Journal of Proteomics 259 (2022) 104542

Contents lists available at ScienceDirect

Journal of Proteomics

journal homepage: www.elsevier.com/locate/jprot

MSSort-DIA^{XMBD}: A deep learning classification tool of the peptide precursors quantified by OpenSWATH

Yiming Li [a,1], Qingzu He [a,b,1], Huan Guo [a], Chuan-Qi Zhong [c], Xiang Li [a,c], Yulin Li [a], Jiahuai Han [c,d,*], Jianwei Shuai [a,b,d,*]

[a] Department of Physics, and Fujian Provincial Key Laboratory for Soft Functional Materials Research, Xiamen University, Xiamen 361005, China
[b] Wenzhou Institute, University of Chinese Academy of Sciences, and Oujiang Laboratory (Zhejiang Lab for Regenerative Medicine, Vision and Brain Health), Wenzhou, Zhejiang 325001, China
[c] State Key Laboratory of Cellular Stress Biology, Innovation Center for Cell Signaling Network, School of Life Sciences, Xiamen University, Xiamen 361102, China
[d] National Institute for Data Science in Health and Medicine, School of Medicine, Xiamen University, Xiamen 361102, China

ARTICLE INFO

Keywords:
Deep convolutional neural networks
Data-independent acquisition proteomics
OpenSWATH
Deep learning

ABSTRACT

OpenSWATH is an analysis toolkit commonly used for data independent acquisition (DIA). Although the output of OpenSWATH is controlled at 1% false discovery rate (FDR), the output report still contains many peptide precursors with low similarity fragments. At the last step of OpenSWATH for peptide quantification, researchers usually need to manually check the similarity of the extracted ion chromatograms (XICs) of fragments to distinguish the high confidence and the low confidence peptide precursors. Here we developed an algorithm with a Graphic User Interface named MSSort-DIA^{XMBD}, which combines the deep convolutional neural network (CNN) and the double-threshold segmentation process, to automatically recognize the high confidence precursors and low confidence precursors. To train the model of MSSort-DIA^{XMBD}, we built a database contained about 50,000 manually classified peptide precursors acquired from different instrument platforms and different species. With the double-threshold segmentation strategy, MSSort-DIA^{XMBD} can reduce the number of the low confidence peptides required for manual inspections to less than 10% and be used as the last step of OpenSWATH to visualize and classify the MS/MS data of peptide precursors.

Significance: Although the output of OpenSWATH is controlled at 1% false discovery rate (FDR), the output report still contains many peptide precursors with low similarity fragments. At the last step of OpenSWATH for peptide quantification, researchers usually need to manually check the similarity of fragment XICs to distinguish the high confidence and the low confidence peptide precursors. However, manual inspection is inefficient. For instance, it takes about 50 h to sort even a small dataset of 1000 MS/MS spectra manually. In this paper we developed a software named MSSort-DIA^{XMBD} to automatically recognize the high confidence precursors. We manually classify 50,000 peptide precursors as training set to train a convolutional neural network. After training finished, MSSort-DIA^{XMBD} takes only a few minutes to automatically classify 20,000 peptide precursors, leaving a small portion of fuzzy ones for manual inspection. On the benchmarked dataset, MSSort-DIA^{XMBD} can significantly improve the efficiency and accuracy of recognition of high confidence peptide precursors.

1. Introduction

Mass spectrometry technology (MS) is widely used for peptide and protein identification/quantification. Data-independent acquisition (DIA) is a deterministic and reproducible strategy for peptide and protein quantification [1–3]. An implementation of DIA methods is named sequential window acquisition of all theoretical mass spectra (SWATH-MS) [3,4], which records all fragments from the corresponding precursor isolation window with a large range of mass-to-charge ratio (m/z). Several software tools, such as OpenSWATH [5], Spectronaut [6], Skyline [7], Group-DIA [8], and DIA-NN [9], have been developed for DIA analysis. Also, QuantPipe is a graphic interface software for targeted analysis of DIA data [10]. OpenSWATH is a common toolkit to analyze SWATH-MS data, firstly extracting chromatograms from the spectral

* Corresponding authors at: National Institute for Data Science in Health and Medicine, School of Medicine, Xiamen University, Xiamen 361102, China.
E-mail addresses: jhan@xmu.edu.cn (J. Han), jianweishuai@xmu.edu.cn (J. Shuai).
[1] These authors contributed equally to this work.

https://doi.org/10.1016/j.jprot.2022.104542
Received 31 May 2021; Received in revised form 17 February 2022; Accepted 18 February 2022
Available online 26 February 2022

library and then filtering peptide precursors according to the scoring algorithms.

Although OpenSWATH includes PyProphet [11] and Percolator [12] for statistical validation, where the false discovery rate (FDR) could be controlled at 1% in the "global" context, there are still lots of peptide precursors with low confidence, because the shapes of the extracted ion chromatograms (XICs) of fragments show low similarity. These low confidence peptide precursors cannot be detected by statistic validation alone. Identifying the high confidence peptide precursors which display strong similarity of peak-shaped XICs is often required for peptide quantification. Therefore, there is a need to develop a software toolkit that can accurately and quickly classify the peptide precursors reported by OpenSWATH.

Over the past several years, machine learning and deep learning techniques have made great progresses in dealing with classification problems [13–16]. The deep learning has been applied in mass spectrometry in recent years. Tran et al. proposed a deep neural network model for de novo peptide sequencing [17]. Zohora et al. designed DeepIso based on deep learning to detect peptide features [18]. Ma et al. used deep learning method to improve the prediction of peptide retention time [19]. He et al. have developed a library-free software combined unsupervised deep learning and machine learning to directly analyze DIA data [20]. Wu et al. applied deep learning method as a substitute for the conventional manual peak picking pipeline of MS/MS data [21]. Xu et al. proposed a semi-supervised Convolutional Transformer for automated peak detection, in which prior information about a peptide precursor is used for multi-channel time series segmentation [22].

Manual inspection of MS/MS data filtered by OpenSWATH still serves as a required step in accurate quantification of peptide precursors. The typical procedure of manual inspection is to draw the fragment XICs of each peptide precursor and save it as a picture, and then researchers observe the similarity between the fragment XICs in each picture, and finally judge whether the inspected peptide precursor is high confident peptide or not. Researchers will discard peptide precursors with dissimilar fragment XICs. Manual inspection is a tedious work, and researchers can process about 20 images per hour. Computer vision with deep learning based methods can accelerate this process. MS/MS data visualization is a key part in the process of manual inspection, and there are some tools available to visualize chromatogram. Skyline [7], TAPIR [23], TRIC [24], and TOPPView [25] are familiar tools for raw chromatogram visualization. DrawAlignR is an interactive tool for cross-run chromatogram alignment visualization [26].

However, there is not any open-source tool combining deep learning method and spectra visualization for the confidence degree assessment of peptide precursors. Here we developed a toolkit named MSSort-DIA^{XMBD} to classify the peptide precursor data filtered by OpenSWATH with deep learning method. It integrates a visualization plugin to locate and show the XICs of peptide fragments and then uses a deep convolutional neural network (CNN) to speed up the confidence degree assessment of the peptide precursor data. We created a database with about 50,000 peptides, which are manually classified into two categories of high confidence and low confidence peptide precursors, to train the model of MSSort-DIA^{XMBD}. Due to the strong ability of CNN in classification, MSSort-DIA^{XMBD} shows excellent performances in distinguishing high confidence and low confidence peptide precursors. MSSort-DIA^{XMBD} is a useful open-source software that fulfil the need for eliminating false identification of peptide precursors by OpenSWATH.

2. Materials and methods

2.1. The workflow of MSSort-DIA^{XMBD}

OpenSWATH-PyProphet-TRIC workflow is commonly used as a quantification workflow for DIA data. Although the FDR value is controlled at 1% by the algorithm, researchers usually need to manually inspect the similarity of the top 6 fragment XICs from the same peptide precursor. The manual inspection results contain high confidence and low confidence classes (Fig. 1 and Fig. 4D). MSSort-DIA^{XMBD} can substitute for manual inspection to classify the top 6 fragment XICs of the same peptide precursor. MSSort-DIA^{XMBD} calculates the similarity scores for each fragment XIC group. Researchers can set the upper similarity threshold and the lower similarity threshold to obtain the high confidence peptide precursors that the 6 XICs show strong peak-shaped similarity and low confidence peptide precursors that the 6 XICs hardly display similarity in shapes, but with strong noises. We defined the peptide precursors with scores between the upper threshold and the lower threshold as fuzzy peptide precursors that the 6 XICs show certain similarity in peak-shaped curve but with high fluctuation. Researchers only need to manually check the fuzzy precursors instead of checking all the data.

2.2. The algorithms of MSSort-DIA^{XMBD} in classification

We applied CNN for classification in MSSort-DIA^{XMBD}. CNN excels at processing multiple arrays [27], and can automatically learn the potential spatial correlation of the given data according to its structure [28,29]. The LeNet-5 developed by LeCun which applied back-propagation algorithm performs well in recognizing hand-written digit characters with relatively few parameters [13,30,31]. Considering that LeNet can overcome the variance of XICs caused by normalization, we referred the structure of LeNet to design our model. We designed the rectangle convolutional kernel to adapt to the input matrix with a shape of 6 rows and 85 columns (Table. 1 and Fig. 2A). The rows of the input matrix represent the top 6 fragment XICs. The columns of the input matrix represent the length of XICs. To ensure that there is one peak included in the XIC, we take 42 points before and 42 points after the peak to obtain an XIC, which gives 85 points in total as the length of XIC. The time interval between each 2 points is 3.6 s. Therefore, with the column number of 85, the time interval is 85*3.6 = 306 s. According to a series of tests, in which the XIC length of 85 was proved to be the best among different intervals of 40, 55, 70, 85, 100, 115, and 130.

When training our model, we randomly shuffled the order of 6 XICs for 4 times for data augmentation to avoid overfitting. And we employed the adaptive moment estimation (Adam) algorithm with batch size of 256 peptides, beta1 of 0.9, beta2 of 0.999, epsilon of 1e-8, weight decay of 5e-4, and learning rate of 0.001. The number of training epochs was set to be 100.

To further evaluate the proposed CNN model, we compared it with other methods, including the Pearson and the Spearman correlations, deep neural network (DNN), recurrent neural network (RNN), support vector machine (SVM), and random forest. The parameters of these models were obtained by manual testing. We tested a series of parameter combinations to obtain the final optimized parameters.

For the Pearson and the Spearman correlations, we calculated the Pearson and the Spearman correlation coefficients of each two XICs and averaged them to get the final similarity scores. The formula of the Pearson correlation is given as follows:

$$\rho_{XY} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E(XY) - E(X)E(Y)}{\sigma_X \sigma_Y}$$

where X and Y represent two fragment XICs, the numerator is the calculation formula of covariance of X and Y, and σ_X and σ_Y represent the standard deviation of X and Y, respectively. $E(X)$ and $E(Y)$ represent the mean values of X and Y, respectively.

The formula of the Spearman correlation is given as follows:

$$\rho_s = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)}$$

where d_i^2 is the difference between the two ranks of each input fragment XIC, and n is the number of the input samples.

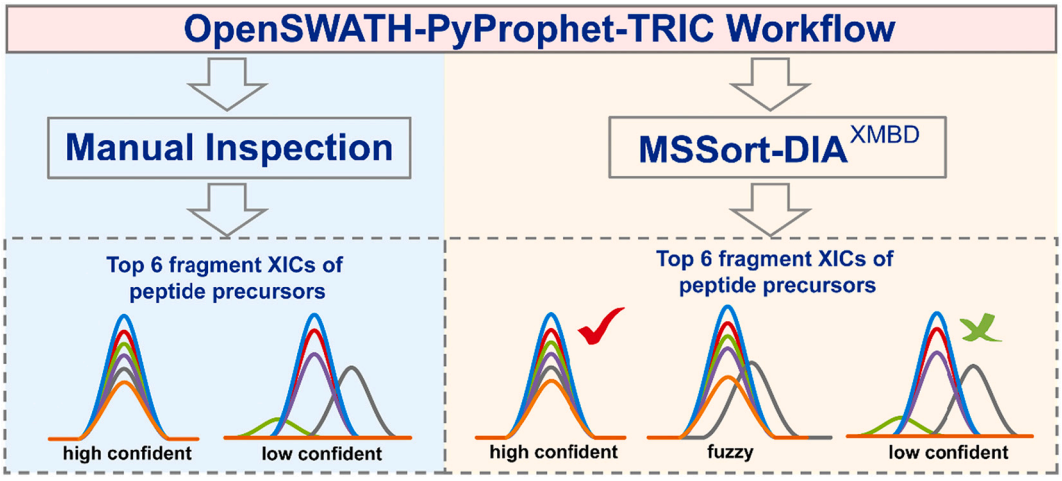


Fig. 1. The workflow of MSSort-DIA^{XMBD}. The blue and yellow parts represent the manual inspection process and MSSort-DIA^{XMBD} analyzing process, respectively. The colored Gaussian curve represents the fragment XICs of peptide precursors. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Detailed structure of our proposed CNN classifier. The size of the input data is (1, 6, 85) with the input format of (channel, height, width). Layer Conv1 is a convolutional layer with the kernel size of (2, 7), the output size of (5, 79), and the ReLU activation. Max pooling1 is a max-pooling layer with the kernel size of (2,2), the output size of (4, 78). Layer Conv2 is a convolutional layer with the kernel size of (2,3), the output size of (3, 76), and the ReLU activation. Layer Max pooling2 is a max-poling layer with the kernel size of (2, 2), the output size of (2, 75). Layer Conv3 is a convolutional layer with the kernel size of (2, 3), the output size of (3, 75), the spatial padding of (1,1), and the ReLU activation. Layer Max pooling3 is a max-poling layer with the kernel size of (2, 2), the output size of (2, 74). Layer Dense1 is a fully connected layer with 512 neurons, the dropout rate of 0.3, and the activation of ReLU. Layer Dense2 is a fully connected layer with 256 neurons, the dropout rate of 0.3, and the activation of ReLU. The output layer is a fully connected layer with one unit and the activation of Sigmoid.

	No. of filters	Size of operators	Padding	Strides	Size of outputs	Activation function
Conv1	64 filters	(2, 7)	–	1	(5, 79)	ReLU
Max pooling1	–	(2, 2)	–	1	(4, 78)	–
Conv2	128 filters	(2, 3)	–	1	(3, 76)	ReLU
Max pooling2	–	(2, 2)	–	1	(2, 75)	–
Conv3	256 filters	(2, 3)	(1, 1)	1	(3, 75)	ReLU
Max pooling3	–	(2, 2)	–	1	(2, 74)	–
Dense1	512 neurons	–	–	–	(1, 512)	ReLU
Dense2	256 neurons	–	–	–	(1, 256)	ReLU
Output	1 neuron	–	–	–	(1, 1)	Sigmoid

The input data of DNN, RNN, SVM and Random Forest are a one-dimensional vector spliced by the top 6 fragment XICs. Since the length of each XIC is 85, the length of input vector is $6 \times 85 = 510$.

Our proposed DNN comprises four fully connected layers, including the input layer with 510 neurons, the first hidden layer with 516 neurons, the second hidden layer with 256 neurons and the output layer with 1 neuron. The activation function of both hidden layer 1 and hidden layer 2 is ReLU, and the dropout parameter was set to be 0.3. The activation function of the output layer is Sigmoid (Fig. 2B). We employed the Adam optimizer with the batch size of 256, the weight decay of 5×10^{-4}, the learning rate of 0.001, and the training epoch of 100.

Our proposed RNN comprises 4 dense layers, including an input layer with the size of (6, 85), the first hidden layer with 256 neurons, the second hidden layer with 256 neurons, and the output layer with 1 neuron. The activation function of the hidden layers is ReLU, and that of the output layer is Sigmoid (Fig. 2C). We employed the Adam algorithm with the batch size of 256, the learning rate of 0.001, and the number of training epochs of 100.

SVM is a kind of machine learning algorithm which can maximize the margin between the training patterns and the decision boundary [32–35]. When training the SVM model in *sklearn.svm.SVC* package, we set the 'regularization' parameter to be 0.5, the 'kernel' parameter to be 'rbf', and the 'gamma' parameter to be 0.1.

Random Forest is an ensemble of classification and regression trees (CART) [36] trained on datasets, which are created from a random resampling on the training set itself [37]. When training the Random Forest model in *sklearn* package, we set the parameter 'n_estimator' to be 250, for it had the best performance among parameters of 50, 100, 150, 200, 250, 300, and 350. We set all the other parameters to be the default value based on empirical tests.

The output of each model is a score representing the probability of the confidence degree assessment of peptide, and the predicted class of certain peptide is determined by the output score.

2.3. Training settings

The experiments in this paper were run under the software platform of Python with version of 3.7. The CNN, DNN, and RNN classifiers were implemented in deep learning framework of MXNet (version 1.5.0) [38] (https://mxnet.apache.org). The SVM and Random Forest classifiers were implemented in Scikit-learn (version 0.24.2, http://scikit-learn.org).

3. Results

3.1. Classified database and data processing of MS/MS

We used the amino acid sequence of peptides and the charge value of precursor identified by OpenSWATH to distinguish the high confidence

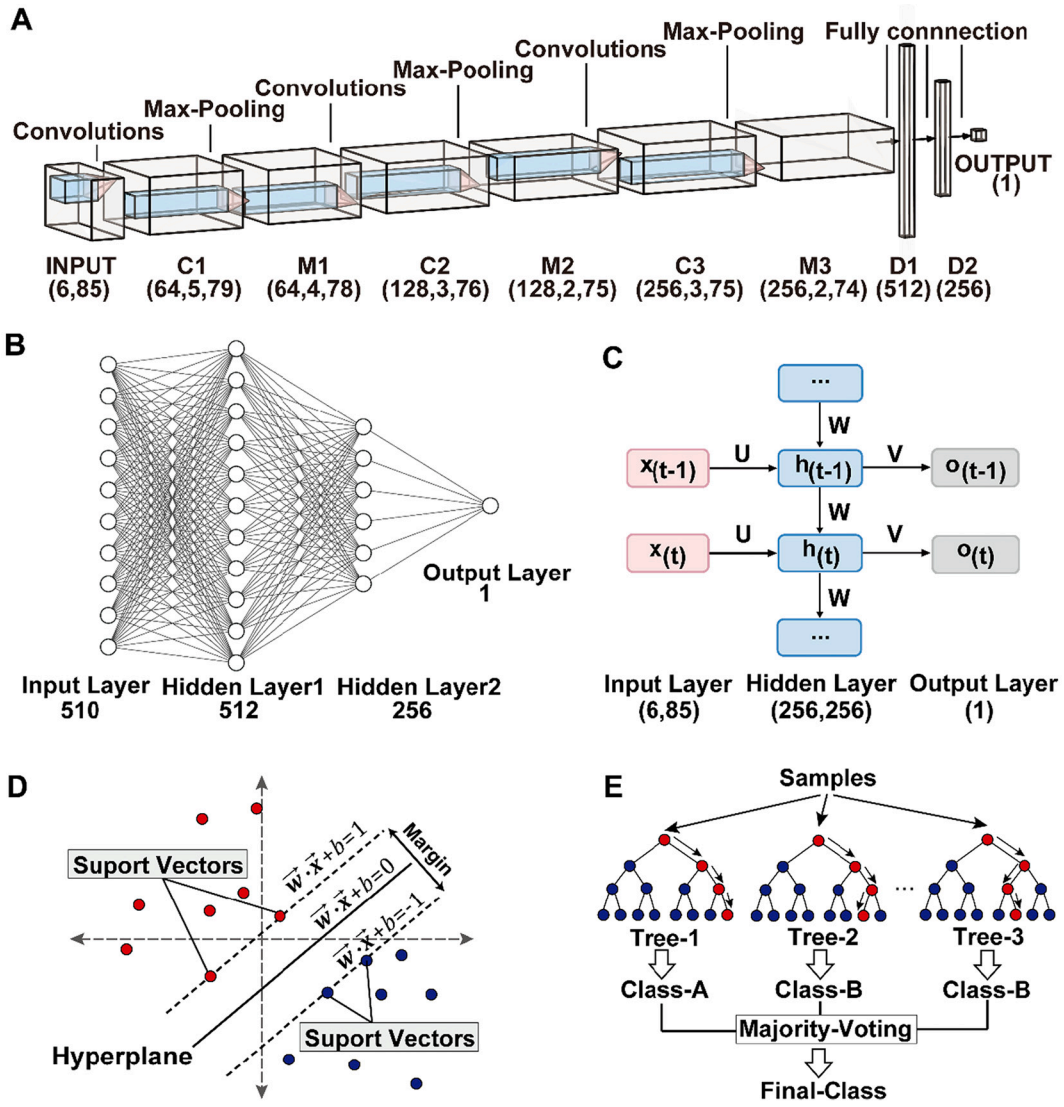


Fig. 2. The proposed neural networks and machine learning models. (A) The model of the proposed CNN. "C" represents the convolutional layer, "M" represents the max pooling layer, and "D" represents the dense layer. The number inside the bracket represents the output size of the corresponding layer. (B) The model of the proposed DNN. The output sizes of the input layer, hidden layer1, hidden layer2, and output layer are set to be 510, 512, 256, and 1, respectively. (C) The model of the proposed RNN. "x" is the value of the input layer, "h" is the value of the hidden layer, "o" is the value of output layer, "U" is the weight matrix from the input layer to the hidden layer, "V" is the weight matrix from the hidden layer to the output layer. "t" is the depth of RNN model on the time dimension. The number inside the bracket represents the output size of the corresponding layer. (D) The model of SVM. The solid line represents the classification hyperplane. "$\vec{w}$" and "b" are the weight and bias. "x" is the input vector. The input vectors located in margin hyperplanes are support vectors. (E) The model of Random Forest. The blue circles and red circles represent all paths and the selected path in decision trees, respectively. Each decision tree predicts the labels for input samples. Random forest integrates the voting results of all decision trees to predict the label of the sample. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

peptides in our deep learning-based classifier. The benchmarked dataset includes SWATH-MS Gold Standard (SGS) human dataset, L929 mouse dataset [8], HYE110 dataset, HYE124 dataset, HeLa dataset, BGS mouse dataset and *E. coli* dataset. The SGS human dataset [5] was acquired from TripleTOF 5600 with 32 fixed windows and gradient length of 2 h. The L929 mouse dataset contains triplicate samples with 100 variable windows measured in SWATH mode on TripleTOF 5600 mass spectrometer. The HYE110 and HYE124 dataset [39] were acquired by TripleTOF 5600 or TripleTOF 6600 mass spectrometers with gradient length of 2 h, the window number of 32 or 64 and fixed or variable window sizes. The HeLa dataset [40] was acquired from Q Exactive HF-X mass spectrometer with 45 windows and gradient length of 2 h. The BGS mouse datasets [41] were acquired from Orbitrap Fusion Lumos mass spectrometers (Thermo Fisher Scientific, San Jose, CA) with 40 windows and gradient length of 2 h.The *E. coli* dataset [42] was acquired from TripleTOF 6600 mass spectrometer (SCIEX) with 100 variable windows on SWATH mode.

The labeled datasets consist of 51,358 peptide precursors, which have been classified into two categories: high confidence and low confidence peptide precursors. We assume that the peptide precursor is highly confident if its 6 XICs of peptide fragments in the dataset coincide well with each other at the crest. On the contrary, a peptide precursor is considered to be low confident if the 6 curves don't coincide well at the crest [43]. After manually checking, we obtained 29,387 high confidence peptide precursors and 21,971 low confidence peptide precursors.

We randomly selected 17,631 high confidence and 13,183 low confidence peptide precursors as the training set, and 5878 high confidence and 4394 low confidence peptide precursors as the cross-

validation set, and 5878 high confidence and 4394 low confidence peptide precursors as the testing set. As a result, the ratios of training set, cross validation set and testing set are 6:2:2.

During the process of building the spectral libraries, the raw files of mass spectrometry data were converted to profile mzXML files using MSConvert (V.3.0.19311) and then a pseudo-DDA mgf file was generated using DIA-Umpire [44]. The mgf files were converted to mzXML files using TPP (Trans-Proteomic Pipeline, Version 5.1.0) software for analysis. A database search of the UniprotKB/Swiss-Prot database for mzXML files was performed using Comet [45] (Version 2017.01) and X! Tandem [46] (Version 2013.06.15.1, native and k-score). The pep.xml search results were validated and scored using PeptideProphet [47] with parameters -p0.05 -l7 -PPM -OAdPE -dDECOY and combined by iProphet [48] with parameters DECOY=DECOY. Mayu [49] (version 1.07) was used to determine the iProphet probability corresponding to 1% peptide FDR. The peptide ions passing the 1% FDR were input into SpectraST for the library building with CID-QTOF setting. The retention time of peptides in sptxt file was replaced with iRT time using spectrast2spectrast_irt.py script (downloaded from www.openswath.org), and the iRT peptides used for retention time normalization were endogenous peptides. The sptxt file was made as the consensus non-abundant spectral library with the iRT retention time using spectraST [50]. The consensus sptxt files were converted to tsv using spectrast2tsv.py script which was then converted to TraML file for OpenSWATH-PyProphet-TRIC workflow. The spectrast2tsv.py script set six transition ions in TraML file with the corresponding parameters. OpenSWATH-PyProphet-TRIC workflow was used to quantify peptides and proteins in spectral libraries. The search parameters are shown in Supplementary Fig. S1 and Fig. S2.

The input of our model consists of the top 6 fragments XICs, and we fixed the length of XIC to be 85 scans based on the window size and our experience. Supplementary Fig. S2 shows the examples of originally high confidence and low confidence peptide precursors.

MSSort-DIA^{XMBD} analyzes the similarity of the extracted ion chromatograms (XICs) of fragments. Different cycling times and mass resolutions influence the process of the XIC similarity extraction. We calibrated for these variations in the data preprocessing stage. To eliminate the effect of different cycling times, we set the time interval of XIC to about 300 s to ensure that each XIC generated by different instruments contains one peak. According to the results of several tests, we set a reasonable binning size of m/z to eliminate the impact of different mass resolutions.

Before training our models, we first applied *minmax scale* method in *sklearn.preprocessing* package provided by Scikit-learn library (http://scikit-learn.org) to normalize each curve. This standardization is to scale the intensity of XIC to lie between zero and one with the transformation formula given as follows:

$$X_{scale} = \frac{X - min(X)}{max(X) - min(X)}$$

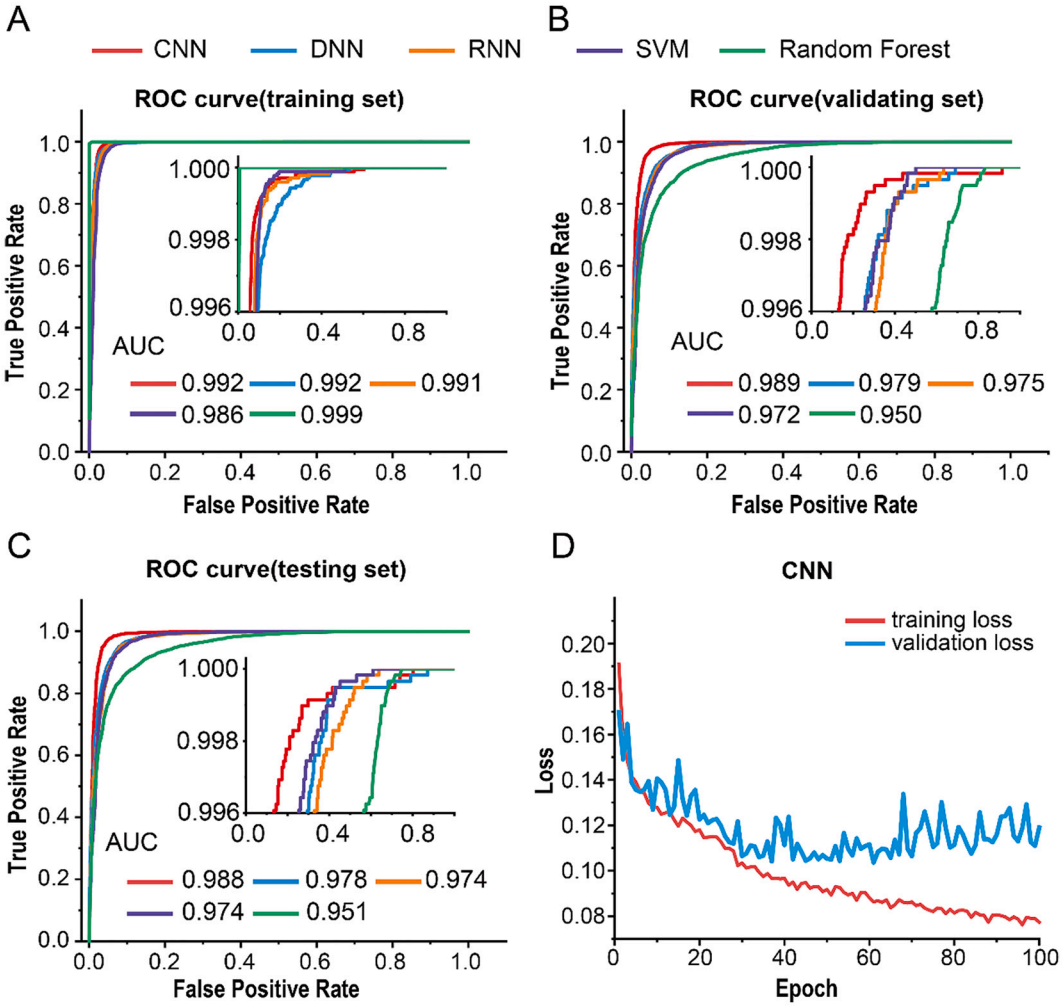


Fig. 3. The ROC curves of different classifiers and the losses of CNN model on the training set and cross validation set. (A) The ROC curves of different classifiers on the training set. (B) The ROC curves of different classifiers on the cross validation set. (C) The ROC curves of different classifiers on the testing sets. (D) The losses of CNN model on the training set and cross validation set. The x-direction is the training number of times.

where $min(X)$ and $max(X)$ are the minimum and maximum of each XIC. Fig. S1B shows the images of peptide precursors of different classes after normalization.

3.2. ROC curves of different classifiers

To compare the performances of CNN model with other four models on binary classification, we computed the evaluation scores including the true high confidence rate and the low confidence rate under different thresholds with the range from 0 to 1, and then compared the ROC curves of the training set, the cross validation set, and the testing set for different classifiers (Fig. 3).

On training set, each classifier has good performance with Area Under Curve (AUC) close to 1 (Fig. 3A). Therefore, all the five classifiers are well-trained without under fitting. As for the Random Forest classifier which performs the best on the training set with the AUC of 0.999, the overfitting is observed for its bad performance on the testing set with the AUC of 0.951.

The ROC curves on validation set and testing set reflect the generalization performance of the classifiers (Fig. 3B and Fig. 3C). According to the AUC results in Fig. 3C, the CNN classifier shows the best generalization performance with the AUC of 0.988 on the testing set, indicating that CNN performs best accuracy for classification. Also, the rank of AUC value on the testing set is the same as that on the cross validation set, proving that our training set is large enough for MS/MS classifier.

The losses of CNN model on both training set and cross validation set are also plotted in Fig. 3D, showing that the CNN model was well trained without overfitting.

3.3. Probability distribution histogram of CNN

According to the ROC curves, the CNN model is proved to be the best in classification. To further evaluate the performances of CNN classifier in distinguishing between the high confidence and low confidence peptide precursors, we plotted the frequency distribution histogram of the predicted score on testing set (Fig. 4A). CNN shows a strong ability in distinguishing the high confidence and the low confidence peptide precursors. As for CNN model, most of the predicted scores are close to either 0 or 1, and only a small part of the predicted scores are in the middle intervals. For the XICs in the middle intervals, although these XICs show certain similarity in peak-shaped curves, but the curves display strong enough fluctuation. As a result, we defined these peptide precursors as the fuzzy peptide precursors, and furthermore employed the bilevel thresholding segmentation method to distinguish the fuzzy peptide precursors with the predicted score in middle intervals.

The output of our model is a probability $P(X)$ of the peptide precursor X, which is an index of confidence degree of peptide precursor. By setting the bilevel threshold values with T_{lower} and T_{upper} to denote the lower and upper threshold, respectively, an original peptide with the probability P calculated by the model can be assigned into one of the following three classes:

$$\begin{cases} Low\ confidence\ if\ P(X) \le T_{lower} \\ Fuzzy\ if\ T_{lower} < P(X) \le T_{upper} \\ High\ confidence\ if\ P(X). > T_{upper} \end{cases}$$

According to the inflection points of the smoothing curve of distribution histogram (Fig. 4A), we set the upper threshold to be 0.92 and the lower threshold to be 0.08. Because the smoothing curve is steep at both ends and flat in the middle, the small change of the two threshold values barely affects the number of predicted fuzzy peptides in the middle range.

3.4. The changes of fuzzy set under different thresholds

Next, we evaluated the proportion changes of fuzzy data in the testing dataset when the bilevel threshold values were changed. Fig. 4B shows the proportions of fuzzy set under different lower thresholds and upper thresholds. The proportion of fuzzy peptide precursors that need to be manually checked is between 1% and 10%. Thus, MSSort-DIA^{XMBD} can reduce the number of manual inspections by more than 90%. The lower threshold and the upper threshold are adjustable parameters, and so the users can change the thresholds according to their requirement. If the upper threshold is increased and the lower threshold is decreased, the percentage of the fuzzy peptide precursors will increase, and therefore a relatively high percentage of fuzzy set is obtained at the expense of increasing confidence degree. If the upper threshold is decreased and the lower threshold is increased, the percentage of fuzzy peptide precursors will decrease at the expense of decreasing confidence degree. Our default values of upper and lower thresholds are 0.08 and 0.92, respectively, which are proved to be the best combination based on our tests.

According to the ROC curves and probability distribution histograms (Fig. 4A and Fig. 4C), we suggest CNN as the best model, for its strong ability to distinguish the high confidence and low confidence peptide precursors with high stable results. As examples, Fig. 4D shows some predicted peptide precursors obtained by CNN with the upper threshold of 0.92, and the lower threshold of 0.08. In the fuzzy XICs in the middle, the parts that do not contain peaks (retention time of 2400–2550 and 2600-2700) are like the low confidence ones, and the parts that contain peaks (retention time of 2550-2560) are similar to the high confidence ones (Fig. 4D). Therefore, the fuzzy one in the middle is hard to be classified to either high confidence or low confidence. The noise comes from the residual pollutants in the sample and the experimental instrument. In the fuzzy XICs, the intensity of peak is relatively low, and the signal-to-noise ratio of peak is relatively low, which is lower than that of high confidence one and higher than that of low confidence one.

3.5. The application of MSSort-DIA^{XMBD}

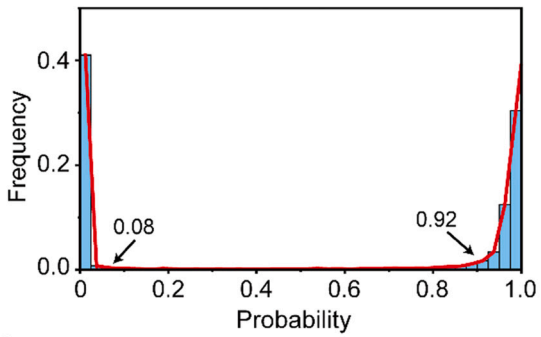
Finally, we designed a user-friendly GUI for MSSort-DIA^{XMBD}, which has two main modules: MS/MS data visualization and classification. The users need to pass the profile mzXML file, window configuration file, and the output of OpenSWATH (version 2.2.0) to MSSort-DIA^{XMBD} for MS/MS data visualization. Then MSSort-DIA^{XMBD} will output the XICs of the fragments of the quantified peptide precursors found by OpenSWATH, and the users can select a certain peptide precursor for XICs visualization. Next, MSSort-DIA^{XMBD} can classify the quantified peptide percursors and output the classification result based on the chosen thresholds. It can also report the numbers and the percentages of the predicted high confidence, fuzzy, and low confidence peptide precursors. The users can modify the values of the two thresholds in order to obtain a satisfied classification (Fig. 5 and Note S1).

4. Discussion

In this work, we developed a tool of MSSort-DIA^{XMBD} combined CNN algorithm and bilevel thresholding method to automatically classify the MS/MS data obtained in the last step of the quantitative analysis of protein by OpenSWATH. To train the model, we generated a database including high confidence peptides and low confidence peptide precursors, which were manually classified from real experiments, to ensure the classifiers can learn the features of the high confidence and low confidence peptide precursors. This database can also be used to train other kinds of neural networks.

As a comparison with other deep learning and machine learning methods including DNN, RNN, SVM, and Random Forest, we show that the performances of CNN in all aspects are good enough for classification application. According to the ROC curves on training, validation and testing sets, the CNN classifier presents a good generalization performance. Especially, CNN was proved to have a strong ability in distinguishing the high confidence and the low confidenve peptide

A

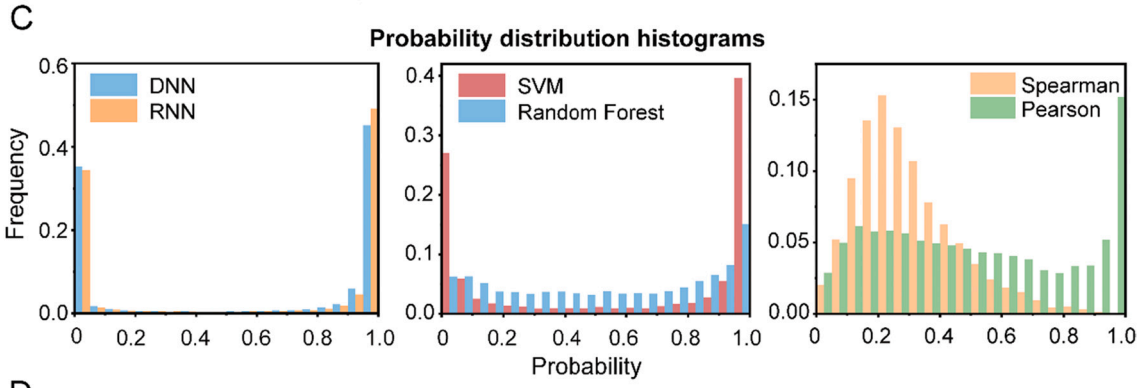


B

The ratios of fuzzy set (CNN)

Lower thresholds	Upper thresholds	Fuzzy ratios (%)	Fuzzy/All
0.08	0.92	10.90	1120/10272
0.16	0.84	6.74	692/10272
0.24	0.76	4.61	474/10272
0.32	0.68	3.04	312/10272
0.40	0.60	1.71	176/10272

C



D

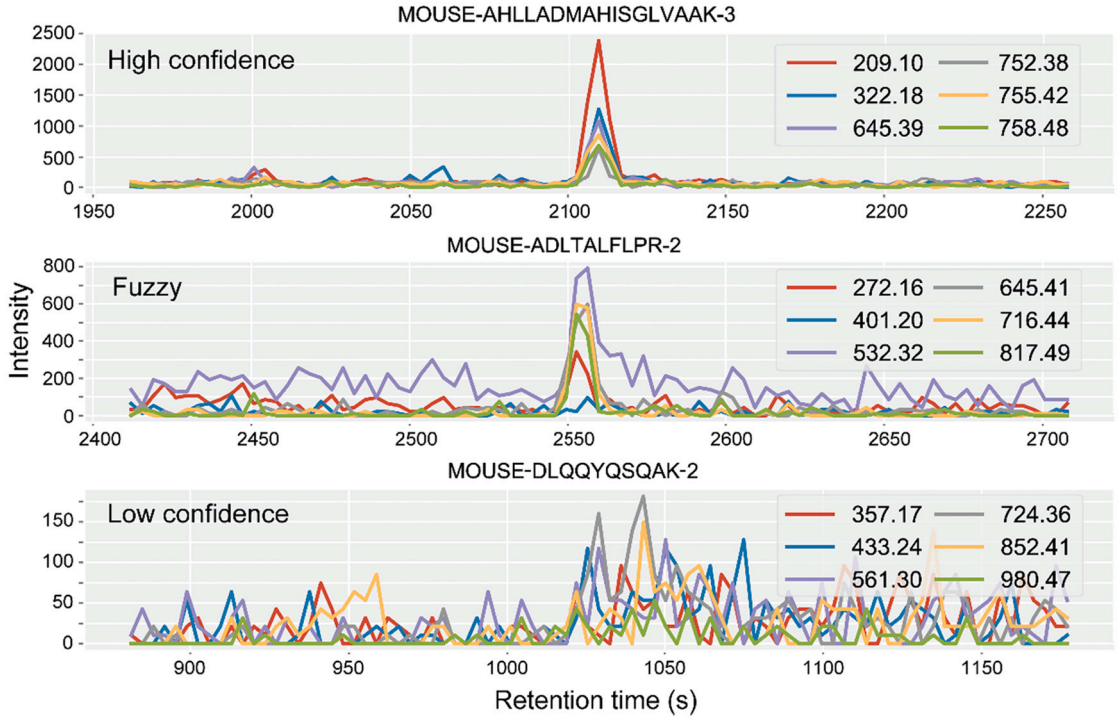


Fig. 4. Results on the testing set. (A) Distribution histogram of the predicted probability of CNN model on testing set. The vertical axis represents the proportions of peptides at a certain bin size of 0.025, while the horizontal axis represents the predicted poss Zibility of confidence assessment of peptide precursor. (B) The changes of fuzzy set under different thresholds. (C) Distribution histograms of the predicted probability of RNN, DNN, SVM, and Random Forest models on testing set. The vertical axis represents the proportions of peptides at a certain bin size of 0.025, while the horizontal axis represents the predicted possibility of confidence assessment of peptide. (D) The top part shows the predicted high confidence peptide by CNN with the probability of 0.997. The middle part shows the predicted fuzzy peptide precursor by CNN with the probability of 0.533. The bottom part shows the predicted low confidence peptide precursor by CNN with the probability of 0.001.

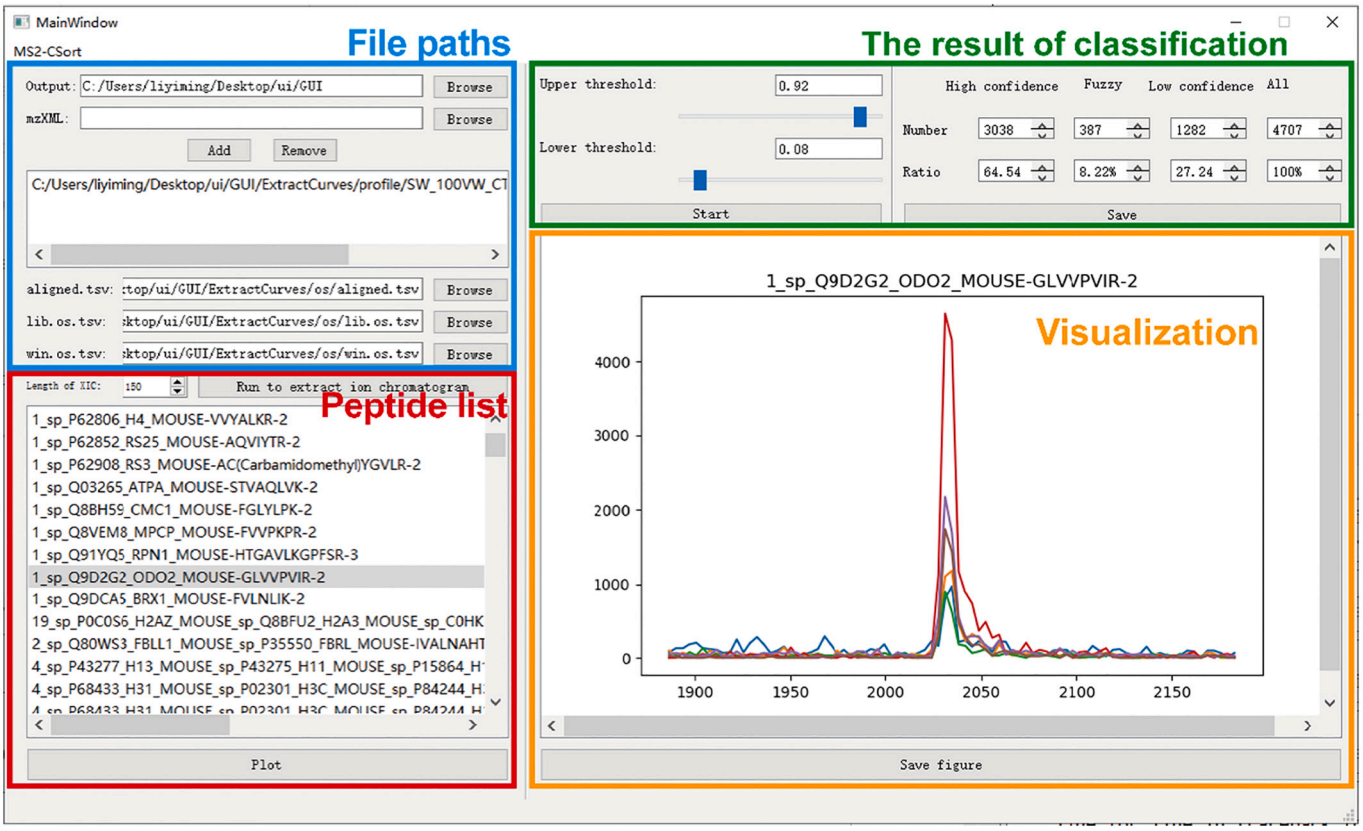


Fig. 5. The interface of MSSort-DIA^{XMBD}. The output directory and the input files are set in the blue box. Aligned.tsv is the output file of OpenSWATH Workflow, which includes OpenSWATH, PyProphet and TRIC software; lib.os.tsv is the output file of spectrast2tsv.py; win.os.tsv is the MS1 isolation file. The function of extracting XICs of fragments of the quantified peptide precursors is included in the red box. In the orange box, XICs of certain peptide can be displayed and saved. The classification function is included in the green box. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

precursors based on the distribution histograms on testing set.

Based on the proposed CNN classifier, we designed a software tool named MSSort-DIA^{XMBD} to visualize and automatically classify the MS/MS data filtered by OpenSWATH. After inputting the MS/MS data by users for classification, MSSort-DIA^{XMBD} can output the name-list of all the peptide precursors from the input. Meanwhile, users can select any peptide precursors and draw the corresponding XIC curves on the interface. If the upper threshold and the lower threshold are given, the software can classify and package the input dataset into three files for high confidence peptide precursors, low confidence peptide precursors and fuzzy peptide precursors, respectively. Besides, it will output an excel file to describe the original file name with the predicted score and the predicted classification where letter N represents the low confidence class, letter F represents the fuzzy class, and letter P represents the high confidence class, for all the peptides in the input dataset.

As a result, the peptides in the file of "low confidence peptide precursor" can be discarded, and the peptides in the file of "high confidence peptide precursor" can be used for further calculation. What left for biologist to manually check are those peptides in the files of "fuzzy peptide precursor". The ratio of the fuzzy peptide precursors can be kept within a certain small range, leaving the manual workload greatly reduced. As a fact, if we set the lower threshold equal to the upper threshold, one classifies the peptide precursors only into the two types of low confidence and high confidence peptide precursors.

As for MSSort-DIA^{XMBD}, setting suitable combination of upper and lower thresholds is still a challenge, and we are exploring other deep learning methods to automatically predict the combination of upper and lower thresholds. MSSort-DIA^{XMBD} automatically checks and filters the low confidence peptides included in the OpenSWATH output files. However, it does not restrict the source of the input spectral library of OpenSWATH. Currently, MSSort-DIA^{XMBD} supports the spectral library built from DIA-Umpire or DDA. In the future, MSSort-DIA^{XMBD} could become compatible with other open-source DIA tools, such as DIANN or MaxDIA, giving its great potential in the field of DIA analysis.

5. Conclusion

We developed MSSort-DIA^{XMBD} for MS/MS data visualization and classification, which performs superior in classifying MS/MS data filtered by OpenSWATH. It can directly acquire the picture of XICs of fragments of certain peptides precursors and replace manual inspection in data-independent acquisition proteomics.

Code availability

The program of MSSort-DIA^{XMBD} is available at https://github.com/jianweishuai/MSSort-DIA-XMBD.

Author information

Jiahuai Han: School of Life Sciences, State Key Laboratory of Cellular Stress Biology, Innovation Center for Cell Signaling Network, National Institute for Data Science in Health and Medicine, Xiamen University, Xiamen, Fujian, 361,102 China; Email: jhan@xmu.edu.cn.

Jianwei Shuai: Department of Physics, State Key Laboratory of Cellular Stress Biology, Innovation Center for Cell Signaling Network, National Institute for Data Science in Health and Medicine, Xiamen University, Xiamen, Fujian, 361,005 China; Wenzhou Institute,

University of Chinese Academy of Sciences, and Oujiang Laboratory (Zhejiang Lab for Regenerative Medicine, Vision and Brain Health), Wenzhou, Zhejiang 325,001, China; Email: jianweishuai@xmu.edu.cn

Author contributions

Jianwei Shuai, Yiming Li, and Qingzu He conceived the project. Yiming Li and Qingzu He developed the algorithm and implemented the software and wrote the manuscript. Huan Guo plotted figures of the manuscript and analyzed data. Chuan-Qi Zhong acquired mass spectrometry data for training deep neural network. Xiang Lidiscussed the algorithms. Yulin Li and Yuer Lu analyzed data. Jiahuai Han and Jianwei Shuai wrote the manuscript and supervised the project.

Declaration of Competing Interest

The authors declare no competing financial interest.

Data availability

Data will be made available on request.

Acknowledgments

This project is supported by the National Natural Science Foundation of China [Grant Nos. 11874310, 12090052 to J.S. and 81788101 to J.H. and No. 11704318 to X.L. and J1310027 to C.Q.Z.], the Fundamental Research Funds for the Central Universities [20720190087] to C.Q.Z, the Fujian Province Foundation (grant no. 2020Y4001), and the 111 Project (grant no. B16029) to J.S.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jprot.2022.104542.

References

[1] J.D. Venable, M.-Q. Dong, J. Wohlschlegel, A. Dillin, J.R. Yates, Automated approach for quantitative analysis of complex peptide mixtures from tandem mass spectra, Nat. Methods 1 (2004) 39–45, https://doi.org/10.1038/nmeth705.
[2] P.C. Carvalho, X. Han, T. Xu, D. Cociorva, M. da Gloria Carvalho, V.C. Barbosa, J. R. Yates, XDIA: improving on the label-free data-independent analysis, Bioinformatics (2010), https://doi.org/10.1093/bioinformatics/btq031.
[3] L.C. Gillet, P. Navarro, S. Tate, H. Röst, N. Selevsek, L. Reiter, R. Bonner, R. Aebersold, Targeted data extraction of the MS/MS spectra generated by data-independent acquisition: A new concept for consistent and accurate proteome analysis, Mol. Cell. Proteomics (2012), https://doi.org/10.1074/mcp.O111.016717.
[4] C. Ludwig, L. Gillet, G. Rosenberger, S. Amon, B.C. Collins, R. Aebersold, Data-independent acquisition-based SWATH - MS for quantitative proteomics: a tutorial, Mol. Syst. Biol. (2018), https://doi.org/10.15252/msb.20178126.
[5] H.L. Röst, G. Rosenberger, P. Navarro, L. Gillet, S.M. Miladinoviä, O.T. Schubert, W. Wolski, B.C. Collins, J. Malmström, L. Malmström, R. Aebersold, OpenSWATH enables automated, targeted analysis of data-independent acquisition MS data, Nat. Biotechnol. (2014), https://doi.org/10.1038/nbt.2841.
[6] R. Bruderer, O.M. Bernhardt, T. Gandhi, S.M. Miladinović, L.Y. Cheng, S. Messner, T. Ehrenberger, V. Zanotelli, Y. Butscheid, C. Escher, O. Vitek, O. Rinner, L. Reiter, Extending the limits of quantitative proteome profiling with data-independent acquisition and application to acetaminophen-treated three-dimensional liver microtissues, Mol. Cell. Proteomics (2015), https://doi.org/10.1074/mcp.M114.044305.
[7] B. MacLean, D.M. Tomazela, N. Shulman, M. Chambers, G.L. Finney, B. Frewen, R. Kern, D.L. Tabb, D.C. Liebler, M.J. MacCoss, Skyline: an open source document editor for creating and analyzing targeted proteomics experiments, Bioinformatics. (2010), https://doi.org/10.1093/bioinformatics/btq054.
[8] Y. Li, C.-Q. Zhong, X. Xu, S. Cai, X. Wu, Y. Zhang, J. Chen, J. Shi, S. Lin, J. Han, Group-DIA: analyzing multiple data-independent acquisition mass spectrometry data files, Nat. Methods 12 (2015) 1105–1106, https://doi.org/10.1038/nmeth.3593.
[9] V. Demichev, C.B. Messner, S.I. Vernardis, K.S. Lilley, M. Ralser, DIA-NN: neural networks and interference correction enable deep proteome coverage in high throughput, Nat. Methods (2020), https://doi.org/10.1038/s41592-019-0638-x.
[10] D. Wang, G. Gan, X. Chen, C.Q. Zhong, QuantPipe: a user-friendly pipeline software tool for DIA data analysis based on the OpenSWATH-PyProphet-TRIC workflow, J. Proteome Res. (2020), https://doi.org/10.1021/acs.jproteome.0c00704.
[11] G. Rosenberger, I. Bludau, U. Schmitt, M. Heusel, C.L. Hunter, Y. Liu, M.J. Maccoss, B.X. Maclean, A.I. Nesvizhskii, P.G.A. Pedrioli, L. Reiter, H.L. Röst, S. Tate, Y. S. Ting, B.C. Collins, R. Aebersold, Statistical control of peptide and protein error rates in large-scale targeted data-independent acquisition analyses, Nat. Methods (2017), https://doi.org/10.1038/nmeth.4398.
[12] M. The, M.J. MacCoss, W.S. Noble, L. Käll, Fast and accurate protein false discovery rates on large-scale proteomics data sets with percolator 3.0, J. Am. Soc. Mass Spectrom. (2016), https://doi.org/10.1007/s13361-016-1460-7.
[13] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE (1998), https://doi.org/10.1109/5.726791.
[14] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 80 (2006), https://doi.org/10.1126/science.1127647.
[15] S. Wang, X. Qian, H. Hu, Q. He, H. Lin, J. Shuai, Study on sleep stages of polysomnography using deep neural network, Biophysics (Oxf) 7 (2019) 15.
[16] Q. Yuan, Z. Hong, X. Wang, J. Shuai, Y. Cao, Application of artificial intelligence in mental illness, Int. J. Psychiatry. 1 (2020) 4–7.
[17] N.H. Tran, X. Zhang, L. Xin, B. Shan, M. Li, De novo peptide sequencing by deep learning, Proc. Natl. Acad. Sci. U. S. A. (2017), https://doi.org/10.1073/pnas.1705691114.
[18] F.T. Zohora, M.Z. Rahman, N.H. Tran, L. Xin, B. Shan, M. Li, DeepIso: A deep learning model for peptide feature detection from LC-MS map, Sci. Rep. (2019), https://doi.org/10.1038/s41598-019-52954-4.
[19] C. Ma, Y. Ren, J. Yang, Z. Ren, H. Yang, S. Liu, Improved peptide retention time prediction in liquid chromatography through deep learning, Anal. Chem. (2018), https://doi.org/10.1021/acs.analchem.8b02386.
[20] Q. He, C. Zhong, X. Li, J. Shuai, J. Han, Deep learning analysis for data-independent acquisition mass spectrometry data, J. Xiamen Univ. Sci. 60 (2021) 97–103.
[21] Z. Wu, D. Serie, G. Xu, J. Zou, PB-net: automatic peak integration by sequential deep learning for multiple reaction monitoring, J. Proteome (2020), https://doi.org/10.1016/j.jprot.2020.103820.
[22] L.L. Xu, H.L. Röst, Peak Detection on Data Independent Acquisition Mass Spectrometry Data with Semisupervised Convolutional Transformers, 2020 arXiv: 2010.13841.
[23] H.L. Röst, G. Rosenberger, R. Aebersold, L. Malmström, Efficient visualization of high-throughput targeted proteomics experiments: TAPIR, Bioinformatics. (2015), https://doi.org/10.1093/bioinformatics/btv152.
[24] H.L. Röst, Y. Liu, G. D'Agostino, M. Zanella, P. Navarro, G. Rosenberger, B. C. Collins, L. Gillet, G. Testa, L. Malmström, R. Aebersold, TRIC: an automated alignment strategy for reproducible protein quantification in targeted proteomics, Nat. Methods (2016), https://doi.org/10.1038/nmeth.3954.
[25] M. Sturm, O. Kohlbacher, TOPPView: an open-source viewer for mass spectrometry data, J. Proteome Res. (2009), https://doi.org/10.1021/pr900171m.
[26] S. Gupta, J. Sing, A. Mahmoodi, H. Röst, DrawAlignR: an interactive tool for across run chromatogram alignment visualization, Proteomics. (2020), https://doi.org/10.1002/pmic.201900353.
[27] Y. Lecun, Y. Bengio, G. Hinton, Deep learning, Nature. (2015), https://doi.org/10.1038/nature14539.
[28] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. (2013), https://doi.org/10.1109/TPAMI.2013.50.
[29] Y.T. Zhou, R. Chellappa, Computation of Optical Flow Using a Neural Network, 1988, https://doi.org/10.1109/icnn.1988.23914.
[30] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L. D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Comput. (1989), https://doi.org/10.1162/neco.1989.1.4.541.
[31] S. Lin, L. Cai, X. Lin, R. Ji, Masked face detection via a modified LeNet, Neurocomputing. (2016), https://doi.org/10.1016/j.neucom.2016.08.056.
[32] B. Boser, I. Guyon, V.V.-P. of the 5th, U, A Training Algorithm for Optimal Margin Classifiers, Gautampendse.Com, 2003, p. 1992.
[33] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, Data Min. Knowl. Disc. (1998), https://doi.org/10.1023/A:1009715923555.
[34] V.N. Vapnik, The Nature of Statistical Learning Theory, 2000, https://doi.org/10.1007/978-1-4757-3264-1.
[35] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, 2000, https://doi.org/10.1017/cbo9780511801389.
[36] A.D. Gordon, L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and regression trees, Biometrics. (1984), https://doi.org/10.2307/2530946.
[37] A. Sarica, A. Cerasa, A. Quattrone, Random forest algorithm for the classification of neuroimaging data in Alzheimer's disease: A systematic review, Front. Aging Neurosci. (2017), https://doi.org/10.3389/fnagi.2017.00329.
[38] T. Chen, M. Li, U.W. Cmu, Y. Li, M. Lin, N. Wang, M. Wang, B. Xu, C. Zhang, Z. Zhang, U. Alberta, MXNet : A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems arXiv: 1512, 01274v1 [cs. DC] 3, Emerald Gr. Publ. Ltd, Dec 2015, p. 2015.
[39] P. Navarro, J. Kuharev, L.C. Gillet, O.M. Bernhardt, B. MacLean, H.L. Röst, S. A. Tate, C.C. Tsou, L. Reiter, U. Distler, G. Rosenberger, Y. Perez-Riverol, A. I. Nesvizhskii, R. Aebersold, S. Tenzer, A multicenter study benchmarks software tools for label-free proteome quantification, Nat. Biotechnol. (2016), https://doi.org/10.1038/nbt.3685.

[40] J. Muntel, T. Gandhi, L. Verbeke, O.M. Bernhardt, T. Treiber, R. Bruderer, L. Reiter, Surpassing 10000 identified and quantified proteins in a single run by optimizing current LC-MS instrumentation and data analysis strategy, Mol. Omi. (2019), https://doi.org/10.1039/c9mo00082h.

[41] J. Muntel, J. Kirkpatrick, R. Bruderer, T. Huang, O. Vitek, A. Ori, L. Reiter, Comparison of protein quantification in a complex background by DIA and TMT workflows with fixed instrument time, J. Proteome Res. (2019), https://doi.org/10.1021/acs.jproteome.8b00898.

[42] M.K. Midha, U. Kusebauch, D. Shteynberg, C. Kapil, S.L. Bader, P.J. Reddy, D. S. Campbell, N.S. Baliga, R.L. Moritz, A comprehensive spectral assay library to quantify the Escherichia coli proteome by DIA/SWATH-MS, Sci. Data 7 (2020) 389, https://doi.org/10.1038/s41597-020-00724-7.

[43] L. Reiter, O. Rinner, P. Picotti, R. Hüttenhain, M. Beck, M.Y. Brusniak, M. O. Hengartner, R. Aebersold, MProphet: automated data processing and statistical validation for large-scale SRM experiments, Nat. Methods (2011), https://doi.org/10.1038/nmeth.1584.

[44] C.-C. Tsou, D. Avtonomov, B. Larsen, M. Tucholska, H. Choi, A.-C. Gingras, A. I. Nesvizhskii, DIA-umpire: comprehensive computational framework for data-independent acquisition proteomics, Nat. Methods 12 (2015) 258–264, https://doi.org/10.1038/nmeth.3255.

[45] J.K. Eng, T.A. Jahan, M.R. Hoopmann, Comet: an open-source MS/MS sequence database search tool, Proteomics. 13 (2013) 22–24, https://doi.org/10.1002/pmic.201200439.

[46] R. Craig, R.C. Beavis, TANDEM: matching proteins with tandem mass spectra, Bioinformatics. 20 (2004) 1466–1467, https://doi.org/10.1093/bioinformatics/bth092.

[47] A. Keller, A.I. Nesvizhskii, E. Kolker, R. Aebersold, Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search, Anal. Chem. 74 (2002) 5383–5392, https://doi.org/10.1021/ac025747h.

[48] D. Shteynberg, E.W. Deutsch, H. Lam, J.K. Eng, Z. Sun, N. Tasman, L. Mendoza, R. L. Moritz, R. Aebersold, A.I., Nesvizhskii, iProphet: multi-level integrative analysis of shotgun proteomic data improves peptide and protein identification rates and error estimates, Mol. Cell. Proteomics 10 (2011), https://doi.org/10.1074/mcp.M111.007690. M111.007690-M111.007690.

[49] L. Reiter, M. Claassen, S.P. Schrimpf, M. Jovanovic, A. Schmidt, J.M. Buhmann, M. O. Hengartner, R. Aebersold, Protein identification false discovery rates for very large proteomics data sets generated by tandem mass spectrometry, Mol. Cell. Proteomics MCP 8 (2009) 2405–2417, https://doi.org/10.1074/mcp.m900317-mcp200.

[50] H. Lam, E.W. Deutsch, J.S. Eddes, J.K. Eng, N. King, S.E. Stein, R. Aebersold, Development and validation of a spectral library searching method for peptide identification from MS/MS, Proteomics. 7 (2007) 655–667, https://doi.org/10.1002/pmic.200600625.