University of California Davis

Honors Contract

Winter 2018

Efficient Dynamic Visualization of Large, Sparse Networks

Author:
Rachel Hanna

Supervisor:
Professor Patrice Koehl

March 15, 2018

1 Introduction

Scientists have made great strides in their understanding of DNA and how it defines the functions of cells, traits of organisms, and creation of proteins. Proteins are responsible for many basic cell functions, from serving as antibodies, to transporting nutrients. Each protein serves a specific function, or set of functions, based on its shape and is formed by a specific combination of amino acids; however, we still have much to uncover about *how* amino acid chains determine protein folding. This "protein structure prediction problem" [Li and Koehl, 2014] inhibits scientists from expanding their knowledge of amino acid sequences to solve many problems in biology and medicine. Researchers have recently begun to solve this problem from a new angle - computer visualizations of networks. For example, researchers have generated 3-d models of protein sequences to improve drug design and understand new aspects of membrane binding. [Petrey and Honig, 2009] Yet, the accuracy of such models is still lacking, creating the need for new network visualizations that can predict protein folds. Each amino acid can be paired with a tuple of values (or three values, or more) and displayed graphically. These amino acid associated vectors can be combined one after another to visualize a complete protein and hopefully provide clues as to how certain patterns produce specific shapes. Despite the advanced technological tools we have developed to research proteins, one of our most powerful assets is still the human ability to visualize a data set and recognize patterns. For this reason, we want to develop an interactive tool to support visualization for a 3-d trace of a protein sequence.

Specifically for this project, I was asked to develop a webpage tool to display vectors that could represent amino acids. For this purpose, I learned several programming languages - including JavaScript and HTML - and made use of JavaScript libraries D3 and Plot.ly. I was also asked to look over the previous and current research related to this problem, and report my experiences and findings after completing several graphical tools for displaying networks. I completed both a 2-dimensional and 3-dimensional version.

2 Method and Results

Our tools for completing the project included HTML and CSS, and multiple JavaScript libraries. I used JavaScript's D3 data visualization library for the beginning of the project, when designing the 2-d network visualization, and eventually switched to plot.ly for the final 3-d network.

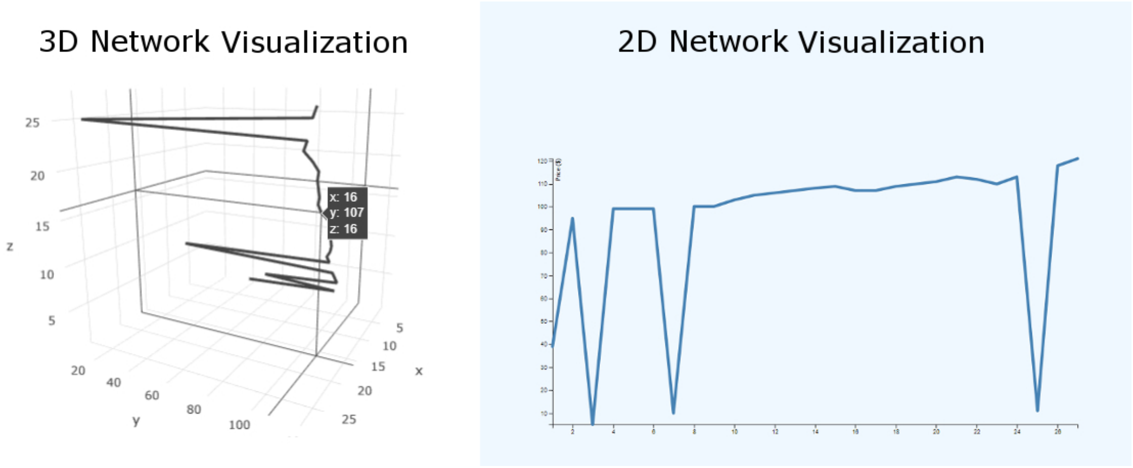
I began developing the network visualizations using JavaScript's D3 library, since there were many examples of 2-dimensional plots similar to our goal. D3 provides an extensive library of tools for manipulating and visualizing data-driven documents; these include countless functions to set domains, reorient graphics, and produce lines or plots from data files, all using JavaScript, CSS, and HTML. Ultimately, D3 gives the programmer greater control over their project than most data visualization frameworks - as I later discovered after switching to Plot.ly. While D3 enables endless possibilities, it also requires a large time investment to examine the documentation and understand its many capabilities. For a simple 2-d graph, however, D3 was straightforward and time-efficient to use.

Working from an example from the d3 gallery, I was able to alter the JavaScript to display a graph with 2 axes. D3's simple tools for reading in tsv files allowed me to separate the data from the code, making it easy to allow user-inputed data in the future.

After completing the initial visualization I continued to use D3 to attempt a 3-d version. While there were several examples of 3-d objects, their design was complicated and difficult to adjust without ruining the structure of the code. For example, D3 has the capability to display a 3-d scatter plot, wherein the data points are given by a random number generator. However, given that the display of the grid was also tied to this function, I could not simply replace it with a tsv input function similar to my 2-d visualization. Instead, my efforts to incorporate both continued to fail. Rather than spend more time working with D3, I turned to a new resource, plot.ly, to design the second half of my project.

Plot.ly was a much simpler library to utilize, especially since there were multiple examples of 3-d plots. While I found that the D3 library had nearly endless capabilities for my project, Plot.ly was most time-efficient. Due to simpler front-end features and common-sense design, it left the important design decisions to me, while dealing with the tedious aspects - such as defining domains and rotations

- on the back-end. The benefits were comparable to using a template rather than organizing an essay from scratch. After working with Plot.ly, I believe it would have made even the 2-d graph appear more professional and allowed for a simpler design process.



Visualizations

3 Discussion

These network visualizations can be further improved in the future by adding new features. For example, grouping both 2-d and 3d networks together would allow for easy access to both, eliminating the need to switch between windows. Writing out the sequence above the visualization tools would also make this more usable for researchers. Perhaps knowledge could also be gained by finding a 4-d or 5-d approach with the introduction of line thickness or color, allowing researchers to view certain amino acids or groupings set apart from each other in yet another way. Ultimately, these two network tools could be organized into a clean interface that displays a title, sequence, both graphs, and options to add more information to the graphs - or even to input the data into a window rather than a separate file.

Through this project, I learned many helpful tools for designing webpages. I developed proficiency in HTML and Apache, allowing me to display my progress easily along the way. I experienced the difficulty and reward of using libraries like

D3, and developed an affinity for Plot.ly. In a greater sense, I practiced the art of learning new libraries, reading through documentation, and learning by example. Up until this experience, I had improved through book learning; after this project I have the confidence to continue learning on my own using online resources.

While I understand how to develop several 2-dimensional objects with D3 and 3-dimensional line plots with plot.ly, I would like to have learned more about basic JavaScript first, before grappling with libraries. This would have allowed me to understand more of the D3 gallery examples and refine my visualizations. While this aspect is somewhat outside the scope of this introductory project, I wanted to witness first-hand what researchers accomplish with such visualization tools for protein sequences. I would be curious to understand how researchers map amino acids to specific values, and even how they scale back the values of a 5-d or 3-d vector to something smaller and easier to handle. With those exceptions, I completed all of my project learning goals. I improved my understanding of HTML and gained the tools needed to display useful data on a webpage.

I found this project both enjoyable and rewarding, and have a desire to program with HTML and JavaScript in the future. The design aspect interests me more than other types of programming, like data structures in C++. I would like to make use of this knowledge in some form of research where I can utilize web programming tools to help solve problems in other disciplines - like biology or chemistry. For now, I will expand upon the basic skills I developed and attempt my own web design projects using HTML and JavaScript. I will work with my website to program new visualization tools to provide data to businesses or online clients. While Plot.ly was more time efficient for this project, D3 sparked my interest because of its wide capabilities. Of the two, I will probably choose D3 for my future projects - especially 2-dimensional ones. It strikes me as a tool with a high learning curve, but with beautiful benefits. I will continue to go through examples from the D3 gallery to complete my understanding of both JavaScript and the Data Driven Document library.

References

[Li and Koehl, 2014] Li, Jie and Koehl, Patrice. (2009). 3D representations of amino acids - applications to protein sequence comparison and classification.

Computational and Structural Biotechnology Journal, 11:47–58.

[Petrey and Honig, 2009] Petrey, D. and Honig, B. (2005). Protein structure prediction: inroads to biology. *Mol Cell*, 20:816–817.