Project Report

Lawrence Owusu, Jordan Sturtz, and Swetha Chittam

***Abstract*—De novo protein sequencing using mass spectrometry or Edman degradation suffers from the problem of producing only partial sequences of target proteins. We explore the possibility of using deep learning techniques to perform the task of predicting gaps in partially sequenced proteins. Our method involves querying the NCBI Protein Blast server for closest matching homologous sequences to a partial sequence, then training two deep learning models to predict in the forward and reverse directions the missing gaps. Our results show high accuracy (98-99%) in filling gaps of a particular partially sequenced de novo sequence of the light chain of alemtuzumab.**

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 then train and deploy two models: one to learn dependencies in the forward direction and another to learn dependencies in the reverse direction. We combine the two models' predictions to fill the gaps in the de novo protein sequence.

We train and evaluate four competing hybrid deep learning models combining LSTM and CNN layers. For the purpose of this project, we tested our model against only a single target sequence: the light chain of alemtuzumab. Our results show high accuracies (98-99%) for all four of our model architectures, with our Bi-LSTM-CNN model outperforming the others at 99.04%.

A. Related Work

Mass spectrometry continues to be a dominant method to perform protein sequencing [1]. 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-translationally 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, and the output of each associated kmer is the next character in the sequence. Thus, for instance, if one of our homologous

The authors are graduate students at NCA&T

sequences contains the substring ”DIQMSQ”, then a single training instance would be the input-output tuple (DIQMS, Q).

From our ten homologous sequences, we generated 2106 number of input-output pairs with an input length of 5. The kmer length represents 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 samples of forward and reverse input-output pairs are divided into training and validation sets with the ratio of 90% and 10% respectively. The validation set is used to prevent overfitting.

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 Selection

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 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, since LSTM is useful for learning sequence dependencies in our input data. For our four hybrid models, we also added two dense hidden layers with ReLu activation and a final output layer with softmax activation.

It is not clear a priori whether LSTM or Bi-LSTM would be more effective in our prediction task; nor is it clear whether the addition of the convolutional layer would improve our results. We thus opted 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.

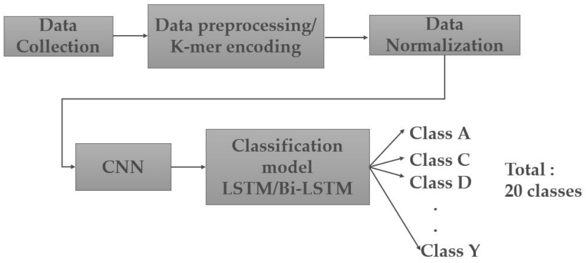


Fig. 1. Model Diagram for Multi-Classification of Sequence Data

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: 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 target dataset. 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 (Figure 2). 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.

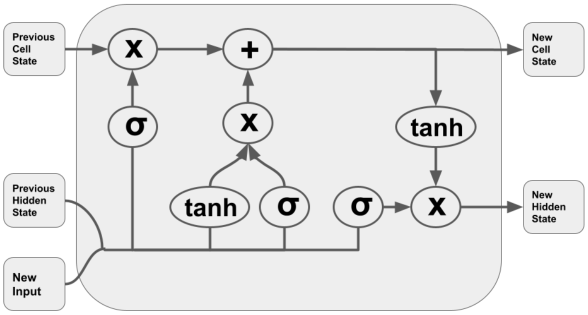


Fig. 2. LSTM Architecture Inside a Memory Cell

3) 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. See Figure 3 for an illustration.

III. Overall 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. After that, the next layer is either the LSTM or Bi-LSTM layer. After that, the results are passed to two hidden densely connected layer 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. See Figure 4.

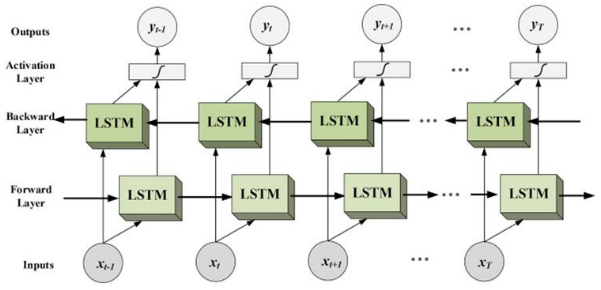


Fig. 3. Bidirectional LSTM Architecture

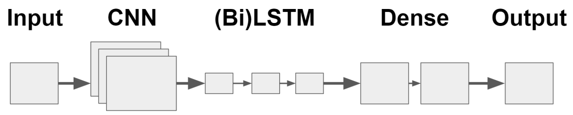


Fig. 4. Deep Learning Model Architecture

IV. Experimental Results

A. Experimental Model Architectures

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

We added two hidden Dense layers after the initial LSTM layer and the accuracy improved marginally. In subsequent model trials, we kept the two hidden layers since they showed marginal improvement over leaving them out.

We settled 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.

Our models all exhibited slight overfitting, though not enough to be significant. Figures 5, 6, 7, and 8 plot training accuracy against validation accuracy for our four models, and Figures 9, 10, 11, and 12 plot training loss versus validation loss for all four models.

TABLE I
Comparing Different Model Architectures

	Test Accuracy	Training Loss	Validation Loss
LSTM	98.56%	0.0964	0.4546
Bi-LSTM	98.09%	0.1282	0.6051
CNN-LSTM	98.56%	0.0948	0.5604
Bi-CNN-LSTM	99.04%	0.0884	0.6719

B. Hyperparameters

Our hyperparameters were chosen through manual trial-and-error. We did not implement any regularization techniques, nor did we use our validation set or cross validation to tune any hyperparameters. All of our results for our different model architectures were generating high accuracy (98-99%) on our single de novo sequence. Table II displays our chosen hyperparameters.

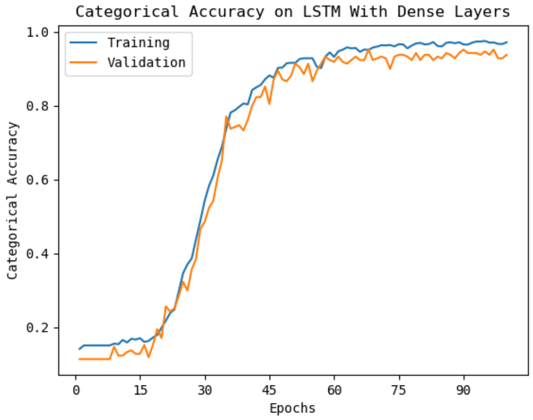


Fig. 5. LSTM Accuracy

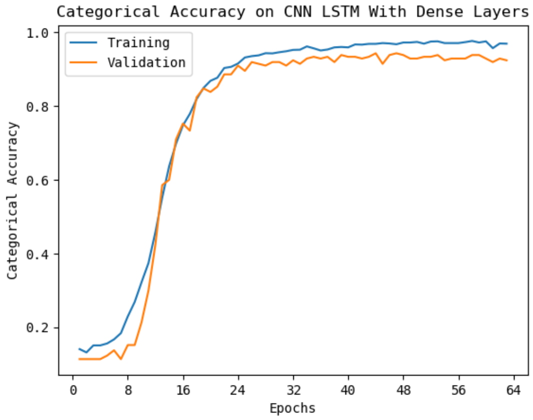


Fig. 6. CNN LSTM Accuracy

TABLE II
Hyperparameters

	Hyperparameter	Value
Shared	Kmer Length	5
	Epochs	100
	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

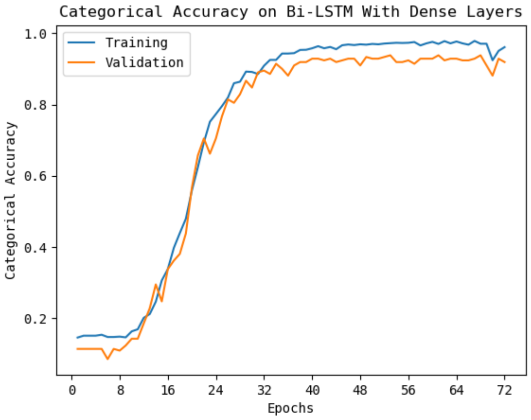


Fig. 7. Bi-LSTM Accuracy

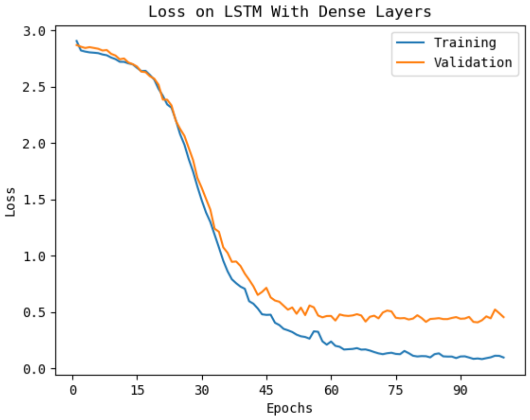


Fig. 9. LSTM Loss

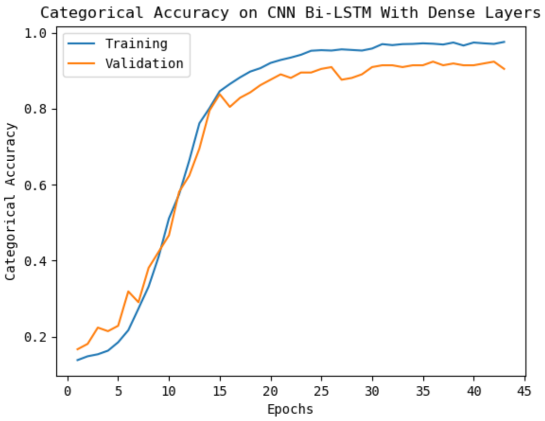


Fig. 8. CNN Bi-LSTM Accuracy

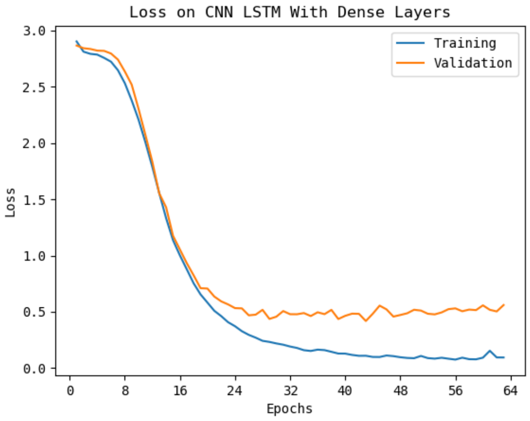


Fig. 10. CNN LSTM Loss

C. Predictions on De Novo Sequence

To deploy our models to predict gaps in our de novo sequence, we trained two models: one to learn the dependencies in the forward direction and one to learn the dependencies in the reverse direction. This was necessary to predict gaps in the beginning or end of the target sequence. 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 valid input. See Figure 13 for an example of two different predictions from one sample run.

To resolve any disagreement between the two models, we opted to choose the prediction with the highest corresponding associated probability. By associated probability, we mean the 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.

With this simple algorithm, we deployed our dual models to predict gaps in our de novo sequence, achieving 100% accuracy for the test gaps. See Figure 14.

V. Conclusion

Our project aims to explore the possibility of using deep learning techniques to solve an otherwise hard problem of predicting gaps in partial peptide de novo sequences. We have demonstrated that with various LSTM-based deep learning models it is possible to achieve high accuracy on peptide sequences with gaps that have high similarity scores in known protein databases.

The approach we describe here generalizes to any other de novo sequence and in fact other sequencing problems beyond protein sequencing. Further research is needed to explore the challenges and limitations of this approach, in particular for de novo sequences for which the highest similarity score in known protein databases is comparatively low. Moreover, our approach relies on an assumption that gap lengths in partial sequences are known in advance. In practice, partial sequence gaps may be unknown. It may be possible to solve this problem

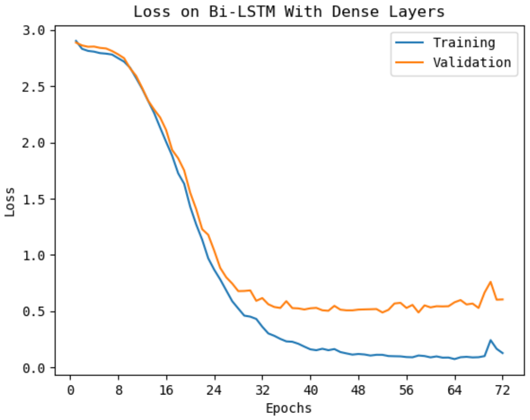


Fig. 11. Bi-LSTM Loss

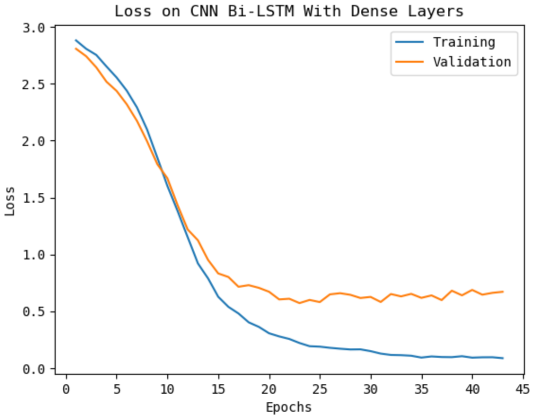
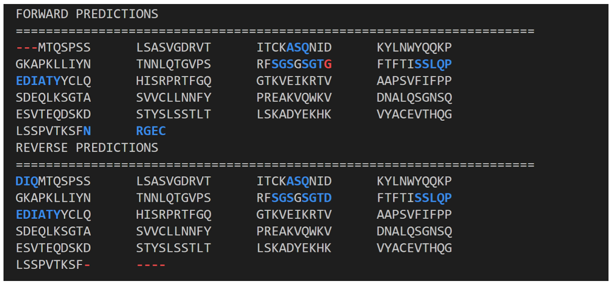


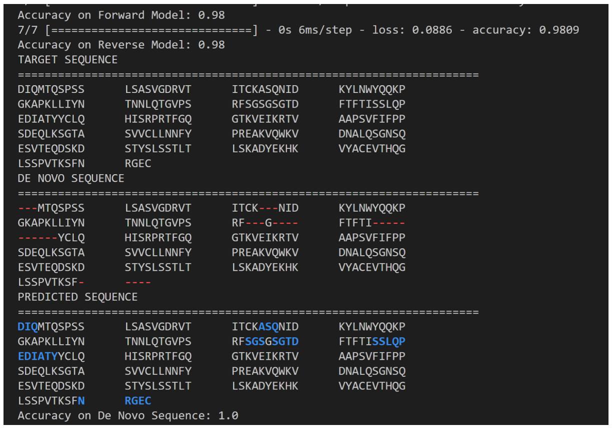
Fig. 12. CNN Bi-LSTM Loss

by employing forward and reverse predictions to generate overlapping sections or to use other data features (for instance, known total mass of the gap) to successfully predict gaps of unknown length.



```
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. 13. Illustration of Conflicting Predictions



```
Accuracy on Forward Model: 0.98
7/7 [==============================] - 0s 6ms/step - loss: 0.0886 - accuracy: 0.9809
Accuracy on Reverse Model: 0.98
TARGET SEQUENCE
================================================================
DIQMTQSPSS      LSASVGDRVT      ITCKASQNID      KYLNWYQQKP
GKAPKLLIYN      TNNLQTGVPS      RFSGSGSGTD      FTFTISSLQP
EDIATYYCLQ      HISRPRTFGQ      GTKVEIKRTV      AAPSVFIFPP
SDEQLKSGTA      SVVCLLNNFY      PREAKVQWKV      DNALQSGNSQ
ESVTEQDSKD      STYSLSSTLT      LSKADYEKHK      VYACEVTHQG
LSSPVTKSFN      RGEC
DE NOVO 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
Accuracy on De Novo Sequence: 1.0
```

Fig. 14. Illustration of Complete Predictions

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.