Group 4: Jonathan Sebastiani, Ilias Kladakis

Assignment: Final Research Project

Date: 3/19/2025

Analysis of Current Protein Structure Prediction Algorithms

Introduction

Protein Structure Prediction

Protein structure prediction is a computational approach that aims to determine the three-dimensional structure of proteins based solely on their amino acid sequences. This process involves sophisticated algorithms that analyze the linear sequence of amino acids (primary structure) to predict how the protein will fold into secondary structures (alpha-helices and beta-sheets), arrange into a complete three-dimensional shape (tertiary structure), and potentially interact with other protein subunits (quaternary structure). As proteins function primarily through their three-dimensional conformations, accurate prediction methods provide crucial insights into how these molecules operate within biological systems.

The importance of protein structure prediction cannot be overstated in modern biological research and medicine. Errors in protein folding are directly linked to numerous diseases, including Sickle Cell Anemia, Alzheimer's, and various cancers. By accurately predicting protein structures, scientists can better understand the molecular mechanisms behind these conditions and develop targeted therapeutic approaches. Additionally, structure prediction accelerates drug discovery by enabling structure-based drug design without the need for expensive and time-consuming experimental methods like X-ray crystallography. This computational approach allows researchers to screen potential drug candidates more efficiently, significantly reducing the cost and time required for pharmaceutical development.

CASP

The Critical Assessment of Protein Structure Prediction (CASP) [7] is a community-wide experiment that serves as the gold standard for evaluating the state of the art in protein structure prediction. Held biennially since 1994, CASP provides an objective testing ground where research groups blindly predict the structures of proteins that have been experimentally determined but not yet publicly released. The competition aims to establish the current capabilities in the field, identify recent progress, and highlight areas where future research efforts should be focused. As the most reputable evaluation platform in the field, high-scoring algorithms in CASP gain significant credibility within the scientific community. Recent CASP competitions have showcased remarkable advancements in AI-based prediction methods, revolutionizing the field and bringing unprecedented accuracy to protein structure modeling.

Algorithmic Background

The algorithms that we chose to compare are the AlphaFold2, RoseTTAFold, ESMFold, and I-TASSER algorithms. These all are tertiary protein structures prediction models that use AI to produce more accurate results. These are all critically acclaimed algorithms [1] - [4] which will help us to compare the top algorithms that are used in the field of tertiary protein structure prediction.

The AlphaFold2 model is created by google. It utilizes deep learning and transformer neural networks to predict protein folding. The algorithm had unmatched accuracy in CASP14 [1] and it can predict novel folds without templates. Although, it requires extensive computational resources and lacks transparency in some algorithmic processes.

The RoseTTAFold algorithm was developed by the University of Washington's Baker Lab. David Baker was awarded the Nobel Prize in Chemistry in 2024 for developing their RoseTTAFold algorithm [5]. Its breakthrough strength is that it can predict protein structures efficiently using a three-track neural network system [2]. But, it has lower accuracy for long protein sequences.

The ESMFold model is developed by Meta AI and is based on evolutionary-scale modeling [3]. It is designed to predict protein structures rapidly and has a good balance between speed and accuracy for best targeting medium sized protein sequences. Unfortunately, this creates long run times for short sequences and inaccuracies for long sequences.

Finally, we have the I-TASSER (Iterative Threading ASSEmbly Refinement) algorithm. This algorithm predicts 3D protein structures by integrating template-based modeling with ab initio approaches. It has high accuracy and is widely used in structural biology. Even though it has consistent performance in CASP [4], it is computationally expensive and accuracy depends on template availability. This highly limits the opportunities that I-TASSER can provide for biologists.

Motivation and Methods

AI Generated Protein Structure Predictions

Finding physical protein structures through biological means is expensive and time intensive [6]. Although, this essential step in biology cannot be overlooked. Errors in protein folding cause diseases like Sickle Cell Anemia. Understanding protein structures aids in drug development and accurate predictions will improve biological research and clinical applications.

Predicting protein structure through computational methods has become more prevalent and accurate recently because of the large cache of data that has been accumulating in recent years. The larger databases that sequence and create protein structures, the more accurate AI models become. But, AI creates a black box for researchers and creates more issues in a new way with tracing back data. The results that AI models present are extremely hard to trace back, making proving results difficult.

To ensure that there is a safe way to compare AI generated models to real structure predictions done in a lab, we want to create a program that graphs overlap of an algorithm to a baseline model. Our goals are to evaluate and compare protein structure prediction algorithms, identify the best algorithm balancing accuracy and efficiency accessibility, improve future bioinformatics tools for faster and more accurate predictions, and provide an overall evaluation and proposed advancement of protein structure prediction algorithms. Gaining protein structure prediction accuracy for specific algorithms will not only help to defend the accuracy and preciseness of the models, but will also help to push prediction model creators to the most accurate areas of data collection, model structure, and many more important areas in the field.

Comparing Algorithms

A .pdb file can be downloaded through each algorithm and used for comparison. These files contain coordinates for each atom in the protein structure that was predicted. These coordinates can then be scaled accordingly and overlapped to see structural similarity. The unit of measurement that we used for testing structural similarly is RMSD (Root Mean Square Deviation). RMSD is calculated as…

$$RMSD=(x2-x1)2+(y2-y1)2+(z2-z1)2$$

The RMSD of two structures gives a great baseline to see how far away each atom is to each other. We can then take an average of the RMSD for every atom to see the overall structural similarity.

For a better understanding, an RMSD value between 0.0 to 1.0 Å indicates an excellent match, suggesting near-identical structures often seen when comparing experimental and high-quality predicted models. A value within the range of 1.0 to 2.0 Å represents good similarity, where structural differences are minimal and usually limited to side chain positions or slight backbone shifts. When the RMSD falls between 2.0 to 4.0 Å, it suggests moderate similarity, often reflecting conformational changes, flexible regions, or minor prediction errors. An RMSD value above 4.0 Å signifies significant differences, which could indicate structural inaccuracies in predictions, large conformational changes, or challenges in the modeling process.

Hemoglobin (HBA1)

Using Hemoglobin HBA1 [8] for the comparison of protein structure prediction algorithms provides a reliable benchmark. The gene name HBA1 encodes a protein with a length of 142 amino acids, which is an ideal size for structural comparison and can be handled by most prediction algorithms. It also has a complex structure, making it a great option to test high end algorithms. The amino acid sequence of HBA1 is…

“MVLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHFDLSHGSAQVKGHGK KVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASL DKFLASVSTVLTSKYR”

HBA1 plays a critical biological function by facilitating the transport of oxygen from the lungs to various peripheral tissues, making it a valuable target for evaluating the accuracy of structural prediction algorithms.

Program Description

Coding Environment

In order to code these visualizations and comparisons, we needed to use a powerful package that would allow us to overlap these visualizations on a 3 dimensional plane with user intractability. We used python to code our program seeing as we are the most familiar with python and it is the best option for data analysis and manipulability, where our coding was done in the Visual Studio Code IDE.

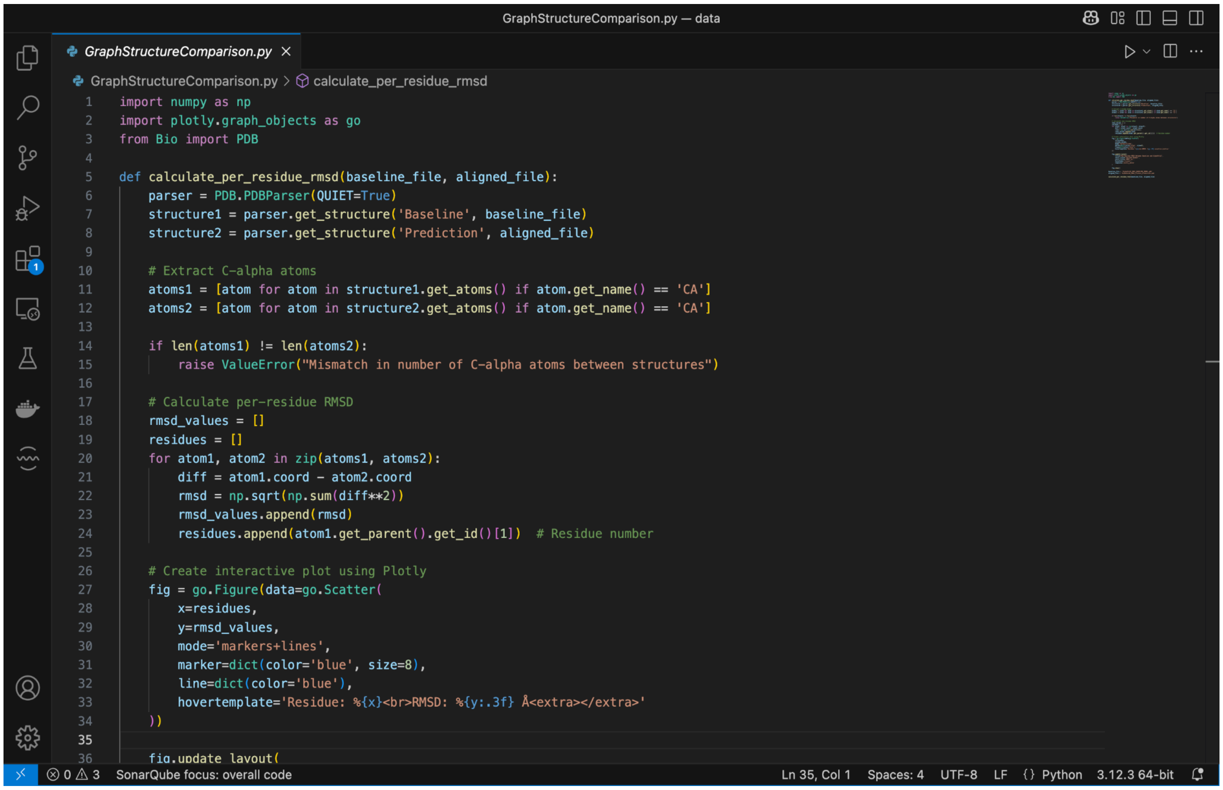
We used the ‘plotly’ package to plot the graphical comparisons with residue distance. This allows us to create interactive visualizations that users can look over to see which areas have the greatest distance from the baseline structure.

We also used the ‘nglview’ package to create 3 dimensional interactive visualizations of the protein structure overlaps. Creating this visualization helps us to better understand variability in protein structures. In future work, researchers would be able to take these visualizations and make quaternary structures to see more clearly how individual protein structures interact with each other. The packages ‘numpy’ and ‘Bio’ were used throughout to help make biological computations and assess amino acid sequence specific data, comparisons, and visualizations.

Our Code

The motivation for our code was to compare both overall RMSD and RMSD for each individual residue. Geographical information from each protein structure prediction algorithm was compared to a baseline protein structure for HBA1 through a .pdb file. We created two python files which produce a line graph of RMSD for the compared protein structure, and a visualization of the overlapping protein structure graphs. Both of these graphs are interactable, meaning that the 3d structure can be spun, zoom in and out, and hover over specific residues to get their information, while the line graph will tell which residue correlates to which data point.

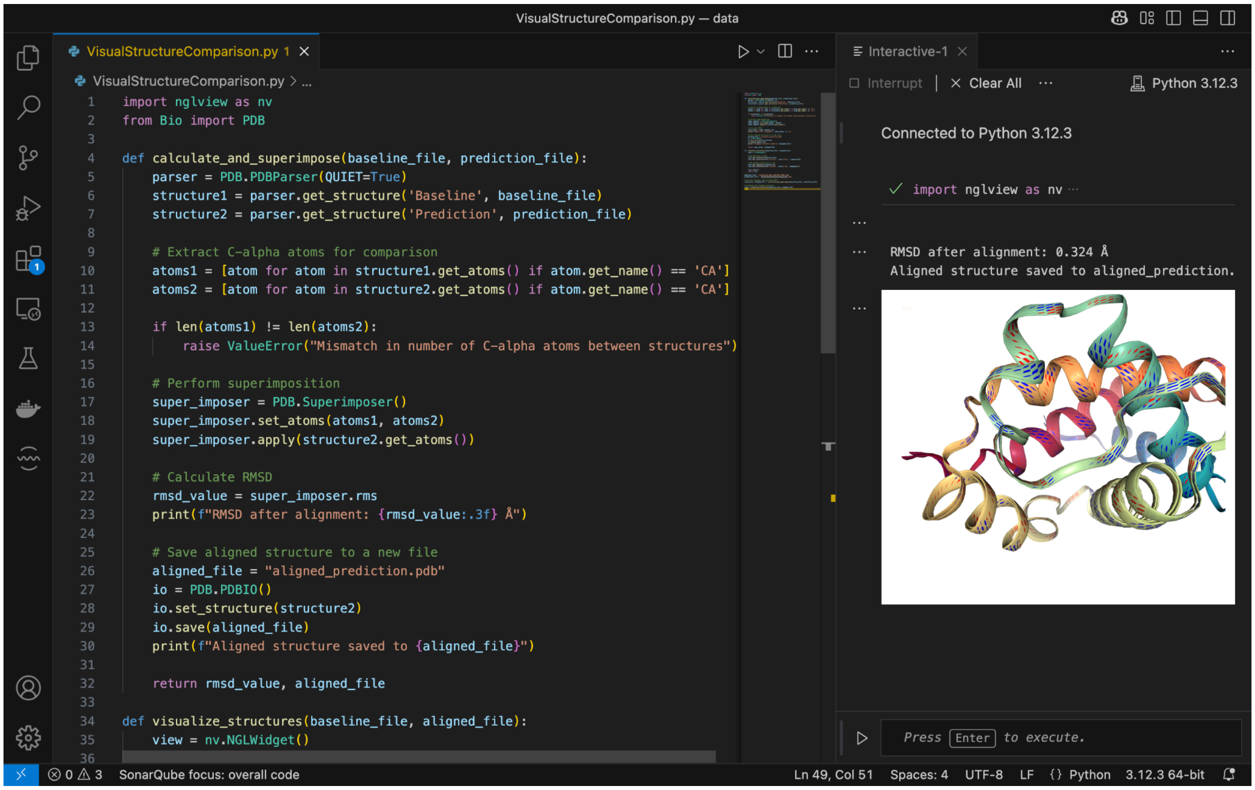
Our line graph was created through a python file named ‘GraphStructureComparison.py’. This code compares the structural similarity of two protein models by calculating the per-residue RMSD. It uses Biopython to parse and extract the C-alpha atoms (representing the backbone structure) from two PDB files: a baseline model and a predicted model. After confirming that both structures have the same number of C-alpha atoms, it calculates the RMSD for each residue by measuring the Euclidean distance between corresponding atoms. The results are stored and plotted using Plotly to create an interactive graph in a separate window through the local machine. The graph displays residue numbers on the x-axis and RMSD values on the y-axis, with hover functionality to show precise residue information. This visualization helps assess structural deviations and identify regions of significant difference between the baseline and predicted models.


GraphStructureComparison.py — data
GraphStructureComparison.py
GraphStructureComparison.py > calculate_per_residue_rmsd
⊗ 0 ⚠ 3
SonarQube focus: overall code
Ln 35, Col 1
Spaces: 4
UTF-8
LF
{} Python
3.12.3 64-bit

```python
import numpy as np
import plotly.graph_objects as go
from Bio import PDB

def calculate_per_residue_rmsd(baseline_file, aligned_file):
    parser = PDB.PDBParser(QUIET=True)
    structure1 = parser.get_structure('Baseline', baseline_file)
    structure2 = parser.get_structure('Prediction', aligned_file)

    # Extract C-alpha atoms
    atoms1 = [atom for atom in structure1.get_atoms() if atom.get_name() == 'CA']
    atoms2 = [atom for atom in structure2.get_atoms() if atom.get_name() == 'CA']

    if len(atoms1) != len(atoms2):
        raise ValueError("Mismatch in number of C-alpha atoms between structures")

    # Calculate per-residue RMSD
    rmsd_values = []
    residues = []
    for atom1, atom2 in zip(atoms1, atoms2):
        diff = atom1.coord - atom2.coord
        rmsd = np.sqrt(np.sum(diff**2))
        rmsd_values.append(rmsd)
        residues.append(atom1.get_parent().get_id()[1])  # Residue number

    # Create interactive plot using Plotly
    fig = go.Figure(data=go.Scatter(
        x=residues,
        y=rmsd_values,
        mode='markers+lines',
        marker=dict(color='blue', size=8),
        line=dict(color='blue'),
        hovertemplate='Residue: %{x}<br>RMSD: %{y:.3f} Å<extra></extra>'
    ))

    fig.update_layout(
```

Figure 1: Python code for graphically comparing RMSD distance from the baseline model through an interactive line graph.

Our 3 dimensional structure comparison model was created through a file named 'VisualGraphComparison.py'. This code uses the Biopython and NGLView packages to perform structural alignment and comparison of two protein models. It begins by reading and parsing two PDB files, one representing the baseline model and the other a predicted structure. The program extracts the C-alpha atoms from both structures, which represent the protein backbone and are commonly used for structural comparisons. After confirming that both models have the same number of C-alpha atoms, the code applies a superimposition using the Superimposer class to minimize the RMSD. The calculated RMSD value is then printed, indicating how closely the predicted structure aligns with the baseline. The aligned model is saved to a new file called "aligned_prediction.pdb" for further analysis. For visual inspection, the code uses NGLView to generate a 3D interactive visualization where the baseline model is shown in blue and the aligned prediction in red using a cartoon representation. This visualization allows users to rotate, zoom, and explore the structural differences, providing both a quantitative and qualitative understanding of the model alignment.



VisualStructureComparison.py — data
VisualStructureComparison.py 1
VisualStructureComparison.py > ...
Interactive-1
Interrupt
Clear All
Python 3.12.3
Press Enter to execute.
0 3 SonarQube focus: overall code
Ln 49, Col 51 Spaces: 4 UTF-8 LF {} Python 3.12.3 64-bit

```python
import nglview as nv
from Bio import PDB

def calculate_and_superimpose(baseline_file, prediction_file):
    parser = PDB.PDBParser(QUIET=True)
    structure1 = parser.get_structure('Baseline', baseline_file)
    structure2 = parser.get_structure('Prediction', prediction_file)

    # Extract C-alpha atoms for comparison
    atoms1 = [atom for atom in structure1.get_atoms() if atom.get_name() == 'CA']
    atoms2 = [atom for atom in structure2.get_atoms() if atom.get_name() == 'CA']

    if len(atoms1) != len(atoms2):
        raise ValueError("Mismatch in number of C-alpha atoms between structures")

    # Perform superimposition
    super_imposer = PDB.Superimposer()
    super_imposer.set_atoms(atoms1, atoms2)
    super_imposer.apply(structure2.get_atoms())

    # Calculate RMSD
    rmsd_value = super_imposer.rms
    print(f"RMSD after alignment: {rmsd_value:.3f} Å")

    # Save aligned structure to a new file
    aligned_file = "aligned_prediction.pdb"
    io = PDB.PDBIO()
    io.set_structure(structure2)
    io.save(aligned_file)
    print(f"Aligned structure saved to {aligned_file}")

    return rmsd_value, aligned_file

def visualize_structures(baseline_file, aligned_file):
    view = nv.NGLWidget()
```

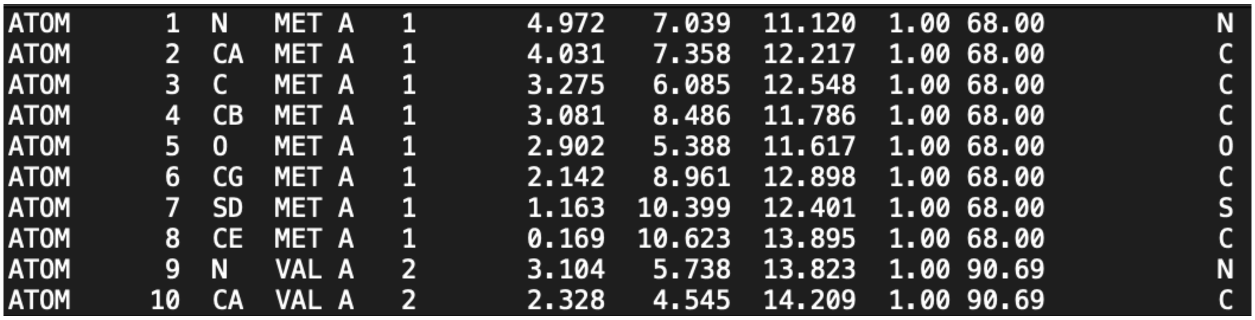
```
Connected to Python 3.12.3

✓ import nglview as nv ···

RMSD after alignment: 0.324 Å
Aligned structure saved to aligned_prediction.
```

Figure 2: Python code for generating 3d interactive visualization of prediction and baseline structures, as well as finding the average RMSD.

Data

The data was collected through the online websites of all the algorithms used in this research project. While AlphaFold2 and ESMFold were able to produce results through their online website, both RoseTTAFold and I-TASSER needed accounts to be created and results to be sent through email. I-TASSER required a scholarly email in order for results to be processed and sent to the user.

The data was collected by downloading a .pdb file which was then used for our analysis. Each file has an introductory section which gives credit to the creator and other important information. The raw data from the file is formatted in the way that it contains the element, amino acid, chain name, sequence number, and then x, y, and z coordinates respectively.



```
ATOM      1  N   MET A   1       4.972   7.039  11.120  1.00 68.00           N
ATOM      2  CA  MET A   1       4.031   7.358  12.217  1.00 68.00           C
ATOM      3  C   MET A   1       3.275   6.085  12.548  1.00 68.00           C
ATOM      4  CB  MET A   1       3.081   8.486  11.786  1.00 68.00           C
ATOM      5  O   MET A   1       2.902   5.388  11.617  1.00 68.00           O
ATOM      6  CG  MET A   1       2.142   8.961  12.898  1.00 68.00           C
ATOM      7  SD  MET A   1       1.163  10.399  12.401  1.00 68.00           S
ATOM      8  CE  MET A   1       0.169  10.623  13.895  1.00 68.00           C
ATOM      9  N   VAL A   2       3.104   5.738  13.823  1.00 90.69           N
ATOM     10  CA  VAL A   2       2.328   4.545  14.209  1.00 90.69           C
```

Figure 3: Example of an .pdb file.

These coordinates can then be calibrated for comparison between other .pdb files which also contain this formatted information.

Testing and Results

AlphaFold2

The usability of the AlphaFold2 website was extremely high. Given a single search bar, you can search either their database of protein models or submit a sequence of your own which would give results between 5-10 minutes. There did not appear to be any length constraints for a given sequence. The results produced only a singular model which had a color overlay to show structure prediction confidence at each residue.

The results of the AlphaFold2 model [9] proved the best overlap with the compared model with an average RMSD of 0.049 Å and a high 17.990 Å. This extremely high overlap shows the high correlation between the AlphaFold2 predicted model and the AlphaFold2 database baseline model.

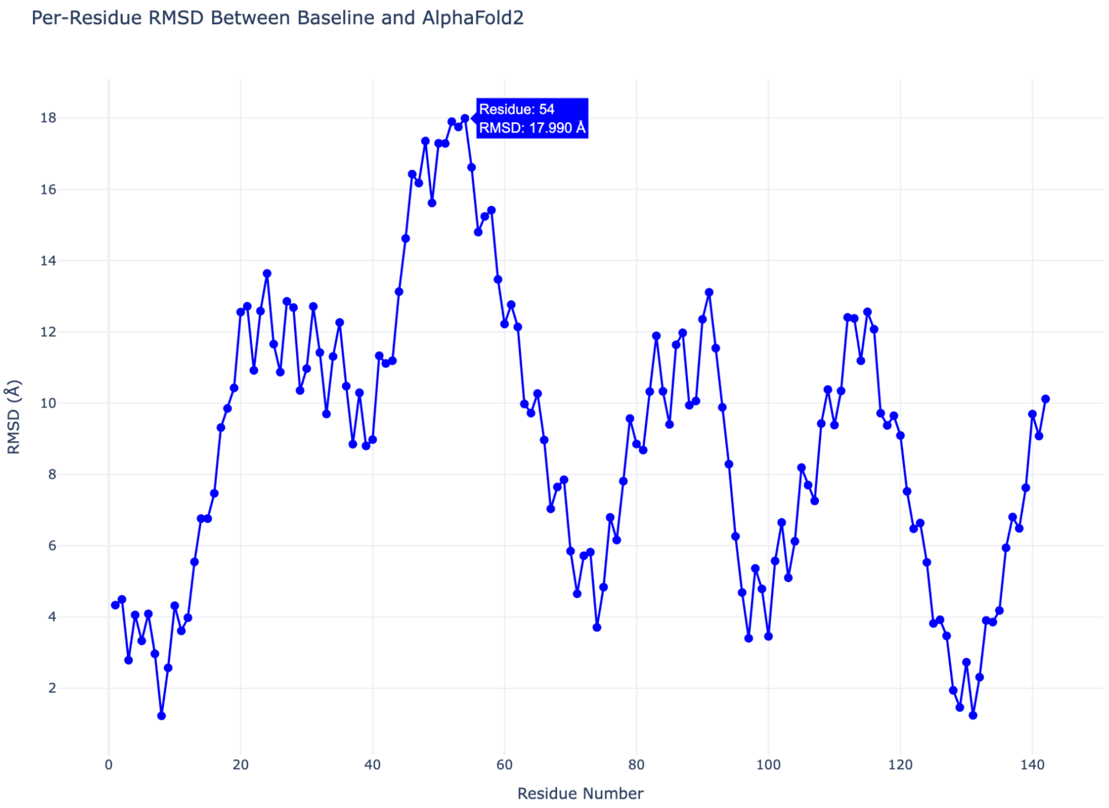


Figure 4: Line graph of AlphaFold2 RMSD. Average of 0.049 Å, high of 17.990 Å.

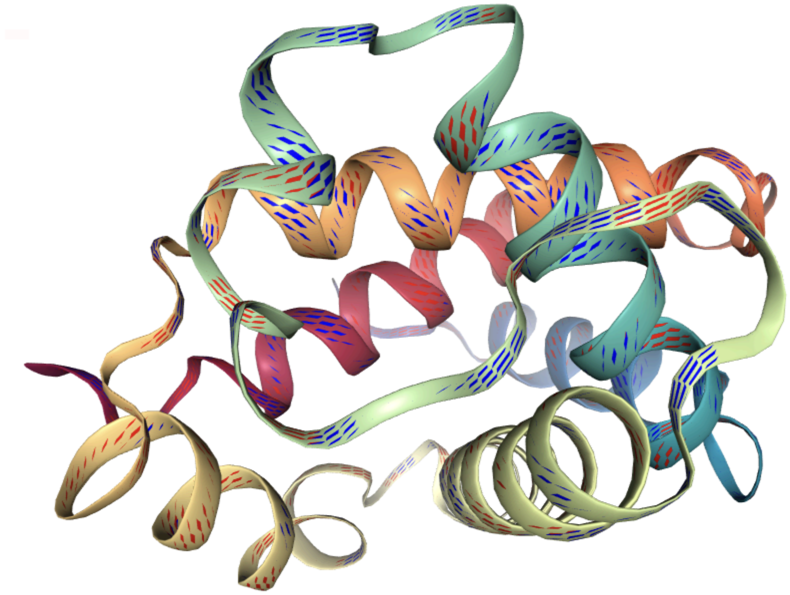


Figure 5: Visual overlap of baseline (blue) and AlphaFold2 model (red).

RoseTTAFold

The usability of the RoseTTAFold website was moderate. I sequence length between 26 and 501 was required which limited the protein sequence length the most out of all the algorithms. Results were not given through the website, but sent through an inputted email after account creation was made. Runtime isn't calculable because of sending the results through email. The results produced five models of best confidence which each had a color overlay to show structure prediction confidence at each residue. I picked model 4 which seemed to have the lowest overall predicted error based on a graph of error produced by the algorithm.

The results of the RoseTTAFold model [10] proved the worst overlap with the baseline model with an average RMSD of 0.621 Å and a high 51.129 Å. Although this is the worst overlap of the compared algorithms in this study, an average RMSD of 0.621 Å is still an excellent match to the baseline model.

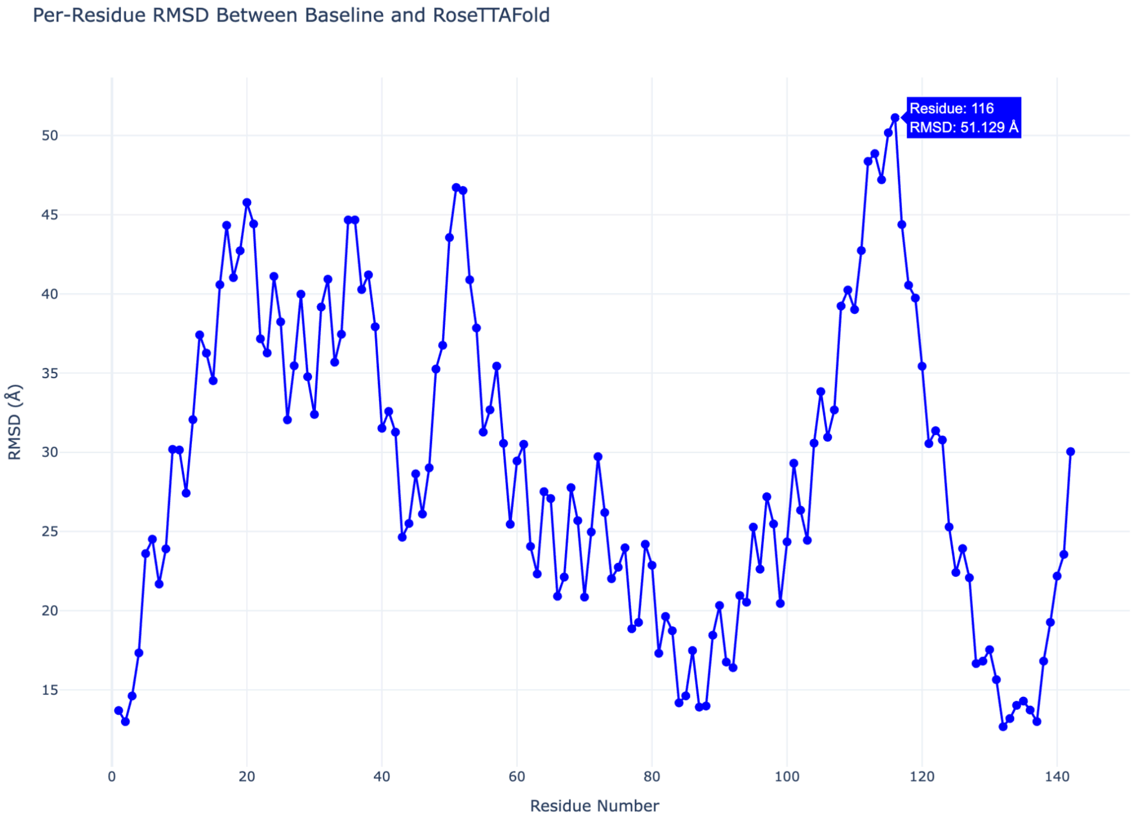


Figure 6: Line graph of RoseTTAFold RMSD. Average of 0.621 Å, high of 51.129 Å.

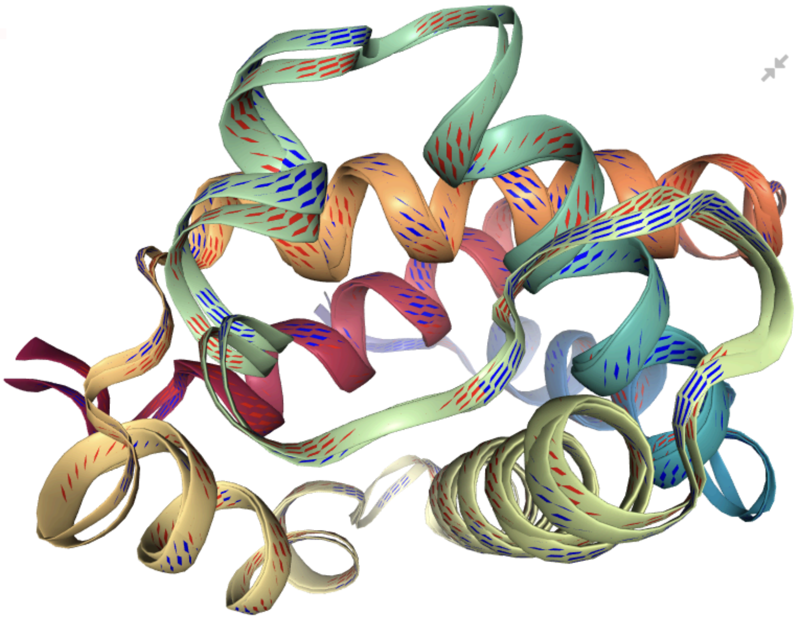


Figure 7: Visual overlap of baseline (blue) and RoseTTAFold model (red).

ESMFold

The usability of the ESMFold website was high and similar to that of the AlphaFold2 website. Since they are both created by large companies, Google and Meta, we infer that the companies have more workers to make their websites look clean and work well. Like the AlphaFold2 website, there did not seem to be a requirement for sequence length. No estimated time for computation was given and only one prediction model was produced.

The results of the ESMFold model [11] provided a moderate overlap with the baseline model giving an average RMSD of 0.324 Å and a high 44.134 Å.

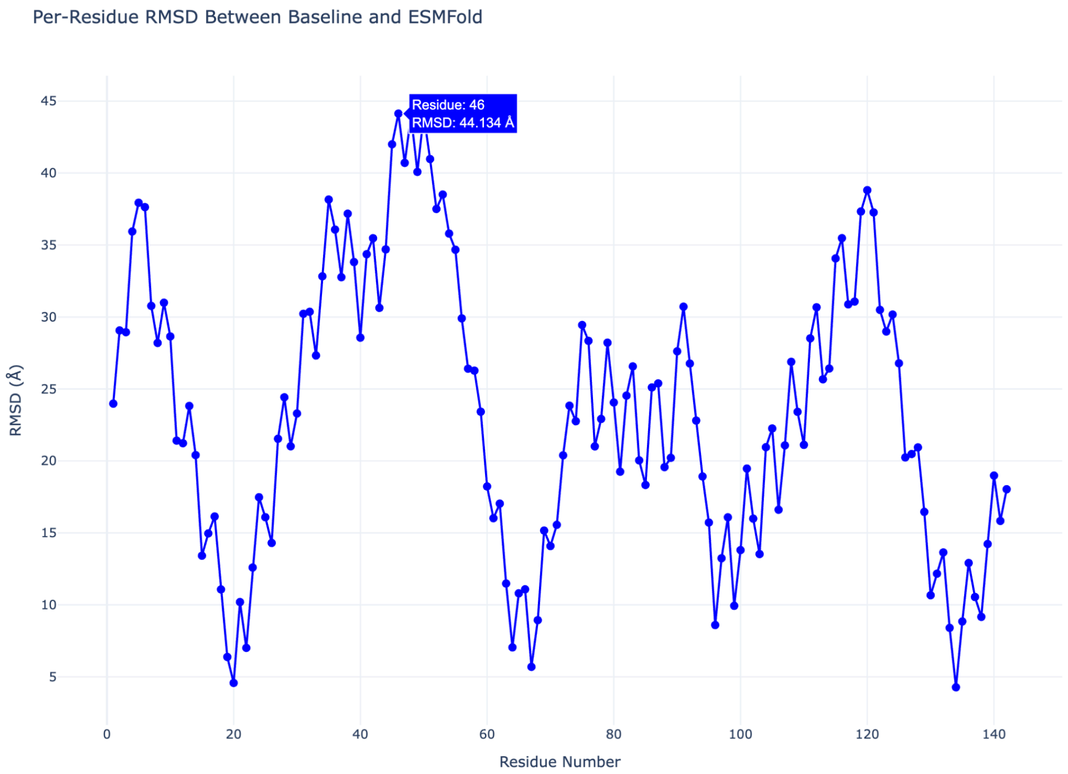


Figure 8: Line graph of ESMFold RMSD. Average of 0.324 Å, high of 44.134 Å.

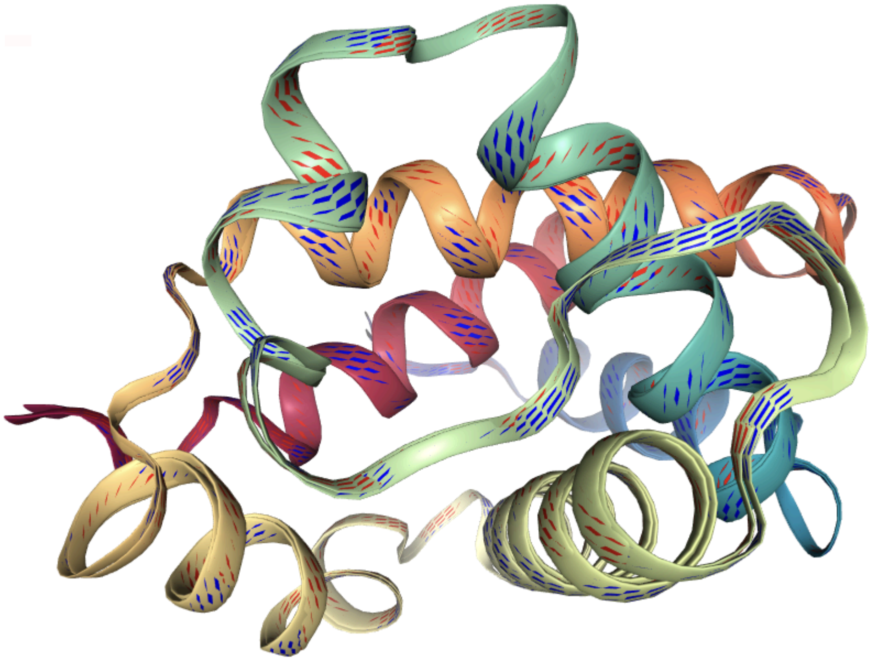


Figure 9: Visual overlap of baseline (blue) and ESMFold model (red).

I-TASSER

The usability of the I-TASSER website was very low. In order to run a prediction through their website, you need to create an account and have a scholarly email. Email verification is required. The sequence length must be in the range of 10-1500. These limitations offer lower options for researchers who want to use this algorithm for their analysis. No estimated time for computation was given because the results were sent through email and 5 separate prediction models were produced and one was ranked the best, which we used for comparison.

The results of the I-TASSER model [12] provided a moderate overlap with the baseline model giving an average RMSD of 0.419 Å and a high 114.183 Å.

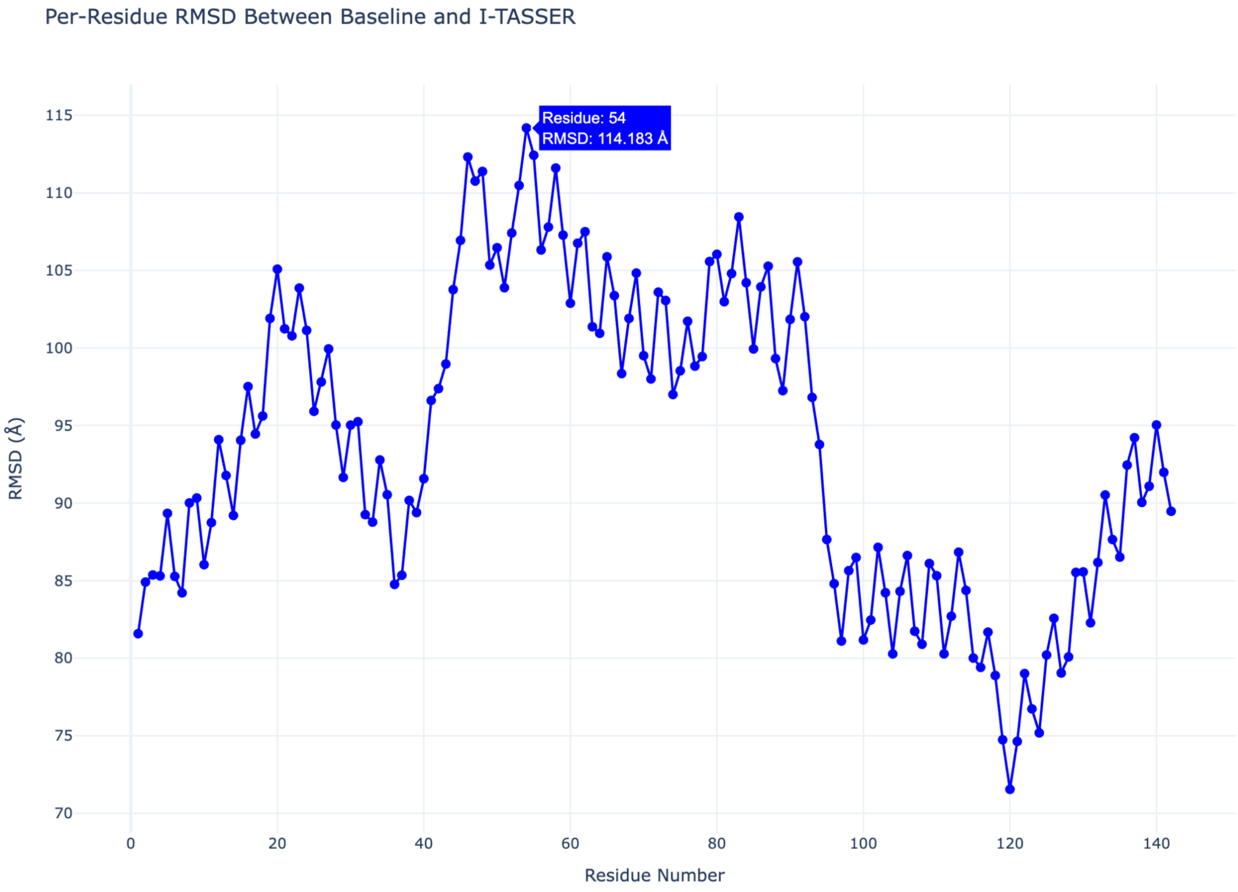


Figure 10: Line graph of I-TASSER RMSD. Average of 0.419 Å, high 114.183 Å.

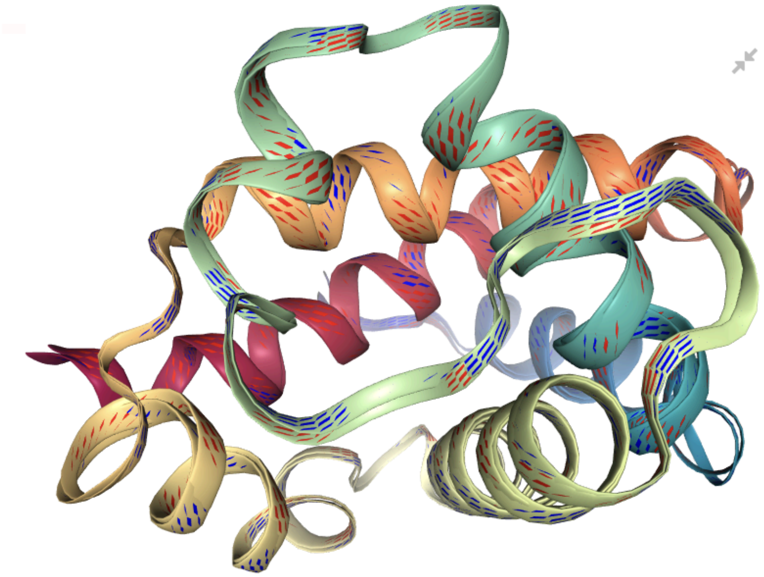


Figure 11: Visual overlap of baseline (blue) and I-TASSER model (red).

Results

After running our code and having firsthand experiences with the websites of the algorithm creators, we created a table to compare each algorithm's strengths and weaknesses. Data points were also collected through various articles [1] - [4] and the CASP results [13].

	AlphaFold2	**RoseTTAFold**	**ESMFold**	**I-TASSER**
Time Complexity (Worst Case)	$O(L^3)$	$O(L^2)$ to $O(L^3)$	$O(L^2)$	$O(L)$ to $O(L^2)$
Public Access	Open-source	Open-source	Open-source	Not fully open-source
Accuracy	High	High	Moderate	Moderate
Speed	Slow	Moderate	Fast	Fast
RMSD	0.049 Å	0.621 Å	0.324 Å	0.419 Å

Figure 12: Table of results comparing top protein structure prediction algorithms.

Conclusion

Limitations of our Data

While our actual techniques of comparison are sound, the specific baseline protein structure that we used provides a limitation for this research project which needs to be mentioned. The baseline model that we used was derived from the AlphaFold2 database itself. This database consists of a massive amount of models for proteins based on lab trials, not only AI predictions. By using the model created through biological means rather than computationally predictive ones, we are able to gain a strong baseline model for comparison. The caveat in using this model is that it is from the AlphaFold2 database itself. This proves complications and bias for then using that model to compare against the predicted model from the AlphaFold2 algorithm. In simpler words, we are then comparing the AlphaFold2 model database, to the raw predicted model using their predictive algorithm. While they should be separate in of themselves, the AlphaFold2 algorithm may have been trained with the database model and other similar models which give it an advantage in comparison to our other compared models. That being said, AlphaFold2 has proved the best database for predictive modeling and the most reputable algorithms for protein structure prediction. Although creating our own baseline through lab analysis would be the best for comparison, provided the limitations of time and availability this is the best comparison model we could use to our knowledge.

Another limitation is that we compared these top algorithms based on a single protein, HBA1. Although this is a great protein to give a reference into these algorithms performance and usability, increasing our data and comparing each algorithm from a large set of proteins would increase our confidence in the results.

Analysis of Results

Each algorithm gave its own advantages and disadvantages. While AlphaFold2 seemed to outperform every other algorithm in terms of results, the process was computationally expensive, making the option more unreasonable for the longevity of real world lab work. Although the convenience of having a protein structure database integrated into the website is a huge advantage for biologists. RoseTTAFold provided a great option for biologists in terms of balancing complexity and accuracy while also using an innovative solution through their three-track neural network system. But for long sequences, RoseTTAFold falters. In our analysis, RoseTTAFold produced the worst overlap of structure, even with a sequence of only 142 amino acids. ESMFold makes great results for their balanced algorithm. Our results show that they've coded a fast algorithm which gives solid results, especially for small algorithms. I-TASSER proved the most inconvenient in terms of accessibility and test results. While leveraging their algorithm to be fast, they've sacrificed results quality and produced the second worst structure overlap with our baseline structure.

While each algorithm is different, we cannot say whether one is better or worse than the other. To make the most of our results, biologists should analyze each algorithm and search for an algorithm that best overlaps with their specific needs. At the moment, no algorithm is a one-size-fits-all. To better advance the field, we should make the advantages and disadvantages of protein structure prediction algorithms transparent. There is not enough information on each individual website about their own optimal performance range. This is why external analysis of algorithms is needed. The reason for this covertness in the field is unknown and should be further investigated.

Future Works

Open-source Analysis of Algorithms

Increasing the transparency of algorithms advantages and disadvantages is extremely important for biologists to make educated decisions pertaining to their algorithm of choice. Making an open-source website where users can gain information on updated versions of protein structure prediction algorithms and be able to compare algorithm results of their own, will help inform the public about which algorithm is best for their specific case. Not only will this help the areas in research which will use these algorithms, but it will encourage algorithm creating companies to progress their algorithms further, explaining weak points and areas which could be improved upon and increasing competitiveness between top protein structure prediction algorithm companies or research labs. An increased competitiveness in this field would force top companies to improve algorithms more rapidly to stay at the top. This platform would also be a place for users to suggest new ideas, share their algorithms, and help find others who would want to collaborate together, improving the field more rapidly.

References

[1] Jumper, John, et al. “Highly Accurate Protein Structure Prediction with AlphaFold.” *Nature (London)*, vol. 596, no. 7873, 2021, pp. 583–89, https://doi.org/10.1038/s41586-021-03819-2.

[2] Liang, Tianjian, et al. “Differential Performance of RoseTTAFold in Antibody Modeling.” *Briefings in Bioinformatics*, vol. 23, no. 5, 2022, https://doi.org/10.1093/bib/bbac152.

[3] Song, Yidong, et al. “Accurately Predicting Enzyme Functions through Geometric Graph Learning on ESMFold-Predicted Structures.” *Nature Communications*, vol. 15, no. 1, 2024, pp. 8180–11, https://doi.org/10.1038/s41467-024-52533-w.

[4] Yang, Jianyi, and Yang Zhang. “I-TASSER Server: New Development for Protein Structure and Function Predictions.” *Nucleic Acids Research*, vol. 43, no. W1, 2015, pp. W174–81, https://doi.org/10.1093/nar/gkv342.

[5] “Nobel Prize in Chemistry 2024.” *NobelPrize.Org*, www.nobelprize.org/prizes/chemistry/2024/baker/facts/. Accessed 19 Mar. 2025.

[6] Author links open overlay panelG. Deléage, et al. “Protein Structure Prediction. Implications for the Biologist.” *Biochimie*, Elsevier, 5 Mar. 2000, www.sciencedirect.com/science/article/abs/pii/S0300908497835249.

[7] *Home - Prediction Center*, predictioncenter.org/. Accessed 19 Mar. 2025.

[8] “Uniprot Website Fallback Message.” *UniProt*, www.uniprot.org/uniprotkb/P69905/entry. Accessed 19 Mar. 2025.

[9] Database, AlphaFold Protein Structure. *Alphafold Protein Structure Database*, alphafold.com/entry/Q5R9M5. Accessed 19 Mar. 2025.

[10] *Log In*, robetta.bakerlab.org/results.php?id=660982. Accessed 19 Mar. 2025.

[11] “ESM Metagenomic Atlas: Meta Ai.” *ESM Metagenomic Atlas by Meta AI*, esmatlas.com/resources/fold/result?fasta_header=%3Eunnamed&sequence=MVLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHFDLSHGSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR. Accessed 19 Mar. 2025.

[12] “I-Tasser Results for Job ID S811368.” *I-TASSER Results*, zhanggroup.org/I-TASSER/output/S811368/. Accessed 19 Mar. 2025.

[13] Pereira J;Simpkin AJ;Hartmann MD;Rigden DJ;Keegan RM;Lupas AN; “High-Accuracy Protein Structure Prediction in CASP14.” *Proteins*, U.S. National Library of Medicine, pubmed.ncbi.nlm.nih.gov/34218458/. Accessed 19 Mar. 2025.