Deep Learning Approaches for the Protein Scaffold Filling Problem

***Abstract*—We are on the verge of a post-genomics era in which whole protein sequencing will be quickly carried out. Protein sequencing plays an important role in identifying protein functions, analyzing protein-protein interactions, and characterizing post-translational modifications, etc. The protein sequencing problem is to determine the complete sequence of amino acids in proteins.**

***De novo* protein sequencing using top-down and bottom-up tandem mass spectrometry suffers from the problem of producing only partial sequences of target proteins, namely scaffold. In this paper, we explore the possibility of using deep learning techniques to perform the task of predicting amino acids in partially sequenced proteins by two phases. First, our methods involve querying the NCBI Protein Blast server to find closest matching homologous sequences to a scaffold as a training dataset. Second, we train several deep learning models based on a convolutional neural network and long short term memory to predict missing amino acids in the scaffold in the forward and reverse directions. We comprehensively evaluate our proposed methods on an alemtuzumab dataset and our results show that the proposed methods achieve high sequence coverage and high sequence accuracy with 100% on the the light chain of alemtuzumab scaffold data.**

***Index Terms*—protein sequencing, scaffold filling, deep learning, alemtuzumab dataset**

I. Introduction

Protein sequencing plays an important role in identifying protein functions, quantifying proteins in complex biological samples, and analyzing protein-protein interactions. Protein sequencing remains one of the challenges in the field of computational biology. Determination of accurate protein sequencing and structure are important in developing deep understanding of the functions of proteins and their applications of drug and inhibitor discovery and design. The protein sequencing problem was initiated as early as in 1950 by Edman, but due to the high cost of traditional Edman degradation experiments, since 1987 tandem mass spectrometry has become a popular method for the protein sequencing. There are two main methods for *de novo* protein sequencing: Edman degradation and mass spectrometry. In the past decades, researchers identify proteins by searching tandem mass spectra through protein databases. However, some important proteins are not included in such databases. For example, the antibody MabCampath and the venom-based drug Captopril are commonly not found in such databases, despite their utility in treating breast cancer and cardiovascular disease respectively. Therefore, mass spectrometry-based *de novo* protein sequencing becomes a popular technique to sequence such unknown proteins.

This technique is usually carried out in the form of whole-protein analysis called top-down proteomics or analysis of enzymatically or chemically produced peptides called bottom-up proteomics. Despite the recent progress in protein sequencing and assembly, many of the currently available assembled proteins come in a draft form. There are still many gaps in the assembled protein sequences even if we combine top-down and bottom-up methods. These gaps may correlate to important protein functions. In other words, at the end of the sequencing steps, we are more likely to see contigs separated with gaps (or scaffolds). Hence, a natural computational biology problem is to fill the missing amino acids in scaffolds to obtain complete protein sequences.

In 2017, Qingge et al. [1] developed several practical polynomial time algorithms based on dynamic programming and local search to fill the remaining gaps in protein scaffolds to construct complete protein sequences with respect to its homologous protein sequences. However, the methods in this paper have several disadvantages. The running time of exact solutions for the protein scaffold filling problems are as high as $O(n^{26})$, where n is the number of missing amino acids in the scaffold. On the other hand, it needs a high quality homologous protein reference. The idea behind filling the gaps in the protein scaffold is based on a simple greedy approach.

In this paper, we develop an approach using deep learning techniques to predict missing amino acids in protein scaffolds to overcome the disadvantages in [1] and improve the prediction results in the gaps of a scaffold. Our approach begins with a partial sequence with gaps of known size obtained from analyzing top-down and bottom-up tandem mass spectrometry. To obtain the training data in our deep learning models, we query the National Center For Biotechnology Information (NCBI) [2] Protein Blast server to retrieve the highest matching homologous sequences to our known scaffold. We then train and deploy two sets of models: one to learn dependencies in the forward direction and another to learn dependencies in the reverse direction. At last, we combine the two models' predictions to fill the gaps in the *de novo* protein sequencing.

In each direction, we train and build four competing hybrid deep learning models combining long short-term memory (LSTM) and convolutional nerual network (CNN) layers. We evaluate our models against a known target sequence (the light chain of alemtuzumab) of the given scaffold to check the performance of filling gaps. Our results show high accuracies (92-94%) for all four of our model architectures, with our LSTM-CNN model outperforming the others at 94.76% on the validation dataset. The comparison results show that our deep learning based approaches outperforms the proposed combinatorial algorithms designed in [1] and improves the results in [3]. We believe our proposed deep learning methods will greatly improve the *de novo* protein sequencing combined with top-down and bottom-up tandem mass spectrometry analysis.

We organize our paper as follows. In section II, we will introduce related sequencing work. In section III, we will discuss data collection, data preprocessing and related deep learning models. In section IV, we will propose our deep learning models, and in section V, we show the experimental results. We will conclude and discuss open problems in the last section.

II. Related Work

The bottom-up and top-down tandem mass spectrometry based analysis continues to be a dominant method to perform protein sequencing [4]. The bottom-up approach consists of three steps. It cleaves the same protein sample using multiple proteases to break the protein into many peptides to generate many overlapping tandem mass spectra of the peptides, and finally stack and align these spectra or their corresponding peptide sequences (obtained from *de novo* peptide sequencing) together to construct the original protein sequence. The top-down tandem mass spectrometry analysis is an emerging technique to cover intact proteins. However, the top-down approach rarely produce full ion fragmentation coverage for a whole protein.

Over the past decades, *de novo* peptide sequencing has been researched extensively in computational proteomics and has been used successfully to deduce peptide sequence of unsequenced organisms, antibodies and post-translationally modified peptides [5]–[7]. For example, Mai et al. reported that their assembling algorithm successfully assembles the *de novo* peptides in contig-scaffold fashion, resulting in 100% coverage and 98.69-100% accuracy on three proteins and replications [8]. Similarly, Yang et al. reported 100% accuracy for full-length *de novo* sequencing for light chains of Herceptin and bovine serum albumin (BSA) and they reported 99.7% accuracy for the heavy chain of Herceptin [9].

Protein sequencing is only one type of sequencing problem in biology. Related types of sequencing problems include single-cell protein sequencing or whole genome sequencing [10], [11]. Most recently, Tran et al. [12] proposed DeepNovo, a deep neural network model that combines convolutional neural network, long short term memory, and dynamic programming to address the *de novo* peptide sequencing problem. This method shows an improvement of up to 64% higher accuracy at the full-peptide level when compared with state-of-the-art *de novo* peptide sequencing tools, such as PEAKS [13], Novor [14] and PepNovo [15]. Tran et al. [16] also presented a *de novo* peptide-sequencing method named DeepNovo-DIA for data-independent acquisition (DIA) mass spectrometry data. Qiao et al. [17] developed PointNovo, a neural network-based *de novo* peptide sequencing model that can handle any resolution levels of mass spectrometry data without substantially increasing the computational complexity, which remains a challenge for *de novo* peptide sequencing tools. Zohora et al. [18] developed a deep learning model named DeepIso for peptide feature detection from liquid chromatography with tandem mass spectrometry (LC-MS) map. Guan et al. [19] proposed a deep learning model to predict three key LC-MS/MS properties from peptide sequences to enhance peptide identification and quantification in data-dependent acquisition and data-independent acquisition (DIA) experiments. Also, single-cell RNA sequencing analysis is one of the most popular techniques which has been widely used in immunology, developmental biology, oncology and cancer genomics to measure transcripts in a mixture of cells. An increasing number of computational algorithms and tools have been developed for single-cell analysis, such as Seurat [20], MAGIC [21], SAVER [22], scImpute [23] for imputation, Seurat CCA [24] and ZINB-WaVE [25] for batch effect removal.

III. Methods

Most of the neural network based models were developed to solve for the *de novo* peptide sequencing, not for the *de novo* protein sequencing. The distinguishing features of our paper is that we try to solve the protein sequencing problem using deep learning techniques based on the constructed scaffold retrieved from the analysis of top-down and bottom-up tandem mass spectrometry. In this paper, we focus on filling missing amino acids in the scaffold to complete the protein sequencing. The protein scaffold filling problem is defined as follows.

The Protein Scaffold Filling Problem: Given a complete target protein sequence S and the scaffold T, fill the missing amino acids in the scaffold T such that $Score(S,T)$ is maximized, where function $Score$ is the total number of one-to-one matches of amino acids between S and T.

A. Data Collection

We use the same techniques proposed in Liu, et al. [3] to generate the protein scaffold of the light chain of alemtuzumab (MabCampath) by the process of converting raw spectra to a prefix residue mass (PRM) spctra, spectral selection and merging, improving the top-down spectrum using bottom-up spectra, spectra mapping, gap filling by extension and gap filling by mass matching. In the end, we generate the MabCampath scaffold consisting of 5 contigs and 6 gaps of missing amino acids. The scaffold information can be seen in Figure 12, in which the red colored dash line represents gaps in the scaffold and green colored amino acids are predicted amino acids in the gaps. More technical details about generating the MabCampath protein scaffold can be found in Liu, et al. [3].

We manually remove the gaps produced by Liu et al.'s [3] combined top-down and bottom-up tandem mass spectrometry approach, then feed that sequence into NCBI's Protein Blast Server to retrieve top ten closest matching homologous sequences [2] as our training data. The similarity between the scaffold and homologous sequences ranges from 82.52% to 89.32%.

B. Data Preprocessing

To obtain input and ouput of training data, we generate all kmers from our homologous sequences. A kmer is a any substring of length K. For our models, we test different lengths of kmers and choose a kmer-length of 5. Each kmer represents a single input, and the output of each associated kmer is the next character (amino acid) in the sequence. For instance, if one of our homologous sequences contains the substring (peptide) ”DIQMSQ”, then a single training instance is the input-output tuple (DIQMS, Q). From our ten homologous sequences, we generate 2106 input-output pairs.

To be fed into a neural network, the input data must be encoded numerically, so we assign integer labels to each character for the training data. The output of each training instance is one-hot-encoded to represent the target classes of 20 different amino acids. Since we are training two models, one for forward prediction and one for reverse prediction, we generate the same training data for both the reverse and forward directions. We train models in both directions, because it produces the higher accuracy instead of training only one direction. More technical details can be discussed in section IV.

The samples of forward and reverse input-output pairs are divided into training and validation sets with the ratio of 90% and 10% respectively. We apply min-max normalization on the input dataset to squash the input values into the range [0, 1]. Normalization helps speed the training process.

C. Model Selection

For our sequence prediction task, we build four deep learning models combining convolutional neural networks (CNN) with long short term memory (LSTM). The CNN layer in theory helps with automatic feature extraction, in particular if there are patterns within each kmer that can be extracted for better predictions. The LSTM layer is the core of our models, because LSTM is useful for learning sequence dependencies in our input data. For our four hybrid models, we also add two dense hidden layers with ReLu activation and an output layer with softmax activation.

It is not clear a priori whether LSTM or bidirectional LSTM (Bi-LSTM) would be more effective in our prediction task, nor whether the addition of the convolutional layer would improve our results. We thus opt to evaluate and compare four hybrid models: LSTM, CNN-LSTM, Bi-LSTM, and CNN-Bi-LSTM. The workflow for the proposed approach is shown in Figure 1.

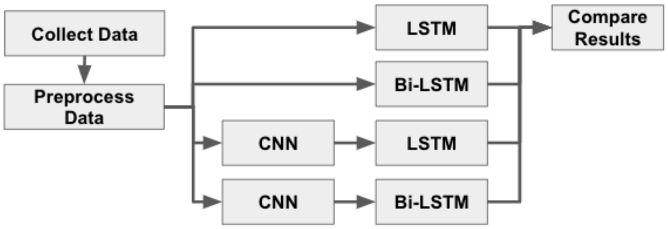


Fig. 1. Proposed Workflow

1) Convolutional Neural Networks (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 meaningful features. The convolutional layer has 128 filters with a filter size of 3, which indicates the kmer length cannot be smaller than 3. The feature maps from this layer form as input to the LSTM or Bi-LSTM layer.

2) Long short-term memory (LSTM): The LSTM architecture consists of 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 input sequences. 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. 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 long-short term memory (Bi-LSTM): Bidirectional LSTM (Bi-LSTM) is a modification of the LSTM architecture that permits the model to learn both the forward and reverse dependencies. Bi-LSTM uses 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.

IV. Our Model Architecture

All four of our hybrid models share common features. If there is a convolutional layer, it is the first layer after the input layer. The next layer is either the LSTM or Bi-LSTM layer. After that, the results are passed to two densely connected layers with ReLu activation, which pass their results to the final dense layer with softmax activation. The number of neurons in the final output layer equals the number of our target classes of 20 different amino acids. See Figure 2 for our proposed model architecture.

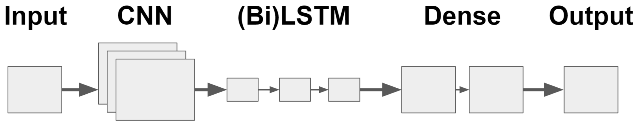


Fig. 2. Deep Learning Model Architecture

V. Experimental Results

A. Experimental Model Architectures

We initially experiment with a simple input-LSTM-output architecture. By itself, a single LSTM layer acheive high accuracy with minimal amount of training. We attribute this largely to the quality of our training dataset. Indeed, all of the training instances have between 82.52% to 89.32% similarity match to the scaffold of our target sequence. The lower the similarity between our scaffold and the training data, the lower our expected accuracy and thus the greater the difficulty in the sequencing task.

We add two hidden Dense layers after the initial LSTM layer and the accuracy improved marginally. In subsequent model trials, we keep the two fully connected layers since they show marginal improvement over leaving them out.

We settle on four competing architectures to run: LSTM, CNN-LSTM, BiLSTM, and CNN-BiLSTM. Their testing accuracies, final loss values, and final validation loss values are displayed in Table I. Figures 3, 4, 5, and 6 plot training accuracy against validation accuracy for our four models, and figures 7, 8, 9, and 10 plot training loss versus validation loss for all four models to show our models are not overfitting.

TABLE I
COMPARING DIFFERENT MODEL ARCHITECTURES

	Train Acc.	Valid Acc.	Train Loss	Valid Loss
LSTM	96.26%	92.38%	0.1020	0.5420
BiLSTM	93.72%	93.33%	0.4004	0.5474
CNN-LSTM	97.68%	94.76%	0.0586	0.3845
CNN-BiLSTM	97.57%	93.81%	0.0604	0.3711

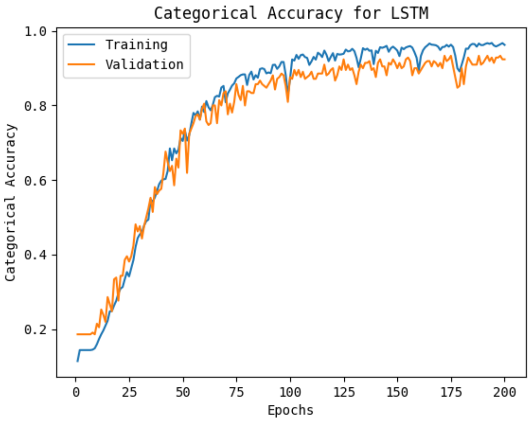


Fig. 3. LSTM Accuracy

B. Hyperparameters

Our hyperparameters were chosen through trial-and-error. We apply L2 regularization to reduce overfitting. We run our models on a range of L2 penalty values from $1e^{-8}$, $1e^{-7}$, $1e^{-6}$, $1e^{-5}$, $1e^{-4}$, $1e^{-3}$, $1e^{-2}$, and $1e^{-1}$. For each of our four model architectures, we choose the L2 penalty which most reduced our validation loss. These L2 values are displayed in Table II. Table III displays our chosen hyperparameters.

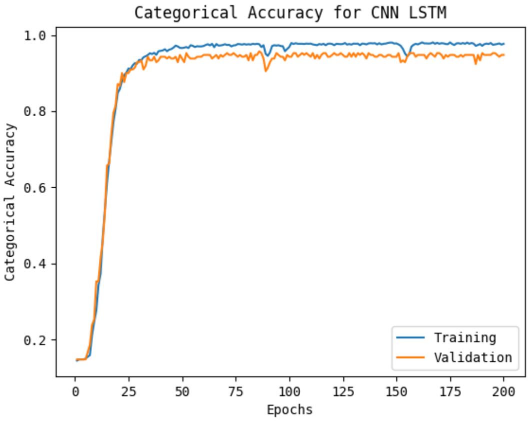


Fig. 4. CNN Accuracy

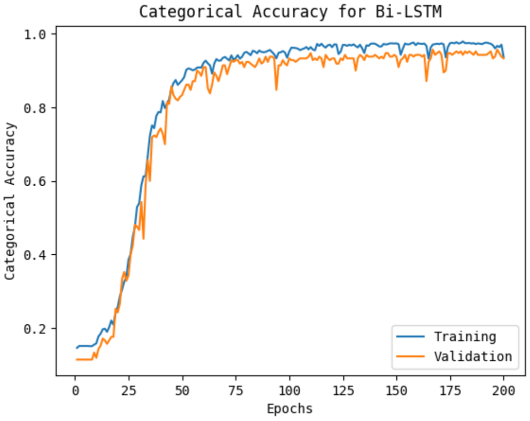


Fig. 5. Bi-LSTM Accuracy

TABLE II
L2 VALUES

Model Type	L2 Penalty
LSTM	$1e^{-8}$
CNN LSTM	$1e^{-3}$
Bi-LSTM	$1e^{-7}$
CNN Bi-LSTM	$1e^{-8}$

C. Protein Scaffold Filling

To deploy our models to predict gaps in the constructed protein scaffold described in section III A., we train two sets of models: one to learn the dependencies in the forward direction and one to learn the dependencies in the reverse direction. It is necessary to predict gaps in the beginning or end of the scaffold. For example, if our target sequence begins with ”DIQMSP” and we want to predict ”DIQ”, then we cannot predict from the forward direction, since ”DIQ” is the absolute start of the sequence. We must therefore be able to predict ”DIQ” given the sequence ahead of it. Similarly, we must predict in the forward direction for gaps at the absolute end of the sequence.

This use of two models implies a potential disagreement: one model may predict a different value than the other for those middle gaps with enough preceding values to create a

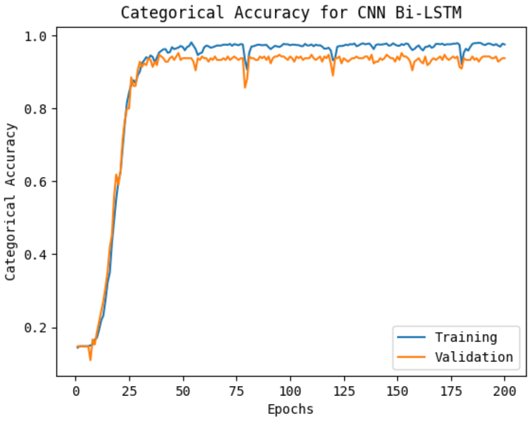


Fig. 6. CNN Bi-LSTM Accuracy

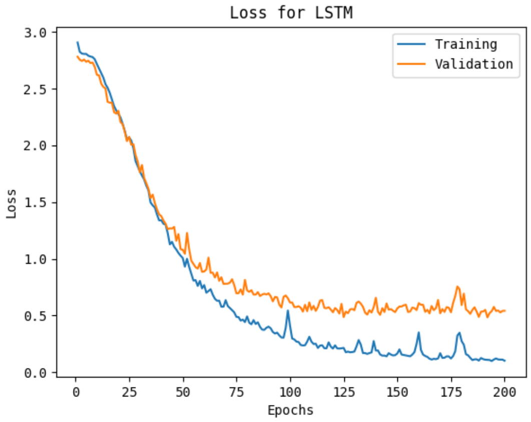


Fig. 7. LSTM Loss

TABLE III
HYPERPARAMETERS

	Hyperparameter	Value
Shared	Kmer Length	5
	Epochs	200
	First Dense Layer Neurons	128
	Second Dense Layer Neurons	64
	Cost Function	Categorical Cross Entropy
	Optimizer	ADAM
	Output Activation Function	Softmax
	Dense Layer Activation Function	ReLu
	Batch Size	64
CNN	Number of Convolutional Filters	128
	Convolutional Filter Size	3
LSTM	Number of LSTM Cells	256

valid input. See Figure 11 for an example of two different predictions from one sample run, in which red dash lines indicate amino acids that can not be predicted from that direction. To resolve any disagreement between two models, we opt to choose the prediction with the highest corresponding associated probability. By associated probability, we mean the

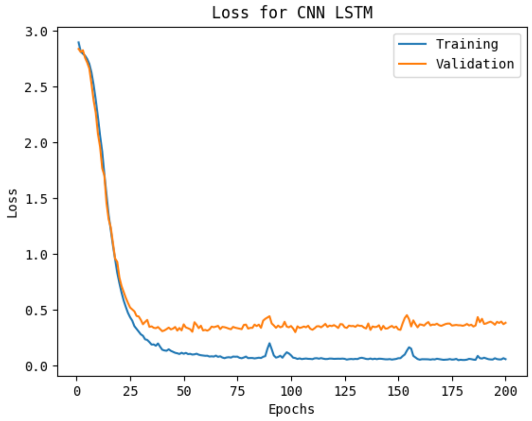


Fig. 8. CNN Loss

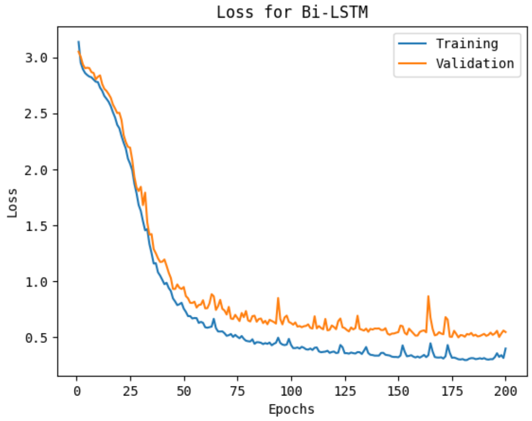


Fig. 9. Bi-LSTM Loss

output value of the output class with the highest probability. Since softmax outputs a probability distribution over the target classes, the prediction with the higher associated probability ought to be preferred since its higher probability represents the ”certainty” the model has in its prediction.

For our purposes, we assume it is known in advance the length of the gaps between the contigs. This assumption is reasonable, as with the top-down tandem mass spectrometry, it is easy to compute the total gap mass of the missing amino acids. Thus, the algorithm for filling gaps is straightforward. We iterate over the scaffold protein starting from amino acid c_k where k is in the kmer length. Until we reach the final amino acid in the entire scaffold, if we encounter a gap, i.e the end of a contig, we predict the next amino acid by deploying our trained model. We then take that prediction and use it to predict the next amino acid. The process is repeated until predictions have been made for all the gaps in the forward direction. We repeat this process in the reverse direction using the reverse model, and then resolve any disagreements by choosing the predicted amino acid with the higher associated probability. See Figure 11 for an illustration of conflicting

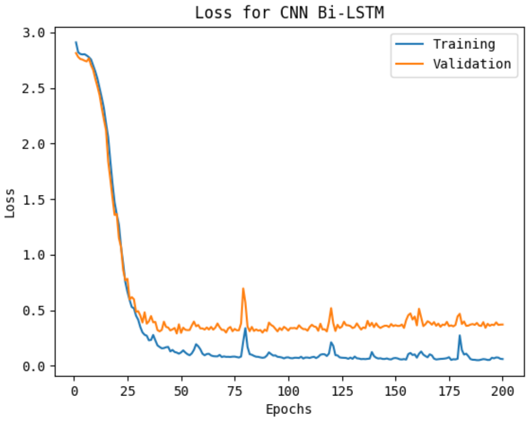
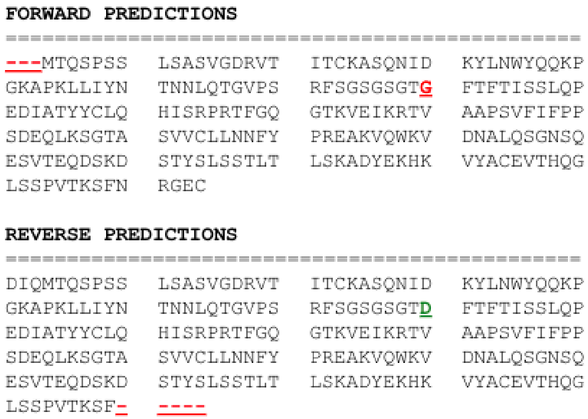


Fig. 10. CNN Bi-LSTM Loss

predicted sequences.



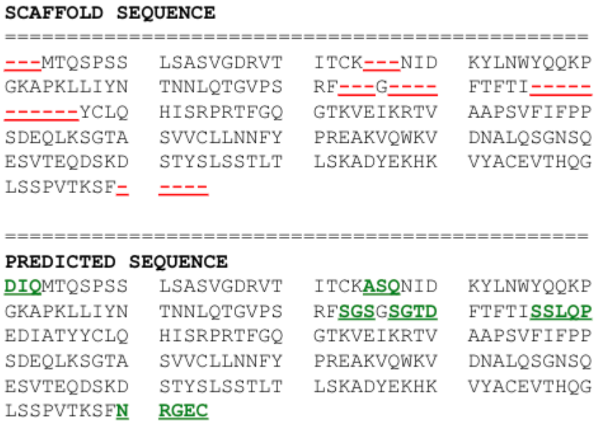
```
FORWARD PREDICTIONS
==============================================
---MTQSPSS   LSASVGDRVT   ITCKASQNID   KYLNWYQQKP
GKAPKLLIYN   TNNLQTGVPS   RFSGSGSGTG   FTFTISSLQP
EDIATYYCLQ   HISRPRTFGQ   GTKVEIKRTV   AAPSVFIFPP
SDEQLKSGTA   SVVCLLNNFY   PREAKVQWKV   DNALQSGNSQ
ESVTEQDSKD   STYSLSSTLT   LSKADYEKHK   VYACEVTHQG
LSSPVTKSFN   RGEC

REVERSE PREDICTIONS
==============================================
DIQMTQSPSS   LSASVGDRVT   ITCKASQNID   KYLNWYQQKP
GKAPKLLIYN   TNNLQTGVPS   RFSGSGSGTD   FTFTISSLQP
EDIATYYCLQ   HISRPRTFGQ   GTKVEIKRTV   AAPSVFIFPP
SDEQLKSGTA   SVVCLLNNFY   PREAKVQWKV   DNALQSGNSQ
ESVTEQDSKD   STYSLSSTLT   LSKADYEKHK   VYACEVTHQG
LSSPVTKSF-   ----
```

Fig. 11. Illustration of Conflicting Predictions

With the CNN-LSTM model, we deploy our dual models to predict gaps in the MabCampth scaffold. The green colored amino acids are predicted amino acids in Figure 12 in the scaffold with the CNN-LSTM mdoel, which achieve 100% accuracy for the test gaps. Note that, to evaluate our models, we know the whole MabCampth target sequence information so that we can easily calculate the gap filling accuracy by comparing the amino acids in the gaps with the corresponding position at the target sequence. See Figure 12 for the scaffold sequence on the top with red dash lines representing missing amino acids. The bottom image in Figure 12 represents the accurate predictions made on the MabCampth Scaffold. The code and results for this paper can be found on the public GitHub repository at *https://github.com/jsturtz/protein_scaffold_filling*.

D. Comparison Results

For the de novo protein scaffold filling problem, Qingge et al. [1] developed several combinatorial algorithms based on dynamic programming, local search and greedy methods. We compare our proposed deep learning models to the algorithms in [1] to show the effectiveness of our methods. Qingge et al.'s results are summarized in the Table IV using different homologous sequence references: Humira, Huamn Germline Antibody, and S1V2-83 light chain, which have 91%, 92% and 90.65% similarity with the target sequence respectively. Qingge et al.'s algorithm [1] only uses one homologous sequence as a reference to fill the gaps in the scaffold, from which we obtain 90%, 84.79% and 77.06% gap filling accuracy (one-to-one match) in the scaffold with respect to the corresponding positions in the target sequence. However, from our deep learning models, we fill the gaps in the scaffold with 100% accuracy for MabCampth data despite using lower similarity scores ranging between 82.52% to 89.32% with respect to our scaffold.



```
SCAFFOLD SEQUENCE
==============================================
---MTQSPSS   LSASVGDRVT   ITCK---NID   KYLNWYQQKP
GKAPKLLIYN   TNNLQTGVPS   RF---G----   FTFTI-----
------YCLQ   HISRPRTFGQ   GTKVEIKRTV   AAPSVFIFPP
SDEQLKSGTA   SVVCLLNNFY   PREAKVQWKV   DNALQSGNSQ
ESVTEQDSKD   STYSLSSTLT   LSKADYEKHK   VYACEVTHQG
LSSPVTKSF-   ----

==============================================
PREDICTED SEQUENCE
DIQMTQSPSS   LSASVGDRVT   ITCKASQNID   KYLNWYQQKP
GKAPKLLIYN   TNNLQTGVPS   RFSGSGSGTD   FTFTISSLQP
EDIATYYCLQ   HISRPRTFGQ   GTKVEIKRTV   AAPSVFIFPP
SDEQLKSGTA   SVVCLLNNFY   PREAKVQWKV   DNALQSGNSQ
ESVTEQDSKD   STYSLSSTLT   LSKADYEKHK   VYACEVTHQG
LSSPVTKSFN   RGEC
```

Fig. 12. 100% Accuracy on MabCampth Scaffold

TABLE IV
GAP FILLING ACCURACY IN QINGGE, ET AL.

Reference	Similarity	Gap Filling Accuracy
Humira	91.1%	90%
Human Germline Antibody	92%	84.79%
S1V2-83 light chain	90.65%	77.06%

Reference column indicates the references to the target sequence. Similarity column shows the similarity between reference and target sequence and the Gap Filling Accuracy column represents the prediction accuracy in the gaps from Qingge et al. [1]

VI. CONCLUSION

We explore the possibility of using deep learning techniques to solve an otherwise hard problem of predicting gaps in the constructed protein scaffold generated from top-down and bottom-up tandem mass spectrometry. In this paper, we demonstrate that the various hybrid CNN and LSTM-based deep learning approaches are very effective by obtaining the complete protein sequence with 100% gap filling accuracy in the scaffold and 100% coverage with the known target sequence, which outperform the traditional combinatorial based algorithms on MabCampth data.

Due to the page limitation of the conference paper, we only show the results running on MabCampth data. In the future (full) version, we will show the robustness of our model on a larger dataset and build an unified computational framework for any *de novo* protein scaffold filling problem combined with top-down and bottom-up tandem mass spectrometry analysis.

Furthermore, top-down and bottom-up tandem mass spectrometry analysis cannot distinguish between amino acids *I* and *L*, because they have the same weight. Deep learning models open the door to distinguish amino acids *I* and *L* in the protein sequencing problem.

Acknowledgment

We thank anonymous reviewers for their insightful comments. We thank Swetha Chittam and Lawrence Owusu for their useful inputs and discussion.

References

[1] L. Qingge, X. Liu, F. Zhong, and B. Zhu, "Filling a protein scaffold with a reference," *IEEE transactions on nanobioscience*, vol. 16, no. 2, pp. 123–130, 2017.

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

[3] 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.

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

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

[12] N. H. Tran, X. Zhang, L. Xin, B. Shan, and M. Li, "De novo peptide sequencing by deep learning," *Proceedings of the National Academy of Sciences*, vol. 114, no. 31, pp. 8247–8252, 2017.

[13] B. Ma, K. Zhang, C. Hendrie, C. Liang, M. Li, A. Doherty-Kirby, and G. Lajoie, "Peaks: powerful software for peptide de novo sequencing by tandem mass spectrometry," *Rapid communications in mass spectrometry*, vol. 17, no. 20, pp. 2337–2342, 2003.

[14] B. Ma, "Novor: real-time peptide de novo sequencing software," *Journal of the American Society for Mass Spectrometry*, vol. 26, no. 11, pp. 1885–1894, 2015.

[15] A. Frank and P. Pevzner, "Pepnovo: de novo peptide sequencing via probabilistic network modeling," *Analytical chemistry*, vol. 77, no. 4, pp. 964–973, 2005.

[16] N. H. Tran, R. Qiao, L. Xin, X. Chen, C. Liu, X. Zhang, B. Shan, A. Ghodsi, and M. Li, "Deep learning enables de novo peptide sequencing from data-independent-acquisition mass spectrometry," *Nature methods*, vol. 16, no. 1, pp. 63–66, 2019.

[17] R. Qiao, "Peptide sequencing with deep learning," 2020.

[18] F. T. Zohora, M. Z. Rahman, N. H. Tran, L. Xin, B. Shan, and M. Li, "Deepiso: a deep learning model for peptide feature detection from lc-ms map," *Scientific reports*, vol. 9, no. 1, pp. 1–13, 2019.

[19] S. Guan, M. F. Moran, and B. Ma, "Prediction of lc-ms/ms properties of peptides from sequence by deep learning*[s]," *Molecular & Cellular Proteomics*, vol. 18, no. 10, pp. 2099–2107, 2019.

[20] R. Satija, J. A. Farrell, D. Gennert, A. F. Schier, and A. Regev, "Spatial reconstruction of single-cell gene expression data," *Nature biotechnology*, vol. 33, no. 5, pp. 495–502, 2015.

[21] D. van Dijk, J. Nainys, R. Sharma, P. Kaithail, A. J. Carr, K. R. Moon, L. Mazutis, G. Wolf, S. Krishnaswamy, and D. Pe'er, "Magic: A diffusion-based imputation method reveals gene-gene interactions in single-cell rna-sequencing data," *BioRxiv*, p. 111591, 2017.

[22] M. Huang, J. Wang, E. Torre, H. Dueck, S. Shaffer, R. Bonasio, J. Murray, A. Raj, M. Li, and N. R. Zhang, "Saver: Gene expression recovery for umi-based single cell rna sequencing," *bioRxiv*, p. 138677, 2018.

[23] W. V. Li and J. J. Li, "An accurate and robust imputation method scimpute for single-cell rna-seq data," *Nature communications*, vol. 9, no. 1, pp. 1–9, 2018.

[24] H. T. N. Tran, K. S. Ang, M. Chevrier, X. Zhang, N. Y. S. Lee, M. Goh, and J. Chen, "A benchmark of batch-effect correction methods for single-cell rna sequencing data," *Genome biology*, vol. 21, no. 1, pp. 1–32, 2020.

[25] D. Risso, F. Perraudeau, S. Gribkova, S. Dudoit, and J.-P. Vert, "A general and flexible method for signal extraction from single-cell rna-seq data," *Nature communications*, vol. 9, no. 1, pp. 1–17, 2018.