

CS315 - Final Project

Ben Le Cam

December 15th, 2014

- Project Name: NeuroCiC
- I believe that the abstract from the class work two years ago explains really well NeuroCiC: "the goal of this project is to facilitate the generation and analysis of head-direction histograms for Gary Muir's neuroscience research. Before this term, the method for building a histogram from experimental data was inefficient and highly time consuming. Additionally, the final histogram was not always built from the correct subset of data in the past, requiring the researcher to start over from scratch. We have already solved the problem of time consuming histogram generation, and are moving on to building a web application to aid the storage, organization, and analysis of experimental data.
- The NeuroCiC web application uses HTML and CSS to render its interface. In order to describe the interface, I will describe the pages, their roles, design and connection.
 - /index - the homepage, it does not anything but welcome the user and directs it to the "Trial List" page through the navigation header generated by Hipercore.
 - /trials - this page shows the Trials stored in the database, they are sorted by date. A button is also present to create a new trial
 - /create-trial - creating a trial shows a pop-up modal window that asks for the name of the trial, the date, notes and ask for two input files: the spikes file and the head positions file. All of these latter are required in order to be able to create the trial.
 - /trial/:trial-id - it shows the summary of a trial. First, there is a histogram graph. Then there is a "Create new histogram" button in order to create new histograms. Then there is a table of histograms. The first column is a button to change the current histogram displayed on this page. The second column is the name of the histogram , if clicked it will redirect to a summary of this histogram. The third column is a delete button to delete this histogram(no confirmation is asked on purpose). At the end of the page is a text area for the notes that were entered when creating the trial.
 - /histogram/:histogram-id - it shows the summary of an histogram. Currently, it only shows the graphics visualization of the histogram and its name. In the future I hope to be able to show the raw datasets in a convenient way.
 - /create-histogram - it creates a new histogram from the input files of its parent trial. The first seen is the clusters of spikes. After selecting a few spikes, it is then possible to generate an histogram from the selection. Scrolling down, it gives the possibility to enter an angle and a range.
- The old version of the project was working with Python2, an older versions of Django and with the old version of Hipercore. The design was already present from the old version of the app and the goal was to update the app so that it would work again.
- The different stages were the following:
 1. Try to get NeuroCiC to display without any errors. This included modifying the routing, replacing old functions from the Hipercore, fixing the paths, change the print to function(major change from Python2 and Python 3).
 2. Now that NeuroCiC was displaying, the features had to be checked. Unfortunately, one package used in the old version of NeuroCiC(called Dajaxice) was not compatible with Python 3. Therefore, I had to rewrite the functionalities offered by this framework by following RESTful guidelines(these are considered as best practices) and to integrate those back in the HTML code.

3. From there, the application was working as expected. However, I wanted to make the app the most flexible possible so I decided to replace fixed urls by dynamic urls using Django.
- I developed this project using the HiPerCiC methodology and HiPerCore. The technologies used are jQuery, CSS, AJAX, HTML, Python3, Django among others.
 - Description of the files
 - *file_analysis.py* - this file contains functions to process the Spike and head positions file. This was not written by myself.
 - *models.py* - contains the data models that are stored in the database. In our case, we have Histogram and Trial.
 - *navbar.py* - contains a Python dictionary of the following text:url. These are used by HiPerCore to generate custom navigation header for each app(the big yellow headers with neurocic in our case).
 - *settings.py* - contains the settings for the application. This file is loaded by hipercore. In Neurocic, we only set the database parameters but we could also define background jobs, etc.
 - *urls.py* - contains the url, the functions to call when they are reached and also the possibility to name them. This is an important file that determines what the urls look like, how they are going to be called. They are a major part of the application design process.
 - *views.py*: contains the functions that are called when an URL is reached. It usually return some kind of HttpResponse.
 - */templates/*: folder that contains the html files for the application.
 - */static/*: folder that contains the CSS and Javascript files.
 - I removed a lot more codes(the framework Hipercore I wrote makes everything much smoother and simpler) than I added. Actually, according to Git stats I removed 12,000 lines of code and added about 1,200.

Please find the code at <http://stogit.cs.stolaf.edu/hipercic/neurocic>.

Here are a couple links to check out in order to get a better understanding of the NeuroCiC application: Links to google presentation, protocol pdf, and Dick Brown classes

- Pr. Brown's class notes on NeuroCiC from Bio 291: BioMaths Math and Biology
- Pr. Brown's notes from Alumni in College
- Manual protocol replaced by NeuroCiC
- Final presentation for CS315 final

Here are a few screenshots:

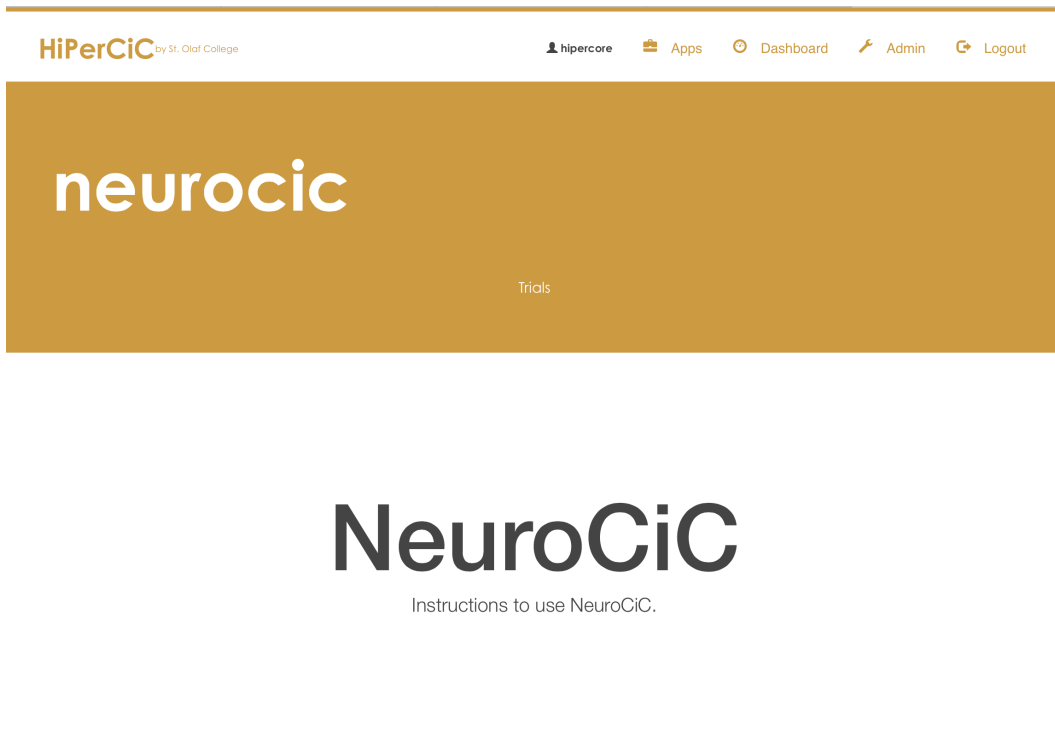


Figure 1: The homepage of NeuroCiC.

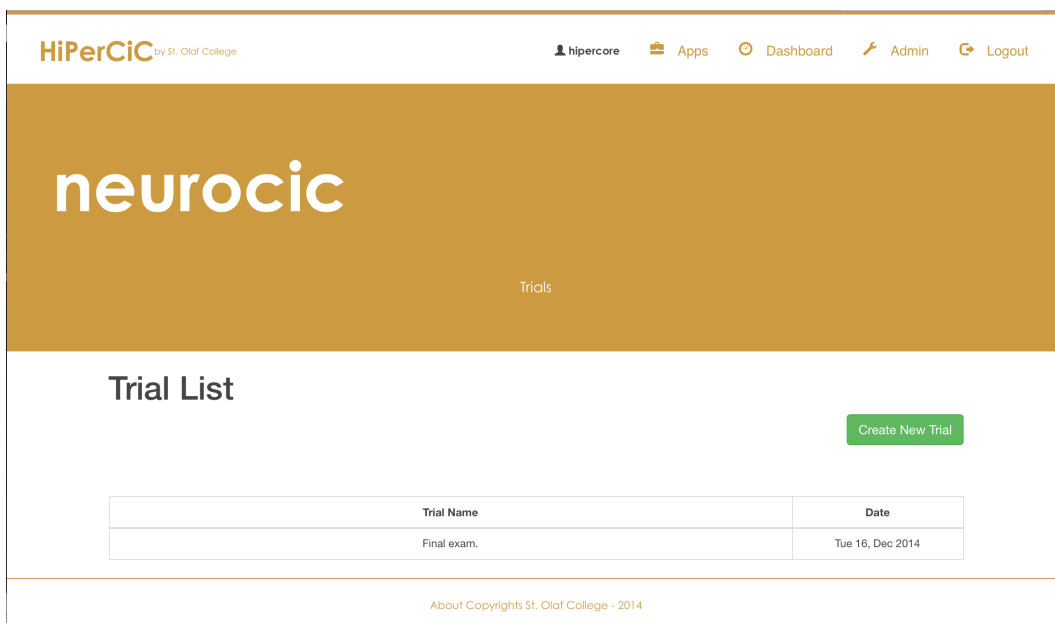


Figure 2: The list of trials.

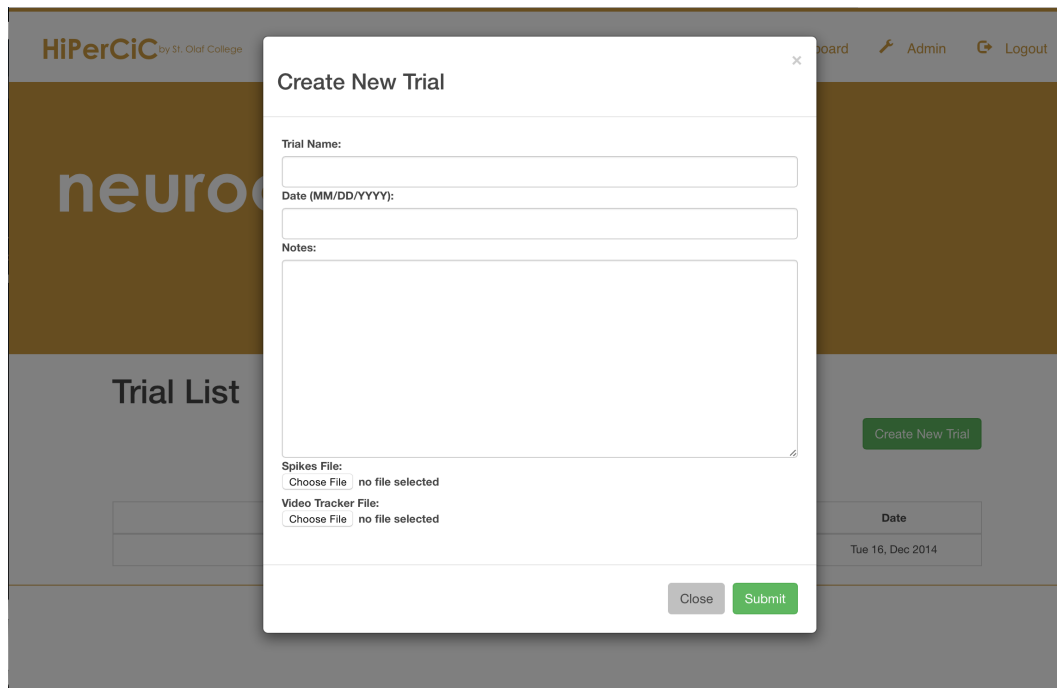


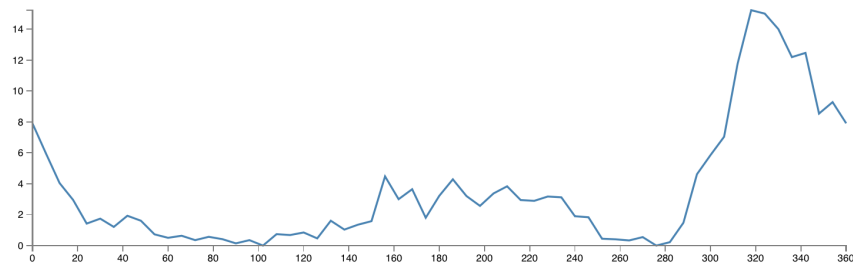
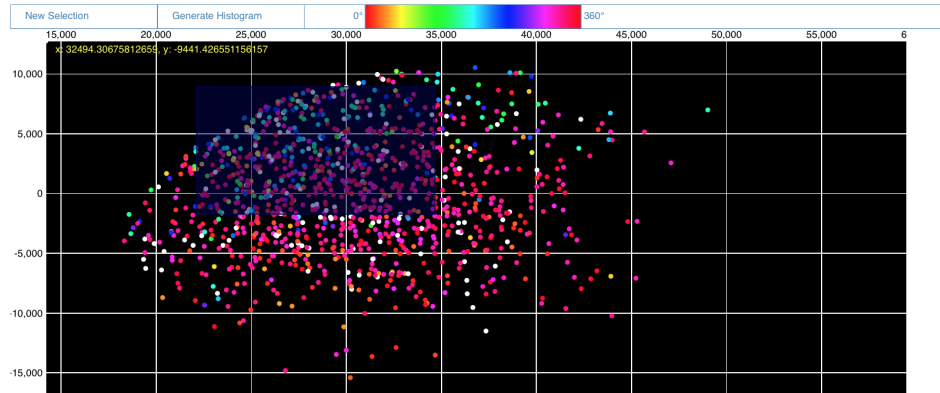
Figure 3: Creating a trial in a modal window.



Figure 4: Trial summary. We can see the histograms.

Create Histogram

◀ Final exam.



Get scatterplot parameters from angle range:

Angle:

Range:

Angle Parameters Generate Histogram from Angle Restricted Data

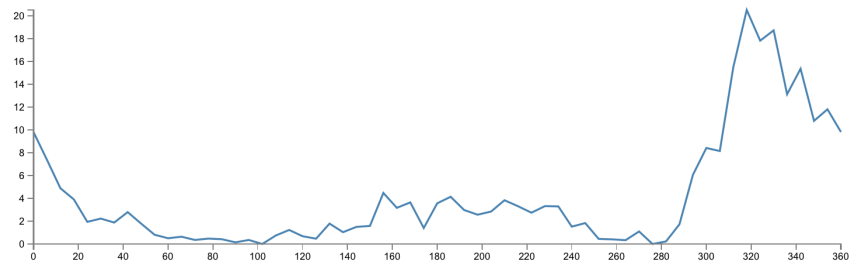
Name:

Save to trial Export to Excel

Figure 5: Creating an histogram involved selecting points from the cluster and generating an histogram from it. We can define parameters.

Random

◀ Final exam.



Delete this histogram

Figure 6: We can for now only see the histogram.