Supporting notes for:

A deep learning-based classification and prediction system for deciphering the characteristics of clinical protome raw mass spectrometry data

Table of Contents

I. Supplementary Notes

I. Supplementary Notes

1. Brief Description

In this study, we presented a free and powerful platform, named MSpectraAI (Mass Spectra Artificial Intelligence), as an easy-to-use stand-alone software for mining and classifying raw LC-MS2-based proteomics or metabolomics data of different samples using deep learning models. Users can also built your own deep neural network model in this software. To date, this platform contains:

1) Feature swath extraction, all collected mass spectra are acquired consistently with sequential windows;

2) Samples classification, different group samples can be tested and predicted using artificial neural networks model;

3) Visualization, the fingerprint of mass spectra and model prediction results are shown as vector graphs or table data.

2. Availability

MSpectraAI is an open source web platform, which initiative available in the GitHub repository: https://github.com/wangshisheng/MSpectraAI. An example is shown here: https://www.omicsolution.org/wukong/MSpectraAI/, to which users can also import their data.

3. Friendly Tips

- Run this tool locally. As we know, the raw data from mass spectrometer are usually very large. You can analyze your data on our web server, but the analysis speed will be slower.
- Be familiar with the basic usage of R language. This web tool is developed with R, therefore, if you know some basic knowledge about R, it will help you understand this tool better. However, you need not worry if you know nothing about R, and you can learn to use our tool expertly as well after reading our manual.

4. How to install third-party softwares

- ■ Install R. You can download R from here: https://www.r-project.org/. We recommend the R version >= 3.5.0.
- ■ Install RStudio (Recommendatory but not necessary). You can download RStudio from here: https://www.rstudio.com/. If you decide to use the script editor, we recommend the version >= 1.1.423.
- ■ Install Anaconda (For Windows users). You can download Anaconda from here: https://www.anaconda.com/download/.
- ■ Install RawConverter [7]. Download from here: http://fields.scripps.edu/rawconv/. Optionally, you can also use similar tools, such as MSConvert [8], which can be downloaded from here: http://proteowizard.sourceforge.net/tools.shtml.

5. Installing R packages

We recommend the R version >= 3.5.0. Particularly, if you use unix-like systems, you may need install some dependent packages in advance, for example, on CentOS 7:

```
sudo yum -y install libxml2-devel igraph-devel libxslt-devel netcdf-devel \
libcurl-devel openssl-devel cairo-devel
pip install virtualenv
##Then open R
#Packages
needpackages<-c("BiocManager","devtools","shiny","shinyjs","shinyBS","ggplot2",
"ggjoy","reshape","openxlsx","gdata","DT","gtools","ggsci","mzR","plyr","tidyr",
"abind","data.table","parallel","ggrastr","ggthemes","viridis",
"glue","ComplexHeatmap","impute","circlize","ROCR","keras")

#Check and install function
CheckInstallFunc <- function(x){
    for( i in x ){
        #    require returns TRUE invisibly if it was able to load package
        if(!require(i, character.only = TRUE )){
            #    If package was not able to be loaded then re-install
            install.packages(i, dependencies = TRUE)
            if(!require(i, character.only=TRUE)) BiocManager::install(i, dependencies = TRUE)
            if(i=="ggrastr"){
                devtools::install_github('VPetukhov/ggrastr')
            }
        }
    }
}

#Start to check and install
CheckInstallFunc(needpackages)
#R interface to Keras: https://keras.rstudio.com/
library(keras)
install_keras()
```

The default installation of Keras is CPU, so you want GPU if your computer supports, you should use this commad: install_keras(tensorflow = "gpu"). And the detailed introduction of GPU installation can be found here: https://keras.rstudio.com/reference/install_keras.html.

6. Browser compatibility

MSpectraAI can be processed on Windows, Linux, and Mac operating system. We have tested it as below:

OS	Version	Chrome	Firefox	Safari
Windows	7	68.0.3440.106	63.0.3	not tested
Linux	CentOS 7	not tested	52.8.0	not tested
MacOS	HighSierra	70.0.3538.110	not tested	12.0.1

7. Data Preparation

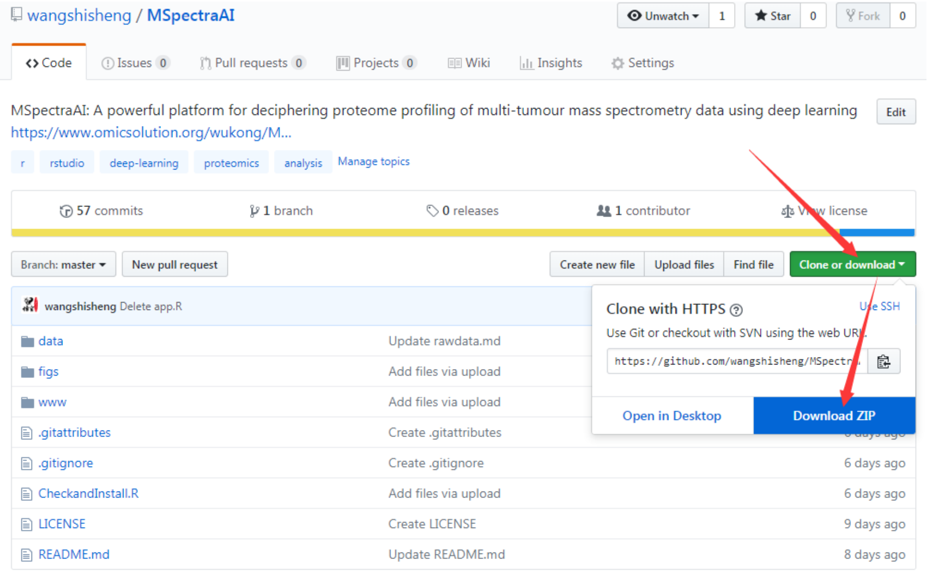
Users can obtain the mass spectra data in their own laboratory. Otherwise, these raw data can be downloaded from some public database, such as ProteomeXchange Consortium (http://www.proteomexchange.org/), where users can search raw data uploaded from other labs across the world. However, users should notice that the ideal raw data are limited and not always found for special analysis. Fortunately, we collected six tumor type data and analysed them with deep neural network model in MSpectraAI. The detailed sample information is listed in supplementary table S1.

In consideration of running speed and time, we reconstruted some small-size raw data from nonsmall cell lung cancer samples. But the whole process of analysis is totally identical in comparison with calculation of large-size data. These small-size raw data were also uploaded to the same github as mentioned above for users to download.

8. Running MSpectraAI locally

Once you install R and relative packages well, it would be quite easy to run this tool locally on by two lines of code.

First, download this tool from the github (https://github.com/wangshisheng/MSpectraAI), like this:



MSpectraAI: A powerful platform for deciphering proteome profiling of multi-tumour mass spectrometry data using deep learning
https://www.omicsolution.org/wukong/M...

Use Git or checkout with SVN using the web URL.

```
https://github.com/wangshisheng/MSpectr
```

wangshisheng Delete app.R

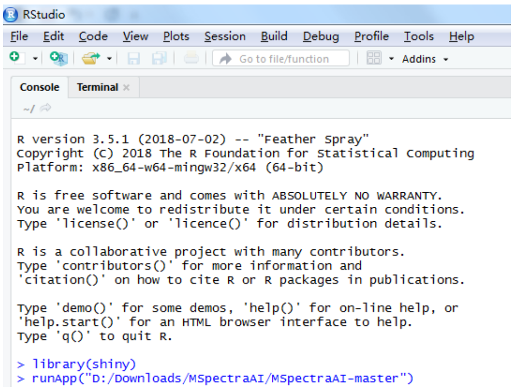
data	Update rawdata.md	
figs	Add files via upload	
www	Add files via upload	
.gitattributes	Create .gitattributes	6 days ago
.gitignore	Create .gitignore	6 days ago
CheckandInstall.R	Add files via upload	6 days ago
LICENSE	Create LICENSE	9 days ago
README.md	Update README.md	8 days ago

The whole file is about 180MB, so it may take some time.

Second, if you download successfully, unzip this file:

```
MSpectraAI-master          2018/11/23 12:57
MSpectraAI-master          2018/11/23 13:04
```

Third, open R-GUI or RStudio. Here, we use RStudio and then find file path, run these codes as below:

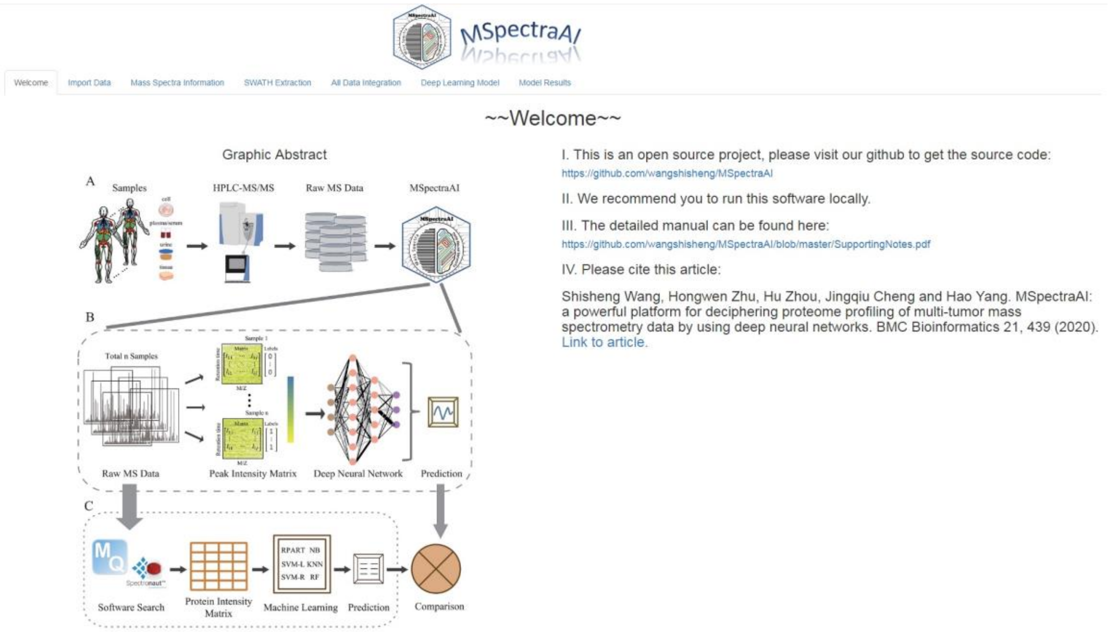


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Console
Terminal
~/

```
R version 3.5.1 (2018-07-02) -- "Feather Spray"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> library(shiny)
> runApp("D:/Downloads/MSpectraAI/MSpectraAI-master")
```

In my computer, the file path is “D:/Downloads/MSpectraAI/MSpectraAI-master”, but yours may be different, so you need change it.

Now, MSpectraAI is activated successfully through listening on a local link. In my computer, it is: http://127.0.0.1:6201. Then you can copy this link to a browser, such as Chrome:



~~Welcome~~

I. This is an open source project, please visit our github to get the source code:
https://github.com/wangshisheng/MSpectraAI

II. We recommend you to run this software locally.

III. The detailed manual can be found here:
https://github.com/wangshisheng/MSpectraAI/blob/master/SupportingNotes.pdf

IV. Please cite this article:

Shisheng Wang, Hongwen Zhu, Hu Zhou, Jingqiu Cheng and Hao Yang. MSpectraAI: a powerful platform for deciphering proteome profiling of multi-tumor mass spectrometry data by using deep neural networks. BMC Bioinformatics 21, 439 (2020). Link to article.

The detailed information about the current R session is shown below:

```
> sessionInfo()
R version 3.5.1 (2018-07-02)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 7 x64 (build 7601) Service Pack 1

Matrix products: default

locale:
[1] LC_COLLATE=Chinese (Simplified)_People's Republic of China.936    LC_CTYPE=Chinese
(Simplified)_People's Republic of China.936
[3] LC_MONETARY=Chinese (Simplified)_People's Republic of China.936 LC_NUMERIC=C
[5] LC_TIME=Chinese (Simplified)_People's Republic of China.936

attached base packages:
[1] grid        parallel  stats       graphics  grDevices utils       datasets  methods    base

other attached packages:
  [1] keras_2.1.6              ROCR_1.0-7                 gplots_3.0.1              circlize_0.4.4
ComplexHeatmap_1.18.1
  [6] glue_1.3.0               viridis_0.5.1            viridisLite_0.3.0        ggthemes_4.0.0
ggrastr_0.1.5
[11] data.table_1.11.8        abind_1.4-5                tidyr_0.8.2                plyr_1.8.4
impute_1.53.0
[16] mzR_2.13.6                Rcpp_0.12.19               ggsci_2.8                    gtools_3.5.0
DT_0.4
[21] gdata_2.18.0              openxlsx_4.0.17          ggjoy_0.4.1                ggridges_0.5.0
ggplot2_3.1.0
[26] shinyBS_0.61              shinyjs_1.0                shiny_1.2.0

loaded via a namespace (and not attached):
  [1] ProtGenerics_1.11.0 bitops_1.0-6            RColorBrewer_1.1-2    tools_3.5.0              R6_2.2.2
KernSmooth_2.23-15
  [7] lazyeval_0.2.1          BiocGenerics_0.26.0 colorspace_1.3-2       GetoptLong_0.1.7
withr_2.1.2               tidyselect_0.2.5
[13] gridExtra_2.3            compiler_3.5.0          Biobase_2.39.2          Cairo_1.5-9
labeling_0.3              caTools_1.17.1.1
[19] scales_1.0.0             tfruns_1.3                stringr_1.3.1           digest_0.6.18
base64enc_0.1-3         pkgconfig_2.0.1
[25] htmltools_0.3.6         htmlwidgets_1.3         rlang_0.3.0.1           GlobalOptions_0.1.0
rstudioapi_0.7           shape_1.4.4
[31] bindr_0.1.1              jsonlite_1.5              tensorflow_1.8          crosstalk_1.0.0
dplyr_0.7.7               magrittr_1.5
```

```
[37] Matrix_1.2-14          munsell_0.5.0          reticulate_1.9          stringi_1.1.7
whisker_0.3-2          yaml_2.1.19
[43] promises_1.0.1          crayon_1.3.4            lattice_0.20-35         zeallot_0.1.0
pillar_1.2.1            rjson_0.2.19
[49] codetools_0.2-15       httpuv_1.4.4.1          gtable_0.2.0            purrr_0.2.4.9000
reshape_0.8.7           assertthat_0.2.0
[55] mime_0.5
```

9. Operation step by step

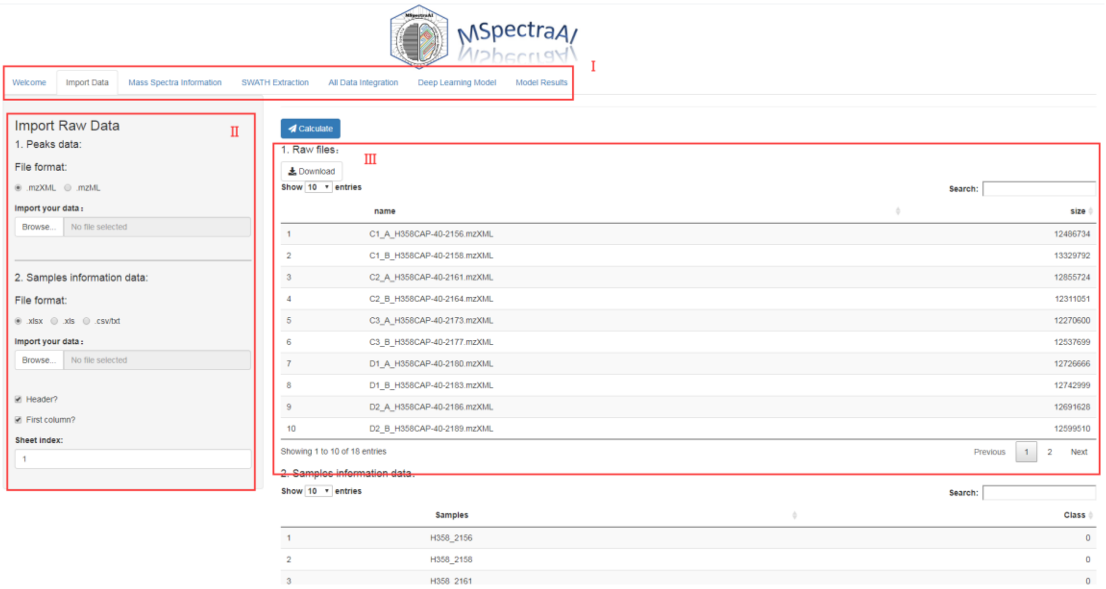
9.1 Graphical user interface of MSpectraAI

There are three main parts in this software:

I. Function names. All principle functions are displayed in the menu.

II. Parameter tuning panel. Users can regulate parameters conveniently here according to their own data.

III. Results panel. After uploading data or adjusting parameter, click “Calculate” button, the results will be shown here immediately.



Import Raw Data

1. Peaks data:

File format:

● .mzXML ○ .mzML

Import your data：

No file selected

2. Samples information data:

File format:

● .xlsx ○ .xls ○ .csv/txt

Import your data：

No file selected

☑ Header?

☑ First column?

Sheet index:

1

1. Raw files：

Show 10 entries

Search:

	name	size
1	C1_A_H358CAP-40-2156.mzXML	12486734
2	C1_B_H358CAP-40-2158.mzXML	13329792
3	C2_A_H358CAP-40-2161.mzXML	12855724
4	C2_B_H358CAP-40-2164.mzXML	12311051
5	C3_A_H358CAP-40-2173.mzXML	12270600
6	C3_B_H358CAP-40-2177.mzXML	12537699
7	D1_A_H358CAP-40-2180.mzXML	12726666
8	D1_B_H358CAP-40-2183.mzXML	12742999
9	D2_A_H358CAP-40-2186.mzXML	12691628
10	D2_B_H358CAP-40-2189.mzXML	12599510

Showing 1 to 10 of 18 entries

2. Samples information data：

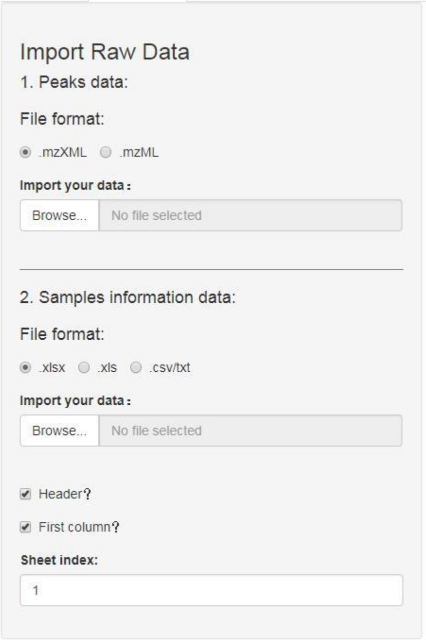
Show 10 entries

Search:

	Samples	Class
1	H358_2156	0
2	H358_2158	0
3	H358_2161	0

9.2 Importing data

Click “Import Data” name in the menu, then you can upload your data from here. In default, the software will load our example data. Once you upload your own data, the results panel will show the results of your data.



Import Raw Data

1. Peaks data:

File format:

● .mzXML ○ .mzML

Import your data:

2. Samples information data:

File format:

● .xlsx ○ .xls ○ .csv/txt

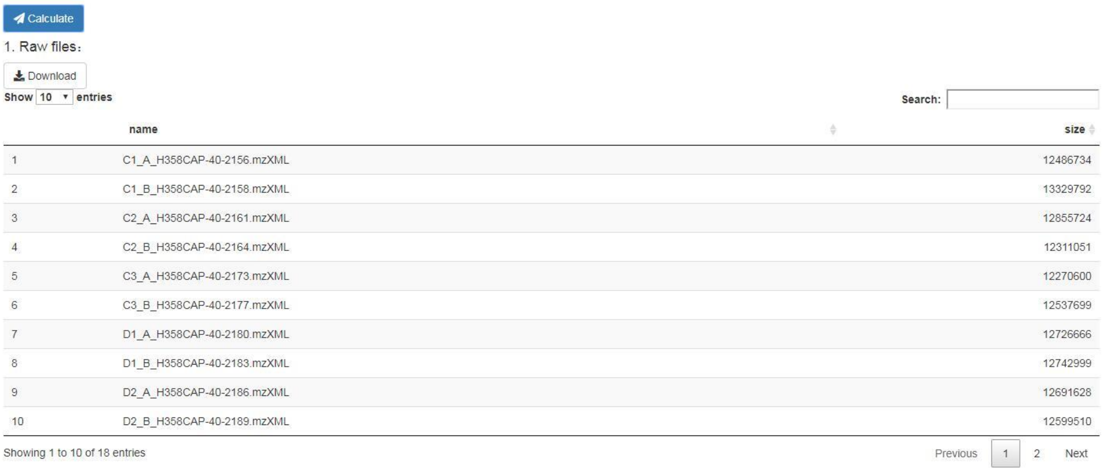
Import your data:

☑ Header?

☑ First column?

Sheet index: 1

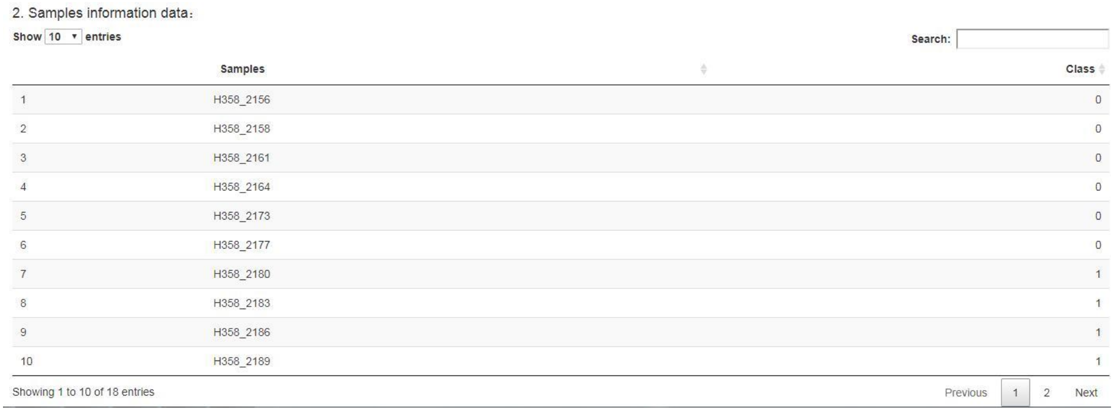
Here you should upload two kinds of data. First, the mzXML or mzML files that converted from raw data using RawConverter or MSConvert software as mentioned above. Second, the sample information data that record the file names and class labels. Once you prepare these data, click “browser”, the results will be shown like this:



1. Raw files:

	name	size
1	C1_A_H358CAP-40-2156.mzXML	12486734
2	C1_B_H358CAP-40-2158.mzXML	13329792
3	C2_A_H358CAP-40-2161.mzXML	12855724
4	C2_B_H358CAP-40-2164.mzXML	12311051
5	C3_A_H358CAP-40-2173.mzXML	12270600
6	C3_B_H358CAP-40-2177.mzXML	12537699
7	D1_A_H358CAP-40-2180.mzXML	12726666
8	D1_B_H358CAP-40-2183.mzXML	12742999
9	D2_A_H358CAP-40-2186.mzXML	12691628
10	D2_B_H358CAP-40-2189.mzXML	12599510

Showing 1 to 10 of 18 entries



2. Samples information data：

Show 10 ▾ entries

Search:

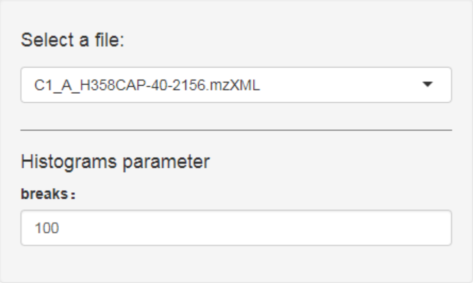
	Samples	Class
1	H358_2156	0
2	H358_2158	0
3	H358_2161	0
4	H358_2164	0
5	H358_2173	0
6	H358_2177	0
7	H358_2180	1
8	H358_2183	1
9	H358_2186	1
10	H358_2189	1

Showing 1 to 10 of 18 entries

The “name” in “Raw files” result means raw data filenames, “size” means the file size. The “Samples” in “Sample information data” result means sample filenames (also raw data filenames), whose order should be same as raw data filenames. “Class” means category labels, which should be numbers starting from 0.

9.3 Mass Spectra Information

The peaks number in every spectra (shown as histogram), the MS1 spectra number, and the MS2 spectra number (shown in a table) are counted in this part.



Select a file:

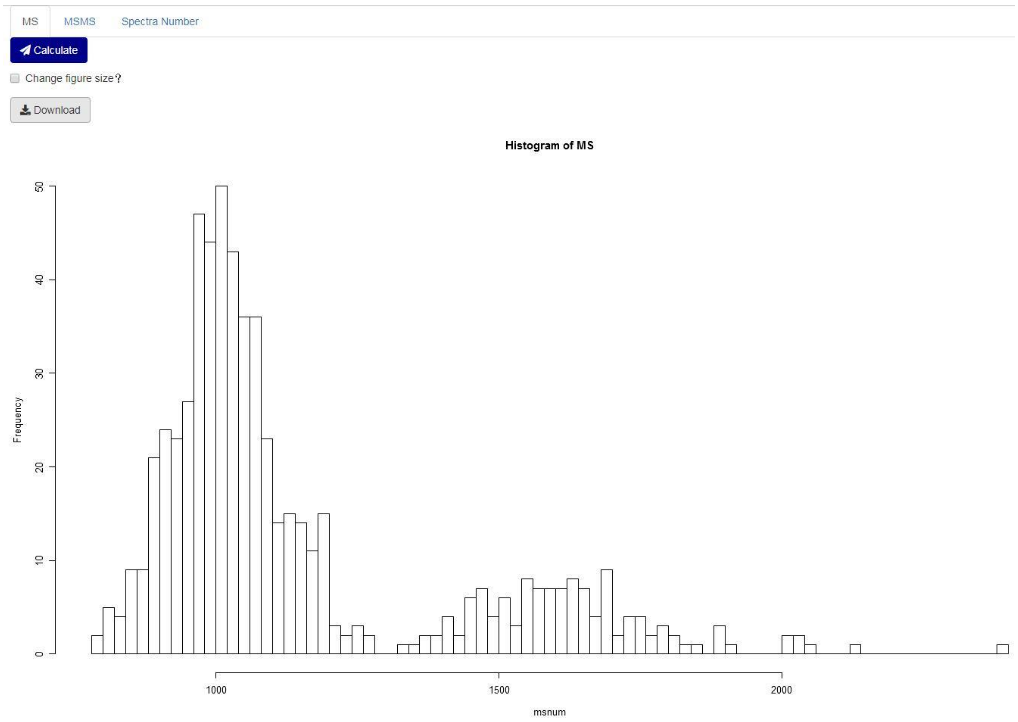
C1_A_H358CAP-40-2156.mzXML

Histograms parameter

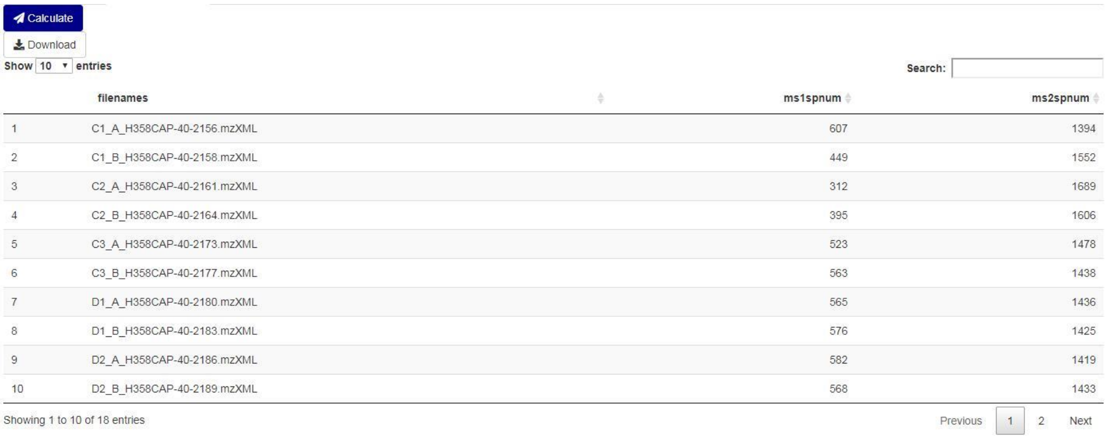
breaks:

100

You can select any file and the tool calculates corresponding results immediately. Then click "Download" button, the figures will be saved as pdf files and the tables will be saved as csv files.



□ Change figure size?



Show 10 entries

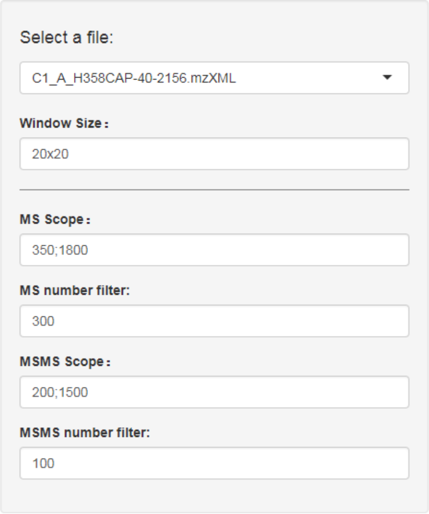
Search:

	filenames	ms1spnum	ms2spnum
1	C1_A_H358CAP-40-2156.mzXML	607	1394
2	C1_B_H358CAP-40-2158.mzXML	449	1552
3	C2_A_H358CAP-40-2161.mzXML	312	1689
4	C2_B_H358CAP-40-2164.mzXML	395	1606
5	C3_A_H358CAP-40-2173.mzXML	523	1478
6	C3_B_H358CAP-40-2177.mzXML	563	1438
7	D1_A_H358CAP-40-2180.mzXML	565	1436
8	D1_B_H358CAP-40-2183.mzXML	576	1425
9	D2_A_H358CAP-40-2186.mzXML	582	1419
10	D2_B_H358CAP-40-2189.mzXML	568	1433

Showing 1 to 10 of 18 entries

9.4 Swath extraction

Features can be extracted in a certain window size. Parameters can be regulated as below:



Select a file:

C1_A_H358CAP-40-2156.mzXML

Window Size:

20x20

MS Scope:

350;1800

MS number filter:

300

MSMS Scope:

200;1500

MSMS number filter:

100

Select a file: users can select a file that they want to analyse.

Window Size: how many windows across the whole *m/z* range. For example, “20x20” means there are total 400 windows and then the whole *m/z* range will be divided into 400 parts.

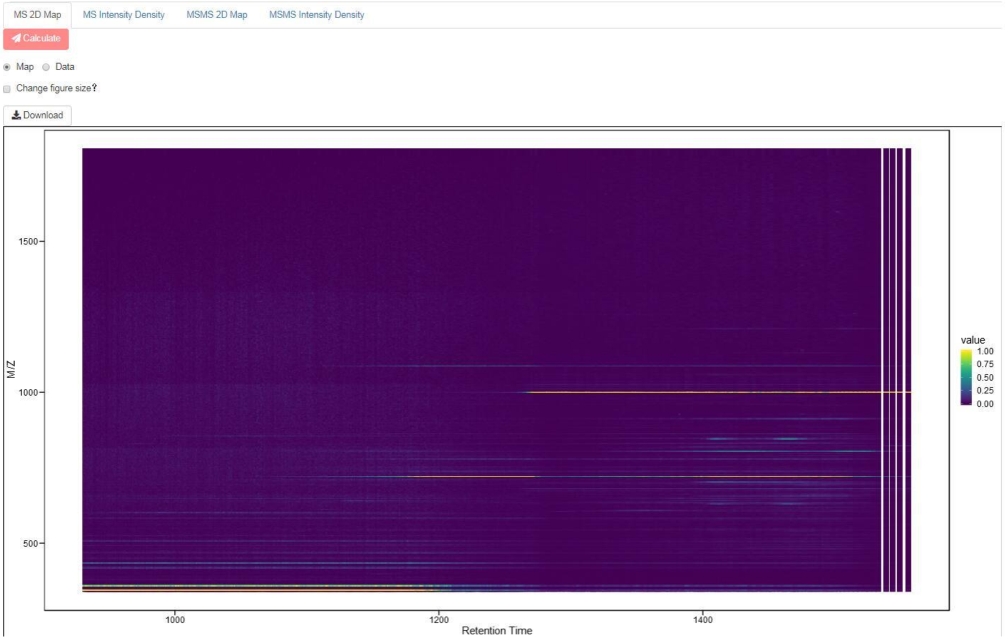
MS Scope: the *m/z* range of MS scan, which in linked by “;”.

MS number filter: those MS scan whose peaks number are below this threshold will be deleted.

MSMS Scope: the *m/z* range of MS2 scan, which in linked by “;”.

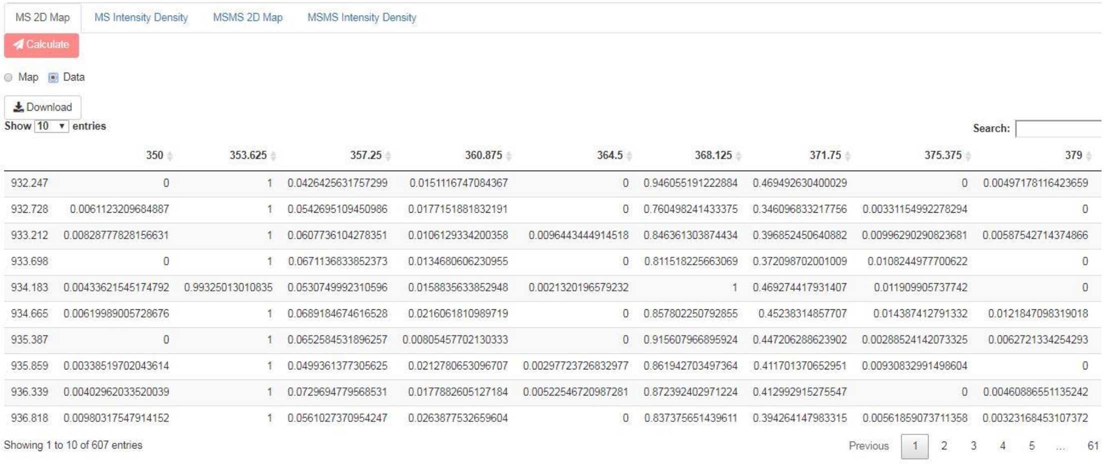
MSMS number filter: those MS2 scan whose peaks number are below this threshold will be deleted.

Then the intensity distribution across *m/z* and retention time dimensions as 2D map and corresponding data matrix table can be calculated here:



● Map ○ Data

□ Change figure size?



○ Map ● Data

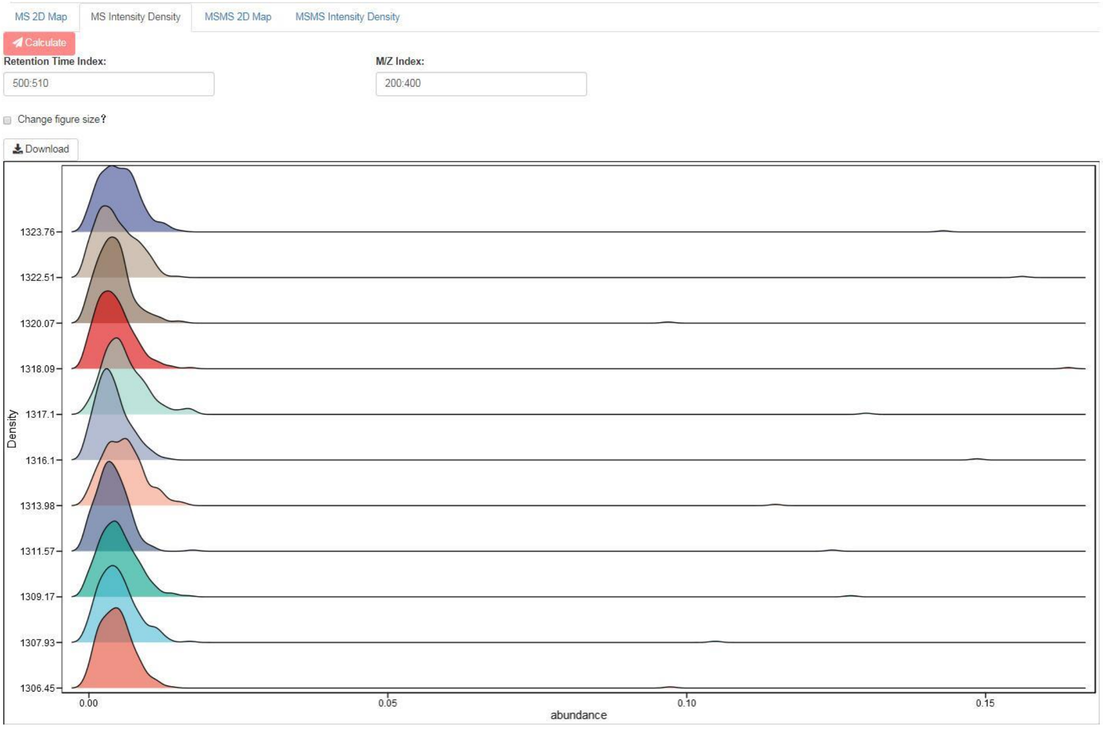
Show 10 ▾ entries

Search:

	350	353.625	357.25	360.875	364.5	368.125	371.75	375.375	379
932.247	0	1	0.0426425631757299	0.0151116747084367	0	0.946055191222884	0.469492630400029	0	0.00497178116423659
932.728	0.0061123209684887	1	0.0542695109450986	0.0177151881832191	0	0.760498241433375	0.346096833217756	0.00331154992278294	0
933.212	0.00828777828156631	1	0.0607736104278351	0.0106129334200358	0.0096443444914518	0.846361303874434	0.396852450640882	0.00996290290823681	0.00587542714374866
933.698	0	1	0.0671136833852373	0.0134680606230955	0	0.811518225663069	0.372098702001009	0.0108244977700622	0
934.183	0.00433621545174792	0.99325013010835	0.0530749992310596	0.0158835633852948	0.0021320196579232	1	0.469274417931407	0.011909905737742	0
934.665	0.00619989005728676	1	0.0689184674616528	0.0216061810989719	0	0.857802250792855	0.45238314857707	0.014387412791332	0.0121847098319018
935.387	0	1	0.0652584531896257	0.00805457702130333	0	0.915607966895924	0.447206288623902	0.00288524142073325	0.0062721334254293
935.859	0.00338519702043614	1	0.0499361377305625	0.0212780653096707	0.00297723726832977	0.861942703497364	0.411701370652951	0.00930832991498604	0
936.339	0.00402962033520039	1	0.0729694779568531	0.0177882605127184	0.00522546720987281	0.872392402971224	0.412992915275547	0	0.00460886551135242
936.818	0.00980317547914152	1	0.0561027370954247	0.0263877532659604	0	0.837375651439611	0.394264147983315	0.00561859073711358	0.00323168453107372

Showing 1 to 10 of 607 entries

The intensity density can also be displayed here:



Retention Time Index: 500:510

M/Z Index: 200:400

□ Change figure size?

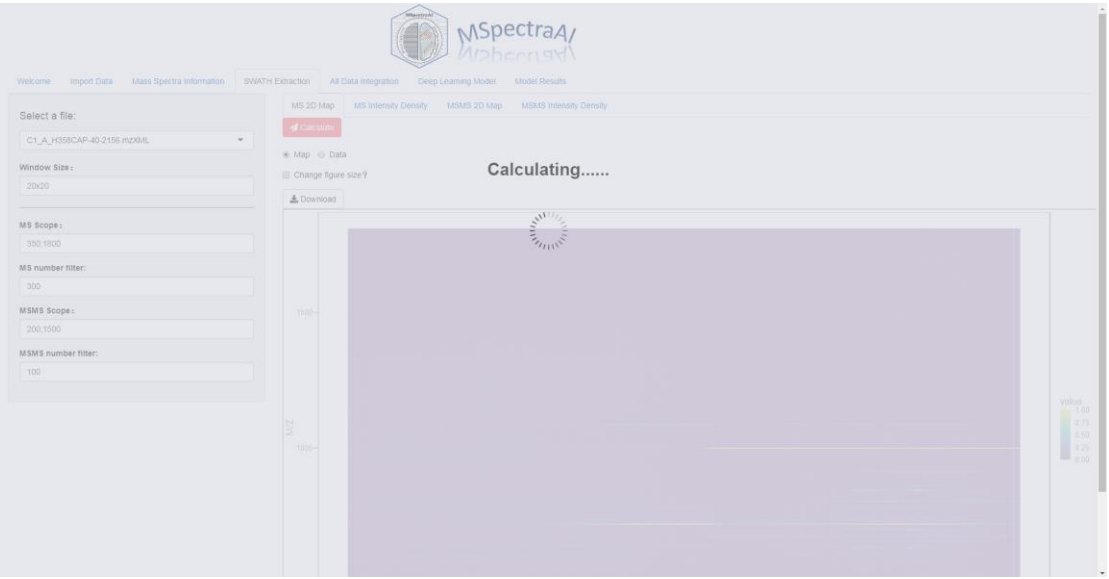
Retention Time Index: which spectra across retention time dimension are extracted to calculate the density.

M/Z Index: which spectra across *m/z* dimension are extracted to calculate the density.

All results are similar in MSMS spectra (not shown here).

9.5 All Data Integration

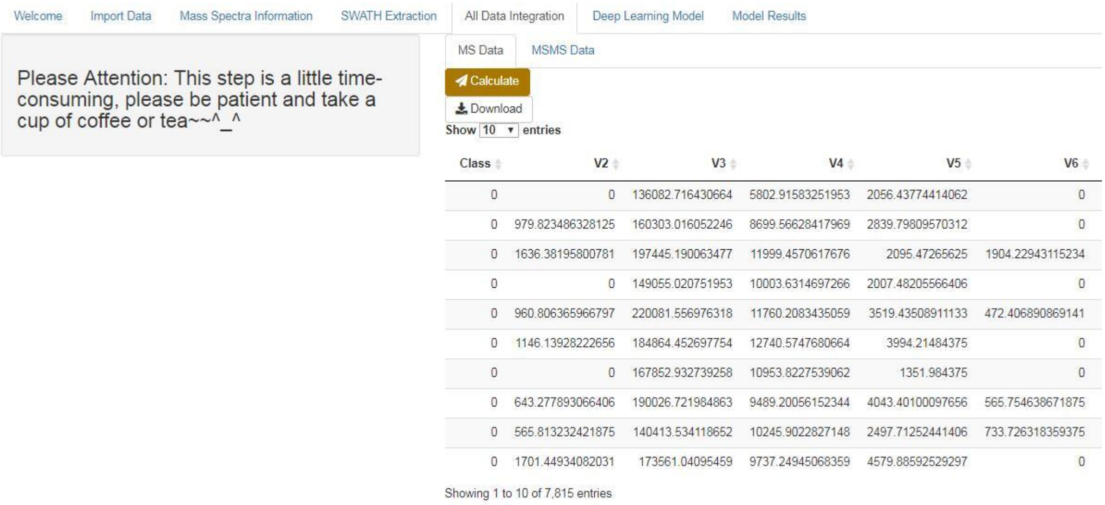
In “9.4 Swath extraction” part, the features are extracted from one file that users select at a time. Here, all files that users upload are extracted and combined together for MS and MS2 spectra data, so the analysis time of this step is a little long. Users should be patient:



Select a file: C1_A_H358CAP-40-2156.mzXML
Window Size: 20x20
MS Scope: 350,1800
MS number filter: 300
MSMS Scope: 200,1500
MSMS number filter: 100

Calculating......

The indicator at the bottom right can tell you which file is processing. And then the whole matrix are shown as below and downloadable by clicking “Download” button:



Please Attention: This step is a little time-consuming, please be patient and take a cup of coffee or tea~~^_^

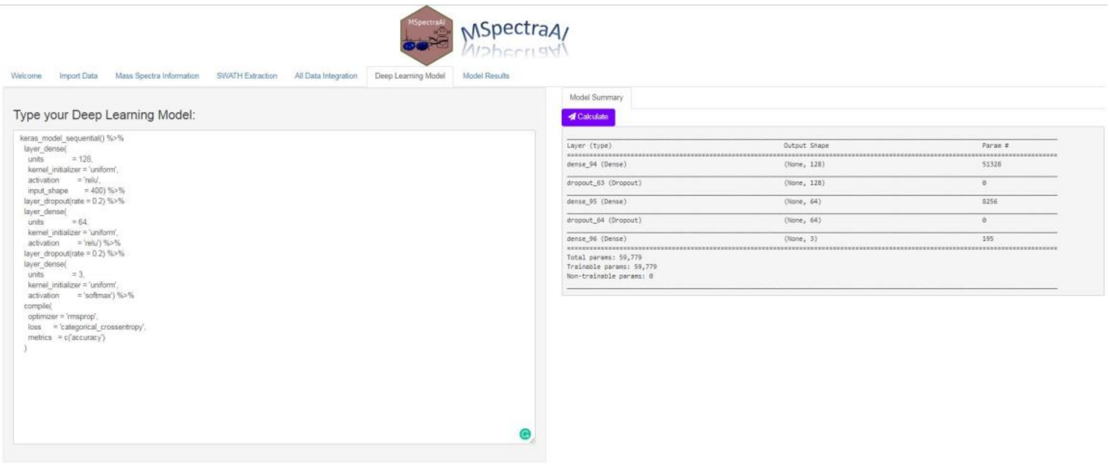
Show 10 entries

Class	V2	V3	V4	V5	V6
0	0	136082.716430664	5802.91583251953	2056.43774414062	0
0	979.823486328125	160303.016052246	8699.56628417969	2839.79809570312	0
0	1636.38195800781	197445.190063477	11999.4570617676	2095.47265625	1904.22943115234
0	0	149055.020751953	10003.6314697266	2007.48205566406	0
0	960.806365966797	220081.556976318	11760.2083435059	3519.43508911133	472.406890869141
0	1146.13928222656	184864.452697754	12740.5747680664	3994.21484375	0
0	0	167852.932739258	10953.8227539062	1351.984375	0
0	643.277893066406	190026.721984863	9489.20056152344	4043.40100097656	565.754638671875
0	565.813232421875	140413.534118652	10245.9022827148	2497.71252441406	733.726318359375
0	1701.44934082031	173561.04095459	9737.24945068359	4579.88592529297	0

Showing 1 to 10 of 7,815 entries

9.6 Deep learning model

In this part, users can obtain the intuition of deep learning model that we build using Keras (https://github.com/fchollet/keras) for the example data, the “Model Summary” will give the general information of the deep learning model we input:



Type your Deep Learning Model:

```
keras_model_sequential() %>%
 layer_dense(
  units             = 128,
  kernel_initializer = 'uniform',
  activation        = 'relu',
  input_shape       = 400) %>%
 layer_dropout(rate = 0.2) %>%
 layer_dense(
  units             = 64,
  kernel_initializer = 'uniform',
  activation        = 'relu') %>%
 layer_dropout(rate = 0.2) %>%
 layer_dense(
  units             = 3,
  kernel_initializer = 'uniform',
  activation        = 'softmax') %>%
 compile(
  optimizer = 'rmsprop',
  loss      = 'categorical_crossentropy',
  metrics   = c('accuracy')
 )
```

```
________________________________________________________________________________
Layer (type)                        Output Shape                      Param #
================================================================================
dense_94 (Dense)                    (None, 128)                       51328
________________________________________________________________________________
dropout_63 (Dropout)                (None, 128)                       0
________________________________________________________________________________
dense_95 (Dense)                    (None, 64)                        8256
________________________________________________________________________________
dropout_64 (Dropout)                (None, 64)                        0
________________________________________________________________________________
dense_96 (Dense)                    (None, 3)                         195
================================================================================
Total params: 59,779
Trainable params: 59,779
Non-trainable params: 0
________________________________________________________________________________
```

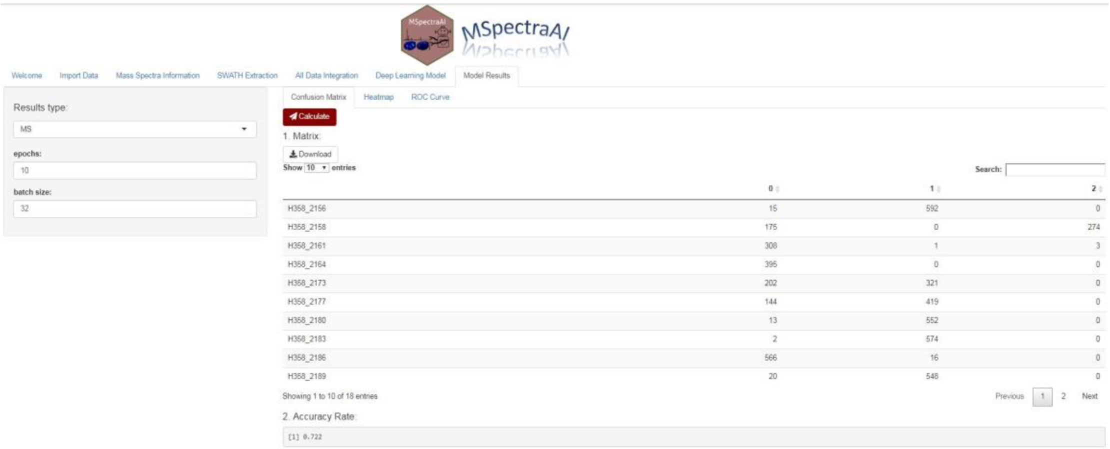
The default code can be changed and shown below:

```
keras_model_sequential() %>%
  layer_dense(
    units                  = 128,
    kernel_initializer = 'uniform',
    activation            = 'relu',
    input_shape          = 400) %>%
  layer_dropout(rate = 0.2) %>%
  layer_dense(
    units                  = 64,
    kernel_initializer = 'uniform',
    activation            = 'relu') %>%
  layer_dropout(rate = 0.2) %>%
  layer_dense(
    units                  = 3,
    kernel_initializer = 'uniform',
    activation            = 'softmax') %>%
  compile(
    optimizer = 'rmsprop',
    loss          = 'categorical_crossentropy',
    metrics     = c('accuracy')
  )
```

In addition, our tool also supports users to design their own deep learning model for their own data in order to obtain more satisfactory results.

9.7 Model Results

Here, the model will train and test mass spectra data. In this process, the mass spectra of one file will be used in testing, the remaining data will be used in training in a for loop, which is similar to “leave-one-out” method. And then the classification results will be displayed including Confusion matrix, Heatmap, and ROC curve.



Results type: MS

epochs: 10

batch size: 32

1. Matrix:

Show 10 entries

Search:

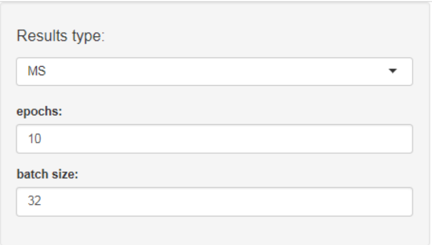
	0	1	2
H358_2156	15	592	0
H358_2158	175	0	274
H358_2161	308	1	3
H358_2164	395	0	0
H358_2173	202	321	0
H358_2177	144	419	0
H358_2180	13	552	0
H358_2183	2	574	0
H358_2186	566	16	0
H358_2189	20	548	0

Showing 1 to 10 of 18 entries

2. Accuracy Rate:

```
[1] 0.722
```

The parameter panel of this part is like this:



Results type: MS

epochs: 10

batch size: 32

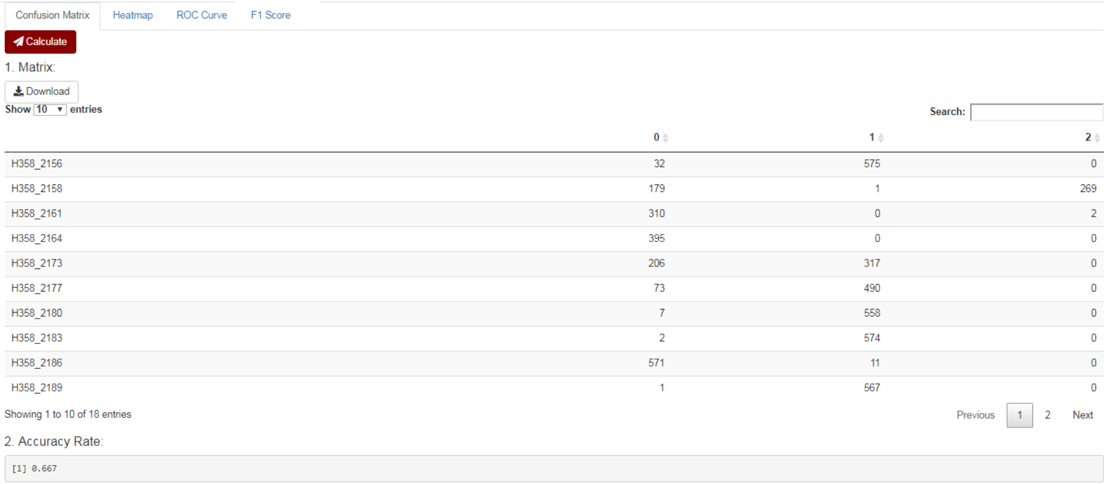
Results type: the results of MS1 or MS2 mass spectra data that users can choose to display.

epochs: number of epochs to train the model in fit function of keras package.

batch size: number of samples per gradient update in fit function of keras package.

Then click “Calculate” button to obtain the results.

For Confusion Matrix:



1. Matrix:

Show 10 entries

Search:

	0	1	2
H358_2156	32	575	0
H358_2158	179	1	269
H358_2161	310	0	2
H358_2164	395	0	0
H358_2173	206	317	0
H358_2177	73	490	0
H358_2180	7	558	0
H358_2183	2	574	0
H358_2186	571	11	0
H358_2189	1	567	0

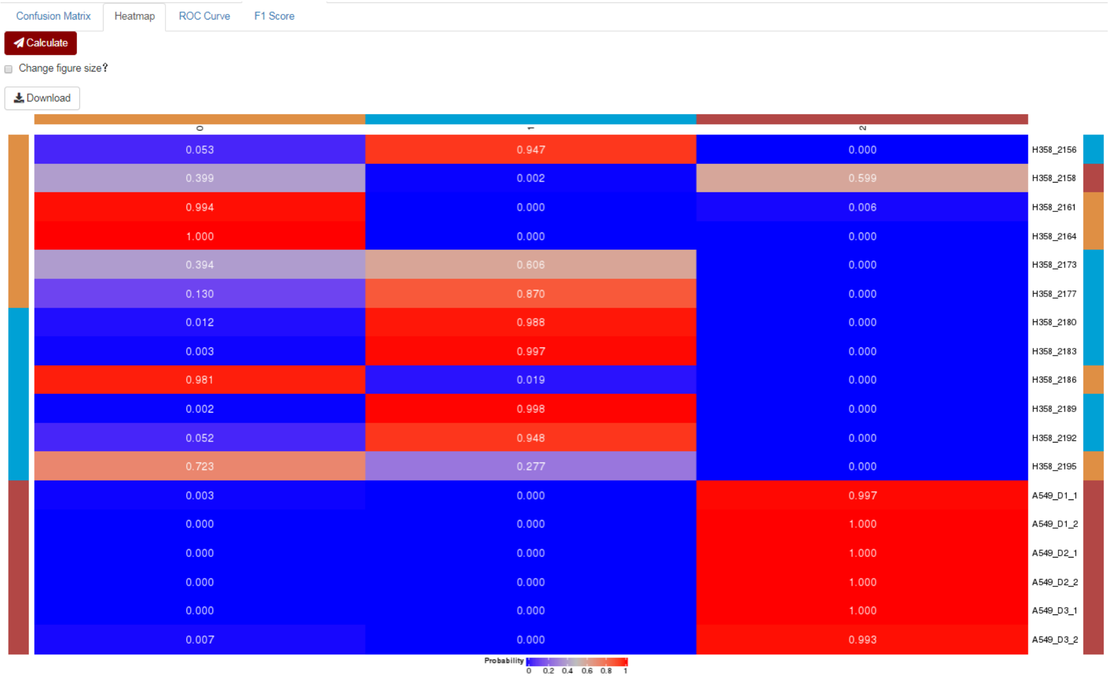
Showing 1 to 10 of 18 entries

2. Accuracy Rate:

```
[1] 0.667
```

Here, the matrix contains predicted mass spectra labels for every sample, on which the final accuracy rate can be calculated based, for example, in “H358_2156” sample, there are total 607 MS1 spectra, and then 592 spectra are predicted as “1” label, whose rate (592/607 = 0.975) is above 0.5 (the default threshold), so this sample is classified as “1”. Repeatedly in this way, every sample can be predicted. If the predicted label is identical to that actual label, we think it is correct. For example data, 13 samples are predicted correctly, so the accuracy rate is 0.722 (13/18).

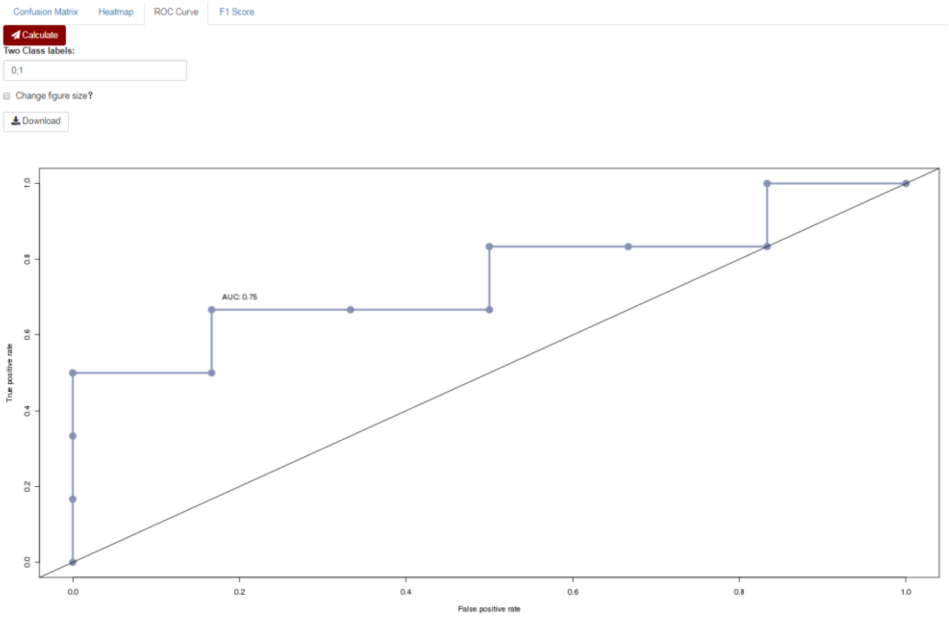
For Heatmap:



	0	1	2
H358_2156	0.053	0.947	0.000
H358_2158	0.399	0.002	0.599
H358_2161	0.994	0.000	0.006
H358_2164	1.000	0.000	0.000
H358_2173	0.394	0.606	0.000
H358_2177	0.130	0.870	0.000
H358_2180	0.012	0.988	0.000
H358_2183	0.003	0.997	0.000
H358_2186	0.981	0.019	0.000
H358_2189	0.002	0.998	0.000
H358_2192	0.052	0.948	0.000
H358_2195	0.723	0.277	0.000
A549_D1_1	0.003	0.000	0.997
A549_D1_2	0.000	0.000	1.000
A549_D2_1	0.000	0.000	1.000
A549_D2_2	0.000	0.000	1.000
A549_D3_1	0.000	0.000	1.000
A549_D3_2	0.007	0.000	0.993

This is mainly visualised for confusion matrix result.

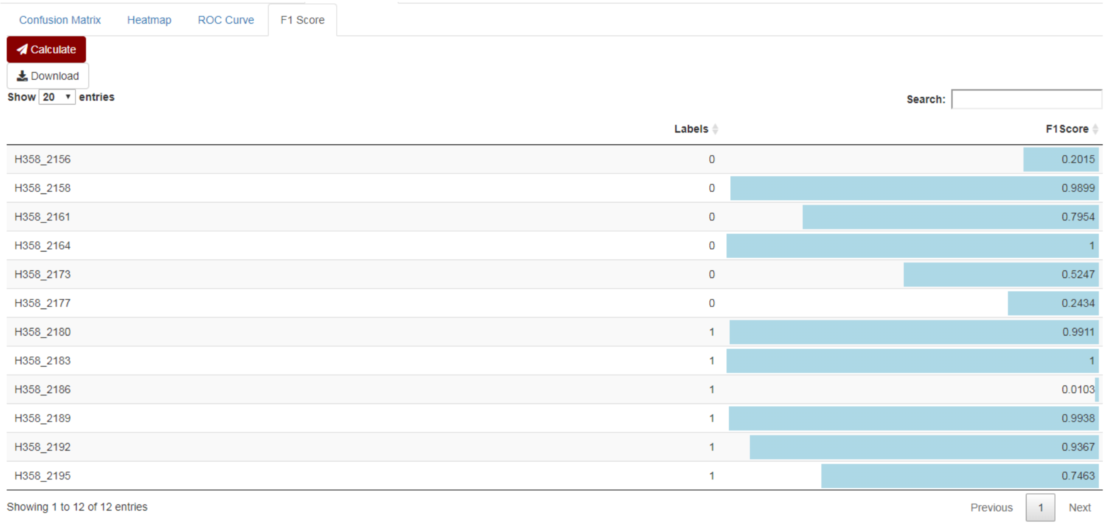
For ROC curve:



Two Class labels:
0;1

The receiver operating characteristic (ROC) curves and the area under the curve (AUC) are calculated using ROCR package [9] for two-category samples which users can assign the class label in “Two Class Labels” box.

For F1 Score:



	Labels	F1Score
H358_2156	0	0.2015
H358_2158	0	0.9899
H358_2161	0	0.7954
H358_2164	0	1
H358_2173	0	0.5247
H358_2177	0	0.2434
H358_2180	1	0.9911
H358_2183	1	1
H358_2186	1	0.0103
H358_2189	1	0.9938
H358_2192	1	0.9367
H358_2195	1	0.7463

Showing 1 to 12 of 12 entries

The F1 score is usually used as a measure of a test's accuracy for statistical analysis of binary classification. The class labels are same as those in “ROC Curve”. The colour bars in the “F1Score” column indicate the magnitude of these F1 scores.

II. References

1. Löffler MW, Kowalewski DJ, Backert L, Bernhardt J, Adam P, Schuster H, Dengler F, Backes D, Kopp H-G, Beckert S: Mapping the HLA ligandome of Colorectal Cancer Reveals an Imprint of Malignant Cell Transformation. *Cancer research* 2018:canres. 1745.2017.
2. Jin J, Son M, Kim H, Kim H, Kong S-H, Kim HK, Kim Y, Han D: Comparative proteomic analysis of human malignant ascitic fluids for the development of gastric cancer biomarkers. *Clinical biochemistry* 2018, 56:55-61.
3. Wiredja DD, Ayati M, Mazhar S, Sangodkar J, Maxwell S, Schlatzer D, Narla G, Koyutürk M, Chance MR: Phosphoproteomics Profiling of Nonsmall Cell Lung Cancer Cells Treated with a Novel Phosphatase Activator. *Proteomics* 2017, 17(22):1700214.
4. Bohnenberger H, Kaderali L, Ströbel P, Yepes D, Plessmann U, Dharia NV, Yao S, Heydt C, Merkelbach ‐ Bruse S, Emmert A: Comparative proteomics reveals a diagnostic signature for pulmonary head ‐ and ‐ neck cancer metastasis. *EMBO molecular medicine* 2018, 10(9):e8428.
5. Zagorac I, Fernandez-Gaitero S, Penning R, Post H, Bueno MJ, Mouron S, Manso L, Morente MM, Alonso S, Serra V: In vivo phosphoproteomics reveals kinase activity profiles that predict treatment outcome in triple-negative breast cancer. *Nature communications* 2018, 9(1):3501.
6. Carnielli CM, Macedo CCS, De Rossi T, Granato DC, Rivera C, Domingues RR, Pauletti BA, Yokoo S, Heberle H, Busso-Lopes AF *et al*: Combining discovery and targeted proteomics reveals a prognostic signature in oral cancer. *Nature communications* 2018, 9(1):3598.
7. He L, Diedrich J, Chu Y-Y, Yates III JR: Extracting accurate precursor information for tandem mass spectra by RawConverter. *Anal Chem* 2015, 87(22):11361-11367.
8. Adusumilli R, Mallick P: Data conversion with ProteoWizard msConvert. In: *Proteomics.* Springer; 2017: 339-368.
9. Sing T, Sander O, Beerenwinkel N, Lengauer T: ROCR: visualizing classifier performance in R. *Bioinformatics* 2005, 21(20):3940-3941.
10. Cox J, Mann M: MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nature biotechnology* 2008, 26(12):1367-1372.
11. Wang S, Li W, Hu L, Cheng J, Yang H, Liu Y: NAguideR: performing and prioritizing missing value imputations for consistent bottom-up proteomic analyses. *Nucleic acids research* 2020.
12. Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, Botstein D, Altman RB: Missing value estimation methods for DNA microarrays. *Bioinformatics* 2001, 17(6):520-525.
13. Ihaka R, Gentleman R: R: a language for data analysis and graphics. *Journal of computational and graphical statistics* 1996, 5(3):299-314.
14. Allaire J, Chollet F: keras: R Interface to'Keras'. *R package version* 2019, 2(4).