A Convolutional Denoising Autoencoder for Protein Scaffold Filling

Jordan Sturtz[1], Richard Annan[1], Binhai Zhu[2], Xiaowen Liu[3], and Letu Qingge[1](✉)

[1] Department of Computer Science, North Carolina A&T State University,
Greensboro, NC, USA
{jasturtz,rkannan}@aggies.ncat.edu, lqingge@ncat.edu

[2] Gianforte School of Computing, Montana State University,
Bozeman, MT, USA
bhz@montana.edu

[3] John W. Deming Department of Medicine, Tulane University,
New Orleans, LA, USA
xwliu@tulane.edu

Abstract. De novo protein sequencing is a valuable task in proteomics, yet it is not a fully solved problem. Many state-of-the-art approaches use top-down and bottom-up tandem mass spectrometry (MS/MS) to sequence proteins. However, these approaches often produce protein scaffolds, which are incomplete protein sequences with gaps to fill between contiguous regions. In this paper, we propose a novel convolutional denoising autoencoder (CDA) model to perform the task of filling gaps in protein scaffolds to complete the final step of protein sequencing. We demonstrate our results both on a real dataset and eleven randomly generated datasets based on the MabCampath antibody. Our results show that the proposed CDA outperforms recently published hybrid convolutional neural network and long short-term memory (CNN-LSTM) based sequence model. We achieve 100% gap filling accuracy and 95.32% full sequence accuracy on the MabCampth protein scaffold.

Keywords: De Novo Protein Sequencing · Convolutional Layer · Denoising Autoencoder · Protein Scaffold Filling

1 Introduction

Protein sequencing plays an important role in many aspects of proteomics, including identification of structure and functions of proteins, new protein biomarkers, construction of phylogenetic tree to find evolutionary relationship and new drug design. De novo protein sequencing refers to the process of determining the primary structure of proteins directly without inferring the full sequence by merely matching against an existing protein database. Complete de novo protein sequencing remains a challenging problem in bioinformatics.

This work is supported by the National Science Foundation of the United States under Award 2307571, 2307572 and 2307573.

X. Guo et al. (Eds.): ISBRA 2023, LNBI 14248, pp. 518–529, 2023.
https://doi.org/10.1007/978-981-99-7074-2_42

Every protein can be defined by its unique sequence of amino acids, which is called its primary structure. Proteins are comprised of 20 different amino acids. We use the term "peptide" to refer to small multi-amino acid sub-units of proteins. The goal of peptide or protein sequencing is to determine the complete unique sequence of amino acids in a peptide or protein. In general, peptide or protein sequencing from mass spectrometry can refer to either de novo sequencing or database searching. With database searching, once a mass spectrum is generated, it is compared to databases of known peptides or proteins to retrieve the sequence with the closest matching mass spectrum. Often, these databases will include only proteins or peptides generated from genomic data [10]. Many proteins of interest are not included in such databases, especially those that are not directly inscribed in genomes such as monoclonal antibodies. Even if a protein sequence is known, it is often still desirable to perform de novo sequencing to discover novel proteoforms [11]. For instance, proteoforms may be created by post-translational modifications, which occur when amino acids of proteins undergo a process of proteolytic cleavage which alters the amino acid in the primary structure by adding a modifying group [8,9]. De novo protein sequencing has been used for many purposes, including full sequencing of proteins, to sequence endogenous peptides [12,13], to characterize mutations in antibodies [14], and to perform proteomic analysis of novel organisms not found in protein databases.

We organize our paper as follows. In Sect. 2, we discuss the problem statement and gap challenges that motivate our research, deficiencies in existing approaches. In Sect. 3, we introduce the methodology that we will use to develop a new convolutional denoising autoencoder (CDA) model as a solution. In Sect. 4, we discuss in detail our proposed CDA model, including data preprocessing, model architecture and hyperparameters tuning steps. In Sect. 5, we show our experimental prediction results both on the original real MabCampth scaffold data and simulation data. Finally, we conclude our paper and discuss the future directions.

2 Preliminaries

The Protein Scaffold Filling (PSF) 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.

The protein scaffold filling problem has been shown to be polynomial solvable in $O(n^{26})$ time [4]. In [4], the authors proposed several practical algorithms based on greedy algorithm, dynamic programming and local search. These algorithms rely on high quality homologous reference proteins. As reported in [4], these algorithms run in a reasonable amount of time when gaps are small. Thus, our goal is to investigate deep learning approaches to the same problem to improve our accuracy, especially when gaps are large or the homologous reference proteins are dissimilar to proteins scaffolds produced in a lab.

Most recently in 2022, the authors [7] developed several deep learning models based on CNN and LSTM models for the PSF problem and achieved high accuracy when filling the gaps in the MabCampath scaffold dataset. The basic idea behind this approach is to iteratively predict each amino acid in sequence by deploying a model that can predict the next amino acid given the preceding K amino acids. From left to right, when a gap is encountered in the protein scaffold, the model predicts the next amino acid as a replacement for that gap. This process is repeated until all gaps are filled. The authors trained a forward model and reverse model so they can predict gaps at the end of any protein scaffold. For training data, the authors query for homologous sequences to their scaffold protein, then generate all kmers of each training instance. Each kmer represents a single training instance input, and the amino acid after the kmer in the sequence is the training output. So, for example, from the sequence DIQMSPIL..., the following input-output pairs would be generated: (DIQMS, P), (IQMSP, I), (QMSPI, L). The authors trained various CNN-LSTM hybrid models to compare their accuracy.

Though their reported accuracy is higher than that reported in [4], this approach suffers from a few flaws. First, since the model is a kmer sequence-based approach, any errors in inference are likely to propagate, leading to subsequent incorrect inferences. See Fig. 1 for an illustration. If this issue is indeed a significant problem for the sequence-based approach, it suggests that such approaches will tend to do worse when the gaps to fill between contigs of a protein scaffold are particularly large.

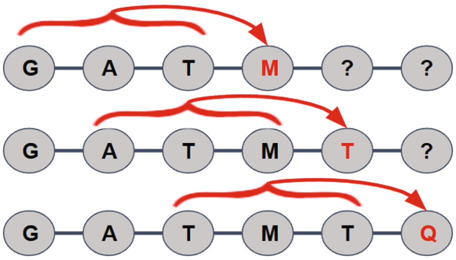


Fig. 1. CNN-LSTM model illustration [7]. Since during inference the model predicts only the next amino acid, if it makes a poor prediction, it will feed that poor prediction into the next inference step, causing future inferences to be unreliable

In this paper, our goal is to develop a deep learning model that can accurately predict the missing amino acids in gaps of the scaffold while improving on the approach described in [7] by also correcting incorrect amino acids in the existing scaffold.

3 Methodology

The approach we use is a convolutional denoising autoencoder (CDA) trained on homologous sequences of our given scaffold. The motivation behind an autoencoder in general is that it imputes all the missing amino acids at once, which is different from the iterative sequence-based approach described in [7]. Not only can the CDA predict gaps in the scaffold but it can also correct any incorrect amino acids in the scaffold. In contrast, the LSTM models designed in [7] can only predict the missing amino acids in the gaps of the scaffold.

Autoencoders. Autoencoders are neural networks that learn how to reconstruct its input through the composition of an encoder and a decoder [1]. Typically, the idea is to encode the original input into a lower dimensional space and then decode the compressed representation into the original input.

Denoising Autoencoders. Simple autoencoders suffer from the problem that the autoencoder may simply learn an identity function, which produces trivially useless results [2]. A common solution to this problem is to intentionally corrupt the original input in some way by adding some kind of "noise" to the data. The goal of the autoencoder, then, is to learn how to denoise the corrupted input, which produces a more robust representation that avoids trivial solutions [3]. A model trained on corrupted inputs can learn an internal representation that can correct those defects.

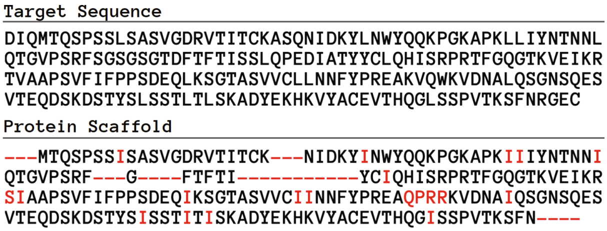
Convolutional Layers. Convolutional layers in a neural network are useful whenever the input contains hidden features created by the relationships among neighboring components of the input. In this way, convolutional layers can be viewed as automatic feature extractors. Since the dataset consists of sequences of amino acids, it is a reasonable hypothesis that there are meaningful features to extract among neighboring values of each sequence.

Pooling and Upsampling. Pooling is in general a useful technique to reduce model complexity to speed up training. In our case, pooling is how the model achieves the compression characteristic of autoencoders. The model convolves the original input to extract features, then compresses those feature maps with pooling into a reduced dimensional space. The decoder portion of the autoencoder performs inverse convolutions and upsampling to produce the final sequence length of the training data.

4 The Proposed Convolutional Denoising Autoencoder Model

4.1 Data Collection

The protein scaffold we use to evaluate our proposed model is the light chain of alemtuzumab (MabCampath). In [5], the authors generated the MabCampath scaffold data by combining top-down and bottom-up tandem mass spectrometry. This scaffold includes five contigs and six contiguous gaps of missing amino acids. The main steps of generating the MabCampth scaffold consists of converting raw spectra to a prefix residue mass (PRM) spectra, 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. More technical details about generating the MabCampath protein scaffold can be found in [5]. The scaffold information can be seen in Fig. 2, in which the red colored dash line represents gaps in the scaffold and the other red characters are non-gap errors in the scaffold. We feed the scaffold into NCBI's Protein Blast Server [6] to retrieve 1000 homologous sequences as our training data.



```
Target Sequence
DIQMTQSPSSLSASVGDRVTITCKASQNIDKYLNWYQQKPGKAPKLLIYNTNNL
QTGVPSRFSGSGSGTDFTFTISSLQPEDIATYYCLQHISRPRTFGQGTKVEIKR
TVAAPSVFIFPPSDEQLKSGTASVVCLLNNFYPREAKVQWKVDNALQSGNSQES
VTEQDSKDSTYSLSSTLTLSKADYEKHKVYACEVTHQGLSSPVTKSFNRGEC
Protein Scaffold
---MTQSPSSISASVGDRVTITCK---NIDKYINWYQQKPGKAPKIIIYNTNNI
QTGVPSRF---G----FTFTI-----------YCIQHISRPRTFGQGTKVEIKR
SIAAPSVFIFPPSDEQIKSGTASVVCIINNFYPREAQPRRKVDNAIQSGNSQES
VTEQDSKDSTYSISSTITISKADYEKHKVYACEVTHQGISSPVTKSFN----
```

Fig. 2. Dashes are missing amino acids, i.e., gap errors. The other red-colored characters are non-gap errors in the given Protein Scaffold. Target Sequence is a ground truth sequence that we will predict. (Color figure online)

As our model depends on padding shorter protein sequences with empty amino acids, we also prune the collected training data by limiting the lengths of acceptable training sequences to those where the length is between 95% to 105% of the length of the target. In this way, we reduce the required amount of padding in our training data to allow for varied sequence lengths while also minimizing biases that may occur due to the model learning the noise of the extra padding. To get a sense for the quality of training data for each test scaffold, we choose the homologous sequences with the range of 205–224 lengths, the range of 98%-100% query coverage, and the range of 44%-89% percent identical similarity among sequences in each training dataset. The query coverage refers to the percentage of the queried sequence that is covered by the returned sequence, whereas the percent similarity refers to the percent of one-to-one matches in the sequence alignments.

4.2 Data Preprocessing

One-Hot Encoding. In general, there are two ways to represent categorical data. The first method is label-encoding, in which each category is assigned a numerical value. The second method is one-hot encoding, in which each category is represented by a binary vector where the position of the 1 in the binary vector represents the category of the datum.

One-hot encoding is often a preferred method for categorical data and it is the type of encoding we choose here. Thus, our network must learn a representation where the full input dataset is a tensor of shape (samples, sequence_length, classes).

Noisification. To add noise to our input data, we add a new class label to represent emptiness. Thus, in data preprocessing, a percent P of the amino acids are replaced by the empty class represented by blank.

Padding. Not all sequences in the training data will have the same lengths. To feed these sequences into a neural network, it is therefore necessary to employ a strategy to either pad or truncate training sequences to get a fixed length. We opt to pad each training sequence with empty amino acids until the lengths reach the maximum length sequence in the training data. Let S be the maximum length of the sequence in the training data. It is important that pooling layers in our model cause a reduction in the size of the feature maps such that upsampling in the decoding phase produces the same shape as our target inputs. For instance, suppose S is 211 and the neural network has two pooling layers. In this case, the encoder will produce a length of 52: $\lfloor \lfloor 211/2 \rfloor /2 \rfloor = 52$. But if a shape of 52 is then upsampled in the decoder, it produces an output length of 208: $52 * 2 * 2 = 208$. We want the output of the neural network to have a length of 211 to match the length of the input. To solve this technical problem, we increment S until $S \mod L = 0$, where L is the product of the shapes of the pooling layers.

The Model Architecture. The final model architecture is illustrated in Fig. 3. There are two convolutional layers in the encoder, each followed by max pooling and dropout layers. Likewise, there are two inverse convolutional layers in the decoder followed by upsampling and dropout layers. We split our dataset into training and validation of 85% and 15% respectively. The more details about the model architecture can be found in Fig. 3. Noise and padding are added to the model input, then it is one-hot encoded before running through the encoder, which ultimately compresses the input into a reduced dimensional space. The decoder portion of the neural network reconstructs the input using upsampling. Dropout is added to reduce overfitting. The model hyperparameters are listed in Table 2. Our developed code can be found from https://github.com/astonish24/QinggeLab_ISBRA23_paper.

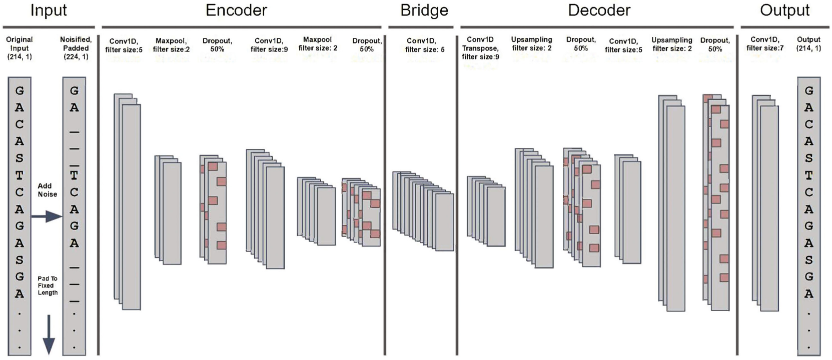


Fig. 3. The proposed convolutional denoising autoencoder (CDA) model architecture.

4.3 Simulation Data

A protein scaffold produced using MS/MS will contain errors both in its contigs (what we will call *non-gap* errors) as well as gaps that need to be filled between contigs to complete the entire protein (what we will call *gap* errors). The number of amino acids to fill between contigs as well as the number of total errors will vary from protein scaffold to protein scaffold. For this reason, and because we are interested in comparing our results to the up-to-date sequence-based approach described in [7], we generate random protein scaffolds from our target protein sequence. Note that, for validation purpose, we know our target sequence that we are constructing.

We generate eleven new protein scaffolds using combinations of three values for errors percentage (20%, 30%, and 40%) and four values (4, 6, 8, 10) for the number of contiguous gaps size in each scaffold. To maintain realistic artificially generated protein scaffolds, we split the percent error into a ratio of 60/40 for gap and non-gap errors respectively, which roughly corresponds to the ratio present in the protein scaffold produced by [5].

Once the eleven protein scaffolds are generated, we collect training data by querying the National Center for Biotechnology Information (NCBI) Protein BLAST server to retrieve the top 1000 most similar reference proteins [6]. Table 1 shows each training dataset and the range of percent identical similarity in the returned reference proteins. The protein scaffolds with smaller values for reference similarity are likely to have worse results, since the training data will be based on less similar reference proteins.

Table 1. Generated Protein Scaffolds and Training Similarity

ID	# Contiguous Gaps	% Incorrect	Reference Similarity
1	6	20%	80.4% - 68.2%
2	8	20%	87.8% - 70.9%
3	10	20%	92.5% - 73.9%
4	4	30%	75.7% - 64.0%
5	6	30%	71.4% - 61.2%
6	8	30%	80.2% - 62.3%
7	10	30%	71.2% - 57.7%
8	4	40%	88.1% - 65.1%
9	6	40%	68.2% - 60.1%
10	8	40%	68.2% - 60.6%
11	10	40%	67.4% - 53.5%

5 Results and Comparison

We compare the performance of our proposed model with the recently developed hybrid CNN-LSTM [7] in terms of gap filling accuracy and full sequence accuracy. The gap filling accuracy is computed by dividing the number of correct predictions on missing gaps by the number of missing gaps in the scaffold, where we use the target sequence as a ground truth sequence. The full sequence accuracy is the percentage of one-to-one matches between the full prediction and the target protein. Note that the CNN-LSTM model only predict the missing amino acids in the gaps. While our proposed denoinsing autoencoder model not only predict the missing amino acids in the gaps but also it has an ability to correct the amino acids in the scaffolds which is obtained from bottom-up and top-down methods. Also, in the bottom-up and top-down methods, it cannot distinguish the same weight amino acids *I* and *L*. However, our proposed model is able to correctly identify both *I* and *L* in the predicted sequence.

Table 2. The CDA Hyperparameters

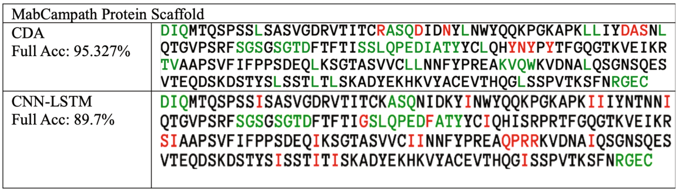
learning_rate	3.061E-4
dropout_percent	0.50
bridge_filters	160
conv_filters1	46
conv_filters2	90
conv_filter_size1	5
conv_filter_size2	9
bridge_filter_size	5
final_filter_size	7
kmer_size	15
noise_percent	40%

5.1 Results on the MabCampath Scaffold

We run both our proposed CDA and the CNN-LSTM based model [7] discussed in Sect. 2 on the original MabCampath scaffold. Both models did not appear to display any overfitting. Figure 5 shows training and validation accuracy for both models, and Fig. 6 shows training and validation losses for both models.

We also display the predictions for both the CDA and the CNN-LSTM models on the original scaffold protein in Fig. 4. In this figure, the green colored amino acids are correctly predicted amino acids and the red colored amino acids are incorrectly predicted amino acids from both CDA and CNN-LSTM models. From our proposed model, we also achieve 100% gap filling accuracy as the CNN-LSTM model produced in [7]. While for the full sequence accuracy, our model obtain 95.32% accuracy compared with the target sequence which outperforms the CNN-LSTM model's 89.7% accuracy [7].

The non-gap accuracy, which is the percentage of correct predictions on non-gap region in the protein scaffold with respect to the target sequence. The non-gap accuracy will always be 0% for the sequence-based approach, since the sequence-based approach cannot in principle attempt to correct non-gap errors. On the other hand, since the CDA imputes the full protein sequence, which is taken as the prediction for all amino acids, the autoencoder may at times incorrectly change amino acids that should not have been altered. It is for this reason that we display the full sequence accuracy.



MabCampath Protein Scaffold	
CDA Full Acc: 95.327%	DIQMTQSPSSLSASVGDRVTITCRASQDIDNYLNWYQQKPGKAPKLLIYDASNL QTGVPSRFSGSGSGTDFTFTISSLQPEDIATYYCLQHYNYPYTFGQGTKVEIKR TVAAPSVFIFPPSDEQLKSGTASVVCLLNNFYPREAKVQWKVDNALQSGNSQES VTEQDSKDSTYSLSSTLTLSKADYEKHKVYACEVTHQGLSSPVTKSFNRGEC
CNN-LSTM Full Acc: 89.7%	DIQMTQSPSSISASVGDRVTITCKASQNIDKYINWYQQKPGKAPKIIIYNTNNI QTGVPSRFSGSGSGTDFTFTIGSLQPEDFATYYCIQHISRPRTFGQGTKVEIKR SIAAPSVFIFPPSDEQIKSGTASVVCIINNFYPREAQPRRKVDNAIQSGNSQES VTEQDSKDSTYSISSTITISKADYEKHKVYACEVTHQGISSPVTKSFNRGEC

Fig. 4. MabCampath Protein Scaffold Predictions from CDA and CNN-LSTM Models

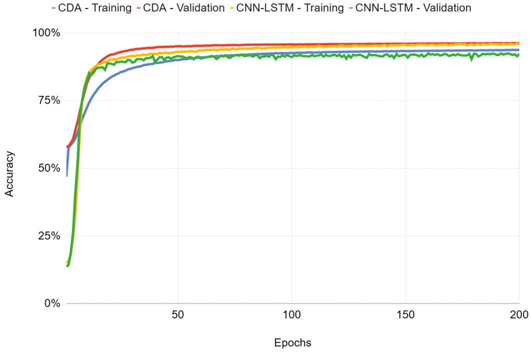


Fig. 5. CDA and CNN-LSTM Training and Validation Accuracies

5.2 Results on Simulation Datasets

To further demonstrate the performance of our proposed CDA and CNN-LSTM [7] model, we test both models on the generated scaffolds as described in Sect. 4.3. The CDA outperforms the sequence-based CNN-LSTM approach on 10 out of the 11 datasets in terms of full sequence accuracy. The chart in Fig. 7 compares the full-sequence accuracies. Our proposed CDA model has a better prediction accuracy for full sequence comparison with the target sequence. The main reason is that CDA is able to predict the missing amino acids in the gaps, also it can fix the errors in the non-gaps regions of the constructed scaffold. While CNN-LSTM model does not have such capability. It only focus on predicting the missing amino acids in the gaps of the scaffold. The CNN-LSTM model approach cannot in principle correct non-gap errors, so the non-gap accuracy is always 0%. The CDA model, on the other hand, suffers from the deficiency that since it outputs a full sequence to be used for its full prediction, it may inadvertently change amino acids that should not be changed. In fact, on the one generated scaffolds (scaffold #9), the CNN-LSTM model achieves higher full sequence accuracy. The reason the lower full sequence accuracy of CDA is merely that the CDA changes too many amino acids that should have remained the same.

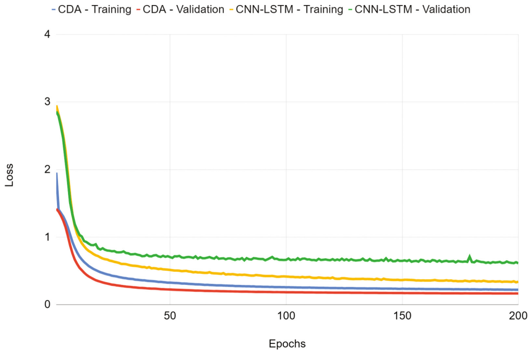


Fig. 6. CDA and CNN-LSTM Training and Validation Loss

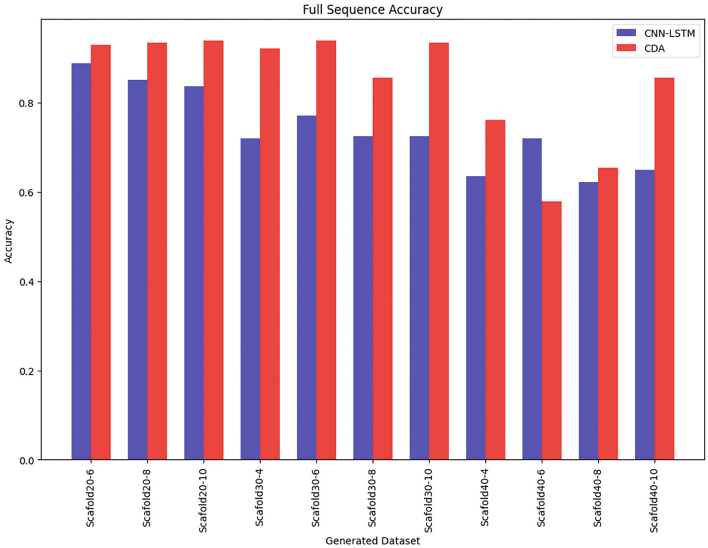


Fig. 7. A Comparison Result Between CDA and CNN-LSTM on Simulation Datasets

6 Conclusion

De novo protein sequencing from mass spectrometry data is still a hard problem in proteomics. Current state-of-the-art approaches are still unable to completely sequence proteins accurately. In this paper, we show that we can apply deep learning methods to aid in a final step in de novo protein sequencing, namely filling gaps in the protein scaffold. Moreover, we have shown that our CDA model is able to perform this task more accurately than the sequence-based approach [7], which also outperforms the existing combinatorial algorithms based on dynamic programming, local search and greedy methods described in [4]. The advantage of this approach is that it is far simpler once the model is built to perform the inference needed to fill the gaps. This simplicity avoids the potential deficiency we identified with the sequence-based approach that predicts one amino acid after another. We conclude that if the constructed scaffold with higher accuracy and smaller gaps, the deep learning based approaches can produce more higher accuracy on protein sequencing predictions. For the future work, we will test our model on the more real protein scaffold dataset and explore other machine learning models for the protein sequencing problem.

Acknowledgement. This work is supported by the National Science Foundation of the United States under Award 2307571, 2307572 and 2307573. We thank anonymous reviewers for their insightful comments and inputs.

References

1. Bengio, Y.: Deep learning of representations for unsupervised and transfer learning. In: Proceedings Of ICML Workshop On Unsupervised And Transfer Learning, pp. 17–36 (2012)
2. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.: Extracting and composing robust features with denoising autoencoders. In: Proceedings Of The 25th International Conference On Machine Learning, pp. 1096–1103 (2008)
3. Vincent Pascalvincent, P., Larocheh, L., Autoencoders, H.: Learning useful representations in a deep network with a local denoising criterion pierre-antoine manzagol. J. Mach. Learn Res. **11**, pp. 3371–3408 (2010)
4. Qingge, L., Liu, X., Zhong, F., Zhu, B.: Filling a protein scaffold with a reference. IEEE Trans. Nanobiosci. **16**, 123–130 (2017)
5. Liu, X., et al.: De novo protein sequencing by combining top-down and bottom-up tandem mass spectra. J. Proteome Res. **13**, 3241–3248 (2014)
6. National Center for Biotechnology Information Blast. https://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE=Proteins
7. Sturtz, J., Zhu, B., Liu, X., Fu, X., Yuan, X., Qingge, L.: Deep learning approaches to the protein scaffold filling problem. In: 2022 IEEE 34th International Conference On Tools With Artificial Intelligence (ICTAI), pp. 1055–1061 (2022)
8. Ramazi, S., Allahverdi, A., Zahiri, J.: Evaluation of post-translational modifications in histone proteins: a review on histone modification defects in developmental and neurological disorders. J. Biosci. **45**(1), 1–29 (2020). https://doi.org/10.1007/s12038-020-00099-2
9. Ramazi, S., Zahiri, J.: Post-translational modifications in proteins: resources, tools and prediction methods. Database (2021)
10. Eng, J.K., McCormack, A.L., Yates, J.R.: An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. J. Am. Soc. Mass Spectrom. **5**(11), 976–989 (1994). https://doi.org/10.1016/1044-0305(94)80016-2
11. Smith, L., Kelleher, N.: Proteoform: a single term describing protein complexity. Nat. Methods **10**, 186–187 (2013)

12. Alhaider, A., et al.: Through the eye of an electrospray needle: mass spectrometric identification of the major peptides and proteins in the milk of the one-humped camel (Camelus dromedarius). J. Mass Spectrom. **48**, 779–794 (2013)
13. Viala, V., et al.: Pseudechis guttatus venom proteome: insights into evolution and toxin clustering. J. Proteomics **110**, 32–44 (2014)
14. Costa, D., et al.: Sequencing and quantifying IgG fragments and antigen-binding regions by mass spectrometry. J. Proteome Res. **9**, 2937–2945 (2010)