SpectraClassifier 3.1.1. - Help and Manual

SpectraClassifier is a Java solution for designing and implementing MRS-based classifiers. The main goal of SpectraClassifier is to allow users with minimum background knowledge of multivariate statistics to perform a fully automated pattern recognition analysis.

SpectraClassifier incorporates feature selection (greedy stepwise approach, either forward or backward), and feature extraction (PCA). Fisher Linear Discriminant Analysis is the method of choice for classification. Classifier evaluation is performed through various methods: display of the confusion matrix of the training and testing datasets; K-fold cross validation, leave-one-out and bootstrapping as well as ROC curves.

SpectraClassifier is composed of the following modules: Classifier design, Data exploration, Data visualization, Classifier evaluation, Reports, Classifier history, Multi-voxel reports, and Batch analysis. It is able to read low resolution (SV and MV) and high resolution MRS (HRMAS) processed with existing tools (jMRUI, INTERPRET, 3DiCSI or TopSpin). In addition, to facilitate exchanging data between applications, a standard format capable of storing all the information needed for a dataset was developed.

SpectraClassifier is a user-friendly software designed to fulfill the needs of potential users in the MRS community. The scope of SC is specified; and it is concluded that the results obtained with SC compare well with previous non-automated analysis.
Contents
SpectraClassifier 3.1.2. Help and Manual ................................................................. 2
1. INTRODUCTION ................................................................................................. 5
  1.1. MAIN MENU ................................................................................................. 6
      1.1.1. File option: ............................................................................................ 6
      1.1.2. Edit option: ........................................................................................... 8
      1.1.3. View option: .......................................................................................... 8
      1.1.4. Perform option: ................................................................................. 8
      1.1.5. Help option: ......................................................................................... 9
  2. MODULES ........................................................................................................... 10
    2.1. TAB: CLASSIFIER DESIGN ................................................................. 10
        2.1.1. Overview ............................................................................................ 10
        2.1.2. Importing datasets ............................................................................ 10
        2.1.3. Setting datasets as training or testing............................................ 14
        2.1.4. Selecting the spectral range for classification.............................. 15
        2.1.5. Classes’ definition............................................................................. 15
        2.1.6. Feature selection and extraction ...................................................... 16
        2.1.7. Classifier definition......................................................................... 20
    2.2. TAB: Data exploration .............................................................................. 21
        2.2.1. Overview ............................................................................................ 21
        2.2.2. Plot cases .......................................................................................... 21
    2.3. TAB: Data visualization ............................................................................ 25
        2.3.1. Overview ............................................................................................ 25
        2.3.2. PCA visualization ............................................................................. 26
        2.3.3. Using the visualizer .......................................................................... 26
    2.4. TAB: Classifier evaluation .............................................................. 30
        2.4.1. Overview ............................................................................................ 30
        2.4.2. Classification results ......................................................................... 30
        2.4.3. Store classifiers info ......................................................................... 32
        2.4.4. Evaluation method .......................................................................... 32
        2.4.5. ROC curve ....................................................................................... 36
    2.5. TAB: Reports ............................................................................................... 37
        2.5.1. Overview ............................................................................................ 37
        2.5.2. Fisher LDA results ......................................................................... 37
# Table of Contents

2.5.3. Fisher LDA probabilities ................................................................. 38
2.5.4. Determining probabilities ............................................................... 39
2.5.5. Weights matrix .............................................................................. 40
2.5.6. PCA results .................................................................................... 41
2.6. TAB: Classifier history ...................................................................... 42
   2.6.1. Overview .................................................................................... 42
   2.6.2. Classifier info ............................................................................ 43
2.7. TAB: Multi-voxel reports ................................................................. 46
   2.7.1. Overview .................................................................................... 46
   2.7.2. Displaying classification results .................................................... 46
   2.7.3. Plotting colour-coded map ............................................................ 47
   2.7.4. Displaying probabilities ............................................................... 49
   2.7.5. Changing class colours ............................................................... 50
   2.7.6. Exporting information ................................................................. 51
2.8. TAB: Batch analysis ......................................................................... 53
   2.8.1. Overview .................................................................................... 53
   2.8.2. Starting the batch analysis ............................................................ 53
   2.8.3. Feature selection or extraction results .......................................... 54
   2.8.4. Evaluation results ....................................................................... 55
   2.8.5. Performance of the classifiers ...................................................... 56
REFERENCES .............................................................................................. 58
1. INTRODUCTION

Currently available methods for classifying magnetic resonance spectroscopy (MRS) data rely on either commercial (SPSS, SAS), non-commercial (R) or “in-house” programs running over MATLAB. Multiplatform, easy to use software programs for fast and robust classification of MRS data are scarce.

SpectraClassifier is a java software solution that uses machine learning techniques to design and implement classifiers based on in-vivo single voxel (SV) and multi voxel (MV) 1H-MRS and high resolution magic angle spinning (HRMAS) data. SpectraClassifier has been developed in java, using well-known multiplatform libraries to carry out specific tasks, such as i) Weka [1], used for selecting and extracting features; ii) JavaStat [2], used with modifications, to apply Fisher’s Linear Discriminant Analysis (LDA), iii) StatGraphics [3], to generate graphs used in classifier results evaluation, iv) KING (Kinemage, Next Generation) [4], for three-dimensional visualization.

At the moment, SpectraClassifier implements Fisher LDA as the method to separate two, three or four classes, depending on the users’ needs. SpectraClassifier is composed by the following modules (tabs): classifier design, data exploration, data visualization, classifier evaluation, reports, classifier history, multi-voxel reports, and batch analysis.

Classifier design tunes the desired inputs for designing the classifier, such as the training datasets, the definition of tumour classes and the selection of relevant features. Three methods have been implemented for selecting or extracting relevant features - Sequential Forward, Sequential Backward and Principal Components Analysis (PCA). The resulting features or combination of features are used as classifier inputs.

Data exploration allows displaying spectral data, mean and standard deviation of a class for analyzing spectra type population structure and visually comparing spectra.

Data visualization is an up to 3D-latent space visualizer of PCA or LDA results.

Classifier evaluation. An essential part of the life cycle of the classifier development is its validation. SpectraClassifier contains several methods for performing this: a) Confusion matrix, b) ROC (Receiver Operating Characteristic) curve, c) Graphs showing the statistics of well predicted cases, using test sets, and d) Evaluation methods like cross-validation and bootstrapping with mean and standard deviation results of correctly classified cases, in general and by group.

Reports. The application also allows generating reports with the results obtained.

Classifier history. Used for storing the main description of the classifiers chosen by the user. It can be used to compare between classifiers, checking variations of obtained depending on different parameters.

Multi-voxel reports. Used for displaying classification results of a given dataset as a color-coded map.
Batch analysis for creating multiple classifiers varying the number of feature selected or extracted.

1.1. MAIN MENU

Main menu options - The main menu contains the main options of the application. These options are divided into the following groups: File, Edit, View, Perform and Help. Next, we describe these groups in more detail.

1.1.1. File option:

The File option enables the user to create a new classifier, import and export data, and exit the application. There are two ways of importing data - if you want to work with a spectrum per case you must select the option Import datasets, and if you want to make a combination of two spectra per case the option Import two datasets and merge them for analysis should be used. The Export datasets option will export all cases in the list, which you select, to a single file. The application will ask which list you want to export (Imported datasets, Training datasets, Testing datasets).
1.1.1.2. Export datasets

The exported file follows the XML schema developed for storing all the information needed for a dataset. See the following figure.

Format of the exported files:

As shown in the figure, the global node is DATASET, with attributes CreatedBy, Version and Date. The CreatedBy and Version attributes express which application built this file and with what version, in this case the value for created by is always SpectraClassifier. The Date is just the date the file was exported. Version and Date are non-mandatory.

Every data set node will have one or more Case nodes. A Case node has an ID attribute for the identification of the case, and a sequence of nodes like Tissue and Spectrum. A case has only one Tissue node and one or more Spectrum nodes. The Tissue node has a Type attribute (non-mandatory). Every Spectrum node has two mandatory child nodes: Parameters and Points; and one non-mandatory node: MapPosition. The Parameters node has three non-mandatory attributes like PointsNumber for the number of points of the spectrum, LastPPM for the last PPM and FirstPPM for the first PPM. The Points node is used to store the spectrum quantitative data, i.e. the intensity value of each point in the frequency domain, and the MapPosition node is used to store the x-y position of each spectrum in each MV grid. Dashed lines are used to indicate non-mandatory elements.
1.1.2. Edit option:

The *Preferences* option from Edit allows the configuration of certain aspects of the interface, like the appearance (also called *look and feel*) and the position of the tabs. When the Preferences option is clicked, the application displays a window that allows the modification of the already mentioned configuration. See the following figure.

![Preferences window](image)

The Save button is used to accept changes made and the Close button to leave this window.

1.1.3. View option:

The *View* option allows the user to switch between tabs.

1.1.4. Perform option:
The Perform option groups the principal operations of the application, like Run feature selection or extraction, Run classifier, Run evaluation, and Append current classifier info to Classifier history.

1.1.5. Help option:

![Help option image]

The Help option allows the user to open the Help contents, and the About SpectraClassifier to see info related to the version and release date of the application, etcetera. See the following image.

![About SpectraClassifier image]
2. MODULES

2.1. TAB: CLASSIFIER DESIGN

2.1.1. Overview

When SpectraClassifier is launched the application starts by displaying the Classifier design tab – this tab allows you to introduce the needed information to make a classifier.

![Classifier design tab](image)

2.1.2. Importing datasets

The first thing to do is to import datasets. There are two ways to import data, as desired. If you want to work with one spectrum per case you must select the option Import datasets (File/Import datasets), if you want to make a combination of two spectra by case the option Import two datasets and merge them for analysis should be used (File/Import two datasets and merge them for analysis).
2.1.2.1. **Importing one spectrum per case**

There are some validations for importing datasets:

1) If you import a file that does not contain the tumour type information, the application is going to ask for it, (see the following figure).

![Image of the interface asking for tumour type](image)

Note: all cases having the tumour type information (either because it was read from the file, or because the user entered it) may be used for training or testing, those that are left empty because the type of tumor is unknown, may only be used as test cases (but, no numerical evaluation will be performed on them, i.e. assigned to a class in the confusion matrix, only visual inspection will be possible).

2) If you import a file that does not contain the information of the spectral range in ppm, the application is going to ask for it, (see the following figure):

![Image of the interface asking for ppm range](image)

3) Once you import a case, the application is going to use the ppm range entered as the reference for the following files to import. So, if you enter a new case with a range of ppm that does not correspond with the one previously entered, the application will inform you of that, and will not allow you to enter this case. If the new case does not contain the range information, the application is going to ask if you want to use the range entered previously (the reference one). If you say yes, the case will be entered with this range, if you say no, you will not be allowed of enter this case in the group of cases being created.

4) Another validation is related with the number of points of the spectra. All cases should have the same number of points, and the number of points of the first case entered is going to be used as the reference for the following files to import.
2.1.2.2. Importing two spectra per case

If you select to import two datasets and merge them for analysis, the application is going to show a window where you can enter the two datasets (see the following image). Dataset 1 and Dataset 2 buttons can be used to select the files you want to import. When you press the OK button, the application will concatenate every spectrum of the first dataset with the corresponding spectrum of the second one. The allowed file extensions of both datasets do not need to be the same, but the number of cases in both has to be the same.

To import two sets, some validations must be met in addition to the validations for each dataset separately (mentioned below in Importing a spectrum per case):

1) The number of cases has to be the same in both datasets.

2) The number of points of every spectrum has to be the same in both datasets.

3) The name of the case, the tumour type, and the spectral range in ppm, are always taken from Dataset 1 (even if Dataset 1 does not have this information and Dataset 2 does). So, if you enter as Dataset 1 a file that does not contain this information, the application is going to ask for it (as in importing a spectrum per case).

If you enter a dataset that contains two spectra per case as Dataset 1, the application is going to ask you if you want to use both spectra merged. If you say Yes, then the application is going to ask you if you want to use them in the same order that they appear (it means that the first spectra is for Dataset 1 and the second for Dataset 2). If you say No to the option of using both spectra merged, then the application understands that you want to select only one spectrum per case, and then it is going to ask you which of those spectrum do you want to use (if the first or the second one).

2.1.2.3. Allowed extensions

For importing dataset files, the preferable format is the XML with the structure described before. It can be used for the three types of MRS data allowed by SpectraClassifier. Other formats can also be used to import dataset files, according to the type of MRS data:
In-vivo SV data, usually with a low number of points per spectrum (512-2048):

- Files with .txt or .art extension in the INTERPRET [5] canonical format, with 512 points in the [7.2; -2.8] ppm range, which only contain the information of one spectrum in one row. Similarly, files with .dat extension, exported with SPSS or similar, and composed by rows of 514 tokens, where the first row is columns labels (not used in SpectraClassifier), and the rest of rows correspond to cases (similar to the INTERPRET canonical format), having the following information each: identifier of the class, identifier of the case, and 512 points of the spectrum.

- Files with .txt extension, processed and exported using the Magnetic Resonance User Interface package (jMRUI)[6]. It is composed by a header and a four-column matrix of data. The header is partially used by SpectraClassifier, because it contains the number of points of the spectrum (PointsInDataset), and the information that allows inferring the spectral range (SamplingInterval and TransmitterFrequency). From the data matrix, only the third column (fft(real)) is read by SpectraClassifier.

- Files with extension .txt, only with the spectra matrix, in which each row corresponds to a different case. The number of points is not fixed, and the tumour type and ppm range can be entered by the user later. This option is useful for users who want to make a classifier without prior feature selection or extraction.

In-vivo MV data, also with a low number of points per spectrum (512-2048), but with a large number of spectra per acquisition (n x m). SpectraClassifier treats each acquisition as one dataset:

- Files with .bsp extension, that correspond to data pre-processed with 3D Interactive Chemical Shift Imaging v1.9.10 (3DiCSI)[7], and exporting the data in ASCII format [8]. It has the following structure: first row for the name of the set (not used in SpectraClassifier), line-break, Number of voxels: (a number), line-break, Number of points per voxel: (a number), line-break, Voxel Index: (with the information of the location of each voxel in a map, it is not used by SpectraClassifier for the development of the classifier, but for the colour-coded map, see TAB: Multi-voxel reports), line-break, and then two columns with Real and Imaginary data.

- File with extension .txt, processed and exported using the Dynamic MRSI Processing Module (DMPM)[7]. It has the following structure: first row for the name of the set (not used in SpectraClassifier), line-break, Number of voxels: (a number), line-break, Number of points per voxel: (a number), line-break, Voxel Index: (with the information of the location of each voxel in a map, it is not used by SpectraClassifier for the development of the classifier, but for the colour-coded map, see TAB: Multi-voxel reports), line-break, Is labelled: (Yes/No), line-break, and then the matrix of frequencies having one spectrum per row. If Is labelled is equal to Yes, at the end of each row, the class label can be found for the corresponding spectrum. If Is labelled is equal to No, this file cannot be used for training.
High resolution data, usually with a large number of points per spectrum (16-32 K points):

- File with extension .txt, for HRMAS. The original file having been processed with TopSpin [8] or similar and exported as text file. The number of points accepted is variable; the most commonly used are from 1600 to 3200, with a [4.5; 0.5] ppm range. Each file only contains the information of one spectrum in one column.

The pre-processing tasks needed for imported files are out of the scope of SpectraClassifier, so they have to be carried out before using this software, including adjustments of sweep width and number of points if spectra from different manufacturers are to be used. On the other hand, all imported datasets, training and testing sets, regardless of the original format, can be exported in the XML file format described before.

2.1.3. Setting datasets as training or testing

Imported datasets can be used as training or testing sets. After an imported file is chosen (see figure below) you can pick between Training file button and Testing file button to indicate that all cases in this file are going to be for training or testing, respectively. The Reset button is to set all inputs as default and restart the design of the classifier.

Note that the names of the imported data sets are formed by the type of the data set followed by the name of the file. The following list shows all types of the data set used.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSS</td>
<td>Text file from the INTERPRET decision support system</td>
</tr>
<tr>
<td>jMRUI</td>
<td>Text file from jMRUI</td>
</tr>
<tr>
<td>HRMAS</td>
<td>HRMAS text file</td>
</tr>
<tr>
<td>DMPM</td>
<td>Text file from Dynamic MRSI Processing Module (DMPM)</td>
</tr>
</tbody>
</table>
2.1.4. Selecting the spectral range for classification

Once the data is imported, the user can change the **Spectral range for classification (in ppm)**, setting the minimum and maximum value.

The **minimum** and **maximum** values of the spectral range for classification are 0.5 and 4.1 by default because this is the region of interest where the resonances of the main metabolites arise and where the contribution of the residual water is expected to be minimal [9]. But you can change these values by those you consider most appropriate to build the classifier. The text field will accept up to 5 characters; the characters may be integers (numbers) and a decimal point ("."). However, the system will not allow the input of invalid values and will reset the range to default values unless the Disable limits checkbox is activated.

Examples: 

- [.5 .9] is a valid range.
- [0.1234 6.2345] is a valid range.
- [1.2 5.4.1] is not a valid range, unless the Disable limits checkbox is activated. In this case both values will be ignored and the default values will be used.

2.1.5. Classes’ definition

The figure below shows the selected field for choosing the number of classes, you can select among two, three or four classes. After setting imported files as training dataset, the **Tumour types** list box is filled in with all the types read from training cases, so you can select them to create classes or groups. To do that you should mark the tumour types and press the button of the class you need, the tumour type is going to be removed from the Tumour types list box and is going to be added in the list box of the selected class. The **Class name** text is optional; the **Reset** button clears the assigned classes.
Note that the tumour type (class) given name is followed by a number in parenthesis. This is the number of cases (samples) in the training dataset that are labelled with this tumour type (class). For example, tumour type (class) “ra” has 18 cases (samples) and class “pi” has 3 cases (samples).

2.1.6. Feature selection and extraction

Once the training data set is imported, and the classes of tumour types are indicated, you can do the feature selection or the feature extraction, according to the method of your choice. The following figure shows an overview of the selectable fields. Note that in the Results box the selected/extracted features based on the training set are shown.
If the **Method** is "Sequential Forward F.S." or "Sequential Backward F.S.", you will be able to set the **Criterion** and the **Number of features**. Both methods are for feature selection. If the **Method** is PCA (see the following figure), you will be able to set the **Number of features**, the **Maximum ppm combination** and the **Variance covered**.

The **Number of features** represents the desirable number of resulting features after running the **Run Feature Selection or Extraction** button. This text field is limited to an input of maximum 3 numbers; other characters cannot be typed in. Another feature of this field is that it autocompletes according to the number of cases in the selected classes. The autocomplete is based on the following formula:

$$F = \frac{(N_1 + N_2 + N_3 + N_4 - 2 \times C)}{C}$$

Where $F$ is the number of features, $N_1 \ldots N_4$ represent the number of cases in classes 1 through 4 and $C$ is the number of classes actually used. The formula ensures that the number of features chosen satisfies Fisher’s criterion [10, 11]. $F$ will be the maximum number of features that the system allows. A higher number of features may be used if the **Disable limits** box is checked.

If this number of features is set automatically, then the number of features will be set to one third\(^1\) of the minimum number of cases in the smallest class. Note that the “**Set automatically**” checkbox will not be accessible if any of the class has less than 3 cases.

---

\(^1\) See section “Some validations for obtaining relevant features” for details
Example: In a 2 class problem one class comprises of 10 cases and the other of 12. The number of features field will autocomplete to $F = (10 + 12 - 2^2)/2 = 9$. If the Set automatically box is activated, then $F$ will be $10/3 = 3$.

The **Maximum ppm combination** is used for the PCA to specify the maximum number of attributes to include in transformed attribute names. You can check the **Set all possible combination** radio button to include all.

The **Variance covered** box tells the system to retain enough principal components to account for this proportion of variance in the original data. The default value has been set to 0.95.

The **Save points** and **Save results** buttons (see previous figure) allows to save the results of the feature selection or extraction. **Save points** button is for saving the results of the feature selection or extraction in points, and **Save results** button is for saving the actual displayed results in XML format. To later reuse the obtained results, you have to have them saved in XML format. The **Enter saved results** button can be used to load previously saved results (see previous figure).

The following figure shows the results of applying a PCA that retained 10 principal components, with 2 combinations of ppm, and retained 95% of variance.

Inside the process of feature selection of "Sequential Forward F.S." or "Sequential Backward F.S." methods, there is an evaluation of these features by means of the **Correlation-based Feature Subset Selection for Machine Learning** technique implemented in the class **CfsSubsetEval** of Weka. This class evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them. Subsets of features that are highly correlated with the class while having low intercorrelation are preferred.

The PCA method performs a principal components analysis and transformation of the data, used in conjunction with a **Ranker** search. Dimensionality reduction is accomplished by choosing enough eigenvectors to account for some percentage of the variance in the original data [default 0.95 (95%)]. Attribute noise can be filtered by transforming to the PC space, eliminating some of the worst eigenvectors, and then transforming back to the original space. The **Ranker** ranks attributes by their individual evaluations.
2.1.6.1. Some validations for obtaining relevant features

1) As in Tate et al.[5], if the user selects a number of features greater than \( n/3 \), where \( n \) is the number of cases in the smallest group of the training set, the application is going to show a warning for possible data overfitting (see figure below). If the \( n/3 \) value is less than the number of classes - 1, the suggested value will be the number of classes - 1.

2) Fields like Number of features, Maximum ppm combinations and Variance covered only accept the input of numeric values.
3) The *Number of features* should be less than the number of points of each spectrum in the dataset.

### 2.1.7. Classifier definition

The **Classifier** panel (see figure) of the **Classifier design** tab allows to select the **Classification Method**. The available classification method at the moment is Fisher LDA, which calculates a set of linear discriminants and obtains the predictions for the data.

![Classifier panel](image)

After setting all information (data sets, classes and features) you can press the **Run classifier** button to calculate the classifier.

#### 2.1.7.1. Fisher LDA

At the moment, SC uses Fisher LDA as the technique of choice for distinguishing cases between two, three or four classes. Each class could be a tumour type, various tumour types forming a super-class, normal tissue, etc.; this is left up to the user’s choice. Fisher LDA is a fundamental and widely used technique, that provides a reasonably way of reducing the dimensionality of the problem.

For the implementation of the Fisher LDA classification method, the *JavaStat* library was used. *JavaStat*, is an open-source, platform-neutral Java library for performing basic statistics. The Discriminant Analysis class implemented in JavaStat mainly follows the formulae in [12] (see Chapter 11.3 and 11.6). For its used inside *SpectraClassifier*, a few modifications have been made in order to allow the use of more than 4 features. Below you can see this modification:

Line 586 from "multivariate/DiscriminantAnalysis.java", was rewritten as:

```java
for (int m=1; m < groupIndex.length; m++)
```

instead of:

```java
for (int m=1; m < (eigenValues.length-1); m++)
```

---

20
2.2. TAB: Data exploration

2.2.1. Overview

Data exploration is the second tab of the application. You can use it to plot cases, to plot the mean and the standard deviation of cases from the same intrinsic class, and to display the selected features on Classifier design tab (if the used method is "Sequential Forward F.S." or "Sequential Backward F.S.").

2.2.2. Plot cases

To plot cases, mean and standard deviation of data sets, first you have to import a data set. All imported data sets files will be listed in the Imported files list box (see Imported files and Cases figure). If you select one of the listed files, all the cases contained in that file (tumour type and identifier) will be listed in the Cases list box.
You can sort the cases of the Cases list box by tumour types or identifiers. The None option leaves data unsorted, just in the order of appearance in the imported file. For example, the figure above shows the cases of the dataset DSS-longMRS.txt ordered by tumour types.

When you press the Plot button, you will have to indicate in which visualizers you want to plot the selected cases. You can select one or more visualizers at the same time. See the following figure.

You can select specific cases from the Cases list box and add them to the Selected cases for visualization list box (see the following figure).

For plotting mean and standard deviation you should add the cases to the Selected cases for visualization list box. Then press the Mean or StdDev button, respectively. You can even plot individual cases from the selected cases for visualization list box, using the right bottom option Plot.
The middle buttons are used as follows:

- **Add all from Cases list box to Selected cases list box.**
- **Add marked cases in Cases list box to Selected cases list box.**
- **Remove marked cases from Selected cases list box and add them to Cases list box.**
- **Remove all cases from Selected cases list box and add them to Cases list box.**

If the used methods for feature selection are "Sequential Forward F.S." or "Sequential Backward F.S.", you can display the selected features on Classifier design tab in the Data exploration tab's visualizers. To do that, you should check the **Display feature selection results** box (see next figure). If you choose the PCA method, then the **Display feature selection results** box will be disabled.

- **Display features disabled**
- **Display features enabled**

After checking **Display feature selection results** box, accordingly to the number of classes, a number of visualizers will show the selected features in each class. (See the following images). Note that you can see the mean (the pink plot), the standard deviation range (the yellow plots) and the selected features (green lines). The numbers in grey are in ppm and the yellow box to the right indicates the exact ppm number where the mouse is located in the visualizer. You can uncheck the **Mean** or **STDev** boxes to hide those plots.
Details of one of the visualizers (from Data exploration tab)

All the visualizers (from Data exploration tab)

You can use the right button of the mouse to change the visualization range. There are two options implemented: *Set full range* and *Set range 5 -> 0.5 ppm*.

Set full range option (from Data exploration tab)
You can clean the visualizers you need using **Select visualizers to clear** as shown below:

![Select visualizers to clear](image)

To clear visualizers (from Data exploration tab)

### 2.3. TAB: Data visualization

#### 2.3.1. Overview

**Data visualization** is the third tab of the application. You can use this tab to visualize resulting data from PCA or Fisher LDA. The following figure shows a Fisher LDA classifier visualization for the following brain tumor classes: low-grade meningioma, aggressive, low-grade glial and normal. The *KiNG* (Kinemage, Next Generation) library was used for three-dimensional visualization.
2.3.2. PCA visualization

To visualize results from PCA, first you have to select PCA in the Visualization method combo box (PCA appears in this combo box only if you make PCA before in Classifier design tab). Then you should indicate the pair of components to be set as horizontal and vertical axes. To set PC as axis, you should select one of the listed in the PCA list box and press Set as horizontal axis or Set as vertical axis button. You cannot select the same component to be both the horizontal and vertical axis. As you can see in the following image, the selected PCs will be shown on the bottom right of the box.

![Selecting the components to visualize (from Data visualization tab)](image)

The visualization will be as follows:

![PCA visualization (from Data visualization tab)](image)

2.3.3. Using the visualizer

The Data visualization panel is a 3D illustration of the corresponding point in the space of each case. In a 3D representation, the best way to take advantage of this visualization is by rotating it and twisting it around. Just click the mouse near the center of the graphic panel and slowly
drag right or left, up or down. If you get lost or disoriented, you can re-plot the visualization by means of the Plot visualizer button to start all over again.

All tumour types involved in the classifier are grouped into two main groups: Training cases and Testing cases. Testing cases group appears only if the user has provided a test set. Both groups are divided into subgroups, one for each class. The groups and subgroups appear in the button panel, to the right of the graphic area. Each one has a check-button to turn it on or off. Groups that are on (visible) have a check mark or an X in their box; a blank button means that the group is hidden.

By left clicking on the cases, you will access the identifiers of the cases. The identifier will appear on the bottom left of the graphics area. Furthermore, the distance from this point to the last one you clicked will also be displayed. For keeping track of which case is selected, markers can be displayed. Two markers are displayed normally. The checkbox Markers is just below the graphic area, with the Pick center checkbox.

You can make any case in the visualization to be the center. The center case will be in the middle of the plot, and the visualization will pivot around that case. There are several ways to set the center: you can hold the Shift key while you click the case, or use the right click, if you have one.

The Data visualization panel allows zooming in to see fine details in the visualization. Use the Zoom slider, below the plot, to control how far you zoom in. You can click the mouse right or left of the knob for small movements or click the knob and drag it for larger ones. A small motion is usually all that is needed. For easier access, one can hold Shift and drag (or drag using the right mouse button) in the plot. Dragging down zooms in; dragging up zooms out. The up/down arrow keys and mouse wheel (Java 1.4 and later only) also control zooming.

In a 3D representation, objects that are too near the viewer, or too far away, are not displayed; otherwise, zooming in would result in a useless superposition. The depth of the clipping slab can be adjusted using the slider along the bottom, or by dragging side-to-side with the right mouse button. The following figure shows some of the basic tools to interact with the visualization, explained above.

Some basic tools to interact with the visualization (from Data visualization tab)

The Show hierarchy button can help to modify the current visualization. The structure can be rearranged by cutting, copying and pasting elements; creating new elements and deleting undesired ones; and reordering elements (using the Up and Down commands). Play with these commands, and their operations will soon become obvious. Also, the properties of individual elements can be adjusted, which allows them to be renamed. Each element can also be toggled on and off. You can change the colors and the size of elements (see next figure).
To set the background of the visualizer, just select the Data visualization element in the Hierarchy window (see previous figure) and click Properties. Then, check the option whiteback and press OK. To return to a black background, then uncheck the whiteback option. On the left of the Data visualization panel, there is a Colour legend of the tumour types already plotted.

Note that each tumour type (pathology/condition) will have assigned a different colour. Then, each case is painted with the colour corresponding to their original tumor type (not the colour of the predicted class), for training cases and for independent test cases tagged with the information of the tumour type. Beware that not labeling the pathology/condition in a case
from an independent set case will cause the case to be given a class label (colour) by the previously developed classifier to be coloured with the assigned class colour, not with the known class colour.

The **Export visualization** button can be used to export the current plot view of the **Data visualization** panel as a standard image file. This button is located below the **Colour legend**, and on the left of the **Data visualization** panel.

As the original version of Fisher LDA does not assume any probability distribution to define the model, the limitation of Fisher LDA for estimating the probability of a case of belonging to a class, this has been overcome by approximating the resulting projections through spherical Gaussian distributions, one for each class. The center of each distribution has been assumed as the class mean estimated from data and the standard deviation common to all. Therefore, the probability of membership of every case to each class is estimated applying the Bayes’ theorem over these distributions:

$$P(\omega_i|X) = \frac{P(X|\omega_i)}{\sum_j P(X|\omega_j)}$$

where $\omega_i$ are each of the classes; $X$ is the projected case after applying LDA; $P(\omega_i|X)$ is the posterior probability; $P(X|\omega_i)$ is the normal distribution in the class $\omega_i$, with the mean of the projected data belonging to that class, and the common variance estimated for the entire set of projected data. The method assumes equal prior probability for each class, as LDA.

Classes’ boundaries are going to be displayed in case of a 2 or 3 -classes classifier visualization with Fisher LDA. The mean of the classes are also going to be displayed as white crosses.
2.4. TAB: Classifier evaluation

2.4.1. Overview

Classifier evaluation tab is the fourth tab of the application.

The following figure shows the number of cases by class. You can see the information related with the original training data set, original testing data set, estimated training data set and estimated testing data set. The Testing radio button will be enabled only if you loaded a testing data set.
Count of cases per class (from Classifier evaluation tab)

The subsequent figure allows seeing the right and wrong number of cases predicted per class. The red colour indicates the correctly predicted ones and the blue colour indicates the wrongly ones. Depending on the radio button selection, you will see the prediction for the training or for the testing data set.

Data prediction per class (from Classifier evaluation tab)

The next figure shows the **Confusion matrix**. Each row of the matrix represents the instances in a predicted class, while each column represents the real value of the instances in the original class. Confusion matrices are a metric for evaluating the classifier.
2.4.3. Store classifiers info

Storing multiple classifiers’ information can be useful for choosing the most suitable model. The practical way of storing information about the model is to add the info of an already built classifier to the Classifier history tab, by means of Append classifier info button, in Classifier evaluation tab. See the following figure.

2.4.4. Evaluation method

You can evaluate the classifier using the following metrics: balanced error rate (Balanced Error Rate radio button), without cross validation (Without crossvalidation radio button), by doing a cross validation with a specified number of repetitions (Fold crossvalidation radio button), using the maximum possible repetitions for the cross validation (Leave-One-Out radio button) or with a bootstrap method (Bootstrapping radio button).
The Balanced Error Rate (BER) [9] is the average of the error rate on the classes.

Cross-validation is used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. One round of cross-validation involves partitioning a dataset into complementary subsets, performing the analysis on one subset (training set), and validating the analysis on the other subset (testing set). In K-fold cross-validation, the original dataset is partitioned into K subsamples. Of the K subsamples, a single subsample is retained as the testing data for testing the model, and the remaining K−1 subsamples are used as training data. The cross-validation process is then repeated K times (the folds), with each of the K subsamples used exactly once as the testing data. The K results from the folds then can be averaged to produce a single estimation.

Leave-One-Out (LOO) method is a special case of a K-fold cross-validation. It uses a single case from the original dataset as the testing data, and the remaining cases as the training data. This is repeated such that each case in the dataset is used once as the testing data. This is the same as a K-fold cross-validation with K being equal to the number of observations in the original dataset.

Bootstrapping [13]: it is implemented by constructing a number N of bootstrap samples of the observed dataset (and of equal size to the observed dataset), each of which is obtained by random sampling with replacement from the original dataset (there is nearly always duplication of individual cases in a bootstrap dataset). The N results from the bootstrap samples then can be averaged to produce a single estimation. Bootstrapping could be better at estimating error rates in a linear discriminant problem, outperforming cross-validation.

The results of the evaluation will be shown in the subsequent panel.; see the following figure. For a four-class classifier (low grade meningioma, aggressive, low grade glioma and normal tissue) using 2 TE (Long plus Short), doing the evaluation with the Bootstrapping method, the overall mean accuracy for the correctly classified cases is 92.28%, with an standard deviation of 1.879%. By class, low grade meningioma was classified with a mean accuracy of 96.43% and a standard deviation of 2.546%.
Evaluation results (from Classifier evaluation tab)

Note that the mean and the standard deviation will be displayed if the cross validation or the bootstrap has been performed. When no-crossvalidation has been performed you will only see the mean.

2.4.4.1. Detailed example of K-fold cross-validation and LOO

For the K-fold cross-validation, we will exemplify a 5-fold cross-validation. The dataset used in this example contains 217 cases. Let’s divide the dataset in 5 subsamples (S1, S2, S3, S4, and S5), therefore K=5 (please see next figure). Of the 5 subsamples, a single subsample is retained as testing data for “evaluating” the model, and the remaining 4 subsamples are used as training data. The cross-validation process is then repeated 5 times (the folds), we have called this “stages” in the following figures. At every stage, 43 (217/5) different cases are used to test the model developed with the 174 remaining cases, obtaining for each stage the mean of the cases correctly predicted of the test group. Then, the total mean and the standard deviation of the correctly predicted cases can be calculated.
Example of a 5-fold cross-validation. (S: subset. C: case.)

In the LOO, the sample is partitioned in the number of cases in the original dataset (217 in this example), and a single case from the original dataset is used as testing data, while the remaining cases are used as training data (please see next figure). Then, the total mean and the standard deviation of the correctly predicted cases can be calculated as in the K-fold cross-validation method.

Example of Leave-One-Out. (C: case.)
2.4.5. ROC curve

A receiver operating characteristic (ROC) curve, or simply ROC, is a graphical plot of the sensitivity vs. (1 - specificity) for a binary classifier system as its discrimination threshold is varied. It is also known as a Relative Operating Characteristic curve, because it is a comparison of two operating characteristics (TPR = true positive rate & FPR = false positive rate) as the criterion changes. In the case of a classifier with more than two classes, each instance is analysed from the perspective of belong to a class or not while the threshold varies.

In this application, each ROC curve per class shows the results of the probabilities generated by the classifier dichotomising each class versus the rest of classes (1 vs. all). In the following image you can see three ROC curves, the red one corresponds to class mm (as shown in the legend), the blue one corresponds to class gl-me, etc.

[Image: ROC curve (from Classifier evaluation tab)]
2.5. **TAB: Reports**

2.5.1. **Overview**

*Reports* tab is the fifth tab of the application. Its purpose is to allow the user to export the results of Fisher LDA and/or PCA, as shown in the following figure.

The following report shows the classifier results for training and testing cases. Each row corresponds to a different case. The **Case** column is the identifier of the case. The **Tumour** column is the tumour type (if applicable, in case of test cases without tumour type, this column will be missing, this is the case of the example shown in the next figure). The **Orig. class** column is the corresponding original class, depending of the tumour type. The **Pred. class** column is the predicted class, obtained with Fisher LDA method.

The **X**, **Y** and **Z** columns correspond to the coordinates (in ppm). Their number will vary depending on the number of classes (It will be the number of classes - 1). The **Export results** buttons can be used to export these data to a text file.
2.5.3. Fisher LDA probabilities

The following report shows the probabilities of each case of being in each class (for training and testing cases). Each row corresponds to a different case. The Case column is the identifier of the case. The rest of columns correspond to the probabilities of being in each class. It will be a column per class. The Export results buttons can be used to export these data to a text file.
2.5.4. Determining probabilities

The general formulas used for determining these probabilities are described below.

Bayes' theorem:

\[
P(G_j|X) = \frac{P(X|G_j)P(G_j)}{P(X)}
\]

where:

- \(P(G_j)\) is the prior probability or marginal probability of \(G_j\).
- \(P(G_j|X)\) is the conditional probability of \(G_j\), given \(X\). It is also called the posterior probability because it is derived from or depends upon the specified value of \(X\).
- \(P(X|G_j)\) is the conditional probability of \(X\) given \(G_j\).
- \(P(X)\) is the prior or marginal probability of \(X\).

Intuitively, Bayes' theorem in this form describes the way in which one's beliefs about observing \(G_j\) are updated by having observed \(X\).
In *SpectraClassifier*, the implemented method to obtain these probabilities does not take into account the prior probability. It is assumed that this prior probability is the same for each group.

\[ P(G_j|X) = \frac{P(X|G_j)P(G_j)}{\sum_j P(X|G_j)P(G_j)} \]

For obtaining \( P(X|G_j) \), the following formula was used:

\[ P(X|G_j) = \frac{1}{\sigma \sqrt{(2\pi)^D}} \exp \left( \frac{1}{2\sigma^2} |X - \mu_j|^2 \right) \]

Since Fisher LDA uses distances to the mean of the classes to predict which class a case belongs, it describes a radial basis function. For this reason the Mahalanobis distance was replaced by Euclidean distance in the normal distribution function to approximate \( P(X|G_j) \).

The calculation of the standard deviation is described by the following formula:

\[ \sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2} \]

### 2.5.5. Weights matrix

The following report shows the weights matrix, associated to the corresponding feature. Those features are expressed in ppm and the number of axes depends on the number of classes (It will be the number of classes - 1). The Export Results button can be used to export this data to a text file. In case of a feature extraction with PCA the Feature column will be missing.
2.5.6. PCA results

The following three reports show the PCA results. The following figure shows the first one with the principal component functions.

PCA functions (from Reports tab)

The following figure shows two other PCA reports with the results of the numerical value obtained after applying each principal component function to every training and testing case. The Export results buttons can be used to export these data to a text file.

PCA results for training and testing cases (from Reports tab)
2.6. TAB: Classifier history

2.6.1. Overview

Classifier history tab is the sixth tab of the application. Its purpose is to have a place to compare classifiers made with different parameters. Every time you create a new classifier, the Append classifier info button in Classifier evaluation tab will allow you to add the principal information of this classifier toClassifier history tab.
2.6.2. Classifier info

Every Classifier info is composed by the names of the training datasets used to create the classifier; the composition of the classes or groups; the features selection or extraction method, the number of features and the list of them; the classifier method; the results of the evaluation of the classifier with the Bootstrapping method (1000 repetitions), and the Area Under the Curve (AUC) from the ROC curve. See the following figure.

Classifier info (from Classifier history tab)
The *Save brief info* button can be used to put this information into a text file. For instance, the corresponding text file for this classifier info will be as follows:

CLASSIFIER INFO:
Training data files:
DSS - longMRS.txt
Classes:
low-grade m (mm)
aggressive (gl, me)
low-grade g (a2, od, oa)
normal (no)
FEATURE SELECTION OR EXTRACTION
Method: SequentialForward
Number of features: 10
Features (in ppm): D51 2.0178ppm, D51 1.2315ppm, D51 1.5191ppm, D51 3.0342ppm, D51 3.7821ppm, D51 2.4589ppm, D51 1.1739ppm, D51 1.9794ppm, D51 2.1136ppm, D51 2.2863ppm
CLASSIFIER
Method: LDA/Fisher
EVALUATION
Total mean: 88.63%, with standard deviation 2.185%
low-grade m
Mean: 94.58%
STD: 3.059%
AUC: 0.987
aggressive
Mean: 81.73%
STD: 3.724%
AUC: 0.964
low-grade g
Mean: 96.78%
STD: 3.218%
The evaluation method is Bootstrapping

The *Save detailed info* button saves the whole information into two XML files, one of them with the information of the classifier and the other with the information of the dataset used for training. The name of the dataset file will be the same that was typed for the classifier, plus ",_dataset"; and the format of this file is explained in Format of the exported files section. The XML file with the information of the classifier has the following schema:
2.7. TAB: Multi-voxel reports

2.7.1. Overview

Multi-voxel reports tab is the seventh tab of the application. It allows displaying classification results of a given dataset as a colour-coded map.

Multi-voxel reports tab

2.7.2. Displaying classification results

For displaying classification results, the first step is loading files containing the information of the test cases investigated, then select which of them conform a complete case, and click the Plot image button for creating the colour-coded map.
2.7.2.1. Loading files

For loading files, the **Load files** button can be used (see the following figure).

![Load files](image)

**Load files (from Multi-voxel tab)**

For loading files, the allowed formats are:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMPM</td>
<td>Text file from Dynamic MRSI Processing Module (DMPM)</td>
</tr>
<tr>
<td>BSP</td>
<td>BSP file from Biospec spectrometers (Bruker)</td>
</tr>
<tr>
<td>XML</td>
<td>XML file with GABRMN format</td>
</tr>
</tbody>
</table>

Note that the voxel positions are mandatory for creating the colour-coded map.

After loading files, the name of them will be displayed in the **Loaded files** box. Using the buttons located between **Loaded files** and **Selected files** boxes as in the Data exploration tab, the files can be added and/or removed from the **Selected files** box. All spectral vectors in the files added to Selected files box will be displayed in the colour-coded map when **Plot image** button is clicked.

2.7.3. Plotting colour-coded map

For creating the colour-coded map, the following steps are carried out: first, the classifier developed in previous tabs is used to predict which class the voxels of the loaded dataset belong to and the probability of that classification. This information is then coded in HSB mode (Hue, Saturation, and Brightness) for each voxel, which in our case is used as follows. The Hue
parameter indicates the colour assigned to each of the predicted classes (e.g. blue, red, green), while the Saturation parameter (i.e. the tone defined by the weighted/mixed contributions of the Hue) defines the probability of each voxel belonging to the predicted class. The Brightness parameter is not used and therefore is assigned with a constant value in all cases (voxels). More details are provided next using an example. Below, there is a graphical illustration of the HSB colour composition of a given spectral vector (represented with an asterisk): the Hue parameter indicates the predicted class, the Saturation the probability of belonging to that class, and the Brightness is a constant value for all the cases. In this example, the classifier predicts that the spectral vector belongs to the class “blue” (Hue = 160), with an 80% of probability (Saturation value).

Finally, the nosologic-type map can be obtained by placing the HSB colour-code of each voxel in its geometric position on the original MRSI grid (see the following figure).
For modifying the number of rows and columns displayed, the Specify grid dimensions option can be used (see next figure). The width is related with the number of columns and the height with the rows. 32 rows and columns are set by default. The Set dimension button is enabled when the width and height values are changed.

2.7.4. Displaying probabilities

The classifier developed in previous tabs is also used to predict the probabilities of each spectral vector of belonging to each class (see next figure). The notation for identifying each voxel is the following: S, a number (a consecutive value for each voxel), open parenthesis, number of the row, x, number of the column, close parenthesis, dash, and the name of the file of this voxel. So, S1(10x11) - secondAligned.txt in the following figure means: spectral vector 1;
position in the original MRSI 10x11 (10 for the row and 11 for the column), and secondAligned.txt is the name of the file containing this spectral vector.

![Probability of belonging to each class](image)

**Predicted probabilities for each class (from Multi-voxel tab)**

2.7.5. Changing class colours

The colour of a class (assigned either by default or by the user) can be changed by clicking the rectangle corresponding to this class. See next the Colour legend of the map.

![Colour legend](image)

**Colour legend (from Multi-voxel tab)**

For instance, if the green rectangle is clicked (with the aim of changing the colour to the "normal" class), a Choose a colour window will be opened for allowing the user to select the new colour for this class (see the Choose a colour window below).

![Choose a colour](image)

**Choose a colour**
If a blue colour is selected like in the previous example, the colours in the map and in the legend will be updated, like in the following figure:

![Multi-voxel tab](image)

2.7.6. Exporting information

The colour-coded map can be exported as image, using the Export map image button at the bottom of the map.

The spectral vectors can be exported in the DMPM format, using the Export map data button at the bottom of the map. The Is labelled field will be equal to Yes, and at the end of each spectral vector row, the predicted class label for this spectral vector will be written.

The Export results (list) and Export results (maps) buttons, at the bottom of the probabilities box, can be used to export these data to text files.

The first one, Export results (list), can be used to save the probabilities per voxel and per class (see next figure).
The second one, Export results (maps), can be used to export two text files, one of them containing the class membership of each voxel, and the other containing the probabilities, both of them in the matrix format (see next two figures).
2.8.  TAB: Batch analysis

2.8.1.  Overview

The **Batch analysis** tab is the eighth tab of the application. It allows creating multiple classifiers with a single click, varying the number of features or principal components, and compare them.

![Batch analysis tab](image)

**Batch analysis tab**

2.8.2.  Starting the batch analysis

After importing the datasets for training, selecting the spectral range for classification, and defining the classes in the **Classifier design** tab, the user will be able to indicate the FS/FE and classification methods in the **Batch Analysis Setup** panel (at the top-left of the Batch analysis tab). The use of the fields in this panel is similar to the **Feature Selection And Extraction** and the **Classifier** panels. The only difference is that in the **Number of features** field now the user can indicate the numbers for which he/she wants to perform the FS/FE followed by the classification. For doing this, he/she can write numbers or intervals of numbers, as shown in the next figure. As in the case of the Number of features text field from the classifier design tab, this field also autocompletes according to the same formula\(^2\). This time the autocomplete

---

\(^2\) See the “Feature selection and extraction” section from the Classifier Design Tab
has the following form: “1–F”, where F is the maximum number of features that will be allowed by the system. Even if the interval [1 F] has a range bigger than 20 the user will be allowed to develop up to 20 different classifiers at one time. More than 20 classifiers can be developed if the Disable limits checkbox is activated. If the Set automatically option is chosen then the interval will be set to “1 − \( \frac{N_{\text{min}}}{3} \)”, where \( N_{\text{min}} \) is the number of cases from the smallest class.

Example: In a 2 class problem one class comprises of 10 cases and the other of 12. The Number of features field will autocomplete to F = 9, so the interval will be set to “1 – 9”. If the Set automatically box is activated then F will be 10/3 = 3. The interval “1-20” is also considered valid in this case but might lead to overfitting. The interval “15-25” is not considered valid and the system will not compute classifiers unless the Disable limits checkbox is activated.

![Batch analysis setup (from Batch analysis tab)](image)

2.8.3. Feature selection or extraction results

The features calculated will be shown in the **FS/FE Results** panel, for each classifier created, as shown in the next figure.
2.8.4. Evaluation results

In the Evaluation Results panel the confusion matrix and the ROC curves can be seen - for each classifier developed, one at a time. By default, the results of the first classifier will be displayed, and to see the rest, the user can select it in the 'Visualize details of' component (see the following figure).

Selecting the classifier to visualize its details (from Batch analysis tab)
The confusion matrix and the ROC curves will be displayed as follows:

Confusion matrix and ROC curves, for the selected classifier (from Batch analysis tab)

2.8.5. Performance of the classifiers

The bootstrapping results can be used to evaluate the performance of the classifier, in the Classifier Performance For Training Set panel. The mean and the standard deviation are shown for each class, and in general. See the following figure as an example.

Classifier performance (from Batch analysis tab)

The Bootstrapping results combo box (next figure) can be used to display the mean and standard deviation, total and by class. The option "All" displays the total mean, and the mean of all the classes in the same chart.
Selecting the option to display the bootstrap results (from Batch analysis tab)

The **Lower percentage** combo box can be used to adjust the minimum value of the left axis in the bootstrapping results chart. The default value is 0. If any of the percentages to be displayed is lower than the selected value by the user, it will be automatically readjusted to allow all the percentages to be shown.

The **balanced error rate (BER)** results can be used also to evaluate the performance of the classifier in the batch tab. See the following figure as an example.

The balanced error rate radio buttons (next figure) can be used to between displaying either training or testing datasets.
REFERENCES


