Tutorial for Mayday 2.14

Alexander Stoppel
2015

Contents

1 Introduction

An essential research field in biology and bioinformatics is the understanding of cellular gene regulation. Through the development of technologies like Microarrays as well as RNA-seq, researchers are able to gain insight into the transcriptome of cells. The analysis of the large amount of data that are produced by these techniques is thereby still a challenging task for bioinformaticians. With Mayday [BSN10], a platform has been published, that combines the most useful methods with a user-friendly interface to facilitate the analyses on transcriptomic data.

This tutorial covers the most common analyses for microarray data using Mayday. Furthermore, a detailed explanation of Mayday's features is given as well as a demonstration on how to apply these. In this tutorial all descriptions of used methods are condensed to its essential, which are necessary to understand its meaning for the analysis. For more detailed information additional references are provided. This tutorial is written for Mayday version 2.14.

1.1 Steps of analyzing microarray data

In general, microarray data analysis consists of several sequential steps:

- Normalization: after getting measures from a microarray chip, technically induced variation need to be minimized to make sure that only biological variations are present in the data.
- Extraction of information: significant information such as genes, which seem to be differentially regulated or are expected to show a response to altered experiment conditions are to be found and extracted.
- Data visualization: visualization of data and results of different analyses is essential and helpful.

We will use these steps as guideline for this tutorial.

1.2 About the example data

For a better understanding, the application of all described Mayday features is demonstrated with example data. We will analyze gene expression data from a microarray time series experiment of *Streptomyces coelicolor* [NBH+10]. The *Streptomyces* bacterium is used for the industrial production of antibiotics like *Streptomycin* and *Avermyctin* [dLPdSM+12]. Moreover, *S. coelicolor* genome is i.a. related to *Mycobacterium tuberculosis* and *Corynebacterium diphtheriae* genomes [JIC] so it can be used to study bacteria causing these deceases. Therefore, *S. coelicolor* is commonly used as an example organism for genetic studies.

Under optimal environmental conditions, that means especially an abundant supply of nutrient, bacteria like *S. coelicolor* proliferate exponentially. When conditions become worse, proliferation stagnates and the bacterium switches from primary to secondary metabolism, what results in alteration of gene expression.

The experiment to measure gene expression during growth was conducted as follows: *S. coelicolor* bacteria were cultivated in a fermenter for 60 hours under phosphate limiting conditions. After 35 hours the nutrient phosphate was depleted, which lead to a stop of exponential proliferation. The example data has been produced using a custom designed *Affymetrix* GeneChip® microarray. The experiments had been conducted every hour from 20 up to 44 hours and every two hours afterwards.

In this tutorial we will analyze the effects of starvation of nutrition that can be observed on the gene expression level and try to identify genes which are possibly involved in this metabolic switch. Our example dataset F199 contains expression values of 22779 *S. coelicolor* transcripts measured over the time points as described above. All data used in this tutorial is available on our homepage.

2 Mayday

Mayday is a workbench for analysis, visualization and storage of microarray data. It combines several steps of microarray data analysis in one single program. Statistical methods as well as graphical presentation is supported.

Mayday is available on our website `http://it.inf.uni-tuebingen.de`. Written in Java programming language Mayday can be used on all platforms supporting the Java runtime environment 1.7. Developed as modularized open source project, several plugins are available that can be adapted and extended.

Mayday is an ongoing project of the *Integrative Transcriptomics* group[1] at the *Center of Bioinformatics* Tübingen, Germany, led by Dr. Kay Nieselt.

2.1 Getting started

Mayday can be used without any installation which makes it flexible and easy to work with. We recommend the experimental webstart version, but alternatively a standalone download version is available as well. For using the webstart version, follow the download instructions on our website and run `Mayday_XG.jnlp` in your download folder. For both versions the amount of available RAM can be specified. Please make sure that the used hardware supports the selected amount of memory.

For the standalone version, extract the downloaded zip archive, navigate to the contained folder `executables` and run `Mayday_XG.sh` (Linux/Mac) and `Mayday_XG.bat` (Windows), respectively. `X` means how much RAM Mayday can use.

When Mayday starts for the first time, a new window will ask the user to set a plugin directory. Set the path to the folder `plugins` inside the extracted archive.

2.2 Data handling in Mayday

Analyzing large amounts of data, efficient and simple data handling is required. Mayday provides a clear data organization. Automatically created probe sets are neatly arranged in probe lists, manually selected subsets are represented in dynamic probe lists and meta informations can be accessed separately. Probe lists, dynamic probe lists and its associated meta informations are organized in a dataset. Several datasets can be loaded at the same time. This design concept with mainly three different types of data handling simplifies management of large scaled data.

[1] `http://it.inf.uni-tuebingen.de/`

Probe lists

In general, a dataset consists of (gene expression) data (=*probes*) measured under certain conditions (=*experiments*). Mayday represents the data in matrix format where the probes are mapped to its rows and the experiments to its columns. These matrices are held in so-called probe lists. When a new dataset is imported, Mayday by default creates a new probe list group named `Complete DataSet` including a probe list which contains all the data.

A probe list may contain the values of the whole dataset or custom subsets which can be created by the user. This can be used for work with subsets of the whole dataset and to process these without altering other subsets or the complete dataset. That is very useful e.g. if one wants to analyze or compare different subsets. For better visual perception, Mayday probe lists are colored.

The changeable probe list order in probe list window (see Figure 1 (**3**)) is used as kind of priority list with descending precedence. In some visualization windows, this probe list priority is used for ordering data visualization.

Dynamic probe lists

Dynamic probe lists are similar to the previous described probe lists with the difference, that the selection of contained genes can be adapted dynamically. With dynamic probe lists, Mayday provides intuitive user-defined data selection and handling. Every dynamic probe list is thereby defined by filtering rules which are applied on other probe lists. All genes matching these rules will be added to the respective dynamic probe list. To create a new filter, Mayday provides many data processors to select probes by names, values, probe lists, MIOs and much more.

In comparison to “normal” probe lists, filtering rules can be edited and changed, and the probe list will be updated dynamically.

Meta information objects

Mayday can add auxiliary information to the probes which are represented as so-called meta information objects (MIOs) organized into meta information groups. For example, in combination with probe lists they can be used for filtering the data using criteria provided in a meta information object. MIOs can either be created within Mayday (e.g. application of statistical methods, compare Sections 2.5 and 4.6) or imported from a file (compare Section 3.1). An example for a MIO is a list of p-values calculated on a probe list using a statistical test.

The meta information window (see Figure 1 (**2**)) features the selection, ordering and grouping of MIOs. Furthermore, the context menu provides several methods to edit, transform and visualize them. Of course, these MIOs can be exported into a separate file as well as together with the expression matrix.

2.3 Mayday graphical user interface

When Mayday is started, the main window of its GUI appears which is described in Figure 1. Mainly, it is separated into three areas:

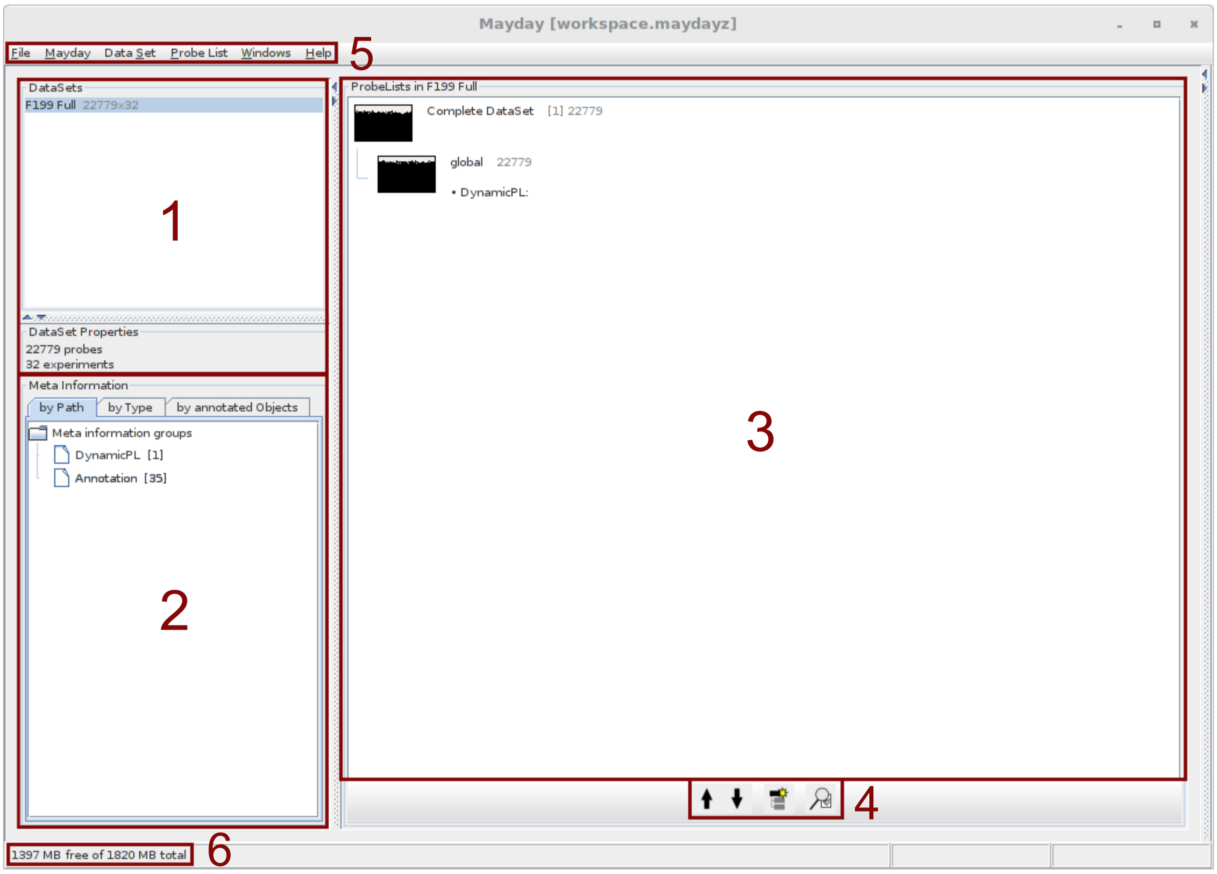
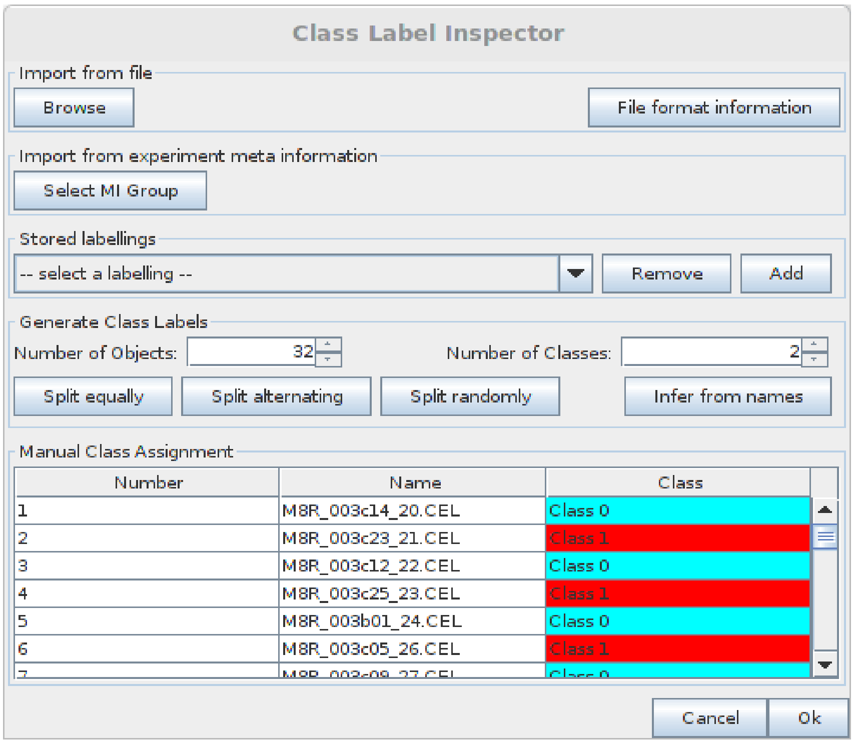


Figure 1: MAYDAY's graphical user interface main window. **1**: DataSet-management shows the loaded datasets as well as information about its size. **2**: Meta Information window displays MIOs and provides functionality for managing and ordering them. **3**: Main area displays all probe lists, additional information and preview profile plots. **4**: Buttons at the bottom rearrange and group probe lists. **5**: The menu bar provides access to further actions. **6**: Status bar shows free RAM.

1. The DataSets window shows the loaded datasets and its size (that means the number of probes and experiments).
2. The meta information group window contains a list of all MIOs which belong to the selected dataset.
3. In the main window probe lists are displayed.
4. The buttons below are for rearranging and ordering them in groups.
5. The menu bar provides access to almost the whole functionality of MAYDAY, which is described later.
6. The status bar indicates, how much RAM is currently available.

2.4 Plugins

Advanced users can expand MAYDAY's functionality by adding further plugins. The advantage is, that one can extend the functionality of MAYDAY individually by writing own plugins. Most of them are only available in the experimental version. Plugins are only usable when path to `plugins` folder is set (compare Section 2.1).



Number of Objects: 32

Number of Classes: 2

Number	Name	Class
1	M8R_003c14_20.CEL	Class 0
2	M8R_003c23_21.CEL	Class 1
3	M8R_003c12_22.CEL	Class 0
4	M8R_003c25_23.CEL	Class 1
5	M8R_003b01_24.CEL	Class 0
6	M8R_003c05_26.CEL	Class 1
7	M8R_003c09_27.CEL	Class 0

Figure 2: `The class label inspector` splits groups of experiments in definded classes.

2.5 Demonstration of the basics based on random data

We will briefly demonstrate the main functionality of probe lists and MIOs by an example. `Data Set` → `Further import options` → `Random Probes` creates random probes of an arbitrary size. In MAYDAY's main window the created probe list with some additional information and a preview of the data appears. Double-click to enlarge the preview. A double-click on the probe list itself enables renaming the probe list and changing its color.

Biological researchers are often interested in the fold-change. To demonstrate how to add a meta information object, we will calculate the fold-change as follows:

- Right-click at the newly created probe list and navigate to `Statistics` → `Derived Statistics` → `Fold-change`.
- A new window will appear were two classes can be defined, from which the fold-change should be calculated. Figure 2 shows the `Class label inspector` window where experiments can be grouped into classes.
- After confirming with `OK`, at the meta information window (see Figure 1 (**2**)), a new folder `(Probe Statistic)` containing the fold-changes as MIO will appear.

This MIO can be used to create a new dynamic probe list containing the probes with the highest fold-change.

- Again, right-click on the probe list and select `Create` → `Dynamic ProbeList`.
- One can rename the new dynamic probe list, for example to `Highest fold-change` and select a color.
- Click on `Add Rule` and select `Meta information value` as `data processor`.

- A new window will open. There, select previously created MIO `FC between Class 0 and Class 1` with the fold-change and proceed.

- The second `data processor` should be `Compare a number (class java.lang.Double)`. To set a threshold, select `Value should be` >= in the drop-down menus and set a threshold, for example 1.0.

- Press `OK` and a new dynamic probe list containing all probes with an fold-change ≥ 1 will be created.

3 Data import

Various file formats are supported in Mayday. Many files can simply be imported for further processing, for others, that are more complex, an import plugin called Mayday SeaSight [BN11] is available. Furthermore, the whole workspace can be saved and loaded again. The following sections describe the appropriate import options for the respective data formats.

3.1 Import from file

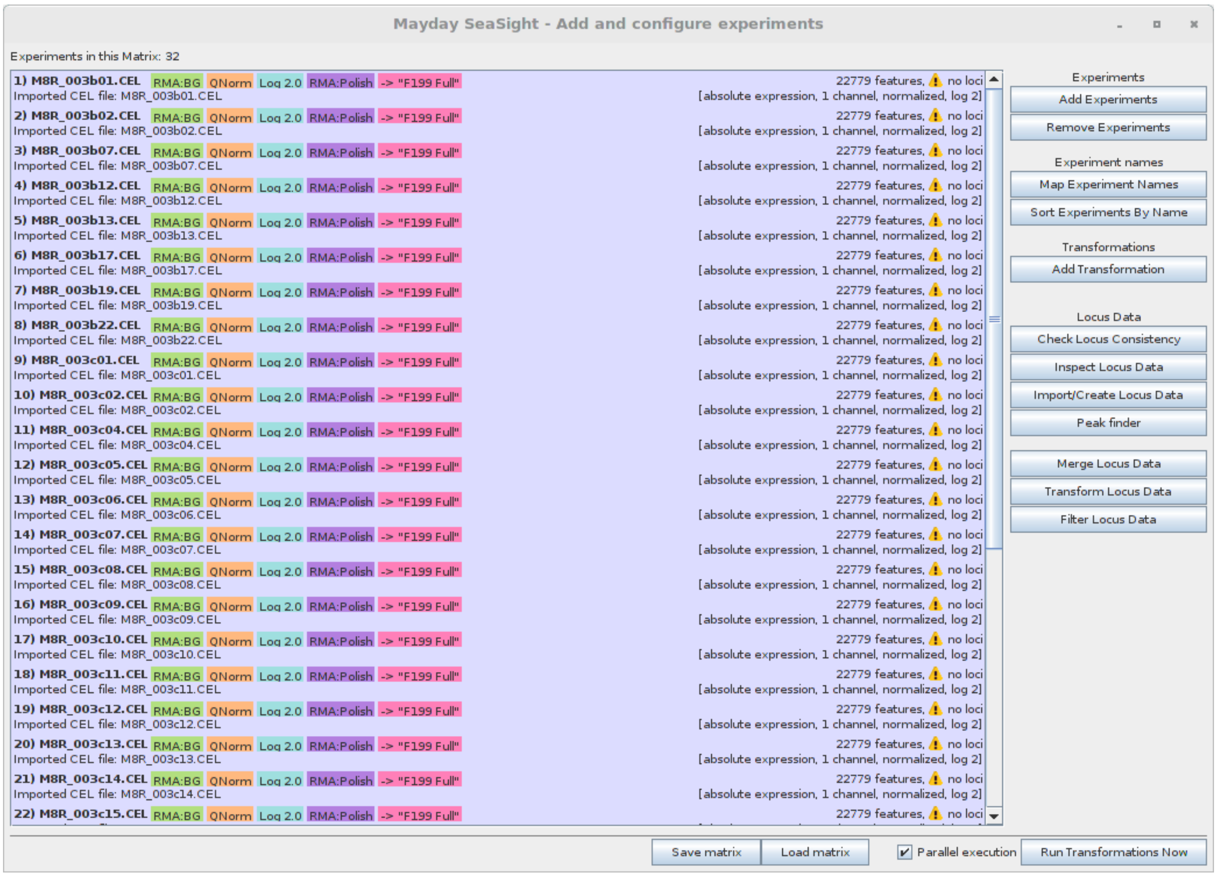
Many files, especially simple text files or `.csv` tables are directly importable into Mayday. Such files can be loaded via `Data Set` → `Import from file` in menu bar (see Figure 1 (**5**)). Mayday is able to import snapshots, whole datasets, probe lists and meta informations from various file formats. To import a dataset or a probe list, select `Import from file` in `Data Sets` menu and `Probe List` menu, respectively.

Meta information from a `.csv` file, that are associated with datasets, experiments or probes, can be imported as well. Right-click on the Meta Information window (see Figure 1 (**2**) to open the context menu where the import of several kinds of meta information such as `Data`, `Experiment` or `Probe Information` and `Locus Data` is provided.

3.2 Mayday SeaSight

Microarrays as well as RNA-seq are the two important technologies, when dealing with transcriptomics data. Generally, data from different technologies are difficult to combine. This also applies for microarray and RNA-seq data. To overcome this problem, the plugin Mayday SeaSight offers a common framework for transcriptomics data pre-processing. It features importers for many file formats as well as several pre-processing methods, such as taking the logarithm, background-correction or normalization using different calculation methods and many more. The great advantage is that raw data from different sources can be individually pre-processed before combining them into a common dataset.

The concept of data analysis using Mayday is based on the assumption, that all data represent true biological information instead of being contaminated by technically induced variations. In order to fulfill this assumption, SeaSight provides a pre-processing pipeline for normalizing the raw data. Figure 3 shows the graphical user interface of SeaSight with an example for such a transformation pipeline.



Experiments in this Matrix: 32

```
1) M8R_003b01.CEL RMA:BG QNorm Log 2.0 RMA:Polish -> "F199 Full"     22779 features, ⚠ no loci
Imported CEL file: M8R_003b01.CEL          [absolute expression, 1 channel, normalized, log 2]
2) M8R_003b02.CEL RMA:BG QNorm Log 2.0 RMA:Polish -> "F199 Full"     22779 features, ⚠ no loci
Imported CEL file: M8R_003b02.CEL          [absolute expression, 1 channel, normalized, log 2]
3) M8R_003b07.CEL RMA:BG QNorm Log 2.0 RMA:Polish -> "F199 Full"     22779 features, ⚠ no loci
Imported CEL file: M8R_003b07.CEL          [absolute expression, 1 channel, normalized, log 2]
4) M8R_003b12.CEL RMA:BG QNorm Log 2.0 RMA:Polish -> "F199 Full"     22779 features, ⚠ no loci
Imported CEL file: M8R_003b12.CEL          [absolute expression, 1 channel, normalized, log 2]
5) M8R_003b13.CEL RMA:BG QNorm Log 2.0 RMA:Polish -> "F199 Full"     22779 features, ⚠ no loci
Imported CEL file: M8R_003b13.CEL          [absolute expression, 1 channel, normalized, log 2]
6) M8R_003b17.CEL RMA:BG QNorm Log 2.0 RMA:Polish -> "F199 Full"     22779 features, ⚠ no loci
Imported CEL file: M8R_003b17.CEL          [absolute expression, 1 channel, normalized, log 2]
7) M8R_003b19.CEL RMA:BG QNorm Log 2.0 RMA:Polish -> "F199 Full"     22779 features, ⚠ no loci
Imported CEL file: M8R_003b19.CEL          [absolute expression, 1 channel, normalized, log 2]
8) M8R_003b22.CEL RMA:BG QNorm Log 2.0 RMA:Polish -> "F199 Full"     22779 features, ⚠ no loci
Imported CEL file: M8R_003b22.CEL          [absolute expression, 1 channel, normalized, log 2]
9) M8R_003c01.CEL RMA:BG QNorm Log 2.0 RMA:Polish -> "F199 Full"     22779 features, ⚠ no loci
Imported CEL file: M8R_003c01.CEL          [absolute expression, 1 channel, normalized, log 2]
10) M8R_003c02.CEL RMA:BG QNorm Log 2.0 RMA:Polish -> "F199 Full"    22779 features, ⚠ no loci
Imported CEL file: M8R_003c02.CEL          [absolute expression, 1 channel, normalized, log 2]
11) M8R_003c04.CEL RMA:BG QNorm Log 2.0 RMA:Polish -> "F199 Full"    22779 features, ⚠ no loci
Imported CEL file: M8R_003c04.CEL          [absolute expression, 1 channel, normalized, log 2]
12) M8R_003c05.CEL RMA:BG QNorm Log 2.0 RMA:Polish -> "F199 Full"    22779 features, ⚠ no loci
Imported CEL file: M8R_003c05.CEL          [absolute expression, 1 channel, normalized, log 2]
13) M8R_003c06.CEL RMA:BG QNorm Log 2.0 RMA:Polish -> "F199 Full"    22779 features, ⚠ no loci
Imported CEL file: M8R_003c06.CEL          [absolute expression, 1 channel, normalized, log 2]
14) M8R_003c07.CEL RMA:BG QNorm Log 2.0 RMA:Polish -> "F199 Full"    22779 features, ⚠ no loci
Imported CEL file: M8R_003c07.CEL          [absolute expression, 1 channel, normalized, log 2]
15) M8R_003c08.CEL RMA:BG QNorm Log 2.0 RMA:Polish -> "F199 Full"    22779 features, ⚠ no loci
Imported CEL file: M8R_003c08.CEL          [absolute expression, 1 channel, normalized, log 2]
16) M8R_003c09.CEL RMA:BG QNorm Log 2.0 RMA:Polish -> "F199 Full"    22779 features, ⚠ no loci
Imported CEL file: M8R_003c09.CEL          [absolute expression, 1 channel, normalized, log 2]
17) M8R_003c10.CEL RMA:BG QNorm Log 2.0 RMA:Polish -> "F199 Full"    22779 features, ⚠ no loci
Imported CEL file: M8R_003c10.CEL          [absolute expression, 1 channel, normalized, log 2]
18) M8R_003c11.CEL RMA:BG QNorm Log 2.0 RMA:Polish -> "F199 Full"    22779 features, ⚠ no loci
Imported CEL file: M8R_003c11.CEL          [absolute expression, 1 channel, normalized, log 2]
19) M8R_003c12.CEL RMA:BG QNorm Log 2.0 RMA:Polish -> "F199 Full"    22779 features, ⚠ no loci
Imported CEL file: M8R_003c12.CEL          [absolute expression, 1 channel, normalized, log 2]
20) M8R_003c13.CEL RMA:BG QNorm Log 2.0 RMA:Polish -> "F199 Full"    22779 features, ⚠ no loci
Imported CEL file: M8R_003c13.CEL          [absolute expression, 1 channel, normalized, log 2]
21) M8R_003c14.CEL RMA:BG QNorm Log 2.0 RMA:Polish -> "F199 Full"    22779 features, ⚠ no loci
Imported CEL file: M8R_003c14.CEL          [absolute expression, 1 channel, normalized, log 2]
22) M8R_003c15.CEL RMA:BG QNorm Log 2.0 RMA:Polish -> "F199 Full"    22779 features, ⚠ no loci
```

☑ Parallel execution

Figure 3: Graphical user interface of Mayday SeaSight. The experiments that have been loaded are labeled with the respective identifiers as well as the selected transformations during the pre-processing pipeline. On the right side, there are the buttons for editing experiments and transformations.

Import and pre-processing raw data

Mayday SeaSight provides an automated data import which spares the users having to load the data manually and reduces the number of possible user-made errors. The user only needs to specify the file format for every type of experiment and Mayday loads the format specific importer. If necessary, the user will be asked to input further information.

After importing raw data, experiment (and file format) specific pre-processing can be selected. In SeaSight, the calculations and steps in order to pre-process raw data and to create a Mayday dataset are called transformations. As part of pre-processing raw data, several transformations can be sequentially combined. The application of the same transformations for experiments of same origin and file format is desirable, due to consistency.

For commonly used data formats, SeaSight sets recommended transformations by default. Of course, additional transformations are selectable. However, not every transformation can sometimes be adapted due to prior set transformations or used data format, but SeaSight provides a help menu which shows why some transformations cannot be applied.

The transformation `Create dataset from common features` combines data from different file formats using the probe IDs as identifier which is necessary to create a Mayday dataset. All probes with IDs that occur in all experiments are combined to a new dataset

3.3 Practical example

In the following, a step-by-step instruction shows, how to apply the previously described SeaSight's functionality to a set of CEL files from *Affymetrix* runs. For using SeaSight make sure that the correct plugin path is set (compare Section 2.1 and 2.4). For demonstration purpose data import and pre-processing will be conducted on the F199 data.

Import example data

- In Mayday menu bar (see Figure 1 (**5**)), `Data Set` → `Further import options` → `Import raw data (SeaSight)` will open the SeaSight plugin which is shown in Figure 3.
- New experiment raw data can be added with `Add Experiments`. This will open the `Import Experiments` window in which the data format can be specified.
- F199 has been produced using *Affymetrix* GeneChips®, so we select `Affymetrix CEL files` as `Import Plugin`.
- Clicking on `Add` opens a file manager to select the files which contain the desired data provided in exampleData.zip. For this example we select all files and proceed.

Apply data transformation

SeaSight will automatically select suitable transformations for the dataset. Here, RMA-transformation will be applied (RMA = Robust Multi-array Average, [HKY99]). RMA will perform the following transformations:

- Background correction and quantile normalization which make sure, that variations in gene expression are only caused by biological instead of technical reasons.
- RMA-polish, which reduces unwanted interferences as well.
- $\log_2$-transformation. Taking the logarithm scales the enormous range of values to a smaller range between 0 and 16.

If you don't want to run RMA, remove the check mark `Perform RMA` in the `Import Experiments` window.

To process the `CEL` files, SeaSight needs further information provided by a `CDF` file (=Chip Definition File). A `CDF` file contains information about how to summarize and interpret the experimental data. If SeaSight cannot find a suitable `CDF` file, a new window shows the message

```
CDF file not found.  Please select a CDF file
        for Chip Type "ScoeA32a520627F".
```

Click `OK` to load the file `ScoeA32a520627F.CDF` which is provided in exampleData.zip as well.

Add further transformation

In addition to transformations applied by default, further transformations can be selected. For this, select all rows and click on `Add Transformation` to open a new window in which transformations can be selected from a list. There is a button `Show me, why some transformations cannot be applied` to list the requirements that are necessary to apply some further transformations. (This window misses an exit-button and can be closed by right-clicking at the top bar and selecting `close`).

As shown in Figure 3, colored labels show the selected transformations. They can be removed by right-clicking on the respective label.

Saving pre-processed data

To avoid repeating this procedure every time before working with a dataset, SeaSight can export the loaded experiments as a so-called *snapshot* which saves the applied pre-processing pipeline as well. This enables an eventual altering of the pre-processing without conducting all import and pre-processing steps again. Select `Save Matrix` in `SeaSight` to export a snapshot of the data as `SeaSight matrix file (.maydayz.SeaSight)`. A `SeaSight matrix file` can be loaded via `Load matrix`. For more information compare Section 6.1.

Create Mayday DataSet

To transfer the data from SeaSight into Mayday, the transformation `Create DataSet from common features` needs to be applied to all experiments. (`CTRL + A` to mark all experiments, select `Add Transformation`). This transformation combines all experiments, especially when they are from different file formats. All information will be merged and translated to a Mayday-readable data format. Optionally one can set a custom name for the dataset.

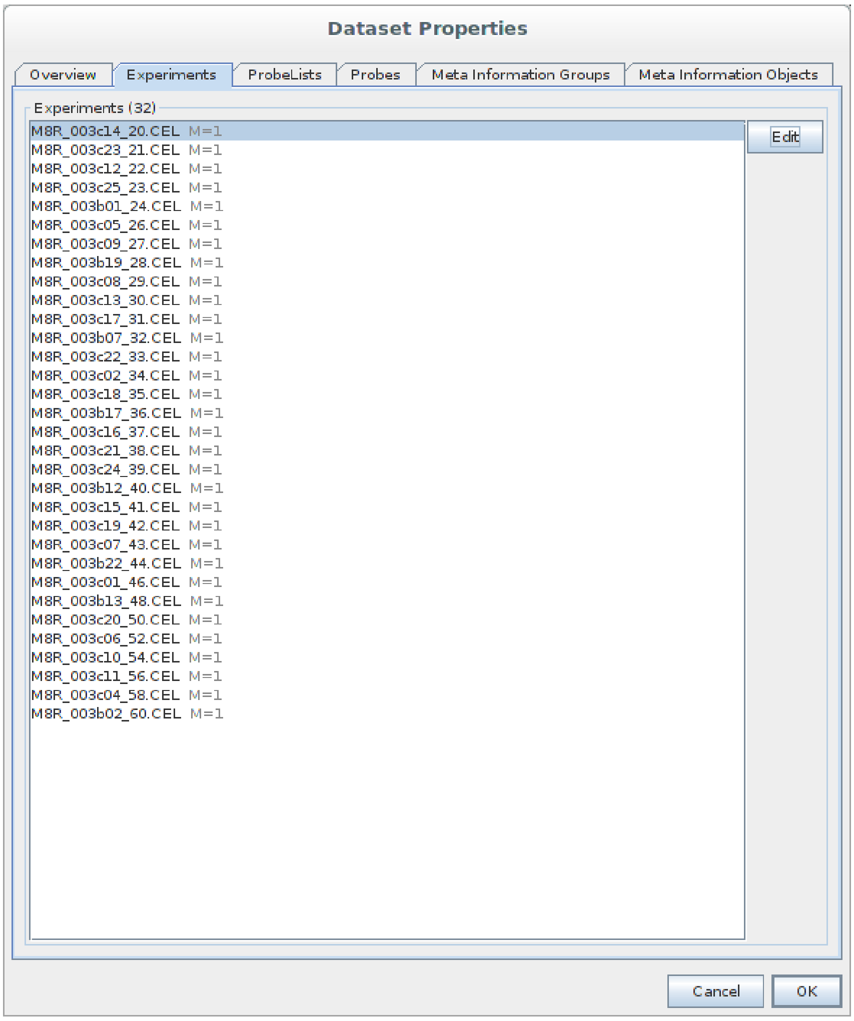
Clicking on `Run Transformation Now` in the bottom corner of the SeaSight window applies the selected transformations and builds a new `Mayday` DataSet from the `CEL` files. This may take some time depending on the numbers of experiments. In Mayday, a new window `Dataset Properties` will open. If you want to add some more information, go on with Section 4, else click `OK`. The SeaSight window remains in the background and can be closed now.

4 Data analysis

Additional information can be added to experiment names in Mayday manually in order to improve clearness and comprehensibility. The following two sections show, how this can be done for the example of the F199 dataset.

4.1 Change experiment names

Often, one wants to add further information about the experiment's conditions for improved visualizations and eased interpretations. Especially for analyzing time series experiments it is important to add information about the time points when experiments were conducted.



Experiments (32)

```
M8R_003c14_20.CEL  M=1
M8R_003c23_21.CEL  M=1
M8R_003c12_22.CEL  M=1
M8R_003c25_23.CEL  M=1
M8R_003b01_24.CEL  M=1
M8R_003c05_26.CEL  M=1
M8R_003c09_27.CEL  M=1
M8R_003b19_28.CEL  M=1
M8R_003c08_29.CEL  M=1
M8R_003c13_30.CEL  M=1
M8R_003c17_31.CEL  M=1
M8R_003b07_32.CEL  M=1
M8R_003c22_33.CEL  M=1
M8R_003c02_34.CEL  M=1
M8R_003c18_35.CEL  M=1
M8R_003b17_36.CEL  M=1
M8R_003c16_37.CEL  M=1
M8R_003c21_38.CEL  M=1
M8R_003c24_39.CEL  M=1
M8R_003b12_40.CEL  M=1
M8R_003c15_41.CEL  M=1
M8R_003c19_42.CEL  M=1
M8R_003c07_43.CEL  M=1
M8R_003b22_44.CEL  M=1
M8R_003c01_46.CEL  M=1
M8R_003b13_48.CEL  M=1
M8R_003c20_50.CEL  M=1
M8R_003c06_52.CEL  M=1
M8R_003c10_54.CEL  M=1
M8R_003c11_56.CEL  M=1
M8R_003c04_58.CEL  M=1
M8R_003b02_60.CEL  M=1
```

Figure 4: The `Change experiment names` window provides editing the names of the experiments. Select an experiment and click on `Edit` to change its name.

To change experiment names, right-click on the dataset (see Figure 1 (**1**)) and select `Properties` to open the `Dataset Properties` window. The tab `Experiments` lists all experiments. The name of a selected experiment can be changed with `edit`. Confirm changes with `OK`.

4.2 Change experiment order

Sometimes, the user wishes to change the order of the experiments. In particular this is essential for time series experiments. It can be done as follows:

- `Data Set` → `Transform` → `Change Experiment Order` opens a new window `Change Experiment Order for data set`.
- Use drag-and-drop to bring the experiments in the correct order.
- Confirm with `OK`.

4.3 Overview of dataset

Before starting with a deeper analysis, we want to get a first visual impression of the data after normalization. Mayday provides many kinds of data visualization which are presented in Section 5 but for now we will only use boxplots, profile plots and scatter plots.

Boxplot

A boxplot [MTL78] shows the distribution of values as plots of quantiles. A quantile contains all values that are less or equal its respective value. Boxplots show the 25%, 50% (mean) and 75% quantile as well as the inter-quartile-range, that means the middle 50% of the values, and the minimum/maximum. Boxplots can be used to visualize the data distribution to prove whether Seasight's quantile normalization had worked successful. Furthermore, experiment outliers can be detected. Figure 5 shows boxplots of the gene expression level of all experiments before (above) and after quantile normalizaton (below): when quantile normalization has been applied successfully, all quantiles are approximately on the same level.

A boxplot can be created with a right-click on a probe list (here: probe list `global`) and navigating to `Visualization` → `Boxplot`.

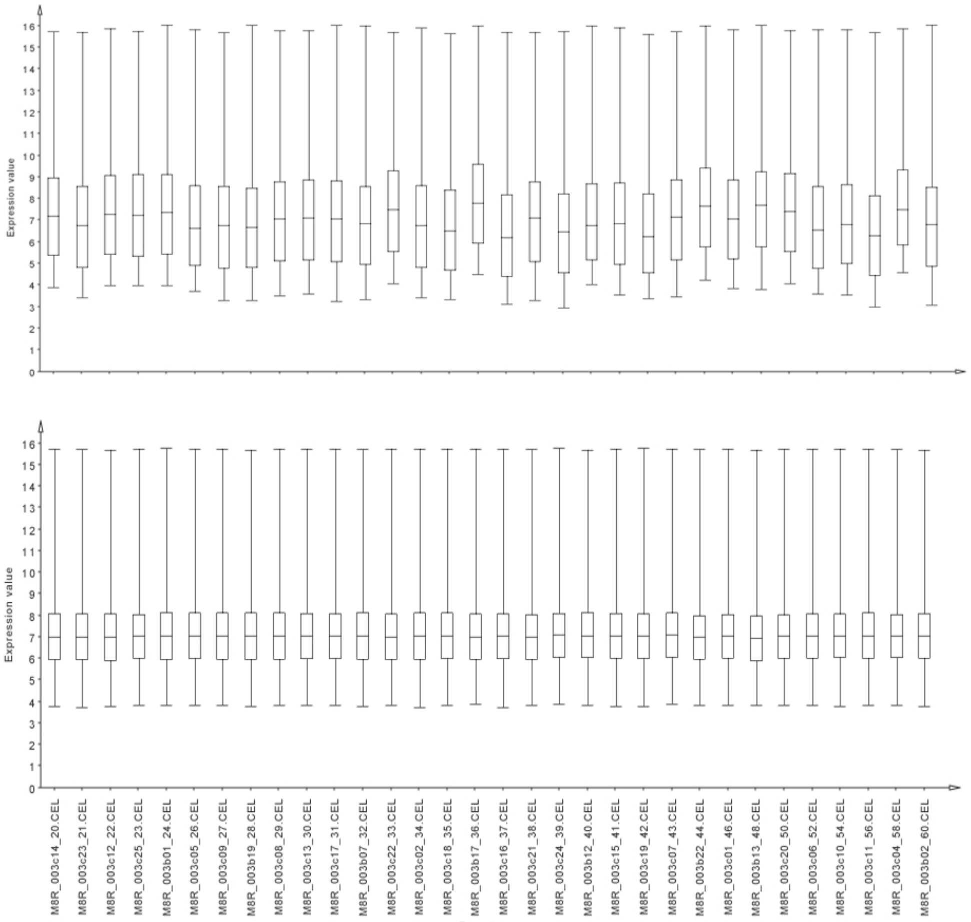


Figure 5: Boxplots are used to visualize and compare the quantiles of the probes during all experiments. At the top, a boxplot shows the quantiles of the F199 data without quantile normalization. Below are the same data but they are quantile normalized successfully; all experiments have approximately the same quartiles.

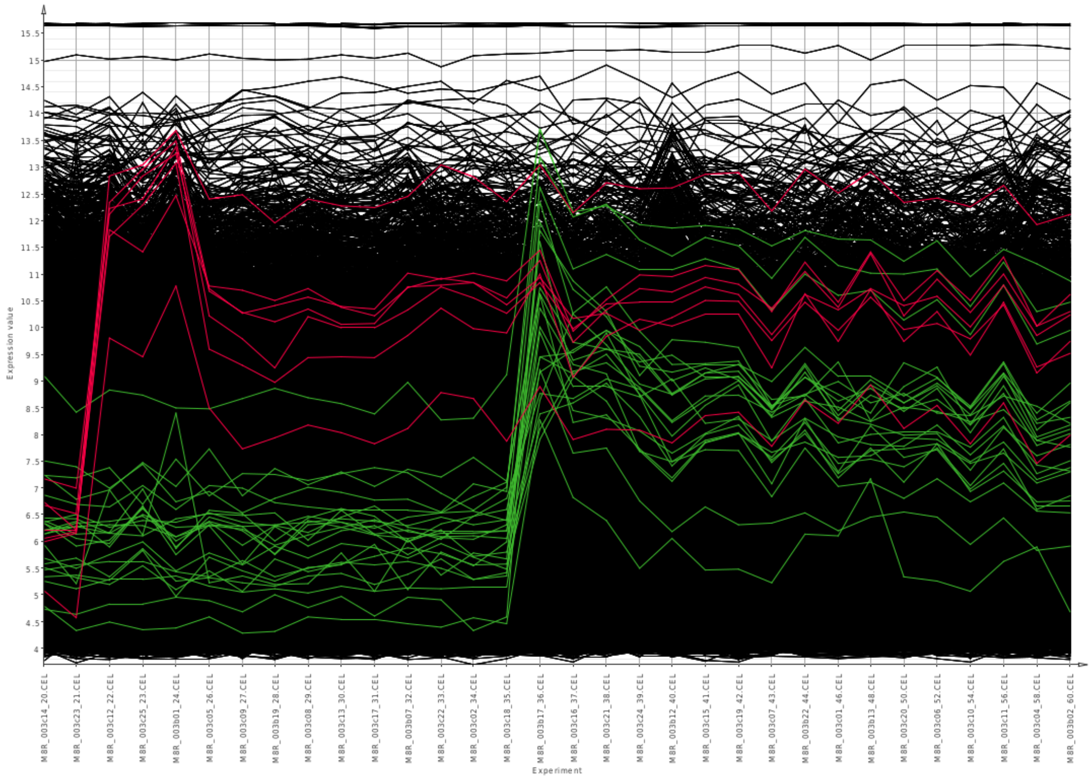


Figure 6: A profile plot shows the expression profiles of all 22779 transcripts contained in F199 (in black). In green and red some genes are highlighted, which show a common expression pattern (genes seems to be co-regulated). This plot illustrates the necessity to find and isolate co-regulated genes.

Profile Plot

The gene expression levels, summarized in Figure 5 can be visualized as more intuitive plot of gene expression profiles as well. A profile plot visualizes the (logarithmic) gene expression values during all experiments. For better visual perception, these points are connected by lines. Figure 6 shows profiles of all genes (black). Additionally, profiles of some differentially regulated genes are highlighted (green and red). This picture illustrates the importance of the following analysis steps, especially the extraction of similarly regulated genes. Double-click on the preview window of probe list `global` shows a plot of all gene profiles during all experiments.

Scatter plot

To determine the dependency of two data variables visually and to detect potential clusters, a scatter plot is appropriate. For example, expression values of two experiments can be plotted against each other. To create such a scatter plot, right-click on a probe list and select `Visualization` → `Scatter Plot`. Via `View` → `X axis (Y axis)` → `Configure experiment`, user can select, which experiments should be plotted. Meta informations can be visualized as well. To visualize a MIO, select `View` → `X axis (Y axis)` → `meta information`. At `Configure meta information` in the same menu, the desired MIO can be selected and manipulations such as taking the logarithm can be applied using `Meta information manipulator`. Figure 7 (left side) shows a scatter plot of expression values from the first two experiments contained in F199 dataset.

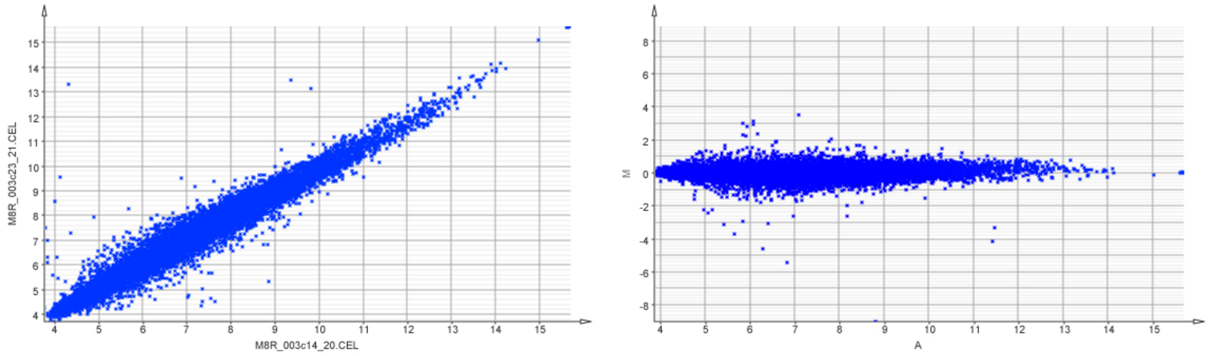


Figure 7: Left side: A scatter plot of the gene expression from the 20th hour vs. the 21st hour. Right side: An Ma-Plot of the same data, which is actually a rotated and scaled scatter plot of log ratio against the average of the respective values.

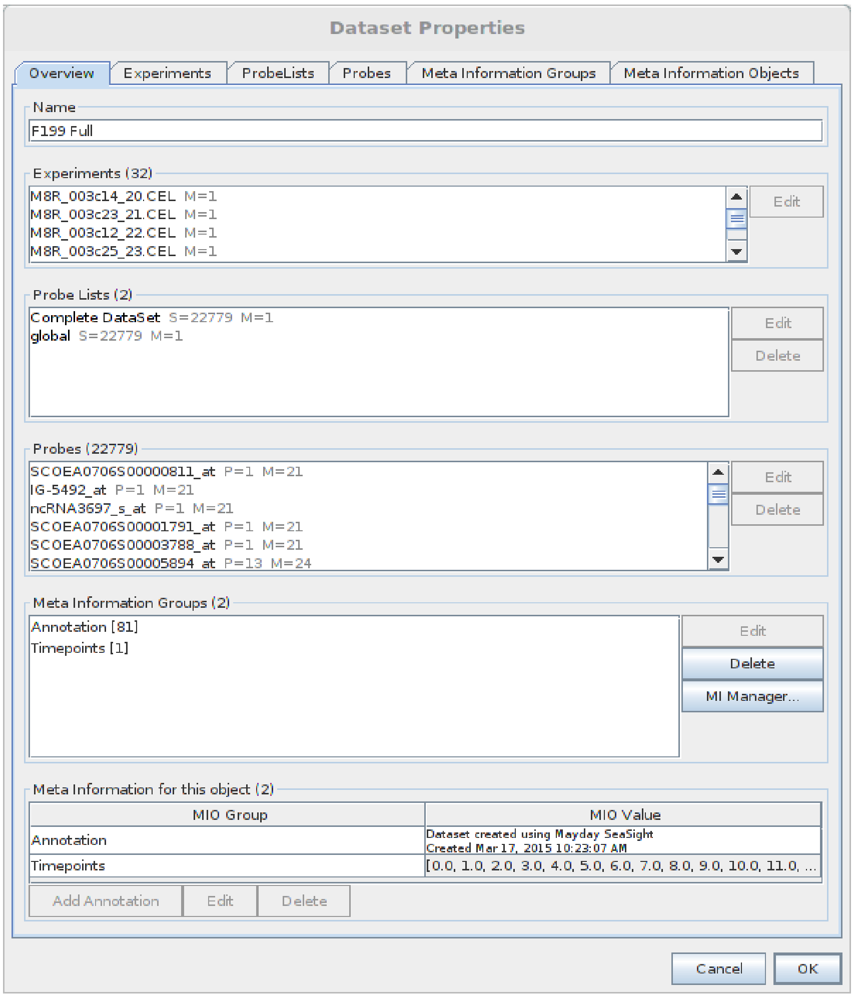
MA-Plot

A special kind of scatter plots is the so-called MA-Plot. MA-Plots are used to study dependences between the log ratio of two variables and the mean values of the same two variables. In context of microarray analysis, gene expression level of two experiments/arrays can be compared. An MA-Plot is a 45° rotated and scaled scatter plot, where the log ratio is plotted against the average of the respective logarithmized values.

Right-click on a probe list and select `Visualization` → `MA Plot` to produce an MA-Plot. Figure 7 (right side) shows an MA-Plot of the same data as the scatter plot on the left side.

4.4 Dataset properties

All information associated with a dataset such as its size, names and number of experiments as well as information about the contained probes, probe lists and MIOs are summarized in the dataset properties. The `Dataset Properties` window can be accessed via right-click on a dataset in the dataset management at `Properties`. Figure 8 shows the `Dataset Properties` window. Use the tabs at the top to navigate through the properties. Most fields can be edited or deleted using the respective buttons on the right side.



Name
F199 Full

Experiments (32)
M8R_003c14_20.CEL M=1
M8R_003c23_21.CEL M=1
M8R_003c12_22.CEL M=1
M8R_003c25_23.CEL M=1

Probe Lists (2)
Complete DataSet S=22779 M=1
global S=22779 M=1

Probes (22779)
SCOEA0706S00000811_at P=1 M=21
IG-5492_at P=1 M=21
ncRNA3697_s_at P=1 M=21
SCOEA0706S00001791_at P=1 M=21
SCOEA0706S00003788_at P=1 M=21
SCOEA0706S00005894_at P=13 M=24

Meta Information Groups (2)
Annotation [81]
Timepoints [1]

Meta Information for this object (2)

MIO Group	MIO Value
Annotation	Dataset created using Mayday SeaSight Created Mar 17, 2015 10:23:07 AM
Timepoints	[0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0, ...

Figure 8: In the `Dataset Properties` window, all information associated with a dataset are accessible. The tabs at the top are for navigating through the properties. The buttons on the right side are for editing the respective information.

4.5 The dynamic probe list, a powerful feature in Mayday

During analysis, we will use the previous described dynamic probe list. A new dynamic probe list can be created as follows: right-click and select `Create` → `Dynamic ProbeList` to open `Dynamic ProbeList Properties` window (see Figure 10). Select `Add Rule` to set the criteria for the new probe list by combining suitable data processors. For example, a MIO can be used to define new filtering rules. Section 4.6 provides an example for how to use dynamic probe lists in combination with MIOs for analysis and data visualization. To open `Probelist Properties` again, double-click on a probe list.

Double-clicking on probe list preview opens a profile plot window which shows the gene expression during the experiments as profiles which are actually points connected by lines for better visual perception. Other visualizations can be selected with a right-click on the respective probe list under `Visualization`.

4.6 Statistics

In general, a gene expression experiment produces a lot of data. Thus an important part of analysis is data reduction and extraction of important information. There are many ways to find genes, which show a response to altered experiment conditions, or in case of a time series experiment, genes that are differentially regulated over time. Gene activity is represented in the measured gene expression, so altered gene expression can be taken as indicator, which genes are active or change their activity. That results in filtering gene expressions with mathematical methods to detect such genes. Mayday provides several methods such as statistical tests or analysis of variance to find such genes.

Looking at our example data we want to analyze, which genes are involved in metabolic switch when phosphate is depleted and exponential growth ends. F199 contains 22779 transcripts from *S. coelicolor* whose expression profiles are shown in Figure 6. Obviously it is necessary to extract genes which seem to be differentially regulated. To reduce the amount of data, the other genes can be ignored here. These can be objects of deeper analysis.

This Tutorial shows several ways how to find differentially expressed genes using Mayday. In particular, we will use gene expression variance and statistical tests to identify significant genes.

Find most variant genes

As described before, the amount of data needs to be reduced in order to find the important, differentially regulated genes. One way to evaluate that mathematically is to calculate the variance in gene expression because differentially regulated genes leads to altered expression values during the time series experiments what causes an increased expression variance during the experiments. This only makes sense, if there are genes that are differentially regulated and others that are not. In such a large dataset like F199, it is very likely to find these conditions, so that we can use the expression variance to create a new probe list containing the subset of the most variant and therefore differentially regulated genes:

- To calculate the gene variance, right-click on probe list `global` and select `Statistics` → `Derived Statistics` → `Probe Variance` (see Figure 9). In the meta information window a new meta information group `(Probe Statistics)` containing the MIO `Expression variance` will appear.
- We will use the expression variance to create a new dynamic probe list. Right-click on the probe list window and select `Create` → `Dynamic ProbeList`.
- Make sure, `All of these rules` is marked and click `Add Rule` to add a new rule.
- As `data processor` select `Meta-information values`. In new window `MIO Group Selection` select `Expression variance` in group `(Probe Statistic)` to calculate the variances and click `OK`.
- As second data processor select `Compare a number`. Because we want to filter for the most variant genes, we have to filter for genes that have a variance at least equal or above a certain threshold.
- The number of matching probes is displayed on the left side. Raise or lower the chosen threshold to see how the size of the matching probes change.
- It is recommended to rename the new probe list, for example to "most variant genes" and optionally choose a different color. Mayday will display the filtering rules such as data processors or used threshold as additional information at the probe list. Confirm with `OK`. Now we've created a dynamic probe list containing the most variant genes.

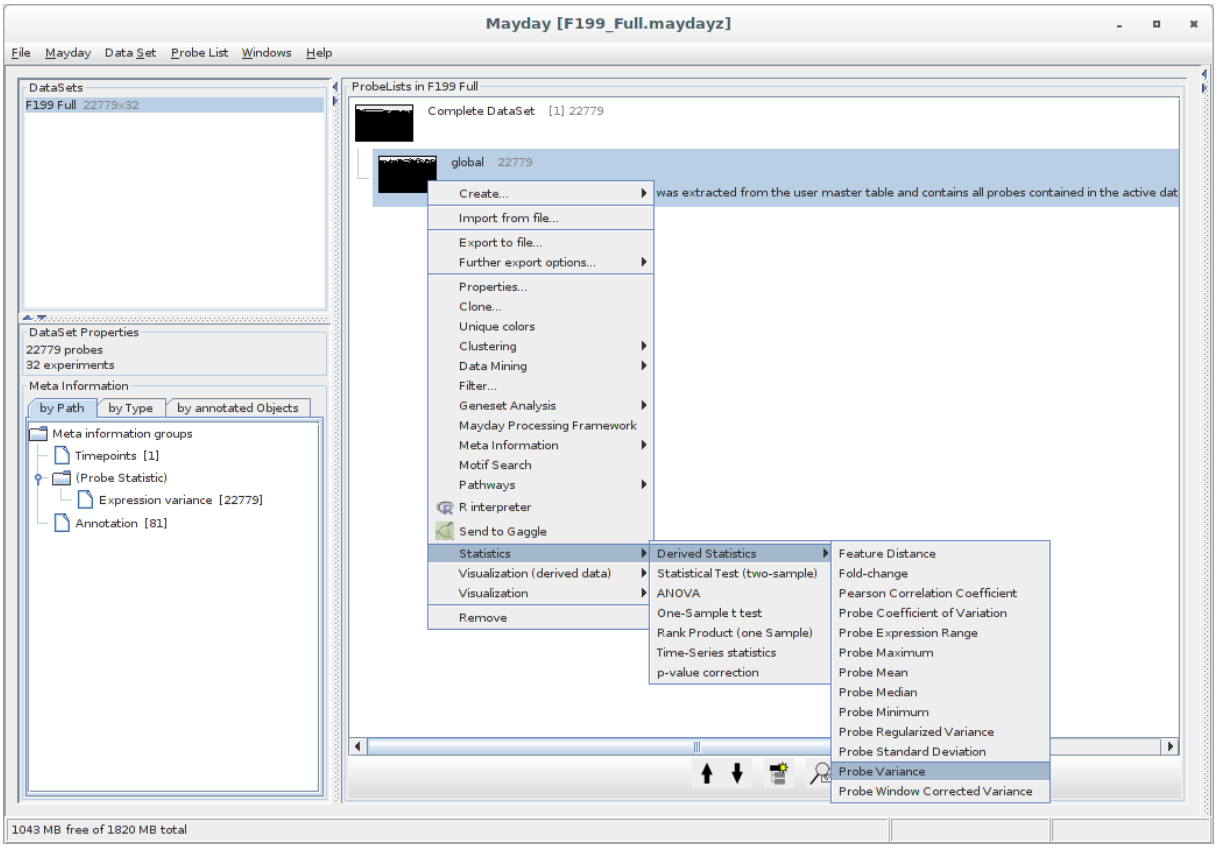
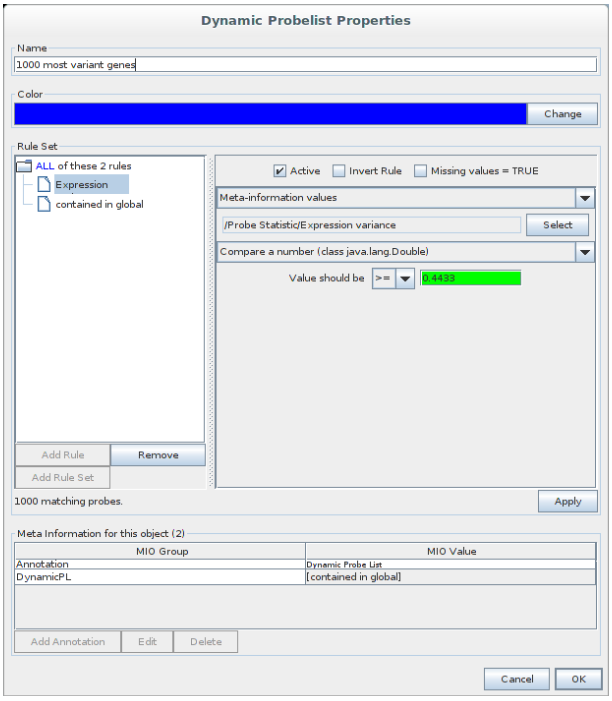


Figure 9: In the contex menu `Statistics`, the expression variance can be calculated. On the right side, a new MIO containing the calculated variances will appear



Name: 1000 most variant genes

Rule Set

- ALL of these 2 rules
  - Expression
  - contained in global

☑ Active ☐ Invert Rule ☐ Missing values = TRUE

Meta-information values

/Probe Statistic/Expression variance

Compare a number (class java.lang.Double)

Value should be >= 0.4433

1000 matching probes.

Meta Information for this object (2)

MIO Group	MIO Value
Annotation	Dynamic Probe List
DynamicPL	[contained in global]

Figure 10: In the `Dynamic Probelist Properties` window, dynamic probe lists can be created by combining filtering rules. Such a rule is defined by sequential data processors. On the left side is shown, how many probes match the current rules. The shown combination of data processors match the 1000 most variant genes.

Find differentially expressed genes using statistical tests

A fundamental aspect of expression analysis is to reveal differences in expression levels of the same gene between different conditions. In Mayday statistical methods can be used to find genes, that show such differences. Two-sided Student's t-test [Hay13] which tests whether two selected groups of values have the same mean is one of them. Testing equality of mean produces the best results, if there are enough differences in gene expression between the tested groups.

However, be careful with the results of any statistical test. Every test makes certain assumptions about the underlying data but the test itself does not verify, whether these assumptions match to the current data. When in doubt, inform yourself about the respective statistical test.

Testing a large number of objects, also known as multiple testing, using for example Student's t-test even with a low error rate of 0.01 e.g, might still cause many false positives. Mayday provides more or less conservative p-value correction methods to address this phenomenon, such as `Bonferroni, FDR, Holm's` and `no correction` (decreasing conservative) which adapt the error rate in order to reduce the number of falsely positive tested values. In this context, *conservative* means, that false positive matches are reduced as much as possible. Especially when dealing with biological data, sometimes it is better to falsely declare some genes as differentially regulated than missing truly differentially regulated genes due to a too conservative correction method.

We will now create a new dynamic probe list using p-values of Student's t-test:

- Right-click on probe list `global` and navigate to `Statistics` → `Statistical Test (two-sample)` will open `Statistical Testing` window (see Figure 11).

- Group the data in two classes that Student's t-test should compare, by `Define exactly 2 classes`. Because we expect to see the metabolic switch, we can divide the experiments in two groups before and after hour 35. Compare Section 2.5 and Figure 2 for details.

- As `Test` method select `Student's t test` and make sure, that `Equal variance` is activated because we assume, that the data have equal variances.

- Choose an p-value correction method. Here, we will use `Bonferroni` correction.

- Activate `Create ProbeList of significant probes` and select a suitable significance level to create a new probe list of significant probes with a specified threshold.

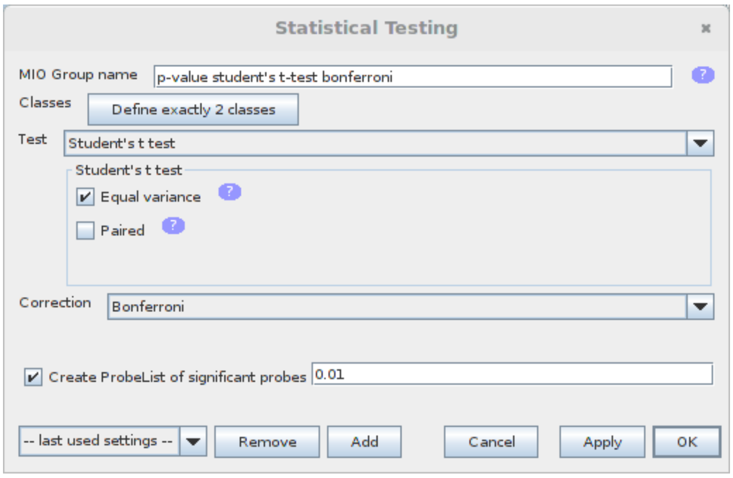
As alternative to Student's t-test, `ANOVA` (=analysis of variance) can be used. `ANOVA` tests equality of means of more than two groups. Similar to Student's t-test, `ANOVA` produces a MIO containing the respective p-values and optionally creates a new probe list according to threshold-filtered p-values which can be error-corrected as well.

- Right-click on a probe list and select `Statistics` → `ANOVA`.

- Define two or more groups to be compared.

- Choose a correction method.

- Activate `Create ProbeList of significant probes`, set threshold and confirm with `OK`.



Statistical Testing

MIO Group name: p-value student's t-test bonferroni

Classes: Define exactly 2 classes

Test: Student's t test

Student's t test

☑ Equal variance

□ Paired

Correction: Bonferroni

☑ Create ProbeList of significant probes: 0.01

Figure 11: `Statistical Testing` window provides several statistical test methods which can be selected in drop-down menu `Test`. Furthermore, a p-value correction can be conducted in `Correction` menu. Optionally, MAYDAY creates a new probe list containing significant probes according to a given threshold.

Under `Statistics` → `Derived Statistics`, many more statistical values can be calculated. For example *fold change*, *Pearson correlation coefficient*, expression range, minimum, maximum, mean, standard deviation or corrected variances.

4.7 Data mining via clustering

Clustering is an approach to group objects in a way, that similar objects are grouped together based on a chosen similarity (or distance) measure. We assume that more than one gene is involved in a cellular process such as metabolic switch of *S. coelicolor*. Additionally, one assumes, that a similar expression profile corresponds to co-regulated genes. Hence, there must be patterns in gene expression which can be used to find and group such co-regulated genes. This is exactly what clustering does.

There are basically two different clustering approaches: hierarchical and partitioning clustering which are both implemented in MAYDAY.

Both approaches can be combined with replaceable distance measures. MAYDAY provides Canberra, Chebychev, Euclidean, Manhattan, Minkowsky, Pearson Correlation, Spearman Rank Correlation, Supremum and Vector Angle distance. Euclidean distance and Pearson Correlation distance are common used distance measures, but of course the others can be chosen as well. Often, several distance measures needs to be tested to produce the best result.

Hierarchical clustering

Hierarchical clustering [Joh67] produces a hierarchy of nested clusters which are represented as a distance based tree, a so-called dendrogram. The cluster hierarchy is built by recursively dividing a set of objects based on their distance or recursively joining every single object with the most similar object in the set, respectively. Optimally, genes or experiments are clustered in a way that the distances in the dendrogram represent the similarity (or dissimilarity) of the respective gene expression profiles. Hierarchical clustering is recommended for a small number of objects, either genes or experiments. A hierarchical clustering of many objects will produce a confusing tree with many edges. Mayday provides several hierarchical clustering algorithms such as UPGMA, WPGMA and Rapid Neighbor Joining. Note that hierarchical clusterings can be conducted according to patterns of genes as well as experiments.

For the example of F199 a hierarchical clustering analysis using Mayday is conducted as follows:

- Right-click on the probe list which contains the 1000 most variant genes created before (or an arbitrary other probe list) and select `Clustering` → `Hierarchical` to open `Hierarchical Clustering` window.
- Several clustering algorithms and distance measures can be chosen.
- To cluster the genes, for example select `Rapid Neighbor Joining` algorithm [SMP08] and as distance measure `Euclidean`.
- To cluster the experiments, activate `Transpose Matrix`, select `WPGMA` algorithm [Car] and `Euclidean` distance for example.
- Confirm with `OK` respectively.

This will create a new probe list named `Hierarchical Clustering` followed by information about the used clustering algorithm. Simultaneously, a new window will open, where the result of the hierarchical clustering is shown as unrooted dendrogram. Figure 12 shows such an unrooted dendrogram, in Figure 13 a rooted dendrogram can be found on the left side of the heatmap.

Tree visualizer

The results of a hierarchical clustering are visualized by a tree structure, a so-called dendrogram. To produce such a dendrogram manually, right-click on a probe list containing hierarchically clustered genes and select `Visualization` → `Tree Visualizer`. The appearance of the dendrogram can be adjusted via `View` → `Layout Algorithm`.

Figure 12 shows an unrooted dendrogram produced by hierarchically cluster the experiments in F199 (transposed matrix) with the 1000 most variant genes using WPGMA algorithm (euclidean distance). The experiments are well separated into two groups before (blue) and after (red) metabolic switch after 35 hours in fermenter.

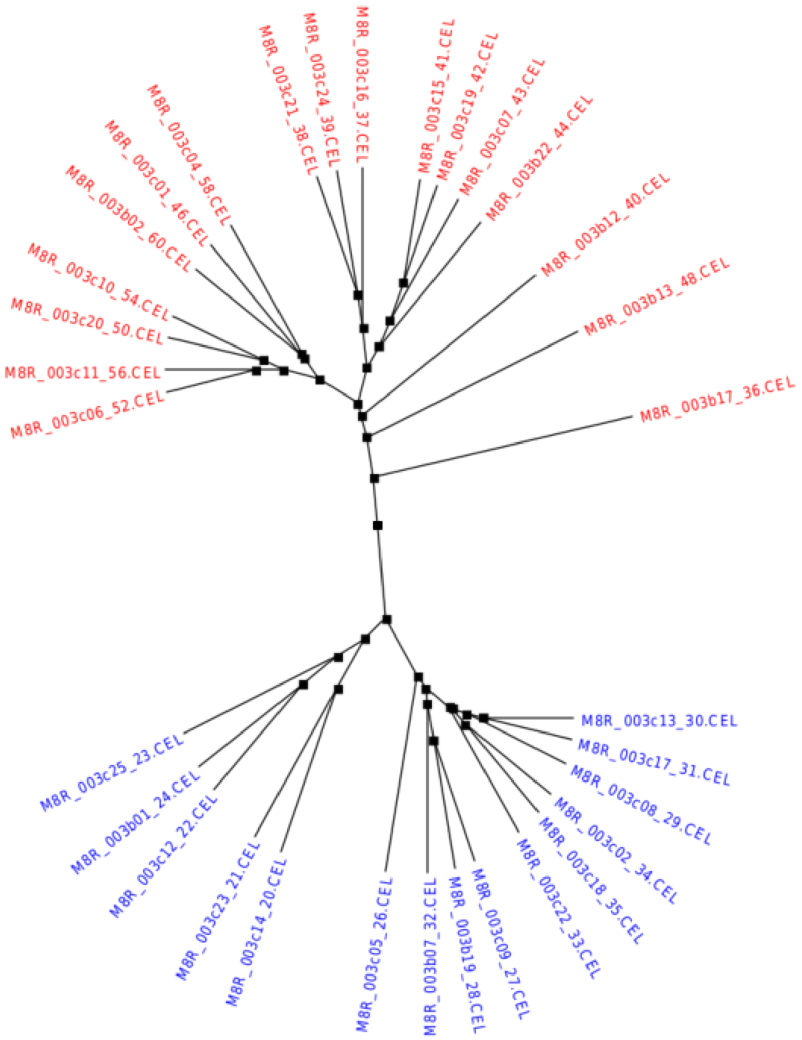


Figure 12: The result of clustering the experiments (transposed probe list of the 1000 most variant genes) hierarchically using WPGMA algorithm with euclidean distance visualized as dendrogram. The clustering nicely separates experiments into two groups before (blue) and after (red) metabolic switch.

Heatmap

Recognizing patterns and correlations in expression level of genes under different conditions as seen in Section 4.7 is important when dealing with gene expression data. A suitable visualization is the so-called heatmap that is a matrix visualization of gene expression [GDN05]. The expression level of a gene is visualized with a color gradient, usually from green (low) to red (high).

A heatmap can be produced by right-clicking on a probe list and selecting `Visualization` → `HeatMap`. The legend at the top resolves the color gradient encoding the gene expression level. The appearance of a heatmap can be adjusted in the menu `View` → `Detach menu`. This will open a new window in which a caption and labels can be added. In tab `Heatmap colors` of the `Column` tab, several predefined or user-defined color gradients can be applied and the way, how the gene expression is mapped to a color gradient can be adapted. The tab `Color enhancement` provides adding meta information values to the heatmap and experiment order can be changed in tab `Sort Experiments`.

Advanced heatmap

Normally, a heatmap of unclustered genes is very hard to interpret. To make best use of heatmaps, a hierarchical clustering is recommended. Hierarchical clustering improves the order in which the genes are visualized, in a way that (in best case) all co-regulated genes are ordered consecutively. It is possible to visualize the data with hierarchical clustering of genes and/or experiments as well.

To produce such a heatmap together with the dendrogram of the genes or/and experiments, visualize the probe lists of the hierarchical clustered genes or/and experiments as heatmap. This will add a dendrogram to the heatmap that shows the hierarchical relationship. A combined heatmap including hierarchical clustering of genes and experiments (transposed matrix) can be produced by visualizing the two corresponding probe lists as a heatmap (use `CTRL + click` to mark both probe lists). On the left side of the heatmap, a dendrogram with the hierarchically clustered genes will be drawn, at the top one containing the experiments.

Figure 13 shows a heatmap of 54 out of the 100 most variant genes of F199 (see Section 4.6). They are clustered hierarchically using Rapid Neighbor Joining algorithm and Pearson correlation distance) to produce the dendrogram and with `Qt-Clustering` algorithm (partitioning, `Distance Measure` = Pearson Correlation, `Diameter threshold` = 0.25, `Minimal cluster size` = 10) to add the colored labels at the left side which indicate the genes that are clustered together. The metabolic switch at hour 35 is well recognizable: some genes (blue) are highly expressed before metabolic switch and down-regulated afterwards. Other genes (red) are up-regulated during metabolic switch and some genes (yellow) show a up-regulation hours later. Through the hierarchical clustering of the genes, these coherences are visible.

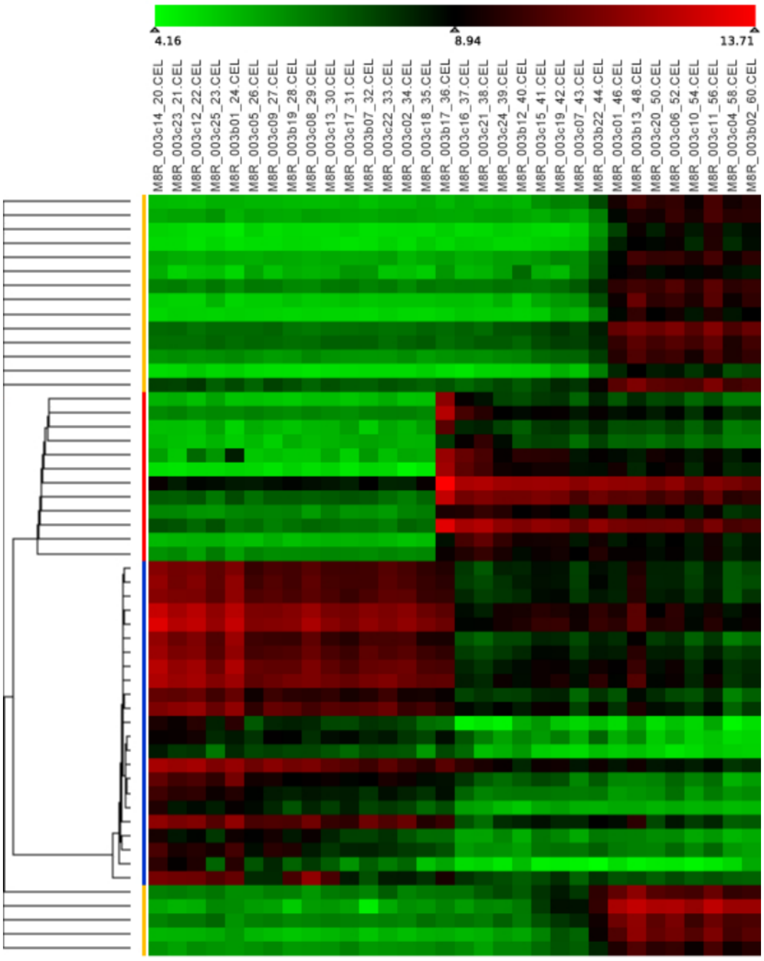


Figure 13: This heatmap was produced using 54 out of the 100 most variant genes. They were clustered using `Qt-Clustering`. `Qt-Clustering` has found 3 clusters and 46 genes which did not match a cluster. The remaining 54 genes were clustered hierarchically using Rapid Neighbor Joining algorithm.

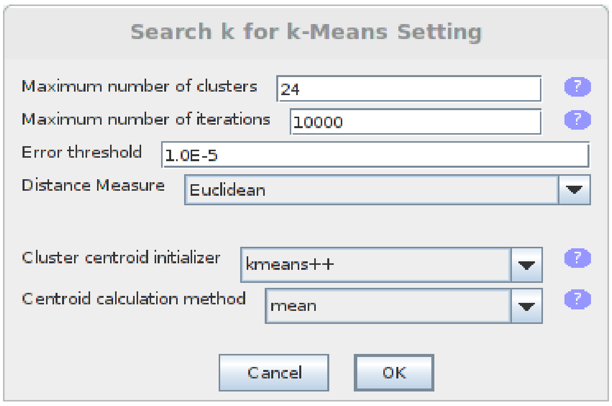
Partitioning clustering

In contrast to hierarchical clustering algorithms which produce nested clusters, partitioning clustering tries to group similar genes into disjoint clusters. Mayday provides several partitioning clustering algorithms such as *k-means*, self-organizing maps (SOM) [Koh90], QT-clustering or density based DBSCAN [EKSX96].

The most popular partitioning clustering algorithm is *k-means* [M$^+$67]. It is a very simple and fast algorithm which clusters the genes. The disadvantage is, that one needs to set the number of clusters k without knowing how many clusters are correct. Mayday provides a plugin to find the optimal k heuristically. Because there are different possibilities how to select cluster centroids and most of them are random, it is hard to get identically reproducible results. Several k and different methods to choose cluster centroids can be tried in order to get a good clustering. Compare “clustering assessment using silhouette plot” in Section 16 to get information about identifying a good clustering.

Originally, *k-means* uses the euclidean distance for clustering, but in Mayday, several other distance measures can be used as well.

To create a partitioning clustering of genes using *k-means*, right-click on the `1000 most variant genes` probe list and select `Clustering` → `Partitioning (k-Means, find`



Search k for k-Means Setting

Maximum number of clusters: 24

Maximum number of iterations: 10000

Error threshold: 1.0E-5

Distance Measure: Euclidean

Cluster centroid initializer: kmeans++

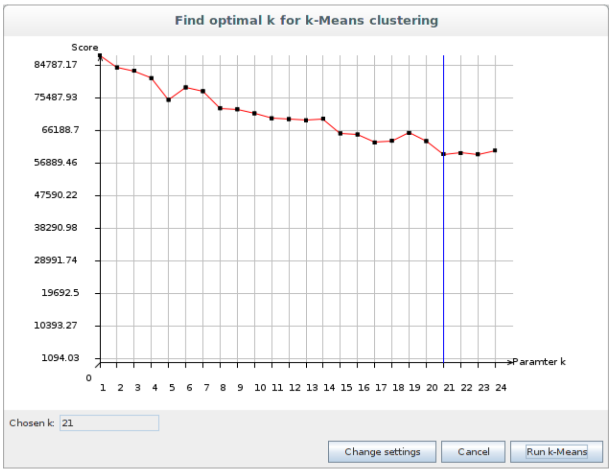
Centroid calculation method: mean

Figure 14: `Search k for k-Means Setting` window provides a heuristic to find an optimal k for *k-means*. Maximum number of clusters and iterations, error threshold and distance measure as well as cluster centroid initialization and calculation can be selected.

`optimal k)`. In the window `Search k for k-Means Setting` (see Figure 14) just click `OK`. Set `Maximum number of clusters` to 24 and `Distance Measure` to `Euclidean`.

Now, Mayday heuristically calculates a score for every number of clusters up to the chosen maximum and produces a scree plot (see Figure 15). Experienced researchers can interpret this plot in order to find a suitable k. An "elbow-like" kink in the plot can be an indicator, which k will produce a good clustering. Alternatively select a k that corresponds to the respected number of gene expression patterns. Click `Run k-Means` to run *k-means* clustering. Figure 18 shows the result of a *k-means* clustering using $k = 21$ and Pearson correlation distance. Later, this tutorial will show how to assess the quality of computed clusters.

Note, that partitioning clustering can only be conducted on genes trivially. To create a partitioning clustering on experiments, the respective probe list needs to transposed and transfered in a new dataset via `Data Set` → `Transform` → `Transpose Matrix`. Then, the former experiments will be represented as probes and former probes as experiments. Performing a partitioning clustering will now cluster the actual experiments.



Find optimal k for k-Means clustering

Chosen k: 21

Figure 15: Mayday uses heuristics to assess cluster quality. Such a scree plot can be used to estimate a suitable number of clusters k. Click on the chart to choose a suitable k.

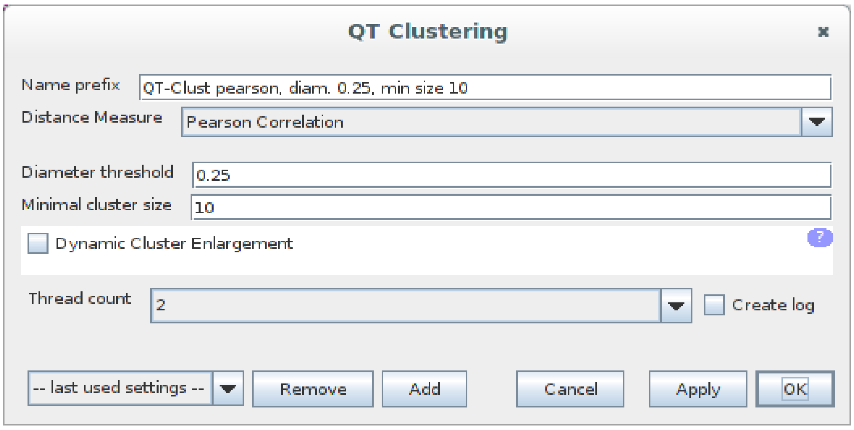
QT-Clustering

Another partitioning clustering algorithm implemented in Mayday is the quality-threshold clustering algorithm *QT-Clustering* [HKY99]. QT-Clustering allows a much more detailed adjustment of clustering criteria, cluster size and error rate. In opposite to *k-means*, no predefined number of clusters needs to be set.

We will now perform a quality-threshold clustering on the F199 dataset.

- Right-click on the probe list `1000 most variant genes` and select `Clustering` → `Quality-based (QT-Clustering)`.
- In the `QT Clustering` window (see Figure 16) select a distance measure, e.g. *Pearson Correlation*.
- Set `Diameter threshold` to 0.25 and `Minimal cluster size` to 4. If required, try different parameters.

Performing QT-Clustering will take some time dependent on the amount of data and selected distance measure. Mayday creates a new probe list for every cluster found by QT-Clustering algorithm and collect genes, which could not be associated to any cluster, in a separate probe list. The resulting probe lists are shown in Figure 17.


QT Clustering

Name prefix: QT-Clust pearson, diam. 0.25, min size 10

Distance Measure: Pearson Correlation

Diameter threshold: 0.25

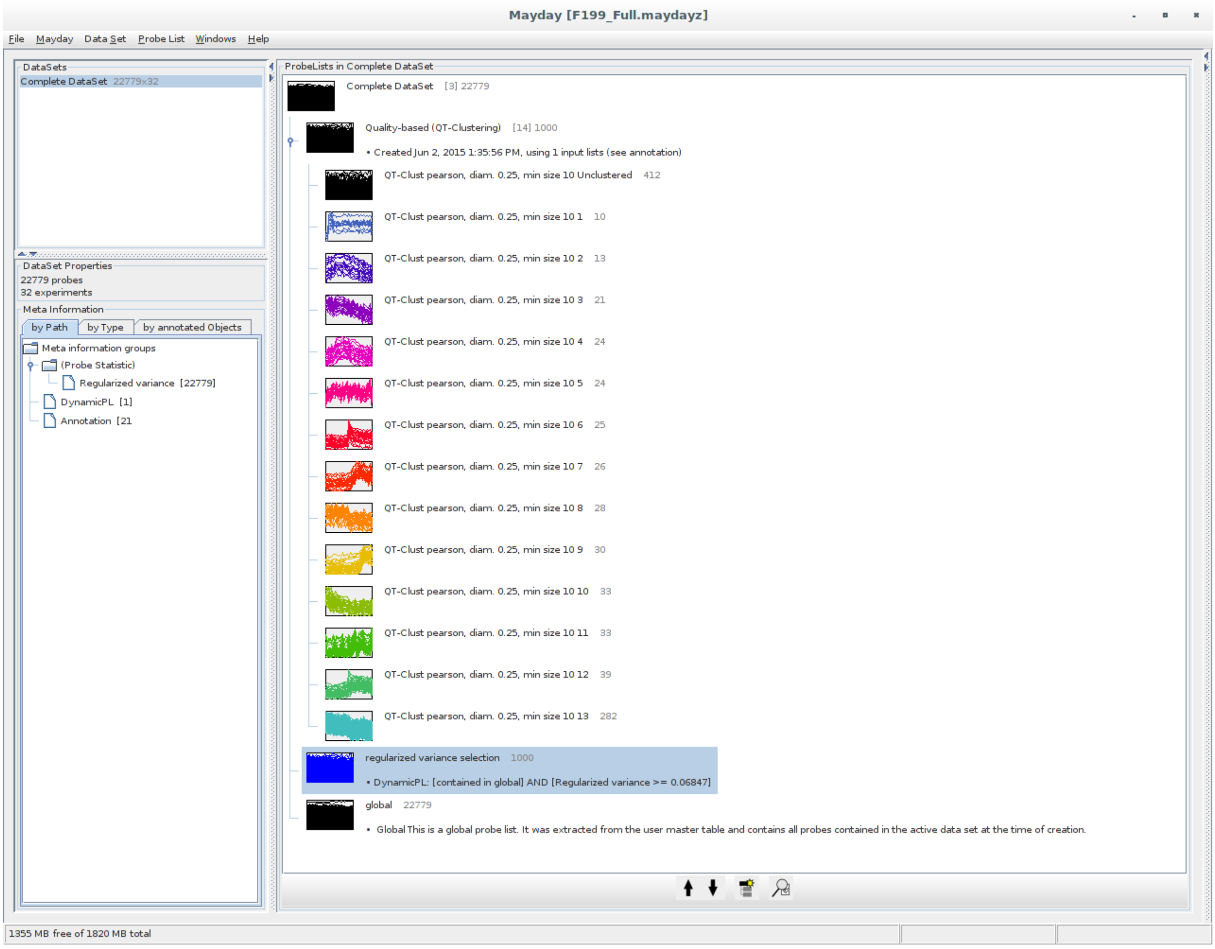
Minimal cluster size: 10

☐ Dynamic Cluster Enlargement

Thread count: 2

☐ Create log

Figure 16: In the `QT Clustering` window, the parameters for QT-Clustering algorithm such as distance measure, diameter threshold and minimal cluster size can be set.


Mayday [F199_Full.maydayz]
File Mayday Data Set Probe List Windows Help
DataSets
ProbeLists in Complete DataSet
DataSet Properties
Meta Information
by Path
by Type
by annotated Objects

Complete DataSet 22779x32

22779 probes
32 experiments

Meta information groups
(Probe Statistic)
Regularized variance [22779]
DynamicPL [1]
Annotation [21

Complete DataSet [3] 22779

Quality-based (QT-Clustering) [14] 1000
• Created Jun 2, 2015 1:35:56 PM, using 1 input lists (see annotation)

QT-Clust pearson, diam. 0.25, min size 10 Unclustered 412
QT-Clust pearson, diam. 0.25, min size 10 1 10
QT-Clust pearson, diam. 0.25, min size 10 2 13
QT-Clust pearson, diam. 0.25, min size 10 3 21
QT-Clust pearson, diam. 0.25, min size 10 4 24
QT-Clust pearson, diam. 0.25, min size 10 5 24
QT-Clust pearson, diam. 0.25, min size 10 6 25
QT-Clust pearson, diam. 0.25, min size 10 7 26
QT-Clust pearson, diam. 0.25, min size 10 8 28
QT-Clust pearson, diam. 0.25, min size 10 9 30
QT-Clust pearson, diam. 0.25, min size 10 10 33
QT-Clust pearson, diam. 0.25, min size 10 11 33
QT-Clust pearson, diam. 0.25, min size 10 12 39
QT-Clust pearson, diam. 0.25, min size 10 13 282

regularized variance selection 1000
• DynamicPL: [contained in global] AND [Regularized variance >= 0.06847]

global 22779
• Global This is a global probe list. It was extracted from the user master table and contains all probes contained in the active data set at the time of creation.

1355 MB free of 1820 MB total

Figure 17: QT-Clustering produces a set of colored probe list corresponding to the result of clustering analyis. All genes of a cluster are grouped in a separate probe list.

Multi profile plot

The results of a clustering algorithm like *k-means* and QT-Clustering can be nicely presented as a multi profile plot which combines profiles of several probe lists in a single window. In contrast to the profile plot shown in Figure 6, gene expression profiles of every selected probe list are plotted in separate coordinate systems.

k-means produces k new probe lists which contain the clustered genes. To create a multi profile plot, select all of them, right-click and select `Visualization` → `Multi Profile Plot`. The result is presented in Figure 18 which shows a multi profile plot of selected probe lists produced by *k-means* clustering using *Pearson correlation distance*.

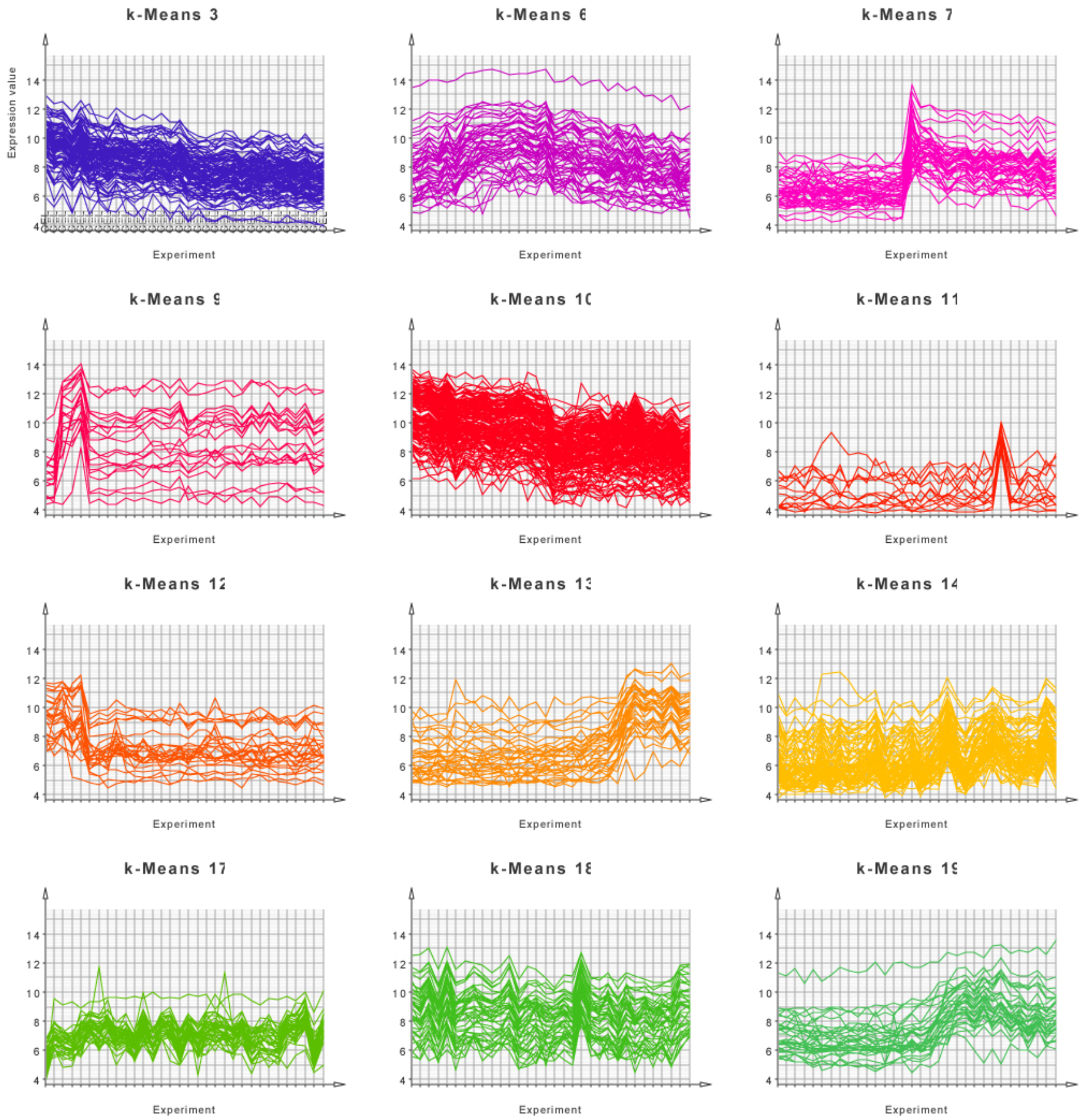


Figure 18: A Multi Profile Plot of selected clusters calculated by *k-means* with $k = 21$ and *Pearson correlation distance.*

Clustering assessment using the silhouette plot

The clustering algorithms presented above, especially *k-means*, calculate clusters which respect the selected preferences like distance measure and number of clusters but do not give any guarantee to produce always a good or correct cluster. Hence, quality of created clusters needs to be reviewed. Assessing the quality of clustering results can be done visually using `Profile Plot` (double-click on preview), `Multi Profile Plot` (right-click, `Visualization` → `Multi Profile Plot`, only makes sense if more than one cluster is selected) and `Scatter Plot` (right-click, `Visualization` → `Scatter Plot`), but Mayday provides algorithmic ways as well.

So-called *silhouette values* represent the distance to all cluster centroids for every single gene. If a gene is relatively close to the centroid of its associated cluster, its silhouette value will be positive. If the distance between a gene and an adjacent cluster centroid is

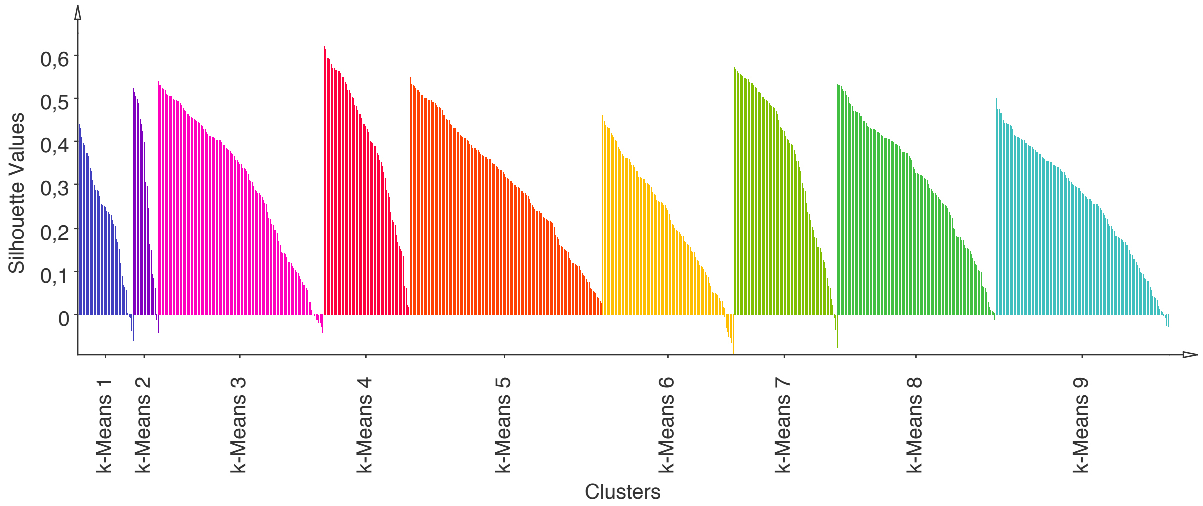


Figure 19: A silhouette plot can be used for visually assessing clustering quality. In general, positive silhouette values represent good cluster affinity, negative silhouette values are an indication, that the respective gene could be clustered better. Here, a *k-means* clustering with $k = 9$ and euclidean distance was evaluated.

smaller than to its own, the silhouette value will be negative. In optimal case, none of the silhouette values is below zero thus all genes are correctly clustered. A so-called *silhouette plot* visualizes the silhouette values for all selected clusters.

To produce a silhouette plot like the one in Figure 19, again select all probe lists created by a partitioning clustering algorithm (in case of QT-Clustering except probe list `Unclustered`), right-click and select `Visualization` → `Silhouette Plot` which visualizes silhouette values for every object in all selected clusters. If distance measure used for clustering is not euclidean, the correct distance measure needs to be set via `View` → `Distance Measure`.

Figure 19 shows a quite good clustering where only few genes have negative silhouette values. In case of many negative silhouette values, a different distance measure and, using *k-means*, a different k should be selected to produce a better clustering.

5 Data visualization

An important way to understand and present the results of data analysis is a graphical visualization. Besides the already presented plots, many different kinds of visualization such as histograms, profile plots, scatter plots, Venn diagrams, and a number of statistical plots can be produced. They can be adapted individually, combined and exported. In the following sections some essential visualizations using the results of the F199 analysis are described.

5.1 Histogram

The frequency distribution of values can be visualized using a histogram, where the frequency of values within certain intervals is plotted. In Mayday, histograms from probe lists and MIOs can be created. To produce a histogram, right-click on a probe list and select `Visualization` → `Histogram`. The values for visualization (experiment data or meta information) can be selected via `View` → `Values`. The resolution, that means the number of intervals, can be adjusted via `View` → `Resolution`. Figure 20 shows

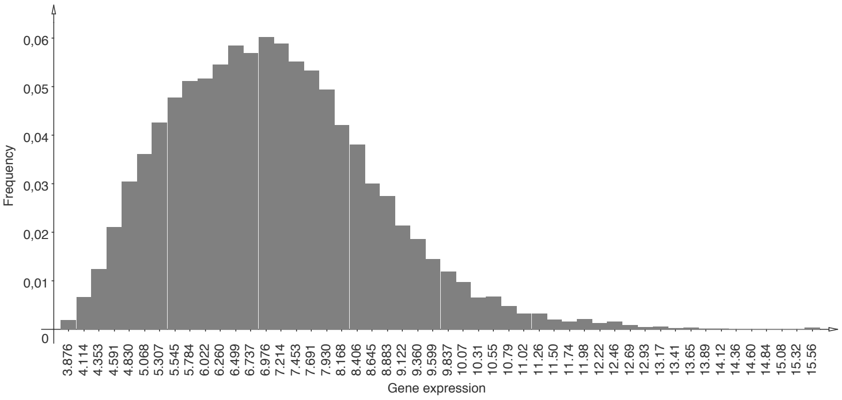


Figure 20: A histogram shows the relative frequency of values. Here, the frequency of expression values of the first experiment in F199 is plotted with a resolution of 50 intervals.

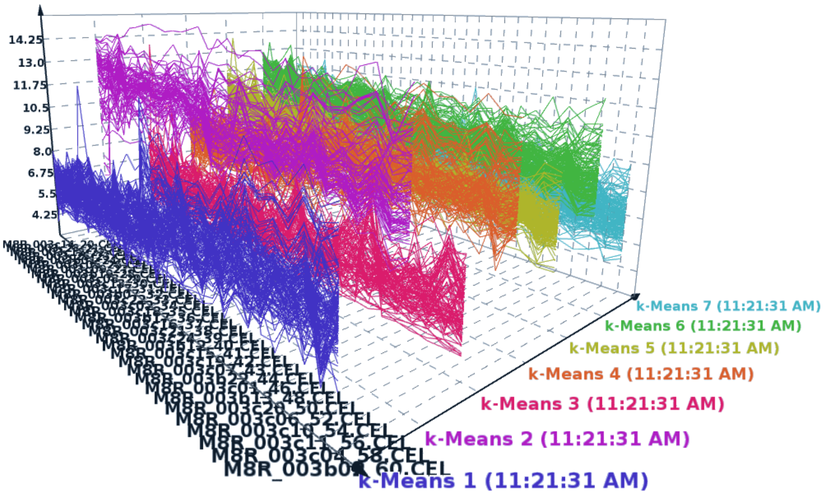


Figure 21: Expression profiles can be visualized as 3D multi profile plot. Here, a *k-means* clustering ($k = 7$, distance measure = euclidean) of the 1000 most variant genes is visualized.

the relative frequency of the logarithmized gene expression measured by the first F199 experiment divided into 50 intervals (resolution = 50).

5.2 Three dimensional plots

All of the plots described yet, do enable a two-dimensional data visualization only. A visual representation of higher dimensional data is hard to create. To overcome this problem, some 3D visualizations such as 3D multi profile plots and 3D scatter plots are available. Zooming in and out as well as a free rotation of the 3D plots is enabled. Right-click on a probe list and select `Visualization` to find a series of three dimensional visualizations. Figure 21 shows a 3D multi profile plot and a multi scatter plot of the gene expression from all experiments in F199 is shown in Figure 22.

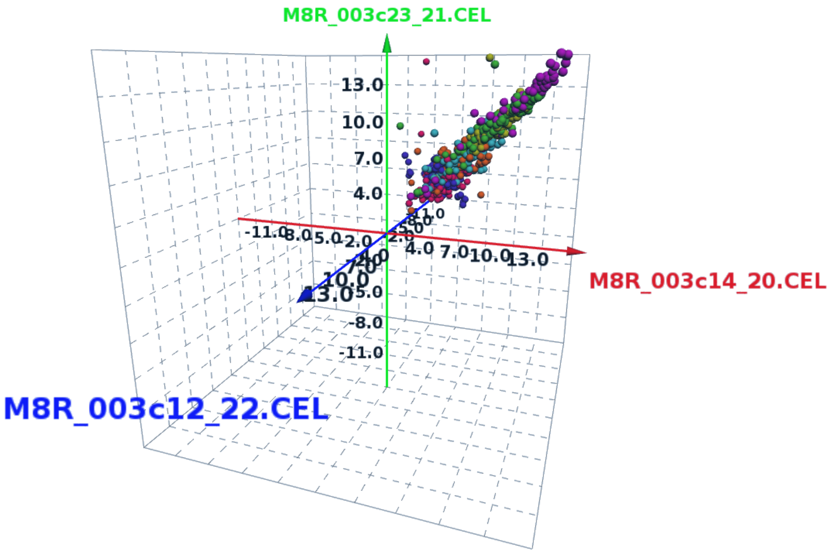


Figure 22: A 3D scatter plot visualizes the 1000 most variant genes of the first 3 experiments clustered with *k-means* using $k = 7$ and euclidean distance.

5.3 Tables

Besides the statistical and visual data representation, MAYDAY provides direct access to the values in a probe list. For that, right-click on a probe list and select `Visualization` → `Expression Matrix` or `Distance Matrix` to display the respective values as a table. Moreover, a meta information table, percentile table and a sample information table can be visualized.

5.4 Dealing with plots

MAYDAY tries to find the optimal presentation for every plot. Nevertheless, one might want to adapt them individually. For that, MAYDAY provides various settings. Click on `View` → `Detach menu` to add a legend and captions, change chart settings or add individual colors. To improve visualizations e.g. for use in publications, the `Visualizer` menu provides features to add and combine several plots in a single window.

The scaling of plots can be adjusted using `CTRL + mouse wheel` to zoom and `CTRL + Shift + mouse wheel` to adjust only vertical zoom. With `ALT + mouse wheel`, horizontal zoom only can be adjusted.

6 Further Mayday features

The following sections describe further features of MAYDAY which can be helpful for data analysis.

6.1 Saving results of analysis

MAYDAY can export the whole workspace (project) as snapshot, which saves all loaded datasets and its containing probe lists as well as MIOs. It is recommended to save a snapshot after data import using SEASIGHT to avoid repeating the pre-processing steps.

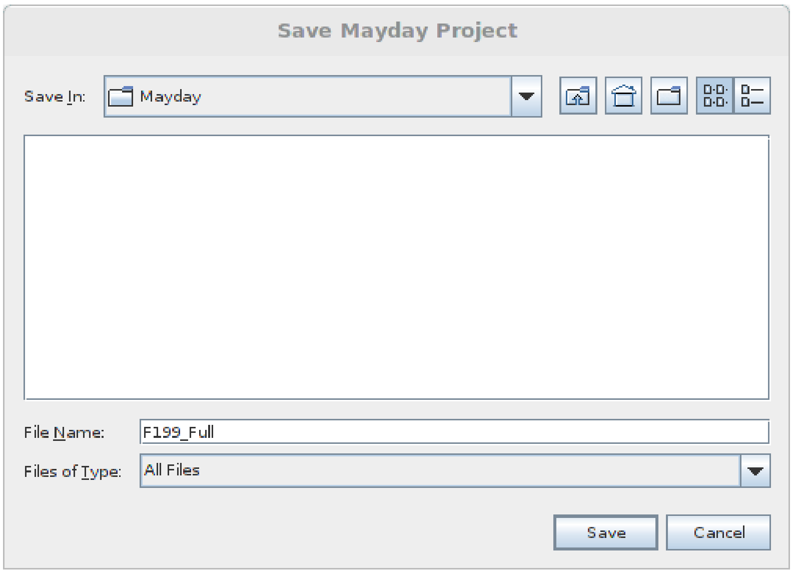


Figure 23: To save a MAYDAY project, select a file path on your disc and a name and confirm with `Save`. The project will be saved as `.maydayz` file.

To create a new snapshot of the whole workspace, navigate to `File` → `Save as`, and enter a file name. Clicking on `Save` will export the workspace as `.maydayz` file (see Figure 23).

To save a single datasets, `DataSet` → `Export to file` will open a new window where the selected dataset can be exported as `.maydayz` file. MAYDAY provides the direct export to a `.csv` table file format that can be read by many other programs. For that, select the file format `Tabular Text file [csv]` in the `Export DataSet` window. Analogously, a probe list can be exported as hierarchical probe list (`.pl`) and other formats via `Probe List` → `Export to file`. Section 3.1 describes, how the exported data can be loaded.

6.2 Export graphics

MAYDAY does not store open plot windows when the workspace is closed or saved as a snapshot. To keep visualized results, most plots can be exported as different file formats with `Plot` → `Export`. In `Graphics Export Settings` window, file format, ratio and anti-aliasing settings can be adjusted. For low quality plots, the file formats `JPEG` and `PNG` will work, better results can be created using the `PDF` format or the vector graphic format `SVG`.

6.3 Export tables

MAYDAY can create new files containing subsets of the whole dataset. Selections, probe lists and MIOs can be exported as separate `.csv` files. This might be useful to extract subsets of informations for further processing with MAYDAY or other platforms. In many views, subsets of probes can be selected and exported by navigating to `Selection` → `Run ProbeList plugin` → `Export to file`.

6.4 Selection menu

For an easy and intuitive interaction with large datasets, MAYDAY provides the `Selection` menu in the menu bar of visualization windows holding many options to work with selected probes. Different visualizer can be synchronized so that selecting genes in one visualizer will also highlight these genes in the other visualizer (`Selection` → `Synchronize selection with Visualizer`). Furthermore, the `Selection` menu provides to invert the selection (`Selection` → `Invert`), to create a new probe list containing the selected probes or to sent them to another probe list (`Selection` → `Send selection to ProbeList`).

For demonstration purpose, we will sent some selected probes from a heatmap to a profile plot:

- Open a heatmap visualization as described in Section 4.7.
- Make a selection of some probes, for example all genes that are down-regulated at hour 35.
- Click on `Selection` → `Plot in separate visualizer` to create a profile plot of the selected probes.

The result is presented in Figure 24. This is very useful to compare behavior of genes in different views.

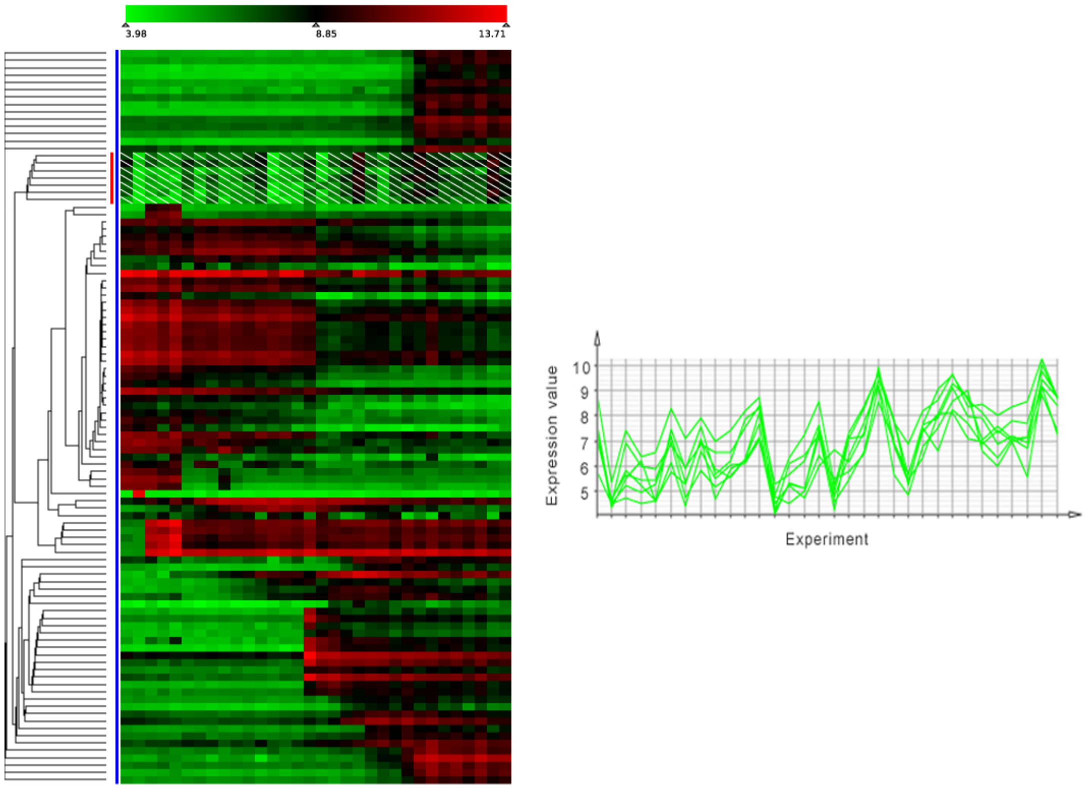
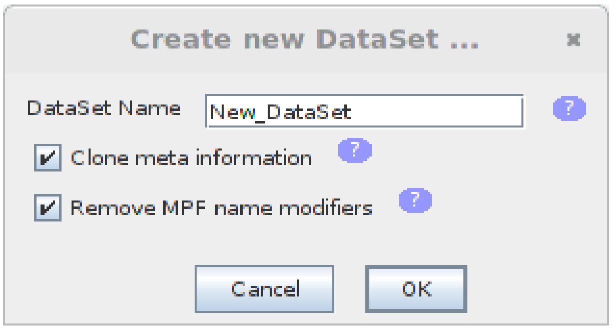


Figure 24: Here, some selected genes from the heatmap are visualized as profile plot. The red label and the white grid in the heatmap mark the selected genes, whose expression profiles are plotted on the left side.

6.5 Move data between datasets

Mayday holds the loaded data in separated datasets. Data can be sent to other opened or new datasets. For that, right-click on a probe list and select `Further export options` → `Sent to DataSet` or `Create new DataSet`. Alternatively, use the menu bar item `ProbeList`. A new window will open where the target dataset or the name of the new dataset can be selected.



DataSet Name: New_DataSet

☑ Clone meta information

☑ Remove MPF name modifiers

Figure 25: When moving data to a new dataset, in the window, the name of the new dataset can be selected.

References

[BN11] Florian Battke and Kay Nieselt. Mayday seasight: combined analysis of deep sequencing and microarray data. *PLoS One*, 6(1):e16345, 2011.

[BSN10] Florian Battke, Stephan Symons, and Kay Nieselt. Mayday-integrative analytics for expression data. *BMC bioinformatics*, 11(1):121, 2010.

[Car] Stephen M. Carr. UPGMA and WPGMA. `http://www.mun.ca/biology/scarr/UPGMA_vs_WPGMA.htm`. 30.04.2015.

[dLPdSM+12] Rudi Emerson de Lima Procópio, Ingrid Reis da Silva, Mayra Kassawara Martins, João Lúcio de Azevedo, and Janete Magali de Araújo. Antibiotics produced by streptomyces. *The Brazilian Journal of infectious diseases*, 16(5):466–471, 2012.

[EKSX96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[GDN05] Nils Gehlenborg, Janko Dietzsch, and Kay Nieselt. A framework for visualization of microarray data and integrated meta information. *Information Visualization*, 4(3):164–175, 2005.

[Hay13] Winston Haynes. Student's t-test. In *Encyclopedia of Systems Biology*, pages 2023–2025. Springer, 2013.

[HKY99] Laurie J Heyer, Semyon Kruglyak, and Shibu Yooseph. Exploring expression data: identification and analysis of coexpressed genes. *Genome research*, 9(11):1106–1115, 1999.

[JIC] John innes centre: Streptomyces genome. `https://www.jic.ac.uk/science/molmicro/Strept.html`. 16.04.2015.

[Joh67] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[Koh90] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[M+67] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[MTL78] Robert McGill, John W Tukey, and Wayne A Larsen. Variations of box plots. *The American Statistician*, 32(1):12–16, 1978.

[NBH+10] Kay Nieselt, Florian Battke, Alexander Herbig, Per Bruheim, Alexander Wentzel, Øyvind M Jakobsen, Håvard Sletta, Mohammad T Alam, Maria E Merlo, Jonathan Moore, et al. The dynamic architecture of the metabolic switch in streptomyces coelicolor. *BMC genomics*, 11(1):10, 2010.

[SMP08] Martin Simonsen, Thomas Mailund, and Christian NS Pedersen. Rapid neighbour-joining. In *Algorithms in Bioinformatics*, pages 113–122. Springer, 2008.