Training a Protein Secondary Structure Prediction Model Core Goal: Train an ML model to predict protein secondary structures (H = Helix, E = Beta Sheet, C = Coil) from amino acid sequences using Kaggle GPUs.

Phase 1: Set Up the Kaggle Environment Tasks:

Create a new Kaggle Notebook →
Enable GPU (T4 or A100). Install necessary dependencies: bash Copy Edit pip install torch tensorflow biopython Prepare structured sections in the notebook for: Data Loading Preprocessing Model Training Evaluation

Phase 2: Get & Preprocess the Dataset Tasks:

Find and load a dataset (e.g., DSSP, UniProt, Protein Data Bank). Preprocess the data: Convert amino acid sequences into numerical representations: One-hot encoding (simple, fast) Pretrained embeddings (ProtBERT, TAPE, ESM2) (better, needs GPUs) Convert secondary structure labels (H, E, C) → numerical categories (0, 1, 2). Split into train (80%), validation (10%), test (10%).

Phase 3: Build & Train the Model Tasks:

Define the model architecture: CNN Layer → Captures local sequence patterns. BiLSTM Layer → Captures long-range sequence dependencies. Fully Connected Layer → Maps features to output classes (H, E, C). Softmax Activation → Assigns probability to each class. Compile with loss function & optimizer: Categorical Cross-Entropy Loss (for multi-class classification). Adam Optimizer (fast convergence). Train the model for 20-30 epochs. Monitor validation accuracy and loss.

Phase 4: Evaluate & Improve Tasks:

Test the model on the unseen test set. Compute evaluation metrics: Accuracy Precision, Recall, F1-score Q3 Score (for secondary structure prediction). Visualize results: Confusion matrix (check class imbalances). Sample predictions vs. actual labels. Iterate & Improve: Adjust learning rate, batch size. Try different sequence embeddings.

Optional: Try a More Advanced Model (If Time Allows)

Tasks:

Replace BiLSTM with Transformers (ESM2, ProtBERT). Train a larger model using Kaggle's A100 GPUs. Phase 5 (Future, Not Tomorrow): Scaling & Applications Once the model is working:

Expand into real-world biotech problems (mutation analysis, drug discovery). Test generative approaches (designing synthetic proteins). Experiment with distributed ML techniques (for larger-scale training).

Introduction Protein secondary structure can be calculated based on its atoms' 3D coordinates once the protein's 3D structure is solved using X-ray crystallography or NMR. Commonly, DSSP is the tool used for calculating the secondary structure and assigns one of the following secondary structure types (https://swift.cmbi.umcn.nl/gv/dssp/index.html) to every amino acid in a protein:

C: Loops and irregular elements (corresponding to the blank characters output by DSSP) E: β-strand H: α-helix B: β-bridge G: 3-helix I: π-helix T: Turn S: Bend However, X-ray or NMR is expensive. Ideally, we would like to predict the secondary structure of a protein based on its primary sequence directly, which has had a long history. A review on this topic is published recently, Sixty-five years of the long march in protein secondary structure prediction: the final stretch?.

For the purpose of secondary structure prediction, it is common to simplify the aforementioned eight states (Q8) into three (Q3) by merging (E, B) into E, (H, G, I) into E, and (C, S, T) into C. The current accuracy for three-state (Q3) secondary structure prediction is about ~85% while that for eight-state (Q8) prediction is <70%. The exact number depends on the particular test dataset used.

Dataset The main dataset lists peptide sequences and their corresponding secondary structures. It is a transformation of https://cdn.rcsb.org/etl/kabschSander/ss.txt.gz downloaded at 2018-06-06 from RSCB PDB into a tabular structure. If you download the file at a later time, the number of sequences in it will probably increase.

Description of columns:

pdb_id: the id used to locate its entry on https://www.rcsb.org/ chain_code: when a protein consists of multiple peptides (chains), the chain code is needed to locate a particular one. seq: the sequence of the peptide sst8: the eight-state (Q8) secondary structure sst3: the three-state (Q3) secondary structure len: the length of the peptide has_nonstd_aa: whether the peptide contains nonstandard amino acids (B, O, U, X, or Z). Key steps in the transformation:

Both Q3 and Q8 secondary structure sequences are listed. All nonstandard amino acids, which includes B, O, U, X, and Z, (see here for their meanings) are masked with "*" character. An additional column (has_nonstd_aa) is added to indicate whether the protein sequence contains nonstandard amino acids. A subset of the sequences with low sequence identity and high resolution, ready for training, is also provided

Final Summary:
✔ Set up Kaggle notebook & install dependencies
✔ Get & preprocess a dataset (DSSP, UniProt, etc.)
✔ Build & train a CNN + BiLSTM model for secondary structure prediction
✔ Evaluate performance using accuracy, Q3 score, and confusion matrix
✔ If time allows, test transformer-based models