System Manual

Installation and Usage

MSpec application is available as a MATLAB based application and a stand alone application. The user can choose the installation steps based on their preferences. If the user already has MATLAB installed, it is recommended to run MSpec as a MATLAB application for additional MATLAB features available. The open source of the application and its standalone can be downloaded in the MSpec Spectral Preprocessing and Analysis Environment github at https://github.com/chayudS/MSpec-Spectral-Preprocessing-and-Analysis-Environment

Prerequisites

To run MSpec as a MATLAB application, the user must install MATLAB and the required toolboxes, as well as the MSpec MATLAB application files.

- Mathworks® MATLAB R2020a or later versions with the following add-on toolboxes
    - Bioinformatics Toolbox version 4.15.1
    - Statics and Machine Learning Toolbox version 12.1

      Available at www.mathworks.com/downloads or execute the application via a web browser by using MATLAB online at https://matlab.mathworks.com/
- MSpec MATLAB Application Available at: https://github.com/chayudS/MSpec-Spectral-Preprocessing-and-Analysis-Environment

To run MSpec as a standalone application, the user must install the MSpec standalone application.

- MSpec Standalone Installer located in folder *"/app-standalone"* in MSpec github or at: https://github.com/chayudS/MSpec-Spectral-Preprocessing-and-Analysis-Environment/tree/main/app-standalone

Installation steps

To execute MSpec as a MATLAB application:

1. Install Mathworks® MATLAB and required add-on toolboxes
2. Import the MSpec application downloaded from MSpec github into the MATLAB workspace
3. Run the MSpec application
    a. To run the MSpec main application, run the file called *"MSpecMainApp.mlapp"*

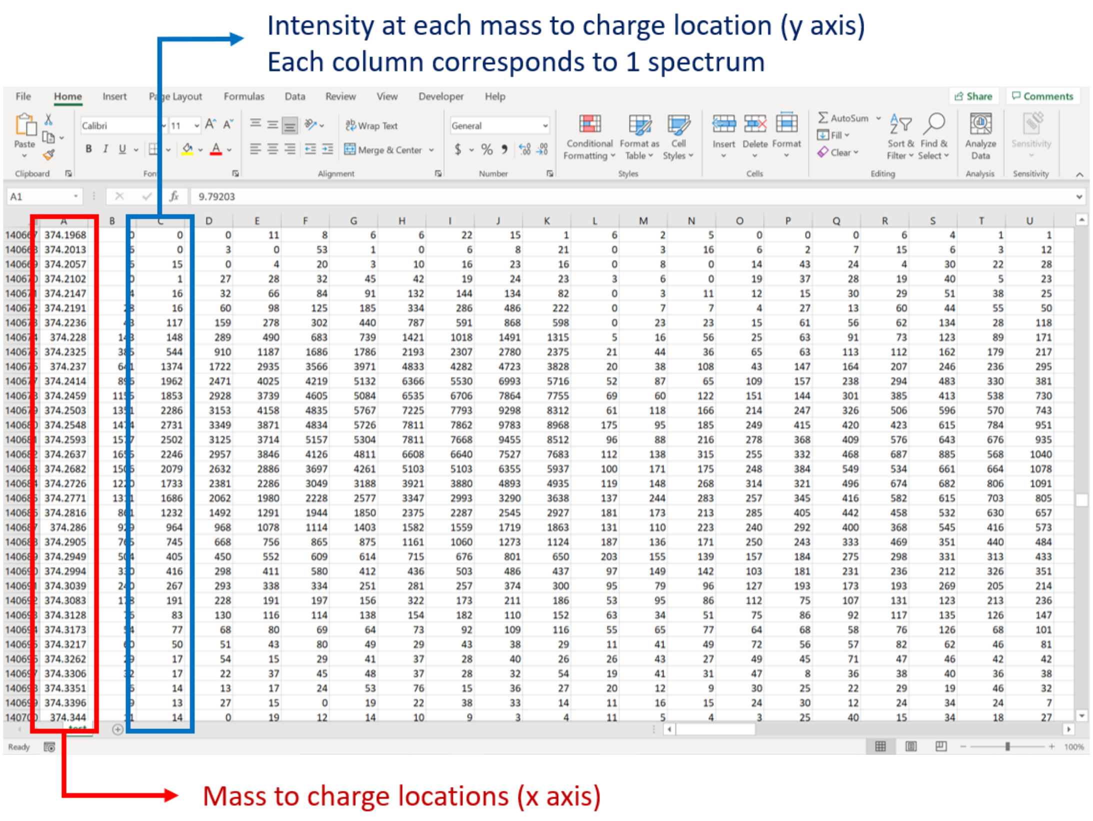
b. If the icons in the application don’t appear, please add source code folder of the MSpec application to be the file path in MATLAB

To install MSpec as a standalone application:

1. Install MSpec standalone application by clicking on *“MSpecMainAppInstaller.exe”*
2. After the installation the file named *“MSpecMainApp.exe”* will appear. MSpec application will be started by double click on the *“MSpecMainApp.exe”*

Raw Data Pre-requirement

The raw data needs to be in the form of a table in a .csv file. Each row of the table should represent the spectra data points i.e. mass/charge in Mass spectra or Wavenumber in Raman spectra. The first column of the table shows each data point on each row and the second to the last column will represent each sample of spectra.


Intensity at each mass to charge location (y axis)
Each column corresponds to 1 spectrum
File Home Insert Page Layout Formulas Data Review View Developer Help
Share Comments
Paste Clipboard Calibri 11 Font Wrap Text Merge & Center Alignment General Number
Conditional Formatting Format as Table Cell Styles Styles Insert Delete Format Cells
AutoSum Fill Clear Sort & Filter Find & Select Editing Analyze Data Analysis Sensitivity
A1 fx 9.79203
test
Ready 100%
Mass to charge locations (x axis)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
140667	374.1968	[illegible]	0	0	11	8	6	6	22	15	1	6	2	5	0	0	0	6	4	1	1
140668	374.2013	[illegible]	0	3	0	53	1	0	6	8	21	0	3	16	6	2	7	15	6	3	12
140669	374.2057	[illegible]	15	0	4	20	3	10	16	23	16	0	8	0	14	43	24	4	30	22	28
140670	374.2102	[illegible]	1	27	28	32	45	42	19	24	23	3	6	0	19	37	28	19	40	5	23
140671	374.2147	[illegible]	16	32	66	84	91	132	144	134	82	0	3	11	12	15	30	29	51	38	25
140672	374.2191	[illegible]	16	60	98	125	185	334	286	486	222	0	7	7	4	27	13	60	44	55	50
140673	374.2236	[illegible]	117	159	278	302	440	787	591	868	598	0	23	23	15	61	56	62	134	28	118
140674	374.228	[illegible]	148	289	490	683	739	1421	1018	1491	1315	5	16	56	25	63	91	73	123	89	171
140675	374.2325	[illegible]	544	910	1187	1686	1786	2193	2307	2780	2375	21	44	36	65	63	113	112	162	179	217
140676	374.237	[illegible]	1374	1722	2935	3566	3971	4833	4282	4723	3828	20	38	108	43	147	164	207	246	236	295
140677	374.2414	[illegible]	1962	2471	4025	4219	5132	6366	5530	6993	5716	52	87	65	109	157	238	294	483	330	381
140678	374.2459	[illegible]	1853	2928	3739	4605	5084	6535	6706	7864	7755	69	60	122	151	144	301	385	413	538	730
140679	374.2503	[illegible]	2286	3153	4158	4835	5767	7225	7793	9298	8312	61	118	166	214	247	326	506	596	570	743
140680	374.2548	[illegible]	2731	3349	3871	4834	5726	7811	7862	9783	8968	175	95	185	249	415	420	423	615	784	951
140681	374.2593	[illegible]	2502	3125	3714	5157	5304	7811	7668	9455	8512	96	88	216	278	368	409	576	643	676	935
140682	374.2637	[illegible]	2246	2957	3846	4126	4811	6608	6640	7527	7683	112	138	315	255	332	468	687	885	568	1040
140683	374.2682	[illegible]	2079	2632	2886	3697	4261	5103	5103	6355	5937	100	171	175	248	384	549	534	661	664	1078
140684	374.2726	[illegible]	1733	2381	2286	3049	3188	3921	3880	4893	4935	119	148	268	314	321	496	674	682	806	1091
140685	374.2771	[illegible]	1686	2062	1980	2228	2577	3347	2993	3290	3638	137	244	283	257	345	416	582	615	703	805
140686	374.2816	[illegible]	1232	1492	1291	1944	1850	2375	2287	2545	2927	181	173	213	285	405	442	458	532	630	657
140687	374.286	[illegible]	964	968	1078	1114	1403	1582	1559	1719	1863	131	110	223	240	292	400	368	545	416	573
140688	374.2905	[illegible]	745	668	756	865	875	1161	1060	1273	1124	187	136	171	250	243	333	469	351	440	484
140689	374.2949	[illegible]	405	450	552	609	614	715	676	801	650	203	155	139	157	184	275	298	331	313	433
140690	374.2994	[illegible]	416	298	411	580	412	436	503	486	437	97	149	142	103	181	231	236	212	326	351
140691	374.3039	[illegible]	267	293	338	334	251	281	257	374	300	95	79	96	127	193	173	193	269	205	214
140692	374.3083	[illegible]	191	228	191	197	156	322	173	211	186	53	95	86	112	75	107	131	123	213	236
140693	374.3128	[illegible]	83	130	116	114	138	154	182	110	152	63	34	51	75	86	92	117	135	126	147
140694	374.3173	[illegible]	77	68	80	69	64	73	92	109	116	55	65	77	64	68	58	76	126	68	101
140695	374.3217	[illegible]	50	51	43	80	49	29	43	38	29	11	41	49	72	56	57	82	62	46	81
140696	374.3262	[illegible]	17	54	15	29	41	37	28	40	26	26	43	27	49	45	71	47	46	42	42
140697	374.3306	[illegible]	17	22	37	45	48	37	28	32	54	19	41	31	47	8	36	38	40	36	38
140698	374.3351	[illegible]	14	13	17	24	53	76	15	36	27	20	12	9	30	25	22	29	19	46	32
140699	374.3396	[illegible]	13	27	15	0	19	22	38	33	14	11	16	15	24	30	12	24	34	24	7
140700	374.344	[illegible]	14	0	19	12	14	10	9	3	4	11	5	4	3	25	40	15	34	18	27

Figure 1 Correct table format of .csv file for MSpec application (Using Mass Spectra as an example)

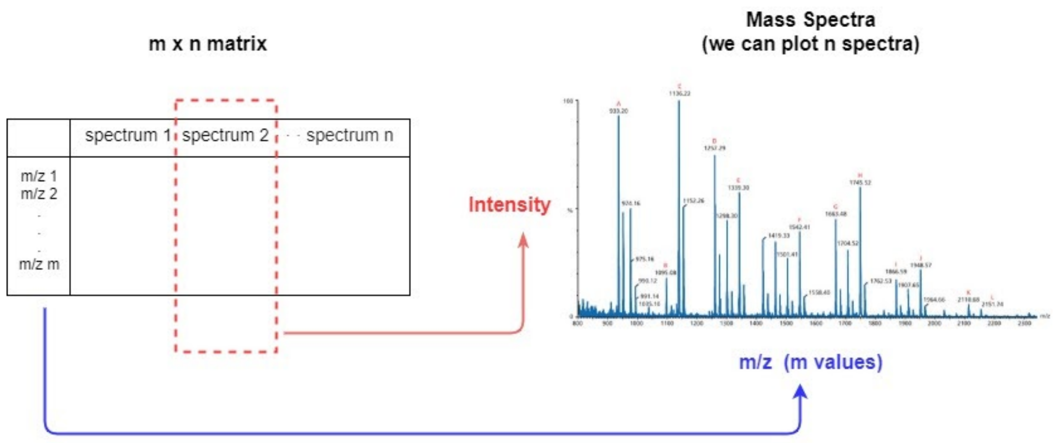
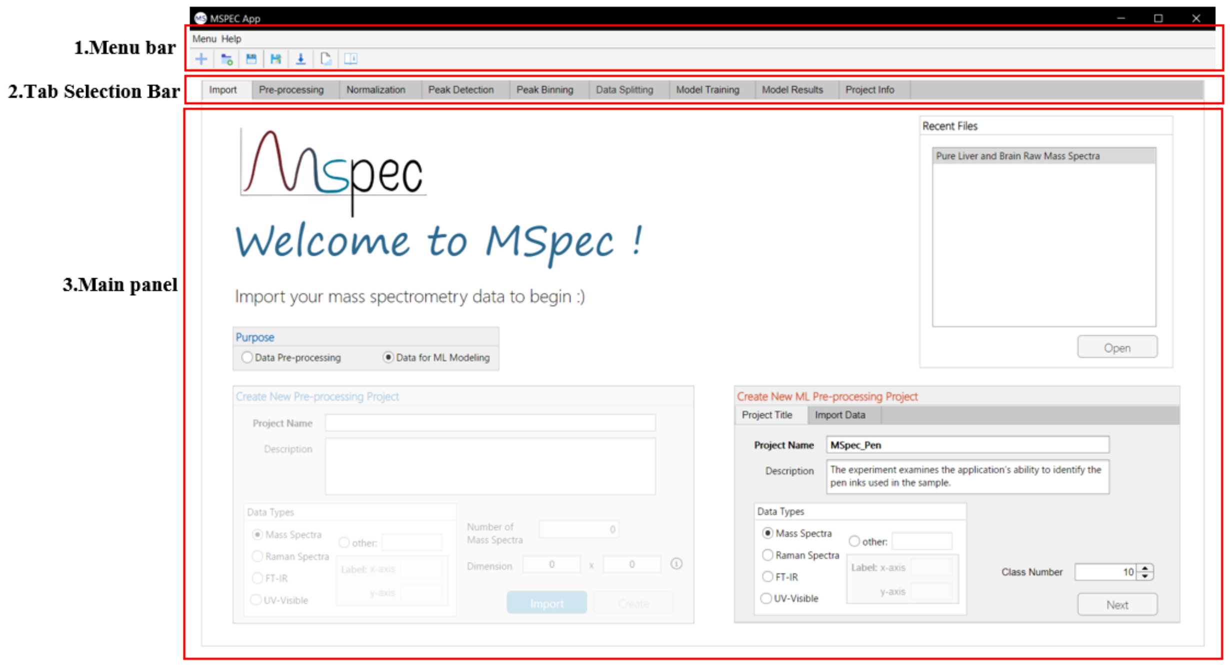


Figure 2 Correct format of .csv file and the graph plotting using the table (Using Mass Spectra as an example)

Application Usage

After finishing the installation either by standalone installation or MATLAB installation, when the MSpec application is started, the first tab of the application that will be displayed is the "Import" tab. The layout of the application consisted of two main parts. The Menu Panel on the top most part of the window and the lower part will be the main panel where the user will operate the imported spectra as well as to select the operation tab on the top of the panel.



Recent Files: Pure Liver and Brain Raw Mass Spectra

Purpose: ○ Data Pre-processing ● Data for ML Modeling

Create New ML Pre-processing Project — Project Title | Import Data

Project Name: MSpec_Pen

Description: The experiment examines the application's ability to identify the pen inks used in the sample.

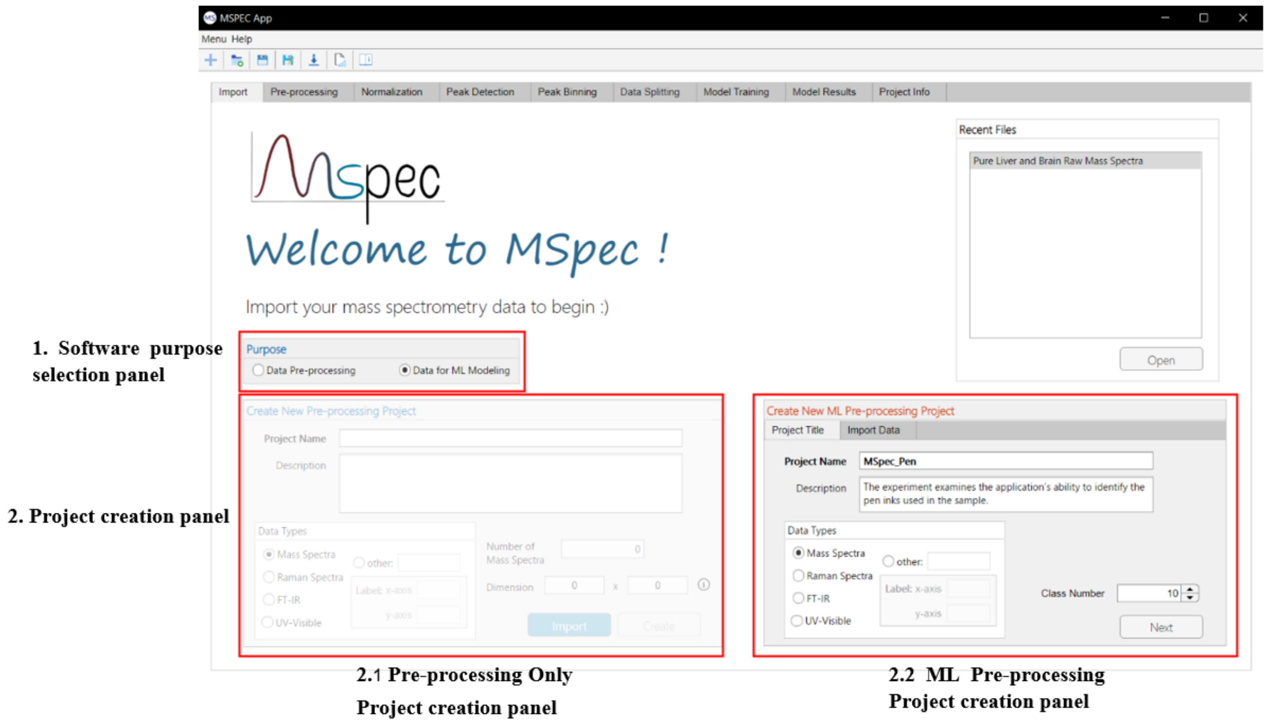
Data Types: ● Mass Spectra ○ Raman Spectra ○ FT-IR ○ UV-Visible ○ other:

Label: x-axis y-axis

Class Number: 10

Figure 3 MSpec application layout overview

After running the MSpecMainApp.mlapp, the user will be taken to the "Import" tab of the software, then the user can select the purpose of the software whether to pre-process the data only or to operate data for Machine Learning. The purpose selection of the software will enable the "Create New Project" sub-panel where the user can import their spectrum data for selected purposes.


MSPEC App
Menu Help
Import
Pre-processing
Normalization
Peak Detection
Peak Binning
Data Splitting
Model Training
Model Results
Project Info
Mspec
Welcome to MSpec !
Import your mass spectrometry data to begin :)
1. Software purpose selection panel
2. Project creation panel
Create New Pre-processing Project
Import
Create
Create New ML Pre-processing Project
Project Title
Import Data
Next
Open
2.1 Pre-processing Only Project creation panel
2.2 ML Pre-processing Project creation panel

Recent Files: Pure Liver and Brain Raw Mass Spectra

Purpose: ○ Data Pre-processing [✓ Data for ML Modeling]

Project Name: MSpec_Pen

Description: The experiment examines the application's ability to identify the pen inks used in the sample.

Data Types: [✓ Mass Spectra] ○ Raman Spectra ○ FT-IR ○ UV-Visible ○ other:

Label: x-axis, y-axis

Number of Mass Spectra: 0

Dimension: 0 x 0

Class Number: 10

Figure 4 "Import" tab layout overview

There is a project creation sub-panel available for each purpose.

- Create New Pre-processing Project
  - Users can import only one Spectra file for the pre-processing only operation; the parameters adjustable tab will be available only for the data pre-processing purpose.

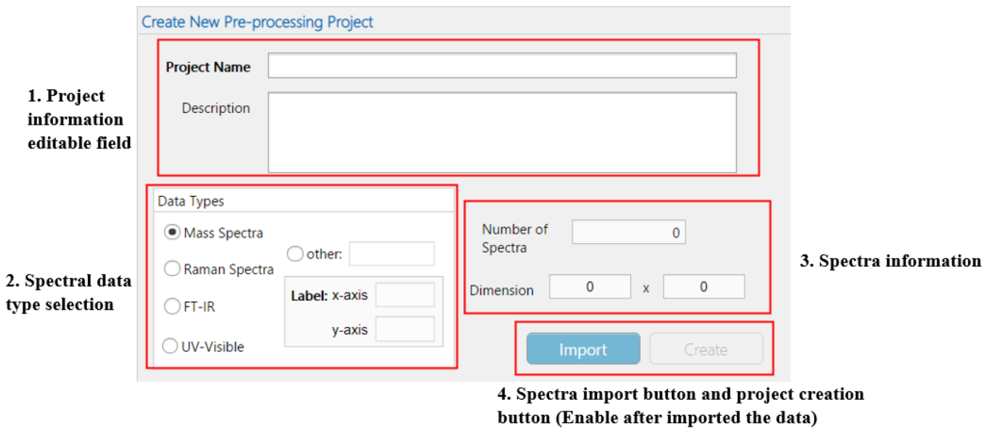

Create New Pre-processing Project
Project Name
Description
1. Project information editable field
Data Types
Mass Spectra
Raman Spectra
FT-IR
UV-Visible
other:
Label: x-axis
y-axis
2. Spectral data type selection
Number of Spectra
0
Dimension
0
x
0
3. Spectra information
Import
Create
4. Spectra import button and project creation button (Enable after imported the data)

Figure 5 Create new pre-processing project panel layout

- Create ML Pre-processing Project: Consisted of the two tabs i.e. "Project Title" and "Import Data" tabs.
    - The first tab that will appear is the "Project Title" tab where the user will record the project information as well as class number of the spectra that will be used in the ML model.

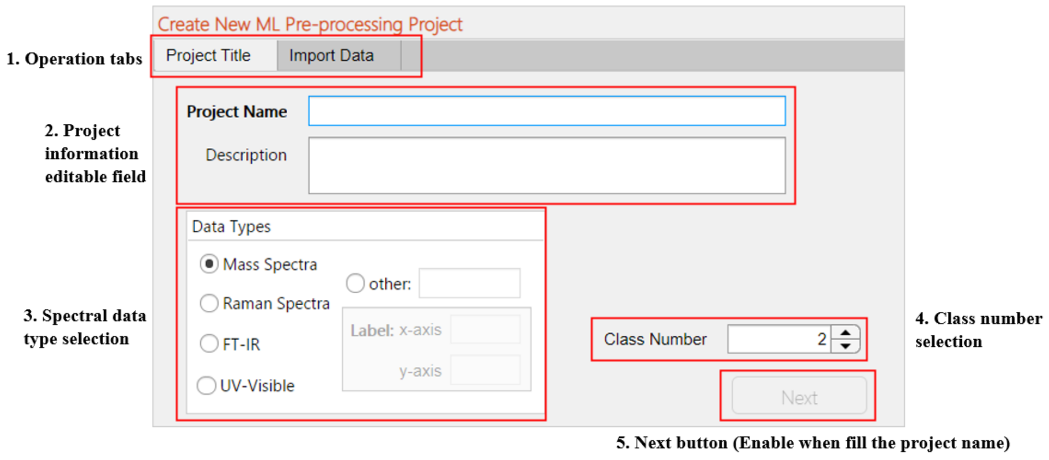
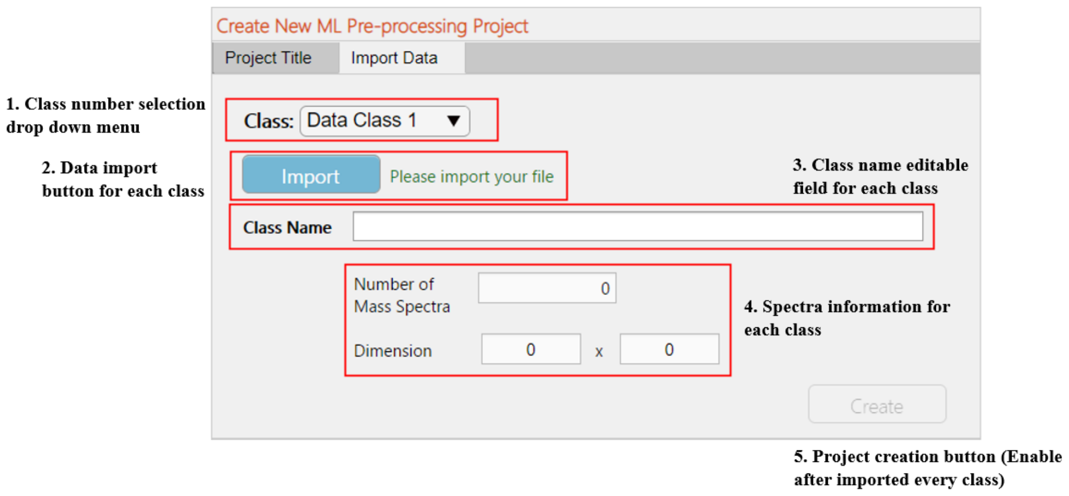

Create New ML Pre-processing Project
Project Title
Import Data
1. Operation tabs
Project Name
Description
2. Project information editable field
Data Types
Mass Spectra
Raman Spectra
FT-IR
UV-Visible
other:
Label: x-axis
y-axis
3. Spectral data type selection
Class Number
2
4. Class number selection
Next
5. Next button (Enable when fill the project name)

Figure 6 "Project Title" sub-tab layout components

-
    - The second tab will appear after clicking the "Next" button on the first tab. The second tab allows the user to import spectral data class by class according to the number of classes being selected on the first tab.


Create New ML Pre-processing Project
Project Title
Import Data
1. Class number selection drop down menu
2. Data import button for each class
3. Class name editable field for each class
4. Spectra information for each class
5. Project creation button (Enable after imported every class)
Import
Create

Class: Data Class 1

Please import your file

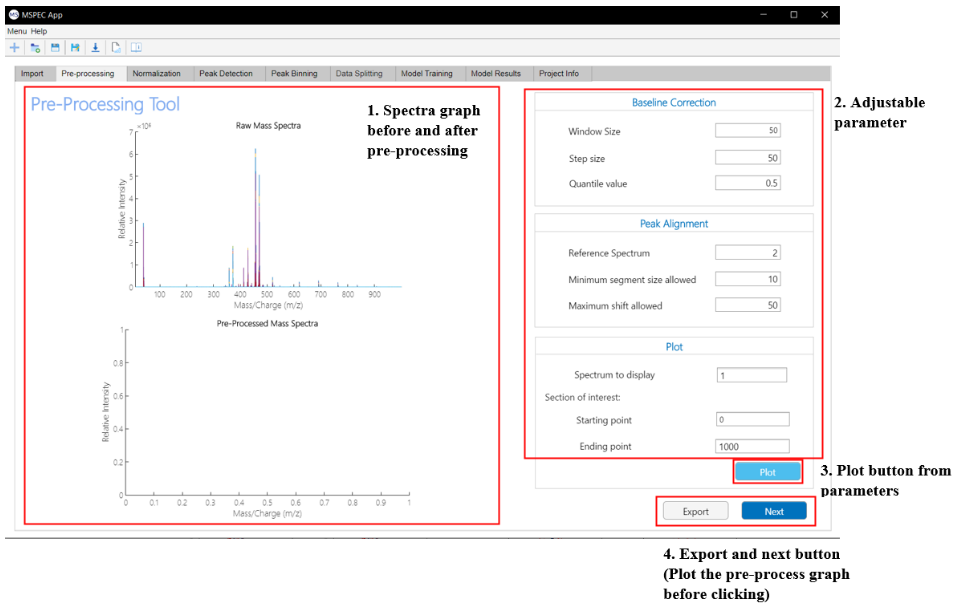
Class Name

Number of Mass Spectra: 0

Dimension: 0 x 0

Figure 7 "Import Data" sub- tab layout components

After successfully importing the dataset, the user can proceed to the "Pre-processing" tab. In this tab, the user can specify the parameters to perform baseline correction and peak alignment. The user can also specify the section of the spectra that they want to display.



Baseline Correction

Window Size	50
Step size	50
Quantile value	0.5

Peak Alignment

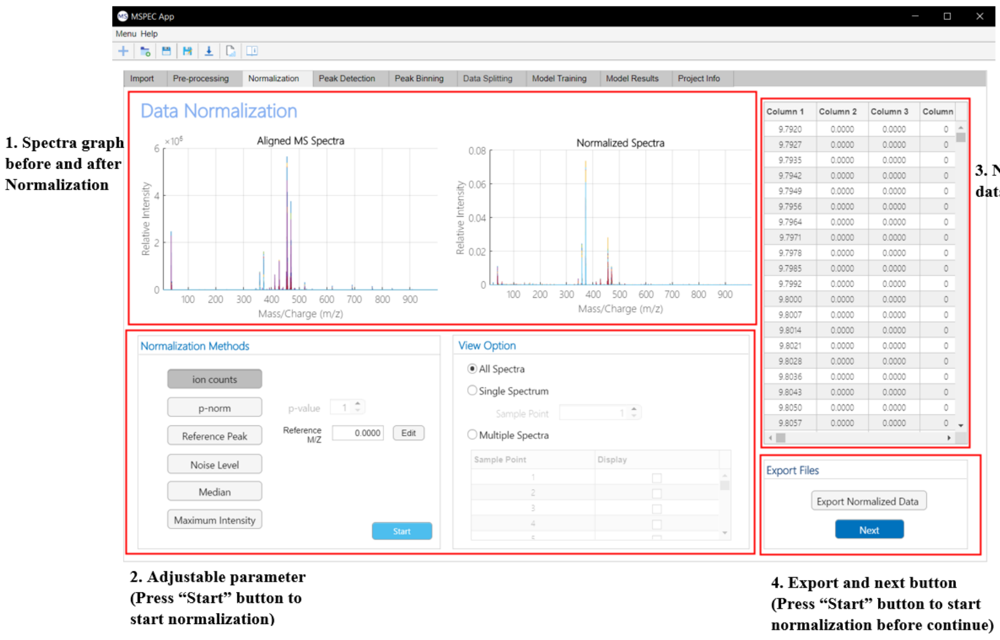
Reference Spectrum	2
Minimum segment size allowed	10
Maximum shift allowed	50

Plot

Spectrum to display	1
Section of interest:	
Starting point	0
Ending point	1000

Figure 8 "Pre-processing" tab layout components

Once completed, the user can continue the preprocessing steps in the "Normalization" tab. The user can choose their normalization method, such as ion counts or reference peak, and view the normalized data.



Column 1	Column 2	Column 3	Column
9.7920	0.0000	0.0000	0
9.7927	0.0000	0.0000	0
9.7935	0.0000	0.0000	0
9.7942	0.0000	0.0000	0
9.7949	0.0000	0.0000	0
9.7956	0.0000	0.0000	0
9.7964	0.0000	0.0000	0
9.7971	0.0000	0.0000	0
9.7978	0.0000	0.0000	0
9.7985	0.0000	0.0000	0
9.7992	0.0000	0.0000	0
9.8000	0.0000	0.0000	0
9.8007	0.0000	0.0000	0
9.8014	0.0000	0.0000	0
9.8021	0.0000	0.0000	0
9.8028	0.0000	0.0000	0
9.8036	0.0000	0.0000	0
9.8043	0.0000	0.0000	0
9.8050	0.0000	0.0000	0
9.8057	0.0000	0.0000	0

Figure 9 "Normalization" tab layout components

After that, the user can detect significant peaks in the "Peak Detection" tab. This can be done automatically or with specific parameters by the user. Those peaks can then be binned in the "Peak Binning" tab.

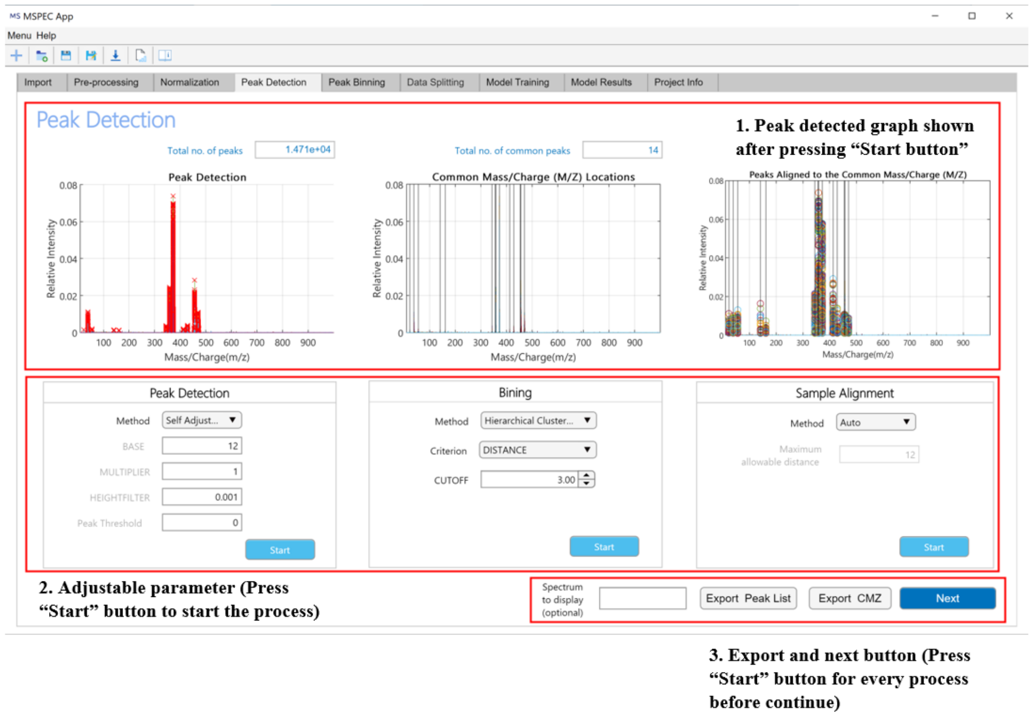
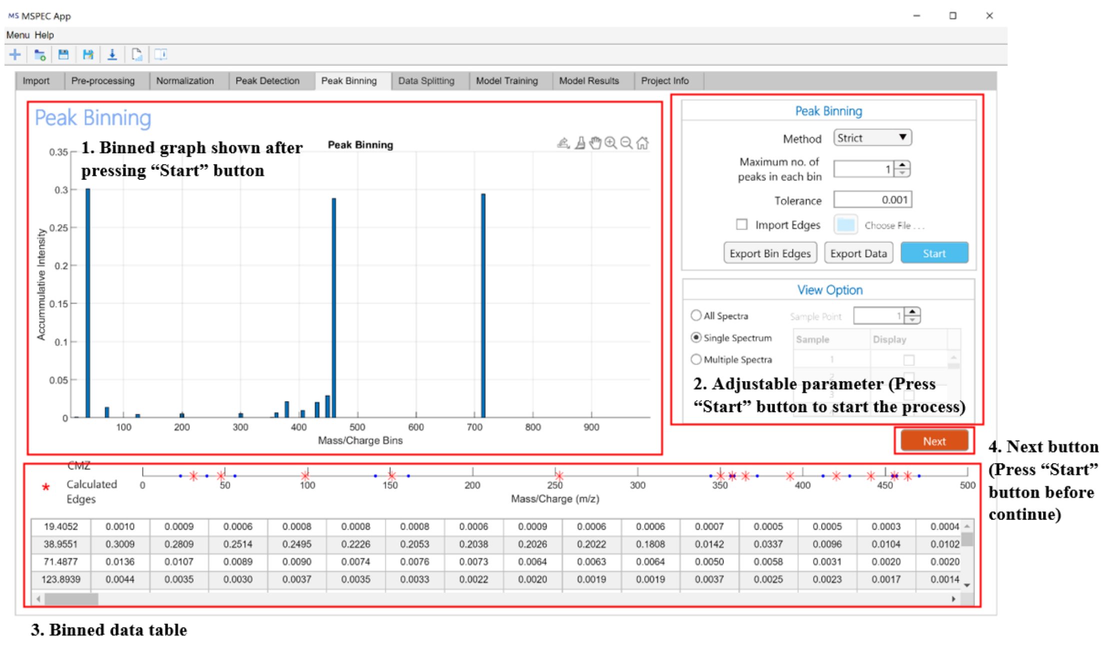


Figure 10 "Peak Detection" tab layout components



1. Binned graph shown after pressing “Start” button

2. Adjustable parameter (Press “Start” button to start the process)

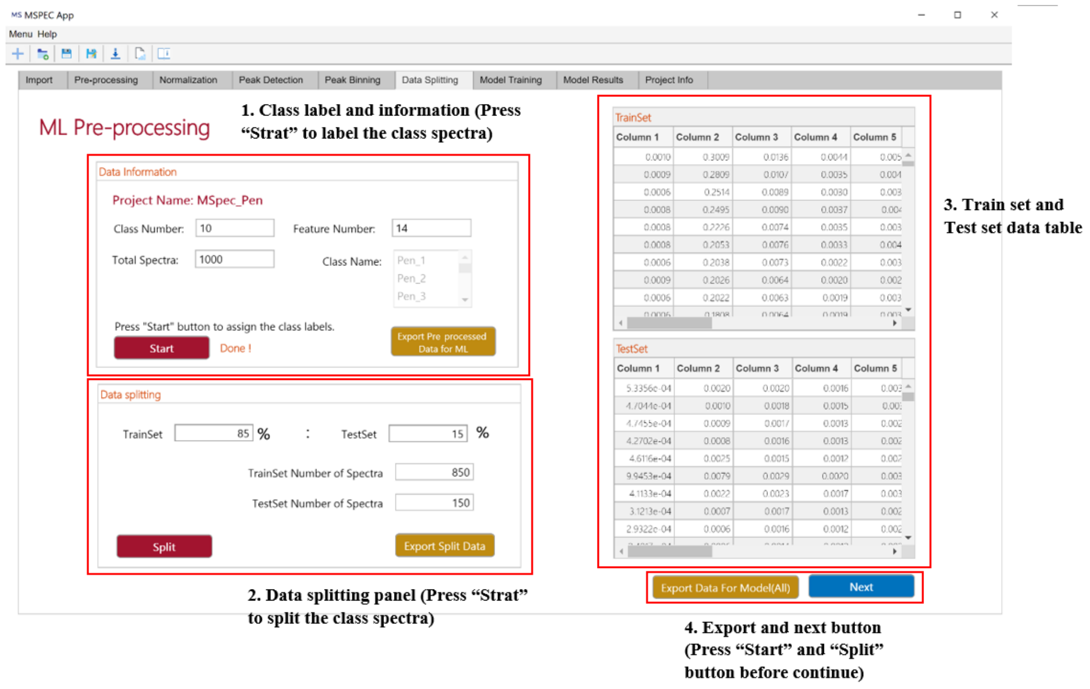
3. Binned data table

4. Next button (Press “Start” button before continue)

19.4052	0.0010	0.0009	0.0006	0.0008	0.0008	0.0008	0.0006	0.0009	0.0006	0.0006	0.0007	0.0005	0.0005	0.0003	0.0004
38.9551	0.3009	0.2809	0.2514	0.2495	0.2226	0.2053	0.2038	0.2026	0.2022	0.1808	0.0142	0.0337	0.0096	0.0104	0.0102
71.4877	0.0136	0.0107	0.0089	0.0090	0.0074	0.0076	0.0073	0.0064	0.0063	0.0064	0.0050	0.0058	0.0031	0.0020	0.0020
123.8939	0.0044	0.0035	0.0030	0.0037	0.0035	0.0033	0.0022	0.0020	0.0019	0.0019	0.0037	0.0025	0.0023	0.0017	0.0014

Figure 11 “Peak Binning” tab layout components

The binned data set is then sent to the “Data Splitting” tab, which allows users to automatically label spectral classes as well as split data into a train set and a test set for further ML model training. The data splitting can be done automatically after users define the percentage of each data set.



1. Class label and information (Press “Strat” to label the class spectra)

Project Name: MSpec_Pen

Class Number: 10

Feature Number: 14

Total Spectra: 1000

Class Name: Pen_1, Pen_2, Pen_3

Press "Start" button to assign the class labels.

TrainSet 85 % : TestSet 15 %

TrainSet Number of Spectra 850

TestSet Number of Spectra 150

2. Data splitting panel (Press “Strat” to split the class spectra)

3. Train set and Test set data table

TrainSet

Column 1	Column 2	Column 3	Column 4	Column 5
0.0010	0.3009	0.0136	0.0044	0.005
0.0009	0.2809	0.0107	0.0035	0.004
0.0006	0.2514	0.0089	0.0030	0.003
0.0008	0.2495	0.0090	0.0037	0.004
0.0008	0.2226	0.0074	0.0035	0.003
0.0008	0.2053	0.0076	0.0033	0.004
0.0006	0.2038	0.0073	0.0022	0.003
0.0009	0.2026	0.0064	0.0020	0.002
0.0006	0.2022	0.0063	0.0019	0.003
0.0006	0.1808	0.0064	0.0019	0.003

TestSet

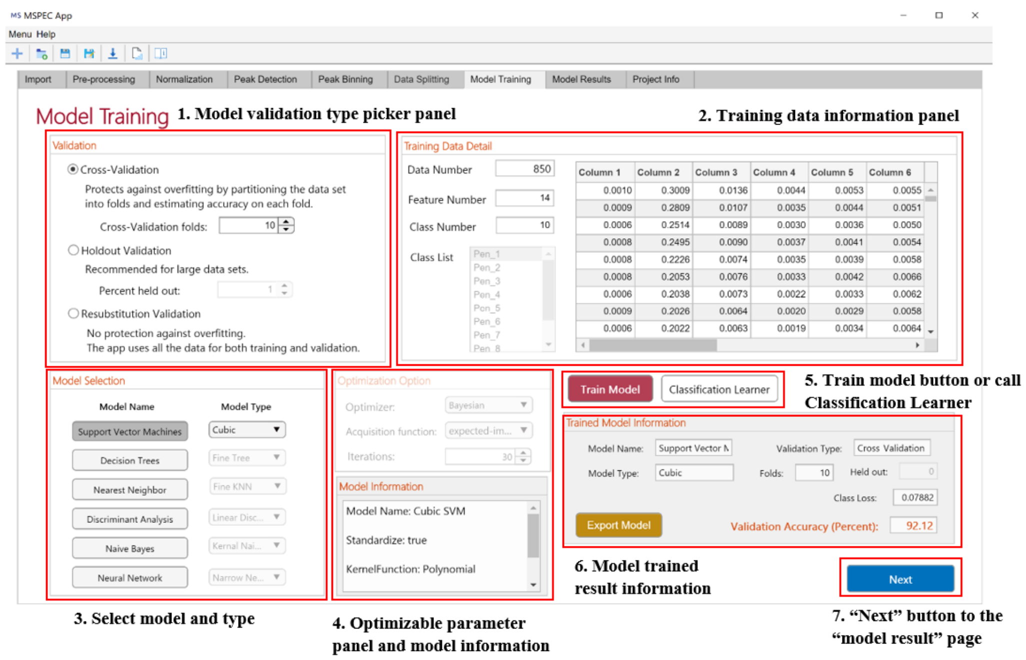
Column 1	Column 2	Column 3	Column 4	Column 5
5.3356e-04	0.0020	0.0020	0.0016	0.003
4.7044e-04	0.0010	0.0018	0.0015	0.003
4.7455e-04	0.0009	0.0017	0.0013	0.002
4.2702e-04	0.0008	0.0016	0.0013	0.002
4.6116e-04	0.0025	0.0015	0.0012	0.002
9.9453e-04	0.0079	0.0029	0.0020	0.003
4.1133e-04	0.0022	0.0023	0.0017	0.003
3.1213e-04	0.0007	0.0017	0.0013	0.002
2.9322e-04	0.0006	0.0016	0.0012	0.002

4. Export and next button (Press “Start” and “Split” button before continue)

Figure 12 “Data Splitting” tab layout components

After pressing the “Next” button from “Data Splitting”, The split data sets are then sent to the “Model Training”. Users are able to train selected models from the split train data and do the validation test to see the validation accuracy of the model being trained. Users can also train the optimizable model using basic ML hyperparameters.

In the case of the MATLAB based MSpec application, users can also use the Classification Learner application in MATLAB to train the model.


MSPEC App
Menu Help
Import
Pre-processing
Normalization
Peak Detection
Peak Binning
Data Splitting
Model Training
Model Results
Project Info
1. Model validation type picker panel
2. Training data information panel
3. Select model and type
4. Optimizable parameter panel and model information
5. Train model button or call Classification Learner
6. Model trained result information
7. “Next” button to the “model result” page
Train Model
Classification Learner
Export Model
Next

Model Training

Validation

(●) Cross-Validation
Protects against overfitting by partitioning the data set into folds and estimating accuracy on each fold.
Cross-Validation folds: 10

( ) Holdout Validation
Recommended for large data sets.
Percent held out: 1

( ) Resubstitution Validation
No protection against overfitting.
The app uses all the data for both training and validation.

Training Data Detail

Data Number: 850
Feature Number: 14
Class Number: 10
Class List: Pen_1, Pen_2, Pen_3, Pen_4, Pen_5, Pen_6, Pen_7, Pen_8

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6
0.0010	0.3009	0.0136	0.0044	0.0053	0.0055
0.0009	0.2809	0.0107	0.0035	0.0044	0.0051
0.0006	0.2514	0.0089	0.0030	0.0036	0.0050
0.0008	0.2495	0.0090	0.0037	0.0041	0.0054
0.0008	0.2226	0.0074	0.0035	0.0039	0.0058
0.0008	0.2053	0.0076	0.0033	0.0042	0.0066
0.0006	0.2038	0.0073	0.0022	0.0033	0.0062
0.0009	0.2026	0.0064	0.0020	0.0029	0.0058
0.0006	0.2022	0.0063	0.0019	0.0034	0.0064

Model Selection

Model Name	Model Type
Support Vector Machines	Cubic
Decision Trees	Fine Tree
Nearest Neighbor	Fine KNN
Discriminant Analysis	Linear Disc...
Naive Bayes	Kernal Nai...
Neural Network	Narrow Ne...

Optimization Option

Optimizer: Bayesian
Acquisition function: expected-im...
Iterations: 30

Model Information

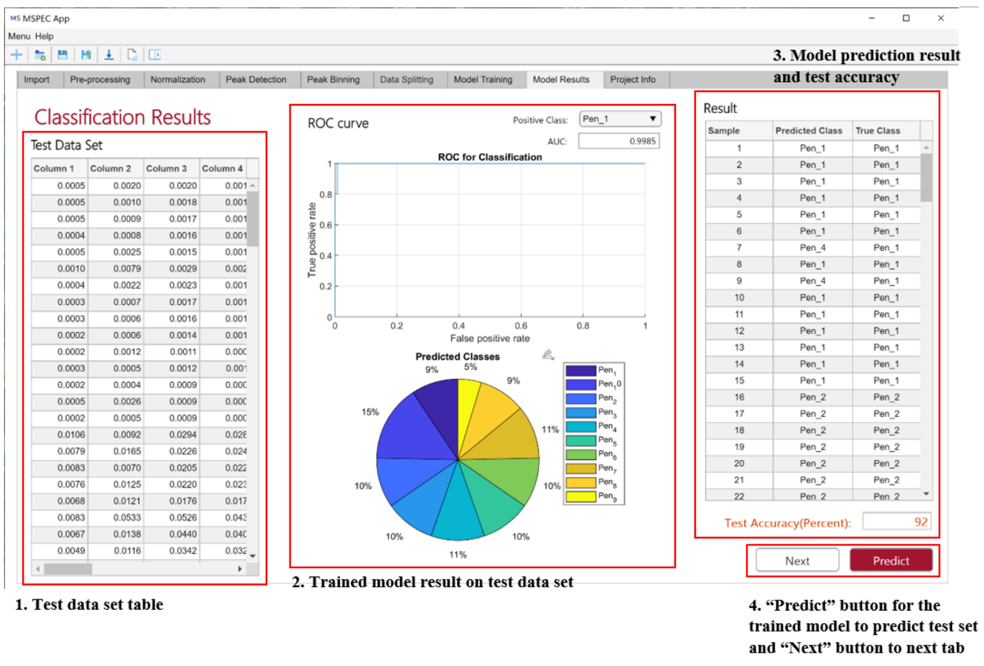
Model Name: Cubic SVM
Standardize: true
KernelFunction: Polynomial

Trained Model Information

Model Name: Support Vector M
Validation Type: Cross Validation
Model Type: Cubic
Folds: 10
Held out: 0
Class Loss: 0.07882
Validation Accuracy (Percent): 92.12

Figure 13 “Model training” tab layout components

After getting the satisfied trained model, the model will be sent to the “Model Result” tab to test the model accuracy and prediction based on the test data set that is being split. The model result will be displayed after pressing the “Predict” button, then as well pop up the confusion matrix for the prediction values from the model.



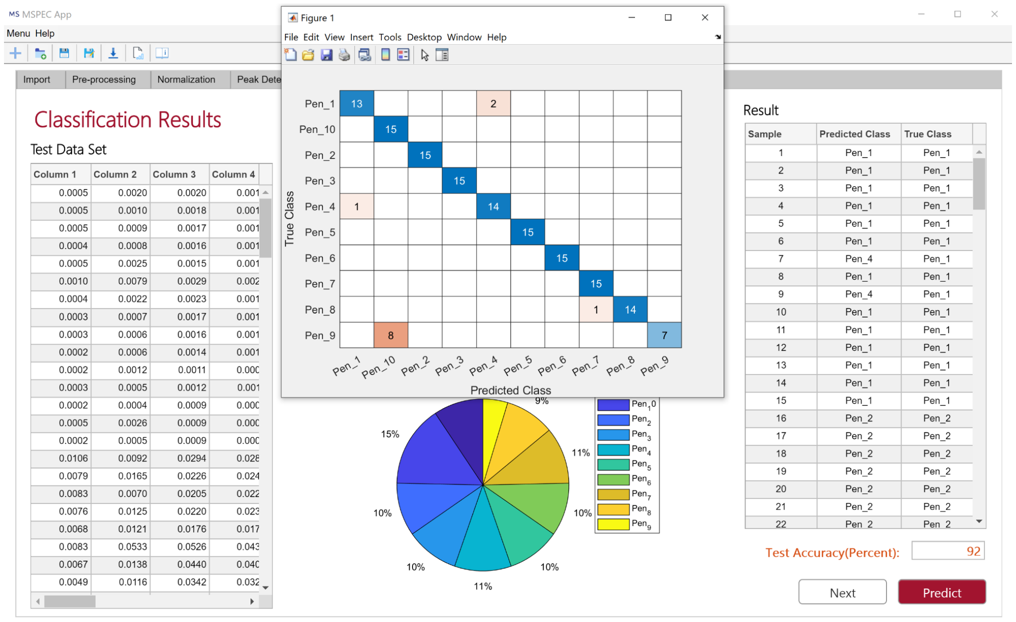
Test Data Set

Column 1	Column 2	Column 3	Column 4
0.0005	0.0020	0.0020	0.001
0.0005	0.0010	0.0018	0.001
0.0005	0.0009	0.0017	0.001
0.0004	0.0008	0.0016	0.001
0.0005	0.0025	0.0015	0.001
0.0010	0.0079	0.0029	0.002
0.0004	0.0022	0.0023	0.001
0.0003	0.0007	0.0017	0.001
0.0003	0.0006	0.0016	0.001
0.0002	0.0006	0.0014	0.001
0.0002	0.0012	0.0011	0.000
0.0003	0.0005	0.0012	0.001
0.0002	0.0004	0.0009	0.000
0.0005	0.0026	0.0009	0.000
0.0002	0.0005	0.0009	0.000
0.0106	0.0092	0.0294	0.028
0.0079	0.0165	0.0226	0.024
0.0083	0.0070	0.0205	0.022
0.0076	0.0125	0.0220	0.023
0.0068	0.0121	0.0176	0.017
0.0083	0.0533	0.0526	0.043
0.0067	0.0138	0.0440	0.040
0.0049	0.0116	0.0342	0.032

Sample	Predicted Class	True Class
1	Pen_1	Pen_1
2	Pen_1	Pen_1
3	Pen_1	Pen_1
4	Pen_1	Pen_1
5	Pen_1	Pen_1
6	Pen_1	Pen_1
7	Pen_4	Pen_1
8	Pen_1	Pen_1
9	Pen_4	Pen_1
10	Pen_1	Pen_1
11	Pen_1	Pen_1
12	Pen_1	Pen_1
13	Pen_1	Pen_1
14	Pen_1	Pen_1
15	Pen_1	Pen_1
16	Pen_2	Pen_2
17	Pen_2	Pen_2
18	Pen_2	Pen_2
19	Pen_2	Pen_2
20	Pen_2	Pen_2
21	Pen_2	Pen_2
22	Pen_2	Pen_2

Test Accuracy(Percent): 92

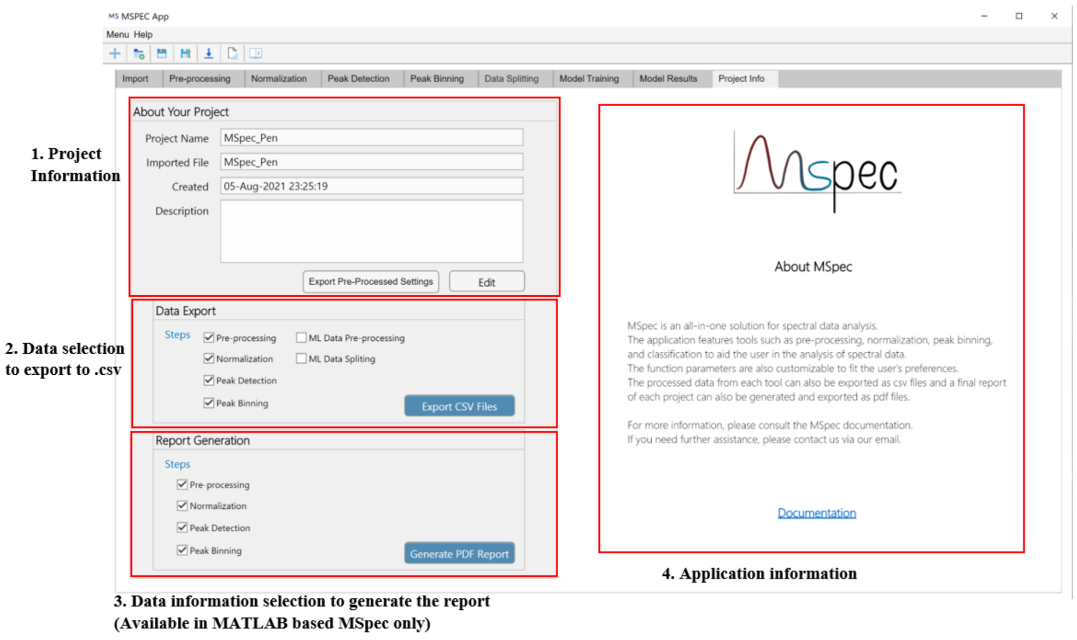
Figure 14 "Model Results" tab layout components



True Class \ Predicted Class	Pen_1	Pen_10	Pen_2	Pen_3	Pen_4	Pen_5	Pen_6	Pen_7	Pen_8	Pen_9
Pen_1	13				2					
Pen_10		15								
Pen_2			15							
Pen_3				15						
Pen_4	1				14					
Pen_5						15				
Pen_6							15			
Pen_7								15		
Pen_8								1	14	
Pen_9		8								7

Figure 15 "Model Results" tab with confusion matrix pop up window

The last part of the application will be shown after finishing the model result test and pressing the next button. The "Project Info" tab which allows users to export data that operated in the previous tabs as well as to generate the report of parameter setting as a .pdf file.



About Your Project

Project Name: MSpec_Pen
Imported File: MSpec_Pen
Created: 05-Aug-2021 23:25:19
Description:

Data Export

Steps
- ☑ Pre-processing
- ☑ Normalization
- ☑ Peak Detection
- ☑ Peak Binning
- ☐ ML Data Pre-processing
- ☐ ML Data Spliting

Report Generation

Steps
- ☑ Pre-processing
- ☑ Normalization
- ☑ Peak Detection
- ☑ Peak Binning

About MSpec

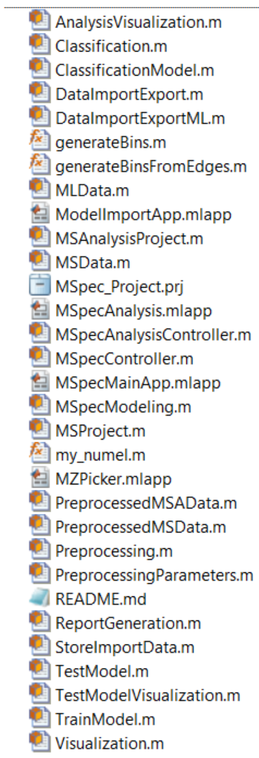
MSpec is an all-in-one solution for spectral data analysis.
The application features tools such as pre-processing, normalization, peak binning, and classification to aid the user in the analysis of spectral data.
The function parameters are also customizable to fit the user's preferences.
The processed data from each tool can also be exported as csv files and a final report of each project can also be generated and exported as pdf files.

For more information, please consult the MSpec documentation.
If you need further assistance, please contact us via our email.

Figure 16 "Project Info" tab layout overview

Implementation Overview

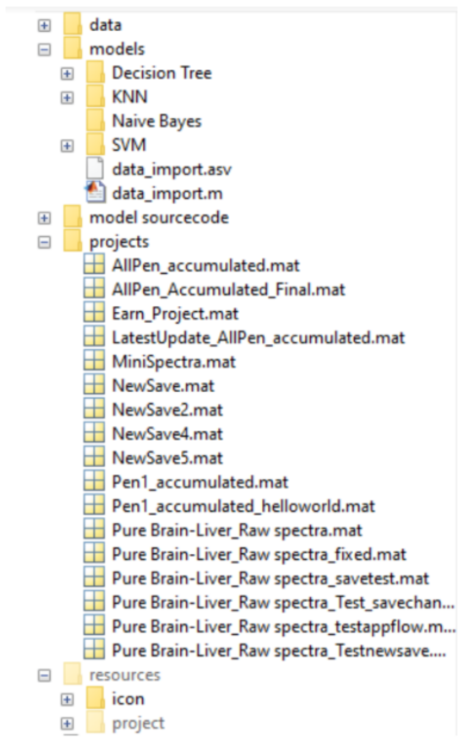
Our application is implemented purely on MATLAB via the MATLAB App Designer, which allows us to create a graphical user interface that can be integrated with our functions.



AnalysisVisualization.m
Classification.m
ClassificationModel.m
DataImportExport.m
DataImportExportML.m
generateBins.m
generateBinsFromEdges.m
MLData.m
ModelImportApp.mlapp
MSAnalysisProject.m
MSData.m
MSpec_Project.prj
MSpecAnalysis.mlapp
MSpecAnalysisController.m
MSpecController.m
MSpecMainApp.mlapp
MSpecModeling.m
MSProject.m
my_numel.m
MZPicker.mlapp
PreprocessedMSAData.m
PreprocessedMSData.m
Preprocessing.m
PreprocessingParameters.m
README.md
ReportGeneration.m
StoreImportData.m
TestModel.m
TestModelVisualization.m
TrainModel.m
Visualization.m

Figure 17 MSpec application files system

- AnalysisVisualization.m: visualization for data analysis function
- Classification.m :model prediction function
- ClassificationModel.m: model and its pre-processing parameters data object
- DataImportExport.m: user input controller class
- DataImportExportML.m: user ML pre-processing input controller class
- generateBins.m: peak binning function
- generateBinsFromEdges.m: peak binning edge finding function
- MLData.m: ML pre-processing data object
- ModelImportApp.mlapp: analysis model import function
- MSAnalysisProject.m: analysis project data object
- MSData.m: project data object
- MSpecAnalysisController.m: main analysis function controller
- MSpecController.m: user interaction controller class
- MSpecMainApp.mlapp: main application user interface
- MSpecModeling.m: control ML model training features
- MSProject.m: user created project data object
- MZPicker.mlapp: peak picking user interface:
- PreprocessedMSData.m: preprocessed project data object:
- Preprocessing.m: preprocessing function
- PreprocessingParameters.m: preprocessing parameter stored
- ReportGeneration.m: report generating function
- StoreImportdata.m: keep the user’s imported data
- TestModel.m: testing trained model features controller
- TestModelVisualization.m: visualize output from the model
- TrainModel.m: model training feature controller
- Visualization.m: visualizing function



```
⊞ data
⊟ models
    ⊞ Decision Tree
    ⊞ KNN
      Naive Bayes
    ⊞ SVM
      data_import.asv
      data_import.m
⊞ model sourcecode
⊟ projects
      AllPen_accumulated.mat
      AllPen_Accumulated_Final.mat
      Earn_Project.mat
      LatestUpdate_AllPen_accumulated.mat
      MiniSpectra.mat
      NewSave.mat
      NewSave2.mat
      NewSave4.mat
      NewSave5.mat
      Pen1_accumulated.mat
      Pen1_accumulated_helloworld.mat
      Pure Brain-Liver_Raw spectra.mat
      Pure Brain-Liver_Raw spectra_fixed.mat
      Pure Brain-Liver_Raw spectra_savetest.mat
      Pure Brain-Liver_Raw spectra_Test_savechan...
      Pure Brain-Liver_Raw spectra_testappflow.m...
      Pure Brain-Liver_Raw spectra_Testnewsave....
⊟ resources
    ⊞ icon
    ⊞ project
```

Figure 18 MSpec application data directory system

- data directory(optional): stores user's data
- models directory: stores classification model imported to the application storage, including an object of class ClassificationModel save as .mat file and .txt file for storing the model information
- model source code: stores matlab functions to generate classification models receiving a train dataset and label as inputs
- projects directory: stores user's saved project from the MSpec application
- resources directory: stores an application project setting as well as other user interface assets