

V-NeuroStack: Open-source 3D time stack software for identifying patterns in neuronal data

Ashwini G. Naik¹ | Robert V. Kenyon¹ | Aynaz Taheri¹ | Tanya Y. BergerWolf² |
Baher A. Ibrahim^{3,4} | Yoshitaka Shinagawa^{3,4} | Daniel A. Llano^{3,4}

¹Department of Computer Science, University of Illinois Chicago, Illinois, Chicago, USA

²Department of Computer Science Engineering, Ohio State University, Columbus, Ohio, USA

³Department of Molecular and Integrative Physiology, University of Illinois Urbana-Champaign, Illinois, Urbana, USA

⁴Beckman Institute for Advanced Science and Technology, Urbana, Illinois, USA

Correspondence

Daniel A. Llano, Department of Molecular and Integrative Physiology, University of Illinois at Urbana-Champaign, Urbana, IL, USA.

Email: d-llano@illinois.edu

Funding information

National Institutes of Health, Grant/Award Number: S10OD023569; National Science Foundation, Grant/Award Number: 1515587

Abstract

Understanding functional correlations between the activities of neuron populations is vital for the analysis of neuronal networks. Analyzing large-scale neuroimaging data obtained from hundreds of neurons simultaneously poses significant visualization challenges. We developed V-NeuroStack, a novel network visualization tool to visualize data obtained using calcium imaging of spontaneous activity of neurons in a mouse brain slice as well as in vivo using two-photon imaging. V-NeuroStack creates 3D time stacks by stacking 2D time frames for a time-series dataset. It provides a web interface to explore and analyze data using both 3D and 2D visualization techniques. Previous attempts to analyze such data have been limited by the tools available to visualize large numbers of correlated activity traces. V-NeuroStack's 3D view is used to explore patterns in dynamic large-scale correlations between neurons over time. The 2D view is used to examine any timestep of interest in greater detail. Furthermore, a dual-line graph provides the ability to explore the raw and first-derivative values of activity from an individual or a functional cluster of neurons. V-NeuroStack can scale to datasets with at least a few thousand temporal snapshots. It can potentially support future advancements in in vitro and in vivo data capturing techniques to bring forth novel hypotheses by allowing unambiguous visualization of massive patterns in neuronal activity data.

KEYWORDS

3D visualization, calcium imaging, dynamic network, neural network, spatio-temporal data

1 | INTRODUCTION

Examination of functional correlations between neurons or brain regions whose correlations vary over time (“dynamic neural networks”) has led to many advancements in neuroscience (Hudson et al., 2014; Liu et al., 2018; Wang et al., 2016). However, even modest-sized networks, for example, 100 neurons, produce nearly 5000 pairwise

correlations when assessed over a single time window. Dynamic network analysis often involves sliding that window for hundreds of frames or more, creating hundreds of thousands of pairwise correlations. Thus, a major challenge in understanding the behavior of dynamic neural networks is visualizing large-scale networks over time. Current applications have not been able to provide an efficient way to view changes in the temporal data beyond a small number of

Edited by Junie Paula Warrington and Eric Michael Prager. Reviewed by Joshua Goldberg and Susan Tappan.

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial-NoDerivs](https://creativecommons.org/licenses/by-nc-nd/4.0/) License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2022 The Authors. *Journal of Neuroscience Research* published by Wiley Periodicals LLC.

nodes. Hence, it has become imperative to find new ways to explore and analyze such complex network structures. Here, we describe our web-based tool that will allow experimenters to visualize many thousands of correlations over time and space. To do this, and to ensure generalizability across *in vitro* and *in vivo* platforms, we used data from two sources: (1) a brain slice imaging experiment that captured the spontaneous activity of a population of cortical neurons and (2) *in vivo* from a population of neurons from the inferior colliculus of an anesthetized mouse exposed to acoustic stimuli. Once the experimenter defined the regions of interest (corresponding to individual neuronal somata), we applied a clustering algorithm that generates clusters of potential functionally correlated neurons. To again ensure broad applicability of the visualization technique, two different clustering algorithms were used. *K*-means clustering, which has been broadly used in the biology to identify networks of interacting biological entities, was applied to the datasets. In addition, a newer algorithm, known as community dynamic analysis (Llano et al., 2021; Tantipathananandh & Berger-Wolf, 2011) was used to infer dynamic communities. These approaches generated clusters that were then used as inputs to our visualization tool for exploration and analysis.

One necessity for analysis is the ability to preserve the information for every timestep or a set of timesteps juxtaposed with a detailed view of a single timestep. Hence, we developed VNeuroStack, a virtual neuron stack created by stacking timesteps for each neuron. In our system, we propose a three-step analysis process to identify patterns in the neuronal activity and to understand functional

correlations between neurons. In the first step, we show an aggregate view of the neuron activity for the entire range of available timesteps. Viewing the NeuroStack in a single color can display large-scale patterns in the activity of the network of neurons. In the next step, we choose a single timestep of interest and use the 2D view for further exploration. This view may display neurons that may be potentially interacting with each other in the chosen timestep. The third step is the viewing of time traces for neurons. This component supports detailed viewing of traces for a single neuron or a group of neurons in its raw and derived forms to examine the dynamic patterns. In addition, there exists a bidirectional connection between the time trace view and the detailed single timestep view. These three components of visualization provide an intuitive and flexible analysis tool for neurobiologists. Figure 1 highlights some of the features of the application and its usage for a single dataset from brain slice spontaneous activity and Figure 2 shows two single-colored clusters with their cluster number and number of data points they contain.

2 | VISUALIZATION APPROACHES

2.1 | Visualization of dynamic networks

Attempts to combat problems that arise while visualizing dynamic networks due to aggregation have inspired novel visualization techniques. Some of these techniques include exploring

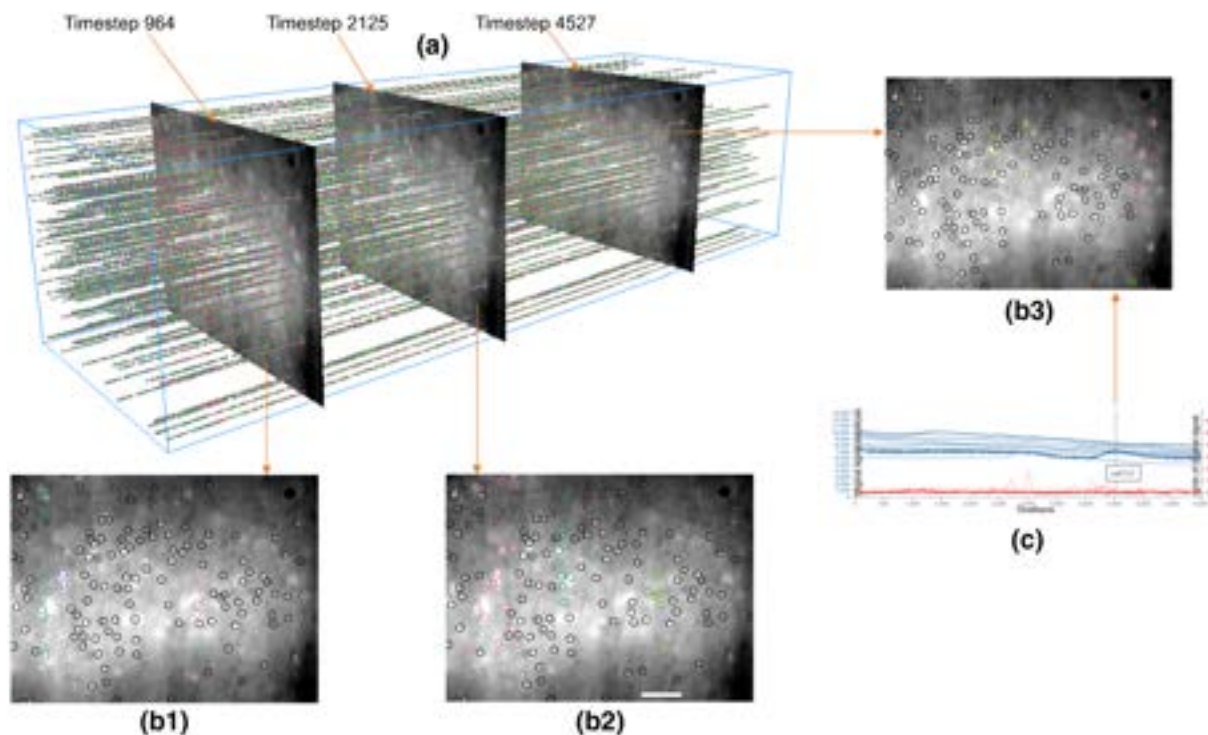


FIGURE 1 Highlights of V-NeuroStack Application applied to *in vitro* data. (a) Shows V-NeuroStack with dataset created using CommDy algorithm (Llano et al., 2021; Tantipathananandh & Berger-Wolf, 2011) with window size 100 frames and correlation coefficient threshold of 0.9. (b1–b3) Show detailed views of timesteps in A with columnar clusters. Scale bar = 100 μ m. (c) Shows dual-line graph which on hover highlights the neuron in the 2D view. Scale bar = 100 μ m.

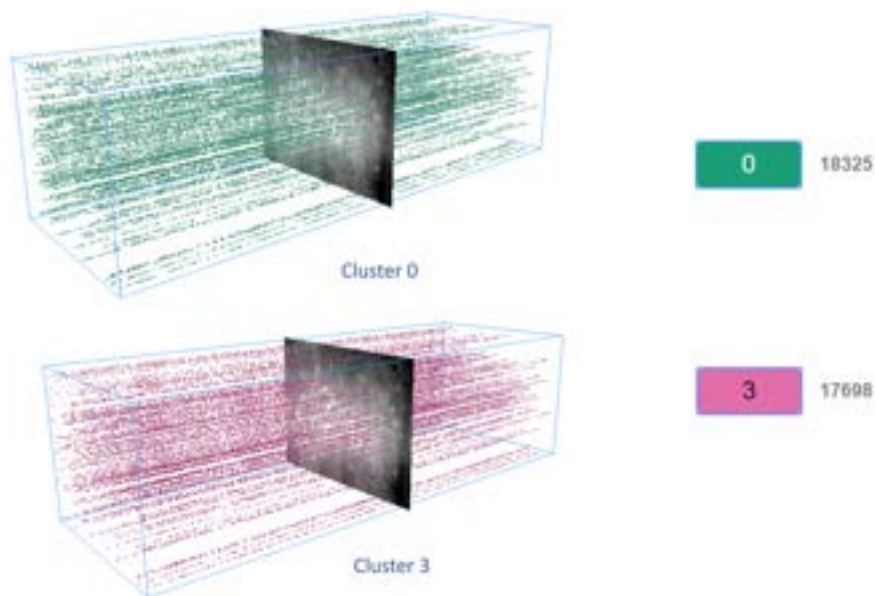


FIGURE 2 Single-colored clusters obtained using CommDy with their cluster from the dataset shown in Figure 1.

network dynamics over time through animation (Bender-deMoll & McFarland, 2006; Ma et al., 2015, 2016, 2018), using abstract spatial components (Bach, Dragicevic, et al., 2014), studying evolving interaction patterns between individuals (Reda et al., 2011) and by reducing snapshots of networks to points in high-dimensional space (van den Elzen et al., 2015). Researchers have also used techniques derived from filmmaking to inform visualizations that combine relationships and perspectives on dynamic networks (Federico et al., 2012).

2.2 | Visualization of brain networks

BrainNet Viewer (Xia et al., 2013), DynamicBC (Liao et al., 2014), EEGNET (Hassan et al., 2015), Connectome visualization utility (CVU) (LaPlante et al., 2014), and eConnectome (He et al., 2011) are visualization utilities implemented as open-source toolboxes to visualize large brain networks. Per the developers, BrainTrawler is a "task-driven, web-based framework that incorporates visual analytics methods to explore heterogeneous neurobiological data" (Ganglberger et al., 2019). Data-driven visualization approaches have been developed to visualize multichannel EEG coherence networks (Ji et al., 2018; Ten Caat et al., 2008). Visual analytics systems have been developed (Al-Awami et al., 2014; Arsiwalla et al., 2015; Bach et al., 2015; Fujiwara et al., 2017; Ganglberger et al., 2020; Vijayalakshmi et al., 2014) and studied (Beck et al., 2017; Chen et al., 2018; Trotts et al., 2007) that enable researchers to explore human brain activity for improved structural understanding of the data.

2.3 | Visualization of calcium imaging data

In the recent years, there has been growing interest among researchers toward making automatic regions of interest

detection and extraction in calcium imaging easier for the community (Weisenburger & Vaziri, 2018). Some of the notable works include MIN1PIPE (Lu et al.,), CAVE (Tegtmeier et al., 2018), and Suite2P (Pachitariu et al., 2017), EZCalcium (Cantu et al., 2020) which incorporates other tools such as CalmAn (Giovannucci et al., 2019) and NoRMCorre (Pnevmatikakis & Giovannucci, 2017). TENASPIS (Kinsky et al., 2020) focuses on calcium imaging in freely moving mice to investigate the emergence and fate of trajectory-dependent activity through learning and mastery of a continuous spatial alternation task. Inscopix' Mosaic ([inscopix.com](https://www.inscopix.com)) and Doric Studio (<https://neuro.doriclenses.com/products/doric-neuroscience-studio>) are commercially available applications for automatic detection, extraction, and refinement of ROIs. Pnevmatikakis in their review (2019) provide a comprehensive list of recent methods for addressing pre-processing problems that arise in calcium imaging data analysis, and available software tools for high-throughput analysis pipelines.

2.4 | Visualization of temporal networks

Email and social networks have been visualized to reveal communication patterns and have explored temporal changes in activities and interests of users (Ahn et al., 2011; Fu et al., 2007; Itoh et al., 2011, 2012). These methods are more suited for visualizing networks of known direct connections between nodes. Space-time cube operation-based visualizations have been used to visualize spatio-temporal data, as described in (Bach, Dragicevic, et al., 2014; Bach, Pietriga, et al., 2014; Gatalsky et al., 2004; Wagner Filho et al., 2019).

The techniques described above do not simultaneously take into account the representation of the spatial components of the data, data sample and extraction of raw signals, or the complexity and size

of data. None of the above-mentioned works report visualization and analysis of networks of neurons post the ROI generation and refinement stage. These techniques are also not scalable to thousands of timesteps while providing the ability to view multiple timeframes of the slice in an aggregated fashion. The V-NeuroStack application provides the researchers with consistent spatial context for comparison of orientation and mobility of neurons between potential neuronal clusters.

2.5 | Significance statement

V-NeuroStack uses a holistic approach by combining two visualization paradigms: (1) Space-time cube and (2) Two-dimensional networks. Additionally, the application also provides interactive capabilities such as the ability to view different clusters by varying parameters, generating time series of identified regions of interest and bidirectional communication between individual components of the visualization. The novel contributions of V-NeuroStack are:

- A unified flexible solution to readily identify patterns in spatio-temporal network datasets.
- An interactive visualization for analyzing cluster-specific individual or multiple neurons.
- Visualization of topology changes over time with respect to functional connectivity.

3 | METHODS AND RESEARCH GOALS

Calcium imaging is a commonly used technique to measure neuronal activity (Grienberger & Konnerth, 2012). Data for this study were generated by imaging calcium signals *in vitro* from a brain slice using GCaMP6s, or *in vivo* from the inferior colliculus using a red-shifted calcium indicator (jRGECO1a). All procedures involving animals were approved by the Institutional Animal Care and Use Committee (IACUC) at the University of Illinois in Urbana-Champaign. The animal facilities at the University of Illinois have been approved by the Association for Assessment and Accreditation of Laboratory Animal Care International (AAALAC). The datasets used in this manuscript are available at: <https://github.com/anaik12/bvis/tree/master/point/data>, and <https://github.com/anaik12/bvis/tree/master/point/invivo/data>.

3.1 | Animal preparation

To obtain brain slice data, a 2-month-old male mouse expressing GCaMP6s on a Thy1 promoter, bred in-house with wild-type C57Bl/J mice and originated from Jackson laboratory (stock No: 024275) was used. Four hundred micrometer thick coronal brain slices through the motor cortex, as previously described (Ibrahim et al., 2017), were exposed to a low concentration of a bath-applied

GABA_A blocker (GABA_Azine, 100 nM) to induce spiking activity. The brain slice was imaged using a U-M49002XI E-GFP Olympus filter cube set (excitation: 470–490 nm, dichroic 505 nm, emission 515 nm long pass) at a magnification of 10× which allowed the capture of a 550 by 750 μm region of the cortex. Images were captured using a Retiga EXi camera and StreamPix software at 10 frames per second. Using this approach, we were able to capture the spontaneous activity of 139 neurons over a 600 s period (10 min). Motion correction was not needed for *in vitro* data. Using ImageJ software (<https://imagej.nih.gov/ij/>), the z-projection was used to compute one single image representing either the sum, the standard deviation, or the median of all the image sequences in the movie. Based on these single images, the region of interest (ROI) was manually drawn around each detectable cell body for both the brain slice and *in vivo* datasets.

For *in vivo* data, a female Tg(Thy1-jRGECO1a)GP8.20Dkim/J mouse bred in-house with wild-type C57Bl/J mice and originated from (Jackson Laboratory, Stock# 030525) of 14 weeks of age was used. A craniotomy was designed to image the dorsal surface of the inferior colliculus (IC). Before surgery, mice were anesthetized with a (ketamine hydrochloride at 100 mg/kg, xylazine at 3 mg/kg and acepromazine at 2 mg/kg) mixture. The anesthesia was maintained during the surgery and imaging using only ketamine (100 mg/kg). To prevent neural edema during or after the craniotomy, an intramuscular injection of dexamethasone sodium (4.8 mg/kg) was given just before the surgery using an insulin syringe. After placing the animal in the stereotaxic apparatus (David Kopf Instruments, USA), both eyes were protected by applying Opti care lubricant eye gel (Aventix Animal Health, Canada). The hair of the scalp was then removed by massaging the scalp with a depilatory cream (Nair) and leaving the cream on the scalp for 4–5 min. The cream was then removed by a thin plastic sheet. The medial incision was made with a scalpel blade #10, and 0.2 ml of 0.5% lidocaine was injected intradermally in the scalp. The skin was completely removed using a pair of micro scissors. The remaining dried or firmly attached pieces of periosteum were removed with a scalpel blade #10. Using the stereotaxic apparatus, a wide area of ~3×4 mm above the left IC was made. A micro drill bit (Size #80, Grainger, USA) was used to drill through the skull starting from the rostralateral region to lambda. To prevent overheating of the superficial brain structures, an ice-cold sterile saline was used to intermittently irrigate the surface. A stream of a pressurized air was also applied also during the drilling procedures to prevent overheating and to remove the debris produced by the drilling. After drilling, the skull was irrigated in a sterile saline and the bone flap (the undrilled bone over the craniotomy area) was gently examined for the complete separation from the rest of the skull. Using a pair of no. 5/45 forceps, the bone flap was gently removed. The dura was carefully and gently lifted after piercing it, and a pair of no. 5/45 forceps were used to grab the dura to gently tear it to the extent of the transverse sinus to avoid bleeding. A pair of forceps was then used to gently place a sterile 5 mm cover glass #1 (Thomas Scientific, USA) over the skull covering the opened area above the IC. Then, the gap between the surface of the IC and the cover glass

was filled with 1% agarose gel that was made in saline at a temperature range of 33–35°C. The extra agarose gel pieces were cut off and removed by the scalpel blade. A titanium headpost was glued carefully on the top of the skull to be at the same level of the cover glass. Following the manufacturer instructions, the C&B Metabond (Parkell, Japan) was used to secure the headpost in its place.

3.2 | Two-photon imaging and acoustic stimulation

Immediately after surgery, the anesthetized animal was taken and secured under the microscope objective by clamping the arms of the head post to two perpendicular metal post mounted on a custom-built two-photon microscope. The imaging of the dorsal surface of the IC was made using a 20× water-immersion objective (LUMPlanFI/IR, 20×, NA: 0.95, WD: 2mm; Olympus Corporation, Tokyo, Japan) using an excitation light at 1040nm, which was generated by InSight X3 laser (Spectra Physics Lasers, Mountain View, CA, USA). A layer of a 1:1 mixture of multipurpose wavelenghts ultrasound gel (National therapy, Canada) with double deionized water was used to immerse the objective. The emitted signals were detected by a photomultiplier tube (Hamamatsu H7422PA-4, Japan) following a t565lp dichroic and a Chroma barrier et595/50m filter. Images (202×191 pixels) were collected at a frame rate of 6.49 Hz in Galvo mode. A movie of tiff images was collected to track either the spontaneous or evoked neuronal activity. Therefore, the movie was started by recording the neuronal activity during 5s of silence (pre-stimulation period), 10s of acoustic stimulation, and approximately 115 s of period of silence (post-stimulation period). The acoustic stimulation paradigm is composed of five stimulations (Duration: 500ms and ISI: 2000ms) of broadband noise at 80dB SPL. The sound stimulus was generated by a TDT RP2.1 processor (Tucker-Davis Technologies, US) and delivered by a TDT ES1 speaker (Tucker-Davis Technologies, US) using a custom-made MATLAB (The MathWorks, Natick, MA, USA) code. The frame timing of the scanner and the sound stimuli were both digitized and time-locked using a Digidata 1440A (Molecular Devices, Sunnyvale, CA, USA) with Clampex v. 10.3 (Molecular Devices, Sunnyvale, CA, USA).

3.3 | Motion and neuropil correction of in vivo data

The folder containing the images was imported into Fiji (<https://imagej.net/software/fiji/>) using the XML file included in the dataset. Motion correction was applied using the Image Stabilizer plugin (https://www.cs.cmu.edu/~kangli/code/Image_Stabilizer.html) with default values, except for outputting to a new stack, and the images were exported as an image sequence of tiff files. These were then imported into Python using the `imread` function from the OpenCv library using the grayscale mode, and converted to a numpy array. The ROI set was imported using the `read_roi_zip` function from the `read_roi` library. The ROI set was then used to create two masks, one that is a replica of the ROIs, and a second surrounding the original

ROI, roughly four times larger in area. The smaller mask was applied to the numpy array to find the average pixel value within the ROI, while the larger mask was applied to find the average pixel value of the neuropil. Neuropil correction was applied using the equation $\text{corrected_value} = \text{data_value} - 0.4 * \text{neuropil_value}$ (Akerboom et al., 2012). The average pixel values for each cell at each time point were exported to an Excel spreadsheet using the pandas library.3.4 Research Goals.

Here we attempt to understand the functional organization of brain networks at the resolution of individual neurons. To do this, we employed clustering algorithms—either K-means clustering or CommDy—to identify dynamic clusters of neurons whose functional connectivity changes over time. The visualization challenge is that the topology of the functional connectivity, the clustering, as well as the neuron membership in the clustering, change over time. This type of complex dataset is becoming routine with large number of neurons imaged over time. However, in previous efforts to visualize dynamic clusters (Tanzi, 2017), the users were unable to view clusters for more than a single timestep. With this context, we set the following goals for our application:

- Research Goal 1: To provide the ability to find patterns in the neuronal activity that exhibit correlations between neurons.
- Research Goal 2: To provide the ability to view multiple time-steps/entire dataset at once.
- Research Goal 3: To maintain clarity by avoiding visual clutter when visualizing the data with multiple attributes.
- Research Goal 4: Reduction in the load time of the dataset and making the visualization faster to promote efficient and faster analyses.

Our design successfully achieves the first three goals and partially achieves the last one. Using V-NeuroStack, we can identify patterns in the dataset and determine how communities differ from each other and evolve at different timesteps for a chosen dataset as it provides the ability to preserve multiple timesteps or the entire dataset throughout the analysis.

4 | DATA PROCESSING

The data were generated using calcium imaging from spontaneous and evoked activity of calcium-indicator-expressing excitatory neurons from adult mice. Data were obtained in the form of TIFF files and time traces of the cells were extracted using ImageJ (<https://imagej.nih.gov/ij/>). These data contained the fluorescence intensity of each neuron at each timestep. We use the following files in our application.:

1. Time traces of the neurons (csv format)
2. First derivative of the time traces (csv format)
3. File containing X, Y, and Z for each point, intensities and communities' information (csv format)

- A composite image of the mouse brain slice or surface of mouse brain, generated from z-stack (third chapter in Tanzi, 2017). (png format, could be any image format)

4.1 | Data extraction analysis and visualization pipeline

The brain slice data were recorded for 600.4 s at 10 frames per second using a single photon epifluorescence microscope, providing a total of 6004 frames. The in vivo data were recorded for 109 s at 6.49 frames per second using a two-photon microscope. We note that the number of frames input to the application is arbitrary and can be easily adjusted by the user. Data extraction comprised five steps: capturing images from tissue preparation, composed image generation, labeling and outlining of ROIs (individual somata), extraction of centroid coordinates from the ROI, and activity time course extraction (Tanzi, 2017). The time series of spontaneous calcium signals were then processed using either a *K*-means clustering or a dynamic community algorithm discussed in (Llano et al., 2021; Tantipathananandh & Berger-Wolf, 2011) which identifies dynamically organized clusters of neurons. These clusters are then visualized using the V-NeuroStack application. In our pipeline we have also filtered the time traces to diminish noise in the data. Both the raw data and the first time derivative ($y' = dy/dt$) of the data were analyzed. A three-point differentiation algorithm was used and the values were smoothed with a second-order polynomial. The first time derivative was used to help accentuate rapid change in the calcium signal. y' values of the neurons were used to compute Pearson Correlation coefficients between all possible pairs of neurons for sliding windows across timesteps. Figure 3 shows the data extraction and visualization pipeline used for our analysis. Time lags were not considered in the current iteration of the algorithm but will be considered in future iterations.

These correlation matrices were then used in the clustering algorithms to generate functional clusters ("communities" under the CommDy rubric) of neurons. A community is a group of neurons having correlated activity over time. CommDy allows both the membership in the communities, as well as the communities themselves to change over time. Other dynamic network clustering algorithms can be used (Rossetti & Cazabet, 2018). Based on the experience with CommDy algorithm in the past in which a range of window sizes were explored (Ma et al., 2015; Reda et al., 2011; Tanzi, 2017), window sizes of 100 and 300 frames (10 and 30 s, respectively) were used to generate the dynamic clusters. The generated dynamic clustering data, along with time courses of the individual cellular calcium traces and the composite image, were then used for the visualization. We add a person-in-the-loop option to the pipeline after the final step to allow the user to vary the choice of the correlation coefficient that generates the network of functional connections and the sliding window size to permit the user to visualize communities identified using other analysis parameters. For *K*-means clustering, we extracted filtered data from csv files, normalized data using min_max scaler (any normalization technique may be used), and generated five clusters for the entire dataset (Jupyter notebook at - <https://github.com/anaik12/bvis/blob/master/K-Means.ipynb>). We then created the following tasks for our application development:

- Add time axis to the 2D spatial view to generate 3D time stack of neurons.
- Provide visual information about each timestep while preserving the data for multiple timesteps.
- Link and connect different parts of the visualization.
- Efficiently store and retrieve point cloud information to minimize loading time during dataset switches.

These tasks translate our project goals into design goals that effectively aid the process of application development. The proposed

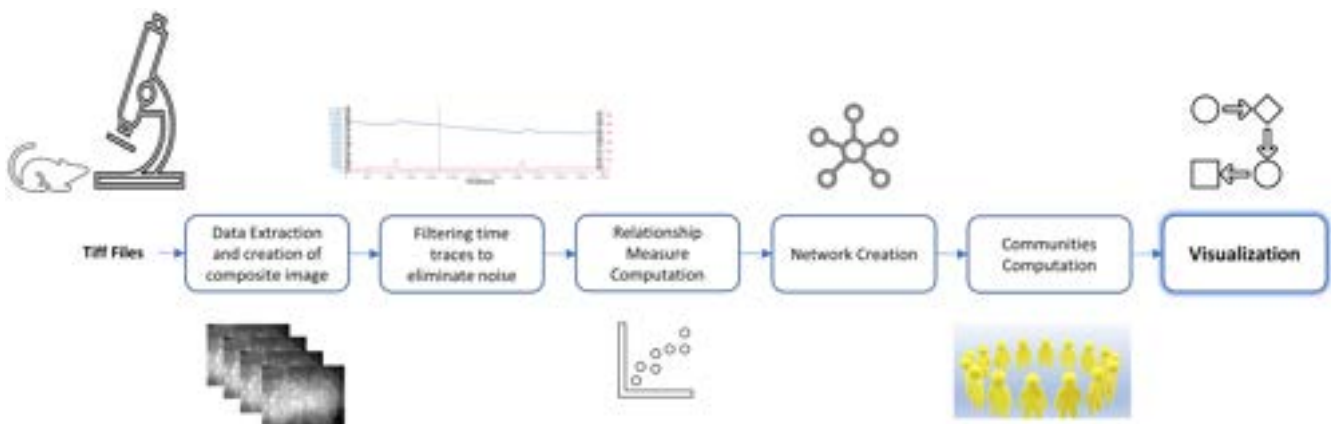


FIGURE 3 Data extraction analysis and visualization pipeline. The input to the pipeline is a movie of 6004 Tiff images. A composite image is created from the movie which acts as a backdrop for the visualization. This step is followed by correlation coefficient computation between each pair of neurons. We then create a network with neurons as nodes and the correlation coefficient between them as edges. This network is then fed to a *K*-means clustering or CommDy algorithm (Llano et al., 2021; Tantipathananandh & Berger-Wolf, 2011) for the network analysis. This is followed by visualizing the networks using V-NeuroStack.

design uses a combination of time cutting, juxtaposition, superimposition, three-dimensional viewing, aggregation, and animation techniques in our visualization.

5 | SOFTWARE DESCRIPTION

V-NeuroStack is divided into three main parts (Figure 4): a 3D point cloud to view the selected dataset in its entirety, a 2D view of a single timestep and a dual-line graph showing raw and first derivative of activity of the neurons for a period of 6004 timesteps. A computer mouse can be used to pan, zoom and rotate the point cloud in three-dimensional space. Demonstrations of the software can be found at:

In vitro data visualization—<https://anaik12.github.io/bvis/point/>.

In vivo data visualization—<https://anaik12.github.io/bvis/point/invivo/>.

All files can be found at: <https://github.com/anaik12/bvis/tree/master/point>.

Application source can be found at: <https://github.com/anaik12/bvis/tree/master/point/src>.

Python script can be accessed at: <https://github.com/anaik12/bvis/blob/master/convertRowsToColumns.ipynb>.

A tutorial video can be found at: https://drive.google.com/file/d/1ANG_95oskqRhhtbxdccyyiwD-9AtBsbUG/view?usp=sharing.

5.1 | Implementation

Our application was intended to be platform-agnostic to alleviate the overhead of software setup and installation on a computer, thus facilitating its use to a broader audience. Creating the tool as a web application would render it easily accessible through any web browser. We used Python (van Rossum, 1995) for initial processing of the data to transform it to the format required for the point cloud in the database. The Web application is built using JavaScript (Eich & McKinney, 1996). The same X and Y coordinates of each neuron are used at each timestep to generate a point cloud. We generate a new Z value (depth of the NeuroStack) for every timestep. This is an essential step in generating the point cloud. This step results in the tubular structure for each neuron in the point cloud. The web application is built in JavaScript mainly using two libraries. The 3D part on the left (Figure 4a) is built using Three.js (Danchilla, 2012) and the 2D part (Figure 4b) is built using D3.js (Michael et al., 2011). We used bootstrap (Michael et al., 2011) framework for better aesthetics in the application. The system currently accepts input in csv format.

5.2 | 3D point cloud

For every dataset in V-NeuroStack, we can generate a 3D point cloud where we view 2D data in three dimensions by adding time

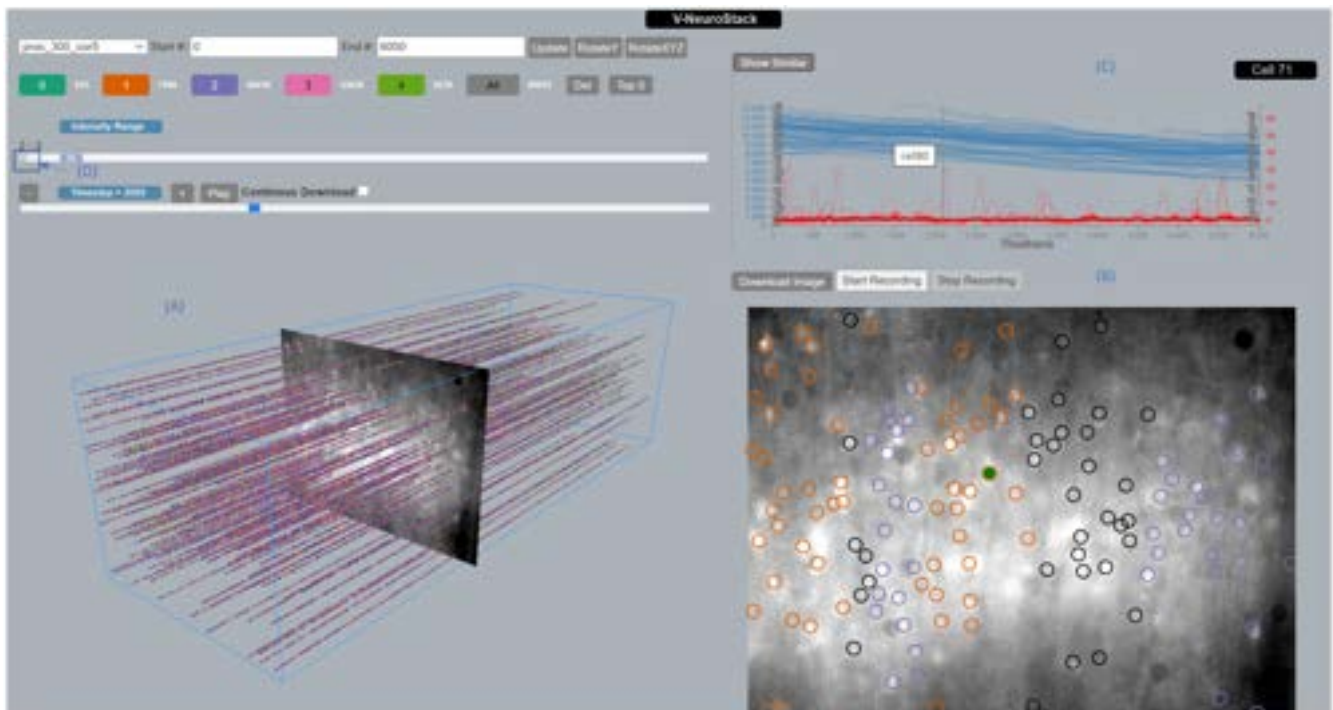


FIGURE 4 V-NeuroStack application with a loaded sample dataset from brain slice data. (a) The 3D point cloud on the left is rotated to view all available timesteps for the dataset. (b) On the right of the image a 2D detailed view of timestep 2095 is shown. (c) We also see a dual-line graph showing the original and first derivative of intensities for all neurons similar to cell number 71 at timestep 2095. Hovering over the line graph for cell number 80 highlights it in the 2D view by filling it in green. (d) Triangular button in the intensity range slider to play/pause the animation.

along the Z axis. The community datasets were generated using the CommDy Algorithm (Llano et al., 2021; Tantipathananandh & Berger-Wolf, 2011). Three-dimensional point clouds may also be generated for raw and the first derivative of intensities (Figure 8). In the visualization for first derivative of intensities we only render data points with positive values (since a negative value for intensities would not be valid). We can also use the intensity range filter to view the entire virtual NeuroStack for a desired range of intensity values. The triangular Play button (Figure 4d) on the slider may be used to run the animation for the entire dataset or by preselecting the range by moving the markers on the slider. Datasets were created from neuronal activity by using various window sizes and correlation coefficients. Each window size and correlation coefficient produced a set of clusters that were then represented by individual colors. These colors do not have precedence over each other and are only a way to distinguish each cluster. These colors are assigned based on the qualitative color scheme for five categories (Harrower & Brewer, 2003). The neurons that do not belong to any cluster are colored black and appear only in the 2D visualization. A user may click on the individual color labels to see neurons belonging to a single color (cluster). We also present the number of neurons belonging to the cluster next to the colored label. In our figures, we show the top five clusters from our dataset (Top5 button) since more than five could lead to visual clutter. However, we can also switch between viewing Top5 clusters or clusters numbered 0–4. The switching feature enables users to focus on individual clusters and get a clear view of particular subparts of their data. For datasets based on the first derivative of the time trace (proc-300-cor5 to proc-100-cor9) data points were divided into more than five clusters. The names of the datasets show parameters used to generate the dataset, with proc referring to processed intensities or first derivative of intensities, number(100/300) being the window size in points, cor being the correlation coefficient and the number after cor used to denote a tenth of the number (cor5 = >correlation coefficient 0.5). A data point is a neuron at any given timestep. Total Data points = Total # of Neurons (Total # of TS – Window Size). Additionally, users can also choose to view only a section of the dataset by using the clipping feature. Users enter the beginning and ending frame numbers then click on the Update button to view this selection in the point cloud. The mouse can be used for panning, zooming and rotating the point cloud. Additionally, we also have buttons that automatically rotate the point cloud on the Y axis or a combination of X, Y, and Z axes. This autorotation can be used to provide enhanced 3D viewing of the point cloud by generating motion parallax.

5.3 | Detailed 2D view

The 2D view consists of all of the neurons which are the identified regions of interest. We mark each of the neurons with a circle. These neurons showed evidence of activity during the entire period of the image capturing process. Neurons whose signal intensities were of

high magnitude and had unvarying time courses throughout the imaging session likely represent unhealthy neurons unable to extrude calcium and were not further analyzed. After generating a point cloud by selecting a dataset from the drop-down menu we notice a 2D view on the right. The timestep slider is at the center of the stack at timestep 3003 when the point cloud is first generated. However, we can move the slider anywhere within the current neuro stack to gain detailed 2D view at that timestep. Alternatively, one could also choose to run the movie by clicking on the Play button. The movie could be paused at any time by clicking the Pause button. For precision selection one could also use the + and – buttons to view a single timestep of interest. Hovering on the neurons in the 2D view highlights the neurons by filling it in red and gives its neuron number.

5.4 | Visualization of the signals

We use a dual-line graph to visualize the raw values and first derivatives of the traces. The Y axis on the left (with blue markings) is used for the raw intensities. The Y axis on the right (with red markings) is used for the first derivative of the intensities. These axis values range between 0 and 68 in arbitrary units generated by Origin software (<https://www.originlab.com/>). The application is programmed to show only the positive values of the first derivative. The X axis represents time as the number of timesteps. When a user clicks on a neuron in the 2D view, a line graph which provides a trace of the neuron in blue and its first derivative in red is generated. The cell number appears on the top right corner. If the user would like to generate traces for neurons belonging to the same cluster, the user clicks on the Show Similar button. This, in turn, generates the raw and first derivative of the traces for all the neurons of the same color on the 2D view. Hovering on any raw trace highlights its corresponding first derivative and vice versa together with the location of the neuron in the 2D view in green. This feature enables a bidirectional connection between the line graph and the 2D view thereby providing flexibility to the users during visualization. The user can also obtain the number of the neuron by hovering on the individual trace. Hovering on a trace lightens the rest of the traces and highlights only the trace that the user is interested in. A vertical line is included on the line graph showing the timestep of the 2D view. This line informs the user of the current timestep for the neuron of interest.

5.5 | Animation components

There are three animation components in the application.

1. Viewing the changing point cloud for the first derivative of neuronal activity using the triangular button on the intensity range slider (Figure 4d)
2. Viewing the communities for each timestep in the 2D view along with a moving image marking the current timestep in the 3D view using the Play button above the timestep slider.

- Automatic rotation of the 3D stack in Y and XYZ directions using the buttons provided. These components assist the users by speeding up the process of exploration and analysis.

5.6 | Record and save options

V-NeuroStack has two features to capture and save any interesting observations/results from the applications.

- The user can download and store any 2D detailed view of interest currently displayed in that section for later reference (png format) by clicking on the Download Image button.
- The user can continuously download and store 2D detailed images by checking the Continuous Download checkbox while playing the 2D visualization.

6 | RESULTS

The V-NeuroStack application was used to gain insights on the neuronal activity time-series datasets from both brain slices from the mouse motor cortex and from the mouse inferior colliculus in vivo. These results uncover the usability of the features and make it evident that V-NeuroStack can be used for effective visualization of time-series data along with a spatial component. We used a three-step process for analysis of these data. The 3D view helps to bring about patterns in the entire length of dataset. However, the point cloud view is not sufficient to provide valuable insight/s into the data. A detailed view of any of the timesteps of interest can then be used to get a better understanding of the data. The interactivity features provide the users with the freedom to explore the data and find similarities and differences between each dataset.

6.1 | Identification of patterns in the 3D point cloud

3D point clouds exhibited patterns in both sets of data which would not be easily visible using a simple 2D view. Figure 5 shows a clipped sample (dataset generated with raw signals with window size of 100 frames, or 10s) from the brain slice dataset. Each bar-like structure represents a neuron for the chosen number of timesteps. The green and orange clusters (Figure 5a) are heavily populated whereas the pink and the blue clusters (Figure 5b) are sparsely populated. In the pink cluster it is observed that there is little to no activity in the first 2000 timesteps. However, between 2500 and 3000 timesteps we find a few dozen neurons belonging to the pink cluster. Between 3783 and 3918 timesteps there are no neurons belonging to the blue cluster. These patterns highlight two major characteristics of the dataset.

- Neurons may have a higher likelihood of membership in some clusters while their appearance in other clusters could be negligible.
- Neurons may not visit a cluster for a period of time while making a return after a certain number of timesteps.

The user may view the dataset in two different modes: Top5 largest clusters and clusters numbered 0–4, to aid the cluster viewing process. The users may rotate, pan, and zoom the point cloud for better exploration. Viewing such patterns in a brain slice exposed to a drug or other manipulation could provide insights about alterations in network connectivity in response to that perturbation. Future work using V-NeuroStack to visualize in vivo data might potentially help us identify distinct sets of functionally correlated neurons while a mouse is engaging in a behavior.

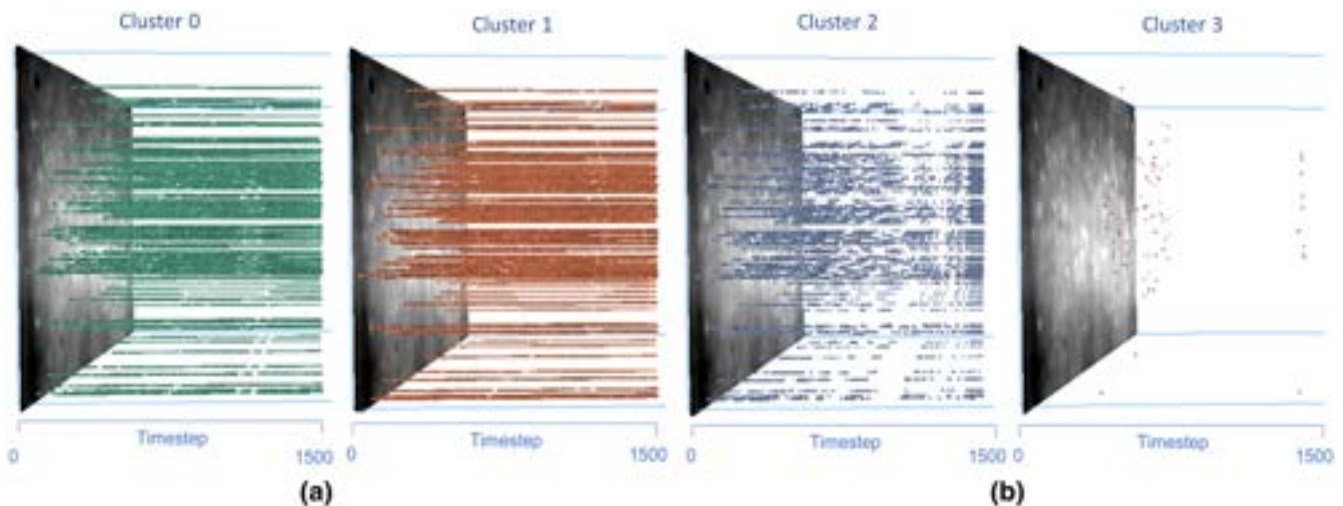


FIGURE 5 Neuronal Activity for clipped dataset from brain slice data. (a) Neuronal activity for first 1500 timesteps of a sample dataset for green and orange color communities which have 92,811 and 96,600 data points, respectively. (b) Neuronal activity for first 1500 timesteps of a sample dataset for blue and pink color communities which have 19,006 and 83 data points, respectively.

6.2 | Columnar clusters in 2D view

Existence of the columnar clusters in some of the timesteps implies that those neurons could potentially be functionally correlated. This finding is in line with the foundational principle of neocortical organization that neurons in a column show strong connectivity (Hubel & Wiesel, 1962; Mountcastle, 1997). Figure 6 shows a columnar cluster in orange suggesting that these neurons could be potentially functionally correlated. Running continuous animation by using the Play button on any dataset can help users find such columnar clusters in the 2D view.

6.3 | Consistency of clusters across timesteps and clustering methods

Comparing timesteps within a single dataset can potentially bring out the most consistent group of neurons that interact together. Figure 7a shows snapshots for in vitro dataset. Clusters

in Figure 7a are obtained using the first derivative of the image data and a correlation coefficient of 0.9, revealing columnar clustering of functional connectivity using CommDy. Similar columnar functional networks were also observed using K-means clustering (Figure 7b). The appearance of columnar clusters in multiple timesteps across different clustering methods further supports our expectation that this application is a versatile and useful visualization tool.

6.4 | Identification of areas of peak activity in the data

V-NeuroStack provides the ability to view original data and the filtered data in grayscale. This dataset is the one available before it goes through the rest of the data analysis pipeline. Figure 8 shows the difference between raw and filtered time traces for brain slice data. Visualizing the first derivative of the raw signals and filtering out data points with negative values helps in eliminating noise

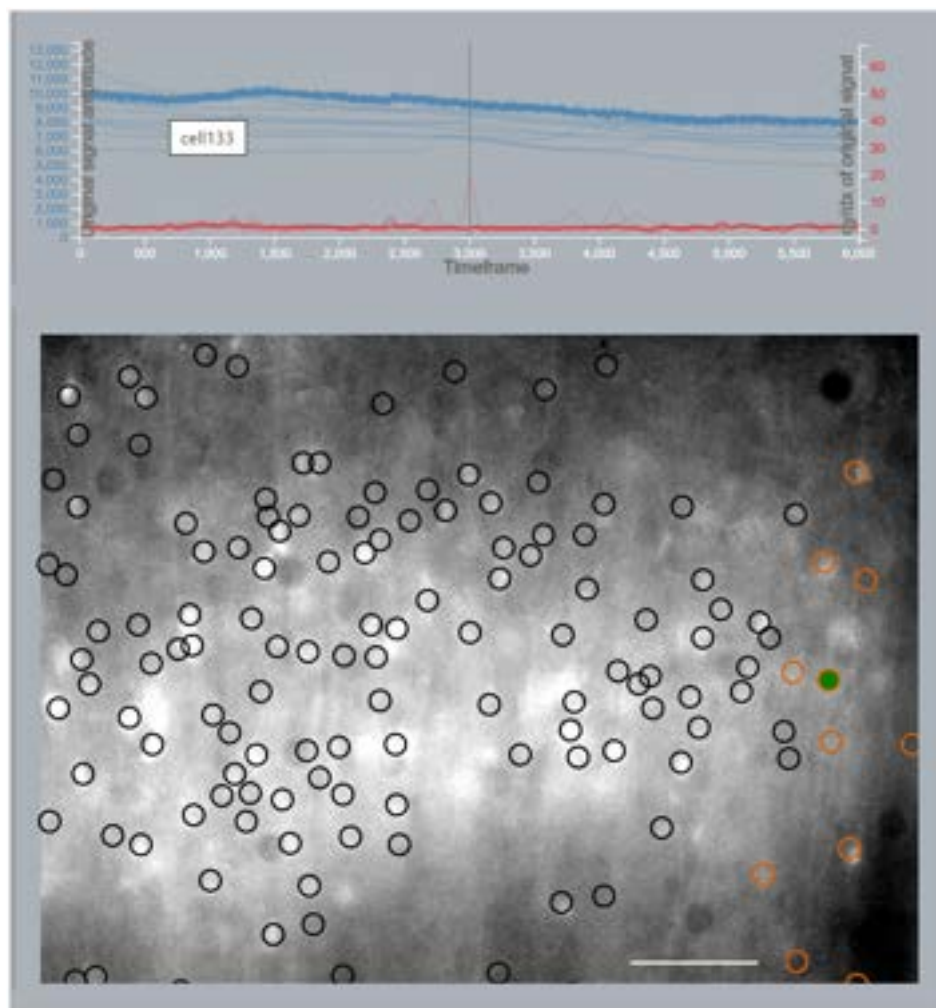


FIGURE 6 Columnar cluster with raw and first derivative of traces. We see an orange columnar cluster in the bottom 2D view. This finding of columnar functional connectivity is in line with the principle of columnar organization of the neocortex (Hubel & Wiesel, 1962; Mountcastle, 1997). We also see that hovering on trace for neuron number 133 highlights it in green in the 2D view. Scale bar = 100 μm .

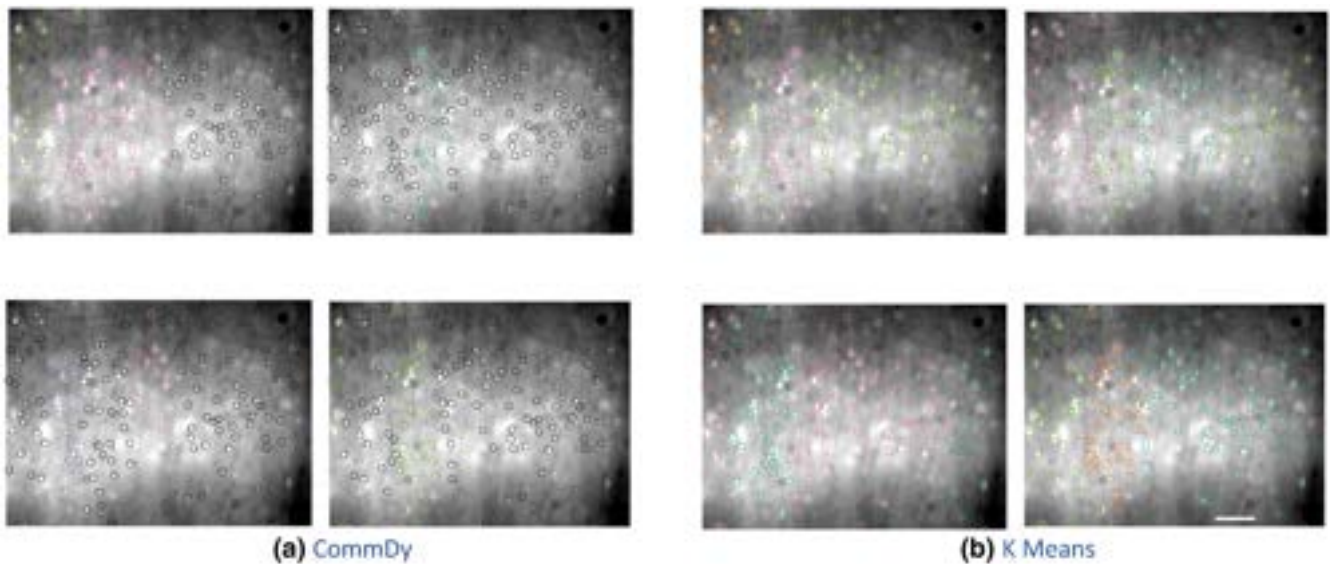


FIGURE 7 Representative snapshots of dataset using CommDy (a) and *K*-means clustering (b) algorithms showing vertically oriented clusters of cortical neurons resembling cortical columns. Scale bar = 100µm.

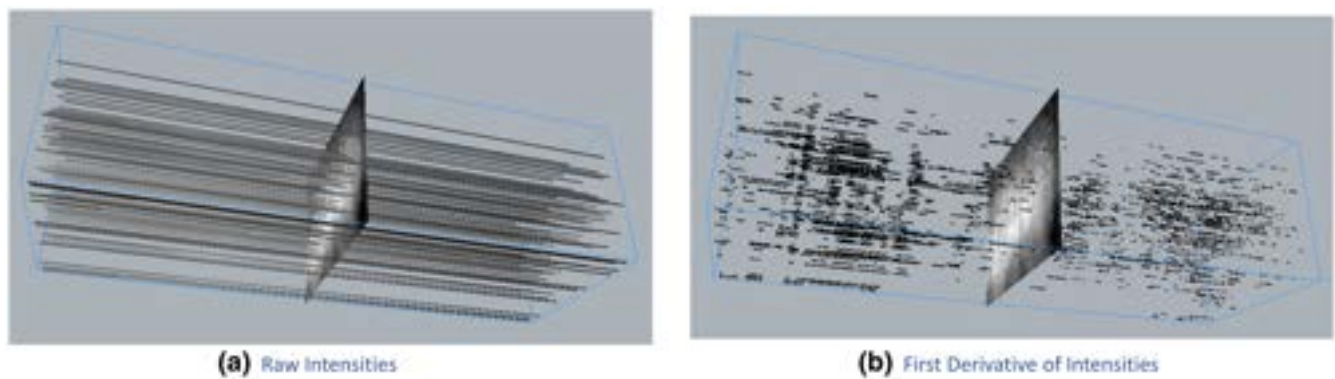


FIGURE 8 Difference between raw and filtered intensities for in vitro dataset. (a) V-NeuroStack with original time trace for 600 s with values ranging between 4715.9 and 14,015. (b) V-NeuroStack with first derivative of time trace after eliminating negative values for 600 s with values ranging between 0 and 68.

in the data and brings us closer to determining true peaks in the spiking patterns. We use grayscale values to visualize this time trace dataset. A darker data point implies it has a lower peak value and a brighter data point corresponds to a higher peak value. The grayscale values of 0.0 to 1.0 (with 0.0 being black and 1.0 being white) are mapped to 4715 to 14,015 for raw intensities and 0 to 68 for filtered intensities. Similar analysis was done using in vivo two-photon imaging data, revealing a series of peaks toward the start of the time series corresponding to the presentation of pure tone stimuli to the animal (Figure 9). Using this approach to view peaks in the filtered intensities would help us determine a good filtering algorithm to be used in our analysis pipeline and can be used as an intermediate step before further proceeding into applying algorithms to generate clusters. These results help in determining the efficacy of our application and creating effective visualizations for analysis.

7 | DISCUSSION, COMPARISONS, LIMITATIONS, AND FUTURE WORK

There are many possible future directions for this application. For example, this tool can be used for in vivo two-photon imaging datasets in behaving animals to understand the correlation between network activity in cortical neurons and behavior. Such analyses may identify functional clusters that assemble during particular behavioral tasks or could identify if abnormal clustering occurs during pathological states. The application currently contains a preprocessed dataset which consists of multiple sub-datasets for individual communities' computation along with raw and first derivative of time traces. Although the application is interactive, it may take between 6 and 10s for the initial data load in case of in vitro dataset, which contains over 6000 timesteps and more than 100 neurons. Less time would be needed for smaller datasets. While visualizing traces for cluster of

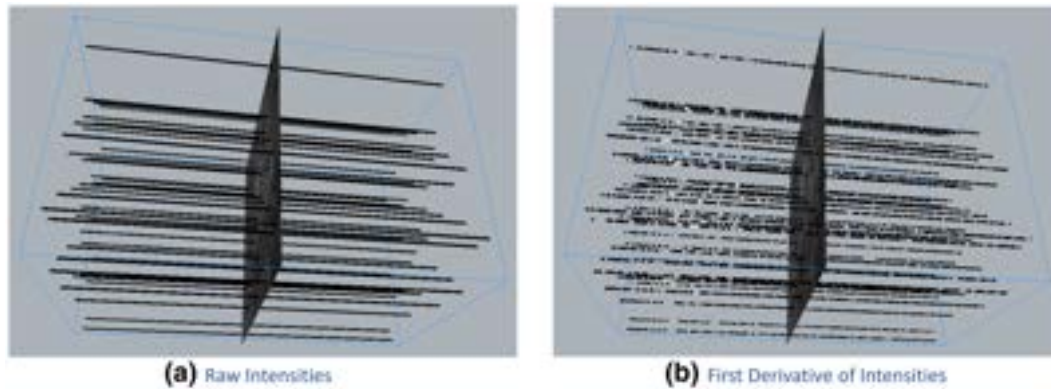


FIGURE 9 Difference between raw and filtered intensities for in vivo dataset. (a) V-NeuroStack with original time trace for 70 s with values ranging between 492 and 1986. (b) V-NeuroStack with first derivative of time trace for 70 s with values ranging between 0 and 276.

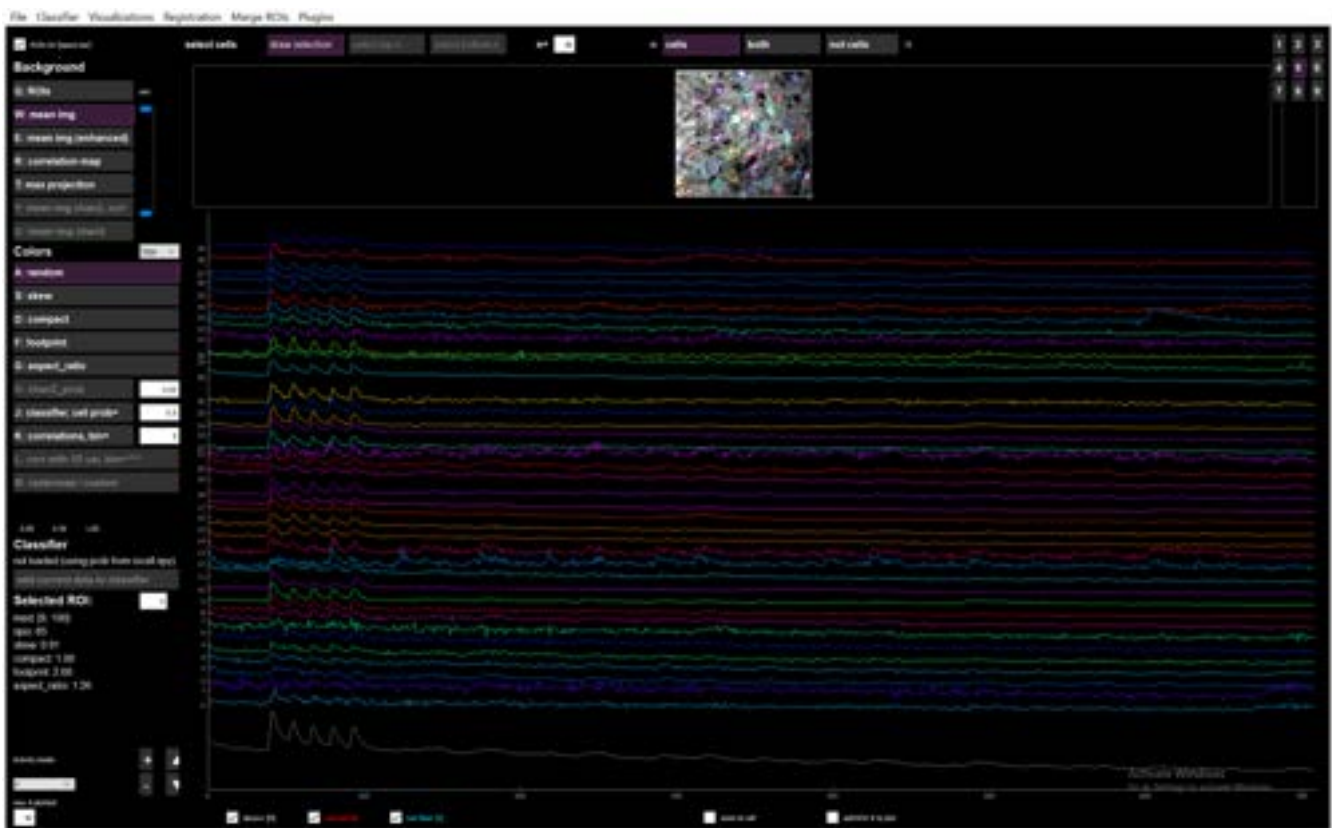


FIGURE 10 Visualization of our in vivo dataset using commonly used software Suite 2P.

neurons there might be a slight lag if there are many neurons belonging to the cluster. These issues can potentially be addressed in future iterations of the software. In the future, it may be helpful to give the users the ability to load datasets in their own formats or to automatically infer spike trains using deconvolution algorithms. Although the application was initially designed for analyzing neuronal data, it can be used for any dynamic time-series data with a spatial component to it such as the analysis of EEG or fMRI data. The visualization of time trace feature of our application would prove to be beneficial for analyzing this type of data. We currently provide the users the

ability to choose between communities numbered 0–4 or the top 5 largely populated communities. Choosing more than five to seven communities may lead to visual clutter depending on the density of the activity and make the application less efficient for the analysis process. However, the ability to visualize a single-colored cluster, may diminish the likelihood of visual clutter. We would also like to consider all the inputs mentioned by the neurobiologists and implement the necessary features to make this application an efficient tool for analysis. V-NeuroStack can also be used on other dynamic networks containing abstract or real spatio-temporal data such as

social and animal networks where the spatial component would also change with respect to time. We note that the goal of the application is to provide the user greater insights into potential patterns in the data. Therefore, it is worth mentioning that users bring their own biases to the analysis. Thus, the V-NeuroStack application is best seen as a hypothesis generation tool to allow the user to see potential patterns in the data and then pursue specific hypotheses about them.

To our knowledge no other similar analysis tool exists that allows 3D visualization of time-series calcium imaging data, therefore no equivalent comparison software exists. To compare V-NeuroStack to another commonly used and well-regarded software package, we imported our *in vivo* dataset into Suite2P, an established free platform for analysis of two-photon imaging data (Figure 10, Pachitariu et al., 2017). Although Suite2P did an excellent job of displaying the time series of each neuron, it does not display their spatial relationships in an interactive cube to allow the user to determine if functionally-organized clusters have particular spatial organizations.

8 | SUMMARY AND CONCLUSION

In this paper we present V-NeuroStack, a web application consisting of three-dimensional time stacks along with juxtaposed 2D view and a dual-line graph for exploration and analysis of neuronal activity time-series data. We use a dataset generated by calcium imaging of spontaneous activity of neurons from a mouse brain slice. These data contain rich spatio-temporal information and require effective use of visual analytics for exploration and analysis. We also propose a three-step analysis pipeline. The first step uses 3D time stacks to identify patterns in the entire length of neuronal activity or clipped sample of the dataset. In the second step, we provide the ability to view each timestep in greater detail by providing a 2D juxtaposed snapshot of the neuronal activity. In the third and final step, each of the neurons in the 2D layout can be further explored using dual-line graph showing the raw and first derivative of traces for the entire length of data capture. Our application currently supports storing 2D visualizations by downloading images in png format. We combine space-time cube and two-dimensional network visualization paradigms in the neuroscience space for visualizing calcium imaging data. The results describe the ability to visualize the commonly understood principles in neuroscience in a user intuitive interactive way. The contribution of our work to the field of visualization is the simplicity with which structures of neural organization can become evident with just a button's click. This application reduces the cognitive load on the user thereby improving the quality of the user experience and potentially enhancing the process of scientific discovery.

AUTHOR CONTRIBUTIONS

Ashwini G. Naik: Conceptualization; formal analysis; investigation; writing (original and review); visualization; software. **Robert V. Kenyon:** Conceptualization; writing (original and review); funding acquisition; visualization; software. **Aynaz Taheri:** Formal analysis; investigation. **Tanya Y. BergerWolf:** Conceptualization; funding

acquisition. **Baher A. Ibrahim:** Investigation. **Yoshitaka Shinagawa:** Formal analysis. **Daniel A. Llano:** Conceptualization; investigation; writing (original and review); funding acquisition.

ACKNOWLEDGMENTS

The initial part of the work was supported by a CRCNS grant from the NSF (1515587) and NIH (S10OD023569). We thank Manuel Tanzi for his thoughts and inputs during this project. We also thank Gang Xiao and Dr. Ekaterina Gribkova for their valuable evaluation and feedback about the application.

CONFLICT OF INTEREST

No conflicting interests exist.

PEER REVIEW

The peer review history for this article is available at <https://publons.com/publon/10.1002/jnr.25139>.

DATA AVAILABILITY STATEMENT

All data used in this study can be found at the following locations: Link for *in-vitro* data (<https://github.com/anaik12/bvis/tree/master/point/data>); Link for *in vivo* data (<https://github.com/anaik12/bvis/tree/master/point/invivo/data>).

REFERENCES

- Ahn, J.-W., Taieb-Maimon, M., Sopan, A., Plaisant, C., & Shneiderman, B. (2011). Temporal visualization of social network dynamics: Prototypes for nation of neighbors. In *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*. Springer.
- Akerboom, J., Chen, T. W., Wardill, T. J., Tian, L., Marvin, J. S., Mutlu, S., Calderón, N. C., Esposti, F., Borghuis, B. G., Sun, X. R., & Gordus, A. (2012). Optimization of a GCaMP calcium indicator for neural activity imaging. *Journal of Neuroscience*, 32(13), 819–840.
- Al-Awami, A. K., Beyer, J., Strobelt, H., Kasthuri, N., Lichtman, J. W., Pfister, H., & Hadwiger, M. (2014). Neurolines: A subway map metaphor for visualizing nanoscale neuronal connectivity. *IEEE Transactions on Visualization and Computer Graphics*, 20(12), 2369–2378.
- Arsiwalla, X. D., Zucca, R., Betella, A., Martinez, E., Dalmazzo, D., Omedas, P., Deco, G., & Verschure, P. F. (2015). Network dynamics with BrainX3: A large-scale simulation of the human brain network with real-time interaction. *Frontiers in Neuroinformatics*, 9, 2.
- Bach, B., Dragicevic, P., Archambault, D., Hurter, C., & Carpendale, S. (2014). *A review of temporal data visualizations based on space-time cube operations*. Eurographics Conference on Visualization.
- Bach, B., Henry-Riche, N., Dwyer, T., Madhyastha, T., Fekete, J. D., & Grabowski, T. (2015). Small MultiPiles: Piling time to explore temporal patterns in dynamic networks. *Computer Graphics Forum*, 34, 31–40.
- Bach, B., Pietriga, E., & Fekete, J.-D. (2014). *Visualizing dynamic networks with matrix cubes*. Proceedings of the SIGCHI conference on Human Factors in Computing Systems.
- Beck, F., Burch, M., Diehl, S., & Weiskopf, D. (2017). A taxonomy and survey of dynamic graph visualization. *Computer Graphics Forum*, 36, 133–159.
- Bender-deMoll, S., & McFarland, D. A. (2006). The art and science of dynamic network visualization. *Journal of Social Structure*, 7(2), 1–38.
- Cantu, D. A., Wang, B., Gongwer, M. W., He, C. X., Goel, A., Suresh, A., Kourdougli, N., Arroyo, E. D., Zeiger, W., & Portera-Cailliau, C.

- (2020). EZcalcium: Open-source toolbox for analysis of calcium imaging data. *Frontiers in Neural Circuits*, 14, 25.
- Chen, W., Shi, L., & Chen, W. (2018). A survey of macroscopic brain network visualization technology. *Chinese Journal of Electronics*, 27(5), 889–899.
- Danchilla, B. (2012). Three.js Framework.
- Eich, B., & McKinney, C. R. (1996). *JavaScript language specification* (96-002). Technologies Ber Netscape Communications.
- Federico, P., Aigner, W., Miksch, S., Windhager, F., & Smuc, M. (2012). *Vertigo zoom: Combining relational and temporal perspectives on dynamic networks*. Proceedings of the International Working Conference on Advanced Visual Interfaces.
- Fu, X., Hong, S.-H., Nikolov, N. S., Shen, X., Wu, Y., & Xuk, K. (2007). Visualization and analysis of email networks. In *2007 6th International Asia-Pacific Symposium on Visualization*. IEEE.
- Fujiwara, T., Chou, J.-K., McCullough, A. M., Ranganath, C., & Ma, K.-L. (2017). A visual analytics system for brain functional connectivity comparison across individuals, groups, and time points. In *2017 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE.
- Ganglberger, F., Swoboda, N., Frauenstein, L., Kaczanowska, J., Haubensak, W., & Bühler, K. (2019). BrainTrawler: A visual analytics framework for iterative exploration of heterogeneous big brain data. *Computers & Graphics*, 82, 304–320.
- Ganglberger, F. J., Kaczanowska, J., Haubensak, W., & Bühler, K. (2020). A data structure for real-time aggregation queries of big brain networks. *Neuroinformatics*, 18(1), 131–149.
- Gatalsky, P., Andrienko, N., & Andrienko, G. (2004). Interactive analysis of event data using space-time cube. In *Proceedings. Eighth International Conference on Information Visualisation, 2004. IV 2004*. IEEE.
- Giovannucci, A., Friedrich, J., Gunn, P., Kalfon, J., Brown, B. L., Koay, S. A., Taxidis, J., Najafi, F., Gauthier, J. L., & Zhou, P. (2019). CalmAn an open source tool for scalable calcium imaging data analysis. *eLife*, 8, e38173.
- Grienberger, C., & Konnerth, A. (2012). Imaging calcium in neurons. *Neuron*, 73(5), 862–885.
- Harrower, M., & Brewer, C. A. (2003). ColorBrewer.org: An online tool for selecting colour schemes for maps. *The Cartographic Journal*, 40(1), 27–37.
- Hassan, M., Shamas, M., Khalil, M., El Falou, W., & Wendling, F. (2015). EEGNET: An open source tool for analyzing and visualizing M/EEG connectome. *PLoS ONE*, 10(9), e0138297.
- He, B., Dai, Y., Astolfi, L., Babiloni, F., Yuan, H., & Yang, L. (2011). eConnectome: A MATLAB toolbox for mapping and imaging of brain functional connectivity. *Journal of Neuroscience Methods*, 195(2), 261–269.
- Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology*, 160(1), 106–154.
- Hudson, A. E., Calderon, D. P., Pfaff, D. W., & Proekt, A. (2014). Recovery of consciousness is mediated by a network of discrete metastable activity states. *Proceedings of the National Academy of Sciences of the United States of America*, 111(25), 9283–9288.
- Ibrahim, B. A., Wang, H., Lesicko, A. M., Bucci, B., Paul, K., & Llano, D. A. (2017). Effect of temperature on FAD and NADH-derived signals and neurometabolic coupling in the mouse auditory and motor cortex. *Pflügers Archiv-European Journal of Physiology*, 469(12), 1631–1649.
- Itoh, M., Yoshinaga, N., Toyoda, M., & Kitsuregawa, M. (2011). 3D Visualization of temporal changes in bloggers' activities and interests. In *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE.
- Itoh, M., Yoshinaga, N., Toyoda, M., & Kitsuregawa, M. (2012). Analysis and visualization of temporal changes in bloggers' activities and interests. In *2012 IEEE Pacific Visualization Symposium*. IEEE.
- Ji, C., Maurits, N. M., & Roerdink, J. B. (2018). Data-driven visualization of multichannel EEG coherence networks based on community structure analysis. *Applied Network Science*, 3(1), 1–24.
- Kinsky, N. R., Mau, W., Sullivan, D. W., Levy, S. J., Ruesch, E. A., & Hasselmo, M. E. (2020). Trajectory-modulated hippocampal neurons persist throughout memory-guided navigation. *Nature Communications*, 11(1), 1–14.
- LaPlante, R. A., Douw, L., Tang, W., & Stufflebeam, S. M. (2014). The connectome visualization utility: Software for visualization of human brain networks. *PLoS ONE*, 9(12), e113838.
- Liao, W., Wu, G.-R., Xu, Q., Ji, G.-J., Zhang, Z., Zang, Y.-F., & Lu, G. (2014). DynamicBC: A MATLAB toolbox for dynamic brain connectome analysis. *Brain Connectivity*, 4(10), 780–790.
- Liu, J., Liao, X., Xia, M., & He, Y. (2018). Chronnectome fingerprinting: Identifying individuals and predicting higher cognitive functions using dynamic brain connectivity patterns. *Human Brain Mapping*, 39(2), 902–915.
- Llano, D. A., Ma, C., Di Fabrizio, U., Taheri, A., Stebbings, K. A., Yudintsev, G., Xiao, G., Kenyon, R. V., & Berger-Wolf, T. Y. (2021). A novel dynamic network imaging analysis method reveals aging-related fragmentation of cortical networks in mouse. *Network Neuroscience*, 5(2), 569–590.
- Lu, J., Li, C., Singh-Alvarado, J., Zhou, Z. C., Fröhlich, F., Mooney, R., & Wang, F. (2018). MIN1PIPE: A miniscope 1-photon-based calcium imaging signal extraction pipeline. *Cell Reports*, 23(12), 3673–3684.
- Ma, C., Forbes, A. G., Llano, D. A., Berger-Wolf, T., & Kenyon, R. V. (2016). SwordPlots: Exploring neuron behavior within dynamic communities of brain networks. *Electronic Imaging*, 2016(16), 1–13.
- Ma, C., Kenyon, R. V., Forbes, A. G., Berger-Wolf, T. Y., Slater, B. J., & Llano, D. A. (2015). *Visualizing dynamic brain networks using an animated dual-representation*. EuroVis (Short Papers).
- Ma, C., Pelloio, F., Llano, D. A., Stebbings, K. A., Kenyon, R. V., & Marai, G. E. (2018). Rembrain: Exploring dynamic biospatial networks with mosaic matrices and mirror glyphs. *Electronic Imaging*, 2018(1), 60404.
- Michael, B., Vadim, O., & Jeffrey, H. (2011). D3: Data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12), 2301–2309.
- Mountcastle, V. B. (1997). The columnar organization of the neocortex. *Brain*, 120(4), 701–722.
- Pachitariu, M., Stringer, C., Dipoppa, M., Schröder, S., Rossi, L. F., Dagleish, H., Carandini, M., & Harris, K. D. (2017). *Suite2p: Beyond 10,000 neurons with standard two-photon microscopy*. BioRxiv.
- Pnevmatikakis, E. A. (2019). Analysis pipelines for calcium imaging data. *Current Opinion in Neurobiology*, 55, 15–21.
- Pnevmatikakis, E. A., & Giovannucci, A. (2017). NoRMCorre: An online algorithm for piecewise rigid motion correction of calcium imaging data. *Journal of Neuroscience Methods*, 291, 83–94.
- Reda, K., Tantipathanandh, C., Johnson, A., Leigh, J., & Berger-Wolf, T. (2011). Visualizing the evolution of community structures in dynamic social networks. *Computer Graphics Forum*, 30, 1061–1070.
- Rossetti, G., & Cazabet, R. (2018). Community discovery in dynamic networks: A survey. *ACM Computing Surveys (CSUR)*, 51(2), 1–37.
- Tantipathanandh, C., & Berger-Wolf, T. Y. (2011). Finding communities in dynamic social networks. In *2011 IEEE 11th International Conference on Data Mining*. IEEE.
- Tanzi, M. (2017). *Visualization of neurophysiological dynamic communities within the mouse brain*. University of Illinois at Chicago.
- Tegtmeier, J., Brosch, M., Janitzky, K., Heinze, H.-J., Ohl, F. W., & Lippert, M. T. (2018). CAVE: An open-source tool for combined analysis of head-mounted calcium imaging and behavior in MATLAB. *Frontiers in Neuroscience*, 12, 958.
- Ten Caat, M., Maurits, N. M., & Roerdink, J. B. (2008). Data-driven visualization and group analysis of multichannel EEG coherence with

- functional units. *IEEE Transactions on Visualization and Computer Graphics*, 14(4), 756–771.
- Trotts, I., Mikula, S., & Jones, E. G. (2007). Interactive visualization of multiresolution image stacks in 3D. *Neuroimage*, 35(3), 1038–1043.
- van den Elzen, S., Holten, D., Blaas, J., & van Wijk, J. J. (2015). Reducing snapshots to points: A visual analytics approach to dynamic network exploration. *IEEE Transactions on Visualization and Computer Graphics*, 22(1), 1–10.
- van Rossum, G. (1995). *Python reference manual*. Department of Computer Science [CS], (R 9525).
- Vijayalakshmi, R., Dasari, N., Nandagopal, D., Subhiksha, R., Cocks, B., Dahal, N., & Thilaga, M. (2014). Change detection and visualization of functional brain networks using EEG data. *Procedia Computer Science*, 29, 672–682.
- Wagner Filho, J. A., Stuerzlinger, W., & Nedel, L. (2019). Evaluating an immersive space–time cube geovisualization for intuitive trajectory data exploration. *IEEE Transactions on Visualization and Computer Graphics*, 26(1), 514–524.
- Wang, C., Ong, J. L., Patanaik, A., Zhou, J., & Chee, M. W. (2016). Spontaneous eyelid closures link vigilance fluctuation with fMRI dynamic connectivity states. *Proceedings of the National Academy of Sciences of the United States of America*, 113(34), 9653–9658.
- Weisenburger, S., & Vaziri, A. (2018). A guide to emerging technologies for large-scale and whole-brain optical imaging of neuronal activity. *Annual Review of Neuroscience*, 41, 431–452.
- Xia, M., Wang, J., & He, Y. (2013). BrainNet Viewer: A network visualization tool for human brain connectomics. *PLoS ONE*, 8(7), e68910.

How to cite this article: Naik, A. G., Kenyon, R. V., Taheri, A., BergerWolf, T. Y., Ibrahim, B. A., Shinagawa, Y., & Llano, D. A. (2022). V-NeuroStack: Open-source 3D time stack software for identifying patterns in neuronal data. *Journal of Neuroscience Research*, 00, 1–15. <https://doi.org/10.1002/jnr.25139>