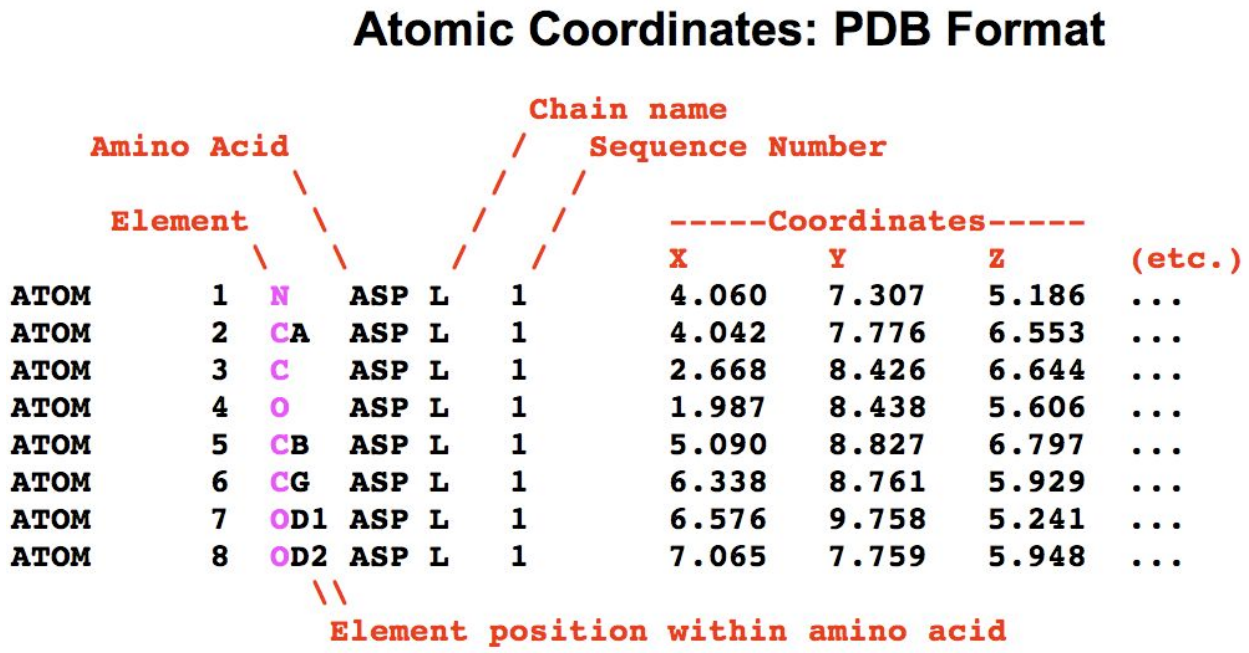
Data Review for Foldit Project
Burhan Azeem & Sabina Chen

Data Structure



The image above illustrates the structure of the .pdb files we will obtain from Prof. Seth Cooper.

The game Foldit is broken into different challenges. For each challenge, there are tens of thousands of users that played the game and within that many different moves, the users made. We have data on most users, snapshots of what moves they made, and a tree structure of how these moves relate to each other (you can return to older checkpoints).

Each snapshot in the game is a .pdb file. The .pdb file list every atom in the protein, its coordinates in 3D space, and how it relates to its neighbors. This data is typically read into a secondary library like biopython, pymol, or rosetta that creates a 3D visualization and outputs the energy score. The data from Foldit already contains the calculated energy scores.

Data Access

We were given access to this data set from Prof. Seth Cooper. He is the creator of the Foldit game.

Quality

For our project, we want to train AI to be able to the best next move when folding a protein. This task has two difficult challenges: 1) there is a large state space of possible actions and 2) to get the highest score end score you take actions that do not maximize the score for that move. The result is that it is difficult to analytically solve protein folding. Human users were important for Foldit

because they were able to identify ‘smart moves.’ We believe that we can train the AI on the game plays of 10,000s of human players such that it would be able to learn from them and make its own smart moves.

Licensing

The data is restricted use because it results from human participants. As a result, we have completed training on using this type of information in a proper manner and have signed terms of use. The data was anonymized before we were given access.

Data Storage

In this case, the dataset is not large in size. The entire history of all the moves by all the players can be downloaded as a .zip file and stored and managed on our computer’s hard drive. If we need more space during the project, Burhan has a 5Tb hard drive.

Preprocessing

As we are obtaining our data from the research group that built the game, our data is incredibly pure. We will not need to cleanse or clear any data for it to be readable. We shall need to identify ‘good players’ which we define as those achieving an end score higher than the initial score and we shall separate those players from the rest of the data. We shall also need to build our AI such that it can read the .pdb file, identify all the atoms (which have an ID in numerical order), and be able to change their coordinates in 3 dimensions.

A data review describing the datasets that you have acquired for training on your final project. This should be approximately two pages long with the following information:

- **What your data looks like** with a few examples. Detail the number of classes/labels, the number of samples per label, the dimensions of each sample (100x100 pixels, 5 seconds per sound clip...etc), and the storage size of the dataset (how many GB/MB?).
- **How/from where you obtained this data**
- **Why you think this data is "good enough"** for what you want to do for your final project.
- **Licensing**: Is this data open sourced or only for restricted use? What restrictions does it have? Does your dataset contain PII (personally identifiable information)?
- **Where you are storing the data**. This cannot just be a link to a kaggle dataset or some website where you plan to download the data. A huge challenge in ML is storing, parsing through, and computing on large datasets that may not fit on your computer. We want to see that you have mostly gotten past this hurdle.
- If transfer learning (using a pre-trained neural network like SqueezeNet or MobileNet) is an important part of your final project, you also need to: (1) describe the base neural network you are working off of, (2) why it is a good fit for your project, and (3) why you think the additional data you have for your project will play well on top of it.
- What/whether **preprocessing** is required to get the data into a form that you can input into your system. You do not have to have this implemented yet, but you do need a reasonable idea of how to do this