Supporting Information

Deep Neural Networks for Classification of LC-MS Spectral Peaks

Edward D. Kantz[1,2], Saumya Tiwari[2], Jeramie D. Watrous[2], Susan Cheng[3], Mohit Jain[2]

[1]Department of Bioengineering, and [2]Departments of Medicine & Pharmacology, University of California San Diego, La Jolla, CA, 92093

[3]Smidt Heart Institute, Cedars-Sinai Medical Center, Los Angeles, CA 90048

Correspondence:

Mohit Jain: mjain@ucsd.edu

Supporting Information

Supporting Text

Supporting Tables

Supporting Figures

Supporting Text

LC-MS metabolomics.

Metabolomics analysis was performed using human plasma samples obtained from Maastricht University (N=78) (Cohort 1) and the Human Functional Genomics Project (N=526) (Cohort 2).[1,2] All study protocols were approved and human samples collected with full participant consent and under Institutional Review Board authorization. For all samples, blood was collected into EDTA-tubes and centrifuged at 1,000xg. Plasma was isolated, snap-frozen and stored at -80°C until analyses.

Metabolites were extracted from human plasma samples using organic solvent consisting of methanol:acetonitrile at a 1:1 ratio (v/v). A volume of 80 μL of cold extraction solvent was added to 20 μL of plasma. Samples were then vortexed for 15 minutes at 4°C followed centrifugation at 4200 rpm and 4°C for 10 minutes. Resulting supernatants were transferred to 96 well microtiter plates for analysis.

Chromatographic separation of compounds was achieved on a Thermo Vanquish UHPLC system using a Merck-SeQuant ZIC-pHILIC PEEK coated HPLC column (100 x 2.1 mm, 5μm particle size) fitted with a Phenomenex KrudKatcher ULTRA HPLC in-line filter (0.5 μm filter x 0.004 in ID) to protect the column from microparticulates. A gradient of mobile phases *A* (20 mM ammonium bicarbonate in water, titrated with ammonium hydroxide to a pH of 9.6) and *B* (acetonitrile) was used for chromatographic separation of metabolites. A constant flow rate of 400 μL min^{-1} and column temperature of 45°C was used. The mobile phase ratio was held at 10% mobile phase *A* from 0 to 0.25 min and linearly increased up to 45% at 4 min, and subsequently maintained for 2 min. Post-run the flow returned to 10% *A* and 90% *B* for 3.25 min to allow for column re-equilibration. During all analyses, the sample tray temperature was held at 4°C. Sample injection volume was 2 μL.

The LC system was coupled to a Thermo Q-Exactive Orbitrap mass spectrometer (Thermo Fisher Scientific, Waltham, MA) equipped with a heated electrospray ionization source (HESI). The following HESI settings were used: sheath gas flow rate 40 arbitrary units, auxiliary gas flow rate 20 arbitrary units, sweep gas flow rate 2 arbitrary units, spray voltage 3.5 kV, capillary temperature 275°C, S-lens RF level 45, auxiliary gas heater temperature 350°C. A scan range of 65 to 975 m/z was used with a resolution of 35,000. All profile data was collected in negative ionization mode.

Data Extraction

Data acquisition for small molecule quantification was performed with the Xcalibur software. Raw MS/MS data were converted to mzXML files using MSConvert.[3] All mzXML files were loaded into MZmine 2.[4] MZmine 2 modules and settings used to preprocess the data, generate chromatograms, detect peaks, and align peaks can be found in **Table S1** and **Table S2**.

Data Preparation

There were 2770 peak groups generated from 78 mzXML files and processed using the MZmine 2 workflow and settings listed in **Table S1** for the typical (more restrictive) settings and **Table S2** for the less restrictive settings. These peak groups were manually reviewed by a single expert reviewer and labeled as either good or bad (see **Figure S1** for examples of good and bad peaks). Groups where all good quality peak shapes occurred near the noise intensity

for that chromatogram and peak groups that were near the decision threshold were excluded from the training data.

The training data was partitioned using a training:validation:test ratio of 47%:27%:26%, such that 1304 of the peak groups were used to train the model, 740 were used for validation and 726 were used for testing (**Table S3**). The training partition was balanced, including an equal number of examples in the good (true positive) and bad (false positive) classes.

Models were also tested on 3000 peak groups selected from Cohort 2 (**Table S3**).

Image based deep neural network

The deep neural network model was trained using the dataset discussed in the previous section. Full peak shapes used for training the deep learning model were extracted from the LC-MS data (mzXML format) with windows specifying upper and lower limits in m/z and retention time. Since the peak feature is judged in the context of its surrounding chromatographic landscape, we extended the observable retention time of each extracted ion chromatogram to capture the neighborhood information for each feature.

The raw peak signals were discretized in the retention time dimension by equal width binning with a bin size of 0.01 minutes. This resulted in a total of 602 bins in the retention time dimension. The raw peak signals were converted to an image format where the Y dimension was comprised of samples while the X dimension was comprised of signal intensities in retention time bins as illustrated in **Figure 2A–C** in the main text. The image size chosen for this model was 64 sample bins x 64 retention time bins, because the majority of peaks spanned a width of less than 64 retention time bins. In each image, the sample rows 1 to 63 were filled by the top 63 samples with highest peak intensities and the last row was filled by a binary vector indicating the retention time range of the peak under consideration, labeled as “Window” in **Figure 2A-C** in the main text. Furthermore, each feature intensity is scaled so that the maximum intensity in each image is one (**Figure 2A**).

The deep neural network model was trained using Keras libraries[5] in Python.[6] The model architecture was as follows:
Layer 1: 2D convolution with 32 Filters, 3x3 Kernel Size, ReLU activation
Layer 2: 2D MaxPooling with Pool Size 2x2, Stride 2x2
Layer 3: Dropout 0.2
Layer 4: 2D convolution with 16 Filters, 3x3 Kernel Size, ReLU activation
Layer 5: 2D MaxPooling with Pool Size 2x2, Stride 2x2
Layer 6: Dropout 0.2
Layer 7: Fully Connected Layer with 64 neurons, ReLU activation
Layer 8: Dropout 0.5
Layer 9: Fully Connected Layer with 2 neurons, Softmax activation

The model was applied at a threshold determined from the receiver-operator characteristic (ROC) curve of the calibration set using the “closest.topleft” method from the R Package “pROC”.[7] The same method was used to determine the threshold for the multiple logistic regression and Random Forest models.

Review of Neural Network Performance

In order to evaluate the size of the training data set and its impact on model performance, the model was trained iteratively using progressively larger subsets of the full training data set. The

performance of the model was evaluated at each iteration using the metric of mean squared error (**Figure S5**)

While we saw a very significant improvement in the data quality after applying the neural network model (retention of most true positive peak features and great reduction in false positive peak features), a fraction of peak features selected were still false hits. We investigated why this was the case by manually reviewing the peaks that the machine learning algorithm picked inaccurately. A common major source of errors in machine learning classification is the difference between data used for training and independent testing. Therefore, it is imperative to understand the data extraction procedure and recognize the differences in the data owing to the set of experiments that we performed here.

In a typical MZmine 2 workflow, peaks are selected by an algorithm using a set of user-provided settings. MZmine 2 does not explicitly generate peak feature windows. In order to extract a feature from raw data, in-house written scripts were used to determine windows around the peaks grouped by MZmine 2's join aligner module. Using these windows, data is then extracted from mzXML files. Either of the feature identification and window setting steps can sometimes erroneously result in two peaks being merged in one window, one peak being split into two windows and peaks being duplicated in the data. The prevalence of each type of windowing error is dependent on the peak selection settings used.

Since the initial training data was developed using less restrictive settings in MZmine 2 (**Table S2**), we anticipated some difference in performance when the model was applied to data exported with restrictive settings (**Table S1**). We were able to categorize these errors in five broad categories, and representative examples from each type are shown in **Figure S3**.

One surprising finding was that the model gave low scores to shoulder peaks in the restricted settings cohort (**Figure S3A**). Here shoulder peaks refer to smaller peaks occurring at the same retention time and m/z window as a larger peak, such that the peak shapes are not fully resolved to baseline. This error seems to come from the low abundance of shoulder peaks in our training dataset, where shoulders were often put in the same window as the higher intensity neighboring peak. Thus, the model did not have a prior basis of judging windows with just the shoulder peaks. As such, we do not anticipate peak losses if the data is consistently prepared with the same settings as data used for model training.

In other cases, the background signal from some samples overwhelmed the peak signal (**Figure S3B**). These signals can occur when certain samples have a peak with the same m/z as the candidate peak that has a much higher intensity than the candidate peak. In these cases, the tail of the much larger peak in those files can be higher than the apex of the true peaks in other files. As illustrated by the listed scores, an acceptable peak overlaid by noise was rejected once and accepted the other time. Such peaks formed a negligible portion of total peaks in the dataset, and an expert may reject such peak features because the high intensity noise could create problems in subsequent statistical analyses. But the answer to the question of whether the window contains a peak would be yes, causing ambiguity in both annotation and classification.

In the third category (**Figure S3C**) were wide peaks that were not Gaussian and had low signal to noise ratio. Upon reexamining these cases, the reviewers agree that these cases are ambiguous and these peaks could have been accepted. The fourth category of misclassification were broad square peaks with low signal to noise ratio (**Figure S3D**). These peaks were called

as ambiguous on the side of being rejected upon repeat examination by the expert. These peaks were scored low but acceptable by the model. In manual curation, experts often determine if the peak apex is captured in the given window. It is possible that this leads to some ambiguous labeling among reviewers, resulting in low confidence of the neural network classifier.

The final category of misclassification was features that the neural network classifier got wrong, but were unequivocally rejected by the reviewers (**Figure S3E**). While we still do not understand the full reason behind why this was the case, it appears that the neural network classifier noted the slight bump in the noise as a possible peak.

Peak group parameter model

The following six peak shape attributes were exported from MZmine 2 for each peak group: peak duration, height, area, full-width half max (FWHM), tailing factor, and asymmetry factor. These attributes were used to define different peak group descriptive statistics (peak group parameters) intended to separate groups of true peaks from groups of noise or peaks of unacceptable quality. These peak group parameters were then used as variables in the predictive models. The peak group parameter definitions are provided in **Table S4**.

Two separate models were generated and used for prediction, a multiple logistic regression model and a Random Forest[8] model. The models were generated and implemented using scripts written in the R programming language.[9] All code used in this paper has been provided to the scientific community at https://github.com/JainLab.

The multiple logistic regression model variables are selected using a simple forward selection procedure. K-fold cross validation is used to determine which individual variable provides the best prediction. Additional variables are added one at a time using the same procedure, selecting the variable which improves performance the most as determined by k-fold cross validation of the training data. When the addition of new variables does not improve performance, variable selection is complete. The calibration data set is then used to prune variables from the model which do not improve prediction. The resulting multiple logistic regression model is parsimonious and unlikely to be an over-fit of the training data.

For the data sets and MZmine 2 settings used, the final model included 15 different peak group parameters. These peak group parameters are indicated with an asterisk in **Table S4**. It should be noted that the selection of variables included in final model is dependent on both the LC-MS dataset as well as the MZmine 2 settings used for peak detection and alignment.

The Random Forest algorithm as developed by Leo Breiman[8] performs classification through the construction of a multitude of decision trees trained on random subsets of the training data. The algorithm implements this method via the R package randomForest[10]. The calibration data set was used to optimize the node size parameter.

The basic workflow for both peak group parameter models is illustrated in **Figure S2.**

Both models are independently evaluated on the test subset of the input data and results are output in the form of various plots including receiver operator curves (ROC) (**Figure 3D** in the main text) as well as histograms (**Figure S4 A** and **B**) and density plots comparing the distributions of true and false positives .

When applying the models to an independent dataset, the script exports a histogram showing the distribution of predicted probability scores for the input peak group set to assist with

threshold selection. Tables are exported with peak groups above the user-set threshold in a format which can directly be used to label the peak groups in MZmine 2 using the “Custom database search” module.

References

(1) Reijnders, D.; Goossens, G. H.; Hermes, G. D. A.; Neis, E. P. J. G.; van der Beek, C. M.; Most, J.; Holst, J. J.; Lenaerts, K.; Kootte, R. S.; Nieuwdorp, M.; et al. Effects of Gut Microbiota Manipulation by Antibiotics on Host Metabolism in Obese Humans: A Randomized Double-Blind Placebo-Controlled Trial. *Cell Metab.* **2016**, *24* (1), 63–74. https://doi.org/10.1016/j.cmet.2016.06.016.

(2) Swertz, M. A.; Kumar, V.; Li, Y.; Toenhake-Dijkstra, H.; Joosten, I.; Netea-Maier, R. T.; Jaeger, M.; Pavelka, N.; van Herwaarden, A. E.; Diavatopoulos, D. A.; et al. Host and Environmental Factors Influencing Individual Human Cytokine Responses. *Cell* **2016**, *167* (4), 1111-1124.e13. https://doi.org/10.1016/j.cell.2016.10.018.

(3) Chambers, M. C.; Maclean, B.; Burke, R.; Amodei, D.; Ruderman, D. L.; Neumann, S.; Gatto, L.; Fischer, B.; Pratt, B.; Egertson, J.; et al. A Cross-Platform Toolkit for Mass Spectrometry and Proteomics. *Nat. Biotechnol.* **2012**, *30* (10), 918–920. https://doi.org/10.1038/nbt.2377.

(4) Pluskal, T.; Castillo, S.; Villar-Briones, A.; Orešič, M. MZmine 2: Modular Framework for Processing, Visualizing, and Analyzing Mass Spectrometry-Based Molecular Profile Data. *BMC Bioinformatics* **2010**, *11* (1), 395. https://doi.org/10.1186/1471-2105-11-395.

(5) Chollet, F. Keras. 2015.

(6) Python Software Foundation. Python Language Reference.

(7) Robin, X.; Turck, N.; Hainard, A.; Tiberti, N.; Lisacek, F.; Sanchez, J.-C.; Müller, M. PROC: An Open-Source Package for R and S+ to Analyze and Compare ROC Curves. *BMC Bioinformatics* **2011**, *12* (1), 77. https://doi.org/10.1186/1471-2105-12-77.

(8) Breiman, L. Random Forests. *Mach. Learn.* **2001**, No. 45, 5–32. https://doi.org/10.1023/A:1010933404324.

(9) R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing.

(10) Wiener, A. L. and M. Classification and Regression by RandomForest. *R News* **2002**, *2* (3), 18–22.

Supporting Tables

Table S1. MZmine 2 settings used for the typical (more restrictive) condition.

Parameters	**Values**
Mass detection	
Scans	MS level: 1
Mass detector	Centroid
Chromatogram Builder	
Scans	MS level: 1
Min time span (min)	0.04
Min height	3.00E+05
m/z tolerance	0.0 m/z or 15.0 ppm
Chromatogram deconvolution	
Algorithm	Local minimum search
Chromatographic threshold	0.00%
Search minimum in RT range (min)	0.03
Minimum relative height	1.00%
Minimum absolute height	4.00E+05
Min ratio of peak top/edge	1.5
Peak duration range (min)	0.040-4.00
Join aligner	
m/z tolerance	0.0 m/z or 15.0 ppm
Weight for m/z	1
Retention time tolerance	0.02 min
Weight for RT	1
Peak list rows filter	
Minimum peaks in a row	25
Keep or remove rows	Keep rows that match all criteria

Table S2. MZmine 2 settings used for the less restrictive condition.

Parameters	**Values**
Mass detection	
Scans	MS level: 1
Mass detector	Centroid
Chromatogram Builder	
Scans	MS level: 1
Min time span (min)	0.04
Min height	1.50E+05
m/z tolerance	0.0 m/z or 15.0 ppm
Chromatogram deconvolution	
Algorithm	Local minimum search
Chromatographic threshold	0.00%
Search minimum in RT range (min)	0.025
Minimum relative height	1.00%
Minimum absolute height	1.50E+05
Min ratio of peak top/edge	1.2
Peak duration range (min)	0.025-4.00
Join aligner	
m/z tolerance	0.0 m/z or 15.0 ppm
Weight for m/z	1
Retention time tolerance	0.02 min
Weight for RT	1
Peak list rows filter	
Minimum peaks in a row	3
Keep or remove rows	Keep rows that match all criteria

Table S3. Summary of training, calibration and test datasets

Cohort 1	Train	1304 peak groups
	Calibrate	740 peak groups
	Test	726 peak groups
Cohort 2	Test	3000 peak groups

Table S4a. Definitions of peak group parameters used in multiple logistic regression and Random Forest models (1 of 4). The 15 peak group parameters used in the final logistic regression model are indicated with an asterisk (*).

Variable Name	Formula
propDetected	$\frac{TotalFilesWithPeakDetected}{TotalFiles}$
logMaxHt	$log(max(peakHeight))$
logMedHt	$log(med(peakHeight))$
*logSdHeight	$log(sd(peakHeight))$
logMaxArea	$log(max(peakArea))$
logMedArea	$log(med(peakArea))$
*logSdArea	$log(sd(peakArea))$
*logMaxDur	$log(max(peakDuration))$
logMedDur	$log(med(peakDuration))$
logStDevDur	$log(sd(peakDuration))$
CvDur	$\frac{sd(peakDuration)}{mean(peakDuration)}$
custVarMeasDur	$\frac{sd(peakDuration)}{max(peakDuration)}$
logMaxFWHM	$log(max(peakFWHM))$
logMedFWHM	$log(med(peakFWHM))$
logMaxRelArea	$log\left(max\left(\frac{peakArea}{peakHeight}\right)\right)$
logMedRelArea	$log\left(med\left(\frac{peakArea}{peakHeight}\right)\right)$
logMinRelArea	$log\left(min\left(\frac{peakArea}{peakHeight}\right)\right)$

Table S4b. Definitions of peak group parameters used in multiple logistic regression and Random Forest models (2 of 4). The 15 peak group parameters used in the final logistic regression model are indicated with an asterisk (*).

Variable Name	Formula
logSdRelArea	$log\left(sd\left(\frac{peakArea}{peakHeight}\right)\right)$
*logCvRelArea	$log\left(\frac{sd\left(\frac{peakArea}{peakHeight}\right)}{mean\left(\frac{peakArea}{peakHeight}\right)}\right)$
custVarMeasRelArea	$\frac{sd\left(\frac{peakArea}{peakHeight}\right)}{max\left(\frac{peakArea}{peakHeight}\right)}$
*sdRatioAreaHt	$\frac{sd(Height)}{sd(Area)}$
logMaxRelFWHM	$log(max(\frac{peakFWHM}{peakDuration}))$
*medRelFWHM	$median(\frac{peakFWHM}{peakDuration})$
*logMedRelFWHM	$log(median(\frac{peakFWHM}{peakDuration}))$
*minRelFWHM	$log(min(\frac{peakFWHM}{peakDuration}))$
logSdRelFWHM	$log(sd(\frac{peakFWHM}{peakDuration}))$
logCvRelFWHM	$log(sd(\frac{peakFWHM}{peakDuration})/mean(\frac{peakFWHM}{peakDuration}))$
custVarMeasRelFWHM	$log(sd(\frac{peakFWHM}{peakDuration})/max(\frac{peakFWHM}{peakDuration}))$
logSdSpecWidth	$log(sd(\frac{peakFWHM}{peakArea}))$
logMaxTfact	$log(max(peakTailingFactor))$

Table S4c. Definitions of peak group parameters used in multiple logistic regression and Random Forest models (3 of 4). The 15 peak group parameters used in the final logistic regression model are indicated with an asterisk (*).

Variable Name	Formula
logMedTfact	$log(median(peakTailingFactor))$
logMinTfact	$log(minimum(peakTailingFactor))$
sdTfact	$sd(peakTailingFactor)$
logSdTfact	$log(sd(peakTailingFactor))$
CvTfact	$\frac{sd(peakTailingFactor)}{mean(peakTailingFactor)}$
logCvTfact	$log\left(\frac{sd(peakTailingFactor)}{mean(peakTailingFactor)}\right)$
custVarMeasTfact	$\frac{sd(peakTailingFactor)}{max(sd(peakTailingFactor))}$
logMaxAfact	$log(max(peakAsymmetryFactor))$
logMedAfact	$log(median(peakAsymmetryFactor))$
AsymLngFact	if peak Assymetry Factor < 1, then $AsymLngFact = \frac{-1}{peakAssymetryFactor}$ else $AsymLngFact = peakAssymetryFactor$
logMaxAbsAsymLngFact	$log\big(max(\lvert AsymLngFact\rvert)\big)$
*medLogAbsAsymLngFact	$log\big(median(\lvert AsymLngFact\rvert)\big)$
logMinAbsAsymLngFact	$log\big(minimum(\lvert AsymLngFact\rvert)\big)$
*sdAsymLngFact	$sd(AsymLngFact)$
logSdAsymLngFact	$log(sd(AsymLngFact)$
CvAsymLngFact	$\frac{sd(AsymLngFact)}{mean(AsymLngFact)}$ if peak Assymetry Factor < 1, then $AsymLngFact = \frac{-1}{peakAssymetryFactor}$ else $AsymLngFact = peakAssymetryFactor$

Table S4d. Definitions of peak group parameters used in multiple logistic regression and Random Forest models (4 of 4). The 15 peak group parameters used in the final logistic regression model are indicated with an asterisk (*).

Variable Name	Formula
logCvAsymLngFact	$log\left(\frac{sd(AsymLngFact)}{mean(AsymLngFact)}\right)$
custVarMeasAsymLngFact	$\frac{sd(AsymLngFact)}{max(AbsAsymLngFact)}$
logMaxT_Afact	$log\left(max\left(\frac{peakAssymetryFactor}{peakTailingFactor}\right)\right)$
logMedT_Afact	$log\left(median\left(\frac{peakAssymetryFactor}{peakTailingFactor}\right)\right)$
logMinT_Afact	$log\left(minimum\left(\frac{peakAssymetryFactor}{peakTailingFactor}\right)\right)$
sdT_Afact	$sd\left(\frac{peakAssymetryFactor}{peakTailingFactor}\right)$
logSdT_Afact	$log\left(sd\left(\frac{peakAssymetryFactor}{peakTailingFactor}\right)\right)$
CvT_Afact	$\frac{sd\left(\frac{peakAssymetryFactor}{peakTailingFactor}\right)}{mean\left(\frac{peakAssymetryFactor}{peakTailingFactor}\right)}$
*logCvT_Afact	$log(CvT_Afact)$
*custVarMeasT_Afact	$\frac{sd\left(\frac{peakAssymetryFactor}{peakTailingFactor}\right)}{max\left(\frac{peakAssymetryFactor}{peakTailingFactor}\right)}$
*logCombFact1	$log((minRelFWHM)(medTfact)(medAfact))$
*combVarMeas1	$\frac{custVarMeasRelArea}{custVarMeasDur}$
*combVarMeas2	$log\left(\frac{custVarMeasRelArea}{custVarMeasDur}\right)$

Supporting Figures

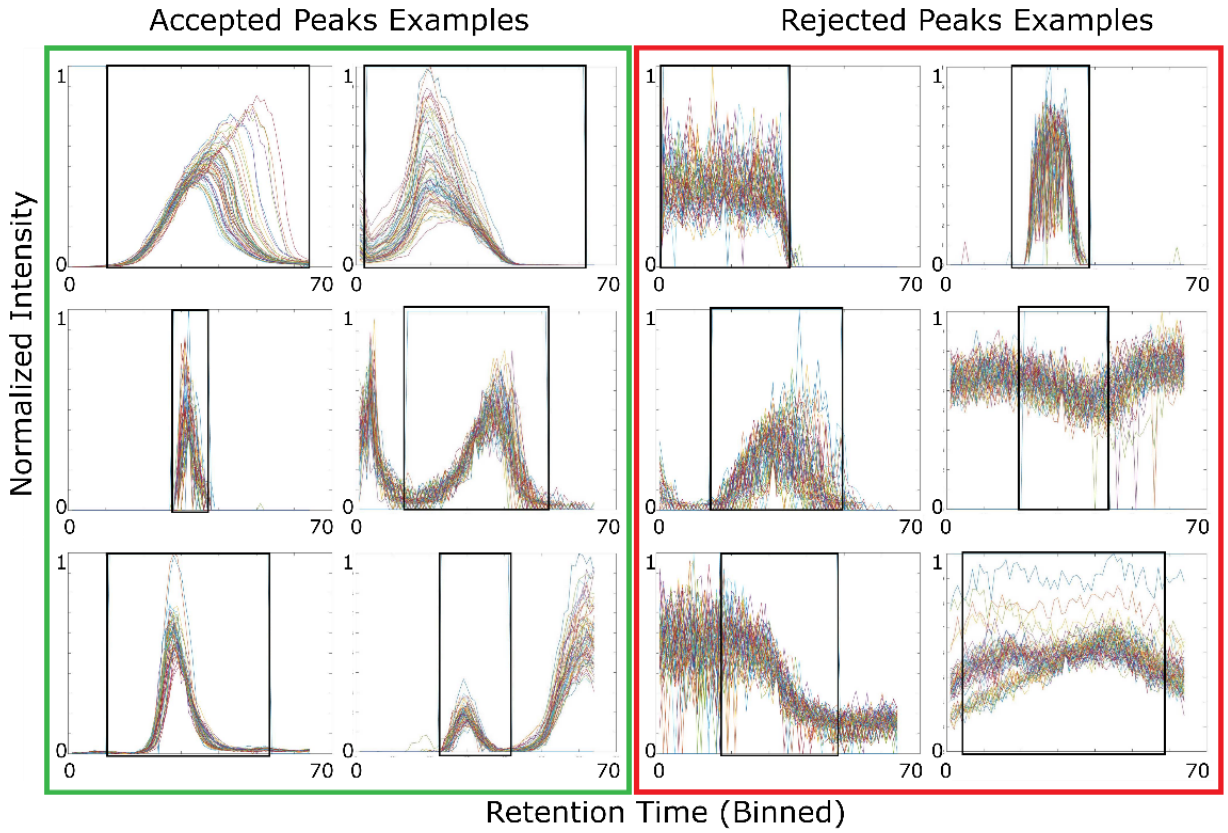

Figure S1 Peak shape examples that were accepted (green box on the left) and rejected (red box on the right) by the human reviewer.

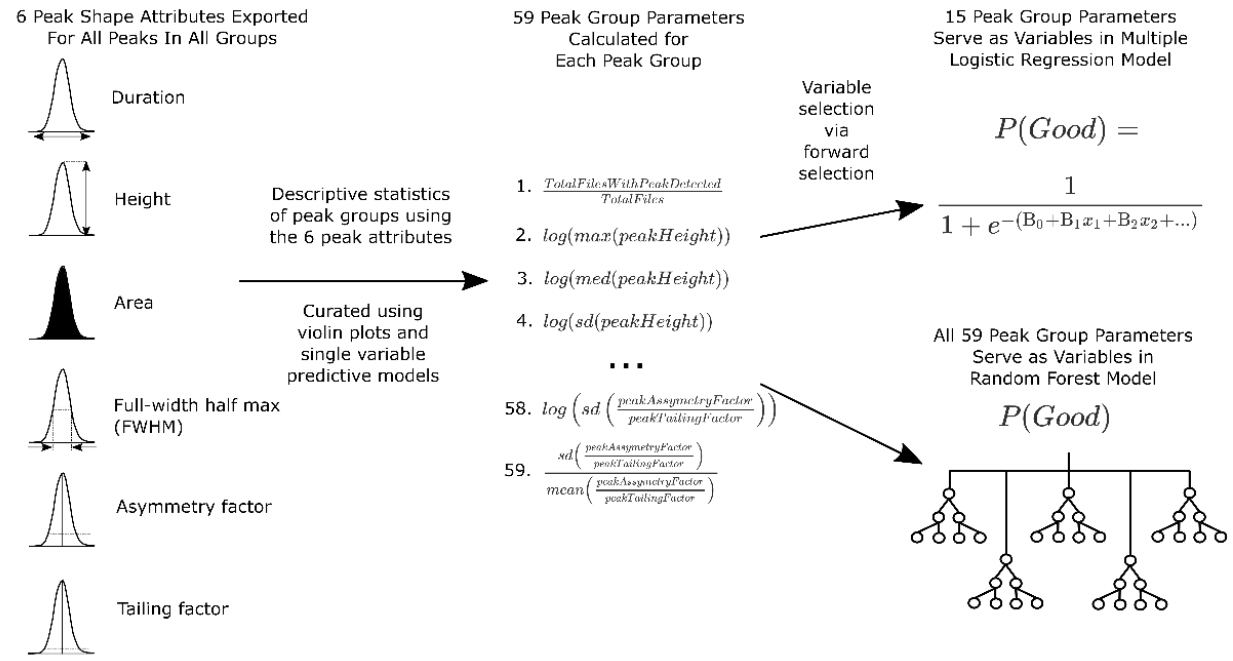

Figure S2. Peak group parameter workflow illustrating the progression from exporting peak shape attributes to generating multiple logistic regression and Random Forest models.

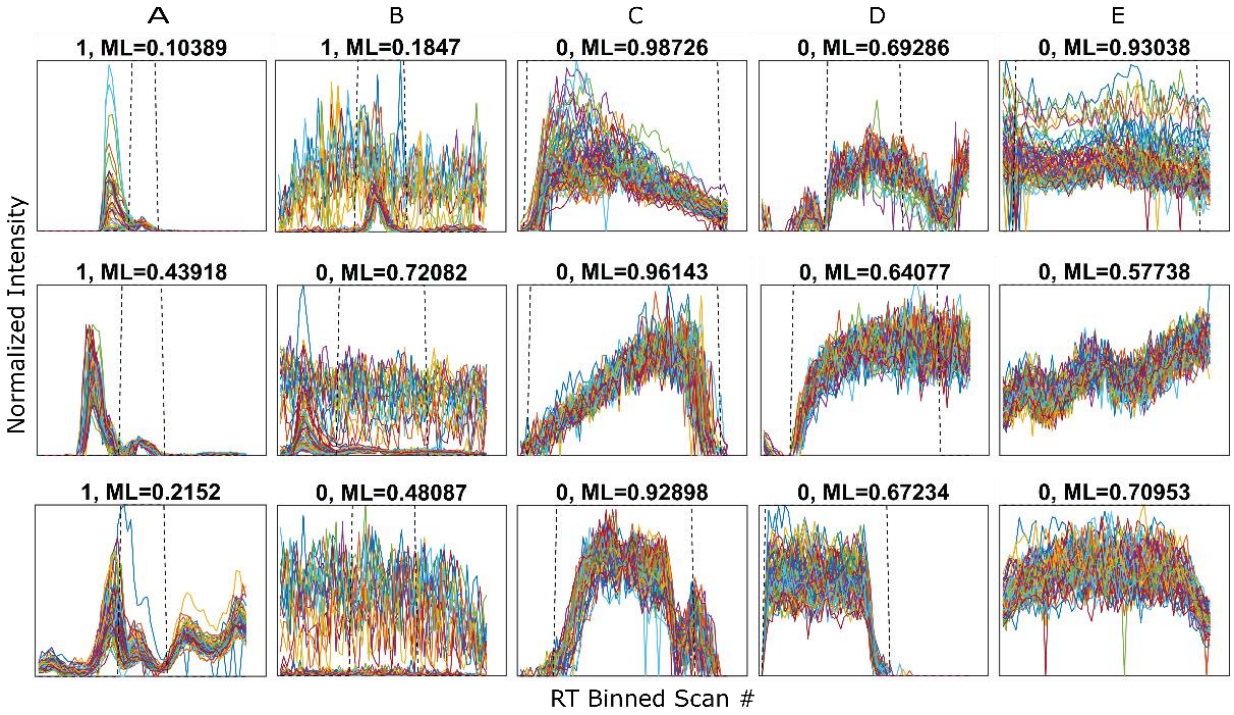

Figure S3 Peak shape examples and the corresponding model scores. (A) “Shoulder peaks” lower intensity peaks occurring on the tail of a higher intensity peak. (B) The background signal from certain samples has a higher intensity than the intensity of the peak occurring in the window. (C) Peaks with unusual (non-gaussian) shapes. (D) Broad peaks with low signal to noise ratios. (E) Incorrect classification by the model. High scoring signal that should have been rejected as a non-peak.

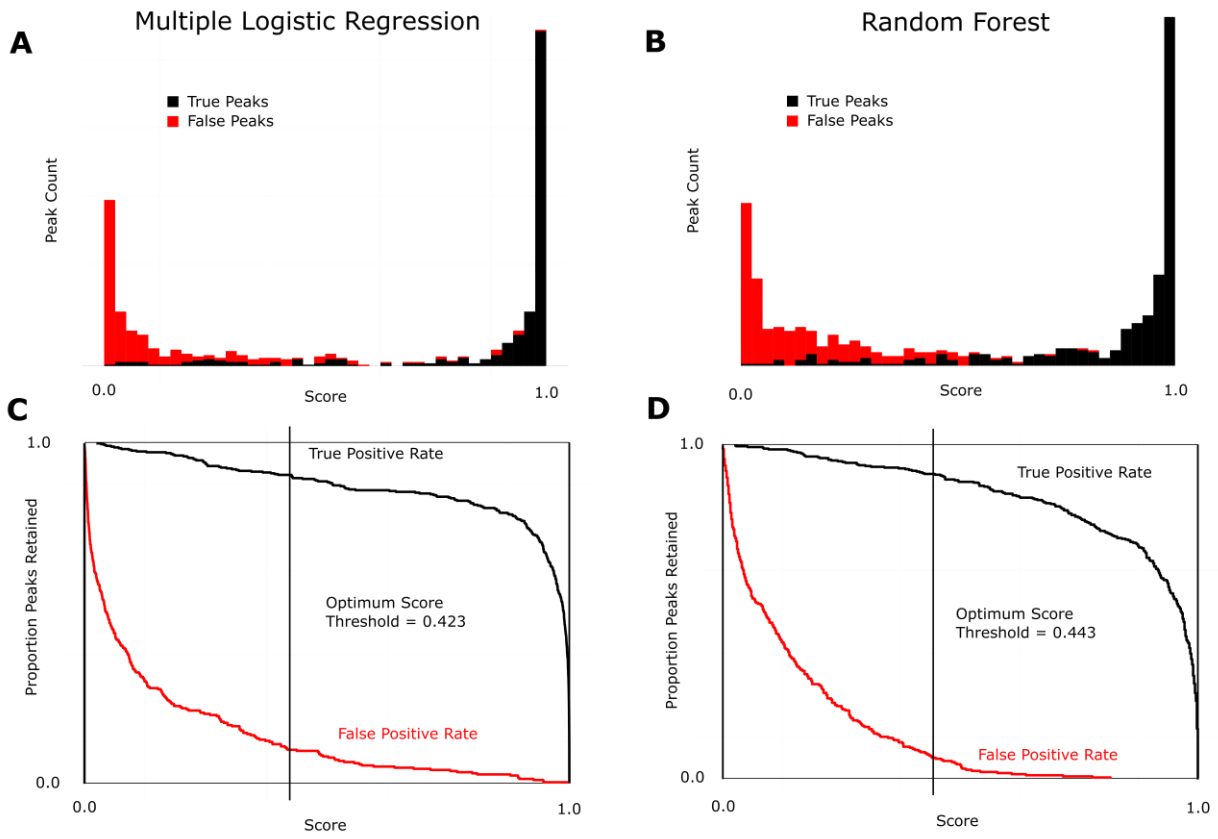

Figure S4. Performance of Multiple Logistic Regression and Random Forest models (A) A stacked-bar histogram of peak probability scores predicted by the multiple logistic regression model for ground truth false positive (red) and true positive (black) peak groups in the test set. (B) Histogram for the Random Forest Model (C) A plot of the proportion of true positive (black line) and false positive (red line) peaks retained at each score threshold from zero to one. Optimum score threshold was selected using the point on the corresponding ROC curve closest to the top left corner (sensitivity = 1, specificity = 1) (D) Corresponding plot for the Random Forest Model.

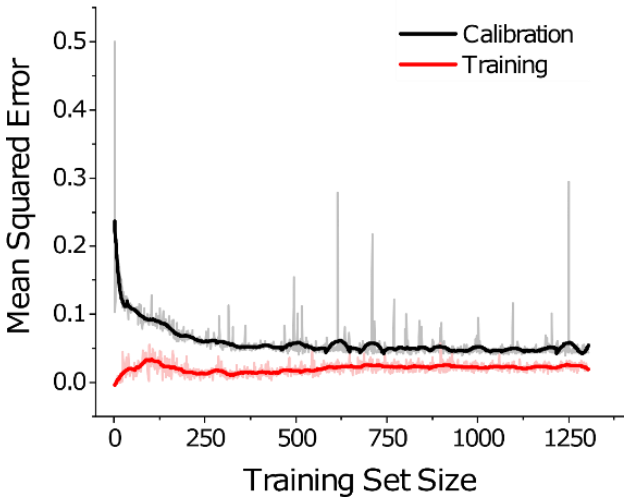

Figure S5 Training set size evaluation. Mean squared error of the model predictions for the training and calibration data sets vs. training set size.