Manual

MultiSensAnalyzer

Version 1.1

JLM Innovation GmbH
Vor dem Kreuzberg 17
72070 Tübingen
Germany
Tel. +49 7071 5667730
Fax: +49 7071 5667731
email: info@jlm-innovation.de

JLM Innovation GmbH
Dr. Jan Mitrovics
Vor dem Kreuzberg 17
72070 Tübingen
Germany
Tel. +49 7071 5667730
Fax: +49 7071 5667731
email: info@jlm-innovation.de

March 2012

Table of Contents:

1 Introduction

1.1 General

This operating manual describes the MultiSens Analyzer software package. It addresses customers who are already familiar with the basics of chemical sensing combined with the numerical recognition algorithms.

MultiSens Analyzer is distributed in different versions that include different sets of functionalities. Depending on your version functionality described in this manual may not be available.

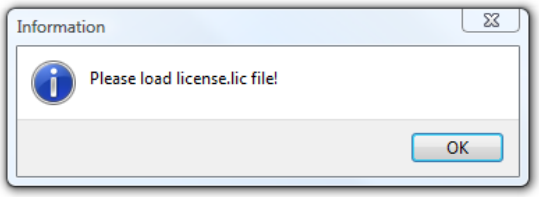
MultiSens Analyzer is constantly improved. New functionality may be introduced, that is not already described in this manual.

1.2 Installation

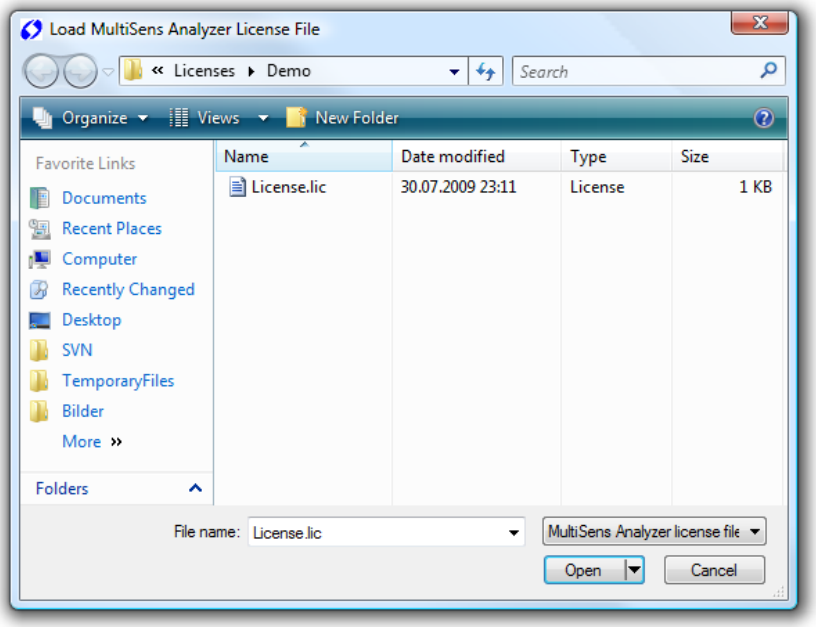
Typically this program is delivered as an executable setup file (setup.exe). Starting setup.exe will guide you through a simple set of dialogs to install MultiSens Analyzer on your PC.

If the installation was carried out successfully, you can continue directly to work with your system. A re-start of Windows is not required.

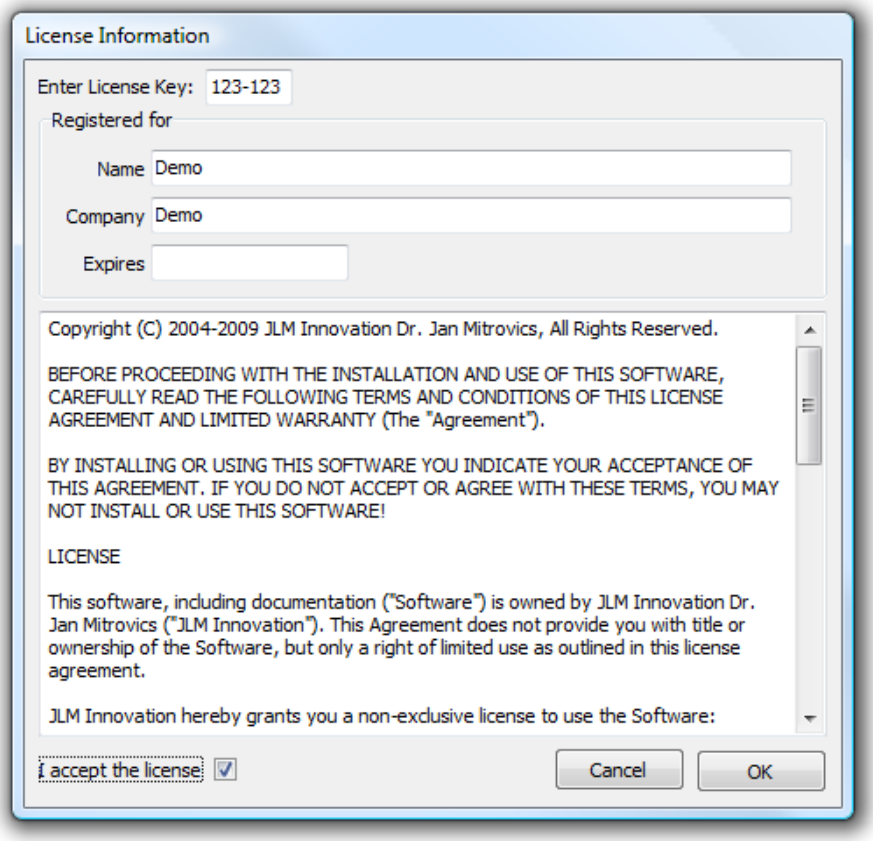
When you are first running MultiSens Analyzer, you will be prompted with a license dialog. You might be asked to provide a license.lic and license.txt file.



Both files are typically provided via Email. Please store both files to your hard disk. MultiSens Analyzer will present a file dialog to locate the files. Please note that file dialogs do look different depending on Windows version and language.



When both files license.lic and license.txt are found the license information is shown. In order to proceed, you have to enter the License Key and click checkbox next to "I accept the license".



Special versions of MultiSens Analyzer are distributed on CD-ROM or USB-Stick and require no explicit installation via a setup program. These versions may directly be started by clicking on MultiSensAnalyzer.exe. Trial versions display the license dialog at every start of the program and may run without entering a license key.

1.3 Checking for updates

MultiSens Analyzer is continually improved. We have included a function in MultiSens Analyzer which automatically checks for updates, whenever the program is started. Your computer needs to be connected to the internet, as this function will fetch information from our servers.
You can disable this function by selecting **Help – Check for Updates** from them menu and disabling the option "**Automatically check for updates during program start**".
Select **Help – Check for Updates** at any time to check for updates manually.

If an update has been found you can download and install the update by clicking on the "Install" button.

1.4 System Requirements

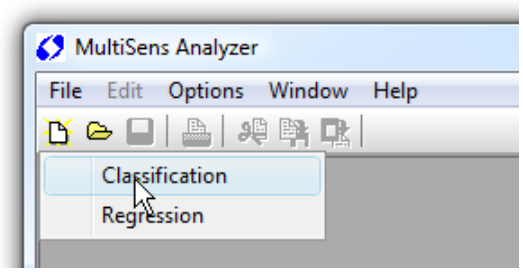
MultiSens Analyzer will run on a computer with Windows XP, Vista or 7.
A processor with 1 GHz, 512 MB RAM, and 100 MB free disk space is the minimum requirement. For complex data sets faster processors (> 2GHz) and more RAM (2 – 4 GB) are recommended.

2 MultiSens Analyzer step by step

2.1 The first evaluation

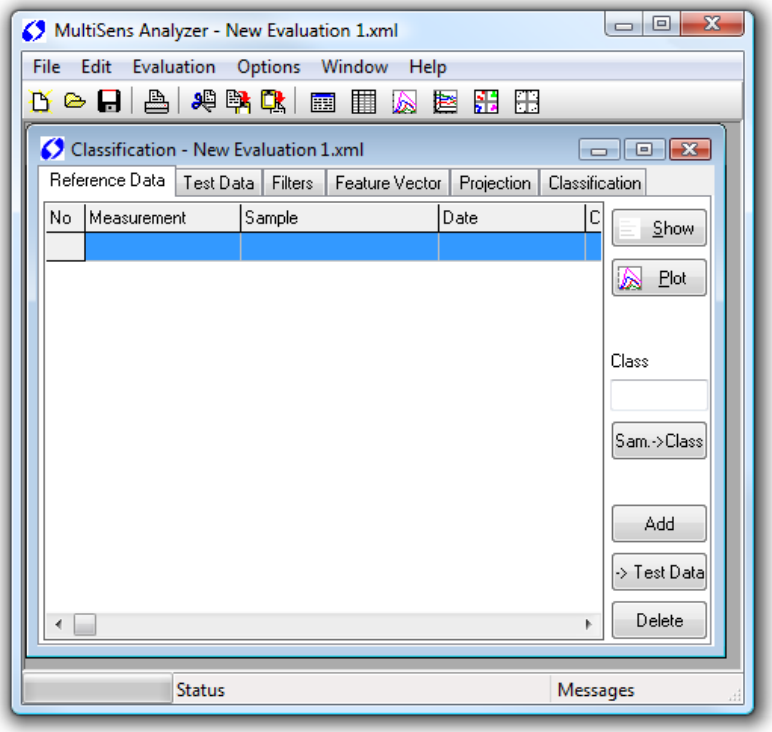
MultiSens Analyzer main purpose is to evaluate measurement data from multi sensor instruments. Two different types of evaluations are available in MultiSens Analyzer: Classification and Regression (the version MultiSens Analyzer Basic only includes Classification).
Classification is used to assign measurements to a specific class from a set of classes (e.g. Apple, Orange or Banana) while Regression is used to calculate a value that describes a quantifiable property (e.g. a concentration)

Step 1: Creating a new classification:
*From the main menu select **File – New – Classification**. Alternatively you can use the icon bar. Through the icon bar you have fast access to the most important menu commands. Click on the leftmost icon in the icon bar. A menu pops up. In this menu choose **Classification**.*



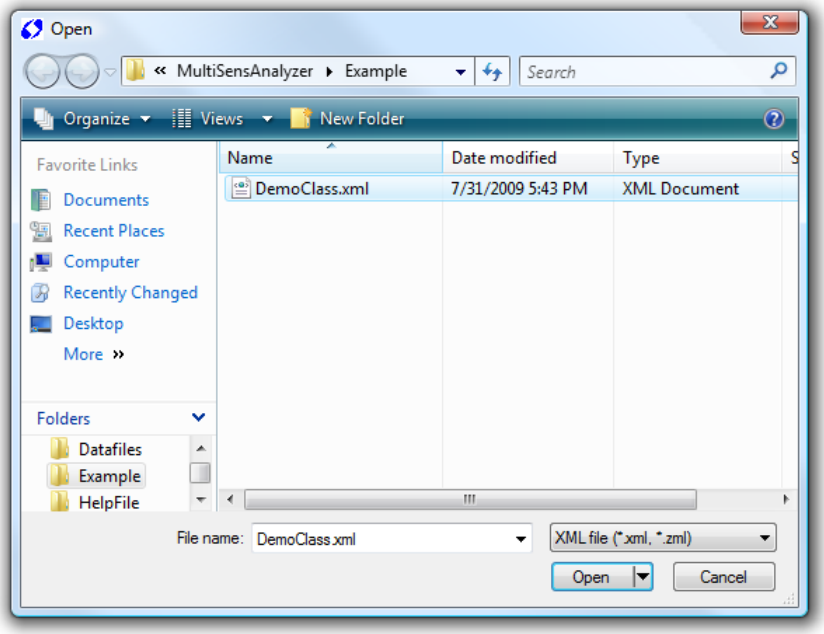
Hint: If the mouse cursor remains over an icon for a few moments without clicking, a short descriptive text will appear.

An evaluation compares several measurements with each other, or with reference data. What exactly are reference data? Reference data contain those measurements that were used to build the mathematical model that is used as the framework to predict new measurements. It is impossible to build a model without reference data. When creating a new evaluation, designating reference data (which must already exist at this time!) is the first step. Consequently, the corresponding page is automatically activated:



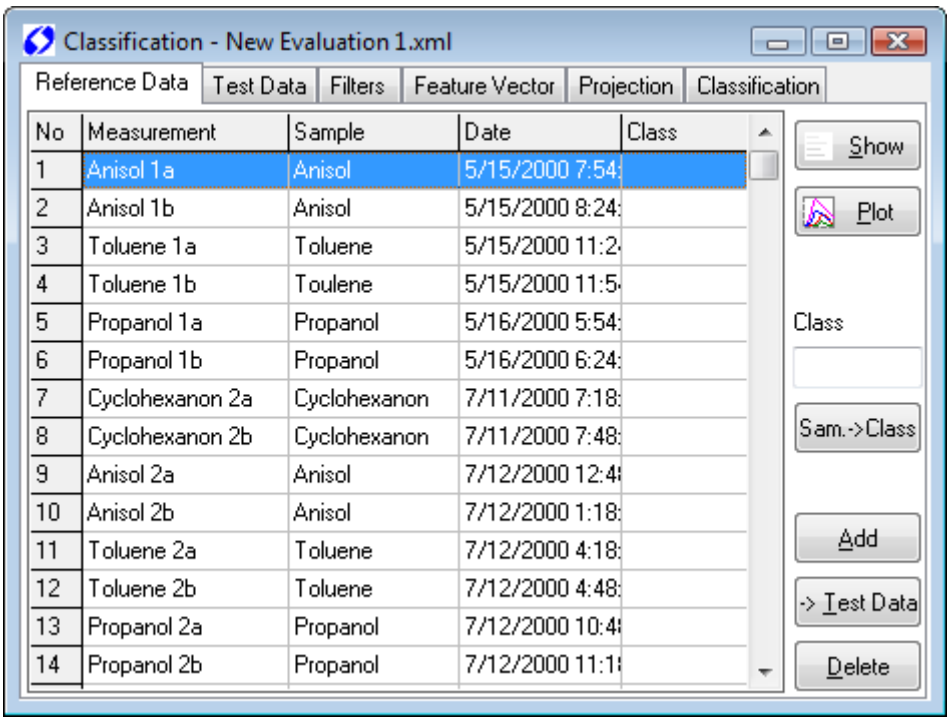
Step 2: Introducing reference data:

*Click on the **Add** button. A new dialog window will appear. MultiSens Analyzer can import data in a number of formats. By default the default import settings are activated (which can be defined via the Options menu). Depending on the file format a single file may contain several or only a single measurement. In this tutorial, we want to open an xml file.Use the drop down list in the lower right corner to select XML file, then find the DemoClass.xml file and click **Open**. Please note, that while MultiSens Analyzer is available only in English this dialog will use the language of your windows installation and also depends on the version of windows.*



Hint: You can repeat this procedure a number of times in order to add further measurements to your reference data. You may also import several files in one step. To highlight many files at once, either click-drag the mouse to draw a marquee around the desired number of file names, or highlight the first file name, then shift-click on the last file name.

Finally, the result should look similar to the example shown below. The sequence of samples is of no importance.



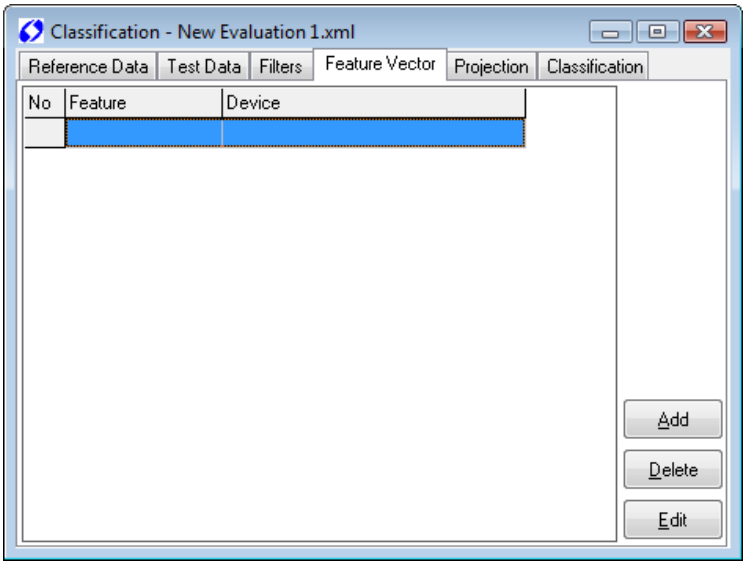
Each measurement contains the entire response of all sensors to one sample. Obviously, the complete response curve contains a lot of redundant information. For many types of pattern recognition algorithms, this is problematic and will in any case increase the processing time drastically. Hence, it is desirable to greatly reduce the amount of data to be processed.

This is achieved by extracting certain traits (e.g. the maximum of the response curve) from the raw signal. Such traits are called 'features' in the MultiSens Analyzer software. The entire set of features is called the 'feature vector'. The 'feature vector' comprises a number of values extracted from the sensors' reaction to one sample.

An optional step might be to remove noise or other influences with a filter before extracting features. In this short introduction we will not use filters and extract features directly from the sensor responses.

Step 3: Generating a feature vector:

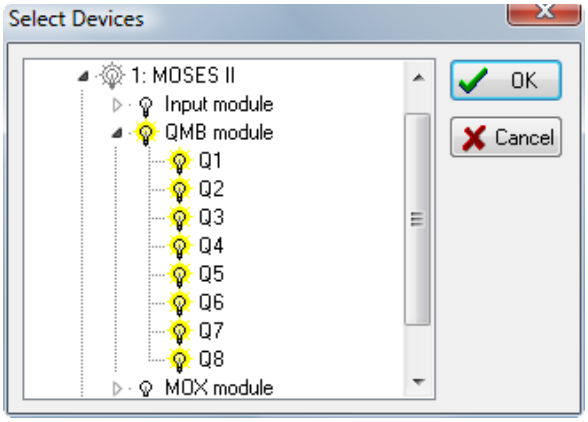
*Click on the tab labeled **Feature vector**. This will bring the corresponding page to the foreground.*



At this point, the feature vector does not contain any features yet. The logical next step is to first select the sensors to use, and to then define the type of feature extraction to apply. It is not necessary to use the same type of feature extraction for all sensors.

Step 4: Choosing the sensors:

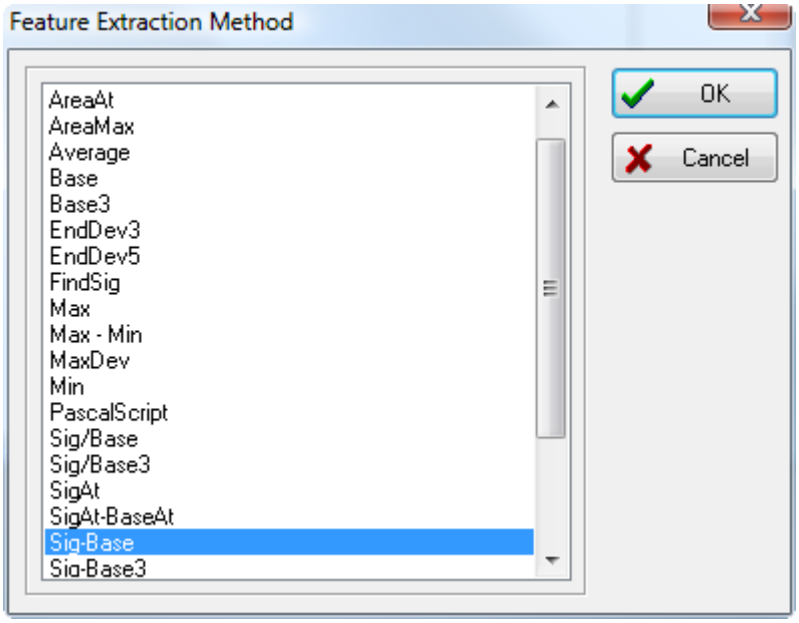
*Click **Add**. You will be presented with a new dialog window with a tree view of all devices in the measurements of the evaluation. Click the + sign to expand a node. Selected sensors are represented by a lit light bulb next to it. For our example, expand the module '**QMB module**', and select all QMB sensors (**Q1-Q8**). Clicking on a decive / module will select/deselect all contained sensors. When done, click **OK**:*



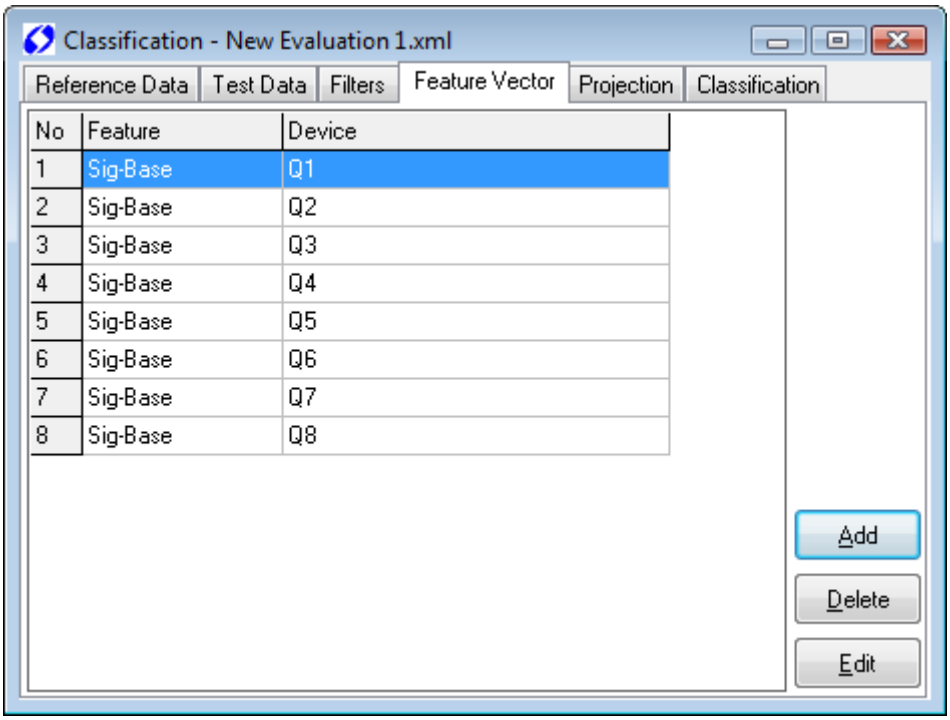
As discussed above, the next step is the choice of feature extraction method.

In this step we will define which value will be extracted from the sensors' response curves. The MultiSens Analyzer software makes a number of different options available to you which we will discuss in more detail later. For the time being, let us just use one of the simple methods, i.e. we will just extract the curves' peak values. The corresponding feature extraction method is called Sig-Base (shorthand for "signal minus baseline").

Step 5: Choosing the feature extraction method:
*In the list select **Sig-Base** and confirm your choice by clicking **OK**:*

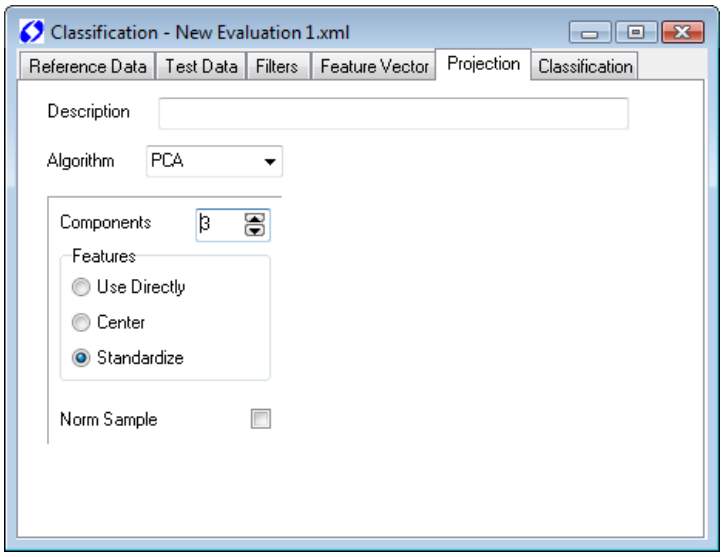


This will add eight entries to the list of features, i.e. one feature called Sig-Base for each sensor.



Hint: The composition of a feature vector can be chosen very liberally. It is possible to choose only one type of sensor or even one single sensor only. Identical or different features from one sensor can be present several times in one feature vector. Obviously, the influence of one particular sensor on the final result increases with its number of appearances in the feature vector.

By now, we have defined what measurements to evaluate and how to extract one value (feature) from every QMB sensor in the measurement. Now we will select the right parameters for the evaluation. Click on the Projection tab in the parameter window.



Step 6: Set the number of components:
On the Parameter tab set the number of ***Components*** *to 3.*

All other parameters should be left at their preset default value for the time being. The parameter **Components** determines how many principal components are calculated during the evaluation.

Before we actually perform the evaluation, let us elaborate a bit more just how the MultiSens Analyzer implements the evaluation. In this version of the MultiSens Analyzer, Principal Component Analysis (PCA) is built in as the standard processing algorithm.

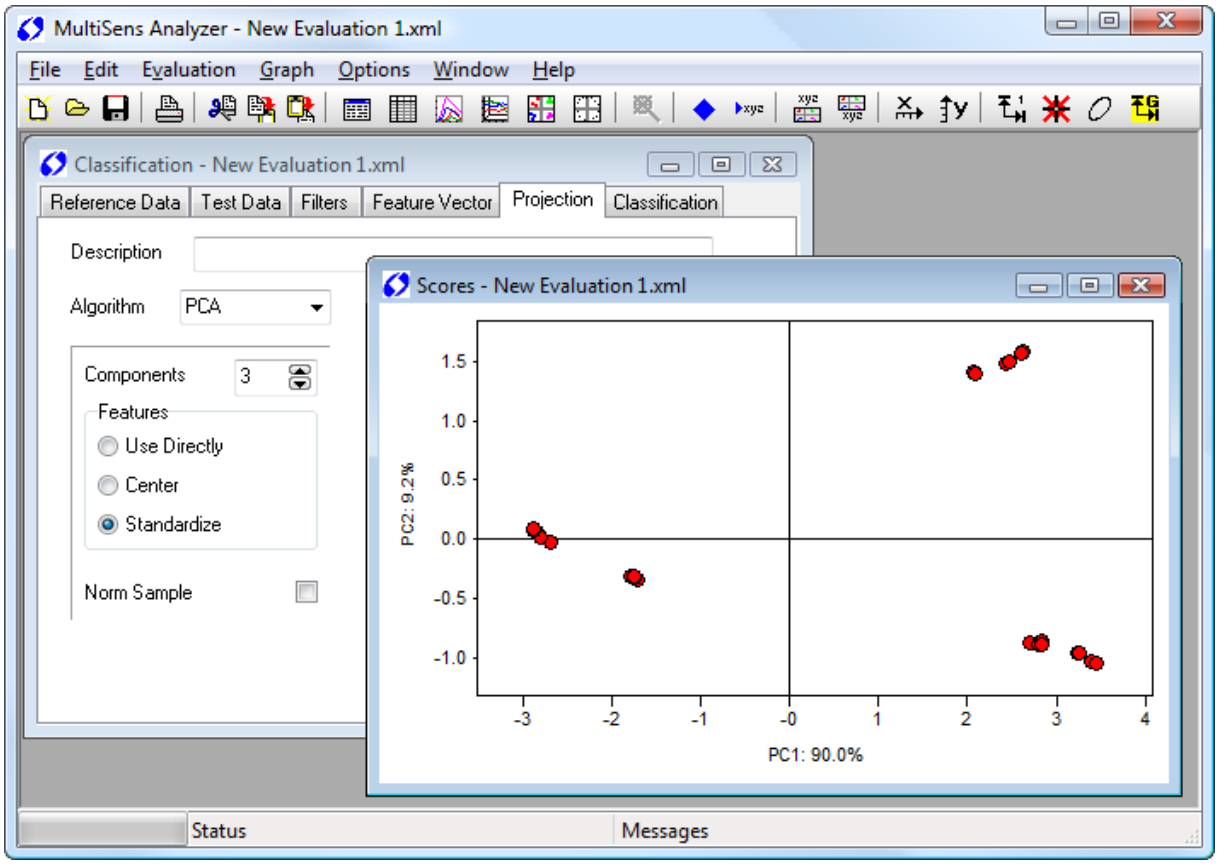
Maybe surprisingly, there is no explicit command to launch the evaluation. MultiSens Analyzer will automatically recognize which actions are required to achieve what you mean to achieve. Hence, to have the evaluation executed, all you have to do is to request the display of the results of the PCA. The program will notice that the principal components have not yet been determined, and will automatically launch the required algorithms. All final and intermediate results will be stored in memory. This will make recalculation a lot faster because the results from a previous calculation may be re-used.

The scores plot is by far the most popular type of display for the results of a PCA. The scores plot shows the different samples with respect to two principal components (normally, the first and second PCs are shown). Similar samples turn out to be close to each other. Different samples lie further apart.

Step 7: Showing the scores plot:
From the menu, select ***Evaluation – Scores plot****, or click the corresponding icon.*

The program will now create a new window to contain the scores plot. Since the data have not been yet calculated, this is going to take place now. The first step is the extraction of features from the measurements. Once this has been completed, the principal component analysis proper will be performed. On a decently fast computer, the whole procedure will only take a few seconds.

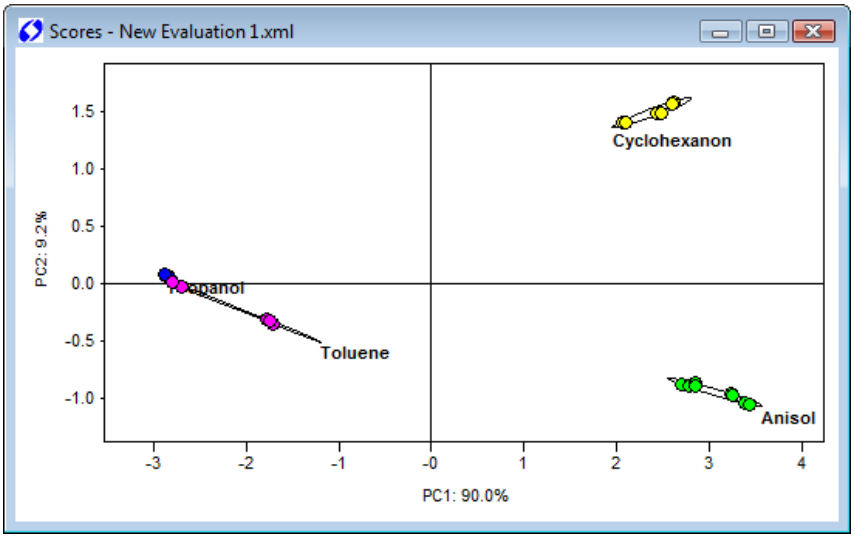
The result should be similar to the example shown next:



The resulting scores plot shows several groups of measurements. We could identify the individual samples by printing their measurement names next to them (Graph – Label – Measurement), however a better way is to assign the measurements to specific classes. While this information is not used in a PCA to calculate the model, the graph will identify measurements belonging to different classes with different colors.

Step 8: Assigning classes to measurements:

*Go back to the **Reference data** tab and click on the button "**Sam.->Class**". This will copy the sample information the class for all measurements.*



The scores plot will automatically reflect the changes. The different classes are indicated by different colors and by special ellipses drawn around the measurements belonging to one class. The class names are drawn next to the ellipses.

Finally we need to save the evaluation.

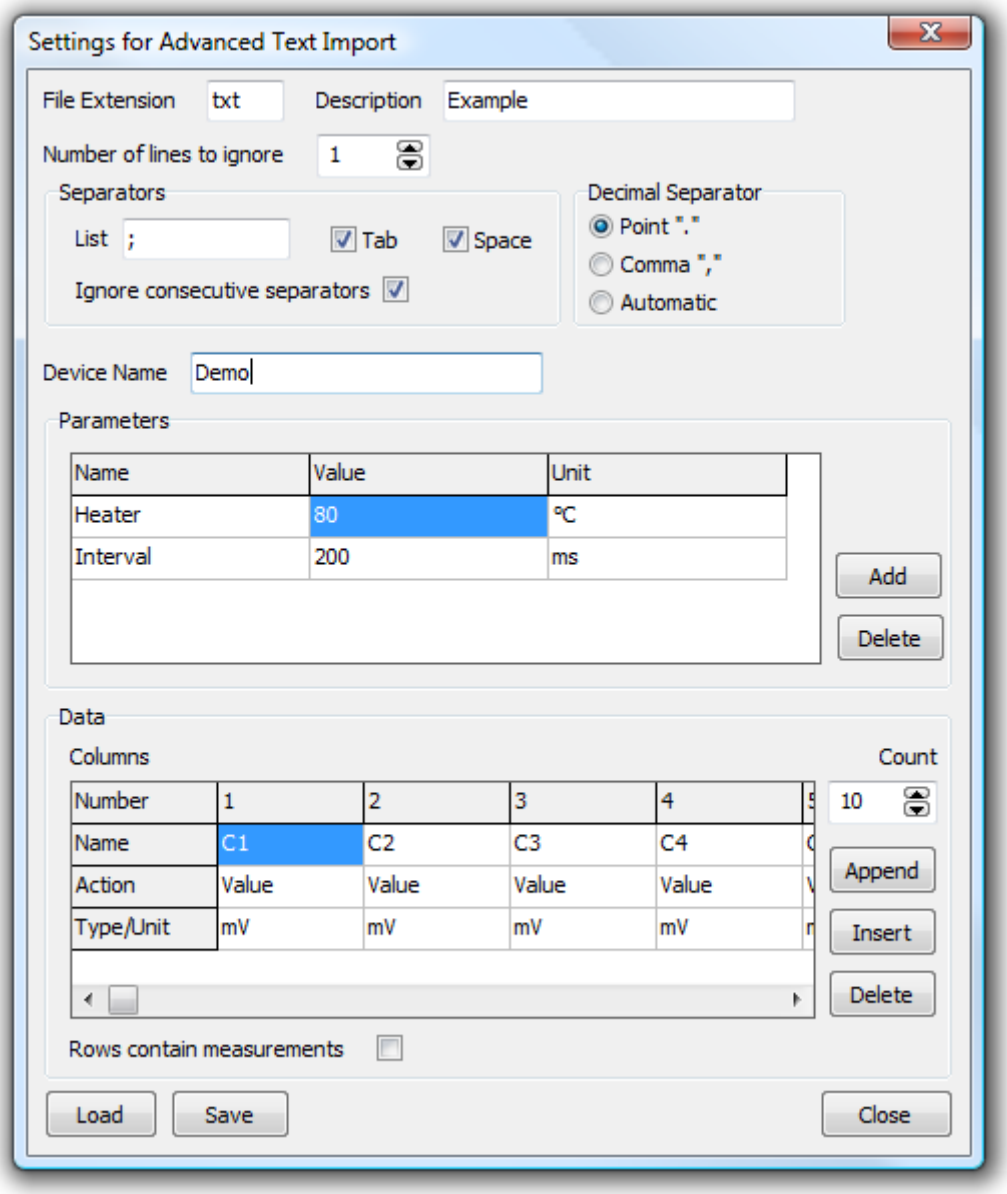
Step 9: Saving the evaluation:
*From the menu, select **File – Save as**. Like in the previous examples of saving files, select the directory that you wish to save the evaluation file to, enter a descriptive name, and click **Save**.*

This concludes the creation of your first evaluation.

2.2 Importing your data

MultiSens Analyzer includes import functionality for a number of different file formats. Whenever you add measurement data to an evaluation you may choose in the file open dialog between the various import formats. Please refer to the appendix for more information in the various input formats.

A typical format that is often used for measurement data is text data (ASCII). However it is necessary, to define in detail how the text data within the file is structured. MultiSens Analyzer includes some import functions that read will typical text files that we came across (see appendix). More often you will need to define how to read your text data files. This can be done via selecting **Options – Default Import Settings** in the menu. This will bring up following dialog:



Enter the extension of the data files that you want to import into the field **File Extension**.

Description may contain a short description for the data file. You could enter the name or type of the instrument that generates the data files that you want to import. The content of this field is shown in file open dialogs.

Many data files contain a header that cannot easily be imported. By setting the value of **Number of lines to ignore** you can skip these lines during import.

Separators define how individual values are separated. Enter any characters into **List** to set them as separators. Additionally **Tab** and **Space** may be activated as separators. If **Ignore consecutive separators** is activated a new value will only be generated, if a non separator

character is read from the text file. If **Ignore consecutive separators is not activated**, every separator will create a new value, that may be empty.

The **Decimal Separator** discriminates the integral from the fractional part in a numeral value. Either a **point "."** or **comma ","** may be chosen. If set to **Automatic** the first comma or point, found in the numeral value, will be taken as decimal separator. Thousands separators are not supported!

Device Name lets you set the name of the device that is used to generate the data.

Parameters is a list of arbitrary parameters that you can add to the import. These are fixed value parameters (all imported files will have the same values (entered in this dialog).

The measurement data in the text file is expected to have the form of a two dimensional array. The format is defined under **Data**:
If **Rows contain measurements** is checked, then each row in the data array contains an independent measurement. For each sensor/device only one measurement point is included in the file. If **Rows contain measurements** is not checked, then only one measurement is included in the file, but for each sensor/device a complete response curve may be included.
The input grid **Columns** contains the definitions for the column of the data matrix in the text file. The number of columns in this grid must match the number of columns in the text file.
For each column we have to assign a **Name** and optionally a **Type/Unit**.
Action defines what to do with the data in the column. Following actions are available:

Action	**Description**	**Format**
Value	Sets the Value of the sensor/device.	Decimal number
Measurement	Sets the name of the measurement.	String
Ignore	Ignores the column.	-
DateTime	Sets date and time of the measurement.	Date and time according to Windows language settings
Date	Sets date of the measurement.	Date according to Windows language settings
Time	Sets time of the measurement.	Time according to Windows language settings
Year	Sets the year of the measurement.	number
Month	Sets the month of the measurement.	1 .. 12

Day	Sets the day of the measurement.	1 .. 31
Hour	Sets the hour of the measurement.	0 .. 23
Minute	Sets the minute of the measurement.	0 .. 59
Second	Sets the second of the measurement.	0 .. 59
Sample	Sets the name of the sample.	String
Parameter	Sets the content of a parameter.	String
Float Parameter	Sets the content of a parameter.	decimal number
Integer Parameter	Sets the content of a parameter.	number
Class	Sets the class that the measurement is assigned to	String
Target Value	Sets the target value that the measurement is assigned to.	Decimal number

3 Detailed description

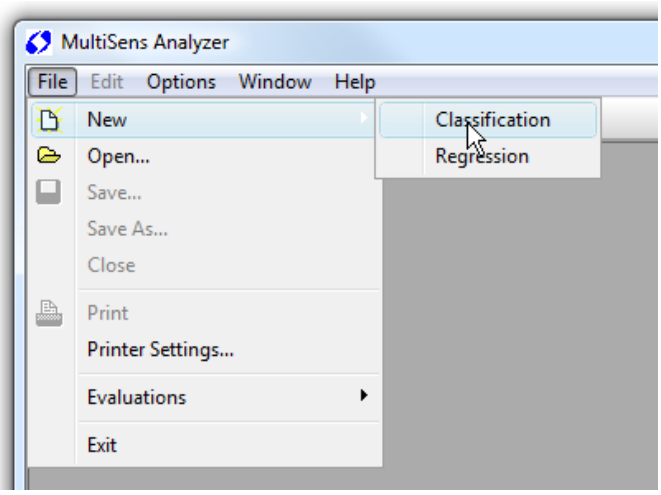
3.1 The concept of the MultiSens Analyzer software

MultiSens Analyzer allows performing of many different analysis methods. Despite this multitude of functions we have focused on an easy usability. All data of an evaluation is stored altogether in a single file. Several evaluation files can be kept open and active simultaneously.
The evaluation files are stored in a well defined XML format.

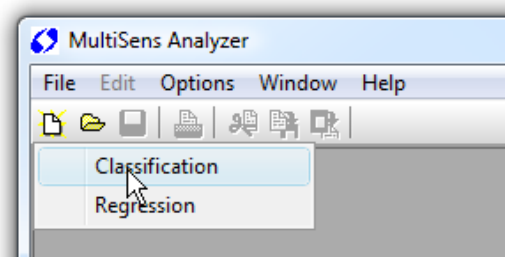
3.1.1 The most important command: File - New

Since MultiSens Analyzer uses a file based concept, new evaluations are created via the command to create a new file. This command can be reached in two ways:

First, by selecting the **New** submenu from the **File** menu (e.g. by pressing Alt-F followed by N),



and second, by selecting the first symbol from the left in the icon bar.



In either case you can select different evaluation types.

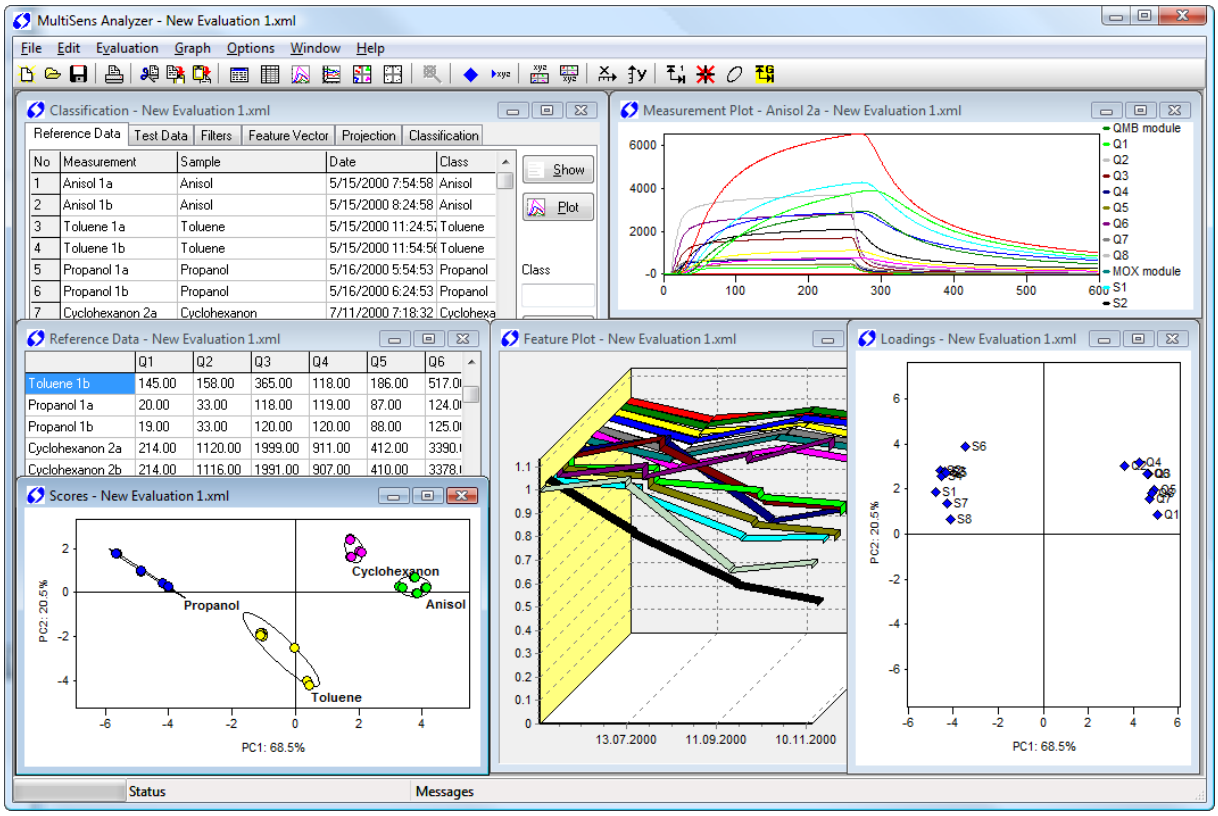
3.1.2 Files and windows within MultiSens Analyzer

There are several possibilities to represent the contents of a file in MultiSens Analyzer and to have several views of one file on the screen simultaneously in different windows. To close all

windows belonging to one file, select **Close** from the **File** menu. If you have previously made any changes to the contents of the respective file, MultiSens Analyzer will prompt you to confirm whether you wish to save these changes. If you reply "No", your changes will be lost. If you close one of several open windows pertaining to the same file, only the closing of the last remaining window will invoke the prompt dialogue.

MultiSens Analyzer is a multiple document application. This means that you can keep several files opened simultaneously. For example, you may want to have two different evaluations on the screen at the same time, making direct comparison easy. At any point, you may simply open the respective file without having to close what you are currently doing. However, this possibility should be used sparingly; there is a very real danger of cluttering the screen with dozens of open windows. Also, a decent monitor size (17 or more inches) is preferable for use with MultiSens Analyzer.

The **Window** menu lets you change between windows, and provides several possibilities for arranging windows on the screen (e.g. **Cascade**, **Tile horziontal**, or **Tile vertical**).



The example above shows an open evaluation with some corresponding windows.

The content of the menu line depends on the currently active window. Most menu commands refer directly to the active window and its file. The title bar of the active window is shown in a different color.

It is possible for the menu line to change its contents when you change from one window to another one (the same is true for the icon bar). In addition, some commands depend on the contents and state of a file.

3.2 The evaluation

An evaluation is used to compare measurements with each other, and/or to classify new measurements according to previously recorded measurements.
Creating a new evaluation is achieved by creating a new file: **File – New – Classification (or Regression)**. Instead of always starting from scratch you may also choose to load an existing evaluation, modify it, and save it under a different name.

3.2.1 The basic procedure

MultiSens Analyzer has a number of different analysis methods built in. These analysis methods are grouped into two main evaluation types.
A **Classification** is used to assign unknown measurements to a set of classes. A **Regression** is used to predict quantitative data, i.e. a value gets assigned to an unknown measurement. The choice between **Classification** and **Regression** is thus based on the type of result that you want to achieve (qualitative vs. quantitative).
Within each evaluation type there will be a number of different algorithms that you can use to perform the analysis. **P**rincipal **C**omponent **A**nalysis (PCA) and **P**rincipal **C**omponent **R**egression (PCR) are two standard algorithms built in MultiSens Analyzer. These algorithms have the striking advantage of being relatively intuitive and easy to use. There are only a limited number of parameters to "tweak", and the result is ready for immediate interpretation.
Modern computers are fast enough to calculate most algorithms in very little time. MultiSens Analyzer makes good use of this property by invoking recalculations automatically as soon as any parameters have been modified or any data are added or removed. You never have to explicitly request a recalculation. There isn't even a command to achieve that. The program will recognize automatically what needs to be done.
Normally you will want to start evaluation work flow by defining which measurements you want to include. Then, define a few basic parameters (definition of the feature vector). You will want to view the result immediately as a scores (scatter) plot. In addition you may want to open further windows containing the features matrix, a loadings plot, etc.

All these windows pertain to one single file though. They just show different views. Changes made in any one of these windows will immediately proliferate to all other windows. Hence, you will hopefully soon find this an inspiring environment that will encourage you to "play" with your data in many different ways. Furthermore it is possible to load several evaluations simultaneously and compare the different results.

3.2.1.1 Reference data

To start an evaluation you will need reference data. Those data that are used to construct the mathematical model on which the evaluation is based are called reference data - no reference data, no model. It is, however, possible to perform an evaluation with only reference data (i.e. without any independent test data)

3.2.1.2 Adding Filters

Filters in MultiSens Analyzer allow recalculating a complete response curve of a sensor. A common use for filters is to remove noise from a sensor response.

Within MultiSens Analyzer a filter actually is implemented like an additional (virtual) sensor. It does not alter the response curve of a sensor. Rather a new (virtual) sensor is created that contains the filtered response curve. This new sensor can then be chosen as source for subsequent algorithms.

One advantage of this approach is, that you can chain filter algorithms. After creating a Filter you can choose the output from this filter as source for a new filter.

Another advantage is that you can create graphs that show filtered and unfiltered data from one sensor and thus compare the effectiveness of a filter algorithm. As all graphs are automatically updated whenever corresponding parameters change, you may even visually optimize filters by opening suitable graphs and changing the filter parameters.

A list of filter algorithms is included in the appendix.

3.2.1.3 The Feature extraction

Theoretically it would be feasible to feed the raw sensor response curves into a regression or classification algorithm, with no prior treatment. However, the curves contain a very large amount of redundant information which would serve no other purpose than to blow up computation time unnecessarily. By extracting characteristic values (features) from the entire curve, a much reduced data set is obtained which (hope-fully) still retains most of the information present in the full curve and emphasizes the relevant information. This procedure is called feature extraction.

As soon as the reference data set contains at least one measurement it is possible to define how to extract the feature vector from the individual sensors response curves. The feature vector thus

defines, which data goes into the subsequent classification or regression algorithm. You need to explicitly choose the algorithms and sensors to be used.

After the feature vector is defined, the calculation does not need to be started manually. Again, the MultiSens Analyzer software will automatically invoke the required algorithms.

The most popular feature is the one that determines by how much the sensor signal has changed during the exposition to the gaseous sample. In MultiSens Analyzer parlance, this feature is called **Sig-Base**, shorthand for Signal minus Baseline. This means that the largest positive or negative deviation (Signal) from the initial value (Baseline) is extracted.

Features whose name ends in "3" use an additional averaging over three adjacent data points. Whether or not this averaging makes sense depends on the particular sensors and application.

A list of feature extraction algorithms is included in the appendix.

3.2.1.4 The principal component analysis (PCA)

After feature extraction, each measurement will contribute one or several feature vectors. The feature vectors span a multi-dimensional space (feature space) where each measurement is represented by a point. The features are the base vectors for this space; if only one feature per sensor is extracted (e.g. **Sig-Base**), the feature space has as many dimensions as there are sensors. PCA tries to find new base vectors (which are linear combinations of the original base vectors) for feature space. Thus, drawing an analogy to our familiar three-dimensional space, PCA rotates the coordinate system. Obviously, a simple rotation of the coordinate system will not help a great deal in reducing the space's dimensionality. Actually, it will not help at all.

The trick with PCA is that the new base vectors are chosen according to the variance of feature vectors. The first base vector will be chosen so that it represents the direction along which there is the largest variance; the second base vector will show the second largest variance, and so on. The base vectors thus found are then called principal components. The first principal component therefore points in the direction where the reference measurements show the greatest degree of variation.

This method will only help if the variance of feature vectors is not sufficiently well represented by the original base vectors. This is the case when there is cross sensitivity between sensors, and when the sensors are not overly selective.

The scores plot (scatter plot) shows data projected onto a plane which is spanned by two principal components. By default, the X axis is the first principal component while the Y axis corresponds to the second PC. As discussed previously, these two PCs represent the bulk of the variance present in the data set. The distance between samples corresponds directly to their degree of likeliness as "seen" by the sensors.

The scores plot is a display technology to get similarities and differences of samples at a glance.

3.2.1.5 Classification of test data

Frequently, the goal of an evaluation is the classification of an unknown sample, i.e. to determine the degree of similarity with known classes of previously analyzed samples. The MultiSens Analyzer software has built-in algorithms to classify an unknown new sample based on the class membersip of the reference data. To this end, the program obviously needs to be told which classes the individual reference data measurements belong to.

One algorithm is to calculate distances of a new sample to the reference data, and select the class most frequently found in the k nearest neighbours, where k is a non-zero, user selectable integer number. This algorithm is also known as the KNN method (k-nearest-neighbours).

Additionally to the classification by the KNN method it is possible to calculate the Mahalanobis distance. The Mahalanobis distance is a value which gives the dissimilarity of a sample in respect to a certain class. A value of 0 means the sample exactly matches the class. Large values are indicating a high dissimilarity to a given class.

The computation of the Mahalanobis distance calculates the variation of samples within a class around its centre in respect to the distance of the new sample. Consequently the Mahalanobis distance takes the area of the reference data into consideration. To allow this calculation it is necessary to have enough reference data for a given class. The number of reference data points always has to be larger than the dimension of the vector space of the computation. If this doesn't hold the program will give the output *NaN* (Not a Number). Since for practical reasons the number of reference data points is low it is useful to use the scores as input data for the Mahalanobis distance. The values for the Mahalanobis distance are displayed next to the sample description. Here all distances to all classes are shown giving the description of the class first followed by the Mahalanobis distance. Additionally to the sample values the Mahalanobis distance values for the centres of the classes will be displayed next to the class description.

3.2.1.6 The linear discriminant analysis (LDA)

LDA is an alternative method to calculate a new coordinate system for the representation of the measurements. In contrast to PCA where the coordinate axes are chosen to represent the maximum variation within the reference data, LDA will explicitly use information about the class memberships of the reference data to find a representation that maximizes the distances between data belonging to different classes and minimizes the distances between data belonging to the same class.

The number of coordinate axes (equivalent to the scores of the PCA), that an LDA will generate, is determined by the number of different classes.

3.2.1.7 The principal component regression (PCR)

Principal Component Regression is an extension of the Principal Component Analysis. During calibration for each sample a certain user defined or independently measured target value (e.g. a concentration value or a quality index) is assigned.

As described before the principal components are calculated. Using these principal components a multi linear regression is subsequently performed with the user defined values as target data. Resulting regression coefficients used together with the Loading values of the PCA allows the prediction of unknown samples.

3.2.1.8 The partial least squares algorithm (PLS)

As an alternative to PCR, the PLS algorithm may be used. PLS takes the target values of the reference data into account when calculating the scores. It optimizes the regression from the features to the target values by choosing a scores projection that minimizes the regression error.

3.2.2 The Classification

3.2.2.1 The parameters

An evaluation's parameters are collectively displayed on several pages within one window. Page one and two contain lists of reference and test data, used in the evaluation. The next page contains settings for the filters. Page four defines the feature vector. Page five contains general parameters and algorithm selection and the last page contains the settings for the classification.

This series reflects the way the algorithms are calculated in sequence. After the definitions of the raw data sets (reference and test data) the filters add new virtual sensors. Then the feature vector extracts the relevant data which is then fed into the core algorithms. The output of this algorithm is then used to assign the class to unknown samples.

3.2.2.1.1 Reference data

On the **Reference data** page you will enter those measurements that are to be included in the calculation of the prediction model. When you click the **Add** button, a dialog window will open which allows you to open several measurement data files simultaneously by holding down the SHIFT or CTRL key while clicking on the files one by one, or by drawing a marquee around several files. In this window you may also change the directory clicking on it or using the symbol status line.

Measurements can be removed altogether by highlighting the corresponding line and clicking **Delete**, or they can be changed from reference to test data by clicking **-> Test data**

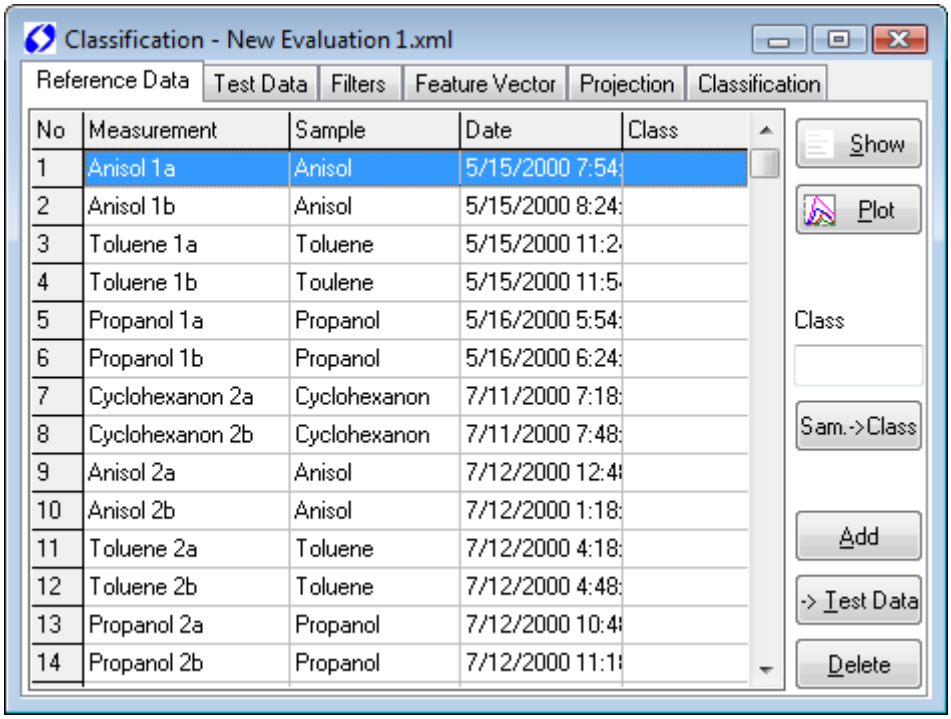
You may change the sequence of measurements by highlighting one line and then click-dragging it to a different place in the list. Most analysis algorithms are not affected by changes in the sequence; however it may help you with entering data or displaying suitable graphs. The sequence

can also be sorted automatically. Right clicking in the list brings up a menu which gives you a number of choices.

By clicking **Show**, windows with the measurement parameters for all selected measurements in the list are opened.

By clicking **Plot**, plots for all selected measurements in the list are opened. The same result can be achieved by double clicking a measurement's file name in the table.

If you want to be able to have newly added test data automatically attributed to a certain class of samples, you will need to first define classes of reference data. To do this, highlight all reference measurements belonging to a certain class, and fill in a descriptive class name in the **Class** field. Alternatively you can press the **Sam.->Class** button, which will copy the contents of the column Sample to the column Class.



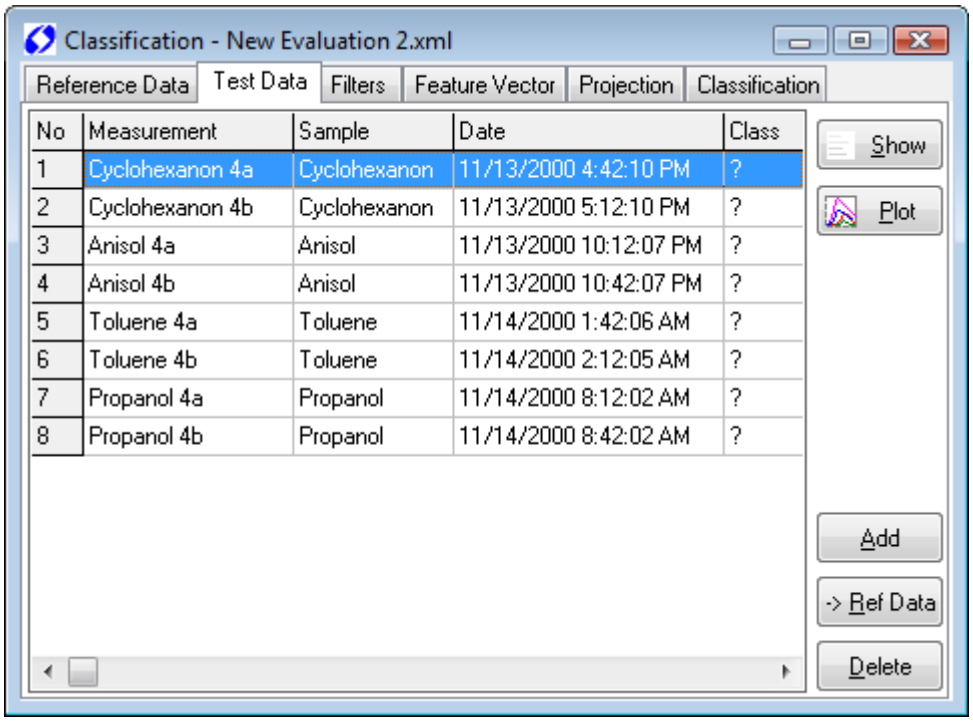
Column	Description
No	Number of the measurement.
Measurement	Name of the measurement.
Sample	Sample that was measured.
Date	Date and time when the measurement was taken.
Class	Class the measurement belongs to.

Tip: You can change the sequence of measurements by highlighting one line and then click-dragging it to a different place in the list.

Tip: Right clicking in the list will bring up a popup menu. In this menu you can sort the list by Measurement / Sample / Date or class. You can also copy the Measurement name or sample to the class. You can also add the Date to the Class, which will effectively split the class into separate classes for each day.

3.2.2.1.2 Test data

On the **Test data** page you can enter those measurements that you want to process according to the previously built model (in other words, adding more test data will not change an existing model). All buttons work in analogy to their Reference data counterparts. However, a class cannot be entered. If no classification has been selected on the **Classification** page (see below for details), or if no classes have been defined for the reference data, a question mark will appear in the Class column. Otherwise, the class determined by the classification algorithm will appear here.



Column	Description
No	Number of the measurement.
Measurement	Name of the measurement.
Sample	Sample that was measured.
Date	Date and time when the measurement was taken.
Class	Class the measurement belongs to.

3.2.2.1.3 Filters

Filters calculate a new response curve based on the response curve of sensors. Using filters is optional (the Basic edition of MultiSens Analyzer does not contain filters).

New filters can be introduced using the **Add** button. Once you have clicked **Add** you will first need to choose the device(s) from which you want to create new filters. Finally, select the desired type of filter. Depending on the filter type a window requiring extra user input may show up.

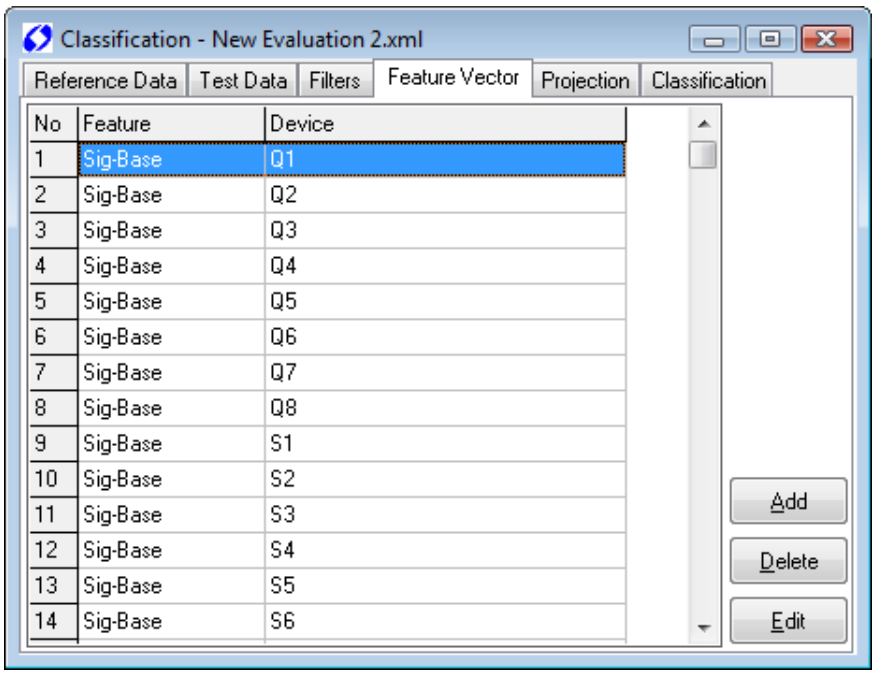
To remove a filter, highlight the respective line, and click **Delete.** The **Edit** button opens the window allowing the modification of filter parameters. In the case of a filter which has no user selectable parameters the Edit button has no function.
A list of filter algorithms is included in the appendix.

3.2.2.1.4 The feature vector

As discussed previously, the feature vector determines how to extract a small number of values from the relatively complex sensor response curves. There is a great deal of latitude in composing a feature vector. All or only some sensors may be used to build the feature vector; one sensor may be used multiple times to give its data more weight. Multiple features from one sensor can be used, too. The Basic version of MultiSens Analyzer provides a choice of features that mainly determine the amplitude of the response. More complex features such as signal rise time may give additional input for a subsequent evaluation and are available in the Standard and Professional version.
The individual components of the feature vector are shown in a tabular form on this page. Each line corresponds to one feature. The sequence of features is irrelevant. Each feature is shown with a number, the type of feature, and the name of the sensor whose response curve was used to generate the feature. The sensor's name includes the module to which the sensor belongs.
New features can be introduced using the **Add** button. Once you have clicked **Add** you will first need to select a module from the list of available modules and then highlight the sensor(s) to be used in the feature extraction. Finally, select the desired type of feature extraction. The following list shows the feature extraction types currently available in the Basic version of MultiSens Analyzer. Depending on the method a window requiring extra user input may show up.

Feature	Description
Sig-Base	Maximum positive or negative amplitude of the curve (signal) after the basline was subtracted. The 1st value after the start of the measurement serves as baseline value.
Sig-Base3	As Sig-Base but averaged over three values.
SigAt	Difference of the response curve at a certain point definable by the user to the baseline (at the start of the measurement).
SigAt-BaseAt	Difference of two different points of the response curve. Both points are definable by the user.

To remove a feature from the vector, highlight the respective line, and click **Delete.** The **Edit** button opens the window allowing the setting or modification of feature extraction parameters. In the case of a feature which has no user selectable parameters the Edit button has no function.



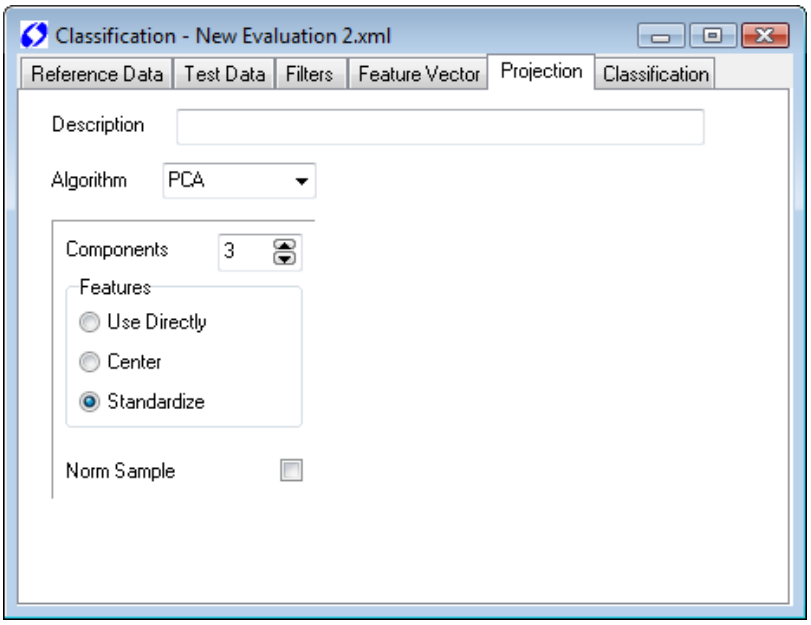
Column	Description
No	Number of the feature.
Feature	Extraction method of the feature.
Module	Module the evaluated sensor belongs to.
Sensor	Description of the sensor.

Tip: You can change the sequence of features by highlighting one feature line and then click-dragging it to a different place in the list. As mentioned before, such changes have no effect on the evaluation results.

3.2.2.1.5 The Projection parameters

On the page named **Projection** you can set up parameters that affect the projection onto the scores / discriminant functions.

In the **Description** field you can enter a short descriptive text about this evaluation. The text entered here may also be made to appear in the title or subtitle of a plot.



Depending on which version of MultiSens Analyzer you have installed, the parameter **Algorithm** lets you choose between different built in projection algorithms. The default algorithm is PCA (Principal Components Analysis).

The field **Components** defines how many principal components will be calculated. The maximum number here is the number of measurements on the Reference data page, or the number of features in the feature vector, whichever is smaller. The content of this field has some influence on the classification if scores are to be used as **Input values** (see below). If Linear Discriminant Analysis (LDA) is selected, the number of components is automatically determined and cannot be changed.

In the **Features** box, you can choose a preprocessing method for the features in the PCA. The following three options are implemented:

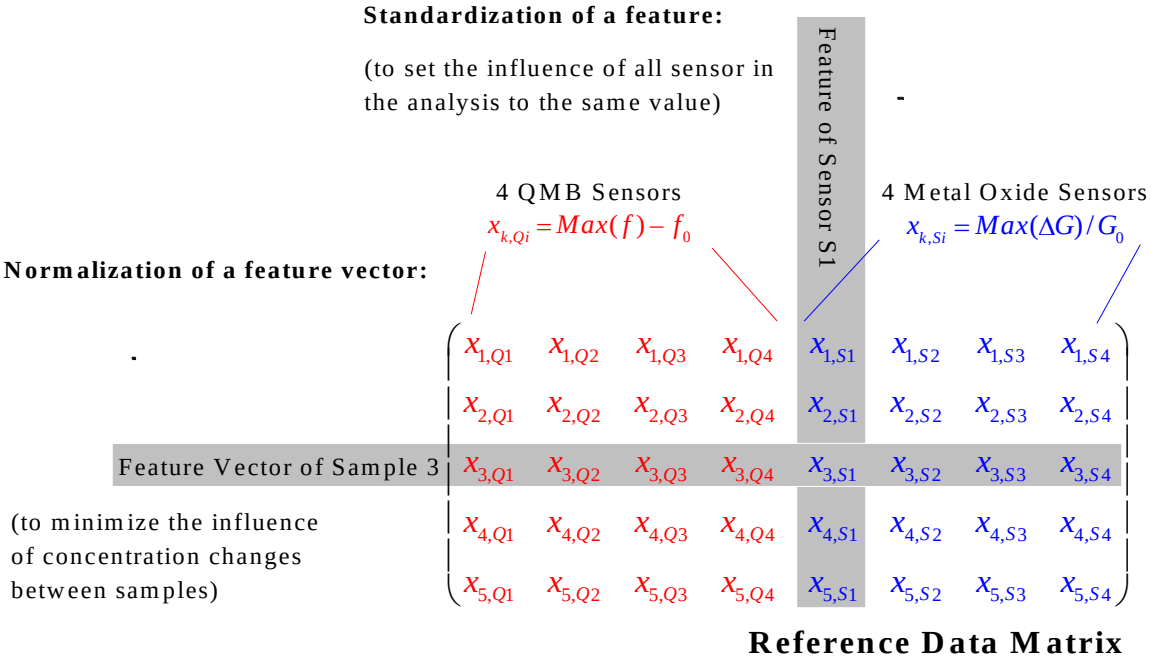
Use directly: The extracted features are not pre-processed in any way; they are used as-is.

Center: The mean value of the features in the reference data is subtracted from all features. This means that the principal components reflect only differences between features, and that the absolute feature value (which normally correlates with intensity) does not play a role. With uncentered data, the first principal component will in many cases contain mainly intensity information which is probably an unwanted property.

Standardize: With this point selected, features will be centered and additionally divided by the standard deviation of features in reference data. This means that all features will receive approximately equal weighting in the evaluation. Standardisation is recommended especially whtn different types of sensors are combined in an evaluation. Standardisation means that grossly different units (say, one sensor type in kilo-ohms, another type in milliampe'res) will cancel out, and not play a role in the evaluation.

There is a separate option **Norm samples** which can be applied in addition to any of the above. With this option activated, the length of each feature vector is forced to be one (1) without changing the relationship of features within this vector. This option is useful to minimise the influence of different intensities because normally, samples with higher concentrations will produce a longer feature vector. This simplistic explanation, however, assumes that only such features have been included whose magnitude correlates with intensity (such as Sig-Base). Additionally, this assumption is only valid for sensors showing a more or less linear response (which is not necessarily the case in real life).

The following sketch shows these relationships. We have a matrix of reference data and use x_1, to designate the i-th feature extracted from the data of sensor j. To keep the sketch intelligible we use a simplified example with four QMB and MOS sensors each and with five samples:

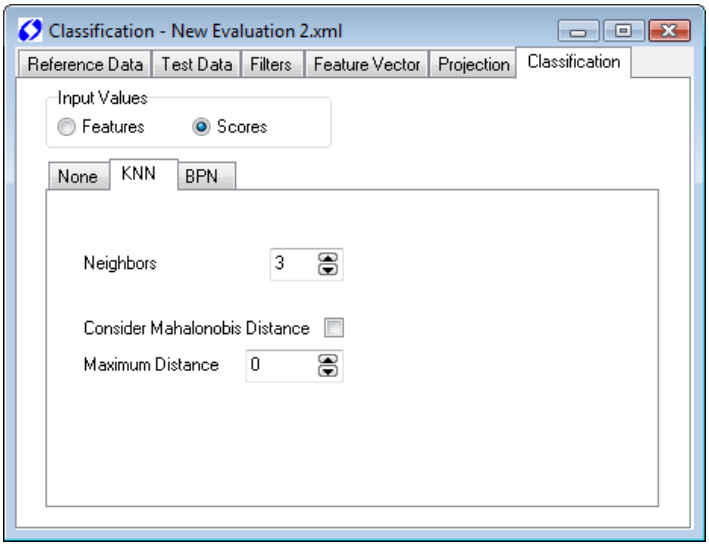


3.2.2.1.6 Classification parameters

The last page **Classification** contains all the settings that affect how the assignment of an unknown measurement to a certain class is achieved.

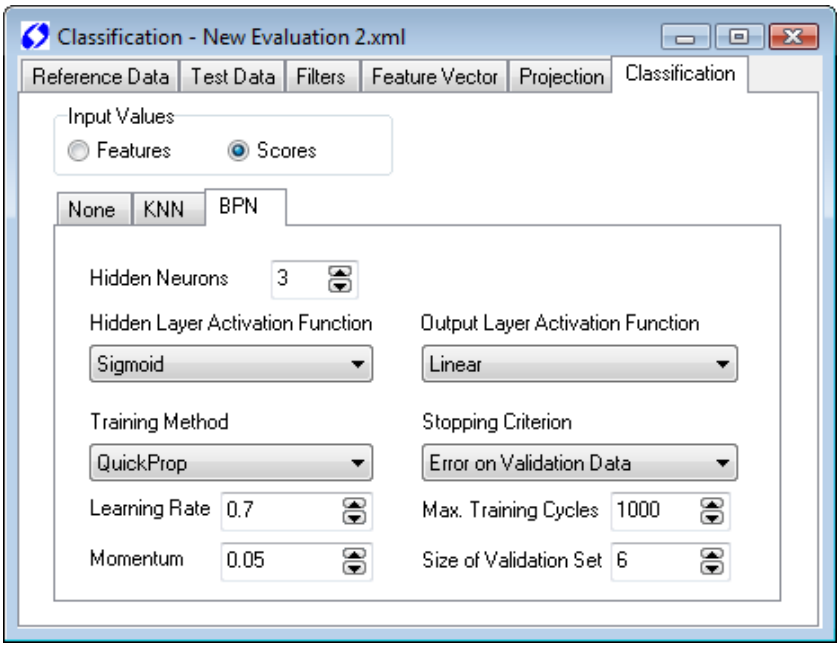
The first choice is to select the input values into the classification algorithm. In most cases Scores will be chosen, which will use the calculated scores from the PCA or LDA projection. Alternatively the **Features** may be used as input values. In this case the PCA or LDA projection does not affect the classification as the data used to determine the assignment is taken from the feature vector.

The next step is to choose the classification algorithm itself. By clicking on **None**, **KNN** or **BPN** the respective algorithm is activated and the parameters belonging to the algorithm are shown. (BPN is only available in the Professional version.)



If **KNN** is active the field **Neighbors** defines how many nearest neighbors are to be used. The class appearing most frequently in the number of neighbors defined here will be chosen. If two classes appear the same number of times (which is possible only if K is an even number), the class whose sum of euclidic distances is smallest will be chosen.

Activating **Consider Mahalanobis Distance** will compute the values for all classes and samples. One problem of the KNN algorithm is, that measurements will always be classified to on of the classes, even if the new measurement is far away from any of the classes in the reference data. Activate **Consider Mahalonobis Distance** and set **Maximum Distance** to a suitable value (typical values are between 2 and 5). In case the Mahalonobis distance exceeds the **Maximum Distance** to the class chosen by the KNN algorithm the measurement will be classified as unkown.



The **BPN** classification uses a back propagation neural network to classify measurements. This is a complex algorithm and should be used by advanced users with a good knowledge of artificial neural networks. A description of the BPN algorithm and its parameters is included in the appendix.

A good starting point is to set the number of **Hidden Neurons** to the number of different classes, set *Sigmoid* as **Hidden Layer Activation Function** and set *Linear* as **Output Layer Activation Function**.

Training Method should be set to *QuickProp* and with a **Learning Rate** of *0.7* and a **Momentum** of *0.05*.

Stopping Criterion should be set to *Error on Validation Data*. Max. **Training Cycles** should be set to a sufficiently high value (*200* to *1000*, depending on complexity of data set).

The Size **of the Validation Set** should be chosen to include at least 2 repetitions per class, or 20 to 50% of the Reference Data. The Validation Set is taken from the end of the Reference Data. You can move reference measurements on the Reference Data Page to the end to be included in the validation data.

Validation Data is not used during training! The Validation Set will therefore reduce the amount of measurements that are used during training.

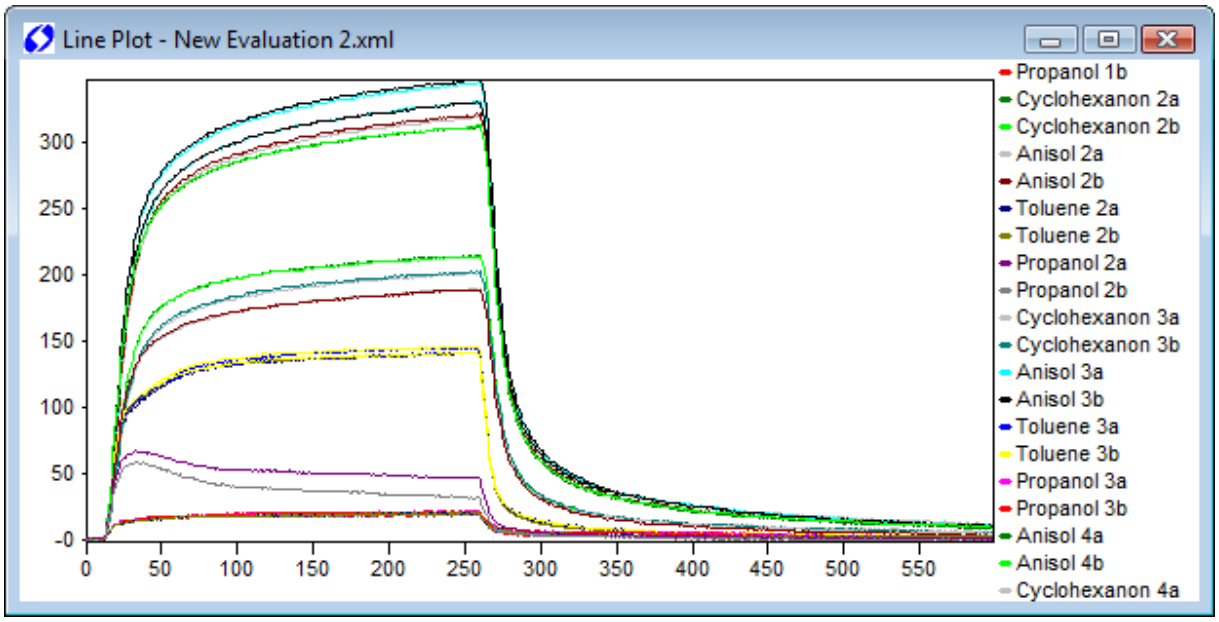
The BPN Training Error Plot shows how the prediction error onto the training data and the validation data develops during the training. You can use this plot to optimize the parameters for the BPN. MultiSens Analyzer does recalculate the complete BPN whenever you change a parameter. This allows to interactively change parameters and watch the results.

The BPN algorithm initializes the training with random values for the weights in the neural network. This may lead to different results whenever the training is performed, even if the parameters are unchanged.

Box	Description
Description	Description of the evaluation (can be displayed as title or subtitle in the graph).
Components	Number of calculated principal components.
Features	Weighting of the features: **Use directly**: Without conversion. **Center**: Subtraction of the average from the reference data. **Standardize**: Division of the reference data by the standard deviation additionally to centering.
Norm samples	All feature vectors are recalculated to the length of one.
Mahalonobis distance	Activate the calculation of the Mahalonobis distance.
KNN-classification	Activate the automatic classification using the k-nearest-neighbor (KNN)-algorithm.
Neighbors	Number of nearest neighbors that should be considered for the classification.
Input values	Values, used for the calculation of the KNN-classification and the Mahalonobis distance: **Features**: The extracted features (after weighting and/or normalization) are used. **Scores**: The input values of the principal component analysis are used.

3.2.2.2 The line plot

The line plot is used to display the response curves of a single sensor within all measurements. It allows to check if a sensor produces consistent results in all measurements. The sensor can be selected from a drop down box in the icon bar.



By activating **Graph – Offset subtraction** you can force all sensor curves to start at a Y value of zero. The first value of the measurement is considered the offset, and will be subtracted from all subsequent values. Depending on the type of module, offset subtraction may or may not be activated by default. In either case you can always choose manually.

By choosing **Graph – Legend** you can enable or disable the legend (list of sensor names and curve colors) on the right. By default, this is enabled.

Axes are normally scaled automatically so that all values fit in the graph frame. If you want to zoom in, simply draw a frame around your area of interest with the left mouse button depressed. To return to full-size (unzoomed) view, double click the graph window, or click on the crossed-out magnifier icon in the icon bar, or choose **Graph – Zoom off** from the menu. This menu point is only enabled when you are currently looking at a zoomed graph.

The measurements can be limited to those belonging to a certain class by choosing **Graph – Classes – *classname*** in the menu or the down box in the icon bar. **Graph – Classes – Group** will add an y-offset depending on the class of the measurement. So that different classes are grouped along the y-axis.

For easier filing and finding diagrams, top and bottom titles may be added to a plot below and/or above the diagram (**Graph – Title – Title above…**) and/or (**Graph – Title – Title below…**). Once either of these two menu points has been selected, a dialog window will pop up which will prompt you for the title text, its alignment (left, right, centered), and, optionally, font and size. If you have made your choice you have an option to make it permanent by clicking **As Standard** which will make these settings appear automatically as the default setting in the future. By clicking **Standard**, the current default settings are applied to the current title. It is possible to use macros

in the tide text, creating variable content that is updated automatically from data entered elsewhere in the evaluation. The following table gives details:

Macro	Function	Description
{Description}	Description	Shows the information of the description box.
{Filename}	Filename	Shows the filename and the path.
{Now}	Now	Shows the current time.
{ProgramVersion}	Software version	Shows the name and the version of the program.
{Today}	Today	Shows the current date.
{User}	User	Shows the name of the user (the login under Windows).

*Tip: A graph can be transferred to the clipboard by **Edit – Copy**. This allows to „copy and paste" graphs into different applications (e.g. Word-processors)*

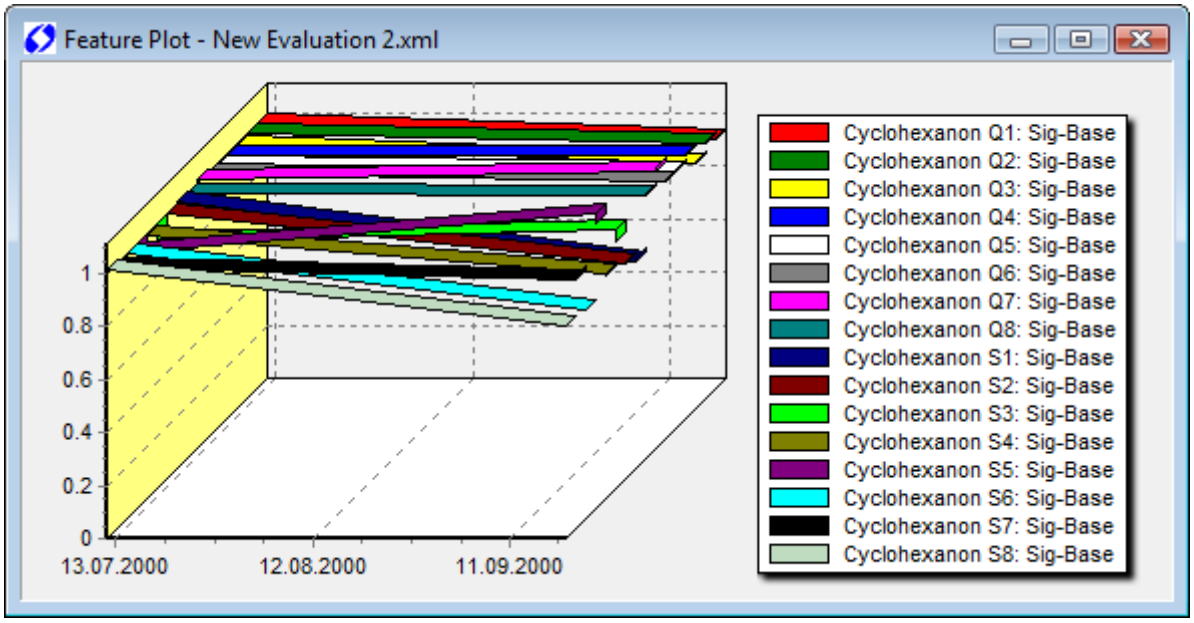
3.2.2.3 The measurement plot

The measurement plot is very similar to the line plot. The difference is that the measurement plot shows all (or selected) sensor responses from one measurement, while the line plot shows the responses of one from all (or selected) measurements.

You can open a measurement plot for a certain measurement by double clicking on the measurement in the reference data or test data list. Alternatively clicking on the Plot button on the reference data or test data page will open measurement plots for all selected measurements.

3.2.2.4 The feature plot

The feature plot displays the feature values that have been extracted from the response curves. It has a number of options that allow present the information in optimal ways for identifying outliers, drift and other effects.



Graph – Normalize is active by default. This uses the values extracted from the first measurement of a series as quotients for the respective features. The same feature in all other measurements of the same series is divided by this quotient. The resulting graph thus shows relative changes to this value. If the feature is constant within this series, the graph will show a flat line with a value of one.

Graph – Classes – classname will only show measurements of the same class. If you combine this with **Graph Normalize**, you will easily see if a feature has reproducible data and spot outliers easily. **Graph – Classes – Group** will plot individual lines for all classes and thus allow examining all classes simultaneously.

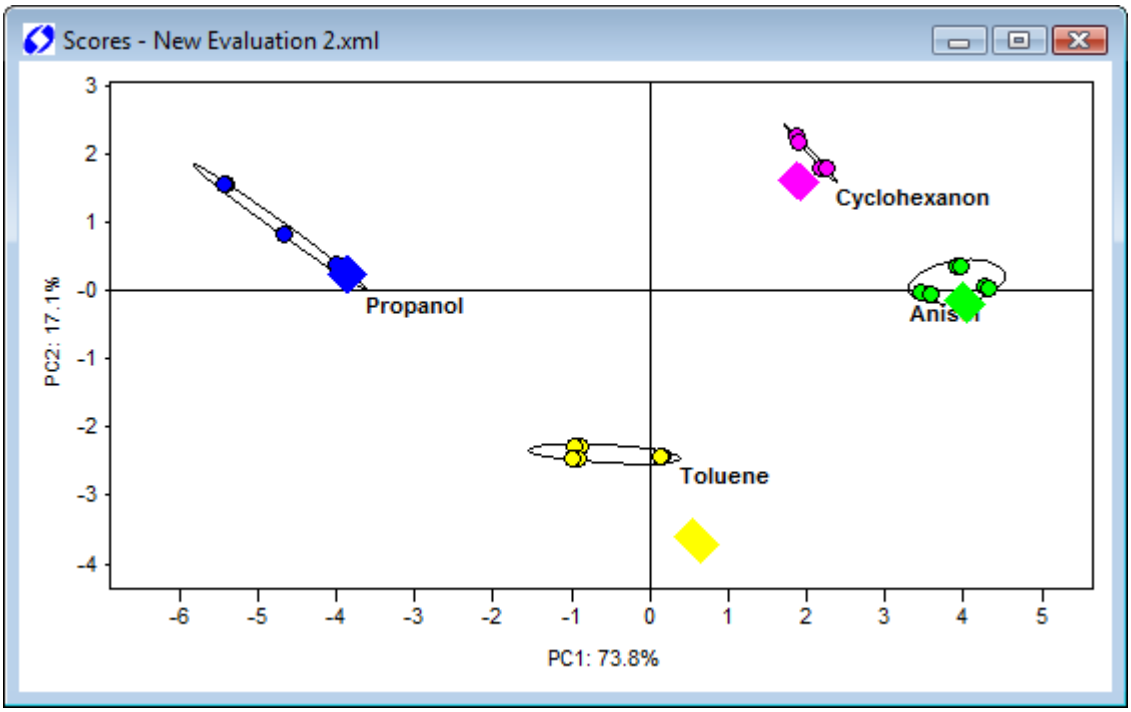
Graph – Show Time is also active by default and will plot the measurements along the time on the x-axis. This allows identifying drift in features over time. If Graph – Show Time is disabled, the sequence of the measurements is determined by the reference / test data sequence.

Graph – Features allows selecting a subset of features to be shown in the graph. For clarity it is often best to show only one single feature in the graph.

Graph – Show 3D activates a quasi 3D look and **Graph – Show Legend** controls the display of the legend.

3.2.2.5 The scores plot

The scores plot is the most used plot in a classification and shows a plane spanned by two principal components. The default configuration is to show the first principal component as the X axis (horizontal) and the second principal component as the Y axis (vertical). Since the first two principal components contain the largest amount of variance in the data, this diagram is a very intuitive measure of how different or similar the samples were "seen" by the sensors. The distance between points is directly proportional to the difference in features. The larger the distance between two samples in the plot the larger is the difference of the samples. The amount of variance represented by the respective principal component is shown in the axis title in the format PCx = zz.z % where X is the index of the principal component (normally 1 or 2).



The scores plot is an ideal tool for quickly grasping an overview about different samples. The score of a measurement with respect to a principal component is simply the value of the measurement on the corresponding axis. There are many options available for customizing the scores plot display.

Axes are automatically scaled so that all measurements are show in the diagram. For zooming into a certain range in the diagram, simply click-drag a rectangle around that range. By double clicking the graph or selecting **Graph - Zoom 0ff** you can revert to the unzoomed display.

By activating **Graph - Symmetric axes** it is possible to adapt scaling so that the origin of X and Y axes lies in the diagram center. If values have a strongly asymmetrical distribution around the origin, this may lead to a great deal of wasted display space.

By activating **Graph - Golden cut**, a special display mode is enforced. The ratio of X to Y axis length is forced to be the square root of two (this is the ratio used in the DIN A paper size system, and does NOT correspond to the mathematical definition of golden cut, but the initial catch phrase "golden cut" turned out to be sticky...). Additionally, X and Y axis ticks and labels are removed and replaced by only the ratio of the ranges displayed on the X and Y axes (PC x : PC y = z.zzz). This display mode may be useful for comparing the results of different evaluations with one another.

Graph – BiPlot will plot the loadings of the features in the scores plot. Features are identified by squares and have a line to the origin of plot. This plot helps to find features that carry information for certain classes.
Graph – Time Path draws lines connecting measurements within one class made on different days. If several measurements were done on the same day, the line will go to the center between those measurements. This plot is ideal to identify long term drift.
By activating **Graph - Equal scale**, scalings are adjusted so that unit lengths along X and Y axes are identical, i.e. 1cm along the X axis corresponds to the same number of units as 1 cm along the Y axis.

The Symmetric axes, Golden cut, and Equal Scale settings may also be combined arbitrarily.
The menu points **Graph - X-axis** and **Graph - Y-axis**, respectively, allow you to define which principal component will be shown along which axis. This way it is also possible to use different combinations than the default (PC 1: X PC 2: Y). You may, for example, choose to display PC 3 vs PC 2.

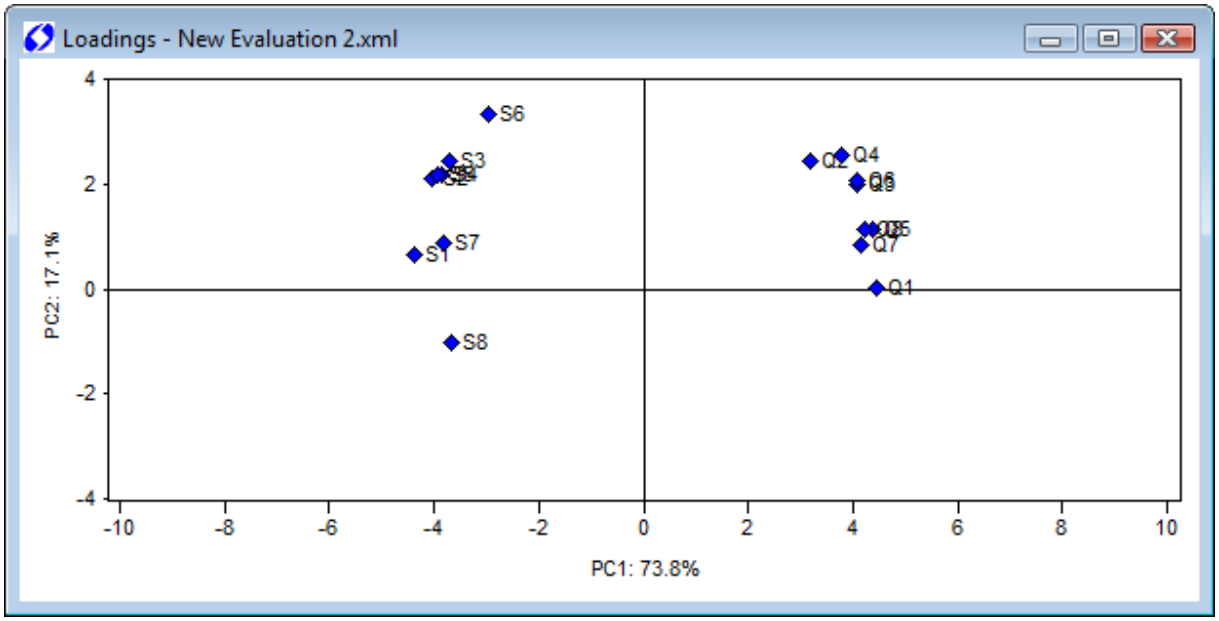
If you have defined classes for your reference data, classes can be emphasized in the plot by ellipses drawn around samples belonging to the same class **Graph - Ellipsis**. There are several options here. First, ellipses can be calculated so that their principal axes correspond to three times the standard deviation of scores with respect to the corresponding principal component, independently for X and Y axes **Graph – Ellipsis Independent**. Alternatively, ellipses can be calculated so that the correlation between the principal components within the class is taken into account **Graph – Ellipsis Correlated**. In this case, the principal axes correspond directly to standard deviation (not three times as in the previous case). N.B.: These two options are not mutually exclusive! If you activate both of them you will get two different ellipses for each class. It is also possible to have the class name displayed next to the ellipse **Graph – Ellipsis Caption**, and to display the ellipse axes **Graph – Ellipsis Draw axes**.

Individual measurements can be displayed in different ways. Classes of reference data can be distinguished by different colours (**Graph - Symbols – Mark classes by color**) or by different shape (**Graph - Symbols – Mark classes by shape**). This is applicable for Test data as well. (**Graph - Symbols – Mark Test data by color**, **Graph – Symbols – Mark Test data by size**). Reference data are represented by circles, Test data are represented by diamonds.
Labels may be chosen (**Graph - Label**) from the number of the measurement in the table of reference or test data (**Graph - Label - Number**), the filename of the corresponding

measurement (**Graph - Label - Filename**), the description of the sample (**Graph - Label - Sample**) and the name of the class to which the measurement belongs (**Graph - Label - Class**).

3.2.2.6 The loadings plot

The loadings plot shows to which extent individual features contribute to the calculation of principal components. The further "inside" (close to the origin) a feature lays, the smaller its influence; the more distant from the origin, the more important. The azimuthal distribution of features gives hints about the data set's redundancy. Features providing very similar information are found very close to each other azimuthally (under the same angle). Features providing vastly different information will be located under clearly separated angles.



The same axis scaling options as for the scores plot (**Graph - Symmetric axes**, **Graph - Equal scale**, **Graph - Zoom off**)are applicable here.

Additionally, loadings can be weighted with their corresponding eigenvalues (**Graph - Weight with Eigenvalues**).

Again, options are available for selecting which principal component to display on which axis (**Graph - X-axis** and **Graph - Y-axis**).

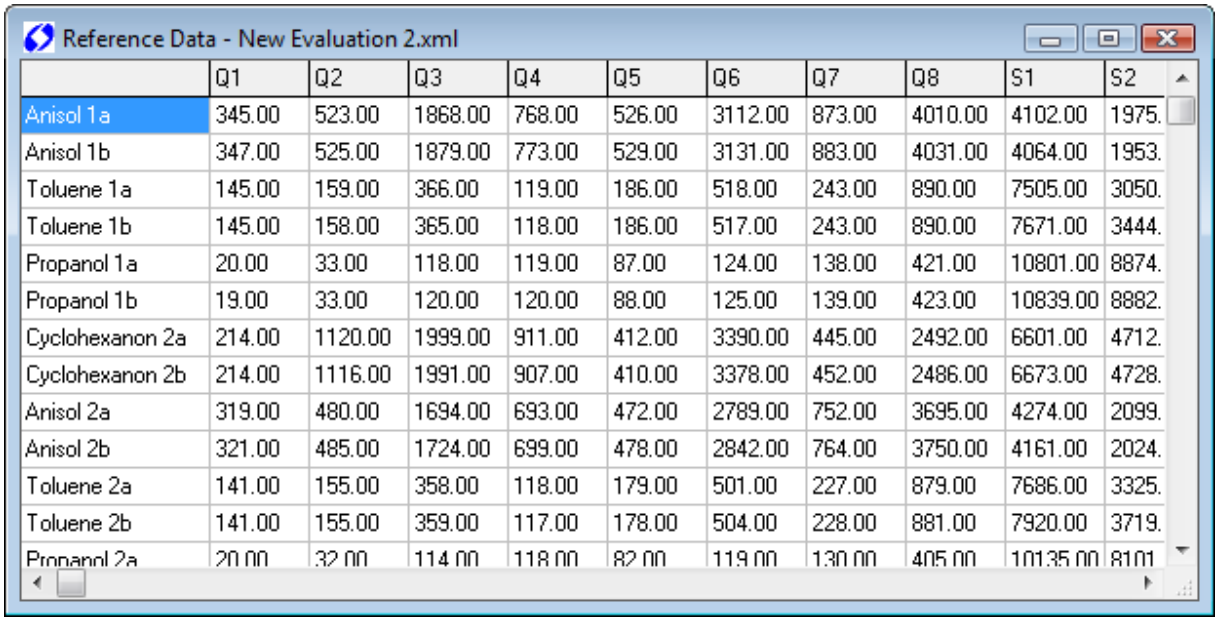
Again, there is the possibility of using symbols and/or captions for the features. If symbols are chosen (**Graph - Symbols activated**), all features are shown as diamonds.

Labeloptions available are the number of the feature within the feature vector (**Graph – Label - Number**) , the name of the sensor used for the extraction of this feature (**Graph – Label - Sensor**), and a comprehensive caption including module, sensor, and feature extraction method names (**Graph – Label - Sensor+Method**).

Title options are identical to those available with the scores plot. The same macros are applicable.

3.2.2.7 Output of numerical values

While it is very convenient and intuitive to view the results graphically, sometimes it may be useful to have access to processing results in tabular or matrix form. MultiSens Analyzer offers that possibility too.



	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	S1	S2
Anisol 1a	345.00	523.00	1868.00	768.00	526.00	3112.00	873.00	4010.00	4102.00	1975.
Anisol 1b	347.00	525.00	1879.00	773.00	529.00	3131.00	883.00	4031.00	4064.00	1953.
Toluene 1a	145.00	159.00	366.00	119.00	186.00	518.00	243.00	890.00	7505.00	3050.
Toluene 1b	145.00	158.00	365.00	118.00	186.00	517.00	243.00	890.00	7671.00	3444.
Propanol 1a	20.00	33.00	118.00	119.00	87.00	124.00	138.00	421.00	10801.00	8874.
Propanol 1b	19.00	33.00	120.00	120.00	88.00	125.00	139.00	423.00	10839.00	8882.
Cyclohexanon 2a	214.00	1120.00	1999.00	911.00	412.00	3390.00	445.00	2492.00	6601.00	4712.
Cyclohexanon 2b	214.00	1116.00	1991.00	907.00	410.00	3378.00	452.00	2486.00	6673.00	4728.
Anisol 2a	319.00	480.00	1694.00	693.00	472.00	2789.00	752.00	3695.00	4274.00	2099.
Anisol 2b	321.00	485.00	1724.00	699.00	478.00	2842.00	764.00	3750.00	4161.00	2024.
Toluene 2a	141.00	155.00	358.00	118.00	179.00	501.00	227.00	879.00	7686.00	3325.
Toluene 2b	141.00	155.00	359.00	117.00	178.00	504.00	228.00	881.00	7920.00	3719.
Propanol 2a	20.00	32.00	114.00	118.00	82.00	119.00	130.00	405.00	10135.00	8101

The menu **Evaluation – Matrix** accesses the following menu points

Features Shows values of the features extracted from the reference measurements (**Reference data**) or the test data (**Test data**).

Scores Shows the calculated scores of the principal components of the **Reference data** or **Test data**.

Loadings Shows the loadings of all features.

Variances Shows the information content as percentage for each principal component.

Mahalonobis distance Shows the Mahalonobis distance between the **Classes**, or between the **Reference data** and the classes, or between the **Test data** and the classes.

Once a matrix is displayed, the menu point **Matrix - Transpose** lets you swap rows and columns.

For the reference data, test data, and scores matrices, **Matrix - Sample description** lets you choose between the number of this feature (**Number**), the measurement's file name (**File name**), the contents of the samp1e field in the measurement (**Samp1e**), or the class attributed to this measurement (**class**).

For the reference data, test data, and loadings matrices, feature labels can be chosen with **Matrix - Feature Label** from the following choices: the number of the feature in the feature vector (**Number**), the name of the sensor used for the extraction of this feature (**Sensor**), and a comprehensive label including module, sensor, and feature extraction method names (**Sensor+Method**).

For the matrix of loadings, you can additionally enable weighting with eigenvalues (**Matrix - Weight with eigenvalues**).

3.2.2.7.1 Export of results

For exporting results to other applications, you can generally use the familiar mechanism of **Edit - Copy** and **Edit - Paste**. For exporting PCA results (scores and loadings), more elaborate possibilities exist. You can choose to copy the entire matrix including all values and all labels (**Edit - Copy all**), or you first highlight the values you wish to copy. Then you have a choice of either copying just the numeric values with **Edit - Copy**, or you use **Edit - Copy with label** to include label information.

The clipboard contains all data as text, the delimiter is the tab-symbol. The decimal delimiter is a dot. This is independent of settings in the control panel of Windows™.

Hint: Before pasting data from the clipboard to an application program it might be necessary to adapt the settings in the control panel. The decimal delimiter must be set to a dot. After adjustment you might have to re-start the application program in which you want to paste the data.

3.2.3 The Regression

Regression is available in MultiSens Anaylzer Standard and Professional.

In a regression quantitative values are predicted for unknown measurement data. The correlation between the measurement data and the target values are determined by building a mathematical model from the reference data. The user has to fill in target values for every measurement of the reference data set.

MultiSens Analyzer offers Principal Component Regression (PCR) and Partial Least Squares (PLS) as algorithms for the regression.

Both algorithms calculate principal components (scores); thus scores plot and loadings plot can be plotted.

To check the quality of the model, the prediction values are also calculated for the reference data. On the true/predicted plot the true values (as defined by the user) are plotted against the predicted values (by the model). This graph visualizes the prediction accuracy of the model. As there are no true values for the test data, only reference data are shown on this graph.

In the results plot the predicted values of all samples (within the reference and test data) are shown as bar graph.

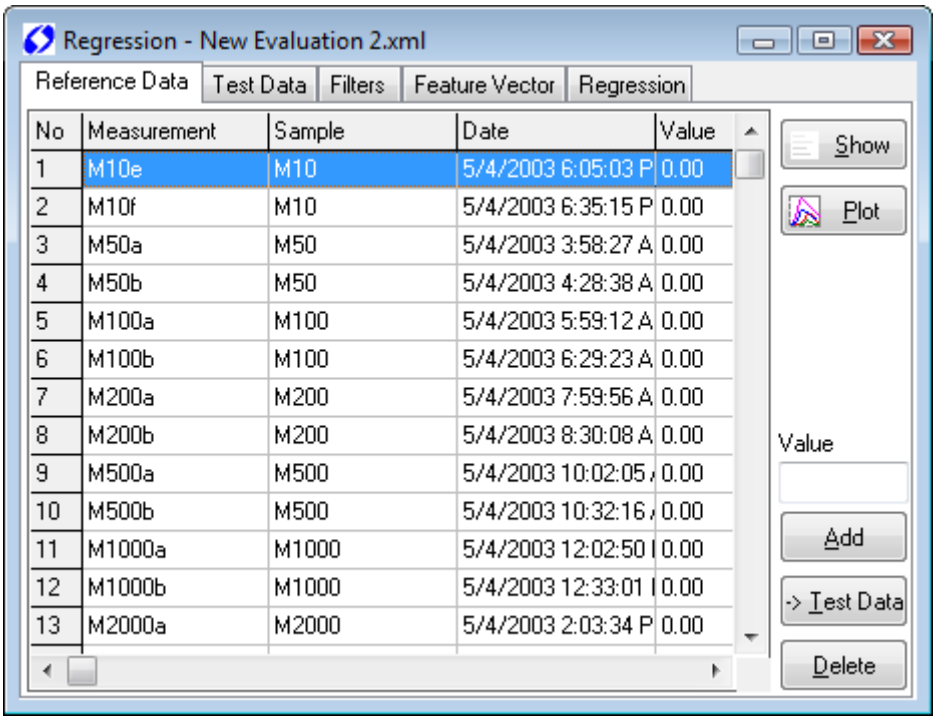
The prediction accuracy of the model within the reference data will increase the more components (scores) are used in the calculation. The prediction accuracy within the test data however, will decrease in general if the number of components is increased beyond a certain number. This paradoxical behavior results from the fact that in the creation of the model the true values of the reference data are known. The more components in the regression are used, the higher the freedom of the model will be to adopt to these known values. As also the reference data will have a certain amount of measurement error, the influence of this error increases (in most cases) with the number of components. If such an overfitted model is used to predict test data the random measurement error of the reference data may lead to totally wrong results. While setting up an evaluation one should always check the model by inserting samples not included in the reference data set into the test data set where true values are known. The error in the prediction these test data should then be checked critically and compared to the error on reference data (validation).

3.2.3.1 The parameters

As for the classification, regression parameters are collectively displayed on several pages within one window. The first two page contain lists of reference and test data, the next pages contains settings for filters and the feature vector and the last page contains general and regression parameters.

3.2.3.1.1 Reference data

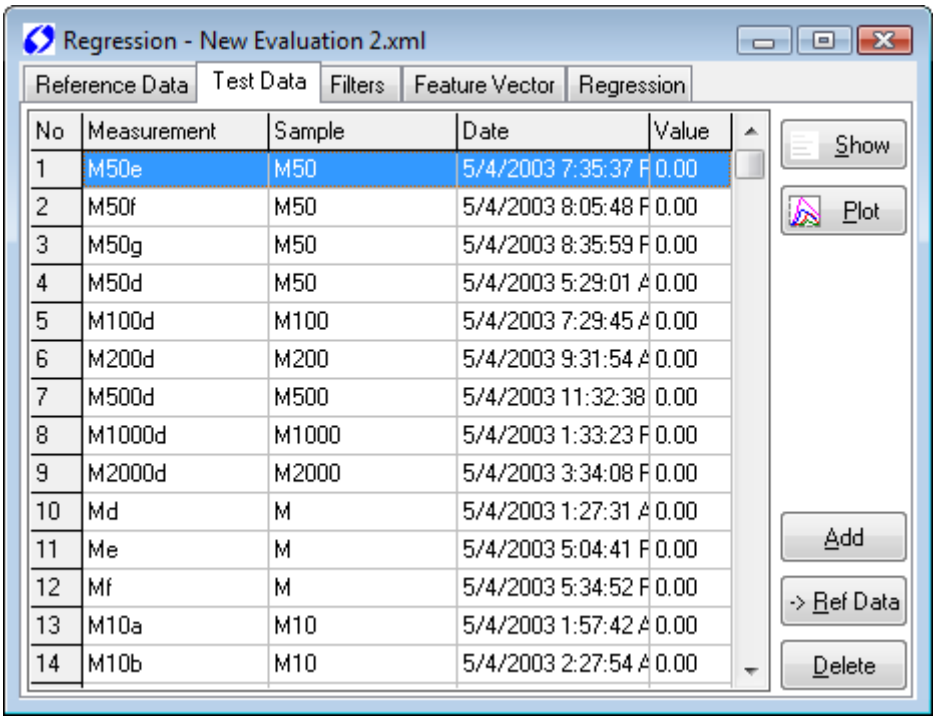
As described in 3.2.2.1.1 on the page reference data the reference data set is composed.
Instead of a class a value is assigned to each individual member of the reference data set. The following table explains the different columns on this page.



Column	Description
No	Number of the measurement.
Measurement	Name of the measurement.
Sample	Sample that was measured.
Date	Date and time when the measurement was taken.
Value	(True) value that should be related to the measurement.

3.2.3.1.2 Test data

On the page test data the measurements are defined, that you want to evaluate on the bases of the model calculated from the reference data set. As the values of the test data are calculated automatically, it is not possible to input any data in the column Value.



Column	Description
No	Number of the measurement.
File	Filename of the measurement.
Sample	Information from the sample box of the measurement.
Value	(predicted) value, calculated for this measurement.

3.2.3.1.3 Filters and Feature Vector

The functionality of filters and the feature vector is identical to the ones in classification described in the previous chapter.

3.2.3.1.4 The projection parameters

On the page named **Regression** you can select the regression algorithm and corresponding parameters.

In the **Description** field you can enter a short descriptive text about this evaluation. The text entered here may also be made to appear in the title or subtitle of a plot.

Depending on which version of MultiSens Analyzer you have installed, the parameter **Algorithm** lets you choose between different built in projection algorithms. The default algorithm is PCR (Principal Components Regression).

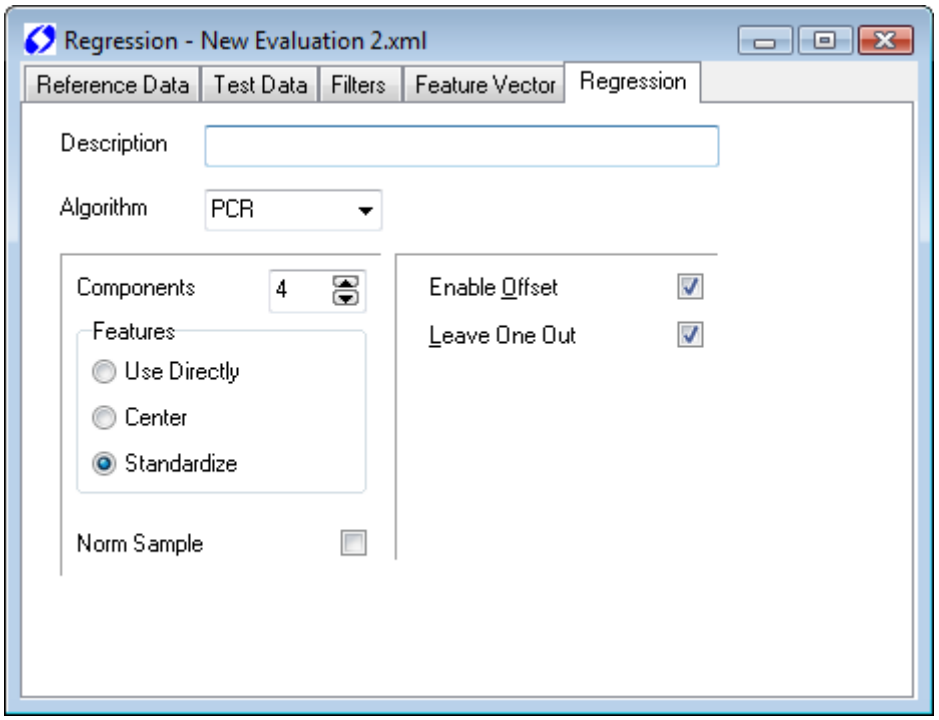
The field **Components** defines how many principal components will be calculated. The maximum number here is the number of measurements on the Reference data page, or the number of features in the feature vector, whichever is smaller.

The contents of this field will also define how many input values are used for the regression. As discussed in 3.2.3 the choice of number of components used in the PCR is a critical parameter that should be chosen with great care.

In the **Features** box you can choose a preprocessing method for the features in the PCR. This parameter and the parameter **Norm samples** have been discussed in great detail in chapter 3.2.2.1.5.

The Option **Enable offset** relates to the calculation of the multilinear regression of the scores onto the result values and in practice should be enabled always. In some rare cases (e.g. when the features are used directly without centering or standardization and a proportional relation without any offset is assumed between the features and the target values) it might be appropriate to disable this option.

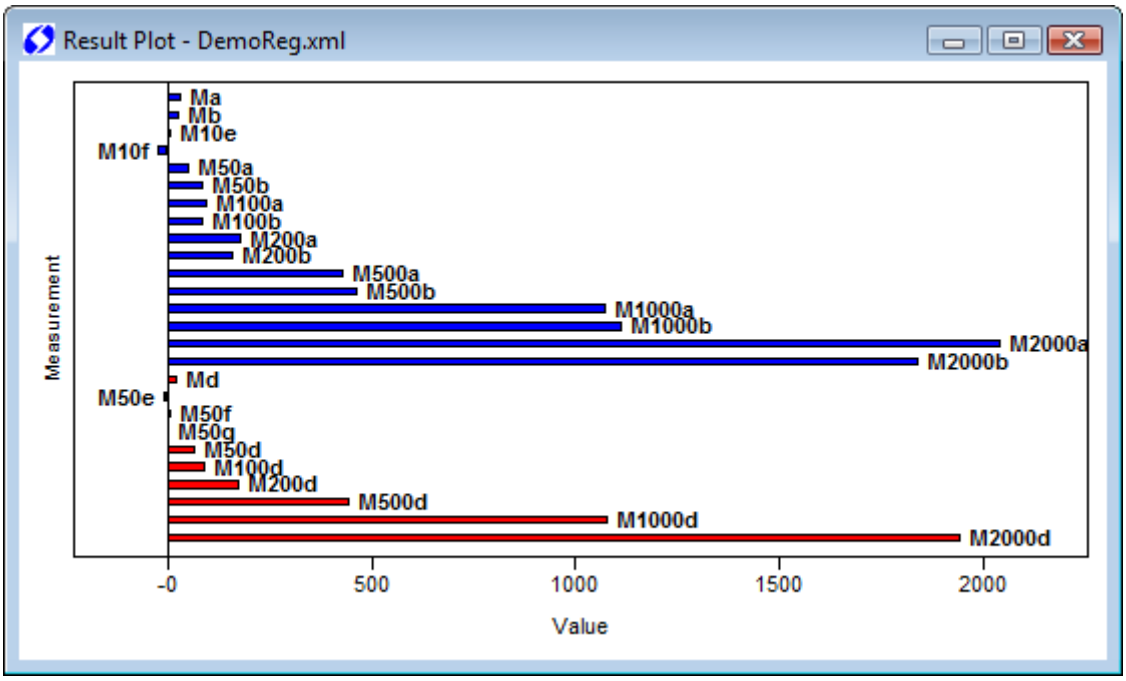
Leave One Out only affects how the predicted values for the reference data are calculated. If **Leave One Out** is active, individual regression models will be calculated for each reference data measurement. Only the other reference data measurements will be used for this calculation. The predicted value for the reference measurement under evaluation will then be calculated from this model. The reference measurement is thus left out during model building for calculating its predicted value.



Box	Description
Description	Description of the evaluation (can be displayed as title or subtitle in the graph).
Algorithm	Selects the algorithm used for the classification.
Components	Number of principal components calculated and considered for the regression.
Features	Weighting of the features: **Use directly**: Without conversion. **Center**: Subtraction of the average from the reference data. **Standardize**: Division of the reference data by the standard deviation additionally to centering.
Norm Samples	All feature vectors are recalculated to the length of one.
Offset Enable	Offset enabled for the calculation of the regression coefficient.
Leave One Out	Use Leave One Out during the calculation of predicted values for reference data

3.2.3.2 The bar chart

The result plot visualizes the results of the regression as horizontal bar chart. The single measurements are represented as bars. Positive values wil be shown as bars extending to the right, negative values will extend to the left. Using the menu option **Graph –Zero** the origin can be included in the horizontal range. Using **Graph - Draw** bars can be removed totally (**Only text**) or can be displayed as **Bars** or **Points**. With **Graph – Label** the text that is displayed for each sample can be chosen (**None**, **Number**, **File name**, **Sample**, **Value**). With **Graph – Title** the titles above (**Title above**) and below (**Title below)** van be defined. Like in any graph macros can be used within the title text.



3.2.3.3 The true/predicted plot

In order to visualize the prediction accuracy of the model, on this graph the predicted values (calculated by the model) are plotted against the true values (as defined by the user).

Each sample of the reference data is represented by a point, whose coordinates on the X-axis are true values and on the Y-Axis are the predicted values. If all samples are represented on a straight line that links zero and the point (1/1) then the reference data is predicted with no error. As mentioned in 3.2.3 for judging the quality of the model one has to check also the error made in the prediction of known test data, that is not included in the reference data set (validation).

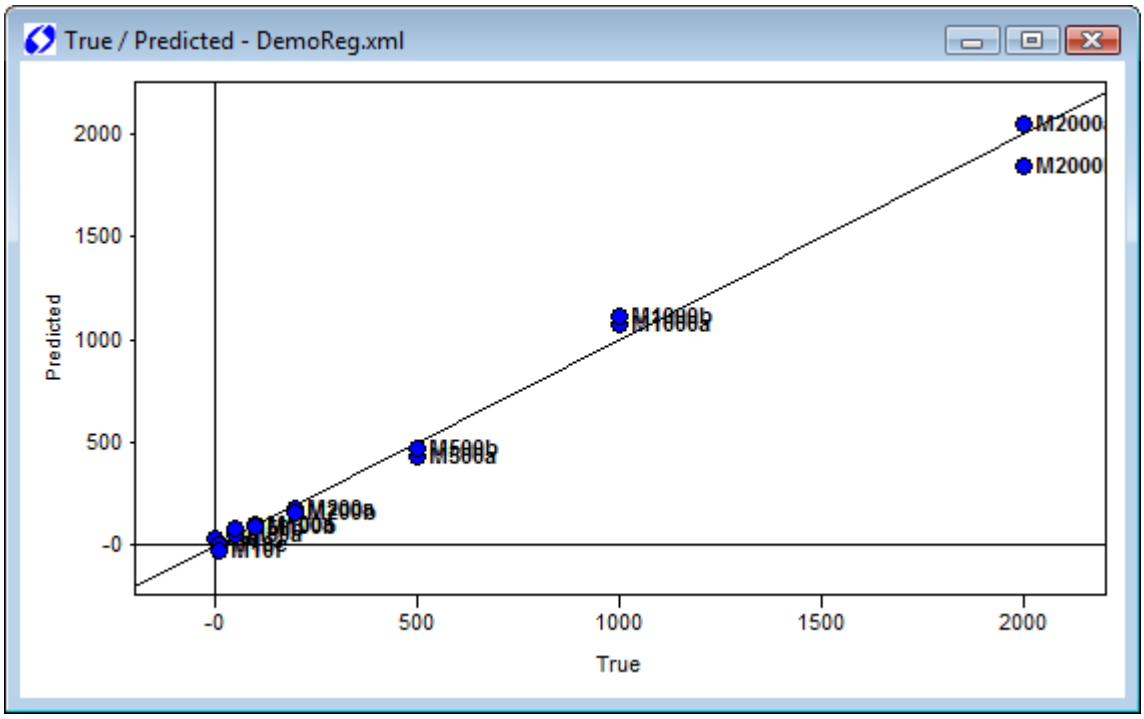
By default the axes will be scaled, so that the origin of the axes will be on the graph. If **Graph – Zero** is deactivated the axes will be scaled in order to show all samples in the diagram without wasting any space.

By activating **Graph - Symmetric axes** it is possible to adapt scaling so that the origin of X and Y axes lies in the diagram center. By activating **Graph - Equal scale**, scalings are adjusted so that unit lengths along X and Y axes are identical, i.e. 1cm along the X axis corresponds to the same number of units as 1 cm along the Y axis. The line representing correctly predicted reference data will then have an angle of 45 degree.

The settings ‚**Zero'**, , **Symmetric axes**' and , **Equal scale**' can be combined.

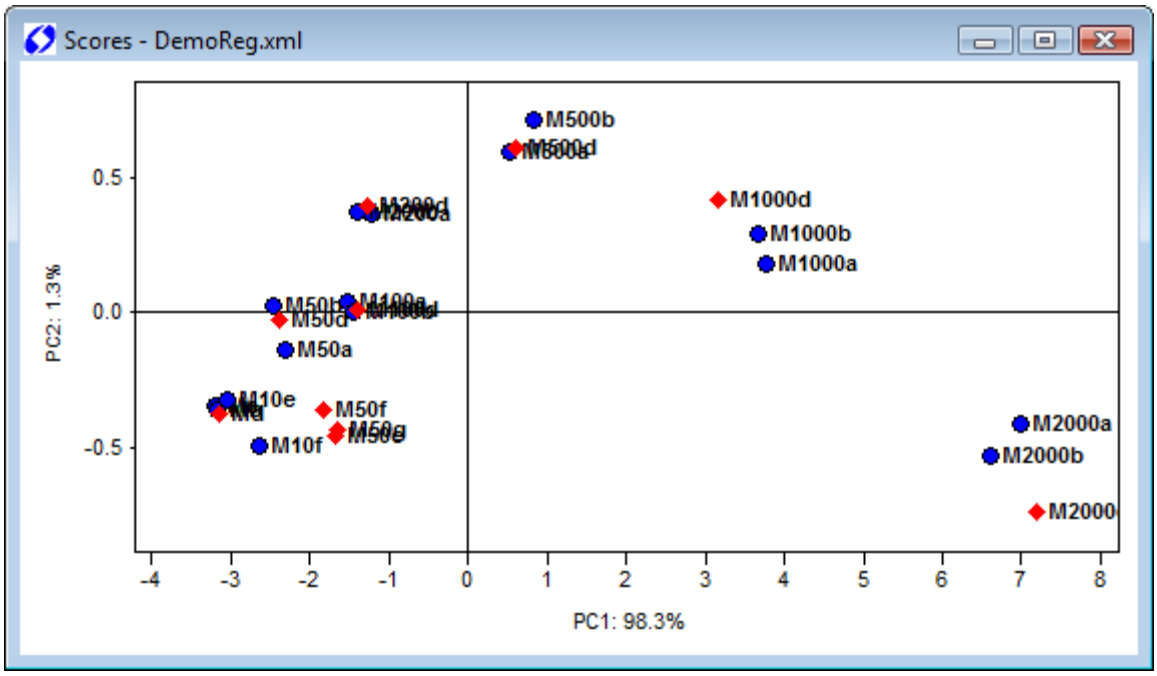
Labels may be chosen (**Graph - Label**) from the number of the measurement in the table of reference data (**Graph - Label - Number**), the filename of the corresponding measurement (**Graph - Label - Filename**), the description of the sample (**Graph – Label - Sample**) and the true value assigned by the user (**Graph - Label - Value**).

For documentation purposes titles can be assigned like in any other graph (**Graph - Title**).



3.2.3.4 *The scores plot*

The scores plot shows a plane spanned by two principal components. The default configuration is to show the first principal component as the X axis (horizontal) and the second principal component as the Y axis (vertical). Since the first two principal components contain the largest amount of variance in the data, this diagram is a very intuitive measure of how different or similar the samples were "seen" by the sensors. The distance between points is directly proportional to the difference in features. The larger the distance between two samples in the plot the larger is the difference of the samples. The amount of variance represented by the respective principal component is shown in the axis title in the format PCx = zz.z % where X is the index of the principal component (normally 1 or 2).



The scores plot is an ideal tool for quickly grasping an overview about different samples. The score of a measurement with respect to a principal component is simply the value of the measurement on the corresponding axis. There are many options available for customizing the scores plot display.

Axes are automatically scaled so that all measurements are show in the diagram. For zooming into a certain range in the diagram, simply click-drag a rectangle around that range. By double clicking the graph or selecting **Graph - Zoom off** you can revert to the unzoomed display.

By activating **Graph - Symmetric axes** it is possible to adapt scaling so that the origin of X and Y axes lies in the diagram center. If values have a strongly asymmetrical distribution around the origin, this may lead to a great deal of wasted display space.

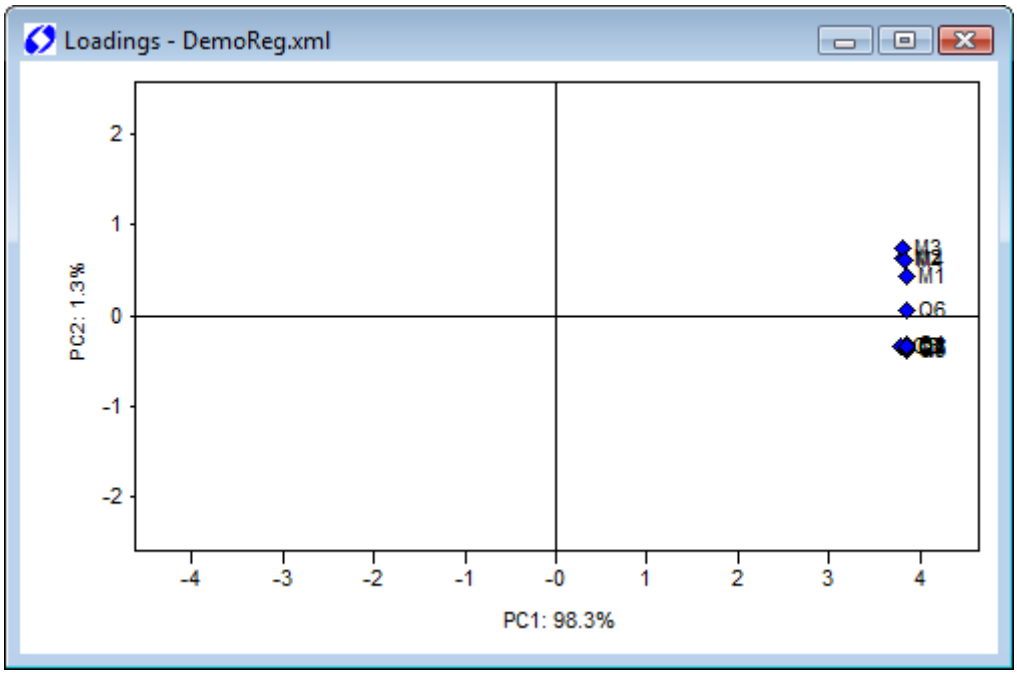
By activating **Graph - Golden cut**, a special display mode is enforced. The ratio of X to Y axis length is forced to be the square root of two (this is the ratio used in the DIN A paper size system, and does NOT correspond to the mathematical definition of golden cut, but the initial catch phrase "golden cut" turned out to be sticky...). Additionally, X and Y axis ticks and labels are removed and replaced by only the ratio of the ranges displayed on the X and Y axes (PC x : PC y = z.zzz). This display mode may be useful for comparing the results of different evaluations with one another.

By activating **Graph - Equal scale**, scalings are adjusted so that unit lengths along X and Y axes are identical, i.e. 1cm along the X axis corresponds to the same number of units as 1 cm along the Y axis.

The Symmetric axes, Golden cut, and Equal Scale settings may also be combined arbitrarily.
The menu points **Graph - X-axis** and **Graph - Y-axis**, respectively, allow you to define which principal component will be shown along which axis. This way it is also possible to use different combinations than the default (PC 1: X PC 2: Y). You may, for example, choose to display PC 3 vs PC 2.
If **Graph - Symbols** is activated reference data are represented by blue circles, and test data are represented by red diamonds.
Labels may be chosen (**Graph - Label**) from the number of the measurement in the table of reference or test data (**Graph - Label - Number**), the filename of the corresponding measurement (**Graph - Label - Filename**), the description of the sample (**Graph - Label - Sample**) and the value assigned by the user in the case of reference data or calculated by the model in the case of test data (**Graph - Label - Value**).
In **Graph – Title** corredsponding titles for the graph can be assigned.

3.2.3.5 The loadings plot

The loadings plot shows to which extent individual features contribute to the calculation of principal components. The further "inside" (close to the origin) a feature lies, the smaller its influence. The more distant from the origin, the more important. The azimuthal distribution of features gives hints about the data set's redundancy. Features providing very similar information are found very close to each other azimuthally (under the same angle). Features providing vastly different information will be located under clearly separated angles.



The same axis scaling options as for the scores plot (**Graph - Symmetric axes**, **Graph - Equal scale**, **Graph - Zoom off**)are applicable here.

Additionally, loadings can be weighted with their corresponding eigenvalues (**Graph - Weight with Eigenvalues**).

Options are available for selecting which principal component to display on which axis (**Graph - X-axis** and **Graph - Y-axis**).

Again, there is the possibility of using symbols and/or captions for the features. If symbols are chosen (**Graph - Symbols** activated), all features are shown as diamonds.

Labels options available are the number of the feature within the feature vector (**Graph – Label - Number**) , the name of the sensor used for the extraction of this feature (**Graph – Label - Sensor**), and a comprehensive caption including module, sensor, and feature extraction method names (**Graph – Label - Sensor+Method**).

Title options are identical to those available with the scores plot. The same macros are applicable.

3.2.3.6 Output of numerical values

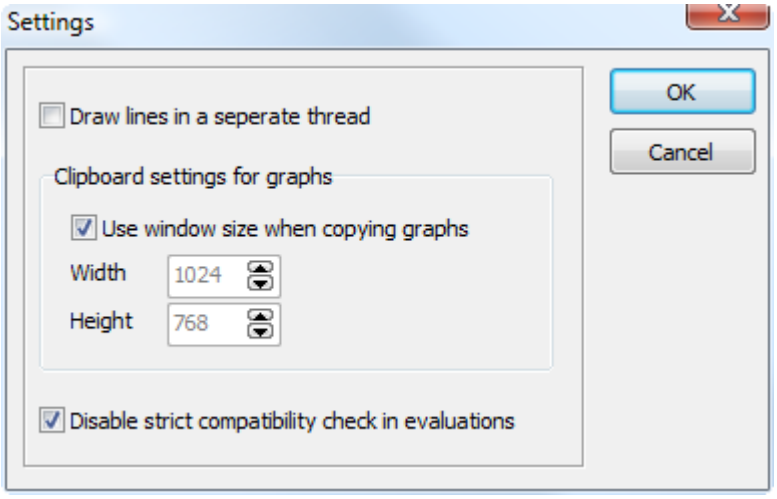
The menu **Evaluation – Matrix** accesses the following menu points

Features	Shows values of the features extracted from the reference measurements (**Reference data**) or the test data (**Test data**).
Scores	Shows the calculated scores of the principal components of the **Reference data** or **Test data**.
Loadings	Shows the loadings of all features.
Result values	Shows the true and predicted values of the **Reference data** or the predicted values of the **Test data**.

3.3 Options

3.3.1 Settings

The Settings dialog allows to set various properties of the MultiSens Analyzer program.



If **Draw lines in a separate thread** is activated, the graph display of measurement data will use separate thread to draw the sensor responses. On slow computers drawing line graphs can take considerable time especially when many measurements and sensors are displayed. As a result the program may not be able to react to user input during this time. This can be prevented, when this setting is enabled.

Clipboard settings for graphs determine how graphs are exported to the clipboard (using **Edit – Copy**). For publications and reports it is often desired, to have consistent scaling of graphics. In that case disable **Use window size when copying graphs** and set **Width** and **Height** to desired values. The graphs are exported as vector graphics (windows meta file), however the resolution and size of fonts depends on these settings.

Disable strict compatibility check in evaluations influences, how MultiSens Analyzer will examine measurements that are combined into one evaluation. Whenever you add a new measurement to an evaluation, a check is performed to make sure, that the same set-up is used in both measurements. Measurements with different set-ups (e.g. different number or types of devices) cannot be combined in one evaluation and will be rejected. If the set-ups are matching another check is performed to compare the parameters used in the measurements. If differences in the parameters are found, a warning message is presented and the user may decide whether to include the measurement or not. If this option is enabled this additional check will not be performed and no warning will be given when combining measurements with diverging parameter sets.

3.3.2 Setting of the status bar

With **Options – Icon bar** and **Options – Status bar** the display of the corresponding bar can be enabled or disabled. On small displays (e.g. Netbooks) you can save some space on your desktop using these options. Some settings might not be accessible without activated icon bar!

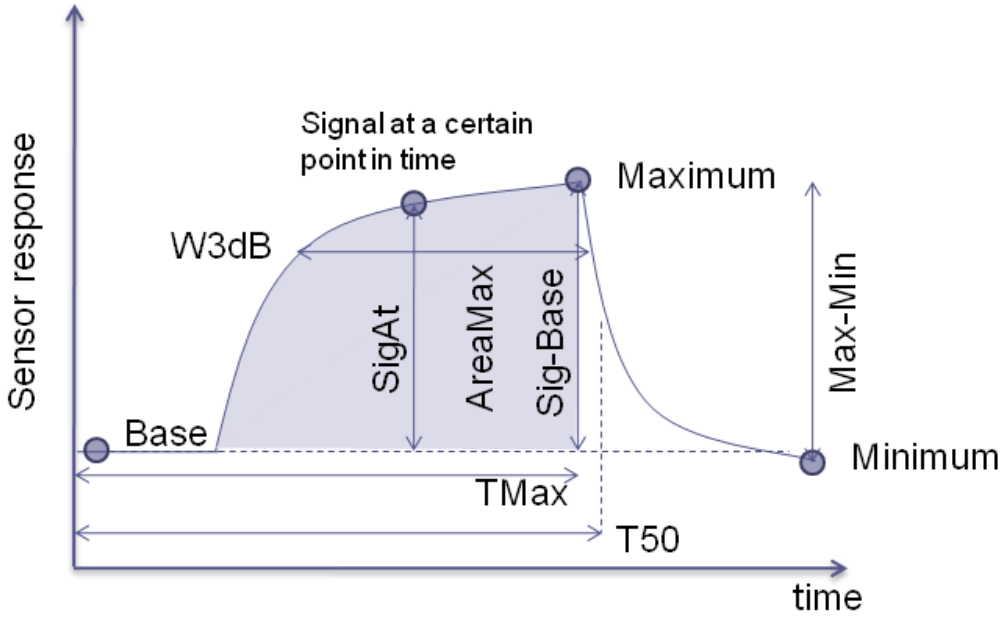
3.3.3 Default Import Settings

The default Import Settings Dialog is described in chapter 2.2. and the appendix

4 Appendix

4.1 Feature Extraction Methods

Various feature extraction methods are implemented within MultiSens Analyzer. The available feature extraction methods also depend on the version of MultiSens Analyzer that you have licensed.



Feature	Description	Version
Area	Calculates the area of the response curve above the baseline (= value of first measurement point).	Base
AreaAt	Calculates the area of the response curve above a baseline. Allows to set start point, end point, and base line determination for the calculation.	Base
AreaMax	Calculates the area of the response curve above the baseline (= value of first measurement point) from the first point to the point at which the maximum of the response is reached.	Base
Average	Calculates the aveage of the response curve.	Base
Base	Returns the baseline = value of the first measurement point.	Base
Base3	Returns the baseline = average of the first 3 measurement points.	Base
EndDev3	Returns the difference between the maximum of the response and the measurement point that occurred 3 points before the maximum.	Standard
EndDev5	Returns the difference between the maximum of the response	Standard

	and the measurement point that occurred 5 points before the maximum.	
FindSig	Returns the maximum deviation to the baseline that is found between the start point and the end point. Start point, end point and the point where to determine the baseline can be set. Additionally averaging can be set, which will average for the selected number of measurement points to calculate baseline and signal.	Base
Max	Returns the maximum of the response curve.	Base
Max-Min	Returns the span of the response curve.	Base
MaxDev	Returns the maximum derivative found in the response curve. The derivative is calculated by subtracting the predecessor point for each measurement point.	Base
Min	Returns the minimum of the response curve.	Base
PascalScript	Allows writing feature extraction algorithms in Pascal.	Professional
Sig/Base	Returns the Signal divided by the baseline. Signal is determined as the maximum deviation to the baseline in the response curve. Baseline is determined as the value of the first measurement point.	Base
Sig/Base3	Same as Sig/Base, but averages over 3 measurement points to determine signal and baseline.	Base
SigAt	Returns the signal at a measurement point. The signal is calculated by subtracting the baseline (= first measurement point) from the selected measurement point. Additionally averaging can be set, which will average for the selected number of measurement points to calculate baseline and signal.	Base
SigAt-BaseAt	Returns the difference of two measurement points in the response curve. Typically this is used to calculate a signal at a certain point by subtracting the baseline that is determined at another point. Additionally averaging can be set, which will average for the selected number of measurement points to calculate baseline and signal.	Base
Sig-Base	Sig-Base is determined as the maximum deviation to the baseline in the response curve. Baseline is determined as the value of the first measurement point.	Base
Sig-Base3	Same as Sig-Base, but averages over 3 measurement points to determine signal and baseline.	Base

T3dB	Returns the number of the measurement point, at which the signal has returned to 3dB (= 70%) of the maximum in the response curve after the maximum had been reached. The signal is calculated by subtracting the baseline (= value of the first measurement point) from the value of each measurement point.	Standard
T50	Returns the number of the measurement point, at which the signal has returned to 50% of the maximum in the response curve after the maximum had been reached. The signal is calculated by subtracting the baseline (= value of the first measurement point) from the value of each measurement point.	Standard
T90	Returns the number of the measurement point, at which the signal has returned to 90% of the maximum in the response curve after the maximum had been reached. The signal is calculated by subtracting the baseline (= value of the first measurement point) from the value of each measurement point.	Standard
TMax	Returns the number of the measurement point, at which the maximum of the response is reached.	Base
W3DB	Returns the number of measurement points from the T3dB point to the point after the maximum when the response curve has again fallen below 3dB (=70%) of the maximum.	Standard

4.2 Filter Methods

Filters are available in MultiSens Analyzer versions Standard and Professional.

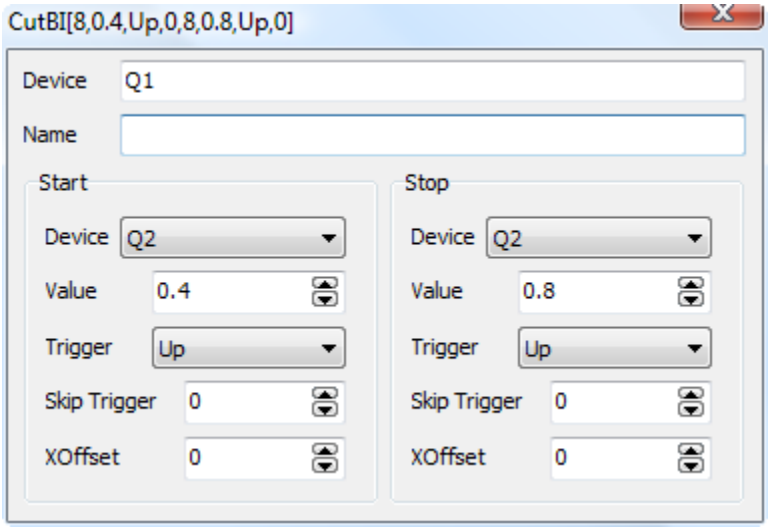
Filter	Description	Version
CutByIndicator	Allows to cut out a part of the response from a sensor. See extended description below.	Professional
IIR	Infinite impulse response filter. The IIR filter has one parameter **Width *w***. The calculation starts by calculating the average over the first ***w*** values ***x*** in the response curve. This average is then used as the starting value x'_0 for the filtered response curve. The calculation then proceeds by calculating new filtered values for all consecutive values in the response curve: $x'_i = \frac{x_i + w \cdot x'_{i-1}}{w+1}$	Standard
LinearDrift	Allows removing linear drift within a response curve. This filter has two parameters **Start** and **Stop**. The filter determines the slope of the response curve between Start and Stop. The slope is then subtracted from the response curve.	Standard
Median	The median filter has one parameter **width *w***. The median filter calculates the median of ***w*** values in the response curve. The median is calculated by sorting the ***w*** values and then picking the value that is in the middle. If ***w*** is even, then the average of the 2 values in the middle is chosen. As for the moving average this procedure is repeated for all values in the response curve.	Standard
MovingAverage	The moving average filter has one parameter **Width *w***. The calculation starts by calculating the average over the first ***w*** values ***x*** in the response curve. This average is then used as the value x'_0 for the filtered response curve. The calculation then proceeds by calculating the moving average of ***w*** values: $x'_i = \frac{1}{w} \sum_{j=i-(w-1)}^{i} x_j$	Standard
PascalScript	Allows to write a filter algorithm in Pascal.	Professional
PassThrough	This filter just passes the response through without any changes. This filter may be used to include unfiltered data in graphs that show all filters.	Standard

4.2.1 The CutByIndicator filter

The purpose of this filter is to cut out a part of a response curve. Often response curves are affected by the sampling process that may go through various steps (e.g. preconcentration, measurement, cleaning, …). Within the evaluation only a certain part of the response curve may thus contain relevant information about the measured sample. When the timing of these steps is highly reproducible a simple feature like SigAt-BaseAt may be used to extract the relevant information. However, there is instrumentation, where timing might vary and a more elaborate method has to be used instead. The CutByIndicator filter allows cutting out parts of a response curve depending on changes in other devices. (A device is merely a source of input data, so it can as well represent a valve setting, pump power, or other inputs).

The field name allows entering a descriptive **Name** for the filter. This name will then be shown in dialogs, forms and graphs instead of the standard CutBI[x,x,x,..] name.

The operation of the filter is controlled by a number of parameters that are grouped into **Start** and **Stop**, which represent the condition for the start and stop of the cut out.



The filter first determines when to start the cut out by looking at the **Device** mentioned in the **Start** box. It scans through the devices response curve to find the point at which the trigger condition becomes true. A trigger condition is described by a **Value** and a **Trigger**. **Trigger** "up" means, that the values in the response curve need to go from smaller than **Value** to larger or equal than **Value**. The point in time, where they first become larger is then the taken as starting point for the cut out.

Skip Trigger allows specifying that the trigger condition needs to happen a couple of times. I.e. if **Skip Trigger** is 2 and **Trigger** is "up", the response values in the **Device** need to go from lower to higher or equal values 3 times.

XOffset allows defining an offset to the trigger position. I.e. if **XOffset** is – 5 and the trigger has fired at position 20 the cut out would start at 15.

Further possibilities for **Trigger** are "down" - the values in the response curve first need to be larger than **Value** and then become smaller or equal - , or level – the values in the response curve must match **Value**.

If no Start-Trigger is found, the cut out will start with the last value in the response.

The conditions in the **Stop** box are working exactly the same way, as for the **Start** box, however the start of the search for the stop trigger conditions begins at the point of the start trigger (neglecting the XOffset setting). If no stop trigger is found, the cut-out will include all points from the start trigger to the end of the response.

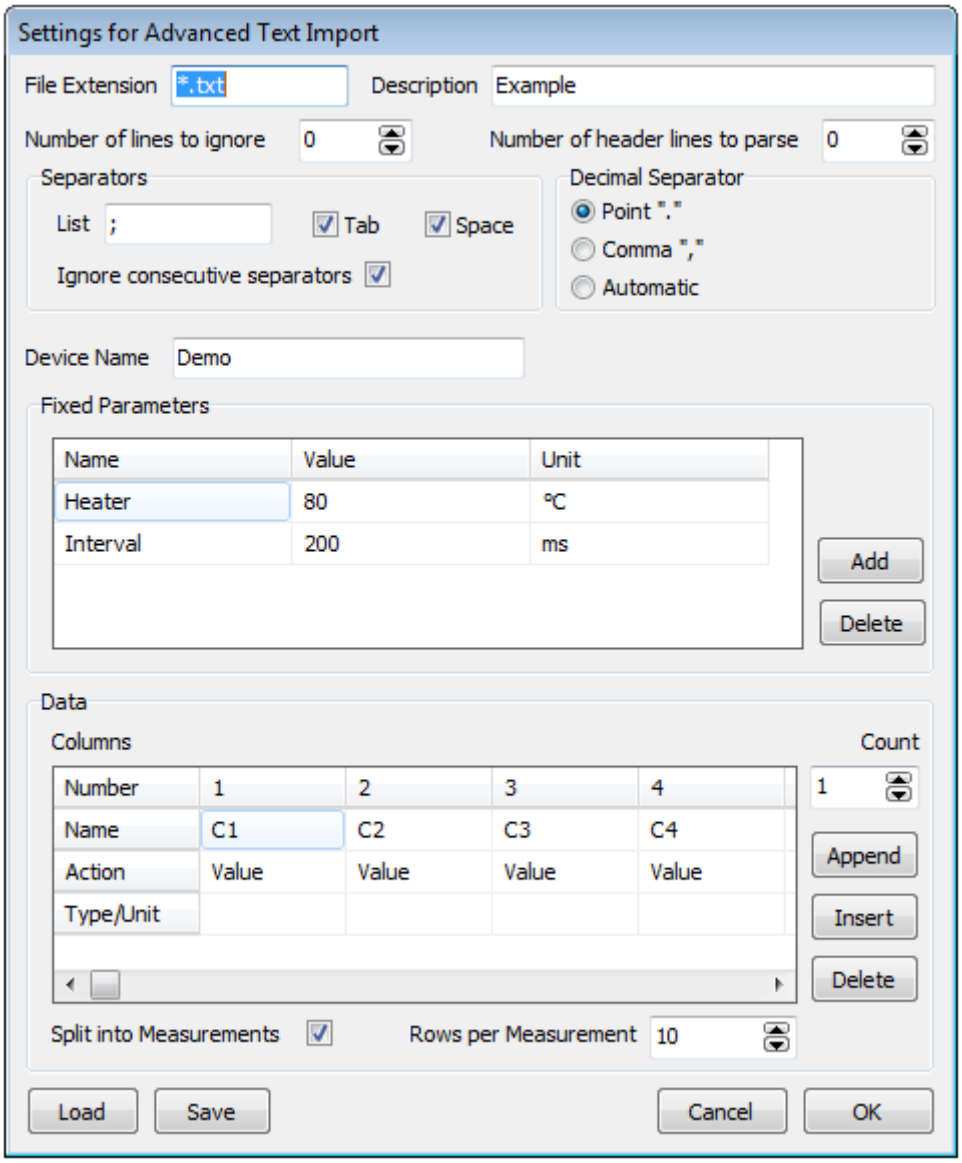
4.3 Importing data files

MultiSens Analyzer includes a number of import filters for various file formats.

Whenever you add data files to an evaluation you can choose which filter to use for the import. By default the advance text import is selected as defined under **Options – Default Import Settings…**

4.3.1 Advanced Text Import

A typical format that is often used for measurement data is text data (ASCII). Advanced Text Import lets you define how to read your text data files. This can be done via selecting **Options – Default Import Settings** in the menu. The following dialog will be displayed:



Enter the extension of the data files that you want to import into the field **File Extension**.

Description may contain a short description for the data file. You could enter the name or type of the instrument that generates the data files that you want to import. The content of this field is shown in file open dialogs.

Many data files contain a header that cannot easily be imported. By setting the value of **Number of lines to ignore** you can skip these lines during import.

Separators define how individual values are separated. Enter any characters into **List** to set them as separators. Additionally **Tab** and **Space** may be activated as separators. If **Ignore consecutive separators** is activated a new value will only be generated, if a non separator character is read from the text file. If **Ignore consecutive separators is not activated**, every separator will create a new value, that may be empty.

The **Decimal Separator** discriminates the integral from the fractional part in a numeral value. Either a **point "."** or **comma ","** may be chosen. If set to **Automatic** the first comma or point, found in the numeral value, will be taken as decimal separator. Thousands separators are not supported!

Device Name lets you set the name of the device that is used to generate the data.

Parameters is a list of arbitrary parameters that you can add to the import. These are fixed value parameters (all imported files will have the same values (entered in this dialog).

The measurement data in the text file is expected to have the form of a two dimensional array. The format is defined under **Data**:
If **Split into Measurements** is checked, then each number of **Rows per Measurement** in the data array contain an independent measurement. For each sensor/device a response curve containing the number of **Rows per Measurement** points are included in the file. If **Split into Measurements** is not checked, then only one measurement is included in the file, but for each sensor/device a response curve containing all data rows will be included.
The input grid **Columns** contains the definitions for the column of the data matrix in the text file. The number of columns in this grid must match the number of columns in the text file.
For each column we have to assign a **Name** and optionally a **Type/Unit**.

Action defines what to do with the data in the column. Following actions are available:

Action	**Description**	**Format**
Value	Sets the Value of the sensor/device.	Decimal number
Measurement	Sets the name of the measurement.	String
Ignore	Ignores the column.	-
DateTime	Sets date and time of the measurement.	Date and time according to Windows language settings
Date	Sets date of the measurement.	Date according to Windows language settings
Time	Sets time of the measurement.	Time according to Windows language settings
Year	Sets the year of the measurement.	number
Month	Sets the month of the measurement.	1 .. 12
Day	Sets the day of the measurement.	1 .. 31
Hour	Sets the hour of the measurement.	0 .. 23
Minute	Sets the minute of the measurement.	0 .. 59
Second	Sets the second of the measurement.	0 .. 59
Sample	Sets the name of the sample.	String
Parameter	Sets the content of a parameter.	String
Float Parameter	Sets the content of a parameter.	decimal number
Integer Parameter	Sets the content of a parameter.	number
Class	Sets the class that the measurement is assigned to	String
Target Value	Sets the target value that the measurement is assigned to.	Decimal number

4.3.2 XML Import

MultiSens Analyzer automatically recognizes 2 different XML file formats:

1. It's own XML file format:
   You may import any evaluation into another evaluation. In this case all the measurement data is imported, but none of the settings for the evaluation.
2. NOSE II XML files:
   Measurement files stored in the standard NOSE II XML file format version 0.4.

4.3.3 Text Import

Besides the advanced text import a simple text import is available, which requires text files to be properly formatted. If you have the ability to generate or convert your data to text files yourself, this might be the easiest way to import your data.

In its simplest form the file consists of a single line containing the word DATA and subsequently an array of data, with the sensors in the columns and the acquisitions in the row. For each measurement/sample that has been measured, an individual file needs to be created:

```
DATA
1.23 4.54 2.15 3.29
2.78 2.96 3.01 3.45
4.89 2.22 5.23 3.08
5.21 3.87 3.68 5.12
```

The following example also contains information about the sensors, the device used, the sample and the date, as well as parameters used during the acquisition:

```
#The text file always starts with a header that allows to describe
#the measurement system, the sample, date and parameters.
#Only one measurement can be included in one text file.
#For each measurement you will have to create an independent text file.
#Remarks start with an "#" sign. MultiSens Analyzer will ignore these lines.
#Lines can be left empty

#The name of the measurement can be explicitly set:
Measurement = Apple Juice 123/a
#The next line describes the sample:
Sample = Apple Juice
```

```
#The next line contains the source of the sample
Source = Crop 123
#With the next line a default class can be set when importing:
Class = Apples

#Target sets a default regression target value
Target = 12.3

#Date contains the measurement date (and time) of the measurement.
#This data is imported using the language dependent date and time settings of Windows!
Date = 2.2.2009 12:20:02

#Device describes the instrument that has been used in the measurement
Device = JLMQ

#Sensors contains a list of the sensors that are installed in the device.
#Sensors are separated by space.
Sensors = QMB1 QMB2 QMB3 QMB4

#Next we add a number of parameters, that were used during the measurement
#MultiSens Analyzer will read the parameters, and store them within the evaluation
Acquisition interval = 1
Chamber temperature = 5
#Parameters can be prefixed with an underscore "_" to be able to add parameters that
#otherwise would be interpreted as non parameter
_Class = MyParameterClass

#The measurement data from the sensors is contained in an array,
#with the sensors in the columns and subsequent acquisitions in the rows.
#The individual values have to be formatted as integers or floating points
#with a point "." as decimal separator.
#A space or tab should separate the values in a line.
#If FirstRow is defined to DateTime, Sec or MSec, the first column in the data will
#be ignored:
```

```
FirstRow = Sec
#The measurement data starts with the word DATA in a line


DATA
0 1.23 4.54 2.15 3.29
1 2.78 2.96 3.01 3.45
2 4.89 2.22 5.23 3.08
3 5.21 3.87 3.68 5.12


#if several measurements are included in one text file they need to be separated by
END_OF_MEASUREMENT
# The second measurement starts here:
Measurement = Apple Juice 123/b
# All parameters, etc. need to be repeated!
Sample = Apple Juice
Source = Crop 123
Class = Apples
Target = 12.7
Date = 2.2.2009 12:35:02
Device = JLMQ
Sensors = QMB1 QMB2 QMB3 QMB4
Acquisition interval = 1
Chamber temperature = 5
_Class = MyParameterClass
FirstRow = Sec
DATA
0 1.11 4.24 2.34 3.79
1 2.76 3.60 3.51 3.76
2 1.23 3.25 5.23 3.34
3 5.44 4.87 3.68 5.56
```

4.3.4 CSV Import

This filter allows importing comma separated text files.

Rows that only contain comma "," separated numbers (or decimal numbers) are treated as measurement data. Each number is assigned to a sensor/device. The number of sensors/devices is automatically matched to the numbers available in the longest row.

The decimal separator for decimal numbers is a point ".".

Rows that start with a string (something that cannot be interpreted as a number) will assign following parameters / information:

String	Description
DATE	Sets the date and time of the measurement. The format needs to follow the Windows language settings.
DEVICE	Sets the name of the device used for the measurement.
SAMPLE	Sets the name of the sample.
SENSORS	Sets the names of the sensors/devices. The remaining comma separated fields in the line are assigned the the sensors/devices.
SOURCE	Sets the source of the sample.
Sensor Number	Sets the names of the sensors. The remaining comma separated fields in the line are assigned the the sensors. Also sets the device in the measurement to Aromascan.
Time Interval	Sets the parameter Time Interval.
Purge Humidity	Sets the parameter Purge Humidity.
Reference Humidity	Sets the parameter Reference Humidity.
Base Resistances	Sets the parameter Base Resistance for each sensor.
any other string	Creates a new parameter and sets the name to this string. Reads the next value from the line and sets the value of the parameter.

4.3.5 Matlab Import

MultiSens Analyzer can read Matlab matrices (*.mat) directly. Depending on the dimensionality of the Matrix, the data is important as a measurement with a single sensor (1 dimension), as a measurement with several sensors (2 dimensions, individual sensors are in the columns) or several measurements with the third dimension containing different measurements.

4.3.6 Other Import formats

A number of other import formats are available in MultiSens Analyzer, that have been implemented on user requests. These include Airsense *.nos files for the PEN electronic nose of

Airsense, NeuroChem raw log files (*.txt) for data files by the EU project NeuroChem and SNet Server Log files.