Project Report

Lawrence Owusu, Jordan Sturtz, and Swetha Chittam

***Abstract*—Abstract - clearly state problem, method, and results.**

I. Introduction

PROTEIN structure sequencing remains one of the challenges in the field of computational biology. Determination of accurate protein structure is important in developing deep understanding of the functions of proteins and their applications of drug and inhibitor discovery and design.

Protein sequencing begins in the lab using techniques such as mass spectrometry or Edman degradation to identify partial sequences of proteins [1], [2]. Various techniques exist for using these partial sequences to construct the full protein sequence [3]–[6].

In our paper, we develop an approach using deep learning techniques to predict missing gaps in peptide sequences. Our approach begins with a partial sequence with gaps of known size. To retrieve the relevant training data, we query the National Center For Biotechnology Information (NCBI) Protein Blast server to retrieve the highest matching homologous sequences to our partial sequence [7]. We train two models: one to predict peptides in the forward direction and one to predict in the reverse direction. We then use both predictions to predict the missing peptides. Our deep learning models use LSTM layers to learn the forward and reverse dependencies.

A. Related Work

The history of using Mass Spectrometry to sequence proteins dates to several years ago. In this approach, multiple proteases are employed to cleave the same protein sample separately. Because the cleavage of protein by the protease generates overlapping peptides, merging the spectral pairs of the overlapping peptides consecutively assembles into long contigs from which de novo sequences are obtained [8].

Some researchers on protein sequencing problems focus on identifying proteins or generating full protein sequences by comparing partial sequences to known protein databases to infer the missing gaps in mass spectral data [4], [5]. Others focus on developing techniques for de novo protein sequencing, which refers to the process of sequencing proteins without the use of a protein database or with minimal use of genomic data [3], [9].

Over the last 10 years, de novo protein sequencing has been researched extensively in computational proteomics and have been used successfully to deduce peptide sequence of un-sequenced organisms, antibodies and post-translastionally modified peptides [10]–[12]. For example, Mai et al. reported that their assembling algorithm, Multiple Contigs and Scaffolding successfully assembles the de novo identified peptides in contig-scaffold fashion, resulting in 100% coverage and 98.69-100% accuracy on three proteins and replications. The Multiple Contigs and Scaffolding algorithm has provided robust and accurate software for full-length protein sequencing after de novo identification of peptides [13]

Similarly, Yang et al. reported 100% accuracy for full-length de novo sequencing for light chains of Herceptin and bovine serum albumin (BSA) when their proposed method was applied to de novo sequencing of bovine serum albumin (BSA) and monoclonal antibody Herceptin. However, the accuracy marginally dropped to 99.7% for the heavy chains of Herceptin [6].

Protein sequencing is only one type of sequencing problem in biology. Related types of sequencing problems include single-cell protein sequencing or whole genomic sequencing [14], [15].

II. Method

A. Data Collection

For our initial project, we used as a target sequence the light chain of alemtuzumab referenced in Liu, et al. [8]. We manually removed the gaps produced by Liu et al.'s tandem mass spectrometry approach, then entered that sequence into NCBI's Protein Blast Server to retrieve the closest matching homologous sequences [7]. We used the top ten homologous sequences as training data. The closest matching homologous sequences in our training data range from 89.32% to 82.52% similarity.

B. Data Preprocessing

We generated from our homologous sequences all kmers. A Kmer is a any substring of length K. For our models, we chose a kmer-length of 5. Each kmer represents a single input. The output of each associated kmer is the next character in the sequence. Thus, for instance, if one of our homologous sequences contains the substring "DIQM", then a single training instance would be the input-output tuple (DIQ, M).

From our ten homologous sequences, we generated 2106 number of input-output pairs with an input length of 5. The kmer length will represent the timesteps in our LSTM layer. To be fed into a neural network, the data must be encoded numerically, so we assigned integer labels to each character. The output of each training instance is a one-hot encoded vector to represent the target classes. Since we are training two models, one for forward prediction and one for reverse prediction, we generate the same training data in both the reverse and forward directions.

The authors are graduate students at NCA&T

The samples of forward and reverse input-output pairs are divided into train and validation sets with the ratio of 90% and 10% respectively. The validation set is used to perform hyperparameter tuning.

C. Data Normalization

The input pairs are normalized by dividing with number of classes to obtain a range between [0, 1]. The number of classes are determined by extracting all the unique characters from the input sequences. This data normalization technique is applied to forward, reverse and test data input-output pairs. Normalization helps speed the training process.

D. Model Overview

For our sequence prediction task, we built four deep learning models combining convolutional neural networks (CNN) with long short term memory (LSTM). The CNN layer helps with automatic feature extraction, in particular if there are patterns within each kmer that can be extracted for better predictions. The LSTM layer learns the dependencies in the sequence data to predict the correct output. For our four hybrid models, we also added two dense hidden layers with ReLu activation and a final output layer with softmax activation.

Our four hybrid models are LSTM, CNN-LSTM, Bi-LSTM, and CNN-Bi-LSTM. The purpose of using four models is to compare their results.

The workflow for the proposed work is shown in Figure 1.

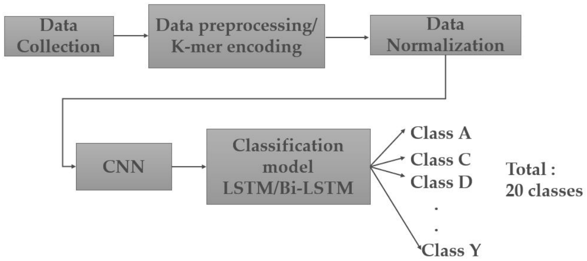

Fig. 1. Model diagram for multi-classification of sequence data

1) CNN: CNN is a powerful deep learning technique for feature extraction. CNN is often used on 2-dimensional and 3-dimensional datasets, e.g. image or video datasets. We use a 1-dimensional convolutional layer that convolves the kmers to extract any meaningful features. The convolutional layer has 128 filters with a kernel size of 3, so minimally the kmer-length cannot be smaller than 3. The feature maps from this layer forms as input to the LSTM or Bi-LSTM layer.

2) LSTM: Characteristic of the LSTM architecture is a set of chained together cells called ”memory blocks”. The number of memory blocks in the chain equals the length of the timesteps in our target dataset (Fig 2). Each memory cell has three gating units (forget, input and output gates) which conditionally regulate how information flows into and out of the memory block (Fig 3). Intuitively, the ”forget” gate can be viewed as a step where irrelevant information from the hidden state is first ”forgotten” before being passed to the input gate, which constructs the new cell state. Before that cell state can output its value to the next memory cell, it is filtered again through an output gate that decides what values to keep internal to the memory cell (i.e. the hidden state) and what to output to the next memory cell or final output layer. The LSTM layer is helpful, therefore, for learning the important sequence dependencies for performing sequence predictions.

3) Bidirectional LSTM: Bidirectional LSTM is a modification of the LSTM architecture that permits the model to learn both the forward and reverse dependencies. The way it acheives this is to have two LSTM layers, one consuming the timesteps in the forward direction and the other consuming the timesteps in the reverse direction. The two layers then merge their results to produce the final output.

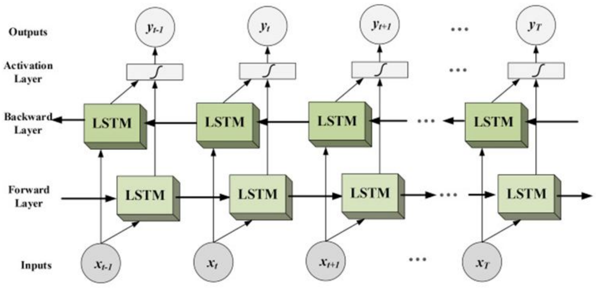

Fig. 2. Bidirectional LSTM Architecture

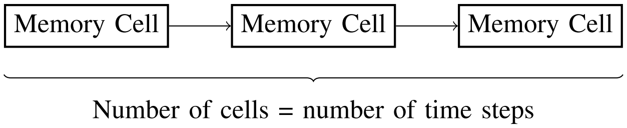

Fig. 3. Illustration of recurrently chained memory cells

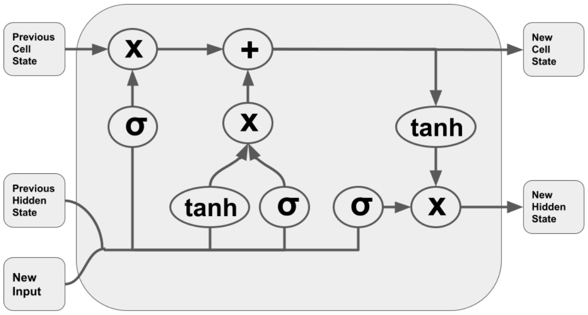

Fig. 4. LSTM architecture inside a memory cell

III. Model Architectures

A. CNN

B. LSTM

C. LSTM-CNN

D. BiLSTM-CNN

IV. Experimental Results

Experimental results: Implementation process, tuning hyperparameters, show predictions in amino acid gaps (deploy the model and show its accuracy on a particular instance)

A. Tuning hyperparameters

B. Accuracy/loss/confusion matrix/whatever

C. Predictions on De Novo Sequence

V. Conclusion

Conclusion: Summarize project

References

[1] M. Mann, “The rise of mass spectrometry and the fall of edman degradation,” *Clinical Chemistry*, vol. 62, no. 1, p. 293, 2016.

[2] P. Edman *et al.*, “A method for the determination of the amino acid sequence in peptides.” *Arch. Biochem.*, vol. 22, pp. 475–476, 1949.

[3] K. G. Standing, “Peptide and protein de novo sequencing by mass spectrometry,” *Current opinion in structural biology*, vol. 13, no. 5, pp. 595–601, 2003.

[4] D. N. Perkins, D. J. Pappin, D. M. Creasy, and J. S. Cottrell, “Probability-based protein identification by searching sequence databases using mass spectrometry data,” *ELECTROPHORESIS: An International Journal*, vol. 20, no. 18, pp. 3551–3567, 1999.

[5] J. K. Eng, A. L. McCormack, and J. R. Yates, “An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database,” *Journal of the american society for mass spectrometry*, vol. 5, no. 11, pp. 976–989, 1994.

[6] C. Yang, Y.-C. Shan, W.-J. Zhang, Z.-P. Dai, L.-H. Zhang, and Y.-K. Zhang, “Full-length protein sequencing based on continuous digestion using non-specific proteases,” *ACTA CHIMICA SINICA*, vol. 79, no. 5, pp. 664–670, 2021.

[7] National Center for Biotechnology Information, “Blast,” May. 8, 2022 [Online]. [Online]. Available: https://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE=Proteins

[8] X. Liu, L. J. Dekker, S. Wu, M. M. Vanduijn, T. M. Luider, N. Tolic, Q. Kou, M. Dvorkin, S. Alexandrova, K. Vyatkina *et al.*, “De novo protein sequencing by combining top-down and bottom-up tandem mass spectra,” *Journal of proteome research*, vol. 13, no. 7, pp. 3241–3248, 2014.

[9] N. Bandeira, V. Pham, P. Pevzner, D. Arnott, and J. R. Lill, “Automated de novo protein sequencing of monoclonal antibodies,” *Nature biotechnology*, vol. 26, no. 12, pp. 1336–1338, 2008.

[10] B. Ma and R. Johnson, “De novo sequencing and homology searching,” *Molecular & cellular proteomics*, vol. 11, no. 2, 2012.

[11] J. A. Veltman and H. G. Brunner, “De novo mutations in human genetic disease,” *Nature Reviews Genetics*, vol. 13, no. 8, pp. 565–575, 2012.

[12] N. Bandeira, “Spectral networks: a new approach to de novo discovery of protein sequences and posttranslational modifications,” *BioTechniques*, vol. 42, no. 6, pp. 687–695, 2007.

[13] Z.-B. Mai, Z.-H. Zhou, Q.-Y. He, and G. Zhang, “Highly robust de novo full-length protein sequencing,” *Analytical Chemistry*, vol. 94, no. 8, pp. 3467–3475, 2022.

[14] Y. Wang and N. E. Navin, “Advances and applications of single-cell sequencing technologies,” *Molecular cell*, vol. 58, no. 4, pp. 598–609, 2015.

[15] C. S. Pareek, R. Smoczynski, and A. Tretyn, “Sequencing technologies and genome sequencing,” *Journal of applied genetics*, vol. 52, no. 4, pp. 413–435, 2011.