New tools for web-enabled interactive visualizations of electrophysiological data

Eric L. Denovellis Emily P. Stephen Daniel Bullock

Abstract

Electrophysiology datasets in neuroscience are becoming richer and more complex as data is collected on multiple scales, dataset sizes increase, and more sophisticated questions are asked of the data. Visualization is an essential tool for understanding these datasets at all stages of analysis, but current practices in visualization of electrophysiological data are limited in their ability to efficiently compare between visualizations (such as between a raster plot of neuronal spiking to a visualization of a firing rate model of the same neuron) and filter complex data (for example, by limiting a visualization to specific brain areas on demand). Such difficulties are only magnified as the amount of data increases.

This paper describes a set of composable, web-enabled interactive visualization tools developed for use in electrophysiological studies. These tools were developed to facilitate (1) exploratory data analysis, (2) checking of raw data and statistical modeling assumptions, and (3) data presentation in the context of large, complex and multi-scale neuroscience data. Data from several experiments were used to test the tools. These visualization tools are viewable in the web browser and open-source, making them easily shareable online and allowing for modification and development by the neuroscience community.

Introduction

Current theories of brain functioning ascribe different roles to different scales: neurons, cortical layers, brain areas, networks between brain areas. For

example, the Communication-through-Coherence Hypothesis postulates that communication in the brain happens primarily through phase coordination between groups of neurons (Fries, 2005). This phase coordination between groups of neurons may differ between different layers, frequencies, within-brain areas, and between brain areas (Buffalo et al., 2011; Buschman et al., 2012; Gregoriou et al., 2009). Through the use of multiple electrode arrays and laminar probes, we are beginning to collect data at these different scales and understand how they interact (Miller and Wilson, 2008). However, as electrode technology progresses, our understanding of the data is not limited by the amount of data we can collect, but by our ability to efficiently understand and model relationships in the data.

Take, for example, a typical analysis of an electrocorticography dataset in which grids of intracranial electrodes are placed across large portions of cortex. These grids span multiple brain areas and are often combined with microelectrode grids to measure both local field potentials and action potentials from individual neurons (Figure 1A, REFERENCE). Given enough data, this allows us to ask questions about the properties at different spatial scales (units, multiunits, local field potentials, brain region summaries) and how they relate (e.g. correlation and coherence between local field potentials, local field potentials and neurons, neurons and neurons, see Figure 1B). Moreover, we can ask questions about how these change over time and/or relate to experimental conditions. This results in a dataset with many complex interrelations.

Understanding a dataset such as this becomes even more challenging as we can record from more electrodes. For example, when assessing relationships between recorded signals, the number of possible associations scales quadratically with the number of signals. That is, 10 electrodes means analyzing 100 relationships between electrodes. Implantation of multielectrode arrays with upwards of 100 electrodes is becoming common (Einevoll et al., 2012; Miller and Wilson, 2008; Siegel et al., 2015) and the number of simultaneously recorded neurons is projected to double every seven years (Stevenson and Kording, 2011).

Visualization of data is one way that we can reduce data complexity — allowing us to make multiple simultaneous comparisons, easing the cognitive burden on working memory by efficiently encoding properties of the data into features salient to the visual system (ref). In addition, visualization is

important in the understanding and checking of statistical assumptions — it helps reveal differences between the expected structure of the data (the model) and the observed data (Anscombe, 1973; Tukey, 1977). This is important, from the initial stages of analysis to publication, for revising our assumptions and models and for understanding and communicating where and how often our models fail (Gelman, 2004).

However, current practice with electrophysiology data relies on static visualization — requiring the generation of figures for each particular view. This makes it difficult to explore and check the data efficiently. For example, Liu and Heer (2014) found that even a 500 millisecond delay between visualizations could reduce the amount of the dataset explored and affect the number of hypotheses and observations formed.

Adding interactivity allows the user to change perspectives and modify analyses on demand, facilitating comprehension and hypothesis generation (Liu and Heer, 2014). Neuroimaging studies, which generate large datasets with complex interrelations, make extensive use of interactive visualization tools (e.g. the freeview module in freesurfer, pysurfer, spm), but there are no such tools that exist for electrophysiology studies. One can design user interfaces using MATLAB, but these are hard to share and require commercial software.

We present a set of three tools aimed at providing basic interactive visualizations for electrophysiology studies: `SpectraVis` is a tool aimed at exploring task-related functional networks over time and frequency; `RasterVis` is a tool for dynamically displaying and sorting spike raster plots and peri-event histograms; and `GLMVis` is a a tool for displaying coefficients from generalized linear models (GLMs) — which are commonly used to describe the receptive field of neurons. Each visualization allows for examination of the electrophysiological signals (or summary statistic of the signal) over time relative to task-relevant events, comparison of different subjects or recording sessions, and aggregation or filtering of signals. The visualizations are composable — two or more of the visualizations can be linked together to give a more comprehensive view of the dataset — and static visualizations can be exported for use in papers. Finally, all visualizations are web-based and open-source, making them easily shareable and operating system independent, and allowing for modification and repurposing by the neuroscience community. To see working examples of all three tools, please visit http://ericdeno.com/research/.

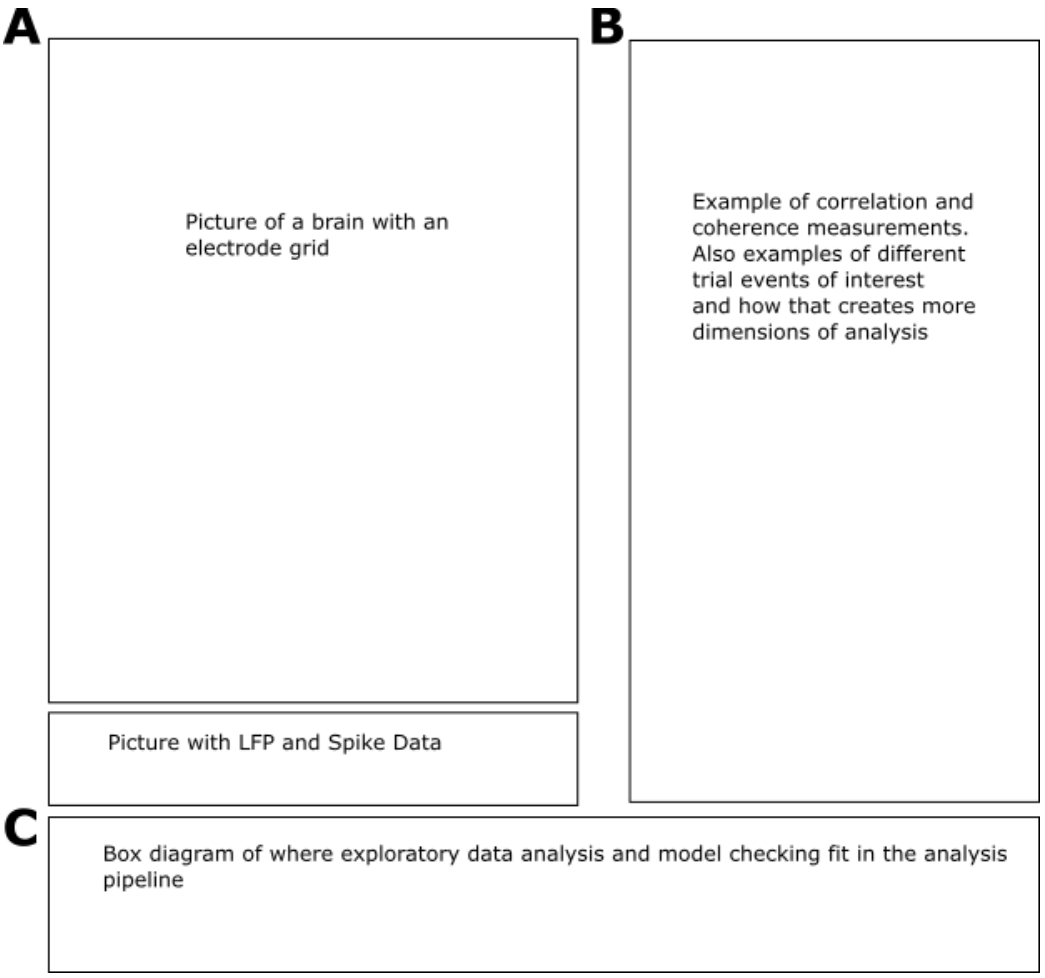

Figure 1: Typical Analysis.

Materials and Methods

Design

In order to make our set of tools accessible to a large number of users, we identified a set of design principles that would make them maximally useful to developers — who may want to extend the visualization code based on our toolkit — and to the end-users of the visualizations (Sherif et al., 2015). To that end, our approach is to create interactive visualizations that are:

1. **Configurable** — so visualizations can dynamically display different datasets or be preset to a particular view state.
2. **Shareable** — so others can easily view the visualizations online or in print.
3. **Modular** — so the visualizations can be used independently or linked together to provide an integrated view of an electrophysiological dataset.
4. **Extendable** — so others can implement their own visualization algorithms and modify the visualizations for their own use.

Configurable

<–! fill in these sections with a few sentences describing how the design principles will make them more useful to developers and end-users –> Configurability ensures that developers will be able to set up the visualizations using their own datasets and customize settings for their own needs. The visualizations are configurable in three ways: parameters can be passed through the URL, parameters can be preset using Javascript via the `init` function for each visualization, and data and data labels can be loaded using the JSON file format.

The JSON file format is a readable, XML-like format allows the visualization to display different datasets — the visualization dynamically adjusts the axes, labels, and the display for each dataset based on the JSON file. In order for this to work, the JSON files must be in the specified format for the visualization, details of which are documented on the wiki for each visualization (Figure 2).

Importantly, JSON files can be exported from MATLAB structures (using, for example, the open-source toolbox JSONlab) and Python — providing an

important bridge between commonly used analysis tools and the visualizations. An example workflow might be to do the analysis in MATLAB, format the data into MATLAB structures that correspond with the visualization, use JSONlab to export the MATLAB structures to the JSON format, and use the visualization tools to explore the dataset (Figure 2). Example MATLAB scripts for exporting JSON in the correct format are included in the Github repositories to make it easier for the user to get started.

Shareable

To make the visualizations shareable, the visualizations were written with modern web technologies — HTML, CSS, and Javascript. They can be deployed via a local or remotely hosted web server and viewed with any modern browsers (Firefox 4+, Chrome 4+, Safari 4+, Opera 9.5+ and IE9+). As a result, the visualizations require no specialized software (beyond a browser) to view.

Users can share a particular state of the visualization using permanent links (permalinks) — each visualization has a button which provides the URL containing the parameters necessary to generate the current view. For example, `SpectraVis` can show correlation networks across time. If a user wanted to share a snapshot of the correlation network at a specific time (e.g. 100 ms after stimulus onset), clicking on the link button would provide a URL that could then be shared with colleagues.

Additionally, static visualizations can be saved for publication purposes. Each visualization includes a button to download the current view of the visualization in scalable vector graphics (SVG) format. This format has the advantage that it can be resized without loss of resolution — making it useful for both presentations and publications — and can be imported into a graphics program of choice such as Inkscape or Adobe Illustrator for further modification. The New York Times, which frequently uses interactive graphics online and in print, has used this workflow successfully and we used it for many of the figures in this paper.

Modular and Extendable

Each visualization is self-contained and works independently of the other

visualizations. The visualizations can be selectively linked together by using the permalinks — which allow specification of a particular state of the linked visualization. For example, `GLMVis` might display a neuron's receptive field response to several experimental stimuli. By a simple modification of the code, this can be linked to the neuron's raster plot in `RasterVis` — showing the spiking response of the neuron to each experimental stimuli. This makes the visualizations composable — the visualizations can be mixed and matched to provide a desired view of the dataset.

The visualizations' internal code is also constructed modularly — separating the internal visualization modules from data loading modules and from user interface elements such as buttons. Developers can import and export these modules selectively or make their own modules, making the visualization customizable to the developers' needs. For example, in `SpectraVis`, a developer might want to customize the layout of the correlation networks, spatially grouping nodes by brain area or another desired metric. Constructing the code modularly allows a developer to implement this new layout without interring with the rest of the code internals.

Finally, each visualization has its own online software repository. The repositories are hosted on Github and can be downloaded and installed — including all software dependencies — using the node package manager (npm). This ensures the development tools, such as deploying a local web server (allowing the user to view the visualization on their own computer without having to host it remotely), are included. This helps developers extending the visualizations to get started developing as quickly as possible. These repositories are also open-sourced under the GNU General Public License (version 2), meaning the code is available to anyone to use and develop as long as the code remains open-source.

Results

Here we describe the functionality and interface for the visualizations and explain why they are useful for analyzing electrophysiological data.

SpectraVis

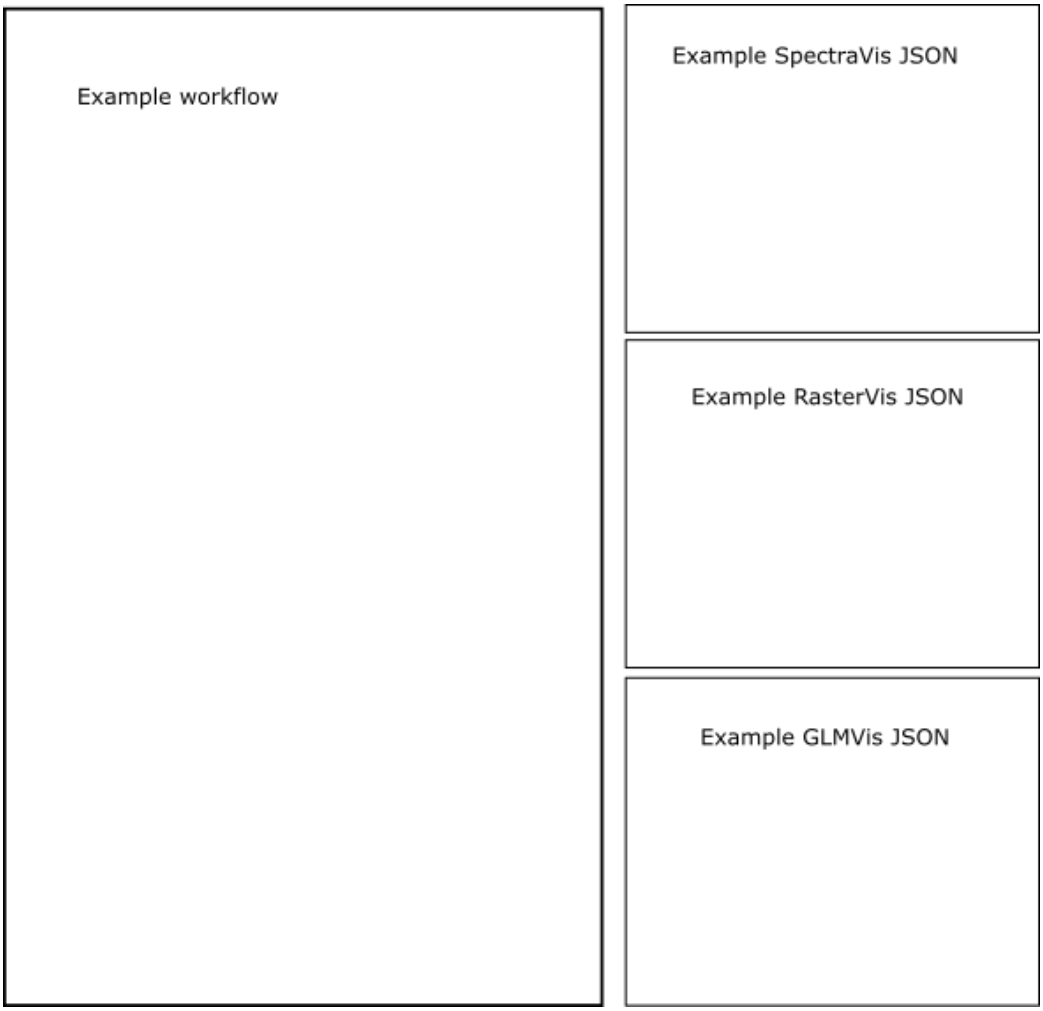

Figure 2: Typical Workflow and File formats.

Functional network analysis is a growing area of neuroscience research, driven in part by technological improvements allowing us to record from more sensors simultaneously. However, as researchers record from more sensors, network analyses can become unwieldy and hard to interpret, because the number of possible network connections scales quadratically with the number of sensors (e.g. electrodes). Further, we expect neural processes to form dynamic networks that vary over time, frequency, and spatial scales (e.g. within and between brain regions), adding numerous dimensions to network analyses.

`SpectraVis` is an interactive visualization aimed at enhancing exploratory analysis of networks by allowing the user to efficiently: (1) compare task-related functional networks over time and frequency, (2) compare individual and associative measures on all sensor pairs (e.g. spectra, coherences), and (3) compare different measures of association (e.g. correlation vs. coherence, binary vs. weighted networks). The different views of `SpectraVis` are dynamically linked, highlighting relationships between the metrics in response to user interaction.

Figure 3 shows a typical view of `SpectraVis`. The network view shows the anatomical location of the sensors (circles with sensor number) and edges (lines) weighted by the edge statistic (color of the line, measure of association between the sensors). In this example, the edges are binary, representing significant changes in local field potential coherence between *Speech* — subjects reading aloud the words of a famous speech or nursery rhyme — and *Silence* at a particular frequency (10 Hz) and time (187.5 ms after speech onset). The network has dense connectivity within and between primary motor and primary somatosensory cortices (M1 and S1). Users can compare between binary edge statistics, which categorically declare associations between sensors, and weighted edge statistics, which use continuous measures such as the raw coherence difference and z-scored coherence difference, via the edge statistic dropdown.

The controls can also be used to examine the evolution of the network over time using either the time slider, which can be dragged to a time of interest, or the play button, which will automatically advance the time slider. The user can compare networks at different frequency bands (for example, comparing a 10 Hz alpha band network to a 20 Hz beta band network) using the frequency slider.

One difficulty of analyzing networks is interpreting the edges between sensors,

particularly if the network is a weighted network and there are many sensors. There often is not enough room in a visualization to display all the edges without much overlap. To solve this problem, we use two strategies: layout and filtering.

The network layout toggle controls where nodes are positioned (and consequently the edges between nodes). The *anatomical* layout places the nodes according to the anatomical position defined in the JSON files. For example, in an ECoG dataset the network nodes often correspond to electrodes, which can be displayed over an image of the brain to give the end-user a sense for the sulci and gyri underlying each electrode. In some cases the anatomical locations of the nodes may be less important than their position induced by the network topology: for example, a node with strong connections to other nodes may serve as a hub. To help with this kind of interpretation, `SpectraVis` offers a *topological* layout that models the nodes and edges as a physical system to limit the number of overlapping edges.

The second strategy to make edges more visible allows the user to isolate subsets of the network by filtering edges and sensors, thus reducing the number of comparisons needed to be made in any one view. Using the edge filter dropdown, the user can isolate the edges between sensors that reside within the same brain area (e.g. only auditory cortex - auditory cortex sensor pairs) or between sensors that have non-matching brain areas (e.g. only auditory cortex – motor cortex sensor pairs).

Once the desired network view has been obtained, users can get further detail by clicking on a pair of sensors. This loads a sensor view (3, dotted box) which depicts the relationship (spectra and coherences) between a selected pair of sensors (circled in black, network view, sensors 85 and 90) at all times and frequencies. Here, the edge between M1 (sensor 90) and S1 (sensor 85) represents a 10 Hz increase in speech coherence relative to silence. The increase co-occurs with higher frequency beta (15-25Hz) power suppression on the M1 sensor. The user can investigate the relationship between the sensor pair and the network view by mousing over a time-frequency bin in the sensor view, which correspondingly updates the network view to the time-frequency bin under the cursor.

<–! Conclusion paragraph –>

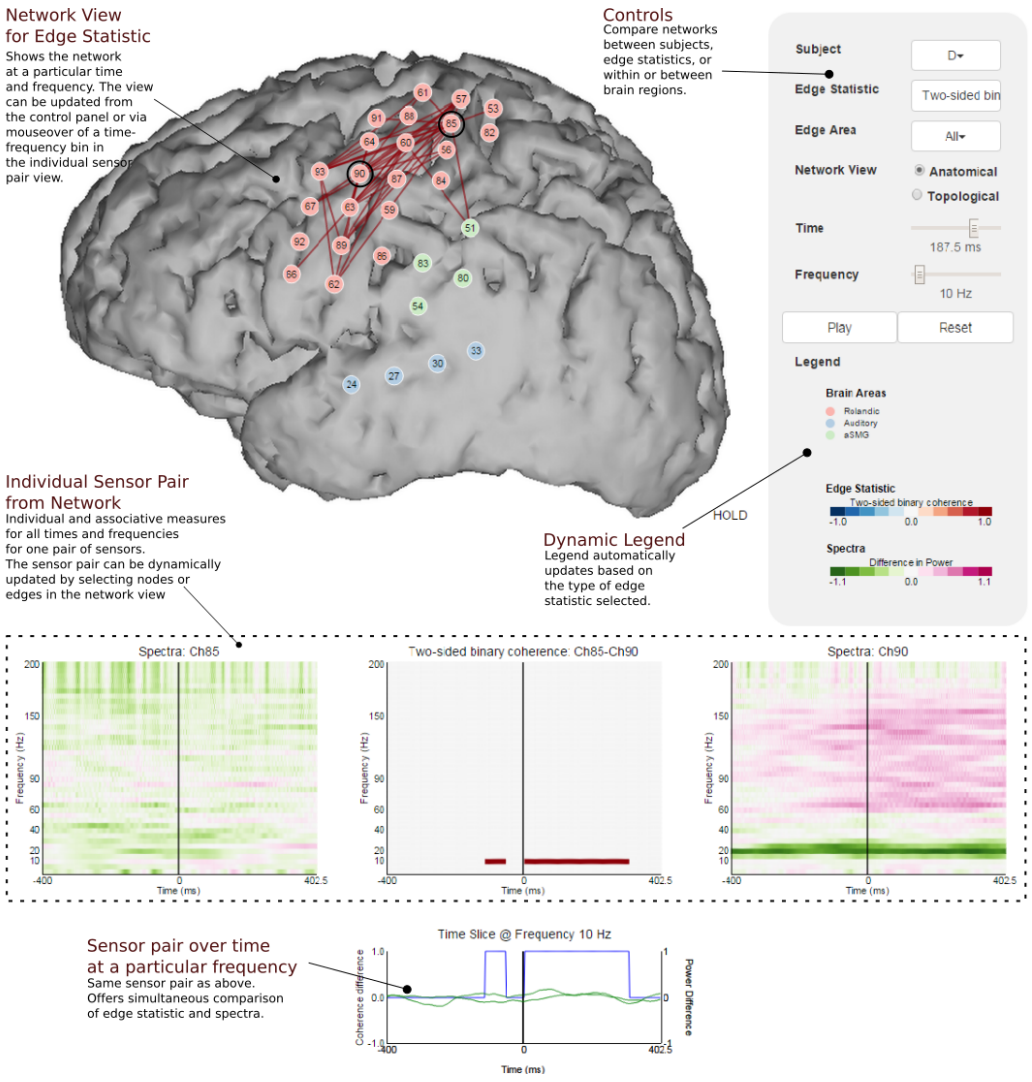

Figure 3: A static screenshot of the `SpectraVis` interface with the ECOG overt reading data.

RasterVis

`RasterVis` incorporates two canonical visualizations for single and multiunit spiking data — the raster plot and peri-event time histogram. The raster plot shows spike times for each trial relative to a trial event. The peri-event time histogram sums the raster over trials, showing the count of spikes that occurred in each time bin relative to the time of a trial event (REFERENCE: OBD or Uri's text?). Because these two types of visualizations are familiar and represent the "raw" spiking data, they are an ideal building-block visualization. Furthermore, they can also be used to compare raw spiking data to model-generated data in order to check statistical modeling assumptions (posterior predictive checks, REFERENCE) — so they can be useful in understanding how models reflect the data.

`RasterVis` uses interactivity and animation to supplement the raster plot and peri-event time histogram in order to make it easier for the user to accomplish typical tasks in the analysis of spiking data (See Figure 4 for a screenshot of the `RasterVis` interface).

For example, `RasterVis` allows for dynamic alignment of spike times and "on-the-fly" computation of peri-event histograms relative to experimental trial events (e.g. visual stimuli, timing of rewards, presentation of fixation points). Animated transitions emphasize how spike timing relative to trial event relates to another trial event. This helps a user quickly compare the timing of individual spikes and aggregate spiking (via histogram) to different cues and conditions. Different levels of aggregation (Gaussian smoothing) for the histogram can be compared as well.

`RasterVis` also allows for dynamic sorting of trials by experimental task factors. This feature creates on-demand plots for each condition within the task factor. For example, if a task factor is a visual cue with two experimental conditions — the color cue and the orientation cue — sorting by the visual cue creates two plots for the color condition and the orientation condition. This is essential for multidimensional analysis which may compare several different factors and conditions.

Finally, `RasterVis` allows users to find and select neurons by subject, recording session, or name. This is useful for fast comparison between neurons, linking to other visualizations (other visualizations can directly link to a specific neuron by name via a parameter passed via the URL), and general

exploratory analysis of the dataset.

<–! Conclusion paragraph –>

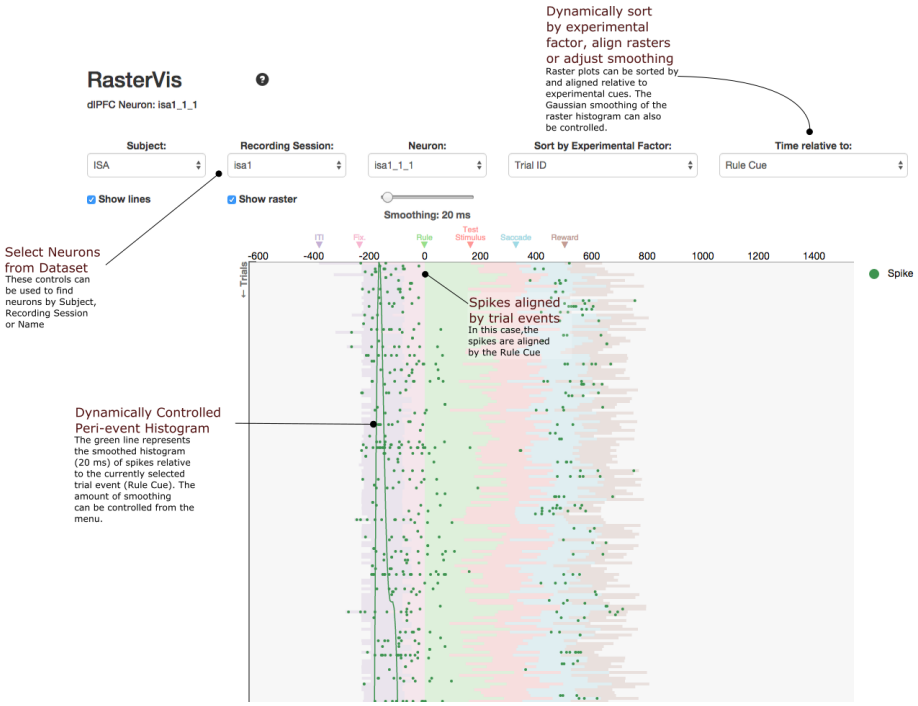

Figure 4: A static screenshot of the `RasterVis` interface.

GLMVis

A common analysis framework for characterizing the spiking response of neurons is the generalized linear model (GLM) (Fernandes et al., 2014; Harris et al., 2003; Mayo et al., 2015; Park et al., 2014; Pillow et al., 2008). GLMs can simultaneously estimate effects of experimental conditions, spike history (refractory period, bursting), non-linear firing rate changes over time, and dependence on other neurons (Truccolo, 2004) — making them useful for analyzing a wide range of experiments. GLMs are especially useful in situations where conditions of interest are interdependent, making them difficult to tease apart using simple tools like peri-event time histograms (REFERENCE – Kyle's time paper, etc?)

One consequence of being able to estimate many covariates simultaneously is that the relationship of the effects becomes hard to understand because of the number of dimensions — particularly if the covariates change over time and there are many neurons. Moreover, understanding the relationship between multiple covariates may be important to understanding *mixed selectivity* neurons (Rigotti et al., 2013). These neurons are sensitive to a combination of sensory, motor and cognitive processes, appear in higher-order association brain regions such as parietal and prefrontal cortex (Park et al., 2014; Rigotti et al., 2013), and may underlie the computation of complex behavior (Rigotti et al., 2010).

Therefore, we built `GLMVis`, an interactive visualization for GLMs, that: (1) shows the relationship between the multiple dimensions of the model fit over time, (2) allows filtering of neurons by effect size, brain area, and experimental subject, and (3) can be used to compare estimates from different models. To show the relationship between multiple dimensions, we use parallel coordinate plots (Inselberg, 1985; Wegman, 1990) — a compact representation of multivariate data that links each dimension on parallel axes by a line.

Figure 5 shows a typical view of `GLMVis`. Each axis is a black horizontal line that corresponds to a dimension of the GLM. Non-parallel lines connect the dimensions and represent a single neuron. The intersection of the axes and non-parallel lines is the computed value of the neuron at that dimension. Dropdown menus allow the user to filter the neurons by their brain area or subject. The user can also filter by effect size by "brushing" along a desired axis — holding and dragging the mouse to select neurons in the range of values. Multiple axes can be selected in order to investigate the associations between values in different dimensions. <–! you really need to describe the meaning of the dimensions here at some point ('effect size' is not descriptive enough) –>

<–! Conclusion paragraph –>

Discussion

We developed a novel interactive visualization toolkit for investigating electrophysiological data. This toolkit allows users to quickly explore raw data via `RasterVis` and intermediate analysis such as receptive fields and net-

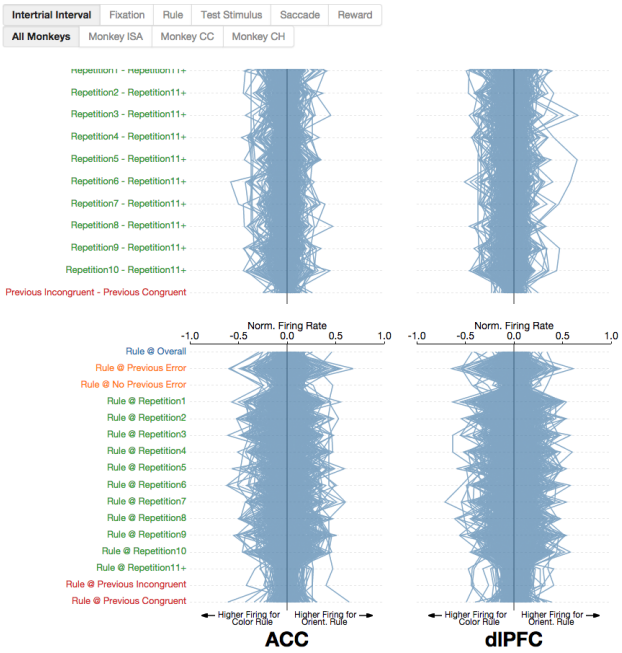

Figure 5: A static screenshot of the `GLMVis` interface.

works via `GLMVis` and `SpectraVis`. We believe these tools will be important going forward as electrode technology progresses and scientists form more complicated hypotheses.

Related Work

Importance of Visualization for Open Neuroscience

Online interactive visualization tools such as ours may provide a way for quick exploration of datasets online — enabling users to understand the datasets before performing more in-depth analyses. Indeed, the Allen Brain Institute — which shares massive neuroscience datasets online — makes extensive use of online visualizations to enable users to find the appropriate datasets. Because our tools encompass many common types of analysis for electrophysiological data, we hope our tools and future extensions can be used in a similar manner.

Sharing datasets is important because (1) it can help ensure the reproducibility

of results, (2) it makes the data available for meta-analyses, (3) the data can be used as benchmarks for computational models, and (4) the data can be used in new ways, maximizing its utility (Poldrack and Gorgolewski, 2014). This is particularly important for datasets that are hard to collect, such as those from non-human primates.

One challenge for the sharing of datasets is providing users with a way to find the datasets they want. Datasets provided in numerical form in non-standardized formats are hard to navigate and limit the usefulness of sharing the data. As electrophysiology dataset sharing becomes more common, it will be important to have tools to do preliminary investigations of these open datasets.

Future Directions

As data formats for sharing electrophysiology are standardized, we would like to change our JSON data structures to match those formats in order to make visualizing data as simple as possible. Unfortunately, there is no dominant standard currently.

We would also like to provide additional "plugin" layout options for `SpectraVis`, `GLMVis`, and `RasterVis`. In particular, there are numerous open source network layouts such as Group-in-a-box layouts, which clusters nodes according to group membership, and edge bundling layouts, which group similar edges together — all with the goal of improving understanding of the network structure. Likewise, with `GLMVis`, alternative views of the GLMs such as scatter plot matrices (SPLOMs) and dimensionality reduction algorithms such as t-Distributed Stochastic Neighbor Embedding (t-SNE) (Van der Maaten and Hinton, 2008) could help identify multivariate patterns in the data.

Finally, we would like to add more visualizations to the toolbox. Laminar electrodes, which have contacts spaced along the shank of the electrode and provide cortical layer information, pose an interesting challenge in terms of incorporating the extra dimension of depth information. As more studies incorporate laminar electrodes, finding effective visualizations and filtering of networks between different cortical layers, with the many possible associations between the layers, could be another good use case for interactive visualizations.

<–! Conclusion paragraph –>

Data Sharing

Acknowledgements

Dr. Gerwin Schalk, Dr. Peter Brunner, Dr. Earl K. Miller

References

Anscombe, F. J. (1973). Graphs in statistical analysis. *The American Statistician* 27, 17–21.

Buffalo, E. A., Fries, P., Landman, R., Buschman, T. J., and Desimone, R. (2011). Laminar differences in gamma and alpha coherence in the ventral stream. *PNAS* 108, 11262–11267. doi:10.1073/pnas.1011284108.

Buschman, T., Denovellis, E., Diogo, C., Bullock, D., and Miller, E. (2012). Synchronous Oscillatory Neural Ensembles for Rules in the Prefrontal Cortex. *Neuron* 76, 838–846. doi:10.1016/j.neuron.2012.09.029.

Einevoll, G. T., Franke, F., Hagen, E., Pouzat, C., and Harris, K. D. (2012). Towards reliable spike-train recordings from thousands of neurons with multielectrodes. *Current Opinion in Neurobiology* 22, 11–17. doi:10.1016/j.conb.2011.10.001.

Fernandes, H. L., Stevenson, I. H., Phillips, A. N., Segraves, M. A., and Kording, K. P. (2014). Saliency and Saccade Encoding in the Frontal Eye Field During Natural Scene Search. *Cerebral Cortex* 24, 3232–3245. doi:10.1093/cercor/bht179.

Fries, P. (2005). A mechanism for cognitive dynamics: neuronal communication through neuronal coherence. *Trends in Cognitive Sciences* 9, 474–480. doi:10.1016/j.tics.2005.08.011.

Gelman, A. (2004). Exploratory Data Analysis for Complex Models. *Journal of Computational and Graphical Statistics* 13, 755–779.

doi:10.1198/106186004X11435.

Gregoriou, G. G., Gotts, S. J., Zhou, H., and Desimone, R. (2009). High-Frequency, Long-Range Coupling Between Prefrontal and Visual Cortex During Attention. *Science* 324, 1207–1210. doi:10.1126/science.1171402.

Harris, K. D., Csicsvari, J., Hirase, H., Dragoi, G., and Buzsáki, G. (2003). Organization of cell assemblies in the hippocampus. *Nature* 424, 552–556.

Inselberg, A. (1985). The plane with parallel coordinates. *The Visual Computer* 1, 69–91. Available at: http://link.springer.com/article/10.1007/BF01898350 [Accessed January 25, 2016].

Liu, Z., and Heer, J. (2014). The effects of interactive latency on exploratory visual analysis. *Visualization and Computer Graphics, IEEE Transactions on* 20, 2122–2131. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6876022 [Accessed June 29, 2015].

Mayo, J. P., DiTomasso, A. R., Sommer, M. A., and Smith, M. A. (2015). Dynamics of visual receptive fields in the macaque frontal eye field. *Journal of Neurophysiology* 114, 3201–3210. doi:10.1152/jn.00746.2015.

Miller, E. K., and Wilson, M. A. (2008). All My Circuits: Using Multiple Electrodes to Understand Functioning Neural Networks. *Neuron* 60, 483–488. doi:10.1016/j.neuron.2008.10.033.

Park, I. M., Meister, M. L. R., Huk, A. C., and Pillow, J. W. (2014). Encoding and decoding in parietal cortex during sensorimotor decision-making. *Nature Neuroscience* 17, 1395–1403. doi:10.1038/nn.3800.

Pillow, J. W., Shlens, J., Paninski, L., Sher, A., Litke, A. M., Chichilnisky, E. J., et al. (2008). Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature* 454, 995–999. doi:10.1038/nature07140.

Poldrack, R. A., and Gorgolewski, K. J. (2014). Making big data open: data sharing in neuroimaging. *Nature Neuroscience* 17, 1510–1517. doi:10.1038/nn.3818.

Rigotti, M., Barak, O., Warden, M. R., Wang, X.-J., Daw, N. D., Miller, E. K., et al. (2013). The importance of mixed selectivity in complex cognitive tasks. *Nature* advance online publication. doi:10.1038/nature12160.

Rigotti, M., Rubin, D. B. D., Wang, X.-J., and Fusi, S. (2010). Internal

representation of task rules by recurrent dynamics: the importance of the diversity of neural responses. *Frontiers in Computational Neuroscience* 4. doi:10.3389/fncom.2010.00024.

Sherif, T., Kassis, N., Rousseau, M.-Ã., Adalat, R., and Evans, A. C. (2015). BrainBrowser: distributed, web-based neurological data visualization. *Frontiers in Neuroinformatics* 8. doi:10.3389/fninf.2014.00089.

Siegel, M., Buschman, T. J., and Miller, E. K. (2015). Cortical information flow during flexible sensorimotor decisions. *Science* 348, 1352–1355. Available at: http://www.sciencemag.org/content/348/6241/1352.short [Accessed August 5, 2015].

Stevenson, I. H., and Kording, K. P. (2011). How advances in neural recording affect data analysis. *Nature Neuroscience* 14, 139–142. doi:10.1038/nn.2731.

Truccolo, W. (2004). A Point Process Framework for Relating Neural Spiking Activity to Spiking History, Neural Ensemble, and Extrinsic Covariate Effects. *Journal of Neurophysiology* 93, 1074–1089. doi:10.1152/jn.00697.2004.

Tukey, J. W. (1977). Exploratory data analysis.

Van der Maaten, L., and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, 85. Available at: http://siplab.tudelft.nl/sites/default/files/vandermaaten08a.pdf [Accessed January 29, 2016].

Wegman, E. J. (1990). Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association* 85, 664–675.