Advanced Practical Computer Concepts for Bioinformatics
Final Project Analysis
Claire Harelson – JHU ID: charels1

Protein Structural Biology Visualization and Analysis Tool

Structural biology is the study of the three-dimensional structures and dynamics of molecules. Protein structure analysis in bioinformatics is incredibly important as structure is directly related to function. Therefore, increased knowledge of a molecule’s structure can ultimately lead to enhanced understanding of its characteristics. Although a wide variety of structural bioinformatics tools exist, this particular tool provides a user-friendly web interface to explore protein information. Users interact with the web application through a browser while data processing occurs server-side via CGI and JavaScript files as well as through interactions with a MySQL database.

Obviously, the protein data bank (PDB) has an extensive trove of information on a wide variety of proteins, so as someone who works in the realm of structural biology, I chose to model my web application after this. Although my tool does not necessarily offer something distinctly different from the PDB (I aspire to one day be able to add additional, unique features!), this project was a great way to practice and demonstrate my ability to execute a full stack development.

To begin, I utilized the UNIX OS and filesystem by properly organizing my project directory to breakdown all the necessary components, CSS, HTML, JavaScript, Python, and the PDB files. Since the PDB has hundreds of thousands of protein files, I decided to pick one organism to focus on, Escherichia coli BL21 (DE3). Within this organism I chose about 30 proteins in order to minimize space taken on the class server while still demonstrating the proof-of-concept of my tool. Next, I utilized BioPython to parse the PDB file data. This was surprisingly one of the more difficult portions of the project for me, yet learning how to understand and implement code from other developers is a crucial skill to hone for the workforce. After extensive reading of documentation, I was finally able to implement the BioPython PDB package to parse the data into a MySQL database through the connector tool. I chose to utilize MySQL for database management as it allows for easy manipulation and extraction of data. This design also provides a method in which my server-side data processing can be executed.

I implemented an HTML form to provide a user with the opportunity to enter a search term. Within this I included a placeholder value to give an example of how the search should be written as well as the addition of autocomplete using JavaScript. For the sake of this smaller tool model, I chose to hard code the options to autofill, however, for a larger scale development of this tool I would instead implement a call to the remote MySQL database. A CGI script was written to communicate with the HTML form, receiving the input data from the user and adding it to a MySQL database query. This query communicated with the PDB table I had created and returned the corresponding information associated with the protein ID of interest. The CGI form then output the results to an HTML template.

Both my HTML form and template utilized CSS to maintain content separation. Web page styling through the use of classes allowed me to create a more uniform and user-friendly visual. The form is simple to use and interactive, allowing the user to input their desired search with ease. The returned result is designed to be visually pleasing and includes various colors and formatting styles simply to give me the opportunity to practice some fun CSS!

There are quite a few components I would have ideally liked to add or change about this project if I had more time and a bit more skill. The amount of time and effort it took to simply display some data without my tool necessarily ‘doing’ anything additional was shocking to me, I definitely have a newfound appreciation for web developers and bioinformaticians! Although this was a small-scale project, I can see how I might implement this same logic to create a larger and more in-depth tool. My same approach could be used to include many more proteins if space on the server were not a concern. Furthermore, with a bit more technical skill I would hope to implement more of an analysis aspect, such as a feature to superimpose or compare molecules. And of course, the biggest challenge I faced was trying to include the ability to view the actual 3D model of the protein! This proved to be much more difficult than anticipated and despite my best efforts I was not able to achieve this. In the future I hope to improve my data visualization skills, starting by taking a Data Visualization class next semester! Overall, I definitely had hoped I could create more with this tool, however, simply honing the skill of stack development and understanding how each component works has provided me with the backbone of technical expertise needed for future endeavors. Most importantly, I now understand the importance of content separation and how useful it is for web development and other bioinformatics tools.