nature portfolio

Peer Review File

An end-to-end deep learning method for mass spectrometry data analysis to reveal disease-specific metabolic profiles

Corresponding Author: Professor Weizhong Li

Version 0:

Reviewer comments:

Reviewer #1

(Remarks to the Author)
Deng et al have developed a deep learning-based pipeline for classification and explainable downstream analysis of raw mass spectrometry metabolomics data. DeepMSProfiler bypasses the need for pre-processing/annotation, instead using the mzML files directly as input to an ensemble model, achieving high prediction accuracy and AUC when discriminating between serum samples of lung adenocarcinoma, benign nodules, and healthy donors. The workflow allows determination of which features contribute to class discrimination which the authors then interpret in terms of inferred biological networks and pathways.

It is perhaps surprising that a raw-data-to-outcome-prediction deep learning approach has not already been published in the metabolomics field; however I am not aware of any such publications. DeepMSProfiler therefore addresses a gap, offering a promising pipeline for disease classification, bypassing the need for pre-processing, batch correction, and annotation which are key bottlenecks in standard metabolomics data analysis. The manuscript is clearly written, but there are a few concerns which the authors should consider addressing prior to publication.

Major concerns:
1. For the work to be useful, the model needs to be applicable to other LCMS metabolomic datasets. The present study employs two datasets, an UPLC-MS serum dataset and a lipidomics cell-line dataset from the Cancer Cell Line Encyclopaedia database. The majority of the work is evaluated on the lung adenocarcinoma dataset, where the model performs extremely well. The authors performed statistical correlation analysis to identify confounding by clinical features, however this only accounts for linear relationships, and does not necessarily mean two features are independent. The CCLE dataset does not appear to be described in the methods. Given that the workflow takes raw data it would be important to give analytical details of the CCLE data. Given the importance of transferability for this work, including another dataset using the same workflow would help indicate how generalisable the model performance is. For future users, discussion or analysis showing what the minimum sample size for robust DeepMSProfiler performance would be beneficial. The authors should also clarify whether the model needs to be re-trained when applied to new data, and what ther requirements (data & compute) are for this.
2. The feature contribution heatmaps are a key output of DeepMSProfiler. Importantly, the authors should clarify in the Methods how the peak contribution heatmaps were derived from the RISE heatmaps (Fig. 5c to Fig 5d/e/f). The authors emphasise the ‘high resolution’ of the RISE technique, but the heatmaps actually look relatively low resolution for interpretation in terms of individual MS peaks. Is the network capturing signals from independent metabolites, or larger regions of the RT-mz domain?
3. Because the RT shift is not corrected in the model, the authors used only m/z to find the corresponding peaks and used this data as input to PIUMet. How confident can readers be that these peaks identified are true positives? Clarity could be improved in the figures (i.e. 5c major axes could be labelled with m/z and RT). Some discussion on how the threshold of 0.7 for feature contribution applies to other datasets would be beneficial for future users.
4. An interesting analysis was done on the ability of DeepMSProfiler to correct for batch effects. Further discussion could be added explaining how this is achieved via progression through the hidden layers (i.e. how does the ‘automatic removal of batch effects’ work in this context). Is it possible to visualise what batch effect is being removed, at the level of RT-mz profiles? Additionally, although the authors showed that the model corrected for 10 batches, the data was collected from 3 distinct hospitals, where additional batch effect would be expected. Some more discussion or analysis on this aspect would be of interest.
5. How is the ensemble model run? Is the data resampled for each model? How? I don’t see any reference to ensemble approach in the methods. This is a key part of the workflow and highlighted by the authors as a strength so needs to be described much more clearly.
6. The biological interpretation portion of the work applied existing methods (e.g. PIUmet and pathway analysis) to place

important features into a biological context. Further detail is required to familiarise the readers with the PIUmet method, in terms of how it creates the sub-networks using both annotated and un-annotated metabolites, and how robust the inference is. Further, it is not clear in the methods whether the proteins inferred by PIUmet, or also the metabolites (both inferred and annotated) were used as input to pathway analysis. It is key that the authors report the specific method of pathway analysis applied via MetaboAnalyst, as well as the database version and background set (in the case of over-representation analysis). Authors should discuss how relevant the pathway analysis results are, given that the inputs may have false positive identifications (mz matching only).
7. I was unable to test the code via Code Ocean (https://codeocean.com/capsule/2328223/tree) with the following error message: 'You don't have permission to access this Capsule. Please contact your administrator for assistance.' Furthermore, on the GitHub documentation there is no indication on how to produce RISE contribution heatmaps, which are an essential part of the biological interpretation. There is a function to produce a binary classification confusion matrix, but there doesn't seem to be anything further to produce a coloured feature importance map such as those in Fig 5. To make this tool readily accessible, the authors should consider expanding on the documentation provided in the GitHub README.md file (i.e. a ReadTheDocs page) so that users can easily find how to reproduce these important outputs.

Minor concerns:
8. The application of deep learning models to raw metabolomics data is an interesting area. It would be helpful if the authors could elaborate in the introduction on where this has been attempted before in previous studies and how this work builds on existing research. Has anyone else tried to perform disease classification using deep learning (or indeed other ML approaches) applied to raw mzML files, for example? How is the interpretation derived from DeepMSProfiler different to other methods?
9. L683: Mass-to-charge ratio and some substance identification information of these metabolites and their classification contribution scores were used as input data. What is meant here by 'substance identification information'?
10. Fig 4f: it is not clear which machine learning models are being compared to DeepMSProfiler, perhaps this could be clarified in the figure legend.
11. Further details are required in the methods about which machine learning library was used to implement the conventional models (SciKitLearn or other?) for purposes of reproducibility.

Reviewer #2

(Remarks to the Author)
I co-reviewed this manuscript with one of the reviewers who provided the listed reports. This is part of the Nature Communications initiative to facilitate training in peer review and to provide appropriate recognition for Early Career Researchers who co-review manuscripts.

Reviewer #3

(Remarks to the Author)
The paper presents an explainable end-to-end deep learning method called DeepMSProfiler for the direct analysis of mass spectrometry data to reveal disease-specific metabolic profiles, which enables the analysis of raw metabolic signals with high accuracy and reliability. The method was applied to differentiate the metabolomic profiles of lung adenocarcinoma, benign lung nodules, and healthy individuals using serum samples. The results demonstrate that DeepMSProfiler successfully differentiated the different groups and detected early-stage lung adenocarcinoma with high accuracy. The method also overcame inter-hospital variability and effects of unknown metabolite signals. Furthermore, DeepMSProfiler provided interpretability and enabled the direct access to disease-related metabolite-protein networks. The method was also applied to lipid metabolomic data to unveil correlations of important metabolites and proteins.

1. Given that the Transformer model excels at processing sequential information, the reviewer suggest that the authors incorporate experiments utilizing Transformer architecture. This would provide a more comprehensive evaluation of the proposed method.
2. In Figure 3(d), it seems that in the initial four layers, DeepMSProfiler cannot effectively differentiate data across various batches, but in the last layer, it successfully eliminates batch effects. Could the authors elucidate the reason behind this observation? Supporting results would also be needed.
3. More recent advances (View, 2023, 4, 20220038; Exploration 2022, 2, 20210222) should be included and discussed to highlight this work.
4. DeepMSProfiler employs an ensemble of five distinct models. It is recommended that the authors conduct further experiments to explore how each model contributes to the overall results, thereby providing deeper insights into the ensemble's effectiveness.

Author Rebuttal letter:

Dear Reviewers,

Thanks very much indeed for your valuable comments and helpful suggestions.
We have now revised our manuscript accordingly. The major modifications in this

revision include the followings.

1) In the Introduction part, we described more about the previous attempts in combining machine learning with LC-MS for in vitro diagnoses of diseases.
2) We applied DeepMSProfiler on a new dataset for colon cancer and achieved an high performance. Please see the additional section of âApplication of model in colon cancerâ in the Results part.
3) We expanded the layer visualization to demonstrate the progression of batch effect removal in Figure 3d and Supplementary Figure 7. The âInsensitivity to batch effectsâ section in the Results part and the Figure 3 legend were updated accordingly.
4) We added a section to detail the data collection of the CCLE dataset and other relevant public data resource. Please see the additional âPublic dataset collectionâ section of the Methods part.
5) We complemented the methods with an explanation for the specific process of going from the feature contribution heatmap to the signal peak contribution. Please see the âFeature contributionsâ section in the Methods part.
6) To detail how the ensemble model runs, a new section of âEnsemble Strategyâ has been added in the Methods part.
7) We added more details about using PIUMet in the âNetwork analysis and pathway enrichmentâ section in the Methods part.
8) Figure 3, 4 and 5 have been updated slightly to enhance image quality and clearness, and we also supplemented new Supplementary Figure 2, 4, 7 and 9.

For code and data availability, please see the sections of âCode availabilityâ and âData availabilityâ. More details are provided in the GitHub README.md file for user to run our tool. We have created an account for the reviewers to access our data in Code Ocean before the article publication. Please visit the URL https://codeocean.com/capsule/2328223/tree, and log on the account with username âyd849@bath.ac.ukâ and password âcode4papeâ. Please note that, the code on GitHub served as an easy-to-use tool for running DeepMSProfiler, and the Code Ocean repository contains our original data and scripts that can be used to replicate our results.

A point-by-point response to your comments is provided below, with your review points in black and our response texts in red colour. Please also note that the major and minor changes in the revised manuscript are also highlighted in red colour.

Your comments and suggestion have enabled us to substantially improve our manuscript. Many thanks again for your time and efforts. We hope that we have satisfactorily addressed all issues and concerns, and that our revised manuscript meets the publication standards.

Yours faithfully,

Weizhong Li, Ph.D.

REVIEWER #1

Remarks to the Author:
Deng et al have developed a deep learning-based pipeline for classification and explainable downstream analysis of raw mass spectrometry metabolomics data. DeepMSProfiler bypasses the need for pre-processing/annotation, instead using the mzML files directly as input to an ensemble model, achieving high prediction accuracy and AUC when discriminating between serum samples of lung adenocarcinoma, benign nodules, and healthy donors. The workflow allows determination of which features contribute to class discrimination which the authors then interpret in terms of inferred biological networks and pathways.
It is perhaps surprising that a raw-data-to-outcome-prediction deep learning approach has not already been published in the metabolomics field; however I am not aware of any such publications. DeepMSProfiler therefore addresses a gap, offering a promising pipeline for disease classification, bypassing the need for pre-processing, batch correction, and annotation which are key bottlenecks in standard metabolomics data analysis. The manuscript is clearly written, but there are a few concerns which

the authors should consider addressing prior to publication.

Response:
Thanks very much for your valuable comments and positive feedback. We have addressed your concerns in this revision and responded point-by-point as underneath.

Major Concerns:
Major Concern 1 â Point 1:
1. For the work to be useful, the model needs to be applicable to other LCMS metabolomic datasets. The present study employs two datasets, an UPLC-MS serum dataset and a lipidomics cell-line dataset from the Cancer Cell Line Encyclopaedia database. The majority of the work is evaluated on the lung adenocarcinoma dataset, where the model performs extremely well.

Response:
To test other MS data with our workflow, we further downloaded a new colon cancer dataset from the MetaboLights database and then applied DeepMSProfiler on this new dataset. In the independent validation dataset of the new data, the model achieved an accuracy of 97.9%, a sensitivity of 97.5%, and a specificity of 100%.
For more information about the model application in colon cancer, please see the additional section of âApplication of model in colon cancerâ in the Results part.

Major Concern 1 â Point 2:
The authors performed statistical correlation analysis to identify confounding by clinical features, however this only accounts for linear relationships, and does not necessarily mean two features are independent.
Response:
The main purpose of the statistical correlation analysis is to identify non-independent confounding by clinical characteristics, and to apply appropriate constraint processing to reduce their impact in subsequent experiments. Three different statistical correlation analysis were used, including Pearsonâs correlation, Spearmanâs rank correlation, and Kendallâs rank correlation (Supplementary Table S3). Pearsonâs correlation coefficient was preferred to find linear correlations. Spearmanâs and Kendallâs rank correlation coefficients were used to discover non-linear correlations. According Supplementary Table 3, compared to other factors, 'diameter' is significant (p-values < 0.05) in all three correlation analyses, so it is believed to be non-independent. And then we applied distributional constraints specifically to 'diameter' in the subsequent analyses. Other factors did not show significant non-independence, but we still compared the performance of different age groups in the subsequent analyses.
In this revision, we mentioned the use of the three correlation coefficients in the âStatistical analysisâ section of the Methods part.

Major Concern 1 â Point 3:
The CCLE dataset does not appear to be described in the methods. Given that the workflow takes raw data it would be important to give analytical details of the CCLE data.

Response:
We have added a section to detail the data collection of the CCLE dataset and other relevant public data source. Please see the âPublic dataset collectionâ section of the Methods part for details.

Major Concern 1 â Point 4:
Given the importance of transferability for this work, including another dataset using the same workflow would help indicate how generalisable the model performance is.

Response:
As mentioned in the response to Major Concern 1 - Point 1, we further downloaded a new colon cancer dataset from the MetaboLights database and then applied DeepMSProfiler on this new dataset. In the independent validation dataset of the new data, the model achieved an accuracy of 97.9%, a sensitivity of 97.5%, and a specificity of 100%.
For more information about the model application in colon cancer, please see the additional section of âApplication of model in colon cancerâ in the Results part.

Major Concern 1 â Point 5:
For future users, discussion or analysis showing what the minimum sample size for

robust DeepMSProfiler performance would be beneficial.
Response:
It is difficult to tell the exact minimum sample size, but about 90 (=30 for cancer + 30 for benign nodes + 30 for healthy) might be good for the classification task. Because we have trained the model in only 1 batch of 90 samples and achieved good performance. Usually, researchers in deep learning would agree that hundreds of samples are needed for a good training for deep learning models, but it is dependent to the different scenarios of studies. Therefore, we collected nearly 900 samples in the beginning of this study.

Major Concern 1 â Point 6:
The authors should also clarify whether the model needs to be re-trained when applied to new data, and what their requirements (data & compute) are for this.

Response:
If users try newly acquired data from the same mass spectrometer and the same parameters, it is better to perform transfer learning on the existing model to fine-tune the model. If users try new data from different mass spectrometers or different sequencing methods, it is necessary to retrain the model.
In this revision, we mentioned the model re-training in dealing with new data in the âApplication of model in colon cancerâ section of the Results part and the âPublic dataset collectionâ section of the Methods part.

Major Concern 2 â Point 1:
2. The feature contribution heatmaps are a key output of DeepMSProfiler. Importantly, the authors should clarify in the Methods how the peak contribution heatmaps were derived from the RISE heatmaps (Fig. 5c to Fig 5d/e/f).

Response:
Using RISE, we can determine the characteristic contributions of RT and m/z for each sample according to its true category. The feature contribution heatmap uses RT as the horizontal axis and m/z as the vertical axis to show the feature contribution of different positions of each sample. The average feature contribution of all samples correctly predicted to be of their true category is taken as the feature contribution of the category. At the same time, by performing peak extraction in the previous steps, we determined the RT value range and the m/z median value for each signal peak. The characteristic contribution associated with the RT and median m/z coordinates is then identified as the distinctive contribution of the signal peak.
We complemented the methods with the above explanation for the specific process of going from the feature contribution heatmap to the signal peak contribution. Please see the âFeature contributionsâ section in the Methods part.

Major Concern 2 â Point 2:
The authors emphasise the âhigh resolutionâ of the RISE technique, but the heatmaps actually look relatively low resolution for interpretation in terms of individual MS peaks.

Response:
Compared to the feature contribution heatmap obtained by gradient-based methods such as GradCAM, the feature contribution heatmap obtained by RISE has a higher resolution. Due to the principle of RISE, the feature contribution heatmap obtained by RISE has the same size as the original input data, which is 1024Ã1024 for the example model, and can distinguish a large number of signal peaks. However, the output obtained by methods such as GradCAM corresponds to the output of the last convolutional layer, which is 10Ã10 for the example model, and cannot meet the requirements for distinguishing most signal peaks.

Major Concern 2 â Point 3:
Is the network capturing signals from independent metabolites, or larger regions of the RT-mz domain?

Response:
The feature extraction module is based on a convolutional neural network to perform classification tasks by extracting category-related features. A certain range of retention time and the m/z value included in this interval were captured as an individual signal to obtain a two-dimensional matrix of 1024Ã1024 ion intensities. The individual RT length we selected was 0.016 minutes, with which we were able to establish a high resolution-network. After the above step, the metabolite-protein

networks were then analysed with independent metabolites. The results of metabolite-protein networks have greatly improved the interpretability of our method.

Major Concern 3 â Point 1:
3. Because the RT shift is not corrected in the model, the authors used only m/z to find the corresponding peaks and used this data as input to PIUMet. How confident can readers be that these peaks identified are true positives?

Response:
Metabolite identification is a major challenge in the metabolomic research area. The primary goal of the current study was to develop a novel method to classify different disease state based on the deep learning end-to-end model. As deep learning methods are usually perceived as âblack-boxâ processes, the purpose of PIUMet analysis was to explore the possible interpretability of the metabolomic profile established by our model DeepMSProfiler, rather than identify accurate metabolites. Nevertheless, we were still able to validate a good number of the metabolites discovered by PIUMet using authentic chemical standards. Furthermore, most of these metabolites belong to the highest ranked pathways analyzed by MetaboAnalyst and they were consistent with the metabolomic alterations in lung cancer previously reported. The results of these metabolites were shown in Supplementary Table 16.
Therefore, in order to specify the purpose of PIUMet and pathway analysis, we modified the text in the section of âMetabolomic profiles in lung adenocarcinoma, benign nodules, and healthy individualsâ in the Results part.

Major Concern 3 â Point 2:
Clarity could be improved in the figures (i.e. 5c major axes could be labelled with m/z and RT).

Response:
We have followed the suggestion to improve the clarification in Fig. 5.

Major Concern 3 â Point 3:
Some discussion on how the threshold of 0.7 for feature contribution applies to other datasets would be beneficial for future users.

Response:
Thanks a lot for the suggestion. We have discussed more about the threshold in the âMetabolomic profiles in lung adenocarcinoma, benign nodules, and healthy individualsâ section of the Results part. As observed in Figure 5b, by retaining metabolic signals with a contribution score above 0.7, the overall accuracy is around 0.8, which manages to maintain an efficient classification impact.
Selecting a lower threshold could potentially improve the classification accuracy of the model, but could also result in the inclusion of too many metabolic signals, diluting the significance of key metabolic peak features. Conversely, choosing a higher threshold may result in the inclusion of metabolic signal peaks that are not accurately classified and are not representative of disease relevance.

Major Concern 4 â Point 1:
4. An interesting analysis was done on the ability of DeepMSProfiler to correct for batch effects. Further discussion could be added explaining how this is achieved via progression through the hidden layers (i.e. how does the âautomatic removal of batch effectsâ work in this context).

Response:
We have expanded the layer visualization to demonstrate the progression of the automatic removal of batch effect in Figure 3d and Supplementary Figure 7. DenseNet121 is a deep neural network with 431 layers, of which 120 are convolutional layers (Supplementary Table 15). The fourth and fifth layers shown in Figure 3d refer to the output of the fourth dense connected module and the output of the fifth dense connected module. There are 112 layers between them. The modified Figure 3d and the new Supplementary Figure 7 illustrate the intermediate change process from the output of the fourth closely connected module to the output of the fifth closely connected module.
Actually, Figure 3e already illustrates that the closer to the output layer, the less relevant the data is to batch labels and the more relevant to class labels. This alos explains how the batch effect removal is achieved via progress through hidden layers (Figure 3d).

The âInsensitivity to batch effectsâ section in the Results part and the legend of Figure 3 have been updated to mention the above information. And more details about the progression visualization can be found in Supplementary Figure 7.

Major Concern 4 â Point 2:
Is it possible to visualise what batch effect is being removed, at the level of RT-mz profiles?

Response:
It is difficult to visualise the batch effect removal at the level of RT-m/z profiles. DeepMSProfiler progressively removes batch effects through neural network layers and improves category correlation. The data output from the non-linear neural network layer is already deformed data and is no longer a one-to-one correspondence with the original data. Traditional methods correct peak areas using reference samples. Regardless of the traditional Ref-M method or the automated process of DeepMSProfiler, once the data is deformed, the RT-m/z coordinate information no longer exists in the output data.
However, as mentioned above, we have expanded the layer visualization at the level of low-dimension space to demonstrate the progression of the automatic removal of batch effect in Figure 3d and Supplementary Figure 7.

Major Concern 4 â Point 3:
Additionally, although the authors showed that the model corrected for 10 batches, the data was collected from 3 distinct hospitals, where additional batch effect would be expected. Some more discussion or analysis on this aspect would be of interest.

Response:
Samples of batch 1-6 and 9-10 were obtained from the Sun Yat-Sen University Cancer Centre, and samples of batch 8 came from the First Affiliated Hospital of Sun Yat-Sen University. Samples of batch 7 were a mixture from three different hospitals. Among them, lung cancer samples came from the Affiliated Cancer Hospital of Zhengzhou University, lung nodule samples came from the First Affiliated Hospital of Sun Yat-Sen University, and healthy samples came from the Sun Yat-Sen University Cancer Centre. 100 healthy human serum samples used as reference during conventional procedures were all from the Sun Yat-Sen University Cancer Centre, which might be the main reason why it is difficult to correct batch effect for batch 7-9.
We have mentioned the above information in the middle of the âInsensitivity to batch effectsâ section of the Results part.
Major Concern 5:
5. How is the ensemble model run? Is the data resampled for each model? How? I donât see any reference to ensemble approach in the methods. This is a key part of the workflow and highlighted by the authors as a strength so needs to be described much more clearly.

Response:
We do agree that the ensemble model is a key part of the workflow, since it avoids overfitting and overcomes the âbackground categoryâ phenomenon which is found the single models (Fig. 5a&b). DeepMSProfiler consists of several trained end-to-end sub-models as an ensemble model, where the average of the classification prediction probabilities of the samples from all sub-models was used as the final prediction probability for classification. The ensemble model calculated a score vector of length 3 in each of the three classifications, and the category with the maximum score was selected as the predicted classification result. Each end-to-end sub-model was trained on the discovery dataset. The architecture of each sub-model is the same, but some hyperparameters are different. Two different learning rates of 1e-3 and 1e-4 were used. Each sub-model participated fairly in the final prediction result without setting a specific weight. The independent testing dataset was not used in model training and hyperparameter selection. And the datasets for model training were not used in performance evaluation. Therefore, the data used in each model is consistent.
The above explanation on the ensemble model has been added into the âEnsemble Strategyâ section in the Methods part.

Major Concern 6 â Point 1:
6. The biological interpretation portion of the work applied existing methods (e.g. PIUmet and pathway analysis) to place important features into a biological context. Further detail is required to familiarise the readers with the PIUmet method, in terms of how it creates the sub-networks using both annotated and un-annotated metabolites, and how robust the inference is. Further, it is not clear in the methods whether the proteins inferred by PIUmet, or also the metabolites (both inferred and annotated)

were used as input to pathway analysis.

Response:
PIUMet built disease-related metabolite-protein networks based on the prize-collecting Steiner Forest algorithm. First, PIUMet integrated iRefIndex (v13), HMDB (v3) and Recon 2 databases to obtain the relationship between m/z, metabolites and proteins, and generated an overall metabolite-protein network. The edges were weighted by the confidence level of the correlation reported in these databases. The disease-related metabolite peaks obtained by DeepMSProfiler were matched to the metabolite nodes of the overall network by their m/z, and directly to the terminal metabolite nodes of the overall network after annotation. The corresponding feature contributions obtained by DeepMSProfiler served as prizes for these metabolite nodes. The network structure was then optimised using the prize-collecting Steiner Forest algorithm to minimise the total network cost and connect all terminal nodes, thereby removing low-confidence relationships and obtaining disease-related metabolite subnetworks. Then, disease-related metabolites and proteins were used to analyse their pathways. These hidden metabolites and proteins from PIUMet were then processed for KEGG pathway enrichment analysis using MetaboAnalyst.
We have added more details about using PIUmet in the main text. Please see the âNetwork analysis and pathway enrichmentâ section in the Methods part.

Major Concern 6 â Point 2:
It is key that the authors report the specific method of pathway analysis applied via MetaboAnalyst, as well as the database version and background set (in the case of over-representation analysis). Authors should discuss how relevant the pathway analysis results are, given that the inputs may have false positive identifications (mz matching only).

Response:
Considering that the primary aim of our study is to establish a classifier model based on end-to-end approach, we did the pathway analysis of the network found by PIUMet in order to test the interpretability and reliability of DeepMSProfiler developed by deep learning method. Indeed, the major metabolic alterations we discovered in lung cancer were consistent with previous reports. For instance, targeting serine and glycine for one-carbon metabolism as therapeutic strategy in non-small cell lung cancer (Reference 54-57 in main text), the aberrant bile acid metabolism in invasive lung adenocarcinoma (Reference 14), and the decrease of tryptophan metabolism intermediates in early-stage lung adenocarcinoma (Reference 53).
More discussion was included at the end of the 4th paragraph of the Discussion part.

Major Concern 7 â Point 1:
7. I was unable to test the code via Code Ocean (https://codeocean.com/capsule/2328223/tree) with the following error message: âYou don't have permission to access this Capsule. Please contact your administrator for assistance.â

Response:
Sorry for the previous inconvenience in using the Code Ocean data. We have created an account for the reviewers to access our data capsule in Code Ocean before the article publication. The account details are listed as below.
URL: https://codeocean.com/capsule/2328223/tree
Username: yd849@bath.ac.uk
Password: code4paper
Major Concern 7 â Point 2:
Furthermore, on the GitHub documentation there is no indication on how to produce RISE contribution heatmaps, which are an essential part of the biological interpretation.
There is a function to produce a binary classification confusion matrix, but there doesnât seem to be anything further to produce a coloured feature importance map such as those in Fig 5.

Response:
We have described more on how to produce and visualise the feature results (the RISE contribution heatmaps) in the GitHub README.md file.

Major Concern 7 â Point 3:

To make this tool readily accessible, the authors should consider expanding on the documentation provided in the GitHub README.md file (i.e. a ReadTheDocs page) so that users can easily find how to reproduce these important outputs.

Response:
As mentioned above, more details have been provided in the GitHub README.md file for user to run our tool and reproduce outputs. Please note that, the code on GitHub served as an easy-to-use tool for running DeepMSProfiler, and the Code Ocean repository contains our original LC-MS data and scripts that can be used to replicate the results in the article.
For the reviewers to access the full data in Code Ocean before the article publication, please use the following URL and account information.
URL: https://codeocean.com/capsule/2328223/tree
Username: yd849@bath.ac.uk
Password: code4paper

Minor Concerns:
Minor Concern 1:
8. The application of deep learning models to raw metabolomics data is an interesting area. It would be helpful if the authors could elaborate in the introduction on where this has been attempted before in previous studies and how this work builds on existing research. Has anyone else tried to perform disease classification using deep learning (or indeed other ML approaches) applied to raw mzML files, for example? How is the interpretation derived from DeepMSProfiler different to other methods?

Response:
Actually, several previous studies have combined machine learning with LC-MS for in vitro diagnosis and improved the efficiency of LC-MS data analysis. For example, Huang et al conducted machine learning to extract serum metabolic patterns from laser desorption/ionization mass spectrometry to detect early-stage lung adenocarcinoma (Reference 11 in the main text); Chen et al adopted machine learning models to conduct targeted metabolomic data analysis to identify non-invasive biomarker for gastric cancer diagnosis (Reference 17); Shen et al developed a deep-learning-based Pseudo-Mass Spectrometry Imaging method and applied it in the prediction of gestational age of pregnant women, as well as the diagnosis of endometrial cancer and colon cancer (Reference 18). However, these studies still face challenges such as batch effects and unknown metabolites in metabolomics (Reference 7). Consequently, a new analytical approach is urgently needed to overcome the experimental bottlenecks and to reveal disease-associated profiles comprising both identified and unknown components derived from LC-MS peaks.
The above description has been added into the Introduction part.
Regarding the difference to the above methods, our DeepMSProfiler has several advantages, such as more comprehensive metabolic signals, better performance, and higher interpretability. These advantages allow DeepMSProfiler to infer disease-related metabolic signals from the LC-MS raw data, and its inference process is not disturbed by incorrect metabolite annotations.

Minor Concern 2:
9. L683: Mass-to-charge ratio and some substance identification information of these metabolites and their classification contribution scores were used as input data. What is meant here by âsubstance identification informationâ?

Response:
For some of the metabolic signal peaks, we have accurately identified their molecular formulae and substance names by secondary mass spectrometry as âsubstance identification informationâ. Due to the limitation of existing databases, many unknown signals cannot be identified through secondary mass spectrometry. Therefore, PIUMet was also adopted to search for hidden metabolites and related proteins. The mass-to-charge ratio refers to the original m/z value of an unannotated metabolic signal peak above the threshold.
We have added the above information about âsubstance identification informationâ in the first paragraph of the âNetwork analysis and pathway enrichmentâ section of the Methods part.

Minor Concern 3:
10. Fig 4f: it is not clear which machine learning models are being compared to DeepMSProfiler, perhaps this could be clarified in the figure legend.

Response:
The main purpose of Figure 4f is to demonstrate the impact of unknown metabolic signals in the raw data on model performance, so we did not use any machine learning model for Figure 4f, but instead used the same architecture as DeepMSProfiler - Ensemble Densenet121. We have improved the legend of Figure 4f to make it clearer. Specifically, we eliminated metabolic signals that were not reported in the original data based on the m/z of known biomarkers. We retained the metabolic signals with publication counts greater than 1, greater than 3, and greater than 8 for modelling development. All ablated data is analysed using the same architecture as the original unprocessed data in our DeepMSProfiler. The vertical axis shows the accuracy of models built on the datasets of different publication counts and our DeepMSProfiler.

Minor Concern 4:
11. Further details are required in the methods about which machine learning library was used to implement the conventional models (SciKitLearn or other?) for purposes of reproducibility.

Response:
Thanks for this suggestion. XGBoost was implemented by the XGBClassifier function in the xgboost library. Other machine learning methods were implemented using the SciKitLearn library. SVM was adopted using the svm function, and the kernel of SVM is 'linear'. RF was implemented through the RandomForestClassifier function. Adaboost was adopted through the AdaBoostClassifier function. DNN was implemented using the MLPClassifier function.
We have added the above details in the âMachine learning models for comparisonâ section in the Methods part.

Remarks on code availability:
See comments to authors

Response:
Thanks very much for your efforts in investigating our code and data. More details have been added in the documentation (e.g., README.md) in the GitHub and the Code Ocean repositories to facilitate the users to run the codes and reproduce our research results.
Additionally, for the reviewers to access the full data in Code Ocean before the article publication, please use the following URL and account information.
URL: https://codeocean.com/capsule/2328223/tree
Username: yd849@bath.ac.uk
Password: code4paper
REVIEWER #2
Remarks to the Author:
I co-reviewed this manuscript with one of the reviewers who provided the listed reports. This is part of the Nature Communications initiative to facilitate training in peer review and to provide appropriate recognition for Early Career Researchers who co-review manuscripts.

Response:
Thanks very much for your efforts and valuable comments.
REVIEWER #3
Remarks to the Author:
The paper presents an explainable end-to-end deep learning method called DeepMSProfiler for the direct analysis of mass spectrometry data to reveal disease-specific metabolic profiles, which enables the analysis of raw metabolic signals with high accuracy and reliability. The method was applied to differentiate the metabolomic profiles of lung adenocarcinoma, benign lung nodules, and healthy individuals using serum samples. The results demonstrate that DeepMSProfiler successfully differentiated the different groups and detected early-stage lung adenocarcinoma with high accuracy. The method also overcame inter-hospital variability and effects of unknown metabolite signals. Furthermore, DeepMSProfiler provided interpretability and enabled the direct access to disease-related metabolite-protein networks. The method was also applied to lipid metabolomic data to unveil correlations of important metabolites and proteins.

Response:
Thanks very much for your valuable comments and positive feedback. We responded to your concerns point-by-point as underneath.

Comment 1:
1. Given that the Transformer model excels at processing sequential information, the reviewer suggest that the authors incorporate experiments utilizing Transformer architecture. This would provide a more comprehensive evaluation of the proposed method.

Response:
We believe that Transformer architecture might have a good performance on processing sequence information and language data, but there is still a big difference between metabolomics raw data and language data. First of all, the metabolic signal peak in the raw data is a 3D peak shape, including three coordinate axes: RT, m/z and intensity. It is not the data with only one coordinate axis. Secondly, the boundaries between different metabolic signal peaks are not clear, rather than isolated signal peaks that can be processed as discrete word embeddings like language data.
In the future, we will try to combine the features of Transformer, change the whole model design and build a novel model to try more interesting downstream tasks. However, this is not the main objective of our current study. The main task of this paper is to propose an end-to-end deep leaning architecture to process metabolomics raw data, and to verify its advantages compared to traditional approaches from multiple perspectives. Therefore, we believe that the architecture based on convolutional neural networks is consistent with the characteristics of the metabolomics data and the purpose of this experiment. As expected, the results demonstrate a high performance in disease classification using metabolomics raw data.
Comment 2:
2. In Figure 3(d), it seems that in the initial four layers, DeepMSProfiler cannot effectively differentiate data across various batches, but in the last layer, it successfully eliminates batch effects. Could the authors elucidate the reason behind this observation? Supporting results would also be needed.

Response:
We have expanded the layer visualization to demonstrate the progression of the automatic removal of batch effect in Figure 3d and Supplementary Figure 7. DenseNet121 is a deep neural network with 431 layers, of which 120 are convolutional layers (Supplementary Table 15). The fourth and fifth layers shown in Figure 3d refer to the output of the fourth dense connected module and the output of the fifth dense connected module. There are 112 layers between them. The modified Figure 3d and the new Supplementary Figure 7 illustrate the intermediate change process from the output of the fourth closely connected module to the output of the fifth closely connected module.
Actually, Figure 3e already illustrates that the closer to the output layer, the less relevant the data is to batch labels and the more relevant to class labels. This also explains how the batch effect removal is achieved via progress through hidden layers (Figure 3d).
The âInsensitivity to batch effectsâ section in the Results part and the legend of Figure 3 have been updated to mention the above information. And more details about the progression visualization can be found in Supplementary Figure 7.

Comment 3:
3. More recent advances (View, 2023, 4, 20220038; Exploration 2022, 2, 20210222) should be included and discussed to highlight this work.

Response:
Thanks for indicating these relevant references. We have cited them and discussed slightly in the Introduction part. Please see the Introduction part and the Reference 4 and 7.

Comment 4:
4. DeepMSProfiler employs an ensemble of five distinct models. It is recommended that the authors conduct further experiments to explore how each model contributes to the overall results, thereby providing deeper insights into the ensemble's effectiveness.

Response:
When using the ensemble strategy, we did not set different weights for each sub-model, so each sub-model is equally involved in the final prediction. In this revision, we have added the confusion matrices of prediction performance for each

sub-model (Supplementary Figure 4) to show their contributions to the overall results. We also evaluated the effect of the number of sub-models on improving performance and consequently chose to use 18 sub-models for DeepMSProfiler (Supplementary Fig. 2).
The above information has been added to the Results part. Please see the sections of âThe overview of the ensemble end-to-end deep-learning modelâ and âModel performance metricsâ in the Results part.

Remarks on code availability:
More details should be provided.

Response:
Thanks very much for your efforts in investigating our code and data. More details have been added in the documentation (e.g., README.md) of the GitHub and the Code Ocean repositories to facilitate the users to run the code and reproduce our reaserch results.
Additionally, for the reviewers to access the full data in Code Ocean before the article publication, please use the following URL and account information.
URL: https://codeocean.com/capsule/2328223/tree
Username: yd849@bath.ac.uk
Password: code4paper

Version 1:

Reviewer comments:

Reviewer #1

(Remarks to the Author)
We thank the authors for the revised manuscript and for the responses to our comments. The authors have made an effort to address all our points and revise the manuscript accordingly, however, there remain a few points of concern before we can recommend this for publication. We use the same numbering as the original review.

1. Other datasets and covariates
a. The authors have added a new colorectal cancer dataset and applied DeepMSProfiler to demonstrate its generalisability. Despite the small sample size of the colon cancer cohort (n=236), the model achieves 97.9% accuracy (with only 8 healthy samples in the independent validation set). To increase reader confidence, it would be helpful if the authors could provide metrics of uncertainty for this performance estimate (e.g. bootstrapping or averaging across multiple train-test splits). The authors should clarify whether they obtained this accuracy using a single discovery and validation dataset, or across different random splits; the latter would have higher confidence. In addition, the authors should clarify the bootstrapping procedure used in the methods (L701): ‘Confidence intervals were estimated using 1000 bootstrap iterations’ – was the model re-trained with different samples each time? How many?
b. In terms of the CCLE/pan cancer dataset, we thank the authors for adding further information to the methods. Perhaps surprisingly, the performance of DeepMSProfiler applied to this dataset is not stated anywhere in the manuscript. For completeness, the authors should also state the accuracy or other relevant performance metrics for this dataset.
c. We thank the authors for highlighting the three different correlation coefficients used. Rank based methods such as Spearman’s Rho and Kendal’s Tau can only detect monotonic relationships but there are other methods, such as mutual information, that could be used to detect non-monotonic relationships between variables.
2. We thank the authors for clarifying how the RISE heatmaps and feature importances were produced. Although the resolution is higher than other methods, the authors may want to acknowledge the limitation that the resolution may still be too low to distinguish all the individual peaks contributing to the classification. While we would expect that peaks with similar physicochemical properties appear close together, they may not be related to the outcome in the same way (e.g. in a close peak pair, one may be important for the prediction and the other not). This means that interpretation of the DeepMSProfiler model cannot be as rich as a more conventional model based on feature detection.
3. Metabolite idenfication and feature contribution to PIUMet
a. The authors mentioned they validated 11 metabolites identified by PIUMet using authentic chemical standards. For completeness, could the authors report how many/which metabolites identified by PIUMet they were not able to validate? Furthermore, in supplementary table 16, could authors please clarify what is meant by ‘score’, ‘pathway robustness_score’, and ‘confidence score’ – are these outputs of PIUMet?
b. We thank the authors for clarifying the use of a 0.7 feature contribution threshold. We suggest the authors also clarify whether the initial model accuracy reported in L194-196 was achieved without using a feature contribution threshold (or a 0.0 mask threshold). There is presumably a trade-off between model interpretability and performance, as the authors mentioned – this may be of interest to readers in the discussion.
4. We thank the authors for adding further information/visualisation about batch effect removal. We appreciate it is difficult to specify exactly what is being removed from the data in order to correct for the batch effects, and the visualisations help illustrate this. With regard to the model architecture, Fig 1c. illustrates the convolutional layers within the feature extractor. It would be helpful for readers to understand which layers in Fig 3d/e correspond to which layers in Fig 1c. Can this be stated

explicitly? Could the authors also clarify whether the blocks in the Convs are sequential or parallel?
5. In terms of the ensemble model the authors state ‘The architecture of each sub-model is the same, but some hyperparameters are different. Two different learning rates of 1e-3 and 1e-4 were used.’ Is the learning rate the only hyperparameter that was different among the sub-models?
6. We thank the authors for the further explanation on PIUMet. The authors remark: ‘The edges were weighted by the confidence level of the correlation reported in these databases.’ – what does this correlation signify? Does it mean metabolites and proteins are involved in the same reaction?
7. If the intention is for others in the field to run the DeepMSProfiler tool, it is key that the software can be readily installed and applied to new datasets. Perhaps the authors could state within the manuscript what level of compute time and resources are required to fit and train a model. The documentation on the README.md describes the running of the mainRun.py script. However, further detail is required for a new user to understand the input parameters (e.g. what file format should the input data be in? how should the disease type labels be formatted? What architectures are possible? Where can you specify the mask threshold for feature contributions?). On the Baidu repository, the input data are numpy data files, whereas in the paper it suggests they should be .mzML files. Additionally, creating a Python package distribution available via PyPI or Anaconda may render installation on a diverse range of platforms more straightforward.

Minor comment L372-274: what is meant by ‘queue’ here? Did you mean ‘set’? We suggest checking to ensure terminology is consistent throughout the manuscript.

Reviewer #2

(Remarks to the Author)
I co-reviewed this manuscript with one of the reviewers who provided the listed reports. This is part of the Nature Communications initiative to facilitate training in peer review and to provide appropriate recognition for Early Career Researchers who co-review manuscripts.

Reviewer #3

(Remarks to the Author)
The paper can be accepted.

Author Rebuttal letter:

Dear Reviewers,

Thanks so much for your valuable comments on our revised manuscript and helpful suggestions for further revision. We have now revised our manuscript according to your feedback. Please find the updated manuscript files in this new submission.

A point-by-point response to your comments is provided below, with your specific comments in black and our response in blue colour. Please also note that the changes made in the revised manuscript are marked in red.

Your comments and suggestion have enabled us to further improve our manuscript. We appreciate your time and efforts, and hope that we have satisfactorily addressed all issues and that our second revised manuscript meets the publication standards.

Best wishes,

Weizhong Li, Ph.D.
REVIEWERS' COMMENTS AND AUTHORSâ RESPONSES

Reviewer #1

Remarks to the Author:
We thank the authors for the revised manuscript and for the responses to our comments. The authors have made an effort to address all our points and revise the manuscript accordingly, however, there remain a few points of concern before we can recommend this for publication. We use the same numbering as the original review.
Response:
We really appreciated your valuable feedback and helpful suggestions. Please see our specific

responses below.

Comment 1:
1. Other datasets and covariates
a. The authors have added a new colorectal cancer dataset and applied DeepMSProfiler to demonstrate its generalisability. Despite the small sample size of the colon cancer cohort (n=236), the model achieves 97.9% accuracy (with only 8 healthy samples in the independent validation set). To increase reader confidence, it would be helpful if the authors could provide metrics of uncertainty for this performance estimate (e.g. bootstrapping or averaging across multiple train-test splits). The authors should clarify whether they obtained this accuracy using a single discovery and validation dataset, or across different random splits; the latter would have higher confidence. In addition, the authors should clarify the bootstrapping procedure used in the methods (L701): âConfidence intervals were estimated using 1000 bootstrap iterationsâ â was the model re-trained with different samples each time? How many?
Response:
The colorectal cancer data was randomly divided into a discovery data set and an independent testing dataset, and the discovery dataset was further randomly divided into the training dataset and the validation dataset with multiple times. This process was done to ensure that our model was not trained on the independent testing dataset. In the independent testing for the colon cancer dataset, our model has achieved an accuracy of 97.9% (95% CI, 97.7%-98.1%), a precision of 98.7% (95% CI, 98.6%-98.8%), a recall of 93.4% (95% CI, 92.9%-94.1%), and an F1 of 95.8% (95% CI, 95.4%-96.2%). We have added the above information about the data divisions and the performance metrics in the independent testing dataset for colorectal cancer in the " Application of model in colon cancer" section of the Results part (Lines 386-392).
When building models for the lung adenocarcinoma dataset, we investigated the potential impact of different data splits on performance. Our experiments showed that, within the constraints of balancing clinical factors, 10 different random split methods could all achieve an accuracy of over 90%. Due to time costs, we chose bootstrapping instead of the average results from different splits. Regarding the bootstrapping procedure used in the Methods part, our model was estimated by an ensemble strategy combining 20 models trained on the discovery dataset. It was not retrained using other samples. In this revision, we classified this in the âPerformance metricsâ section of the Methods part (Lines 730-732).

b. In terms of the CCLE/pan cancer dataset, we thank the authors for adding further information to the methods. Perhaps surprisingly, the performance of DeepMSProfiler applied to this dataset is not stated anywhere in the manuscript. For completeness, the authors should also state the accuracy or other relevant performance metrics for this dataset.
Response:
This model, trained on the CCLE dataset, is designed to distinguish between 23 different types of cancer. Due to the limited number of samples for most cancer types, particularly for biliary tract, pleura, prostate, and salivary gland cancers, each with less than 10 samples, we did not create a separate independent testing dataset for the performance validation. 20 sub-models have been trained, and in each sub-model training, 80% of all samples were randomly allocated for training to ensure that every sample could contribute to the training process, especially for cancer types with very few samples. The final ensemble model used for explainable analysis achieved 99.3% accuracy, 97.2% sensitivity and 100% specificity.
We have now provided the above information in the âDiscovery of metabolic-protein networks in pan-cancerâ section under Methods (Lines 404-411).

c. We thank the authors for highlighting the three different correlation coefficients used. Rank based methods such as Spearmanâs Rho and Kendalâs Tau can only detect monotonic relationships but there are other methods, such as mutual information, that could be used to detect non-monotonic relationships between variables.
Response:
In the Supplementary Table 2 of this revision, we have added additional information on the calculation results of mutual information. It is important to note that while mutual information can measure the similarity of clustering results, it does not provide a clear p value to determine the significance of the correlation, unlike other statistical methods.

Comment 2:
2. We thank the authors for clarifying how the RISE heatmaps and feature importances were produced. Although the resolution is higher than other methods, the authors may want to acknowledge the limitation that the resolution may still be too low to distinguish all the individual peaks contributing to the classification. While we would expect that peaks with similar physicochemical properties appear close together, they may not be related to the outcome in the same

way (e.g. in a close peak pair, one may be important for the prediction and the other not). This means that interpretation of the DeepMSProfiler model cannot be as rich as a more conventional model based on feature detection.
Response:
We are very grateful for your thought-provoking question about how to differentiate the importance of two closely related metabolites with similar physicochemical properties in prediction. In fact, the principle of explainability method was to mask the raw data and test how this affected the results. If two substances with similar properties can be distinguished by traditional methods, they can also be masked separately in the raw data to see their different classification contributions.
However, we do acknowledge the limitation that the resolution may still be relatively low in distinguishing all the individual peaks contributing to the classification. In this revision, we described this limitation in the Result part (Line 360-361) and the Discussion part (Lines 483-485).

Comment 3:
3. Metabolite idenfication and feature contribution to PIUMet
a. The authors mentioned they validated 11 metabolites identified by PIUMet using authentic chemical standards. For completeness, could the authors report how many/which metabolites identified by PIUMet they were not able to validate? Furthermore, in supplementary table 16, could authors please clarify what is meant by âscoreâ, âpathway robustness_scoreâ, and âconfidence scoreâ â are these outputs of PIUMet?
Response:
As the goal for using PIUMet in this study is to explore the biological explainability of the features we extracted, we did not intend to validate the metabolites detected by PIUMet. Therefore, we only selected those with available authentic standards in our laboratory to justify the presence of these metabolites in the lung cancer serum samples. Indeed, these metabolites were identified in the lung cancer serum as described in our previous study (Reference 40 in main text). This explanation has also been added into the âNetwork analysis and pathway enrichmentâ section of the Result part (Lines 361-366).
Please also note that, according to the journal formatting guidance, the previous Supplementary Table 16 has been renamed to Supplementary Table 5. In the current Supplementary Table 5, the âscoreâ values were derived from the results of PIUMet, while the âpathwayâ, ârobustness_scoreâ, and âconfidence_scoreâ were obtained from the outputs of KEGG pathway enrichment analysis using MetaboAnalyst. This explanation has been added into the note under the Supplementary Table 5.

b. We thank the authors for clarifying the use of a 0.7 feature contribution threshold. We suggest the authors also clarify whether the initial model accuracy reported in L194-196 was achieved without using a feature contribution threshold (or a 0.0 mask threshold). There is presumably a trade-off between model interpretability and performance, as the authors mentioned â this may be of interest to readers in the discussion.
Response:
The model accuracy reported in Lines 194-196 (in the previous version) was achieved without the use of any specific feature contribution threshold. Because explainability analysis is an independent module that occurs after the model training and does not require any changes to the model. It does not affect the modelâs performance on the independent testing dataset. So, there is also no trade-off between model interpretability and performance.
In the Results part (Lines 348-351), we have discussed the balance between the threshold and the impact of masked data on performance, but this was just to help us pick a suitable threshold.

Comment 4:
4. We thank the authors for adding further information/visualisation about batch effect removal. We appreciate it is difficult to specify exactly what is being removed from the data in order to correct for the batch effects, and the visualisations help illustrate this. With regard to the model architecture, Fig 1c. illustrates the convolutional layers within the feature extractor. It would be helpful for readers to understand which layers in Fig 3d/e correspond to which layers in Fig 1c. Can this be stated explicitly? Could the authors also clarify whether the blocks in the Convs are sequential or parallel?
Response:
Similar to Fig 1c, Conv1 to Conv5 in Fig3 d & e represent the outputs of the first to the fifth pooling layers in the feature extraction module. This information has now been mentioned in the figure legend for Fig 3. We selected DenseNet121 as our final backbone, which means that the blocks in the Conv layer are not arranged in a sequential or parallel manner. Instead, they are densely connected, and the specific connection relationships between them was reported in Supplementary Data 1.

Comment 5:
5. In terms of the ensemble model the authors state âThe architecture of each sub-model is the same, but some hyperparameters are different. Two different learning rates of 1e-3 and 1e-4 were used.â Is

the learning rate the only hyperparameter that was different among the sub-models?
Response:
In the previous version of the revised manuscript, we have described how to obtain the hyperparameters. In this revision, we have described all the hyperparameters in the "Ensemble Strategy" section of the Methods part. Please see Lines 702-705 in the manuscript main text.

Comment 6:
6. We thank the authors for the further explanation on PIUMet. The authors remark: âThe edges were weighted by the confidence level of the correlation reported in these databases.â â what does this correlation signify? Does it mean metabolites and proteins are involved in the same reaction?
Response:
PIUMet integrated iRefIndex (v13), HMDB (v3) and Recon 2 databases to obtain the relationship between m/z, metabolites and proteins, and generated an overall metabolite-protein network. The correlation refers to the connection between m/z, metabolites, and proteins. iRefIndex provides details about the physical interactions between proteins, which are detected through different experimental methods. The protein-metabolite relationships in Recon2 are based on their involvement in the same reaction. HMDB includes proteins that are involved in metabolic pathways or enzymatic reactions, as well as metabolites that play a role in protein pathways, based on known biochemical and molecular biological data. We have now added the above explanation in the âNetwork analysis and pathway enrichmentâ section under Methods (Lines 817-822).

Comment 7:
7. If the intention is for others in the field to run the DeepMSProfiler tool, it is key that the software can be readily installed and applied to new datasets. Perhaps the authors could state within the manuscript what level of compute time and resources are required to fit and train a model. The documentation on the README.md describes the running of the mainRun.py script. However, further detail is required for a new user to understand the input parameters (e.g. what file format should the input data be in? how should the disease type labels be formatted? What architectures are possible? Where can you specify the mask threshold for feature contributions?). On the Baidu repository, the input data are numpy data files, whereas in the paper it suggests they should be .mzML files. Additionally, creating a Python package distribution available via PyPI or Anaconda may render installation on a diverse range of platforms more straightforward.
Response:
To help users to try the software, we have now added details regarding computation time and resources in the "Ensemble Strategy" section under Methods (Lines 704-705). The batch size was set to 8 and the training was run for 200 epochs. A model typically took about 2 hours to train on a GP100GL (16GB) GPU server.
According to your suggestion, we have also added more details in the README.md file to better describe each input parameter. In addition, we now provide a mzml2itmx function in our GitHub code for converting .mzml data to .npy format files, which take up less storage space. The details of the data conversion procedure have been descried in the âData format conversionâ section of the Methods Part. We also packaged DeepMSProfiler and submitted it to the PyPI repository as suggested. The detailed usage instructions can be found in the README.md file on our GitHub repository.
Minor comment L372-274: what is meant by âqueueâ here? Did you mean âsetâ? We suggest checking to ensure terminology is consistent throughout the manuscript.
Response:
Yes, âqueueâ is âsetâ here. We have changed âqueueâ to âdatasetâ throughout the entire manuscript. We have also checked other terminologies and made sure their consistency.

Remarks on code availability:

See comments in our review above
Response:
Many thanks for the helpful comments on the code availability.

Reviewer #2

Remarks to the Author:
I co-reviewed this manuscript with one of the reviewers who provided the listed reports. This is part of the Nature Communications initiative to facilitate training in peer review and to provide appropriate recognition for Early Career Researchers who co-review manuscripts.

Response:
We greatly appreciated your time and efforts and the helpful suggestions.

Reviewer #3

Remarks to the Author:
The paper can be accepted.
Response:
We greatly appreciated your efforts and valuable comments.