Advanced Practical Computer Concepts for Bioinformatics Final Project

Final Report

Sebastian Kyllmann

OdorBase: A Web Tool to Visualize Drosophila Melanogaster Olfactory Genes

Tool Description

Relative to the original project proposal, the name of the web tool was changed from SmellBase to OdorBase. Nonetheless, the project maintains the original dataset envisioned in the project proposal, consisting of 128 gene entries from FlyBase.org that share the gene ontology 0007608 corresponding to the description of “sensory perception of smell” as a molecular function. These data include a unique FlyBase ID, a gene feature type, a gene name, a gene symbol, descriptions of the coded protein’s function, and descriptions of the coded protein’s family. The data are included in text file within the SQL scripts directory, which also includes the python scripts that were used to convert the text file into an SQL table hosted on the SQL database skyllma1. The final browser consists of “Tinderesque” cards that display a given gene’s data. The web tool can be accesed at http://bfx3.aap.jhu.edu/skyllma1/final_project/code/odorbase.html. Using HTML buttons to surf the cards mimics the sensation of flipping through a deck of cards. As the user flips forward, new cards are generated by choosing a gene entry at random. Additional search functionality allows users to search for gene entries. The search is executed as an exact match against the FlyBase ID, the gene name, and gene symbol while a partial match is executed against the UniProt function and Protein Family columns. The user’s browser will alert for successful or erroneous searches and include the number of matches generated. For each match, a new card will be added to the end of the deck, which the user must then navigate to then visualize.

Design Process

I approached programming tasks by trying to make all the individual parts work first. I first refreshed my SQL and SQL connector skills to build the database. Then I worked towards getting a functioning CGI file to pull data from the SQL database. Then I worked towards having visually appealing HTML and CSS files. Finally, I worked on the JavaScript file, and tested the web tool to make sure all parts were integrated into the functioning of the web tool. Due to my relative unfamiliarity with JavaScript and JQuery, this final part was the longest and most arduous. As we learned this semester, I made sure to organize the files directory to respect the content separation principle.

This design process is archived in a folder on the class server title fp_brainstorms. This arose from a haphazard process of generating many "brainstorming" files. I was hesitant to use GitHub at first, but once I had a working minimal product, I migrated to exclusively using GitHub for versioning control on my local machine and committing changes to the online repository. The repository may be accessed at https://github.com/snk5040/AdvCC_finalProject.

Lessons Learned & Possible Improvements

Having become more familiar with GitHub through the project, I would now feel comfortable with starting and ending a project while using GitHub for version control the entire time. I would also pay consideration to having good documentation and having a good README file from the start.

During the project, I wrote some scripts that were able to access the FlyBase Reports via APIs (as described in the project proposal) to obtain associated XML data. It was interesting to work on this track, although in the end it did not seem appropriate as it was outside the scope of what we learned in class. However, I would like to continue learning and using APIs for future projects. To supplant this functionality, hyperlinks to FlyBase.org and UniProt were included in the gene cards for users to peruse more thorough data on the genes described in the card.

Throughout the project I strove to maintain good code commenting practices and code organization. I hope external feedback might confirm good code commenting or ask for improvements in this area. Nonetheless, the JavaScript code would benefit from better organization by defining more functions. I hope I can write better JavaScript code as I become more familiar with the language. I also hope to perhaps learn and use other popular JavaScript frameworks aside from JQuery.

The web tool itself could use improvements in the user experience. Firstly, there may be ways for the cards to be generated faster and with less lag when connecting to the database. Secondly, the web tool might benefit from a more unified or systematic visual design. Thirdly, when a search is executed, it may be useful to automatically surf the deck stack to the first match and its associated card. Overall, I strove to stress test the web tool as much as possible, yet some glitches may emerge as external users test the tool.