

Jonathan Potter
Andrew Lutsky
Rohit Nandakumar
Shashank Katiyar

Predicting Secondary Structure Protein Folding

Background

The earliest concepts of secondary structure protein prediction emerged after the work of Linus Pauling and Robert Corey, who discovered the alpha-helix and beta-sheet in the early 1950s. Among the first well-known algorithms to identify secondary structure is the Chou-Fasman method. In the early 1970s, Peter Chou and Gerald Fasman developed a statistical method to analyze the frequencies of alpha-helices, beta-sheets, and turns. This method yielded about a 50-60% accuracy with predicting secondary structure. However, some limitations of Chou-Fasman include the fact that apart from immediate neighbors, Chou-Fasman does not account for longer-range interactions [1].

Later on, during the late 1970s, as an improvement to the Chou-Fasman method, the GOR algorithm was developed by Garnier, Osguthorpe, and Robson. This algorithm utilized information theory to identify the likelihood of alpha-helices, beta-sheets, and turns. However, this algorithm had similar limitations to the Chou-Fasman method, with longer-range interactions not being accounted for. The GOR algorithm has an accuracy of 60-70% [2].

In the 1990s to present, machine learning techniques have been used to predict secondary structure. Most notably, AlphaFold was developed by Alphabet's DeepMind to predict protein structure at a remarkable 92.4% accuracy [4]. This is considered the state-of-the-art approach.

In our project, we will be focused on developing the Chou-Fasman and GOR algorithms. We will then visualize and compare the predictions of both algorithms using both 2D and 3D visualization techniques. We will use CASP classifications into three different secondary

structure types. The typical method for validating secondary structure prediction is to compare to DSSP classification[8]. The DSSP classification categorizes things into 8 larger groups of secondary structures. However, it is very common to translate these 8 larger groups into just three different structures and it is common to narrow down into H for Helices, E for sheets, and C/L for coils/loops[8]. This is why in our project we narrowed down secondary structures into only three different classifications, as there is precedence. Additionally, Chou-Fasman and GOR I only have predefined parameters for four secondary structure classifications for these types of classifications. We aim to compare the two algorithms using the DSSP classifications of x-ray crystallography data which have been cleaned and made into a test dataset in the SECNET 2018 validation dataset.

Chou-Fasman Method

The Chou-Fasman method primarily involves utilizing a set of predefined statistics of the frequencies of particular amino acids in certain secondary structures. For example, amino acids that were more likely to be found in an alpha-helix secondary structure would have a higher propensity value for helices for that amino acid. In this way, a dictionary of sorts can be defined for specific amino acids. Chou-Fasman primarily relies on these propensity values for specific amino acids in conjunction with a series of predefined steps in order to determine the overall secondary structure prediction.

There are several limitations of the Chou-Fasman method, including but not limited to low accuracy, independence assumption, small dataset for defining the parameters, and failure to account for local environment. It has been reported that Chou-Fasman is in fact only approximately 50-60% accurate in identifying correct secondary structure. This inaccuracy can

be chalked up to several defining characteristics of the algorithm. The initial set of parameters was only defined off of a dataset of approximately 15 different proteins and was reported as up to 80% accurate on globular proteins; this in fact was later defunct. Additionally, later methods employed the inference of independent neighboring residues, although Chou-Fasman does use a four residue window to determine the probability of a turn secondary structure.

The actual algorithm for the Chou-Fasman method takes the following steps [9], as shown in Figure 1.

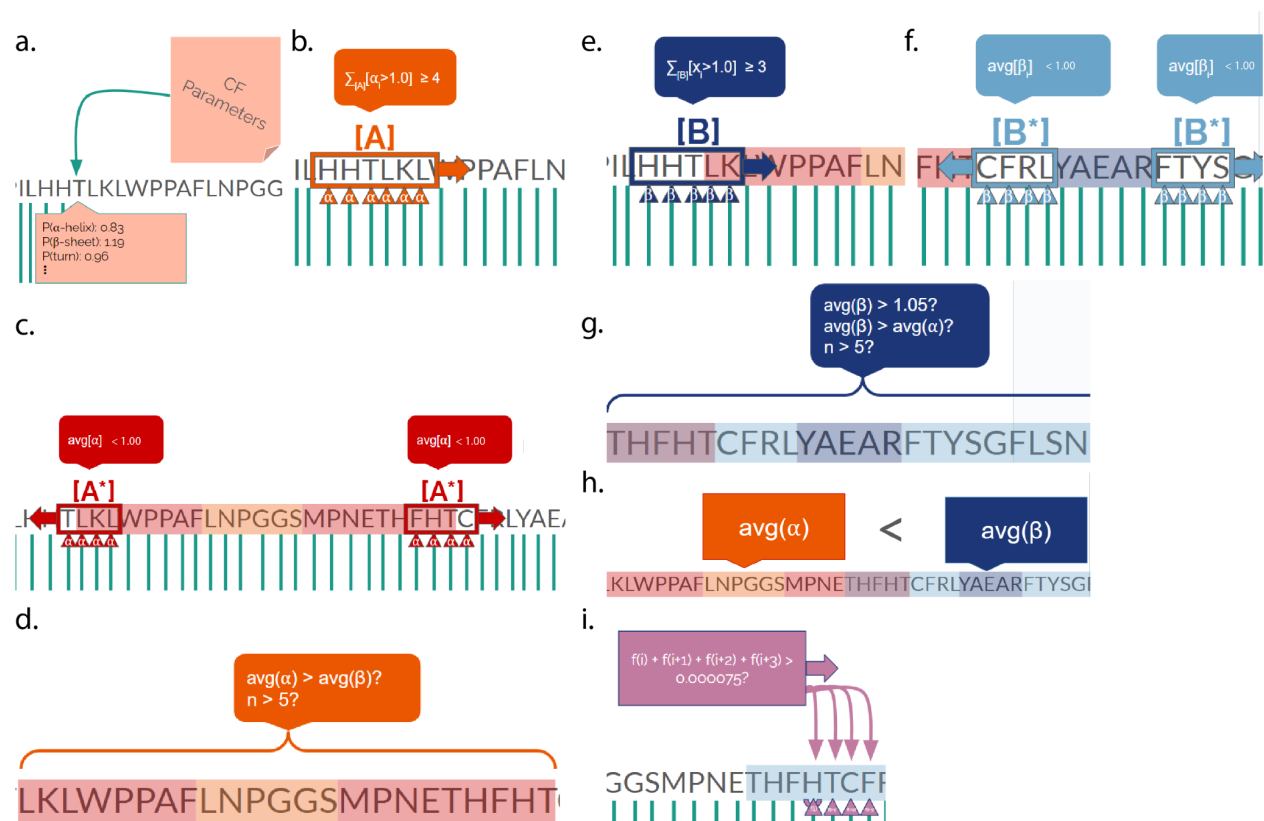


Figure 1 - A visual representation of the Chou-Fasman algorithm.

- 1) The amino acids in the peptide are all associated with their respective parameters from a table of weights for alpha helices, beta sheets, and bends that were determined in the original Chou-Fasman paper (Figure 1a). These scores were initially determined by using relative frequencies heuristics calculated from around 30 proteins with known structures

(This small dataset became a large criticism of the paper, one that was never updated due to better algorithms being created that relied on machine learning).

- 2) The peptide is then scanned with a sliding window, starting at one end, until a region is found where four of six contiguous amino acids are found with helix parameter scores (assigned in step 1) greater than 1.00 (Figure 1b). This region is then labeled as the root of a helix. The ends of the root are then extended in their respective directions one at a time until the 4 terminal amino acids have an average helix score of less than 1.00 (Figure 1c). The whole section is then evaluated: if the selected region is longer than 5 acids and the average helix score is greater than the average sheet score, the region is added to a list of alpha helices (Figure 1d).
- 3) Step 2 is repeated until the entire peptide has been evaluated for helices.
- 4) A similar method is then performed for identifying beta sheets. The major differences include: The initial sliding root window requires 3 of 5 acids to have a sheet score greater than 1.00 (instead of an acid score) and when evaluating the selected region (Figure 1e), the average beta score of the flanking windows must be less than 1.00 (Figure 1f), and the average sheet score must be greater than 1.05 (instead of 1.00), with an average sheet score higher than the average helix score (Figure 1g).
- 5) Now, armed with a list of all possible sheets and helices, the selected regions are scanned for overlaps. If two helices overlap, the one with the highest helix score is kept, with the others discarded. The same applies for sheets using sheet scores. If a helix and a sheet overlap, the region with the highest score out of both helix and sheet scores wins, and the region is kept and labeled as such, with the other regions discarded (Figure 1h).

- 6) Finally, bends are identified by considering the turn score and calculating the “bend score”, which is the sum of the $f(i)$, $f(i+1)$, $f(i+2)$, and $f(i+3)$ scores (all assigned in step 1) of the acid being investigated as well as the three subsequent acids, respectively. If the bend score is greater than 0.000075, the average turn score of the four acids is greater than 1.00, and the average beta sheet and alpha helix scores of the tetrapeptide are less than the turn score, then the initial amino acid is labeled as a beta-turn (Figure 1i).

GOR Method

The GOR method primarily relies on the idea of using a window of parameters and information theory to determine the secondary structure of a given residue. The original dataset of predetermined parameters were derived from the study of directional information plots created by Robson & Suzuki[7] and only from a sample size of approximately 25 proteins. These information plots were fitted and smoothed to give better parameters, particularly challenged by small sample size. Similar to Chou-Fasman, there is a set of parameters for each residue. Unlike Chou-Fasman these parameters correspond to the information content of a particular residue appearing in a window of residues in a specific position. The information content is a quantification of the probability of a particular event occurring from a random variable, in this case that residue being one of predetermined secondary structures, helix, sheet, or coil/loop. Additionally, in contrast to Chou-Fasman, we find that instead of using a single residue and extending out those regions, we evaluate the information content or self-information across a 17 window range. A simplistic implementation of GOR 1 can be approximated as the overall equation for summing the information content, which can be found in Figure 2. The steps involved can be broken down into the following three simplistic steps.

3. We then find the maximum information content for each secondary structure and assign that structure to residue j , where residue j is the central residue of the window.

The GOR I method just involves taking the maximum information content from the available measurements and choosing that secondary structure. There have since been major improvements to the GOR I method, seeing an increase of up to approximately 8-9% for globular proteins(GOR V was released in 2005). These improved methods involve using larger sample databases to derive the parameters, improved statistics, and using pairs and triplets of amino acids instead of single residues.

Visualization

To visualize the results of our algorithm, we implemented two types of visualization, 2D and 3D. 2D visualizations- In these types of visualizations, we draw a simple 2-dimensional image, with a collection of vertical lines corresponding to each amino acid in the protein. The color of each line is according to the type of secondary structure allotted to the corresponding amino acid. For this, we used the go graphics package for Golang [5].

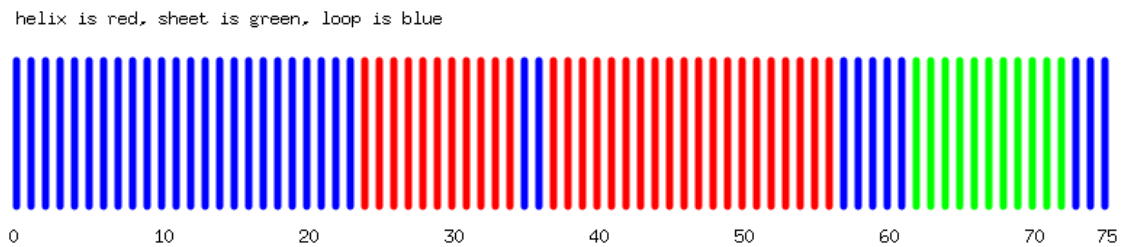


Figure 3 - 2D visualization for ubiquitin protein. Green: Beta sheets; Red: Helices; Blue: Loops

3D visualizations- For these visualizations, the user must operate the program in the “array” mode. In the array mode, the input is in the form of an array of ensembl-ids of genes. The gget package [6] is used to extract the protein sequence corresponding to the gene/s and the algorithm is run on those sequences. Also, from the Alphafold database, the pdb for the protein is downloaded by the code. The output of the algorithm is overlaid on the pdb structure in the form of colors, and an existing HTML template with the code to display a protein 3d structure is modified to display this structure with the prediction’s colors. The result looks like the following for ubiquitin protein.

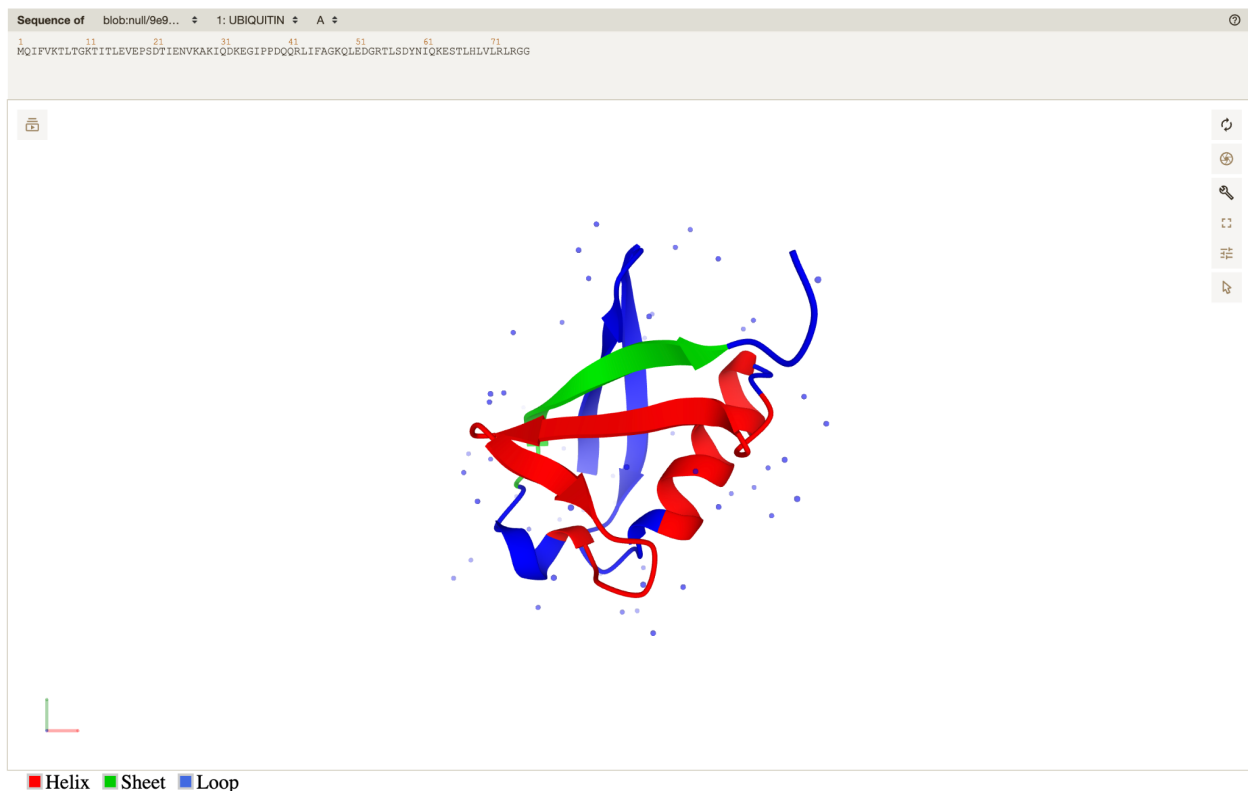


Figure 4 - 3D visualization for ubiquitin protein. Green: Beta sheets; Red: Helices; Blue: Loops

Results/Analysis

In order to assess the accuracy of each prediction algorithm we tested their accuracies using a small selection of proteins randomly selected from the SECNET 2018 validation dataset. (<https://github.com/sh-maxim/ss/tree/master>) The accuracies of each implemented secondary structure prediction algorithms were written out into a csv file, along with their secondary structure accuracies. This information has been compiled into Figure 5.

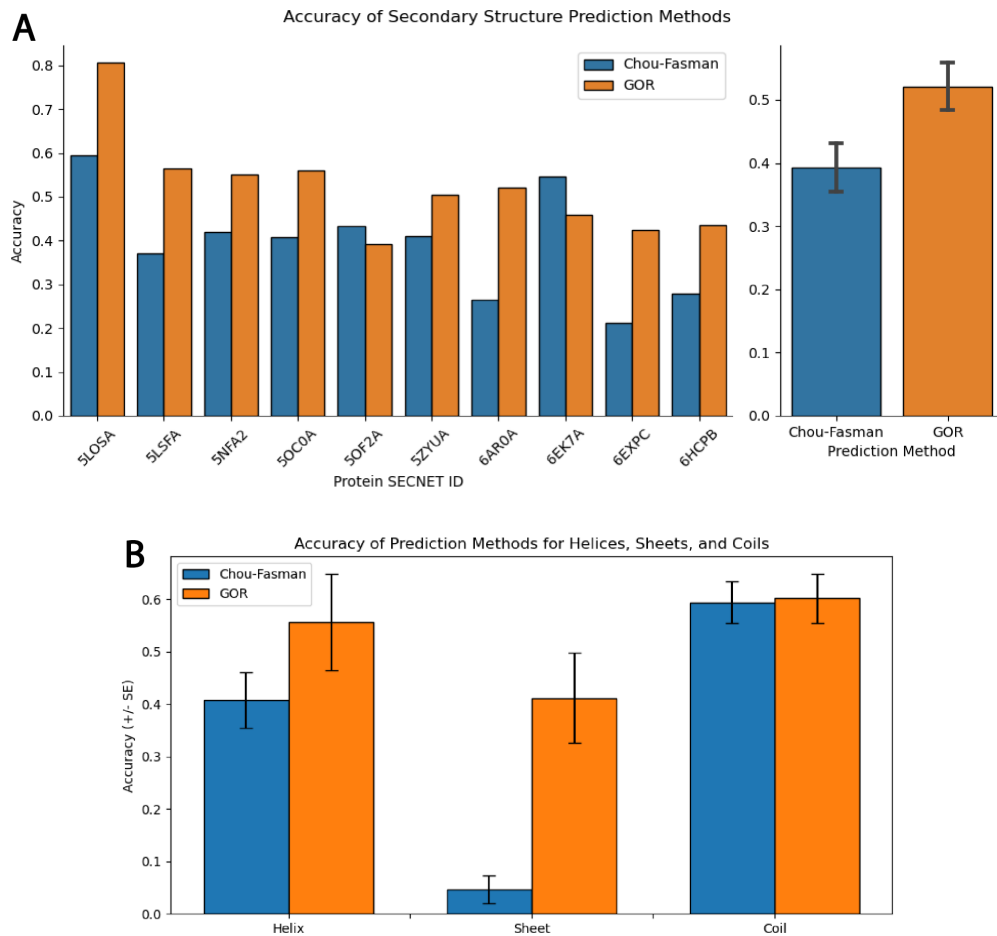


Figure 5 - Comparative accuracy of Chou-Fasman and GOR secondary structure prediction

methods. (A) Accuracy of individual protein prediction and overall accuracy (+/- SEM). (B) Prediction of each secondary structure prediction algorithm per secondary structure (+/- SEM).

Comparing the results of Chou-Fasman and GOR to real biological examples is extraordinarily useful at assessing how useful they are as tools. We can see that although originally reported to be anywhere from 50-60 % accuracy, these two algorithms fall somewhat short on real world examples (from Secnet2018). Our internal assessment of the algorithms appear to demonstrate that the algorithms themselves are about ~5-10 % less accurate than reported. This can be attributed to small validation test sets. Additionally, if we examine Figure 5B we see that Chou-Fasman performs quite terribly in comparison to GOR at identifying sheets. This can be attributed to multiple different issues with the validation and the nature of Chou-Fasman in the first place. First, Chou-Fasman does not examine as large of a window comparatively to GOR, and beta sheets are relatively global secondary structures of proteins by nature. That is that completely different parts of the primary structure align, so having a larger window size would lend itself to more accurate predictions for these more “global” secondary structures. Additionally, the validation dataset only contains relatively small proteins, which could also lend to the low accuracy of sheets. Lastly, it could be possible that since Chou-Fasman was derived from such a small dataset of proteins, that there were not enough sheets present in the proteins to lend accurate propensities for the beta sheet secondary structure. Interestingly enough, the accuracies of GOR and Chou-Fasman for the other secondary structures perform within the margin of errors of each other. Overall, we can conclude that GOR is significantly better than Chou-Fasman at predicting beta sheets and in secondary structure in general, despite its simplicity relative to Chou-Fasman.

Future Advancements in the Field


Most recent advancements in the field have been spearheaded by the adoption and growth of machine learning-based models. The first notable usage of machine learning models for the prediction of secondary protein structure was PSIPRED [10], which achieved an accuracy of 76.5% for three-state (helix, sheet, and turn) prediction. Later ML-based algorithms such as JPred4 and RaptorX had an accuracy of 82.0% and 84.0% accuracy with three-state prediction [11][12].

Among the biggest advances in the field has been the development of AlphaFold by Alphabet's DeepMind. AlphaFold predicted tertiary structure at an astonishing 92.4% accuracy, and has even been termed as one of the most important achievements in the field of artificial intelligence [13][14].

We predict future advancements in the field to be based on current trends in artificial intelligence, notably with rapid advances in attention-based models such as transformers and large language models (LLMs). Historically, attention-based models were largely used for natural language processing (NLP) tasks. Attention-based ML models can be trained to look at sequences of amino acids, similar to how natural language is a sequence of words, to better predict protein structure. Meta's ESMFold (released in 2022) [15] has been able to leverage LLM-based architecture to rival AlphaFold's accuracy while being substantially faster.

References

1. "Chou–Fasman Method." Wikipedia, Wikimedia Foundation, 26 Aug. 2023, en.wikipedia.org/wiki/Chou%E2%80%93Fasman_method#:~:text=The%20method%20is%20at%20most,modern%20machine%20learning%E2%80%93based%20techniques.

2. “Secondary Structure Prediction.” *Secondary_structure_prediction*, www.bionity.com/en/encyclopedia/Secondary_structure_prediction.html. Accessed 29 Nov. 2023.
3. Garnier J, Gibrat JF, Robson B. GOR method for predicting protein secondary structure from amino acid sequence. *Methods Enzymol.* 1996;266:540-53. doi: 10.1016/s0076-6879(96)66034-0. PMID: 8743705.
4. Hale, Conor. “Google DeepMind’s Protein-Predicting Alphafold, OCT Imaging Inventors Claim Lasker Awards.” *Fierce Biotech*, 21 Sept. 2023, www.fiercebiotech.com/medtech/google-deepminds-protein-predicting-alphafold-oct-ima-ging-inventors-claim-2023-lasker.
5. fogleman/gg: Go Graphics - 2D rendering in Go with a simple API. GitHub. Published September 28, 2021. Accessed November 29, 2023. <https://github.com/fogleman/gg>
6. pachterlab/gget:  gget enables efficient querying of genomic reference databases. GitHub. Published November 16, 2023. Accessed November 29, 2023. <https://github.com/pachterlab/gget>
7. Robson, B. & Suzuki, E. (1976). *J. Mol. Biol.* 107, 327-356.
8. Taner Z. Sen, Robert L. Jernigan, Jean Garnier, Andrzej Kloczkowski, GOR V server for protein secondary structure prediction, *Bioinformatics*, Volume 21, Issue 11, June 2005, Pages 2787–2788, <https://doi.org/10.1093/bioinformatics/bti408>
9. Center for Molecular and Biomolecular Informatics (CMBI): Chou-Fasman Parameters. Published Jan 29, 2018. Accessed November 29, 2023. <https://swift.cmbi.umcn.nl/teach/aainfo/chou.shtml>

10. DT, M. L. K. (n.d.). *The PSIPRED protein structure prediction server*. Bioinformatics (Oxford, England). <https://pubmed.ncbi.nlm.nih.gov/10869041/>
11. Drozdetskiy, A., Cole, C., Procter, J., & Barton, G. J. (2015, April 16). *JPRED4: A protein secondary structure prediction server*. OUP Academic. <https://academic.oup.com/nar/article/43/W1/W389/2467870>
12. Wang, S., Li, W., Liu, S., & Xu, J. (2016). RaptorX-Property: A web server for protein structure property prediction. *Nucleic Acids Research*, 44(Web Server issue), W430–W435. <https://doi.org/10.1093/nar/gkw306>
13. Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Židek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., ... Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873), Article 7873. <https://doi.org/10.1038/s41586-021-03819-2>
14. Toews, R. (2023, October 5). *Alphafold is the most important achievement in AI-ever*. Forbes. <https://www.forbes.com/sites/robtoews/2021/10/03/alphafold-is-the-most-important-achievement-in-ai-ever/>
15. Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Židek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., ... Hassabis, D. (2021). Highly accurate protein structure prediction with

AlphaFold. *Nature*, 596(7873), Article 7873.

<https://doi.org/10.1038/s41586-021-03819-2>

Contributions

Jon Potter

Jon contributed to the project by constructing the code for most of the Chou-Fasman algorithm backend, everything along the pipeline from a string of amino acids to the collection of labeled features. Jon also laid the foundation for the GOR I method. Jon contributed to the presentation by constructing visual diagrams for the Chou Fasman algorithm slides. Jon contributed to the final report by writing about the Chou-Fasman algorithm and how it was applied to the problem at hand. Finally, Jon attended the weekly meetings to discuss the project's direction and group work assignments.

Andrew Lutsky

Andrew contributed to the overall project by implementing the identifying turns section of the CF algorithm as well as completing the overall implementation the GOR I algorithm and performing exploratory data analysis to visualize comparative accuracies of the two algorithms. Andrew contributed to the final report by writing about the GOR algorithm and how it was applied to the problem, DSSP classification, and the results section of the report. Additionally, Andrew attended the weekly meetings to discuss the project's direction and group work assignments.

Rohit Nandakumar

Rohit contributed to the project by creating the input files for FASTA and CIF sequences. Additionally, Rohit assisted in the creation of the Chou-Fasman method, focusing on beta sheet identification. Rohit was heavily involved with creating testing functions for ten functions. In regards to the non-coding aspects of the project, Rohit wrote up the “Background” and “Future Advancements” sections of the paper and presentation. Lastly, Rohit attended weekly meetings with the group to discuss direction and progress.

Shashank Katiyar

Shashank was heavily involved in creating the visualization tool for the project as well as implementing CLI workflow options. He tied together the various components of our project and provided a CLI for the user to interact with. Subsequently, he also created detailed instructions for the usage of the program in the Readme of the project and the demonstration video. Shashank also attended weekly meetings with the group to discuss direction and progress. Shashank contributed to the presentation by talking about the various different approaches taken to visualize the protein structure and to discuss future direction and progress.