eLife

Reviewed Preprint
v1 • July 31, 2026
Not revised

For correspondence:
Jordan.Farrell@Childrens.Harvard.edu

* These authors contributed equally.

Competing interests: No competing interests declared

Funding: See page 15

Reviewing editor: Laura L Colgin, University of Texas at Austin, United States

Toothy: an interactive platform for dentate spike curation

Kathleen N Esfahany*,1,2,3, Amanda L Schott*,1,2, Julia M Grocott1,2, Jordan S Farrell1,2,4

[1]F.M. Kirby Neurobiology Center, Boston Children's Hospital, Boston, United States • [2]Rosamund Stone Zander and Hansjoerg Wyss Translational Neuroscience Center, Boston Children's Hospital, Boston, United States • [3]Program in Neuroscience, Harvard University, Boston, United States • [4]Department of Neurology, Harvard Medical School, Boston, United States

eLife Assessment

This paper describes a **valuable** tool for the detection and analysis of dentate spikes, network events in the dentate gyrus that are common yet understudied. This tool could help standardize dentate spike detection and analysis across labs and is therefore likely to be of interest to hippocampal neurophysiologists. However, the strength of evidence for its broad usefulness was viewed as **incomplete**, due to several identified bugs in the program and insufficient explanations of parameter selection and methods.

https://doi.org/10.7554/eLife.112308.1.sa4

Abstract

Dentate spikes (DSs) are hippocampal population events that occur during low-arousal states, defined by large-amplitude positive voltage peaks recorded in the hilus of the dentate gyrus (DG). DSs can be classified into two types (DS1 and DS2), with DS2 linked to transient increases in arousal, brain-wide neural activation, and functional relevance for memory encoding. Despite growing interest in their physiological and functional properties, no standardized framework exists for detecting and classifying DSs. We present Toothy, an open-source tool for DS detection and classification in large-scale electrophysiological recordings. Toothy offers a modular and interactive workflow comprising three key steps: (1) ingestion and preprocessing of local field potential (LFP) recordings, (2) detection of hippocampal population events, and (3) classification of DS types using peri-event current source density (CSD) profiles. To support both flexibility and reproducibility, features include compatibility with multiple recording formats, customizable processing parameters, interactive event review tools, and comprehensive logging of event detection and classification parameters. In addition to DS detection, Toothy can also detect sharp wave-ripples (SPW-Rs; an oscillatory population event recorded in the CA1 region), enabling comparative analysis between DSs and SPW-Rs. Toothy provides a standardized, reproducible pipeline for DS detection and classification, advancing broader efforts towards investigating hippocampal dynamics across diverse settings.

Introduction

The hippocampus is a critical structure for encoding, consolidating, and retrieving memories, with much of this processing occurring during offline behavioral states such as rest and sleep[1–4]. During these periods of low arousal, marked by large-amplitude, irregular local field potential (LFP) activity[5], the hippocampus exhibits two prominent network events: sharp wave-ripples (SPW-Rs) and dentate spikes (DSs). In rodents, SPW-Rs are characterized by high-frequency oscillations (100-180 Hz) in CA1 lasting 20-200 ms[6,7], whereas DSs are brief (<50 ms), large-amplitude peaks (>1 mV) in the hilus of the dentate gyrus (DG)[8]. Decades of research on SPW-Rs have highlighted their role in supporting memory through replay and consolidation[9–14], with consensus guidelines recently introduced to standardize their detection[15]. Meanwhile, despite

being first characterized around the same time as SPW-Rs, DSs remain relatively understudied. Given the mounting interest in DSs and their role in hippocampal function – including offline/remote memory reactivation[16,17], neural state switching[18], and associative memory formation[18,19] – tools for DS analysis would be especially timely for this emerging field, promoting standardized detection methods and comprehensive reporting of parameters used for defining DSs across studies.

DSs were originally reported to occur at a rate of up to 15 per minute, particularly during immobility and slow wave sleep, and did not occur during locomotion[8]. Since then, reported DS rates have spanned from 2-3 per minute to 2-3 per second, even showing positive modulation by locomotion in some cases[16–22]. This variability in DS rates is likely due to the amplitude-based thresholding for detection, which needs to be chosen carefully to avoid including other positive voltage peaks, such as movement artifacts and gamma oscillations during locomotion. It was also originally reported that DSs come in two types DS1 and DS2[8] (though see[23]), which later studies demonstrated have strikingly different properties, including local and brain-wide spiking differences as well as coupling of DS2 to arousal[16–18]. Although not universally performed across studies, the gold standard for DS classification is to apply current source density (CSD) analysis[24] to identify DS1 versus DS2, based on maximal current sinks in the outer or middle molecular layer of the DG, respectively. Precise CSD classification requires the use of linear probes with uniformly spaced electrodes spanning DG molecular layers down to the hilus, although waveform-based classification models, trained on CSD-classified data, can be used with over 80% accuracy in some cases[21]. Given the vast differences in reported rates and differences between DS types, a standardized and comprehensive set of tools for curating DSs would be beneficial to this growing field.

Here we present Toothy, an open-source graphical user interface (GUI) for detecting hippocampal population events (DS1s, DS2s, and SPW-Rs) in large-scale electrophysiological recordings. The GUI provides interactive controls and dynamic visualizations to facilitate the curation of high-quality event datasets. Toothy accommodates diverse experimental contexts and analysis goals by enabling parameter customization, while ensuring reproducibility through automatic documentation of parameter values. The graphical interface makes the workflow accessible to researchers without coding expertise, while the underlying open-source codebase supports bespoke feature development. By consolidating previously fragmented approaches into a single user-friendly platform, Toothy significantly reduces technical barriers to high-quality and reproducible DS detection and classification, thereby opening the doors to new insights into the properties and functional relevance of this understudied hippocampal population event.

Results

Overview of Toothy workflow

Toothy is written in Python and utilizes the cross-platform QT framework for its graphical interface, ensuring compatibility with MacOS, Windows, and Linux operating systems. The Toothy software leverages Python's robust suite of scientific tools to support a modular, interactive pipeline for processing and analyzing multichannel electrophysiological recordings from the hippocampus (Fig. 1). Researchers can install Toothy by downloading the underlying code repository from GitHub (*https://github.com/Farrell-Laboratory/Toothy/*) and following the documentation on the repository page.

Upon running Toothy, the user is greeted with a home screen comprising several buttons, including the main workflow ("Step 1: Load data" and "Step 2: Analyze events") and convenience menus for functions that can also be achieved within the main workflow (namely setting paths and parameters and creating probe configurations) (Fig. 2). The main workflow begins with data ingestion, in which the user selects a recording, specifies the associated probe configuration, and sets parameters for further processing. This step is followed by an internal processing stage in which Toothy scales and optionally downsamples the data, then detects events across all channels. Next, the user enters the main analysis phase of the workflow, channel selection, in which the user

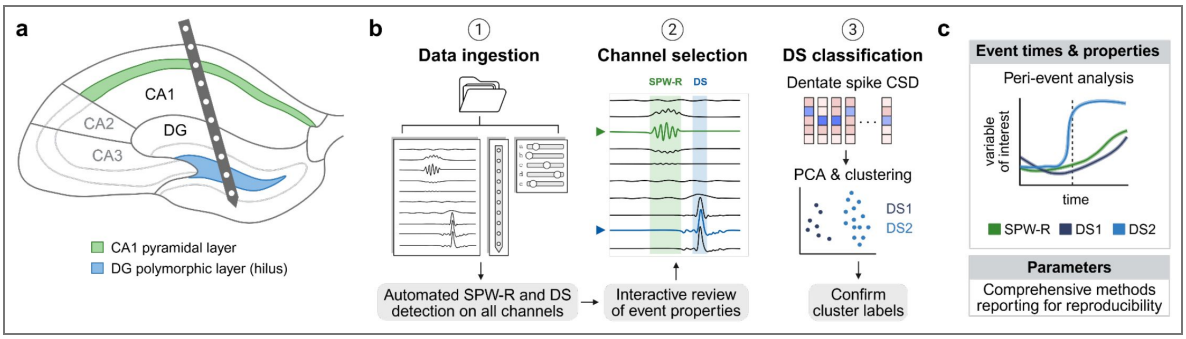


Figure 1. Toothy analysis pipeline for detecting hippocampal population events from electrophysiological recordings.

(a) Schematic of a laminar probe recording local field potentials (LFP) across hippocampal subregions in the mouse brain, including the CA1 pyramidal layer (green) and dentate gyrus (DG) polymorphic layer or hilus (blue). The multi-channel LFP recording can be analyzed with Toothy to detect sharp wave-ripples (SPW-Rs) and dentate spikes (DSs). **(b)** The Toothy pipeline consists of three main steps: (1) Data ingestion: The user loads LFP data, specifies the associated probe configuration, and sets parameters for further processing. Toothy performs automated event detection across all channels in the recording. (2) Channel selection: For each event type (SPW-Rs and DSs), the user selects a representative channel based on the channel's properties (selected channel marked by triangle and colored trace; green for SPW-R and blue for DS). Toothy provides interactive features for visualizing and comparing event properties across channels. (3) DS classification: For each DS, Toothy computes a current source density (CSD) profile, finds a 2D representation using PCA, then clusters events into DS1 and DS2. Toothy displays the mean waveforms and CSD profiles for each cluster for the user to use to refine the clustering before confirming the labels for DS1 and DS2. **(c)** Toothy output: Toothy returns a table of event times and properties as well as a comprehensive log of the parameters used for event detection and classification parameters. The event times enable peri-event analysis of other simultaneously-recorded variables of interest, including neural activity (e.g., spiking, photometry) and behavior (e.g. pupil size, whisker movement). The parameter log promotes comprehensive reporting of event detection methods, supporting reproducibility.

selects an optimal channel for identifying DSs or SPW-Rs, facilitated through several visualizations. Finally, the workflow concludes with a DS classification step, in which Toothy performs CSD analysis on the LFP for each DS, performs principal component analysis (PCA) on the CSD, then clusters the CSD to label each DS as type 1 or type 2. Toothy produces two key outputs: (1) a table of event times and properties, which can be used for any number of downstream analyses across concurrent modalities such as imaging and behavior, and (2) a log of the parameters used for each step of analysis, which can be used for methods section reporting to promote reproducibility.

Data Ingestion

The main Toothy workflow begins with data ingestion, which refers to the process of selecting a recording, as well as mapping it to a probe configuration and setting relevant parameters for further processing. These steps can be completed in the Data Ingestion Window of the GUI. To open the Data Ingestion Window, users click "Step 1: Load data" on the Toothy home screen. The Data Ingestion Window proceeds modularly through the following steps: (1) select a recording (Fig. 3a and 3b), (2) map the recording to a probe configuration (Fig. 3c), and (3) select processing parameters (Fig 4). After completing these three steps, users click "Process data" at the bottom of the window to trigger a signal processing pipeline for detecting events across all channels of the recording.

To select a recording, users click the folder icon to open a file explorer and select a file (Fig. 3a). Toothy supports six file formats, spanning both popular electrophysiology acquisition and storage formats (Neurodata Without Borders, OpenEphys, NeuroNexus, and Neuralynx) and generic array files (NumPy[25] and MATLAB arrays). Proprietary electrophysiology formats not directly supported can be converted into Neurodata Without Borders (NWB)[26] files using the *NeuroConv* Python package or the *NWB GUIDE* desktop app before using Toothy. Toothy uses the SpikeInterface Python package[27] to extract LFP data, timestamps, and relevant metadata (sampling rate and voltage units) from each file. For array files without embedded metadata, Toothy prompts the user to manually enter the information. When a recording is successfully loaded, the window expands to display the properties of the recording (number of channels, number of samples, and sampling rate) and timestamps (start time, end time, and duration) (Fig. 3b). Internally, following this step, all recordings are treated identically, regardless of source format.

Next, LFP signals must be mapped to the appropriate recording probe(s). The Data Ingestion Window automatically displays the number of signals detected in the raw data, as well as the number assigned to a valid probe channel (Fig. 3c). Users may press the "Load" button to upload a probe object from a saved configuration file. Alternatively, the "Create" button initiates a Probe Designer GUI where users can easily generate a new probe object. The "View" button launches a pop-up table displaying the probe index assigned to each data row, with blank cells indicating unassigned signals. Assigned probes will appear by name in the main window and can be duplicated or deleted using the associated icons. When each data row (i.e. channel) is successfully assigned to a unique probe channel, these items will appear in green and the "Set processing parameters" button at the bottom of the window becomes enabled. Clicking this button initiates Toothy's internal data processing step, in which LFP signals are filtered and event detection is performed for all channels using previously established methods[18].

Finally, users confirm the parameters used for processing the recording and performing event detection. Before performing filtering and other signal processing operations to detect events, the data may first be downsampled to reduce the time and storage consumed by Toothy.

Customizing parameters

Before initiating the data processing step, the user should confirm their chosen analysis parameters used in filtering and event detection. Parameters can be adjusted before starting the Toothy workflow (through the "Set parameters" button on the home screen) (Fig. 2c) or from within the Data Ingestion Window. Toothy's framework for managing numerous configuration

Figure 2. **Toothy home screen and convenience windows.**

(a) Home screen: Central access point for all Toothy modules. **(b) Path setting window:** Accessed from the "Set paths" button on the home screen, the path setting window allows the user to set default paths for folders and files to which Toothy will point at later stages in the workflow. **(c) Parameter setting window:** Accessed from the "Set parameters" button on the home screen, the parameter setting window allows the user to choose between using default parameters for signal processing and event detection (optimized to mouse hippocampal data) or using custom parameter files.

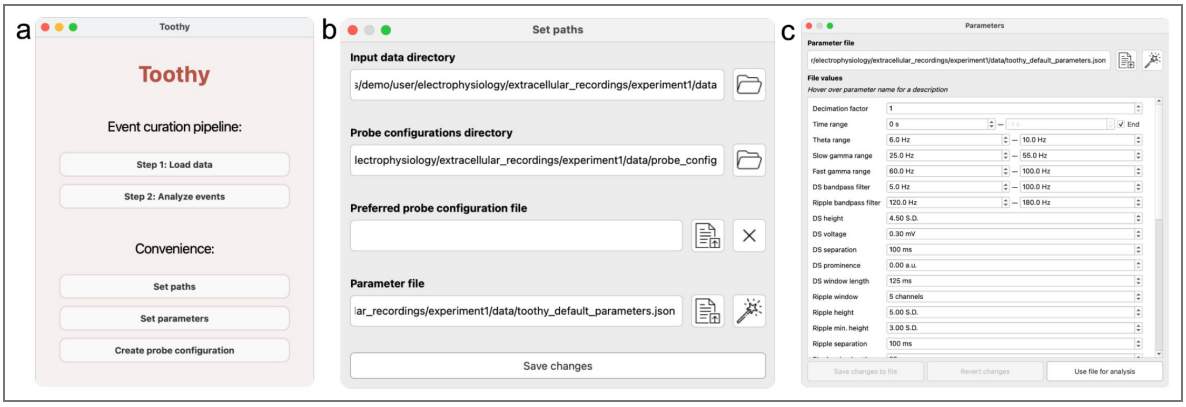
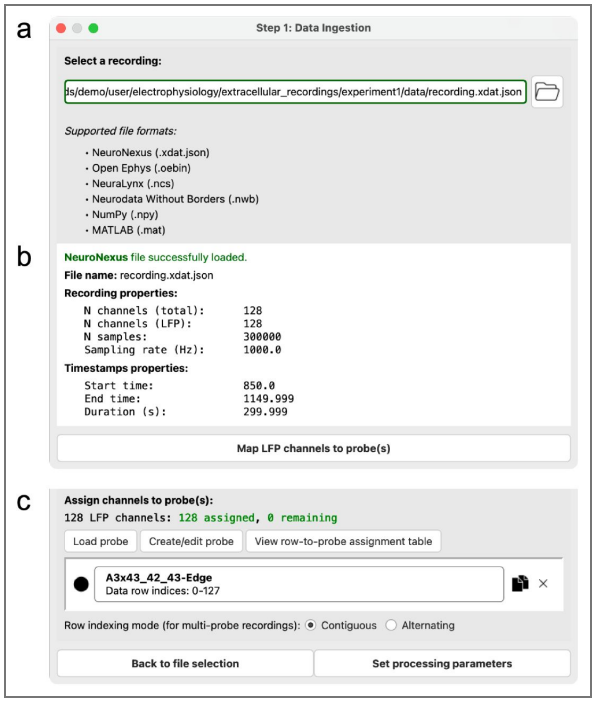


Figure 3. **Data Ingestion Window.**

(a) Step 1: Select a recording. Users begin data ingestion by selecting a recording. The user clicks on the folder icon to open a file explorer and select a recording file. Once a file is selected, the box with the file path turns green and the next sections of the window appear, shown in (b), alongside the "Map LFP channels to probe(s)" button. **(b) Metadata from the selected recording**. The properties of the recording (number of channels, number of samples, and sampling rate) and the associated timestamps (start time, end time, duration) are displayed. In this example, a NeuroNexus file with 128 channels sampled at 1000 Hz was loaded. **(c) Step 2: Map the recording to probe file(s)**. A probe file can be loaded or created to map the channels in the file to specific probe(s) and shank(s). Here, a probe configuration corresponding to a NeuroNexus A3x43/42/43-Edge probe. Once all channels have been mapped to a probe, the "Set processing parameters" button is enabled.



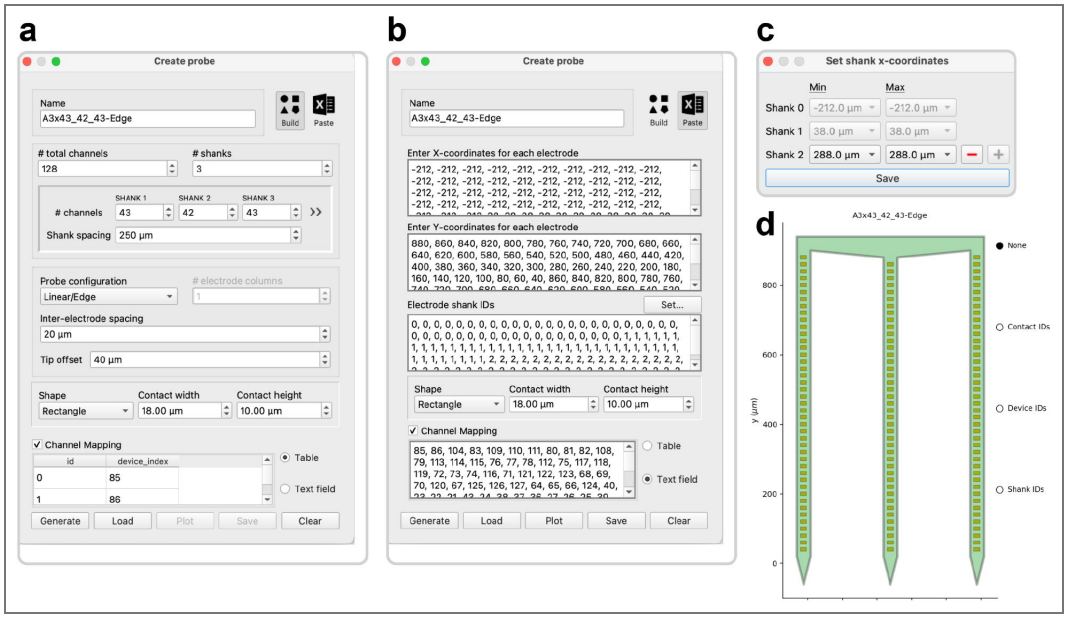


Figure 4. **Designing a probe object.**

(a) The Build tool generates a probe object from descriptive parameters, including channel count, shank count, electrode configuration, and contact spacing. Channel mappings can be entered as table rows or comma-separated values to set the probe's wiring configuration, otherwise the physical probe layout is assumed to match the logical data order. **(b)** The Paste method generates a probe object from arrays of electrode X and Y coordinates. Electrode shank IDs can be entered as a corresponding array, or they can be assigned to specific X-values through a dedicated interface **(c)** This method supports irregular or asymmetrical layouts that violate the assumptions of the "Build" tool. **(d)**The generated probe object is displayed with interactive labeling of contact indices, device indices, and shank IDs. Probe configurations are saved in JSON format for compatibility with the probeinterface package[27].

parameters allows the user to exert a granular level of control over the inputs to data analyses. The main settings file dictates the values for over fifty parameter inputs that affect all stages of the data pipeline, from signal processing and event detection to CSD estimation and cluster analysis.

Settings are stored in plain text (TXT) files for wide cross-platform compatibility and accessibility outside the GUI environment. Directly editing the parameter values through the text file is possible but not recommended, as this may disrupt the precise file structure required to accurately parse its contents; instead, editing through the GUI is advised. Toothy offers a dedicated interface for viewing and modifying the contents of the current parameter file, accessed by "Set parameters" on the home screen (Fig. 2c). The order of parameter rows in the GUI corresponds to the sequential stage of analysis; the first inputs govern raw data processing (e.g. downsampling factor, frequency band cutoffs, and event detection thresholds), followed by inputs for CSD calculation (e.g. estimation method) and finally DS classification (e.g. clustering algorithm). Changes can be saved to the existing settings file or used to create a new one, and the "Reset values" button erases any unsaved changes by reloading the original file.

Probe configurations

Multi-channel data analysis depends on the unique configuration of the neural probe and its accurate representation in the processing pipeline. While this may be straightforward in a single lab using a few standard probes, it becomes increasingly challenging when analyzing data from varied sources, each with its own probe geometry, size, and channel mapping. A flexible probe designer window allows users to easily create probe configuration files for each of these diverse probe types, enabling rapid customization to match the exact electrode geometry and wiring configuration of any physical probe in use (Fig. 4). Toothy uses the probeinterface Python package[28] to create software representations of probes, stored in a data format based on JSON to ensure compatibility and ease of data sharing across different systems and software environments.

A probe object is defined by its electrode geometry and wiring, edited through two main methods: the Build tool, which generates standard configurations (linear, polytrode, or tetrode) based on user parameters, and the Paste tool, which allows manual entry of arbitrary electrode coordinates for full customization. Additional options include specifying electrode spacing, tip offsets, and contact shapes/sizes, as well as defining channel mappings to link physical electrode contacts with device indices. By default, contact indices follow a consistent ordering scheme, but users can manually override this when hardware rearranges signals.

When all critical parameters for a chosen configuration are set to valid values, the Probe Designer GUI automatically enables the "Generate" button at the bottom of the window. This button synthesizes the current inputs to construct a unique probe object, plotted in a separate window for visual verification of the electrode geometry (Fig. 4d). The plot window also allows users to review the probe channel mapping by interactively displaying the contact indices, device indices, or shank IDs above the corresponding electrodes. Once a probe object is generated, the "Save" button allows the data to be written to a file and stored for later use. The default file name uses the text in the "Name" field as a base probe identifier, appends a "_config" tag signifying a probe configuration file, and recommends the `.json` extension for optimal data preservation. The GUI also supports exports to MAT and PRB file formats used by other software packages[29,30], although some information is lost as a result. A previously saved probe object can be imported by using the "Load" button to select the target configuration file and read in its probe data, which will update the GUI accordingly.

Event detection

The Toothy data processing pipeline prepares LFP signals for analysis through a series of transformations, manipulations, and preliminary analyses according to the parameters set in the previous phases. The first step is to organize, scale, and downsample the raw data. Using channel mappings from one or more probes in the recording, signals are organized by probe, with each channel array sorted by depth, from shallowest to deepest. Signals are then scaled to standard

units of millivolts for consistency. By default, Toothy does not downsample LFP data when the sampling rate is already at 2.5kHz or below. However, to facilitate efficient processing and visualization of data at higher sampling rates, Toothy can downsample data from its original sampling rate (e.g.,kHz, 10 kHz, or 30 kHz) to a user-defined rate (1 kHz by default) by averaging bins of consecutive samples.

Once organized, the LFP signals are bandpass filtered to isolate specific frequency bands with the following default values: theta (6-10 Hz), slow gamma (25-55 Hz), fast gamma (60-100 Hz), ripple (120-180 Hz), and a broader 5-100 Hz band used for DS detection. Each filtered signal is then analyzed for amplitude and calculated as the standard deviation over the entire recording, which provides a mean value for each channel. To facilitate comparisons, these values are normalized across all channels on each probe for visualization the relative power in each frequency band across depth (Fig. 5d). Both raw and normalized power values are saved in a summary table for further analysis.

In the next phase, preliminary detection of DSs and SPW-Rs is performed on each channel. For DS detection, positive peaks are identified in the filtered LFP signal using the `find_peaks` function from Python's SciPy library[31]. Peaks must meet specific criteria: exceeding a user-defined height threshold, having a minimum prominence, and being separated by a minimum time interval to ensure that DSs with multiple peaks are not extracted as discrete events. For the qualifying events, additional waveform properties such as half-width, peak height, and asymmetry are computed using SciPy's `peak_widths` function within an analysis window around each peak. Since filtered signal peaks may not align precisely with raw signal peaks, each DS is indexed by the maximum raw LFP value within a 20 ms window around the filtered signal peak. This analysis is repeated for all channels, and results are compiled into a single table for storage.

For SPW-R detection, Toothy first calculates the SPW-R envelope by taking the absolute value of the Hilbert transform of the ripple-filtered signal. The instantaneous frequency is then computed by translating the phase from the Hilbert transform across cycles, converting it into frequency values that are clipped to the ripple band (120-180 Hz). SciPy's `find_peaks` function is again used to detect peaks in the SPW-R envelope which meet the user-defined height and inter-peak interval thresholds. A lower envelope threshold is then applied, and the `peak_widths` function is used to measure SPW-R widths. Additional criteria are used to filter detected SPW-Rs, including minimum duration and mean instantaneous frequency thresholds. Qualified SPW-Rs are indexed by the largest positive cycle within 40 ms of the filtered peak. As with DSs, SPW-R analysis is performed for all channels, and results are consolidated into a table for storage, along with threshold values for each channel.

The final outputs of the data processing phase include the original and filtered LFP signals (scaled and downsampled), frequency band power values, and detailed tables of DS and SPW-R detection results. Data files are organized by probe in a central recording directory, along with probe configuration data and the parameter settings used in processing. This directory serves as a base resource for subsequent analyses and ensures reproducibility across different recordings.

Channel selection

The channel selection stage of the analysis pipeline is dedicated to refining and validating the detection of DSs and SPW-Rs after automated processing, with the goal of curating high-quality datasets for downstream analysis. Although Toothy detects candidate events across all recorded channels, true physiological events are expected to occur within specific hippocampal subregions, with DSs localizing to the hilus of the DG, due to its unique folded cytoarchitecture[32]. Any detected "events" outside these regions are likely to be spurious or to reflect unrelated activity patterns, such as large gamma oscillations, leaving a narrow range of potential channels for each event type. These channels can be rigorously compared using the Channel Selection Window (Fig. 5), which integrates time-series visualization, statistical insights, and interactive tools to identify the optimal event channels and manually edit the data.

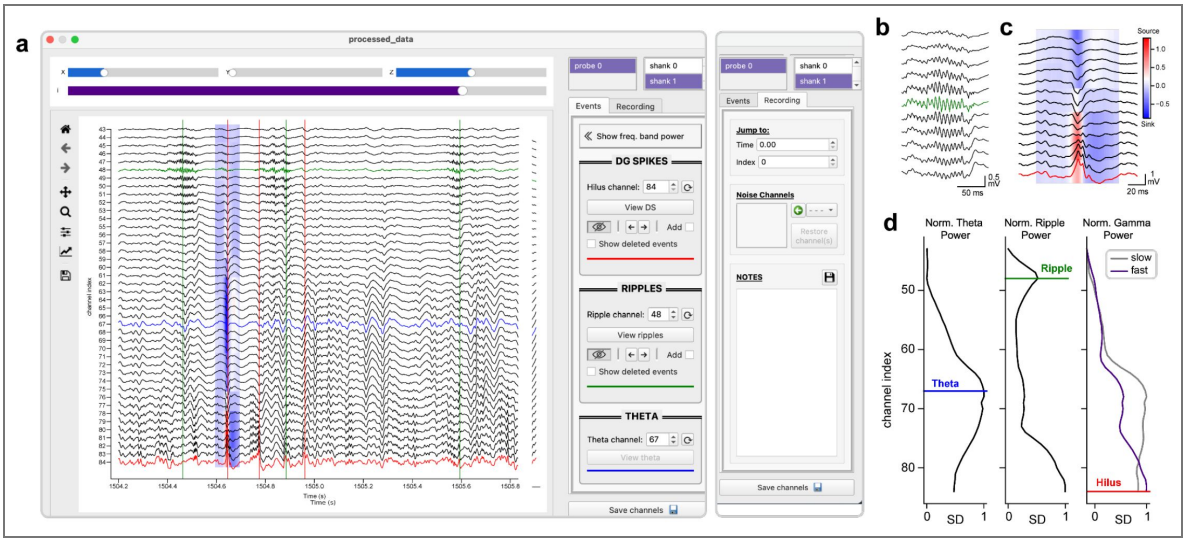


Figure 5. Selecting event channels and curating DS and SPW-R datasets

(a) The Channel Selection Window displays raw LFP traces from the selected probe and shank, with the color-coded signals representing the current event channels for DSs (red), SPW-Rs (green), and maximal theta amplitude (blue). The red and green vertical lines mark the timepoints of detected DSs and SPW-Rs, respectively, and the CSD surrounding the first DS event (blue patch) is displayed behind the LFP traces. The Events tab in the sidebar contains tools for setting event channels and analyzing DS and SPW-R datasets, while the Recording tab (right) provides general utilities for navigating, cleaning, and documenting the data. **(b)** Example SPW-R LFPs in CA1, expanded from the upper dashed box in panel **a**. **(c)** Example DS LFPs and CSD in the hilus, expanded from the lower dashed box in panel **a**. **(d)** Normalized amplitude across channels for the theta, ripple, and slow and fast gamma frequency bands, with event channels marked by color-coded lines.

The main interface (Fig. 5a) consists of a central plot rendered using Python's Matplotlib library[33], and a settings sidebar with a suite of interactive elements. In the upper right corner, users can switch between the available probes for the recording, and between the available shanks for the currently selected probe. LFP signals from the specified probe and shank are vertically stacked according to depth and displayed over a given interval, providing an adjustable "viewing window" into the recording. The main horizontal slider (purple) controls the position of the viewing window, enabling users to quickly navigate across the recording. Pressing the left and right arrow keys shifts the window backward or forward by 25% of its time range, providing higher temporal resolution for stepping through local data. The default size of the viewing window is 2 seconds, which can be reduced or expanded by the "X" slider to inspect specific waveforms or evaluate broader patterns. The "Y" slider changes the height of the central figure, effectively stretching the vertical arrangement of LFP channels for enhanced clarity on smaller displays, while the "Z" slider scales the amplitude of individual LFP traces. Another interactive tool is the span selector, a red box dragged horizontally across the plot to visually select a time range. Pressing the "Enter" key calculates an instantaneous CSD for the highlighted interval, temporarily visualized as a colormap overlaying the corresponding LFP signals (Fig. 5a,c).

The right-hand sidebar is organized into an "Events" tab with specific tools for analyzing DSs and SPW-Rs, and a "Recording" tab with general utilities. In the latter, the "Time" and "Index" inputs move the viewing window to a given timestamp or recording index, allowing users to efficiently jump between points of interest. Users can also designate individual broken or artifact-contaminated channels as "noise", excluding them from event detection, CSD computation, and frequency band analyses. Noise channels appear as flat gray traces in the main display, and they can be restored by setting the designation back to "clean". Finally, the "Notes" section provides built-in documentation through an editable text field, allowing users to record their observations directly within the GUI. This input is linked with a plain text file (`notes.txt`) in the recording folder, enabling preservation of the information across multiple analysis sessions.

The "Events" tab of the settings sidebar is organized into event-specific panels, and editable inputs are used to specify the optimal channels for DS and SPW-R events. A third input specifies the channel with the maximal theta amplitude, used to identify the hippocampal fissure as a landmark for subsequent CSD analysis of DSs (section 5). Initial event channels are roughly estimated based on their spectral properties and identified in the main plot by color-coded LFP signals, using red for the DS (or hilus) channel, green for the SPW-R channel, and blue for the theta (or fissure) channel. Event markers are displayed for DSs and SPW-Rs on the current channels, represented by red and green vertical lines. Any changes to the event channel indexes are reflected in real time by dynamically updating the color-coding and event markers on the main plot, and channel inputs can be reset to their initial values using the adjacent buttons. Toggle buttons in the DS and SPW-R sections allow users to show or hide the corresponding event markers, while pairs of left and right arrow buttons enable direct navigation between detected events. These features are useful for initial event review, facilitating a preliminary understanding of their distribution and waveform characteristics.

At the top of the "Events" tab is a toggle button to show the relative amplitude across theta, ripple, and gamma frequency bands (Fig. 5d). Channels on the Y-axes of the frequency band plots are aligned with the main LFP plot for cross-referencing, and the designated event channels are marked by color-coded lines, which are dynamically updated when the channel inputs are modified. These plots compare the current event channels with peaks in the relative amplitude of the corresponding frequency band, providing valuable context for informing channel selections. Together, these visualizations provide a foundation for an informed initial hypothesis of the optimal event channels.

To refine these initial selections, DS- and SPW-R-specific interfaces (Fig. 6) are used to compare detailed event properties across channels. The central plot (Fig. 6a) is split into two rows, with static quantitative insights on top and interactive waveform visualizations on bottom. The three statistical subplots in the top row summarize key parameters across all channels, including event count and peak amplitude for both DSs and SPW-Rs, with data points color-coded by magnitude

for enhanced clarity. The third subplot is event-specific, showing the peak height above the surround for DSs (Fig. 6a) or the ratio of ripple to theta power for SPW-Rs (Fig. 6b). The distribution of these properties can identify strong candidate channels for each event; in particular, a peak in the ripple-to-theta band ratio and a peak in DS amplitude guides selection of the SPW-R and DS channel, respectively. Data from the currently selected channel can be highlighted with a red outline, using a toggle button in the settings sidebar. In the bottom row, LFP signals are analyzed using two different visualization modes. In "Single" mode (Fig. 6d,e), a horizontal slider is used to scroll through individual waveforms in sequence. Users can sort events by attributes such as timestamp, size, and width to view the full range of the distribution and inspect the waveforms at each extreme. In "Average" mode (Fig. 6a,c), the mean event waveforms from multiple channels are overlaid, enabling side-by-side comparisons of their morphology. Both modes allow users to adjust the time window, scale the data, and toggle between raw LFPs and the filtered signals used for event detection. Additional options in the settings are used to show and hide various detection thresholds and waveform annotations. These tools balance quantitative metrics with heuristic reasoning to facilitate the selection of optimal event channels.

A powerful feature of the Channel Selection Window is the ability to manually curate event datasets, refining the set of automatically detected events by removing artifacts and adding missed events. To add an undetected DS or SPW-R to the dataset, the user must double-click the desired timepoint on the plot while the "Add" option is checked in the corresponding event section. Newly created events are indexed to the LFP peak closest to the clicked point, and they appear as dotted lines for visual differentiation. Users can remove specific DSs or SPW-Rs by dragging the span selector around the desired event markers and pressing the "Backspace" key to temporarily delete the events, or the "Escape" key to permanently erase them. Deleted events can be viewed through a separate toggle button (appearing as dashed lines), and they can be automatically restored by selecting the markers with the span selector and pressing the spacebar. Erased events, however, are permanently removed from the dataset and must be manually re-added by the user. This iterative refinement process generates high-quality, rigorously validated DS and SPW-R datasets for downstream analysis.

The final step of exporting the data to the recording folder is executed using the "Save" button (Fig. 5). Event channels are written to an NPY file, while DS and SPW-R datasets are saved as CSV tables. The current probe and shank index is appended to each file name as a unique identifier to avoid confusion in multi-probe and multi-shank recording setups. Successful completion of this step allows access to the DS classification module, the final phase of the Toothy pipeline. While Toothy is primarily geared towards DS curation, users may refer to the previously published consensus statement on methods for further curating SPW-Rs[15].

Classification of DS1 and DS2

DSs can be separated into two types based on current sinks in the outer (type 1, DS1) or middle (type 2, DS2) molecular layer of the DG, identified by performing CSD analysis. To classify DS events into DS1 and DS2, Toothy provides a DS Classification Window that streamlines CSD estimation and clustering analyses (Fig. 7). The interface offers four modular visualization options: i) an interactive CSD window (Fig. 7a), ii) heatmaps of raw and processed signals (Fig. 7b), iii) mean event profiles for DS1 and DS2 (Fig. 7c), and iv) clustering results (Fig. 7d). The workflow begins by selecting channels for CSD analysis using an interactive slider. By default, the CSD window spans from the selected DS channel (i.e. the hilus) to the selected theta channel (i.e. approximately the hippocampal fissure) to encompass the entire molecular layer of the DG. Users should exclude channels above the fissure from CSD estimation, as the strong sink exhibited by the stratum radiatum during SPW-Rs, which can co-occur with DSs[8,18,22], introduces an additional dimension into the clustering analysis and may prevent accurate classification of DS types.

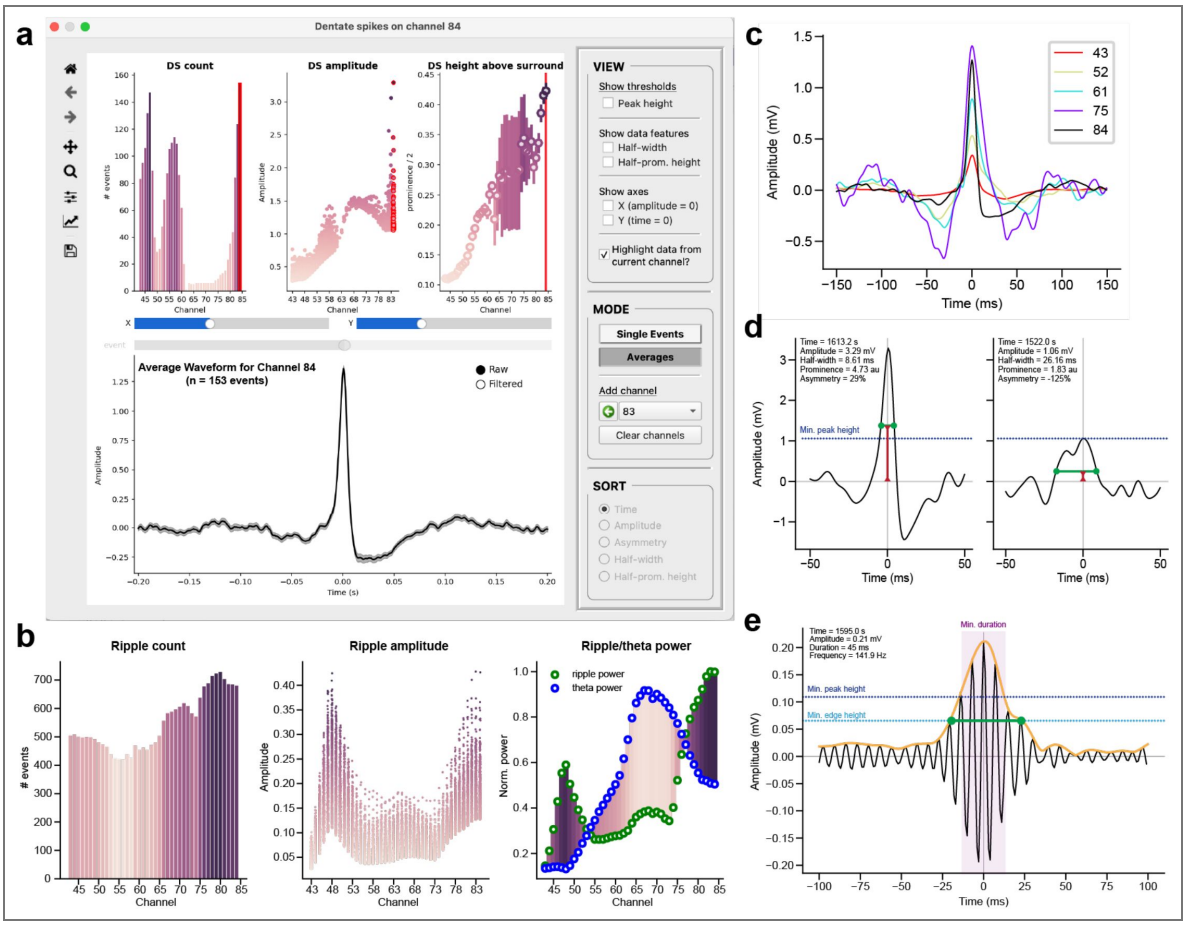


Figure 6. **Event-specific analyses.**

(a) Interface for comparing DS and SPW-R events. “Single” mode reviews the individual waveforms in a dataset, and “Average” mode compares the mean event waveforms across multiple channels. **(b)** Parameter distributions for SPW-R events, showing the event count, amplitude, and ripple-to-theta ratio across all channels. **(c)** Mean DS waveforms from five example channels distributed across the probe. The overlaid waveforms highlight laminar differences in polarity, latency, and waveform shape across the hippocampal circuit. **(d)** Individual waveforms from two example DSs with the highest and lowest peak amplitude. The peak detection threshold is indicated by the dashed line, and waveforms are annotated to show with their half-width (green) and half-prominence height (red). **(e)** Example SPW-R waveform and ripple envelope (orange) in the filtered LFP. Detection thresholds are displayed for minimum peak height and edge height (dashed lines) and minimum duration (purple shaded region), and the event duration is indicated by the green bar.

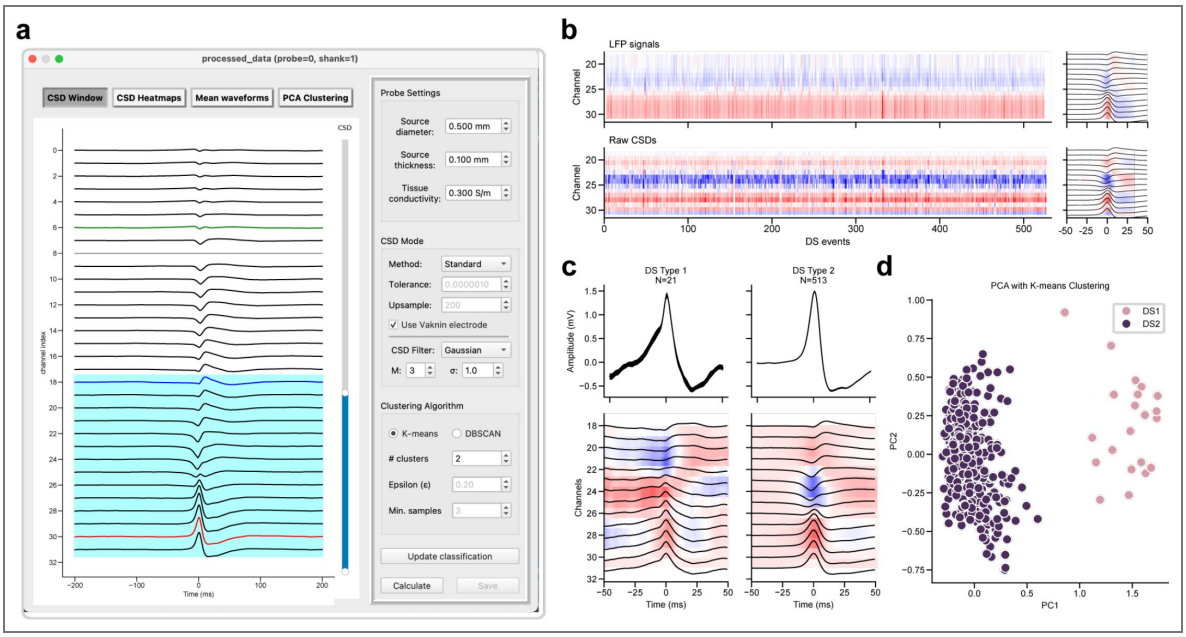


Figure 7. Classification of DS1 and DS2 using CSD estimation and event clustering.

(a) Interface for CSD-based classification. The interactive CSD window is used to define the range of channels for analysis, and the sidebar enables configuration of CSD calculation settings and clustering algorithm parameters. The "Calculate" button initiates the CSD estimation and PCA-based clustering, and the results are displayed in three additional visualizations accessible through the inputs above the plot. **(b)** Heatmaps of raw LFPs (top) and raw CSDs (bottom) for each DS event. Right, mean LFPs and CSD surrounding DSs. **(c)** Mean LFP waveforms and filtered CSDs for DS1 and DS2 events. **(d)** Two-dimensional PCA projection of the normalized filtered CSDs. Events are color-coded by cluster label (DS1 or DS2).

Users may choose the standard CSD method or one of several inverse CSD (iCSD) approaches, which include δ-source, step, and spline methods[24]. By default, Toothy uses the standard CSD method with Vaknin's procedure and third-order Gaussian smoothing (σ=1), but alternative CSD methods will have a negligible impact on the subsequent clustering. Heatmaps are used to display the LFPs and the computed CSDs (Fig. 7b). Summary plots display the averaged LFPs and CSDs within a 100 ms window surrounding each DS, with the color gradient indicating the signal intensity across channels and events.

Following CSD estimation, principal components analysis (PCA) is performed on the normalized filtered CSD profiles. Under physiological conditions, the PCA-based analysis should produce two distinct clusters, corresponding to DS1 and DS2 events, as evident on the PCA scatterplot (Fig. 7d). Points are color-coded by classification as DS1 or DS2, based on the clustering results from K-means or DBSCAN algorithms, available for comparison via toggle buttons on the plot. K-means requires a predefined number of target clusters, while DBSCAN requires parameters for epsilon (the local radius for expanding clusters) and the minimum number of samples per cluster (note DBSCAN can produce a third "undefined" cluster). Toothy automatically labels the clusters by comparing their mean CSDs, designating the cluster with a maximal sink closer to the fissure as DS1 and that with a sink closer to the granule cell layer as DS2. The mean LFP waveforms and associated peri-event CSDs are displayed for DS1 and DS2, along with the number of events in each category (Fig. 7c). These plots allow users to compare the classification results with expected physiological patterns, and users can switch the label assigned to each cluster if visual inspection determines that this automatic labeling was inaccurate. Following successful classification, the user can press "Save" to add the event types to the DS dataset. This step concludes the Toothy workflow, producing a set of CSV files with event times and types that facilitate any number of downstream analyses.

Discussion

Toothy addresses a critical gap in the field of hippocampal dynamics: the lack of standardized and accessible methods for dentate spike detection and classification. By providing a user-friendly, open-source platform for curating high-quality datasets of hippocampal events, Toothy opens the door to a wide range of downstream analyses to investigate the properties and functional relevance of DSs. The modular, customizable pipeline – encompassing data ingestion, channel selection, and DS classification – promotes flexibility across experimental contexts while ensuring reproducibility through automatic parameter logging.

Toothy functions not as a black-box detector but as an interactive tool that supports user-guided curation. While Toothy reduces technical barriers by consolidating complex analytical steps into a graphical interface, effective use still requires familiarity with hippocampal LFP events. For example, an ideal user would know to consider the theta/ripple band ratio in selecting a SPW-R channel, the DS height and mean waveform in selecting a DS channel, and the relative sink position in confirming the automatically-assigned DS1/DS2 classification labels. While Toothy selects event channels and classification labels using heuristics developed primarily from dorsal hippocampal recordings in mice, the unique nature of each recording necessitates a user with some prior knowledge of hippocampal events to ensure proper completion of each step. In addition to user expertise, Toothy's performance depends on the quality and properties of the input data. The curation of appropriately separated DS types requires sufficient channels in the molecular layer and hilus and sufficient sampling of quiescent behavioral states in which DSs occur (awake rest and non-REM sleep). Nevertheless, given that Toothy's clustering of event types uses PCA, which is agnostic to the location of current sinks and sources, irregular sampling of DG LFP with alternative electrode configurations (e.g., wire bundles or tetrodes) may still yield distinct DS clusters that can be appropriately classified based on waveform differences[21] and behavioral features[18].

Ultimately, Toothy provides a foundation for reproducible analyses of dentate spikes. Standardized detection and classification of DSs and SPW-Rs will facilitate comparisons across studies and enable integration with behavioral or imaging data.

Acknowledgements

K.N.E. is supported by an NSF GRFP award. A.L.S. is supported by NIH NINDS T32 training program (T32NS007473). Dentate spike research in the laboratory of J.S.F. is supported by NIH NINDS (R00NS126725), Esther A. & Joseph Klingenstein Fund, Simons Foundation, CureSHANK, CURE Epilepsy, and the Harvard Brain Science Initiative Bipolar Disorder Seed Grant.

Additional files

Supplementary information.

Supplementary sequence data.

Additional information

Funding

Funder	Grant reference number	Author
Citizens United for Research in Epilepsy (CURE)		Jordan S Farrell
Klingenstein Third Generation Foundation (KTGF)		Jordan S Farrell
Harvard Medical School (HMS)		Jordan S Farrell
Simons Foundation (SF)		Jordan S Farrell
HHS | NIH | National Institute of Neurological Disorders and Stroke (NINDS)	R00NS126725	Jordan S Farrell
NSF | National Science Foundation Graduate Research Fellowship Program (GRFP)		Kathleen N Esfahany
HHS | NIH | National Institute of Neurological Disorders and Stroke (NINDS)	T32NS007473	Amanda L Schott

Author ORCID iDs

Jordan S Farrell: https://orcid.org/0000-0002-1209-5559

References

1. **Buzsáki G** (1989) Two-stage model of memory trace formation: A role for "noisy" brain states. *Neuroscience* **31**:551-570 https://doi.org/10.1016/0306-4522(89)90423-5 | PubMed
2. **Scoville WB,** Milner B (1957) Loss of recent memory after bilateral hippocampal lesions. *J Neurol Neurosurg Psychiatry* **20**:11-21 https://doi.org/10.1136/jnnp.20.1.11
3. **Eichenbaum H** (2000) A cortical–hippocampal system for declarative memory. *Nat Rev Neurosci* **1**:41-50 https://doi.org/10.1038/35036213 | PubMed
4. **Brodt S,** Inostroza M, Niethard N, Born J (2023) Sleep—A brain-state serving systems memory consolidation. *Neuron* **111**:1050-1075 https://doi.org/10.1016/j.neuron.2023.03.005 | PubMed
5. **Vanderwolf CH** (1969) Hippocampal electrical activity and voluntary movement in the rat. *Electroencephalogr Clin Neurophysiol* **26**:407-418 https://doi.org/10.1016/0013-4694(69)90092-3 | PubMed
6. **Buzsáki G** (2015) Hippocampal sharp wave-ripple: A cognitive biomarker for episodic memory and planning. *Hippocampus* **25**:1073-1188 https://doi.org/10.1002/hipo.22488 | PubMed
7. **Buzsáki G** (1986) Hippocampal sharp waves: Their origin and significance. *Brain Res* **398**:242-252 https://doi.org/10.1016/0006-8993(86)91483-6 | PubMed

8. **Bragin A,** Jando G, Nadasdy Z, Van Landeghem M, Buzsaki G (1995) Dentate EEG spikes and associated interneuronal population bursts in the hippocampal hilar region of the rat. *J Neurophysiol* **73**:1691-1705 https://doi.org/10.1152/jn.1995.73.4.1691 | PubMed

9. **Foster DJ,** Wilson MA (2006) Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature* **440**:680-683 https://doi.org/10.1038/nature04587 | PubMed

10. **Skaggs WE,** McNaughton BL (1996) Replay of Neuronal Firing Sequences in Rat Hippocampus During Sleep Following Spatial Experience. *Science* **271**:1870-1873 https://doi.org/10.1126/science.271.5257.1870 | PubMed

11. **Diba K,** Buzsáki G (2007) Forward and reverse hippocampal place-cell sequences during ripples. *Nat Neurosci* **10**:1241-1242 https://doi.org/10.1038/nn1961 | PubMed

12. **Fernández-Ruiz A,** Oliva A, Fermino De Oliveira E, Rocha-Almeida F, Tingley D, Buzsáki G (2019) Long-duration hippocampal sharp wave ripples improve memory. *Science* **364**:1082-1086 https://doi.org/10.1126/science.aax0758 | PubMed

13. **Girardeau G,** Benchenane K, Wiener SI, Buzsáki G, Zugaro MB (2009) Selective suppression of hippocampal ripples impairs spatial memory. *Nat Neurosci* **12**:1222-1223 https://doi.org/10.1038/nn.2384 | PubMed

14. **Gridchyn I,** Schoenenberger P, O'Neill J, Csicsvari J (2020) Assembly-Specific Disruption of Hippocampal Replay Leads to Selective Memory Deficit. *Neuron* **106**:291-300.e6 https://doi.org/10.1016/j.neuron.2020.01.021 | PubMed

15. **Liu AA,** Henin S, Abbaspoor S, et al. (2022) A consensus statement on detection of hippocampal sharp wave ripples and differentiation from other fast oscillations. *Nat Commun* **13**:6000 https://doi.org/10.1038/s41467-022-33536-x | PubMed

16. **Dvorak D,** Chung A, Park EH, Fenton AA (2021) Dentate spikes and external control of hippocampal function. *Cell Rep* **36**:109497 https://doi.org/10.1016/j.celrep.2021.109497 | PubMed

17. **McHugh SB,** Lopes-dos-Santos V, Castelli M, et al. (2024) Offline hippocampal reactivation during dentate spikes supports flexible memory. *Neuron* **112**:3768-3781.e8 https://doi.org/10.1016/j.neuron.2024.08.022 | PubMed

18. **Farrell JS,** Hwaun E, Dudok B, Soltesz I (2024) Neural and behavioural state switching during hippocampal dentate spikes. *Nature* **628**:590-595 https://doi.org/10.1038/s41586-024-07192-8 | PubMed

19. **Nokia MS,** Gureviciene I, Waselius T, Tanila H, Penttonen M (2017) Hippocampal electrical stimulation disrupts associative learning when targeted at dentate spikes. *J Physiol* **595**:4961-4971 https://doi.org/10.1113/JP274023 | PubMed

20. **Lensu S,** Waselius T, Penttonen M, Nokia MS (2019) Dentate spikes and learning: disrupting hippocampal function during memory consolidation can improve pattern separation. *J Neurophysiol* **121**:131-139 https://doi.org/10.1152/jn.00696.2018 | PubMed

21. **Santiago RMM,** Lopes-dos-Santos V, Aery Jones EA, Huang Y, Dupret D, Tort ABL (2024) Waveform-based classification of dentate spikes. *Sci Rep* **14**:2989 https://doi.org/10.1038/s41598-024-53075-3 | PubMed

22. **Headley DB,** Kanta V, Paré D (2017) Intra-and interregional cortical interactions related to sharp-wave ripples and dentate spikes. *J Neurophysiol* **117**:556-565 https://doi.org/10.1152/jn.00644.2016 | PubMed

23. **Tarcsay G,** Saxena R, Long R, Shobe JL, McNaughton BL, Ewell LA (2025) Dentate spikes comprise a continuum of relative input strength from the lateral and medial entorhinal cortex. *bioRxiv* https://doi.org/10.1101/2025.10.27.684857 | PubMed

24. **Pettersen KH,** Devor A, Ulbert I, Dale AM, Einevoll GT (2006) Current-source density estimation based on inversion of electrostatic forward solution: Effects of finite extent of neuronal activity and conductivity discontinuities. *J Neurosci Methods* **154**:116-133 https://doi.org/10.1016/j.jneumeth.2005.12.005 | PubMed

25. **Harris CR,** Millman KJ, Van Der Walt SJ, et al. (2020) Array programming with NumPy. *Nature* **585**:357-362 https://doi.org/10.1038/s41586-020-2649-2 | PubMed

26. **Rübel O,** Tritt A, Ly R, et al. (2022) The Neurodata Without Borders ecosystem for neurophysiological data science. *eLife* **11**:e78362 https://doi.org/10.7554/eLife.78362 | PubMed

27. **Buccino AP,** Hurwitz CL, Garcia S, et al. (2020) SpikeInterface, a unified framework for spike sorting. *eLife* **9**:e61834 https://doi.org/10.7554/eLife.61834 | PubMed

28. **Garcia S,** Sprenger J, Holtzman T, Buccino AP (2022) ProbeInterface: A Unified Framework for Probe Handling in Extracellular Electrophysiology. *Front Neuroinformatics* **16**:823056 https://doi.org/10.3389/fninf.2022.823056 | PubMed

29. **Pachitariu M,** Sridhar S, Pennington J, Stringer C (2024) Spike sorting with Kilosort4. *Nat Methods* **21**:914-921 https://doi.org/10.1038/s41592-024-02232-7 | PubMed

30. **Yger P,** Spampinato GL, Esposito E, 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 https://doi.org/10.7554/eLife.34518 | PubMed

31. **Virtanen P,** Gommers R, Oliphant TE, et al. (2020) SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat Methods* **17**:261-272 https://doi.org/10.1038/s41592-019-0686-2 | PubMed

32. **Fernández-Ruiz A,** Muñoz S, Sancho M, Makarova J, Makarov VA, Herreras O (2013) Cytoarchitectonic and Dynamic Origins of Giant Positive Local Field Potentials in the Dentate Gyrus. *J Neurosci* **33**:15518-15532 https://doi.org/10.1523/JNEUROSCI.0338-13.2013 | PubMed

33. **Hunter JD** (2007) Matplotlib: A 2D Graphics Environment. *Comput Sci Eng* **9**:90-95 https://doi.org/10.1109/MCSE.2007.55

Peer reviews

Reviewer #1 (Public review):

Summary:

Esfahany et al. describe a new platform (Toothy) to identify and analyze dentate spikes and sharp wave ripples from silicon probe electrophysiology data. The goal is to facilitate and standardize the extraction of DS1 and DS2 events, which have highly variable properties across recordings from different labs. The manuscript describes the basic workflow of the Toothy pipeline, including loading data, assigning channels along a linear probe, customizing parameters, selecting ideal channels for analysis, and classifying DS1 and DS2 events.

Strengths:

The manuscript is clear and easy to follow and does a good job of describing the platform. Overall, this will be a useful analysis pipeline that can help to standardize DS analysis across labs and datasets.

Weaknesses:

The current version has several bugs that prevent analysis, and the documentation of analysis parameters needs to be improved.

(1) In limited testing, the pipeline had several bugs, and I was not able to complete the full analysis of a dataset. Loading data from .mat or .npy files gave errors (it seemed that the metadata was not loaded correctly from the pop-up window). I was able to load a .nwb file, which worked well. The probe configuration tool was a bit difficult to understand, and there was not much documentation to help, although it worked when simply entering the x-y coordinates of the channels. It also crashed several times while trying to make a probe configuration due to it trying to save when a small typo was briefly entered. The initial

analysis worked well, and the auto-selected channels matched our recording notes and seemed appropriate. DSs and ripples were extracted. An error came when trying to classify DSs, and the program repeatedly crashed across a variety of parameters. Overall, parts of the pipeline worked well, but others had significant bugs that need to be addressed.

(2) The authors should provide test data that can be run through the pipeline. Ideally, this could use a variety of data types, probes, and conditions so that it is clear how they differ.

(3) There are a lot of parameters that can be adjusted, but very little information about how they are chosen and what goes into parameter selection for a dataset. Additional documentation with more information on adjustable parameters, channel selection, and best practices would help improve the utility of the tool. Ideally, this could also integrate citations (either in the manuscript or documentation) to support some of the choices made during parameter selection.

(4) There is no validation presented against other analysis methods or datasets. While there is no ground truth of when DSs occur, this may limit the ability of this tool to become the standard for DS analysis. A section comparing the analysis used in the pipeline to other published analyses would be helpful.

(5) In the manuscript, it would be helpful to further describe the rationale for initially detecting DSs and SPW-Rs on all channels, when they are network events that occur across channels.

(6) A section on what hardware and software are necessary to run the pipeline should be added.

https://doi.org/10.7554/eLife.112308.1.sa3

Reviewer #2 (Public review):

Summary:

This work provides an open-source, Python-based, graphical user interface for curating the detection and classification of dentate spikes (DSs) from hippocampal local field potential (LFP) recordings. The tool may also be used to detect, but not classify, sharp wave-ripples (SPW-Rs). The tool utilizes previously published Python packages for loading LFP files and creating experiment-specific probe objects. Detection and classification parameters are clearly defined and logged in a parameter file before starting processing. Once LFP data has been mapped to the probe object, event detection occurs across all channels. DSs are detected as qualifying peaks in the filtered DS band LFP, while SPW-Rs are detected as qualifying peaks in the filtered ripple band amplitude envelope. An initial curation step allows visualization of the LFP, instantaneous current source density (CSD), and depth-by-frequency band power plots for determining the approximate channel locations of key anatomical regions (i.e., CA1, the hippocampal fissure, and the hilus of the dentate gyrus). The optimal channel for detection is further refined in the next step by comparing event waveforms and quality metrics across channels. Artifacts and noisy waveforms can also be manually excluded during this step. Finally, DSs detected from the optimal channel are classified by computing the CSD profile around events and then clustering the first two principal components of all CSDs. The authors claim that this customizable tool will standardize DS detection and classification.

Strengths:

Toothy's detection and classification algorithms are appropriate and well-validated in the literature. The ability to change many parameters, the CSD calculation method, and clustering algorithm is helpful for precise replication of methodology that has varied previously. Default

parameters optimized for mouse recordings provide a standardized starting point for rodent researchers.

The authors' commitments to transparency and user-friendliness are to be commended (e.g., clear instructions, defined and logged parameters, multiple visualization options, etc.) and are likely to be appreciated by new users. Researchers with little-to-no coding experience should find this tool especially powerful for jumpstarting their own DS analyses.

While not the focus of the paper, the capability to detect SPW-Rs provides an additional use case for Toothy and streamlines simultaneous analysis of SPW-Rs and DSs.

Weaknesses:

I encountered unexpected errors while trying to load LFP data into Toothy for testing, indicating that the "data ingestion" stage of Toothy requires minor code revision.

Toothy's utility for recordings that do not produce an LFP depth profile is unclear. According to the authors, Toothy allows probe designs with irregular spatial sampling (e.g., tetrodes) to be used. However, recording from a linear probe with electrodes spanning from approximately the hippocampal fissure to the hilus of the dentate gyrus is required for Toothy's full functionality. For example, Toothy uses a DS type classification algorithm that relies on sufficiently sampled CSD depth profiles that tetrode recordings cannot provide. As such, usage is currently restricted to detection only for certain recording setups.

The documentation on Toothy's output could be improved. Specifically, the work does not state which files different data are saved to or list the properties saved per detected event. Furthermore, the work does not discuss the potential importance of DS properties that are saved besides those related to the timing of the DS and its type.

https://doi.org/10.7554/eLife.112308.1.sa2

Reviewer #3 (Public review):

Summary:

Esfahany et al present a novel, UI-based tool to detect dentate spikes from hippocampal local field potential recordings, called Toothy. Toothy is easily accessible, compatible with many popular recording formats, and guides users entirely via UI through the dentate spike curation and analysis process. The functional and interactive visualizations enable users to gain a detailed understanding of their data and rigorously analyze dentate spike phenomena. This tool will be broadly useful for anyone who studies hippocampal electrophysiology. Furthermore, by expanding access to dentate spike analysis, it may encourage more scientists to explore this understudied but critical phenomenon.

Strengths:

(1) Toothy provides several ways for users to interact directly with parameters, revealing the ramifications of these choices. Most parameters are adjustable and made obvious via a UI panel. Their effects are then visualized across channels and individual events. This will help users think critically when selecting parameters.

(2) Toothy is fully UI-based and pip-installable, lowering the barrier to entry far below what most electrophysiology analysis tools offer.

(3) The channel selection tool is broadly useful for identifying DG hilus and CA1 pyramidal locations. Since subregional and laminar localization of electrode sites is critical to correctly interpret hippocampal recordings, this tool could be more generally used to identify site locations across the hippocampus.

Weaknesses:

(1) The rationale behind parameter choices is not explained. In order to function "not as a black-box detector", as the authors state, all initial parameter choices should be explained with citations. If possible, these citations would also be available from Toothy directly, alongside citations describing alternative parameter choices. This will help users make informed choices. For instance, a user analyzing data from rats would need to adjust the default ripple frequency band upwards (150-250Hz), and would benefit from guidance to adjust this properly.

(2) The Results describe the functions of Toothy from the perspective of the user, but there is no Methods section describing what Toothy does between UI displays. This would allow readers to compare the tool directly to analysis pipelines as described in the Methods sections from other papers. Particular attention should be paid to justifying the analysis decisions that cannot be changed by the user, such as detecting events off of a single representative channel instead of across a consensus of multiple channels.

(3) It's unclear whether or how Toothy evaluates data quality to confirm that its analyses return interpretable results. At a minimum, the tool should confirm adequate sampling rate (e.g. <=1kHz) and inter-site spacing for CSD (e.g. <=50um).

(4) The paper does not put Toothy into context among the other common open-source electrophysiology analysis toolboxes. Consider Rippl-AI (Navas-Olive & Rubio et al, 2024) or pynapple (Viejo et al, 2023), to give a few examples. The paper would be strengthened by addressing how Toothy extends beyond the capacities of these other tools and how Toothy can be integrated into a workflow that also uses these other tools.

https://doi.org/10.7554/eLife.112308.1.sa1

Author response:

We are very happy that our work was positively received by the reviewers and editors and we are looking forward sharing our results via eLife.

In addition, we have added more details on the generation of the brainbow lines, and we have submitted the respective plasmids to Addgene and give the respective IDs. Some additional minor changes were done to make the text more clear.

We have to confess that we were not completely happy with the term “solid” for the strength of evidence. That assessment seems to be based on the fact that we reached no single cell resolution with our brainbow imaging. Admittedly, this is true and we have openly discussed this technical restriction. However, to our knowledge, within arthropods transgenic single-neuron marking had been restricted to *Drosophila melanogaster*, i.e. the most advanced genetic model system. Our work presents the first generation of respective transgenic lines and the first application of the brainbow system in any arthropod apart from *Drosophila*. Therefore, we tend to consider our methodological approach to go beyond current state of the art. We would also like to point out an additional aspect of our technical approach: We transgenically marked a subset of neural cells expressing a given transcription factor. This allowed us relating projection patterns and the expression of classic neural markers (e.g. neuropeptides) to the expression of a transcription factor (i.e. a molecular subset of neurons). We are not aware of other arthropod studies doing this (apart from flies, of course).

In summary, we wonder whether the chosen level of strength of evidence might be reconsidered from the perspective of emerging model organisms and in comparison with other recent studies published in eLife, where the well-known classic neural markers were used to determine spider brain morphology.

> ***Public Reviews:***
>
> ***Reviewer #1 (Public review):***
>
> *Summary:*
>
> *Pang et al. investigated the expression pattern of the transcription factor foxQ2II in an adult beetle brain. They find nine distinct clusters, with many neurons expressing Glut/ChaT and dopamine. Some of the dopamine neurons resemble cell types described in Drosophila. Several neurons seem to project to prominent higher brain regions such as the MB and CX, and might even connect to both.*
>
> *Strengths:*
>
> *The authors use state-of-the-art labeling techniques for the analysis of individual cell types, such as beetle brainbow, to investigate the until now unknown expression of the transcription factor in the adult beetle brain.*

We would add that this work establishes and introduces brainbow for the first time in an arthropod outside *Drosophila melanogaster* and that we are the first (outside flies) to relate the expression of a neural transcription factor with neural projection and neurotransmitter content.

> *Rigorous cell reconstruction and image analysis revealed a better understanding of the anatomy of the labeled cells.*
>
> *Weaknesses:*
>
> *The brainbow labeling seems to include all cells labeled by the enhancer trap line, as well as the ones not expressing foxQ2II. Thus, it is unclear how useful this data is to compare individual cells to other insects.*

We kindly disagree with the first statement: not all cells of the enhancer trap are labelled but a subset. Therefore, we call it “sparse labelling” in our manuscript while admittedly we do not reach “single cell labelling”, which admittedly limits both precision and use.

> *The functional relevance of this transcription factor in the adult brain cell is still unknown. It is therefore unclear if the described neurons have any specific function and*

> *if they require this transcription factor for normal function.*

Previously, we found that this gene has an important function in neural development during embryogenesis. Actually, we have done extensive RNAi experiments to test for an effect during postembryonic development. We found surprisingly small defects when looking at the projection patterns of several imaging lines. However, we found some changes in behaviour. Given the extensive data presented in the current paper, we decided to publish these functional data (another 12 figures/suppl. figures) separately.

We also note that the identity/function of neurons is determined by a mix of transcription factors. Disentangling the individual role of each of those transcription factors would indeed be an exciting question and a major endeavour.

> *Overall, the neural reconstructions are missing single-neuron details; it is difficult to compare the shown cell types to specific cell types in Drosophila based on the presented data, and this finding remains speculative.*

Indeed, we do not reach single-cell resolution, which is below the standards of fly neurobiology. However, compared with other arthropods we reach a unique level of precision. Specifically, we are able to relate the expression of a transcription factor to neural projection and neurotransmitter content.

We also think that combining our transgenic line with dopamine expression was sufficient to compare the labelled cells to fly neurons. We feel that most homology assessments of neurons across such large evolutionary distances will remain hypothetical to some degree.

> ***Reviewer #2 (Public review):***
>
> *Summary:*
>
> *The authors provide the first thorough profiling of neurons in Tribolium characterized by the expression of the transcription factor foxQ2, which will be useful for developmental neurobiology. They use state-of-the-art methods convincingly to not only identify the neurons, but also to further characterize them anatomically and neurochemically.*
>
> *Strengths:*
>
> *Thorough and meticulous application of state-of-the-art anatomical methods in a nonstandard laboratory organism.*

Thank you for this encouraging comment.

> *Weaknesses:*
>
> *No weaknesses were identified by this reviewer.*
>
> *Comments:*
>
> *I don't really have any major suggestions at all. Loved the work.*
>
> *There is only one tiny nitpicking aspect:*
>
> *P21: "Biogenic amines are involved in learning and memory and setting arousal threshholds (Davis, 2023), which are functions performed by the mushroom bodies and related to the function of the central complex in goal directed navigation, respectively."*
>
> *MBs mainly process olfactory memory. At least in Drosophila, most other kinds of memories are being supported elsewhere.*

https://pubmed.ncbi.nlm.nih.gov/10454381/

such as, e.g., visual pattern learning in the CX

https://pubmed.ncbi.nlm.nih.gov/16452971/

or motor learning in motor neurons

https://pubmed.ncbi.nlm.nih.gov/38779314/

or ventral ganglion, antennal lobes, and median bundle for place learning:

https://pubmed.ncbi.nlm.nih.gov/10706599/

If the authors focus on MBs, this sentence ought to reflect the fact that the function of the MBs is much narrower than the current sentence appears to suggest.

Thanks for this clarification – we have rephrased:

"Biogenic amines are involved in learning and memory and setting arousal thresholds (Davis, 2023). This relates to the mushroom bodies' function in olfactory memory, and the function of the central complex in visual pattern learning and goal-directed navigation, respectively."

https://doi.org/10.7554/eLife.112308.1.sa0