

# **A Guide to the LCB-DWH**

**Computer Exercise**

**By**  
**The Linnaeus Centre for Bioinformatics**  
**and**  
**The WCN Expression Array Facility**  
**at Uppsala University**

# Table of Contents

<b>Webpages</b> .....	<b>2</b>
<b>Account free access Quick Start Guide</b> .....	<b>3</b>
<b>Introduction</b> .....	<b>3</b>
<b>LCB Data Warehouse</b> .....	<b>5</b>
The experiment.....	5
Add experiment .....	5
Add Raw data sets .....	5
Create BioAssaySet .....	6
<b>Array plots</b> .....	<b>8</b>
<b>PCA</b> .....	<b>10</b>
<b>Normalization</b> .....	<b>11</b>
Background correction.....	11
Normalize within arrays.....	11
Normalize between arrays.....	14
<b>Filtering</b> .....	<b>15</b>
Create filters .....	15
Spot filter.....	15
Merge spots .....	16
Gene filter .....	16
<b>Statistical analysis</b> .....	<b>18</b>
B-statistics .....	19
<b>Experiment Explorer</b> .....	<b>22</b>
<b>SAM</b> .....	<b>24</b>
<b>Gene Ontology (GO)</b> .....	<b>27</b>
The GO statistics .....	27
The GO plot tool.....	30
<b>Visualization</b> .....	<b>31</b>
Hierarchical clustering.....	31
K-means clustering .....	32
<b>Reporter list</b> .....	<b>34</b>
<b>References</b> .....	<b>36</b>

## Webpages

<b>BASE</b>	<a href="https://base.lcb.uu.se">https://base.lcb.uu.se</a>
<b>LCB Data Warehouse</b>	<a href="https://dw.lcb.uu.se">https://dw.lcb.uu.se</a>
<b>WCN Array Platform</b>	<a href="http://www.wcn.se">http://www.wcn.se</a>

# Account free access Quick Start Guide

There are two options for testing the functionality of the LCB-DWH system without being a registered user. Those that only want to get a quick overview:

- Log in as the demo user (user name: demo, password: demo)
- Click on 'Analyze data', then 'Experiments'
- Then click on the experiment 'DEMO MA-Course May2005'
- After that you will see all results from the analysis described in this document.

The demo user can only look at results and is not allowed to run any analysis. If you instead want to execute the analysis tools on pre-uploaded data, then:

- Log in as the 'test' user (user name: test, password: test)
- Click on 'Analyze data', then 'Experiments'
- Either choose one of the available experiments or create a new experiment as described below.
- Try out the analysis tools. It may be suitable to follow the outline of this document.

## Introduction

The laboratory exercise will give an introduction to the analysis of DNA microarray expression data, in particular cDNA microarray data. Please see the references for reviews on the microarray technology and analysis (Howell 1999; Hess, Zhang et al. 2001; Butte 2002).

The course participants will be given hands-on experience in handling the data from microarray experiments. The wet lab part will not be covered.

The computer exercise will first of all cover the process of normalization. The purpose of normalization is to identify and remove systematic variation present in microarray data. It is an essential step in the analysis process, to make sure the differences in measured intensities on the microarrays are indeed due to differential expression, i.e. true biological variation.

The analysis will also include statistical tests for identification of genes that are differentially expressed between for example different conditions or tumour classes. The data can also be visualised using clustering or further analysed using multivariate techniques as classification.

During the lab we will also investigate the results and interpret them, for example by using the newly developed GO (Gene Ontology) tools at the Linnaeus Centre.

## Biological samples

The data set we will use today consists of 8 arrays. The arrays are four pairs of dye swaps, there are two technical replicates for each biological sample. All the experiments were performed at the Wallenberg Consortium North Expression Array Platform in Uppsala, located at the Rudbeck laboratory.

The biological samples in this case are bone marrow samples from ALL patients and they can be divided into two groups depending if they are of B-cell origin or if they have a T-cell origin.

In the clinic these origins can be detected using molecular markers, so in this practical we will benefit from the fact that we know some things we are likely to find when trying to find genes differentially expressed between the two groups.

Our samples are named as follows:

<b>98/365</b>	<b>T-cell origin</b>
<b>00/134</b>	<b>T-cell origin</b>
<b>98/278</b>	<b>B-cell origin</b>
<b>98/2</b>	<b>B-cell origin</b>

## **Hybridizations**

These four ALL samples are hybridized to a common reference in a so called indirect comparison, meaning that all the samples in the two groups are hybridized and compared to a common reference.

The information about these samples and the experimental procedure has already been added to our local LIMS BASE, which is a convenient, secure and reliable way of storing microarray results. These steps are normally performed by the user in the same way as using a lab book to document laboratory information, but in this exercise we focus on the analysis and will therefore skip this first part.

We will start from the analysis part of the process. The initial dataset you will see contains the four dye-swapped microarray experiments in the order as explained above, the first four assays describe the T-cell samples and the last four describe the B-cell samples.

# LCB Data Warehouse

Point your browser to <https://dw.lcb.uu.se>.

Log in using the Username and Password you received.

## The experiment

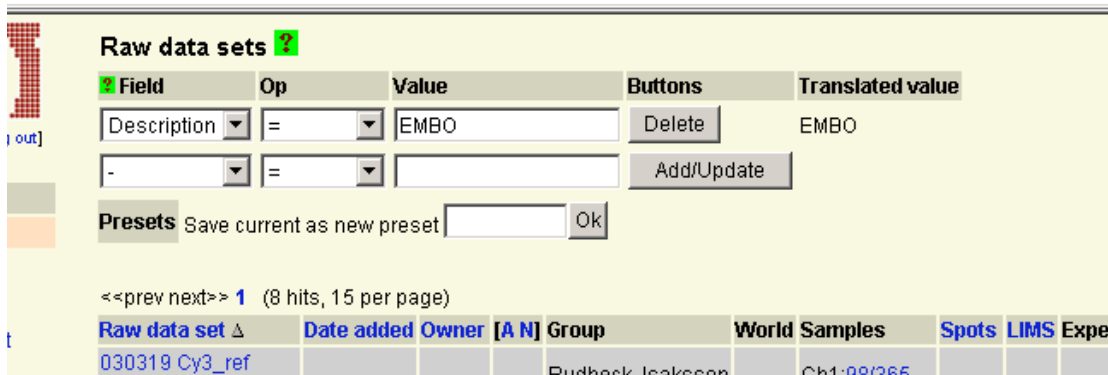
First we will ask you to make your own copy of the experiment described above, since there is only one original **CourseExperiment\_ALL**. To the experiment you create you will add the same datasets present in our original experiment, so that you will end up with an exact copy of that experiment.

## Add experiment

Click on “**Analyze data**”, “**Experiments**” and “**Add Experiment**”. Type a suitable name including your username to distinguish your experiment from the other course participants’. Press “**Accept**”.

## Add Raw data sets

Click on “**Raw data sets**” in the left menu. Since there are a lot of raw data sets available you can try to filter out the ones we will use for this lab by typing the following:



The screenshot shows the 'Raw data sets' interface. At the top, there is a search filter with the following fields:

Field	Op	Value	Buttons	Translated value
Description	=	EMBO	Delete	EMBO
-	=		Add/Update	

Below the search filter, there is a 'Presets' section with a text input field and an 'Ok' button.

At the bottom, there is a table of results with the following columns: Raw data set, Date added, Owner, [A N], Group, World, Samples, Spots, LIMS, and Expe. The first row of data is:

Raw data set	Date added	Owner	[A N]	Group	World	Samples	Spots	LIMS	Expe
030319 Cy3_ref				Rudbeck Isaksson		Ch1:98/365			

You can now see *only the 8 bioassays* which will be used in the lab (the same 8 bioassays present in the original data set). Check all the checkboxes (or simply click the capital A in the header) and in the box after the line “marked raw data sets to/from experiment”, select your previously constructed experiment.

These 8 assays should be:

<<prev next>> 1 (8 hits, 15 per page)									
Raw data set $\Delta$	Date added	Owner	[A N]	Group	World	Samples	Spots	LIMS	Experiments
<a href="#">030319 Cy3_ref Cy5_98365 BP8_s56_1246347</a>	2003-03-31	marten	<input checked="" type="checkbox"/>	Rudbeck, Isaksson (rw)	--	Ch1:98/365, Ch2:Ref (ALL)	15552	No	hanna:CourseExperiment
<a href="#">030319 Cy3_98365 Cy5_ref BP8_s58_1246347 #2</a>	2003-07-11	maria	<input checked="" type="checkbox"/>	Rudbeck, Isaksson (rw)	--	Ch1:98/365, Ch2:Ref (ALL)	15552	No	hanna:CourseExperiment
<a href="#">030313 Cy3_ref Cy5_00134 BP8_s49_1242695</a>	2003-03-27	marten	<input checked="" type="checkbox"/>	Rudbeck, Isaksson (rw)	--	Ch1:00/134, Ch2:Ref (ALL)	15552	No	hanna:CourseExperiment
<a href="#">030313 Cy3_00134 Cy5_ref BP8_s50_1242698 #2</a>	2003-07-14	maria	<input checked="" type="checkbox"/>	Rudbeck, Isaksson (rw)	--	Ch1:00/134, Ch2:Ref (ALL)	15552	No	hanna:CourseExperiment
<a href="#">030227 Cy3_ref Cy5_98278 BP8_s31_1242699</a>	2003-03-31	marten	<input checked="" type="checkbox"/>	Rudbeck, Isaksson (rw)	--	Ch1:98/278, Ch2:Ref (ALL)	15552	No	hanna:CourseExperiment
<a href="#">030227 Cy3_98278 Cy5_ref BP8_s32_1242698 #2</a>	2003-07-14	maria	<input checked="" type="checkbox"/>	Rudbeck, Isaksson (rw)	--	Ch1:98/278, Ch2:Ref (ALL)	15552	No	hanna:CourseExperiment
<a href="#">030226 Cy3_ref Cy5_982 BP8_s29_12458863</a>	2003-03-31	marten	<input checked="" type="checkbox"/>	Rudbeck, Isaksson (rw)	--	Ch1:98/2, Ch2:Ref (ALL)	15552	No	hanna:CourseExperiment
<a href="#">030226 Cy3_982 Cy5_ref BP8_s30_12458864 #2</a>	2003-07-14	maria	<input checked="" type="checkbox"/>	Rudbeck, Isaksson (rw)	--	Ch1:98/2, Ch2:Ref (ALL)	15552	No	hanna:CourseExperiment

Click on “Go”. Note that the “Experiments” column is updated with your experiment name.

## Create BioAssaySet

Click on “Experiments” in the left tab. Click on your newly created experiment. Check all the checkboxes (or simply click the capital A in the header) and type in a suitable name in the box after “Create new BioAssaySet called”. Select “Median FG” to calculate the ratios using the median of the foreground signal for the uploaded intensity values from all the spots.

A BioAssaySet is more or less a data matrix where all raw data sets are concatenated. Think of it as a tab-delimited text file with genes (reporters) on the rows and the samples (arrays) on the columns. This is the data matrix we will use for analysis. By allowing the creation of multiple BioAssaySets, the LCB Datawarehouse makes it possible to easily remove or add new BioAssays to a BioAssaySet and thereby compare the results from different analyses.

Raw data set $\Delta$	Date added	Owner	[A N]	Group	World	Samples	Spots	LIMS	Experiments
<a href="#">030319 Cy3_ref Cy5_98365 BP8_s56_1246347</a>	2003-03-31	marten	<input checked="" type="checkbox"/>	Rudbeck, Isaksson (rw)	--	Ch1:98/365, Ch2:Ref (ALL)	15552	No	hanna:CourseExpe
<a href="#">030319 Cy3_98365 Cy5_ref BP8_s58_1246347 #2</a>	2003-07-11	maria	<input checked="" type="checkbox"/>	Rudbeck, Isaksson (rw)	--	Ch1:98/365, Ch2:Ref (ALL)	15552	No	hanna:CourseExpe
<a href="#">030313 Cy3_ref Cy5_00134 BP8_s49_1242695</a>	2003-03-27	marten	<input checked="" type="checkbox"/>	Rudbeck, Isaksson (rw)	--	Ch1:00/134, Ch2:Ref (ALL)	15552	No	hanna:CourseExpe
<a href="#">030313 Cy3_00134 Cy5_ref BP8_s50_1242698 #2</a>	2003-07-14	maria	<input checked="" type="checkbox"/>	Rudbeck, Isaksson (rw)	--	Ch1:00/134, Ch2:Ref (ALL)	15552	No	hanna:CourseExpe
<a href="#">030227 Cy3_ref Cy5_98278 BP8_s31_1242699</a>	2003-03-31	marten	<input checked="" type="checkbox"/>	Rudbeck, Isaksson (rw)	--	Ch1:98/278, Ch2:Ref (ALL)	15552	No	hanna:CourseExpe
<a href="#">030227 Cy3_98278 Cy5_ref BP8_s32_1242698 #2</a>	2003-07-14	maria	<input checked="" type="checkbox"/>	Rudbeck, Isaksson (rw)	--	Ch1:98/278, Ch2:Ref (ALL)	15552	No	hanna:CourseExpe
<a href="#">030226 Cy3_ref Cy5_982 BP8_s29_12458863</a>	2003-03-31	marten	<input checked="" type="checkbox"/>	Rudbeck, Isaksson (rw)	--	Ch1:98/2, Ch2:Ref (ALL)	15552	No	hanna:CourseExpe
<a href="#">030226 Cy3_982 Cy5_ref BP8_s30_12458864 #2</a>	2003-07-14	maria	<input checked="" type="checkbox"/>	Rudbeck, Isaksson (rw)	--	Ch1:98/2, Ch2:Ref (ALL)	15552	No	hanna:CourseExpe

marked raw data sets to/from experiment

Update marked: Group:  with  access. World:  access.

Create new BioAssaySet  called  from

Click “Go” on the last line and you should see the following screen:

**Experiment hanna:CourseExperiment\_ALL**

[Return](#)

[Info](#) [Raw data sets](#) [Analysis steps](#) [Reporter lists](#)

Processing request - please have patience.  
 Processing '030319 Cy3\_ref Cy5\_98365 BP8\_s56\_1246347'...  
 Processing '030319 Cy3\_98365 Cy5\_ref BP8\_s58\_1246347 #2'...  
 Processing '030313 Cy3\_ref Cy5\_00134 BP8\_s49\_1242695'...  
 Processing '030313 Cy3\_00134 Cy5\_ref BP8\_s50\_1242698 #2'...  
 Processing '030227 Cy3\_ref Cy5\_98278 BP8\_s31\_1242699'...  
 Processing '030227 Cy3\_98278 Cy5\_ref BP8\_s32\_1242698 #2'...  
 Processing '030226 Cy3\_ref Cy5\_982 BP8\_s29\_12458863'...  
 Processing '030226 Cy3\_982 Cy5\_ref BP8\_s30\_12458864 #2'...  
 Counting spots per reporter and position...  
 Counting spots, reporters, assays...

**BioAssaySet 'ALL'  [Close]**

[Edit](#) [Delete](#)

<b>Name</b>	ALL
<b>Description</b>	
<b>Creation date</b>	2004-05-05
<b>BioAssays</b>	8
<b>Created from raw</b>	Median FG
<b>Total # of values</b>	124416
<b>Total # of reporters</b>	7651
<b>Extra columns</b>	No

The total number of reporters is 7651. This is the number of unique genes present on each microarray. The number of spots is equal to all occurrences of these genes on the eight arrays; in this case they are printed in duplicate on each array.

# Array plots

This is a plotting tool to visualize your data in LCB DWH. To apply it click on “**Run app**” (*Run Application*) below the raw data sets in the hierarchical overview table to execute a plugin.

Select “**Visualization: Plots v4.0**”, which is a plugin written in R at LCB to display several different plots of individual arrays. This plug-in uses various R/Bioconductor ([www.bioconductor.org](http://www.bioconductor.org) for further information) functions to construct the following plots of two-channel micro arrays:

- 1) Intensity plots
- 2) Background plots
- 3) M-value plots
- 4) Flag plots
- 5) MA-plots with loess curves for each print-tip group
- 6) Box plots for each print tip group
- 7) Density plots
- 8) Overall plot

The option 'Make reduced number of plots' only creates the plots 1, 3, 7 and 8 together with MA-plots (without loess curves for print-tip groups). This option may be useful in later stages of analysis, for example after the data has been merged.

<b>Plug-in</b>	Visualization: Plots (Bioconductor)																					
<b>About plug-in</b>	<p>This plug-in uses various R/Bioconductor functions to construct the following plots of two-channel micro arrays:</p> <ol style="list-style-type: none"><li>1) Intensity plots</li><li>2) Background plots</li><li>3) M-value plots</li><li>4) Flag plots</li><li>5) MA-plots with loess curves for each print-tip group</li><li>6) Box plots for each print tip group</li><li>7) Density plots</li><li>8) Overall plot</li></ol> <p>The option 'Make reduced number of plots' only creates the plots 1, 3, 7 and 8 together with MA-plots (without loess curves for print-tip groups). This option may be useful in later stages of analysis, for example after the data has been merged.</p> <p>If array layout has not been specified for this data set, such information can be given as an argument to the plug-in. The arrays have the same layout.</p>																					
<b>Job name</b>	Visualization: Plots (Bioconductor)																					
<b>Description</b>	<input type="text"/>																					
<b>Parameter</b>	<table border="1"><thead><tr><th>Parameter</th><th>Type</th><th>Value</th></tr></thead><tbody><tr><td>Make reduced number of plots?</td><td>Enum</td><td>no</td></tr><tr><td>Specify array layout dimensions?</td><td>Enum</td><td>no</td></tr><tr><td>Grid rows per slide</td><td>Float</td><td>0</td></tr><tr><td>Grid columns per slide</td><td>Float</td><td>0</td></tr><tr><td>Spot rows per print-tip group</td><td>Float</td><td>0</td></tr><tr><td>Spot columns per print-tip group</td><td>Float</td><td>0</td></tr></tbody></table>	Parameter	Type	Value	Make reduced number of plots?	Enum	no	Specify array layout dimensions?	Enum	no	Grid rows per slide	Float	0	Grid columns per slide	Float	0	Spot rows per print-tip group	Float	0	Spot columns per print-tip group	Float	0
Parameter	Type	Value																				
Make reduced number of plots?	Enum	no																				
Specify array layout dimensions?	Enum	no																				
Grid rows per slide	Float	0																				
Grid columns per slide	Float	0																				
Spot rows per print-tip group	Float	0																				
Spot columns per print-tip group	Float	0																				
	<input type="button" value="Start job"/>																					

The Grid rows/columns per slide define the print tip block layout and Spot rows/columns the spot layout within a print-tip group. In our case, there is no need to specify the array layout since this information has already been entered into the database. For your information the layout is 4x12 blocks (print-tip groups) and 18x18 spots within each print-tip group.

Press **“Start job”** to execute the plugin with these parameters.

LCB DWH will respond “The application was successfully started”. Press “Check the status of the job” and see in the status row that the application is running. This operation will take a couple of minutes, and you may see the status by pressing the [refresh] option in the status row. When the status has changed to **“All done”**, the resulting plots can be viewed in the **“Results”** file.

Press **“View”** on the results file and you will get an html page with a lot of plots which will be described below. Move the mouse over the blue links on the left side of the page to see the images for the eight arrays one at a time.

**Image plot** - This plot shows the array divided into print-tip groups.

**Background plot** - This is a plot of the background intensities.

**MA-plot with print-tip group intensities** - This plot shows the M values ( $\log_2(R/G)$ ) against the A values ( $\log_2(\sqrt{R*G})$ ) with the lines of the individual print-tip groups.

**Box-plot** - This plot describes the mean values with 50% distribution (boxes), 90% of distribution (ending lines) and outliers (circles) for individual arrays.

**Density plot** - This plot shows the density distribution.

**Overall box-plot** - This plot compares the mean ratio between all arrays.

# PCA

This is another plotting tool to visualize your data in LCB DWH. Principal Component Analysis is a method used to reduce the dimension of that data through finding directions with maximum variance, and creating a new coordinate system defined by the two directions with most variance. This tool produces a plot of all arrays in this 2D coordinate system. Arrays that are close in this 2D plot are likely to contain similar data. Therefore, arrays with replicate measurements should ideally group together in the plot. However, this is not always the case, especially before data has been normalized. In general, this tool can be used to investigate which sources of variation there is in the dataset and if it is necessary to perform any additional processing of the data, i.e. normalization in this case, in order to remove or at least reduce the unwanted source of variation.

To apply the plug-in to your data sets make sure the dataset is marked and then click on **“Run app” (Run Application)** below the raw data sets in the hierarchical overview table to execute the plug-in.

Select **“Visualization: PCA v4.0”**.

This plug-in performs a Principal Component Analysis (PCA) of microarray data and produces the following two plots:

- 1) Histogram with variances of the first few principal components.
- 2) 2D-plot of all samples projected onto the plane spanned by the two first principal components.

Both array plots and PCA can be used for comparison between before and after normalization and also between the different normalization methods.

## Normalization

We will now normalize the data within individual arrays to try to compensate for systematic errors in the printing/hybridization or any other part of the experimental procedure. The aim is to center the data on zero, since one of the fundamental assumptions for applying this type of normalization is that the expression levels of most genes will remain unchanged. Genes that are unchanged should have a mean value of the ratios of 1, which is 0 on a  $\log_2$  scale.

### Background correction

First of all one usually tries to separate the foreground signal from the background signal in the raw data. We believe this to have an influence on our results, to what extent is depending on the slides and the dataset.

Background correction can be done through estimating the background using simple subtraction of the mean or median value of the local background (just outside the spot) or using more complex measurements that makes an estimate taking the spatial variation across the array into account

If the estimation of the background signal is done in a non-optimal way the process will actually introduce more variation into the system, so it is important to investigate the consequences of this step before proceeding.

For now, we will continue *without subtracting the background*. Feel free to go back and perform background subtraction later if you have time. In that case choose to **“Run app”**, **“Normalization: Background correction v4.0”** on the raw data. This plug-in is a collection of methods for background correction of two channel microarray data, try for instance to subtract the local median background.

### Normalize within arrays

The next step in the analysis procedure will attempt to remove the dye specific effects present in our data.

Choose **“Run app”** on the same BioAssaySet as before, but this time choose the plugin **“Normalization: Within arrays v4.0”**.

## Print-tip lowess normalization

Select **“Print-tip loess”** (local polynomial regression fitting for each print-tip group). Please see the references for more information about the lowess algorithm (Yang, Dudoit et al. 2002). The print-tip version of this normalization, which is a local fitting of the regression line, is usually preferred because of the spatial effects present on the array.

There is no need to specify the array layout in this plug-in either, since this information has already been entered into the database. For your information the layout is 4x12 blocks (print-tip groups) and 18x18 spots within each print-tip group.

There is also an option of assigning weights to your spots, depending on their quality. When exporting GenePix (software for image analysis) numerical data to spreadsheet programs spot flags appear as *Good* (100) or *Bad* (-100). If a feature cannot be found during an auto-alignment of the blocks, it is flagged *Not Found* (numerically reported as -50).

Quality weights may be associated with these flag values. A quality weight is a number in the interval [0, 1] that represents how much a spot is taken into account in the normalization. This plug-in makes it possible to give a comma separated string of flag values, for each of the weights 0, 0.25, 0.5 and 0.75. Weights that are not associated with flags get the default value 1.

For now we will leave these parameters at their default value, i.e. all the weights will be set to 1, and perform the first normalization. Feel free to go back and perform the normalization again later and change these values, for example down-weighting the bad spots (-100) to 0.25.

Press **“Start job”**, and wait for the normalization to finish.

Parameter	Type	Value
Normalization method	Enum	Print-tip loess
Specify array layout dimensions?	Enum	no
Grid rows per slide	Float	0
Grid columns per slide	Float	0
Spot rows per print-tip group	Float	0
Spot columns per print-tip group	Float	0
Associate quality weights with flag values?	Enum	no
Flag values with weight 0:	String	
Flag values with weight 0.25:	String	
Flag values with weight 0.5:	String	
Flag values with weight 0.75:	String	

Start job

Check the status of the job, click on [refresh] until status is changed to “**All done**”.

You can now also use “**Visualization: Plots v4.0**” again to create plots and look at your normalized data; using the same parameter settings as before. Look at the produced result-file “**Results.html**” and try to compare the plots with the plots of your raw data. Hopefully you will see that the data is now centered on zero.

*Has the normalization been successful?*

*Can you see any systematic variation in the normalized data?*

### **Global lowess normalization**

Now we will instead perform a global normalization using the same lowess algorithm as before.

Choose “**Run app**” on the same BioAssaySet with raw data as before, and again choose the plug-in “Normalization: Within arrays (Bioconductor)”. Select “**Global loess**”. Use the same parameter settings as above to be able to compare the results from this algorithm to the previous one.

Press “**Start job**”, and wait for the normalization to finish.

You can now run “**Visualization: Plots v4.0**” again to create plots and look at your normalized data; using the same parameter settings as before. Look at the produced result-file “**Results.html**” and try to compare the plots with the plots of your print-tip normalized data.

*Has the normalization been successful?*

*Can you see any systematic variation in the normalized data?*

*Can you see any differences between the two normalizations performed?*

If you for example compare the plot of the first array when using the print-tip version of lowess to using the global method, there are clear differences and you can see that the global method has not been successful in the sense that the data is not centered on zero.

You can also investigate your data by running the PCA plug-in again; use “**Run app.**”, and then **Visualization: PCA v4.0**”.

*Has the variation within the raw data has changed?*

*Is this due to the normalization and if so in what way and why?*

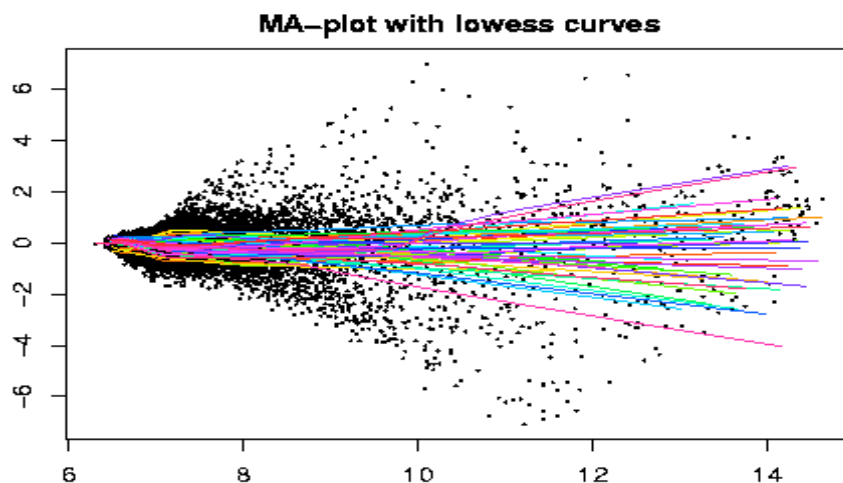


Fig.1 Global lowess normalization

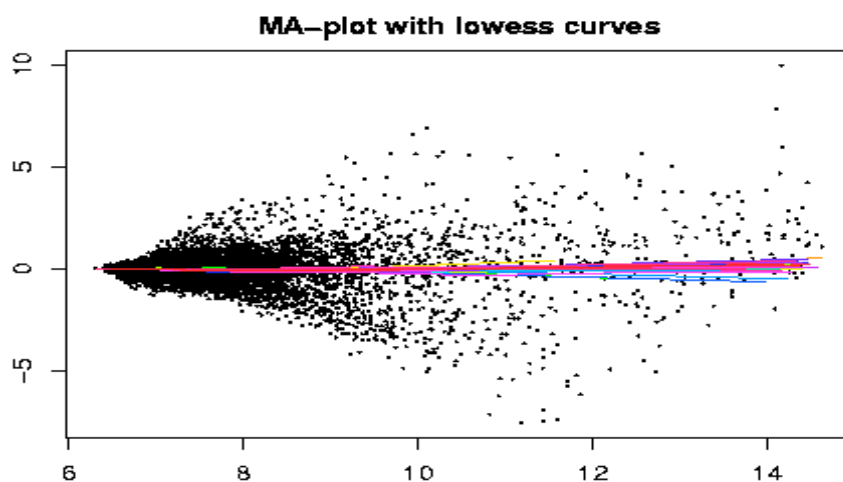


Fig.2 Print-tip lowess normalization

### Normalize between arrays

If we see substantial differences between the arrays, we can compensate for this with the plugin **“Normalization: Between arrays v4.0”**.

If you would like to try this then start with the previously created normalized BioAssaySet ending with “transf.” (transformed) as a default name, select **“Run app”**. Pick this plugin, run it and compare the graphs in **“Results.html”**. We have now scaled the arrays against each other. Usually this step is not necessary unless the range of intensities from the arrays is very different, but you can try it if you have time.

# Filtering

Now we have tried to remove the systematic variation present in our microarray data through the process of normalization, but of course we will still have lots of noise that is not containing any interesting information. Sometimes it can be useful to discard some of this, and this can be done through a process called filtering.

## Create filters

Choose (click on) the *data set normalized through print-tip lowess normalization*. Press the **“Filter”** button on the last row of the analysis procedure to select the normalized BioAssaySet.

## Spot filter

In the software used for image analysis, i.e. analysis to quantify the signals from the scanned microarray slide, there are often various options of flagging the spots depending on for example their morphology or signal/lack of signal. As we have seen earlier in this exercise these flags can be assigned with quality weights and used in the normalization but also in this process to actually discard certain measurements of gene expression levels.

We will now filter out all spots flagged as bad, absent or not found and therefore we should create a filter for this.

Below Spot Filter, select *Field* = **“[RAW] Flags”**, *Op (Operation)* = **“>=”**, *Value* = **“0”**. Press **“Add/Update”** on the same row, to only sort out the spots with the value **“>=0”** from the image software, thus are not marked with a flag as bad, absent or not found.

The screenshot shows a software window titled "Transformation: filter". At the top, there is a "Return" link and a "BioAssaySet" label indicating "ALL transf. (all of 8 assays selected)". Below this is a "Spot filter" section with a table for defining filters. The table has columns for "Field", "Op", "Value", "Buttons", and "Translated value". The first row is pre-filled with "[Raw] Flags", ">=", "0", a "Delete" button, and "0". The second row is empty with "-" in the "Field" column, ">" in the "Op" column, and an "Add/Update" button. Below the table is a "Presets" section with a text input field and an "Ok" button. A "Gene filter" section is also visible below the spot filter, with a similar table structure.

Field	Op	Value	Buttons	Translated value
[Raw] Flags	>=	0	Delete	0
-	>		Add/Update	

Scroll down and change the *Filtering name* to for example “**Flags>=0**” for convenience. Press “**Accept**”, wait for the filtering to complete, then press “**to the new BioAssaySet**”.

If you have time, feel free to try some different filters of your own, even though this may be difficult if you are not familiar with the outcomes of a microarray experiments.

## Merge spots

This plug-in merges all duplicate spots a set of two-channel micro arrays by computing the geometric mean of the two intensity channels. Each array is merged independently, and all spots that have the same Reporter ID are considered to be replicates.

Choose “**Run app**” on the resulting BioAssaySet from the filtering process, but this time choose the plug-in “**Miscellaneous: Merge spots**”. Press “**Continue**” and then “**Start job**”.

Since this calculation only merges the values from the different spots for the same gene, the number of reporters should still be the same as before.

## Gene filter

Now we will look at the data set after filtering away the flagged spots and merging the replicates.

We have filtered some spots, but it would be wise to also filter out the genes which are present on all or most arrays after the previous filter, i.e. where we still have values left after the filter above. Our next step after filtering is to perform some statistical analysis on our data and we would like to base these calculations on as many values as possible per gene, in order to end up with reliable results. Therefore, since we are only interested in genes that exist on many plates, we want to remove the ones where we have values on just a few arrays.

We will create and apply a **Gene filter**. In this tutorial we will require the genes to be present on 6 or more out of the 8 microarray slides in the experiment.

So, again press “**Filter**” on the last BioAssaySet.

Below Gene Filter, select *Field* = “**In # of assays**”, *Op* = “**>=**”, *Value* = “**6**”.

Press “**Add/Update**” and input *Filtering name* to be for example “**Filtering, # assays>=6**”.

**Transformation: filter**

[Return](#)

**BioAssaySet** Filtered ALL transf. (all of 8 assays selected)

**Spot filter**

Field	Op	Value	Buttons	Translated value
-	>		Add/Update	

**Presets** Save current as new preset

**Gene filter**

Field	Op	Value	Buttons	Translated value
In # of Assays	>=	6	Delete	6
-	>		Add/Update	

**Presets** Save current as new preset

**Information**

**Experiment** CourseExperiment\_ALL

**Parent BioAssaySet** Filtered ALL transf.

Press **“Accept”** and wait for filtering to complete. Go to the new BioAssaySet.

This reduces the number of reporters substantially, and should make you end up with around 4000 reporters.

## Statistical analysis

Now we will perform a few statistical tests to try to find differentially expressed genes(Cui and Churchill 2003).

We will take advantage of the fact that we have several measurements of the gene expression levels in our both biological and technical replicates, which will help us to determine whether a certain gene is differentially expressed between the two groups of samples.

Remember from the introduction that your data set consists of 8 arrays: 4 arrays with ALL samples with B-cell origin hybridized together with a common reference( dye swaps of 2 biological samples) and 4 arrays with ALL samples with T-cell origin hybridized together with a common reference(dye swaps of 2 biological samples).

**Samples with T-cell origin:** 98365 (raw data set **1&2**) and 00134 (raw data set **3& 4**)

**Samples with B-cell origin:** 98278 (raw data set **5 & 6**) and 982 (raw data set **7 & 8**)

So the first group of arrays (**1-4**) corresponds to the samples with **T-cell origin** and the last four (**5-8**) correspond to the samples having a **B-cell origin**. The aim of the test we are about to perform is thereby to find genes separating these two groups. In practice this means for example detecting a marker for T-cells.

This information will help you decide on how to create your design matrix and group your samples for the B-statistic, but since the experimental design has not been thoroughly covered in the course we will give you the correct design matrix below and also try to help you with the design issues you might have for your own datasets. For information about how to create the design matrix, see (Glonck and Solomon 2004).

Since we have a reference design in our experiment, we need to use two parameters to be able to estimate the indirect difference between the two groups of interest. The first parameter will estimate the difference between the reference and the T-cell samples (which in practice then will pick up all differences between the leukemia samples and the reference pool so this is not really relevant for our interest). The difference between the T-cell samples and the B-cell samples will then be estimated through the second parameter, where only the genes with an expression that differs between the two groups will receive a high B score, but this would not have been possible using only one parameter since the comparison is indirect.


## B-statistics

Use the data set with about 4000 genes that you ended up with after normalization and filtering in the previous step.

This plug-in calculates B-statistics(Smyth, Yang et al. 2003) values for ranking genes for each parameter in the design matrix. A number of graphs are produced and the B-statistics values for each gene and parameter are stored in the resulting data set. It requires that all arrays in the experiment have exactly the same layout, i.e. reporters at the same positions.

Note that there should be no negative values in the design matrix for dye-swap experiments if the data was loaded into BASE using different file formats for the two cases.

Choose “**Run Application**” and select “**Hypothesis testing: B-statistics v4.0**”.

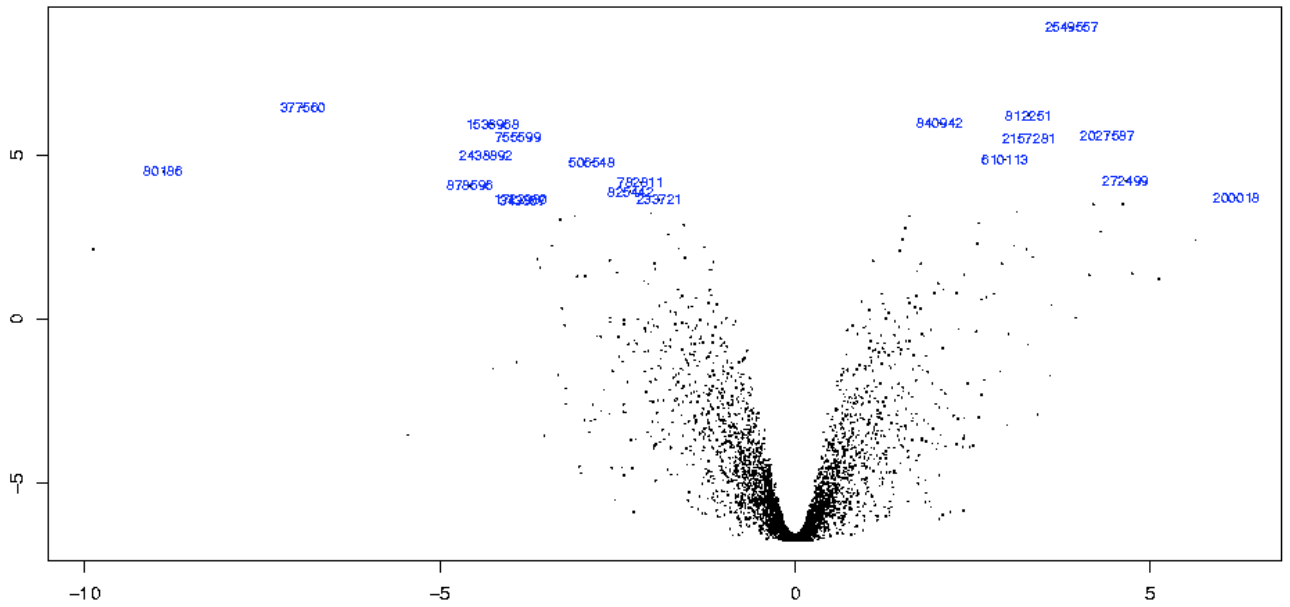
Parameter 	Type	Value
Mark # top genes in plot	Integer	<input type="text" value="20"/>
Rows in design matrix	Integer	<input type="text" value="8"/>
Columns in design matrix	Integer	<input type="text" value="2"/>
Design Matrix (comma separated, fills rowwise)	String	<input type="text" value="1,0,1,0,1,0,1,0,1,1,1,1,1,1,1,1,1"/>

Create a design matrix with 8 rows representing the 8 arrays/slides in the experiment and 2 columns representing the two parameters we will estimate and fill it row-wise with the values “**1,0,1,0,1,0,1,0,1,1,1,1,1,1,1,1,1**”.

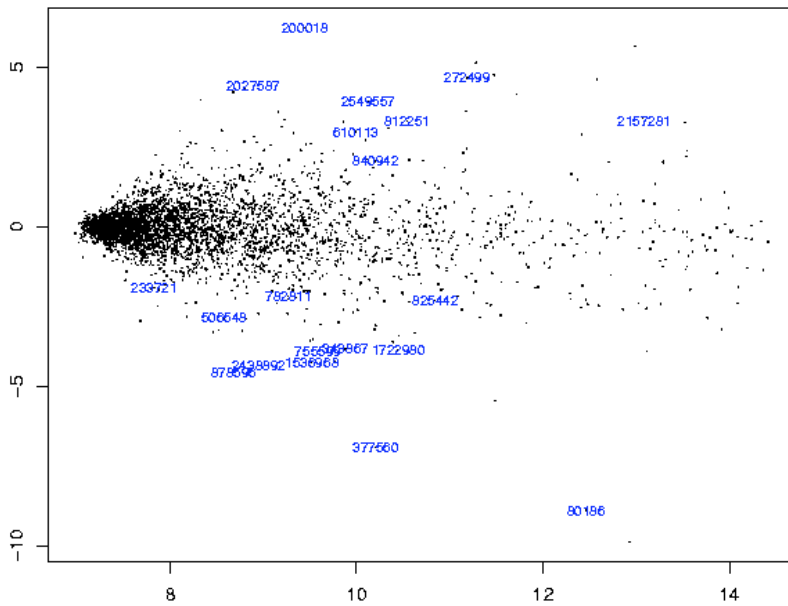
Press “**Start job**”.

When finished, press “**View**” on the resulting file “**Result.html**” and you should see a list of the top 20 ranked genes, MA-plots with these genes marked and Volcano plots of the B-statistics. On the next page there is an output of the resulting plots.

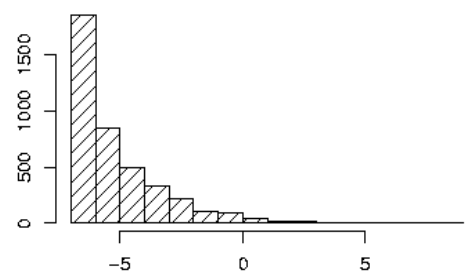
B-stat plot (with reporter ids as labels)



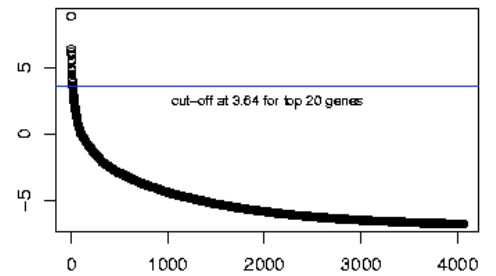
MA-plot



Histogram of B-statistics



Number of genes over cut-off



## Investigation of your results

If you want to know more about any of your interesting genes you can use the features within the LCB DWH or you can go to some biological database to get some biological information about your top genes.

To use the features available for further exploring the genes within the system, you can use the **Experiment Explorer** tool, described in the next section.

Here are some markers used clinically in ALL studies...are any of them among your top genes?

**Table 2.** Immunophenotype in ALL.

**A. Pertinent markers available for immunohistochemical studies**

General: TdT, CD34

B-cell: CD20, CD79A

T-cell: CD3, CD4, CD8, CD5, CD45RO

Myeloid: MPO, CD68, lysozyme, glycophorin A, Factor VIII, CD61

Other: keratin, NSE, myogenin, CD99

**B. Commonly used markers for flow immunophenotyping in acute leukaemia**

General: CD34, HLA-DR, TdT, CD45

B-cell markers: CD10, CD19, cCD22, CD20, CD79A, CD24

T-cell markers: CD1a, CD2, cCD3, CD4, CD8, CD5, CD7

Myeloid: MPO, CD117, CD13, CD33, CD11c, CD14, CD15

**C. B-lineage ALL phenotypes**

Pro-B: TdT+, CD19/22/79A+, CD10-, cμ-, slg-

Common precursor-B: TdT+, CD19/22/79A+, CD10+, cμ-, slg-

Pre-B: TdT+, CD19/22/79A+, CD10+, cμ+, slg-

Burkitt: TdT-, CD19/22/79A+, CD10+, slg+

**D. T-lineage ALL phenotypes**

Pro/immature thymocyte: TdT+, cCD3+, CD2/5/7+/-

Common thymocyte: TdT+, cCD3+, CD2/5/7+, CD4+/CD8+, CD1a+

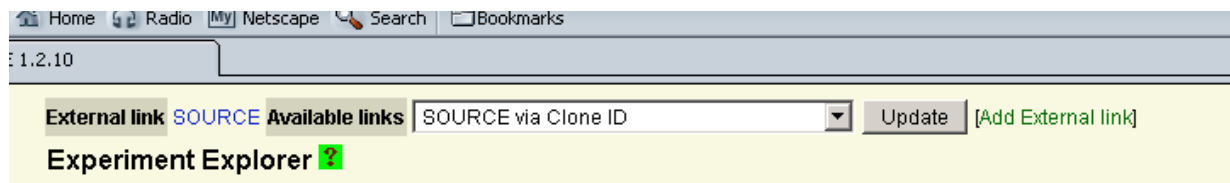
Mature thymocyte: TdT+/-, CD3+, CD2/5/7+, CD4+ or CD8+, CD1a-

## Experiment Explorer

At any stage you can click the **EExplorer** (short for Experiment Explorer) to the right of your data sets in order to browse the result from the analysis so far. In the top right corner there is a section Page – click on “[all]” to display duplicated genes on an array on the same screen.

Maybe you want to know more about the top genes from the statistical tests. Then you can **click on** for example the **“EExplorer”** to the right of your *normalized data set* or the data set you sent as *input to the statistical test* depending on which values you want to investigate.

At the top of the page you will see:



This means that you now have the option of linking your Reporter IDs to any public data base that will contain some information about these IDs, for example you might have GenBank Acc nr as your ID. In this case we have **Clone IDs** from the SOURCE database, so we will use that external link.

Then choose **“Reporter search”** and fill in the Reporter Name or Reporter ID of the gene of interest.

Then click on **“Explore”** and the data for the gene is displayed.

This can for example be useful if you are interested to see the individual values for this gene and also offer a possibility to check whether the individual values in either of the channels are close to background, i.e. close to 0.

The example below displays the information about one of the top genes from SAM, having **Gene Symbol CD3D**.

<b>Reporter search</b>	Gene symbol = CD3D	<b>Matching reporters</b>	1	<b>Total reporters</b>	79
<b>Annotation</b>	- none -	<b>LIMS info</b>	None	<b>Spot images</b>	None
<b>Raw data</b>	Interesting	<b>Current index</b>	1	<b>Page</b>	1 / 1
<b>Reporter name</b>	CD3D antigen, delta polypeptide (TiT3 complex)				<<prev next>> <<first [
<b>Reporter ID</b>	377560	<b>Unigene cluster</b>	.95327	<b>Gene symbol</b>	CD3D
<b>Length</b>	0	<b>Accession</b>	NM_000732	<b>NID</b>	
<b>Chrom.</b>	11	<b>Cytoband</b>	11q23	<b>Markers</b>	
<b>LocusLink</b>	915	<b>OMIM</b>		<b>Start pos.</b>	0
<b>End pos.</b>	0				

Assay $\Delta$	Ch1 int	Ch2 int	Ratio	log2	Spot size	Rgn. ratio	Flags	M-value	CV	Pos
030319 Cy3_ref Cy5_98365 BP8_s56_1246347	12075	1336.6	9.03	3.18	120	0.56	0	0	0	2302
030319 Cy3_98365 Cy5_ref BP8_s58_1246347 #2	18050	1611.4	11.20	3.49	120	2.47	0	0	0	2302
030313 Cy3_ref Cy5_00134 BP8_s49_1242695	3252.9	756.12	4.30	2.11	110	5.917	0	0	0	2302
030313 Cy3_00134 Cy5_ref BP8_s50_1242698 #2	3240.9	648.54	5.00	2.32	110	3.219	0	0	0	2302
030227 Cy3_ref Cy5_98278 BP8_s31_1242699	206.97	3195.1	0.06	-3.95	110	2.364	0	0	0	2302
030227 Cy3_98278 Cy5_ref BP8_s32_1242698 #2	81.448	534.80	0.15	-2.72	90	7.21	0	0	0	2302
030226 Cy3_ref Cy5_982 BP8_s29_12458863	118.30	4340.5	0.03	-5.20	50	22.62	0	0	0	2302
030226 Cy3_982 Cy5_ref BP8_s30_12458864 #2	120.63	3281.5	0.04	-4.77	100	0.623	0	0	0	2302

You can now look at the raw data and the individual ratios or proceed through clicking either the Reporter ID or LocusLink to connect to other sources of information linked to the gene. Feel free to try this feature for some of the other genes if you have time.

## SAM

SAM (Significance Analysis of Microarrays) is a statistical tool to try to find differentially expressed genes in your data set.

SAM identifies genes with statistically significant changes in expression by assimilating a set of gene-specific  $t$  tests. Each gene is assigned a score on the basis of its change in gene expression relative to the standard deviation of repeated measurements for that gene. Genes with scores greater than a threshold are deemed potentially significant. Please see the references for more information about this algorithm(Tusher, Tibshirani et al. 2001).

First, you need to run the plug-in “**Miscellaneous: Imputation v4.0**” to compensate for missing values (MV). This is required since SAM does not handle this. The method uses  $k$  nearest neighbors’ imputation to fill in missing values for spot intensities, on both channels.

Choose a number of neighbors you would like for the imputation or leave the default value, then press “**Start job**” and wait for the imputation to finish.

Remember from the introduction that your data set consists of 8 arrays: 4 arrays with ALL samples with B-cell origin hybridized together with a common reference( dye swaps of 2 biological samples) and 4 arrays with ALL samples with T-cell origin hybridized together with a common reference(dye swaps of 2 biological samples).


Samples with T-cell origin: 98365(raw data set 1&2) and 00134(raw data set 3& 4)  
Samples with B-cell origin: 98278(raw data set 5 & 6) and 982(raw data set 7 & 8)

So the first group of arrays (**1-4**) corresponds to the samples with **T-cell origin** and the last four (**5-8**) correspond to the samples having a **B-cell origin**. The aim of the test we are about to perform is thereby to find genes separating these two groups. In practice this means for example detecting a marker for T-cells.

Now, go to the new BioAssaySet and run the plugin **“Hypothesis testing: SAM v5.0”**.

Choose the parameters shown below.

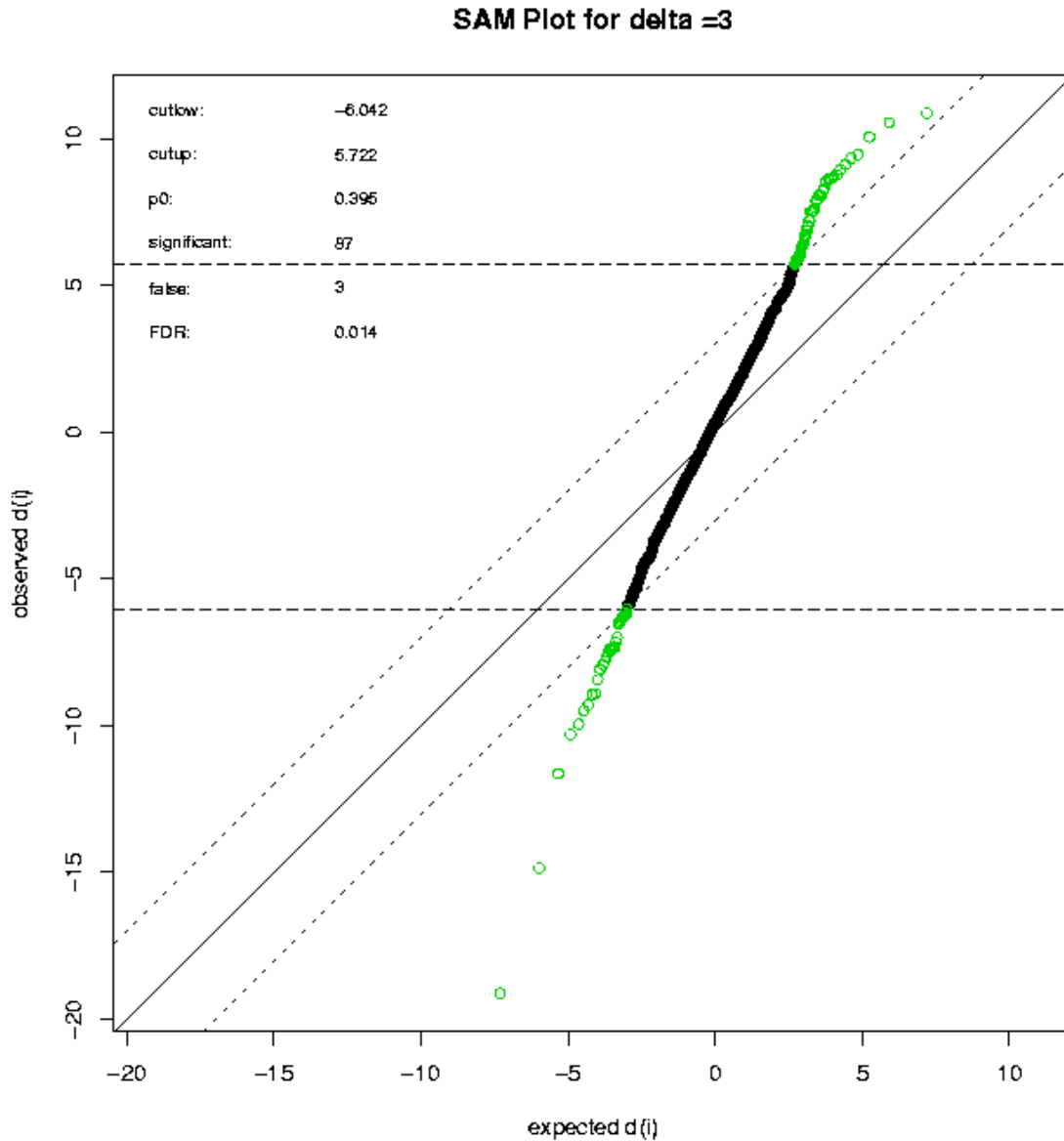
<b>Plug-in</b>	Hypothesis testing: SAM v5.0
<b>About plug-in</b>	This plug-in detects significant genes using the Significance Analysis of Microarrays (SAM) method. It produces graph scores as well as SAM delta values are stored in the resulting data set. Requires that all arrays in the experiment have same layout, i.e reporters at the same positions.  Note that 1 group SAM analysis is not working in this version!  This plug-in is written in R using functions from the Bioconductor library siggenes.
<b>Job name</b>	<input type="text" value="Hypothesis testing: SAM v5.0"/>
<b>Description</b>	<input type="text"/>

Parameter 	Type	Value
Number of permutations	Integer	<input type="text" value="100"/>
Lowest delta value	Float	<input type="text" value="0"/>
Highest delta value	Float	<input type="text" value="6"/>
Distance between delta values	Float	<input type="text" value="0.2"/>
Array group 1	String	<input type="text" value="1,2,3,4"/>
Array group 2	String	<input type="text" value="5,6,7,8"/>
Create SAM plot for delta:	Float	<input type="text" value="3"/>

Press **“Start job”** and wait for the plugin to finish.

To identify potentially significant changes in expression, SAM uses a scatter plot of the observed relative difference *delta* vs. the expected relative difference between the two groups. For the vast majority of genes the difference is very low, but for a number of genes there is a distance greater than a threshold. For example, the delta threshold 3 in the example yielded around 80 genes that were “called significant”.

The resulting plot should look something like:



Each of the rings represent genes, and the ones having the furthest distance to the diagonal line are the ones most likely to be differentially expressed genes and this is also what the ranking in the list is based on. The genes marked green are the ones above the delta threshold set by the user so feel free to go back and change this threshold and run the plug-in again if you would like another delta value to be the cut-off.

## Gene Ontology (GO)

The Gene Ontology (GO) project is a collaborative effort to address the need for consistent descriptions of gene products in different databases.

The GO collaborators are developing three structured, controlled vocabularies (ontologies) that describe gene products in terms of their associated biological processes (BP), cellular components (CC) and molecular functions (MF) in a species-independent manner. For example, if you were searching for new targets for antibiotics, you might want to find all the gene products that are involved in bacterial protein synthesis, and that have significantly different sequences or structures from those in humans. But if one database describes these molecules as being involved in 'translation', whereas another uses the phrase 'protein synthesis', it will be difficult for you - and even harder for a computer - to find functionally equivalent terms.

GO is not a database of gene sequences, nor a catalog of gene products. Rather, *GO describes how gene products behave in a cellular context.*

A couple of newly developed tool at the LCB DWH lets the user connect GOs to the genes of interest, and also offers the possibility to compare the GO for a certain subset of genes compared to the rest of the genes on the array or in the dataset.

## The GO statistics

Normally, we would first use a plug-in here to annotate the genes with Gene Ontology ids. The output will be imported into the data warehouse in the columns GOCC, GOMF and GOBP. In this plug-in “**GO: Background Distribution V2**” the GO frequencies of all reporters on all assays are calculated. This background will then be accessible for all “**GO: Significance Test**” runs throughout the whole experiment, and in the second step the background will be used for comparison to a subset of the genes, i.e. a list of interesting and hopefully significant genes.

However, it can take up to a long time to create the background, so in this case we will ask you to use a background already made. The dataset used to create the background should be considered; in this case it is the **dataset after filtering away the flagged spots and merging the duplicates.**

Below you can see what it looks like when you choose to run the plug-in “**GO: Background Distribution V2**” and decide which sources of annotations to use. A recommendation would be to not include the IEA (electronic annotation), which is known to contain a lot of noise.

**Description**

Parameter	Type	Value
Background filename (To be used later in "GO:Significance Test")	String	"background"
Database to use:	Enum	2nd March 2005
TAS (traceable author statement)	Enum	Yes
IDA (direct assay)	Enum	Yes
IC (curator)	Enum	Yes
IMP (mutant phenotype)	Enum	Yes
IGI (genetic interaction)	Enum	Yes
IPI (physical interaction)	Enum	Yes
ISS (sequence or structural similarity)	Enum	Yes
IEP (expression pattern)	Enum	Yes
NAS (non-traceable author statement)	Enum	Yes
IEA (electronic annotation)	Enum	Yes
NR (not recorded)	Enum	Yes
ND (no biological data available)	Enum	Yes
Field to map to GO	Enum	LocusLink (Entrez GeneID)
Field to Annotate genes	Enum	Gene Symbol

Start job

*Now we will go on and use the already made "background" file in the next step.*

Then we will filter out the top ~100 genes from the statistical ranking, and then we will compare the GO distribution of these to all the genes we could have measured. Since we only want about the top hundred genes, we need to use the information we got from the results in the B-test as a cutoff to filter out the top genes.

Click on the resulting data set from the B-test and then choose "Filter".

[Return](#)

**BioAssaySet** Filtered ALL transf. transf.2 (all of 8 assays selected)

**Spot filter**


Field	Op	Value	Buttons	Translated
[Extra] Bstat2	>	0	Delete	0
-	>		Add/Update	

Now we can create a filter where only the genes with a delta values above a certain threshold will be filtered out and saved into the new data set. Since the B value 0 proposed results in about 100 genes from visual inspection of the plot, we will choose

that value here. This number is a reasonable size to work with, but feel free to try other filters if you have time.

The newly created dataset with only 100 genes will now be tested to see if the GO group distribution differs from the GO groups present in the background, i.e. to see if there is some significance in terms of annotations in the selected group of genes.

Choose **“Run App”** on the 100-gene-dataset or another dataset you choose to use as your “interesting list”, i.e. the subset of genes you would like to investigate, and then choose to run the plug-in **“GO: Significance Test V2”** and decide which sources of annotations to use. A recommendation would be to use the same as you did for the background. There is also a choice of what background to use, so in order to use the background we already created for you, use the name shown below.

Parameter 	Type	Value
Background filename	String	"/CourseBackground"
How to divide assays, ex "1,2,3-6", "" gives all by themselves	String	"1-8"
Ontologies to consider, ex "CC;MF;BP"	String	"CC;MF;BP"
Cut Off, level of (two sided) significance	Float	0.025
TimeProfile (series)	Enum	No <input type="button" value="v"/>
Use Backgrounds Evidence Codes (overrides choices below)	Enum	Yes <input type="button" value="v"/>
TAS (traceable author statement)	Enum	Yes <input type="button" value="v"/>
IDA (direct assay)	Enum	Yes <input type="button" value="v"/>
IC (curator)	Enum	Yes <input type="button" value="v"/>
IMP (mutant phenotype)	Enum	Yes <input type="button" value="v"/>
IGI (genetic interaction)	Enum	Yes <input type="button" value="v"/>
IPI (physical interaction)	Enum	Yes <input type="button" value="v"/>
ISS (sequence or structural similarity)	Enum	Yes <input type="button" value="v"/>
IEP (expression pattern)	Enum	Yes <input type="button" value="v"/>
NAS (non-traceable author statement)	Enum	Yes <input type="button" value="v"/>
IEA (electronic annotation)	Enum	Yes <input type="button" value="v"/>
NR (not recorded)	Enum	Yes <input type="button" value="v"/>
ND (no biological data available)	Enum	Yes <input type="button" value="v"/>
Field to map to GO	Enum	LocusLink (GeneID) <input type="button" value="v"/>
Field to annotate genes	Enum	Gene Symbol <input type="button" value="v"/>

Press **“Start job”** and wait for the significance test to finish.

When finished, the results will appear in three CC, MF, BP forms, and you can view each one of them to see whether any GO groups ended up significant with the parameter settings used, and if these groups seem to be reasonable results!

## The GO plot tool

Go to the dataset that you want to examine, for example a dataset you are interested in investigating the GO terms for.

You need to check so that most reporters in the set have Entrez (Locus Link) ids  
If not, the reporters need to be updated first! In this case, the reporters are already updated and you can proceed to the next step.

Run the plug-in “**GO: Import Annotations**”. This plugin annotates genes with Gene Ontology ids and stores all GO ids for MF,BP and CC for all reporters in the dataset.

The output will be imported into the datawarehouse in the columns GOCC,GOMF and GOBP. This annotation will only exist for the current experiment. The GO ids are stored in three extra columns, with links to the amiGO browser

When the first plug-in is finished, it is time for the visualization tool.  
Run the plug-in “**GO: Visualization**” on the resulting dataset and choose which evidence codes you want to display.

# Visualization

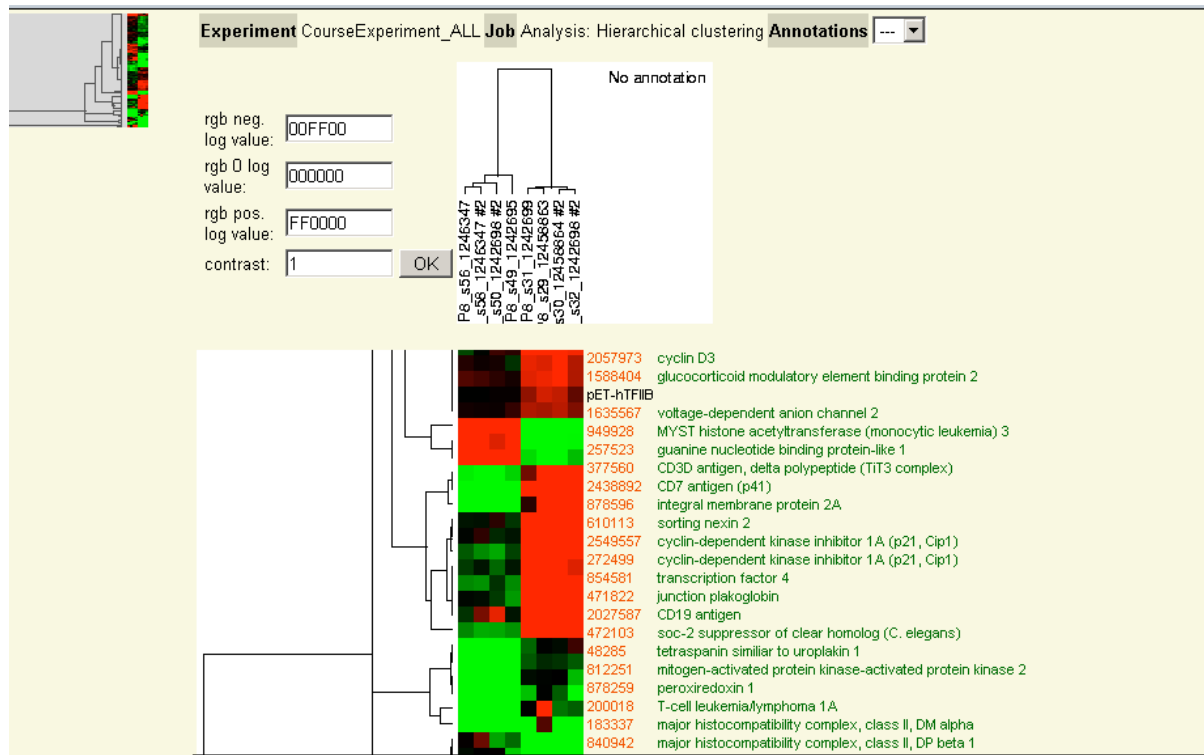
Even though the statistical test gives a ranking of the differentially expressed genes, it is difficult to get an estimate of how these are related. Which ones might have a similar profile? Then it can be useful to use a clustering algorithm for visualization of our data. Clustering can also be used to detect patterns in an unsupervised way to group genes or biological samples together, i.e. to find for example groups of co-regulated genes.

We will end this computer exercise by using some clustering algorithms for visualization of our data, for example the ~100 genes we filtered out to look at GO.

## Hierarchical clustering

The plug-in used in our system is a bottom-up hierarchical clustering. In the algorithm the two closest points are merged and the new cluster is represented by a weighted (median) or weighted (center of mass) average of the two points in gene expression space. This takes little RAM, allowing you to cluster a large number of genes, but the clustering results could be inferior to those obtained with e.g. average linkage.

Choose **“Run app”** on the resulting BioAssaySet from the statistical test with ~100 genes to visualize the results. Choose the plug-in **“Analysis: Hierarchical clustering”**. Leave the parameter settings at default values and press **“Start job”**. When finished use the **“Visualize”** option to the right in the table “Files created by this job”.



When investigating the results of your clustering you can click on different parts of the tree in the top left corner to zoom in and you can also click directly on specific genes in the part of the tree displayed on the screen. Take for instance a gene that you are interested in, and the link will take you straight to the **Experiment Explorer tool**, where much more information about the gene will be displayed.

Since this tool performs clustering in two dimensions, it does not only visualize the data from the gene selection. It also groups the genes with similar profiles together which may be an indication of for example similar regulation of these genes.

## **K-means clustering**

This plug-in performs a k-means clustering of the expression levels of genes, so that genes with similar expression patterns are grouped together into k groups. It requires that all arrays in the experiment have exactly the same layout, i.e. reporters at the same positions.

Before running this plug-in, you need to run the plug-in “**Miscellaneous: Imputation**” to compensate for missing values (MV). This is required since our K-means clustering algorithm does not handle this.

When finished imputing the missing values, choose “**Run app**” on the same BioAssaySet as used for Hierarchical clustering to visualize the results with this clustering algorithm instead. Choose the plug-in “**Clustering: K-means v4.0**”.

Here, you should enter the number of clusters (K). Since we have the top genes distinguishing between the two groups maybe a reasonable number would be four or five clusters. An indication of how many clusters there are can be drawn from the hierarchical clustering result, where you can see approximately how many clusters to specify for K-means clustering. There is also an option of reordering the samples, but since we have them grouped together already we leave that option.

Enter the number of clusters, eg. (K) = 4, to separate the genes into 4 clusters. Press “**Start job**”.

When the clustering is finished, look at the results through clicking on “**View**” to the right of “**Results**” in the table “Files created by this job”.

### **In which cluster is the gene?**

Where is the gene that we investigated before, CD3D?

You can use the Experiment Explorer tool to find out in which cluster this gene ended up. There is now an extra column to the right of the log ratio saying which cluster the gene belongs to.

If you do not remember how to use the tool to look at a certain gene, please go back to the instructions for Experiment Explorer.

### Which genes are in a certain cluster?

So, now we found out the gene CD3D was in a certain cluster, or maybe we looked at the clusters and saw where it was. Then maybe we are interested in other genes related to this gene? Then we can use the filtering option to filter out all genes that ended up in the same cluster. If CD3D for example ended up in Cluster 1:

**Transformation: filter**  
[Return](#)  
**BioAssaySet** Filtered ALL transf. transf.2 transf.3 transf.4 transf.5 (all of 8 assays selected)

**Spot filter**

Field	Op	Value	Buttons	Translated val
[Extra] inCluster	=	1	Delete	1
-	>		Add/Update	

**Presets** Save current as new preset

Then we can use the Experiment Explorer tool again, this time if we press the Reporter Search button we can look at all the genes present in this cluster.

## Reporter list

Another useful feature within the system is to use **“Reporter Lists”**. If you want to filter out or know more about any of your interesting or favourite genes you can use this feature to sort out a list of genes.

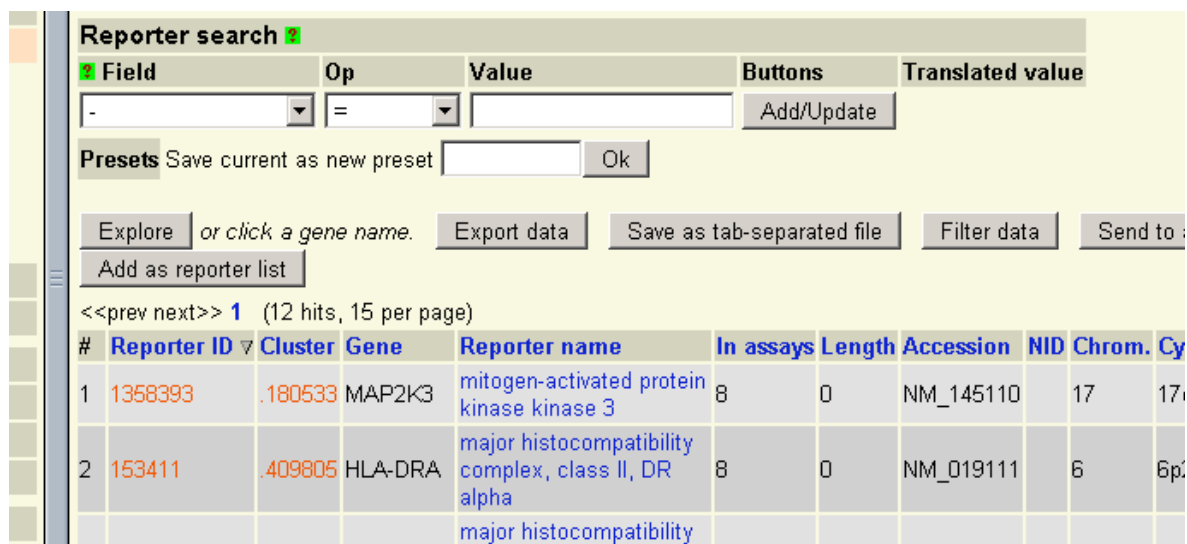
The list can be a text sheet uploaded from your local computer for example, and maybe containing your 100 favourite genes that you want to sort out from the dataset with all the interesting gene expression data you have obtained and download, or maybe a list of genes you want to use for GO term investigation.

We will use another example of using reporter lists to see how robust our K-means clustering is.

We will **choose one of the clusters** from the previous clustering by **filtering out all the genes** that was in this cluster, for instance like we did above. We took a cluster that was easy to recognise because it contained a lot of HLA-genes. In this case that was Cluster 2, but since the start guess is different every time this cluster number is not necessarily the same next time, but the content should stay similar if the algorithm and parameters used are fairly robust.

We filter out the genes in Cluster 2 like we did above, and for this new dataset we press the **“EExplorer”** button.

Then if you are not at the page for reporter search, press **“Reporter Search”** and use the option of adding this dataset as a reporter list through pressing **“Add as reporter list”** and come up with a name of your own choice.



The screenshot shows the 'Reporter search' interface. At the top, there is a search form with a 'Field' dropdown (set to '-'), an 'Op' dropdown (set to '='), and a 'Value' input field. Below the search form is a 'Presets' section with a 'Save current as new preset' button and an 'Ok' button. Further down, there are several action buttons: 'Explore or click a gene name.', 'Export data', 'Save as tab-separated file', 'Filter data', and 'Send to...'. Below these buttons is an 'Add as reporter list' button. The main part of the interface is a table of results. The table has columns: '#', 'Reporter ID', 'Cluster', 'Gene', 'Reporter name', 'In assays', 'Length', 'Accession', 'NID', 'Chrom.', and 'Cy'. The table shows two rows of data:

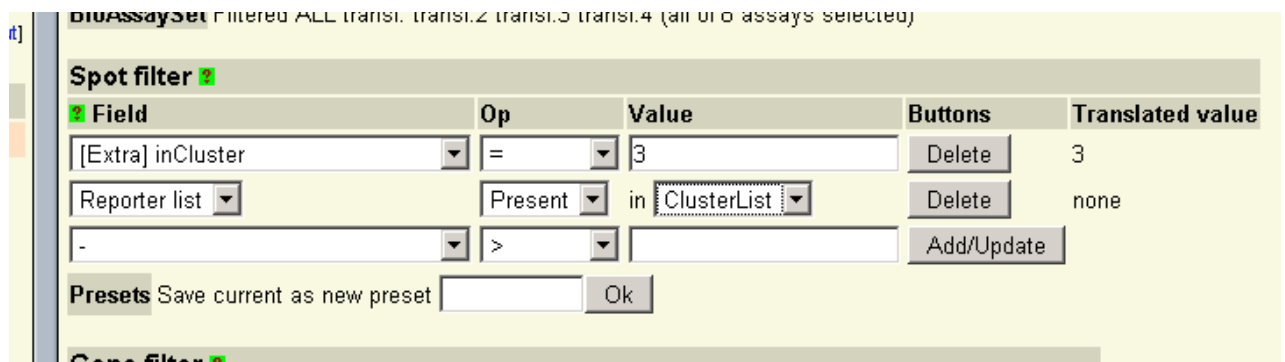
#	Reporter ID	Cluster	Gene	Reporter name	In assays	Length	Accession	NID	Chrom.	Cy
1	1358393	.180533	MAP2K3	mitogen-activated protein kinase kinase 3	8	0	NM_145110		17	17
2	153411	.409805	HLA-DRA	major histocompatibility complex, class II, DR alpha	8	0	NM_019111		6	6p

Now we will perform the clustering again and see if we can find this cluster again and if it contains the same genes.

Choose the plug-in **“Clustering: K-means v4.0”** just like you did before on the same dataset as before, and use the same number of clusters as before.

When finished try to look at the results and see which cluster is the one that you investigated before.

Then press **“Filter”** for the resulting dataset and try to filter out the genes that were in the same cluster for both procedures, i.e. the ones present in both the reporter list and the cluster you recognized as the same this time.



In our example case, it turned out that this cluster was very stable and all the genes were present both times, but it can be a good way of investigating your clustering and see if the genes are not the same at all maybe you need to increase the number of iterations or the number of clusters!

If you want to visualize a specific dataset there is a plug-in producing simple heat maps that might be useful. Feel free to for instance use this plug-in on the results from above, and look at the genes present in both clusters in a heat map.

Choose the plug-in **“Visualization: Heat map v4.0”** to create a heat map for your data.

The tools described above can be helpful when you are trying to investigate the results of your analysis and there are many more ways to use them, but hopefully the exercises gave you an idea of how to use them.

Good luck in the future!!

## References

- Butte, A. (2002). "The use and analysis of microarray data." Nat Rev Drug Discov **1**(12): 951-60.
- Cui, X. and G. A. Churchill (2003). "Statistical tests for differential expression in cDNA microarray experiments." Genome Biol **4**(4): 210.
- Glonek, G. F. and P. J. Solomon (2004). "Factorial and time course designs for cDNA microarray experiments." Biostatistics **5**(1): 89-111.
- Hess, K. R., W. Zhang, et al. (2001). "Microarrays: handling the deluge of data and extracting reliable information." Trends Biotechnol **19**(11): 463-8.
- Howell, S. B. (1999). "DNA Microarrays for Analysis of Gene Expression." Mol Urol **3**(3): 295-300.
- Smyth, G. K., Y. H. Yang, et al. (2003). "Statistical issues in cDNA microarray data analysis." Methods Mol Biol **224**: 111-36.
- Tusher, V. G., R. Tibshirani, et al. (2001). "Significance analysis of microarrays applied to the ionizing radiation response." Proc Natl Acad Sci U S A **98**(9): 5116-21.
- Yang, Y. H., S. Dudoit, et al. (2002). "Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation." Nucleic Acids Res **30**(4): e15.