**Avir.R User Instruction**

(Version 1.0, Jun 08, 2023)

Zixuan Zhang, Tao Huan\*

Department of Chemistry, Faculty of Science, University of British Columbia, Vancouver Campus, 2036 Main Mall, Vancouver, V6T 1Z1, BC, Canada

\* Author to whom correspondence should be addressed:

Dr. Tao Huan

Tel: (+1)-604-822-4891

E-mail: thuan@chem.ubc.ca

Website: <https://huan.chem.ubc.ca/>

• Avir.R user instruction is a support vector machine model trained in R to predict the quality of peak integration in metabolic features.

• Avir.R script is freely available for non-commercial use.

• The instructions are given below:

1) Download and install R studio following the instruction on the RStudio website

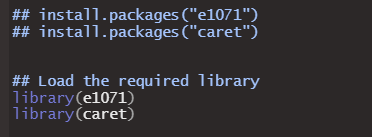
(<https://www.rstudio.com/>)

2) Download the Avir.R, training data, testing data and the demo file from (https://github.com/HuanLab/AVIR.R) then save it in a folder.

3) If the R package “e1071” and “caret” are not installed. Please run the following code:

install.packages("e1071")

install.packages("caret")



4) Within the same folder, prepare two input files

A sample metabolite-intensity table from peak height measurement (file 1)

A sample metabolite-intensity table from peak area measurement (file 2)

in the following formats:

Sample metabolite-intensity table containing all real samples (file 1 & 2, prepared

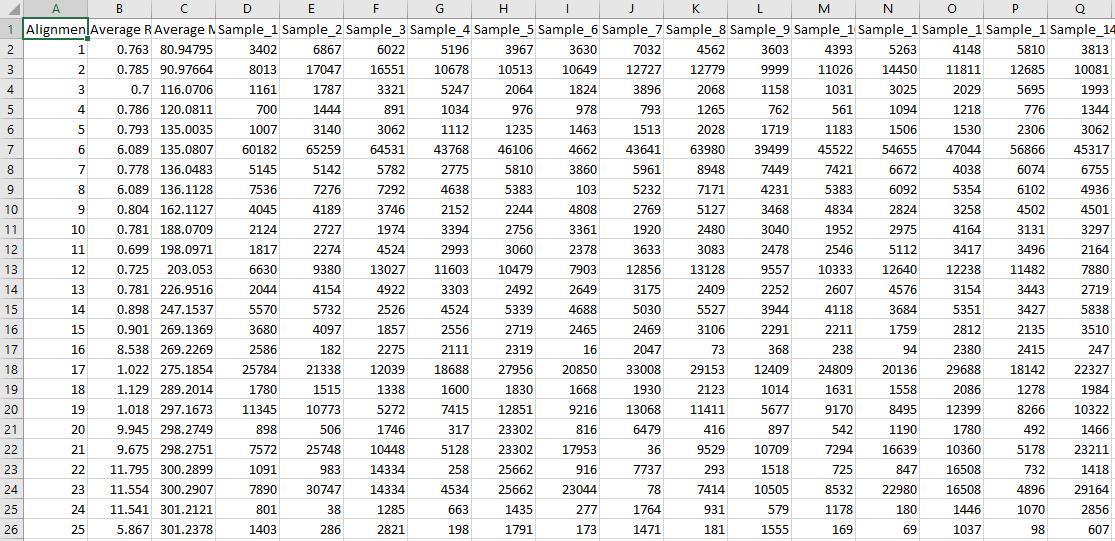
in .csv format)

Column 1: alignment ID.

Column 2: retention time.

Column 3: m/z value.

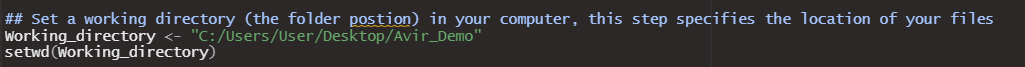
Column 4 to the last column: MS signal intensities of real samples.



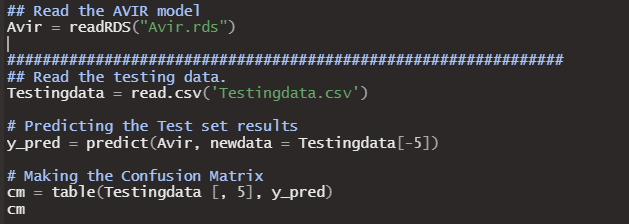
**Note**: in this step, you can try with the provided Demo data.

5) Open the Avir.R script (see below) in RStudio and assign the data path for the folder that contains Avir.rds and other required input files.

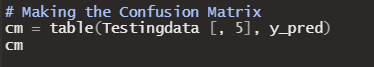
Set up the working directory, for demonstration, here we created a folder in the desktop named” “Avir\_Demo”. Running these two lines will specify the data processing location in RStudio. Then you can read and output the files in this folder.

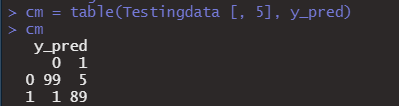


6) Read the model Avir from “Avir.rds”. And use the provided testing data with labels to check the performance of Avir.



Running the following code allows you to get the confusion matrix, you should get the output shown below:





7) To apply Avir in your data processing, it is convenient to just use provided code to calculate the input values of Avir model. Follow the same format as my demo files, prediction can be finished easily using the following code.

Sample metabolite-intensity (both in peak area and peak height) table containing all real samples should be .csv format, the format should be prepared as follows:

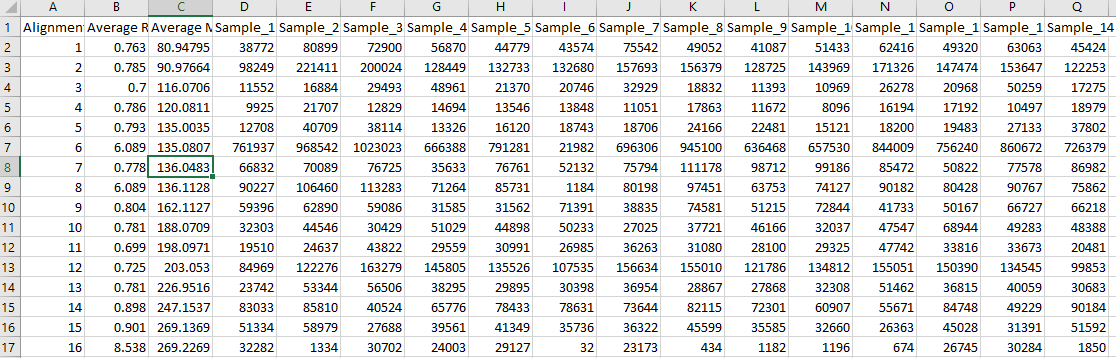
Column 1: alignment ID.

Column 2: retention time.

Column 3: m/z value.

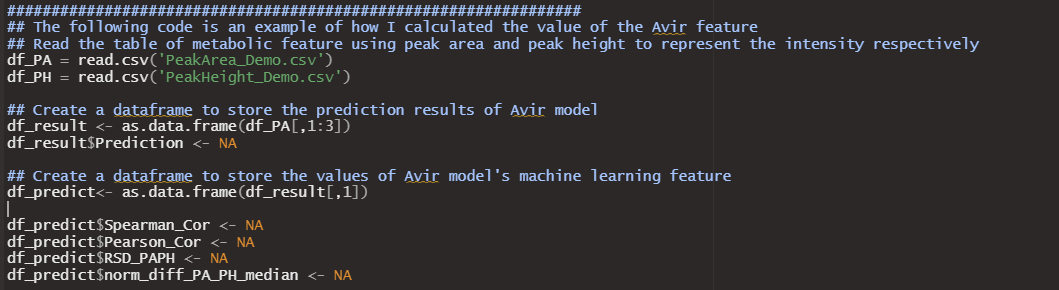
Column 4 to the last column: MS signal intensities of real samples.

**Note:** for customized input ofsample metabolite-intensity table, any software that can output intensity as peak area and peak height are acceptable (e.g., XCMS, MS-DIAL).

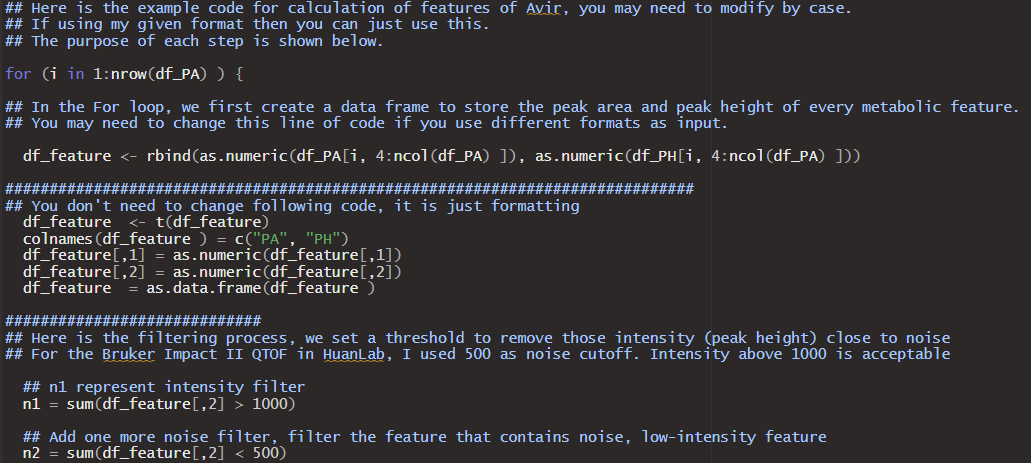


8)Then read the table of metabolic features in peak area and peak height respectively.

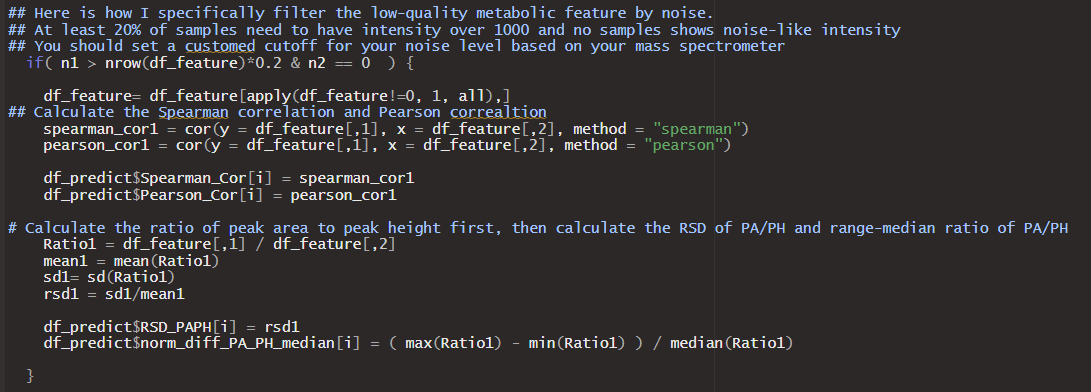
Run the following code directly, and it will create data frame to store the values needed for Avir prediction.

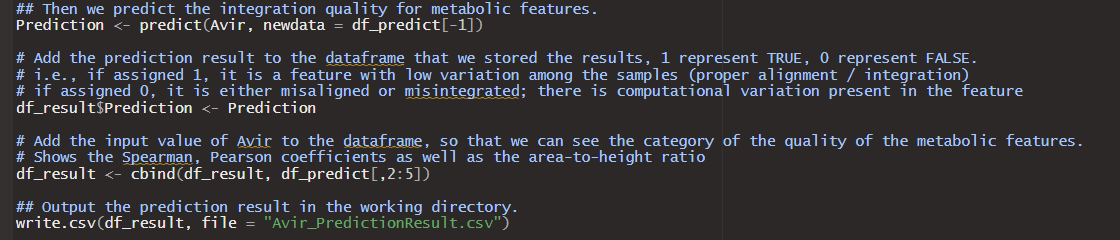


9) Follow the same format of input table above, following code can be directly run to generate the statistical input for Avir. However, it is strongly recommended set an intensity threshold and noise cutoff before Avir prediction. Noise-like peak will interfere the model performance and may decrease the prediction accuracy. It is better to only keep the high-quality metabolic features for prediction. Here is how to filter the metabolic features of low quality. For Impact II QTOF of Bruker, we set 1000 counts as the intensity threshold and 500 as the noise cutoff. Intensity below 500 will be considered noise and the metabolic features that contain noise will be excluded. The metabolic features of low reproducibility are also excluded. At least 20% of the samples need to show intensity above 1000, so that those will be considered as high-quality features.



The intensity and noise level should be set based on your own mass spectrometer, and the reproducibility filter can be customized in the given code.



10) After obtaining the required Avir input, run the following the code to get the Avir prediction and output it in the folder. 

**Design of Avir and Discussion**

The design of Avir is to detect outlier in a consistent peak integration pattern. We assume that the same metabolic feature generates similar peak shape, an ideal peak integration in all samples should be similar. Notice that peak area is positively correlated with peak height in an ideal situation, where all samples have a good Gaussian peak shape and it is integrated correctly. When there is wrong peak integration, the correlation of peak area and peak height looks like a outlier, therefore, we can use the correlation between peak area and peak height to quantitatively represent the similarity of peak integration among all samples.

At first, I try to establish linear regression model between peak area and peak height. I used Cook’s distance and studentized residual to represent the deviation of each sample in the regression model. But later I found that it may not be general enough to distinguish good integration pattern from the bad one. I also tried the Pearman correlation, Spearman correlation and the intensity of metabolic feature. All these factors alone cannot achieve the separation of good and bad. Later I came up with the idea of the ratio of peak area to peak height, which turns out to work well. The ratio of peak area to peak height is a simple and effective measurement to reflect the peak shape, which help achieve the classification task for most of cases. To obtain a robust and comprehensive model, I decided to combine all the factors I thought of into making a machine learning model.

Since different LC-MS platform have different levels of intensity and peak width, to generalize this model to other LC-MS platform, I removed the machine learning feature of intensity of mass spectrometry and the expected ratio of peak area to peak height.

For feature engineering, I designed two machine learning features of PA/PH to detect outliers: relative standard deviation of PA/PH and normalized range-to-median of PA/PH. Each machine learning feature is metabolic-feature specific, which means that I established an evaluation model for the metabolic feature in all samples to identify if the peak integration is correct. I keep the Pearman correlation and Spearman correlation here as it can improve the robustness of the model.

The false negative rate is a bit high in some LC-MS platforms when I did external validation, but it is fine because false positive here hurts more in untargeted metabolomics. Manual inspection and correction of the prediction of FALSE is needed if you care about all metabolic features. I noticed that including noise-like sample in the prediction process will influence the model performance, so I set a noise cut-off to exclude those metabolic features, and these metabolic features require manual inspection as well.

I would agree that building a customized machine learning model to complete this classification in your own LC-MS platform will yield a more accurate and robust performance. Labeling your in-house data to train the model and add more LC-MS specific machine learning feature will be helpful. Feel free to contact me for developing your own model or discuss any details about the topic, I am happy to help.