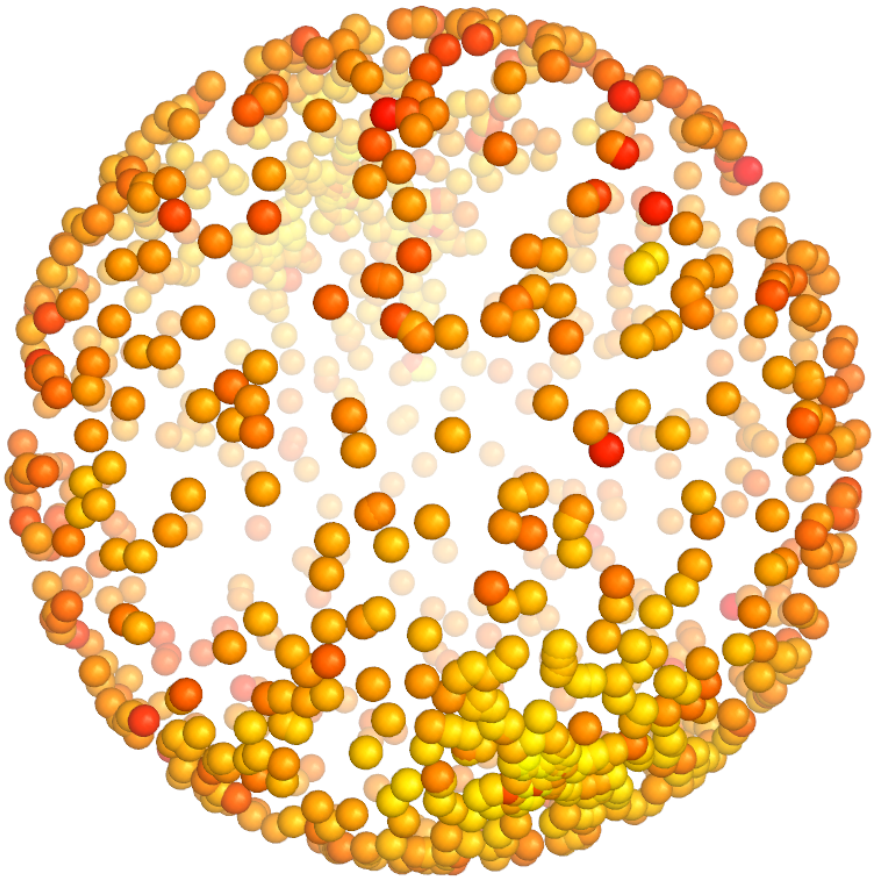
molegro data modeller

user manual

MDM 2013.3.0 for Windows, Linux, and Mac OS X



CLC bio

Molegro – A CLC bio company

Table of Contents

1 Introduction to Molegro Data Modeller....6
1.1 Contact Information....7
1.2 System Requirements....7
1.3 Reporting Program Errors....8
1.4 Text Formats Used in the Manual....8
1.5 Keyboard Shortcuts....8
1.6 Screenshots Used In the Manual....8
1.7 Future Updates....8
2 User Interface....10
2.1 Basic Concepts....10
2.2 GUI Overview....11
2.3 Workspace Explorer....11
2.4 Properties Window....12
2.5 Toolbar....14
2.6 Spreadsheet Window....16
2.7 Changing Spreadsheet Color Scheme....17
2.8 Log Window....20
2.9 Custom Data View....20
2.10 Dataset Finder....21
2.11 Workspace Properties....21
3 Managing Data and Models....23
3.1 Supported File Formats....23
3.2 Importing Datasets....23
3.3 Importing Workspaces....26
3.4 Reorder and Delete Dataset Columns....27
3.5 Dataset Scaling and Normalization....28
3.6 Convert Discrete Descriptors....29
3.7 Cross-Term Generator....31
3.8 Convert Between Numerical and Textual Descriptors....32
3.9 Handling Constant Columns....32
3.10 Deleting, Replacing, or Repairing Invalid Cells....32
3.11 Scrambling Data Columns....33
3.12 Exporting Datasets and Derived Models....33
3.13 Creating a New Dataset....34
3.14 Create New Dataset From Selected Columns....35
3.15 Data Transformation Dialog Box....36
3.16 Correlation Matrix Dialog Box....38
3.17 Bivariate Statistics....41
3.18 Diversity Statistics....42
3.19 Using Derived Regression or Classification Models....44
3.20 Compare Training / Test Set....45
3.21 Offline and command line predictions....47
3.22 Recommendations....49

4 Data Visualization..... 51
4.1 1D Plot Dialog Box.....51
4.2 2D Plot Dialog Box.....53
4.3 3D Plot Dialog Box.....55
4.4 Changing Colors in 2D and 3D Plots..... 58
5 Visualizing High-Dimensional Data.....59
5.1 The Spring-Mass Map 59
5.2 The High Dimensional Visualization Dialog.....60
5.3 Visualizing the Correlation Matrix.....63
6 Outlier Detection.....65
6.1 Quartile-Based Method.....65
6.2 Density-Based Method..... 69
7 Creating Subsets.....72
7.1 Where Can Subsets be Used?.....72
7.2 Creating Subsets From Selected Rows..... 74
7.3 Creating Subsets Using Random Selection.....75
7.4 Create Subset Using 'Subset' Column..... 76
7.5 Creating Subsets From Selected Descriptors.....76
8 Regression..... 84
8.1 Multiple Linear Regression..... 84
8.2 Partial Least Squares..... 85
8.3 Neural Networks..... 86
8.4 Support Vector Machines..... 88
8.5 Choosing a Regression Method..... 90
8.6 Creating Regression Models Using the Regression Wizard.....91
8.7 Inspecting Regression Models.....109
8.8 How to Make Predictions Using an Existing Model.....114
9 Classification..... 115
9.1 K-Nearest Neighbors..... 115
9.2 Support Vector Machines..... 116
9.3 Choosing a Classification Method.....116
9.4 Creating Classification Models Using the Classification Wizard.....117
9.5 Inspecting Classification Models.....121
9.6 How to Make Classifications Using an Existing Model.....125
10 Principal Component Analysis..... 126
10.1 General Overview..... 126
10.2 Performing a Principal Component Analysis.....126
10.3 PCA Regression..... 131
10.4 Subset Creation from Principal Components.....132
11 Clustering and Similarity.....133
11.1 K-Means Clustering..... 133
11.2 Density-Based Clustering..... 138
11.3 Threshold-based Clustering..... 140
11.4 Visualization of Clusters.....141
11.5 Example: Iris Dataset.....142

11.6 Similarity Browser.....143
11.7 How To Use The Similarity Browser.....144
11.8 Customizing the Similarity Browser.....146
11.9 Clustering/Similarity Measures.....147
12 The Chemistry Module.....149
12.1 Importing Chemical Structures.....150
12.2 Working with Molecular Depictions.....153
12.3 The Layout Engine and Internal Molecule Representation.....157
13 Customizing Molegro Data Modeller.....159
13.1 General Preferences.....159
13.2 Command Line Parameters 161
14 Help.....163
14.1 PDF Help.....163
14.2 The Molegro Website 163
14.3 Technical Support.....163
15 Appendix I: Statistical Measures.....164
15.1 General Symbols Used.....164
15.2 Univariate Analysis.....164
15.3 Bivariate Analysis.....166
16 Appendix II: References.....170
17 Appendix III: Third Party Copyrights.....171

1 Introduction to Molegro Data Modeller

Molegro Data Modeller (MDM) is an integrated environment for analyzing and mining data.

The Molegro Data Modeller can be used to:

- Create regression and classification models using imported numerical descriptors
- Identify relevant numerical descriptors using feature selection algorithms
- Fine-tune parameters used by regression and classification algorithms using grid-based search
- Predict numerical properties of imported records using a derived regression model
- Classify imported records using a derived classification model
- Inspect and analyze numerical descriptors and regression/classification models
- Cluster data using various similarity measures
- Prepare datasets (e.g. normalize/scale/transform data, handle missing values)
- Perform principal component analysis (PCA)
- Detect outliers
- Visualize high-dimensional data using Spring-Mass Maps

The preferred way to get started with MDM is:

- Read the instructions on how to use the GUI (Chapter 2)
- Read how to import and prepare datasets (Chapter 3)

- Read how to make a regression model (Chapter 8)
- Read how to make a classification model (Chapter 9)
- Read how to cluster data (Chapter 11)

Overall, Chapters 3 to 11 describe various aspects of MDM from importing and preparing datasets, detecting outliers, to making regression/classification models or clustering datasets, and inspecting the solutions found. Chapter 12 introduces the chemistry module that extends MDM with several features for working with molecular structures. Chapter 13 describes how to customize MDM using the built-in preferences and Chapter 14 concludes with a short overview of where to obtain more information about the usage of MDM.

1.1 Contact Information

Molegro Data Modeller is developed by:

Molegro – a CLC bio company

Finlandsgade 10-12

8200 Aarhus N

Denmark

http://www.clcbio.com

VAT no.: DK 28 30 50 87

Telephone: +45 70 22 55 09

Fax: +45 70 22 55 19

E-mail: **info@clcbio.com**

If you have questions or comments regarding the program, you are welcome to contact our support function:

E-mail: **support@clcbio.com**

1.2 System Requirements

The system requirements for Molegro Data Modeller are:

- Windows 7, Vista, XP, and 2003.
- Linux: Most standard distributions. We provide both 32 and 64 bit builds. Please send a mail to support@clcbio.com if the program does not work on a particular distribution – and we will try to provide a new build.
- Mac OS X 10.5 Intel (and later versions).

1.3 Reporting Program Errors

If you discover a program error, please mail the information to:

support@clcbio.com

Remember to specify how the error can be reproduced, the version number of Molegro Data Modeller in question, and the operating system that was used. If possible, please include dataset files used (e.g. CSV, MDM). This will make it easier for us to reproduce (and correct) the error.

1.4 Text Formats Used in the Manual

The following formatting styles are used in this manual:

- All GUI text, labels, and keyboard shortcuts are written in bold face with initial capital letters.

  Examples: **Workspace Explorer**, **Training algorithm**, **Ctrl-O**
- Menus and menu items are identified using dividing lines and bold face.

  Example: **Modelling** | **Transform Data...** indicates that the user should first select the **Modelling** menu and then select the **Transform Data...** menu item.
- Filenames are written in mono-spaced font.

  Example: `\Molegro\MDM\bin\mdm.exe`

1.5 Keyboard Shortcuts

The keyboard shortcuts used in the manual are for Windows and Linux versions of MDM. On Mac OS X, the **CTRL** key is replaced by the **command** key and function key shortcuts (e.g. **F1**) are invoked by pressing the function key and the **fn** key (e.g. **fn+F1**).

1.6 Screenshots Used In the Manual

The screenshots used in the manual are taken from the Windows XP version of MDM. Therefore, dialogs and other GUI related material may differ slightly on Linux and Mac OS X versions.

1.7 Future Updates

Molegro Data Modeller contains a built-in version checker making it easy to check for new program updates including new features and bug fixes. To check for new updates, select **Help** | **Check for Updates**. A window showing available updates and details about changes made will appear (see Figure 1).

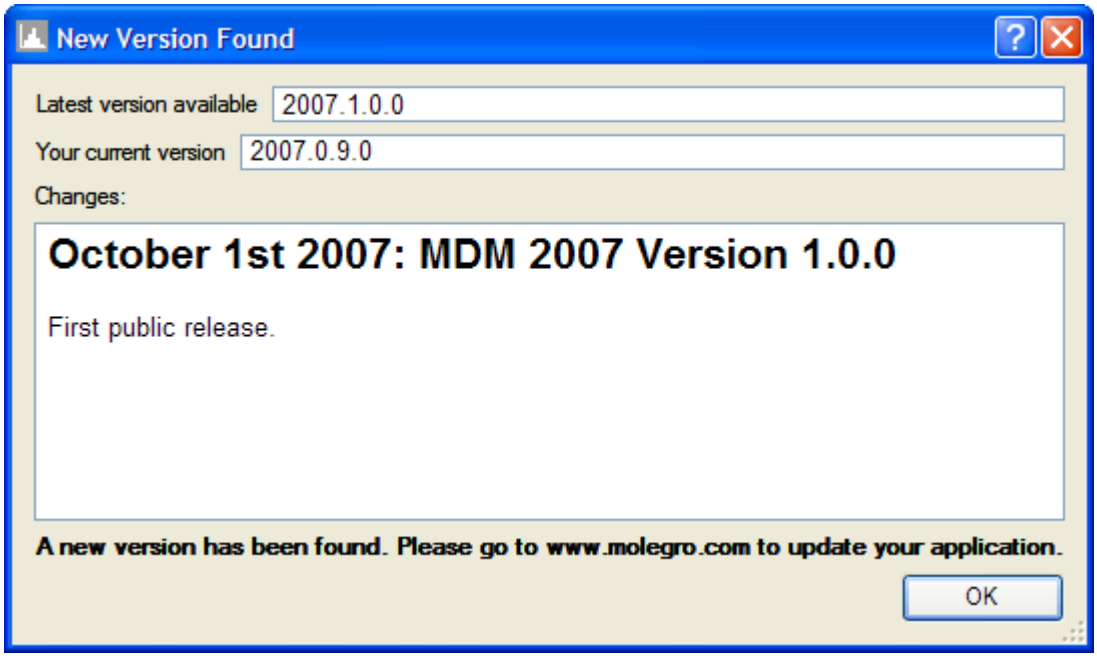


Figure 1: Check for updates.

2 User Interface

2.1 Basic Concepts

Molegro Data Modeller is based on the notion of workspaces, datasets, models, descriptors, predictions, and classifications.

The *workspace* is the central component and represents all the information available to the user in terms of *datasets*, regression and classification *models*, *predictions*, and *classifications*. By default, an empty workspace is shown when starting Molegro Data Modeller. A workspace can be saved, cleared, merged with or replaced by other workspaces (datasets are added to the current workspace when they have been imported).

A dataset consists of a number of numerical and textual descriptors (columns). Each row in the dataset corresponds to a given data record in the dataset. Numerical descriptors are columns containing numerical values only – all other columns are categorized as textual descriptors. Molegro Data Modeller does not impose any limits to the number of descriptors or data records that can be used. However, the number of cells (number of data records × (number of descriptors + number of predictions)) is limited by the amount of memory available on the computer.

Models representing regression or classification models made with Molegro Data Modeller contain information about descriptors used, data transformations performed (e.g. normalization of raw data), and target descriptor used. After training a model using the built-in regression or classification algorithms, the model can be used to make predictions on other datasets.

When a prediction is made on a dataset using one of the models available in the workspace, a new prediction/classification column is added to the dataset. The new prediction column containing the predicted values will be similar to

other descriptors available except for some statistical information (e.g. Pearson correlation coefficient, Mean Squared Error, Classification accuracy) that is stored in the workspace. The statistics information can be inspected later on using the **Properties Window** (introduced in Section 2.4).

2.2 GUI Overview

The main user interface in Molegro Data Modeller is composed of a central spreadsheet view (referred to as the **Spreadsheet Window**), a **Workspace Explorer** window, a **Properties Window**, and a **Log Window**.

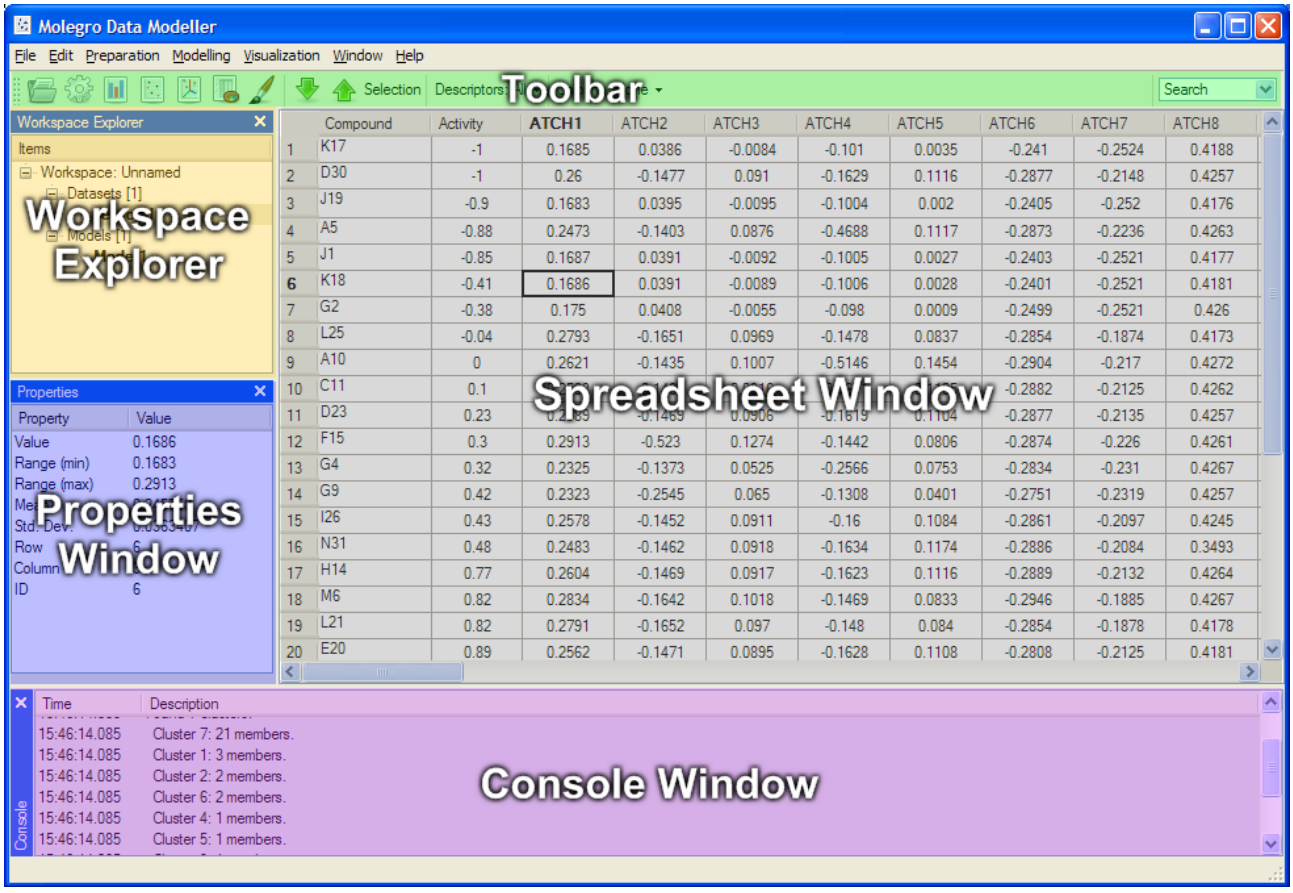


Figure 2: Main application window.

2.3 Workspace Explorer

Molegro Data Modeller includes a **Workspace Explorer** window, which contains information about datasets (containing numerical and textual data columns) and the regression and classification models available in the current workspace.

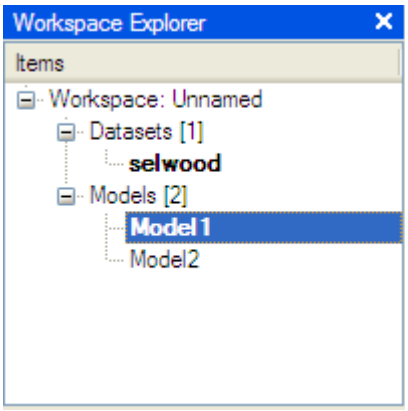


Figure 3: Workspace Explorer window.

The Workspace Explorer context menu (invoked by pressing the right mouse-button) allows the user to:

- Export and rename the current workspace.
- Edit workspace properties/notes.
- Export, rename, clone, and delete datasets.
- Revert to original sorting order, i.e. sort the current dataset records according to their order of occurrence when imported to MDM.
- Split a dataset (using the 'Subset' column). See Section 7.1 for more details.
- Extract one subset (using the 'Subset' column) from a dataset. See Section 7.1 for more details.
- Export, rename, and delete regression/classification models.
- Show regression and classification model details (e.g. descriptors used by a model). See Section 8.7 for more details.
- Make predictions using a selected regression or classification models. See Section 3.19 for more details.

2.4 Properties Window

The **Properties Window** contains information about the currently selected objects in the Workspace Explorer or in the Spreadsheet Window. Figures 4-7 show examples of different properties for a *model* selected in the Workspace Explorer window, a *numerical cell* in the Spreadsheet Window, a *predicted cell* in the Spreadsheet Window, and multiple selections in the Spreadsheet Window, respectively.

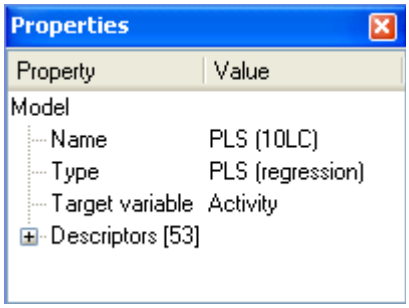


Figure 4: Properties for a model selected in the Workspace Explorer window.

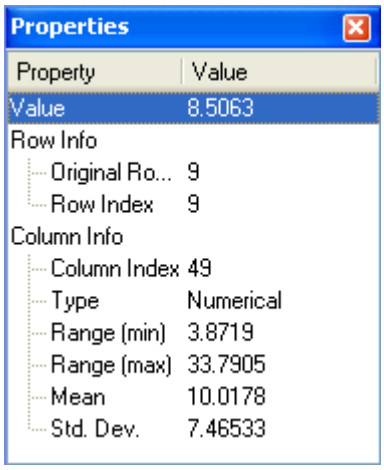


Figure 5: Properties for a numerical cell in the Spreadsheet Window.

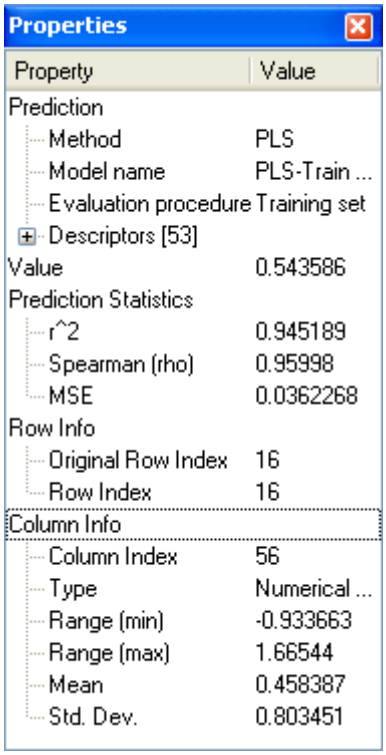


Figure 6: Properties for a predicted cell selected in the Spreadsheet Window.

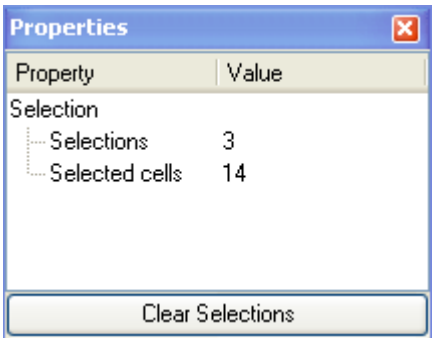


Figure 7: Properties for multiple selections in spreadsheet. All selections can be cleared by pressing the 'Clear Selections' button.

2.5 Toolbar

The **Toolbar** provides easy access to the most commonly used actions ins Molegro Data Modeller, such as importing datasets, invoking regression/classification/clustering wizards, and inspecting numerical descriptors and predictions using the **Visualization** (histogram, 2D/3D plots) and **Show Correlation Matrix** dialog boxes.

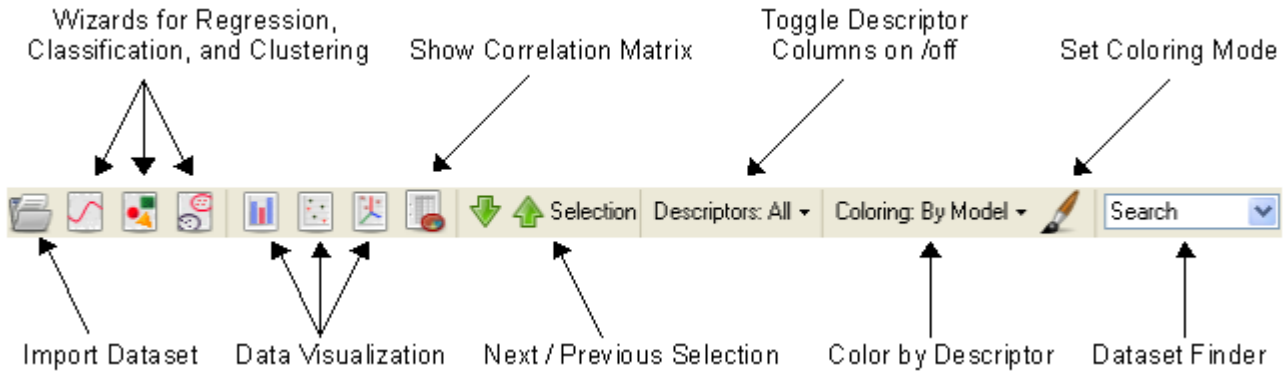


Figure 8: The Molegro Data Modeller toolbar.

The toolbar contains two selection buttons (down/up arrows) to jump to the next or previous selection in the Spreadsheet Window. This is particularly useful when browsing records selected using the plot dialog boxes (see Chapter 4 for more details).

The toolbar also contains a toggle button that makes it possible to switch between different view modes in the Spreadsheet Window (only applicable if regression/classification models are available in the Workspace Explorer):

- **Descriptors: All** shows all descriptors available for the current dataset.
- **Descriptors: Used** shows only the descriptors used by the model currently selected.
- **Descriptors: None** hides all numerical descriptors.

For all three views, target variable, textual, and prediction columns are shown.

The last toggle button on the toolbar makes it possible to switch between different coloring modes in the Spreadsheet Window:

- **Coloring: Default** turns on default coloring mode: Textual descriptors are colored gray and numerical descriptors are colored white. Predicted regression columns are colored dark-green and classification columns are colored light-blue.
- **Coloring: By Model**. The Spreadsheet Window is colored using the following scheme: Textual descriptors are colored gray, numerical descriptors are colored blue. The target variable column (indicating the numerical descriptor that the current model estimates) is colored light-

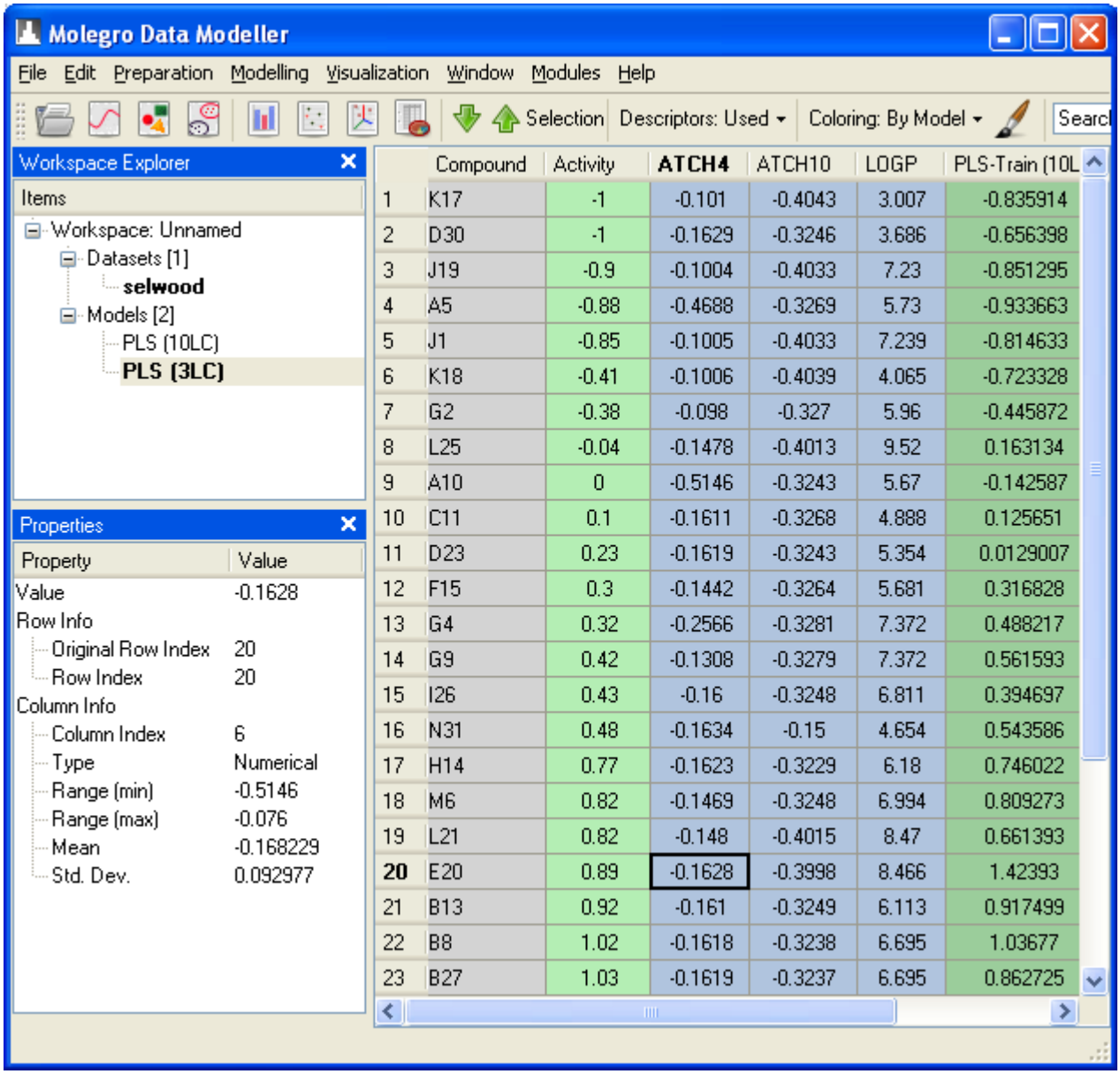
green, and the predicted columns are colored dark-green and light-blue (for regression and classification columns, respectively). Notice that target variable and numerical descriptor columns are only colored if a model has been selected in the Workspace Explorer. When selecting other models in the Workspace Explorer, the coloring may change depending on the model selected. An example of this coloring mode is shown in Figure 9.

- **Coloring: By Descriptor** uses a coloring setting defined by the user. If no color settings are defined, the **Color By Descriptor** dialog box will be invoked allowing the user to define the color scheme (based on a user-selected descriptor). Section 2.7 describes the **Color By Descriptor** dialog box in more details. An example of this coloring mode is shown in Figure 12.
- The **Define descriptor color scheme...** menu option is available by pressing the small arrow on the right-hand side of the toggle button. This option invokes the **Color By Descriptor** dialog box which allows the user to change the color settings for the **Coloring: By Descriptor** mode described above.

The **Color by Descriptor** button (pen icon) is used to change the color of the current spreadsheet and the coloring of data points in the 2D/3D plots (see Section 2.7 for more details).

Finally, the **Dataset Finder** located at the far right side of the toolbar can be used to quickly search for descriptor names and values in the current dataset (see Section 2.10 for more details).

2.6 Spreadsheet Window



	Compound	Activity	ATCH4	ATCH10	LOGP	PLS-Train (10L
1	K17	-1	-0.101	-0.4043	3.007	-0.835914
2	D30	-1	-0.1629	-0.3246	3.686	-0.656398
3	J19	-0.9	-0.1004	-0.4033	7.23	-0.851295
4	A5	-0.88	-0.4688	-0.3269	5.73	-0.933663
5	J1	-0.85	-0.1005	-0.4033	7.239	-0.814633
6	K18	-0.41	-0.1006	-0.4039	4.065	-0.723328
7	G2	-0.38	-0.098	-0.327	5.96	-0.445872
8	L25	-0.04	-0.1478	-0.4013	9.52	0.163134
9	A10	0	-0.5146	-0.3243	5.67	-0.142587
10	C11	0.1	-0.1611	-0.3268	4.888	0.125651
11	D23	0.23	-0.1619	-0.3243	5.354	0.0129007
12	F15	0.3	-0.1442	-0.3264	5.681	0.316828
13	G4	0.32	-0.2566	-0.3281	7.372	0.488217
14	G9	0.42	-0.1308	-0.3279	7.372	0.561593
15	I26	0.43	-0.16	-0.3248	6.811	0.394697
16	N31	0.48	-0.1634	-0.15	4.654	0.543586
17	H14	0.77	-0.1623	-0.3229	6.18	0.746022
18	M6	0.82	-0.1469	-0.3248	6.994	0.809273
19	L21	0.82	-0.148	-0.4015	8.47	0.661393
20	E20	0.89	-0.1628	-0.3998	8.466	1.42393
21	B13	0.92	-0.161	-0.3249	6.113	0.917499
22	B8	1.02	-0.1618	-0.3238	6.695	1.03677
23	B27	1.03	-0.1619	-0.3237	6.695	0.862725

Figure 9: Spreadsheet Window with different coloring styles for columns depending on the column type (textual, numerical, target variable, prediction).

The **Spreadsheet Window** is the central window in Molegro Data Modeller listing the descriptors (numerical and textual) and predictions (if any) of the currently selected dataset (shown in boldface in the **Workspace Explorer** window).

It is possible to perform basic editing in the spreadsheet, such as manually editing a cell by double clicking on it using the mouse. For numerical cells, only valid numerical values will be accepted. Copy-and-paste operations can be done using **CTRL+C** to copy one or more selected cells and **CTRL+V** to paste the selected cells into another region. If the selected region in the spreadsheet

is larger than the content in the clipboard buffer – the entire region will be filled with the clipboard content by repeatedly copying from the clipboard (e.g. useful for filling out a region with identical or repetitive values). Notice that cells containing textual information cannot be pasted into numerical cells.

The context menu (invoked by pressing the right mouse-button) on a spreadsheet cell allows the user to:

- Insert numerical or textual columns.
- Rename a column.
- Add new rows. The new rows will be added to the bottom of the spreadsheet. The number of rows suggested corresponds to the number of lines in the current clipboard buffer.
- Sort column in ascending/descending order.
- Revert to original sorting order (the order of occurrence when dataset was imported).
- Select entire column/row or all cells.
- Delete selected rows or columns.
- Create a subset from selected rows (see Chapter 7 for more details).

These actions are also available from the **Edit** menu located in the main menu bar (except for **Create subset from Selected Rows** which is available from the **Preparation** menu).

2.7 Changing Spreadsheet Color Scheme

The **Color By Descriptor** dialog box can be used to change the colors used in the Spreadsheet Window.

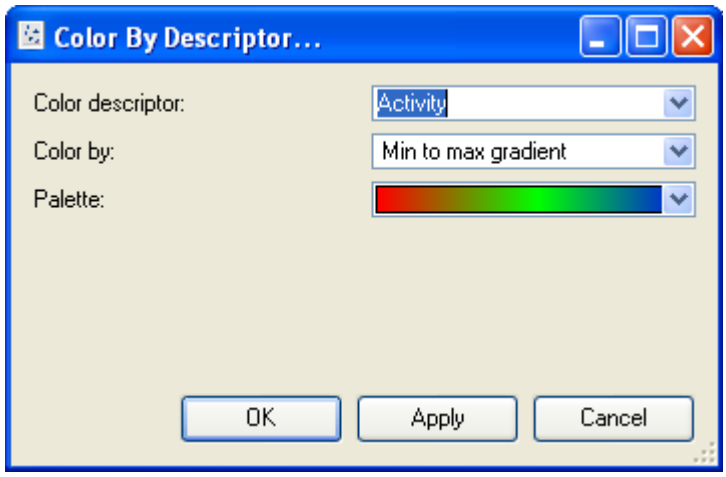


Figure 10: Color By Descriptor dialog box.

The dialog box can be invoked from the **Color By Descriptor** button (pen icon) or the **Coloring** mode toggle button on the **Toolbar** or from the **Visualization** | **Color By Descriptor...** main menu.

The **Color descriptor** specifies which descriptor should be used for the new color scheme. The **Color by** option is used to define whether the color scheme should be gradient-based (**Min to max gradient**) or (**Max to min gradient**), based on discrete classes (**Discrete classes**), or based on user-defined intervals (**User-defined intervals**). Finally, the **Palette** combo box offers a set of pre-defined color palettes to chose from. Notice: Textual descriptors are restricted to use **Discrete classes** only.

The user-defined intervals are typed into the dialog box as a comma separated list of interval boundaries. In Figure 11, all records with *Activity* values below 0.5 will be colored red, all records with values between 0.5 and 1.0 will be colored green, and all records with values above 1.0 will be colored blue.

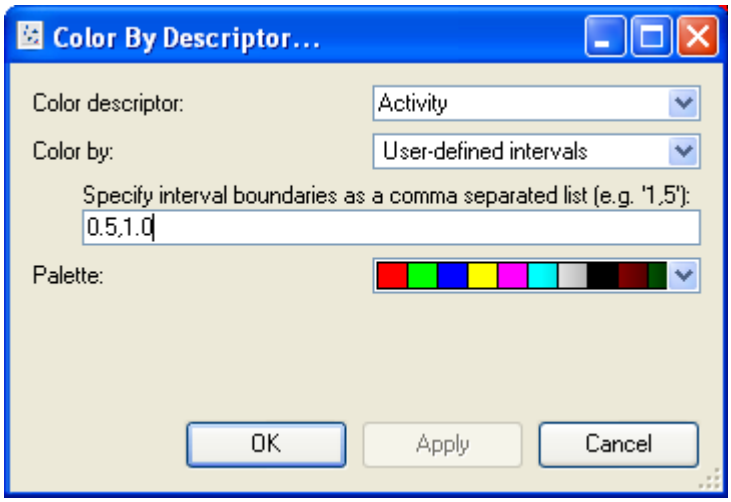


Figure 11: Using user-defined intervals for coloring spreadsheet.

Notice: The color scheme defined is *static* meaning that when it has been applied to the spreadsheet, modifications in the spreadsheet (e.g. changing a descriptor value or adding/removing records) will not alter the coloring of the spreadsheet. To update the coloring to reflect the new changes, the **Color By Descriptor** dialog box has to be invoked again.

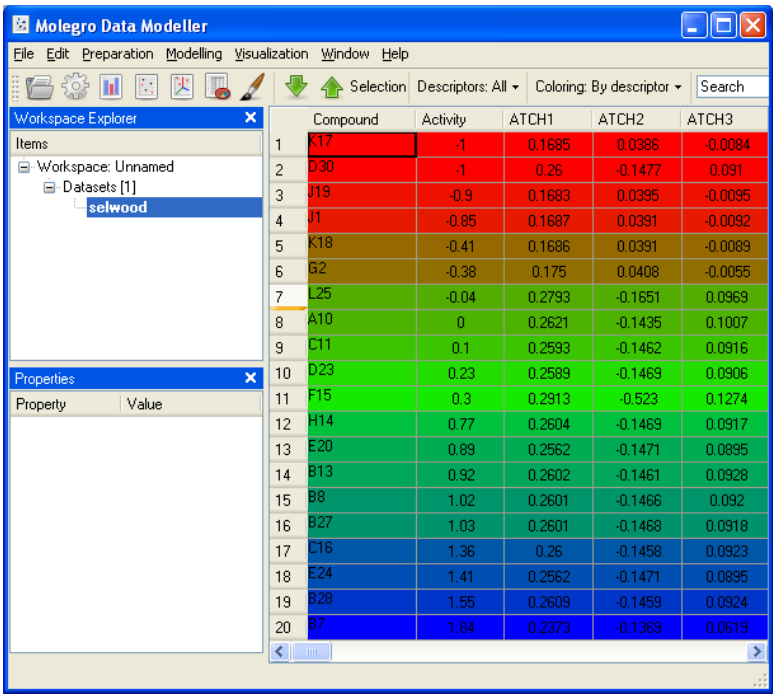


Figure 12: Example of gradient-based coloring scheme.

It is also possible to manually select an entry and color all entries of the same kind. This is done by invoking the context-menu on the desired entry and selecting **Color Selected Values in 'xxx'**. From the sub-menu it is possible to choose a color from either a palette of standard colors or from a color chooser dialog.

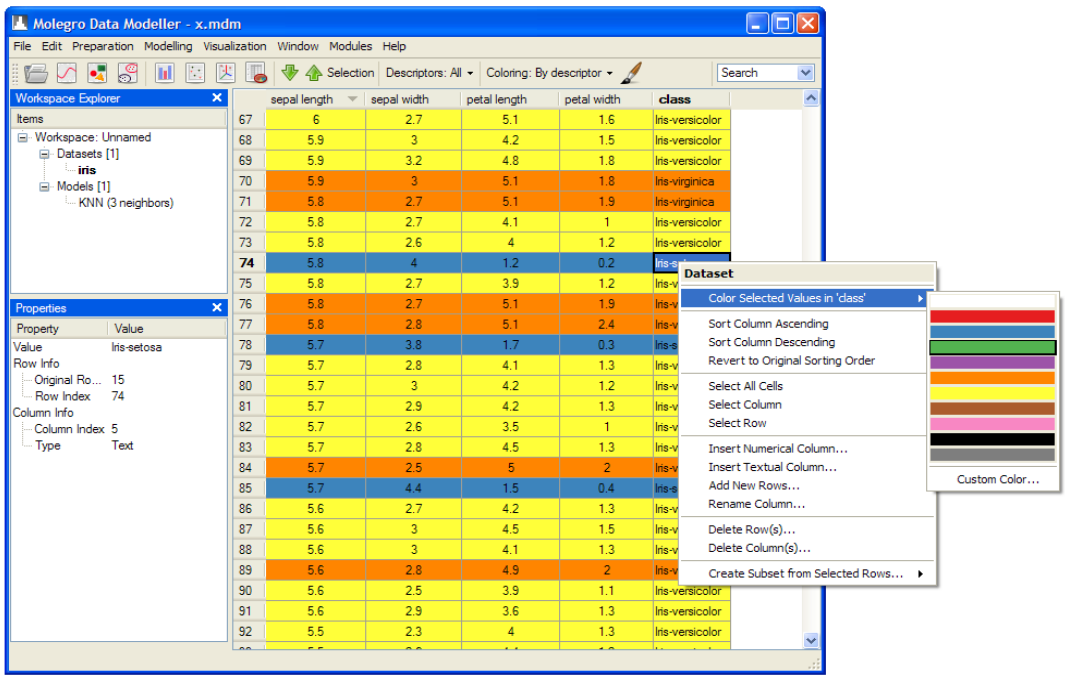


Figure 13: Manually choosing coloring for a particular class.

Manual coloring is in particular useful when dealing with classification since it is fast to select and color particular classes.

2.8 Log Window

The **Log Window** (at the bottom of the screen) displays information, warnings, and errors. The amount of information in the log can be controlled with the associated context menu (right mouse button click) - e.g. info, warnings, and debug messages can be turned off and console information can be copied to the clipboard. The **Log Window** can be toggled on and off using the **Window** | **Log** menu.

2.9 Custom Data View

The **Custom Data View** dialog box can be toggled on and off using the **Window** | **Custom Data View** menu. **The Custom Data View** dialog box (see Figure 14) can be used to display a second view of the currently selected rows in the Spreadsheet Window focusing on user-selected descriptors. To include a descriptor in the window, select the descriptor in the combo box and press the **Add** button. The descriptors shown in the window can be toggled on and off using the context menu. It is also possible to sort the items according to a given descriptor by clicking on the column header.

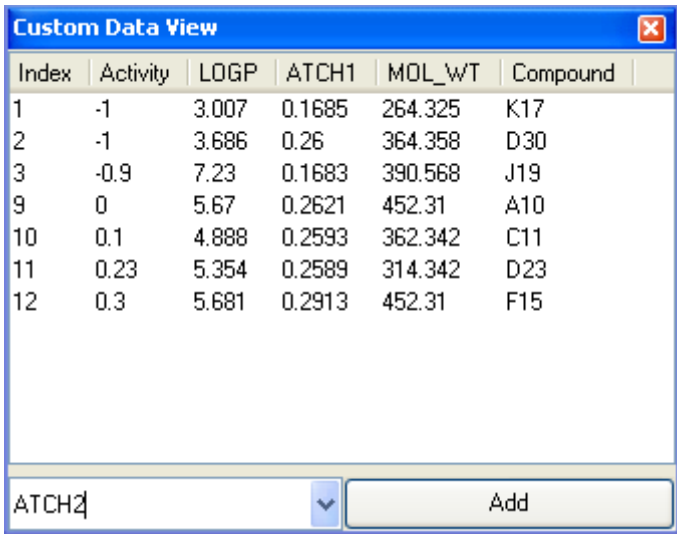


Figure 14: Custom Data View dialog box.

Notice: When changing dataset in the Workspace Explorer, the current selection of descriptors in the **Custom Data View** will be updated so that only descriptors available in the new dataset will be shown.

2.10 Dataset Finder

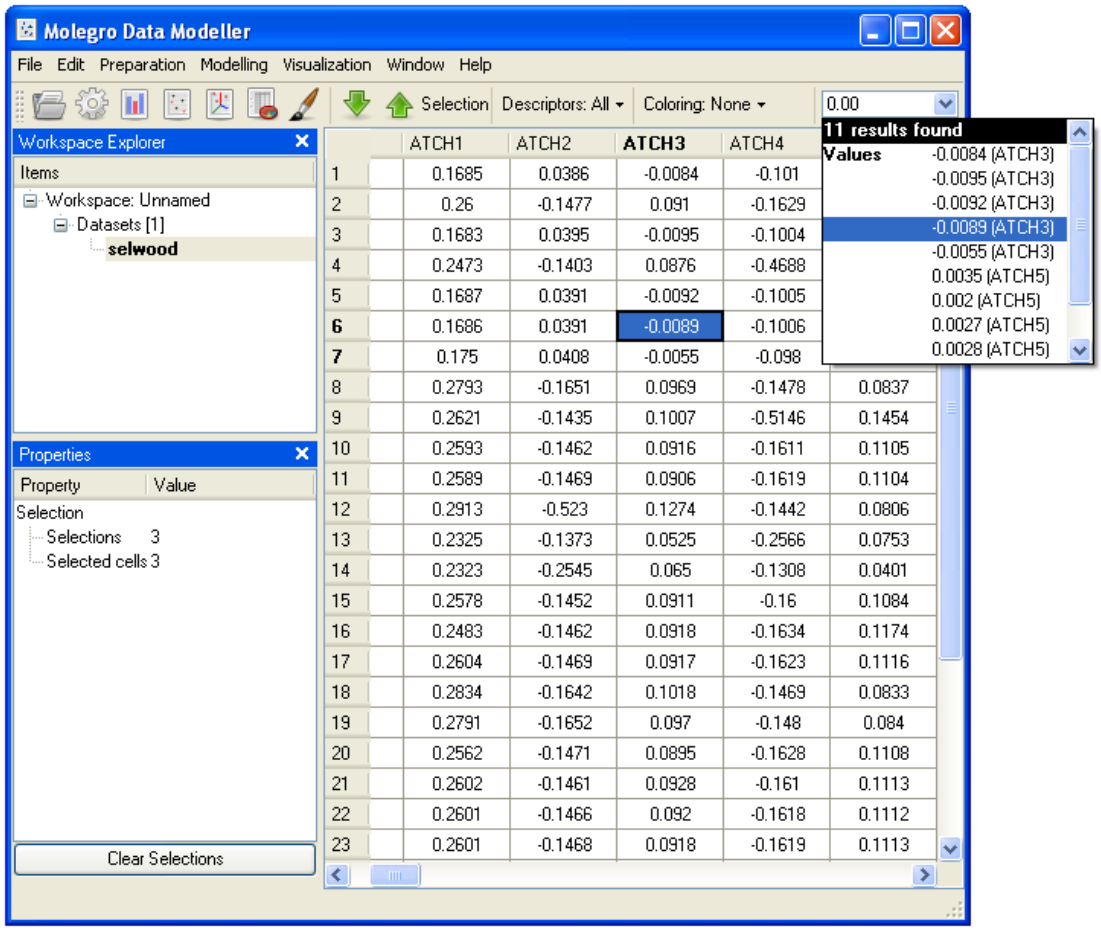


Figure 15: Dataset Finder dialog box.

The **Dataset Finder** located on the **Toolbar** (see Figure 15) allows you to quickly search for descriptor names, numerical values, and text entries in the current dataset. When a textual name or a numerical value (or part of it) is typed in the search box, the **Dataset Finder** displays a list of matches (a maximum of 30 matches is returned). The **Dataset Finder** can be invoked from the **Edit** | **Edit Search Query...** menu or by typing characters in the search box (text field) located at the far right side of the **Toolbar**. A shortcut is provided using the **CTRL+F** keyboard shortcut.

To select a result, press the **Return** key. Pressing the **Escape** (Esc) key or mouse-clicking outside the **Dataset Finder** window will cancel the current search query.

2.11 Workspace Properties

Workspaces can contain user-specified notes that can be edited using the **Workspace Properties** dialog box. The workspace title and notes will be stored when the workspace is saved.

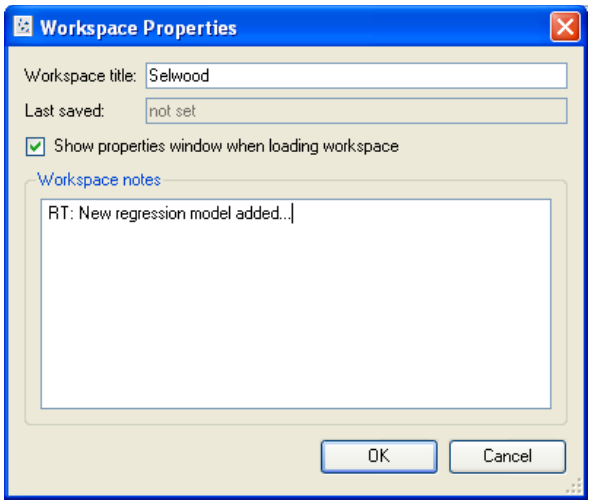


Figure 16: Workspace Properties dialog box.

The **Workspace Properties** dialog box can be invoked from the **Edit Properties...** context menu on the **Workspace** item in the **Workspace Explorer** or from the **Edit** | **Workspace Properties...** main menu.

3 Managing Data and Models

3.1 Supported File Formats

Molegro Data Modeller supports the *Text CSV* file format for importing datasets, where data are separated by either commas, tabs, spaces, or semicolons. Moreover, *MVD Results (mvdresults)* files (tab-separated files containing various numerical descriptors calculated by the Molegro Virtual Docker software) can also be imported using the **Import Dataset...** dialog box.

Molegro Data Modeller also uses its own data modeling XML format with file extension *MDM (*MDM is a shorthand notation for *Molegro Data Modeller markup language)*. In general, MDM can be used to store the following information:

- Datasets (including statistical information about predictions/classifications made)
- Regression models (introduced in Chapter 8)
- Classification models (introduced in Chapter 9)
- Workspace properties (see Section 2.11 for details)

3.2 Importing Datasets

Datasets can be imported into Molegro Data Modeller using the **Import Dataset...** menu option located in the **File** menu. To invoke the dialog box select the File folder icon on the toolbar or use the **CTRL+O** keyboard shortcut. CSV and MVD Results files can also be imported by dragging-and-dropping a given file into the main window.

When importing Text CSV (comma separated) or MVD Results (tab separated) files - a CSV Import Wizard is shown allowing for customization of CSV import settings and dataset preview.

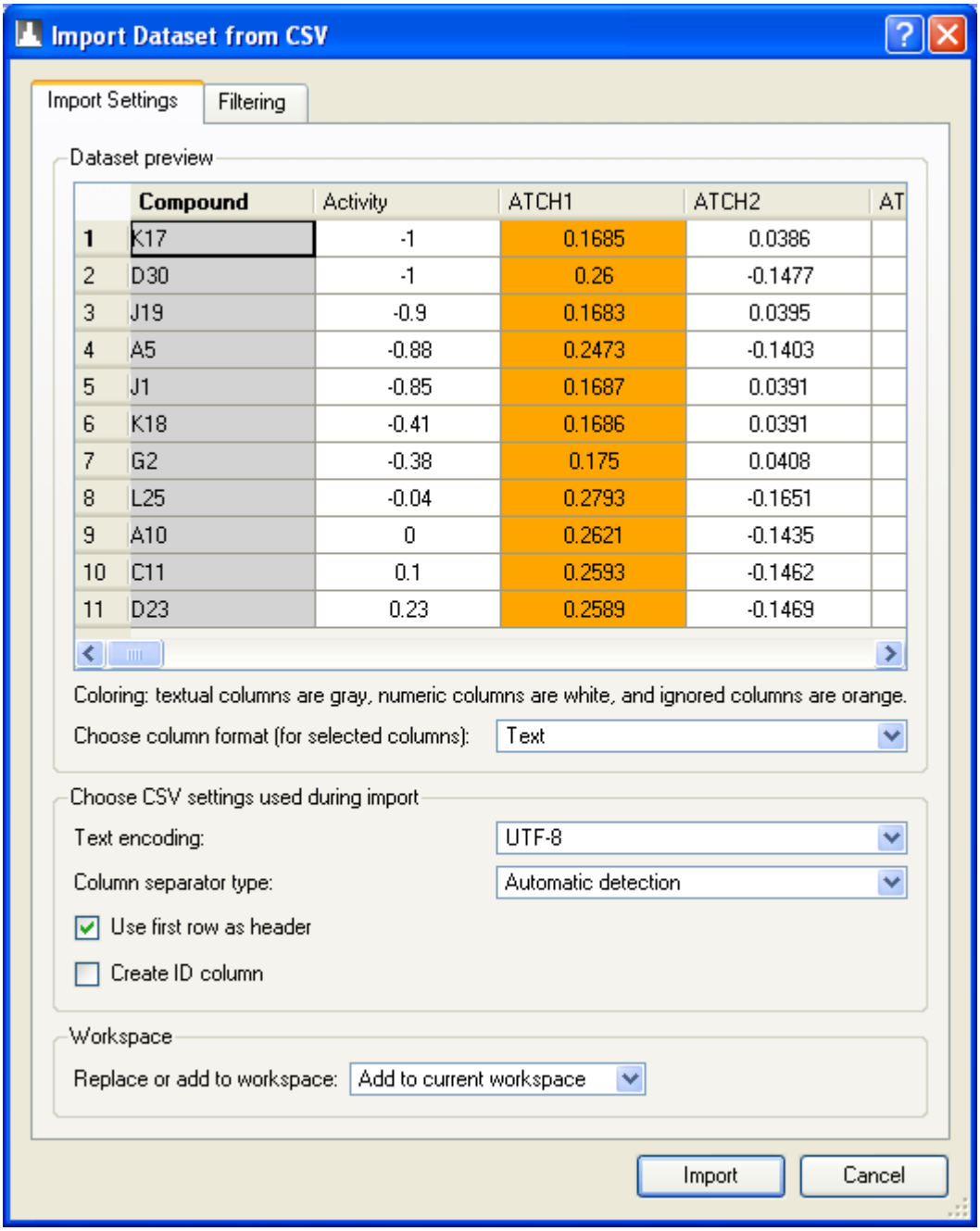


Figure 17: Import Dataset from CSV file. It is possible to preview the dataset and to change specific CSV import settings.

In the Dataset preview table it is possible to customize the column format for selected columns or for individual columns using the context menu. Numerical columns can be converted to text columns and vice versa. However, a text

column can only be converted to a numeric column if all cells in the given column can be interpreted as numeric values. In addition, it is possible to ignore columns during import.

The following CSV import settings are available (see Figure 17):

- **Text encoding:** In most cases importing files as Unicode will work as expected. Files stored as 8-bit ANSI/ASCII files will also be imported correctly as Unicode *if they do not contain any special national characters*. If they do, it might be necessary to change the encoding to Locale 8-bit. Notice that it is recommended to always work with data in Unicode, because of the greater flexibility and portability.
  Per default MDM stores data in UTF-8 – this can however be changed from the Preferences dialog.
- **Column separator type**: By default **Automatic detection** is used which means that MDM will try to identify the separator symbol automatically, i.e., the most frequently occurring symbol (comma, tab, space, semicolon) in the first text line). If the automatic detection fails, it is possible to select a separator symbol manually.
- **Use first row as header**: When this option is enabled, header information (i.e. column names) will be extracted from the first row in the text file. If the file does not contain any header information, this option should be disabled resulting in column names being automatically generated (named 'Col 1', ' Col 2', etc.).
- **Create ID column**: When this option is enabled, an ID column will be added to the dataset with a numeric index shown for each data point / row imported.

The final option makes it possible to add the imported dataset to the current workspace or to replace the current workspace with the new dataset (the dataset is automatically renamed to ensure that datasets in the workspace have unique names).

The **Filtering** tab page allows for dataset filtering during import (see Figure 18).

The **Filter dataset** option makes it possible to limit the total number of rows / data points to import. The data points selected for import are the ones that have the highest or lowest values of a user-defined numerical descriptor.

The **Select subset** option can be used to specify a subset of records to import.

Notice: Both options can be combined so that the filtering based on descriptor values is applied after a subset has been chosen for import.

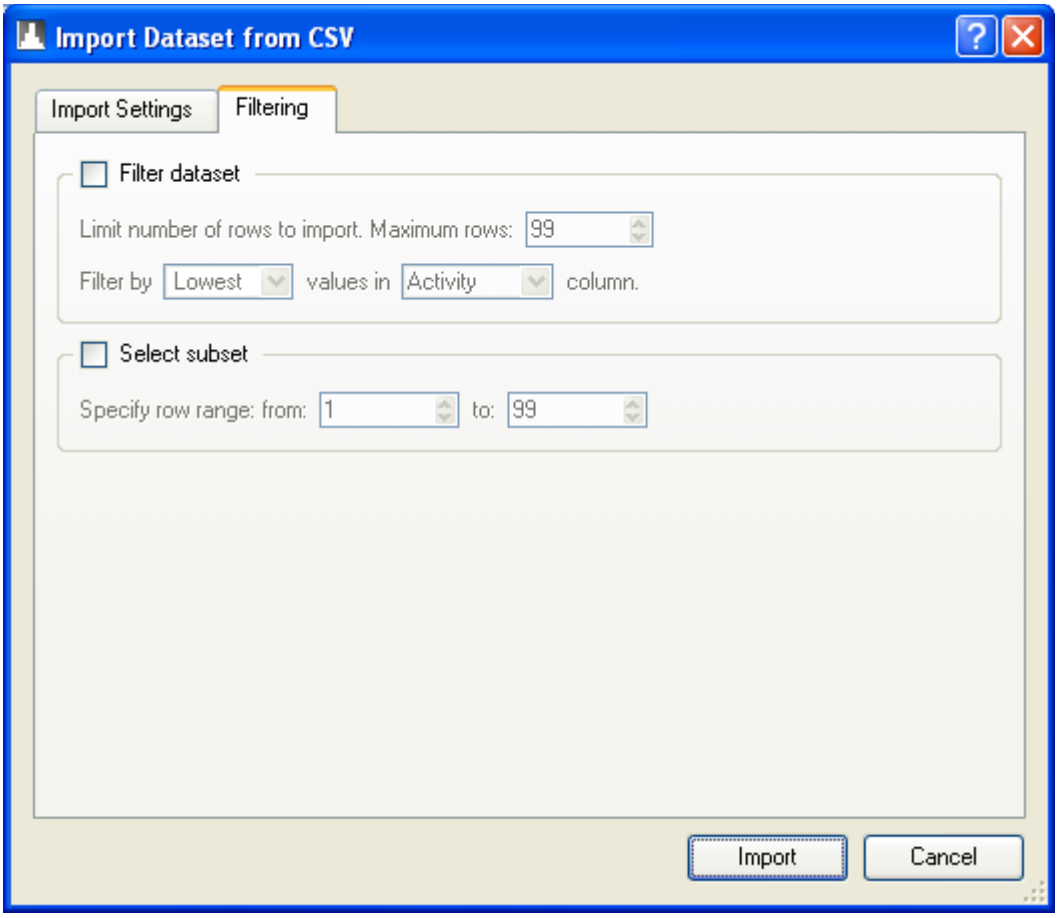


Figure 18: Filtering options available during CSV import.

When pressing the **Import** button, the dataset will be imported using the settings specified in the dialog. If a problem occurs during parsing of the imported file, a **Warning** dialog will be shown.

Typical warnings are:

- Missing value in numerical column.
- Text in numerical column.
- Mismatch between number of columns in header and number of columns imported from data row.

Missing (or invalid) values in numerical columns will be indicated with a red '*nan*' (not a number) label when shown in the **Spreadsheet Window**. Missing values can be removed or repaired (see Section 3.10 for details). Moreover, invalid columns containing one or more invalid cells will not be available during e.g. model creation.

3.3 Importing Workspaces

Workspaces can be imported into Molegro Data Modeller using the **Import Workspace (Datasets/Models)...** menu option located in the **File** menu. To

invoke the dialog box use the **CTRL+SHIFT+O** keyboard shortcut.

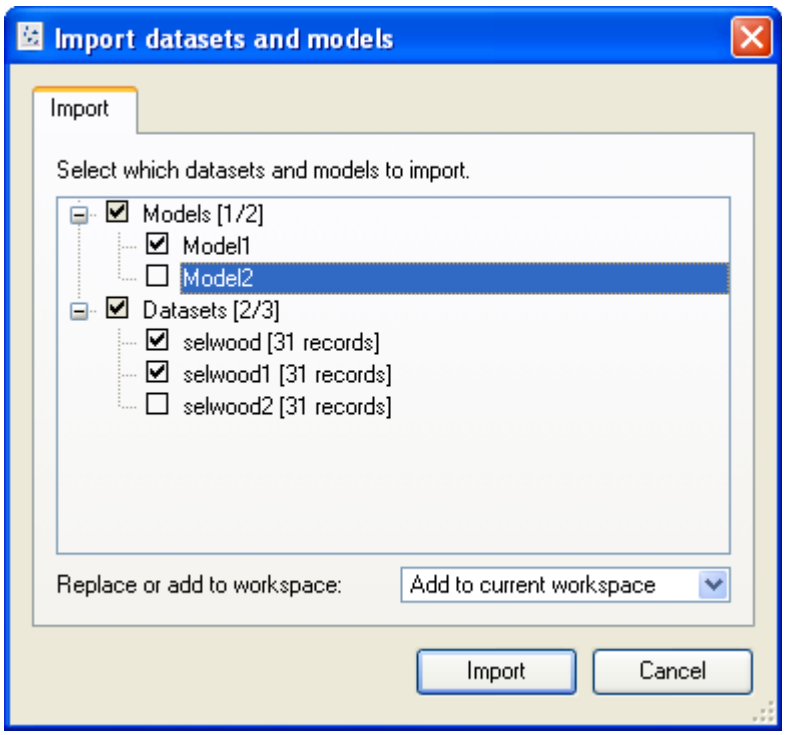


Figure 19: Import Workspace dialog box.

A workspace can contain both datasets and regression/classification models and the user can select which datasets or models to import. It is possible to add the selected datasets/models to the current workspace or to replace the current workspace with the new workspace (datasets and models are automatically renamed to ensure they have unique names).

3.4 Reorder and Delete Dataset Columns

The **Reorder Columns** dialog makes it possible to choose the order of the columns in a given dataset. It is also possible to delete columns by deselecting them in the Column listview.

The dialog can be invoked using '**Edit** | **Reorder Columns**...'.

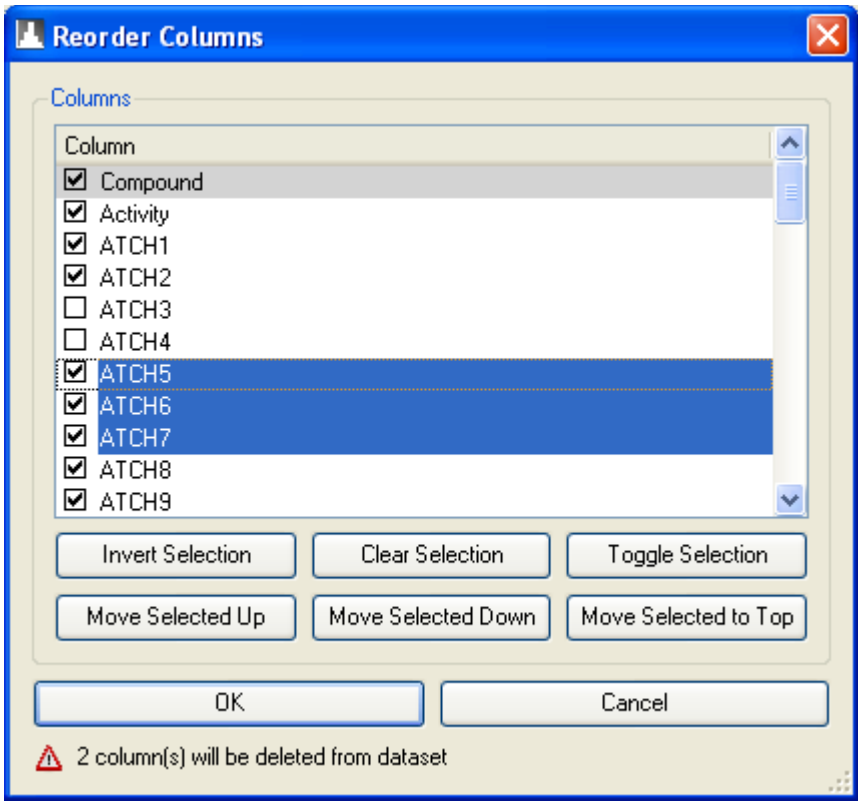


Figure 20: Change the order of the columns in the dataset using the Reorder Columns dialog.

3.5 Dataset Scaling and Normalization

Numerical columns can be scaled or normalized using the **Scale and Normalize Values...** menu option located in the **Preparation** menu. From the **Scale and Normalize Values...** dialog box shown in Figure 21, it is possible to choose a scaling or normalization method and to select which numerical columns the scaling/normalization should be applied to.

Unit variance scaling (UVS) divides each data point with the standard deviation of the specific column. For **Mean centering (MC)**, the mean of the specific column is subtracted from each data point.

Auto Scaling makes it possible to perform both UVS and MC, whereas the **normalization** option normalizes the data points to values between the specified **min** and **max** values.

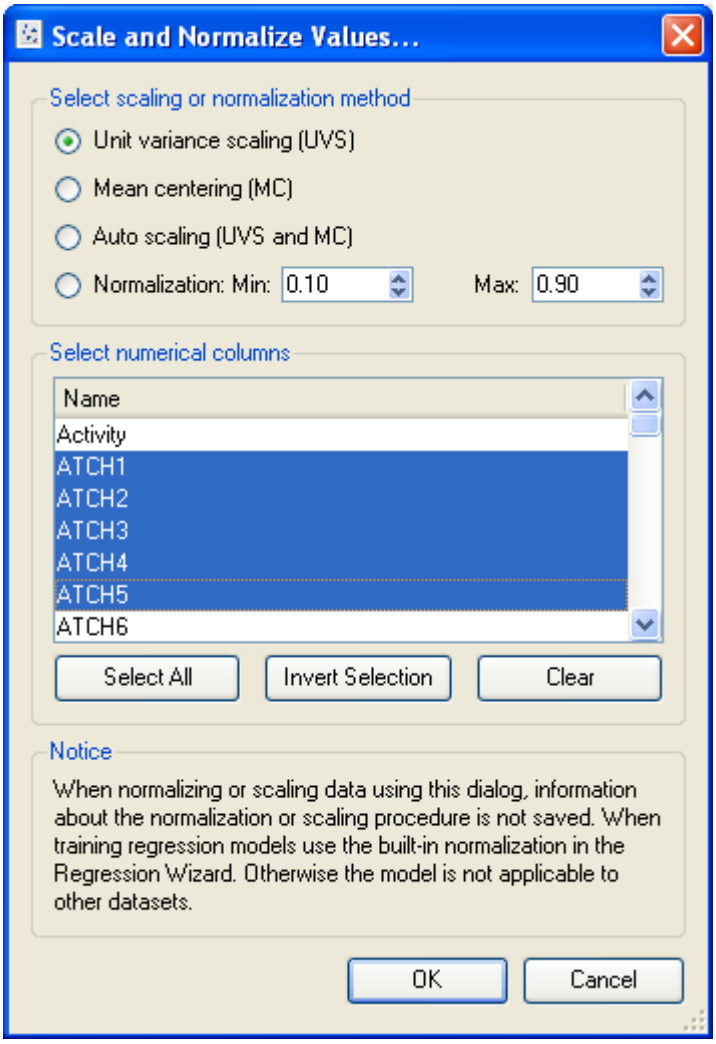


Figure 21: Selected numerical columns can be scaled or normalized.

Notice: It is advisable to perform the scaling or normalization of the dataset in the **Regression Wizard** (introduced in Chapter 8) or **Classification Wizard** (introduced in Chapter 9) since the scaling/normalization applied will be saved as part of the regression and classification models. This will make it possible to use the same scaling/normalization transformation on other datasets that the regression or classification model is applied to without changing the original dataset. If the dataset is modified using the **Scale and Normalize Values...** dialog box, the data transformation done by the scaling/normalization procedure is not saved and cannot be applied to other datasets afterwards.

3.6 Convert Discrete Descriptors

Molegro Data Modeller has a simple tool for converting a discrete descriptor to either **integer representation** or **binary representation**. This can be very useful if class information is provided in textual format (e.g. 'true', 'false') or

an integer-based numerical descriptor should be converted to a binary representation.

For example, using the **Convert Discrete Column** dialog box, the textual descriptor can be converted to a numerical descriptor containing integer values (assigning a unique integer value to each class instance) or a number of numerical descriptors representing a binary representation of the class instances (see Table 1 for an example). The new descriptors (columns) are appended to the dataset.

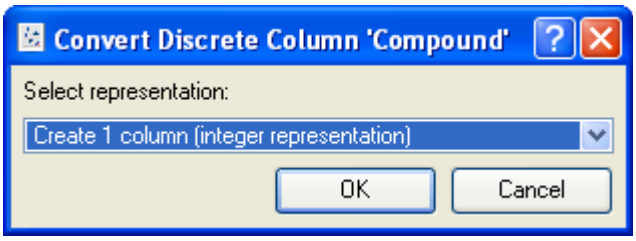


Figure 22: Convert Discrete Column dialog box.

To convert the currently chosen column (marked with boldface in the spreadsheet header), invoke the **Convert Discrete Column** dialog box by selecting **Preparation** | **Convert Discrete Descriptor...**.

Class	Class_Bin1	Class_Bin2	Class_Bin3
Iris-setosa	1	0	0
Iris-setosa	1	0	0
Iris-setosa	1	0	0
Iris-versicolor	0	1	0
Iris-versicolor	0	1	0
Iris-versicolor	0	1	0
Iris-virginica	0	0	1
Iris-virginica	0	0	1
Iris-virginica	0	0	1

Table 1: Example of binary representation of discrete Class descriptor.

3.7 Cross-Term Generator

The **Cross-Term Generator** makes it possible to generate squares and pairwise products of the numerical descriptors in the dataset.

It may be invoked by choosing **Preparation** | **Generate Cross-Terms...** from the menu.

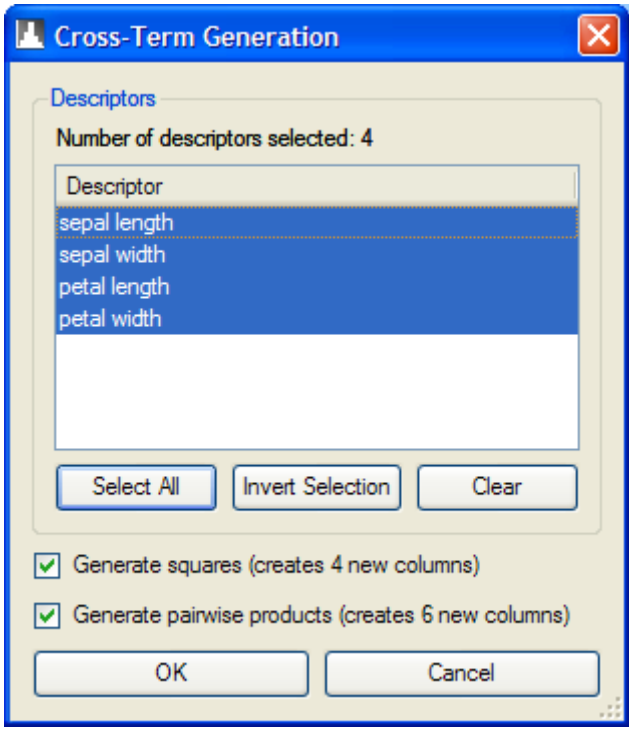


Figure 23: The Cross-Term Generator.

In order to use the generator select the desired descriptors and choose whether to create the squares, the pairwise products, or both. The new columns will be appended at the end of the current data set.

The names of the new columns are automatically generated. For instance, for columns 'A','B', and 'C', the generator will create the new columns 'A*A', 'B*B', and 'C*C' for the squares, and 'A*B','A*C', and 'B*C' for the cross-terms (if needed, the column names are automatically renamed to ensure they are unique).

Cross-terms are usually included in order to account for non-linear terms when doing multiple linear regression. However caution should be taken when adding cross-terms since the complexity of the model is increased, and chance correlation becomes more likely. In general we suggest trying one of the non-linear models (e.g. SVM) before resorting to creating cross-terms.

Cross-terms may however be a valuable tool when trying to uncover relations between the various descriptors in a dataset.

3.8 Convert Between Numerical and Textual Descriptors

It is possible to convert a numerical descriptor to a textual descriptor or a textual descriptor to a numerical descriptor using the **Preparation** | **Convert Descriptor (numerical <-> text)...** option. Converting a numerical descriptor to a textual might be an advantage if the numerical descriptor should not be included in the regression or clustering analyses (for instance if the descriptor represents compound identifiers).

Notice: When converting from a textual descriptor to a numerical descriptor, textual entries representing integers or doubles will be converted automatically whereas non-valid entries will be represented by 'nan'.

3.9 Handling Constant Columns

Descriptors containing the same data value for all records (i.e. constant columns) do not contribute with any valuable information with respect to creating regression/classification models or performing a cluster analysis. It is therefore recommended to remove these columns. To identify constant columns, select **Preparation** | **Select Constant Columns**. All constant columns in the current dataset will be selected and can be removed by choosing **Edit** | **Delete Columns...**.

3.10 Deleting, Replacing, or Repairing Invalid Cells

Invalid record entries (spreadsheet cells) can occur if the imported dataset contains invalid entries such as *NaN* (not a number). Also, invalid cells can occur in the dataset later on if modifications of the data values results in invalid numerical values. For example, dividing entries by zero or taking the logarithm of a negative number in the **Data Transformation** dialog box will result in invalid entries being created.

Numerical descriptors containing one or more invalid cells cannot be used in regression, classification, or clustering analysis. Before using descriptors containing invalid cells, the invalid cells should either be repaired or removed.

Several options are available for repairing invalid cells:

- Manually, repair invalid cells by editing them in the Spreadsheet Window.
- Automatically replace invalid cells with estimated values using the column mean of the specific numerical descriptor that contains the invalid cell(s). Select **Preparation** | **Replace Invalid Cells with Column Mean** to perform this action.
- Automatically replace invalid cells with randomly distributed numbers. It is possible to either use normally distributed values with same mean and variance as the specific column that contains the invalid cell(s) or to use uniformly distributed values using min and max values from the specific

column that contains the invalid cell(s). First, select **Preparation** | **Select Invalid Cells** to select all invalid cells in the dataset (or manually select the ones that should be repaired). Second, select **Preparation** | **Set Selected Cells to Random Distribution** to invoke the dialog box shown in Figure 24.

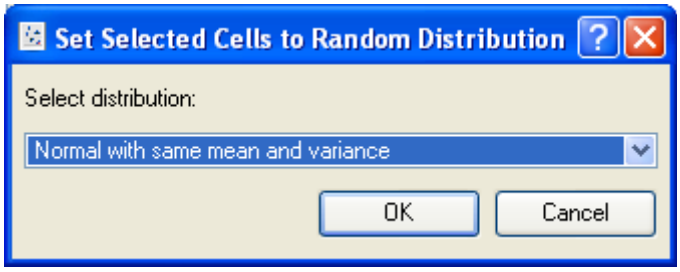


Figure 24: Set Selected Cells to Random Distribution dialog box.

Another solution is to remove the invalid values from the dataset. **Preparation** | **Delete Columns with Invalid Cells** is used to remove all numerical columns containing one or more invalid cells. This is particularly useful if several of the entries in a given column are invalid. In a similar manner, **Preparation** | **Delete Rows with Invalid Cells** can be used to remove all rows (data records) containing invalid cells.

3.11 Scrambling Data Columns

It is possible to scramble selected columns (i.e. shuffle data records) from the **Preparation** | **Scramble Selected Columns** menu option. This option can be useful for e.g. detecting random correlations between the independent variables and the dependent variable, see Section 8.5 for more details.

3.12 Exporting Datasets and Derived Models

To export datasets in **Text CSV** format, select **File** | **Export Dataset...** or **Export Dataset...** from the dataset context menu in the **Workspace Explorer** (by right-clicking on a specific dataset). If one or more *predictions* or *classifications* are present in the dataset, they are by default included in the exported file, but can be excluded by toggling off the **Include regression predictions** or **Include classification predictions** options in the **Export Dataset...** dialog box. Notice that predictions included in **Text CSV** files are parsed as numerical descriptors and not as predictions if imported by Molegro Data Modeller later on. In a similar manner, classifications included in **Text CSV** files are parsed as textual or numerical descriptors and not as classifications if imported by Molegro Data Modeller. To save information about predictions and classifications (e.g. name of model used in prediction, evaluation procedure used, descriptors used in model, correlation coefficient, etc.) the *Molegro Data Modeling* format (MDM) should be used.

The **Export Workspace...** dialog box can be used to export all (or a selection of) datasets, regression/classification models, and predictions/classifications available in the workspace in MDM format (see Figure 25). Notice: The predictions and classifications are not shown in the list since they are associated with the datasets.

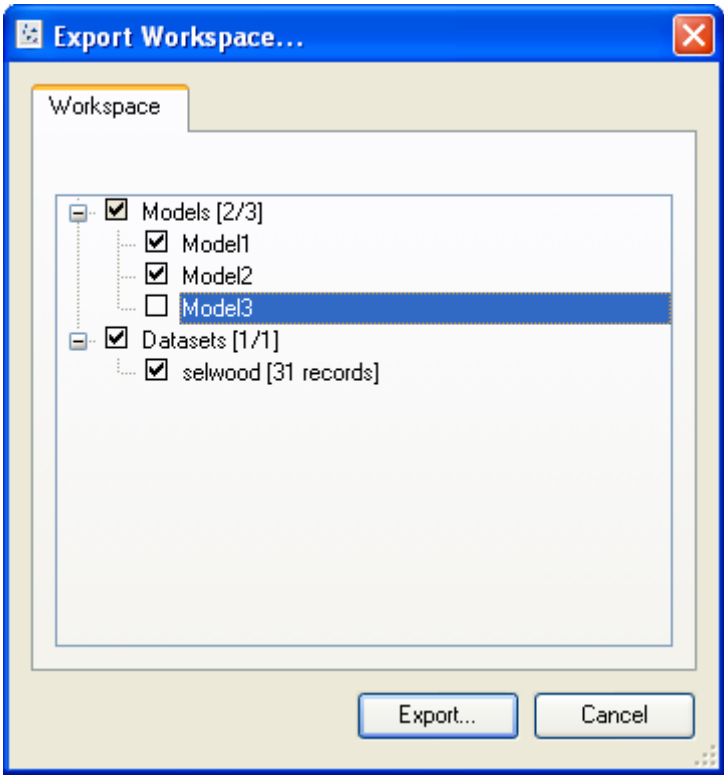


Figure 25: Export Workspace dialog box: Select which models and datasets to export.

The **Export Workspace** dialog box is invoked by selecting **File** | **Export Workspace...**. Alternatively, the **Export Models** dialog box can be used if only regression and classification models should be exported (in MDM format). The **Export Models** dialog box is invoked by selecting **File** | **Export Models...**.

3.13 Creating a New Dataset

New datasets can be created using the **New Dataset...** menu option located in the **File** menu or using the **CTRL+N** keyboard shortcut. From the **New Dataset...** dialog box shown in Figure 26, it is possible to choose a name for the new dataset and the number of columns and rows that the dataset should contain. Notice that only numerical columns are created – textual columns can be added afterwards.

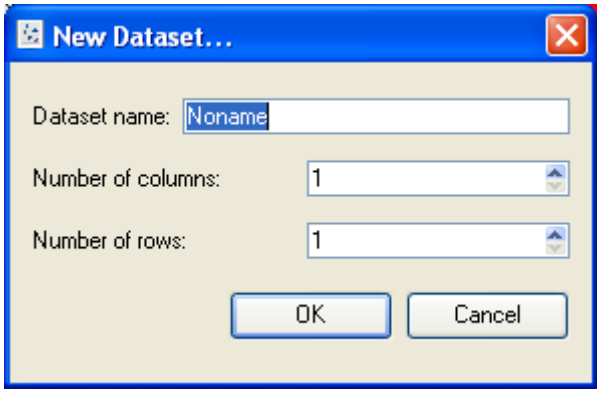


Figure 26: Creating a new dataset.

New datasets can be populated using copy-and-paste from another text source or using the **Data Transformation** dialog box (see Section 3.15 for more details).

3.14 Create New Dataset From Selected Columns

The **Create New Dataset From Selected Columns** dialog may be invoked by choosing **Preparation** | **Create New Dataset From Selected Columns**. The dialog is also available from the Model Details dialog by selecting the **Create new dataset from selected descriptors** button.

This dialog makes it easy to prune the columns in large datasets. It is possible to calculate maximum Pearson correlation coefficients directly from the dialog: either to the selected set of descriptors, or to any descriptor that appear earlier in the Column listview (see figure below). The resulting correlation coefficients are shown in the **Max R^2** column. The **Max Column** column contains the listview index (shown in the **#** column) to the column with the highest Pearson correlation coefficient found for each descriptor.

The calculated correlation coefficients makes it possible to work iteratively when selecting descriptors from a dataset with many descriptors. The dialog is also useful in cases where the dataset has too many columns to display in the Correlation Matrix dialog.

The **Constant (%)** column shown in the dialog, lists the percentage of values that are constant for a given descriptor/column, e.g. a value of '90.3' means 90.3% of the descriptor values are equal. This information makes it possible to prune descriptors that are *nearly constant*.

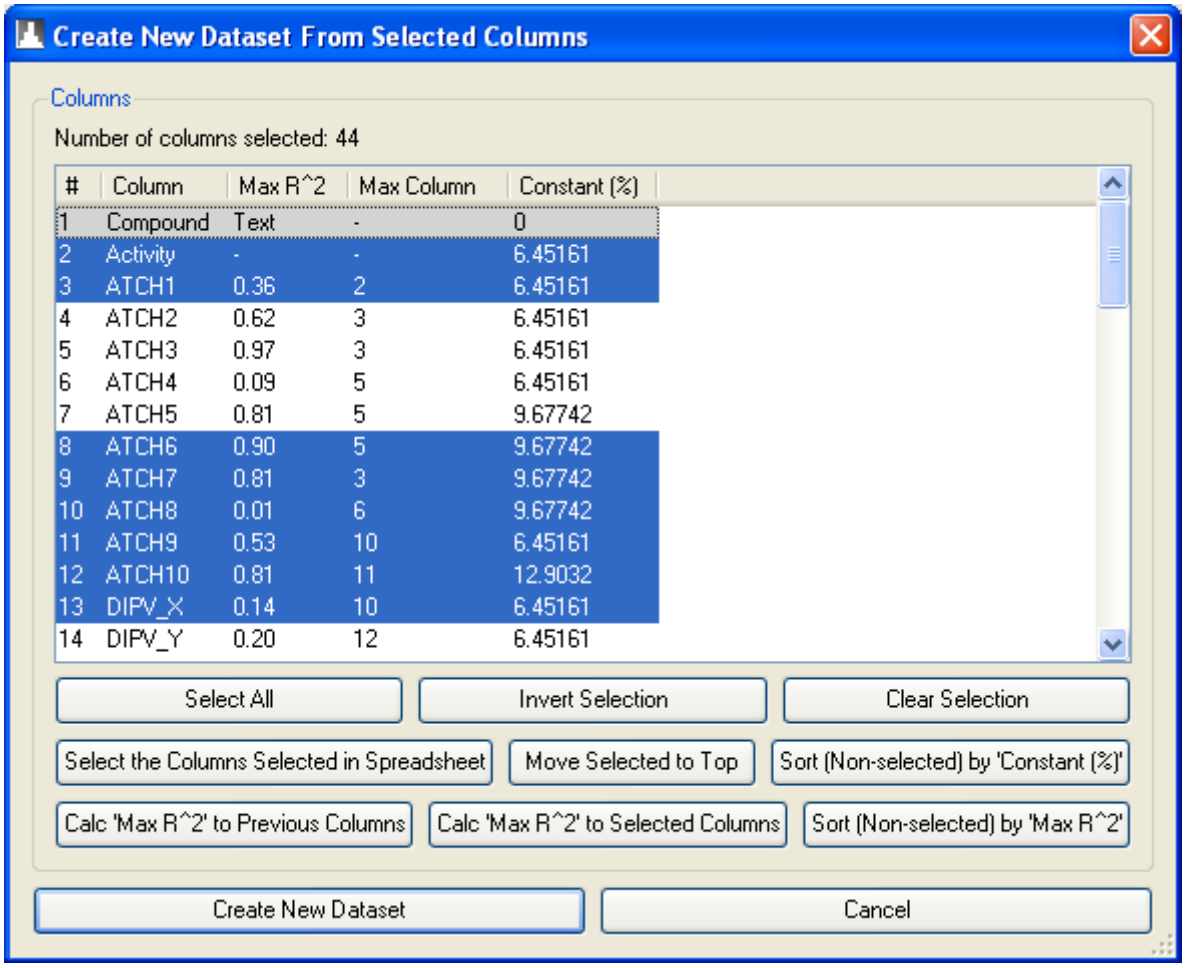


Figure 27: Create a new dataset based on selected columns. It is easy to identify constant or nearly constant columns and to identify correlations between columns using the 'Create New Dataset From Selected Columns' dialog.

3.15 Data Transformation Dialog Box

The **Data Transformation** dialog box is a tool for transforming existing columns and/or creating new columns from existing ones. The dialog box allows the user to specify an algebraic transformation and apply it to one or more columns. To invoke the dialog box select **Modelling** | **Transform Data...** or use the **CTRL+D** keyboard shortcut.

The Data Transformation dialog box is useful for changing the units of a column, or for creating new derived columns from existing ones – for instance it might make sense to try out multiple linear regression on a set of descriptors where new descriptors have been added by transforming existing ones (e.g. by squaring the values).

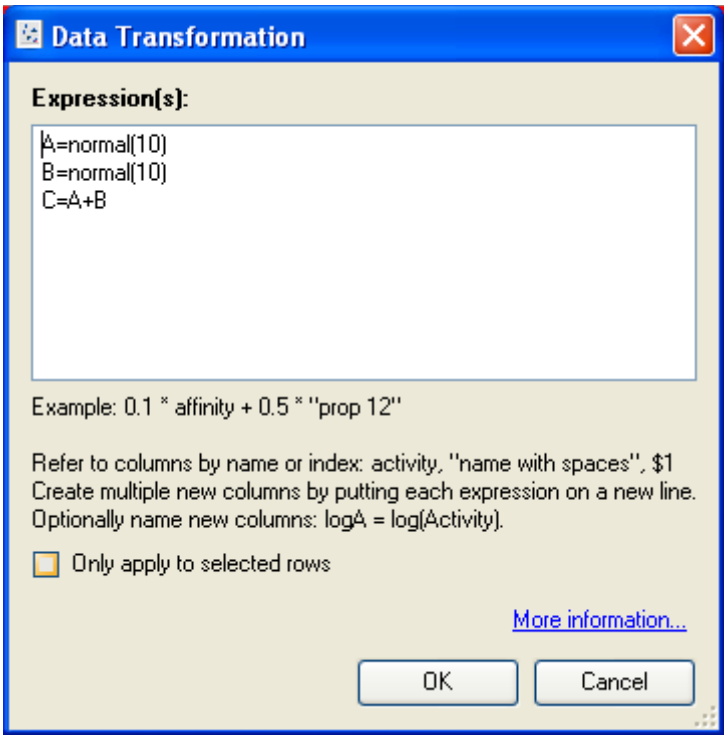


Figure 28: The Data Transformation dialog box.

The upper half of the dialog box is occupied by a text area named **Expression(s)**. Each line in the the text area counts as one single expression.

To transform an existing column, use its name on the left side of the expression, e.g.:

```
Activity = log(Activity)
```

This will replace all values in the 'Activity' column with their natural logarithm. (If the **Only apply to selected rows** check box is checked only rows that are part of a selection in the spreadsheet will be affected)

To create a new column simply use a non-existing name on the left side of the expression, e.g.:

```
NewActivity = log(Activity)+5
```

If a 'NewActivity' column does not exist, it will be created.

It is also possible to refer to columns by their header index instead of their name, e.g.:

```
Sum = $1+$2+$3+$4
```

creates a new column 'Sum' (if a 'Sum' column does not already exist) containing the sum of the first four columns.

Notice that column indices are 1-based (the first column is $1, not $0)

It is also possible to create 'anonymous' columns, by omitting the equal sign:

```
$1*$2
```

will create a new column with the sum of the first and second columns. The system automatically chooses a unique name for the new column.

If a column name contains spaces, it is necessary to enclose the column name in quotes:

```
Diff = "Hydrogen Donors"-"Hydrogen Acceptors"
```

Data Transformation Syntax

The algebraic parser understands the following operators:

+,**-**,*****,**/**,**^**: standard arithmetic operators. '**^**' is the power operator.

<,**>**,**<=**,**>=**,**!**,**==**,**!=**,**&&**,**||**: boolean operators (numbers are interpreted as boolean values as follows: 0 is *false*, everything else *true*).

cos(arg), **sin**(arg): the argument is specified in radians.

exp(arg), **ln**(arg), **log**(arg): **ln** is the base-e logarithm, and **log** is the base-10 logarithm.

rand(max): returns a uniformly distributed number in the interval (-max;max).

normal(var): returns a number from a zero-centered normal distribution with variance 'var'.

abs(arg): returns the absolute (numerical) value of 'arg'.

sigmoid(arg): returns the sigmoid function value of 'arg'.

Min(A,B): returns the minimum of A and B.

Max(A,B): returns the maximum of A and B.

If(condition,ifTrue,ifFalse): evaluates a condition. If the condition is true (different from 0), the function evaluates and returns the 'ifTrue' statement, otherwise 'ifFalse' is returned.

Step(A): returns 0 if A < 0.5 and 1 if A >= 0.5.

Sign(A): returns -1 if A < 0 and 1 if A >= 0.

Refer to columns by their name or by their index (using the $id syntax). Enclose column names containing spaces in quotes. The algebraic parser is not case sensitive.

3.16 Correlation Matrix Dialog Box

Another useful tool for inspecting numerical descriptors is the **Correlation Matrix** dialog box, which can be invoked by selecting **Modelling** | **Show Correlation Matrix**... or by clicking on the Table icon on the toolbar.

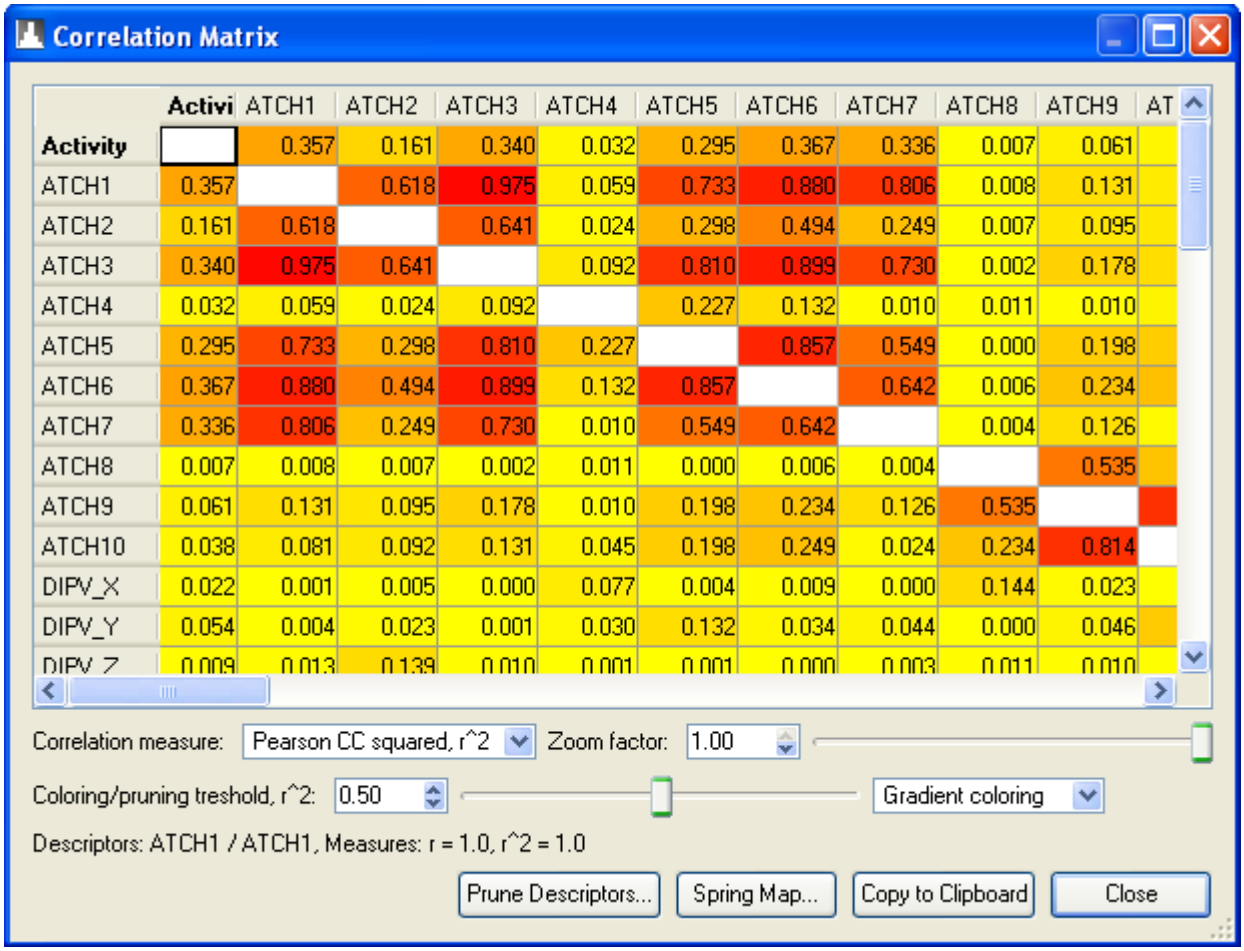


Figure 29: Correlation Matrix dialog.

When invoking the Correlation Matrix dialog the squared Pearson correlation coefficient (r^2) between all pairs of numerical descriptors is shown in the table. From the **Correlation measure** combo box it is also possible to select the non-squared Pearson correlation coefficient (r).

Items with a correlation coefficient above a user-defined threshold (**Coloring/pruning threshold**) can be colored for quick inspection of important descriptors. Using the **Gradient coloring** scheme, a color gradient is shown ranging between low (yellow) and highly (red) correlated entries. Notice: if the non-squared Pearson correlation coefficient measure is used, the absolute value of the entries is compared with the threshold value. The other coloring scheme, **Threshold coloring**, only highlights (red color) entries with values higher or equal to the threshold value. For both coloring schemes, invalid or constant descriptors are indicated by a dark-gray color.

The **Coloring/pruning threshold** is also used when pruning descriptors. After setting the threshold value it is possible to prune descriptors by pressing the **Prune Descriptors...** button. Afterwards, an overview of the descriptors selected for pruning is presented (see Figure 30). The descriptors selected for

pruning are identified in the following manner: First, all invalid or constant descriptors are automatically selected to be pruned. Second, for each descriptor all other descriptors that have a correlation coefficient equal to or above the **Coloring/pruning threshold** are selected for pruning (the descriptors are inspected in the order of occurrence shown in the Correlation Matrix table).

Notice: prediction columns are shown in the Correlation Matrix table but they are not included in the pruning procedure.

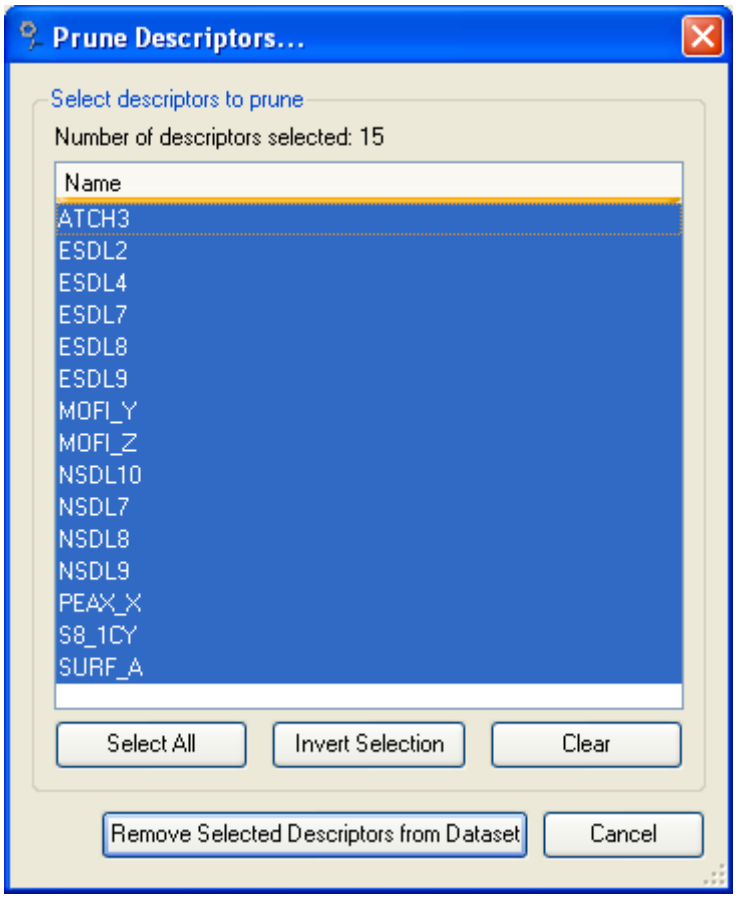


Figure 30: Pruning descriptors using selected correlation coefficient threshold.

From the **Prune Descriptors...** dialog it is possible to manually select which descriptors to prune. To remove the pruned descriptors from the dataset simply press the **Remove Selected Descriptors from Dataset**.

For datasets containing a lot of numerical descriptors it can also be advantageous to zoom out and only focus on the coloring of the table entries indicating regions with high or low correlation. To zoom in or out, simply use the **Zoom factor** spin box or slider and the table entries will resize using the current zoom setting.

By pressing the **Spring Map...** button it is possible to visualize the correlation of the descriptors in the dataset using the Spring-Mass Map method. See Section 5.3 for more details.

Finally, the table entries can be copied to the clipboard by pressing the **Copy to Clipboard** button.

3.17 Bivariate Statistics

The **Bivariate Statistics** dialog is used to explore the statistical relationship between two descriptors.

For numerical descriptors the 2D Plot dialog may be used instead, since the same statistical information is available there, but for discrete classes (such as textual classes), the Bivariate Statistics dialog is able to show classification statistics.

The Bivariate Statistics dialog operates in two different modes depending on the data types for the chosen descriptors. If both descriptors are numerical columns (either a standard numerical column, or a numerical prediction column) it will show the *regression statistics*: Pearson Correlation (and its square), the Spearman rank correlation and the mean square deviation (and its root). These statistics are the same as in the 2D Plot dialog and are described in detail in Appendix I.

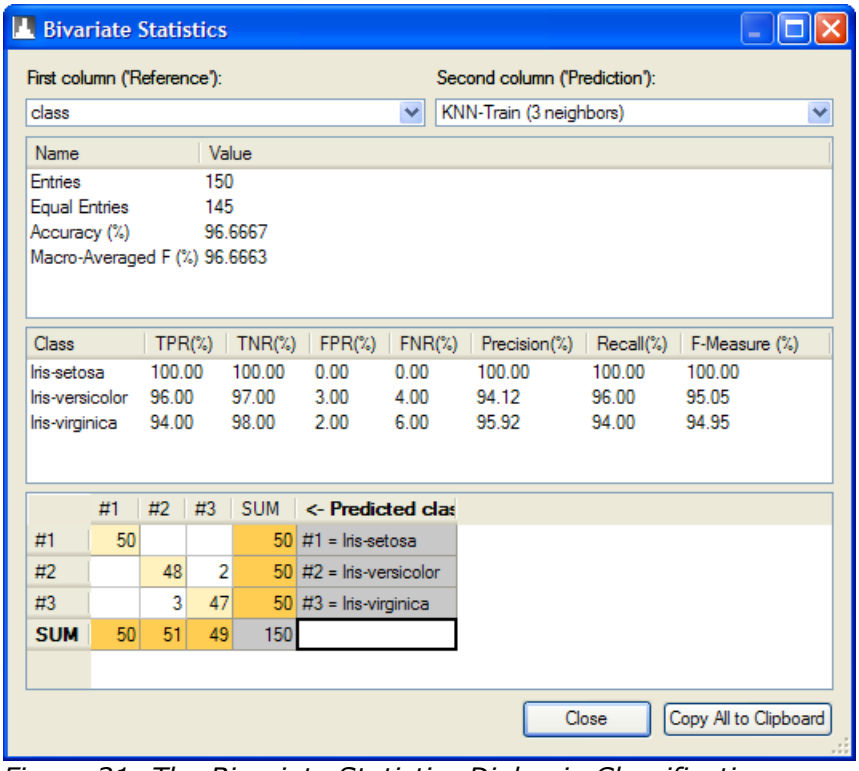


Figure 31: The Bivariate Statistics Dialog in Classification Mode.

If one or both of the descriptors are textual columns (either a standard textual column, or a textual prediction column) the Bivariate Statistics dialog will show *classification statistics* (which assumes that the descriptors are organized in discrete classes): In the upper list box, the number of entries, the number of equal entries, the accuracy and the macro-averaged F-measure are shown. The middle list box shows statistics for the individual classes: the true positive rate (TPR), the true negative rate (TNR), the false positive rate (FPR), the false negative rate (FNR) and the Precision, Recall and F-measure. The F-measure displayed is the weighted harmonic mean of the precision and the recall (sometimes referred to as a F_1-measure or a balanced F-score, because it weights recall and precision evenly). Notice that in order to evaluate the predictive power of a model, the accuracy can be a poor choice: please refer to Appendix I (see under classification measures) for a discussion about this and for formal definitions of the various statistical measures.

The bottom table displays a *confusion matrix*: each of the reference descriptor classes is represented here as a row, and each column corresponds to the predicted class. For a perfect prediction only values in the diagonal would appear. Example: In the confusion matrix on Figure 31, there is a cell with the value '3'. This means that for the class 'Iris-virginica', 3 instances were incorrectly identified as belonging to the 'Iris-versicolor' class.

Notice that for the confusion matrix to be properly interpreted the order of the descriptors is important. The reference descriptor must be chosen in the left list box, and the prediction descriptor must be the chosen in the right list box.

The **Copy to Clipboard** button copies all information in the tables to the clipboard.

3.18 Diversity Statistics

The **Diversity Statistics** dialog is used to explore the diversity of data points in the dataset. The diversity statistics shown for a given dataset is created by measuring the distance from each data point to all other data points in the dataset. It is possible to select which descriptors/columns to include in the distance calculation and which distance (similarity) measure to use: Euclidean Distance, Manhattan Distance, Cosine Similarity, Tanimoto Distance. The distance measures are further described in Section 11.9. Moreover, it is also possible to scale or normalize the dataset using interval-based normalization, auto-scaling or mean-centering.

The diversity results are summarized in three lists: Average Distance, Maximum Distance, and Minimum Distance. These lists contain an entry for each data point. Each of these three lists are again summarized in three columns: Min, Max, and Average. For instance, the 'Average' entry for 'Minimum Distance (All)' refers to the average of the distances to the closest neighbor for each data point.

Furthermore, there are two variants: (All) and (Previous). (All) measures the distance from one data point to all others, whereas (Previous) only measures the distance to earlier data points in the spreadsheet. Notice that the (Previous) measure thus depends on the order of the spreadsheet - and since this order is randomized when calling the Diversity Statistics dialog, the (Previous) measures will vary each time the dialog is invoked.

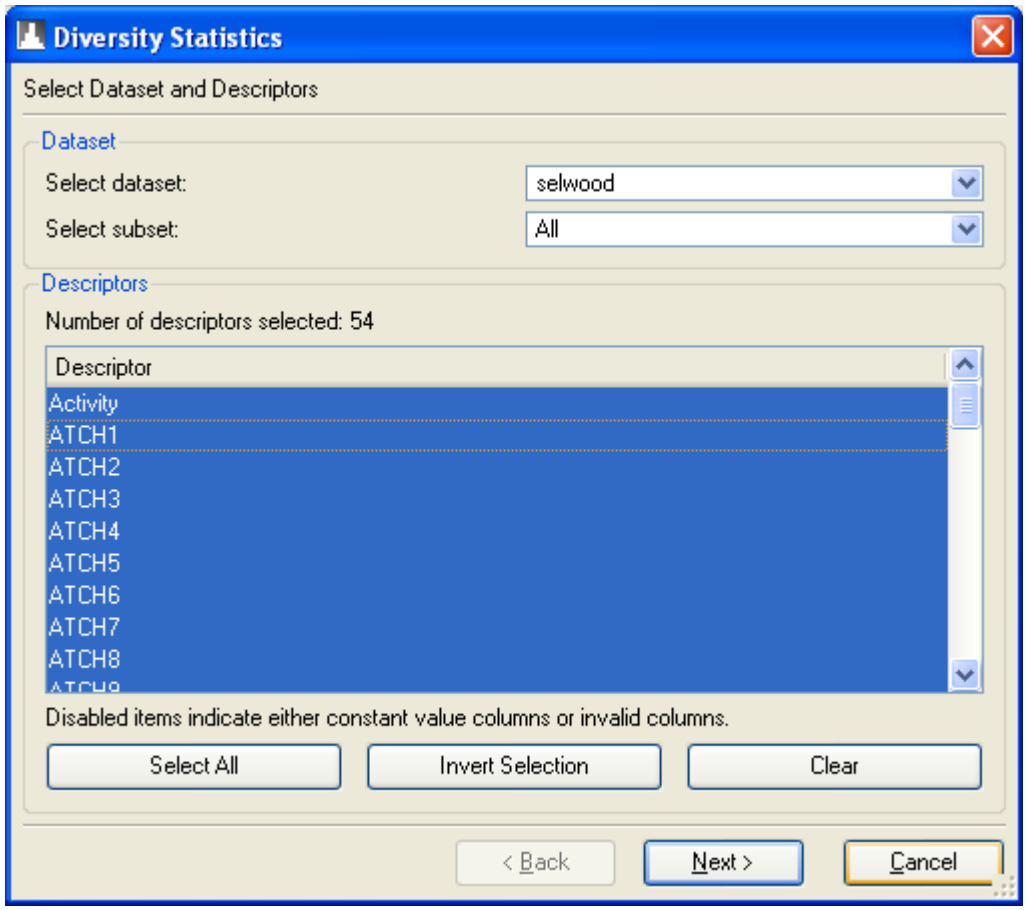


Figure 32: Select which descriptors to include in the Diversity Statistics calculations.

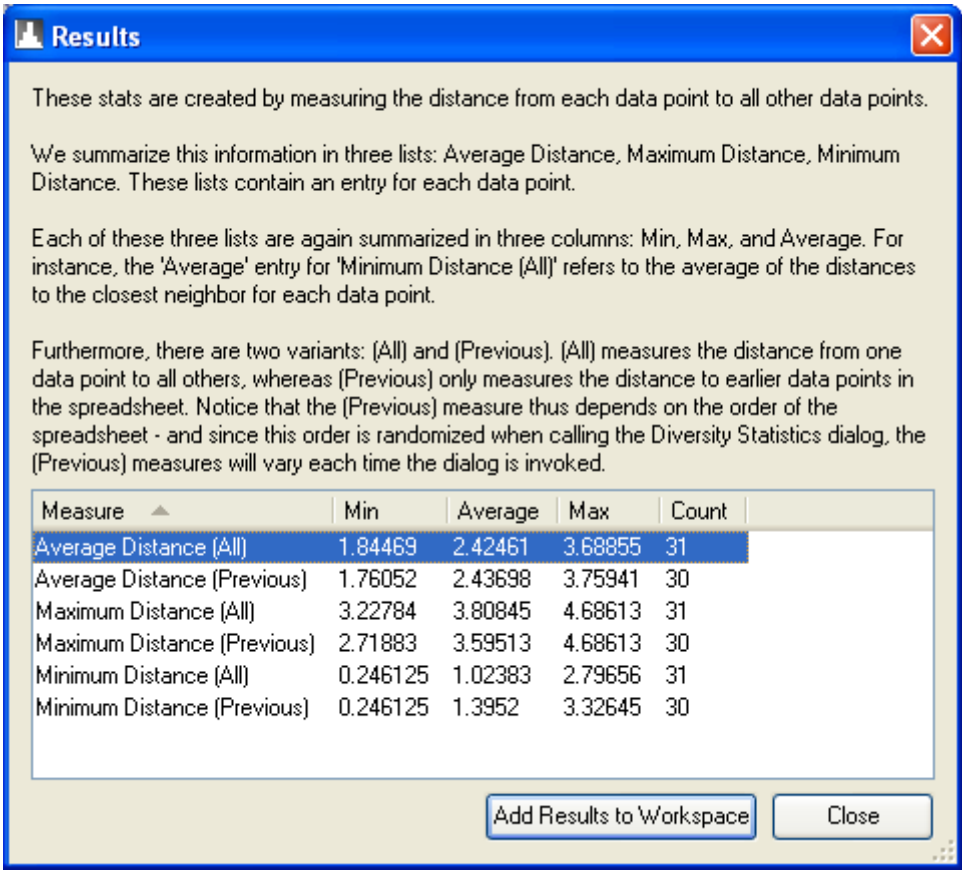


Figure 33: Results obtained by invoking the Diversity Statistics dialog.

3.19 Using Derived Regression or Classification Models

Once a regression or classification model has been created (see Chapters 8 and 9 for details) or imported from a workspace (MDM file) it can be used to predict properties (defined by the model's *target* variable) of other datasets present in the workspace. To make a model prediction, simply invoke the **Make Model Prediction** dialog box from the context menu of the selected model (by right-clicking on the model with the mouse) and selecting the **Make Prediction...** item.

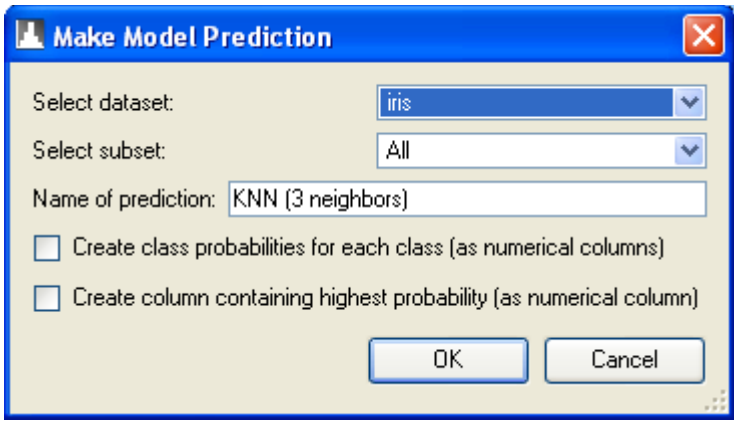


Figure 34: Make Model Prediction dialog box: Select dataset and name of prediction.

In the **Make Model Prediction** dialog box (see Figure 34) it is possible to select which dataset to perform the prediction on and to specify the name of the new prediction column. It is also possible to only predict part of the dataset (using the **Select subset** option) if subsets are available for the dataset. Moreover, the regression or classification model contains all the information needed to perform normalization or scaling of the data values, i.e. the dataset does not need to be normalized or scaled beforehand. Finally, the name of the prediction is automatically altered if another prediction with the same name is present in the dataset (to ensure uniqueness of names). For classification models supporting class probability estimation, it is also possible to create columns containing estimated class probabilities (for each class and the overall highest probability estimated).

Notice that only datasets compatible with the model are listed in the dialog box (a dataset is compatible if it contains numerical descriptors with the same names as those used by the model).

When the *prediction* (numerical predictions for regression models or classifications for classification models) is made (by pressing the **OK** button), it will be available in the dataset. Various statistical information can be inspected using the Bivariate Statistics dialog (see Section 3.17), or by pressing a cell in the prediction/classification column (see Figure 6 for an example).

3.20 Compare Training / Test Set

When training regression or classification models, and applying them to new, independent datasets (here called test sets), it is important to consider whether the test data belongs to the same domain as the training data. If the data points in the test set are located too far away from the training set, it is not safe to assume the model predictions can be extrapolated into that regime.

The **Compare Training / Test Set** dialog makes it possible to measure how similar data points in a test set are to the data points in the training set used

to train a regression or classification model. For a data point in the test set, it can also be used to find a number of nearest neighbors in the training set, and report for instance the training accuracy for these points. If the training accuracy of a neighborhood in the training set is poor, the prediction accuracy of the test data point is likely to be poor as well.

In order to use the Compare Training / Test Set dialog, both the training and test set must be part of the workspace. Invoke the dialog by choosing **Modelling** | **Compare Training / Test Set...**.

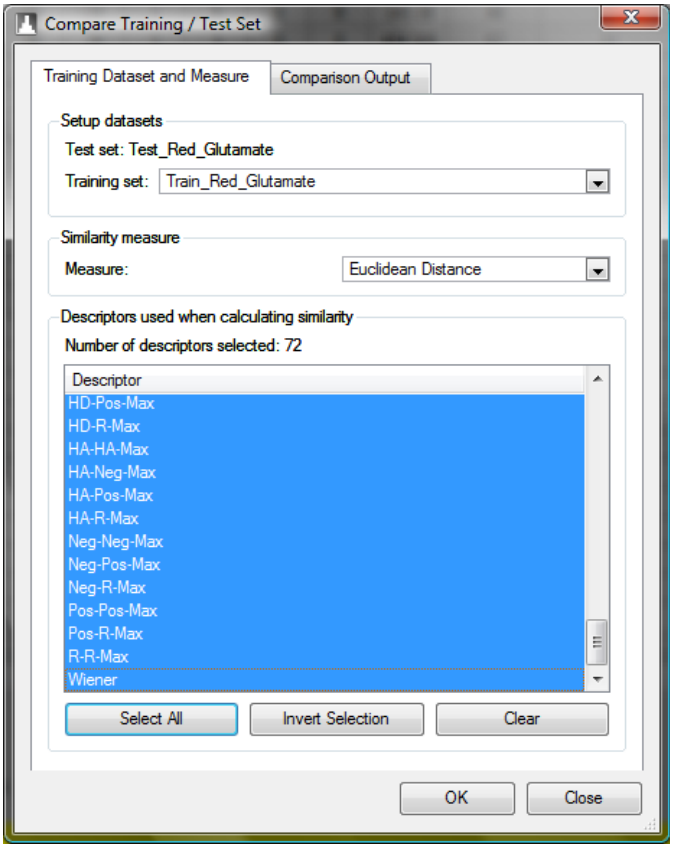


Figure 35: Setting up the similarity (distance) measure.

The first tab, **Training Dataset and Measure**, makes it possible to choose the training set (the test set is always the current data set), and the similarity (distance) measure. It is also possible to choose which descriptors should be used by the similarity measure.

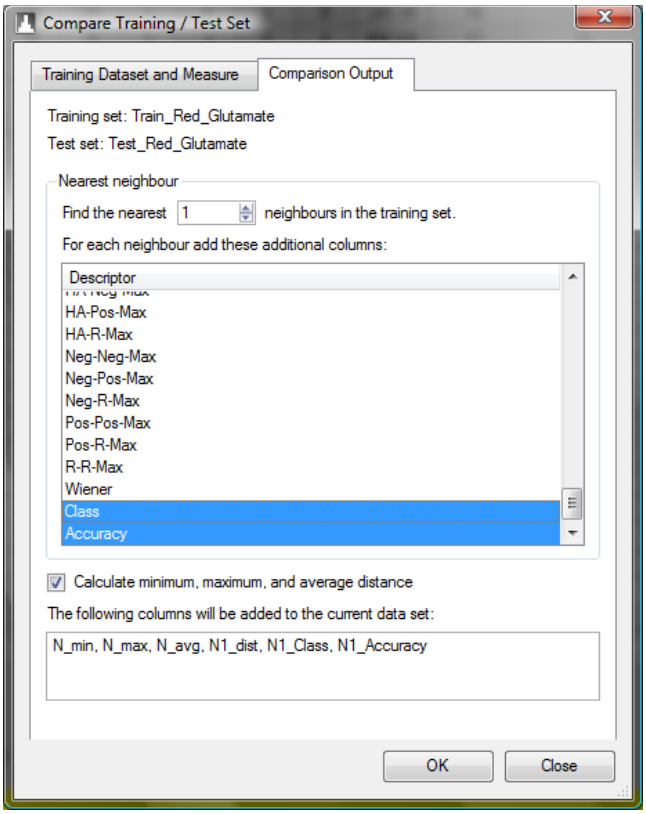


Figure 36: Choosing which information to generate.

The next tab, **Comparison Output**, customizes which information is generated.

From the **Nearest neighbor** panel it is possible to add information for a given number of nearest neighbors in the training set, i.e. for each point in the test set, the closest neighbors in the training set are found, and a column with distances is added to the test set (called N1_dist, N2_dist, ...). It is also possible to add extra information from the training set: for instance it is often useful to add a name or identifier for the data point, and the training accuracy.

The **Calculate minimum, maximum, and average distance** checkbox, makes it possible to add information to the test set about these statistical measures calculated from the data points in the training set. It can thus be identified which data points in the test set, that are most similar to the data points in the training set.

When **OK** is pressed, the comparison is performed, and the necessary number of columns are added to the test set.

3.21 Offline and command line predictions

Normally, when a model is applied to a data set in MDM, the data set is located

entirely in memory.

For very large datasets, it may not be possible to import them into memory. The **Apply Model to External Dataset...** dialog makes it possible to make a prediction using a model in the workspace to a CSV file stored on disk, without importing the CSV file into memory.

Notice that it is not possible to train models on external dataset. For training, the dataset must always be part of the workspace. It is only possible to make predictions on external data.

In order to make predictions on an external file, choose **File** | **Apply Model to External Dataset...**, or choose **Apply Model to External Dataset...** from the context menu on a model in the workspace.

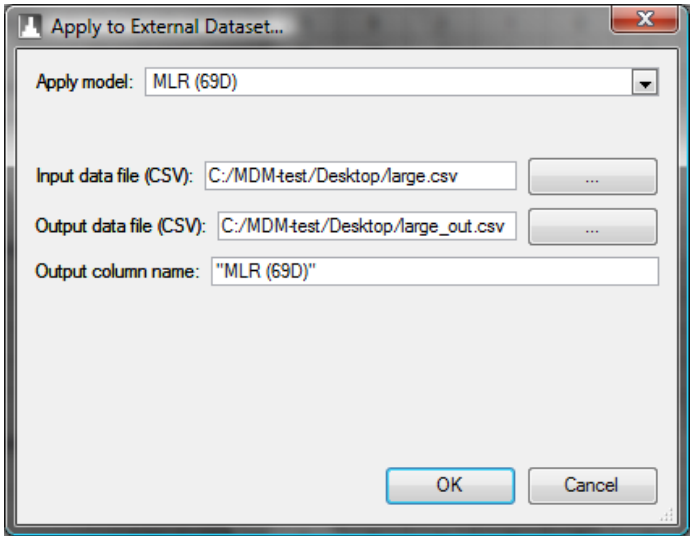


Figure 37: The Apply to External Dataset dialog.

You must choose an input CSV file, an output CSV file, and choose a name for the prediction column. After pressing OK the **Import Dataset from CSV** wizard will appear, making it possible to setup e.g. column formats, text encoding, and separators. It is not possible to apply filtering in this dialog.

After pressing **OK**, MDM will begin processing the external dataset.

Notice, that if the external dataset is too large to fit in memory, it might not be possible to import it to inspect the predictions. However, the **Import Dataset from CSV** wizard makes it possible to filter the dataset while importing it – for instance making it possible to only import a subset with the highest predicted values (see section 3.2 for more information).

Command line predictions

It is also possible to apply models to external datasets using the command

line. In order to do this, store your model in a MDM-file. The workspace should not contain other models. Command line predictions make it possible to integrate MDM in automated workflows. Example:

```
mdm.exe f:\model.mdm -input "f:\my models\input.txt" -output "f:\my
models\output.txt" -outputcolumn prediction
```

The order of the parameters is not important, but the first argument *must* be the model file. Quote any argument containing spaces.

Mandatory arguments

The first argument must be the name of a MDM-file containing exactly one model and no datasets.

-input. Specifies the input file (CSV).

-output. Specifies the output (CSV) file. See also the '-separator' argument.

Optional arguments

-outputcolumn. Specifies the name of the prediction column.

-showGUI [true|false]. Default is true. If false, the log will be written to stdout (only on Linux and Mac).

-debug [true|false]. Default is false. Opens a small log window during processing. Notice, that if an error is encountered the debug window will always automatically open (if showGUI is true).

-overwrite [true|false]. Specifies whether the output file should be overwritten without asking. The default choice is true - thus MDM asks (with a GUI messagebox) before overwriting.

-separator [auto, space, tab, ';', ',', ':']. Specifies the separator for CSV files. Default is auto-detection when parsing CSV, and tab when exporting CSV. Notice that this setting applies to both CSV input and output.

3.22 Recommendations

When preparing a new dataset for regression, classification, or clustering analysis the following actions are highly recommended:

- Remove constant columns (they do not provide any useful information).
- Repair or remove missing values. Depending on the number of missing values present in the dataset either remove the problematic records/descriptors or try to estimate the missing values.
- Remove redundant descriptors. If possible, try to lower the number of numerical descriptors by removing the ones that do not seem relevant (from visual inspection). If some descriptors are highly correlated (e.g.

using the Correlation Matrix), keep one of them and remove the rest. However, it can be difficult to decide which descriptors to focus on. Alternatively, feature selection methods can be applied to identify the most promising ones (see Chapters 8 and 9 for more details).

4 Data Visualization

Numerical and predicted descriptors can be inspected visually using one of the visualization dialog boxes available: **1D Plot** (histogram), **2D Plot**, and **3D Plot**.

4.1 1D Plot Dialog Box

The **1D Plot** dialog box can be invoked by selecting **Visualization**| **1D Plot (Histogram)...** or pressing the histogram icon on the toolbar.

It is possible to select which descriptor to plot and the **Number of bins** slider (or the mouse-wheel) can be used to adjust the number of bins used.

Bins can be selected by pressing the left-mouse button on a bin. When selected, a bin is colored red and the corresponding data points are selected in the spreadsheet.

The context menu (invoked by pressing the right-mouse button on the histogram canvas) offers the following options:

- **Export to CSV**. Saves the histogram data in CSV format.
- **Export to Gnuplot**. Exports the histogram to a Gnuplot script and data file **[GNUPLOT]**.
- **Copy to Clipboard**. Copies the histogram data to the clipboard.
- **Save Screenshot**. Takes a snapshot of the histogram and stores it on disc in either PNG, BMP, or JPEG format.
- **Clear Selection**. (*Requires a current selection*).

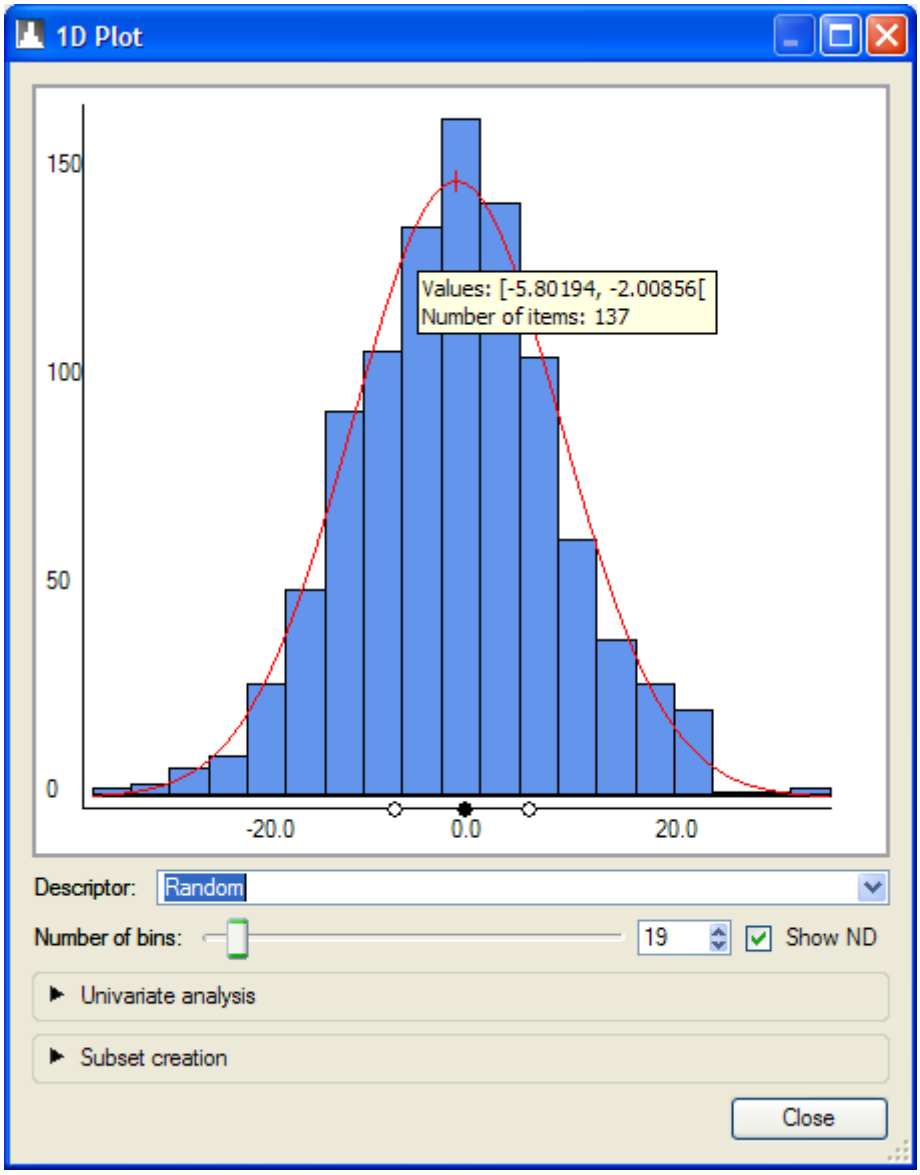


Figure 38: The 1D Plot dialog box.

Univariate analysis listing Range, Median, Mean, Standard Deviation, Skewness, and Excess Kurtosis is provided for the selected descriptor. For more details about the statistical definitions used see Appendix I: Statistical Measures.

The red curve shows an overlayed probability density function for a normal distribution with the same mean and standard deviation as the chosen descriptor. The normal distribution is scaled to cover the same area as the histogram. This makes it possibly to visually inspect if the data samples follow a normal distribution. The normal distribution overlay can be toggled using the **Show ND** checkbox.

Finally, quartile information is provided on the x-axis. The filled circle represents the 50th percentile (median) whereas the two white circles

represent the 25th and 75th percentiles, respectively.

The **Subset creation** options will be introduced in Chapter 7.

4.2 2D Plot Dialog Box

The **2D Plot** dialog box can be invoked by selecting **Visualization** | **2D Plot...** or pressing the scatter plot icon on the toolbar.

It is possible to select which descriptors to plot on the X and Y axes. The plot canvas can be in either *selection* (default) or *zoom* mode. The mode can be changed in the context menu (by pressing the right mouse button on the plot canvas). In selection mode, data points can be selected by left-clicking with the mouse on each data point. Data points within a specific region can be selected by holding down the left mouse button and dragging the mouse. The selected data points in the plot canvas are also selected in the spreadsheet. Further, selections made in the spreadsheet also selects the corresponding data points in the plot canvas.

In zoom mode, the left-mouse button can be used to select a specific region to zoom into (hold down the left mouse button and drag the mouse) and the mouse-wheel can be used to zoom in and out. Numerical data points can be inspected by moving the mouse over the data points.

The context menu offers the following options:

- **Zoom to Fit**.
- **Zoom Out**.
- **Zoom In**.
- **Export** | **Export to CSV**. Saves the 2D plot data in CSV format.
- **Export** | **Export to Gnuplot**. Exports the 2D plot to a Gnuplot script and data file **[GNUPLOT]**.
- **Export** | **Copy to Clipboard**. Copies the 2D plot data to the clipboard.
- **Save Screenshot**. Takes a snapshot of the 2D plot and stores it on disc in either PNG, BMP, or JPEG format.
- **Clear Selection**. (*Requires a current selection*).

The **Jitter** slider can be used to add random noise to the data point positions making it easier to identify overlapping data points. The **Auto Redraw** option continually toggles whether or not jitter is applied to the data points.

The size of the data point circles can be changed using the **Point Size** slider. The **Fill** option toggles whether the circles should be filled or not.

The **Connect** option can be used to connect the data points by drawing lines

between them. The lines are connected using the order of occurrence in the spreadsheet. The **Sort by x** button is used to sort the spreadsheet by the x-axis descriptor (if needed).

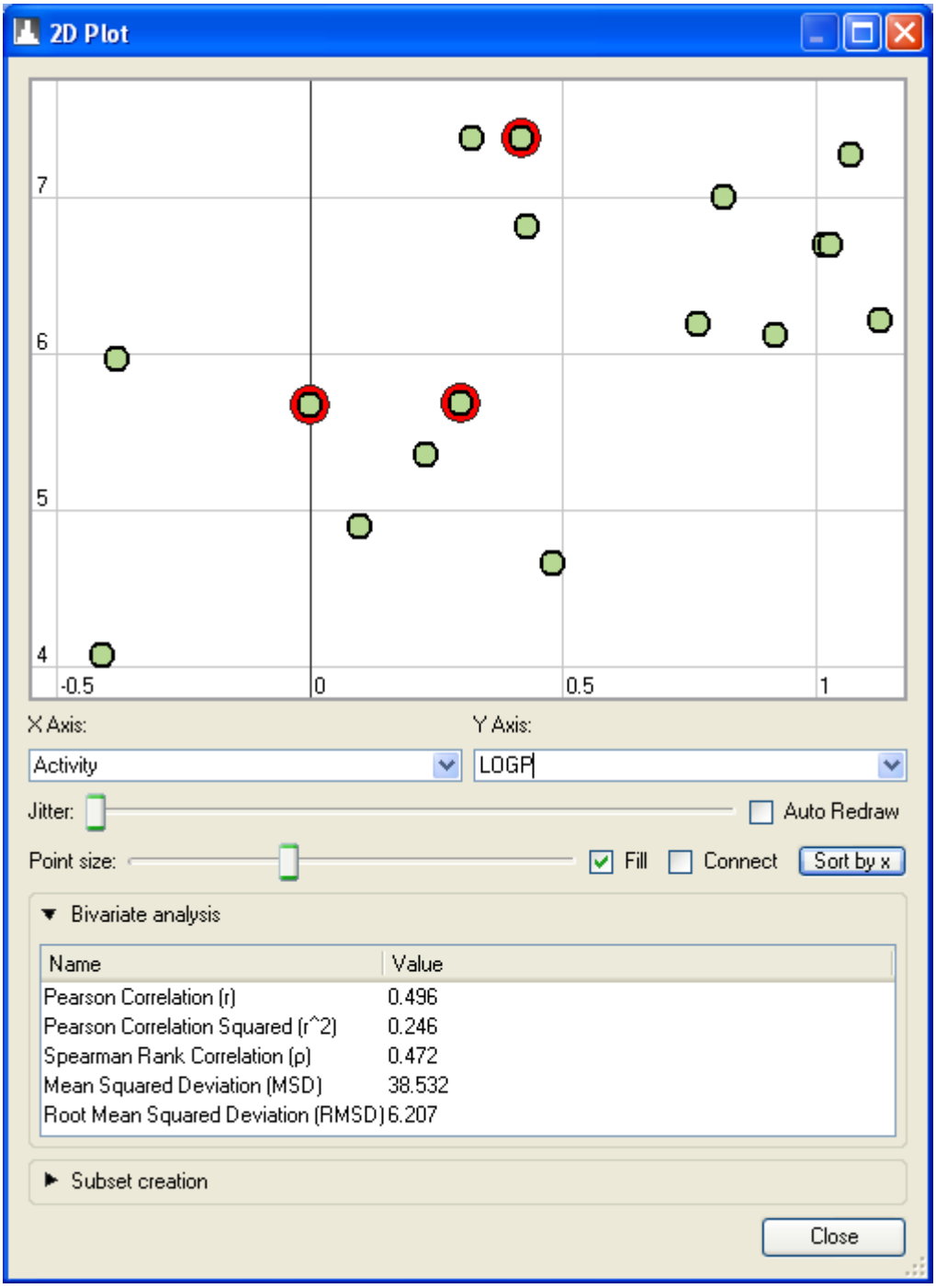


Figure 39: The 2D Plot dialog box.

Finally, bivariate analysis listing Pearson correlation coefficient and Spearman Rank Correlation Coefficient, Mean Squared Deviation, and Root Mean Squared Deviation (see Appendix I: Statistical Measures for more details) is provided

for the selected descriptor.

The **Subset creation** options will be introduced in Chapter 7.

4.3 3D Plot Dialog Box

To invoke the **3D Plot** dialog box, select **Visualization** | **3D Plot** from the main menu, or press the 3D plot icon on the toolbar.

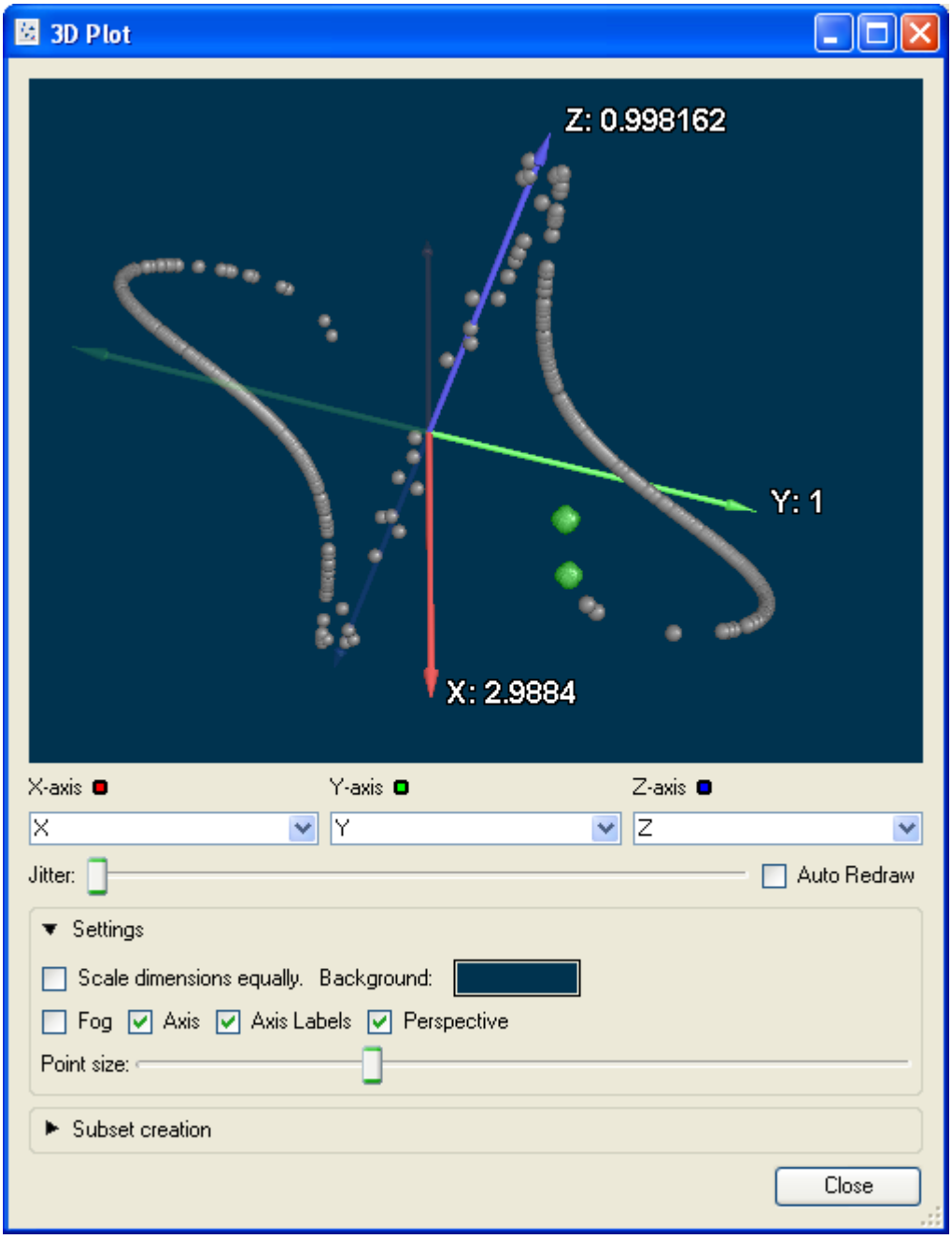


Figure 40: The 3D Plot dialog box.

The 3D Plot dialog box plots the three descriptors (or numerical predictions)

which are specified from the combo boxes at the center of the dialog box. The **Jitter** slider can be used to add random noise to the data point positions making it easier to identify overlapping data points. The **Auto Redraw** option continually toggles whether or not jitter is applied to the data points.

Navigating in the 3D View

The following mouse actions are available in the 3D world:

Function	Action
Zoom	Press both mouse buttons and moving up and down. Use scroll wheel. Use shift and left mouse button.
Free Rotation	Drag mouse cursor while holding down left mouse button.
Drag Rotation	Drag mouse (left mouse button down) while holding mouse over a data point. This will force the data point to follow the mouse cursor.
Translation	Drag mouse cursor while holding down right mouse button.
Show Context Menu	Click and release right mouse button.

All rotations are centered about the rotational center which can be changed using the context menu (see below).

The context menu offers the following options:

- **Zoom to Fit**. Scales the 3D objects to fit the window.
- **Zoom Out**.
- **Zoom In**.
- **Look Down Z-Axis**.
- **Set as Pivot Point (rotational center)**. (*Requires the mouse to hover on a data point*). The selected data point will be the center for all mouse rotations.

- **Clear Selection**. (*Requires a current selection*).
- **Save Screenshot**. Takes a snapshot of the screen and stores it on disc in either PNG, BMP, or JPEG format.

By clicking on the **Settings** toggle box, it is possible to adjust the visual appearance. The following options are available:

- **Scale dimensions equally**. If the dimensions are scaled equally, the units are the same on each axis: therefore, if a selected descriptor spans a smaller interval than another descriptor it may be difficult to see its variations. By default all dimensions are graphically normalized to equal sizes in 3D space.
- **Background**. Sets the background color of the 3D view.
- **Fog**. Enables depth cuing by fading distant objects.
- **Axis**. Toggles the visualization of the axes on and off.
- **Axis Labels**. Toggles axis labels on and off.
- **Perspective**. When perspective is enabled, distant objects appear smaller than objects closer to the viewer. When disabled, objects appear the same size independent of the distance from the viewer (this is sometimes referred to as *orthographic projection*).
- **Point size**. Sets the point size. Notice: If the point size is set to the minimum value, data points will no longer be drawn as spheres made of polygons, instead each data point will be drawn as a pixel point. This is much faster for large datasets. The plotter will automatically default to this drawing mode for datasets with more than 10,000 points.

Notice that it is possible to select a data point in the 3D view by clicking on it. The selection also selects the corresponding row in the spreadsheet. This makes it possible to easily remove outliers by graphically inspecting a dataset. It is not possible to select data points if the **Point size** is set to minimum size. Further, selections made in the spreadsheet automatically selects the corresponding data points in the 3D view.

The **Subset creation** options will be introduced in Chapter 7.

4.4 Changing Colors in 2D and 3D Plots

By default data points in the 2D and 3D Plots are colored gray. However, it is possible to change the coloring of data points using the **Color by Descriptor** dialog box introduced in Section 2.7. The data points will inherit the colors defined by the current color scheme used for coloring the active spreadsheet. Figure 41 shows an example of a gradient coloring style visualized in the 2D Plot dialog box.

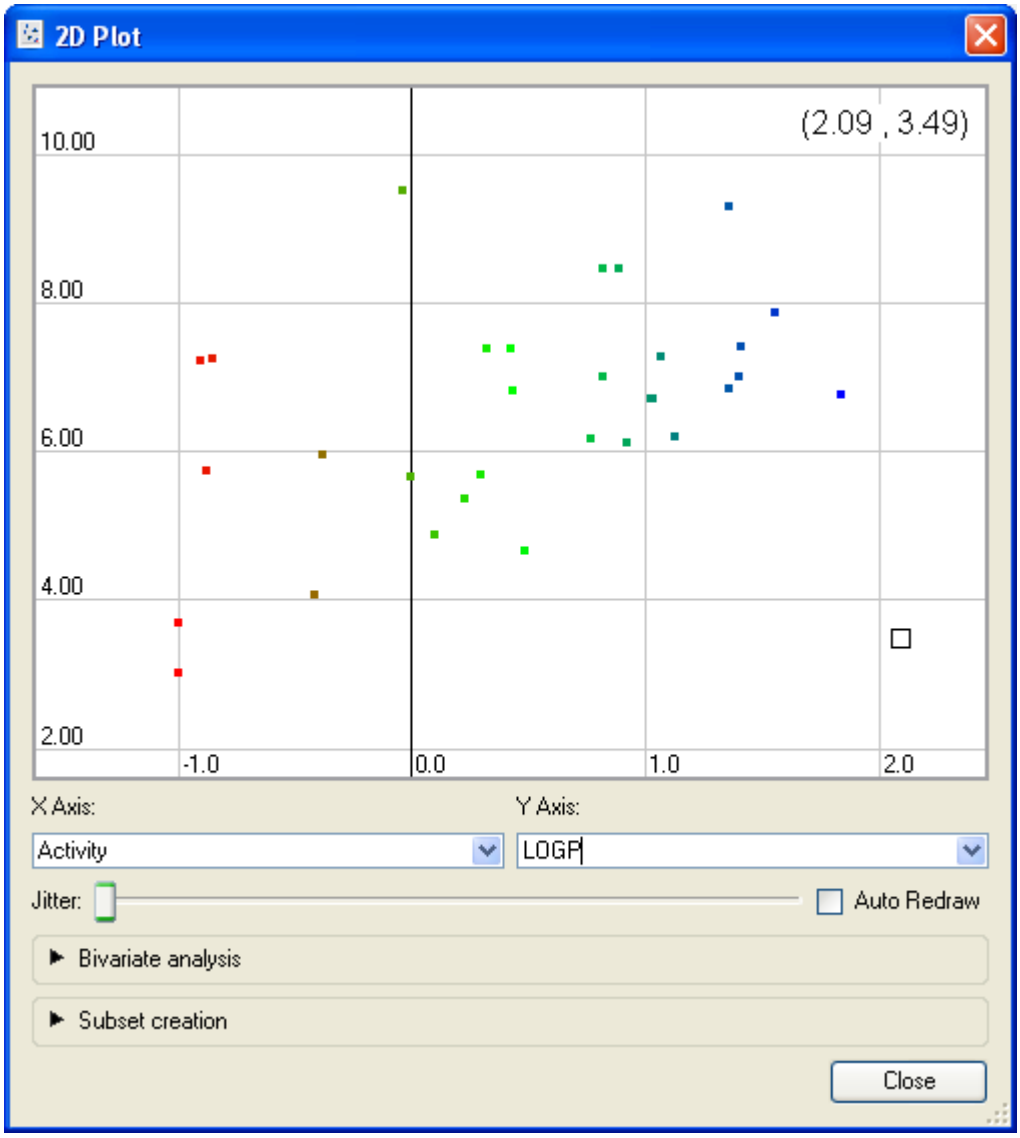


Figure 41: Example of gradient color scheme (using 'Activity' descriptor).

5 Visualizing High-Dimensional Data

The purpose of high-dimensional data visualization is to depict multi-dimensional datasets in two or three dimensions while trying to preserve the underlying structure from the original dataset.

In general it may not possible to make a perfect mapping from a higher-dimensional space to a lower-dimensional space without distorting the structure of the original dataset. However, visualizing the data in a lower-dimensional space may still be useful for identifying clusters or spotting outliers.

5.1 The Spring-Mass Map

The **Spring-Mass Map** model in Molegro Data Modeller offers a simple and intuitive method for reducing the number of dimensions in a given dataset:

1. A distance metric is defined and it is chosen which columns should be included when calculating the distances. The available metrics are the same as those used by the Similarity Browser (Euclidean Distance, Manhattan Distance, Cosine Similarity, Tanimoto Distance described in Section 11.9).
2. A distance matrix containing the distances between every pair of data points in the dataset is constructed using the measure chosen above.
3. A simple physical model inspired by the physics of a network of springs is used to map the distances. In this model each data point is modeled as a point mass being connected to all other data points by a spring model, where the spring equilibrium distances are defined by the distances in the distance matrix. These springs will exercise a force proportional to the displacement from the equilibrium distance: $\mathbf{F}(x)=-kx$, where x is the deviation from the equilibrium distance and k is an arbitrary spring constant. This system is then solved in order to find the state with the

lowest energy (the state with the lowest spring tension).

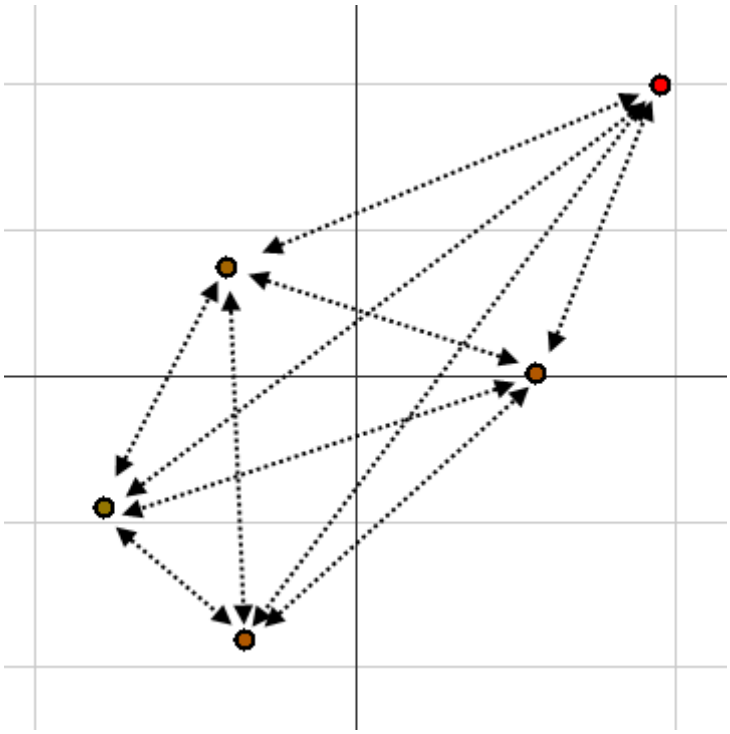


Figure 42: The Spring-Mass Model. Each data point is represented as a point mass connected to all other data points by a network of springs. The equilibrium distance for a given spring is equal to the distance between the data points in the higher-dimensional space.

5.2 The High Dimensional Visualization Dialog

To invoke the **High Dimensional Visualization** dialog choose '**Visualization | Visualize High-Dimension Data (in 2D/3D)...**'

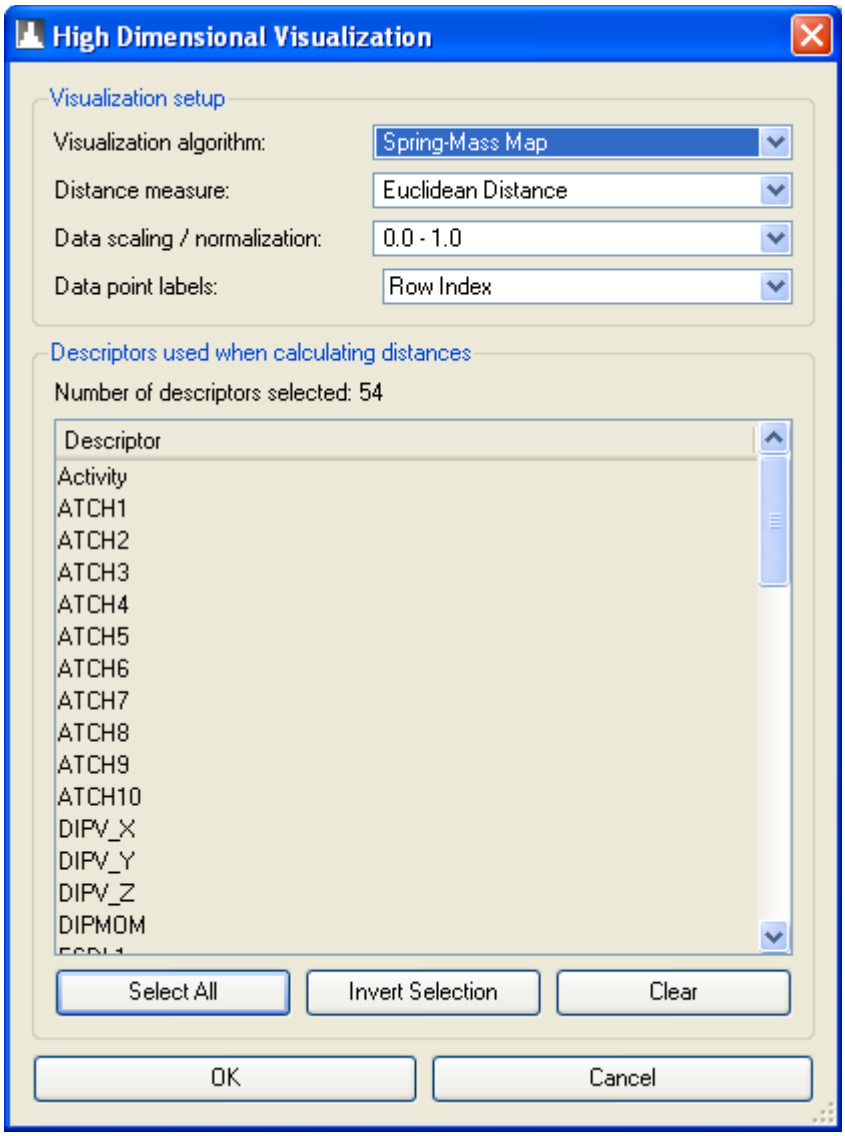


Figure 43: The High Dimensional Visualization dialog.

The dialog box makes it possible to setup the distance measure and presentation. The following parameters can be adjusted:

Visualization algorithm: as of now only **Spring-Mass Map** can be selected.

Distance measure: the distance in the high-dimensional space is calculated using this measure. The measures are described in more details in Section 11.9).

Data scaling / normalization: scaling or normalization can be applied (interval, auto-scaling or mean-centering) to the dataset before the visualization is created (if the dataset has not been normalized beforehand).

Data point labels: in order to identify the data points when depicted in two or three dimensions a label can be assigned. The label can either be the row index in the spreadsheet or the value of either a textual or numerical

descriptor.

Descriptors used when calculating distances. This list view controls which descriptors are taken into account when calculating the distance measure.

After pressing **OK** a visualization window is displayed:

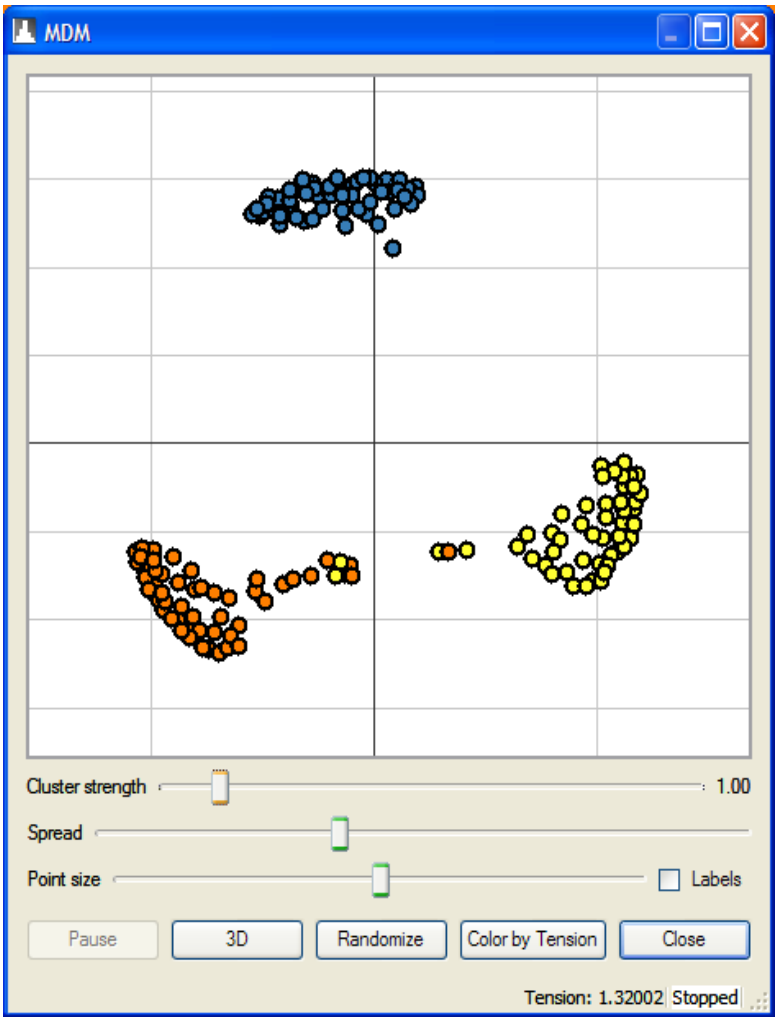


Figure 44: The Spring-Mass map.

While the simulation is running and the positions are being adjusted in order to minimize the tension of the system, the display is being continuously updated. It is possible to pause the simulation using the **Pause** button.

It is also possible to randomize the positions of the data points and restart the simulation using the **Randomize** button – since it is not always the case the system finds the optimal equilibrium configuration, it can be useful to randomize the system and run the simulation several times to see how robust the equilibrium configurations are with respect to the initial conditions.

The **Color by Tension** button determines how the points are colored. If this button is not enabled, the coloring from the spreadsheet will be used. If the

button is enabled, the data points are colored according to their spring map tension, where the color of the data points indicates how far from an equilibrium the data point is. Red data points are far from an equilibrium configuration indicating that the simulation was not able to map the red data points in a way where the distances from the higher-dimensional space were preserved. Green data points indicate that the data points are close to an equilibrium configuration thus more reliably reflecting the structure of the higher-dimensional space.

The **Spread** slider makes it possible to avoid overlapping data points. By turning on a spread penalty, data points will be subject to an additional repulsive force between them. This makes it possible to get an idea of the size of a cluster of points by spreading out the data points, which otherwise would just appear on top of each other.

The **Point size** slider adjusts the size of the data points in both 2D and 3D. If the **Labels** checkbox is checked, each data point will be labeled according to the label chosen in the High Dimensional Visualization dialog box. The **3D** button can be used to toggle between 2D and 3D visualization.

It is possible to manually drag data points (in the 2D view only). This can be used to explore the stability of the equilibrium configurations. It is also possible to select data points in both 2D and 3D. Selecting a data point in the visualization window will select the corresponding row in the spreadsheet and vice versa.

The **Cluster strength** slider can be used to emphasize the clustering of the dataset. If the cluster strength is set to low values, the data points will be more equally spaced. If the cluster strength is increased to higher values, the clustering of the dataset will be more exaggerated. (The cluster strength is implemented as a transformation of the distance matrix. When the distance matrix is created, the entries are calculated from the chosen metric, and the entries are normalized to be within [0;1]. However, the value used as equilibrium distance in the spring-mass model is calculated as d^c where *d* is the distance calculated by the metric, and *c* is the cluster strength. Thus a cluster strength of 1.0 corresponds to an unmodified distance matrix.)

5.3 Visualizing the Correlation Matrix

It is possible to visualize the structure of the Correlation Matrix using the Spring-Mass Map model. This can be done by invoking the correlation matrix, and pressing the **Spring Map...** button. As a measure of the distance between the different descriptors, one minus the Pearson correlation coefficient (r) is used. Descriptor columns with high correlation will therefore be positioned close to each other.

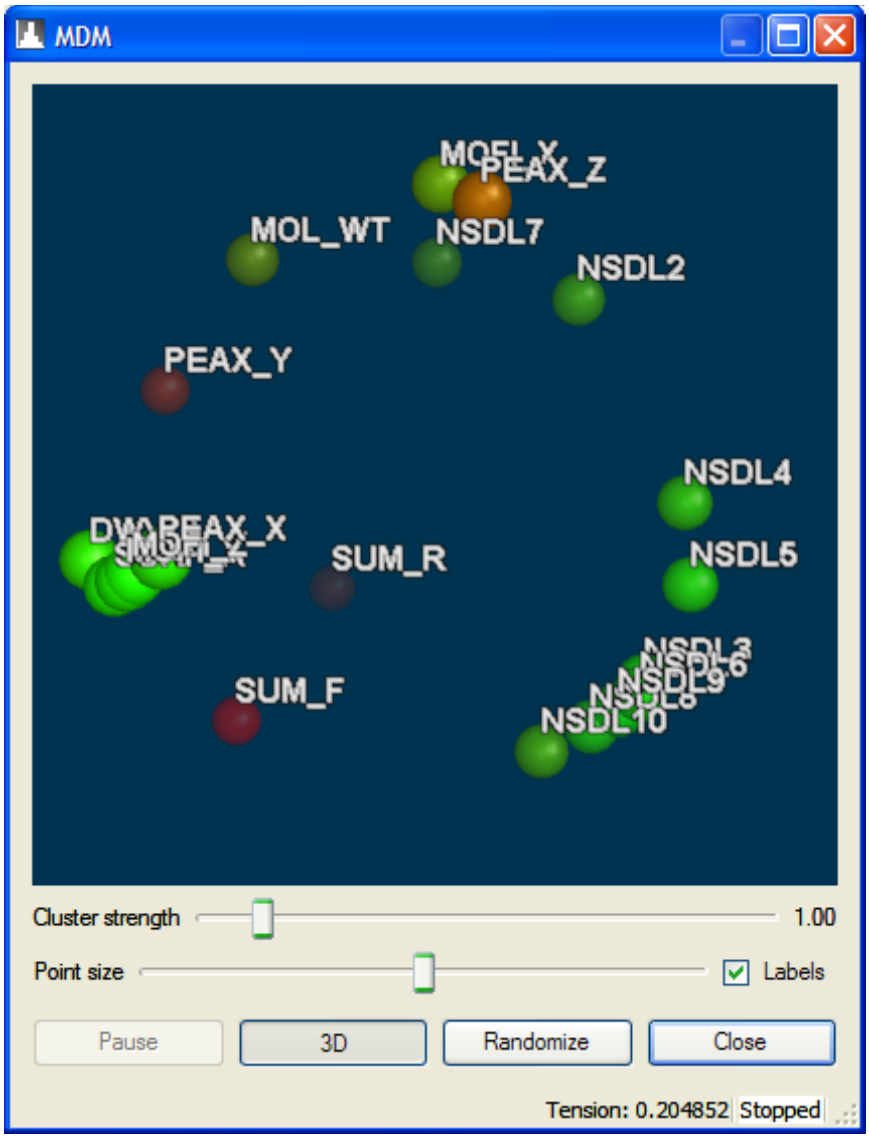


Figure 45: Visualization of the correlation matrix. Highly correlating descriptors are shown close to each other.

6 Outlier Detection

Outliers are observations that are numerically far away from other observations in the dataset.

In practice it is difficult to decide how far away a point must be before it is considered abnormal - and it can be a scientifically questionable practice to delete outliers from a dataset. In particular it can be difficult to identify outliers if the dataset is small or if the underlying distribution is not known. However, outlier detection can be useful for identifying faulty data: for instance data errors occurring because of data collection or data processing faults.

Some regression methods (in particular Multiple Linear Regression) are very sensitive to outliers. Even a single erroneous entry may have a large impact on predictions made by a Multiple Linear Regression model.

Often outliers can be identified by visually inspecting the data, but MDM also provides more sophisticated methods for detecting outliers. This is especially convenient when working with datasets with many descriptors where manual inspection can be tedious and time-consuming.

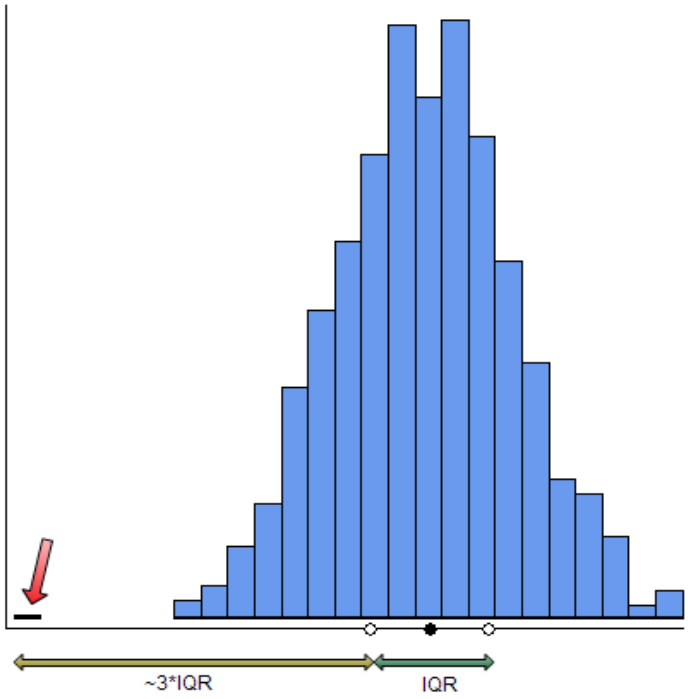
MDM provides two methods for outlier detection: a quartile-based method, and a density-based method.

6.1 Quartile-Based Method

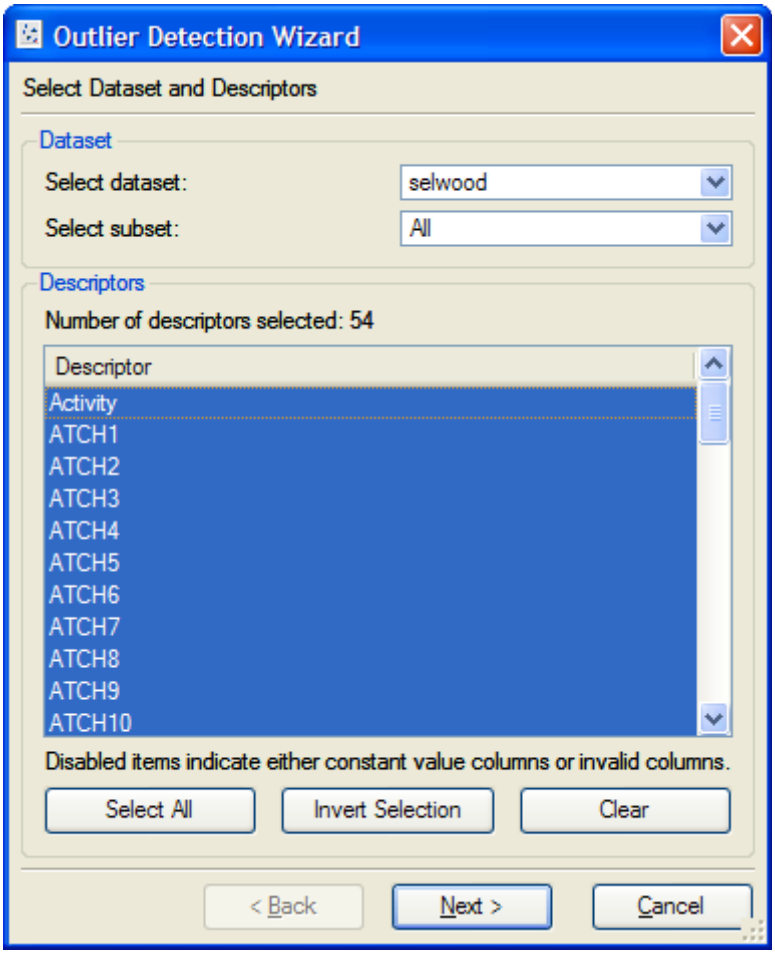
In statistics the q*uartiles* are three numbers dividing the dataset into four equally sized partitions, i.e., the first quartile splits the dataset at the lowest 25% of the data, the second quartile (the *median*) divides the dataset into two equally sized halves, and the third quartile splits the dataset at the highest 25% of the data.

The quartile-based method for outlier detection works by inspecting each

descriptor independently: for each value in a descriptor column, it is determined if the value is outside the region between the first and third quartiles. If it is outside this region, it is given a score equal to the distance to the closest quartile in units of the *Inter-Quartile Range* (see Figure 46).

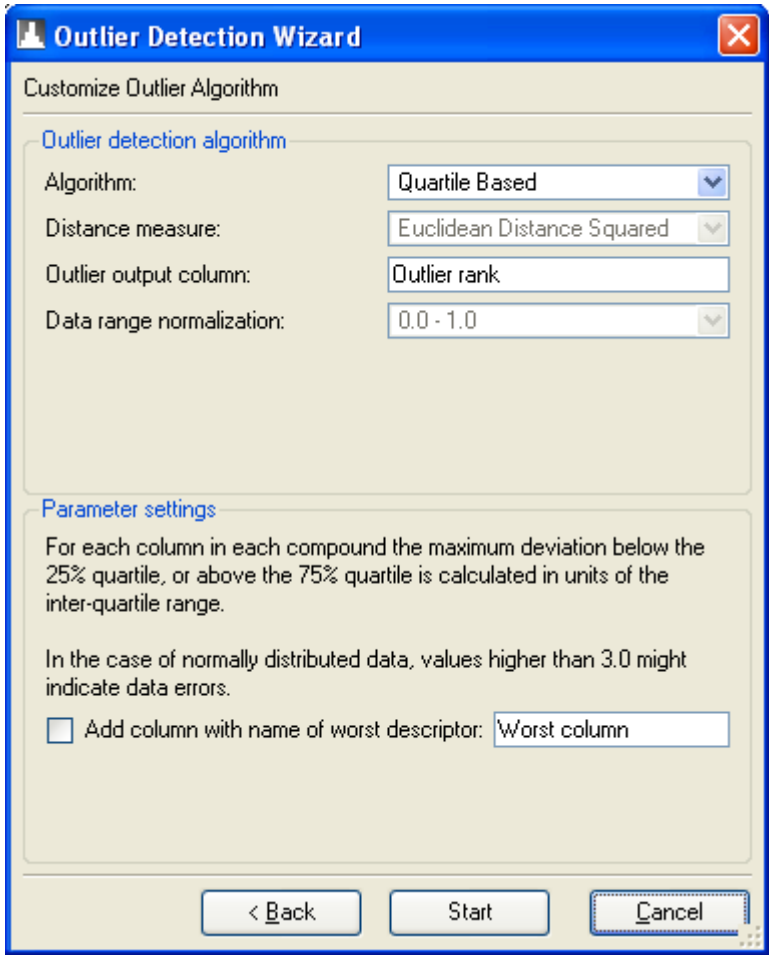


To invoke the **Outlier Detection Wizard**, select **Modelling** | **Outlier Detection...** from the menu bar.



On the first page in the wizard it is possible to specify the descriptors that should be taken into account when performing the outlier detection. This makes it possible to exclude columns with numerical identifiers or other types of data which should not be modelled.

On the next page it is shown how to select and customize the outlier detection algorithm. The default choice is the **Quartile Based** method.



Both outlier detection algorithms assign an outlier score to each row in the spreadsheet. This information is added as an extra column with the name specified in the **Outlier output column** text edit field.

The outlier score is calculated for all specified descriptors in each row. Therefore, only the numerically highest outlier score is reported back. This means that it is not possible to see which descriptor the outlier score originated from.

It is possible to add an extra column to the spreadsheet with the name of the "worst" column, by enabling the **Add column with name of worst descriptor** checkbox.

Notice that if the output column names are already used in the spreadsheet their values will be overwritten.

The **Distance measure** and the **Data range normalization** settings are not used for the quartile-based outlier detection.

Figure 49 shows an example of how the output from an outlier detection may look like.

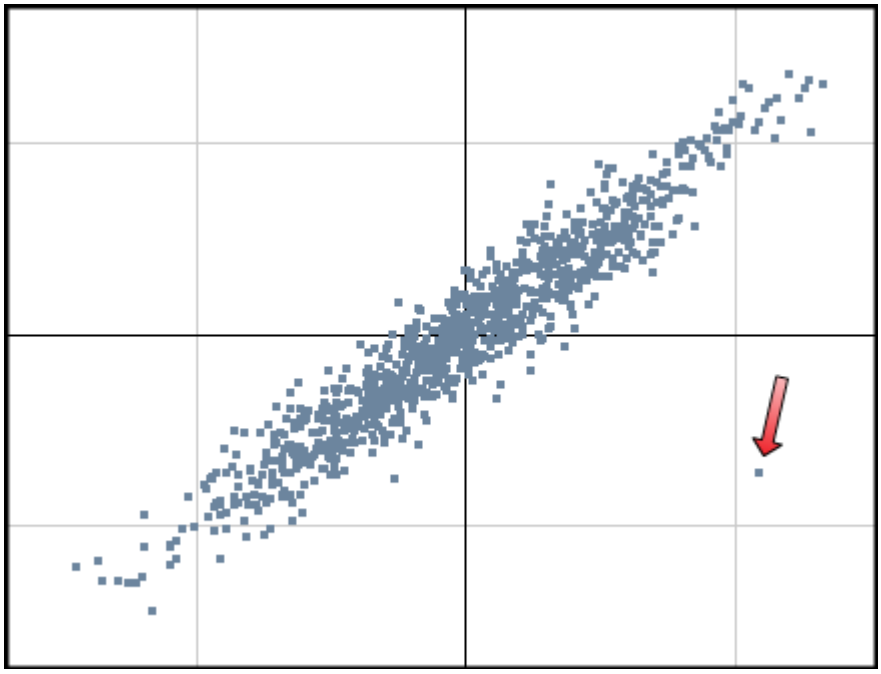
	Random 1	Random 2	Random 3	Random 4	Outlier rank	Worst column
1	422	-29	12.4067	34	20.5482	Random 2
2	-8.40634	-0.550327	16.2749	-32	13.1926	Random 4
3	-5.93436	0.976414	-3.98	10	4.1648	Random 4
4	15.7437	1.99965	14.5842	-0.462729	0.73554	Random 2
5	-9.83912	-0.394284	-7.71088	1.91983	0.633119	Random 4
6	-15.8303	0.0476588	10.2801	-1.8166	0.292576	Random 1
7	-12.6037	0.704279	14.815	-2.28974	0.206799	Random 4
8	-0.504574	1.25945	-7.20177	-0.563664	0.203455	Random 2
9	13.6492	1.12613	5.18097	-0.387682	0.147034	Random 1
10	10.6383	0.246939	8.42639	-1.36684	0	
11	-7.06065	-0.41472	0.0184916	0.146567	0	
12	3.05888	0.51851	1.88288	0.471309	0	

It is difficult to make a rule of thumb for how large the outlier rank should be before data should be discarded, but for the quartile-based outlier detection an outlier rank of more than 3.0 is sometimes referred to as an *extreme outlier*. For *normally distributed data* only about 1 out of 425,000 data points would be an extreme outlier **[OUTLIER]**. However for other distributions (or for an insufficient number of records) the acceptable outlier threshold can be much larger.

We recommend that the quartile-based method is used initially to screen the dataset for obvious data errors, such as those arising from the import or conversion of data.

6.2 Density-Based Method

Sometimes outliers cannot be detected by looking at each column independently. Figure 50 shows an example where the dependence of the descriptors must be taken into account to identify the outlier.



MDM also offers a density-based method for identifying outliers. This method works by calculating a local density measure for each data point. The data points with the lowest densities can be interpreted as the outliers in the set.

The exact formula for calculating the density for a data point, p_i , is:

$$density(p_i) = \sum_{i \neq j} \frac{1}{d(p_i, p_j) + d_{min}}$$

$d(p_i,p_j)$ is the distance between points p_i and p_j . d_{min} is a small constant added to avoid singularities between close points.

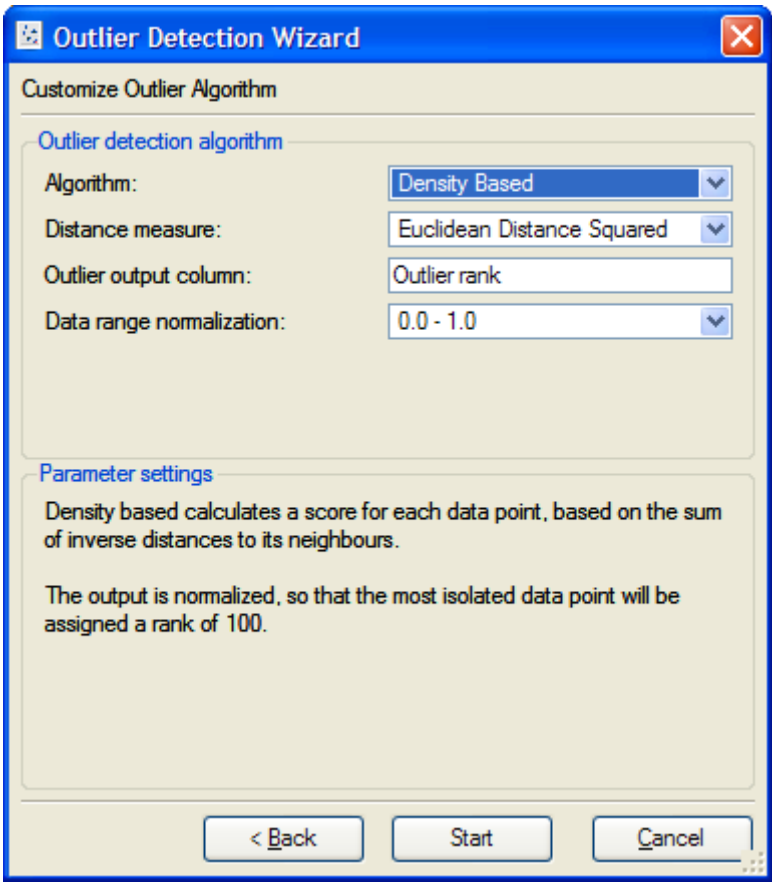
It is possible to specify how the distance between two points on the outlier detection customization page is measured (see Figure 51). We suggest to use **Euclidean Distance Squared**, but it is possible to use other measures as well – see Section 11.9 for a description of the available measures. Notice that the scale of the descriptors heavily influences the distance between data points – therefore unless the descriptors are known to have a meaningful relative scaling, we suggest to use the standard choice of normalizing the descriptors.

The outlier rank is derived by finding the reciprocal of the density and normalizing the resulting values from 0 to 100:

$$outlier\ rank(p_i) \propto \frac{1}{density(p_i)}$$

Thus an outlier rank of 100 corresponds to the most isolated point in the dataset (notice that because the Cosine Similarity and Tanimoto Coefficient measures are numerically higher if two data points are close, the opposite is

true for these measures: numerically high outlier ranks would correspond to points in dense regions).



7 Creating Subsets

7.1 Where Can Subsets be Used?

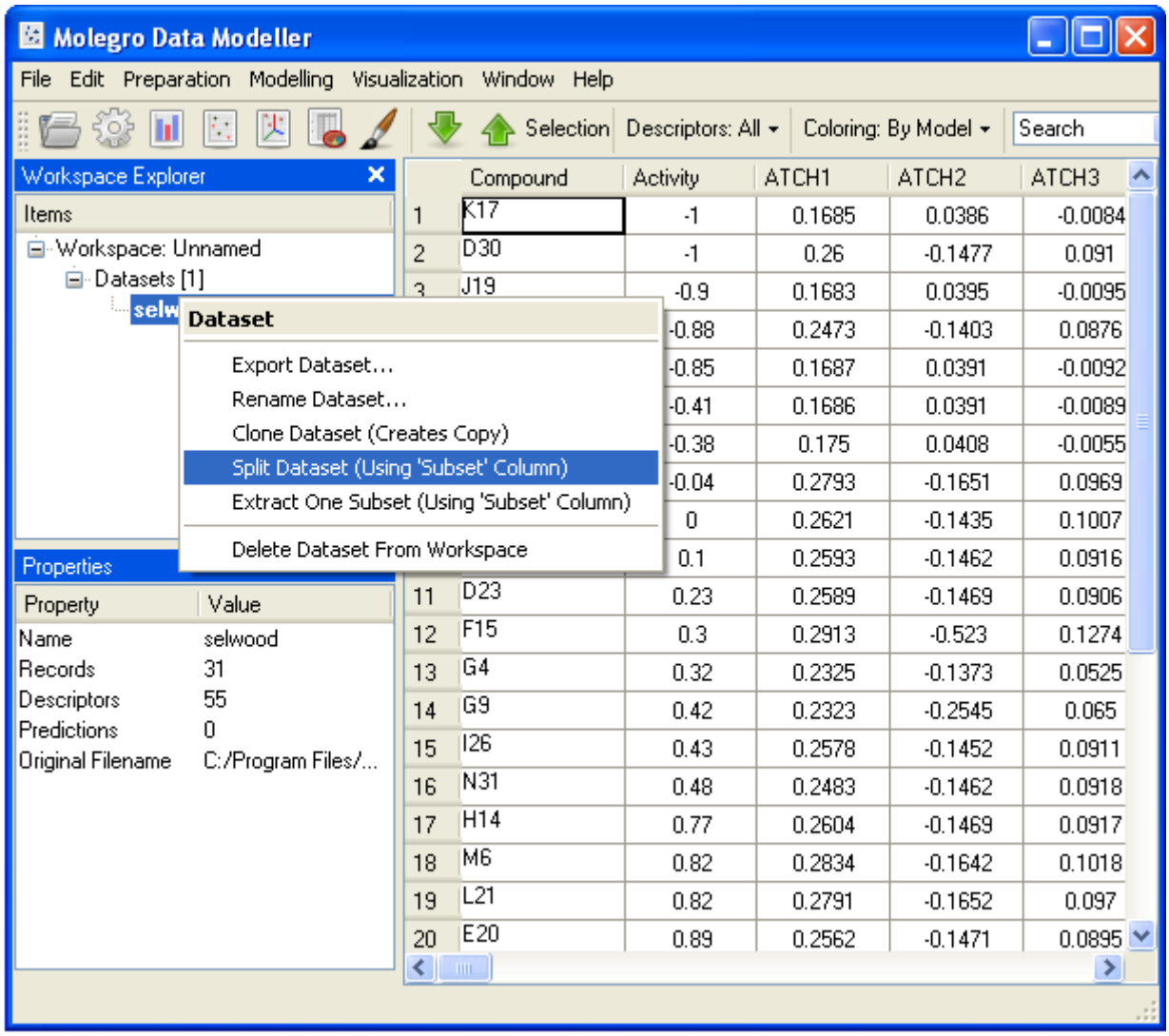
In Molegro Data Modeller it is possible to group dataset records into subsets. Subsets are represented by an integer identifier listed in the **Subset** column (by default all records belong to subset 0).

Subsets can be used to:

- Split a dataset into several datasets: From the **Split Dataset (Using 'Subset' Column)** dataset context menu in the Workspace Explorer it is possible to split the given dataset into a number of sub-datasets using the subset identifiers. Each new dataset will be named after the original dataset and the subset identifier (e.g. selwood_0, selwood_1, etc.). The original dataset will not be modified.
- Extract a subset from a dataset: From the **Extract One Subset (Using 'Subset' Column)** dataset context menu in the Workspace Explorer it is possible to extract a single subset from the given dataset. From the dialog box invoked, it is possible to select which subset to extract. Afterwards, a new dataset is created containing all records with the corresponding subset identifier. The new dataset will be named after the original dataset and the chosen subset identifier (e.g. selwood_2 if subset 2 was chosen). The selected records are removed from the original dataset.
- Perform cross validation of regression/classification models: If a dataset contains subsets, it is possible to perform a N-fold cross validation of a given regression/classification model where the number of folds equals the number of subsets available. The subset-based cross validation option is available via the **Experimental Setup** tab page in the

Regression Wizard or **Classification Wizard** (See Section for more details).

- Make regression or classification models using a reduced training set: Using the subset-creation methods introduced in Section 7.5 it is possible to make a regression/classification model on a subset of the original dataset. Using a subset can lower the total time needed for model training, since the number of records used for training can be significantly reduced compared with the total number of records available in the full dataset.



7.2 Creating Subsets From Selected Rows

Subsets can be created manually from selected records in the Spreadsheet Window. After selecting the records that should be part of a given subset, a subset is created using the **Create Subset from Selected Rows...** menu invoked from the spreadsheet context menu or from the **Preparation** menu. The following options are available:

- **As New Dataset (Keep in Current Dataset)**. A new dataset containing the selected records only will be added to the workspace. The dataset will be given a name similar to the original dataset with the addition of a count indicating the number of selected records in the new dataset and total records in the original dataset. The original dataset is not modified.
- **As New Dataset (Remove from Current Dataset)**. Similar to the option above except that the selected records will be removed from the original dataset.
- **Write Subset IDs to 'Subset' Column**. The selected records will be assigned a unique subset identifier shown in the **Subset** column (the column will be created if it does not exist).

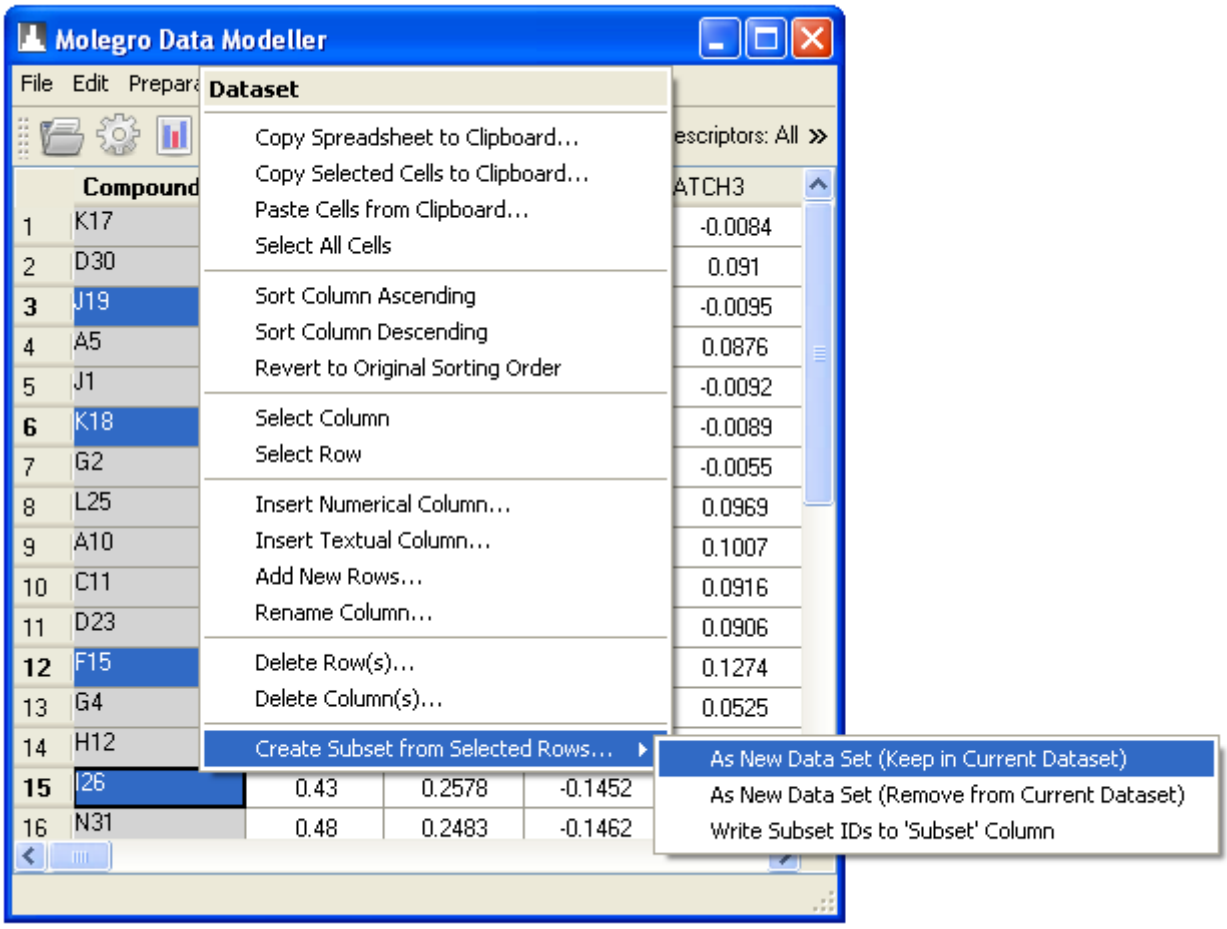
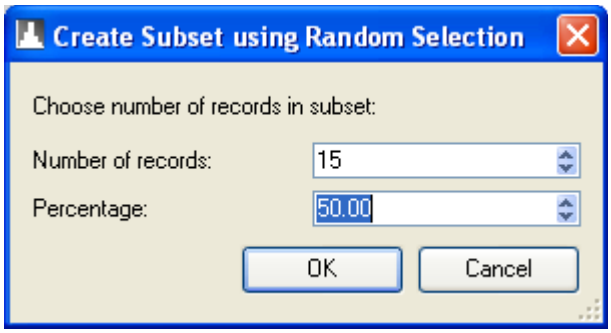


Figure 53: Creating a subset from selected records in the Spreadsheet Window.

7.3 Creating Subsets Using Random Selection

Subsets can also be created from randomly selected records using the **Create Subset using Random Selection...** menu invoked from the **Preparation** menu. From the sub-menu it is possible to select how the new subset should be created. The options available are identical to the ones described in Section 7.2.

It is possible to choose the number (or percentage) of records that should be part of the subset. The new subset containing the randomly selected records is created when pressing the **OK** button.



7.4 Create Subset Using 'Subset' Column

Subsets can also be created from the subset identifiers listed in the 'subset' column (if available) using the **Create Subset using 'Subset' Column...** menu invoked from the **Preparation** menu. This option can be used to create subsets based on a clustering of a given dataset since the cluster association for each record is provided in the 'subset' column. From the sub-menu it is possible to select how the new subset should be created. The options available are identical to the ones described in Section 7.2.

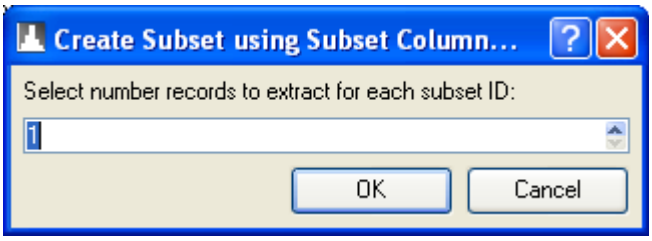


Figure 55: Creating a subset using 'subset' column.

It is possible to choose the number of records to extract for each subset identifier that should be part of the new subset. The maximum number of records that can be extracted for each subset identifier corresponds to the number of records of the subset identifier with the lowest number of records (to ensure that the same number of records are extracted for each subset identifier).

The new subset containing the randomly selected records is created when pressing the **OK** button.

7.5 Creating Subsets From Selected Descriptors

In Molegro Data Modeller, it is possible to automatically create subsets based on a set of selected numerical descriptors. The subset creation procedure is built into the 1D, 2D, and 3D Plot dialog boxes allowing for visual inspection of the records to be included in the subsets. Further, a **Create Subset using N-dimensional Grid** dialog box is available making it possible to create subsets

from more than three descriptors. Each subset creation method will be introduced in the following sections.

Creating a Subset Using the 1D Plot Dialog Box

From the 1D Plot dialog box it is possible to create subsets from the selected descriptor.

A subset is created by selecting one data point from each non-empty histogram bin. If more than one data point is available within a bin, the data point closest to the bin center is selected.

The number of records selected for the current subset is shown in the **Subset creation** toggle box. Further, the total number of histogram bins available (named **compartments**) and empty bins are shown. The **mean occupancy** shows the average number of data points located in each bin.

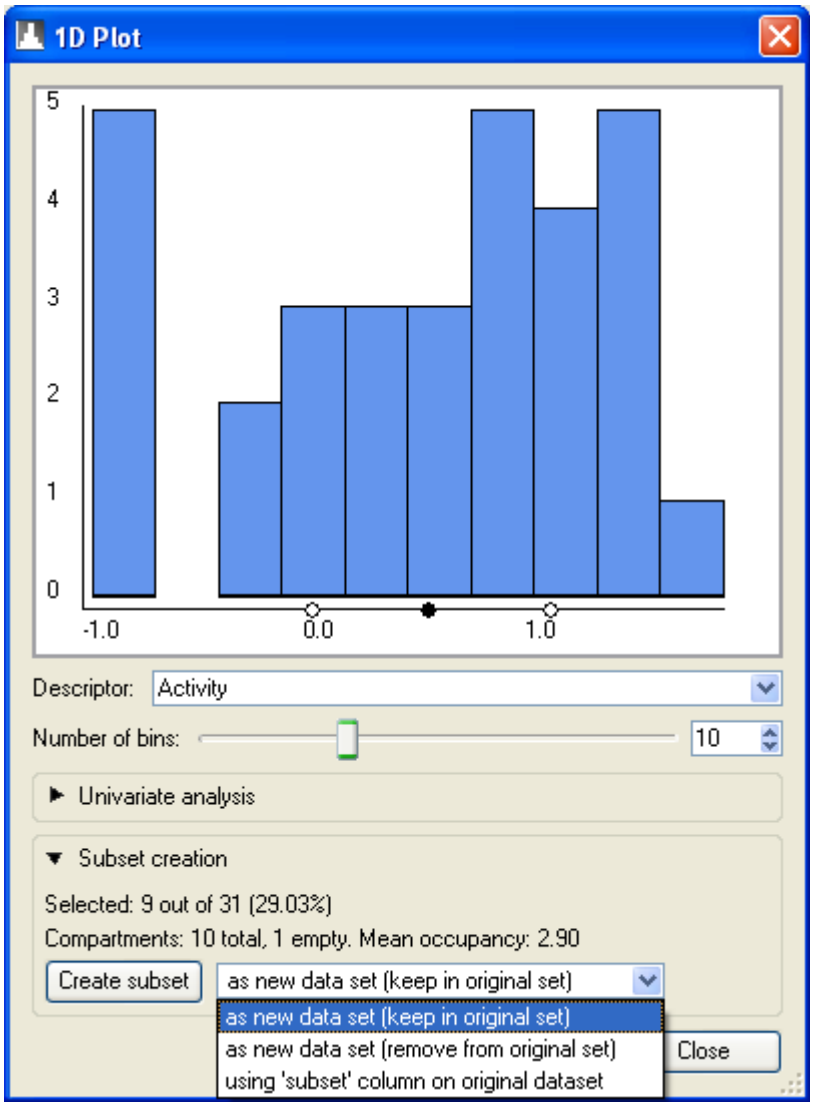


Figure 56: Creating a subset from 1D Plot dialog box.

To create a new subset press the **Create subset** button. The following options for how the new subset should be created are available:

- **As New Dataset (Keep in Current Dataset)**. A new dataset containing the selected records only will be added to the workspace. The dataset will be given a name similar to the original dataset with the addition of a count indicating the number of selected records in the new dataset and total records in the original dataset. The original dataset is not modified.
- **As New Dataset (Remove from Current Dataset)**. Similar to the option above except that the selected records will be removed from the original dataset.
- **Using 'Subset' Column on Original Dataset**. The selected records will be assigned a unique subset identifier shown in the **Subset** column (the column will be created if it does not exist). If a subset column exists, the new subset will be assigned a unique identifier (by adding +1 to the highest subset ID currently listed in the subset column). Notice: using the 'subset' column option, it is not possible to create a new subset from records that already have a subset ID higher than 0 assigned.

Creating a Subset Using the 2D Plot Dialog Box

From the 2D Plot dialog box it is possible to create subsets from a 2D grid that spans the two descriptors selected. A subset is created by selecting one data point from each non-empty grid cell. If more than one data point is available within a cell, the data point closest to the center of the cell is selected.

For example, the red data points in Figure 57 represent data points selected for the current subset that will be created when pressing the **Create subset** button.

By clicking on the **Subset creation** toggle box, it is possible to adjust various options. The **Grid center** option specifies where the center of the 2D grid should be placed. The center can be set to: **Center of bounding box**, **Center of mass**, or **Center of selected data points**.

The **Grid span** options are used to set the span for each dimension (x and y axes) making it possible to increase or decrease the grid. The **lock** toggle option is used to toggle whether grid span settings should be individual for each dimension or not.

The **Grid divisions** options are used to specify the number of grid cells for each dimension. The **lock** toggle options are used to toggle whether grid divisions are the same for both dimensions or not.

For a given subset selection, based on selected descriptors and grid settings, the number of data points selected is shown (e.g. **Selected: 18 out of 31 (58.06%)**). Moreover, the total number of grid cells available (named

compartments) and empty grid cells are shown. The **mean occupancy** shows the average number of data points located in each grid cell.

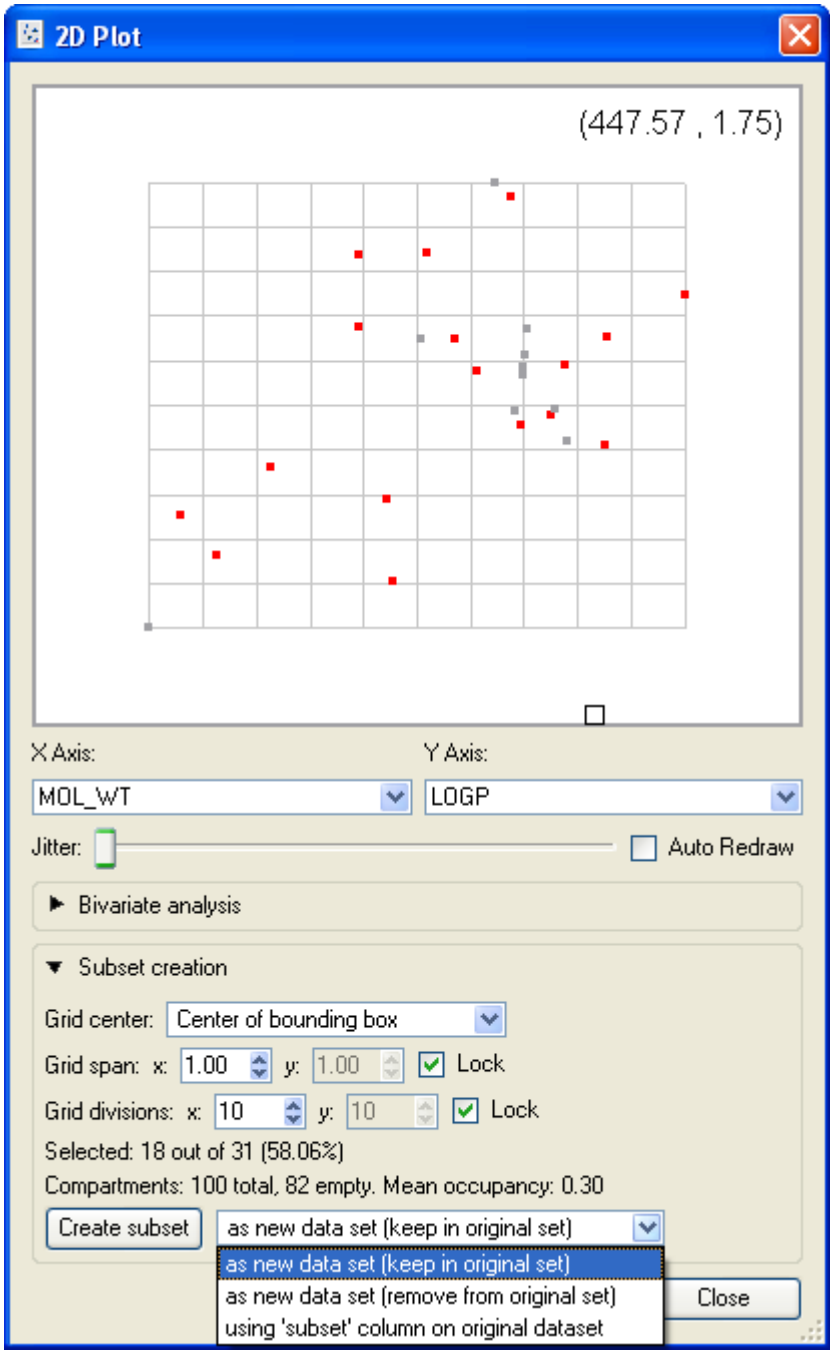


Figure 57: Creating a subset from 2D Plot dialog box.

To create a new subset, press the **Create subset** button. The options available for how the new subset should be created (as new dataset, using 'subset' column) are identical to the ones described in Section 'Creating a Subset Using the 1D Plot Dialog Box'.

Creating a Subset Using the 3D Plot Dialog Box

From the 3D Plot dialog box it is possible to create subsets from a 3D grid that spans the three descriptors selected. A subset is created by selecting one data point from each non-empty grid cell. If more than one data point is available within a cell, the data point closest to the center of the cell is selected.

For example, the red points in Figure 58 represent data points selected for the current subset that will be created when pressing the **Create subset** button.

By clicking on the **Subset creation** toggle box, it is possible to adjust various options. The **Grid center** option specifies where the center of the 3D grid should be placed. The center can be set to: **Center of bounding box**, **Center of mass**, or **Center of selected data points**.

The **Grid span** options are used to set the span for each dimension (x, y, and z axes) making it possible to increase or decrease the grid. The **lock** toggle option is used to toggle whether grid span settings should be individual for each dimension or not.

The **Grid divisions** options are used to specify the number of grid cells for each dimension. The **lock** toggle options are used to toggle whether grid divisions are the same for all dimensions or not.

For a given subset selection, based on selected descriptors and grid settings, the number of data points selected is shown (e.g. **Selected: 21 out of 31 (67.74%)**). Moreover, the total number of grid cells available (named **compartments**) and empty grid cells are shown. The **mean occupancy** shows the average number of data points located in each grid cell.

To create a new subset press the **Create subset** button. The options available for how the new subset should be created (as new dataset, using 'subset' column) are identical to the ones described in Section 'Creating a Subset Using the 1D Plot Dialog Box'.

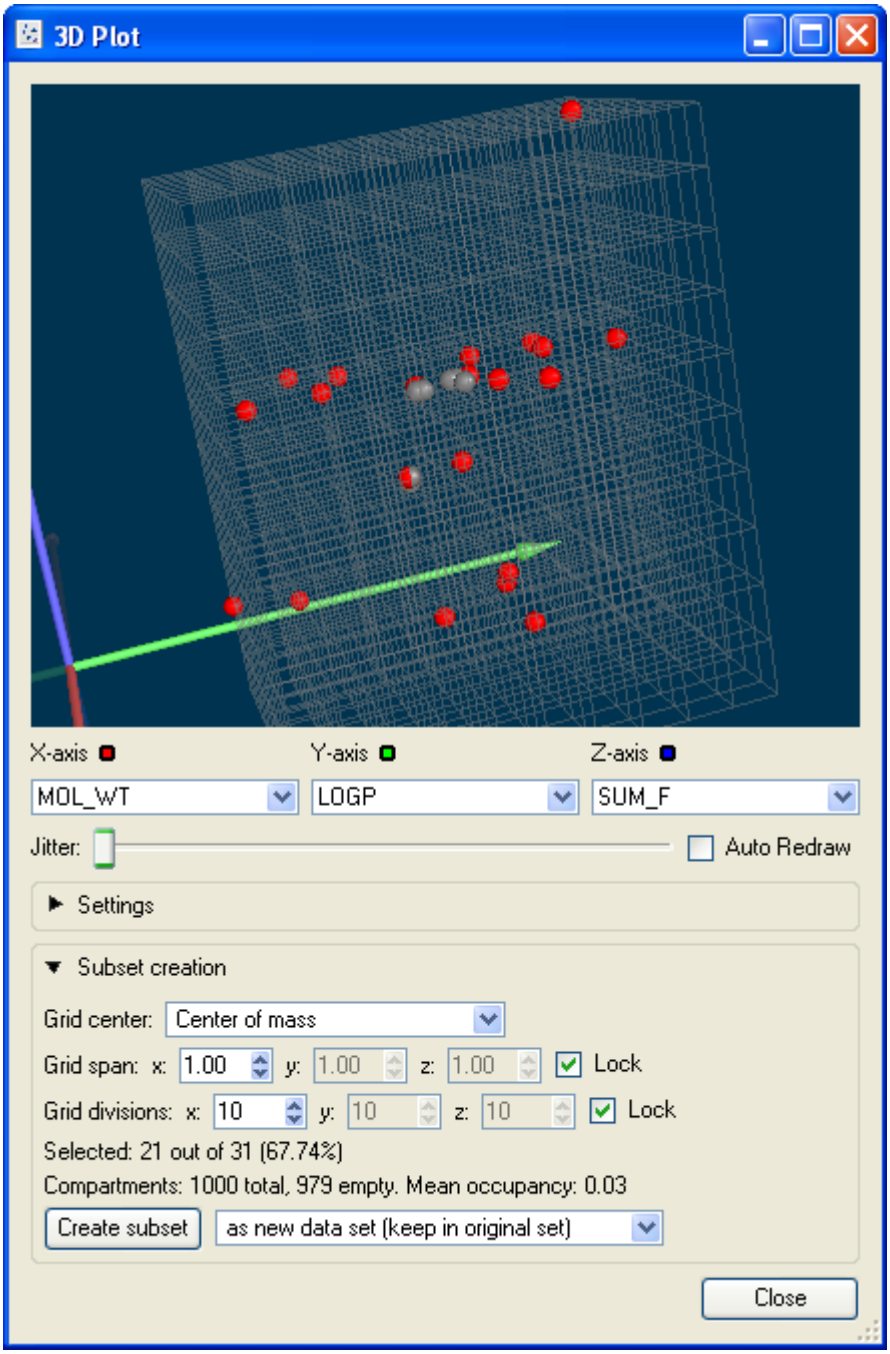


Figure 58: Creating a subset from 3D Plot dialog box.

Creating a Subset Using a N-Dimensional Grid

Creating subsets from more than three numerical descriptors is possible using the **Create Subset using N-dimensional Grid** dialog. To start the dialog box, select **Preparation** | **Create Subset using N-dimensional Grid...**.

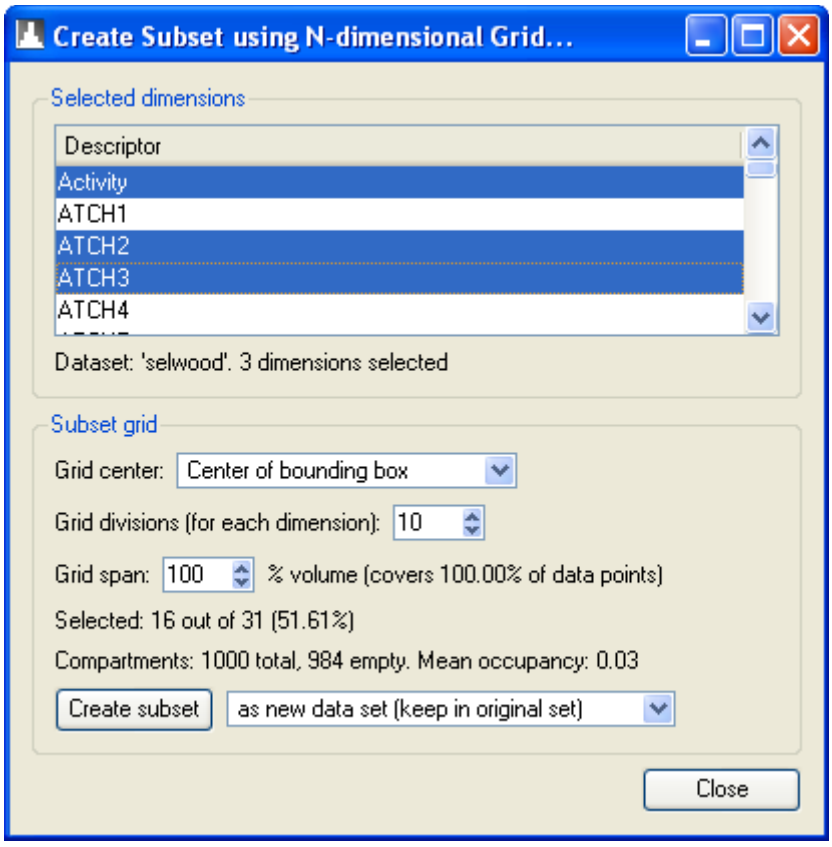


Figure 59: N-dimensional Subset dialog box: Creating a subset from selected descriptors.

The numerical descriptors that should be used to create the subset are selected in the **Selected dimensions** listview. The settings listed in the **Subset grid** box are similar to the settings available in the 2D and 3D Plot dialog boxes. The main difference is that the **Grid span** option is specified in percentage of the volume and the number of **Grid divisions** (for each dimension) are the same for all descriptors used.

Notice: The maximum number of compartments allowed is 10,000,000. If the number of compartments exceeds this threshold a warning is shown and the number has to be reduced by lowering the number of grid divisions or the numerical descriptors used.

To create a new subset press the **Create subset** button. The options available for how the new subset should be created (as new dataset, using 'subset' column) are identical to the ones described in Section 'Creating a Subset Using the 1D Plot Dialog Box'.

8 Regression

Regression methods model the relationship between a dependent variable *Y* (sometimes called the *target variable* or the *response*) and a set of independent variables X_i (here: numerical descriptors). Regression models make it possible to model and discover relationships in existing data, and to make predictions on unseen data.

Molegro Data Modeller provides four methods for regression analysis, namely *multiple linear regression*, *partial least squares*, *neural networks*, and *support vector machines*. This chapter gives a short introduction to the various methods and describes how to create and evaluate regression models using the **Regression Wizard**.

8.1 Multiple Linear Regression

In multiple linear regression (MLR) the model assumes that the dependent variable *Y* is a linear function of the independent variables, X_i. The model can be written as:

$$Y = c_0 + c_1 X_1 + c_2 X_2 + \ldots + c_N X_N$$

where the c_i's are the regression coefficients in the linear model.

To apply MLR successfully, the number of records (observations) must be larger than the number of descriptors selected.

Notice that if the independent variables are proportional or highly correlated with each other, MDM may emit a warning: 'Matrix is rank deficient'. The algorithm will automatically try to handle this by introducing a small artificial perturbation (*ridge regression*), but it is preferable to inspect the descriptors and try to reduce their internal correlation. This can be done by manual pruning, by feature selection, or by principal component analysis.

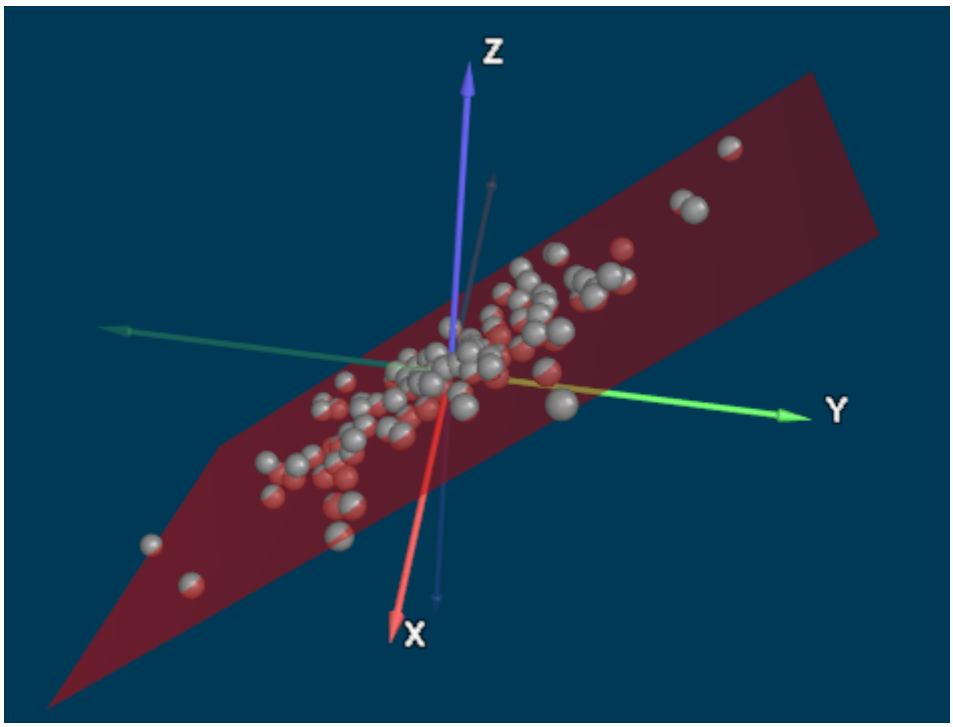


Figure 60: Example of MLR: the red plane shows the linear model as a function of the independent variables X and Y.

8.2 Partial Least Squares

Partial least squares (PLS) regression (also referred to as *Projection to Latent Structures*) is a regression technique that generalizes and combines features from principal component analysis (introduced in Chapter 10) and multiple linear regression. In PLS, the overall strategy is to extract a smaller set of factors called latent components from the set of available descriptors (independent variables X_i), which models the dependent variable Y.

In contrast to PCR (multiple linear regression using principal components as descriptors), PLS regression creates latent components from the independent variables, X_i, while taking the dependent variable Y into account. Specifically, PLS regression searches for a set of latent components that maximizes the covariance to the response, Y. These latent components are used to train a regression model predicting Y.

MDM uses the *PLS1* algorithm for partial least squares regression **[PLS 2007]**. See **[PLS 2007]** for more details about the PLS1 algorithm and how the latent components are derived.

8.3 Neural Networks

Artificial neural networks are inspired by real-world biological neural networks. Although neural networks are very simplified models of the neural processing found in the human brain, they have shown good performance on regression and classification problems.

An example of a network structure is shown in Figure 61. The neural networks are constructed by assigning each independent variable to a neuron in the *input layer*. Connections are then formed to neurons in the next layer. Each connection multiplies the neuron output by a *weight* before the output enters the connected neuron. The neuron output is then calculated by summing all inputs and applying a sigmoid function:

$$output = sigmoid\left(\sum input\right)$$

where $$sigmoid\left(x\right) = \frac{1}{1+e^{-x}}$$

The number of layers and neurons in each layer can be adjusted, but typically one or two *hidden layers* are inserted between the input and output layers. Finally the connections from the last hidden layer are connected to the output layer, which will be trained to estimate the dependent variable.

In addition to the input and hidden layer neurons, it is also possible to include so-called *bias neurons*. A bias neuron connects to all the neurons in the next layer and has a constant value of 1.Therefore, the weight of the bias neuron becomes an offset of the sigmoid function and acts as a threshold for the output neuron.

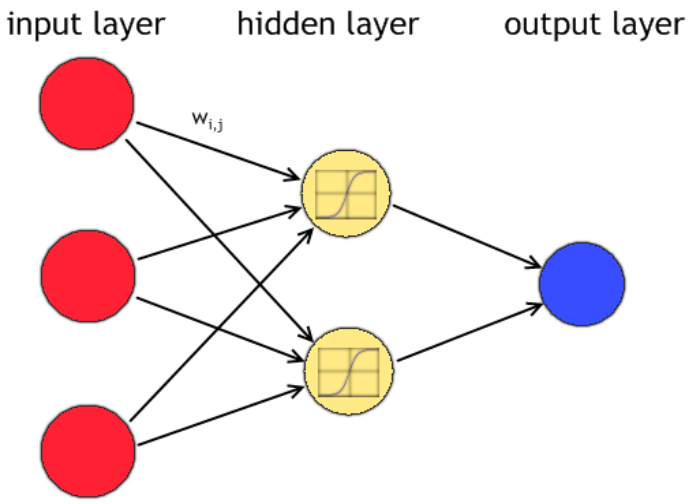


Figure 61: Neural network overview.

The neural networks used in MDM are always *fully connected*, *feedforward networks*, in which a neuron has outgoing connections to all the neurons in the next layer, but not to neurons in other layers than this.

The weights in a neural network are adjusted in order to minimize the error between the output from the output neuron and the dependent variable in the dataset. The algorithm used in MDM for training neural networks models is called **back-propagation** (described in detail in **[HAYKIN 1999]**).

One of the advantages of neural networks is their ability to model complex and highly non-linear relations (see Figure 62). However, the complexity of neural networks makes it easy to overfit the training data. As with all regression models always try to choose the simplest model.

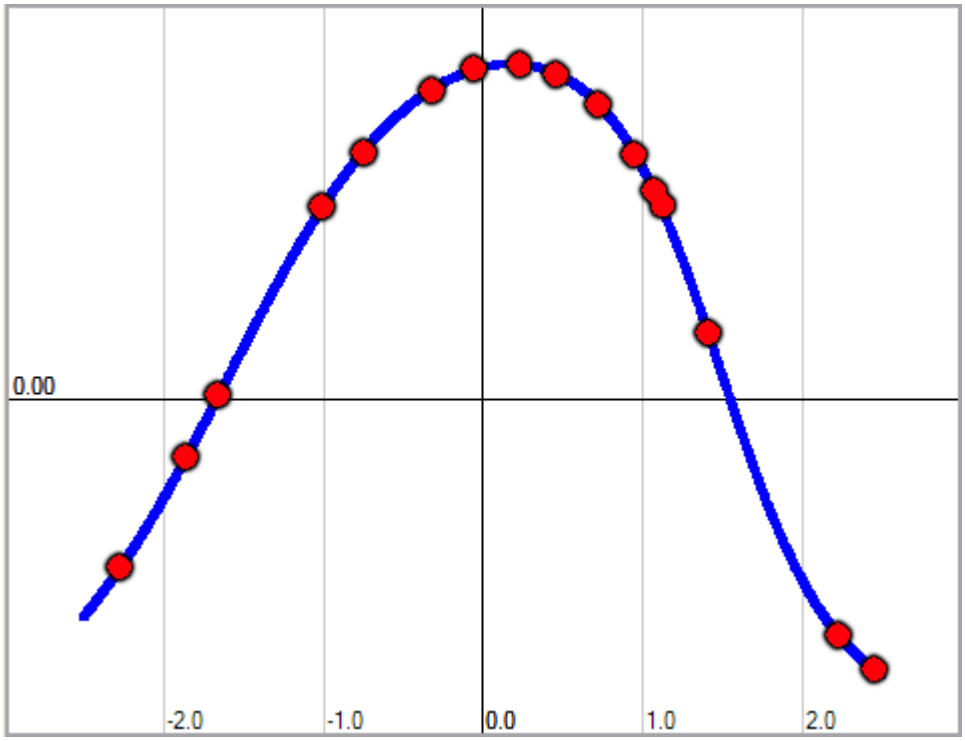


Figure 62: Example of neural network model modelling a function of one parameter. The red dots are the samples, and the blue curve the model prediction. Neural Networks are able to model highly non-linear relationships.

Neural Network Algorithm Customization

It is possible to customize the neural network algorithm in various ways (this is done in the **Regression Wizard**, see Figure 69). Most of these settings are normally acceptable at their default setting, but the most important step is to choose a model that is neither too simple nor too complex.

The model complexity is determined both by the number of descriptors

(corresponding to the number of input neurons) and by the number of hidden layer neurons. Each hidden layer neuron is connected to all neurons in the next layer, and each connection is assigned a weight, which quickly introduces a lot of free parameters to the model. So always try to use as few hidden layer neurons as possible.

8.4 Support Vector Machines

Support vector machines were introduced by Boser, Guyon, and Vapnik in 1992 **[BOSER 1992]**.

Originally, support vector machines were used for linear classification. Consider Figure 63 where two types of objects (red and blue) are positioned on a 2D plane. We are interested in a classifier capable of predicting the type of an object given its position in the plane. In this case the data are linearly separable with several possible choices of lines dividing the plane into red and blue regions. Support vector machines try to find the *maximum separating hyperplane*, which in 2D corresponds to the line with the widest borders. In Figure 63 the maximum separating hyperplane (which here is a line) is shown together with its two borders (the two dotted lines). The data points located on these two dotted lines are the *support vectors*. It is important to notice that the model is determined completely by the support vectors. The other data points are not used in the model.

It is of course not always possible to find a perfect linear separation for a given dataset. Therefore support vector machines were extended to allow for misclassified examples as well (see left side of Figure 64).

Another extension came with the introduction of the *kernel trick*. This makes it possible to go beyond simple linear separation and use more complex boundaries by choosing a suitable kernel (see right side of Figure 64).

MDM uses the algorithms provided with the LIBSVM library **[LIBSVM 2001]** to train the two types of support vector machines that are available in MDM: nu-SVR and epsilon-SVR. These types of support vector machines have been further extended to handle regression instead of classification. See **[LIBSVM 2001]** for more details about SVM and the nu-SVR and epsilon-SVR variants.

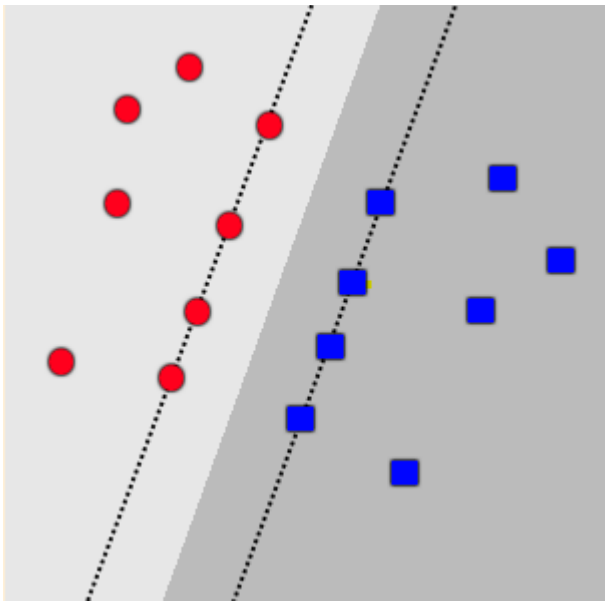


Figure 63: SVM example. The background color shows the resulting classification. The two dotted lines are the support vectors.

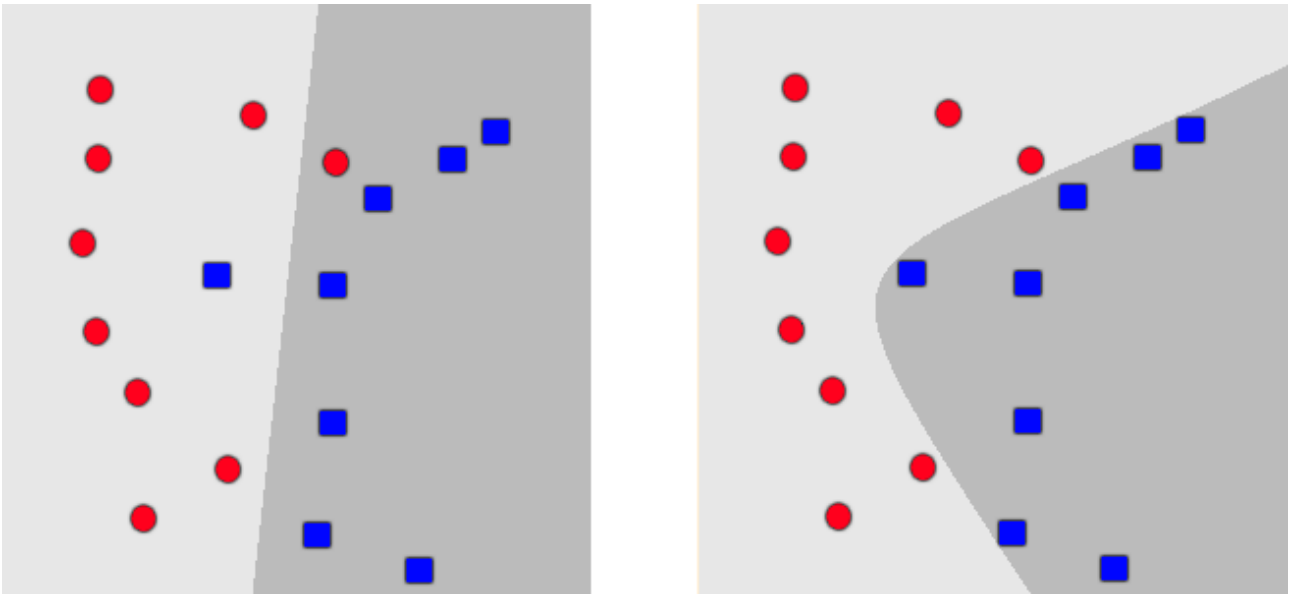


Figure 64: A SVM with soft-margins is able to handle misclassified data points, as in the example to the left, where no perfect linear separation exists. To the right the same example is shown, this time using a Radial Basis Function as kernel.

Support Vector Machine Customization

Compared to neural networks, support vector machines are fast to train. However, it can be difficult to find the optimal parameter settings and to avoid overfitting the training data. The best way to avoid overfitting is to choose the

simplest model suitable for describing the data. The complexity of a support vector machine is determined by the number of support vectors and the complexity of the kernel.

With regard to the kernel we recommend keeping the *Radial Basis Function* kernel as the default choice. The most important (and most difficult) parameters to fine-tune are the *Cost* and *Gamma* parameters. We recommend trying several choices of pairs on a coarse scale (perhaps doubling or halving the parameters at each step) or use the built-in fine-tuning option to identify good parameter values (introduced below). Be sure to use cross-validation while performing the search for the optimal parameters, otherwise the constructed models will most likely be overfitted on the training data.

8.5 Choosing a Regression Method

Each regression method has certain advantages and disadvantages:

Multiple linear regression is very fast, there are no parameters to fine-tune, and is not very likely to overfit (as long as there are are more data records than descriptors). On the negative side, it can only model simple (linear) relationships between data. It can also be very sensitive to outliers.

Partial least squares is also a fast regression method. One advantage compared with multiple linear regression, is that partial least squares might work better than e.g. multiple linear regression when the number of descriptors is larger than the number of data records (observations) available in the dataset. However, contrary to multiple linear regression models, the models created by partial least squares can be more difficult to interpret and the user needs to manually specify the number of latent components to use in the model.

Support vector machines are fast and can handle non-linear types of data. They are also more robust with regard to outliers, but it can be difficult to find optimal parameter settings.

Neural networks are also able to fit very complex and non-linear types of data, but can be slow to train. They are also prone to overfitting.

Recommendations

Choose the simplest model. Since simple models are less likely to overfit the training data, always try to find the simplest acceptable solution. For all regression models the complexity decreases if the number of independent variables is lowered. For neural networks the complexity also depends on the number of hidden layer neurons. For support vector machines, the complexity depends on the number of support vectors and the chosen kernel. It may be possible to reduce the number of independent variables by performing a PCA analysis or using feature selection.

Validate using external test set. The best way to validate the performance of a regression model is to use an external dataset which has not been involved in the training of the model. Unfortunately, the number of data records may be too small to construct an independent set. In this case, it is necessary to rely on cross-validation methods instead.

Watch out for chance correlation. When dealing with a large number of descriptors and a small number of samples, there is always the possibility that a relation between the independent variables and the dependent variable may arise by chance. Notice that cross-validation does not automatically guard against chance correlation: if the number of descriptors is large enough some combinations of the descriptors will be able to describe the dependent variable.

In particular, be careful when using feature selection together with cross-validation as model selection criteria: in this case, many combinations of descriptors will be tested, and the combination with the best cross-validated correlation will be found. But this correlation may have arisen by chance, simply by trying enough combinations.

Chance correlation can be detected by validation on an external test set, but even if this is not possible, a simple procedure exists that makes it possible to estimate the amount of chance correlation for a dataset: *y-Randomization* (sometimes called *y-Scrambling)* suggests that whenever a model has been trained on a dataset, the same procedure should be applied to a dataset where the order of the dependent variable (the target variable) has been randomized.

If the model trained on the randomized dataset yields a high cross-validated accuracy, the correlation is caused by chance. Notice that it is important to start the model building from scratch on the randomized dataset: if feature selection was performed on an initial set of descriptors, perform the feature selection once again on the randomized dataset – do not try to build a model on the randomized set with the descriptors chosen from the initial dataset. It is the whole procedure that must be repeated in order to estimate the chance correlation. Also notice that the randomization and model building should be repeated a number of times in order to get an estimate of the magnitude of the chance correlation. Y-Randomization can be performed in MDM by choosing **Preparation** | **Scramble Selected Columns** (see Section 3.11).

Check for obvious outliers. It may be difficult to decide whether abnormal data are outliers or not – and it may be scientifically questionable to remove them. However, the datasets should always be checked for *obvious* data errors arising from e.g. preparation or conversion faults.

8.6 Creating Regression Models Using the Regression Wizard

Once one or more datasets have been imported into the Workspace, new regression models can be created using the **Regression Wizard**. A shortcut is provided by clicking on the regression wizard icon on the toolbar or using the

keyboard shortcut (**CTRL+W**). It is also possible to select regression algorithms individually from the main menu: **Modelling** | **Multiple Linear Regression...**, **Modelling** | **Partial Least Squares Regression...**, **Modelling** | **Neural Network Regression...**, or **Modelling** | **Support Vector Regression...**.

Select Dataset and Target Variable

The first step is to choose a dataset (see Figure 65). It is possible to work on only a subset of the dataset by using the **Select subset** (the subsets are defined by a **Subset** column in the spreadsheet, see Chapter 7 for more details). Notice: If working on a subset, the N-fold cross validation option will be disabled. By default, all subsets in the dataset are included.

Next, a *target variable* must be selected (see Figure 65). The *target variable* indicates which specific numerical descriptor the regression model should try to estimate or predict. Notice that columns containing invalid numerical data or constant data values will be shown in the list, but it will not be possible to use them as target variables. Further, prediction columns cannot be used as target variables.

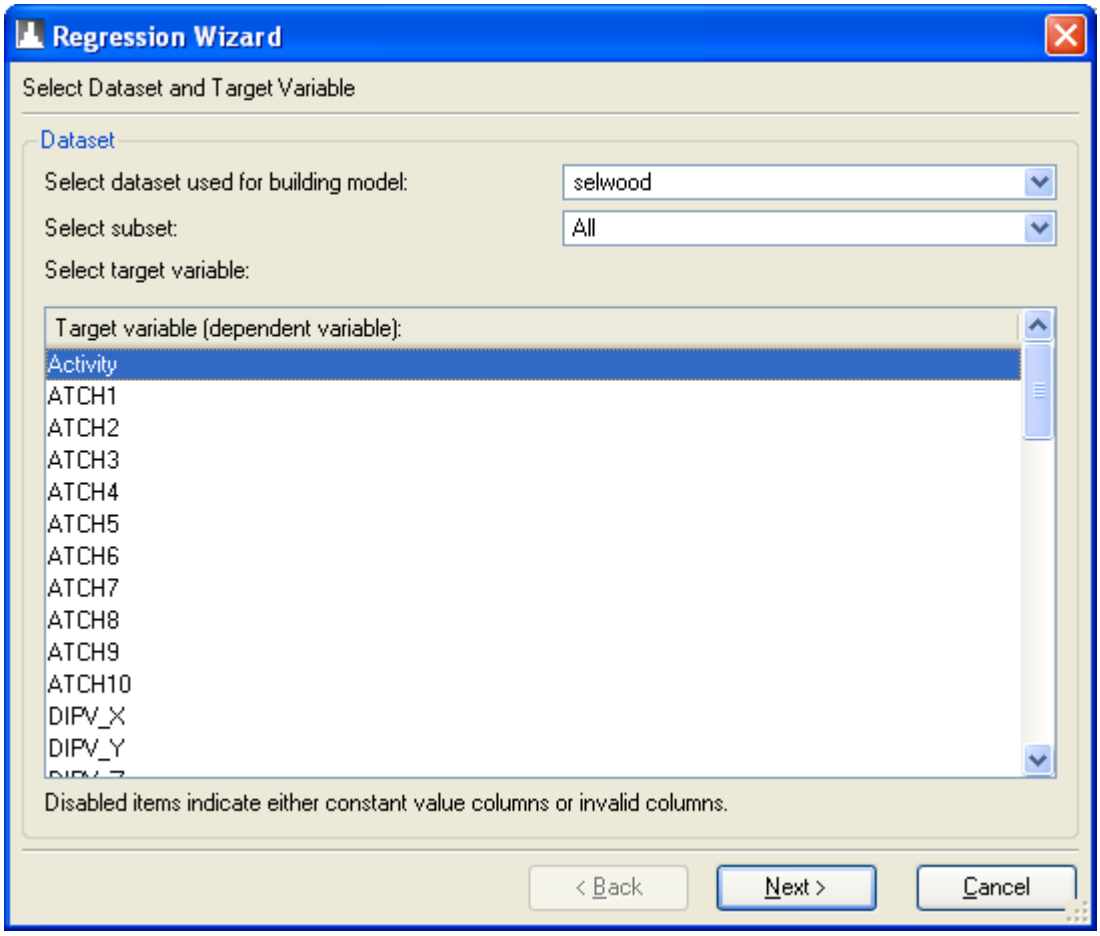


Figure 65: Select which dataset to use and what numerical descriptor to model.

Select Descriptors

The **Select Descriptors** page (Figure 66) contains a list of all the numerical descriptors available for building the regression model. As above, spreadsheet columns containing invalid numerical data or constant data values will be shown in the list, but it will not be possible to include them in the model. Prediction columns cannot be used as descriptors.

The **Descriptor selection** drop-down box allows the user to select descriptors manually or to perform feature selection: the **Manual selection from list below** option allows the user to manually select which descriptors should be included in the model. The **Feature selection (using all descriptors)** and **Feature selection (using selected descriptors)** options make it possible to perform automated selection of relevant descriptors from all descriptors or a manually selected subset of descriptors, respectively. The feature selection options are further described in the next section ('Customizing Training Algorithm').

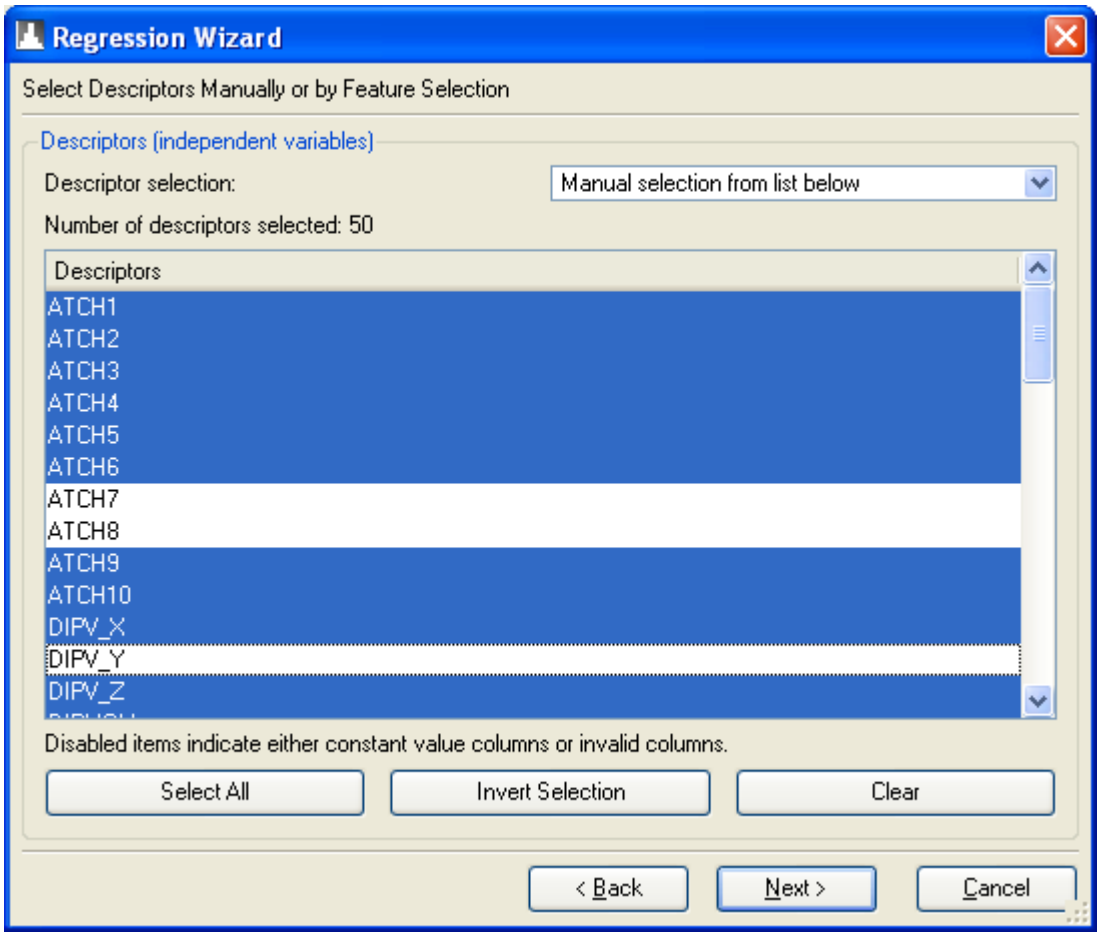


Figure 66: Select which descriptors to include in the regression model.

Customizing Training Algorithm

The algorithms used for training regression models can be customized in the **Customize Training Algorithm** page (see Figures 67-70).

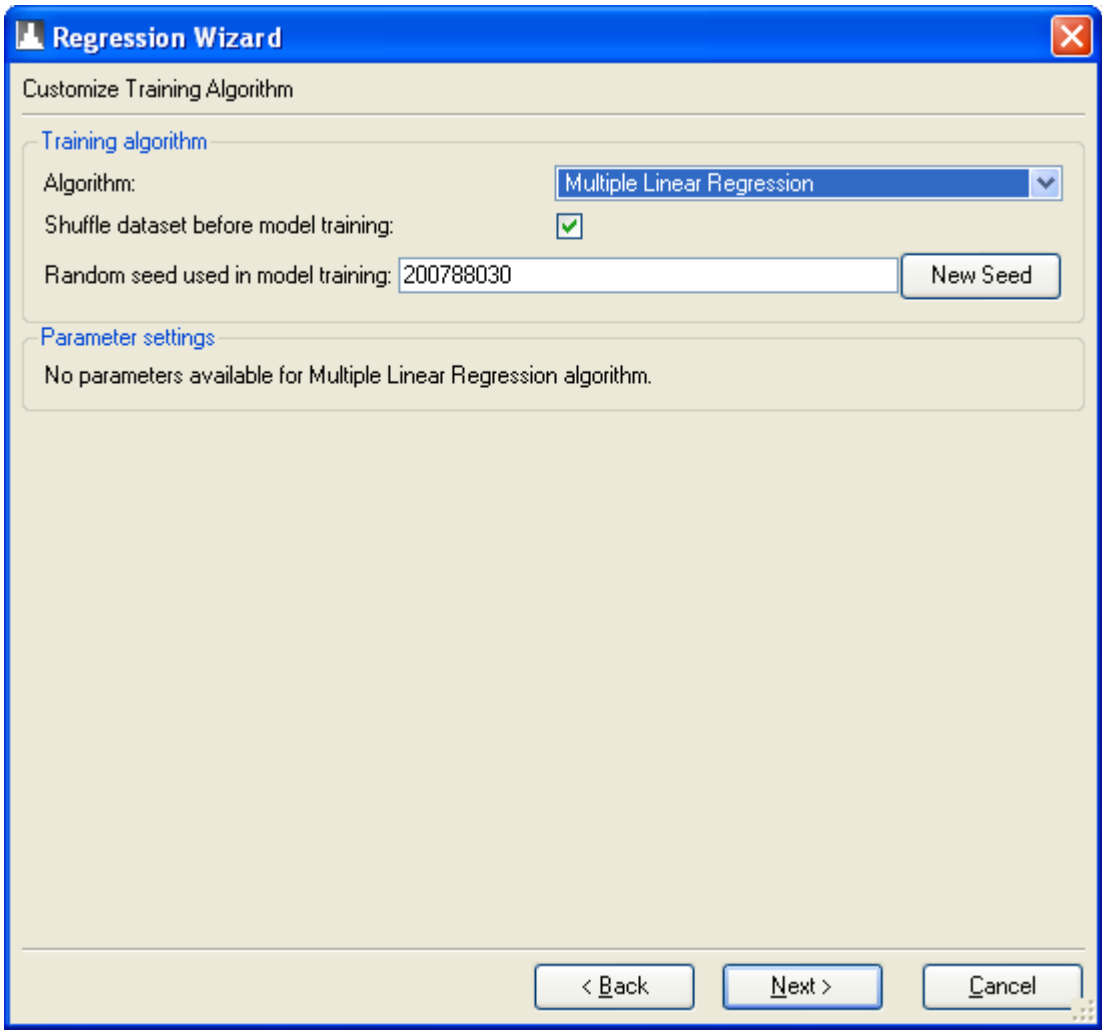


Figure 67: Parameter settings for Multiple Linear Regression models.

The **Training algorithm** box lists the training algorithms available. As described in the introduction four regression methods are available: *multiple linear regression*, *partial least squares*, *neural networks*, and *support vector machines*.

The **Shuffle dataset before model training** option toggles whether or not the order of the records in the dataset should be shuffled before the regression model is trained or evaluated. In particular the shuffling of the dataset, ensures that the folds are random, when performing N-fold cross-validation. Notice: the shuffling is performed on a cloned copy of the dataset, i.e. so the original dataset is not modified.

The **Random seed used in model training** option makes it possible to reproduce experiments by setting the random seed to the value used in the previous experiments. In addition, the **New Seed** button can be used to change the random seed currently used in the random number generator. The random numbers are used when shuffling the dataset, performing feature selection, and internally by the neural network algorithm. Notice: since the neural network model and the feature selection algorithms use random numbers, changing the random seed can produce different results compared with previous runs.

The **Parameter settings** box shows the parameters used by the training algorithm. There are no settings for multiple linear regression.

Most of the parameter settings have a **Finetune** toggle button that is used to toggle automated fine-tuning of the specific parameter on or off. The parameter fine-tuning method will be introduced later on in this chapter.

For both neural networks and support vector machines, the **Data range normalization** option indicates which normalization procedure should be applied to the dataset before the model is trained or if none should be applied (if the dataset has been normalized beforehand). Notice that the normalization is stored as part of the model, which makes it possible to reuse the model on other datasets without the need for manually normalizing the data.

Partial Least Squares Settings

The **Number of latent components** parameter is used to specify the number of latent components that will be used when performing the partial least squares procedure. The default value is 10.

Notice that it is not possible to set the number higher than the number of descriptors (independent variables) available. If the number of latent components used equals the number of descriptors in the dataset, partial least squares is equivalent to multiple linear regression.

There is no general criterion for deciding how many latents to employ. A common approach is to build models with increasing numbers of latent components using cross validation and choose the model with minimum prediction error or highest correlation coefficient.

The **Data scaling / normalization** option indicates which scaling procedure should be applied (auto-scaling or mean-centering) to the dataset before the model is trained or if none should be applied (if the dataset has been normalized beforehand). The default choice (**Auto-scaling**) is recommended.

Notice that the changes made to the dataset is stored as part of the model, which makes it possible to reuse the model on other datasets without the need for manually scaling the data.

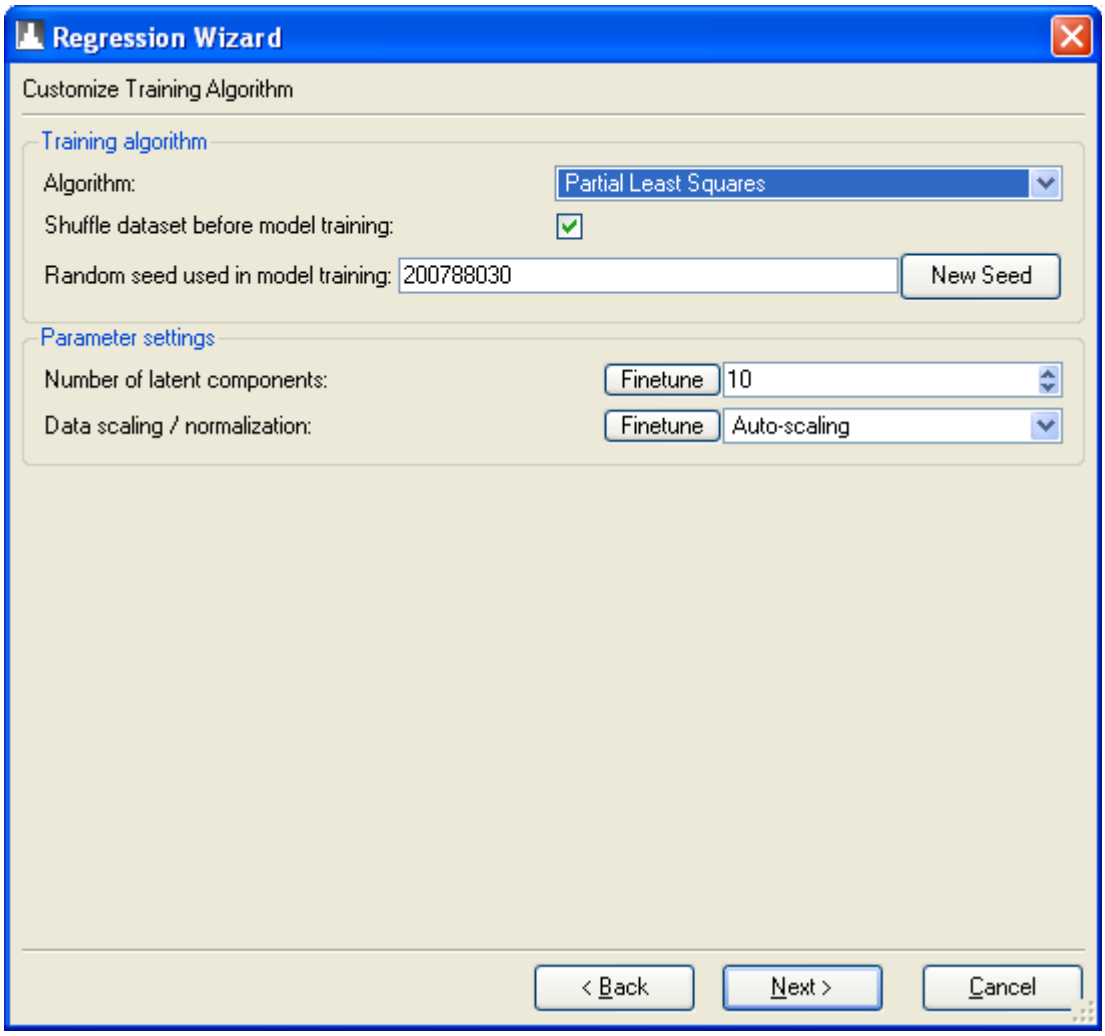


Figure 68: Parameter settings for the Partial Least Squares algorithm.

Neural Network Settings

The **Max training epochs, Learning rate**, **Output layer learning rate**, and **Momentum** parameters are used when updating neuron weights during the neural network training and can be used to speed up the convergence of the back-propagation training algorithm. Usually the default settings are sufficient.

The **Number of neurons in 1st hidden layer** and **Number of neurons in 2nd hidden layer** specify the number of neurons for each hidden layer. Often only one hidden layer is needed. Sometimes more accurate models can be build if a second hidden layer is included, but more complex models are also more prone to overfit the training set. The optimal number of hidden neurons is dependent on the actual regression problem so it may take several runs to identify the most suitable choice.

The **Initial weight range (+/-)** value indicates the range (e.g. from -0.5 to 0.5) used by the random number generator when initializing the neurons before model training is started. The default value is generally suitable for most model training tasks.

Finally, the **Use bias neurons** option can be used to set if bias neurons (in input and hidden layers) should be used or not. Typically, including bias neurons will improve the performance of the back-propagation algorithm.

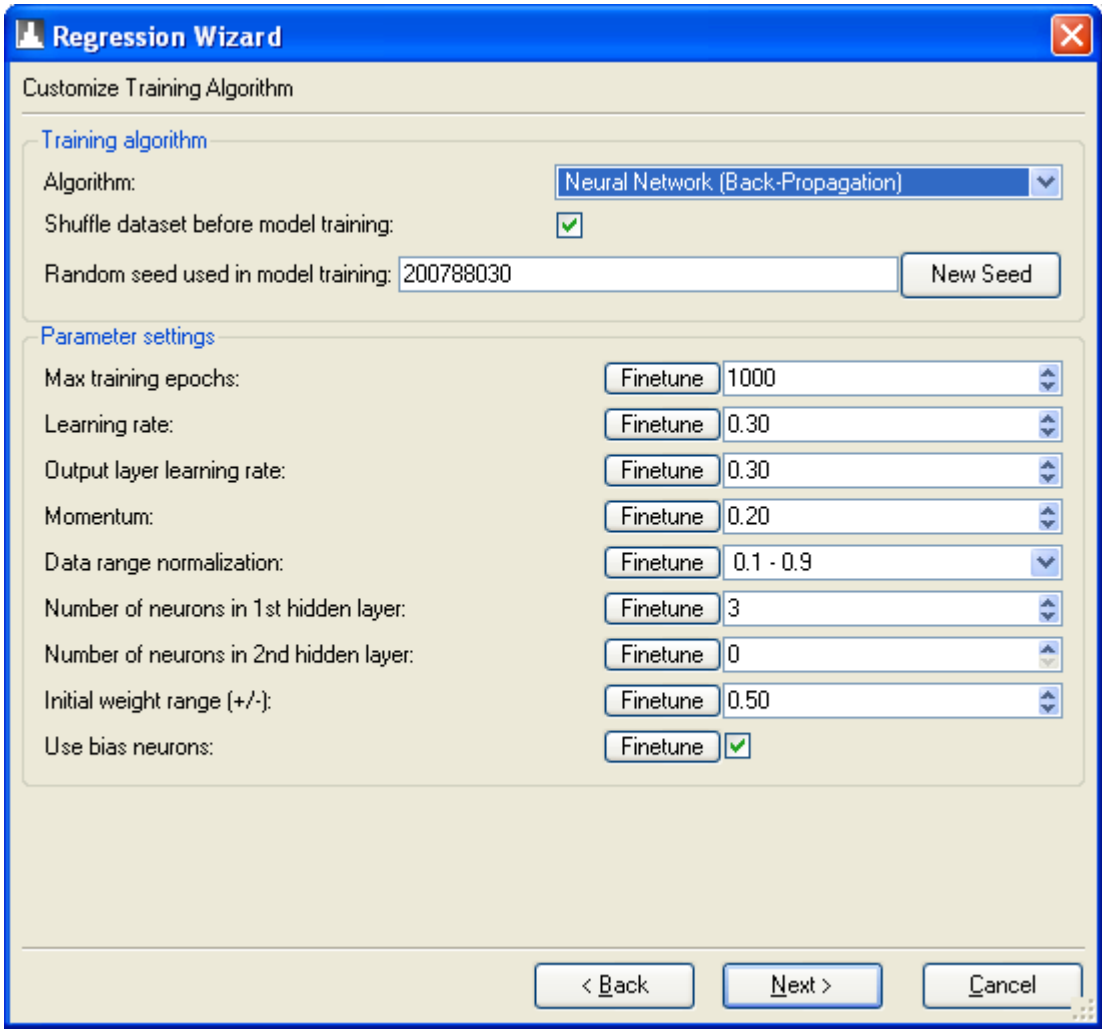


Figure 69: Parameter settings for the Neural Network model.

Support Vector Machine Settings

For the SVM training algorithm the following parameter settings are available:

Two types of SVM models for regression are available in MDM. From the **Model type** combo box it is possible to choose either the **nu-SVR** or the **epsilon-SVR** model.

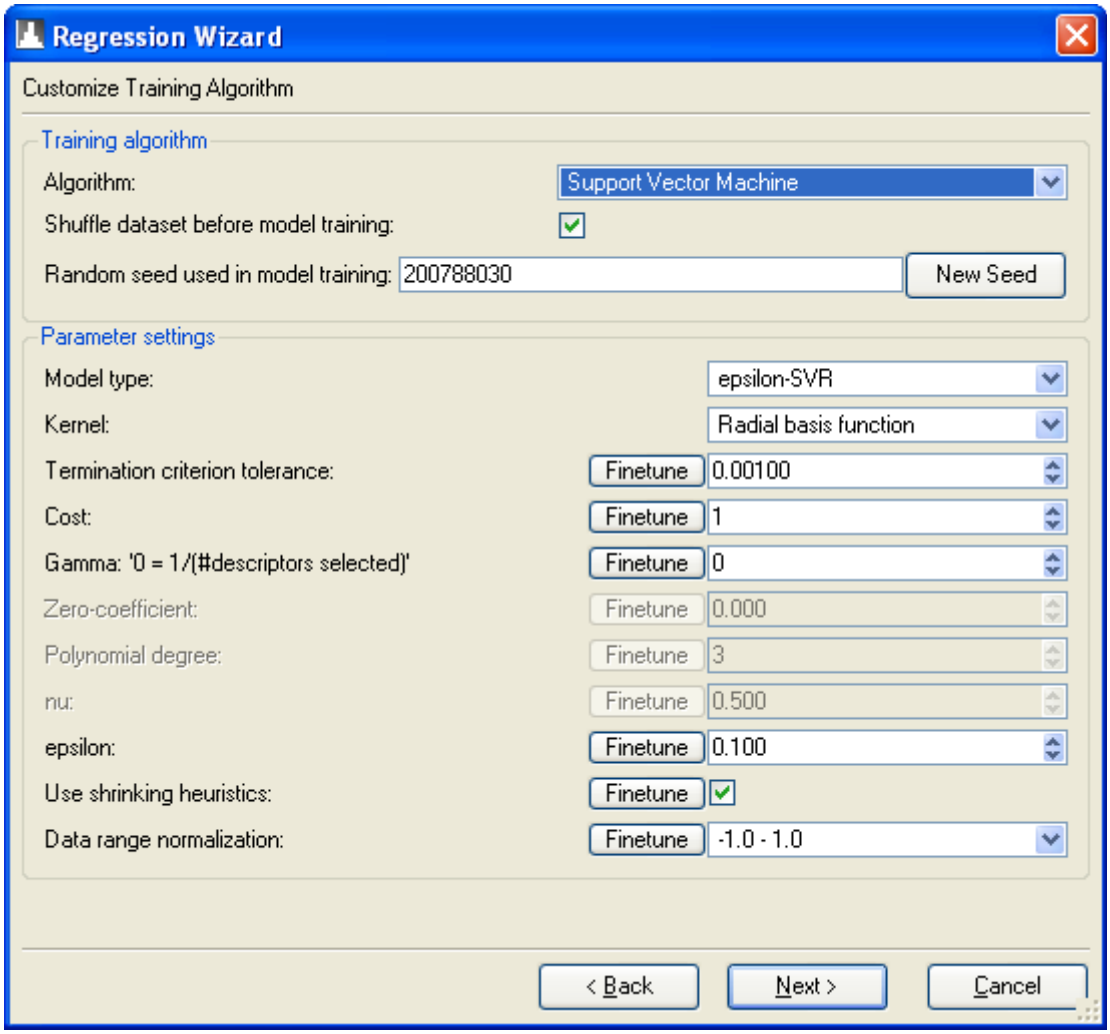


Figure 70: Parameter settings for the Support Vector Machine algorithm.

The **Kernel, Termination criterion tolerance**, **Cost**, **Gamma**, and **Use shrinking heuristics** parameters are used by both SVM model types, whereas the **Zero-coefficient**, **Polynomial degree**, **nu**, and **epsilon** parameters depend on the choice of **Model type** and **Kernel** function. See **[LIBSVM 2001]** for more details about the SVM parameters.

Fine-Tuning Parameter Settings

If one or more of the **Finetune** toggle buttons shown next to the specific parameter settings have been toggled on, automated fine-tuning of these parameters will be enabled and the following dialog box will be shown when pressing the **Next** button:

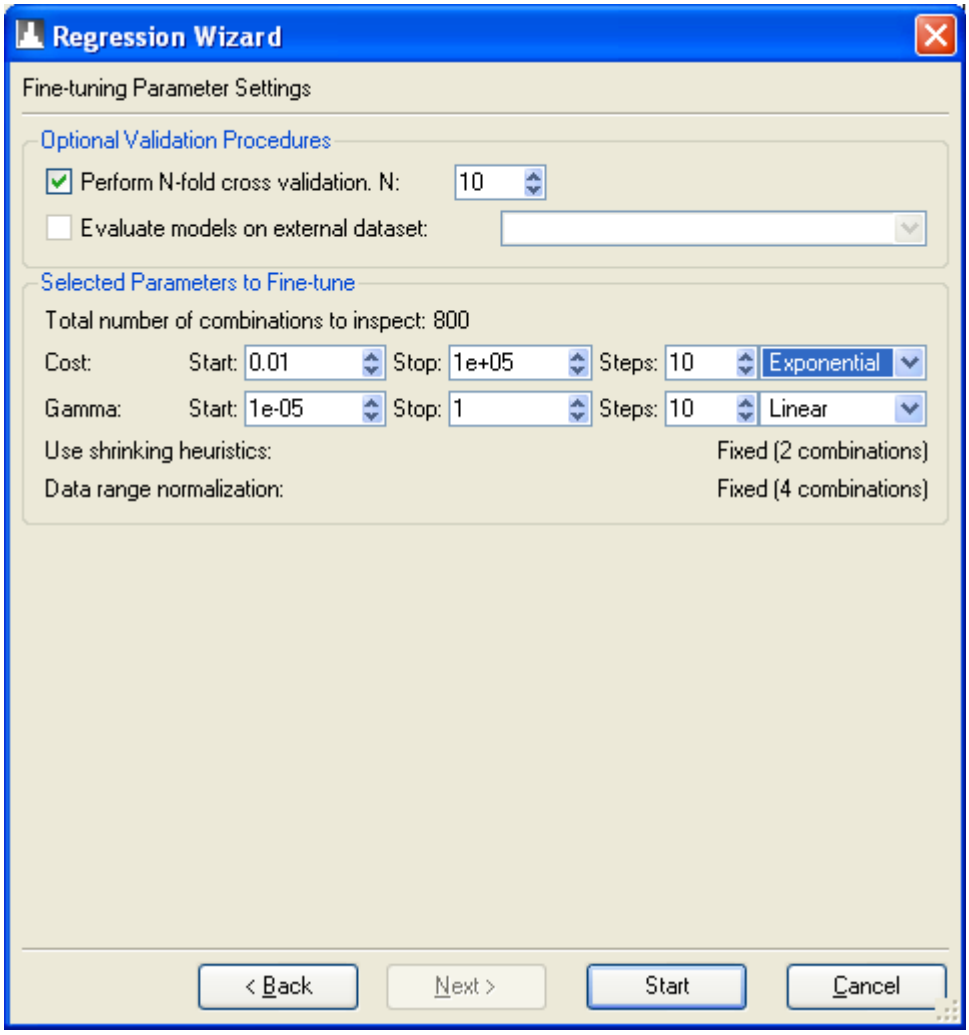


Figure 71: Fine-Tuning Parameter Settings dialog box.

The parameter fine-tuning procedure uses a grid-based search method where the parameters that are selected for fine-tuning are probed in the given intervals (with the intervals being divided by the number of steps). A regression model is then created for each set of parameter settings probed (with all other parameters kept fixed).

The quality of each model is evaluated on the provided training set using the Pearson Correlation Coefficient and the Bayesian Information Criterion (BIC) measure. BIC is used to evaluate model performance balancing model accuracy (Mean Squared Error) and model complexity (number of descriptors used in the model):

$$BIC = \ln(MSE) + (k+1)\left(\frac{\ln(n)}{n}\right)$$

where *MSE* is the Mean Squared Error, *k* is the number of descriptors used in the model, and *n* is the number of records.

In addition to the training set evaluation, it is possible to **Perform N-fold**

cross validation or **Evaluate models on external dataset** (only datasets in the workspace which contains the same descriptors as the regression model will be available in the combo box). Notice that cross validation and external dataset evaluation makes the fine-tuning process slower.

The **Selected Parameters to Fine-tune** box shows all the parameter settings that will be fine-tuned. For each parameter setting, start, stop, and step values are shown indicating the start/stop values for the given parameter and the number of steps probed. In addition, parameters using floating point representations can be probed in either linear or exponential steps (exponential stepping can be useful when probing very large ranges, e.g. The SVM cost parameter).

For example, probing the interval [0.01, 1E5] using 8 steps and exponential stepping will result in the following values being probed: 0.01, 0.1, 1, 10, 100, 1000, 10000, 100000.

Using the same interval and number of steps but linear stepping will result in: 0.01, 14285.7, 28571.4, 42857.1, 57142.9, 71428.6, 85714.3, 100000.

Notice: Parameter settings which are represented by either check boxes (true/false) or combo boxes (fixed number of entries) will have all possible options probed during the fine-tuning procedure.

After selecting start, stop, and step values for all numeric parameters, the fine-tuning procedure can be initiated by pressing the **Start** button.

When the fine-tuning procedure has finished or has been canceled by the user, the solutions found are presented in the **Parameter Fine Tuning Results** dialog box (see Figure 72).

For each solution, the corresponding Pearson correlation coefficient, BIC, and RMSE values are shown for the training set. Results for N-fold cross validation and external dataset evaluation are provided if they were included. In addition, the specific parameter value for each fine-tuned parameter is shown.

Pressing the **Add Selected Model to Workspace** button will create a regression model using the selected parameter settings and add it to the current workspace.

When pressing the **Add Results to Workspace** button, the fine-tuning results (all table entries) will be copied to a dataset (named **Fine Tuning Results**) and added to the current workspace. This option is particular useful for analyzing the fine-tuning results in more details and makes it possible to visually inspect different parameter setting combinations.

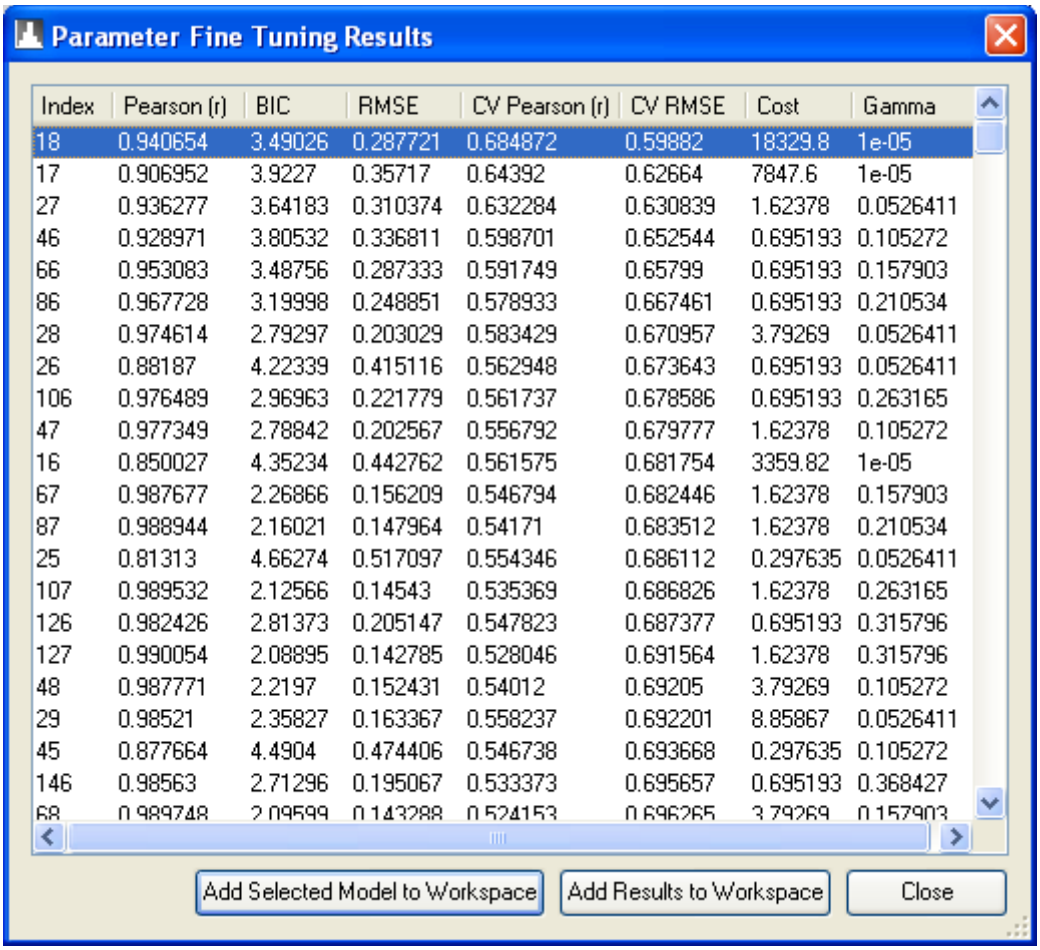


Figure 72: Parameter Fine Tuning Results dialog box.

Experimental Setup

On the final page (**Experimental Setup**), it is possible to either:

- Create and train a new regression model using the data from the dataset.
- Validate the generality of selected model parameters using cross-validation, leave-one-out validation, or percentage split validation.
- Perform feature selection.

Notice: The feature selection option is only available if feature selection has been selected in the **Select Descriptors** page.

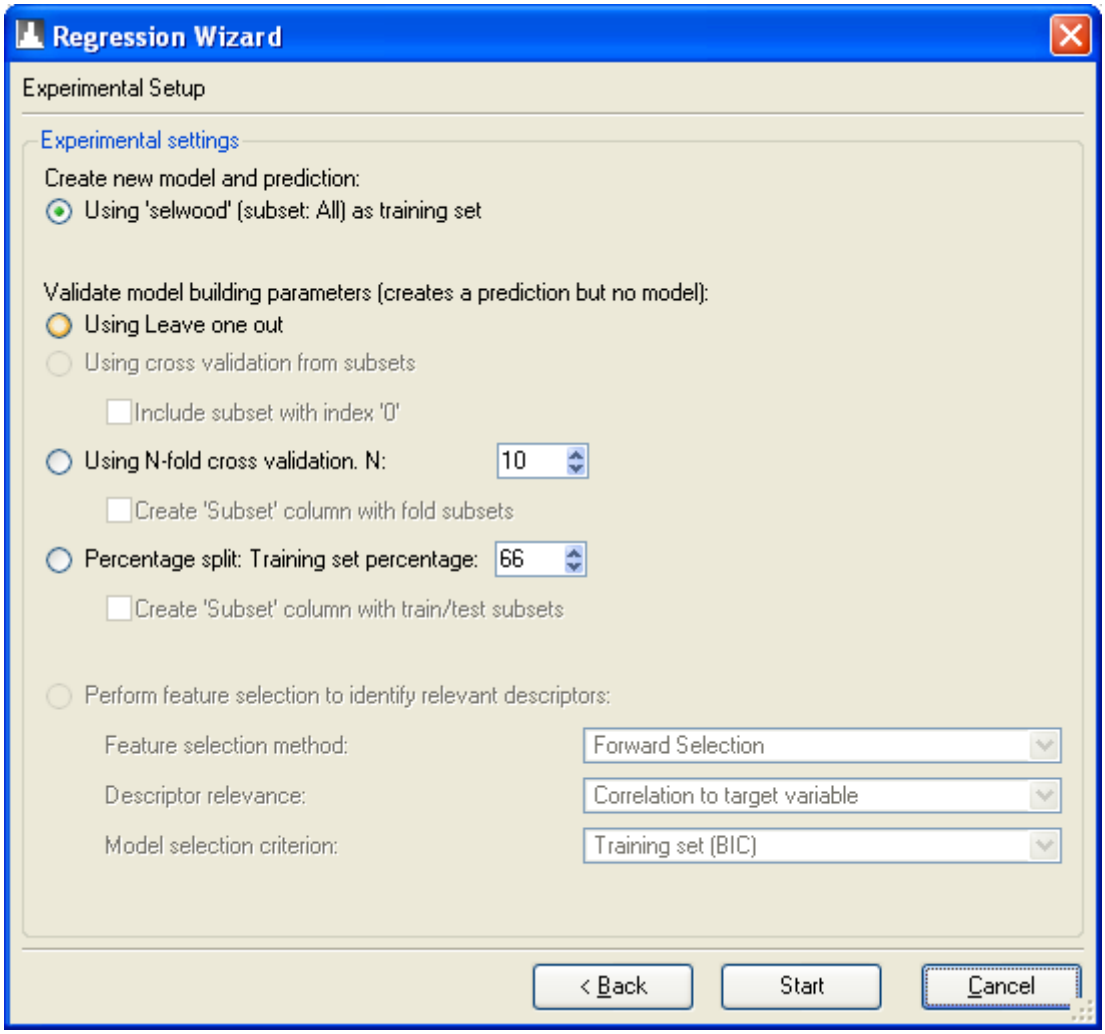


Figure 73: Choose experimental setup.

Creating a Model

When the **Create new model and prediction** option is chosen, a new regression model will be created. The model will be available in the **Workspace Explorer** window, and can be used to make predictions on other datasets. In addition, a prediction column (with predicted values of the target variable) is appended to the dataset that was used for training the model.

Validating a Model

Sometimes regression models are over-fitted, resulting in regression models performing much worse on unseen data than on the training set. Overfitting may occur if the regression model is too complex or too few records are available for model training. The complexity of a model is determined by the number of chosen descriptors (which can be pruned using feature selection)

and by the number of internal parameters in the model (such as the number of hidden layer neurons in a neural network).

There are different ways to validate the generality of a regression model:

- Test the generated model on an independent test set.
- N-fold cross validation on the training set.
- Leave-one-out validation on the training set.
- Percentage split validation.

Using an independent test set is the best solution, but is only possible when sufficient data records exist. Section 8.8 describes how to make a prediction on an external dataset using a given regression model.

In *N-fold cross validation* (N-CV), the dataset is partitioned into N subsets. N-1 subsets are then used for model training and the remaining subset is used for validation (prediction). The cross validation process is repeated N times, with each of the N subsets used exactly once for validation. Afterwards, the model accuracy (generality) is estimated as the Pearson correlation coefficient calculated from the combined prediction. Usually, N is chosen between 5 and 10. If the **Overwrite 'Subset' column with fold subsets** option is toggled on, the fold ID that identifies what fold a given record was assigned to, is stored in the Subset column.

The **Using cross validation from the 'x' subsets** option makes it possible to perform a N-fold cross validation using the subsets defined in an existing Subset column, where the number of folds corresponds to the number of subsets available. It is also possible to toggle whether the 0-subset should be included or not (records with subset ID equal to 0 may indicate that the records have not been assigned to a subset).

Leave-one-out validation (LOO) is similar to N-fold cross validation, where N is equal to the number of samples (e.g. records or observations) in the dataset.

N-CV is typically used when the dataset contains a lot of samples since LOO can be very time-consuming. However, for small datasets (e.g. less than 50 samples), LOO may provide more accurate estimates.

The *Percentage split* validation procedure divides the dataset into a training set and a test set using the percentage provided by the user (default is 66%). A regression model is trained using the training set and a prediction is made on the held-out test set afterwards. If the **Create 'Subset' column with train/test subsets** is enabled, data records will be assigned a subset ID of 1 (for training set records) and 2 (for test set records).The subset Ids will be stored in the Subset column.

Notice that the validation procedures do not create a regression model since several models are created during the validation process. Only a prediction is created indicating the accuracy of the current model setup. Notice: For

Percentage split validation, the prediction is only made for the test set (training set entries are set to 'NaN').

It is possible to create general regression models by first training a model using the N-CV or LOO procedure in order to identify promising descriptors and model training parameter settings. Therefore, the **Regression Wizard** must be invoked more than once. To aid in the selection of descriptors and parameter settings, the wizard remembers the previously used settings making it easier to adjust the parameters. When a model of high generality has been identified (using the correlation coefficient as a measure of generality), a regression model can be created using the **Create new model and prediction** option.

A way to check whether a regression model is over-fitted or not is to compare the correlation coefficient of the trained model (R_{train}) with the correlation coefficient obtained from N-CV or LOO validation (R_{cv}). If R_{train} is much higher than R_{cv}, the model is probably over-fitted.

Feature Selection

The built-in feature selection algorithms can be used to identify relevant descriptors (see Figure 74). Reducing the number of descriptors makes it easier to interpret the model, and makes overfitting less likely.

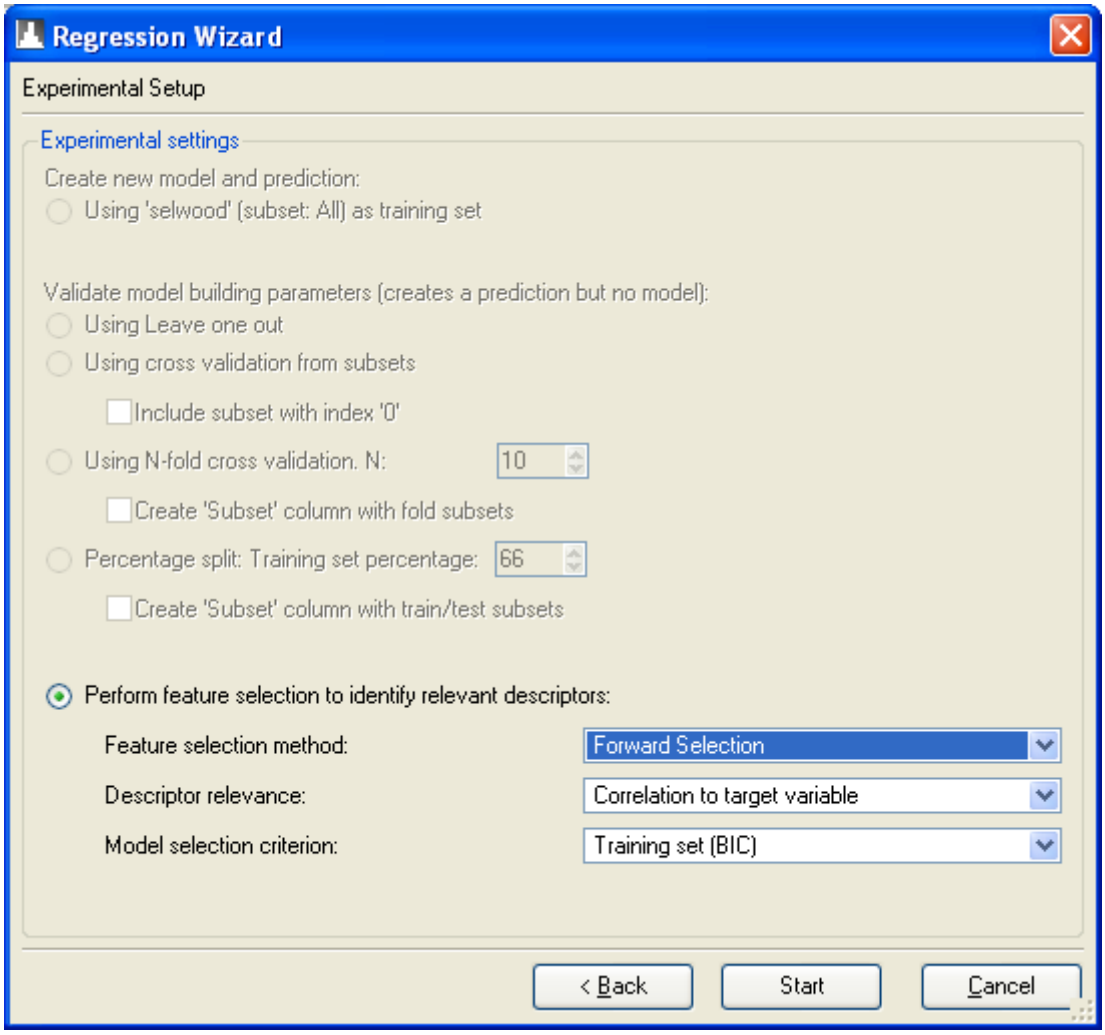


Figure 74: Feature selection options available in the Regression Wizard.

In the **Feature selection method** box it is possible to select whether *Forward Selection*, *Backward Elimination*, or *Hill Climber* should be used to identify relevant descriptors:

- *Forward Selection* begins with one descriptor and continues to add descriptors one at a time until no further improvement is possible. The descriptors are added in the order given by the chosen **Descriptor relevance**. The maximum number of descriptors that are probed at each step of the algorithm can be set in the Preferences dialog (see Chapter 13). By default all descriptors available (i.e. not already selected in previous steps) will be probed. Model improvements are evaluated using the Model selection criterion introduced below.
- *Backward Elimination* starts with all descriptors available and iteratively removes a descriptor (one at a time) until no more improvements is possible. The maximum number of descriptors that are probed at each step of the algorithm can be set in the Preferences dialog. By default all

descriptors available (i.e. not already removed in previous steps) will be probed. Model improvements are evaluated using the Model selection criterion introduced below.

- The *Hill Climber* starts with an initial solution (by default a model containing the 3 highest-ranked descriptors, see **Descriptor relevance** below for more details). The initial solution is modified using one of the three variation operators: (i) Add a randomly chosen descriptor from the set of available descriptors, (ii) Remove a randomly chosen descriptor from the current solution (if more than one descriptor is present in the solution), or (iii) Exchange a randomly selected descriptor with another descriptor from the set of available descriptors. Only one variation operator at a time is applied to modify the current solution and the operator is chosen randomly with 10% chance of using the first operator, 10% chance of using the second operator, and 80% chance of using the third operator. New solutions are created iteratively using the variation operators above. A new solution is accepted if it is better than the previous using the **Model selection criterion** described below. The algorithm is terminated when a maximum number of iterations has occurred. The number of descriptors in the initial solution and maximum number of iterations can be customized in the Preferences dialog (see Chapter 13).

Notice: When performing feature selection using the partial least squares method, the number of latent components used in the model might be altered during the feature selection procedure. If the number of descriptors selected by the feature selection method is lower than the number of latent components specified in the PLS parameter settings, the number of latent components used will be equal to the number of descriptors selected (making the PLS equivalent to MLR).

Before applying one of the feature selection methods described above, the descriptors are sorted according to the **Descriptor relevance** scheme selected. The following schemes are available:

- **Correlation to target variable**: descriptors are ranked according to the Pearson correlation coefficient between each descriptor and the target variable.
- **Coefficient Relevance** (MLR models only): Coefficient relevance scores are calculated from a MLR model using all available descriptors. Each coefficient relevance score is calculated by multiplying the coefficient value with the standard deviation of the corresponding descriptor and dividing the product with the standard deviation of the target variable. Notice: Coefficient relevance scores are only meaningful to calculate if the number of records is higher than the number of numerical descriptors used.

- **Descriptor Relevance** (PLS models only): Descriptor relevance scores are calculated from a PLS model created using all available descriptors. First, for each descriptor a *descriptor relevance score* is derived directly from the *regression vector* computed during the PLS procedure (see **[PLS 2007]** for details). Afterwards, all relevance scores are scaled by multiplying by the range of the specific descriptor they are referring to. Finally, the scores are normalized to be in the range between 0 and 100 based on the highest scoring value observed. The highest scores highlights the most relevant descriptors.
- **Relevance Score**: (Neural Network models only). The Relevance Score is calculated by following all paths from the input neuron to the output neuron (including hidden layers). For each path, the product of all the connection weights (in absolute values) is added to the score. Afterwards, all relevance scores are normalized to be in the range between 0 and 100.
- **Random Ranking**: The descriptors are assigned a random rank.

The quality/performance of each feature selection solution is evaluated using the criterion chosen in the **Model selection criterion** box. The **Cross validation (Pearson-r)** option evaluates each model using a N-fold cross validated Pearson Correlation Coefficient whereas the **Cross validation (RMSE)** option evaluates each model using N-fold cross validated root mean squared error. The **Training set (BIC)** option uses a Bayesian Information Criterion to evaluate model performance balancing model accuracy (Mean Squared Error) and model complexity (number of descriptors used in the model):

$$BIC = \ln(MSE) + (k+1)\left(\frac{\ln(n)}{n}\right)$$

where *MSE* is the Mean Squared Error, *k* is the number of descriptors used in the model, and *n* is the number of records.

In general, we recommend the BIC evaluation criterion since it avoids using the cross-validated correlation coefficients during the feature selection process (with the risk of fitting the selection of descriptors to the cross validated results). In addition, the BIC evaluation criterion gives a significant speedup compared to the N-fold cross validation approach since it only uses the training set once for each evaluation of a models performance. Afterwards, the generality of the BIC-derived model can be evaluated using the N-fold cross validation scheme described above.

When the feature selection process has finished the solutions found are presented in the **Feature Selection Results** dialog box (see Figure 75).

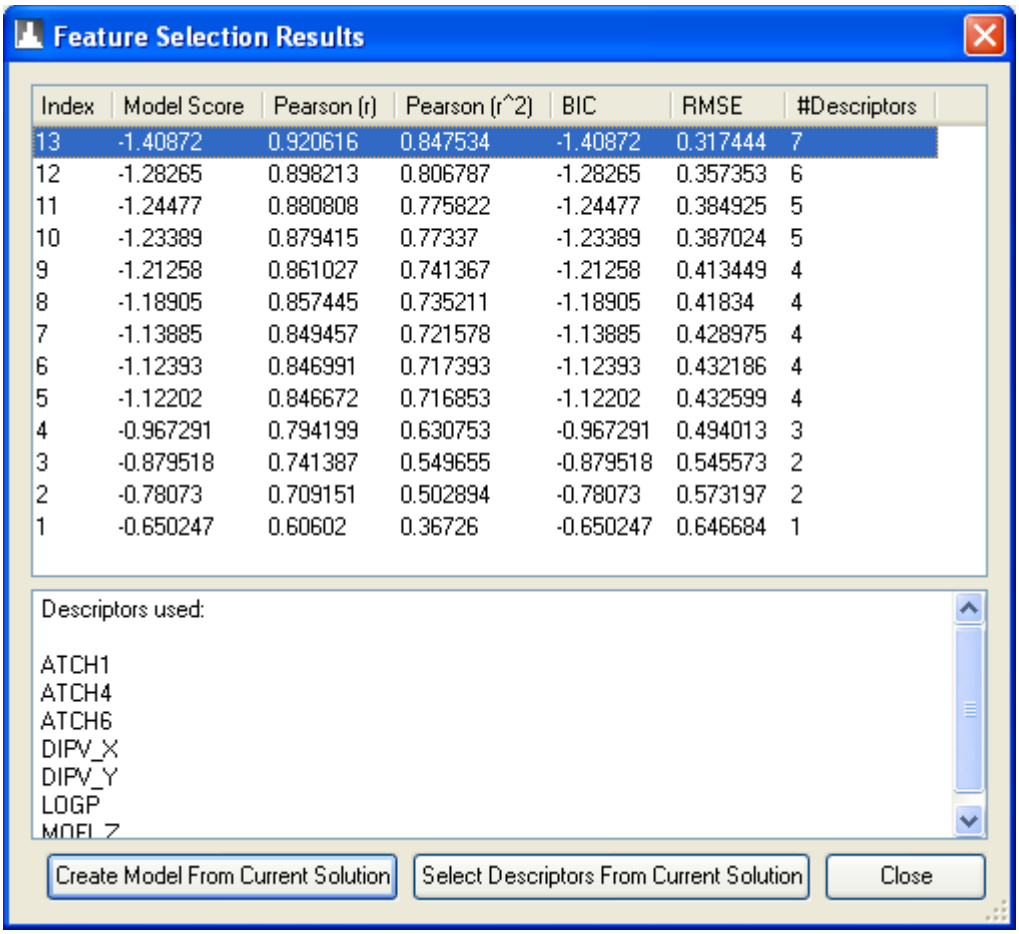


Figure 75: Feature Selection Results dialog box.

For each solution, the corresponding **Model Score** (either BIC, Pearson correlation coefficient, or RMSE), Pearson correlation coefficient, BIC, RMSE, and number of descriptors are shown.

Pressing the **Create Model From Current Solution** button will create a new model using the current solution and add it to the current workspace.

It is also possible to set the descriptors from the current solution as the default choice in the Regression Wizard by pressing the **Select Descriptors From Current Solution** button. Afterwards, a regression model can be created using the training procedure or evaluated using the LOO or N-CV procedures.

8.7 Inspecting Regression Models

Once a model has been created (or imported from a MDM file) it is possible to inspect the model details by invoking the **Model Details** dialog box from the context menu of the selected model (by right-clicking on the model with the mouse) and selecting the **Show Details...** item. An example is shown in Figure 76 where a summary of a *neural network* model is provided. For *multiple linear regression, partial least squares,* and *support vector machine* models a similar tab page is shown (except for the algorithm-specific settings).

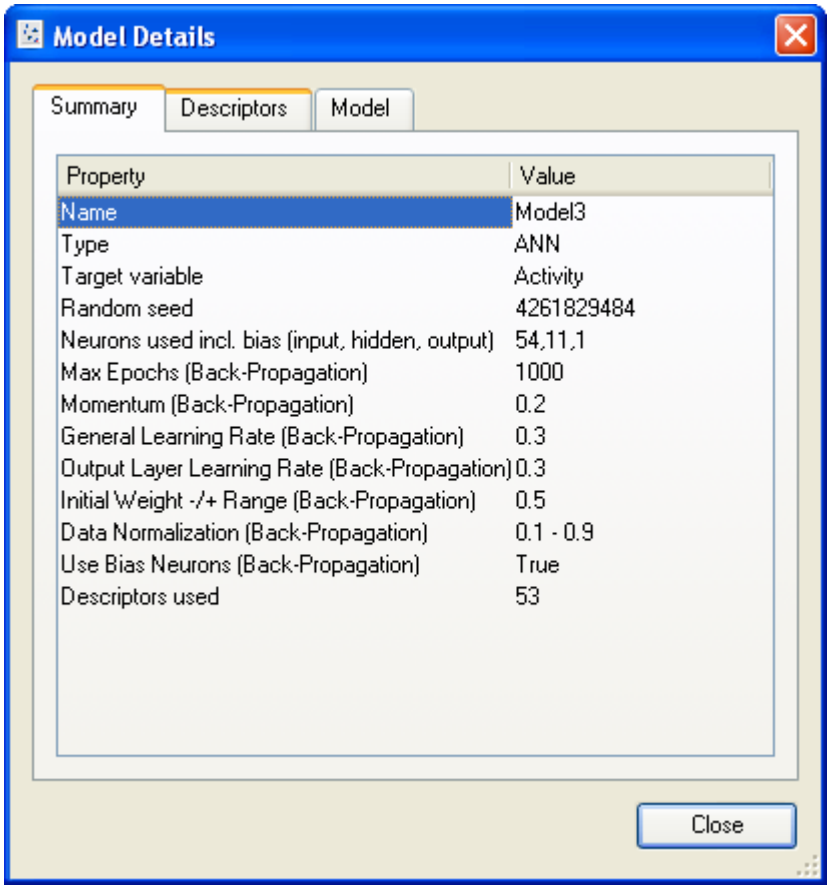


Figure 76: Model Details dialog box: Summary

The **Descriptors** tab (see Figures 77-79) lists all descriptors the model uses. More importantly, it also provides a **Relevance Score** for each descriptor (for neural network models) or a **Coefficient Relevance** for each coefficient (for multiple linear regression models) indicating how *relevant* the descriptor was during model building with respect to modeling the target variable.

Therefore, the relevance scores can be used to identify which descriptors were most suitable for modeling the target variable and new models can be built omitting descriptors with low scores (useful for manual feature selection). The **Use highlighted descriptors in Regression Wizard** button can be used to set the default choice of descriptors selected in the Regression Wizard to the descriptors currently highlighted in the list view.

The **Relevance Score** for neural networks is calculated by following all paths from the input neuron to the output neuron (including hidden layers). For each path, the product of all the connection weights (in absolute values) is added to the score. Afterwards, all relevance scores are normalized to be in the range between 0 and 100.

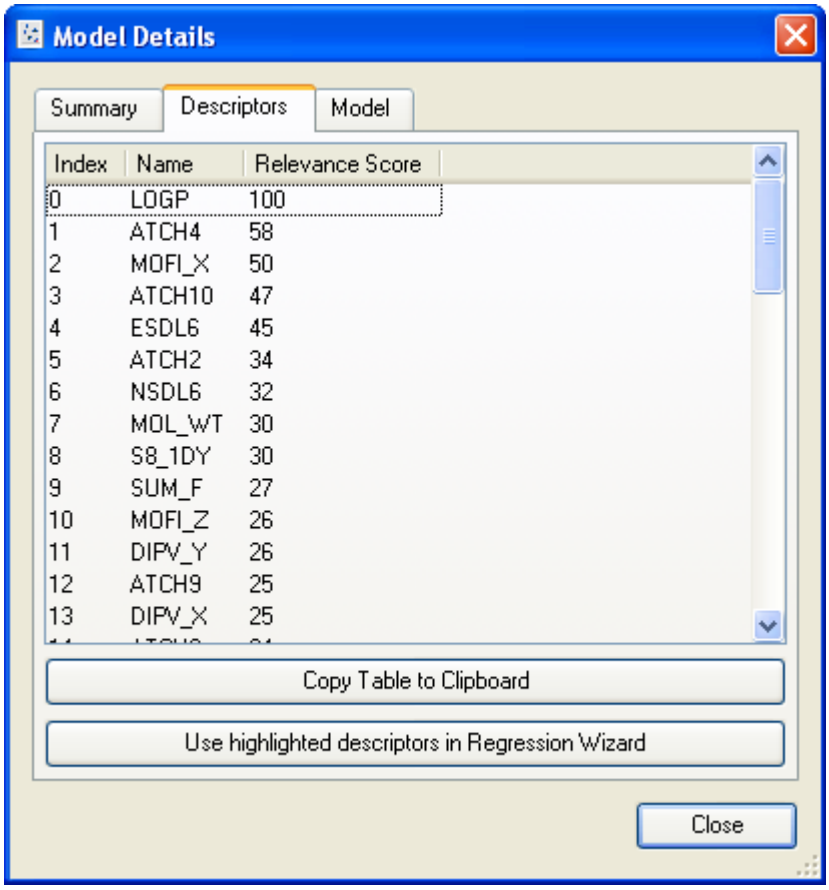


Figure 77: Model Details dialog box: Relevance scores for Neural Network models.

The **Coefficient Relevance** score for multiple linear regression is the product of the specific coefficient and the standard deviation of the corresponding numerical descriptor divided by the standard deviation of the target variable (see Figure 78).

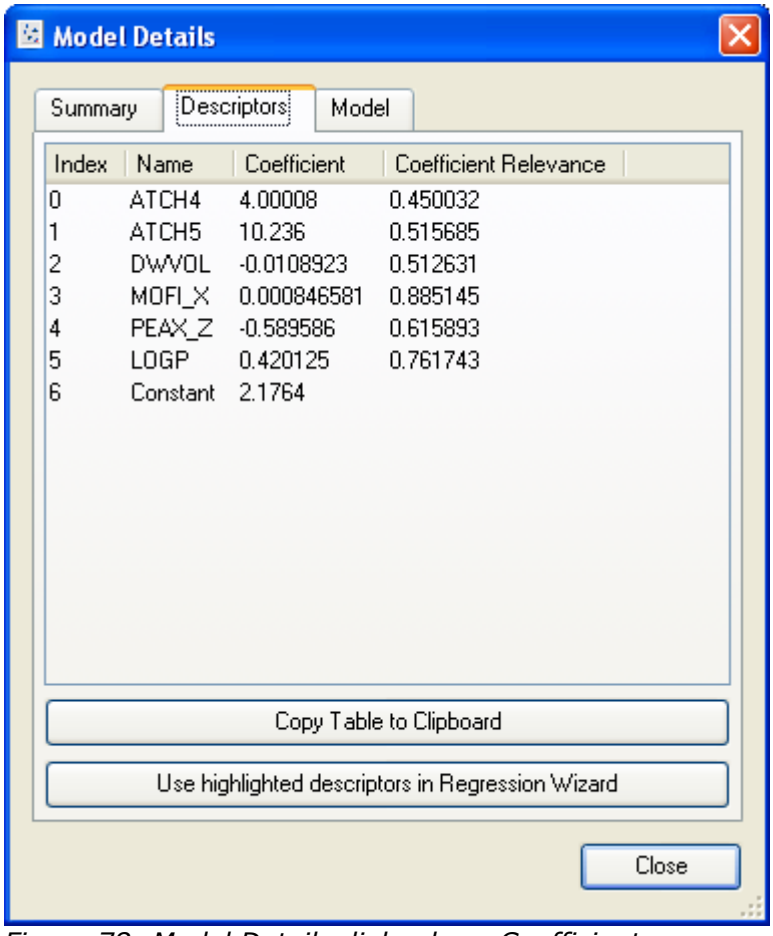


Figure 78: Model Details dialog box: Coefficient relevance scores for Multiple Linear Regression models.

The **Descriptor Relevance** score for partial least squares is derived directly from the *regression vector* computed during the PLS procedure (see **[PLS 2007]** for details). Afterwards, all relevance scores are scaled by multiplying by the range of the specific descriptor they are referring to. Finally, the scores are normalized to be in the range between 0 and 100 based on the highest scoring value observed (see Figure 79).

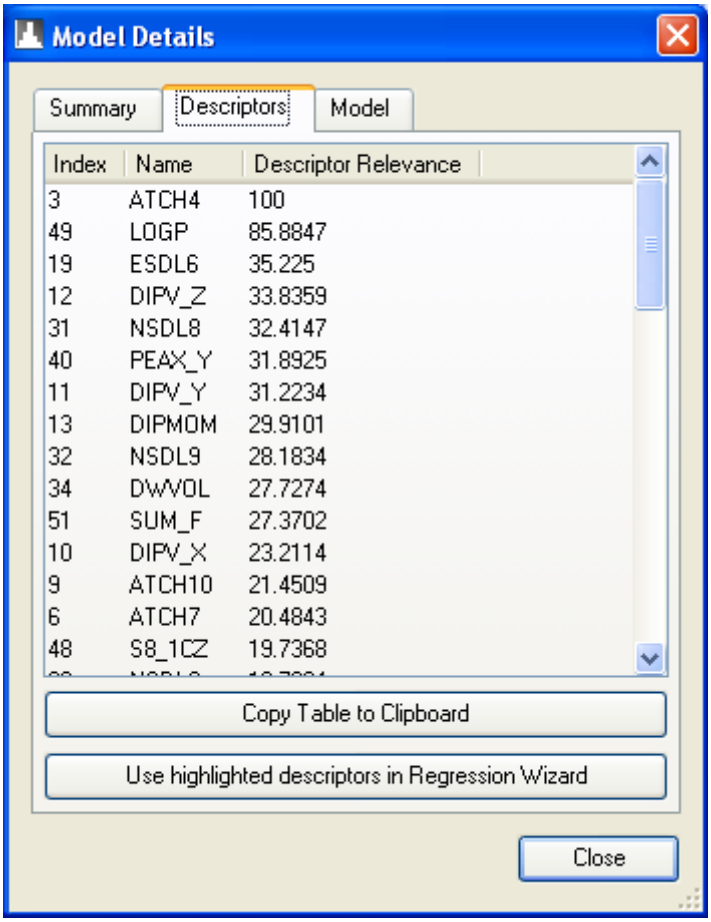


Figure 79: Model Details dialog box: Descriptor relevance scores for Partial Least Squares models.

For multiple linear regression, partial least squares, and neural network models, the final tab **Model** shows the model in details (see Figure 80-81). It is possible to copy-and-paste the pseudo-code into the Data Transformation dialog box for further usage (see Section 3.15 for more details).

For support vector machines, the final tab **Support Vectors** shows the support vectors used in the model (screenshot not shown).

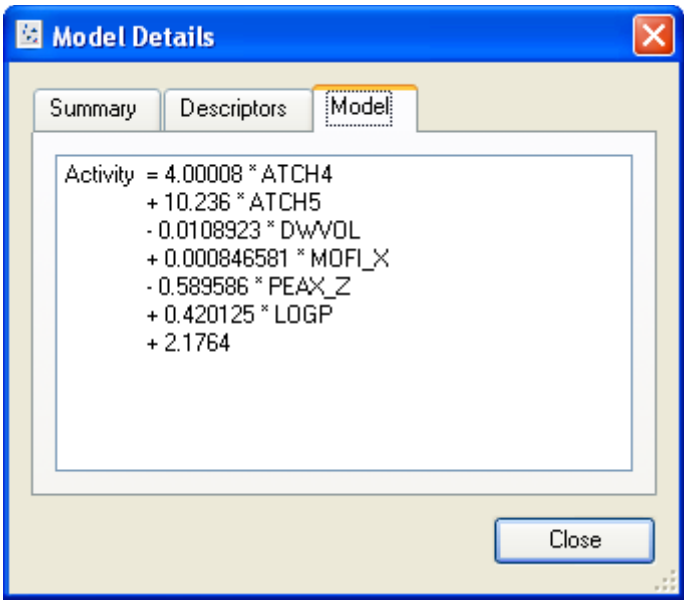


Figure 80: Example: Multiple Linear Regression model pseudo-code.

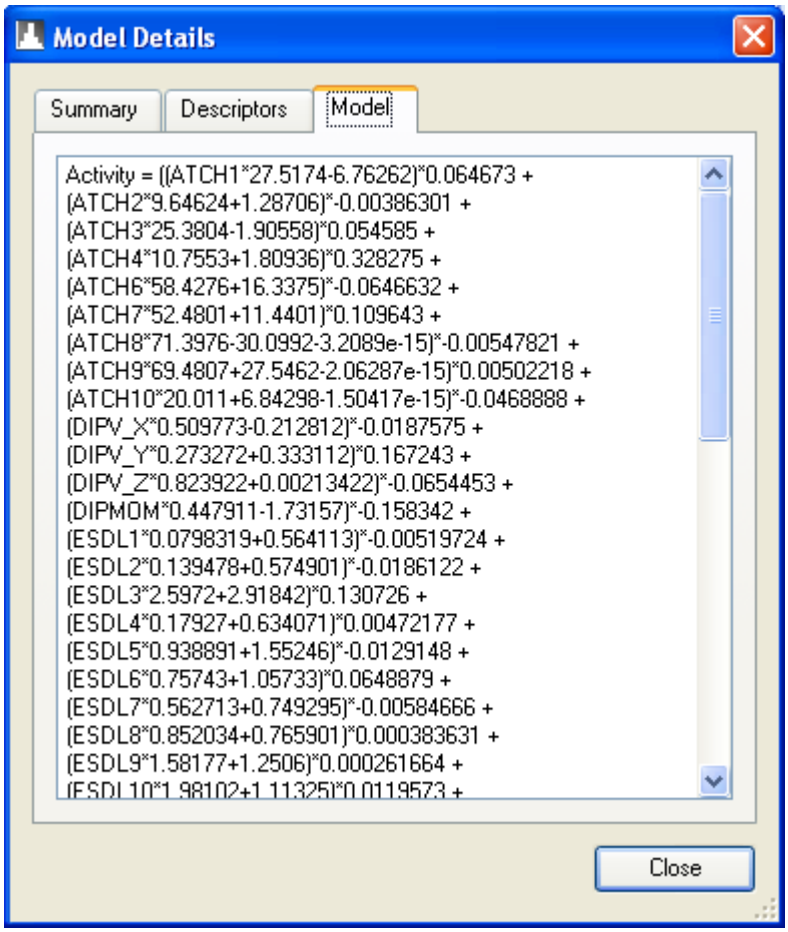


Figure 81: Example: Partial Least Squares model pseudo-code.

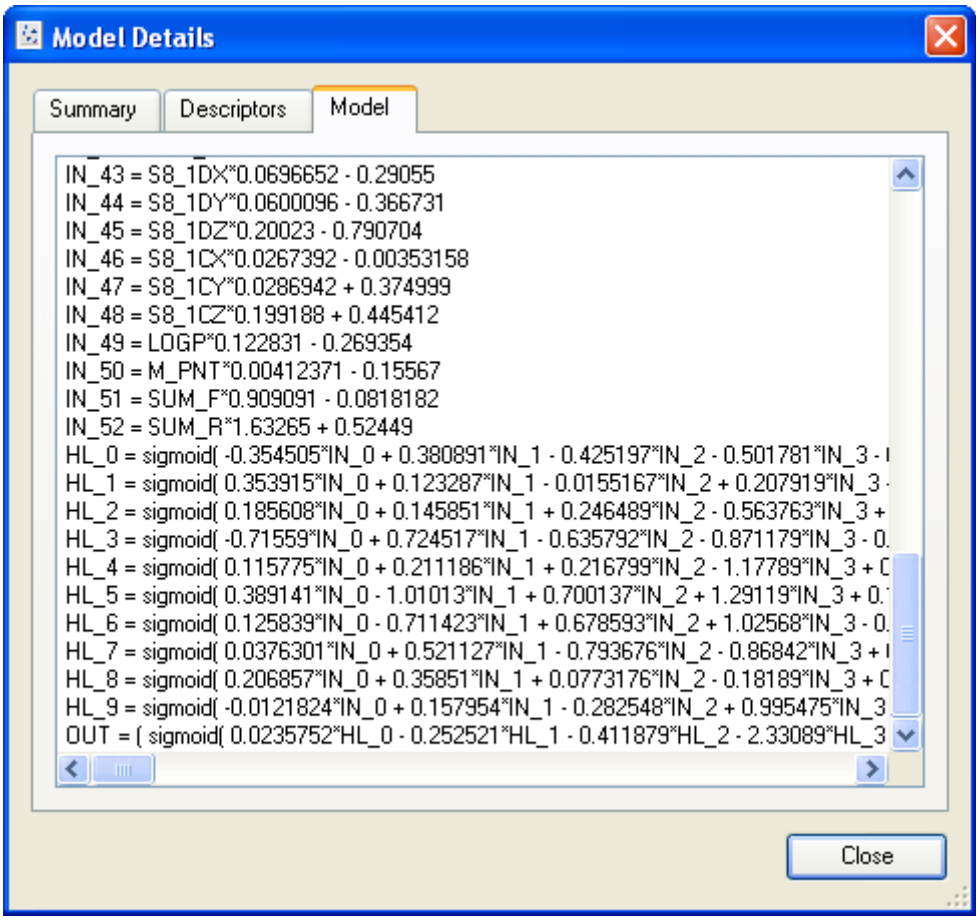


Figure 82: Example: Neural Network model pseudo-code. IN, HL, and OUT equations represent neurons in the input layer, hidden layer, and output layer, respectively.

8.8 How to Make Predictions Using an Existing Model

Once a model has been created it can be used to predict properties (defined by the model's *target variable*) of other datasets present in the workspace. See Section 3.19 for more details about making model predictions.

9 Classification

Classification is the task of assigning objects (here: data points) to one of several predefined categories (classes). In MDM, this is done by training a classification model that maps each data point (consisting of numerical descriptors X_i) to one of the pre-defined class labels Y_j.

MDM supports the following types of classification:

- The class label can be provided as either text (in a textual column) or as an integer number (in a numerical column).
- The independent descriptors used for classification have to be numerical. Categorical descriptors may be converted to numerical descriptors (see Section 3.6 for details).
- Multiple classes: MDM can perform multiclass classification.
- Singlelabel classification. MDM does not support multilabel classification: a data point may only belong to one of the multiple classes.

MDM provides two methods for classification, namely *k-nearest neighbors (KNN) and support vector machines (SVM)*. This chapter gives a short introduction to the two methods and describes how to create and evaluate classification models using the **Classification Wizard**.

For a general introduction to classification, see Chapter 4 in **[TAN 2006]**.

9.1 K-Nearest Neighbors

One of the simplest and most widely used classification algorithms is the *k*-nearest neighbors algorithm (KNN), which classifies objects based on neighboring training examples in the selected descriptor space **[TAN 2006]**.

KNN is a so-called *lazy learning* classification algorithm where the model is composed of all the data points available in the training set, i.e. no underlying model is trained explicitly. Therefore, creating the model is computationally cheap whereas classifying unknown objects is relatively expensive (requires distance calculations from the unknown object to all the objects in the model). Despite its simplicity KNN performs well in many situations.

The k-nearest neighbors (KNN) algorithm works as follows:

First, a model is created containing all the data points from the training set. Each data point is represented by a number of selected numerical descriptors and a known classification label.

Each data point to be classified is assigned to the most frequent class label (majority voting) among the *k* nearest data points found in the training set. The *k* nearest neighbors are found using a customizable distance measure, e.g. Euclidean distance. If the majority voting results in a tie, the neighbor with shortest distance to the predicted data point wins.

The best choice of k depends upon the dataset. If *k* is set too small the classification is sensitive to noise/outliers and overfitting may occur. If *k* is too large, the neighborhood may be to large to catch specific features.

A major drawback of using the majority voting scheme is that the most frequent classes in the training set tend to dominate the classification of new data points. One way to overcome this problem is to take into account the distance of each *k* nearest neighbor, e.g. weight the contribution from each class vote with the distance between the point to be classified and the neighboring data point. This weighting option is available in MDM.

9.2 Support Vector Machines

Support vector machines (SVMs) were introduced by Boser, Guyon, and Vapnik in 1992 **[BOSER 1992]**. A short introduction to SVMs and their parameter settings is provided in Section 8.4.

Similar to SVM regression, MDM uses the algorithms provided with the LIBSVM library **[LIBSVM 2001]** to train the two types of classification SVMs that are available in MDM: *c-SVM* and *nu-SVM*. See **[LIBSVM 2001]** for more details about SVMs and the c-SMV and nu-SVM variants.

9.3 Choosing a Classification Method

Each classification method has certain advantages and disadvantages:

K-nearest neighbors is very fast and there is only a few parameters to fine-tune. However, the classification accuracy depends on the distance measure and the number of neighbors (k) used. Also, the prediction of unknown data points can be computationally slow when working with large training sets.

Support vector machines are more robust with regard to outliers, but it can be difficult to find optimal parameter settings.

For all classification methods, one important aspect of model evaluation concerns the composition of the training set. If the training set used for building/training the classification methods is unbalanced, i.e. the frequency of each class label differs a lot, some classes will tend to dominate the other and lead to overoptimistic classification accuracies being reported. Instead, we recommend to use the Macroaveraged F-measure (see Appendix I: Statistical Measures for more details).

Please see Section 8.5 for more recommendations about model selection and validation.

9.4 Creating Classification Models Using the Classification Wizard

Once one or more datasets have been imported into the Workspace, new classification models can be created using the **Classification Wizard**. To start the wizard, select the classification wizard icon on the toolbar. It is also possible to select classification algorithms individually from the main menu: **Modelling** | **KNN Classification...**, or **Modelling** | **Support Vector Classification...**.

Select Dataset, Target Variable, and Descriptors

Selection of a dataset, target variable, and numerical descriptors is identical to the procedures for regression models introduced in Section 8.6.

The only exception is that the target variable used for classification can be chosen from both numerical and textual descriptors.

Customize Training Algorithm

The algorithms used for training classification models can be customized in the **Customize Training Algorithm** page (see Figures 43-84). Details about the shuffle dataset and random seed options are described in Section 8.5. Specific details about the classification algorithm settings are described below.

K Nearest Neighbors Settings

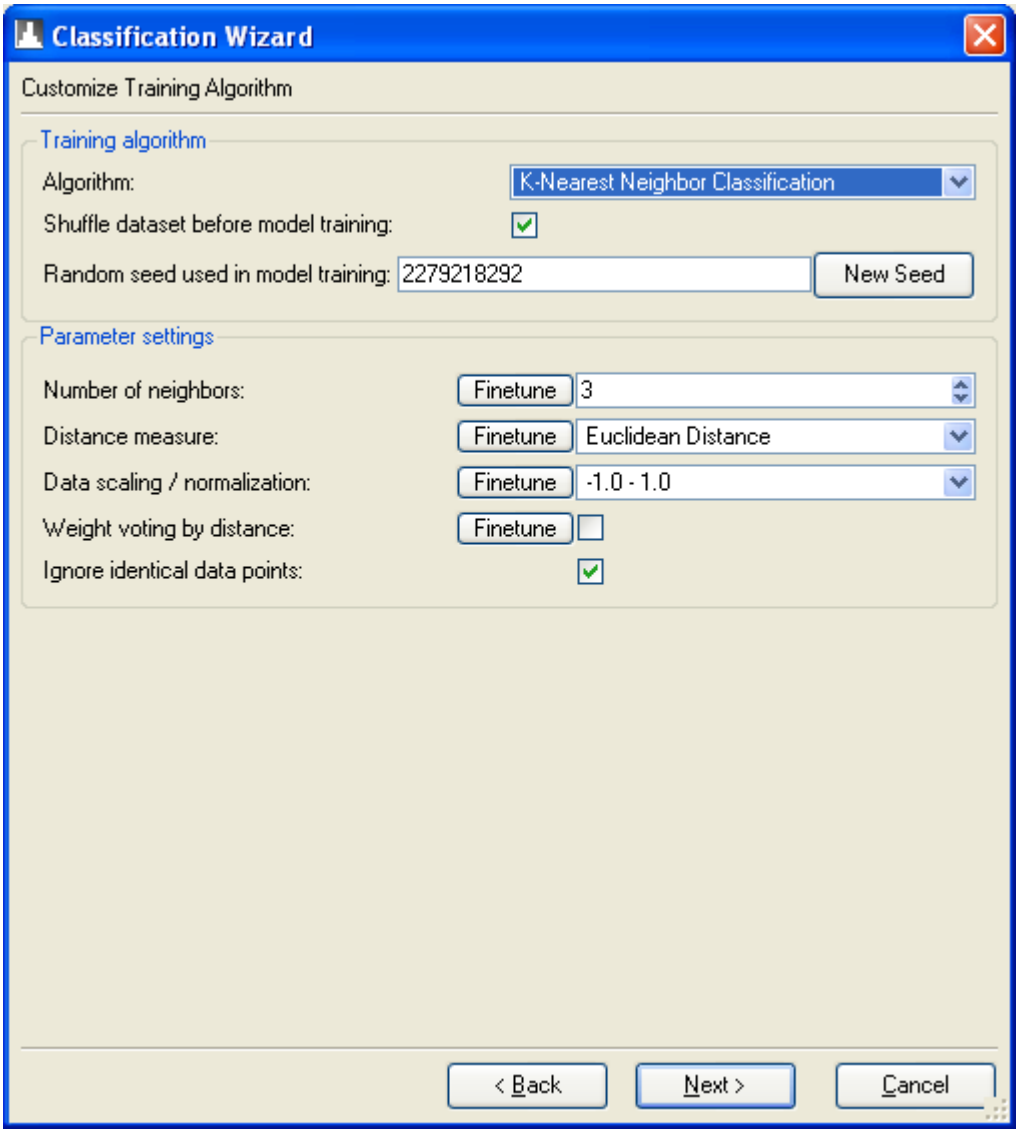


Figure 83: Parameter settings for the K-Nearest Neighbor algorithm.

The **Number of neighbors** parameter is used to specify the number of neighbors that will be used when classifying a given data point. The default value is 3.

There is no general criterion for deciding how many neighbors to employ. A common approach is to build models with increasing numbers of neighbors using cross validation and choose the model with highest classification accuracy or highest Macroaveraged F-measure value. In MDM, this is easily done by fine-tuning the **Number of neighbors parameter** (by toggling on the **Finetune** button next to the parameter setting).

The distance in the high-dimensional space is calculated using the specified **Distance measure**. The measures are described in more details in Section

11.9). By default, the Euclidean distance measure is used.

The **Data scaling / normalization** option indicates which scaling/normalization procedure should be applied (interval, auto-scaling or mean-centering) to the dataset before the model is created or if none should be applied (if the dataset has been normalized beforehand). The default choice (**-1.0 – 1.0**) is recommended. Notice that the changes made to the dataset is stored as part of the model, which makes it possible to reuse the model on other datasets without the need for manually scaling/normalizing the data.

The **Weight voting by distance** parameter is used to toggle on distance weighting when performing majority voting of the neighbors found. If the distance measure is to be minimized, each vote will be weighted by *1/distance*. For maximization measures (e.g. Tanimoto Coefficient), each vote is weighted by *distance*.

Ignore identical data points toggles whether identical data points should be included in the distance calculation or not.

Support Vector Machine Settings

The SVM settings are identical to the ones used for regression-based SVMs (described in Section 8.6 'Parameter settings for the Support Vector Machine algorithm') except for the **Include probability estimation** option.

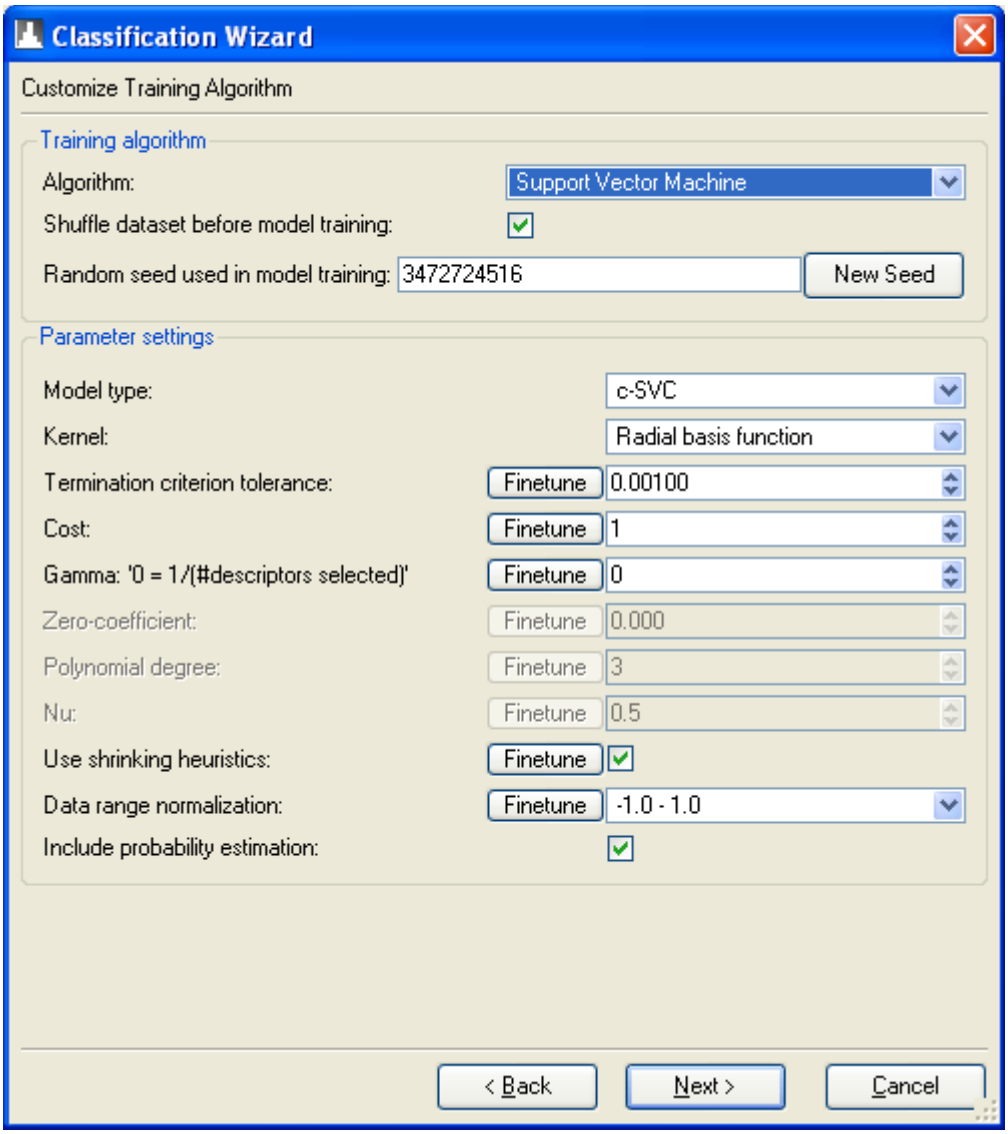


Figure 84: Parameter settings for the Support Vector Machine algorithm.

Include probability estimation toggles whether or not the model should provide a probability estimate of each classified data point. The probability estimate is a measure for how reliable the class assignment made by the classification method was (from the model's perspective). It is important to note that an estimate of 0.95 does not guarantee that the class assignment has a chance of 95% of being correct - it is only an estimate of the quality of the prediction. The probabilities for SVMs are estimated using a method introduced by Wu et al. **[WU 2004]** (referenced in the paper as the 'second approach').

Notice: The KNN algorithmic settings does not include an option for probability estimation since class probabilities are not derived during the model training for KNN models. For KNN models, class probabilities are estimated during classification using the ratio between the number of majority votes obtained

and the number of neighbors (k) queried. For low k-values the probability estimates get very rough.

Experimental Setup

The procedures for creating and validating classification models are very similar to the procedures described for regression models so they will not be introduced further. See Section 8.6 for details.

Fine-Tuning Parameter Settings

Similar to regression models, the parameters used by the training algorithm can be fine-tuned. The procedure for fine-tuning parameters is identical to fine-tuning regression parameter settings except for the evaluation measures used (here: Classification accuracy and Macroaveraged F-measure). See Section 8.6 for more details.

Notice: For unbalanced datasets the Macroaveraged F-measure should be used as the primary evaluation measure (see Appendix I: Statistical Measures for more details).

Feature Selection

Relevant descriptors for classification can be identified using the built-in feature selection algorithms. The feature selection algorithms and settings are similar to the ones introduced for regression models (see Section 8.6 for details).

The only differences between feature selection for regression models and classification models are:

- **Descriptor relevance** option is restricted to **Random Ranking** only.
- The quality/performance for each feature selection solution is evaluated using either the cross-validated Macroaveraged F-measure or the cross-validated Classification Accuracy.

9.5 Inspecting Classification Models

Similar to regression models, details about a classification model can be inspected by invoking the **Model Details** dialog box from the context menu of the selected model (by right-clicking on the model with the mouse) and selecting the **Show Details...** item.

Another useful approach is to visualize the class probabilities estimated by the classification model. The class probabilities can be created as numerical columns in the dataset and visualized using the 2D or 3D plotters. Class probability columns are created from the Classification Wizard (by toggling on

the class probability check-boxes located in the experimental setup tab page) or from the Model Prediction dialog (see Section 3.19 for details).

The Spring-Mass Maps (introduced in Chapter 5) can be used to project the probabilities to 2D or 3D. Figure 85 below shows a Spring-Mass Map that visualizes the class probabilities obtained by a SVM classification model (using the Iris plant dataset, see e.g. Section 11.5 for details). The data points are colored according to their 'true' class labels. Overall, most of the data points are correctly classified as indicated by the three main clusters (the data points in the clusters have been *spread out* using the spread slider in order to see if some data points were misclassified).

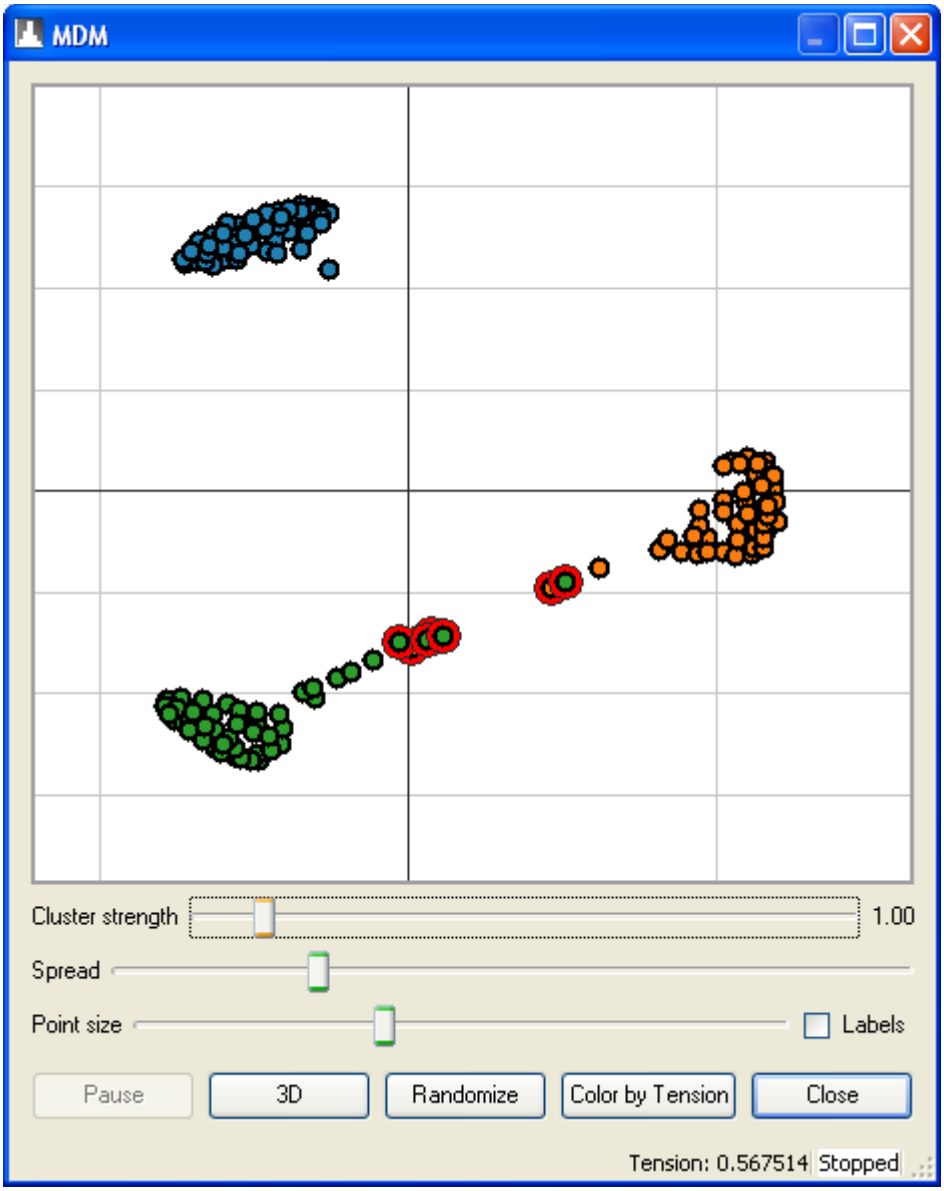


Figure 85: Visualization of class probabilities using a Spring-Mass Map.

Most of the data points have a class probability near 1 but some of the data points are located between the green and orange-colored clusters representing uncertainties in the class prediction.

Some of these 'uncertain' data points (indicated with red circles) are actual borderline cases, which could belong to either of the two classes (green or orange-colored). Figure 86 below show a 2D plot of the same data points in descriptor-space (for two selected descriptors). Here, the same data points are located between the two classes, as was also observed from their estimated class probabilities.

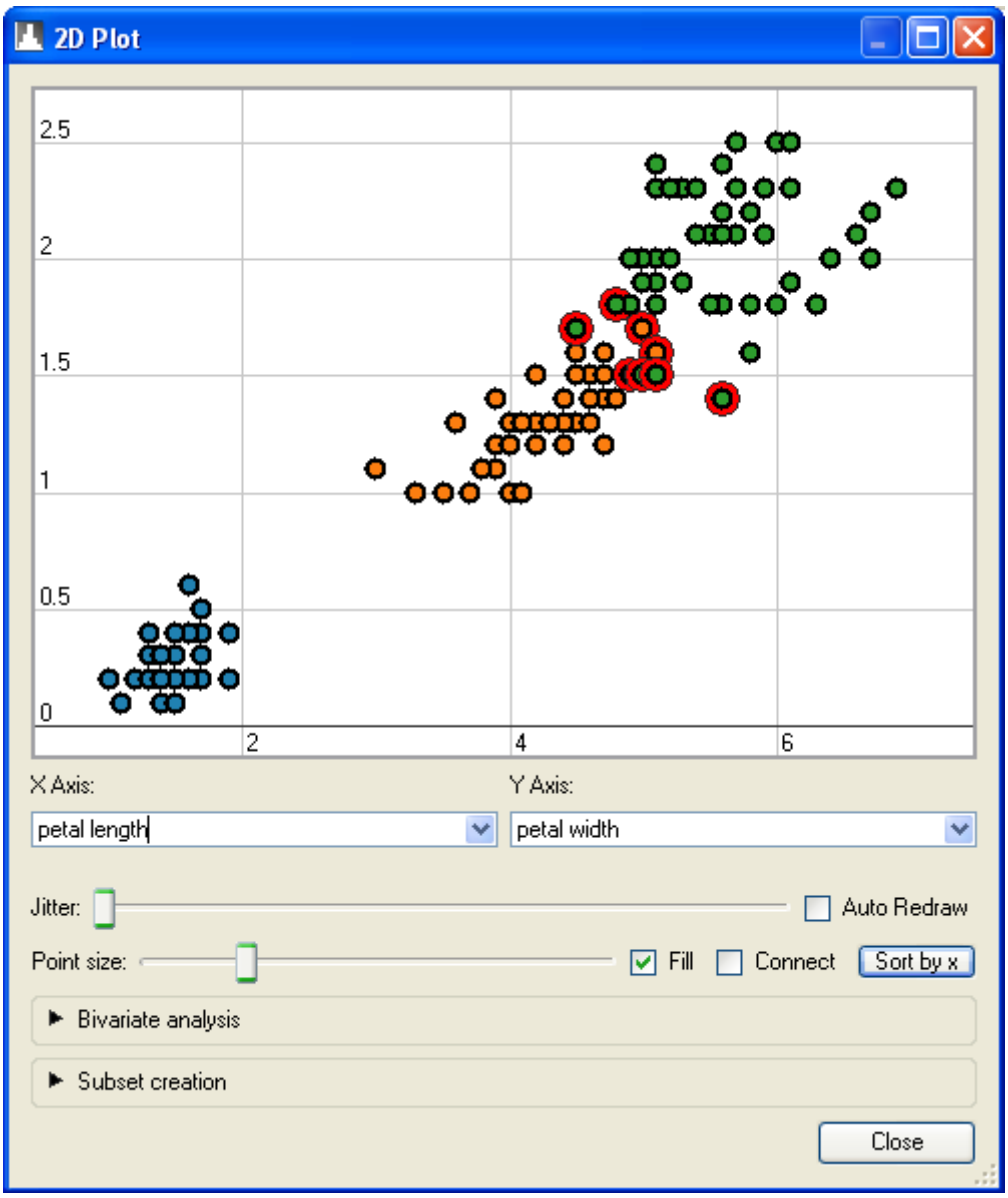


Figure 86: Plot of Iris data. Data point colors indicate class membership. Selected data points indicate borderline classifications.

Finally, it is also possible to get an idea of the overall classification performance by coloring each data point according to its highest probability estimation obtained (stored in the 'Class Probability' column). Figure 87 shows a SVM classification of the Iris plant dataset. Here, most data points obtain very high class probability estimations (colored yellow). Some of the data points have very low class probability estimations (colored red) indicating that the class assignment might not be correct. For this particular example, these low probability estimations are occurring for borderline cases only.

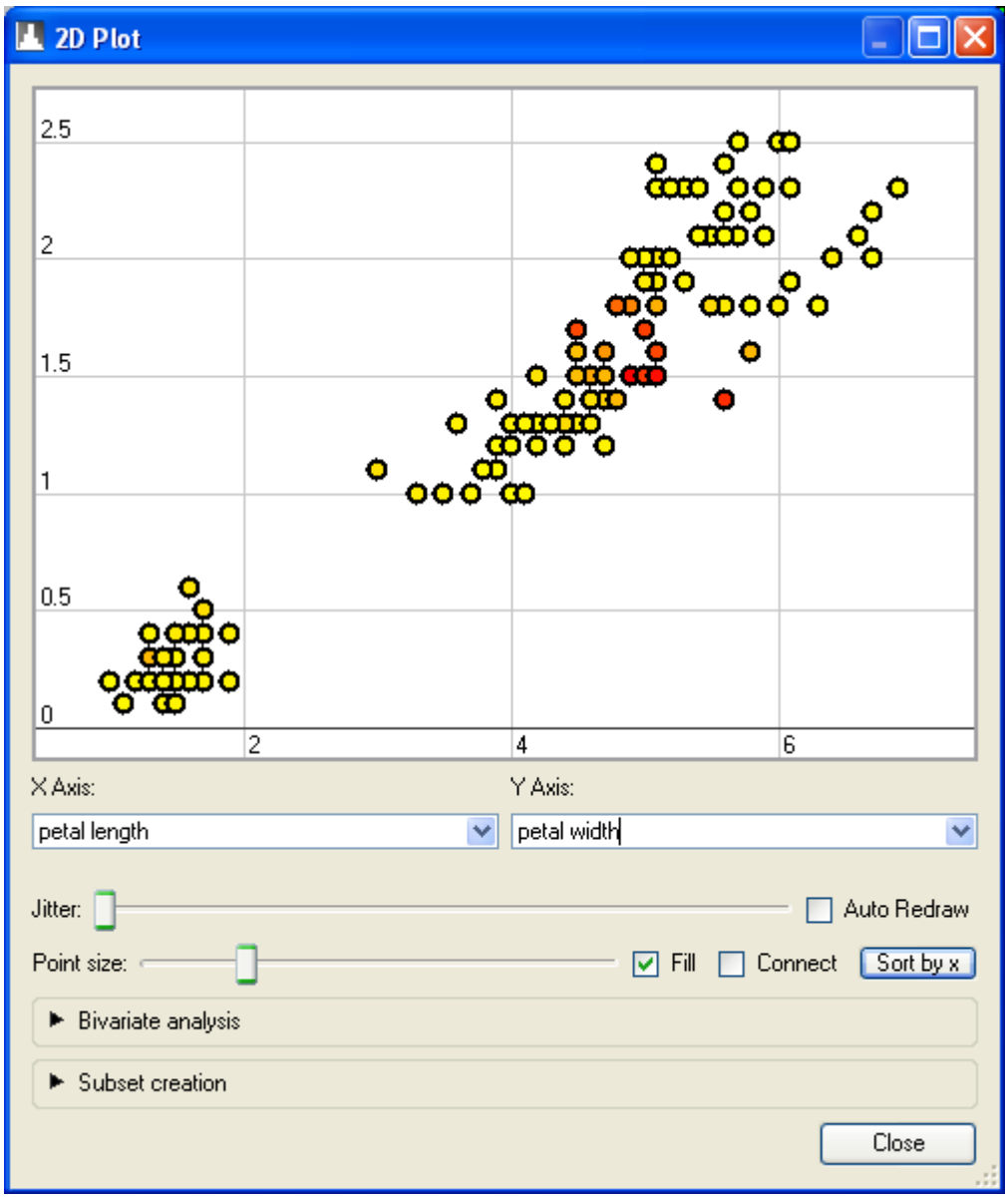


Figure 87: Visualization of estimated class probabilities. Yellow points indicate high probabilities whereas red points indicate low probabilities.

9.6 How to Make Classifications Using an Existing Model

Once a model has been created it can be used for classification of other datasets present in the workspace.

Notice that classification predictions are represented as textual columns in the dataset (even for numerical classes) but they can be converted to numerical columns if needed.

See Section 3.19 for more details about making classifications.

10 Principal Component Analysis

10.1 General Overview

Principal component analysis (PCA) is a widely used method for reducing the dimensionality of a dataset, where dimensionality refers to the number of numerical descriptors used to describe the dataset. When applying PCA on a dataset, the result is a new set of descriptors referred to as *principal components*. Each of the derived principal components is a linear combination of the numerical descriptors included in the analysis (see example in Figure 92). PCA is useful for datasets where significant correlations exist between some of the numerical descriptors.

Overall, the first principal component is chosen to describe the maximum amount of variance in the dataset and the following principal components account for the variance in the dataset not already explained by previous principal components. The total number of principal components available equals the smallest number of either records or selected numerical descriptors from the dataset.

Mathematically, PCA is done by finding the eigenvectors of the covariance matrix by diagonalization. See **[TAN 2006]** for a detailed description of the mathematical basis of PCA.

10.2 Performing a Principal Component Analysis

To perform a principal component analysis, invoke the **PCA** dialog box from the menu bar: **Modelling** | **Principal Component Analysis...**.

The first page in the dialog box makes it possible to choose which **numerical columns** (descriptors) should be included in the analysis. Remember to exclude columns with numerical identifiers or other types of data which should

not be included in the analysis (e.g. subset columns or previous principal components). It is also possible to **select a scaling or normalization method** although the default choice (**Auto scaling**) is recommended (see Section 3.5 for details about the scaling and normalization options).

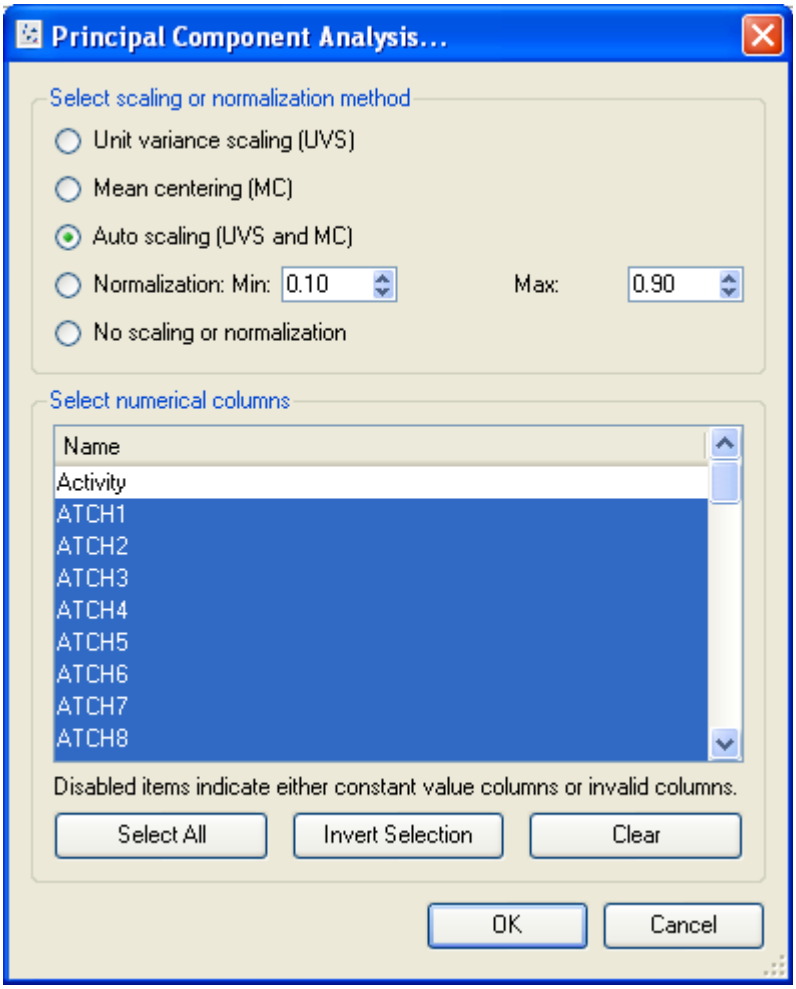


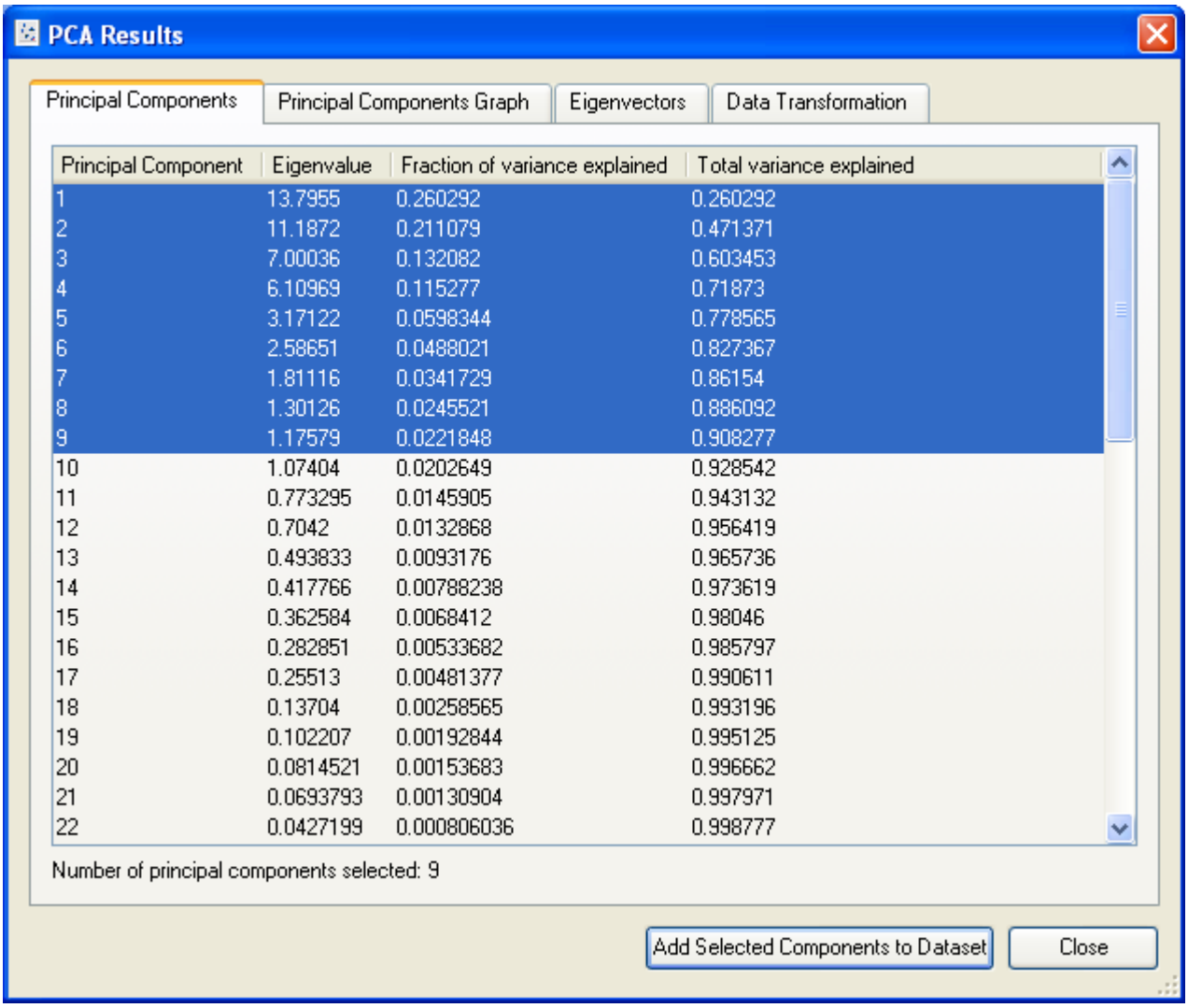
Figure 88: PCA dialog box: Select numerical descriptors to include in the principal component analysis.

When pressing the **OK** button the **PCA Results** dialog box is shown (see Figure 89).

On the first tab page (**Principal Components**), each of the principal components is listed together with the fraction of the variance explained in the dataset. The first principal component corresponds to the largest eigenvalue, the second principal component corresponds to the second-largest eigenvalue and so forth. In addition, the **Fraction of variance explained** and **Total variance explained** are shown.

It is possible to select the number of principal components to add to the current dataset by clicking on the items in the table. Pressing the **Add Selected Components to Dataset** button will add the selected principal

components to the dataset. The principal components will be named PC 1, PC 2, etc. If these names are already used in the dataset, each name will be appended with an unique index.



Principal Component	Eigenvalue	Fraction of variance explained	Total variance explained
1	13.7955	0.260292	0.260292
2	11.1872	0.211079	0.471371
3	7.00036	0.132082	0.603453
4	6.10969	0.115277	0.71873
5	3.17122	0.0598344	0.778565
6	2.58651	0.0488021	0.827367
7	1.81116	0.0341729	0.86154
8	1.30126	0.0245521	0.886092
9	1.17579	0.0221848	0.908277
10	1.07404	0.0202649	0.928542
11	0.773295	0.0145905	0.943132
12	0.7042	0.0132868	0.956419
13	0.493833	0.0093176	0.965736
14	0.417766	0.00788238	0.973619
15	0.362584	0.0068412	0.98046
16	0.282851	0.00533682	0.985797
17	0.25513	0.00481377	0.990611
18	0.13704	0.00258565	0.993196
19	0.102207	0.00192844	0.995125
20	0.0814521	0.00153683	0.996662
21	0.0693793	0.00130904	0.997971
22	0.0427199	0.000806036	0.998777

Figure 89: PCA Results: A list of the found principal components.

Often, principal components accounting for 80-90% of the **Total variance explained** are selected, the rest are ignored (for instance when doing principal component regression, see Section 10.3).

On the second tab page (**Principal Components Graph**), it is possible to visually inspect how much accumulated variance a subset of the principal components explains (see Figure 90).

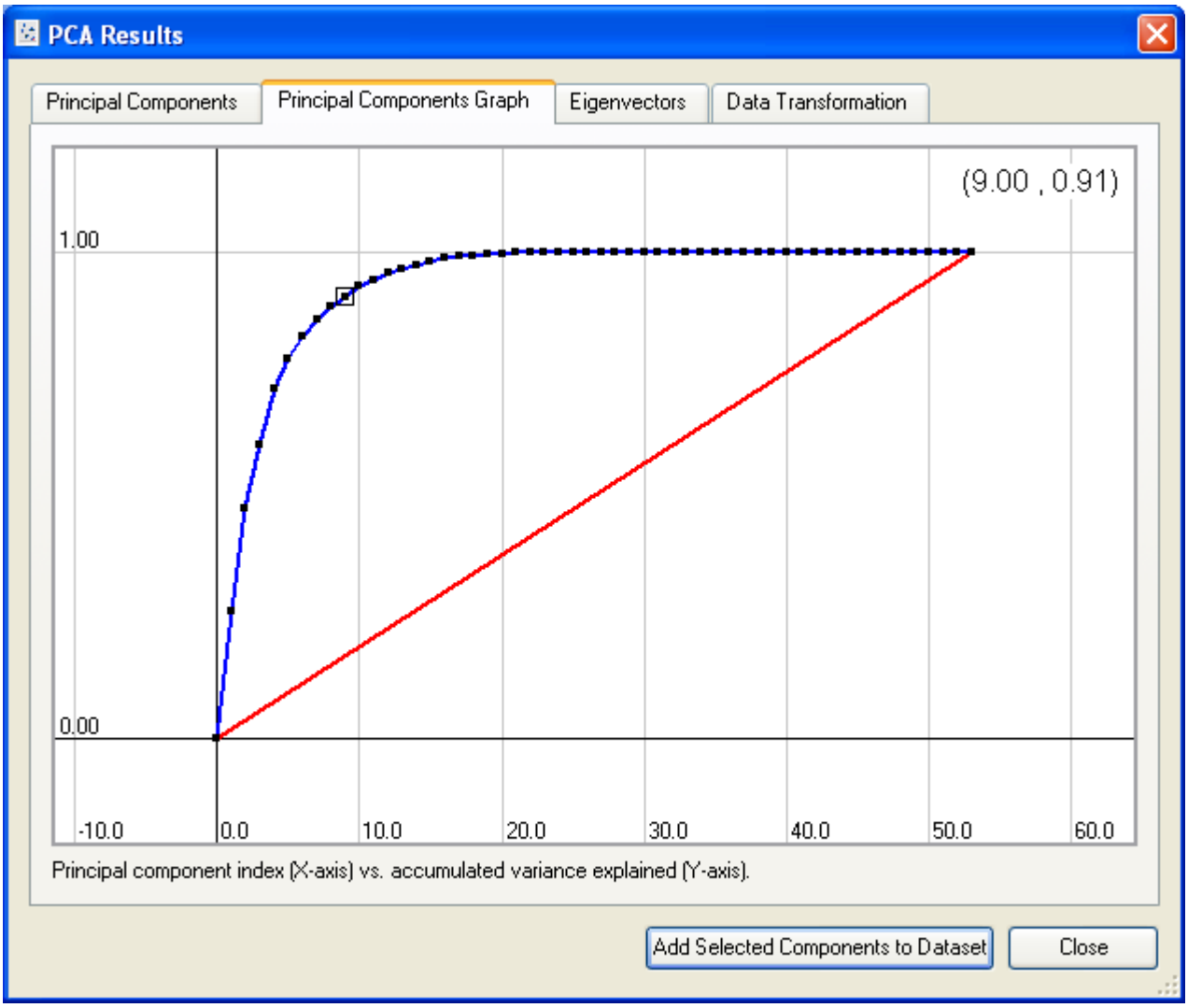
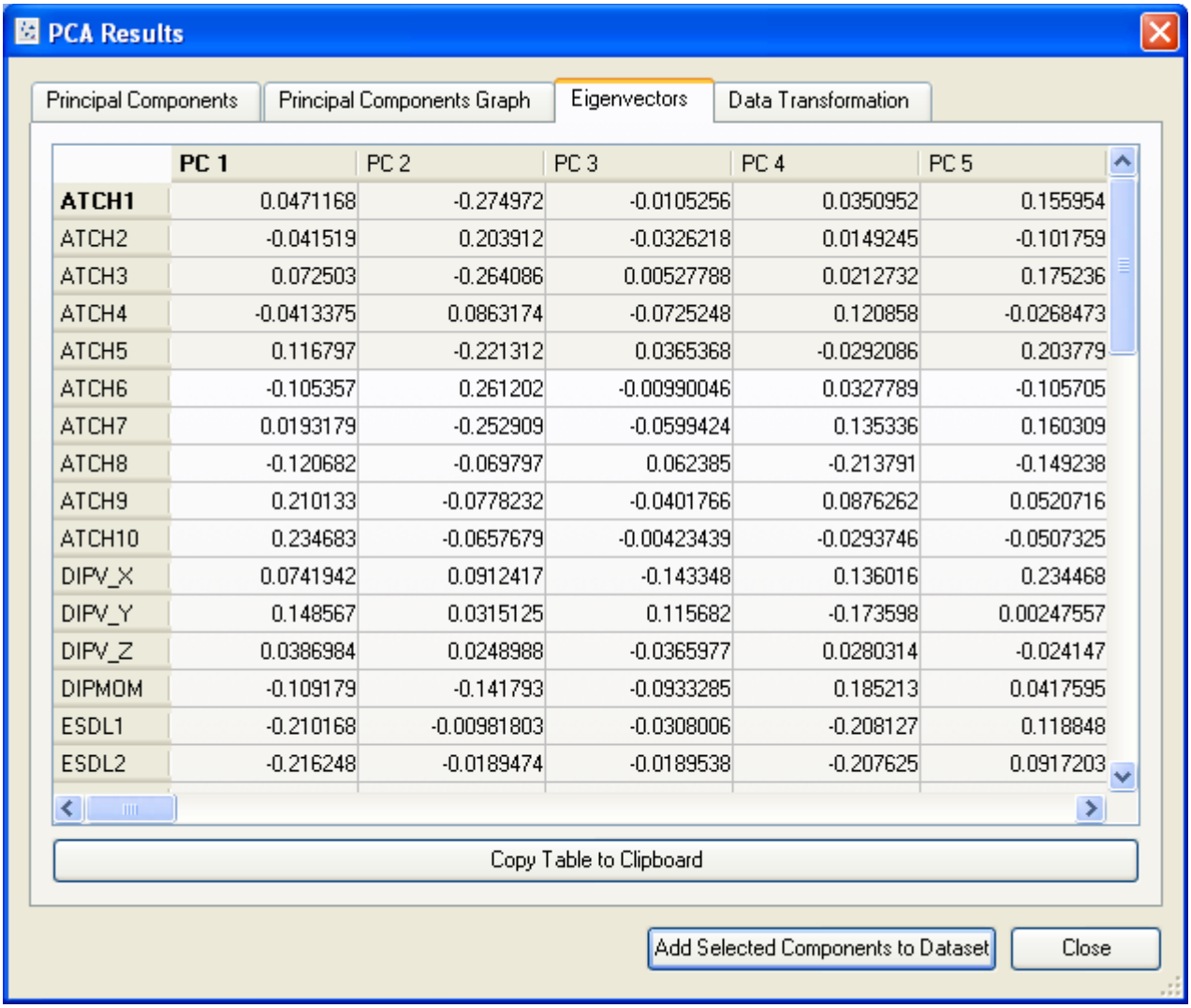


Figure 90: PCA Results: A plot of the principal components included versus the accumulated variance explained. For comparison, the red line shows the expected graph for completely uncorrelated components.

The third tab page (**Eigenvectors**) shows all the eigenvectors calculated from the PCA run. The table can be copied to the clipboard by pressing the **Copy Table to Clipboard** button.



	PC 1	PC 2	PC 3	PC 4	PC 5
ATCH1	0.0471168	-0.274972	-0.0105256	0.0350952	0.155954
ATCH2	-0.041519	0.203912	-0.0326218	0.0149245	-0.101759
ATCH3	0.072503	-0.264086	0.00527788	0.0212732	0.175236
ATCH4	-0.0413375	0.0863174	-0.0725248	0.120858	-0.0268473
ATCH5	0.116797	-0.221312	0.0365368	-0.0292086	0.203779
ATCH6	-0.105357	0.261202	-0.00990046	0.0327789	-0.105705
ATCH7	0.0193179	-0.252909	-0.0599424	0.135336	0.160309
ATCH8	-0.120682	-0.069797	0.062385	-0.213791	-0.149238
ATCH9	0.210133	-0.0778232	-0.0401766	0.0876262	0.0520716
ATCH10	0.234683	-0.0657679	-0.00423439	-0.0293746	-0.0507325
DIPV_X	0.0741942	0.0912417	-0.143348	0.136016	0.234468
DIPV_Y	0.148567	0.0315125	0.115682	-0.173598	0.00247557
DIPV_Z	0.0386984	0.0248988	-0.0365977	0.0280314	-0.024147
DIPMOM	-0.109179	-0.141793	-0.0933285	0.185213	0.0417595
ESDL1	-0.210168	-0.00981803	-0.0308006	-0.208127	0.118848
ESDL2	-0.216248	-0.0189474	-0.0189538	-0.207625	0.0917203

Figure 91: PCA Results: The eigenvectors found.

The final tab page (**Data Transformation**), lists all the principal components as linear combinations of the numerical descriptors included in the analysis. Notice: The values used for scaling or normalizing the data points are also included. It is possible to copy the equations to the clipboard (by pressing **CTRL+C**).

The equations can be used directly in the **Data Transformation** dialog box to recalculate the principal components (pasting the equations from the clipboard buffer using **CTRL+V**). The data transformation step is important when using PCA regression (see Section 10.3 for details) to make a prediction of another external dataset, since the external dataset needs to include the principal components used in the PCA regression model.

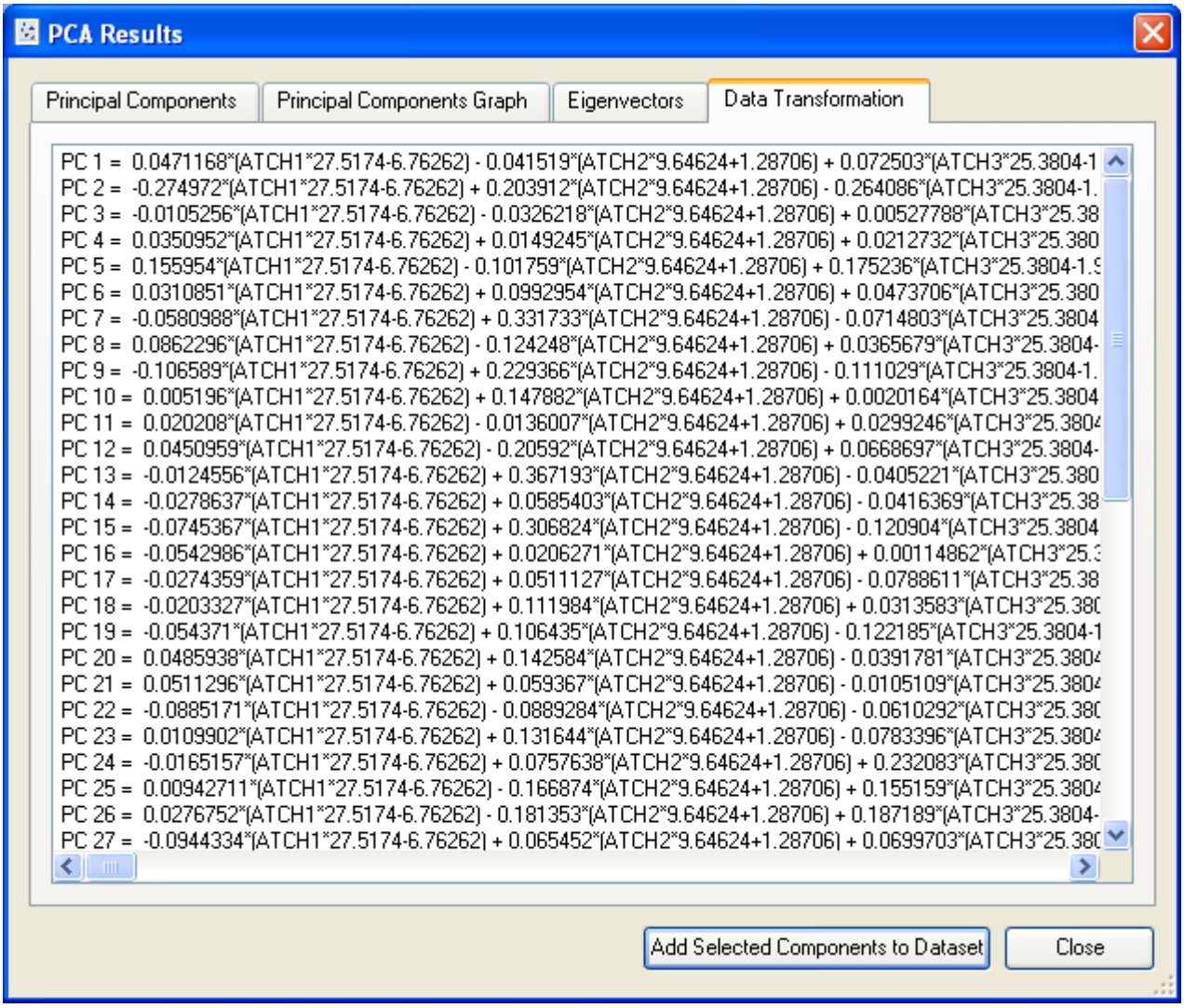


Figure 92: PCA Results: principal components shown as linear combinations of the numerical descriptors included in the analysis.

10.3 PCA Regression

In PCA regression (PCR), the principal components are used as descriptors when creating the regression model.

The main advantage of using PCR is that the number of model parameters (descriptors used in the regression model) is reduced, making over-fitting less likely. However, regression models derived using PCR may be more difficult to interpret since the resulting model is based on the principal components instead of numerical descriptors that are more easily interpreted.

To create a PCR model in MDM, first perform a PCA on the dataset as described in Section 10.2. Afterwards, invoke the Regression Wizard and select the principal components (PC 1, PC 2, etc.) to include in the regression model and

select the Multiple Linear Regression algorithm. When completing the wizard, the resulting regression model will be available in the Workspace Explorer.

To apply the derived PCR model on other datasets, the principal components used by the PCR model have to be computed for the new datasets. The components can be calculated using the Data Transformation dialog box as explained in Section 10.2 (see the description of the **Data Transformation** tab page).

10.4 Subset Creation from Principal Components

The subset creation methods introduced in Chapter 7 are used to create non-redundant subsets from a number of selected numerical descriptors. Instead of using the numerical descriptors available in the dataset, principal components (PCs) can be used to create subsets. To use PCs create the PC columns using the PCA dialog box and select the new PC descriptors (named PC 1, PC 2, etc.) in the subset creation dialog boxes.

11 Clustering and Similarity

Clustering (or cluster analysis) is about dividing data points into groups based on their similarity. The goal is to obtain a set of clusters, where the data points in a given cluster are as similar to one another as possible and where the clusters are as distinct from one another as possible. MDM provides three methods for clustering, namely the well-known K-means algorithm, a density-based clustering algorithm, and a threshold-based clustering algorithm.

The clustering (similarity) measures used by the clustering algorithms are based on numerical values. Therefore, only numerical descriptors can be included in the cluster analysis. Textual descriptors representing discrete values should be converted to numerical representations using the Convert Discrete Descriptor option (see Section 3.6 for details).

For a general introduction to cluster analysis, see Chapter 8 in **[TAN 2006]**.

11.1 K-Means Clustering

One of the oldest and most widely used clustering algorithms is the K-means algorithm **[TAN 2006]**. The K-means algorithm clusters a number of data records into K partitions based on the chosen numerical descriptors and the clustering measure used (introduced below).

The K-means algorithm works as follows:

1. Assign *K* initial centroids, where *K* is a user-defined parameter specifying the number of clusters that should be created (each centroid corresponds to a cluster, where the centroid is the average of all the data points assigned to the cluster). The K centroids are either randomly picked records or randomly generated data points since no records have been assigned to the clusters yet.

2. Assign each record to the closest centroid, using a specific cluster measure (introduced below). All the records assigned to a given centroid belongs to the same cluster.
3. Update all centroids (the centroid position is set to the mean of all records assigned to the centroid).
4. Repeat steps 2 and 3 until the centroids do not change (i.e. no records change cluster) or until a maximum number of iterations has occurred.

Notice that empty clusters may occur if e.g. the randomly created centroid is too far away from the data points compared with other centroids. In this case, the empty cluster centroid will be replaced with a new randomly created centroid and the centroids will be updated. However, there is no guarantee that empty clusters will not occur when using centroids created from randomly generated data points. To avoid empty clusters, the initial centroids should be created from existing records.

Using K-Means Clustering

To invoke the K-Means clustering wizard, choose **Modelling** | **K-means Clustering...** from the menu bar.

The first page in the wizard gives you the possibility to choose which dataset to perform the clustering analysis on and which descriptors should be used (i.e. included in the cluster measure) when performing the clustering.

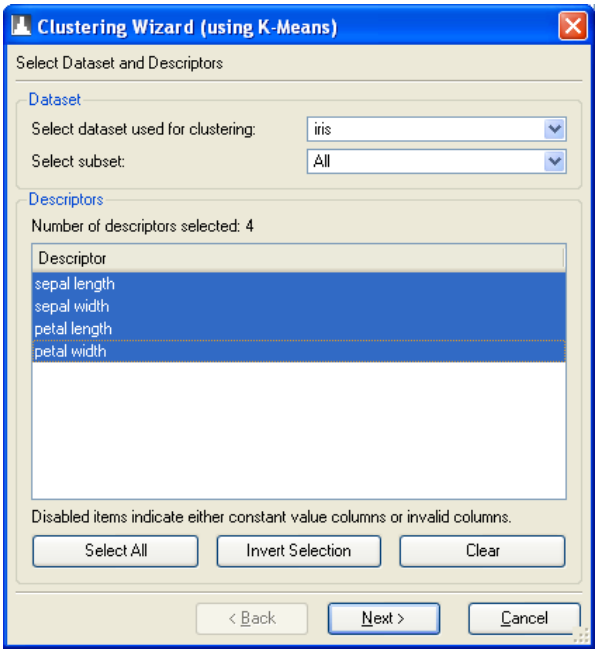


Figure 93: Clustering Wizard: Selecting dataset and numerical descriptors for the cluster analysis.

On the next page in the wizard it is possible to customize the clustering algorithm.

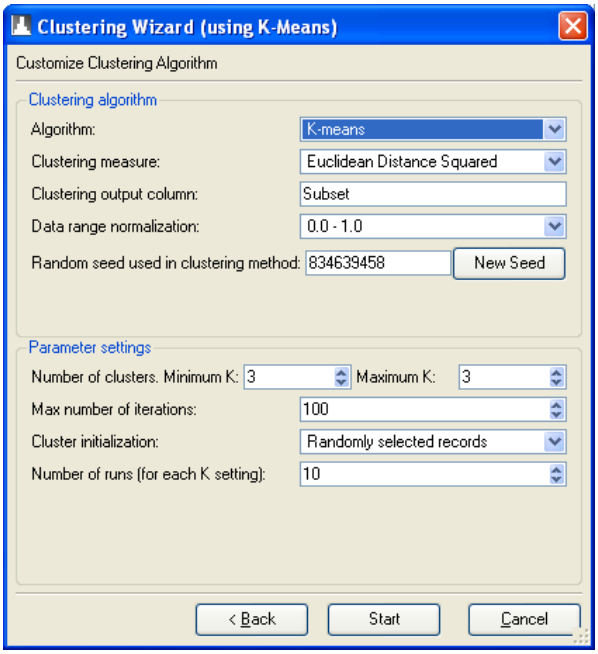


Figure 94: Clustering Wizard: Setting parameters for K-means clustering method.

By default the **algorithm** option is set to **K-means**, but it is possible to choose a density-based clustering algorithm (introduced in the next section).

The **Clustering measure** option shows the cluster measure currently used (Euclidean Distance Squared). The clustering measure is used during the cluster analysis to compute the distance (similarity) between two given data records. In K-means clustering, the distance between records is used when assigning records to the nearest cluster. Section 11.9 provides a short description of the clustering measures available in MDM.

When the cluster analysis is done, each record will be assigned a *cluster identifier* indicating which cluster the record has been assigned to. The **Clustering output column** is used to specify the name of the spreadsheet column where the cluster identifiers will be stored. A new column will be created if the column does not exist. Existing values in the **Clustering output column** will be replaced when doing a new clustering run.

The scale of the numerical descriptors used heavily influences on the distance between data points – therefore unless the descriptors are known to have a meaningful relative scaling, it is suggested to normalize the descriptors. The following data ranges are available from the **Data range normalization** combo box: 0 to 1, 0.1 to 0.9, -1 to 1, and None. The normalization is only used during the clustering analysis and will not make any modifications to the dataset.

It is also possible to manually alter the **random seed** that is used when creating random numbers. The K-means algorithm uses random numbers when selecting the records for initial centroids and when creating new random centroids.

In terms of clustering quality, K-means is not guaranteed to find the best solution. The quality of the final solution depends largely on the initial set of centroids. To locate good clustering solutions, K-means may be executed several times using the **Number of runs (for each K value)** option (default is 10 runs). In the **Parameter settings box** it is also possible to set the **number of clusters** (*K*) that should be used in the algorithm. It is also possible to iterate through different *K*-values to identify the most promising setting. The K-means algorithm will terminate when the cluster centroids no longer change or when a maximum number of iterations has occurred. The maximum number of iterations can be set using the **Max number of iterations** option. It is also possible to select which **Cluster initialization** method to use: **Randomly selected records** will randomly select *K*-data records and use them as initial centroids whereas **Random centroids** will create *K*-random centroids (within the data point ranges defined by the data records available in the dataset).

When the clustering analysis is done (after pressing the **Start** button), a **Clustering Results** dialog box will be shown (see Figure 95).

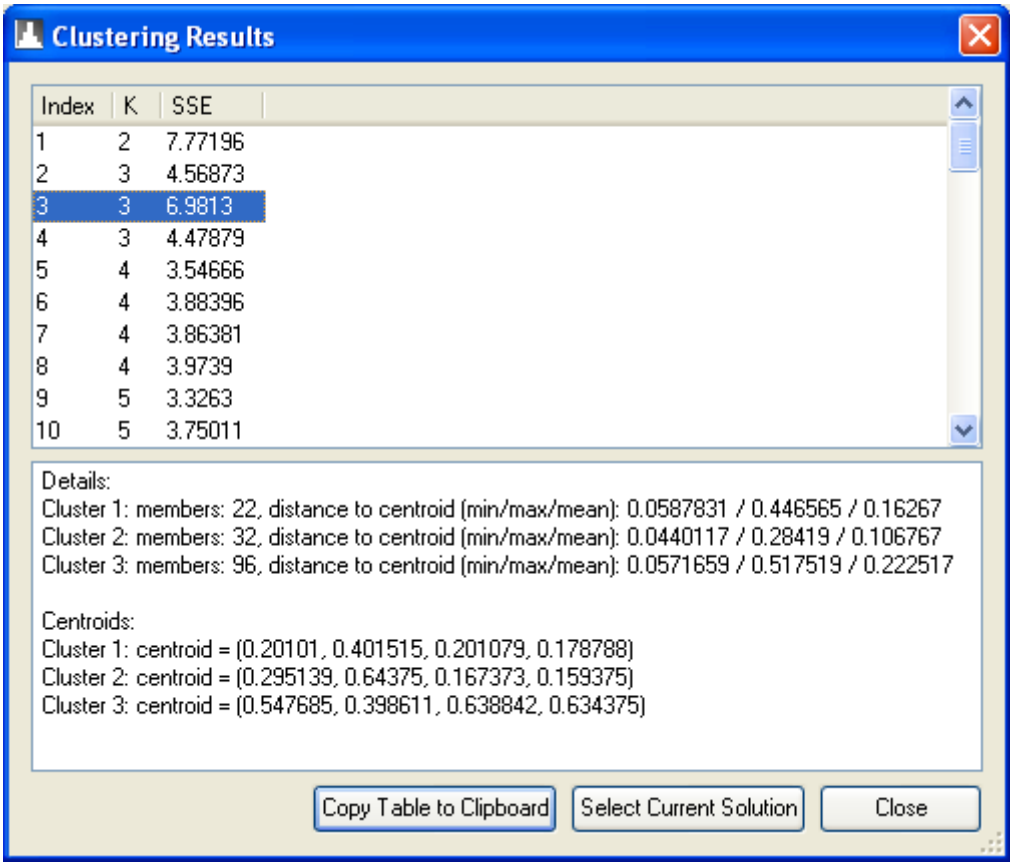


Figure 95: Clustering Results dialog box: Select which clustering solution should be used.

The **Clustering Results** dialog box shows the clustering results obtained using the selected parameter settings for the K-means algorithm. The **K-**column shows the number of clusters found for each solution. The last column **SSE**, shows the sum of the squared error (SSE) defined as:

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist(C_i, x)^2$$

where *K* is the number of clusters, *x* is a data point, C_i is the *i*'th cluster and $dist(C_i, x)$ is the Euclidean distance between centroid for cluster C_i and the data point *x*.

Thus SSE is the sum of squared error, where the error of each data point is the Euclidean distance between the data point and the nearest cluster centroid. SSE can therefore be used to evaluate the quality of different clustering runs, since lower SSE values indicate that the centroids are better representatives than solutions with higher SSE values. Notice: SSE can only be used to evaluate clustering runs using the same clustering measure, the same numerical descriptors, and the same K-value (number of clusters). Consequently, the SSE values cannot be used to e.g. identify which descriptors should be included in a cluster analysis.

From the Clustering Results dialog box, the user can select which solution to apply by selecting a given row in the list view. When pressing the **Select Current Solution** button, a **Clustering output column** containing the cluster identifiers (integer numbers) given by the chosen solution will be added to the current dataset/spreadsheet.

The table below the list view contains details about the current solution, such as the number of members for each cluster and the min/max/mean distances to the cluster centroids. In addition, the centroids are listed with both normalized and renormalized values (if normalization was used during clustering). The table values can be copied to the clipboard using the **Copy Table to Clipboard** button.

11.2 Density-Based Clustering

While K-means clustering is simple and efficient, the shapes of the clusters are defined by the neighborhood of the centroids. Sometimes this results in an unnatural clustering (see Figure 96 for an example). K-means clustering also requires that the number of clusters is known in advance.

Density-based clustering is an alternative clustering scheme. It clusters data by taking both the distance and the local density into account. In areas with low density points must be closer to each other to belong to the same group than in areas with higher density.

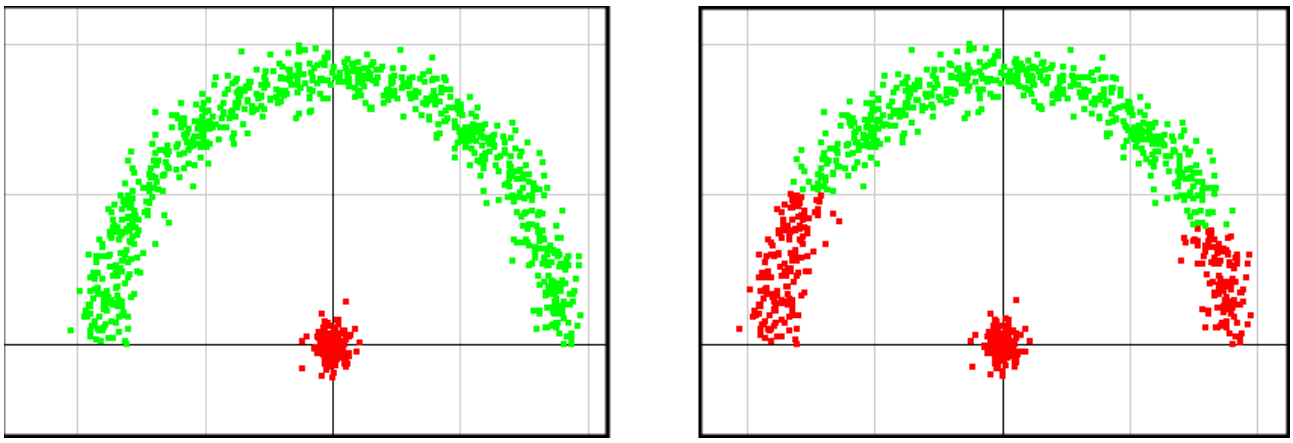


Figure 96: Density-based clustering versus K-means clustering (with K=2).

Using Density-Based Clustering

To invoke the Density-Based clustering wizard, choose **Modelling** | **Density-based Clustering...** from the menu bar.

Similar to K-means clustering, the first page in the wizard gives you the possibility to choose which descriptors to use when performing the clustering.

On the following page the density-based clustering parameters can be customized (see Figure 49). Normally the default clustering measure and data range normalization settings are adequate for density-based based clustering, but the optimal clustering threshold depends on the dataset.

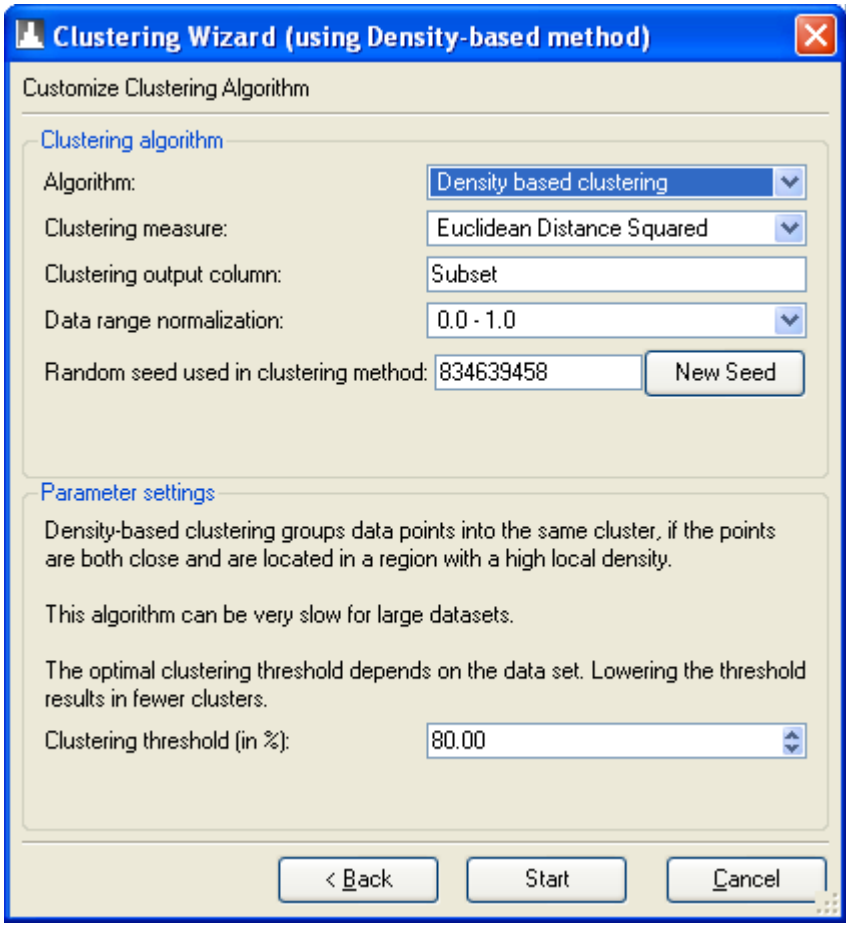


Figure 97: Customizing Density-based clustering.

The clustering threshold is specified in the **Clustering threshold (in %)** text field. Raising the value results in more clusters. It is often necessary to try several settings before finding an acceptable setting. A detailed description of the clustering threshold is provided in the next section.

The random seed is only used when shuffling the dataset since the density-based clustering scheme is not stochastic.

Density-Based Clustering in Details

The clustering scheme has some similarities with the density-based outlier method.

In particular the first step is to calculate the local density, $density(p_i)$, for each data point, which is done the same way as for the density-based outlier detection method (see Section 6.2).

Next, each data point is assigned to its own cluster and the *binding strength* between any two data points is calculated. The binding strength is defined as:

$$binding\ strength(p_i, p_j) = \frac{density(p_i)}{distance(p_i, p_j)^2}$$

Notice that the distance depends on the chosen measure, and that the binding strength is not symmetric ($binding\ strength(p_i, p_j) \neq binding\ strength(p_j, p_i)$).

Two data points are assigned to the same cluster if their binding strength is higher than a user-specified threshold. For distances equal to zero the binding strength is infinite, and the data points will always belong to the same cluster.

The binding strength threshold is the crucial step in the algorithm. If the threshold is too high, too many clusters will be generated, and if it is too low too few clusters are generated.

The threshold is specified on the **Parameters Settings** group (see Figure 49).

The threshold parameter is normalized using the data point with the highest density (p_{max}) as a reference. A value of 100% corresponds to the threshold required for this data point to make a cluster with only its nearest neighbor (thus resulting in a cluster for each data point, except for a single cluster with two data points). A value of 0% would correspond to a situation where the p_{max} data point would group together all data points in a single cluster.

Also notice that the density-based clustering algorithm can be very slow for large datasets.

11.3 Threshold-based Clustering

Threshold-based clustering is a simple clustering scheme that utilizes a user-defined threshold value. Initially, a cluster is formed containing the first data point in the dataset. Afterwards, each data point is added to the most similar cluster if the data point is closer to the cluster's representative than the specified threshold – otherwise the data point forms a new cluster.

Using Threshold-Based Clustering

To invoke the Threshold-Based clustering wizard, choose **Modelling** | **Threshold-based Clustering...** from the menu bar.

Similar to K-means and density-based clustering, the first page in the wizard gives you the possibility to choose which descriptors to use when performing the clustering.

On the following page the threshold-based clustering parameters can be customized (see Figure 98).

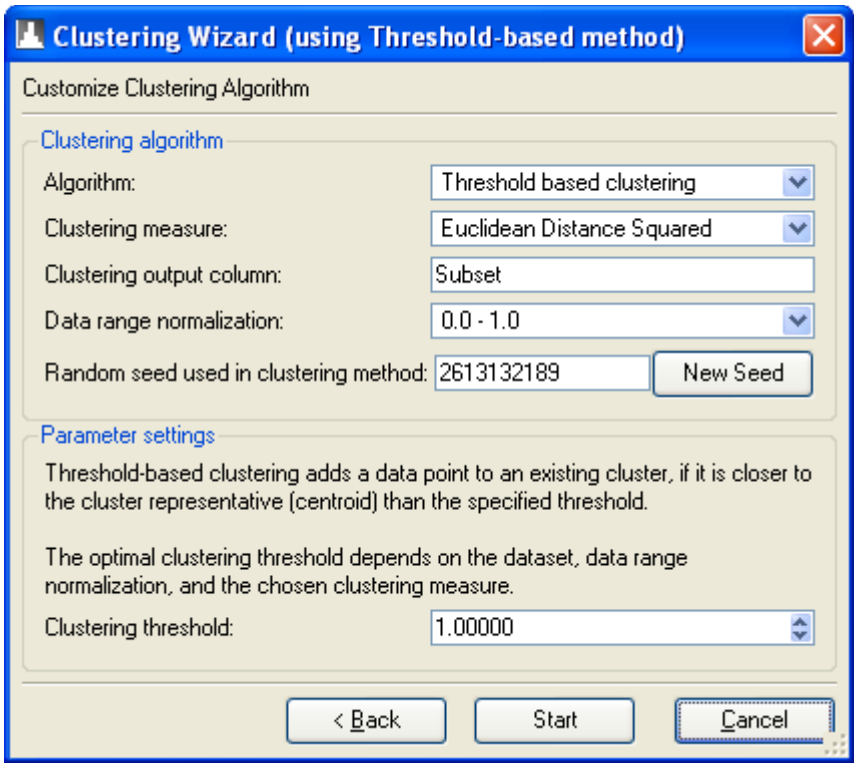


Figure 98: Customizing Threshold-based clustering.

Normally the default clustering measure and data range normalization settings are adequate for threshold-based clustering, but the optimal clustering threshold depends on the dataset, data range normalization, and the chosen clustering measure.

The clustering threshold is specified in the **Clustering threshold** input field. It might be necessary to try several settings before finding an acceptable setting.

The random seed is only used when shuffling the dataset since the threshold-based clustering scheme is not stochastic.

11.4 Visualization of Clusters

The clustering results obtained running either K-means, density-based clustering, or the threshold-based clustering method can be easily visualized in the 2D or 3D plotters. Using the **Color By Descriptor** dialog box, it is possible to color the data points according to the cluster that they are assigned to (using the cluster identifier listed in the Subset column). See Section 4.4 for more details on using the **Color By Descriptor** dialog box.

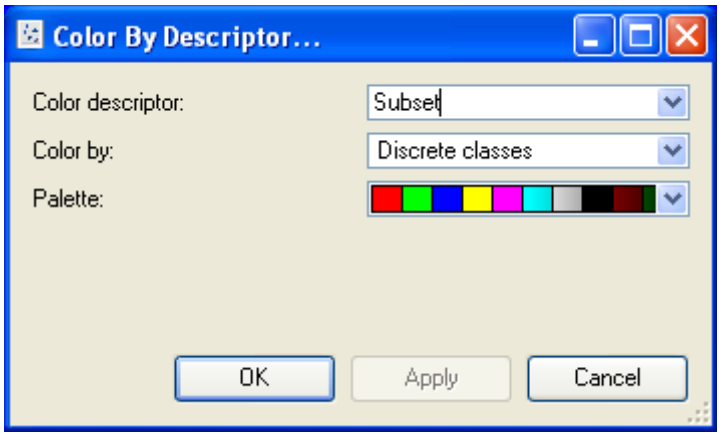


Figure 99: Selecting color used for visualizing found clusters. The 'Subset' column contains the cluster assignment for each record.

Figure 100 shows an example of a 2D visualization of clusters in the Iris dataset. The three clusters shown are based on the original cluster assignment specified in the *class* descriptor (see next section for more details about the Iris dataset).

11.5 Example: Iris Dataset

One of the most cited datasets to be found in the pattern recognition literature is the Iris plant dataset **[FISCHER 1936]**, which is often used to illustrate and evaluate cluster analysis methods. The dataset contains 3 classes of 50 instances each, where each class refers to a type of the Iris plant (Setosa, Versicolor, Virginica). One class (Setosa) is linearly separable from the other two, whereas the latter two are not linearly separable from each other. The four numerical descriptors characterizing the Iris plant are: sepal length, sepal width, petal length, and petal width (all lengths and widths are given in centimeters).

The Iris dataset is publicly available from the UCI Machine Learning Repository **[ASUNCION 2007]**. The dataset (`iris.csv`) is also included in the examples directory, which is located in the MDM installation folder. Using the K-means clustering method introduced in Section 11.1 it is possible to correctly cluster 133 out of the 150 records using default settings in the Clustering Wizard (K=3, Euclidean Distance Squared measure as clustering measure, and using all four numerical descriptors available). The sum of squared error (SSE) using these settings is: 6.99811.

Figure 100: Scatter plot of petal width versus petal length (Iris dataset). The data points are colored according to class (red: Setosa, green: Versicolor, blue: Virginica).

11.6 Similarity Browser

A common data mining task is to identify data points which are similar (in some sense) to a given element.

Although clustering makes it possible to group similar data points, it is sometimes useful to focus on one or more specific data points and find other data points that are either similar or different.

The **Similarity Browser** makes it possible to:

- Find data points (where the data points corresponds to the rows in the dataset) that are similar to the current selected data points.
- Rank an entire dataset according to its similarity to one or more data points, and create a new column with the rankings.
- Choose different measures of 'similarity' (e.g., Euclidean distance or Tanimoto coefficient) on all or on a subset of the descriptors.
- Find similar data points in another dataset than the currently chosen.
- Find data points that are different from the currently selected data points.

11.7 How To Use The Similarity Browser

The **Similarity Browser** is invoked by choosing **Modelling | Similarity Browser...** or by using the keyboard shortcut **CTRL+B**.

This will open a new window, with the **Similar Rows** tab chosen.

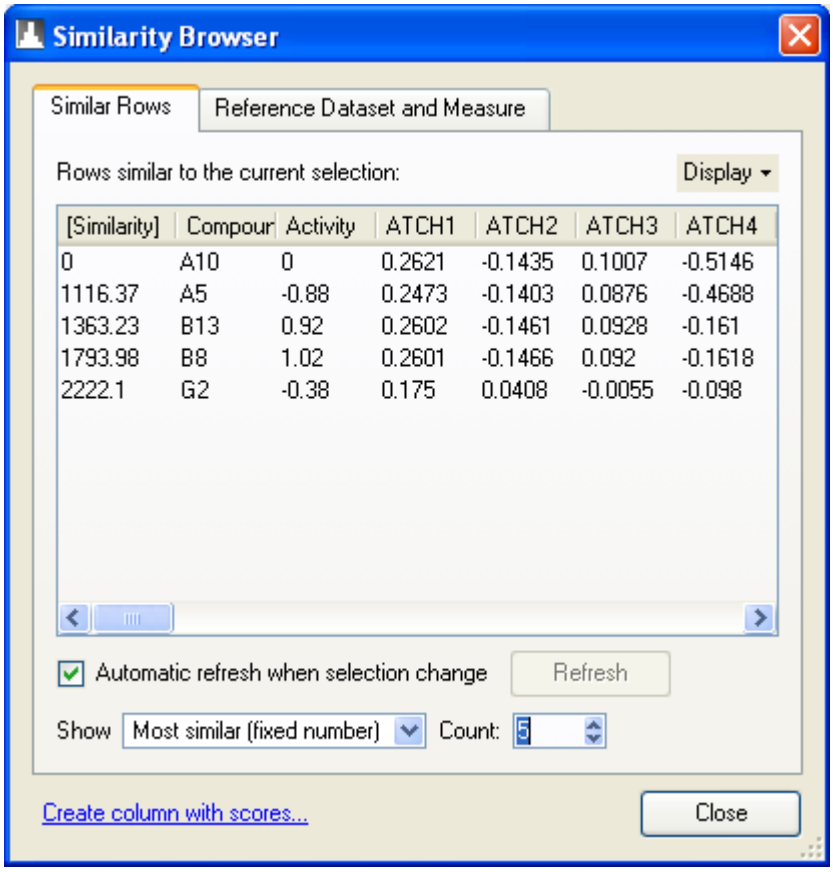


Figure 101: Similarity Browser dialog.

When a new row is selected in the current dataset, the **Similar Rows** list view will be updated with the data points that are most similar to the current

selected rows. Notice that one or more rows can be selected in the current dataset - for multiple rows the similarity calculation depends on the settings specified on the **Reference Dataset and Measure** tab page.

The list with similar rows is automatically updated whenever the selection changes. If this is undesirable, (for instance when working with very large datasets), turn off the **Automatic refresh when selection change**. The list may then be manually updated by pressing the **Refresh** button.

Notice that a row is considered to be selected if it contains one or more selected cells (it is not necessary to select all cells in a row - and selecting multiple non-connected cells in the same row also just counts as one selection).

Initially when the **Similarity Browser** is opened, it shows the five most similar rows for the current dataset. It is possible to change this behavior using the **Show [Most Similar (...)]** drop-down combo box. It is possible to show either all rows, a fixed number of rows, or a percentage of all rows in the dataset.

Per default, the **Similarity Browser** will list similar rows from the current dataset (using a Euclidean distance measure). These settings can be changed from the **Reference Dataset and Measure** tab.

It is also possible to calculate a similarity score for every single entry in a dataset, and add it as a new column. This is done by pressing the **Create columns with scores...** button. After choosing a name for the new column, the column will be added to the reference dataset.

When the **Similarity Browser** updates the list view, all columns from the reference dataset are shown. Often this is more information than is needed. By using the **Display** drop-down button in the top-right corner, it is possible to choose to view only a subset of the columns. The following choices are available:

- **All rows** – all rows are displayed
- **Textual and measure rows** – all textual rows and all rows that the similarity measure currently use are shown.
- **Measure rows** – Only the rows that the similarity measure currently use are shown.
- **Custom** – Shows a list of descriptors making it possible to choose manually. This menu can also be invoked by using the context menu (click with right mouse button) on the list view.

11.8 Customizing the Similarity Browser

The **Reference Dataset and Measure** tab makes it possible to customize the Similarity Browser.

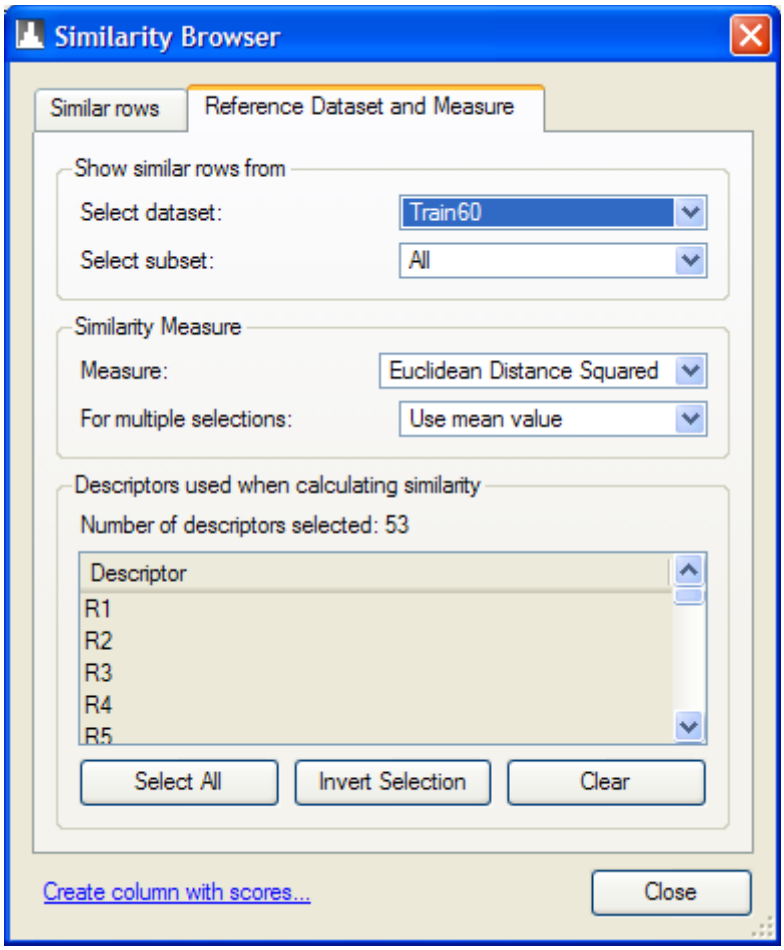


Figure 102: Customizing Similarity Browser settings.

Per default the rows listed on the **Similar Rows** tab are calculated from the *current* dataset – that is, the dataset currently being displayed and holding the current selection in the main application window.

It is possible to show rows from another dataset by selecting it from the **Select dataset** drop-down box. It is also possible to only work with a subset of the chosen dataset by using the **Select subset** drop-down box.

The similarity measure used is chosen from the **Measure** drop-down box. The measures are described in detail in Section 11.9.

Multiple Selections

If only a single row is selected on the currently displayed dataset, the similarity measure between the chosen descriptors from this row and all rows in the selected dataset is calculated. If multiple rows are selected, the behavior depends on the settings for the **For multiple selections** drop-down combobox. The following choices are possible:

- **Use mean value**. For each row in the chosen reference dataset, the similarity is calculated to each of the rows in the multiple selection. The

mean of these similarity values is calculated for each row in the reference dataset. (This means that if five rows are selected in the current dataset, we will calculate five similarity values for each row in the *reference* set. The mean of these five values is the value displayed in the **[Similarity]** column on the **Similar Rows** tab.)

- **Use minimum value**. Same as above, except the lowest of the similarities calculated for the multiple selected rows is used.
- **Use maximum value.** Same as above, except the highest of the similarities calculated for the multiple selected rows is used.

Choosing Descriptors

The last group (**Descriptors used when calculating similarity**) shows which descriptors are taken into account when calculating the similarity values. For a descriptor to appear on this list, it must exist in both the current dataset and the reference dataset specified at the top of the dialog. It is valid to set the reference to be the same as the current dataset, in which case all descriptors in the dataset are shown (except textual, invalid, and reserved descriptors).

Notice that columns are matched between two dataset based on their names – the actual order of the columns does not matter.

11.9 Clustering/Similarity Measures

The following clustering/similarity measures are available in MDM:

Euclidean Distance: The Euclidean Distance between two data points *x* and *y* in *n*-dimensional space (where *n* is the number of numerical descriptors chosen in the wizard):

$$d(x,y)=\sqrt{\sum_{k=1}^{n}(x_k-y_k)^2}$$

Euclidean Distance Squared: This measure is the same as the Euclidean Distance measure above except that the square root is omitted.

$$d(x,y)=\sum_{k=1}^{n}(x_k-y_k)^2$$

Manhattan Distance: The Manhattan Distance summarizes the absolute differences between two records with *n* numerical descriptors. If all the

numerical descriptors are binary, the Manhattan Distance equals the number of bits that are different between the two records.

$$manhattan(x, y) = \sum_{k=1}^{n} |x_k - y_k|$$

Cosine Similarity: The Cosine Similarity measure returns the cosine of the angle between the data points *x* and *y*, i.e., if *cos(x,y) = 1*, the angle between *x* and *y* is 0 degrees and *x* and *y* are proportional. If *cos(x,y) = 0* the angle between *x* and *y* is 90 degrees, which means that *x* and *y* are orthogonal.

$$\cos(x, y) = \frac{x \cdot y}{||x|| ||y||}$$

Tanimoto Coefficient: The Tanimoto Coefficient (also referred to as the Extended Jaccard Coefficient) is defined as:

$$tanimoto(x, y) = \frac{x \cdot y}{||x||^2 + ||y||^2 - x \cdot y}$$

If all the numerical descriptors are binary, the Tanimoto Coefficient is the proportion of the 1-bits which are shared between data record *x* and *y*.

12 The Chemistry Module

The chemistry module extends MDM with several features for working with molecular structures. These features include import of chemical data in the form of either SDF files or SMILES strings, the creation of 2D depictions of molecules, and depictions of molecules in the spreadsheet, in one or more grid views, or in the 2D plotter. By being able to inspect chemical structures visually in MDM, it becomes much more easy to interpret and understand chemical data.

The chemistry module in MDM is designed to work with small (typically drug-like) organic molecules - it is not designed to work with large macro-molecules such as proteins.

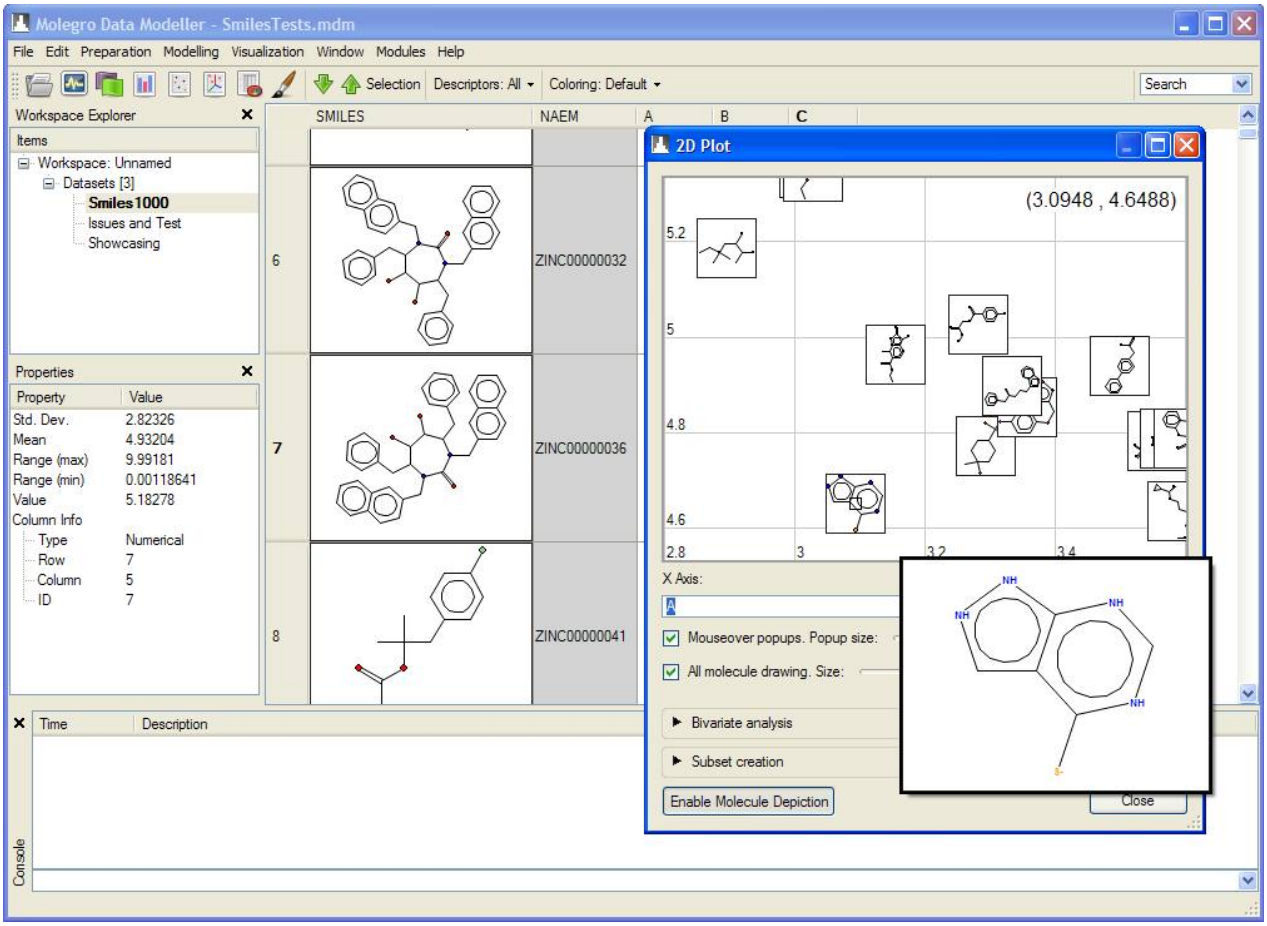


Figure 103: Molecule Depiction in the MDM GUI.

12.1 Importing Chemical Structures

MDM supports two ways of importing chemical structures. Either by importing structures from an SDF file or by importing SMILES descriptions. SMILES descriptions are ordinary text strings and can be imported the same way as other text files are imported in MDM (see Section 3.2).

SDF Files

MDM supports the parsing of Symyx SDF files (formerly MDL SDF files) which are typically used for storing larger libraries of small molecules. SDF files contain atom and bond (connectivity) information, together with optional 'data' fields for each compound. These data fields may contain arbitrary information, and they are imported as either textual or numerical columns in MDM.

SDF files can be imported by choosing **Modules** | **Chemistry** | **Import from SDF...,** or by choosing **File** | **Import Dataset...** (and selecting the '*.sdf' file type), or by dragging and dropping an SDF file onto the spreadsheet window.

An SDF file may contain either three dimensional coordinates for each atom position, two dimensional coordinates (by setting the Z-coordinate to zero), or no coordinate information at all (in which case the coordinates in the file are all zero).

MDM does not store the SDF file after it has been imported into a spreadsheet. Instead the molecular structure is represented and stored as a SMILES string. This conversion is done the following way:

- If the SDF file does not contain coordinates, MDM will convert the structure to a SMILES string and use its internal layout engine to generate a 2D depiction of the molecule.
- If the SDF file contains 2D coordinates, the importer will ask the user whether to use these coordinates or use its internal layout engine to generate new coordinates. The default choice is use the 2D coordinates specified in the SDF file. Again MDM will convert the molecular structure to a SMILES string. However, SMILES strings cannot contain atomic coordinates. Therefore MDM uses a slightly modified notation where the coordinates are appended to the generated SMILES string if the user chooses to preserve the 2D coordinates.
- If the SDF file contains 3D coordinates, it is possible to use the X and Y parts of the 3D coordinates to form a 2D depiction, or to use the layout engine in MDM to create a new 2D depiction. The default method is to generate new 2D coordinates (this usually produces better depictions than projecting the 3D structure onto the X-Y plane).

In order to convert from the atom and connectivity data in an SDF file, MDM uses its internal SMILES generator. Notice that a given chemical compound may have several equally valid SMILES representations. Several schemes have been proposed for generating unique (sometimes called canonical) SMILES strings for a given molecule, but currently MDM does not use any of them. However, it does try to create a somewhat compact and human-readable SMILES string - for instance it will identify and base the SMILES string generation on the longest covalent chain in a given molecule.

The following restrictions apply when importing SDF files:

- The SMILES generator in MDM will not take stereochemistry into account, even if it is explicitly given by an SDF file with 3D coordinates. Neither will the layout generator that generates 2D depictions recognize stereochemistry (for instance it will not be possible to recognize cis/trans conformations from the 2D depiction, though this is likely to change in future versions of MDM).
- The files must be in V2000 connection table format.
- Disconnected structures (where a 'single' molecule has atoms not covalently connected to some of the other atoms) are not supported. If a

single 'Molfile' entry in an SDF file contains multiple, disconnected structures, only the first of the structures is imported.

- The 'SText' and 'Properties' SDF fields are ignored. For storing additional data in an SDF file use the optional 'Data' fields.

After an SDF file has been imported, a new dataset will be present with the following columns:

SDFHeaderName - The name of the compound, as specified in the first line of the SDF header for each compound.

Generated SMILES - The smiles description generated from the structural information in the SDF file. The column is automatically set as a SMILES column, so it will appear as a graphical column with a 2D depiction of the molecule. Notice that it is possible to see and edit the generated SMILES string by double-clicking a cell.

Also any information in the optional 'Data' entry format will appear as either textual or numerical columns in the spreadsheet (multi-line data fields are concatenated into a single line).

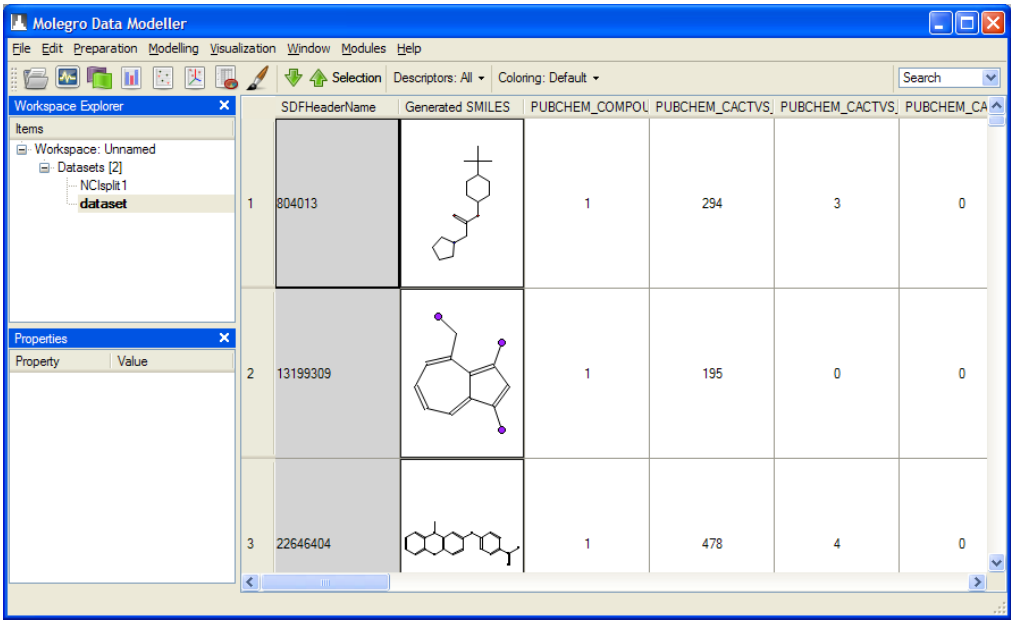


Figure 104: Imported SDF file. The 'Generated SMILES' column is generated from the molecule data in the SDF file.

SMILES Strings

In order to create depictions from SMILES strings, simply import or create a text column with the SMILES strings. Any textual column in MDM can be interpreted as containing SMILES descriptions. In order to specify that a given column contains SMILES descriptions, choose **Modules** | **Chemistry** | **Setup**

SMILES Column.... The column will change to the 2D molecule depiction style. Notice that it is possible to continue working with a SMILES column as any other text column – the text may be copied or edited by double clicking a cell in the column.

12.2 Working with Molecular Depictions.

Whenever a SMILES column has been specified (either manually by choosing **Modules** | **Chemistry** | **Setup SMILES Column...** or automatically when importing SDF files) the column appears as a graphical column with 2D depictions of the molecules. It is possible to show the SMILES string instead of the graphical 2D depiction by toggling **Modules** | **Chemistry** | **Draw Molecules in Spreadsheet**. Working with SMILES strings in text mode makes it possible to see a larger portion of the spreadsheet, and molecules can still be inspected by opening one or more molecule depiction windows (introduced below).

It is possible to change the column size by dragging the cell separators in the row border. If the SMILES parser is unable to parse a SMILES string, the cell will appear with a red background and a short error message.

The context menu for a SMILES column offers a few items not found for ordinary spreadsheet columns (these items are also accessible from the **Modules** | **Chemistry** menu):

- **Embed Coordinates in SMILES column**. A SMILES string does not specify atom coordinates. After MDM has parsed a SMILES string, it uses its internal layout engine to assign 2D coordinates to the structure. These coordinates are calculated whenever MDM needs to draw a molecule, and are cached until MDM is closed. The generated layout is normally not saved together with the MDM dataset, so it needs to be regenerated when the files is loaded. By embedding the coordinates to the SMILES column, the 2D coordinate may be stored by appending them as a list after the SMILES string. This makes it faster to depict the coordinates when the file is subsequently loaded. This also makes it possible to preserve a 2D layout imported from an SDF file. A SMILES string with embedded coordinates may look like this: CCC {0,0;0.86,0.5;1.73,0}.
- **Remove Coordinates from SMILES column**. While it may be faster to embed the coordinates in SMILES column, this is not a standard extension, and may cause problems with other software programs when exporting the data. Use this option to remove any coordinates appended to the SMILES column before exporting the data.
- **Open New Molecule Depiction window**. This creates a new molecule depiction window (see the next section).

The Molecule Depiction Window.

It is possible to create one or more Molecule Depiction windows for having multiple views of the molecules. A Molecule Depiction window offers more flexibility than the default visualization inside a spreadsheet column does.

A new Molecule Depiction Window may be opened by choosing **Modules** | **Chemistry** | **Open New Molecule Depiction window** or by choosing **Open New Molecule Depiction Window** from the context menu for a cell item in a SMILES column.

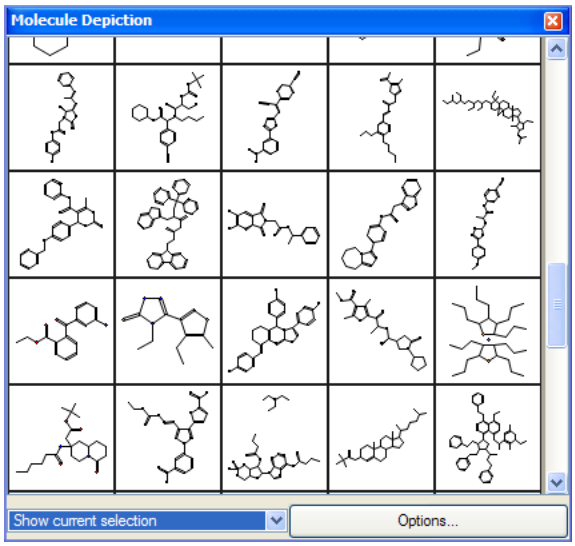


Figure 105: The New Molecule Depiction Window.

The combo box in the lower left corner toggles whether the window should show the current selection in the spreadsheet (**Show current selection**) or whether the molecules currently viewed should be held fixed (**Freeze current display**). It can be useful to freeze the view when comparing molecules (remember that it is possible to open multiple Molecule Depiction Windows).

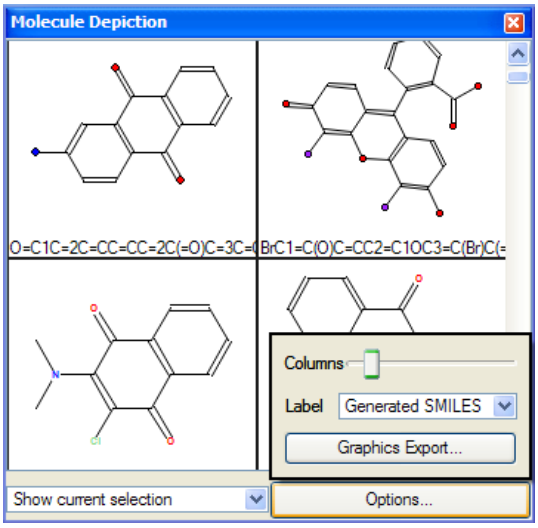


Figure 106: The Options popup menu for the Molecule Depiction Window.

The **Options** menu makes it possible to customize the appearance of the molecule window. The **Columns** slider may be used to create a grid view of the selected molecules. The slider determines the number of columns of the grid. It is also possible to label the molecules by the information from any textual or numerical column in the spreadsheet by choosing a column in the **Label** combo box.

Finally, the **Graphics Export...** button can be used to export molecule depictions. It is possible to output either in vector graphics format (SVG) or in bitmap format (either PNG, JPG, or BMP). The bitmap images will be identical to the ones displayed in the molecule depiction window, and the size of the images will be same as displayed on screen. In contrast, images saved in SVG do not suffer quality loss when scaled – notice vector graphics depictions may look slightly different from the bitmap depictions though.

It is possible to store the images in three different ways:

- **Single image file with all molecules**. Generates one large image file with the molecules in the grid layout from the Molecule Depiction window.
- **One image file for each molecule (filename by index)**. MDM will prompt for an output directory, and the files will be stored as e.g. 0.PNG, 1.PNG, 2.PNG, 3.PNG.
- **One image file for each molecule (filename by label)**. Same as above except that the files will be labelled with the name specified by their label (notice that this requires that the label names are unique. Also the filename is stripped for characters which are not standard letters, numbers or spaces, and truncated to a maximum file name length of 64 characters).

Depiction in the 2D Plotter.

It is also possible to view molecules in the 2D plotter. In order to do this, first make sure that a SMILES column is specified in the spreadsheet and then select **Visualization** | **2D Plot**...

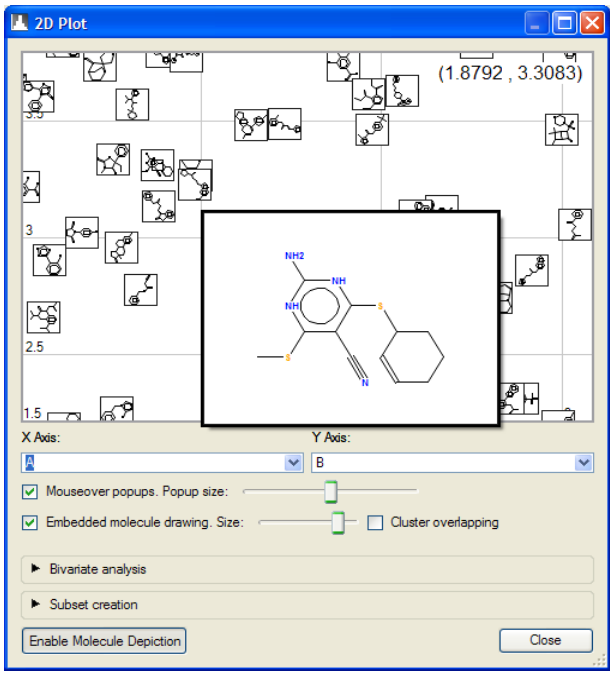


Figure 107: Molecule depictions in the 2D plotter. Here both popup and embedded molecule depictions are shown.

Whenever a SMILES column is present in the spreadsheet the default behavior for the 2D plotter is to enable molecule depictions. This can be toggled by clicking the **Enable Molecule Depiction** button in the lower left corner.

The 2D plotter offers two ways to visualize molecules: popup *visualization* which appears whenever the mouse hovers over a data point, and *embedded visualization* where the molecules are drawn directly on the graph canvas instead of the data points. Both may be used simultaneously.

Popup visualization may be toggled using the **Mouseover popups check box.** The size of the popup window may be adjusted using the **Popup size** slider.

Embedded visualization may be toggled using the **Embedded molecule drawing check box.** The **Size** slider adjust the size of the molecules drawn in the graph window.

Notice that when embedded molecules are drawn, the molecules may overlap. This is in particular likely to occur whenever one of the axis contains *discrete values* (such as hydrogen donor counts or number of rotatable bonds). It is possible to avoid this by enabling the **Cluster overlapping check box.** A cluster is marked by a red border frame.

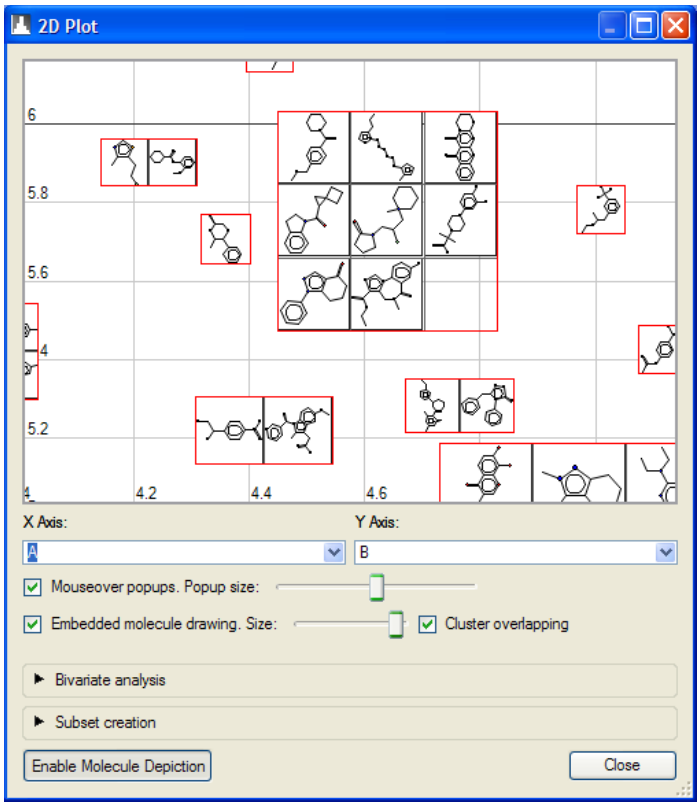


Figure 108: Example of clustered molecules.

A few notes about molecules in the graph plotter:

It is possible to select rows in the spreadsheet by clicking on the relevant molecule.

- A frame with a red background indicates that the SMILES parser encountered an error, and was unable to create a depiction.
- A frame with a yellow background indicates that Layout engine is working on creating a molecule depiction. This is a background task, and the molecules will appear automatically when they are ready.
- A frame with a grey background (and no molecule) indicates that the frame was too small to draw a useful depiction of the molecule. Grey frames may also occur when too many molecules are present at the same time on the graph canvas.

12.3 The Layout Engine and Internal Molecule Representation.

The layout engine is responsible for creating a 2D depiction for a given molecule.

The layout engine and the molecule representation do have a few caveats:

- It is not always capable of layouting complicated ring structures and large molecules correctly.
- Stereochemistry is not supported. The layout engine will not properly depict cis or trans configurations even if they are present in the SMILES or SDF encoding. 'Up' and 'Down' bond types ('/' and '\') are treated as single bonds and the '@' chiral property is silently ignored.
- Hydrogens atoms are always implicit. Even if hydrogen atoms are explicitly stated in the SMILES string or as individual atoms in the SDF file, they are converted to a property of the heavy atom they are attached to. Normally, this is not a problem, but for instance dihydrogen ('[H][H]' as a SMILES string) cannot be expressed in this implicit model.
- Whenever a molecule is loaded from an SDF file, it is automatically converted into a SMILES string. If the molecule contains explicit hydrogens, the hydrogen count will be deduced from these. If no hydrogens are present, a simple valence model will automatically assign implicit hydrogens for the 'organic' elements (B, C, N,O, P, S, F, Cl, Br, I).
- Notice that when displaying the atom element names, if the size of the letters are below a given threshold, the layout engine will paint the atoms as small colored discs instead of displaying the element abbreviations.

The Molecule Cache.

Whenever one or more molecule depictions are needed, MDM will send the requests to a background task. The background task will create the depiction and cache it for the rest of the MDM session. Since the molecule depictions are only calculated when needed there is no initial delay when enabling molecule depiction or when importing SDF files, but there may be a delay if it is necessary to create a lot of depictions simultaneously (for instance in the 2D plotter). Since the molecules are cached, this delay only occurs the first time the molecules are displayed.

By default the molecules stays in a molecule depiction cache for as long as MDM is running. Normally, the memory penalty for this is not very large, but it is possible to clear the memory cache in order to release the memory by choosing **Modules** | **Chemistry** | **Clear Molecule Depiction Cache**.

13 Customizing Molegro Data Modeller

13.1 General Preferences

Molegro Data Modeller can be customized using the **Preferences** dialog box, which can be invoked from the **Edit** menu.

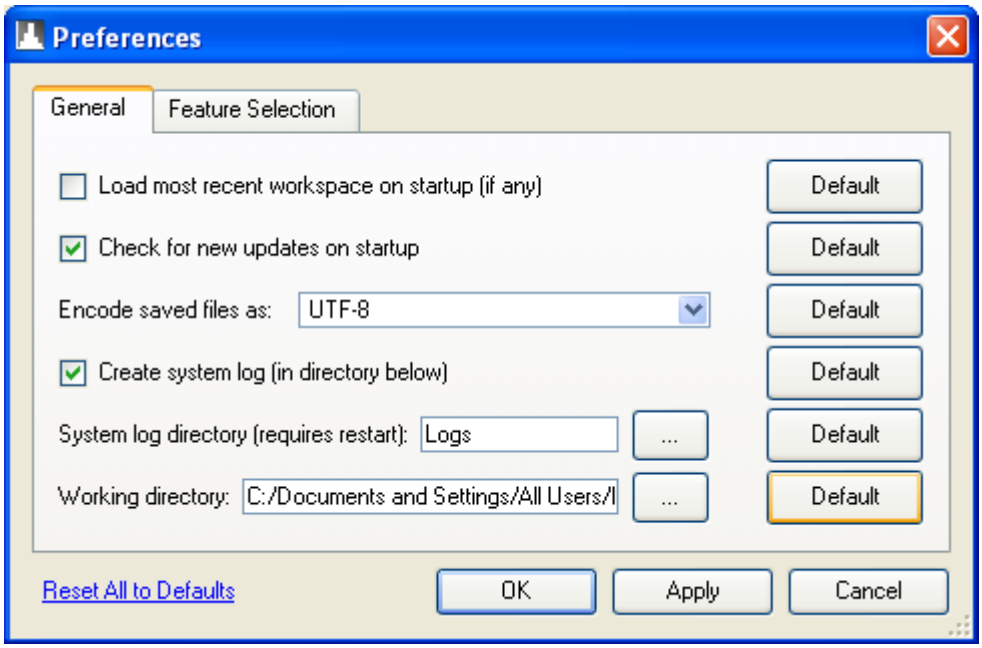


Figure 109: Preferences dialog box.

The following settings are available on the first tab page:

- The **Load most recent workspace on startup (if any)** option toggles automatic import of the last used workspace.
- The **Check for new updates on startup** option enables MDM to automatically check for new updates during startup.

- The **Encode saved files as** drop down menu allows you to choose in which encoding files (xml and csv) should be saved. It is recommend to choose one of the Unicode encodings, either **UTF-8** or **UTF-16**, in order to represent the largest number of possible characters. In most cases, if you data does not contain a lot of special characters (such as Japanese characters), UTF-8 will be the most compact format. It is also possible to store data as **Locale 8-bit**. In this encoding all characters are stored as a single byte, meaning only 256 characters can be represented. The actual characters included in this set depends on the current codepage settings on the machine. This option should only be used when exporting data to older software not capable of parsing Unicode text.
- The **Create system log (in directory below)** option is used to toggle whether a system log is created for each execution of MDM. The system log contains information about user actions conducted and is used to track potential bugs and performance problems. By default, the log files are stored in the `Logs` directory located in the same directory as the `mdm` executable file but another directory can be used if needed (e.g. if user has no write permissions to the directory used). *Notice:* If you encounter problems with MDM please email the log file created before the crash to: support@clcbio.com
- The **Working directory** setting is used to set the current **Working directory**, which is the root path for file related operations (e.g. loading and saving dataset and workspace files).

The Feature Selection tab page contains specific parameter settings for the Feature Selection algorithms introduced in Chapter 8.

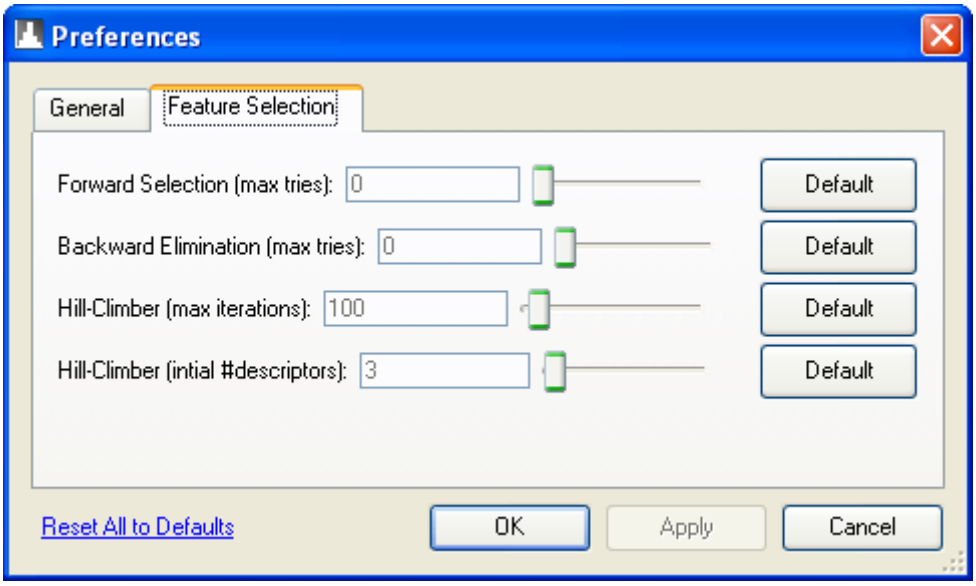


Figure 110: Preferences for feature selection algorithms.

The following options are available:

- The **Forward Selection (max tries)** setting determines the maximum number of descriptors that are probed at each step of the algorithm. Notice: A value of '0' is used to indicate that all descriptors available (i.e. not already taken) will be probed.
- The **Backward Elimination (max tries)** setting is identical to **Forward Selection (max tries)** except that this setting is used by the Backward Elimination algorithm.
- The **Hill-Climber (max iterations)** value defines the maximum number of iterations the Hill-Climber algorithm will be running before returning the solutions found.
- For the Hill-Climber algorithm, the number of descriptors used in the initial candidate solution is set to the **Hill-Climber (#initial descriptors)** value. If the value is higher than the total number of descriptors available, the actual value used will be equal to the number of descriptors available.

13.2 Command Line Parameters

The following command line parameters are available:

```
<filename>
-currentPath
```

The `<filename>` parameter can be used to import a dataset (in CSV format) or a workspace (MDM format) during MDM startup.

Example: `/Molegro/MDM/bin/mdm ../examples/selwood.csv`

The `-currentPath` parameter can be used to override the working directory specified in the general preference settings with the current path. This is particularly useful when running MDM from different working directories (using a terminal window) or when using a script to start up MDM.

Example: `/Molegro/MDM/bin/mdm -currentPath`

14 Help

14.1 PDF Help

The documentation for Molegro Data Modeller is available as a PDF file. In order to invoke the PDF help using the built-in PDF reader, choose **Help** | **Molegro Data Modeller Manual** from the menu bar. The executable for the PDF reader can be specified in the Preferences.

14.2 The Molegro Website

The Molegro website also offers certain help facilities. Please visit http://www.molegro.com/mdm-product.php to see Movies, Tutorials, and other information available.

14.3 Technical Support

Technical support is available for commercial licenses (industrial and academic) only. To obtain additional support, send an email to support@clcbio.com.

15 Appendix I: Statistical Measures

This appendix defines the statistical measures used in Molegro Data Modeller.

15.1 General Symbols Used

- N : Number of data points (records/observations) in a dataset.
- x_i : The value of variable x for data point i.
- $\overline{x}$: The mean of variable x.

15.2 Univariate Analysis

Mean

The *mean* is the arithmetic average of a set of values. The mean of variable x is defined by:

$$\overline{x} = \frac{\sum_{i=1}^{N} x_i}{N}$$

Median

The *median* is a number dividing the higher half of a distribution from the lower half, i.e., at most half the data points in the distribution have values less than the median and at most half have values greater than the median.

The median can be found by numerically sorting all records and picking the middle one. If there is an even number of records, the median is taken as the mean of the two middle values.

Sample Variance

The *sample variance* measures the spread of values in a sample about the mean and is defined as:

$$\sigma^2 = \frac{\sum_{i=1}^{N} (x_i - \overline{x})^2}{N-1}$$

Standard Deviation

The *standard deviation* describes the spread of a distribution and is defined as the square root of the variance. (If the values are close to the mean, the standard deviation is small).

Skewness

Skewness is a measure of the asymmetry of a distribution.

The Skewness measure is defined as:

$$\mu = \frac{\sum_{i=1}^{N} (x_i - \overline{x})^3 / N}{RMSD^3}$$

where

$$RMSD = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \overline{x})^2}{N}}$$

Negative skewness implies that the "mass" of the distribution is shifted to the right whereas a positive skewness implies that the "mass" of the distribution is shifted to the left. Normal distributions have a skewness of zero as they are symmetrical around the mean.

Kurtosis

Kurtosis is a measure of the "peakedness" of a distribution. Kurtosis is defined as:

$$kurtosis = \frac{\sum_{i=1}^{N} (x_i - \overline{x})^4 / N}{RMSD^4} - 3$$

(which strictly speaking is the *excess kurtosis)*.

The -3 at the end of the formula is a convention to make the kurtosis of the normal distribution equal to zero.

15.3 Bivariate Analysis

Pearson Correlation Coefficient

The *Pearson correlation coefficient (*r*)* is a measure of the correlation of two variables x and y (i.e. a measure of the tendency of the variables to increase or decrease together). The Pearson correlation coefficient is defined as:

$$r = \frac{cov(x, y)}{\sigma_x \sigma_y}$$

where

$$cov(x, y) = \frac{\sum_{i=1}^{N} (x_i - \overline{x})(y_i - \overline{y})}{N - 1}$$

is the *covariance* between variable *x* and *y*.

The range of *r*-values is between -1 and 1. A value of 1 shows that a linear equation describes the relationship perfectly with all data points lying on the same line and with y increasing with x. A value of −1 shows that all data points lie on a single line, but that y increases as x decreases. A value of 0 shows that there is no linear relationship between the two variables.

Often, r^2 is used instead of *r* where the range of r^2 values is between 0 and 1. A value of 0 indicates that the two variables are not correlated and a value of 1 indicates that the two variables are perfectly correlated.

Adjusted r^2

Adjusted r^2 is a modification of the Pearson correlation coefficient that adjusts for the number of explanatory terms in a *multiple linear regression* model (i.e. number of descriptors used in the model). *Adjusted r^2* values can be negative and will always be less than or equal to the Pearson correlation coefficient. *Adjusted r^2* is defined as:

$$Adjusted\ r^2 = 1 - (1 - r^2)\frac{N - 1}{N - P - 1}$$

where N is the number of data points and P is the number of descriptors used in the model.

Spearman's Rank Correlation Coefficient

The *Spearman's Rank Correlation Coefficient (ρ)* is a rank-ordered correlation coefficient that uses the ranking of the data points instead of the raw data points. The *Spearman's Rank Correlation Coefficient* is defined as:

$$\rho = 1 - \frac{6\sum_{i=1}^{N} d_i^2}{N(N^2-1)}$$

where the raw data points are converted to ranks. d_i is the difference between the ranks of corresponding values of x and y and N is the number of data points.

Notice that data points with identical values are assigned a rank which is the mean of the respective ranks that would be assigned if they were not identical.

Predictive Sum of Squares

The *Predictive sum of squares (PRESS)* is defined as:

$$PRESS = \sum_{i=1}^{N} (x_{pred,i} - x_{obs,i})^2$$

where $x_{pred,i}$ and $x_{obs,i}$ refer to the predicted and observed values of variable x_i, respectively.

Notice that *PRESS* is only applicable when performing cross validation experiments, i.e., the predicted values are calculated for the hold-out dataset using a regression model trained on the remainder of the dataset.

Cross Validated Correlation Coefficient

The *Cross validated correlation coefficient* is the cross validated equivalent of r^2. The Cross validated correlation coefficient is often denoted Q^2 or q^2 and is defined as:

$$q^2 = 1 - \frac{PRESS}{\sum_{i=1}^{N} (x_{obs,i} - \overline{x_{obs,i}})^2}$$

The closer the value of q^2 is to 1.0, the better is the predictive power of the regression model being evaluated. If q^2 is much lower than r^2, the regression model is likely to be over-fitted and the predictive power of the regression model will be limited.

Notice that q^2 is only applicable when performing cross validation experiments, i.e., predicted values are calculated for the hold-out dataset using a regression model trained on the remainder of the dataset.

Classification Measures

Classification accuracy reports the percentage of data points correctly classified.

Classification Accuracy is defined as:

$$Classification\ Accuracy = \frac{C}{N} * 100$$

where *C* is the number of data points correctly classified and *N* is the total number of data points.

For a classification prediction, and a given class, we have the following statistical measures:

- True positive (**TP**) count is the number of instances where the class was correctly predicted ("hit").
- True negative (**TN**) count is the number of instances where the class was correctly rejected.
- False positive (**FP**) count is the number of instances where the class was incorrectly predicted ("false alarm").
- False negative (**FN**) count is the number of instances where the class was incorrectly rejected ("miss").
- True positive rate (**TPR**) (also called recall or sensitivity) is defined as TPR = TP/P = TP/(TP+FN)
- False positive rate (**FPR**) (also known as false alarm rate or fall-out) is defined as FPR = FP/N = FP/(FP+TN)
- True negative rate (**TNR**) is defined as TNR = TN/(TN+FP)
- False negative rate (**FNR**) is defined as FNR = FN/(TP+FN)
- Precision = TP/(TP+FP)
- Recall is the same as the true positive rate: Recall = TP/P = TP/(TP+FN)
- F-measure = 2*Precision*Recall/(Precision+Recall). The F-measure is the harmonic mean between the recall and precision. It is sometimes called the F_1-measure.

Macro-averaged F-measure

In macro-averaging, the F-measure is first calculated locally for each class (category) and then the averaged.

The Macro-averaged F-measure is defined as:

$$F\,(macro-averaged) = \frac{\sum_{i}^{M} F_i}{M}$$

where *M* is the total number of classes and F_i is the F-measure for class *i*. The macro-averaged F-measure value is defined in the interval [0,1] where larger values correspond to higher classification accuracy.

When estimating the predictive power of model, the accuracy can be

misleading. This is especially the case when dealing with *unbalanced* data: consider a data set with 100 samples, where 95 of the samples belongs to class A and 5 samples belongs to class B. A classification model could be constructed which always predicted any sample to belong to class A. This model would have an accuracy of 95%, yet it would not have any predictive power. For unbalanced data we recommend using the macro-averaged F-measure: For our trivial model, we would have a F-measure of 94,74% for class A and a F-measure of 0% for class B, which would result in a macro-averaged F-measure of 47,36%. So the macro-averaged F-measure would reflect that only one of the classes had been properly predicted.

16 Appendix II: References

[HAYKIN 1999] Haykin, S. Neural Networks: A Comprehensive Foundation. Prentice-Hall, Inc., New Jersey, 1999.

[ASUNCION 2007] Asuncion, A. and Newman, D. J. UCI Machine Learning Repository (http://www.ics.uci.edu/~mlearn/MLRepository.html). Irvine, CA: University of California, Department of Information and Computer Science, 2007.

[FISHER 1936] Fisher, R. A. The Use of Multiple Measurements in Taxonomic Problems, *Annual Eugenics*, 1936, 7, Part II, 179-188.

[OUTLIER] http://en.wikipedia.org/wiki/Outlier

[TAN 2006] Tan, P.-N.; Steinbach, M.; Kumar, V. Introduction to Data Mining. Addison Wesley, Pearson International Edition, 2006.

[GNUPLOT] http://www.gnuplot.info

[LIBSVM 2001] Chang, C.-C. and Lin, C.-J. LIBSVM: a library for support vector machines, 2001, http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf.

[FAN 2005] R.-E. Fan; P.-H. Chen; and C.-J. Lin. Working set selection using second order information for training SVM. Journal of Machine Learning Research 6, 1889-1918, 2005.

[BOSER 1992] Boser, B. E.; Guyon, I.; Vapnik, V. A training algorithm for optimal margin classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pages 144-152, ACM Press, 1992.

[PLS 2007] Jørgensen, B. and Goegebeur, Y. Course notes on partial least squares, 2007, http://statmaster.sdu.dk/courses/ST02/module07/index.html

[WU 2004] T.-F. Wu; C.-J. Lin; R. C. Weng. Probability Estimates for Multi-class Classification by Pairwise Coupling, Journal of Machine Learning Research, 5, 975-1005, 2004.

17 Appendix III: Third Party Copyrights

Mersenne Twister Random Number Generator

Molegro Data Modeller uses a derivate of the Mersenne Twister random number generator (http://wwwpersonal.umich.edu/~wagnerr/MersenneTwister.html), under the following license:

Copyright (C) 1997 - 2002, Makoto Matsumoto and Takuji Nishimura,
Copyright (C) 2000 - 2003, Richard J. Wagner, All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The names of its contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED

AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

LIBSVM

Molegro Data Modeller uses derivates of the LIBSVM algorithms (http://www.csie.ntu.edu.tw/~cjlin/libsvm/), under the following license:

Copyright (c) 2000-2006 Chih-Chung Chang and Chih-Jen Lin, All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither name of copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

MD5

Molegro Data Modeller uses a derivate of the MD5 hash algorithm "RSA Data Security, Inc. MD5 Message-Digest Algorithm", under the following license:

You may use this software free of any charge, but without any warranty or implied warranty, provided that you follow the terms of the original RSA copyright, listed below.

Original RSA Data Security, Inc. Copyright notice

Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All rights reserved.

License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing this software or this function. License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing the derived work. RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind. These notices must be retained in any copies of any part of this documentation and/or software.

Icons

The icon set used in Molegro Data Modeller is taken from:

The Tango Icon Library: http://tango.freedesktop.org/Tango_Desktop_Project

They are released under the 'Creative Commons Share-Alike license': http://creativecommons.org/licenses/by-sa/2.5/