Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)
Special Track on AI for Computational Sustainability and Human Well-being

Improving Tandem Mass Spectra Analysis with Hierarchical Learning

Zhengcong Fei

Key Lab of Intell. Info. Process., Inst. of Comput. Tech., CAS, Beijing, China
University of Chinese Academy of Sciences, Beijing, China
feizhengcong@ict.ac.cn

Abstract

Tandem mass spectrometry is the most widely used technology to identify proteins in a complex biological sample, which produces a large number of spectra representative of protein subsequences named *peptide*. In this paper, we propose a hierarchical multi-stage framework, referred as DeepTag, to identify the peptide sequence for each given spectrum. Compared with the traditional one-stage generation, our sequencing model starts the inference with a selected high-confidence guiding tag and provides the complete sequence based on this guiding tag. Besides, we introduce a cross-modality refining module to asist the decoder focus on effective peaks and fine-tune with a reinforcement learning technique. Experiments on different public datasets demonstrate that our method achieves a new state-of-the-art performance in peptide identification task, leading to a marked improvement in terms of both precision and recall.

1 Introduction

Identifying proteins in complex biological samples is an elementary task in medicine and biology, such as analysis of components in the blood. *Tandem mass spectrometry* (MS/MS) is widely employed to accomplish this task. At a general MS/MS experiment, a collection of spectra is created in a particular order, each of which is representative of a protein subsequence called a *peptide*. The pair consisting of a matched peptide sequence and spectrum is considered as a *peptide-spectrum match* (PSM). Accurately identifying the peptide sequences responsible for each experimental spectrum becomes a challenging task in Proteomics.

Recent advances in deep learning technology have substantially improved the performance of peptide identification. The essential practice of neural peptide sequencing models follow encoder-decoder paradigm [Tran *et al.*, 2017; Tran *et al.*, 2019; Qiao *et al.*, 2019]. In between, convolutional neural network (CNN) is utilized to encode an input spectrum and recurrent neural network (RNN) is adopted as a sequence decoder to produce the entire peptide sequence, one *amino acid* at each time step. Most of these peptide sequencing approaches are trained by maximizing the likelihood of each ground-truth amino acid based on previously generated amino acids and the spectrum with backpropagation.

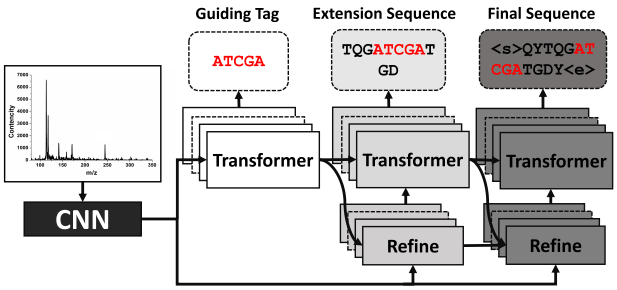

Figure 1: Illustration of our proposed hierarchical sequencing framework. The model consists of one spectrum encoder (CNN) and a sequence of peptide sequence decoders (Transformer plus Refine), and it takes the experimental spectra as input and expands the high-confidence guiding tag layer by layer.

However, there exist two major problems in these peptide sequence identification methods: (1) It is unreliable for the sequence decoders to generate amino acid one by one as a peptide sequence from left to right. Since the fragments of the peptide are more likely to occur in the middle position in a biological experiment, which results in a lower abundance of effective *signal peaks* on both sides of the spectrum and easy to be concealed by *noise peaks* [Li *et al.*, 2005]. On the other hand, the peptide sequencing process is accomplished by a greedy search or with a beam search, which predicts the next amino acid according to local maximum occurrence probability. There is a case that some total matched peptide sequences may be filtered at early steps which are with low probability to be recognized in similar noise peaks by using the local maximum probability network alone. Such a top ranking-based approach assumes that the log-probability of every amino acid in a match sequence must be among top choices. Actually, this is not necessarily true. (2) Despite involving two different modalities in peptide sequencing, former approaches seldom explore the interactions between CNN and RNN structure. A common method is directly feeding the spectrum feature from CNN into the RNN as the initial node [Tran *et al.*, 2017]. However, such a naive method treats features in the experimental spectrum the same

and ignores the major influence of local features when generating each amino acid. Uncorrelated global spectrum information may interfere with current peptide sequence generation and severely limits the capacity of complex reasoning.

Considering the great challenge of generating peptide sequences from left to right in one stage, we propose a novel hierarchical multi-stage framework, namely DeepTag, that generates peptide sequences with the help of *guiding tag*s [Tabb *et al.*, 2003], which refer to high-score peptide subsequences, will be expanded in later decoders. Technically, our model consists of a spectrum encoder and a sequence of Transformer-based peptide decoders that gradually expand peptide sequences in two sides. To refine the information of decoder at the input time, we adopt a refining module, which can fuse historical subsequence information and spectrum information to conduct the sequence generation. Different from [Tran *et al.*, 2017], our model utilize cross-modality refining mechanism [Vaswani *et al.*, 2017] to weight spectrum features based on the preceding output of Transformer decoder layer. Furthermore, inspired by the recent work [Zhang *et al.*, 2017], we design a similar RL-based training method, but extend it from one-stage to our multi-stage framework, where rewards are incorporated at each stage as intermediate supervision. Figure 1 illustrates our proposed hierarchical framework, which consists of three peptide sequence decoder networks. Specifically, the first Transformer decoder generates the guiding tag for the observed spectrum, and the subsequent Transformer decoders serve as sequence extension. At each stage in our DeepTag, refined spectrum features and hidden vector produced by the preceding decoding stage are adopted as inputs to the subsequent stage.

Our contributions are summarized as follows:

- We introduce a hierarchical multi-stage framework to improve the peptide identification from mass spectra, which iteratively expands the peptide sequences layer by layer based on high-confidence guiding tags.
- We present a new multi-layer cross-modality refining mechanism to refine the spectrum features before feeding to the sequence decoders, which significantly boosts the performance of peptide sequencing.
- We incorporate a modified RL-based fine-tune technique that can optimize the multi-stage model with the normalized intermediate rewards.

2 Background

Tandem mass spectrometry is the key technology for mixed sample detection and biology pathological research. It has numerous successful applications in medical, biology, and pharmacy [Craig and Beavis, 2004]. In practical MS/MS experiments, the digesting enzyme is first employed to cleave proteins into peptides. Next, the generated peptides are successively imported to the mass spectrometry in two rounds. In the first round, the mass and charge of enact peptide, called *precursor mass* and *precursor charge*, are measured. In the second round, the peptides are further fragmented and a sequence of mass spectra are provided, which corresponds to a small part of peptide [Li *et al.*, 2005]. Specifically, each output mass spectrum is a set of $(\frac{m}{z}, intensity)$ pairs where $\frac{m}{z}$ denotes the mass-to-charge ratio and $intensity$ value is the abundance of these ions.

In this study, we aim to reconstruct the amino acid sequence of a peptide on the basis of the given spectrum. Nevertheless, the major challenge lies that: (1) a large number of candidate combinations lead to high computational complexity. Since there will be exponential situations to be considered as the candidate prediction in an extreme case. (2) peptide fragmentation produces multiple types of ions that hold quite different intensity values. The division rule is critical but remains understudied [Tran *et al.*, 2017]. (3) there are plenty of noise peaks blending with the real ions that limits the determination of correct sequences.

To cover the above issues, early pioneering sequencing models integrate graph algorithms and dynamic programming to reduce the task complexity [Dasari *et al.*, 2010]. However, these methods set strong assumptions, and the accuracy of the identification results is restricted. Later on, deep learning technology was introduced to peptide sequencing [Tran *et al.*, 2017; Zhou *et al.*, 2017; Tran *et al.*, 2019]. These methods exploit the encoder-decoder paradigm that firstly utilizes CNN to encoder spectrum and then adopt an RNN-based decoder to generate the output sequence, leading to promising results for this task. Nevertheless, during inference, most sequence models employ a common decoder mechanism using a greedy or beam search. That is, they always predict the next amino acids with top local score and produce the total peptide in one stage. Notably, such a mechanism can miss correct peptides at early steps. In contrast, our hierarchical learning framework incorporates high-confidence guiding tags and a multi-stage sequence extension process to compensate such errors and effectively reduce the search space.

3 Methods

3.1 Task Formulation

Peptide sequencing is the task of automatically producing a peptide sequence $\hat{\mathbf{P}}= \{\hat{a}_1, \hat{a}_2, \dots \hat{a}_T\}$ to identify a given spectrum $\mathbf{S}$, where $\hat{a}_t \in \mathcal{A}$ is the predicted amino acid letter, $\mathcal{A}$ is the space including 20 candidate amino acids, <s> and <e> symbol, and T denotes the peptide sequence length.

Our model builds a hierarchical framework with the same target as those one-stage models, but with the additional intermediate layers between the output layer and the input layer. Specifically, we first train the model via maximizing the log-likelihood of each guiding tag conditioned on the input spectrum and the target peptide sequence $\mathbf{P} = \{a_1, a_2 \dots a_T\}$, and then optimize the model with peptide-level evaluation metrics. We denote the predicted peptide subsequence of the i^{th} extensive stage decoder as $\hat{\mathbf{P}}^i$, $i \in \{1, \dots, N\}$, and N is the number of stages. Notably, we treat stage $i = 1$ as the guiding tag decoder and $\hat{\mathbf{P}}^1 = \{\hat{a}_i, \dots, \hat{a}_{i+k-1}\}$ represents the guiding tag of spectrum $\mathbf{S}$, where k is the length of tags and $1 \leq k \leq T$. As a result, each intermediate sequence decoder provides the increasingly expanding peptide subsequence, and the prediction of the last decoder is taken as the final peptide sequence.

3.2 Spectrum Encoding

Following [Qiao *et al.*, 2019], we select top n=500 most intense peaks and represent the spectrum as a tuple set $\{(\frac{m}{z}, intensity)\}_1^n$ to tackle the accuracy-speed / memory trade off problem. Integrated with spectrum feature extraction matrix, the experimental spectrum $\mathbf{S}$ is encoded to the spatial spectrum features $\mathbf{V}$ as,

$$\mathbf{V} = \text{CNN}(\mathbf{S}), \tag{1}$$

Practically, we employ pre-trained T-Net [Qiao *et al.*, 2019] as our spectrum information encoder, which is designed for the kind of order invariant data.

3.3 Hierarchical Decoding

The overall hierarchical decoding framework consists of one guiding tag decoder and a sequence of Transformer-based extensive decoders that repeatedly extend the prediction of a guiding tag from the preceding decoder. The first stage of our model decoding is a guiding tag decoder which generates high-confidence guiding tags based on the global spectrum features. In the subsequent stages, each stage $i \in \{2, \ldots, N\}$ is a sequence expansion decoder which extends the peptide subsequence based on spectrum features and the outputs of the previous stage. More formally, our proposed hierarchical decoder handle the final peptide sequence as:

$$p(\hat{\mathbf{P}}^N|\mathbf{V}) = p(\hat{\mathbf{P}}^1|\mathbf{V}) \prod_{i=2}^{N} p(\hat{\mathbf{P}}^i|\hat{\mathbf{P}}^{i-1}, \mathbf{V}). \tag{2}$$

In addition, we consider the hidden states of the preceding stage to provide the following stage weighted spectrum regions for better peptide sequence prediction. In the following, we will introduce the adopted guiding tag decoder, tag extension decoder, and refining module in detail.

Guiding Tag Decoder

We start by decoding in a guiding tag search in the first stage ($i = 1$), where we learn a guiding tag decoder with an Transformer network, named TM$_{tag}$. At each time step $t \in [1, k]$, the input to TM$_{tag}$ consists of the previous output sequence $\hat{\mathbf{P}}^1_{<t} = \{\hat{a}_i, \ldots, \hat{a}_{i+t-1}\}$ and the intial spectrum features $\mathbf{V}$. Hence, we get the condition probability:

$$p(\hat{\mathbf{P}}^1|\mathbf{V}) = \prod_{t=1}^{k} p(a_t^1|\hat{\mathbf{P}}^1_{<t}, \mathbf{V}). \tag{3}$$

In practical terms, instead of applying RNN or LSTM, we introduce the basic Transformer model to the peptide sequence generation in this task. In our preliminary experiments, we have found that the Transformer can achieve better performance in this task, and thus we apply this model to the sequence decoder.

Briefly, the Transformer decoder consists of an embedding layer and multiple decoder layers. Each decoder layer has a mask self-attention module and a point-wise feed-foward network (FFN). The detail tag decoder flow is described as:

$$\text{TM}_{tag}(\hat{\mathbf{P}}^1_{<t}, \mathbf{V}) = \text{FFN}(\text{ATT}_{crs}(\mathbf{V}, \text{ATT}_{self}(\hat{\mathbf{P}}^1_{<t}))), \tag{4}$$

where ATT$_{crs}(\cdot)$ and ATT$_{self}(\cdot)$ denote cross-attention and self-attention layer respectively.

Sequence Extension Decoder

In the subsequent stages, each i^{th} sequence extension decoder predicts the t^{th} remaining amino acid $\hat{a}_t^i$ based on the spectrum featrues $\mathbf{V}$, the refining weights α_t^{i-1} and the previously generated peptide sequence $\hat{\mathbf{P}}_t^{i-1}$ from the preceding decoder. The probability of sequence can be calculated as:

$$p(\hat{\mathbf{P}}^i|\hat{\mathbf{P}}^{i-1}, \mathbf{V}) = \prod_{t=1}^{N} p(a_t^i|\hat{\mathbf{P}}^i_{<t}, \hat{\mathbf{P}}^{i-1}, \mathbf{V}). \tag{5}$$

In fact, each sequence extension decoder consists of an TM$_{ex}$ network and a refining module. At each time step t, the input to TM$_{ex}$ incudes the refined spectrum feature x^{i-1}, the previous output amino acid a_{t-1}, and the output of the preceding TM$_{ex}$. Therefore, the updating procedure of TM$_{ex}$ can be written as:

$$\text{TM}_{ex}(\hat{\mathbf{P}}^i_{<t}, x^i) = \text{FFN}(\text{ATT}_{crs}(x^i, \text{ATT}_{self}(\hat{\mathbf{P}}^i_{<t}))), \tag{6}$$

$$x_t^i = [g(\mathbf{V}, \alpha^{i-1}, h^{i-1}); \hat{\mathbf{P}}_t^{i-1}], \tag{7}$$

where h^{i-1} is the hidden state from preceding Transformer and $g(\cdot)$ is the spatial attention function which feeds refined spectrum features as additional inputs to TM$_{ex}$ at each time step to emphasise the effective peak information and ignore the noise peaks. When i=2, it represents the information transmits from the guiding tag decoder to the sequence extension decoder; when i>2, it represents the increasely improved process of the sequence extension decoder itself.

Refining Module

As mentioned above, global spectrum features is adopted for our sequence decoder to generate the amino acids. However, in most real cases, each amino acid is only related to a small region of a spectrum. Directly incorporating the whole spectrum feature for amino acid prediction can lead to sub-optimal identification results due to the noises integrated from the irrelevant regions [Vaswani *et al.*, 2017].

Towards that end, the attention mechanism has been proposed to effectively improve the performance of peptide identification [Xu *et al.*, 2015]. It typically produces a spatial feature map attending spectrum peaks regions relevant to each predicted amino acid. In this work, to extract more available spectrum information for each amino acid prediction, we adopt a Transformered-based cross-modality module to filter out noises gradually and pinpoint the effective peaks that are highly relevant to the current sequence prediction [Anderson *et al.*, 2018]. In each sequence extension stage i, our refining model operates on both spectrum features $\mathbf{V}$ and importance weights α_t^{i-1} from the preceding stage.

Formally, for the time step t of stage i, the work process of our refining model can be defined as:

$$g(\mathbf{V}, \alpha_t^{i-1}, h_t^{i-1}) = \alpha_t^{i-1} \cdot (W^i \cdot \mathbf{V} + b^i) \tag{8}$$

$$\alpha_t^{i-1} = \text{TM}_{ref}(\mathbf{V}, h_t^{i-1}, \hat{\mathbf{P}}^i_{<t}) \tag{9}$$

where TM$_{ref}$ denotes Transformered-based refining layer and please note that when $i = 1$, we set α_t^1 to zero.

Data set	Lab	Instrument	Species	#Spectra	Publication
Mann-Human-QE	Mann	Q Exactive	Human	27,570	[Michalski *et al.*, 2011]
Mann-Mouse-QEHF	Mann	Q Exactive HF	Mouse	172,000	[Sharma *et al.*, 2015]
Gygi-Human-QE	Gygi	Q Exactive	Human	176,000	[Chick *et al.*, 2015]
Dong-Ecoli-QE	Dong	Q Exactive	Escherichia coli	15,000	[Liu *et al.*, 2014]
Xu-Yeast-QEHF	Xu	Q Exactive HF	Yeast	243,000	[Chi *et al.*, 2018]

Table 1: Basic dataset information.

3.4 Training Procedure

The introduced hierarchical framework results in a deep architecture. Correspondingly, it tends to cause the vanishing gradient problem during training, where the magnitude of gradients decreases dramatically when backpropagated through multiple intermediate sequencing layers [Anderson *et al.*, 2018]. An effective approach to tackle this problem is to integrate supervised training objectives for the intermediate sequencing layers. Each stage of the hierarchical subsequence decoder is trained to predict the peptide sequence repeatedly. Here, we first introduce a cross-entropy (XE) loss to optimize the network parameters, *i.e.*,

$$\mathcal{L}^i_{XE}(\theta_{1:i}) = -\sum_{t=1}^{T} log(p_{\theta_{1:i}}(a_t|\mathbf{P}_{<t}, \mathbf{S})), \tag{10}$$

where a_t denotes the ground-truth amino acid letter, and $\theta_{1:i}$ is the parameters up to the stage-i peptide sequence decoder. We thus acquire the overall training objective for the full architecture via adding up the losses at each stage i:

$$\begin{aligned}\mathcal{L}_{XE}(\theta) &= -\sum_{i=1}^{N} \mathcal{L}^i_{XE}(\theta_{1:i}) \\ &= -\sum_{i=1}^{N}\sum_{t=1}^{T} log(p_{\theta_{1:i}}(a_t|\mathbf{P}_{<t}, \mathbf{S})),\end{aligned} \tag{11}$$

where $p_{\theta_{1:i}}(a_t|\mathbf{P}_{<t}, \mathbf{S})$ represents the confidence probability of amino acid a_t given by the i-th decoder. The weights of the models are shared across all time steps.

However, optimizing the loss function $\mathcal{L}_{XE}$ is usually not sufficient, since the current log-likelihood training objective causes the *discrepancy* problem. To be specific, the decoder is trained to focus on the correctness of predicting each amino acid separately. However, at each step in the test stage, the decoder is fed with the predicted amino acid from previous step rather than ground truth. This leads to the gap between training and test and linits the performance in test. To address this gap, we can regard the peptide sequence generation process as a reinforcement learning process. In detail, given an environment, we want to train an agent to take an action (next amino acid) conditioned on the current environment (spectrum features and previously generated amino acids). After producing a complete peptide sequence, the agent will receive a peptide-level reward and update its internal state accordingly.

Inspired by [Ren *et al.*, 2017; Gu *et al.*, 2018], we fine-tune our multi-stage sequence generation model with RL technique. The sequence decoder of each stage can be considered as an agent that consistly interacts with the environment. The policy network determines a policy $p_{\theta_{1:i}}$ which receives a state including preceding subsequence outputs, internal state and spectrum features, and produce an action $\tilde{a}^i_t$ corresponds to the next amino acid at time step t. Once we get the complete predicted peptide sequence $\tilde{\mathbf{P}}^i$, the agent will acquire a reward $r(\tilde{\mathbf{P}}^i)$. The goal of RL-based training is to minimize the expected punishments of multi-stage decoding as,

$$\mathcal{L}_{RL}(\theta) = -\sum_{i=1}^{N} \mathbb{E}_{\tilde{\mathbf{P}}^i \sim p_{\theta_{1:i}}}[r(\tilde{\mathbf{P}}^i)] \approx -\sum_{i=1}^{N} r(\tilde{\mathbf{P}}^i), \tag{12}$$

where $\tilde{\mathbf{P}}^i = \{\tilde{a}^i_1, \dots \tilde{a}^i_T\}$ and $\tilde{a}^i_t$ is sampled from the stage i at time step t. $r(\tilde{\mathbf{P}}^i)$ is calculated by comparing the generated peptide sequence to the corresponding ground-truth sequence. On this basis, we can compute the expected gradient with the *Monte-Carlo* to gain $\tilde{\mathbf{P}}^i$ from $p_{\theta_{1:i}}$ as:

$$\begin{aligned}\nabla_\theta \mathcal{L}_{RL}(\theta) &= \sum_{i=1}^{N} \nabla_{\theta_{1:i}} \mathcal{L}(\theta_{1:i}) \\ &\approx -\sum_{i=1}^{N} r(\tilde{\mathbf{P}}^i) \cdot \nabla_{\theta_{1:i}} log p_{\theta_{1:i}}(\tilde{\mathbf{P}}^i).\end{aligned} \tag{13}$$

4 Experiments

4.1 Experimental Preparation

Datasets. All the experiments are conducted on five public data sets from different labs and species. Table 1 presents the basic information of the data sets where "#Spectra" represents the number of spectra used in training or testing. The forms of these data were all high-resolution HCD. Open-pFind [Chi *et al.*, 2018] was employed to deal with these raw data sets and the five data sets were searched against the corresponding reviewed database of human, mouse, *E.coli*, and yeast, respectively, which were all downloaded from Uniprot and their versions are consistent with [Chi *et al.*, 2018]. To be specific, we configure the precursor ion tolerance as 20 ppm and the fragment ion tolerance as $\pm$20 ppm. We also controlled the FDR at 1% at the spectrum level. Furthermore, in order to maintain the matching quality of all spectra in these data sets, we removed the PSM whose matched peak number was less than its peptide length as well as the spectra with peptide length longer than 20. Finally, $\sim$ 920,000 high-quality PSMs were acquired for later experiments. Importantly, the peptide sequences identified from Open-pFind were assigned to the corresponding MS/MS spectra and then used as final ground truth for testing the correctness of sequencing results.

Dataset	Mann-Mouse-QEHF			Gygi-Human-QE			Dong-Ecoli-QE			Xu-Yeast-QEHF		
Metric	AAR	AAP	PR	AAR	AAP	PR	AAR	AAP	PR	AAR	AAP	PR
PEAKS	0.342	0.483	0.145	0.365	0.421	0.152	0.425	0.462	0.178	0.382	0.453	0.155
Novor	0.371	0.502	0.152	0.383	0.412	0.187	0.461	0.501	0.218	0.401	0.512	0.171
DeepNovo	0.427	0.512	0.241	0.454	0.428	0.251	0.513	0.521	0.321	0.466	0.561	0.253
DeepNovoV2	0.467	0.532	0.266	0.484	0.448	0.281	0.533	0.538	0.345	0.482	0.583	0.262
DeepTag	**0.492**	**0.568**	**0.289**	**0.515**	**0.486**	**0.307**	**0.581**	**0.580**	**0.382**	**0.512**	**0.605**	**0.271**

Table 2: Total recall and precision of PEAKS, Novor, DeepNovo, DeepNovoV2 and our multi-stage DeepTag on different matched data sets. AAR represents amino acid recall, AAP represents amino acid precision and PR represents peptide recall.

Compared approaches. We compared the following state-of-the-art peptide sequencing methods: PEAKS [Ma *et al.*, 2003], Novor [Jeong *et al.*, 2013] DeepNovo [Tran *et al.*, 2017] and DeepNovoV2 [Qiao *et al.*, 2019]. Note that the first two methods are constructed with a traditional search strategy, while the last two methods are deep learning-based.

Evaluation metrics. In this paper, the generated amino acid can be regarded as correct when the mass difference between the predicted amino acid and a ground-truth amino acid is less than 0.1 Da, and the prefix mass before them as well as the suffix mass behind them are different by less than 0.5 Da. Following [Qiao *et al.*, 2019], we adopt three types of metrics: *precision*, *recall* and *area under curve* (AUC) to evaluate the performance of peptide sequencing. More subdivided, the ratio of the total number of matched amino acids over the total number of amino acids in the generated peptide sequences is considered as *amino acid level precision* while the fraction of true peptide sequences in total predicted peptide sequences is served as *peptide level precision*. Similar definitions can be applied to recall and AUC as well.

Implementation details. The structure of our DeepTag model was presented in Figure 1. We utilized T-Net [Qiao *et al.*, 2019] as our spectrum feature extractor. The final output of T-Net was resized to 256 dimensions. The length of the guiding tag was set to 5 ($k = 5$) and the total number of the stage was set to 3 ($N = 3$). For guiding tag decoder, we use the small transformer ($d_{model} = 256, d_{hidden} = 256, p_{dropout} = 0.1, n_{layer} = 3$, and $n_{head} = 2$). For extended decoder, we use the base transformer by [Vaswani *et al.*, 2017]($d_{model} = 512, d_{hidden} = 512, p_{dropout} = 0.1, n_{layer} = 3$ and $n_{head} = 8$). We first train our model under the cross-entropy cost using Adam optimizer (lr=0.002) and a momentum parameter of 0.9. Later on, we adopt the proposed RL-based approach on the just trained sequencing model to further optimize. During this stage, we use Adam with a learning rate of 0.0002. After each epoch, we evaluate the model performance on the validation set and choose the sequencing model with the best rewarding. Overall, our model was first trained on the Mann-Human-QE data set, and then tested on Mann-Mouse-QEHF for cross-species validation and Gygi-Human-QE data set for cross-lab validation. The rest of the data sets were adopted to test the robustness of our model. Please note that the training dataset and testing dataset come from different species [Zhou *et al.*, 2017]. The cross-validation is used to guarantee unbiased training and testing and does not give our model any advantage.

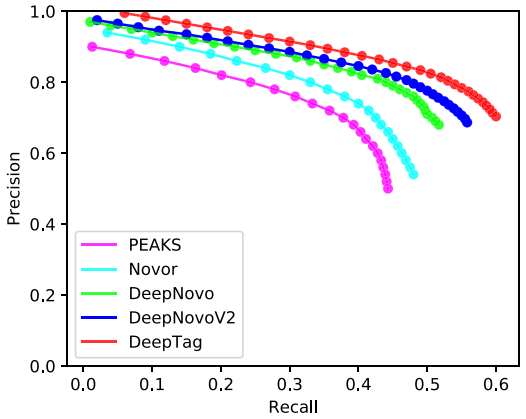

Figure 2: The precision-recall curves of PEAKS, Novor, DeepNovo, DeepNovoV2, and our DeepTag on Mann-Mouse-QEHF dataset.

4.2 Comparing with State-of-the-art Methods

Table 2 summaries the performance comparisons between the state-of-the-art peptide sequencing models and our proposed DeepTag on different species and labs datasets. In general, our DeepTag consistently exhibits better performance than other sequencing models, which include the traditional search methods (PEAKS, Novor) and deep learning-based methods (DeepNovo and DeepNovoV2). The PR score of our DeepTag can achieve 0.289 on the Mann-Mouse-QEHF dataset, which is to-date the best performance and makes the absolute improvement over the best competitor DeepNovoV2 by 2.3%. The performance improvements generally demonstrate the key advantage of incorporating multi-stage decoding for peptide sequencing. In particular, we can observe from the two types in Table 2 that deep learning-based approaches outperforms top human-designed search approaches $\sim$ 5.0 % in AAR, $\sim$ 2.1 % in AAP and $\sim$ 6.5% in PR. In our opinion, compared with the traditional search methods, the model based on deep learning has greater advantages in extracting spectral information and results in accurate peptide sequence.

On the other hand, we should also be aware that all sequencing tools report confidence scores for their predictions, and setting a higher threshold of confidence score will lead to a smaller part of peptides with high precision but will make

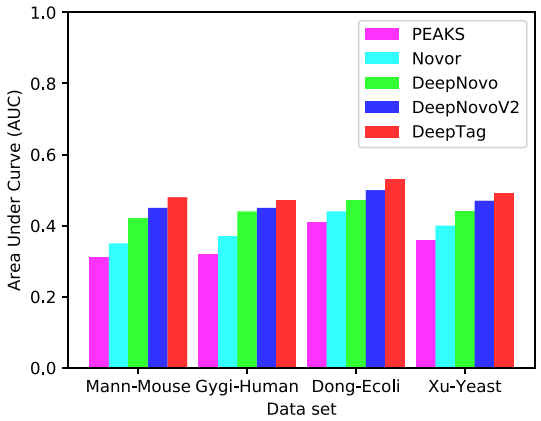

Figure 3: The area under curve of PEAKS, Novor, DeepNovo, DeepNovoV2, and our hierarchical DeepTag on different species and labs data sets.

the rest of the dataset without results [Tran *et al.*, 2017]. Hence, it is reasonable to depict precision-recall curves and incorporate the area under curve (AUC) as metrics of peptide sequencing quality. Figure 3 and 2 display the AUC of different peptide sequencing methods on different data sets and the precision-recall curves on Mann-Mouse-QEHF dataset respectively. We can view that our multi-stage DeepTag model still maintains superiority against other sequencing methods. For example, for Mann-Mouse-QEHF dataset, the AUC of our DeepTag was 54.8% higher than that of PEAKS ((0.48-0.31) / 0.31 = 0.548) and 37.1% higher than that of Novor ((0.48-0.35) / 0.35 = 0.371). In addition, DeepNovoV2 and our method often came in the first two places, probably because of the improvement of deep learning technology in spectrum information encoder (both models adopts T-Net to handle spectra). In summary, the sequence generation method based on deep learning generally present better performance compared with traditional search-based strategy. More importantly, extensive experimental results demonstrated the improvement of our method was efficient and reliable.

4.3 Neural Network Architecture Analysis

In this paper, we propose a novel hierarchical framework whose architecture is worth further looking into. A natural question arises "why do we choose these structures here?". To answer this question, we construct an *ablation study* on four variants based on our neural network structure: (1) we implement a one layer Transformer-based guiding tag decoder and one layer Transformer-based sequence extension decoder model named as $\mathbf{TM}_{1+1\ layers}$. (2) We add two additional Transformer networks after one layer Transformer sequence extension decoder model, which is named as $\mathbf{TM}_{1+3\ layers}$. If the number of layers is not indicated, then N = 3 (1+2) in default. Then, we implement two types of refined-based peptide sequencing models: (1) the output of spectrum features are directly input to the hierarchical decoder, which is named as **TM+no Refine**. (2) the soft-refined model (**TM+Refine**$_{Soft}$) proposed by [Xu *et al.*, 2015].

Methods	AAR	AAP	PR	AUC
$\text{TM}_{1+1\ layers}$	0.431	0.512	0.228	0.42
$\text{TM}_{1+3\ layers}$	0.424	0.503	0.224	0.40
TM+no Refine	0.406	0.498	0.216	0.38
TM+Refine$_{Soft}$	0.450	0.525	0.246	0.43
DeepTag	**0.473**	**0.537**	**0.255**	**0.45**

Table 3: Performance comparisons on Mann-Mouse-QEHF for different metrics optimized by cross-entropy loss.

Methods	AAR	AAP	PR	AUC
$\text{TM}_{1+1\ layers}$	0.435	0.526	0.234	0.42
$\text{TM}_{1+3\ layers}$	0.428	0.511	0.230	0.41
TM+no Refine	0.410	0.503	0.221	0.39
TM+Refine$_{Soft}$	0.482	0.552	0.278	0.46
DeepTag	**0.492**	**0.568**	**0.289**	**0.48**

Table 4: Performance comparisons on Mann-Mouse-QEHF for different metrics optimized by the RL-based method later.

More concretely, in this experiment, we first train the baselines and our proposed DeepTag with a standard cross-entropy loss. We report the performance of our model and the baselines in Table 3. We can find that our hierarchical learning framework achieves the best performances in all metrics. The soft refining models, **TM+Refine**$_{Soft}$, provides slightly lower performance than our DeepTag. Note that directly adding extra one additional Transformer layer in $\mathbf{TM}_{1+3\ layers}$ decreases the performance of our DeepTag as the model may experience overfitting. Our hierarchical approach which optimizes the network gradually with the intermediate supervision can effectively avoid overfitting to some degree. We also view that the refining mechanism (**TM+Refine**$_{Soft}$ and DeepTag) can significantly improve the performance of peptide identification, compared with **TM+no Refine**. After optimizing the model with cross-entropy loss, we optimize with the RL-based algorithms to fine-tune the models. The performance of five models are presented in Table 4. Similar to the previous observations, our DeepTag still obtains significant gains across all metrics.

5 Conclusion

Peptide sequencing is a challenging problem that incorporates both pattern recognition and optimization in data analysis. In this paper, we propose a hierarchial multi-stage framework that utilized a refining module in conjunction with multiple Transformer networks to improve MS/MS analysis. Our model generates peptide sequences from high-confidence guiding tags to both sides extended sequences, which we found to be very beneficial for peptide identification. The model was compared with other prevalent peptide identification tools and experiments show that our DeepTag achieves higher precision at both amino acid and peptide levels. Besides, the effectiveness of various network architecture was investigated in detail.

References

[Anderson *et al.*, 2018] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018.

[Chi *et al.*, 2018] Hao Chi, Chao Liu, Hao Yang, Wen-Feng Zeng, Long Wu, Wen-Jing Zhou, Rui-Min Wang, Xiu-Nan Niu, Yue-He Ding, Yao Zhang, et al. Comprehensive identification of peptides in tandem mass spectra using an efficient open search engine. *Nature biotechnology*, 36(11):1059–1061, 2018.

[Chick *et al.*, 2015] Joel M Chick, Deepak Kolippakkam, David P Nusinow, Bo Zhai, Ramin Rad, Edward L Huttlin, and Steven P Gygi. A mass-tolerant database search identifies a large proportion of unassigned spectra in shotgun proteomics as modified peptides. *Nature biotechnology*, 33(7):743–749, 2015.

[Craig and Beavis, 2004] Robertson Craig and Ronald C Beavis. Tandem: matching proteins with tandem mass spectra. *Bioinformatics*, 20(9):1466–1467, 2004.

[Dasari *et al.*, 2010] Surendra Dasari, Matthew C Chambers, Robbert J Slebos, Lisa J Zimmerman, Amy-Joan L Ham, and David L Tabb. Tagrecon: high-throughput mutation identification through sequence tagging. *Journal of proteome research*, 9(4):1716–1726, 2010.

[Gu *et al.*, 2018] Jiuxiang Gu, Jianfei Cai, Gang Wang, and Tsuhan Chen. Stack-captioning: Coarse-to-fine learning for image captioning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, pages 6837–6844, 2018.

[Jeong *et al.*, 2013] Kyowon Jeong, Sangtae Kim, and Pavel A Pevzner. Uninovo: a universal tool for de novo peptide sequencing. *Bioinformatics*, 29(16):1953–1962, 2013.

[Li *et al.*, 2005] Dequan Li, Yan Fu, Ruixiang Sun, Charles X Ling, Yonggang Wei, Hu Zhou, Rong Zeng, Qiang Yang, Simin He, and Wen Gao. pfind: a novel database-searching software system for automated peptide and protein identification via tandem mass spectrometry. *Bioinformatics*, 21(13):3049–3050, 2005.

[Liu *et al.*, 2014] Chao Liu, Chun-Qing Song, Zuo-Fei Yuan, Yan Fu, Hao Chi, Le-Heng Wang, Sheng-Bo Fan, Kun Zhang, Wen-Feng Zeng, Si-Min He, et al. pquant improves quantitation by keeping out interfering signals and evaluating the accuracy of calculated ratios. *Analytical chemistry*, 86(11):5286–5294, 2014.

[Ma *et al.*, 2003] Bin Ma, Kaizhong Zhang, Christopher Hendrie, Chengzhi Liang, Ming Li, Amanda Doherty-Kirby, and Gilles Lajoie. Peaks: powerful software for peptide de novo sequencing by tandem mass spectrometry. *Rapid communications in mass spectrometry*, 17(20):2337–2342, 2003.

[Michalski *et al.*, 2011] Annette Michalski, Eugen Damoc, Jan-Peter Hauschild, Oliver Lange, Andreas Wieghaus, Alexander Makarov, Nagarjuna Nagaraj, Juergen Cox, Matthias Mann, and Stevan Horning. Mass spectrometry-based proteomics using q exactive, a high-performance benchtop quadrupole orbitrap mass spectrometer. *Molecular & Cellular Proteomics*, 10(9):111–120, 2011.

[Qiao *et al.*, 2019] Rui Qiao, Ngoc Hieu Tran, Lei Xin, Baozhen Shan, Ming Li, and Ali Ghodsi. Deepnovov2: Better de novo peptide sequencing with deep learning. *arXiv preprint arXiv:1904.08514*, 2019.

[Ren *et al.*, 2017] Zhou Ren, Xiaoyu Wang, Ning Zhang, Xutao Lv, and Li-Jia Li. Deep reinforcement learning-based image captioning with embedding reward. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 290–298, 2017.

[Sharma *et al.*, 2015] Kirti Sharma, Sebastian Schmitt, Caroline G Bergner, Stefka Tyanova, Nirmal Kannaiyan, Natalia Manrique-Hoyos, Karina Kongi, Ludovico Cantuti, Uwe-Karsten Hanisch, Mari-Anne Philips, et al. Cell type–and brain region–resolved mouse brain proteome. *Nature neuroscience*, 18(12):1819–1830, 2015.

[Tabb *et al.*, 2003] David L Tabb, Anita Saraf, and John R Yates. Gutentag: high-throughput sequence tagging via an empirically derived fragmentation model. *Analytical chemistry*, 75(23):6415–6421, 2003.

[Tran *et al.*, 2017] Ngoc Hieu Tran, Xianglilan Zhang, Lei Xin, Baozhen Shan, and Ming Li. De novo peptide sequencing by deep learning. *Proceedings of the National Academy of Sciences*, 114(31):8247–8252, 2017.

[Tran *et al.*, 2019] Ngoc Hieu Tran, Rui Qiao, Lei Xin, Xin Chen, Chuyi Liu, Xianglilan Zhang, Baozhen Shan, Ali Ghodsi, and Ming Li. Deep learning enables de novo peptide sequencing from data-independent-acquisition mass spectrometry. *Nature methods*, 16(1):63–66, 2019.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[Xu *et al.*, 2015] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[Zhang *et al.*, 2017] Li Zhang, Flood Sung, Feng Liu, Tao Xiang, Shaogang Gong, Yongxin Yang, and Timothy M Hospedales. Actor-critic sequence training for image captioning. *arXiv preprint arXiv:1706.09601*, 2017.

[Zhou *et al.*, 2017] Xie-Xuan Zhou, Wen-Feng Zeng, Hao Chi, Chunjie Luo, Chao Liu, Jianfeng Zhan, Si-Min He, and Zhifei Zhang. pdeep: Predicting ms/ms spectra of peptides with deep learning. *Analytical chemistry*, 89(23):12690–12697, 2017.