nature methods

BRIEF COMMUNICATION

https://doi.org/10.1038/s41592-018-0260-3

Deep learning enables de novo peptide sequencing from data-independent-acquisition mass spectrometry

Ngoc Hieu Tran[1], Rui Qiao[2], Lei Xin[3], Xin Chen[3], Chuyi Liu[4], Xianglilan Zhang[5], Baozhen Shan[3], Ali Ghodsi[2] and Ming Li[1*]

We present DeepNovo-DIA, a de novo peptide-sequencing method for data-independent acquisition (DIA) mass spectrometry data. We use neural networks to capture precursor and fragment ions across *m/z*, retention-time, and intensity dimensions. They are then further integrated with peptide sequence patterns to address the problem of highly multiplexed spectra. DIA coupled with de novo sequencing allowed us to identify novel peptides in human antibodies and antigens.

Personalized immunotherapy is revolutionizing cancer treatment[1–4]. However, challenges remain in identifying and validating neoantigens that can elicit effective antitumor T cell responses in each individual. The current process of exome sequencing, somatic-mutation calling, and prediction of major-histocompatibility-complex binding is a long and unreliable detour for the identification of neoantigens brought to the cancer-cell surface[3,4]. This process can be complemented and validated by mass spectrometry (MS) technology[3–5]. In addition to the need for sufficient tumor samples for MS analysis, the following two requirements must be addressed: (i) sufficient sensitivity to detect low-abundance peptides and (ii) the capability to discover novel sequences that do not exist in any database.

Recent advances in DIA strategies[6,7] allow the fragmentation of all precursor ions within a certain range of *m/z* and retention time in an unbiased and untargeted fashion, in contrast to data-dependent acquisition (DDA) and selected-reaction monitoring. DIA experiments produce a complete record of all peptides that are present in a sample, including those with low abundance. A remaining question is how to decode these data to extract meaningful information. MS/MS spectra from DIA are notoriously hard to interpret because they are highly multiplexed. Each spectrum contains fragment ions from multiple precursor ions, and the link between a precursor ion and its fragment ions is unknown. This challenge prevents many DIA database search engines from achieving identification power comparable to that of their DDA counterparts[7–10]. The problem is even more acute for the de novo sequencing approach, and no method has been proposed to address it. Indeed, the complexity of dynamic programming or graph-based algorithms in most de novo sequencing methods for DDA may increase exponentially with the multiplexity of MS/MS spectra from DIA. Thus, a new viable solution is needed for de novo sequencing using DIA data.

We recently described DeepNovo[11], a deep-learning-based model for de novo sequencing using DDA data. The key idea is the application of neural networks to better learn features of fragment ions and peptide sequences. We have observed that, in contrast to many complicated optimization algorithms, the iterative sequencing framework of DeepNovo makes it possible to extend to DIA without any increase in complexity. More important, to address the problem of highly multiplexed spectra, we restructure the neural networks to utilize the extra dimensionality of DIA data (*m/z* and retention time) to identify coeluting patterns of a precursor ion and its fragment ions, as well as fragment ions across multiple neighbor spectra. This evidence allows DeepNovo-DIA to pick up the correct signal for de novo sequencing amid a large amount of noise in a DIA spectrum. Taking all these considerations into account, we redesigned DeepNovo-DIA to enable de novo sequencing using DIA data.

Figure 1 presents our de novo sequencing workflow. First, precursor features are detected together with their *m/z*, charge, retention time, and intensity profile from the LC-MS map[12]. Next, for each precursor, we collect all MS/MS spectra so that they are within the precursor's retention-time range, and ensure that their DIA *m/z* windows cover the precursor's *m/z*. Because the number of spectra collected for a precursor may vary, we select a fixed number of spectra that are closest to the center of the precursor's retention time. The closer a spectrum is to the center, the stronger its fragment ion signals are for de novo sequencing. The correlation between the precursor's intensity profile and its fragment ions is also a good indicator for de novo sequencing. Thus, we feed the precursor and its associated MS/MS spectra into DeepNovo-DIA neural networks to learn (i) the 3D shapes of fragment ions along *m/z* and retention-time dimensions, (ii) the correlation between the precursor and its fragment ions, and (iii) the peptide sequence patterns. Our de novo sequencing framework operates in a recurrent and beam-search fashion: at each iteration, the model predicts the next amino acid by conditioning on the output of previous steps and keeps track of only a constant number of top candidate sequences. As a result, its complexity does not increase with the number of peptides or with the number of ions in the spectrum. Finally, de novo peptides can be validated through an augmented database search with a controlled false discovery rate (FDR) to ensure that they are supported by significant peptide–spectrum matches. More details can be found in the Methods.

[1]David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada. [2]Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, ON, Canada. [3]Bioinformatics Solutions Inc., Waterloo, ON, Canada. [4]Department of Electrical & Computer Engineering, University of Waterloo, Waterloo, ON, Canada. [5]State Key Laboratory of Pathogen and Biosecurity, Beijing Institute of Microbiology and Epidemiology, Beijing, China. *e-mail: mli@uwaterloo.ca

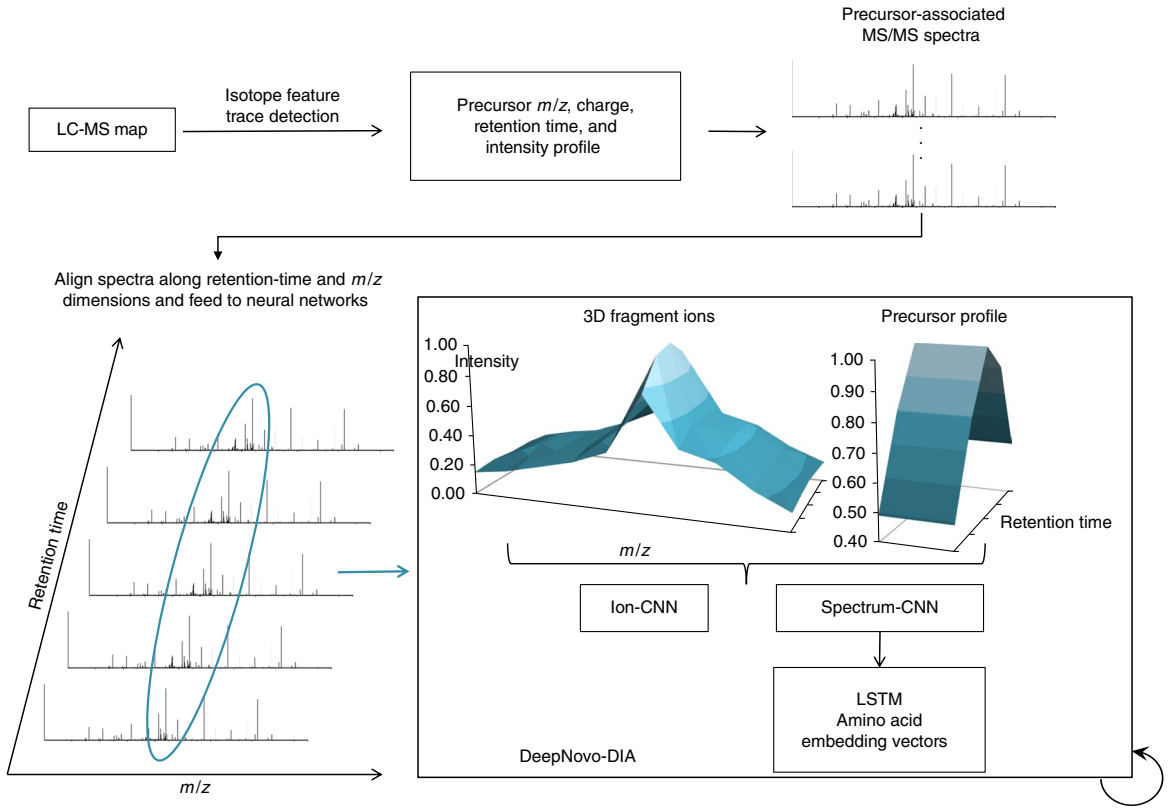

Fig. 1 | The workflow of DeepNovo-DIA for de novo sequencing of DIA data. Precursor ion features are detected together with their *m/z*, charge, retention time, and intensity profile from the LC-MS map. The MS/MS spectra associated with each precursor (i.e., within its retention-time and *m/z* range) are aligned along the retention-time and *m/z* dimensions and then fed to the neural networks. Ion-CNN and spectrum-CNN learn the 3D shapes of fragment ions and the correlation between the precursor and its fragment ions. The long-short-term memory (LSTM) network learns peptide sequence patterns in association with spectrum-CNN. DeepNovo-DIA conducts de novo sequencing in a recurrent fashion, predicting the next amino acid by conditioning on the output of previous steps.

We trained DeepNovo-DIA on a previously obtained dataset of urine samples from 64 subjects[13]. We evaluated DeepNovo-DIA on two other datasets from different subjects who had been diagnosed with ovarian cyst (OC; six subjects) or urinary tract infection (UTI; six subjects). We also tested DeepNovo-DIA on a previously obtained dataset of plasma samples[10]. The test datasets were not used during model development.

We built an in-house database search tool to generate training data. In particular, we followed the approach of DIA-Umpire[9] to generate a pseudo-spectrum from each precursor feature and its associated spectra. Then we used a conventional DDA database search tool, PEAKS DB[12], to search the pseudo-spectra against the Swiss-Prot human database. The peptides identified at 1% FDR were assigned to the corresponding precursors and were used as ground-truth labels for training. Our training set included 2,177,667 spectra, 202,114 labeled precursor features, and 14,400 unique peptides (Supplementary Table 1). For evaluation, we compared DeepNovo-DIA to DIA database search tools including PECAN[10], Spectronaut[14], and OpenSWATH[7]. Such comparisons illustrate (i) the accuracy of de novo sequencing (based on overlapping identifications) and (ii) DeepNovo-DIA's identification of new peptides not found in the database.

We first calculated the accuracy of DeepNovo-DIA using labeled features from the in-house database search. For each labeled feature, we compared the de novo peptide predicted by DeepNovo-DIA with the ground-truth sequence on the basis of the alignment of their mass fragments[11]. We measured the sequencing accuracy at the amino acid level (i.e., the ratio of the total number of matched amino acids to the total length of predicted peptides) and at the peptide level (i.e., the fraction of fully matched peptides). As shown in Fig. 2a, DeepNovo-DIA accurately predicted 63.8–68.1% of amino acids and 37.4–52.4% of peptides of the labeled features. Moreover, DeepNovo-DIA provides a confidence score for each predicted amino acid. Figure 2b shows the distribution of sequencing accuracy with respect to confidence score that allows one to select high-confidence de novo peptides with a certain expected accuracy.

We then applied DeepNovo-DIA to all features, labeled and unlabeled, and used the confidence-score distribution in Fig. 2b to select high-confidence predicted peptides with an expected sequencing accuracy of 90%. Figure 2c shows the substantial overlap of precursor features with peptide identifications by the database search and DeepNovo-DIA. The amino acid accuracy of overlapping features was close to 90%, as expected (Fig. 2d), thus demonstrating the reliability of the DeepNovo-DIA confidence score for quality control. More important, DeepNovo-DIA identified peptides for 33–72.6% of extra features (e.g., plasma dataset 33% = 2,529/(4,207 + 3,466)). We also observed that DeepNovo-DIA's performance was better for the UTI and OC datasets than for the plasma dataset; we suggest that this is because the UTI and OC datasets were more similar to the training data.

Next, we compared DeepNovo-DIA to PECAN and Spectronaut, using the plasma dataset (Supplementary Note 1, Supplementary Figs. 1–5, Supplementary Table 2). DeepNovo-DIA correctly predicted the full sequences of 1,023 database peptides that were reported by PECAN or Spectronaut (Fig. 2e). Among 2,091 peptides reported by both PECAN and Spectronaut, which can be

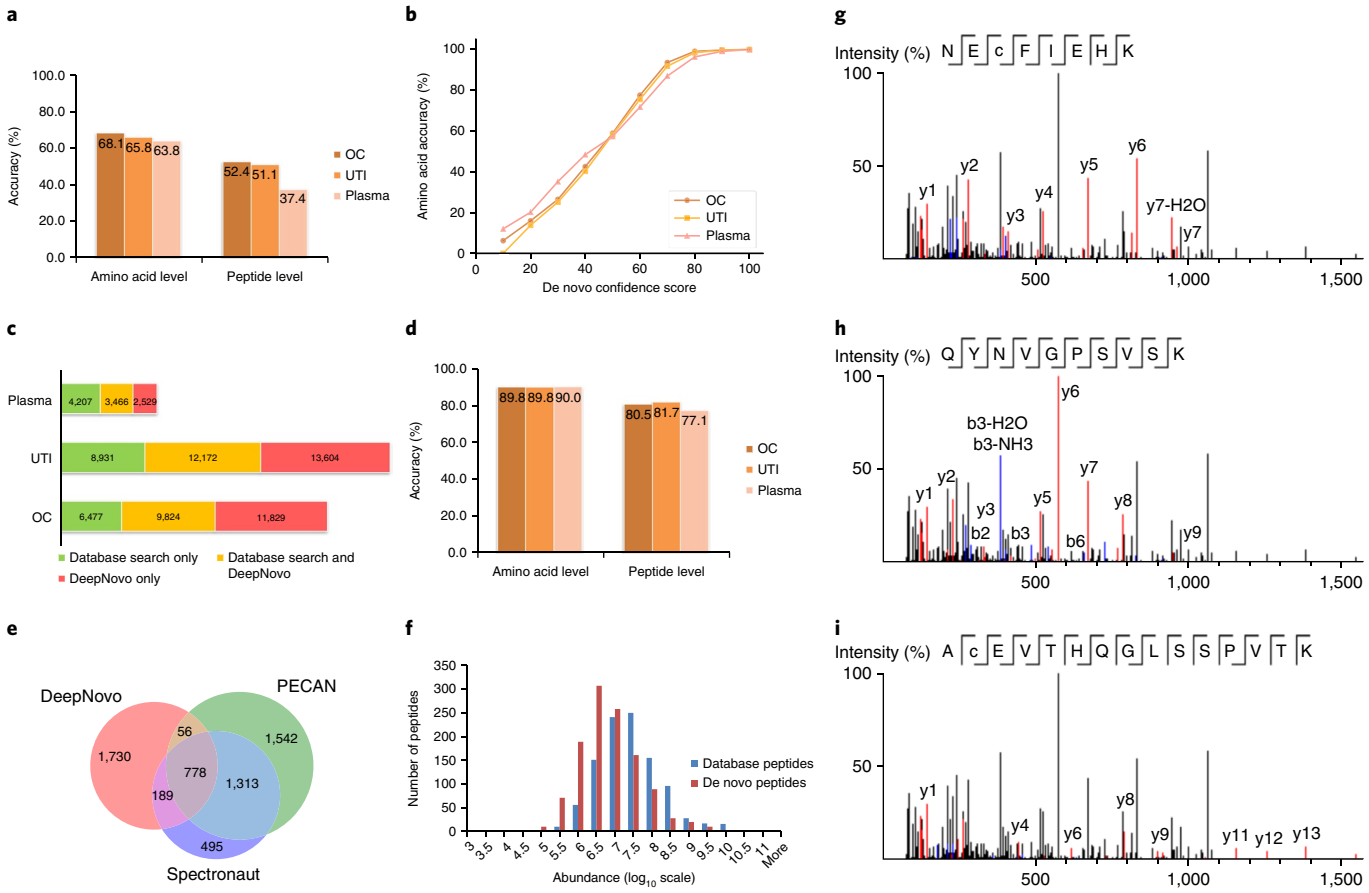

Fig. 2 | DeepNovo-DIA evaluation of three datasets: ovarian cyst (OC), urinary tract infection (UTI), and plasma. a, Accuracy of DeepNovo-DIA on labeled features. **b**, Distribution of DeepNovo-DIA accuracy and confidence scores. **c**, Precursor features with peptide identifications by in-house database search or DeepNovo-DIA. **d**, DeepNovo-DIA accuracy on overlapping features in Fig. 1c. **e**, Comparison of unique peptides identified by DeepNovo-DIA, PECAN, and Spectronaut from the plasma dataset. **f**, Abundance distributions of 1,143 de novo peptides identified by DeepNovo-DIA and 1,023 database peptides identified by DeepNovo-DIA and PECAN or Spectronaut. **g**–**i**, Examples of a DIA spectrum that contains three different peptides, all of which were predicted by DeepNovo-DIA. In each panel, the fragment ions supporting the corresponding peptide are highlighted (red, y ion; blue, b ion).

considered as high-quality database search results, DeepNovo-DIA identified 778 (37.2%). This is comparable to the performance of de novo sequencing tools for DDA data (25–40% at the peptide level[11]). Among peptides reported only by DeepNovo-DIA, 587 could be found in the database and 2,011 were de novo. To ensure that the de novo peptides were supported by significant peptide–spectrum matches, we augmented the database FASTA file with the de novo peptides and re-ran the in-house database search. We found that 1,143 de novo peptides passed 1% FDR after the search was re-run. Thus, 1,730 peptides were identified only by DeepNovo-DIA (587 + 1,143 = 1,730).

Figure 2f shows the abundance distribution of 1,143 de novo peptides compared with that of 1,023 database peptides identified by DeepNovo-DIA and PECAN or Spectronaut. The abundance of de novo peptides ranged from 10^5 to 10^{10}, demonstrating that DeepNovo-DIA scales across multiple levels of abundance. The distribution of de novo peptides showed a shift toward lower abundance compared with that of database peptides. For instance, 270/1,143 (23.6%) de novo peptides had an abundance of $\leq 10^6$, compared with 69/1,023 (6.7%) database peptides. Supplementary Data 1–3 show several examples of low-abundance de novo peptides with supporting peptide–spectrum matches and coelution profiles of precursor and fragment ions.

Next, we ran BLAST on those 1,143 de novo peptides against the broad NCBI human nonredundant protein database to find supporting evidence from previous studies. We found 463 peptide matches with more than 90% identity, and 123 with 100% identity. We found 113 peptides in the variable regions of immunoglobulin light and heavy chains (Supplementary Tables 3 and 4), and 6 peptides with human natural variants (Supplementary Table 5). Note that such variable peptides change from one individual to another and cannot be found via the standard database-search approach. Supplementary Fig. 6 shows an example of three de novo peptides aligned to the variable region of a recently published human antibody for malaria vaccine design[15].

We also applied DeepNovo-DIA to identify novel peptides from data on human leukocyte antigen[16] (Supplementary Note 2, Supplementary Table 6, Supplementary Data 4). Supplementary Figs. 7 and 8 show the results of DeepNovo-DIA, OpenSWATH, and Spectronaut on the Jurkat-Oxford dataset[16]. DeepNovo-DIA correctly predicted the full sequences of 102 database peptides that were reported by OpenSWATH or Spectronaut. Of 106 peptides reported by both OpenSWATH and Spectronaut, DeepNovo-DIA identified 35 (33.0%). Of 202 peptides reported only by DeepNovo-DIA, 72 were found in the database, and the remaining 130 were de novo peptides.

Finally, we show an example of DeepNovo-DIA's application to a DIA spectrum from the plasma dataset that contained mixed fragment ions from three different peptides (Fig. 2g–i). DeepNovo-DIA was able to identify all of them. The last two peptides

were predicted by both DeepNovo-DIA and the database search; however, the first one did not exist in the database. Thus, the combination of DIA and de novo sequencing has the potential to help scientists discover novel peptides and enable more complete profiling of biological samples.

Online content

Any methods, additional references, Nature Research reporting summaries, source data, statements of data availability and associated accession codes are available at https://doi.org/10.1038/s41592-018-0260-3.

Received: 15 May 2018; Accepted: 9 November 2018;
Published online: 20 December 2018

References

1. Ott, P. A. et al. *Nature* **547**, 217–221 (2017).
2. Sahin, U. et al. *Nature* **547**, 222–226 (2017).
3. Anonymous. *Nat. Biotechnol.* **35**, 97 (2017).
4. Vitiello, A. & Zanetti, M. *Nat. Biotechnol.* **35**, 815–817 (2017).
5. Bassani-Sternberg, M. et al. *Nat. Commun.* **7**, 13404 (2016).
6. Venable, J. D., Dong, M. Q., Wohlschlegel, J., Dillin, A. & Yates, J. R. *Nat. Methods* **1**, 39–45 (2004).
7. Röst, H. L. et al. *Nat. Biotechnol.* **32**, 219–223 (2014).
8. Egertson, J. D., MacLean, B., Johnson, R., Xuan, Y. & MacCoss, M. J. *Nat. Protoc.* **10**, 887–903 (2015).
9. Tsou, C. C. et al. *Nat. Methods* **12**, 258–264 (2015).
10. Ting, Y. S. et al. *Nat. Methods* **14**, 903–908 (2017).
11. Tran, N. H., Zhang, X., Xin, L., Shan, B. & Li, M. *Proc. Natl Acad. Sci. USA* **114**, 8247–8252 (2017).
12. Zhang, J. et al. *Mol. Cell. Proteomics.* **11**, M111.010587 (2012).
13. Muntel, J. et al. *J. Proteome. Res.* **14**, 4752–4762 (2015).
14. Bruderer, R. et al. *Mol. Cell. Proteomics.* **14**, 1400–1410 (2015).
15. Tan, J. et al. *Nature* **529**, 105–109 (2016).
16. Caron, E. et al. *eLife* **4**, e07661 (2015).

Acknowledgements

This work was funded in part by NSERC (grant OGP0046506), China's Research and Development Program (grants 2016YFB1000902 and 2018YFB1003202), the NSFC (grant 61832019), and the Canada Research Chair program for M.L. N.H.T. was supported by the Mitacs Elevate Fellowship. The authors thank N. Keshav, K.P. Choi, and K. Xiong for discussions and proofreading of the manuscript.

Author contributions

M.L., B.S., and N.H.T. conceived the research idea. N.H.T. designed the model, implemented the software, and analyzed the results. R.Q. and X.C. contributed to the model design, software development, and data analysis. M.L., B.S., and L.X. supervised the research project. C.L., X.Z., and A.G. contributed to the data analysis. N.H.T., M.L., and R.Q. wrote the manuscript.

Competing interests

L.X., X.C., and B.S. are employees of Bioinformatics Solutions Inc., Waterloo, Ontario, Canada.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41592-018-0260-3.

Reprints and permissions information is available at www.nature.com/reprints.

Correspondence and requests for materials should be addressed to M.L.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© The Author(s), under exclusive licence to Springer Nature America, Inc. 2018

Methods

DeepNovo-DIA model. *Data preprocessing.* Because a DIA spectrum is highly multiplexed, it is important to use high resolution to distinguish fragment ions from different precursors that happen to have similar masses. In this study, we used 50 bins to represent 1.0 Da, that is, a resolution of 0.02 Da. We also defined a maximum mass value of 3,000.0 Da. Thus, each spectrum was represented by a vector of size 150,000, in which the mass of an ion corresponded to an index and the ion intensity was the vector value at that index. For the retention-time dimension, we fixed this number and selected those spectra closest to the feature's retention-time mean. If there were not enough spectra, we appended zeros. In this study, we used five spectra (the use of ten spectra led to minor improvements). We stacked the spectra along the retention-time dimension so that the middle one was the closest to the feature's retention-time mean (Fig. 1).

The five selected MS/MS spectra of a feature were stored in a matrix of size 5×150,000. To normalize the intensities, we divided the matrix element-wise by its maximum. We also extracted the MS1 intensity profile of a given feature at the respective retention times of those five MS/MS spectra. The resulting normalized 5×150,000 matrix together with the length-5 MS1 intensity profile vector were then fed to the DeepNovo-DIA model for de novo sequencing.

De novo sequencing framework. In general, the de novo sequencing framework is the same for DDA and DIA data, except that extra preprocessing is needed to add the retention-time dimension of DIA data. The framework is illustrated in Supplementary Fig. 9. The framework operates in a recurrent and beam-search fashion: at each iteration, the model predicts the next amino acid by conditioning on the output of previous steps and keeps track of only the top five candidate sequences. For each iteration, the input is a prefix, that is, a sequence including a 'start' symbol and the amino acids that have been predicted up to the current iteration. The output is a probability distribution over 26 candidates, including 20 amino acid residues, their modifications (oxidation (M) and deamidation (NQ)), and three special symbols ('start', 'end', and 'padding'). Similarly to the use of DeepNovo on DDA data[11], we used the knapsack search algorithm to limit our search space.

Given the input prefix, DeepNovo-DIA calculates the probability of the next amino acid on the basis of information extracted from two separate branches. In the first branch, called ion-CNN (Supplementary Fig. 10), DeepNovo-DIA first computes the prefix mass, that is, the total mass of the N-terminal and amino acids in the prefix. Next, DeepNovo-DIA tries to add each amino acid type to the current prefix and updates its mass accordingly. For each resulting candidate sequence, the corresponding masses of b ions and y ions are calculated. In the current implementation, we use eight ion types: b, y, b(2+), y(2+), b-H2O, y-H2O, b-NH3, and y-NH3. Given an ion mass, DeepNovo-DIA identifies its location on the intensity vector of feature-associated spectra. DeepNovo-DIA then extracts an intensity window of size 10 around the ion location. Thus, for each input prefix, DeepNovo-DIA computes a four-dimensional array of shapes (26, 8, 5, 10) and feeds it to ion-CNN together with the MS1 intensity profile, a vector of length 5. Ion-CNN then encodes the inputs into a vector h_{ion} of length 512 + 26. The structure of ion-CNN is explained in Supplementary Fig. 10 and in the following section.

The second branch, spectrum-CNN coupled with long–short-term memory (LSTM) (Supplementary Fig. 11), is designed to learn amino acid patterns of the peptide in association with the feature's spectra. In this branch, we directly feed the 5×150,000 normalized feature matrix into spectrum-CNN, and then use the output as the initial input to LSTM. At each step, LSTM takes the embedding vector of current amino acid as input. LSTM then outputs a vector h_{lstm} based on the input and the current hidden state, which contains information about the previous amino acids that it has seen. The structure of spectrum-CNN is explained in Supplementary Fig. 11 and in a later section.

Finally, DeepNovo-DIA concatenates (h_{ion}, h_{lstm}) together and feeds the result to a softmax layer with 26 neurons. The output is interpreted as a probability distribution on the 26 possible candidates of amino acids and tokens. The whole DeepNovo-DIA model is illustrated in Supplementary Fig. 9.

Ion-CNN. The role of ion-CNN is to learn (i) 3D shapes along m/z and retention-time dimensions of fragment ions and (ii) correlation between the precursor and its fragment ions (Fig. 1). The first input to ion-CNN is a five-dimensional fragment-ion intensity array of shape (128, 26, 8, 5, 10), where

- the first dimension is the batch size,
- the second dimension is the number of amino acids,
- the third dimension is the number of ion types,
- the fourth dimension is the number of associated spectra, and
- the fifth dimension is the model's window size.

Thus, for DIA data, we have the extra dimensionality of retention time where multiple associated spectra can be used to predict the peptide sequence of the feature. Moreover, the second input to ion-CNN, a two-dimensional array of shape (128, 5), is the MS1 intensity profile of the feature over its retention-time period.

Theoretically, all the true fragment ions of the feature should be correlated to the feature and to each other. The fragment ions should form 3D shapes along the m/z and retention-time dimensions (Fig. 1). We use three convolutional layers followed by one max pooling layer and one fully connected layer to learn those patterns (Supplementary Fig. 10). For each layer, we choose RELU as the activation function. Eventually, the fully connected layer outputs a matrix of shape (128, 512). To let the model use information of the correlation between the fragment ions and the feature, we also calculate the Pearson correlation between the feature and its fragment ions and concatenate that indicator with the output of the fully connected layer. In particular, for each five-dimensional fragment ion intensity array I, we define I_{frag} as the sum over the third and fifth dimensions. Thus, I_{frag} is a tensor of shape (128, 26, 5). Then, we repeat the MS1 intensity profile matrix 26 times and stack the results to get I_{prof} with size (128, 26, 5). Finally, we compute the correlation between I_{frag} and I_{prof} along their third axis. The resulting correlation tensor, with shape (128, 26), is then concatenated with the output of the fully connected layer. Our experiment showed that when the correlation feature was included, DeepNovo-DIA's amino acid accuracy improved by 5.4%. In addition, it was easier to get overfitting with DIA data than with DDA data. We use dropout layers after the final convolutional layer and the fully connected layer.

Compared with the DDA data, we found that more convolutional layers yielded better performance for DIA data. We also found that max pooling was very important to account for shift invariance, especially along the retention-time dimension.

Spectrum-CNN and LSTM. The model of spectrum-CNN coupled with LSTM is designed to learn amino acid sequence patterns of the peptide in association with the feature's spectra. We use spectrum-CNN to encode the intensity vector of the spectra and LSTM to decode the amino acids. This is similar to the idea of automatically generating a description for an image, where a convolutional neural network is used to encode, or 'understand', the image and a recurrent neural network is used to decode, or 'describe', the content of the image. The input to spectrum-CNN is a four-dimensional tensor of shape (128, 5, 150,000, 1) that is a batch of normalized feature matrices. Spectrum-CNN consists of one max pooling layer, two convolution layers, and one fully connected layer (Supplementary Fig. 11). It encodes the input into an array h_{init} of size (128, 512). Then, DeepNovo-DIA initializes the LSTM module with zero states and feeds h_{init} to LSTM to 'warm up' its hidden state (Supplementary Fig. 9).

Focal loss function. Previously in a DeepNovo model for DDA, we used cross-entropy loss as the loss function. For DIA, the presence of multiple peptides in the same spectrum inspired us to view de novo sequencing as a multi-label classification problem with dense signals, and hence to apply focal loss[17] as the suitable objective function for DIA. Our experiment shows that the switch to focal loss improved DeepNovo-DIA's performance considerably.

Lin et al. proposed focal loss to solve the class-imbalance issue in object detection[17]. The focal loss down-weights the contribution of easy predictions and puts more focus on hard predictions, and therefore could help to address the problems of noisy targets and class imbalance. In object-detection problems, the neural networks need to classify whether a patch of an image is an object or background. Because of the nature of this problem, most patches neural networks can see are background, and this causes problems for end-to-end training with cross-entropy loss. To deal with this problem, Lin et al. proposed a dynamically scaled cross-entropy loss that they named focal loss. For a binary classification problem, we denote $y \in \{0,1\}$ as the ground-truth class for a data point, and p as the model's predicted probability for class 1. Then the focal loss is defined by the following formula:

$$\text{Focal loss} = -(1-p_t)^{\gamma}\log(p_t)$$

where $p_t = p$ if y is class 1 and $p_t = 1 - p$ if y is class 0, and where γ is a hyperparameter greater than 1.

From the definition, we can see that, compared with cross-entropy loss, focal loss scales down the loss by a factor of $(1 - p_t)^{\gamma}$. This means that focal loss down-weights the contribution of easy examples (where $1 - p_t$ is small), and the model is likely to focus more on hard examples.

In our case, we found that the DeepNovo-DIA model also had a class-imbalance problem, as the frequency for amino acids varies a lot. Therefore, we suspected that focal loss could help us to better train the DeepNovo-DIA model. During training, we changed the activation function of the last layer from a softmax function to a sigmoid function, which led the model to give a probability between 0 and 1 for each of the 26 classes (note that here the sum of these 26 probabilities might not amount to 1). Then, for each class we computed the focal loss using the formula above, and used the average of those 26 losses as the final loss. At inference time, we switched the activation function back to softmax because we found that this led to better performance. Overall, our experiments show that the focal loss improved the amino acid accuracy by 20% on the plasma dataset.

Model training. We trained DeepNovo-DIA on a previously published[13] DIA dataset of urine samples from 64 subjects. The training dataset included 2,177,667 spectra, 202,114 labeled features, and 14,400 unique peptides (Supplementary Table 1).

We divided the data into three sets—training, validation, and testing—with ratios of 90%, 5%, and 5%, respectively. For de novo sequencing purposes, we made sure that the training, validation, and testing sets did not share common peptides. During the training process, we used 'early stopping'—that is, we periodically evaluated the model and saved it only if there was improvement on the validation set. We found that DIA data were overfitted more easily than DDA data were, and usually the training process stopped after about five epochs. To train the model, we used the Adam optimizer with the default hyperparameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and a fixed learning rate of 0.001. To prevent the gradient exploding problem, we clipped the gradient so that the global L2 norm of the gradient tensor was <5.

Data analysis. The lists of precursor features, DIA spectra, and de novo predictions that we used for data analysis have been uploaded to the MassIVE repository under accession number MSV000082368. Readers can re-run the de novo sequencing process from the given feature lists and spectra, or from the raw files (see instructions on GitHub or in the Supplementary Protocol). The raw files can be downloaded from the original publications (refs. [10,16]). In the following subsections, we explain how we analyzed the de novo predictions and obtained the results in our paper.

Precursor feature detection. For precursor feature detection from LC-MS maps, we used an existing peak caller from ref. [12]. Other peak callers, such as MaxQuant[18], can also be used in this role. The output of this step is a list of precursor features, each of which should include the following information: feature ID, *m*/*z*, charge, abundance level (area), retention-time center, and intensity values over the retention-time range. Moreover, given the *m*/*z* and retention-time range of a feature, we collected all MS/MS spectra so that they were within the feature's retention-time range, and their DIA *m*/*z* windows had to cover the feature's *m*/*z*. For example, the CSV file "testing_plasma.unlabeled.csv" in the testing data (folder "plasma") shows all precursor features that we detected from the plasma dataset. The file's columns include the following information:

- "spec_group," the feature ID; "F1:6427" means feature number 6,427 of fraction 1
- *m*/*z*, the mass-to-charge ratio
- *z*, the charge
- "rt_mean," the mean of the retention-time range
- "seq": the column is empty during de novo sequencing. In training mode, it contains the peptides identified by the in-house database search for training.
- "scans," a list of all MS/MS spectra collected for the feature as described above. The spectra's IDs are separated by a semicolon; "F1:101" indicates scan number 101 of fraction 1. The spectra's IDs can be used to locate the spectra in the MGF file "testing_plasma.spectrum.mgf."
- "profile," the intensity values over the retention-time range; the values are "time:intensity" pairs and are separated by semicolons; the time points align to the time of spectra in the column "scans."
- "feature_area," the precursor feature area estimated by the feature detection

In-house database searching. To generate the training data for DeepNovo-DIA, we built an in-house database search tool for DIA data. We used an approach similar to DIA-Umpire[9]. First, from each precursor feature and its associated MS/MS spectra, we generated a pseudo-spectrum. In particular, we calculated the Pearson correlation coefficient between the LC eluting profiles of the precursor and MS/MS fragment ions. Then, we selected fragment ions with Pearson correlation coefficients greater than 0.6 and used up to 500 of the most correlated ones to form the pseudo-spectrum.

The pseudo-spectra and corresponding precursor information such as *m*/*z* and charge were then searched against the Swiss-Prot human database. This step can be done with any conventional DDA database search engine; here we used PEAKS DB[12]. We used common parameter settings such as trypsin digestion, fixed modification C (carbamidomethylation), precursor mass tolerance 30 p.p.m., and fragment mass tolerance 0.02 Da for the plasma dataset. For human leukocyte antigen (HLA) datasets, we used non-enzyme digestion, no modification, 20 p.p.m., and 0.05 Da. The peptides identified at a 1% FDR cutoff were then assigned to the corresponding precursor features and were used as labels for training. For example, the CSV file "testing_plasma.feature.csv" in the testing data (folder "plasma") shows all labeled features identified from the plasma dataset. The database FASTA file is included in the Supplementary Software.

Post-processing analysis. For each labeled feature, the de novo peptide predicted by DeepNovo-DIA was compared to the ground-truth sequence identified by the database search. A simple way to do this is on the basis of exact sequence matching. However, it is very common for de novo peptides to have one or two sequencing errors such as swapping, or different amino acid combinations with the same mass. Hence, we calculated sequencing accuracy on the basis of the alignment of the mass fragments[11]. We measured the sequencing accuracy at the amino acid level, that is, the ratio of the total number of matched amino acids to the total length of predicted peptides, and at the peptide level, that is, the fraction of fully matched peptides.

We provide a script in the Supplementary Software to calculate DeepNovo-DIA accuracy ("deepnovo_dia_script_test.py"). The script compares the file of labeled features (e.g., "testing_plasma.feature.csv") and the output file from DeepNovo-DIA (e.g., "testing_plasma.unlabeled.csv.deepnovo_denovo"). It searches for overlapping features and calculates accuracy on those features. The results are printed out to the file "testing_plasma.unlabeled.csv.deepnovo_denovo.accuracy." The script can be used to reproduce the results in Fig. 2a–d.

The confidence score of a peptide sequence is the sum of its amino acids' scores. The score of each amino acid is the log of the output probability distribution—that is, the final softmax layer of the neural network model—at each sequencing iteration. The score was trained using only the training dataset. When applying the process to a new specific dataset, one might want to select a cutoff to filter the de novo results of that dataset. This is similar to the case of a database search: when a 1% FDR is set, the cutoff score changes from one dataset to another. However, there is no such target–decoy method to estimate FDR for de novo sequencing. Hence, we compared database-search and de novo results on the basis of their overlapping features, calculated de novo accuracy, and plotted the distribution of de novo confidence scores with respect to de novo accuracy (Fig. 2b). Then, from the distribution, we selected a cutoff for the de novo confidence score so that the de novo accuracy was 90% at the amino acid level on the overlapping features. Finally, we applied that cutoff to de novo results for all features.

We provide a script in the Supplementary Software to filter high-confidence predictions from DeepNovo-DIA ("deepnovo_dia_script_select.py"). The script was used to filter the DeepNovo-DIA high-confidence results that we have described in this paper. The results are printed out to the file "testing_plasma.unlabeled.csv.deepnovo_denovo.top90."

The selection of high-confidence de novo predictions does not provide a way to control the FDR. Thus, de novo peptides can be validated by means of the following approach. We augmented the original database FASTA file with the de novo peptides identified by DeepNovo-DIA. Then, we re-ran the database search using the new FASTA file; other parameters remained unchanged. Finally, we selected only de novo peptides that passed 1% FDR after the search had been re-run. We performed this analysis with our in-house database search and Spectronaut. We found that about 56.8% (1,143/2,011 = 56.8%) of de novo peptides passed 1% FDR for the plasma dataset (Supplementary Note 1). Such peptides should be supported by significant peptide–spectrum matches and coelution profiles between precursors and fragment ions.

Validation of de novo results with PECAN, OpenSWATH, and Spectronaut. We used three different database search tools to validate the de novo results of DeepNovo-DIA.

For the plasma dataset, we compared DeepNovo-DIA to PECAN and Spectronaut. PECAN results were downloaded from the original publication[10]. We ran Spectronaut using their directDIA workflow with trypsin digestion, fixed modification C (carbamidomethylation), and 1% FDR; other parameters were defaults.

For the HLA datasets, we compared DeepNovo-DIA to OpenSWATH and Spectronaut. OpenSWATH results were downloaded from the original publication[16]. We calculated and filtered peptides at 1% FDR from their results, and used those peptides for comparison. We ran Spectronaut with the directDIA workflow with non-enzyme digestion and 1% FDR; other parameters were defaults.

We compared DeepNovo-DIA to PECAN, OpenSWATH, and Spectronaut on the basis of unique peptides. For such a comparison, DeepNovo-DIA had to accurately predict the full sequence of a peptide in order for it to be considered a match.

Reporting Summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Software availability. DeepNovo-DIA is implemented in Python, using the TensorFlow library for neural networks. The software and documentation are provided in the Supplementary Software and the Supplementary Protocol, as well as on GitHub (https://github.com/nh2tran/DeepNovo-DIA).

Data availability

Data and a pretrained model are publicly available in the MassIVE repository under accession number MSV000082368. Source data for Fig. 2 are available online.

References

17. Lin, T. Y., Goyal, P., Girshick, R., He, K. & Dollár, P. Focal loss for dense object detection. *arXiv* Preprint at https://arxiv.org/abs/1708.02002 (2017).
18. Tyanova, S., Temu, T. & Cox, J. *Nat. Protoc.* **11**, 2301–2319 (2016).

nature research

Corresponding author(s): Ming Li

Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see Authors & Referees and the Editorial Policy Checklist.

Statistical parameters

When statistical analyses are reported, confirm that the following items are present in the relevant location (e.g. figure legend, table legend, main text, or Methods section).

n/a	Confirmed
☒	☐ The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☒	☐ An indication of whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☒	☐ The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐ A description of all covariates tested
☒	☐ A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☒	☐ A full description of the statistics including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☒	☐ For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐ For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐ For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	☐ Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated
☒	☐ Clearly defined error bars *State explicitly what error bars represent (e.g. SD, SE, CI)*

Our web collection on statistics for biologists may be useful.

Software and code

Policy information about availability of computer code

Data collection	No software was used.
Data analysis	DeepNovo is freely available on GitHub: https://github.com/nh2tran/DeepNovo-DIA. Scripts for data analysis are included in the supplementary materials. DeepNovo source code can be made available to editors and reviewers upon request. For PEAKS DB search, we used PEAKS Studio version 8.5. For Spectronaut, we used version 11. For BLAST, we used BLASTX on the NCBI web server, version August 2018. For PECAN and OpenSWATH, we used the results reported in the original studies (references 10 and 16). We did not run the softwares ourselves.

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors/reviewers upon request. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

Data is available on MassIVE repository, accession number MSV000082368.
Data can also be downloaded from https://github.com/nh2tran/DeepNovo-DIA.

Field-specific reporting

Please select the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/authors/policies/ReportingSummary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	We used datasets that were published previously in other studies (references 10 and 16). For database search, we used all peptide identifications at 1% False Discovery Rate. For training, validation, and testing, we partitioned data according to the ratios 90%, 5%, and 5%, respectively. Those settings are standard in proteomics and machine learning research.
Data exclusions	No data were excluded.
Replication	The results, including figures and tables, can be reproduced by running the provided software and scripts on the testing datasets.
Randomization	The data were randomly partitioned into training, validation, and testing sets according to the ratios 90%, 5%, and 5%, respectively. The testing datasets were not used in model development.
Blinding	Blinding was not relevant to our study because the data were randomized. The only control was that the testing datasets were not used in model development so that the model was not biased towards the training data.

Reporting for specific materials, systems and methods

Materials & experimental systems

n/a	Involved in the study
☒	☐ Unique biological materials
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology
☒	☐ Animals and other organisms
☒	☐ Human research participants

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging