https://buzsakilab.com/wp/projects/entry/4919/</a>   |
| Visual cortex dataset   | Senzai et al., 2019        | <a href="https://buzsakilab.com/wp/projects/entry/22682/">https://buzsakilab.com/wp/projects/entry/22682/</a>   |
| UCL dataset             | Steinmetz et al., 2019     | <a href="https://figshare.com/articles/dataset/Eight-probe_Neuropixels_recordings_during_spontaneous_behaviors/7739750">https://figshare.com/articles/dataset/Eight-probe_Neuropixels_recordings_during_spontaneous_behaviors/7739750</a> |
|                         |                            | <a href="https://buzsakilab.com/wp/projects/entry/52347/">https://buzsakilab.com/wp/projects/entry/52347/</a>   |
| Allen Institute dataset | Siegle et al., 2021        | <a href="https://allensdk.readthedocs.io/en/latest/visual_coding_neuropixels.html">https://allensdk.readthedocs.io/en/latest/visual_coding_neuropixels.html</a>   |
|                         |                            | <a href="https://buzsakilab.com/wp/projects/entry/52635/">https://buzsakilab.com/wp/projects/entry/52635/</a>   |
| Software and algorithms |                            |   |
| CellExplorer            | Petersen and Buzsáki, 2020 | <a href="https://cellexplorer.org">https://cellexplorer.org</a>   |
| MATLAB                  | MathWorks                  | <a href="https://www.mathworks.com">https://www.mathworks.com</a>   |

### RESOURCE AVAILABILITY

#### Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, György Buzsáki ([gyorgy.buzsaki@nyulangone.org](mailto:gyorgy.buzsaki@nyulangone.org)).

#### Materials availability

This study did not generate new unique reagents.

#### Data and code availability

All data are available from our databank at <https://buzsakilab.com/wp/database/> (Petersen et al., 2018). All code is available at GitHub: <https://github.com/petersenpeter/CellExplorer>. All resources, including tutorials and documentation are on the CellExplorer website at: <https://cellexplorer.org/>.

### EXPERIMENTAL MODEL AND SUBJECT DETAILS

All experimental data were collected in previous studies. Details of experimental methods and subject information are available in the original studies (Petersen and Buzsáki, 2020; Senzai et al., 2019; Siegle et al., 2021; Steinmetz et al., 2019).

### METHOD DETAILS

#### CellExplorer layout and preferences saved between sessions

The display preferences are saved between sessions, which provides a simpler interface for novice users, while maintaining a more advanced display for experienced users. The preferences can be reset from the view menu by clicking Reset Layout/Preferences, or by deleting the preference file `last_preferences_CellExplorer.mat` located in the folder `calc_CellMetrics`.

#### General tutorial

This tutorial covers the processing, from generating the necessary session metadata using the template, running the processing pipeline, opening multiple sessions for manual curation in CellExplorer, and finally using the `cell_metrics` for filtering cells. The tutorial is also available as a MATLAB script: (`tutorials/CellExplorer_Tutorial.m`).

1. Define the basepath of the dataset to process. The dataset should ideally consist of the raw data `basename.dat` and spike sorted data.  
`basepath = 'your/data/path/basename/';`  
`cd(basepath)`

2. Generate the session metadata struct using the template script and display the metadata in the session GUI ([Figure S2](#))  
`session = sessionTemplate(basepath, 'showGUI', true);` You can use the GUI to inspect and manually add metadata. Make sure the extracellular tab is filled out correctly for your data ([Figure S2C](#)). The template script can extract existing metadata from a NeuroScope compatible basename.xml file, from intan's info.rhd, from KiloSort's rez.mat file and from a basename.sessionInfo.mat file (buzcode).
3. Run the cell metrics pipeline `ProcessCellMetrics` using the session struct as input `cell_metrics = ProcessCellMetrics('session', session, 'showGUI', true);` Setting `showGUI` to `true` will display the session GUI with a *CellExplorer* tab allowing you to verify parameters and settings (screenshot shown in [Figure S2D](#)). You can click the button *Verify metadata* to show a summary table with metadata relevant to the processing. Fields requiring your attention will be highlighted in red; optional fields in blue.
4. Visualize the cell metrics in *CellExplorer*  
`cell_metrics = CellExplorer('metrics', cell_metrics);`
5. You can repeat step 1-4 on a couple of datasets and load them together in *CellExplorer*, providing several paths  
`basepaths = {'path/to/session1', 'path/to/session2'};`  
`basenames = {'session1', 'session2'};`  
`cell_metrics = loadCellMetricsBatch('basepaths', basepaths, 'basenames', basenames);`  
`cell_metrics = CellExplorer('metrics', cell_metrics);`
6. Curate your cells in *CellExplorer* and save the metrics via the file menu in *CellExplorer*.
7. You may use the script *loadCellMetrics* for further analysis using the metrics as filters:
  1. Get cells labeled as *Interneuron*  
`cell_metrics_idx1 = loadCellMetrics('cell_metrics', cell_metrics, 'putativeCellType', {'Interneuron'});`
  2. Get cells that have the groundTruthClassification label *Axoaxonic*  
`cell_metrics_idx2 = loadCellMetrics('cell_metrics', cell_metrics, 'groundTruthClassification', {'Axoaxonic'});`

### Expandability tutorial: add you own custom metrics

This tutorial covers how to add your custom cell metrics. For single value metrics you have two options: numeric values or string arrays. Numeric metrics can be plotted in the custom group plot in *CellExplorer*. String arrays allow you to group your data by the unique strings set within features, and can be plotted in discrete values. All features in the cell metrics are automatically available in *CellExplorer* if they contain N values (N: number of cells).

### Add a string metric to your cell\_metrics

Let's say you want to add a cell metric describing cortical layers for each cell, using predefined labels (Layer 1 to Layer 6). This can be stored as a char cell array, e.g.:

```
cell_metrics.corticalLayer = {'layer 5', 'layer 4', 'layer 2', 'layer 2/3', 'layer 1'}; % nCells = 5
```

### Add numeric values to your cell metrics

Let's say you want to add the preferred orientation of a drifting grating presented to cells in the visual cortex. This will be stored as numeric values, e.g.:

```
cell_metrics.pref_ori_dg = [90, 25, 45, 80, 30]; % nCells = 5
```

Now, load the cell metrics into *CellExplorer* to visualize them. The fields will appear in the drop-down menus in the custom group plot:

```
cell_metrics = CellExplorer('metrics', cell_metrics);
```

If you open multiple sessions in *CellExplorer*, the custom metrics will automatically be imported. Cells without numeric values will have NaN values assigned and empty strings for missing char fields.

You can also incorporate response curves and other more advanced metrics, and perform custom calculations in the *ProcessCellMetrics* script by using the custom calculation implementation: <https://cellexplorer.org/pipeline/custom-calculations/>.

For further plotting options, please see the website: <https://cellexplorer.org/datastructure/expandability/>.

**Neuron, Volume 109**

**Supplemental information**

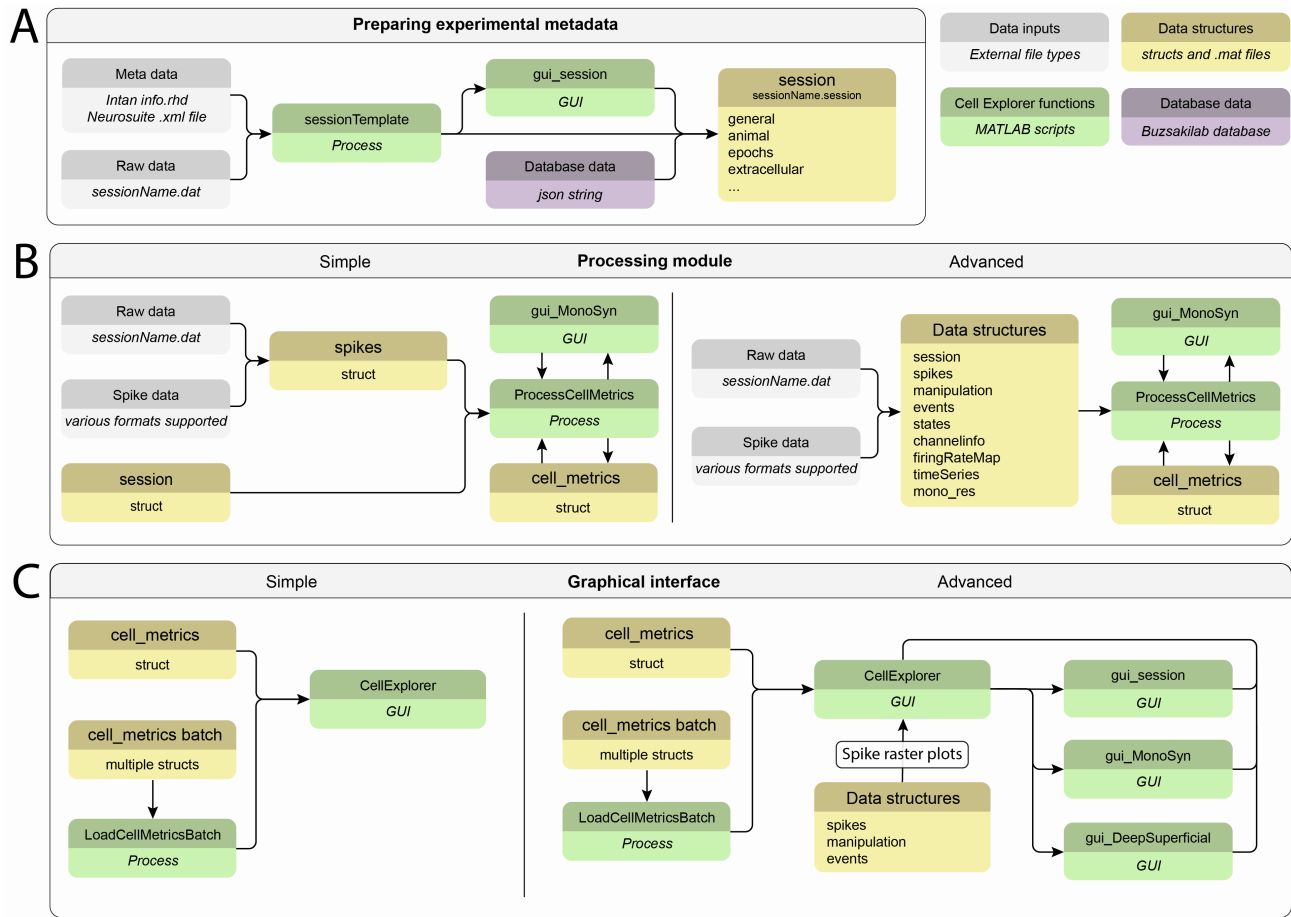
**CellExplorer: A framework for visualizing  
and characterizing single neurons**

**Peter C. Petersen, Joshua H. Siegle, Nicholas A. Steinmetz, Sara Mahallati, and György Buzsáki**

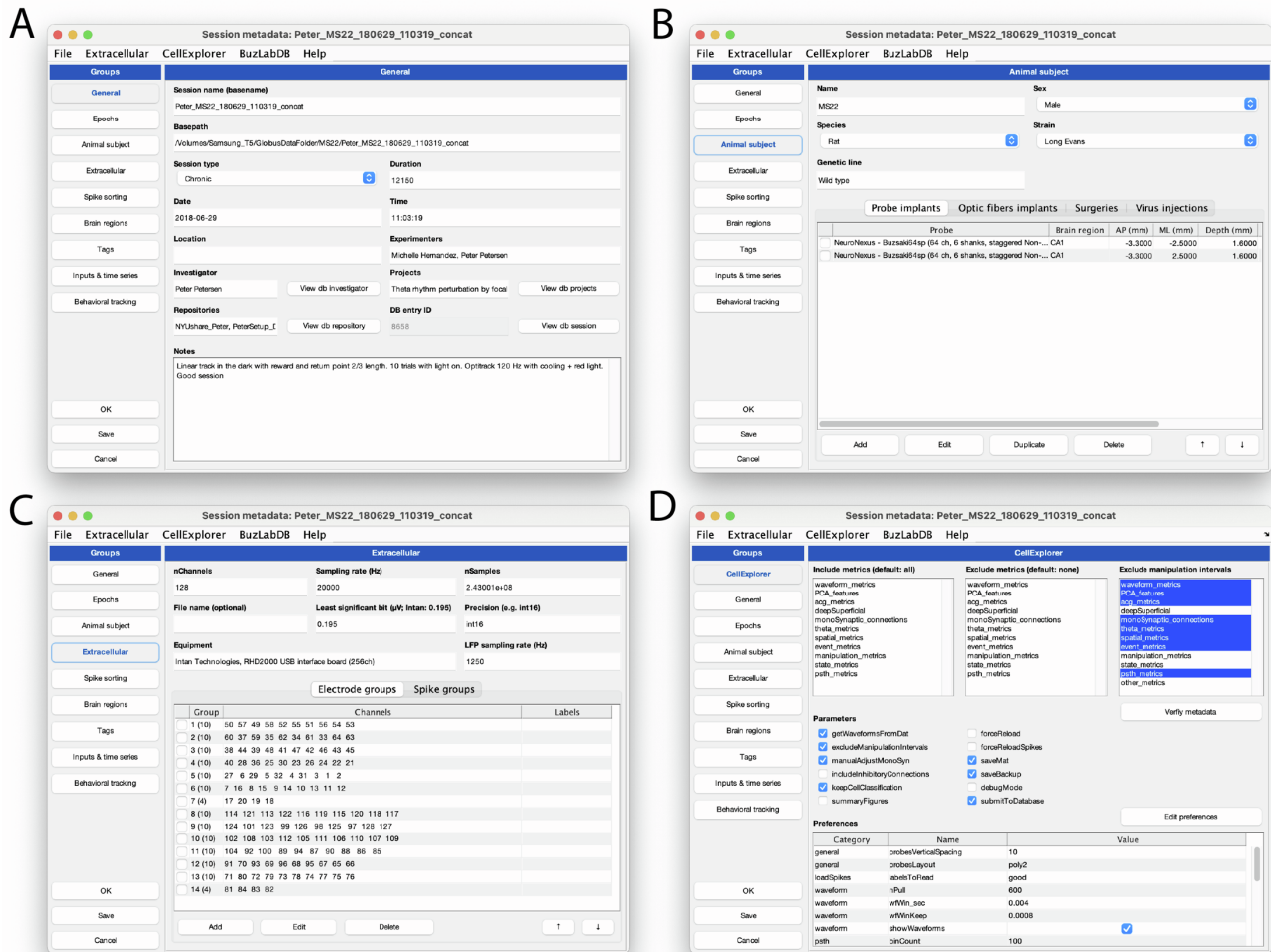


# SUPPLEMENTARY MATERIAL

## Supplementary Figures and Tables

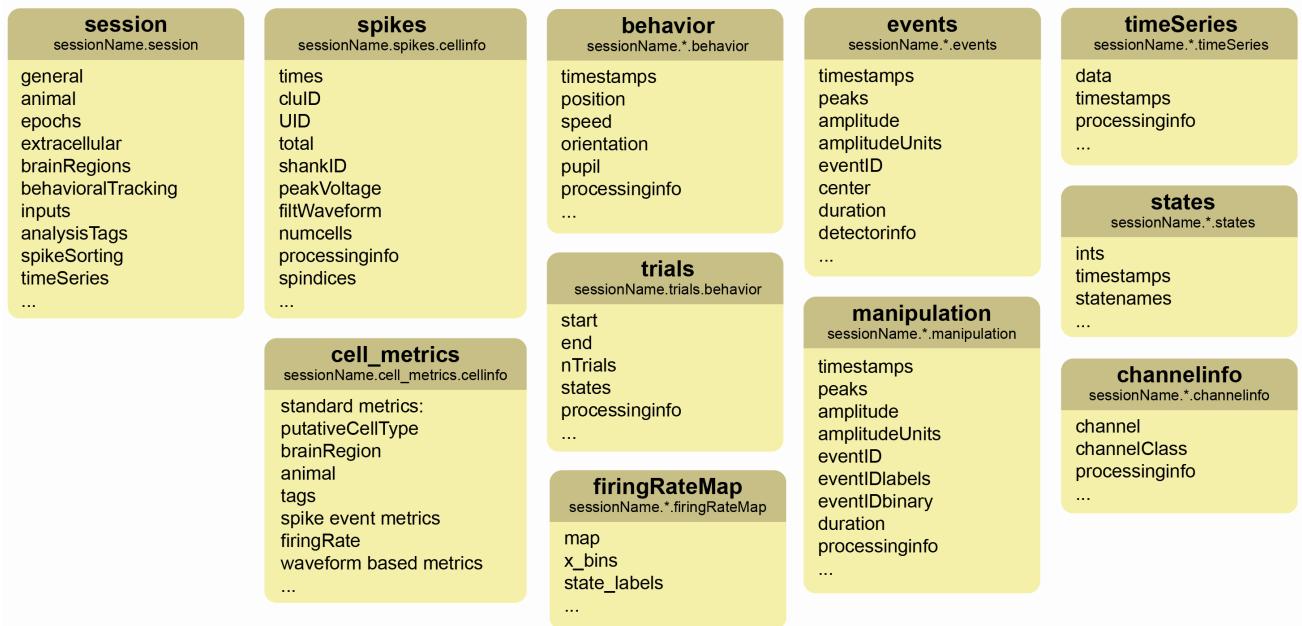


**Supplementary Figure 1. Flow charts, related to figure 2. A)** Generating the metadata structure for a recording session. **B)** Running the processing pipeline. **C)** Running the CellExplorer module for manual curation and exploration. CellExplorer data structures are shown in yellow, MATLAB functions in green, and the input data in grey. Input from the Buzsáki lab database is shown in purple (Petersen et al., 2020).

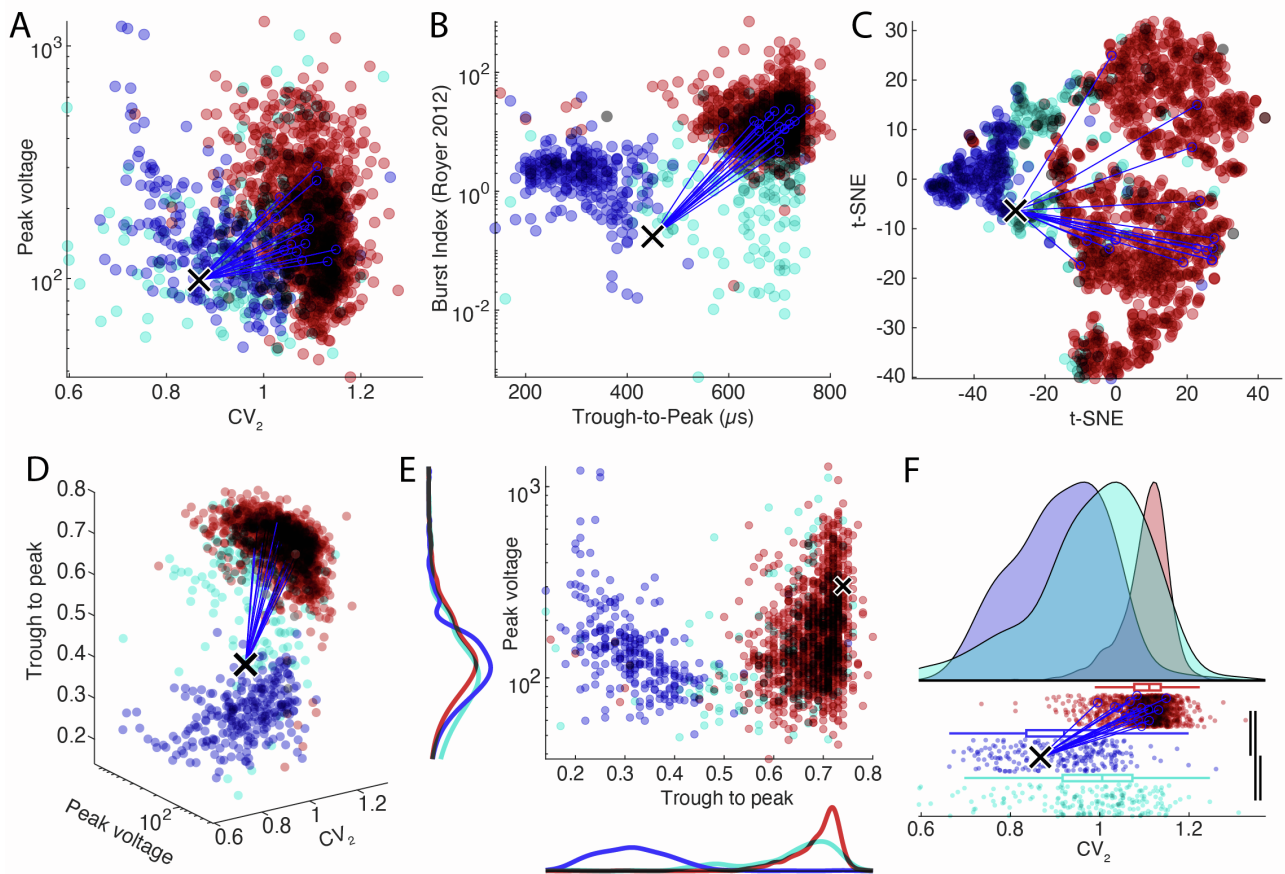


**Supplementary Figure 2: Session metadata GUI, related to figure 2.** The graphical interface for inspection and entry of session level metadata follows the organization of the Matlab struct, with a tab for each field type. **A.** General information about the session, including name, data, duration, location, and notes. **B.** Animal metadata capturing sex, species, strain and genetic line, but also action performed on the animal including probe implants, optic fiber implants, surgeries, and virus injection. **C.** Basic metadata for the extracellular data, including channel count, sampling rate, equipment and electrode groups. **D.** The session GUI is also used as a graphical interface for the processing pipeline (with a dedicated tab), allowing the user to change parameters, view settings, validate metrics, and see and edit the full session metadata structure, that serves as input to the pipeline.

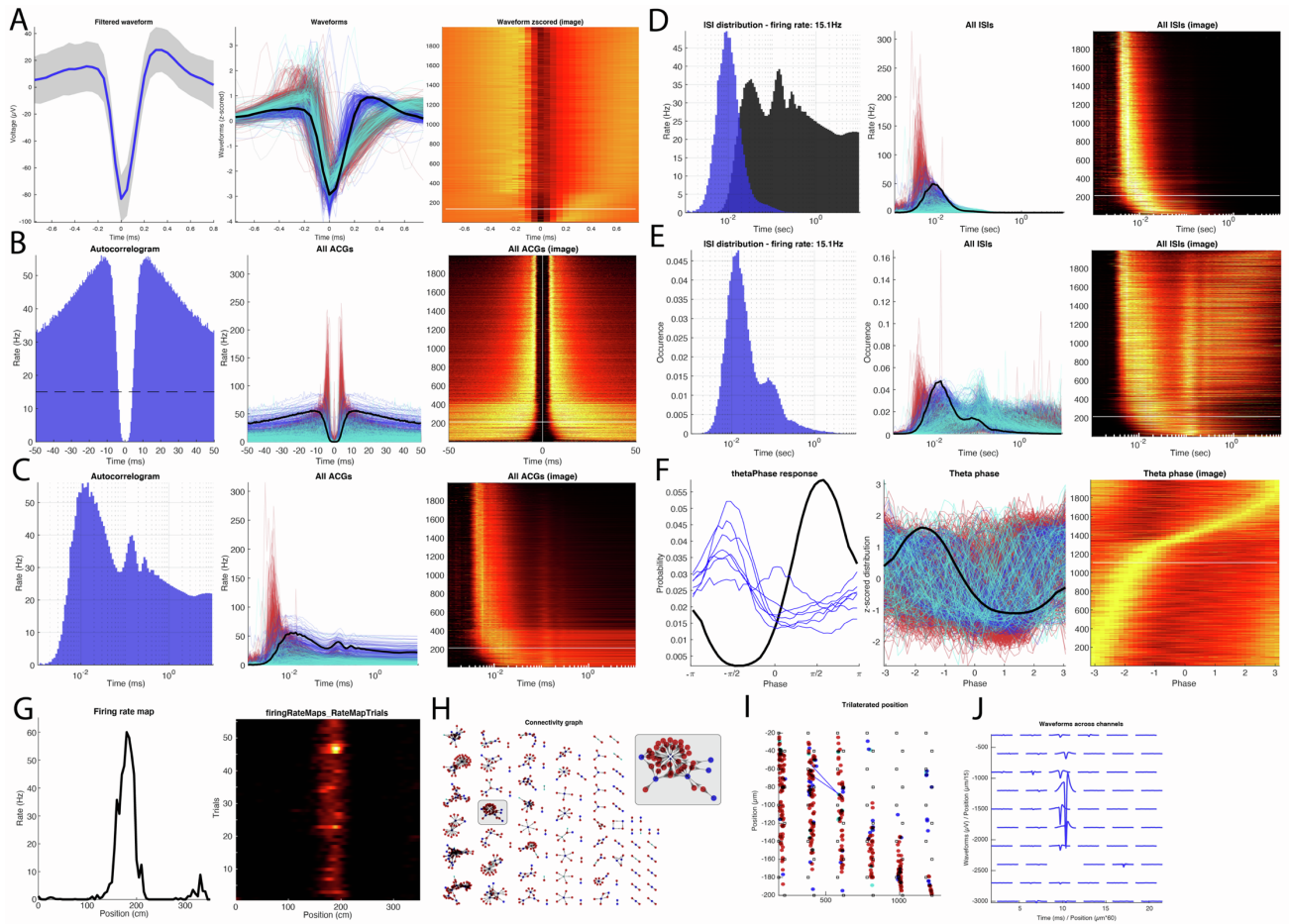
## CellExplorer data structure



**Supplementary Figure 3: Datatypes related to figure 2.** The data structure. A detailed description is available online at [CellExplorer.org/datastructure/data-structure-and-format](https://CellExplorer.org/datastructure/data-structure-and-format). session, spikes, cell\_metrics, trials are defined data types, while behavior, firingRateMap, events, manipulation, timeseries, states, channelinfo are data containers.

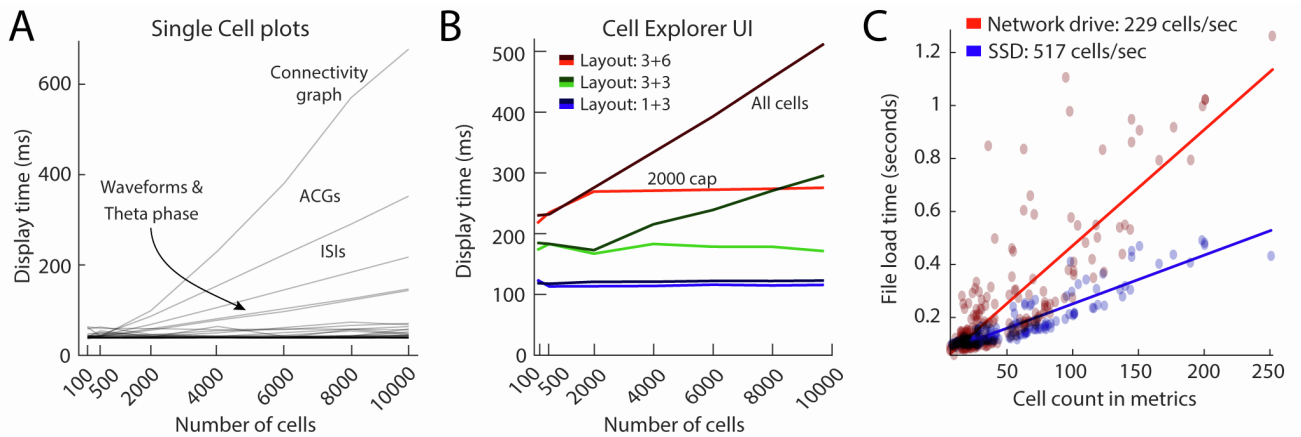


**Supplementary Figure 4. Population data plots, related to figure 3.** Top row: The three standard representations: custom plot (A), classic representation (B), and t-SNE plot (C). Bottom row: The custom plot has 3 further data representations: a 3-dimensional plot with custom marker size (D), 2D plot with marginal histograms (E), and one-dimensional raincloud plots (F), combining 1D scattered neurons with error bars histogram and KS significance test (line thickness represent significance levels). Color-coded according to cell types: pyramidal cell (red), narrow interneuron (blue), wide interneuron (cyan).

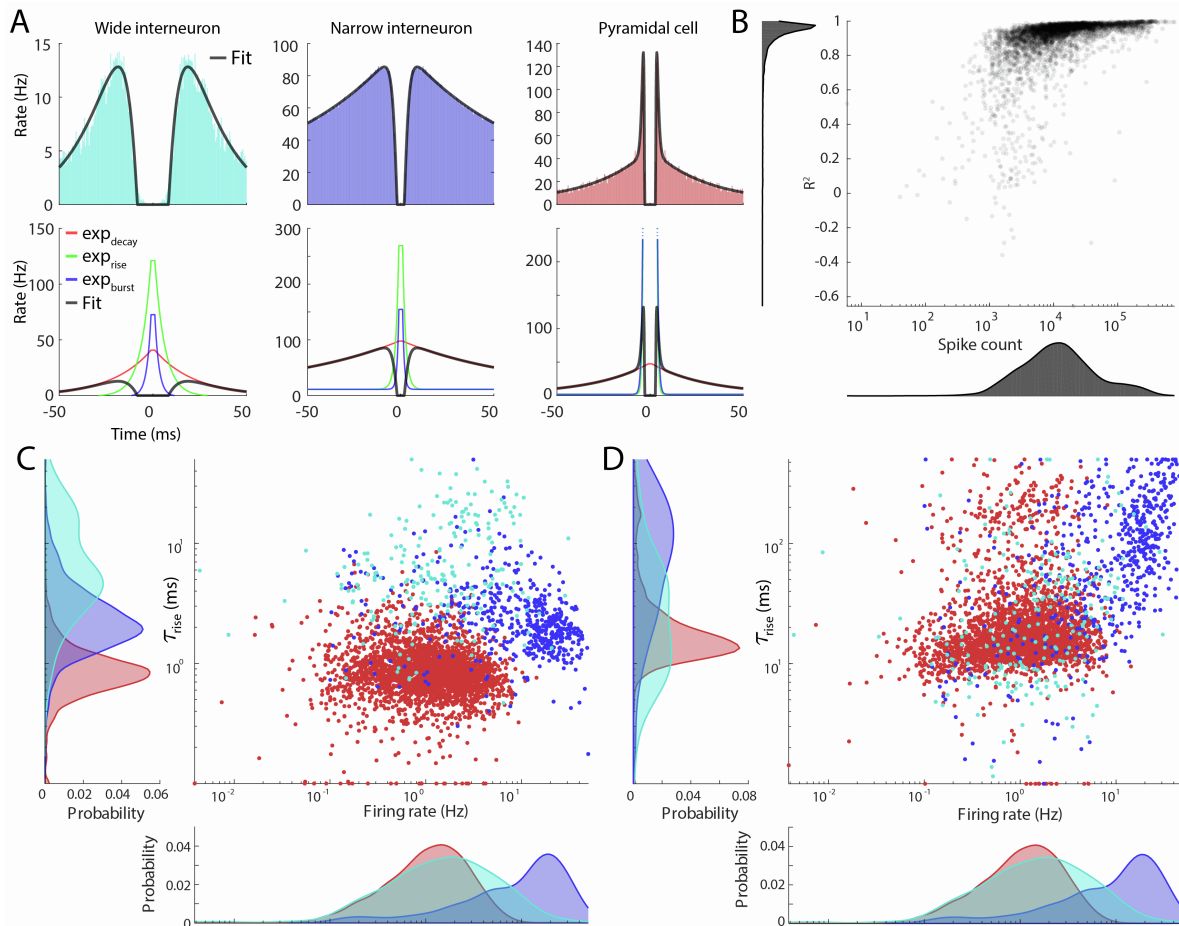


**Supplementary Figure 5. The various single-cell plots, related to figure 3.** Most single cell data-visualizer have three representations: single neurons (with neuronal connections highlighted for a subset of the plots), all neurons (absolute or normalized representations), and an image representation (normalized data, with selected cell highlighted by a white line). **A.** Waveform representations: waveform of a selected single neuron, waveforms of all neurons (z-scored), and their image representation. The white line in the image representation corresponds to the selected neuron. **B.** Autocorrelograms (ACGs) for the single neuron, ACGs for all neurons and their image representation. **C.** ACGs on a log scale (single, all, image). **D, E.** Interspike interval distributions (ISIs) on a log scale (single, all image) for two different normalizations (**D**, rate (Hz); **E**, occurrence). **F.** Theta phase spike histogram for the single interneuron (black line) and those of pyramidal neurons monosynaptically connected to the interneurons (blue lines; left) and all neurons in the same session (middle and right panels). **G.** Firing rate map for a pyramidal cell. Session average (left) and trial-wise heatmap. **H.** Connectivity graph showing all monosynaptic modules in the dataset. A module is highlighted and enhanced (top right). **I.** Physical location of neurons recorded in the same animal using trilateration. Eight-shank silicon probe recording (8 sites on each shank). Red, pyramidal cells. Blue, interneurons. Monosynaptic connections between two pyramidal cells and a target interneuron are also shown (blue lines) **J.** Average waveform across channels of the single interneurons shown in most panels. A-F, H-J: a narrow interneuron, G: Spatial firing rate of a pyramidal cell on a linear track.

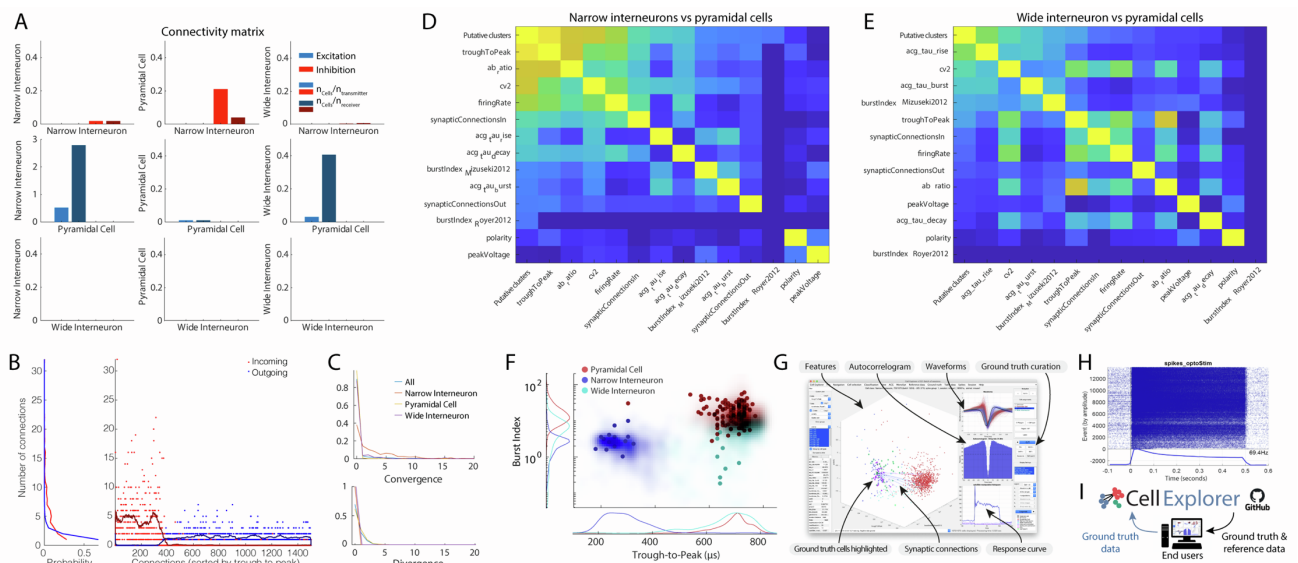




**Supplementary figure 6. Benchmarks of the CellExplorer user interface (UI) related to figure 3.** **A.** Display times for single-cell plots, quantified by the number of cells displayed. The slowest plots are the ones with a trace for each cell (ACGs, ISIs, waveforms, ISIs, theta phase) and the connectivity graph. By default, a maximum of 2000 traces are drawn capping the processing time below ~80 ms for all plots except the connectivity graph for which all connections are shown. **B.** UI display times when switching between units for the three layouts shown in figure 3B (approximately 110 ms for layout 1+3 with 4 subplots; blue lines. 180 ms for layout 3+3 with 6 subplots; green lines) and 290 ms (layout 3+6 with 9 subplots; in red), respectively. Dark gradient colored lines (dark red, green, and blue) indicate where there were no limits on the number of traces plotted for single-cell plots, and the light gradient lines show display times with a maximum of 2000 random traces. **C.** Benchmarks of cell metrics file loading time. On average, 230 cells can be loaded per second quantified across 180 sessions with various cell count (red dots and linear fit in red). By storing the data on a local SSD, the loading time can be decreased and attain cell loading above 500 cells per second. Graphical benchmarks were performed on an iMac from 2017 with a 4.2GHz Quad-Core Intel i7 with 32GB of ram. File load time tests were performed on a custom PC running Window 10, with a 512GB Samsung 870 EVO SSD (SSD) and the NYU Langone Health network storage solution “Research Isilon” (Network drive).



**Supplementary Figure 7. ACG fits related to figure 5.** **A.** Three examples of typical autocorrelograms for a wide interneuron (left column) narrow interneuron (middle column) and a pyramidal cell (right column). The exponential components are plotted in the lower row. **B.** The  $R^2$  values for each fit across the 4000 cells plotted against the number of spikes. **C-D**  $\tau_{\text{rise}}$  (C) and  $\tau_{\text{decay}}$  (D) values plotted against the firing rate. Color coded by putative cell type.



**Supplementary Figure 8. Cell type separation, ground truth- and reference data related to figure 5.**

**A.** Connectivity by cell types. Each panel is a projection pattern showing connection from one cell type to another, both excitatory (blue) and inhibitory (red). The two bars in the same color are normalized by the transmitter and receiver population count. **B.** Every synaptic connection is sorted by the spike waveform trough-to-peak, showing a clear separation between which cells transmit and receive on the basis of spike waveform features. **C.** Convergence and divergence by cell types. **D-E.** Correlation between putative clusters and various metrics. **D:** Narrow interneuron vs pyramidal cells, **E:** Wide interneurons vs pyramidal cells. **F.** Single session (dots) data compared with data from 30 reference sessions (shaded zones). **G.** Opto-tagged data can be processed and curated directly in CellExplorer. **H.** Example of a PSTH of a PV-expressing neuron to 500 ms square light pulses. Raster plot and average responses to the light pulses are visualized in CellExplorer. **I.** The CellExplorer framework allows for sharing ground truth and reference data directly with the end-user. End users can upload their ground truth data to the CellExplorer GitHub repository for communal sharing (see the opto-tagging tutorial at [CellExplorer.org](http://CellExplorer.org)).

| Metrics                              | Description/Calculation   | Type                                |
|--------------------------------------|---|-------------------------------------|
| <b>General metrics</b>               |   |                                     |
| general                              | struct containing general information about the session   | struct                              |
| .basename                            | the name of the session   | char                                |
| .basepath                            | the path to the raw data  | char                                |
| .cellCount                           | number of cells in the current session  | double                              |
| .cgg                                 | cross correlogram matrix between cell pairs within a session  | 201xNxN double                      |
| .cgg_time                            | time vector describing the time bins in the cgg (standard: -100ms:1ms:100ms)                                | 201x1 double                        |
| animal (name)                        | Name of animal subject  | 1xN cell array of charactor vectors |
| general.animal                       | struct containing animal specific information   | struct                              |
| .sex                                 | Sex of the animal [Male, Female, Unknown]   | char                                |
| .species                             | Animal species [Rat, Mouse,...]   | char                                |
| .strain                              | Animal strain [Long Evans, C57B1/6,...]   | char                                |
| .geneticLine                         | Genetic line of the animal  | char                                |
| sessionName                          | Name of session   | 1xN cell array of charactor vectors |
| general.session                      | struct containing session specific information  | struct                              |
| .sessionType                         | [Acute, Chronic]  | 1xN cell array of charactor vectors |
| .spikeSortingMethod                  |   | char                                |
| .investigator                        | Name of the investigator  | char                                |
| general.processinginfo               | Struct containing processing info: date of the processing, version of the script, function name...          |                                     |
| .params                              | Struct containing the input parameters used by ProcessCellMetrics   |                                     |
| UID                                  | The ID for each cell unique within a session (1:nCells)   | 1xN double                          |
| cellID                               |   | 1xN double                          |
| clulID                               | clustering ID from spike sorting pipeline   | 1xN double                          |
| batchIDs                             | only present in batch sessions. The batch ids the cells   | 1xN double                          |
| putativeCellType                     | Putative cell type  | 1xN cell array of charactor vectors |
| brainRegion                          | Brain region acronyms from Allan institute Brain atlas.   | 1xN cell array of charactor vectors |
| shankID                              | Shank number / electrode group  | 1xN double                          |
| labels                               | Custom labels   | 1xN cell array of charactor vectors |
| groups                               | struct containing groups  | struct                              |
| tags                                 | struct containing tags  | struct                              |
| <b>Spike event-based metrics</b>     |   |                                     |
| spikes.times                         | struct containing spike times   |                                     |
| spikeCount                           | Spike count of the cell from the entire session   | 1xN double                          |
| firingRate                           | Firing rate in Hz: Spike count normalized by the interval between the first and the last spike.             | 1xN double                          |
| cv2                                  | Coefficient of variation  | 1xN double                          |
| refractoryPeriodViolation            | Refractory period violation (‰): Fraction of ISIs less than 2ms.  | 1xN double                          |
| burstIndex_Mizuseki2012              | Burst index: Fraction of spikes with a neighboring ISI < 6ms as defined in Mizuseki et al. Hippocampus 2012 | 1xN double                          |
| <b>Waveform metrics</b>              |   |                                     |
| waveform                             | struct containing waveform information  | struct                              |
| .filt                                | Average filtered waveform from peak chanel (μV)   | 1xN cell array of numeric vectors   |
| .filt_std                            | Std of average filtered waveform (μV)   | 1xN cell array of numeric vectors   |
| .raw                                 | Average raw waveform from peak chanel (μV)  | 1xN cell array of numeric vectors   |
| .raw_std                             | Std of average raw waveform (μV)  | 1xN cell array of numeric vectors   |
| .time                                | Time vector (ms)  | 1xN cell array of numeric vectors   |
| maxWaveformCh                        | peak channel (0-indexed)  | 1xN double                          |
| maxWaveformCh1                       | peak channel (1-indexed)  | 1xN double                          |
| maxWaveformChannelOrder              | linearized channel position   |                                     |
| polarity                             | waveform polarity   |                                     |
| troughToPeak                         | waveform trough to peak interval (μs)   | 1xN double                          |
| ab_ratio                             | waveform peak to peak ratio   | 1xN double                          |
| peakVoltage                          | amplitude of the filtered waveform (μV). max(waveform)-min(waveform).                                       | 1xN double                          |
| troughToPeakDerivative               | derivative of waveform trough to peak interval (μs)   | 1xN double                          |
| <b>Autocorellogram (ACG) metrics</b> |   |                                     |
| acg                                  | struct containing autocorrelogram information   | struct                              |
| .wide                                | [-1000ms:1ms:1000ms]  | 1xN cell array of numeric vectors   |
| .narrow                              | [-50:0.5:50]  | 1xN cell array of numeric vectors   |
| .log10                               | [log-intervals spanning 1ms:10s]  | 1xN cell array of numeric vectors   |

|  |  |                                     |
|--|--|-------------------------------------|
| thetaModulationIndex                         | defined by the difference between the theta modulation trough (mean of autocorrelogram bins 50-70 ms) and the theta modulation peak (mean of autocorrelogram bins 100-140ms) over their sum.   | 1xN double                          |
| ACG fit metrics                              | Fit to the autocorrelogram with a triple-exponential equation ( fit = cexp(-x/τ_decay)-dexp(-x/τ_rise) )   | 1xN double                          |
| acg_asymptote                                | the asymptote of the ACG fit   | 1xN double                          |
| acg_c  | ACG fit: amplitude   | 1xN double                          |
| acg_d  | ACG fit: amplitude   | 1xN double                          |
| acg_fit_rsquare                              | ACG fit R-square (the goodness of the fit)   | 1xN double                          |
| acg_h  | ACG fit: amplitude   | 1xN double                          |
| acg_refrac                                   | ACG fit: refractory period (ms)  | 1xN double                          |
| acg_tau_burst                                | ACG fit: tau bursts (ms)   | 1xN double                          |
| acg_tau_decay                                | ACG fit: tau decay (ms)  | 1xN double                          |
| acg_tau_rise                                 | ACG fit tau rise (ms)  | 1xN double                          |
| burstIndex_Royer2012                         | Burst index (Royer 2012)   | 1xN double                          |
| burstIndex_Doublets                          | Burst index doublets   | 1xN double                          |
| <b>Interspike Intervals (ISI) metrics</b>    |  |                                     |
| isi  | struct with interspike interval information  | struct                              |
| .log10                                       | [log-intervals spanning 1ms:10s]   | 1xN cell array of numeric vectors   |
| <b>Putative connections</b>                  |  |                                     |
| putativeConnections                          | putative connections determined from cross correlograms  | struct                              |
| putativeConnections.excitatory               | Excitatory connection pairs  | 2xP double                          |
| putativeConnections.inhibitory               | Inhibitory connection pairs  | 2xP double                          |
| synapticEffect                               | Excitatory' or 'Inhibitory'  | 1xN cell array of charactor vectors |
| synapticConnectionsIn                        | Synatic connections count  | 1xN double                          |
| synapticConnectionsOut                       | Synatic connections count  | 1xN double                          |
| <b>Event metrics</b>                         |  |                                     |
| events                                       | event time series  | struct                              |
| .name  | the event curve  | 1xN cell array of numeric vectors   |
| _modulationIndex                             | modulation index for each event types  | 1xN double                          |
| _modulationSignificanceLevel                 | modulation significance level for each event types   |                                     |
| _modulationPeakResponseTime                  | modulation peak response time for each event types   | 1xN double                          |
| <b>Firing rate map metrics</b>               |  |                                     |
| firingRateMaps                               | struct with (spatial) linearized firing rate maps  | struct                              |
| .ratemap                                     | Primary firing rate map  | 1xN cell array of numeric vectors   |
| .ratemapName                                 | Other firing rate maps   | 1xN cell array of numeric vectors   |
| spatialCoverageIndex                         | Spatial coverage index. Defined from the inverse cumulative distribution, where bins are sorted by decreasing rate. The 75 percentile point defines the spatial coverage by the fraction of bins below and above the point (defined by Royer et al., NN 2012)                                      | 1xN double                          |
| spatialGiniCoeff                             | Spatial Gini coefficient. Defined as the Gini coefficient of the firing rate map   | 1xN double                          |
| spatialCoherence                             | Spatial Coherence. Defined by the degree of correlation between the firing rate map and a hollow convolution with the same map   | 1xN double                          |
| spatialPeakRate                              | Spatial peak firing rate (Hz). Defined as the peak rate from the firing rate map   | 1xN double                          |
| placeFieldsCount                             | Place field count: Number of intervals along the firing rate map that fulfills a set of spatial criteria: minimum rate of 2Hz and above 10% of the maximum firing rate bin and minimum of 4 connecting bins. The cell further has to have a spatial coherence greater than 0.6 (Mizuseki et al ?). | 1xN double                          |
| spatialSplitterDegree                        |  | 1xN double                          |
| placeCell                                    | Place cell (determined from the Mizuseki spatial metrics)  | 1xN binary                          |
| <b>Manipulation metrics</b>                  |  |                                     |
| manipulations                                | manipulations time series  | struct                              |
| .manipulationName                            |  | 1xN cell array of charactor vectors |
| <b>Response curves metrics</b>               |  |                                     |
| responseCurves                               | response curves  | struct                              |
| .responseCurveName                           |  | 1xN cell array of charactor vectors |
| <b>Quality metrics</b>                       |  |                                     |
| refractoryPeriodViolation                    | Refractory period violation (%): Fraction of ISIs less than 2ms  | 1xN double                          |
| isolationDistance                            | Isolation distance as defined by Schmitzer-Torbert et al. Neuroscience. 2005.  | 1xN double                          |
| IRatio                                       | L-ratio as defined by Schmitzer-Torbert et al. Neuroscience. 2005.   | 1xN double                          |
| <b>Hippocampal sharp wave ripple metrics</b> |  |                                     |
| deepSuperficial                              | Deep-Superficial region assignment [Unknown, Cortical, Superficial, Deep]  |                                     |
| deepSuperficialDistance                      | Deep Superficial depth relative to the reversal of the sharp wave (μm)   | 1xN double                          |
| <b>Hippocampal theta oscillation metrics</b> |  |                                     |
| thetaPhasePeak                               | Theta phase peak   | 1xN double                          |
| thetaPhaseTrough                             | Theta phase trough   | 1xN double                          |

|                                      |   |            |
|--------------------------------------|---|------------|
| thetaEntrainment                     | Theta entrainment   | 1xN double |
| thetaModulationIndex                 | Theta modulation index. determined from the ACG   | 1xN double |
| <b>Firing rate stability metrics</b> |   |            |
| firingRateGiniCoeff                  | The Gini coefficient of the firing rate across time   | 1xN double |
| firingRateCV                         | Standard deviation of the "firing rate across time" divided by the mean'  | 1xN double |
| firingRateInstability                | Mean of the absolute differential "firing rate across time" divided by the mean.<br>abs(diff(firingRateAcrossTime)) | 1xN double |

**Supplementary Table 1: Cell metrics, related to figure 1.** An incomplete list of the standard cell metrics. The full list is available online at [CellExplorer.org/datastructure/standard-cell-metrics](https://CellExplorer.org/datastructure/standard-cell-metrics)

| Functions                             | Description  |
|---------------------------------------|--|
| <b>sessionTemplate</b>                | A template script which automatically extracts and imports relevant metadata   |
| <b>gui_session</b>                    | A graphical user interface (GUI) for inspection and entry of metadata  |
| <b>ProcessCellMetrics</b>             | The processing module  |
| <b>CellExplorer</b>                   | The main graphical interface of CellExplorer   |
| <b>preferences_CellExplorer</b>       | Preferences for the graphical interface  |
| <b>preferences_ProcessCellMetrics</b> | Preferences for the processing module  |
| <b>NeuroScope2</b>                    | An Ephys visualizer built upon the data structure of CellExplorer  |
| <b>gui_MonoSyn</b>                    | GUI for manual curation of monosynaptic connections  |
| <b>gui_DeepSuperficial</b>            | GUI for manual curation of the depth assignment of neurons based on depth-related changes of sharp-wave-ripples (Mizuseki et al., 2011)                  |
| <b>loadCellMetricsBatch</b>           | Batch loading script for combining cell_metrics structs across sessions  |
| <b>loadCellMetrics</b>                | Script for loading cell_metrics with built-in common text filters (putative cell type, brain region, synaptic effect, label, animal, tags, groups, etc.) |
| <b>loadSpikes</b>                     | Script for importing and loading spike data  |
| <b>saveStruct</b>                     | Saving to various file containers (e.g. cellinfo, behavior, session, events, states)   |
| <b>saveCellMetrics</b>                | Saving cell metrics  |

**Supplementary Table 2. Primary MATLAB functions of the CellExplorer framework, related to Star method, related to figure 2.** All code is available at GitHub: <https://github.com/petersenpeter/CellExplorer>.