

Visualizing Neuron Electrophysiology via NeuroElectro.org

Michael Gottlieb mikemgottlieb@gmail.com

Emily Hindalong ehindalong@gmail.com

Dmitry Tebaykin dmitry.tebaykin@gmail.com

1. Domain Situation

Overview and Motivation

Neuroscientists have conducted extensive research on the electrophysiological (ephys) properties of different neuron types, but there are barriers to comparing and aggregating results across different studies. This can be attributed to a lack of standard definitions and procedures and paywalls maintained by closed-access journals. To alleviate this, our colleagues have developed Neuroelectro - a freely available web-tool that allows users to directly compare data from different neuroscience studies. The primary goal is to “facilitate the discovery of neuron-to-neuron relationships and better understand the role of functional diversity across neuron types.”

However, the current visualizations provided by Neuroelectro are lacking, as the user is only able to view data for one neuron type or one ephys property at a time using scatter plots. Our goal is to develop a new interface that supports seamless browsing and analysis of select subsets of the data.

Data

The dataset is a collection of electrophysiology property measures taken from many different studies. Each measure is associated with a particular neuron type, experimental conditions, and other attributes. This data is stored in our database with a website interface (neuroelectro.org). All data is initially text-mined from html representations of articles and then curated by field experts. We currently have ~1k curated articles and ~2k text-mined but not curated ones. The volume of the articles is small because text-mining relies on specific electrophysiology tables. This might change in the future, so the visualization solution needs to be scalable.

Requirements

We met with our stakeholder in order to ascertain likely goals of system users. In the process, we identified the following questions:

1. What are the ephys characteristics of different neuron types?
2. How do different neuron types compare in terms of ephys properties?
3. How do experimental conditions relate to ephys measurements?
4. Where are different neuron types located in the brain?
5. How much data exists to support specific discoveries?

2. Data and Task Abstraction

Data

Our dataset is a *table* where each *item* (row) is data for particular neuron type taken from a single article. That is, each item is uniquely identified by the neuron type and article ID attributes.¹ The *attributes* of the data include 18 experimental condition indicators, 36 ephys properties, and ten that provide additional information about the article, neuron type, and organism.

Some of the experimental condition attributes are categorical and others are quantitative. All ephys property attributes are quantitative. Some of the quantitative attributes appear to be discrete while others are continuous - we will need to consult with our stakeholder to confirm which are which.

Most items have data for only a few of the experimental conditions and ephys properties. In other words, the data table is sparse.

Additionally, our stakeholder has offered to supply a dataset relating each neuron type to its associated brain regions (in voxels). In this dataset, each item will be a *list* of voxel locations for a particular neuron type.

Task

The primary usage context for this tool is *discovery*. The purpose of the tool is to help Neuroscientists achieve new insights and develop new hypotheses. This tool is not concerned with any *produce* tasks at this time.

Under the umbrella of *discovery*, a number of mid-level tasks can be identified:

1. *Explore* relationships between neuron types, ephys properties, and experimental conditions.
 - a. *Browse* attributes available for analysis.
 - b. *Identify distributions* of quantitative attribute values for a particular categorical attribute values (e.g. view all ephys_property_x values for neuron_type_y).
 - c. *Compare distributions* of quantitative attribute values for different categorical attribute values (e.g. ephys_property_x versus neuron type).
 - d. *Identify correlations* between quantitative attributes (e.g. ephys_property_x versus experimental_condition_y).
2. *Identify* regions of the brain where particular neuron types are found.
3. *Identify* how many data points exist for different combinations of neuron types, ephys properties, and experimental conditions.
4. *Identify* how many data points support a specific analysis.
5. *Summarize* the data in the current analysis scope.
6. *Lookup* details for individual data points (items).

¹ This is true of the current dataset, but we will need to discuss with our stakeholder whether this is a valid assumption going forward.

3. Personal Expertise

Mike and Dmitry are both in their 2nd year of the Bioinformatics M.Sc. program, and Emily is a recent graduate. Dmitry's thesis work relies heavily on the dataset in question. Mike and Emily have performed rotations in the Pavlidis lab and are familiar with the NeuroElectro project.

Emily has an undergraduate degree in Cognitive Systems and Mike has an undergraduate degree in Psychology and experience working in a cognitive neuroscience lab. Emily and Mike both have industry software development experience. Mike has experience in developing R packages. Dmitry has undergraduate degrees in Biology and Computer Science.

In summary, all three team members have experience in Bioinformatics, Neuroscience, Computer Science, and Software Development. Furthermore, we plan to consult with a neuroscience postdoc employed in the Pavlidis lab (the original developer of Neuroelectro.org), who is our client and target audience.

4. Proposed InfoVis Solution

In this section, we describe the idioms we intend to use to support the tasks above. We discuss our specific design choices (e.g. widgets) in the illustration section.

We will *facet* the display into three linked multiform views. The first view will show where different neuron types are found the brain. The second view will consist of an *overview* of the number of items for each combination of neuron type, ephys property, and experimental condition in the current analysis scope. The third view will contain plots for detailed analyses. The content of all three views will be linked via a common control panel.

Control panel

The control panel will *encode* attributes available for analysis in the form of a *list* (supporting task 1a). This list will contain, at the very least, neuron type, each ephys property, and each experimental condition. Each of these list items may be expanded to reveal possible attribute values - either in the form of a list (for categorical attributes) or range (continuous quantitative attributes). The user may then *filter* the data in the scope of analysis by either *selecting/deselecting* values to include or by specifying a range. All three views will dynamically apply the any filters selected here. (Supports tasks 3 and 4, in conjunction with the other views.)

Brain view

The brain view will use the *slice* idiom, showing a 2D view of the brain on the axial plane (and potentially other planes). The user will be able to change the coordinate of the third dimension. The locations of each neuron type in the brain will be *embedded* in the brain image using transparent *layers*, each of a different colour. The neuron types displayed will reflect those selected in the control panel. (Supports task 2.)

Data overview view

This view will contain a hive plot (or alternative encoding) with three axes - neuron type, ephys property, and experimental condition. The coordinates along the ephys property and experimental condition axes are attributes, and the coordinates along the neuron type axis are attribute values (one for each type of neuron). A link between axis coordinates means that data exists for that pair.² These links will be size coded according to the number of items that exist for that pair. When the analysis scope is the entire dataset, this will allow users to see which combinations have been considered in previous studies at a glance. When the analysis scope is a subset of the data, it will allow users to see the degree to which particular combinations are represented in the current scope. (Supports tasks 3 - 5).

Plots view

The plots view will *partition* the data into a number of *side-by-side* plots. The user will be able to add and remove plots to the display, up to a preset maximum. The user will be able to *select* and *change* attributes for the x and y axis of each plot, resulting in a scatterplots or strip plots depending on the nature of the selected attributes. Each point on the plot will encode a single item. The user will be able to get details about each data item on demand by selecting points (supporting task 6). This view enables comparison of properties of interest. The data on the plots will reflect the filters applied in the control panel. If time allows, we will consider adding support for aggregation. (Supports tasks 1b - 1d).

² For example, a link from the neuron type axis to the ephys property axis means “there exists a row for which neuron_type == x and ephy_prop_y is not null.” A link from the ephys property axis to the experimental conditions axis means “there exists a row for which ephys_prop_y is not null and experimental_condition_z is not null.”

5. Illustration

(See description below. Specific design decisions are tentative.)

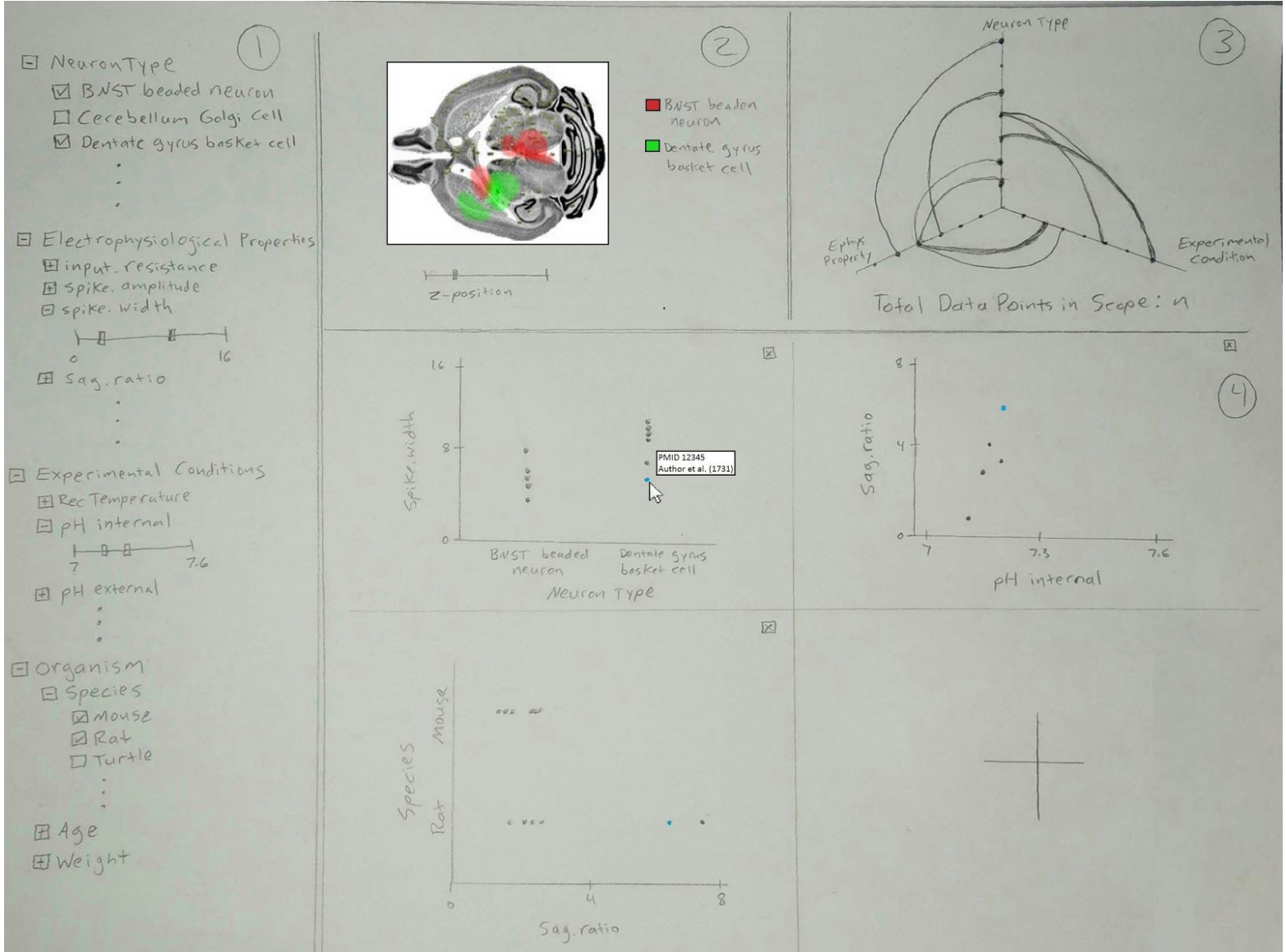


Fig 1. Main View

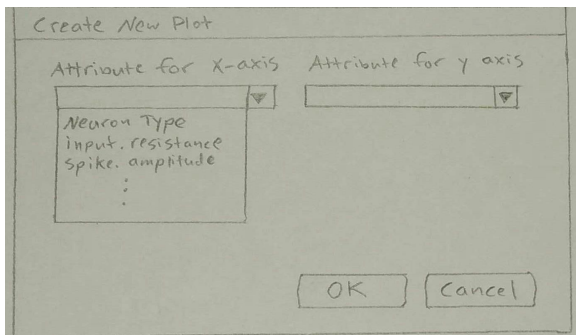


Fig 2. Create New Plot Dialog

Interface is divided into three resizable panels.

Section 1 is the Control Panel. The attribute list is implemented as an expandable tree view with tickboxes for categorical attribute values and range sliders for quantitative attribute values. In addition to neuron type, ephys properties, and experimental conditions, we have added a section for organism attributes. There will be “Select All” and “Deselect All” options for all tick box lists (not shown).

Section 2 is the brain view. The user may navigate through 2D slices of a mouse brain along the z-axis using a slider. The regions associated with each neuron type are shown on the brain in different colours.

Section 3 is the overview view. The axis coordinates will be arranged alphabetically. Hovering over an axis coordinate will reveal its label. Hovering over a link will reveal the relevant axis coordinate labels and the number of items that exist for that pair. This number will also be encoded in the width of the link. Below the plot, we will display the total number of data points in the current analysis scope. (Note that the mock-up only shows a subset of the links.)

Section 4 is the plots view. In the mock-up, the user has already created three plots. The user may create a new plot by clicking the + in an empty plot pane. This brings up the “Create New Plot” dialog, where the user selects attributes for the x and y axes. Hovering over a point reveals details about that item (including which article it came from) and highlights the point in all plots.

Unresolved Design Questions

The specifics of the current design are tentative and we have left several design questions open intentionally, including but not limited to:

1. Should we use a static grid in the plots view or resize based on the number of plots in the display?
2. Should axis ranges in the plots view reflect the full range for an attribute or the filtered range?
3. Should the user be allowed to put categorical attributes on both plot axes? If so, how should we show overlapping points?
4. How many neuron types should we restrict the brain view to? How do we select which ones go on the view if more neuron types are selected?
5. Should all values in the control panel be ticked or unticked by default?
6. Should we use a selection widget that is more scalable than tick boxes?

6. Use-Case Scenario

- Compare spike width for two different neuron types and see what regions of the brain are associated with these neuron types.
 - Spike width comparison:
 - User expands 'Neuron Type' node in tree viewer in the control panel on
 - User clicks "deselect all" (if necessary) and then ticks the neuron types of interest
 - This filters the data points that will be displayed in the plots
 - User clicks on add new plot button in plot view
 - User selects 'neuron type' for x-axis variable in 'Create New Plot' dialog
 - User selects 'spike width' for y-axis variable in 'Create New Plot' dialog
 - User clicks 'OK' on dialog box
 - Strip plot appears showing spike width measures partitioned by neuron type
 - Brain region identification:
 - On top left corner of interface just to the right of the control panel, user moves the z-axis slider to manipulate the axial brain viewer
 - User notes colour coded key and discovers the regions associated with these neuron types based on coloured regions in the brain

7. Implementation Approach

Neuroelectro.org has an HTML / Javascript(JQuery) / CSS front-end and a Python/Django back-end hooked up to a MySQL database. We will develop our system within this paradigm.

We plan to explore a few implementation options, potentially including D3.js + angular.js and RShiny. It is possible that we will settle on a combination of things. There are number of existing solutions for integrating RShiny and D3 with django projects. For example, this repo [<https://github.com/mfcovington/djangocms-shiny-app>] provides a way to connect RShiny with Django projects, and D3 would easily allow this via JSON. Our group is interested in D3.js but has much more R experience than JavaScript experience, so it seems likely at this point that we will use R Shiny in the final version.

8. Milestones

Milestone 1 (Nov 13): Visualization project setup

- Create a web page that will host the visualizations
- Create a github branch in the neuroelectro repository for the project
- Link database to the webpage for seamless data transfer
- Meet with stakeholder to confirm understanding of requirements
- Get all project members on the same page in terms of website and database local installs

Milestone 2 (Nov 27): Interface implementation

- Explore options for the implementation approaches
 - Literature review of the previous work done for similar projects
 - Explore the feasibility of D3.js, angular.js and node.js for this project (default is R Shiny)
 - Search for existing implementations that could suit our needs and integrate well with javascript
- Create interface frame with three resizable panels
- Create and populate control panel widgets
- Design and implement the layout for the plots view
- Create a non-interactive hiveplot for data overview in the top right view
- Add mouse brain with slider to brain view
- Implement add new plots dialog and creation of plots based on selected attributes
- Implement ability to remove plots

Milestone 3 (Dec 11): Functionality implementation and optimizations

- Implement control panel event handlers
- Implement display of layers on brain view
- Implement dynamic filtration logic across all views
- Implement mouse-hover pop-ups and linked highlighting in plots view
- Implement mouse-hover pop-ups for coordinates and links in the hive-plot view
- Add nice-to-have features (in separate document)

Milestone 4 (Dec 15): Code freeze/presentation

- Fix minor bugs, test the visualization, perform target user assessment of the vis performance
- Create project presentation
- Write up the final report

9. Previous work

Neuroelectro was developed as a Django text mining application [<http://neuroelectro.org/>] and was published as an article in *Frontiers in Neuroinformatics* [1]. Through the combined use of text mining and manual curation, it hosts experimental data from over 500 articles. The data of each article can be accessed by type of neuron, electrophysiological property, or via a table of articles. Neuroelectro also presents some of the data visually as scatterplots but does not allow for any interactivity, limiting its usefulness. We have direct access to Neuroelectro's database and will be able to make use of that in our work. While Neuroelectro is the only text mining project in the neuroinformatics and neuroscience domain, we will also consider the types of visualizations used in other text mining projects [e.g 2, 3].

We will likely take advantage of work done in interactive visualizations such as D3.js [4] and AngularJS [<https://angularjs.org/>] or R Shiny [<http://shiny.rstudio.com/>]. Overviews of the data will make use of the work done by Krzywinski et al. [5] for our proposed hive plot and Rieder et al. [6] for spatial navigation of neuronal locations in the brain. As we are leaning towards using R Shiny, we will most likely take advantage of previous work available via github and the Shiny website. For example, we have found solutions to set up multiple plots [<https://gist.github.com/wch/5436415>], use linked views [<http://shiny.rstudio.com/gallery/plot-interaction-zoom.html>], filter data [<http://shiny.rstudio.com/gallery/basic-datatable.html>], and rearrange plot layout [<http://shiny.rstudio.com/articles/layout-guide.html>].

Bibliography

1. Tripathy, Shreejoy J., et al. "NeuroElectro: a window to the world's neuron electrophysiology data." *Frontiers in neuroinformatics* 8 (2014).
2. Van Eck, Nees Jan, and Ludo Waltman. "Text mining and visualization using VOSviewer." *arXiv preprint arXiv:1109.2058* (2011).
3. Hinrichs, Uta, et al. "Trading Consequences: A Case Study of Combining Text Mining and Visualization to Facilitate Document Exploration." *Digital Scholarship in the Humanities* (2015).
4. Bostock, Michael, Vadim Ogievetsky, and Jeffrey Heer. "D³ data-driven documents." *Visualization and Computer Graphics, IEEE Transactions on* 17.12 (2011): 2301-2309.
5. Krzywinski, Martin, et al. "Hive plots—rational approach to visualizing networks." *Briefings in bioinformatics* 13.5 (2012): 627-644.
6. Rieder, Christian, et al. "Interactive visualization of multimodal volume data for neurosurgical tumor treatment." *Computer Graphics Forum*. Vol. 27. No. 3. Blackwell Publishing Ltd, (2008).