Contents of the Document

The code referenced in this document is available at: https://github.com/Shashank979/Projects

1. Update of college courses fall 2020

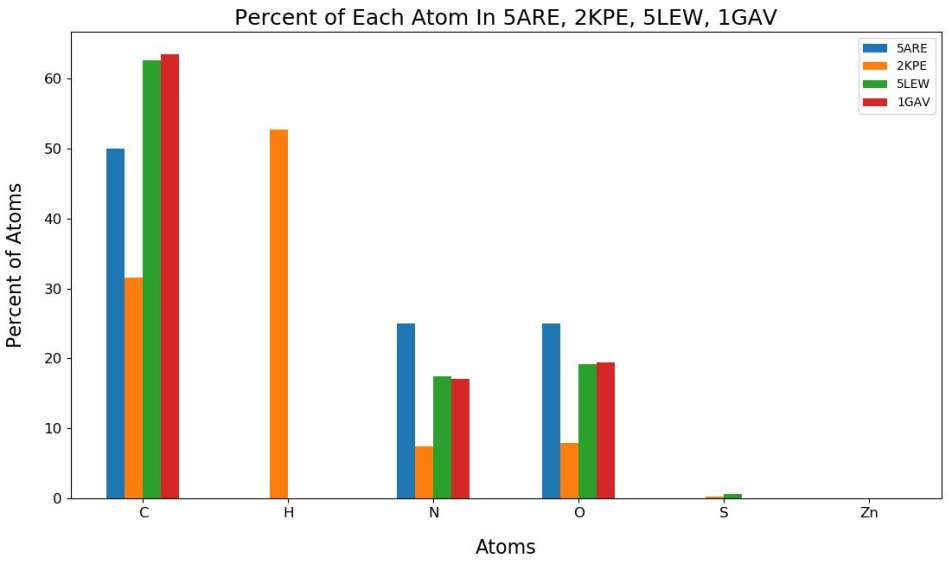
Molecular Genetics (MCB 250) at the University of Illinois, Urbana-Champaign : Grade A+

2. Summaries of Self-Directed Programming Projects

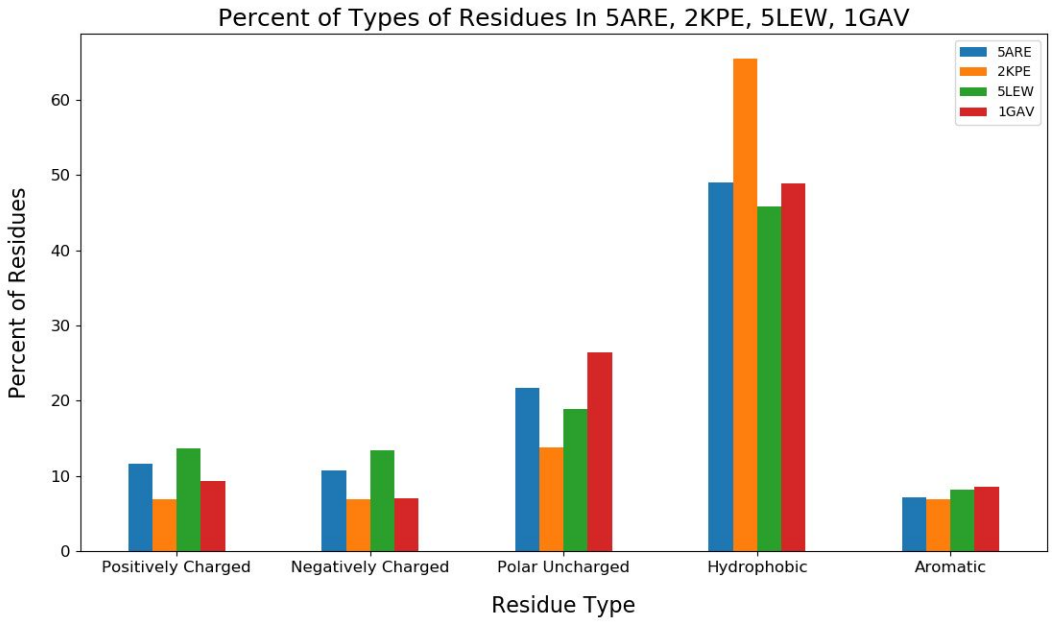
Project 1 : Protein Comparison

Description: Program allows a user to graph and compare selected proteins for various properties. The properties are : atomic composition, amino acid composition or types (for example hydrophobic, polar uncharged etc.) of amino acids. The number of proteins that can be graphed is based on the type of graphing done whether it be atoms or amino acids. Proteins are fetched from a protein data bank using the Python library prody. The proteins are fetched by their four letter codes. Graphs are done using matplotlib. Some examples of the graphs produced are shown below and in the examples folder. Another program also allows the user to compare these different characteristics of the proteins but for the sum of all the proteins in the whole database. So the user can see, for example, what the average percentage of negatively charged amino acids is for all the proteins in the database. This is under the “FullStats” folder. Some questions for future are : Can one graph the percentage of turns, helices and sheets that make up the smaller regions of the protein’s structure? How might this connect to other characteristics?

Example 1: Graph of the percent of each atom (carbon, hydrogen, nitrogen, oxygen, sulfur, and zinc) in four different proteins (5ARE, 2KPE, 5LEW, 1GAV). We can see that the protein (5ARE) has a higher percentage of carbon atoms than the protein 2KPE.



Example 2: Graph of the percent of different types of amino acids (positively charged, negatively charged, polar uncharged, hydrophobic, aromatic) in four different proteins (5ARE, 2KPE, 5LEW, 1GAV). We can see that the protein (5LEW) has a higher percentage of negatively charged amino acids than the protein 2KPE.



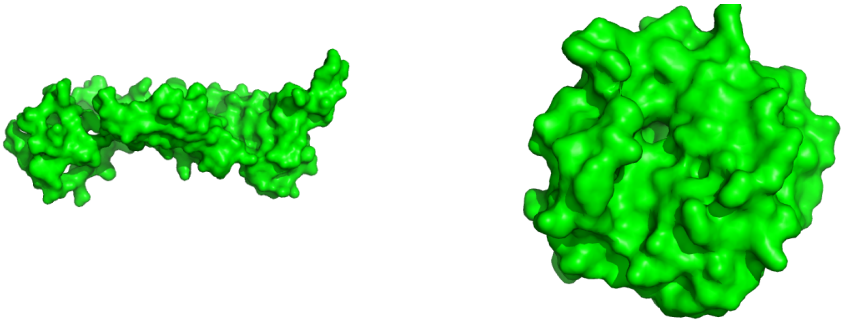
Project 2 : Soccer Teams Comparison

Description: Allows a user to graph different statistics of multiple soccer teams over time. The user can specify a time range, a statistic and then pick soccer teams from almost any top league. The output is a graph with time as x-axis and the statistic as y-axis. For example one can graph: Goals scored per season from 2004-2016 for Barcelona, Real Madrid and Sevilla. In this way a user is able to actually graphically compare statistics from multiple different teams instead of just seeing statistics of one team on a website. Thus one can easily investigate questions like “are the top Spanish league teams winning more games per season than the top English league teams.” The program grabs and properly organizes this data from the website: fbref.com, and utilizes some methods to prevent from being detected and banned from the website.

Project 3 : Machine Learning Classification of Proteins

Description: The program trains a machine learning model to recognize from a protein image, whether it is globular (spherical in shape) or fibrous (elongated in shape). The result of the program is a machine learning model that can be given images of proteins and guess whether the protein is globular or fibrous. Another program I wrote as part of this project used commands from pymol (molecular visualization application) to automate download of images of globular and fibrous proteins from the 3d protein database. These images were for the model to be trained on and validate its training (test its accuracy). The model was able to reach an accuracy of 100% in some runs but the data size was small and not so diverse. I used the PyTorch library for the machine learning part. Some questions for future are : Can I incorporate the amino acid sequences along with the training images? Does the sequence affect if a protein is globular or fibrous?

The following images are examples of a fibrous protein (left) and globular protein (right) that were used in the program for validation:



Notes
Project 1 and 2 were completely self-directed. Project 3 was with the help of a mentor.

3. Advent of Code Challenge

Advent of Code (AoC) is an annual online coding challenge. Everyday from December 1 to December 25 a problem is released. Each problem has two parts and for each part you complete you get one star. Thus there are a total of 50 stars per year. This year I raced alongside more than 150,000 people to earn these stars as fast as possible. I started participating in 2018 and I was able to get 17 stars that year. In 2019 I was able to get 22 stars. This December in AoC 2020 I was able to get 48 stars. One of my favorite problems this year was day 11. In the problem you are given a grid of characters which change based on a set of rules related to their adjacent characters. In part 1 the goal is to figure out when the grid state stays the same for two turns in a row. Doing these problems there are many concepts that I've used and learned such as: circular lists, chinese remainder theorem, 2d to 4d arrays and many types of algorithms. These kinds of problems have made me better at and even more excited about problem solving and programming.

Advent of code 2020 problems : https://adventofcode.com
My solutions : https://github.com/Shashank979/Advent-Of-Code-Solutions