
Mayday User Guide



<http://www.zbit.uni-tuebingen.de/pas/mayday/> Version 2.1;

November 2005

Authors:

Janko Dietzsch
Nils Gehlenborg
Stephan Symons
Matthias Zschunke

Kay Nieselt
Center for Bioinformatics Tübingen
University of Tübingen
Sand 14
D-72076 Tübingen



Copyright (C) 2004-2005 . Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are retained on all copies. The free version of the MAYDAY software package is a copyrighted work that may be freely distributed and modified under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. Alternative license terms may be obtained (for instance, for commercial purposes) from ???

The software is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. For a copy of the full text of the GNU General Public License, see www.gnu.org/licenses.



Contents

1	Introduction	4
1.1	How to read this User Guide	4
1.2	Where to learn almost everything about microarrays	4
1.3	Introduction	5
1.4	Acknowledgements	6
2	Installation	7
2.1	System requirements	7
2.2	Installation	8
2.3	Setting up preferences	9
3	Quick start tutorial	10
3.1	<i>Mayday</i> data organization	10
3.2	Open a data set	11
3.3	Global information	13
3.4	Context menu	13
3.5	Analyzing data	15
3.5.1	<i>k-Means</i> plug-in	15
3.5.2	Load analyzed data	19
3.6	Visualization of data	21
3.6.1	Expression image	22
3.6.2	Profile Plot	27
3.6.3	Box plot	31
4	The <i>Mayday</i> How-to series	32
4.1	Visualization	32
4.2	Clustering	41
4.3	R connection to <i>Mayday</i>	42
4.4	Meta Information Objects in <i>Mayday</i>	48
4.5	Databases in <i>Mayday</i>	55



1 Introduction

1.1 How to read this User Guide

A first word: this User Guide is not yet finished. You are right now reading a preliminary version of it. Please check the webpage from where you downloaded this document for updates!

1.2 Where to learn almost everything about microarrays

STILL TO COME!



1.3 Introduction

Each microarray experiment requires comprehensive and careful analysis of the obtained data. Particularly in the field of gene expression data analysis, a number of software applications exists, supporting the experimenter or data analyst in conducting this task.

Many different methods, ranging from statistical tests to clustering algorithms, data visualization tools and highly sophisticated techniques, have been and are constantly newly developed for microarray data analysis. Especially interactive visualizations, that support the data analyst in exploring the data, can be crucial to the successful interpretation of a microarray experiment. Easy and user-friendly access to a multitude of methods is of high significance to the outcome of the data analysis.

Mayday is a freely available **microarray data analysis** platform and is designed to be a flexible solution for processing microarray data. *Mayday* features interactive data visualization as well as a very generalized plug-in framework to support analytical tools. The intended audience of *Mayday* is on the one hand researchers performing and analyzing microarray experiments and, on the other hand, researchers developing new methods for microarray data analysis.



1.4 Acknowledgements

STILL TO COME!



2 Installation

2.1 System requirements

Mayday is based on the Java programming language, thus you will need at least the Java runtime environment 1.5.0 for your operating system. Please see <http://java.sun.com/> for further information, how to install Java on your machine. Note for Macintosh users: Java 1.5 is only available for Mac OS 10.4.x. See <http://www.apple.com/support/downloads/java2se50release1.html> for further information.



2.2 Installation

- First, please get the files listed below from <http://www.zbit.uni-tuebingen.de/pas/mayday/download/index.html>. Either download the complete package, including core and all plug-ins provided by this site `Mayday-Complete [date stamp]` (`Mayday-Complete-[date stamp].zip`), or download the individual JAR files of the core and the plug-ins that should be used.
- If you selected the complete package: Create an adequate home directory for *Mayday* copy `Mayday-Complete-[date stamp].zip` into this directory and unzip it. Now you will find the main file of *Mayday* `mayday-core-[date stamp].jar` and the plugin directory `plugins` inside the *Mayday* home directory:
 - `Mayday` home directory
 - + `mayday-core-[date stamp].jar` (main program file of *Mayday*)
 - + `plugins`

Inside the plugin directory `plugins` reside all plugin JARs `mayday-[plugin category]-[date stamp].jar` that are available from the *Mayday* home page.

Please make sure that you have downloaded all third party libraries that are needed by the plugins you want to use. You can find the links to necessary libraries in the section `External Libraries Quick Links` of the *Mayday* download page. Create a separate directory for every library under the plugin directory and copy the according library into it.

- If you selected the core and individual plugin JARs, it is advisable for the first time to create a similar structure. But you are free to use different names and locations for the *Mayday* home and the plugin directory.
- To start *Mayday* change to the *Mayday* home directory and call from the command line of your system `java -jar mayday-core-[date stamp].jar`. If necessary you can provide additional arguments to adjust the behavior of the Java Virtual Machine.

If you experience any problems during installation please report them to dietzsch@informatik.uni-tuebingen.de. Please mention name and version of your operating system, the Java runtime environment you use and the *Mayday* release and give a short description of your problems.



2.3 Setting up preferences

If you start *Mayday* the first time you should set up the preferences.

- Open *Mayday* as described in the previous section.
- Select the menu item **File** → **Preferences...**
- On the rider **Browser**, type in your favorite browser. Probably your browser needs an URL switch. This is a command line option some browsers need to determine that a given string has to be interpreted as URL.
- Change to the rider **Plug-ins** and set up your plugin directory path to the plugin directory that you created during installation.
- On the rider **Editor**, specify your favorite editor.

(see *Figure 1*).

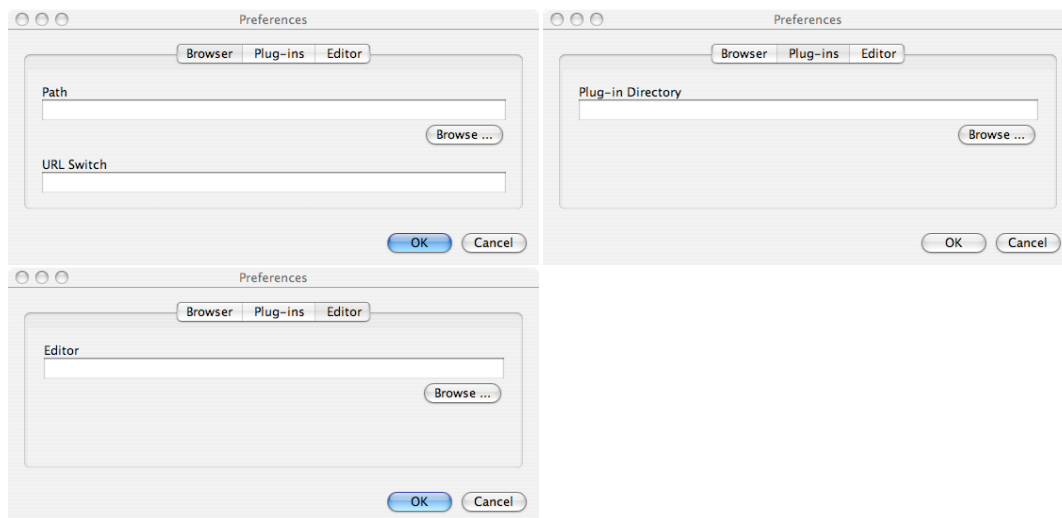


Figure 1: Edit preferences for browser, plugin and editor



3 Quick start tutorial

CAUTION: This tutorial still refers to an older version of *Mayday*. Differences in screenshots, etc. are therefore possible. However, most of this can be done with the new version of *Mayday*.

If you use *Mayday* the first time we suggest to work through the following sample session. From now on you need the sample data set (see *Section 2.2*).

3.1 *Mayday* data organization

To understand the handling of *Mayday* you need to know how the data is organized. *Mayday* is an application intended to analyze microarray data, so the underlying data set is an expression matrix. This matrix contains the expression values of microarray experiments. The values of a row belong to one probe and the values of a column belong to one experiment. Every probe (or gene profile) has an identifier. The identifiers of probes and experiments are taken from the expression matrix. They are expected in the first column (probe identifiers) and the headline (experiment identifiers).

The data structure representing the expression matrix is called master table. The results of the analysis tools working on a master table are subsets of this master table. These subsets are called probe lists. A probe list contains only the identifiers of the included probes and is internally connected to the master table. Probe lists are sets in a mathematical sense, so every probe identifier is contained only once in one probe list. However, a probe identifier can be contained in several probe lists. The interface between *Mayday* and the analysis tools use the probe lists to refer to the expression matrix. To guarantee that at least one probe list exists, a global probe list will be created automatically. This global probe list consists of all probes.



3.2 Open a data set

- Select the menu item **Data Set** → **Open...**
- Find the directory where the sample data set has been stored.
- Open the file `Spellman_alpha_25.dat`.

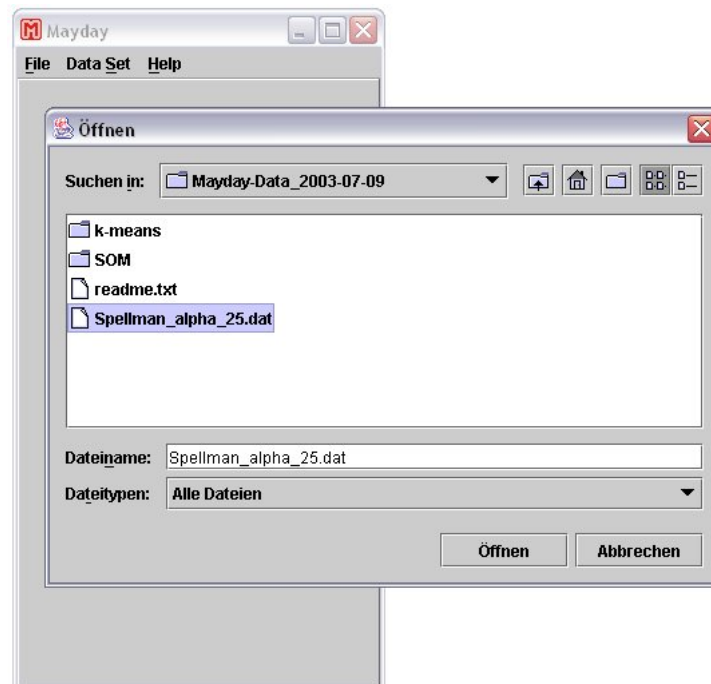


Figure 2: Open a data set

The file contains a tab-separated matrix of expression values from 528 yeast genes. It was extracted from Spellman's experiment to identify cell-cycle-regulated genes of the yeast *Saccharomyces cerevisiae* [?].

- Type in a name for the data set, or confirm by pressing **<Enter>**.
- Choose the data mode *log2 ratio*.

The data mode is a parameter that *Mayday* needs to interpret the data in order to identify allowed and forbidden operations. It determines whether the file contains absolute, logarithmic, or ratio values. What data mode you choose depends on the loaded data set.

Now a global probe list has been created.



If you want to change the name of the data set and give some explaining information, do the following:

- Open the menu item **Data Set** → **Properties...**

Quick Info is meant to contain a short description of the data, e.g. one short phrase or sentence.

Info is meant to contain some further information, maybe a whole article formatted in HTML.

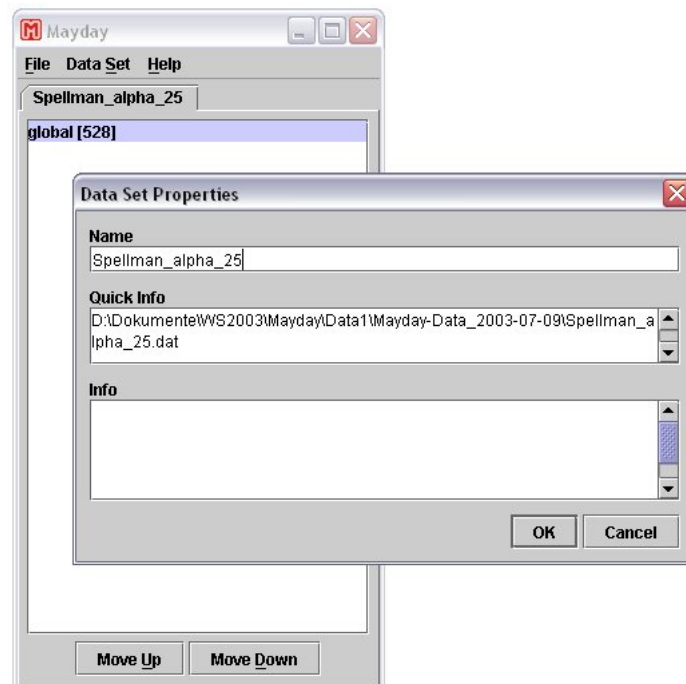


Figure 3: Data Set Properties



3.3 Global information

- Double click on the *Spellman_alpha_25* rider to get some information about the data set. You will see minimum, maximum etc.

For example, you see that the *Spellman*-data consists of 528 probes (genes) and for every probe there are 17 experiments.

In *Mayday* there is a difference between explicit and implicit probes. Explicit probes are those read from an input file. Implicit probes are implicitly contained in the expression matrix such as the mean over all explicit probes or the centers of a k-means cluster. In *Figure 4* you see an explicit global maximum/minimum which means the maximum/minimum of the whole expression matrix. Here there is no implicit probe yet, so there is no implicit maximum/minimum.

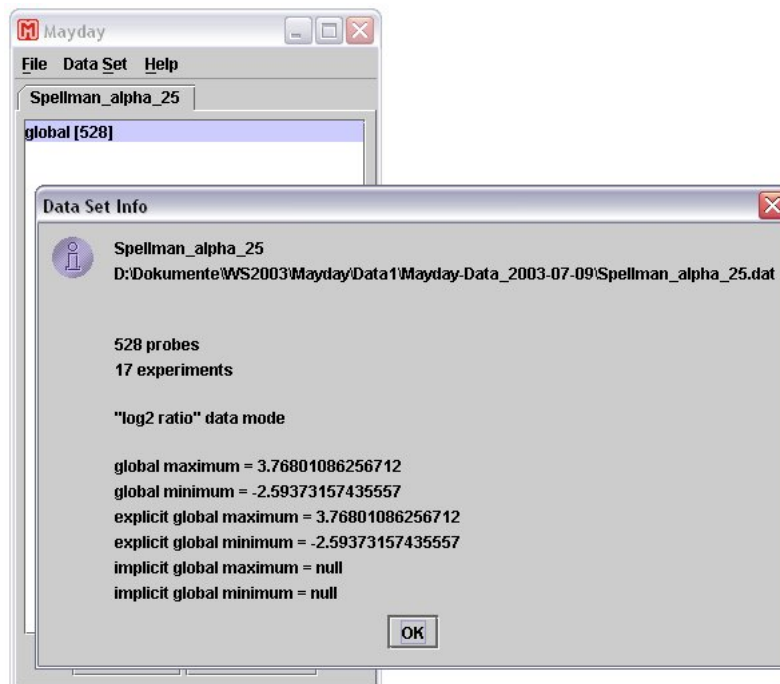


Figure 4: Data Set Info

3.4 Context menu

An important concept of *Mayday* is the context menu which can be opened by clicking the **right mouse button**. The context menu offers almost the whole functionality of *Mayday*.



- On the *global*-entry, click the **right mouse button**.

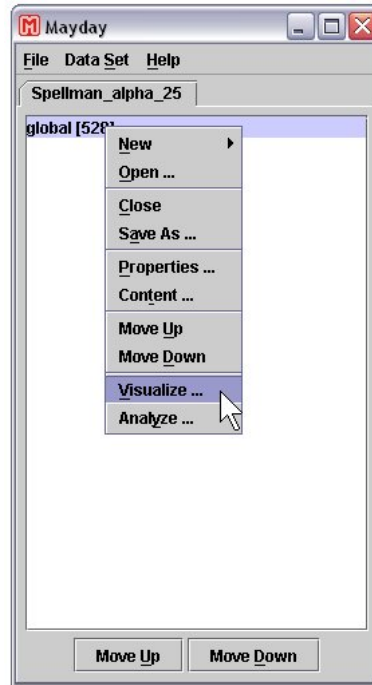


Figure 5: The context menu



3.5 Analyzing data

Mayday has two possibilities to get analyzed data. The first is to analyze the data via plug-ins, for example the *k-means* cluster plug-in. The second possibility is to load pre-analyzed data from a file (see *Section 3.5.2*).

Mayday provides a flexible mechanism to integrate a multitude of established and new data analysis methods. Plug-ins are a concept to capture distinct functional units in interchangeable software modules. The collaboration of these software modules is the basis for the functionality of the whole platform.

On the one hand, the plug-in interface allows *power users* to customize *Mayday* to their needs and on the other hand, it is possible for experts to test their new methods within an existing infrastructure for handling and visualization of data. For available plug-ins check the URL <http://www.zbit.uni-tuebingen.de/pas/mayday/mayday.html>.

3.5.1 *k-Means* plug-in

To apply the *k-means* cluster algorithm:

- Click the right mouse button (context menu) over the *global [528]* entry.
- Open the menu item **Analyze...**

There you will find the Analyzer (see *Figure 6*) which contains all usable plug-ins of your *Mayday*-installation, sorted by categories.

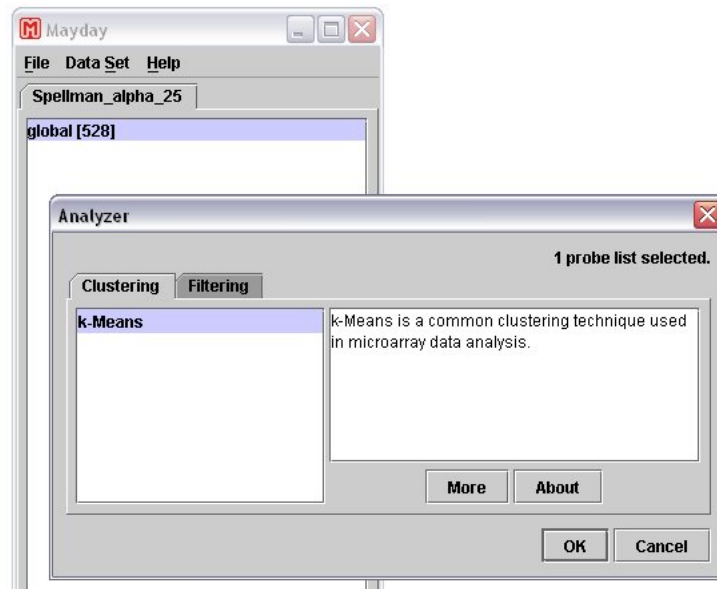


Figure 6: Analyzer

- Select the rider **Clustering** (see Figure 6).
- Choose the entry *k-Means* and press **OK**.
- Set the number of clusters to 9 (see Figure 7).
- Press **Run** to confirm.

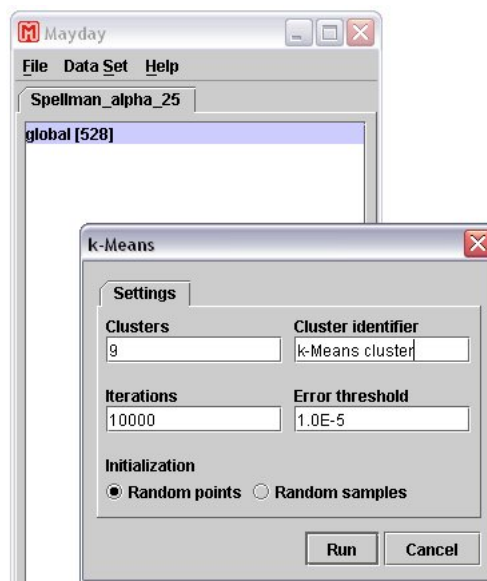


Figure 7: *k-Means* parameters



Other parameters are the identifier for storing the resulting clusters in the master table, the iterations and the error threshold. At the bottom you can select which kind of method should be used to generate the initial cluster centers. *Random samples* means that the centers are randomly chosen from the given data set. *Random points* computes virtual centers.



The resulting 9 clusters will be shown in the master table with different colors. These colors are used for the visualizations. To change the color by hand you can use the probe list properties which are available via the context menu (*right mouse button*).

Notice that almost every functionality is accessible via the right mouse button context menu.



3.5.2 Load analyzed data

Mayday provides the possibility to load pre-analyzed data from a file. Imagine an analysis procedure that is only available with a third party software. *Mayday* allows to visualize the results of this procedure, given that it is saved in a format that *Mayday* can read. The input file contains several clusters given by a grouping of the probe identifiers which must be stored in the XML-based probe list file format. Since probe lists (that only contain probe identifiers that occur in the expression matrix) are the central data concept in *Mayday*, you have to guarantee that only those identifiers occur in the probe list file that can also be found in the expression matrix. An example for a probe list file is given in *Appendix A*.

- Select the **context menu**.
- Select the item **Open...** (*NOT* the **Data Set** → **Open...**-menu).
- Change to the Spellman directory, there you can find the SOM directory.
- Open all 9 files.

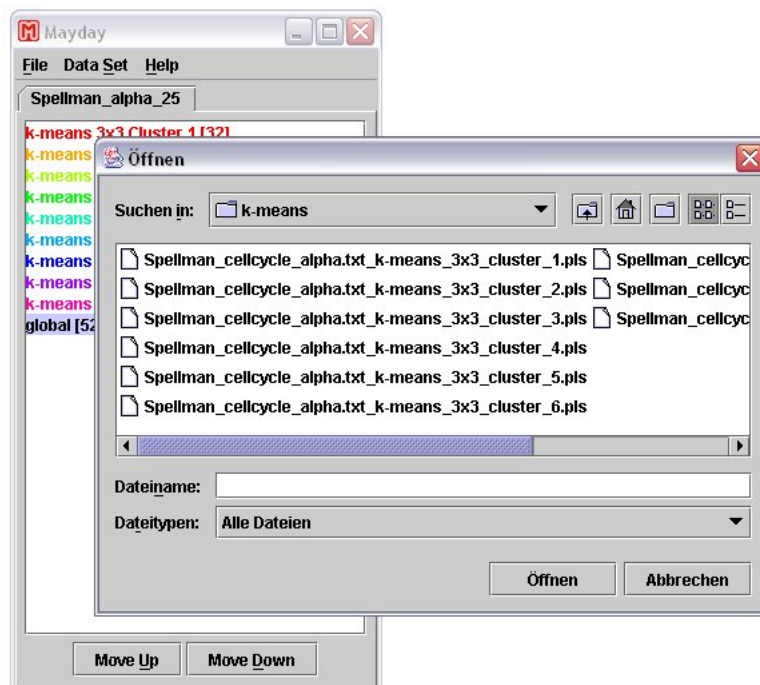


Figure 8: Load pre-analyzed data



3.6 Visualization of data

So far, three different graphical viewers are implemented in *Mayday*: profile plot, box plot, and expression image (heatmap). They are available via the menu item **Viewers** → **New**.

- Select all clusters.
- Open the **context menu**.
- Select the item **Visualize...**

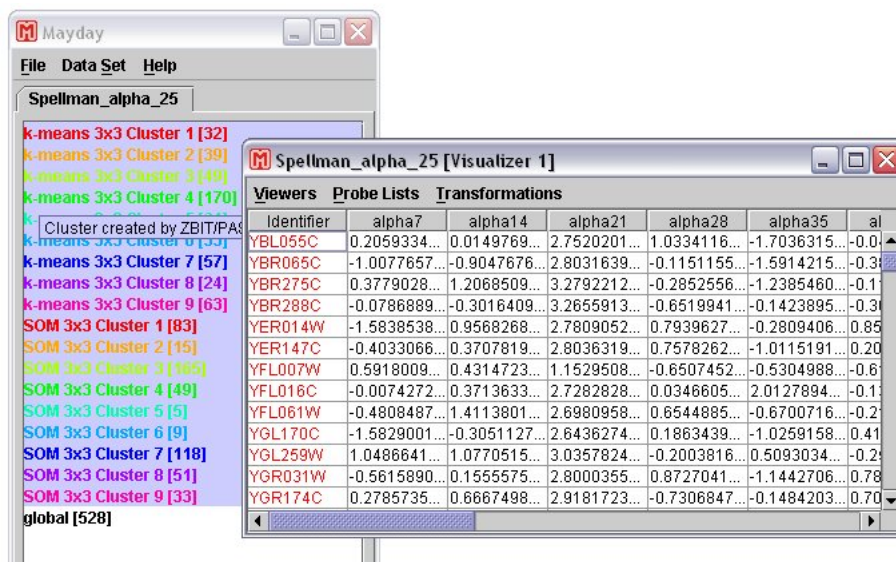


Figure 9: The visualizer

You can see the expression matrix in a tabular view. You see only those probes which you chose in the master table. For example, if the first cluster of the *k-means* analysis is selected, you will only see the 32 probes of this cluster in the tabular view (see Figure 9). Additionally you should know that every probe will occur only once in this table, no matter how often it is contained in different clusters.

An important feature of all views is the export to different file formats. The tabular view of the expression matrix can be exported to a plain-text file, so that it can be opened with Excel e.g.

- Select the menu item **Viewers** → **Table** → **Export...**

The graphical views can be exported to several graphic formats (see Section 3.6.1).



3.6.1 Expression image

The expression image is often called *heatmap*. The heatmap visualizes the expression matrix by coding the expression values of a probe with a given color palette (see *Figure 10*).

- In the *Visualizer* select the menu item **Viewers** → **New** → **Expression Image**.

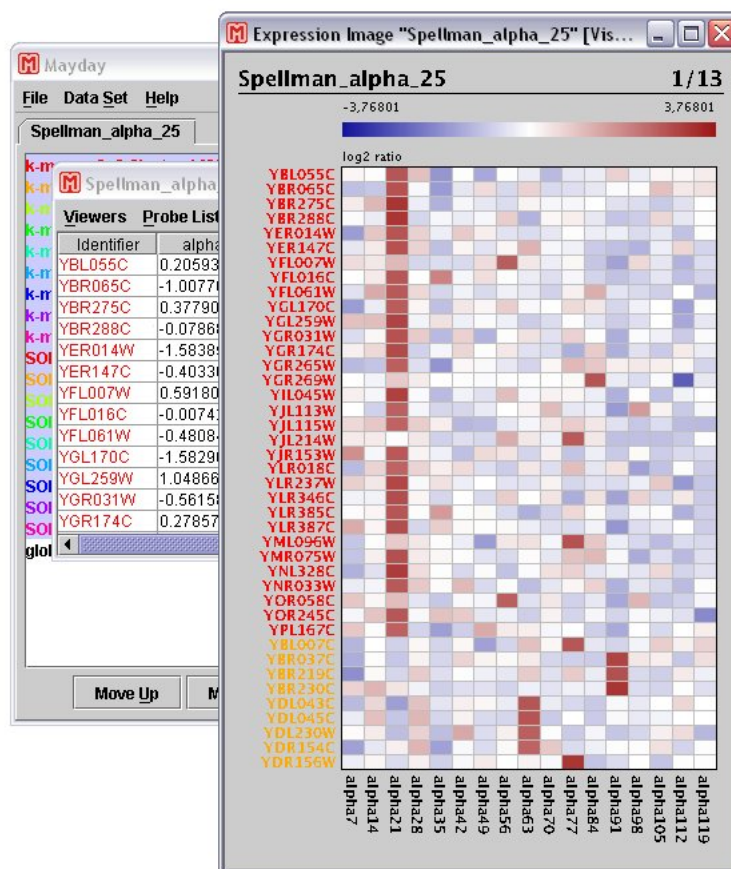


Figure 10: Expression Image

The heatmap will be scaled automatically (see *Figure 10*).

- Press the hotkeys <CTRL><+> or <CTRL><-> to zoom in or out.
- Double click on the expression image to get further information about a specific probe.

You will see the expression value and the probe lists containing this probe, such as the cluster names, etc.



- Open the menu **context menu** → **Settings**.

Color Via the **Color**-menu item you can change the color range of the expression image, maybe you prefer the widely-used green/black/red palette.

Probes Per Page Via the **Probes Per Page** → **User-defined...**-item it is possible to modify the number of probes shown on one page. You will get the heatmap of the whole expression matrix on the current page, if you set the number to 528 (or above).

An important feature is to export this image to a file. To export an image from *Mayday* to a picture file format you need an installed Batik SVG Toolkit (see *Section 2.2*). **Notice that only the shown page will be exported.**

- Open the menu item **context menu** → **Export**.

You are able to choose between different file formats, SVG as a vector based format, the others are pixel based.

A feature in all views allows to create new probe lists by selecting probes in the expression image.

- Hold the <**CTRL**> key down and click on the image. You can select a number of probes.
- Apply the **context menu** → **Probe List from Selection...**-item to get a new probe list.

Again, you can edit the name of the probe list, provide a short description or change the color of the newly created probe list (see *Figure 11*).

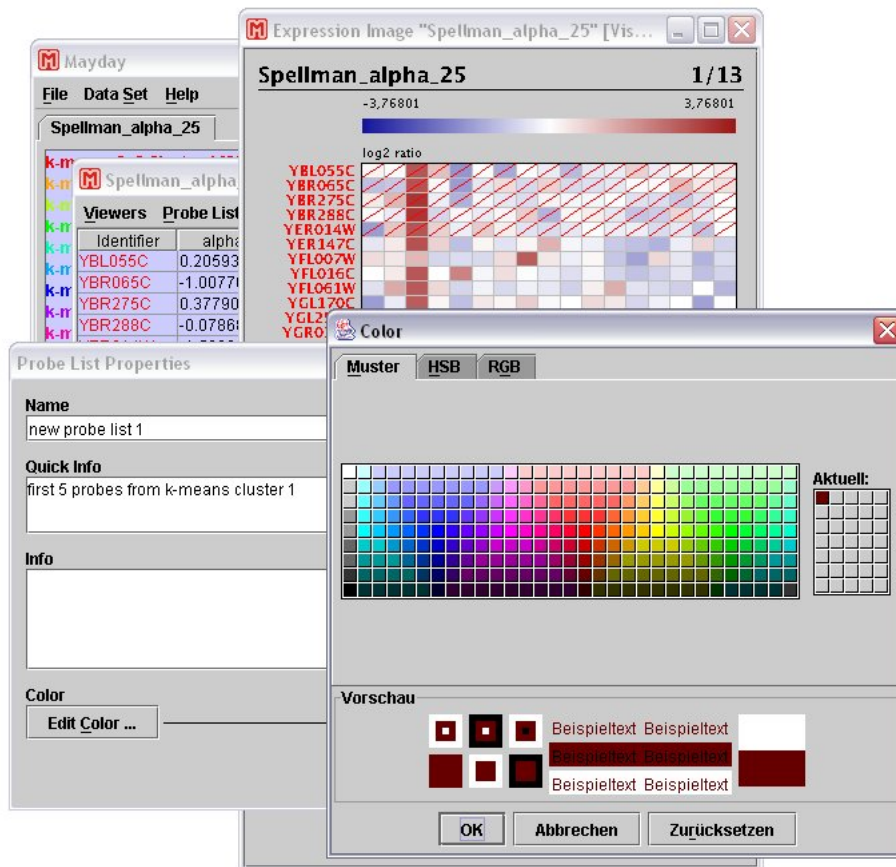


Figure 11: Choosing color

The new probe list is immediately added to the master table. The color of the selected probes has changed to the color chosen in the step before. This change has an effect in the tabular view of the visualizer and in the viewer.

The color in which the probe identifiers are displayed depends on its membership in the probe lists and the position of the related probe lists in the master table. One probe can be a member of more than one probe list.

For example, the second probe in the expression image with the identifier *YBR065C* is a member in 4 probe lists (*new probe list 1*, *k-means cluster 1*, *SOM 3 × 3 cluster 7*, and *global*). The assigned colors for this probe are dark red, red, blue, and black, respectively. The order is important, because one probe gets the color of the highest priority probe list. The order is taken from the master table. That is why the color of the first 5 probe identifiers in the heatmap has changed.

- Bring the main frame of *Mayday* to the front.
- Select *new Probe List 1* and use the **Move Down** button to bring the selected probe list down to one before the global probe list.



You will notice that the color of the first five probes will turn back to red, because now the highest probe list is *k-Means cluster 1*.

This ordering is important not only for colorizing the probe identifiers, but also for the order in the *Visualizer* and in the viewers.

- Select all *SOM clusters*.
- Move them on top of the main frame (**Move Up**-button).

For the result see *Figure 12*.

- Close the expression image viewer.

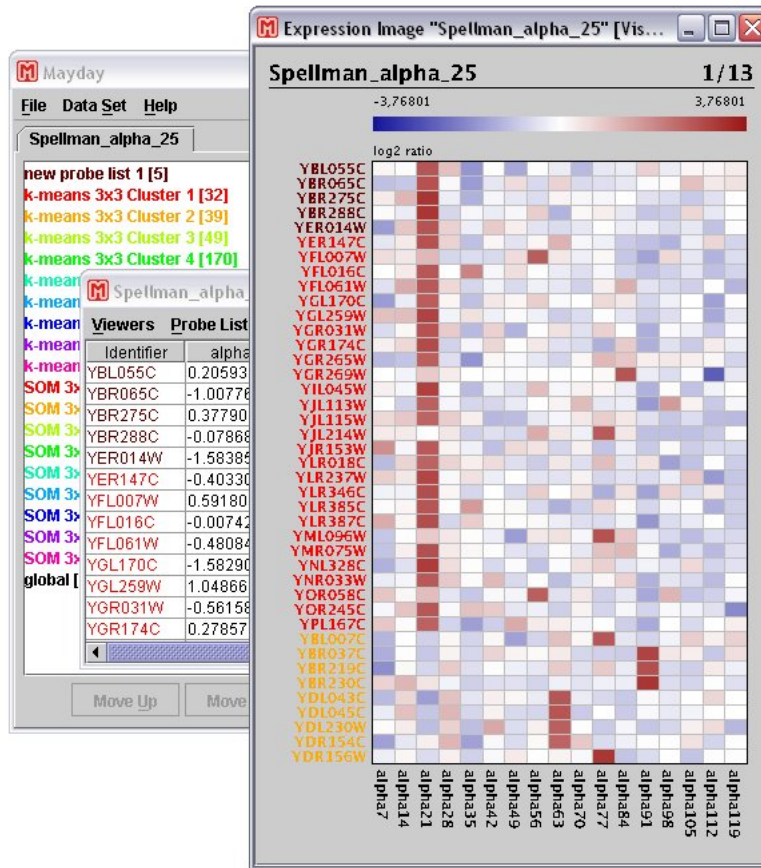


Figure 12: Expression Image, changed identifier's color



3.6.2 Profile Plot

Another important kind of visualization method is the profile plot. You are able to open a single or a multiple profile plot. *Single* means all probes are plotted in the same diagram. The multiple profile plot can show several plots simultaneously.

- Open the menu item **Viewers** → **New** → **Profile Plot** → **Single**.

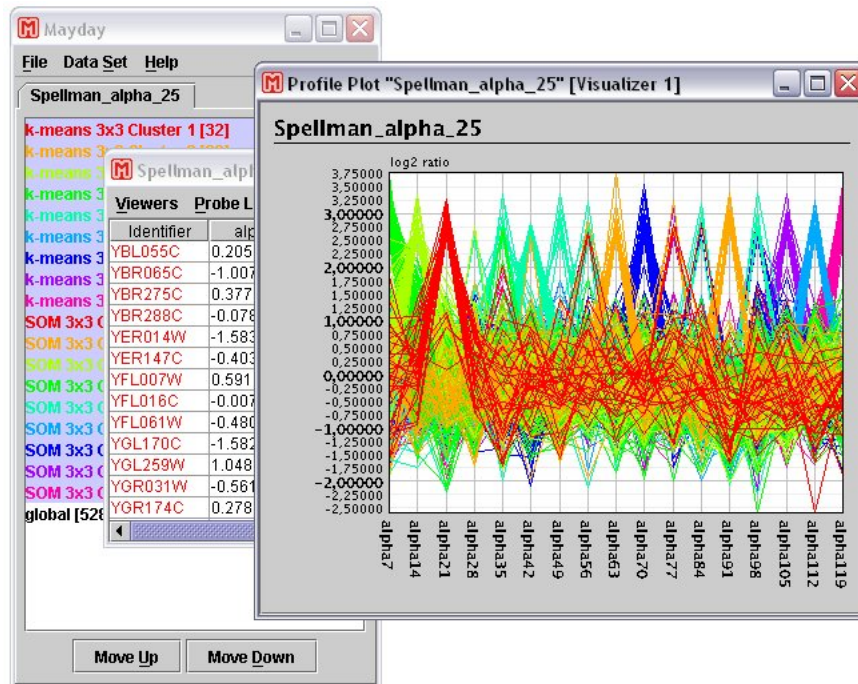


Figure 13: Single Profile Plot

- On this view, apply the **context menu** → **Export...** item to Export the view.
- Close the single profile plot.
- In the Visualizer, open the menu item **Viewers** → **New** → **Profile Plot** → **Multi**.
- Type in the number of diagrams (grid-dimensions) to plot in. Here type 3×3 to plot the 9 profile plots of the clusters computed by *k-means*.
- Make sure that all 9 *k-Means* clusters are spread over the 9 grid cells (see Figure 14).

The result is a 3×3 grid with 9 profile plots (see Figure 15).

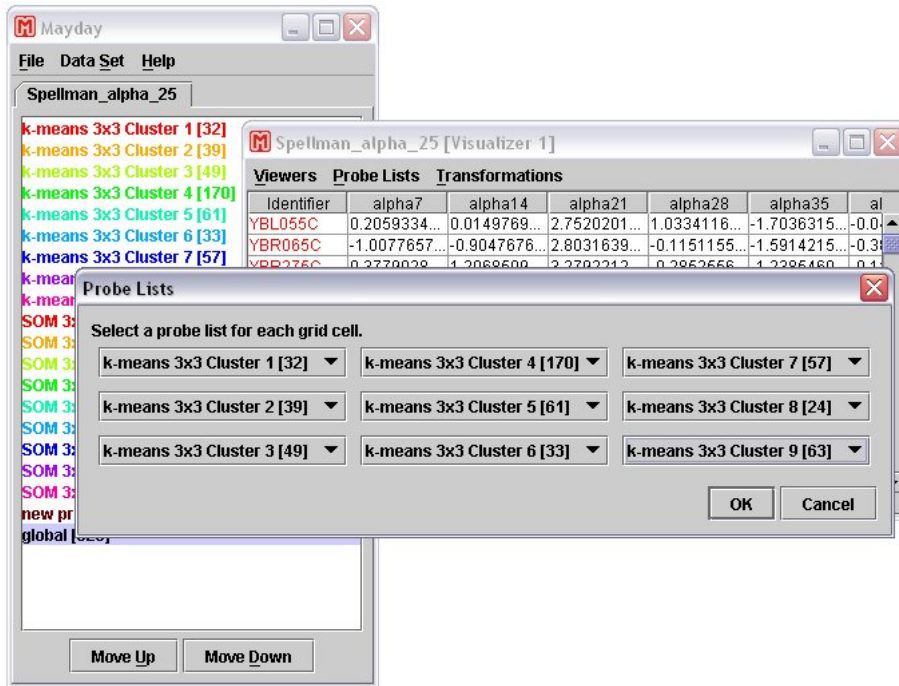


Figure 14: Select a probe list for each grid cell

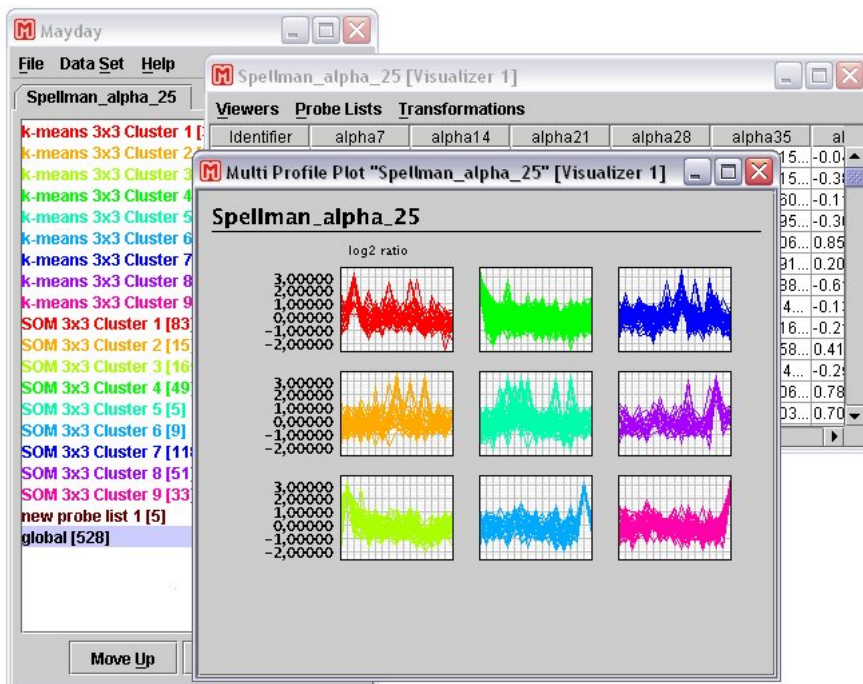


Figure 15: Multi Profile Plot



- To zoom in press <CTRL><+>.
- Click on a profile to to select a probe.

The selected probe will be marked in red color.

- Or, open the **context menu** → **Go To** → **Probe...** and type in the probe identifier which you are interested in, e.g. *YBR065C* (see *last section*).

Remember the color priority ordering from the last section. This allows to compare the results of two different clusterings, for example.

- Click on the main frame of *Mayday* and bring the *SOM* cluster probe lists to the top of the master table (**Move Up**-button).

The colors of the plotted lines change immediately. Subplots with only few different colors show that the two clusters of the different algorithms are very similar to each other, while many colors in one subplot represent a large diversity between the two algorithms (see *Figure 15*). Notice that the movement of probe lists can take some time. The reason is that the plots are recalculated.

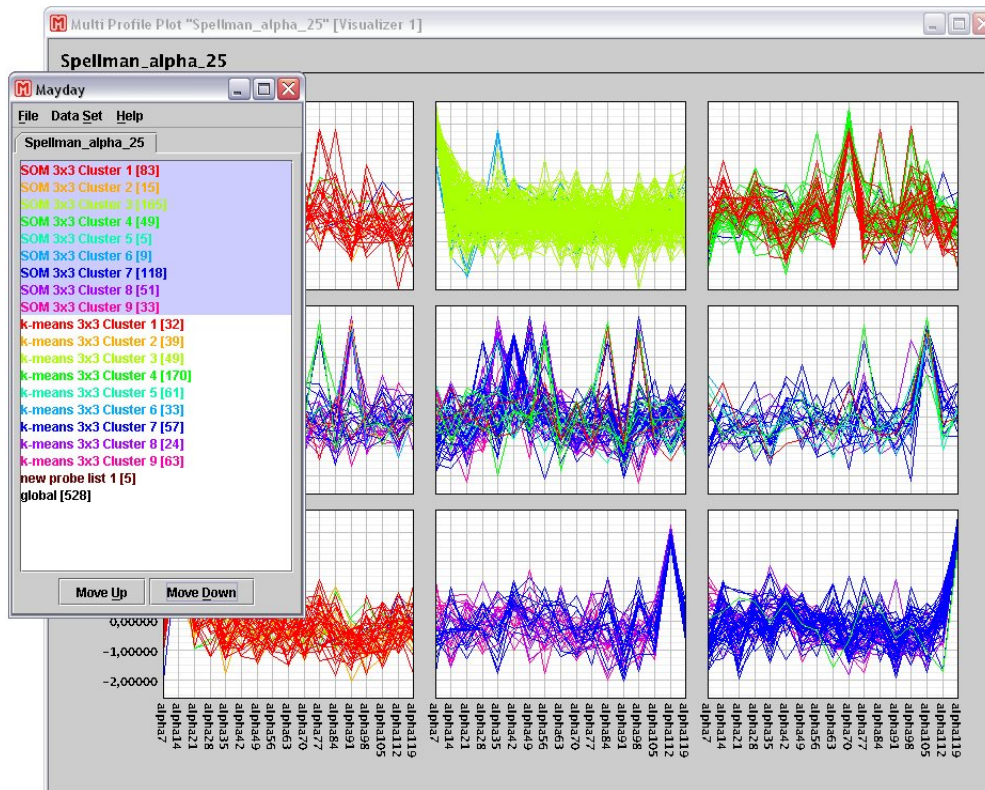


Figure 16:

Have a look at the grid in the middle of the top row. There you can see profiles of two different colors. The blue curves are somewhat hidden. This is the result of the layer concept realized in *Mayday*.

- In order to bring them to the front open the **context menu** (on this subplot) → **Layers** → **SOM 3 × 3 cluster 6** → **Bring To Front**.

Now the blue curves are on top of the green ones.



3.6.3 Box plot

The box plot is a method, often used in statistics, to investigate data variation. For every experiment there is a bar chart representing the minimum, maximum, median, 1st quartile and 3rd quartile over all probes of a specific probe list.

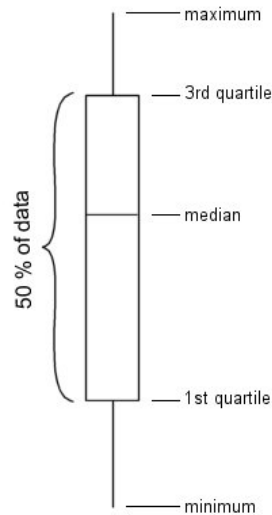


Figure 17: The box plot

The box plot is the third implemented viewer in *Mayday*. It gives a visual overview of the complete data set and offers an easier identification of the difference between several probe lists. The box plots can be shown in single and multiple mode.

-
- Use the menu item **Viewer** → **New** → **Box Plot** → **Multi**
 - Open an 1×2 box plot.
 - Choose *SOM cluster 1* and *SOM cluster 6* to discover the differences of these two clusters. (see *Figure 18*)



4 The *Mayday* How-to series

4.1 Visualization

Getting Started with Visualization in *Mayday*

Nils Gehlenborg

Visualization in *Mayday*

Before we get started a word of warning: This How-To describes only fundamental features of *Mayday*'s visualization facilities. This is neither an in-depth introduction nor a tutorial. Be aware of this. Also keep in mind that *Mayday*'s visualization features are continuously being extended and improved, thus the information in this document might not always reflect the current state of the software.

At the moment there exist six visualization plug-ins for *Mayday*:

- A tabular viewer that visualizes the actual values of the expression data in form of a table.
- A box plot.
- A multi box plot, i.e. a viewer that displays several box plots in a grid.
- An interactive profile plot.
- An interactive multi profile plot, i.e. a viewer that displays several interactive profile plots in a grid.
- An interactive enhanced heatmap.

Each of these plug-ins and their common features will be briefly described below.

Downloading and Installing the Required Software

Mayday offers functionality through plug-ins that are developed independently of the core software. Thus it is necessary that you download and install the visualization plug-ins for *Mayday* before you can visualize your data.

Visit <http://www.zbit.uni-tuebingen.de/pas/mayday/download/index.html> and download the latest snapshot or release version of the visualization plug-ins. Snapshots are usually uploaded to the website when new features were added and when bugs were fixed, they might not have been tested thoroughly.



Currently all visualization plug-ins are bundled into a single jar-archive file named `mayday-visualization-*.jar`. The full name depends on whether you want the snapshot or the release version.

Download the jar-archive into the plug-in directory of your *Mayday* installation. If you plan to export visualizations of your data to image formats such as PNG, JPEG, TIFF or SVG you also have to download the *Batik* library provided on the Mayday website. Alternatively you can download it from the website of the Apache project (<http://xml.apache.org/batik/>). For the time being it doesn't matter where you install the *Batik* library, as long as you remember where!

If you were running *Mayday* while you installed the plug-ins close your session and restart *Mayday* so the new plug-ins will be loaded. Go to "File → Plugins" and select the tab "Visualization". What follows is a slightly tedious process: for every visualization plug-in you have to specify the path to the *Batik* library installation. To do so select the plug-in from the list on the left and click the "Preferences" button. Then click on the "Add ..." button and go to your *Batik* installation. Do you still remember where you placed it? Go to the "lib" subdirectory and select all jar-archives beginning in "batik-" and confirm. Hit "OK" in the preferences dialog. Do this for each visualization plug-in.

Some Important Background Information

As for all other plug-ins the basic unit is a probe list. Thus you can only visualize probe lists, i.e. sets of probes, never individual probes¹.

In order to visualize a probe list you have to select it (or more) in the probe list manager. The probe list manager is the list that displays all probe lists of a data set. Click the selected probe list with the right mouse-button (right-click it) to open the context menu. At the bottom of the menu you'll find the plug-in categories. The sub menu "Visualization" lists all plug-ins that you can use to visualize your data. Select one of the visualization plug-ins. The following will (should?) happen: Two windows will open, one contains the viewer you selected in the sub menu and the other one will contain the tabular viewer. Currently it is not possible to open any other viewer without opening the tabular viewer. Nor is it possible to open the tabular viewer individually as it is part of the so-called "visualizer".

Here is why: A visualizer manages one or more viewers and will be created every time you select a visualization plug-in via the probe list manager. This means that you can have more than one visualizer! They keep track of which viewers have been activated and link them to each other. If you want to add a viewer to a visualizer you can do so by selecting one from "Viewers → New → ..." in the window that contains the tabular viewer. The number of the visualizer that a viewer belongs to is displayed in the viewer's window title. You can get an overview of all viewers of a visualizer by clicking on "New". At the bottom of the menu is a list of all viewers

¹Well, this is a little confusing since you could create a probe list that contains only a single probe and then you would actually be able to visualize an individual probe.



that are managed by the visualizer. Select one to bring its window to the front.

As mentioned before, a visualizer links its viewers to each other. For instance, if you have two profile plots that belong to the same viewer and you select the profile of a probe (and thus the probe) in one of them it will also become selected and highlighted in the other profile plot (if it is not hidden, more on that later). This works with all kinds of viewers that allow you to select probes.

A visualizer not only manages viewers but also the data that those viewers display, i.e. one or more probe lists. It is possible to add and remove probe lists from a visualizer and in consequence from the viewers of that visualizer. If you close a visualizer all of its viewer will be closed automatically without a warning. Keep this in mind!

Common Features of all Viewers

The Almighty Context Menu

As you will notice quickly there are no menu bars or tool bars in the viewer windows². This is to save space for visualization of the actual data. The features of the viewers can be accessed using a context menu similar to the one of the probe list manager in the main window. The context menu can be accessed by right-clicking anywhere in the area of the plot.

Color

In the enhanced heatmap viewer, the tabular viewer and profile plots probe identifiers and profiles, respectively, are plotted in the color of their top level probe list. How the top level probe list is determined is too complex to describe here. In general the top level probe list corresponds to that probe list out of all probe lists the probe is contained in, which is assigned the highest priority in the probe list manager. The higher up the probe list is in the probe list manager, the higher the probe lists priority.

Please note that probes cannot be assigned a color individually. Colors can be assigned to probe lists using their properties dialog, which is for instance accessible from the probe list manager.

Layers

The enhanced heatmap viewer, the tabular viewer and profile plots are organized into layers. Layers are a concept adopted from image editing and desktop publishing software. In *Mayday* each probe list contained in a viewer represents a layer. Think of a stack of transparencies and you get a good idea of what the layers concept does. Layers can be rearranged to either the top or the bottom of the stack. Further they can be hidden, i.e. temporarily removed from the stack.

²The tabular viewers is a special case because it is integrated into the visualizer.



In profile plots the layer concept enables you to get an undisturbed view on a probe list of interest by making it the top most layer. In the enhanced heatmap and the tabular viewer rearranging layers changes the order in which the probe lists occur in the respective viewer *if the probes are sorted according to the probe list they belong to*.

Sorting

As mentioned in the section on layers, it is possible to modify the order in which probes appear in the enhanced heatmap and the tabular viewer. It is possible to sort probes according to their assigned identifier, their assigned top level probe list and according to meta information objects. Sorting according to meta information objects is only possible if the meta information objects can be sorted themselves.

Selection

In a viewer that displays a probe as an individual entity, i.e. all viewers but box plots, you can select one or more probes for further processing.

Probes are selected by clicking on their representation in the viewer. If multiple probes are to be selected the Ctrl key has to be pressed during clicking. The selection can be cleared by clicking in an empty area of the plot³. Selected probes are highlighted depending on the viewer you are using. Another way of selecting a single probe is by using the “Go To →Probe ...” function of the context menu. There you can enter the name of a probe and if it is found it will become selected. So what can you do with selected probes? For instance, it is possible to look up probe identifiers of selected probes in databases like Ensembl or the Saccharomyces Genome Database (SGD). Database lookup can be accessed via “Look up selection →...” in the context menu. Besides database look up it is also possible to create a new probe list from a set of selected probes. This is done by selecting “Probe List from Selection ...” in the context menu.

If you double-click an unselected probe representation in one of the viewers mentioned above, you will see the properties of that probe displayed in a dialog. In this dialog you can double click on probe lists and meta information objects displayed in the corresponding lists to obtain more information.

It is important to note that the selection is a property of the visualizer, i.e. if you select a probe representation in one viewer of a visualizer, the probe will also become selected in all other viewers of that visualizer. This feature is very helpful for data exploration.

Export

You can export data displayed in any viewer to an external file selecting “Export ...” in the viewer’s context menu. The tabular viewer is exported to a text file, while the graphical viewers are exported to image files.

³In this case you actually have to click within the area of the coordinate system not into the gray area surrounding it.



Please note that you need to have the Batik library installed and setup for use with *Mayday* to export image files.

Special Features of Graphical Viewers

Zooming

All graphical viewers can be zoomed. You can zoom in and out of a visualization by selecting “Zoom →Zoom In” or “Zoom →Zoom Out” respectively. Alternatively you can use the corresponding keyboard shortcuts.

If you want to fit the frame of the containing window to the size of your visualization click “Zoom →Fit Frame”.

Axes, Grids and Labels

Minima and maxima of axes, or of the color gradient in case of the heatmap viewer, are equal to the minimum and maximum values in the whole data set. The color gradient is artificially made symmetrical by using the maximum absolute value.

Labels on axes are only displayed if there is enough room to show them. If you don't see labels on axes you should try to zoom in. If you see labels on axes but actually don't want to see them you can turn them of. See “Settings →Scales →...” for options.

All graphical viewers feature a grid that can be turned on or off. Grid lines can be turned on and off individually per axis. In profile and box plots the grid can be displayed on top of everything else contained in the plot. Features concerning the grid can be accessed via “Settings →Grid →...”.

Caption and Legends

In order to include information about probe lists and settings in exported viewers you can add legends to your graphical viewers and set a descriptive caption. For options see “Settings →Legends →...” and “Settings →Caption →...” respectively.

Multi Plots

Multi profile and multi box plots are viewers that contain a grid of several profile plots respectively box plots. If you open a new multi plot you will be asked to define the dimensions of the grid and which probe lists should be displayed at which position in the grid.



A Little More Detail on the Enhanced Heatmap Viewer

General

The heatmap viewer implemented in *Mayday* has several unique features. For instance, the heatmap viewer can integrate meta information about probes into the visualization using various graphical representations. More information about the concept and theory of the enhanced heatmap viewer can be found in Nils Gehlenborg, Janko Dietzsch and Kay Nieselt, A Framework for Visualization of Microarray Data and Integrated Meta Information, *Information Visualization* (2005) 4, 164 - 175 (doi:10.1057/palgrave.ivs.9500094). See the Meta Information How To for more information on how generate and load meta information in *Mayday*.

Browsing the Heatmap

The heatmap viewer is organized into pages that each hold a certain number probes. You can define the number of probes per page depending on your need. See the “Settings → Probes Per Page →...” sub menu for options.

Pages can be browsed using either the “Pages →...” sub menu in the context menu or the following shortcut keys:

- Ctrl + Page Up: Go to previous page.
- Ctrl + Page Down: Go to next page.
- Ctrl + Home: Go to first page.
- Ctrl + End: Go to last page.

Enhancing the Heatmap with Nominal Information

You can add additional columns containing nominal, i.e. categorical, meta information to the enhanced heatmap viewer. See the Meta Information How To for more information on how to obtain and handle meta information in *Mayday*.

Assuming you have loaded nominal meta information for at least some of the probes contained in the heatmap, you can add an additional column via “Enhance → Additional Columns ...” in the context menu. Select “Add Nominal Column” in the dialog and one or more meta information object groups. You can change the colors assigned to the categories by clicking on the corresponding cells in the table. You can also change the name of the new column.

Once you have added the new column you can change its settings (colors, name) either by holding down the Alt key and double-clicking the a cell of the corresponding column or via “Enhance → Additional Columns ...” in the context menu.

Enhancing the Heatmap with Scoring Information

If you want to enhance the heatmap viewer with scoring information you need to have loaded scoring meta information objects for at least some of the probes



contained in the heatmap. See the Meta Information How To for more information.

Additional columns Adding further columns containing scoring information to the enhanced heatmap works exactly like adding nominal columns to the heatmap viewer. However, instead of one color per category a color gradient is employed that ranges from 0 (not relevant at all = lowest score) to 1 (most relevant = highest score). You have to define only the colors for 0 and 1.

Height Scaling Meta information for scoring probes can also be used to scale the vertical height of heatmap rows. You can access this feature via “Enhance →Height ...” in the context menu. Using the threshold you can determine how many rows will be scaled up and how many will be scaled down. Further you can set a maximum scaling factor and a minimum scaling factor. Remove or edit the height scaling by clicking “Remove” or “Edit ...” respectively, in the meta information selection dialog.

Additional Color Gradient Another way to include meta information into the heatmap viewer is through use of an additional color gradient. The additional color gradient is orthogonal to the basic green-black-red color gradient. It can be either be adding a blue component or an alpha (transparency) value to the original gradient. Probes with high scores (close to 1) will either be represented with a large blue component (i.e. colors are from a cyan-blue-magenta gradient) or very opaque (i.e. not transparent or faded out). Access this feature via “Enhance →Color →...”. Remove the additional color gradient by clicking “Remove” in the meta information selection dialog.

Changing the Color Gradient

If you don't feel comfortable with the default green-black-red color gradient you can change it via “Settings →Color Gradient ...”. Besides colors you can choose between two different types of gradients, either a linear gradient or a sigmoid gradient. Further it is possible to reduce the resolution of the gradient and thus the of the heatmap, which is sometimes useful to highlight different classes of expression profiles.

Accessing Data

The values underlying the color-encoding in the heatmap can be accessed by holding down the Alt key and right-clicking a cell of interest in the heatmap. A tool tip like display will show the value encoded by the corresponding color. This works both for regular cells as well as for cells in additional columns.



More Visualization Options

Further visualization options in *Mayday* are offered by the hierarchical clustering plug-in, which provides several variants of a tree view of a hierarchical clustering. The R interpreter plug-in offers a manifold of options for visualization of data, however, most of those visualizations are not interactive.

Known Issues

CPU Usage

Profile plots and in particular multi profile plots are very expensive to compute. Depending on the size of your data set and your machine it might take up to several minutes to compute a multi profile plot. The plots have to be recomputed every time you interact in any way with the visualization. Currently there is no progress indicator and the machine might seem to be stalled. It is not! Keep this in mind if you work with large data sets.

Memory Usage

Currently some viewers cause “memory leaks”. That means that they do not return the memory resources they were using even after you close them. This will make your machine slow and eventually cause the Java Virtual Machine to reach it’s maximum heap size. We are working on this problem and it will be fixed as soon as possible.

Multi Plots

There are several known issues with the multi profile plot and the multi box plot.

- Zooming into the visualization sometimes causes the visualization to be partially drawn outside the visible area. A workaround for this bug is to zoom in one more time and then to zoom out.
- Exporting multi plots is generally possible, however there are problems with defining exact dimensions. In some cases the exported images were found to be much smaller or larger than defined. If you want to export a multi plot just zoom it to the size you would like to export and then export it without defining new dimensions.

Keyboard Shortcuts

On some computers the keyboard shortcuts displayed in the context menus work, on others they don’t. They don’t work properly on Macs and they don’t work properly with some keyboards with an American keyboard layout. You can always



use the corresponding menu entries, even though that might be annoying in some cases. That's why we introduced shortcuts in the first place ...





4.2 Clustering



4.3 R connection to *Mayday*

How To Use The R For Mayday Plugin.

Matthias Zschunke

R for Mayday - Quick start tutorial

R for Mayday provides an extensible interface for applying functionality available with the *R project* to the Mayday data structures.

System Requirements

R for Mayday requires

- at least one working *R* installation ($\geq 1.9.0$, see www.r-project.org),
- a command interpreter (*sh* on Linux/Unix/MacOSX, *cmd.exe* on Windows NT/2000/XP, *sh.exe* on Windows 95/98/Me)
- and the *Batik SVG Toolkit* plus the *RSvgDevice* package for R – if you like to use plotting facilities in the SVG format.
- It is recommended – but not necessarily required – to have *Perl* installed for R maintenance purposes.

Download

Please download the following required files from our website www.zbit.uni-tuebingen.de/pas/mayday/download/index.html:

- the latest snapshot release of the R Interpreter plugin – the Java part of the plugin,
- the RForMayday package in *zip* or *tar.gz* format – the R part of the plugin,
- the R code examples.

Installation

1. Copy the plugins jar into the Mayday plugin directory. You can find this directory in Mayday: File→Preferences...→Plug-ins.
2. Install the RForMayday package into R. There are two possible ways:



- (a) Simply extract the zip version of the package and copy the directory into your R installations library dir, e.g. *path-to-R/library/*. This method is thought for users who do not want to bother with the installation of *Perl*. Since there are no binary libraries contained in the package, this method should work on all systems.
- (b) Or, on a command interpreter (e.g. *command line, bash*) type `R CMD INSTALL RForMayday_x.y.tar.gz` (reset *x* and *y* by the version numbers). This installs the package properly into the R library. This variant requires *Perl*!

For further information about the R package installation process have a look into the R administration documentation supplied with the R distribution.

Make sure you have R at least in version 1.9.0 (type `R --version` in your command interpreter to find this out).

3. Check the proper installation calling R and execute the following command:

```
> library(RForMayday)
```

If this yields any error messages, make sure you installed the package into the right directory, especially if you have multiple R installations on your system.

4. Extract the R code examples into your favorite directory.

Start-up

The R interpreter plugin is available through the *Data Set* → *Data Import...* or the context menu of the data sets, respectively.

Example execution

1. Make sure you have opened a data set, say the artificial data set obtainable from the Mayday homepage.
2. Using the context menu (right mouse click on the *global* probe list) open the *Interpreter* → *R Interpreter* item.
3. A dialog shows up that has two tabs, *Sources* and *File Settings*.
4. Open *File Settings*. This step is only needed at the very first plugin execution. *R for Mayday* will remember those settings.
 - (a) Select the appropriate R binary which can be found in *path-to-R/bin*, e.g. *C:/Program Files/R-2.2.0/bin/*. Make sure you have the *RForMayday* package installed in the selected R.



- (b) Select a *working directory*.

This directory is used to place the communication files and will become the working directory of the R session.

- (c) Select a filename for the log-file.

The function calls including the parameters, the exit state of the R session and the thrown exceptions are reported there. If you leave this field empty the output is directed to StdOut, i.e. is visible with the Message Dialog (see *Help*→*Show Message Window*).

- (d) Select what should be done with the communication files on exit (of each single R session).

There are two types of them:

Input files bring information from Mayday to R.

Output files take the information from R to Mayday.

In both cases you should choose *always delete*. The other choices are just for debugging purposes, i.e. for applicable function and Mayday developers.

- (e) In the *RPlot* box you can choose which image format should be used to return R plots (if there are any).

It is recommended to read the section about graphics devices in the R manual if there are any problems. Initially you should set this to JPG (\equiv JPEG) or PS (postscript). Note that R plots stored in PS or PDF files cannot be shown in Java frames yet.

5. Open the *Sources* tab.

Here you can find the functions to be used. Initially the list is empty.

- Use *Add...* to select *applicable functions* which are contained in *.R files. *Applicable function* is the keyword related to R functions that can be used by the R for Mayday interface. There are some requirements connected with applicable functions:
 - (a) An applicable function has a specific signature.
 - (b) It is defined in a *.R file.
 - (c) There is a corresponding description file with the same filename, but the extension *.xml* containing meta information of the applicable function, e.g. parameter descriptions. The description files are provided by the applicable functions developers and should not be changed.

Open the directory you extracted the example functions to. And select the *kmeans.R* file. Press *Open*.

- Press *Next >* to go on. A dialog shows up where the parameters for the selected functions can be set. Here, in the *kmeans* example with the



artificial dataset mentioned above, you set k (the number of clusters) to 3 and let the number of iterations be 100.

Hold the mouse cursor a short period over the parameter lines. A tooltip with a short parameter description shows up.

- Finally, press *Run*. A dialog is shown with information about the progress of the R session or the functions execution, respectively.

The *Cancel* button is used to kill the R session explicitly, i.e. do not press it. That can be used if an execution seems to never return.

When the function returns, press *Ok* to go back to Mayday. If there were any problems they will be shown at this stage of progress.

- If all works fine there are 3 new probe lists shown in the Mayday main frame.

Example execution, Vol.II

The first example was intended to get acquainted with the R interpreter plugin itself. Now we combine two function calls to test the plots:

1. Open the artificial example data set, like in the previous example.
2. Multidimensional scaling:
 - (a) Open the R interpreter plugin.
 - (b) *Add ...* and select the applicable function *Multidimensional scaling (MDS.R)*.
 - (c) Press *Next* >.
 - (d) Set the parameters to:
 - *method* = "cmdscale"
 - *k* = 3
 - *n.clust* = 3
 - (e) Run.
 - (f) A new data set has been created with 3 extra columns (experiments) and 3 probe lists defined by a clustering executed subsequently to the MDS algorithm.

[Since Mayday now has a meta information object model, this may change in the future so that k -dimensional points will be returned as MIOs.]
3. Scatter plots:
 - (a) Select the 3 colored probe lists created in the MDS run.
 - (b) Start the R interpreter plugin.
 - (c) *Add ...* and select the *Scatter Plots (2D/3D)* applicable function (*scatter-Plot.R*).



- (d) Press *Next* >.
- (e) Set the parameters to: (the 3 dimensions to plot, each a column in the master table)
 - *exp1* = "MDS_1 " (Do not forget the quotes!)
 - *exp2* = "MDS_2 "
 - *exp3* = "MDS_3 "
- (f) Run.
- (g) If you have not already installed the R-package *scatterplot3d* the execution ends with an error. The exit state is 1, i.e. the functions application was not successful.
To solve this problem:
 - Open R with your favorite R GUI or the R terminal.
 - Install the package *scatterplot3d*, e.g. type `install.packages("scatterplot3d")` and follow the instructions.
- (h) Back in Mayday, repeat steps 3a to 3f.
- (i) If all works fine, 3 plots show up, each from another perspective.

Solving errors

Errors might occur on three stages:

Pre-execution of R Errors occurring before and during the start-up of the R interpreter lead to problems with the R installation.

R execution Errors occurring during the execution of the R interpreter are reported by R itself. In most cases the R interpreter will halt on errors. There is no result and the execution has to be repeated after examining those problems. Here are some of the possible causes:

A specific library/package could not be found. Please open R (either an a terminal or your favorite GUI) and install the missing package.

Each provider of applicable functions should make clear which library his/her function makes use of.

Parameter checks fail. Each writer of applicable functions is responsible for the validity of the function arguments. Thus parameter checks are the matter of the writer of applicable functions.

Make sure you read the instructions of the provider of the applicable function. Have you selected all necessary parameter values? Do these values have the proper format? Note that each character string you want to be interpreted as character sequence (and not as symbol) has to be quoted with single (') or double (") quotes! Check whether you have used R executable code as parameter value and then if this code is correct in the matter of the R language.



Other errors occurring during the run of an applicable function. Please contact the provider of this applicable functions for error information.

Warnings. Also warnings may occur in an R session. Warnings are used if a minor problem has drawn out that does not crash the implemented algorithm. But you should be careful with warnings because the results might disagree your expects.

Result parsing These errors occur, if the output of the R session is incomplete or improper. The first case can occur as an artifact of preceding errors in the execution. The latter occurs if the result structure returned by the applicable function is improper in a matter the output method cannot recognize.

Known issues

PNG plotting problems In some cases the R plots in PNG format cannot be read by Java throwing the exception 'LUT has improper length! File: Rplot001.png'. This problem could not yet be solved. The plots can be found in the working directory and opened with another image viewer, or you choose the JPEG format before running the applicable function.



4.4 Meta Information Objects in *Mayday*

How to Use Meta Information Objects in *Mayday*

Nils Gehlenborg

Why Meta Information Objects?

This How-To describes only fundamental features of *Mayday*'s meta information facilities. This is neither an in-depth introduction nor a tutorial. Be aware of this. Also keep in mind that *Mayday*'s meta information features are continuously being extended and improved, thus the information in this document might not always reflect the current state of the software.

Mayday's core data structures were designed to handle gene expression data efficiently. However, many methods of microarray data analysis require additional information about the probes on the microarray. This information can be database identifiers, sequence information, results of calculations such as statistical tests and so on. In order to be able to store and process such information on a high level of abstraction *Mayday* has meta information facilities that are described in this how-to.

Plug-ins to Handle Meta Information

At the moment there exist several plug-ins for *Mayday* that handle meta information:

- Import
 - Import: Integers (Plug-in Package Data Import)
 - Import: Doubles (Plug-in Package Data Import)
 - Import Strings (Plug-in Package Data Import)
 - Import: Complex Numbers (Plug-in Package Data Import)
 - Import: String Lists (Plug-in Package Data Import)
- Creation
 - Statistics: Binning (Plug-in Package Statistics)
 - Statistics: Probe Mean (Plug-in Package Statistics)
 - Statistics: Probe Standard Deviation (Plug-in Package Statistics)
 - Statistics: Probe Variance (Plug-in Package Statistics)
 - Statistics: Pearson Correlation Coefficient (Plug-in Package Statistics)



- Scoring: Direct Mapping (Plug-in Package Relevance Functions)
- Scoring: Exponential Mapping (Plug-in Package Relevance Functions)
- Scoring: Linear Mapping (Plug-in Package Relevance Functions)
- Scoring: Invert (Plug-in Package Relevance Functions)

- Visualization
 - Enhanced Heatmap (Plug-in Package Visualization)

- Utilities
 - Probe Lists: Nominal Data To Probe Lists (Plug-in Package Statistics)

Most of these plug-ins will be briefly described below after some general discussion about meta information in *Mayday*.

Downloading and Installing the Required Software

Mayday offers functionality through plug-ins that are developed independently of the core software. Thus it is necessary that you download and install at least some of the plug-ins listed above before you can use meta information in *Mayday*. Visit <http://www.zbit.uni-tuebingen.de/pas/mayday/download/index.html> and download the latest snapshot or release version of the plug-in packages you are interested in. Snapshots are usually uploaded to the website when new features were added and when bugs were fixed, they might not have been tested thoroughly.

Download the jar-archives into the plug-in directory of your *Mayday* installation. If you were running *Mayday* while you installed the plug-ins close your session and restart *Mayday* so the new plug-ins will be loaded. Please note that you should have only one version of each plug-in in your plug-in directory. Only one version of the plug-in will be loaded and with the current naming scheme this will be the oldest version.

Some Important Background Information

Meta information in *Mayday* is stored in so-called meta information objects, MIOs for short. MIOs can be attached to either probes or probe lists. Since currently only MIOs attached to probes are of interest we will mainly focus on this case in this document.

MIOs come in a variety of different flavors. There are MIOs that represent strings, integers, doubles and more complex objects such as lists of strings or complex numbers (in the mathematical sense). Further MIOs are classified into categories such as numerical (e.g. integer, double), categorical (e.g. integer, string), comparable



(e.g. integer, double, string) or compound MIOs (e.g. complex number). These categories are sometimes used to determine which MIOs can be processed by which method. There are a couple of methods that can process MIOs of several types as long as they fall into a certain category. An example is the sorting of heatmap columns, which requires comparable MIOs.

Whenever MIOs are created or imported they are assigned to a new meta information object group (MIO group). An MIO belongs to exactly one MIO group, which in turn belongs to exactly one data set. Each MIO group contains MIOs of exactly one type (integer, double, etc.).

MIO groups give you access to a semantically related set of MIOs. All operations on MIOs require you to select an MIO group as input. This makes it more convenient to work with meta information because otherwise you would have to define which MIO to use for every single probe or probe list. For instance, if you compute p -values from several test statistics for all probes of a probe list and you want to use the p -value of a particular test later on for visualization in an enhanced heatmap you just select the MIO group that was created for the p -values of that test statistic instead of defining which p -value to use for every probe individually.

As probes and probe lists are parts of a data set MIOs are also considered to be part of a data set. They are managed by a Meta Information Object Manager that is described in the next section.

Meta Information Object Manager

Open the MIO Manager (short for Meta Information Object Manager) of the selected data set via “Data Set → Meta Information Objects ...”. Currently the MIOManager allows you to browse meta information based on probe lists and probes⁴.

Select a probe list in from the list on the very left, then a probe list from the middle list and eventually the list on the very right will display all MIOs associated with the selected probe.

Double-clicking on a entry in either of those three lists will bring up a dialog with more information about either the selected probe list, probe or MIO. Those dialogs are so-called “property dialogs”.

You can edit the name of a MIO group by selecting an MIO from that group and double-clicking it to get the properties dialog. There click the “More ...” button under description to get the MIO group property dialog. Change the description and click on “Ok”. The change will be propagated instantly.

The MIO Manager is not a modal dialog, i.e. you can leave the window opened and progress with your work in *Mayday*. If you load additional meta information the MIO Manager will be updated in the background. The same holds for changes to probe lists and probes.

⁴In the future we will add an option that allows you to browse meta information based on type or category and MIO group.



MIO and MIO Group Properties

You can access MIO and MIO Group properties using the respective property dialogs. MIOs obtain some of their properties directly from their “parent” MIO group, this is for semantic reasons and also to save memory. Here is a quick overview of the properties you can edit or read:

MIO properties

- *Description* (read only, from group) This is basically the name of the MIO group and used to identify MIOs.
- *Creation Date* (read only, from group) Date of creation. Later this might be used to find and remove old, obsolete meta information. Currently it’s useless.
- *Write Protected* (read-only, from group) If this is set you won’t be able to delete⁵ the MIO. This makes sense if the creation of the MIO is costly or if the MIO is fundamental to all further work.
- *Compound* (read only, from group) This flag indicates that the MIO is a compound of several MIOs. One example is the ComplexNumberMIO which is a compound of two DoubleMIOs, one for the imaginary part and one for the real part. Such compounds can be decomposed into their basic building blocks and operations can be performed on those building blocks.
- *Embedded* (read only, from group) This flag indicates that the MIO is part of a compound MIO and represents a building block.
- *Invalid* (read only, from group) This flag indicates that the MIO has been invalidated for some reason, e.g. if the MIO’s value was based on the content of a probe list and the probe list content has changed.
- *Object* (read only) The object the MIOs is associated with. Currently this is either a probe list or a probe.
- *Value* (read only) The actual value of the MIO, basically the meta information itself. If you are talking casually about MIOs this is what you mean.

MIO Group properties

- *Description* This is basically the name of the MIO group and used to identify MIOs.
- *Creation Date* (read only) Date of creation. Later this might be used to find and remove old, obsolete meta information. Currently it’s useless.

⁵Currently meta information can only be created⁵ but not deleted. In the future this will change.



- *Write Protected* If this is set you won't be able to delete⁶ the MIOs of the group. This makes sense if the creation of the MIOs is costly or if the MIOs are fundamental to all further work.
- *Compound* (read only) This flag indicates that the MIOs of this group are a compound of several MIOs.
- *Embedded* (read only) This flag indicates that the MIOs of this group are part of a compound MIO and represent a building block.
- *Invalid* (read only) This flag indicates that the MIO group has been invalidated for some reason, e.g. if the MIOs' values were based on the content of a probe list and the probe list content has changed.

Meta Information Plug-ins Explained

This section contains a quick overview of the functionality offered by the plug-ins that handle meta information in *Mayday*.

Meta Information Import

The import plug-ins listed above all read meta information from a flat file and assign meta information to probes of the data set. The plug-ins don't work on probe lists!

The first column of the file contains probe identifiers and the second column (separated from the first by a tab character) contains the meta information. The complex number and the string list import plug-ins read more than one column. Please see the description of the plug-ins in the Plug-in Manager ("File →Plugins ...") for details.

If the first column contains probe identifiers that are not contained in the data set those probes will be ignored. Some of the probes in the data set are not assigned meta information in the file they won't be assigned an MIO. The plug-ins don't fail in such cases.

Statistical Methods

There are several plug-ins that compute descriptive statistics (mean, variance, standard deviation) on the expression values of probes. The input for those plug-ins is a probe list. They will attach the corresponding values to the probes in that probe list and create a new MIO group.

The Pearson correlation coefficient is computed with respect to the expression values of a probe that you chose before you run the plug-in. The same reference probe will be used for all probes of the probe list. Otherwise the plug-in works like the ones describe before.

⁶Currently meta information can only be created but not deleted. In the future this will change.



Relevance Scoring Functions and the Enhanced Heatmap

Please note: If you are not familiar with the enhanced heatmap read the section on it in the “Getting Started With Visualization in *Mayday*” How-To first. This will certainly help you to understand the information presented here. It will also help to have a look at the paper by Nils Gehlenborg, Janko Dietzsch and Kay Nieselt, A Framework for Visualization of Microarray Data and Integrated Meta Information, *Information Visualization* (2005) 4, 164 - 175 (doi:10.1057/palgrave.ivs.9500094).

Currently there are three plug-ins that can be used to score the relevance of meta information, i.e. to create scoring MIOs: direct mapping, linear mapping and exponential mapping. The input for those plug-ins is a group of numerical MIOs that will be mapped onto a scale from 0 to 1. The linear and the exponential mapping plug-in map the largest input value on 1 and the smallest on 0. All other values will be mapped using either a linear or an exponential function. For the exponential mapping you can define the steepness of the function: the larger that value the steeper the function the less values will be close to 1.

The direct mapping plug-in basically converts the type of the MIOs to scoring MIOs. Only values larger than 1 will be set to 1 and values smaller than 0 will be set to zero.

The output of all three plug-ins can be also used as input for those plug-ins since scoring MIOs belong to the category of numerical MIOs.

The “invert” plug-in can be used to invert the relevance scoring information by computing $score' = 1 - score$. This is necessary because per definition scores close to 1 indicate high relevance and score close to 0 indicate low or no relevance. However, for instance in the case of *p*-values the meaning of 0 and 1 is exactly opposite. Relevance scoring MIOs can be used in the enhanced heatmap visualization to guide you during data exploration and hypothesis generation. How to this is done is described in the “Getting Started With Visualization in *Mayday*” How-To.

Others

If you want to create probe lists from a group of categorical MIOs, i.e. assign all probes with the same value to a new probe list, you should try the “Probe Lists: Nominal Data To Probe Lists” plug-in.

In order to create categorical MIOs from numerical MIOs you may use the “binning” plug-in that takes a MIO group and a number *x* of bins as input. It splits the range of the numerical MIOs into *x* equidistant bins, determines which probes are assigned to which bin according to the selected MIO group and finally creates a new MIO that contains the bin information for each probe.



Known Issues

Storing Meta Information

Currently it is not possible to store meta information objects created during your work in *Mayday*. You will have to ensure that you can reproduce the information if you want to use it across several sessions.

Deleting Meta Information

In the future it will be possible to remove meta information objects if they are no longer needed. Currently this is not implemented.

Creating Large Meta Information Groups

If you are importing or creating meta information objects for a large set of probes the process might take a little longer. But this is only during creation of the meta information group and is not an issue when you are accessing meta information.



4.5 Databases in *Mayday*

How to setup and use *MayDB*

Stephan Symons

Preparations

Short Version

To set up PostgreSQL for usage with *MayDB*, do the following:

- Install and run PostgreSQL
- Set up TCP/IP connections

For the long version see the rest of this section. The steps to set up the database are described using command line programs, but they can also be executed using the graphical front end `pgAdmin` which is available for linux, Mac OS and windows.

Note: If you have already *MayDB* up and running on your system, skip all but the last step in the next section.

Installing PostgreSQL

The first and most important step for preparing PostgreSQL is of course installing it on your machine. PostgreSQL sources and binaries for some linux distributions and windows are available from www.postgresql.org/download. It is also included in many linux distributions.

For Mac OS see <http://www.entropy.ch/software/macosx/postgresql/> for instructions.

As instructions for installing (let alone compiling) PostgreSQL would go too far here, please refer to the documentation (enclosed to PostgreSQL or at <http://www.postgresql.org/docs/> for details.

Make sure that the PostgreSQL backend application (linux: `postmaster`; windows: `postgres.exe`) is running.

TCP-IP Connections

The standard way to use databases with Java is the JDBC interface. In order to use PostgreSQL with JDBC, the PostgreSQL backend must accept TCP/IP connections. If this is not the case on your machine, you need to do the following:

- The main configuration file of PostgreSQL is `postgres.conf`. This must contain the line



```
tcpip_socket = true
```

Also check the other connection settings for compatibility with the used system.

- The file `pg_hba.conf` is used to set up the privileges of the users which use the database.

Add the following (or a similar) line to this file:

```
host      all          all          127.0.0.1      \\  
          255.255.255.255  trust
```

This line allows all local users using IPV4 (for IPV6 see file and documentation) to connect to PostgreSQL without password. This is very convenient (→ Spooling) but somewhat dangerous. Change `trust` to something other to use identification.

For details, see the (brief) help in the file itself and the PostgreSQL documentation.

- Also remember to configure your firewall to allow all necessary programs to access the net.

MayDB Setup

Now that you have PostgreSQL up and running, the next step is preparing the *MayDB* users and tables.

Short Version

1. `createdb maydb`
2. Create a group `maydb`.
3. Add all users which may use *MayDB* to this group
4. use all files in `scripts.zip` to create schemas and tables.

Database Creation

Execute this command as user `postgres` (which is the system user owning the PostgreSQL configuration files and processes):

```
$> createdb maydb
```

to create the database.



Users and Groups

- Create new users by executing `createuser <user> as user postgres`.
- Connect to PostgreSQL as user postgres:
`$> psql maydb`
Create a new group: Execute
`CREATE GROUP maydb;`
- Add all users that should be able to use *MayDB* to the group `maydb`:
`ALTER GROUP maydb ADD USER <user>;`

Schema and Tables

Finally the *MayDB* schema and tables must be created. Connect to PostgreSQL as user postgres:

```
$> psql maydb
```

Then:

- Create a new schema `maydb` and grant privileges to the users.
`CREATE SCHEMA maydb;`
`GRANT ALL ON SCHEMA maydb TO GROUP maydb;`
- Run `\i maydb.sql` to create the tables and sequences.
- Run `\i grant.sql` to grant the needed privileges to the users.

The scripts `maydb.sql` and `grant.sql` are in the `scripts.zip` file.

Congratulations. You have successfully installed PostgreSQL for usage with *MayDB*.

Using *MayDB*

To use *MayDB*, you need the *Mayday* Database Plugin, available from <http://www.zbit.uni-tuebingen.de/pas/mayday/download/index.html>. Make sure the `.jar` file is in the *Mayday* plugin directory or one of its subdirectories.

The *MayDB* Wizard

All *MayDB* functions implemented so far are accessible from the “MayDB Wizard. To run the *MayDB* Wizard, start Mayday, open the DataSet menu, the Data Import sub menu and select *MayDB* Wizard.

A new window opens offering five options:

- Store ImaGene Data in *MayDB* (see section 4.5)
- Store Affymetrix Data in *MayDB* (see section 4.5)



- Store expression matrices in *MayDB* (see section 4.5)
- Fetch Data from *MayDB* (see section 4.5)
- Inspect Database content (see section 4.5)

Select an option and click on ok and proceed to the corresponding section.

ImaGene data and expression matrices

In general, submitting data to *MayDB* consists of the following steps:

1. **Choose Files:** In the “Choose Files dialog, click on “Add and select all files to import. All files you selected are shown in the list. To remove a file from the list, select the file and click on the “Remove button. To remove all files from the list, click on “Clear. When you are pleased with your selection, click on “Ok.
2. **Connect to *MayDB*:** In the “Connect to MayDB dialog, enter the database connection details: host name, database name, user name, password. Select the JDBC drivers by clicking the Browse button. Leave all other settings at their preselected default values.
3. **Experiment Dialog:** Storing data in *MayDB*, the first step is creating a new experiment. An experiment is a collection of hybridizations on the same microarray design. Click on new experiment. A new “Experiment Settings dialog opens. Enter all information about your experiment in the appropriate text fields and click on ok. If you have already stored data for this experiment, select it from the list on the left and click on “Ok.
4. **Array Design Dialog:** To create a new Array Design, click on the “New button, and enter all information about the array design in the text fields of the new Array Design Settings. Then, click on “Ok.

Each Array Design must be stored only once, and can then be used on multiple experiments. Select the appropriate array design from the list on the left of the Array Design Settings dialog and click on “Ok.

5. **Hybridization Dialog:** In this dialog, you can set the name, color channel, and protocol of each hybridization. You can also group hybridizations to BioAssays. To group two hybridizations to one bio assay, set the number of the Associated Bio Assay to the same number for both of them (and no other hybridization). ImaGene files are supposed to be sorted by name, and the green channel files are supposed to have a name “smaller than the red channel. Based on this, the channel and BioAssay are predicted.
6. **BioAssay/Array Dialog:** Finally, you may enter information about the BioAssays and the Arrays used. Click on “Ok when you are done.

After you completed the last dialog, all information necessary to store your data is collected, and your data is stored in *MayDB*.



Affymetrix Data

Storing Affymetrix GeneChip data (many copyright marks here) in *Mayday* requires two steps:

1. Array Design:

Before you submit an Affymetrix chip design to *MayDB*, you might want to check if it is already there. To do so, click on the Check button. You will be asked for a database connection (see above). A list of available array designs appears. If the chip design necessary for your chips is already stored, proceed to step two.

To submit an Affymetrix Chip design to *MayDB*, select the "Store array design (.cdf file) option in the "Affymetrix Options dialog. After you selected the .cdf file, you will be prompted for database connection settings (as described in 4.5, step 2). No further user work is required here.

2. .cel files:

After you have submitted the array design to *MayDB*, you can store matching .cel file data. Select the "Store CEL file data option in the "Affymetrix Options dialog. Storing .cel file data requires the same steps as described in section 4.5.

Fetching Data from *MayDB*

The steps necessary for fetching data from *MayDB* to *Mayday* are the following:

1. Connect to Database:

Enter your database settings in the "Connect to MayDB dialog (as described in 4.5, step 2).

2. Select Experiments:

In the following dialog, select an experiment from the list. Click on ok to proceed fetching the experiment.

3. Select Hybridizations:

Select the hybridizations you want to fetch. If you want to fetch all hybridizations, simply click on ok. If you have a lot of hybridizations, you might want to fetch just some of them. Click on the "Filter button to open the filtering dialog. You can filter by name, channel, comment and/or protocol. Activate each filter by checking the box next to it and enter a string the probe's name (comment/protocol) must contain.

4. Filter mode: Here you can select if, and how to filter probes. You have four options:

- (a) No filtering.



- (b) Filter by name. You will be prompted for a value that each probe must match.
 - (c) Filter by value. Use this function to filter out flagged or outlying probes.
 - (d) Manual filtering. Select all probes to be fetched from a list.
5. Import Mode: You can choose from three import modes:
- (a) Simple Import: Import one measurement. (Ideal for Affymetrix data or simple expression matrices)
 - (b) Default Import: Import the quotient or difference from two measurements (good for ImaGene data)
 - (c) Ratio Import: Import the quotient or difference of measurements from two hybridizations. You will be prompted to arrange the hybridizations (pre-arranged by bio assays if possible).

In each case, you will be asked for what measurement(s) to import after you made your choice.

Thereafter the data is fetched from *MayDB* and a new dataset in *Mayday* is created. You can also invoke this function by clicking on DataSet, Import, Database Fetch in the *Mayday* main window.

Inspect Database Content

This function allows you to manipulate the *MayDB* content to some extent. You can run this plugin also by clicking Dataset, Import, MayDBManager.

Running this plugin, you will first be asked for a database connection (see 4.5, step 2). When connected a dialog with all experiments stored in *MayDB* shows up. There, you can edit the properties of each experiment by clicking the "Edit..." button (set the new properties in the "Edit Experiment dialog), delete an experiment using the "Delete button and browse the Hybridizations and BioAssays of this Experiment by clicking the "Browse..." button.

After clicking the "Browse..." button, you can select one from the available objects and a new dialog, similar to the one described above appears.



How to use the *Mayday* Project Database

Stephan Symons

What is ProjectDB?

The standard way of moving gene expression data into *Mayday* is opening a data set (and possibly some probe lists) stored in external files or fetching a data set from *MayDB*. A general advantage of databases over file systems is that they make data more manageable by enforcing data consistence and order and combining data in the way necessary for the application.

In this specific case, ProjectDB is used to organize your *Mayday* data sets in a way that it saves the current state of an open dataset in *Mayday*. Such a data set is part of a project, and this project has several states, which may or may not depend on other ones. Therefore, ProjectDB allows the user to organize data sets in a hierarchical way - a tree.

Some definitions:

- *Project*: A set of data sets for a specific study, or a collection of data used for the same purpose. Projects are the topmost organizational entity in *Mayday* and contain zero or more states.
- *State*: A collection of datasets in one project, which have undergone a specific treatment, or are related in an other way. States may contain other states or data sets.
- *Data set*: The current file open in *Mayday*, including all annotation, meta information and probe lists.

For example: For some microarray study, you open a data set, and perform some operations on it. Save the data set to ProjectDB in a state called "Prepared Data. Afterwards, you can reopen the data set and perform a clustering analysis on it and save this dataset in a child state of "Prepared Data, possibly called "Clustered (k-means).

Preparations

Short Version

To set up PostgreSQL for usage with ProjectDB, do the following:

- Install and run PostgreSQL
- Set up TCP/IP connections



For the long version see the rest of this section. The steps to set up the database are described using command line programs, but they can also be executed using the graphical front end `pgAdmin` which is available for linux, Mac OS and windows.

Note: If you have already *MayDB* up and running on your system, skip all but the last step in the next section.

Installing PostgreSQL

The first and most important step for preparing PostgreSQL is of course installing it on your machine. PostgreSQL sources and binaries for some linux distributions and windows are available from www.postgresql.org/download. It is also included in many linux distributions.

For Mac OS see <http://www.entropy.ch/software/macosx/postgresql/> for instructions.

As instructions for installing (let alone compiling) PostgreSQL would go too far here, please refer to the documentation (enclosed to PostgreSQL or at <http://www.postgresql.org/docs/> for details.

Make sure that the PostgreSQL backend application (linux: `postmaster`; windows: `postgres.exe`) is running.

TCP-IP Connections

The standard way to use databases with Java is the JDBC interface. In order to use PostgreSQL with JDBC, the PostgreSQL backend must accept TCP/IP connections. If this is not the case on your machine, you need to do the following:

- The main configuration file of PostgreSQL is `postgres.conf`. This must contain the line

```
tcpip_socket = true
```

Also check the other connection settings for compatibility with the used system.

- The file `pg_hba.conf` is used to set up the privileges of the users which use the database.

Add the following (or a similar) line to this file:

```
host    all            all            127.0.0.1      \\  
        255.255.255.255  trust
```

This line allows all local users using IPV4 (for IPV6 see file and documentation) to connect to PostgreSQL without password. This is very convenient (→ Spooling) but somewhat dangerous. Change `trust` to something other



to use identification.

For details, see the (brief) help in the file itself and the PostgreSQL documentation.

- Also remember to configure your firewall to allow all necessary programs to access the net.

ProjectDB Setup

Now that you have PostgreSQL up and running, the next step is preparing the ProjectDB users and tables.

Short Version

1. `createdb project`
2. Create a group `maydb`.
3. Add all users which may use *Mayday* databases to this group
4. use all the `project2.sql` file in `scripts.zip` to create schemas and tables.

Database Creation

Execute this command as user `postgres` (which is the system user owning the PostgreSQL configuration files and processes):

```
$> createdb project
```

to create the database.

Users and Groups

- Create new users by executing `createuser <user>` as user `postgres`.
- Connect to PostgreSQL as user `postgres`:

```
$> psql project
```

Create a new group: Execute

```
CREATE GROUP maydb;
```
- Add all users that should be able to use `maydb` to the group `maydb`:

```
ALTER GROUP maydb ADD USER <user>;
```

Schema and Tables

Finally the ProjectDB schema and tables must be created. Connect to PostgreSQL as user `postgres`:

```
$> psql project
```

Then:



- Run `\i project2.sql` to create the tables and grant the needed privileges to the users.

The script `project2.sql` is in the `scripts.zip` file, which is available on the *Mayday* download page (<http://www.zbit.uni-tuebingen.de/pas/mayday/download/index.html>) Congratulations. You have successfully installed PostgreSQL for usage with ProjectDB.

How to...

- Start the ProjectDB plugin
To use ProjectDB, you need the *Data Import Plugin*, available from <http://www.zbit.uni-tuebingen.de/pas/mayday/download/index.html>. Make sure the `.jar` file is in the *Mayday* plugin directory or one of its subdirectories.
Run the ProjectDB plugin from *Mayday* by clicking on Data Set, Data Import, ProjectDB. You will be asked for a database connection. After that, the main window of ProjectDB appears, the “Project Browser
- What do the symbols in the “Project Browser mean?
A project is represented by a red folder icon, a state by a yellow folder icon, and a dataset is represented by an iconified microarray image.
- Create a Project
In the “Project Browser, right-click on the root symbol, and choose “New Project from the menu. You will be prompted for the name and annotation of the project in a new window.
Projects are depicted by a red folder symbol.
- Create a State
Right-click on a project or a state symbol, and choose “New State from the menu. You will be prompted for the name and annotation of the state in a new window.
States are symbolized by a yellow folder symbol.
- Store a data set / Store all open data sets
To store a dataset currently opened in *Mayday*, select a state, right-click on the state icon to open the context menu, and click on “Store... and select the data set to store.
To store all open data sets, simply right-click on the state icon and click on “Store all Datasets here
A dataset is represented by a microarray image-like symbol.



- Edit the properties of a Project/State/Data set
Right-click on the symbol of the object and choose "Edit..." from the menu.
You can set the values in a new window.
- Delete a Project/State/Data set
Right-click on the symbol of the object and choose "Delete from the context menu."
- Retrieve a data set
Select a data set in the "Project Browser, and click on the "Retrieve button, or use the context menu and click on "Retrieve
- Retrieve all data sets in a state
Select a state in the "Project Browser, and click on the "Retrieve State

Known Issues

Please note the following issues with this software:

- ProjectDB is still under development. In the future, ProjectDB will be extended to cover the Meta Information Object concept of *Mayday*. Currently, only some basic MIOs are supported.
- Only (Probe-)MIOs in the *Mayday* core can be retrieved from ProjectDB, but all (Probe-)MIOs can be stored.
- MIOs attached to probe lists can not be stored or retrieved
- The name changes when i have a better one for it.

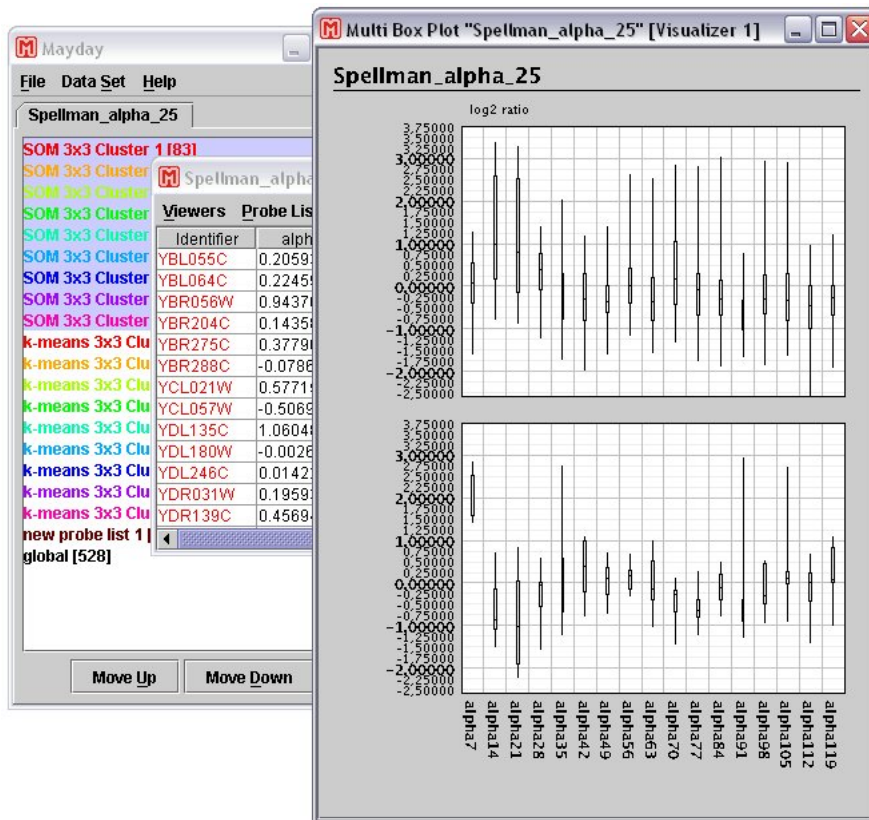


Figure 18: Multi Box Plot