Protein Structure: Unraveled

By ERKS: Ethan Ye (eye5), Russell Chiu (rhchiu), Kelley Tu (kjtu), Sophia Li (sli347)

Github: *https://github.com/lisophia19/rawMSA-replica.git*

Introduction

Proteins are fundamental biological molecules that perform a vast array of essential functions, like catalyzing reactions, supporting cell structures, and transmitting cell signals. A protein's function is intimately tied to its shape—this shape is determined by its secondary structure, which folds into a more complex tertiary structure. Protein misfolding is associated with various diseases, such as Alzheimer's and Parkinson's, making accurate structural predictions a valuable tool in drug design and biomarker identification.

The problem we aim to address is the prediction of secondary structure for proteins using deep learning techniques on raw multiple sequence alignments (MSA). We re-implemented a paper that tackled predicting secondary structure, relative solvent accessibility, and contact maps using deep learning, which eliminates the need for feature engineering typically required in traditional methods. The model focuses on applying Natural Language Processing methodologies like Long Short-Term Memory (LSTM) networks. The paper's approach is to treat protein sequences as sentences and residues as words in an MSA window that are directly aligned with advancements in NLP. We can think of each MSA entry as a collection of one master sequence of amino acids and many other body sequences of amino acids: the master sequence corresponds to a specific sequence with classification data that we hope to predict, while the body sequences provide additional context and align with the master sequence in a predetermined format. Analyzing amino acids through the help of MSA has direct applications in structural biology, as understanding the secondary structure and solvent accessibility of proteins is key to elucidating their 3D structures and functions.

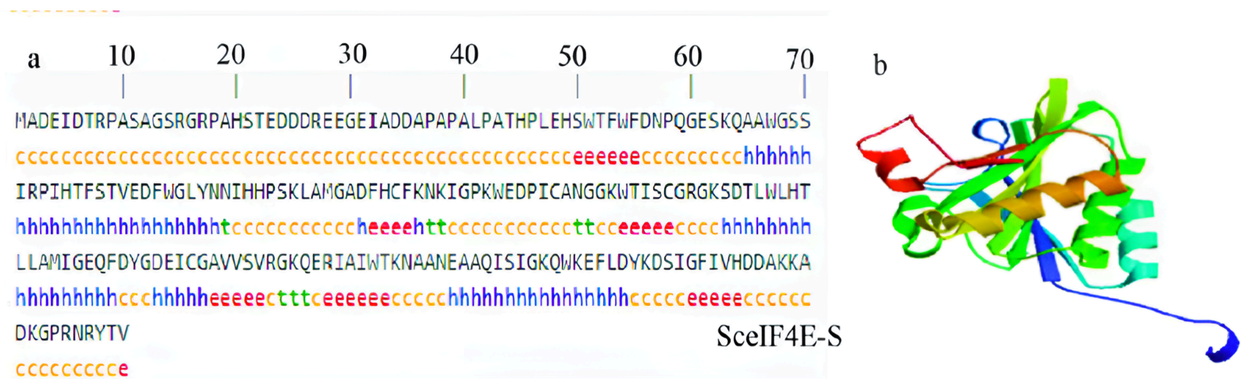
a
10
20
30
40
50
60
70
MADEIDTRPASAGSRGRPAHSTEDDDREEGEIADDAPAPALPATHPLEHSWTFWFDNPQGESKQAAWGSS
cccccccccccccccccccccccccccccccccccccccccccccccccceeeeeeccccccccchhhhhh
IRPIHTFSTVEDFWGLYNNIHHPSKLAMGADFHCFKNKIGPKWEDPICANGGKWTISCGRGKSDTLWLHT
hhhhhhhhhhhhhhhhhhtccccccccccchheeeehttccccccccccctttcceeeeecccchhhhhhhh
LLAMIGEQFDYGDEICGAVVSVRGKQERIAIWTKNAANEAAQISIGKQWKEFLDYKDSIGFIVHDDAKKA
hhhhhhhhhccchhhhheeeeeectttceeeeeeccccchhhhhhhhhhhhhhhhcccccceeeeeccccccc
DKGPRNRYTV
ccccccccce
SceIF4E-S
b

Figure 1: example amino acid sequence (black) and corresponding secondary structure assignments (color)

Methodology

Data Collection and Preprocessing

The dataset used for training consists of multi-sequence alignment (MSA) data representing protein sequences. We gathered MSA data from various protein families with the help of the *InterPro* database, a database operated under EMBL's European Bioinformatics Institute. The family domains of the proteins were selected randomly from this database. The data for all of the family domains were split such that 23 family domains were used for the training dataset, 5 for the validation dataset, and 4 for the testing dataset. With this, the collected data for each family domain was consistently mapped from "Stockholm" formatting into unique integers from 1 to 25, representing the 25 possible amino acids. Similarly, each master sequence's secondary structure was mapped into another set of integers from 1 to 4, representing 4 possible secondary structure classifications. Given that secondary structure is determined primarily by local neighborhoods, we employed a custom sliding window approach of size 31 residues. Padding was then applied to maintain consistent sequence lengths. Each window was flattened to form inputs of shape (batch_size, 31 × 15), facilitating batch processing. The model was trained to predict the secondary structure of the master sequence's central residue using information from the full MSA.

Model Architecture

The architecture of our model is inspired by the LSTM-based design used in the original paper. Our secondary structure model is a sequence labeling task, where each residue in a protein sequence is classified as one of four secondary structure types: helix, strand, coil, and "other." After data collection and preprocessing, the input tensor into our model was of shape (311, 200, 15), as there were 311 batches, 200 amino acids per sequence, and 15 sequences to predict the secondary structure of the master sequence. Thus, each batch is of the size (1, 200, 15).

The model consists of 8 layers: 1 embedding layer, 1 convolution layer with max pooling, 2 Bi-directional LSTM layers, 3 linear layers, and 1 log softmax layer. The Bi-directional LSTM layers help capture long-range dependencies between residues, while the log softmax layer outputs the final classification for each residue. An Adam optimizer with a learning rate of 0.005 was used, with a dropout rate of 0.25 to mitigate overfitting layers during training.

The model specifically predicts the secondary structure of the master sequence's central residue within each sliding window, ensuring that learning is anchored to the target sequence rather than aggregated across all homologs. The loss function employed is negative log-likelihood loss (NLLLoss), chosen for its compatibility with the log softmax output.

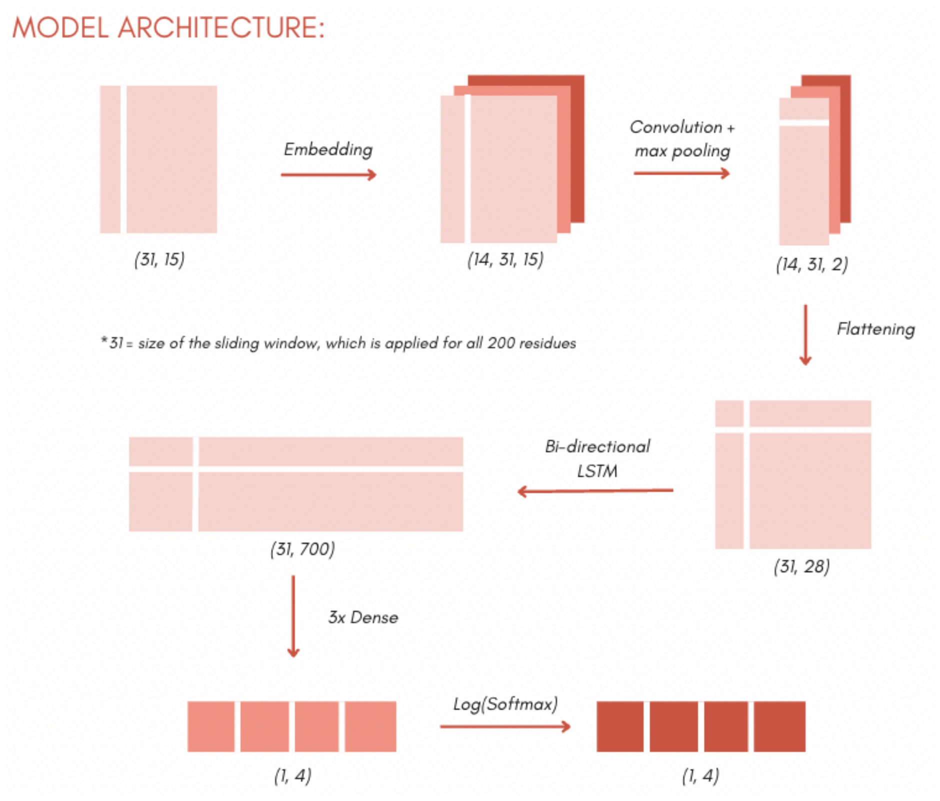

Figure 2: Depiction of the model architecture

Training and Testing

Given the computational demands of training with large MSA datasets, we used OSCAR equipped with GPUs to accelerate training. Due to limited GPU access, our smaller collected dataset was trained for 5 epochs to evaluate its performance. Success was measured by the model's accuracy in classifying residues into the correct secondary structure types.

Results and Discussion

After modifying hyperparameters, our model, under the testing process, achieved an accuracy of 0.895 with a running loss of 59.42, which was higher than the paper's model accuracy of 0.807 across 100 sequences. However, as seen in the graph below, both the training and validation accuracy plateau, indicating that the model is underfitting and could benefit from additional complexity, such as adding more data. Using only three of the eight total secondary structure classes (coil, helix, and strand) and simplifying the other five into the "other" category may have led to an inflated accuracy, as the model could have developed a bias toward classifying residues as "other," which is a more general category.

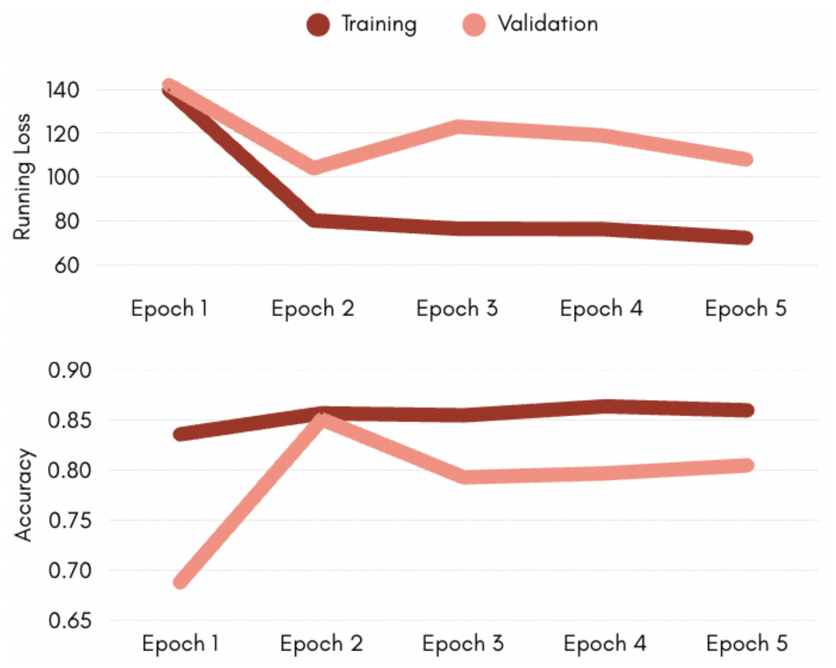

Figure 3: Graphical representation of the model's accuracy and running loss

Challenges

We encountered many challenges throughout the process. We discovered that there was no available code to guide our preprocessing or model development, aside from the h5 files containing the model architecture. Additionally, the data link had expired. Despite these setbacks, we were still able to design and implement a functional model, achieving relatively high accuracy that beat the baseline from the paper. We recognize that training on more data would lead to a better model, but we had trouble with gathering enough data to train our model and parsing each Stockholm file due to the lack of a consistent formatting structure across data from different family domains. Moreover, when trying to implement our model, we had to choose a model based on our limited computation power and implement everything from scratch based on context clues from the paper, which is why preprocessing the data took longer than expected. Finally, it was difficult to navigate the PyTorch framework when establishing our model. In particular, PyTorch differs from TensorFlow in that all layers must be given an input and output size: this caused grief in the form of shape errors during the implementation step of the project. Finally, it is important to note that we decided to split the data to be mutually exclusive between master and body sequences. Such a decision could have introduced biases due to incomplete utilization of all the data during randomized batching.

Reflection

The project turned out to be better than expected because, despite all the challenges, we were still able to develop a model with relatively high accuracy. We were able to reach our target goal to develop a model that can make decently accurate predictions (greater than 60%) about the secondary structure of test proteins. We had greater accuracy than the average model accuracy from the paper, which was shocking but made sense because we trained and tested on a lot less data and analyzed a shorter sequence length (only 200 amino acids compared to around 1000 in the paper). The protein families we collected were on the smaller side and had less secondary structure information than the data the paper had, but we did not have access to their data. If we had more time, we could collect RSA data to train our model on, implement the CMAP model, and evaluate its accuracy in comparison to our SS-RSA predictions. With this extra visual information from the contact map, we would be better equipped to tackle our stretch goal of extrapolating our predictions to real-world applications, such as analyzing the impact of genetic sequence variations on protein structure, and looking at which sequences are highly conserved (common) across species to analyze genetic sequences during evolution.

Working through figuring out how to gather, preprocess, and utilize MSA data gave us an appreciation for the complexity involved in managing biological data, as well as the challenges of ensuring data quality, consistency, and accuracy to build a robust deep learning model. We learned that the model architecture and implementation are only a fraction of the overall process. Equally important are the considerations of data handling, fine-tuning of hyperparameters, and ensuring sufficient compute power for training the model. Equipped with this knowledge, we are now better prepared to approach future projects with a deeper understanding of data management, model optimization, and managing system resources.

Citations

Original paper: "End-to-End Deep Learning Using Raw Multiple Sequence Alignments" by Claudio Mirabello and Björn Wallner. Edited by Yang Zhang. Published: 15 Aug. 2019. journals.plos.org/plosone/article/authors?id=10.1371/journal.pone.0220182.

Related Article: "Protein Secondary Structure Prediction Based on Data Partition and Semi-Random Subspace Method" by Yuming Ma. Nature News, Nature Publishing Group, 29 June 2018, www.nature.com/articles/s41598-018-28084-8.