Robin and RobiNA
Users' Manual

May 2012

Table of Contents

1 RobiNA – RNA-Seq analysis

1.1 Introduction

Before going into detail on the newly introduced workflow for RNA-Seq based analysis of differential gene expression, we want to point out that the RobiNA software package also contains the microarray workflows described in section 2. If you installed RobiNA you will thus have access to all the functionality that was until now provided by Robin. If you want to use RobiNA to analyse microarray data, please continue reading in section 2.

Analysis of differential gene expression using sequencing data is a relatively new approach that takes advantage of the new high-throughput (HT) sequencing technologies developed e.g. by Illumina/Solexa. The basic assumption of this approach is that the frequency with which a certain RNA molecule in a given RNA mixture is sequenced is proportional to the number of copies of this RNA molecule in the mixture. Using this assumption it is possible to extract transcript profiles from HT RNA sequencing (RNA-Seq) data both at a high sensitivity (when using a sufficiently high number of sequence reads) and specificity, because the reads can be directly mapped to a reference transcriptome or genome. Of course, the amount of reads that can be uniquely mapped to specific transcripts also depends on the length of the reads: The longer the reads are the more likely it is to be able to unambiguously find the transcript they originate from. Hence, a large amount of tens of millions of very long (e.g. 1kb) reads would be the ideal data set for RNA-Seq. Unfortunately, current sequencing technology does not yet allow the generation of such datasets (at affordable prices). To perform RNA-Seq based transcriptomics, the "depth" of sequencing is more important than the length of the individual sequence reads because it defines the dynamic range of the transcript profile that can be extracted from the data, given the length is long enough to disambiguate different isoforms in most of the cases, which is usually the case even with shorter reads (as of 2012) Therefore, the most frequently used technique for RNA-Seq is currently Illumina/Solexa sequencing that provides up to 200 million reads of a length of up to 100bp per RNA sample library. The Roche/454 sequencing technology usually yields much longer reads of up to 1kb but a smaller total number of reads (around 1 million) per sample and is thus much better suited for transcriptome and genome assembly.

Analogous to the microarray analysis workflows, the RNA-Seq workflow allows you to:

1) Thoroughly quality check the raw sequencing data
2) Filter out low quality reads based on a range of freely combinable criteria
3) Map the reads to a user-provided genome- or transcriptome reference sequence
4) Detect differentially expressed genes using state of the art statistical methods
5) Generate graphical summary plots and detailed tabular results that can be browsed directly inside the application and exported to downstream tools like MapMan and PageMan (or any other tool that can read simple tab-separated tables)

1.2 Input data format

The raw data has to be provided in FASTQ format [1], which is the standard output format for Illumina/Solexa data. It is also possible to use bzip2- or gzip-compressed FASTQ files – however, RobiNA will extract the files during the workflow and write the uncompressed data to the project directory (which will require extra disk space). So it is recommended to use uncompressed data directly.

Alternatively, users who have already aligned their reads to a reference, using a tool that generates BAM or SAM files, can directly import these by checking the "Alternative inputs" box on the import panel (see Figure 2). Using BAM/SAM files will skip the quality checking, filtering and mapping steps.

1.3 RNA-Seq Walkthrough

The following section will guide through all steps of a RNA-Seq analysis starting from raw data input. When reanalysing data from third parties that was obtained from the NCBI Short Read Archive (SRA), you'll first need to extract the reads as FASTQ files. This is easily done using the sratoolkits' "fastq-dump" tool (http://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?cmd=show&f=software&m=software&s=software). Files from the European Nucleotide Archive (ENA) can used directly Before actually starting the analysis workflow, the user has to define a project directory in which all data, that is generated during the analysis is saved (see Figure 1).

Figure 1: Project setup dialog. Create a new directory in which the analysis results and all other relevant data will be stored

1.3.1 Quality checking

The first step in the workflow is the import of raw sequencing data – so after choosing the RNA-Seq workflow, clicking the "Add" button will open a file browser in which the input files can be selected. To-date, RobiNA accepts Illumina/Solexa-type FASTQ files only, since this is the preferred data type for RNA-Seq based analysis of differential gene expression. However, in future releases, FASTA sequence files with separate FASTA quality file originating from other technologies like e.g. 454/Roche will also be supported (although the depth of sequencing provided by standard runs at the current state of this technology is, to our knowledge, not sufficient for in-depth sensitive analysis of differential gene expression). Upon clicking "Next" the workflow moves on to the quality checking step (see Figure 1). RobiNA offers a choice of quality test modules that cover a range of quality aspects of the data:

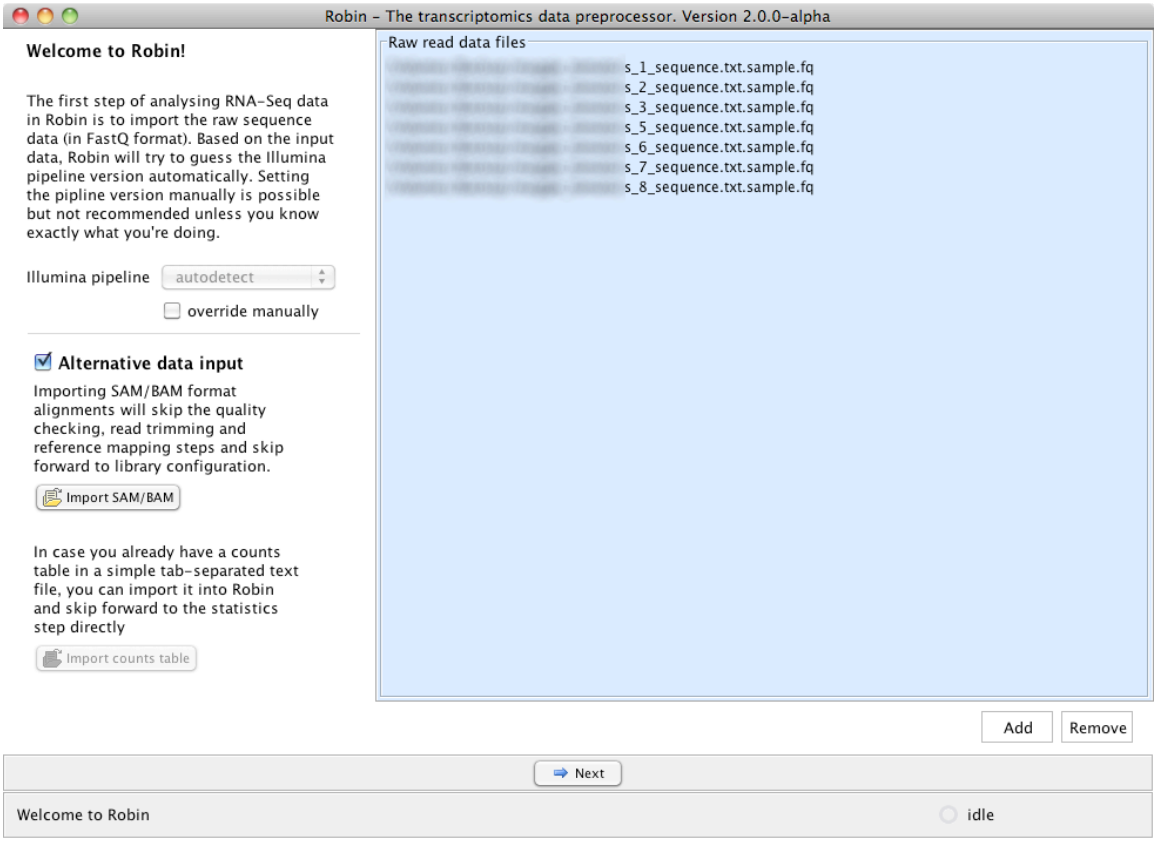

Figure 2: Raw data import panel. RobiNA accepts Illumina/Solexa fastq files as input. The quality encoding version will automatically be deduced from the data. Alternatively, BAM/SAM alignment files or precomputed counts tables can be imported.

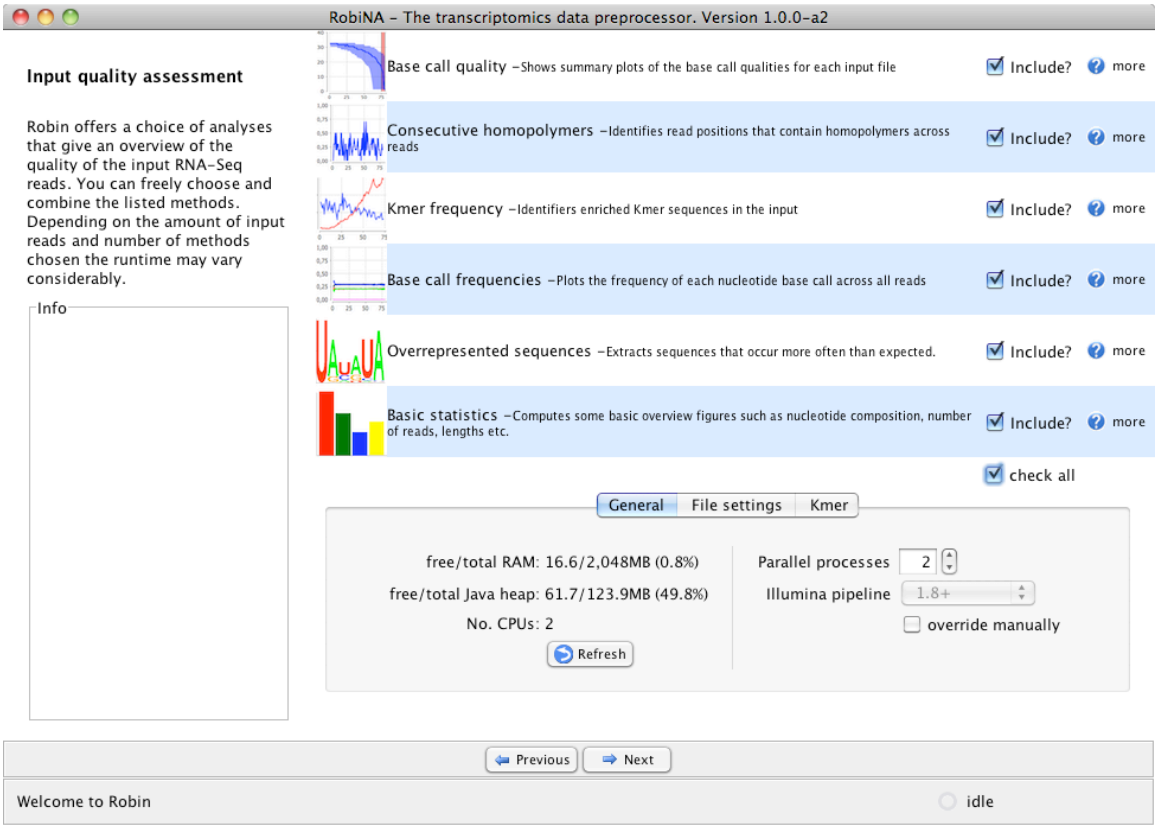

Figure 3: Quality checking options for raw deep sequencing data.

1) Base call quality: This module summarizes the base call quality scores included in the FASTQ file. All high-throughput sequencing technologies include a base calling step in their processing pipelines. The purpose of this step is to transform the raw signals recorded by, e.g. in the case of Illumina/Solexa sequencing, the fluorescence scanner into nucleotide base calls (i.e. A, C, G, T or N). The software component that accomplishes this step evaluates the raw signal and tries to assign a nucleotide to each signal. For each assignment, an error probability is computed, which, in turn, is transformed into a quality score according to the following equations:

Equation 1: PHRED score

$$Q_{PHRED} = -10 * \log_{10}(P_e)$$

Equation 2: Solexa score

$$Q_{Solexa} = -10 * \log_{10}(\frac{P_e}{1 - P_e})$$

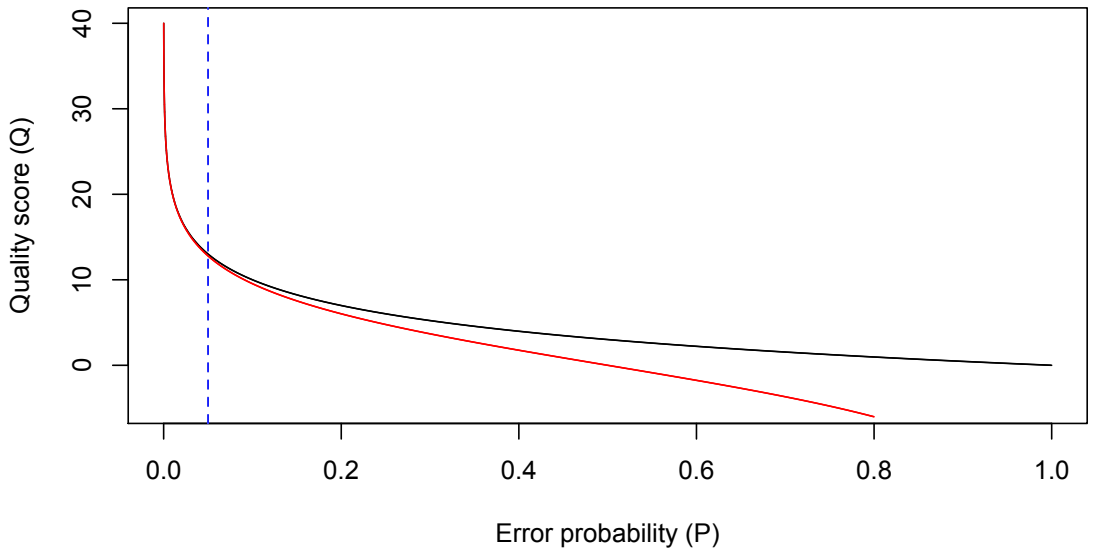

Figure 4: Plot showing the difference between quality scores computed according to equation 1 and equation 2. The blue dashed line denotes p=0.05. The diversion for higher error probabilities is stronger while quality of approx. Q=13 and greater have almost identical error probabilities in both equations.

with P_e being the probability of error. Depending on the processing pipeline version used to generate the data, either Equation 1 (version >= 1.3) or Equation 2 (prior to pipeline version 1.3) are used for computing the quality scores. In FASTAQ files, these quality scores are usually encoded as ASCII characters. Again depending on the pipeline version used, the scores are encoded by adding either 64 (Illumina pipeline until version 1.5+) or 33 (version 1.8+) to the quality score. For example a quality score of 35 would be encoded as ASCII character 64+35=99, i.e. a lower case 'c' in all versions prior to 1.8 and as 33+35=68 i.e. an upper case 'D' in version 1.8+.

Note: When working with RobiNA, the user usually does not have to deal with the quality encoding manually. RobiNA tries to extrapolate the encoding version by scanning the first 10000 entries of each file and evaluating the range of encoding characters observed. In most cases this yields a correct guess of the version used in the data. In case, however, this approach fails, the user can override the version manually on the "General" settings tab of the quality checking options panel (Figure 3)

1) Consecutive homopolymers. In rare cases, we have encountered raw sequencing data that showed the occurrence of short homopolymer stretches in every or almost every read starting at a specific nucleotide position that was the same across all reads. The base that was called, however, was not the same across the reads, i.e. read X might show a AAAAA pentamer starting at position 22 while read Y contains a GGGGG stretch at the same position. We are not sure what technical defect is causing this effect, but it will manifest itself as a clear spike towards a value of 1.0 on the consecutive homopolymer plot (see Figure 5)

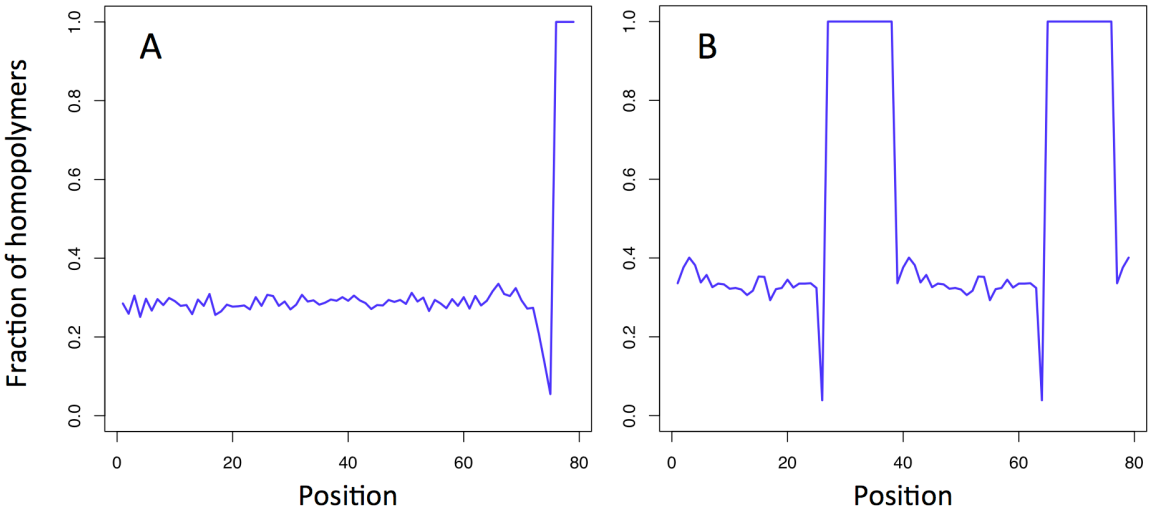

Figure 5: Example of homopolymer plots. Panel A) shows the homopolymer plot of a sequence set whose quality deteriorates strongly at the end of the reads leading to "N"-only base calls at the end of all reads from approximately position 75 on. This is a rather expected and, in the case shown, hardly alarming result, since there is no indication of an increased fraction of homopolymer base calls until the very low quality end of the reads. Panel B) shows a severe case with two blocks of homopolymer calls caused by a defect in the sequencing pipeline.

2) *K*mer frequency scan. Sequence biases in a large set of reads can be detected by breaking up the reads into shorter fragments of a defined size *K* (i.e. *K*mers) and conting the frequency of occurrence of each such fragment. The expected frequency of occurrence of each *K*mer can be computed from the nucleotide composition of the sampled sequences. If the observed frequency of a specific *K*mer exceeds its expected frequency by a factor of 3, the *K*mer is considered as enriched and recorded. The 10 most enriched *K*mers will be reported in the quality results summary, also graphically showing the positional enrichment within the reads. By default, RobiNA will only scan for enriched 5mer *K*mers. The user can, however, choose to record a wider range of *K*mer lengths and also restrict the maximal number of individual *K*mers to record for the enrichment analysis by changing the settings on the "Kmer" settings tab. Changing the settings there to higher values will also lead to a strongly increased computation time and memory consumption of the Java Virtual Machine (JVM) – so please

make sure that you have a fast computer that provides the necessary. Please note that the *K*mer enrichment analysis is computed for each input file individually – so if you run the quality checks in more than one parallel thread (see "parallel processes" in the "general" settings tab), the memory requirement is multiplied by the number of threads.

3) Base call frequencies. This module simply records the frequency in which each base (A, C, G, T and N) was seen at each position in the reads. Depending on the sample that was sequenced, the expected result will vary. Usually, when sequencing poly-A mRNA, you would expect to see roughly the same nucleotide composition as that of the underlying genome. In the case of many plant genomes, the composition usually shows a slight AT bias.

4) Overrepresented sequences. Analogous to the Kmer frequency module, this module identifies longer sequences that occur more often than expected. Usually these would be adapter sequences carried over from the library generation etc.

After choosing the quality check modules, clicking "Next" will start the processing of the input sequences. The "General" section of the settings tab allows to control some technical aspects related to the performance of the quality checking modules. The left part of the tab shows how much random access memory (RAM) is available and free on the machine in total, how much is available and free for RobiNA and how many processors (CPU) have been detected on the system. Based on these measurements, RobiNA automatically sets up the number of parallel quality checking and filtering processes in a very conservative way: The number of parallel processes automatically configured equals the number of CPUs found. On a machine with a lot of memory, it should usually be safe to run more parallel processes than CPUs – e.g. on a 32GB 8-core machine it would be safe to run 10-12 parallel quality checking threads. The more threads run in parallel, the faster the quality checking step will finish, given the underlying hardware supports it.
RobiNA will display the progress of each quality checking process – as soon as a process is finished, you can view the results by clicking on the respective status box (see Figure 6). In case a file is corrupted and cannot be processed, RobiNA will display a warning triangle and highlight file in red. Clicking it will display an error message indicating why the processing failed.

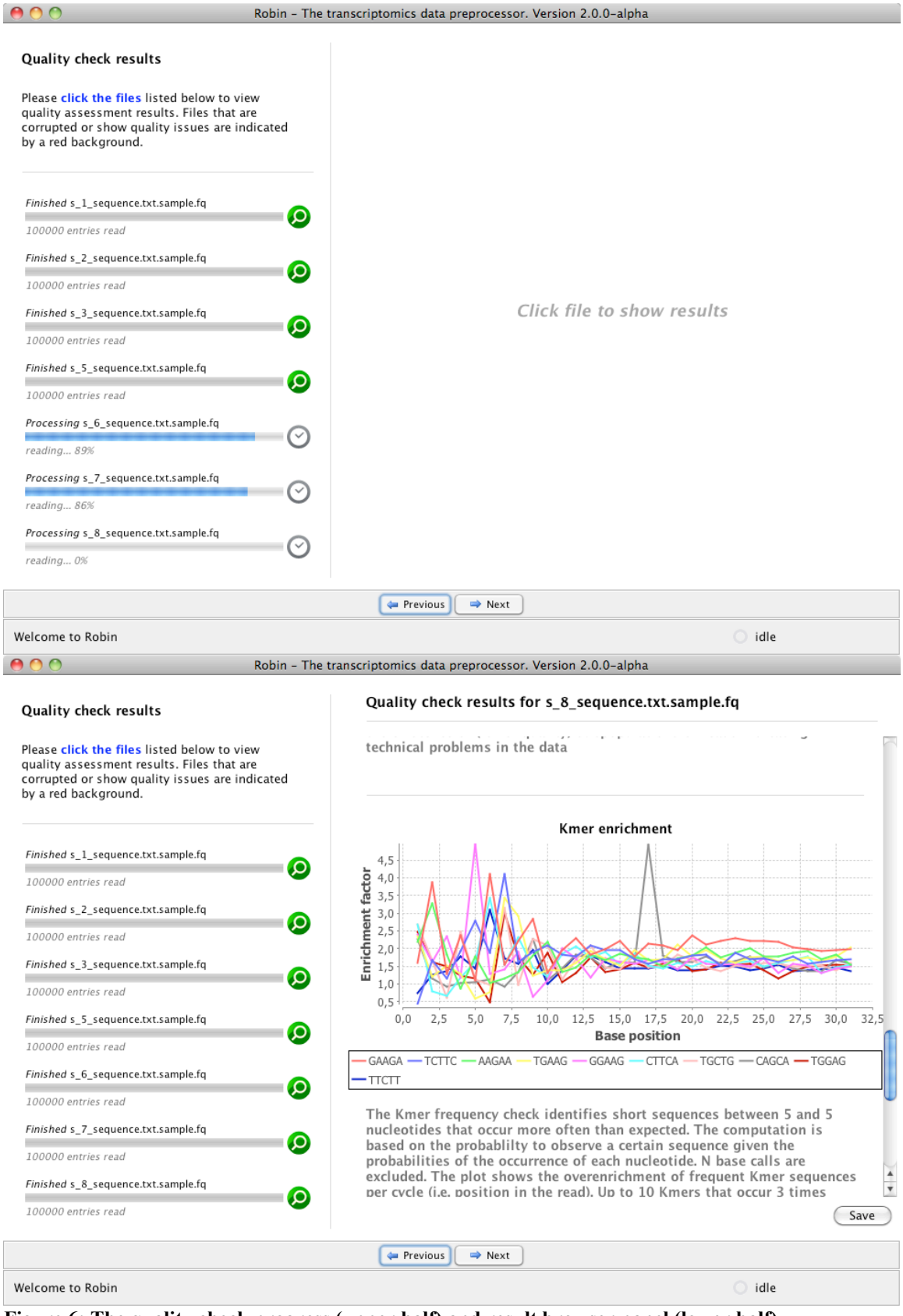

Figure 6: The quality check progress (upper half) and result browser panel (lower half).

1.3.2 Read filtering

When importing raw next generation sequencing data, it is very important to filter out low quality data and (identifiable) contaminating sequences. RobiNA provides a flexible way of doing this by offering a range of trimmer modules that can be combined freely to build a filtering pipeline that best suits the users' needs. The trimmer pipeline is also available as a stand-alone command line tool at http://www.usadellab.org/cms/index.php?page=trimmomatic. The trimming pipeline can be pieced together simply by dragging individual modules from the list on the lower left side of the Trimmomatic panel and dropping them into the white area on the right side with the mouse. For users who are unsure about what modules to use, we have included a default set of trimmers that can be added by clicking the "Standard" button on the lower right. The following trimmer modules are available:

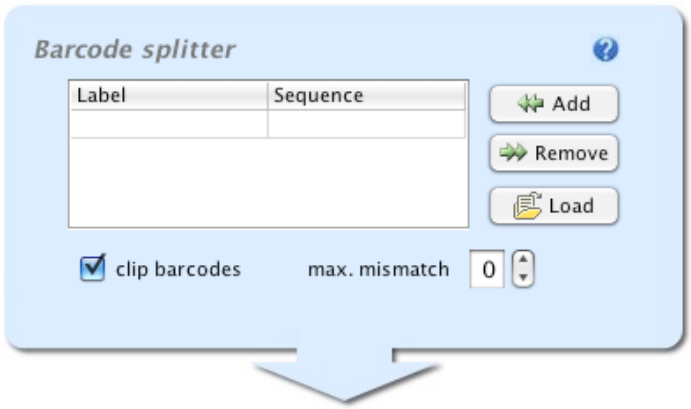

Figure 7: The barcode splitter module

1) Barcode Splitter: This module is not really a trimmer, because it does not remove any reads from the input. Instead, it splits the reads based on user-supplied barcode sequences. When working with multiplexed data, that means several different sample libraries have been sequenced in one sequencing run (e.g. one lane on a Illumina/Solexa platform), the different sample libraries are usually tagged by a short barcode sequence at the 5' end of the reads. To further analyse the data, the multiplexed reads need to be split into different files - one for each of the samples. To do so, you have to supply the barcodes used plus a short human readable label either by typing the information in the table provided in the barcode splitter box or by loading it from a tab-separated text file. You can set the barcode splitter to accept mismatches in the barcode sequence, however this is not really recommended. Reads that have a barcode that does not match to any of the

user-provided ones will be collected in a separated file with the file name prefix "UNKNOWN".

Figure 8: The adapter clipper module

2) Adapter clipper. In many cases, the raw sequence data contains a significant amount of reads that originate from adapter molecules that were used during the library preparation protocol. These reads should be removed prior to analyses since they constitute an artificial contamination and could, depending on their abundance, bias downstream analyses. The adapter clipper module clipper modules can be used to remove known adapter (and other short nucleotide sequences with a length <= the read length) sequences from the input data. These sequences can be provided by the user as a standard FASTA file. The "max. seed mismatch" and "match stringency" settings can be used to tweak the adapter search performance. The settings chosen should work robustly for most cases. However, if you expect very short adapters, the match stringency might have to be reduced in order to reliably detect them. For legal reasons we are not allowed to bundle known adapter sequences with the RobiNA software bundle. Feel free to contact us for further information on adapter sequences (please visit http://mapman.gabipd.org/web/guest/forum).

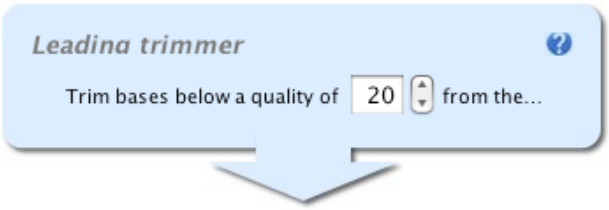

Figure 9: The leading trimmer module.

3) Leading trimmer. The leading trimmer simply removes bases below the specified minimal quality score from the start of each read, leaving the rest of the read untouched (even if there are more low quality reads in the middle of the end of the read – see “Trailing trimmer” and “sliding window trimmer” below).

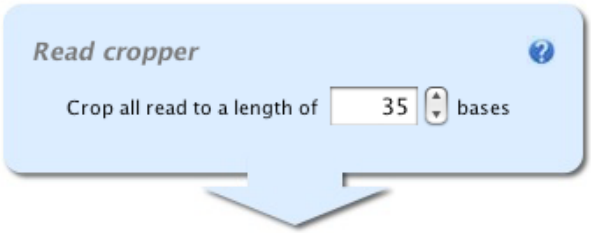

Figure 10: The read length cropper module.

4) Read cropper. The read cropper is very simply only cutting off all bases beyond the specified maximal length from all reads in the input. In the example given in Figure 10 this would mean that after processing all reads are 35 bases long or shorter.

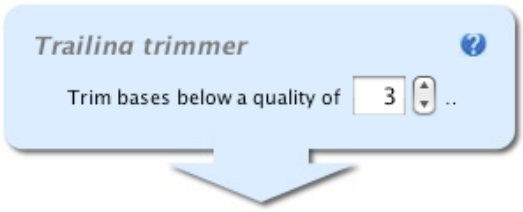

Figure 11: The trailing trimmer module.

5) Trailing trimmer. This trimmer module works analogous to the leading trimmer, but starts removing bases from the end. In its default setting, the trimmer is very conservative in only removing very low quality (score <= 3) bases.

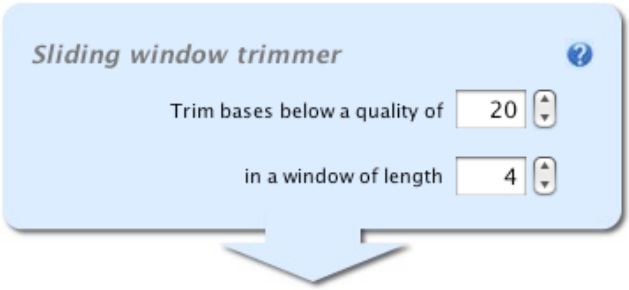

Figure 12: The sliding window trimmer module.

6) Sliding window trimmer. The sliding window trimmer will scan across each read in a window of the specified length and trim the read if the average quality within the window drops below the specified threshold. That means that long reads that contain a low base call quality stretch might get truncated at this stretch even if the quality downstream of this stretch is acceptable again. Depending on whether you want to be more conservative or stricter you might consider increasing (more conservative) or decreasing (stricter) the window size.

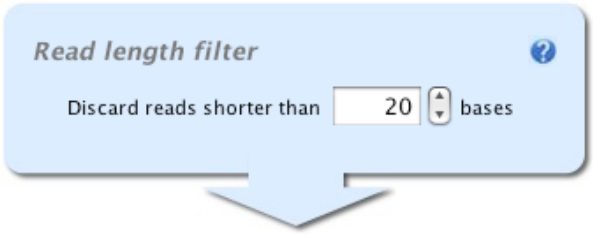

Figure 13: The read length filter module.

7) Read length filter. Removes all reads that are shorter than the specified minimal length. This module is very useful (and recommended) as the last processing step in a multistep pipeline that removes bases based on quality. If it is omitted, reads that are very short – possibly too short to be unambiguously mappable to a reference gene or transcript – will be carried through to the mapping step (which would unnecessarily extend the computation time).

1.3.3 Sample definition

Now that low quality reads and detectable contaminants have been removed, the next step in the analysis is the definition of the different treatments and samples. Every individual sequence file was generated by sequencing an RNA sample that was taken from biological material treated in a defined way. This information has to be entered on the "Experiment set up" panel (see Figure 14). First, all experimental conditions, or treatments, have to be listed in box in the lower left corner of the panel. For example, when different tissues are to be compared, the user would have to list which tissues these are (e.g. "root", "flower"; or "liver", "brain" etc.). After all conditions have been entered, the samples can be defined simply by selecting an input file and a condition and then clicking the "add" button.

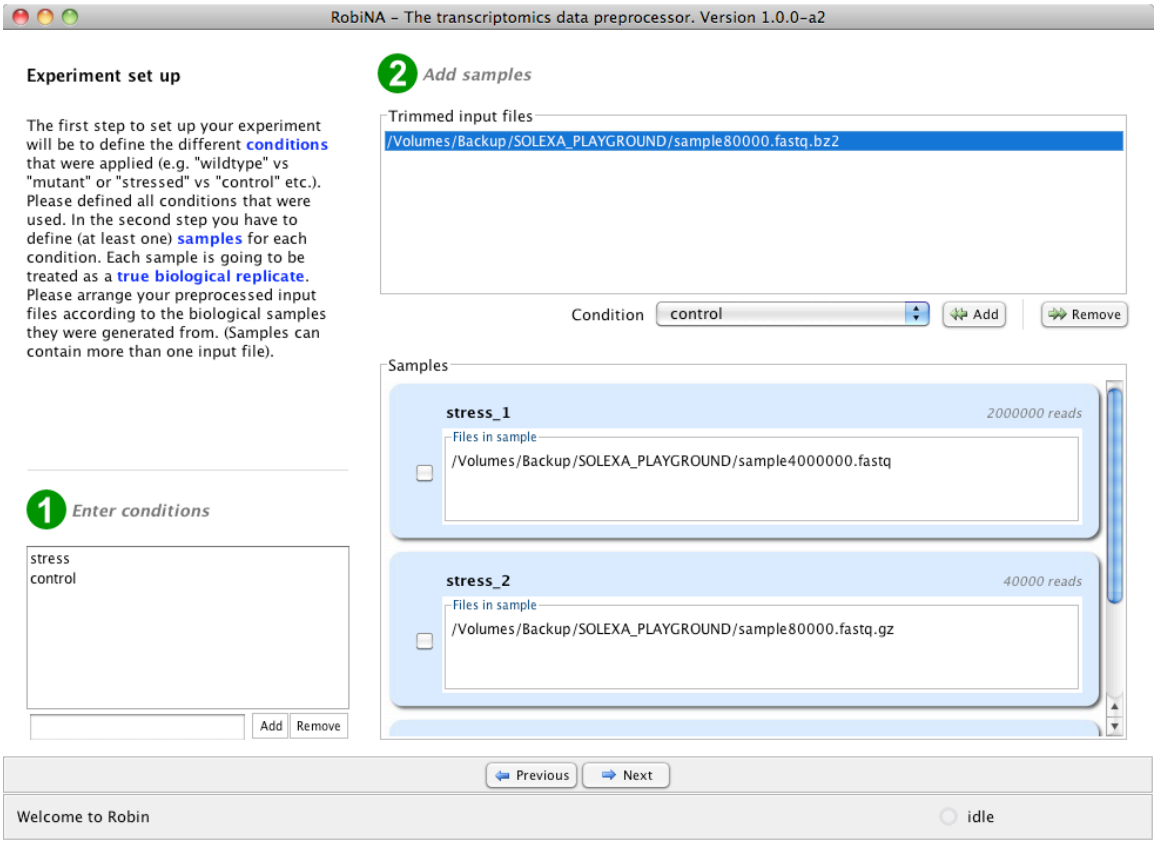

Figure 14: Experiment set up panel. In order to continue to the mapping step and generate a counts table for each sample, the samples and treatments in the experiment have to be defined.

More than one sequence file can be chosen per sample to account for raw data that was generated by sequencing the same original RNA sample more than once to obtain a sufficient amount of reads.

1.3.4 Read mapping

Since the final aim of the RobiNA analysis is to identify differentially expressed genes that are significantly responding to the different experimental treatments, it is necessary to compute a measure of transcript abundance from the reads in each of the defined samples. This is done by mapping the reads in each sample separately to a reference sequence (genome or transcriptome) and subsequently counting the number of reads that aligned unambiguously to each of the transcripts or genes of the reference. The intermediate result of this step is a counts table that contains a column for each sample and a row for each gene that has been see at least once in at least one of the samples. This table will be saved in the "detailed_results" folder of the project directory and can be easily imported into e.g. spreadsheet applications or other tools. The counts table will be used as the basis to estimate the expression level of each of the genes contained and hence is the most important input into the statistical analysis of differential gene expression. Figure 15 shows the mapping panel user interface – First, the user has to

choose whether a transcriptome or a genome is to be used as the reference sequence. When using a transcriptome, the data has to be supplied in the form of a multiple FASTA file containing the nucleotide sequences of all transcripts. While working with a transcriptome usually makes the mapping process faster, it is also biased, because only reads that are mapping to the known transcripts supplied in the FASTA file will be recorded. If the reference transcript set is incomplete and / or inaccurate (as might very well be the case for an ill described organism or a primary transcriptome assembly), a substantial fraction of the input reads might be lost because they map to transcripts that are not included in the reference set.

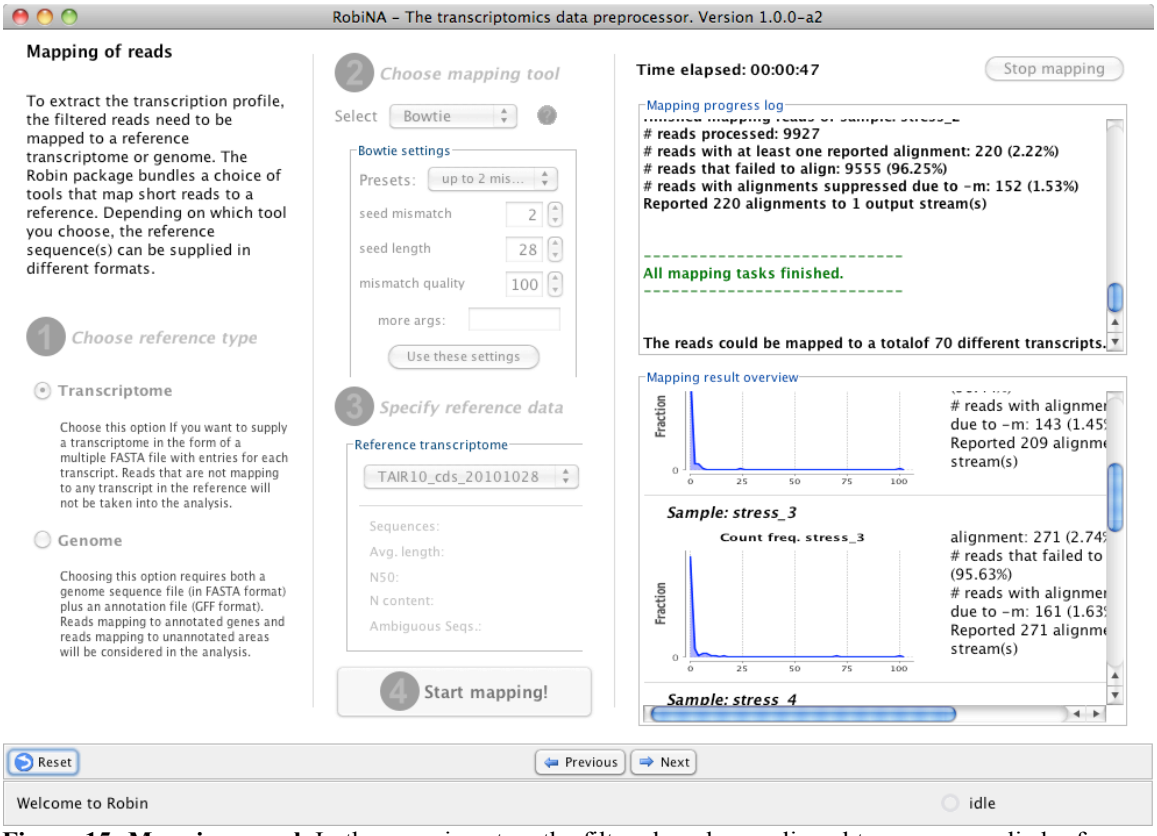

Figure 15: Mapping panel. In the mapping step, the filtered reads are aligned to a user-supplied reference sequence to determine how "often" each transcript was sequenced.

When working with a genome sequence as the reference, both the raw sequence data (again in FASTA format) and a matching annotation file in GFF3 format (http://www.sequenceontology.org/gff3.shtml) has to be supplied. After choosing your desired reference format, the next step is to configure the mapping tool. The current version of RobiNA uses BOWTIE [2] to align the reads. The settings for the alignment can be modified using the "Bowtie settings" panel. While there are a lot of options available for tweaking BOWTIE, RobiNA limits itself to controlling the mapping by 1) setting an option that will ignore any non-ambiguously mappable reads (-m1) and 2) allowing the user to control the alignment stringency by explicitly setting a) the maximal number of mismatches allowed in the seed region, b) the seed region length and c) the maximal sum of mismatch quality scores. The default setting will accept no mismatches

at all which is very conservative and might lead to a lot of well-aligning reads being discarded just because of 1 mismatch. Especially when working with reads that originate from e.g. a different ecotype (or other genetic variant) than the reference, this setting might be too strict. So we recommend to either use the 'up to 2 mismatches' preset or change the settings manually to even more permissive values.

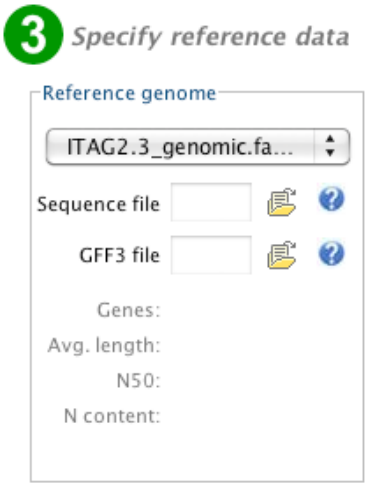

Figure 16: Reference genome panel. Users can either provide new genomic sequence (in a FASTA file) and annotation (in a GFF3 file) or choose an already generated reference index from the drop box.

To confirm the alignment settings and continue the workflow, click 'use these settings'. Before starting the actual mapping process, the reference sequence has to be provided. Since the generation of a BOWTIE sequence index can be quite time consuming when working with large reference files, RobiNA requires the user to do this only once. All indices that have been generated via the RobiNA application will be saved and are subsequently available via the drop-down box in the reference data panel (see Figure 16) so that the indices don't have to be created every time a new analysis is run. If a new genomic reference was entered, the indexing procedure will start as soon as the input is complete. If an existing index was chosen, the main mapping process can directly be started by clicking the 'start mapping' button.

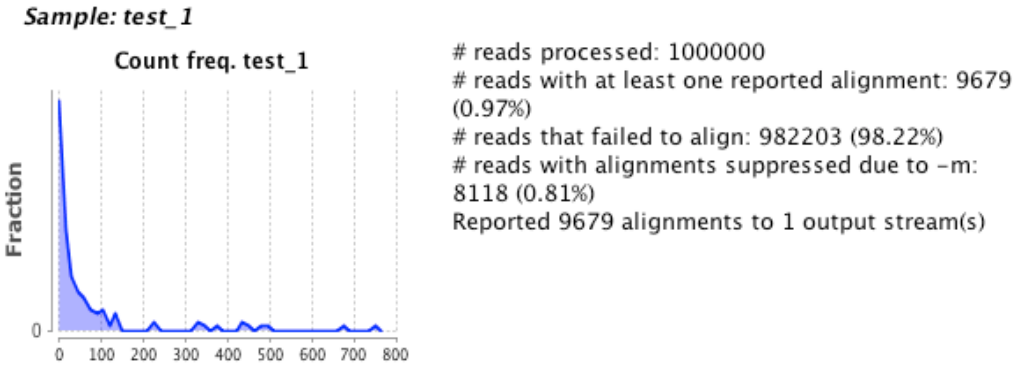

Figure 17: Short summary of single sample mapping. The graph on the left shows a scaled histogram of count observations. The expected shape of the plot would be similar to an exponential decay curve, since

most of the genes have a low count while only a few are counted very often. The right half shows a textual summary of the mapping process as reported by the BOWTIE alignment tool.

As soon as the mapping process is finished, the 'Mapping result overview' box in the lower right corner will show a short result summary for each of the samples. The summary consists of a count frequency histogram and further textual summary data giving the number of reads processed, discarded and unambiguously mapped to the reference. Figure 17 shows a typical count frequency histogram.

1.3.5 Statistical assessment of differential gene expression

After the mapping step, the counts table is computed and can be found in the 'detailed_results' subfolder of the project. To identify genes that are expressed differentially between two experimental conditions, the user first has to define which conditions or treatments are to be compared. This is done on the experiment designer panel (Figure 18). Each of the conditions or treatments entered previously will be represented by a blue box on the experiment designer panel. To define a comparison between two of them you simply have to draw an arrow by holding down the control key and then click-dragging from one box to the other. After releasing the mouse and control key, you will see a picture similar to Figure 18, which means that the comparison STRESS minus CONTROL has been defined. You can define any number of comparisons you are interested in. To delete an arrow, simply right-click on it and choose delete from the context menu.

The lower half of the experiment designer panel allows you to choose the statistical analysis method and modify the settings of the analysis. To-date, two different methods, based on two different Bioconductor libraries, are available:

1) **edgeR** [3]
   The statistical assessment methods implemented in the edgeR package are based on the assumption that the read counts recorded in a RNA-Seq experiment follow a negative binomial distribution. It uses the discrete reads counts recorded in the counts table for the assessment of differential gene expression (including correction for library size and compositional bias). edgeR employs empirical Bayes methods to estimate the gene-specific biological variation and thereby can also be used for experiments with little or even no biological replicate samples. **However, we want to emphasize that true biological replication should always be a main focus when designing an experiment**. The edgeR-based statistics computed by RobiNA uses exact tests as described by [4, 5]. Future releases of RobiNA will also provide the option to compute more complex multifactorial statistics based on generalized linear models according to [6]. For further in-depth information on the statistical approaches implemented in edgeR, please see the edgeR User's Guide

(http://www.bioconductor.org/packages/2.9/bioc/vignettes/edgeR/inst/doc/edgeRUsersGuide.pdf)

2) **DESeq** [7]
The method implemented in the DESeq package is essentially based on the same assumption of a negative binomial distribution of RNA-Seq data as the edgeR method. However it uses a slightly different approach for the computation of differentially expressed genes. Please see the publication cited above for an in-depth description of the method.

When the main analysis step is finished successfully, you can annotate the result data given that a matching MapMan mapping is available – please refer to section 4.4 for details as the process is identical for microarray and RNA-Seq data sets).

1.3.6 Result Browser

Once the analysis is finished, you can browse a summary of the whole analysis directly within the RobiNA application. The summary gives an overview of all steps of the workflow comprising input data, trimming, settings of the statistical analysis and various overview plots visualizing the results of the differential expression analysis. Additionally, full lists of differentially expressed genes, including the log fold change, and p-values are given.

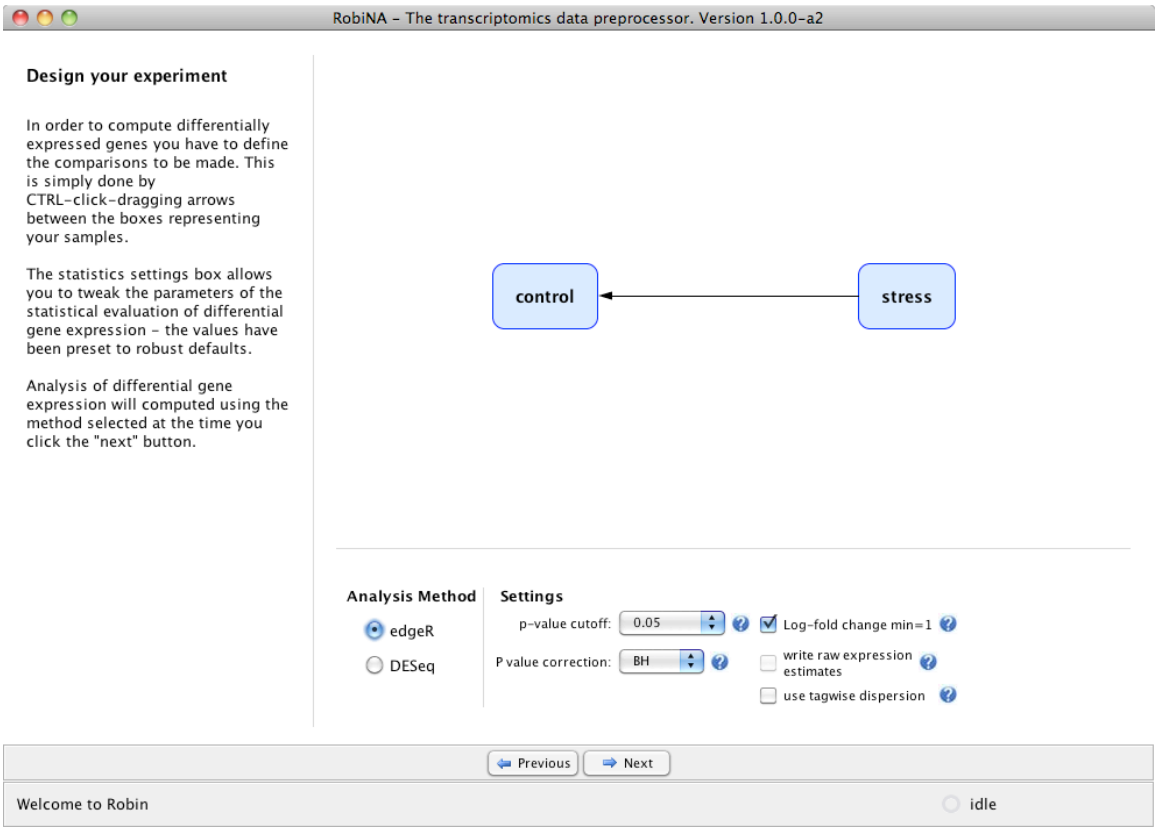

Figure 18: Experiment designer panel. Comparisons of interest between the different treatments that were entered earlier can be defined by simply drawing arrows between the boxes representing the different treatments. The lower part of the panel allows modifying the settings of the statistical assessment.

1.4 RobiNA Troubleshooting

1.4.1 Installation on Linux

When using RobiNA on Linux (or any other UNIX-based operating system), the RobiNA installer package will not include an embedded R engine and hence requires that R (R version 2.14.2 when working with RobiNA version 1.1.0) be installed.

The first time RobiNA is started, it will ask the user for the location of an R installation on the host system and try to install all required packages (if not already present). Depending on the configuration of the R installation and the permissions owned by the user running RobiNA this might cause problems, e.g. when the user lacks the permissions to install R packages in the standard R library path. This, however, can be resolved by setting a user-specific library path: Start an R session and type

```
> Sys.getenv("R_LIBS_USER")
[1] "~/Library/R/2.14/library"
```

to have R report the default location for the user specific library. In a standard installation, the reported directory might not exist, so you'll have to switch to a terminal and create the directory first. In e.g. a bash shell this would be accomplished by typing

```
mkdir ~/Library/R/2.14/library
```

(make sure to replace the path with the actual path reported on your system). If you subsequently try to install new packages, these should be installed in the user-specific library path. The next time RobiNA is started, it should be able to install all required packages and set up your local R environment for being used with RobiNA.

2 Analysing microarray data with Robin

2.1 Introduction

Robin represents an easy to use graphical interface for microarray (Affymetrix GeneChip, other single channel (e.g. Agilent) and two colour) analysis functions from R/BioConductor. It is available on all Java-enabled computer platforms that are also supported by the R Development team. The main objective of Robin is enabling the individual biologist to use state of the art microarray pre-processing and analysis tools that are provided by the BioConductor project without in-depth knowledge of programming in R. To this end Robin provides documented, standard workflows for the quality assessment, normalization and statistical analysis of microarray data originating from many commonly used technical platforms. These workflows should allow for the analysis of most experimental setups that are conducted in microarray experiments carried out in labs around the world.
This manual gives a detailed guideline through the Robin analysis workflows for different types of microarray experiments (e.g. Affymetrix, two colour, Agilent single channel…) and explains the concepts and methods of quality assessment, normalization and analysis of differential gene expression. The output that is generated by Robin can directly be imported into meta-analysis tools like MapMan and PageMan for further visualization and analysis of the data in a biological context or into Microsoft Excel.

2.2 In brief: What can Robin do for you?

You can use Robin for…

(i) quality assessment of your data
(ii) normalization of your microarray data
(iii) detection of differentially expressed genes
(iv) preparation of the data for an import into MapMan and/or excel
(v) generation of informative plots on your experiment

3 Preconditions and Glossary

3.1 Commonly used Terms

Robin helps in evaluating microarrays using advanced normalization strategies and statistics from R/BioConductor. Nevertheless, please bear in mind that most statistics and most normalization techniques make some strong assumptions and have some general terminology.
When dealing with microarrays, almost always one will deal with values which have been transformed by taking the logarithm to the base of 2. The reason is, that by logging the data, the data becomes roughly normally distributed (Gauss shaped), which then allows using tests, like student's t-test, making assumption about standard deviations etc..
Unlogged data is almost always **NOT** normally distributed, meaning t-tests are **NOT**

applicable (even though they might still perform reasonably well). Thus, a difference of 1 unit means a two-fold increase or decrease in expression.
Often data is not represented as treatment value and control value, but instead of **M** and **A**. Here, **M** stands for treatment minus control (on the log scale, being a division on the normal scale), and **A** stands for (treatment plus control)/2. So **M** is a measure of your treatment effect and **A** of the expression level of that gene.
(Actually another reason for using M and A values is that it is easier to see if values deviate from the zero line as if they were deviating from a line with a slope of one. Please see Figure 19)

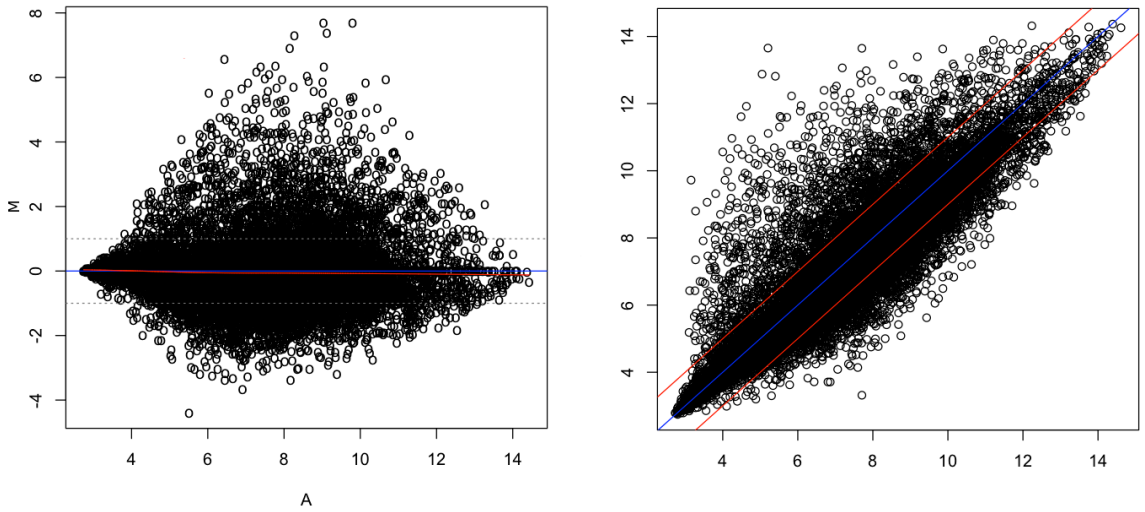

Figure 19: Comparison of MA plot versus Scatter plot of normalized expression values. The left panel shows an MA plot of the log2-fold changes when comparing two chips (M) plotted on the Y axis and the average log2 intensities (A) plotted on the X axis. On the right panel the same two chips' expression values have been plotted against each other. The MA plots gives a clearer representation of the cganges in gene expression when comparing the two chips.

3.2 Affymetrix Files

When dealing with Affymetrix chips, you will be confronted with .CEL and .CDF files, the former describes the scanned intensity for every spot (usually there are 2 times ~11 spots per gene). The CDF file describes where the spots for a probe-set are to be found on the chip, since these are not clustered to compensate for local effects such as bubbles, smears, etc.

3.3 Other single channel and two color data files

Data derived from other microarray experiments may come in a variety of different file formats depending on the microarray scanner hardware and software used. Robin supports direct import of generic file types that contain the data in text files with each column of data points separated by a special character (e.g. semicolon, TAB etc.). Import of generic data is managed by a generic data import dialog that allows you to specify

which column contains what kind of data. Using this dialog it should be possible to import arbitrary microarray data. Since the generic import mechanism does not work for some data formats (like the tab-separated raw text files produced by Agilent scanners), customized settings have been supplied to allow import of these formats. Please set the import type on the file import panel according to your microarray data type if it is listed. If not, try generic import settings. If these fail we will be happy to create a customized filter for your data, if you supply us with a sample of the format. When working with generic data you'll also have to know the layout of the chips you want to analyze – presets for commonly used chip types are already included in Robin. This list can be completed with your custom layouts.

3.4 Assumptions

The strongest assumption probably being, that not much changes in your experiment. I.e. the assumption is that let's say not more than 5, 10 % of your genes are changing and that thus everything is comparable.
If this assumption is violated, you may not get satisfactory results, or worse wrong results. To demonstrate this issue, just consider the probably oldest, easiest normalization, namely median centering. Here, one just subtracts the median of one experiment from each data point. In this extreme example, Gene1 and Gene2 are completely switched off.

	XP1	XP2
Gene1	10.2	0
Gene2	3.2	0
Gene3	4.5	4.7
Gene4	7.8	7.9
Gene5	9.9	9.8
Gene6	10	10.2
median	8.85	6.3

Table 1: Experiment before normalization

	XP1	XP2
Gene1	1.35	-6.3
Gene2	-5.65	-6.3
Gene3	-4.35	-1.6
Gene4	-1.05	1.6
Gene5	1.05	3.5
Gene6	1.15	3.9

Table 2: Experiment after normalization

As an effect, Genes 5 and 6 seem to be upregulated, even though they were unchanged. These effects would disappear in this case, if also some genes were turned on, which often might be the case, but if you have strong suspicions, that very many genes change, and/or that these change in one direction only, you might have to consult an expert statistician.

4 Walkthroughs

The following sections of the manual provide step-by-step walkthroughs through microarray data analysis using Robin. Since Affymetrix data analysis is the most common task it is described in all detail. The workflows for two color and generic single channel analysis resemble the Affymetrix workflow and are hence described in an abbreviated fashion, focusing on the steps that are different from the Affymetrix analysis procedure.

A common step at the beginning of all workflows is choosing the project directory. This directory will be used to store all data and results related to the analysis at hand. If you want to continue or modify a previous analysis, you can do that by simply choosing an existing project directory. Robin will import the data and settings from the project folder allowing you to conveniently modify the settings and run a new analysis. To distinguish the new analysis results from the imported data, Robin requires you to specify a name extension that will be appended to the imported project's name to generate a new folder holding the results of the modified analysis run. If you import e.g. project AFFYTEST and specify "NEW" to be the extension, the new results will be placed into the directory AFFYTEST_NEW to make sure that previous results won't be overwritten.

PLEASE NOTE that the project import feature only works with analysis projects that have been generated using Robin version 1.1 or higher. Trying to import a project from an older version will generate an error message. The version number is always displayed in the title bar of Robin's main window.

4.1 Using Robin to analyze Affymetrix microarray data

Firstly, when using Robin, you have to localize your CEL files. Robin comes preinstalled with specialized CDF files for a small selection of organisms (arabidopsis, maize, lotus, yeast etc.), when dealing with other organisms, you will need an internet connection, so Robin can use the Bioconductor framework to install missing CDF files. The INFO button can be used to display some details about the imported CEL files such as microarray type, algorithm parameters and all the technical data included in the header section of the CEL file.

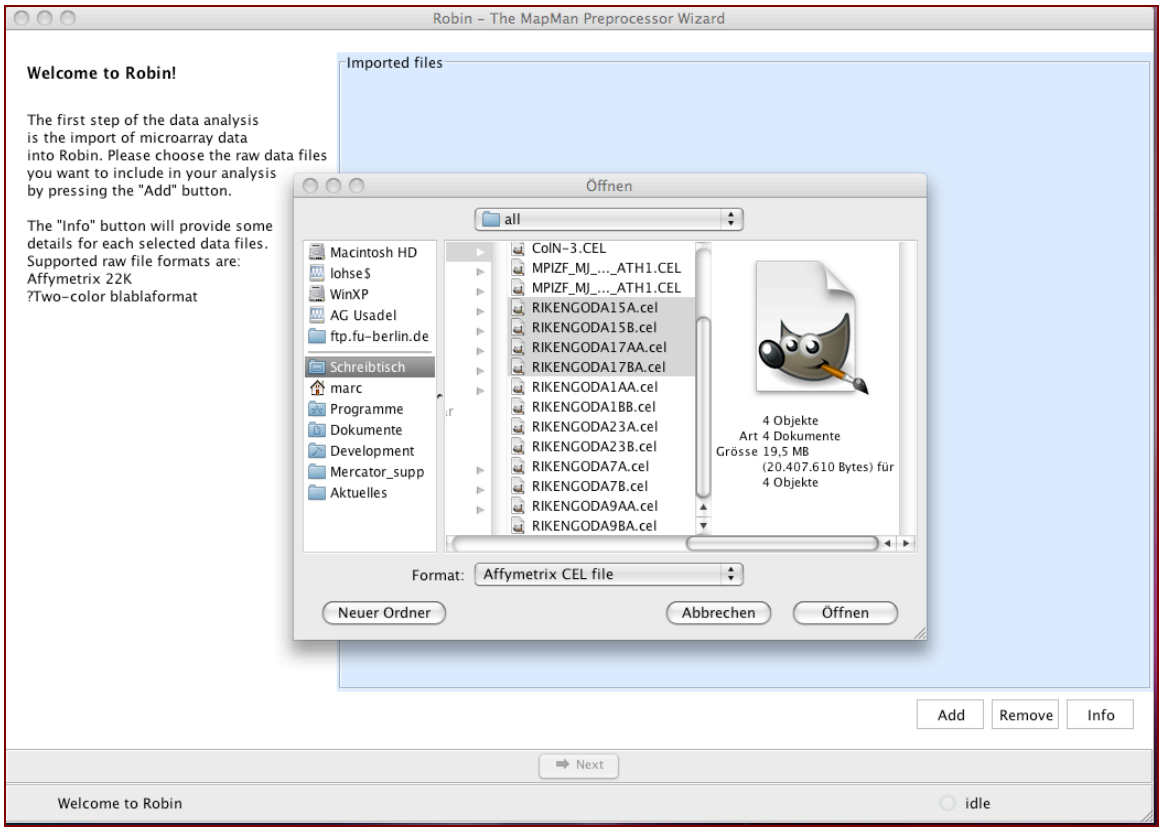

Figure 20: Importing CEL files into Robin

After having selected your CEL files, you are presented with various options to investigate into the quality of the arrays.

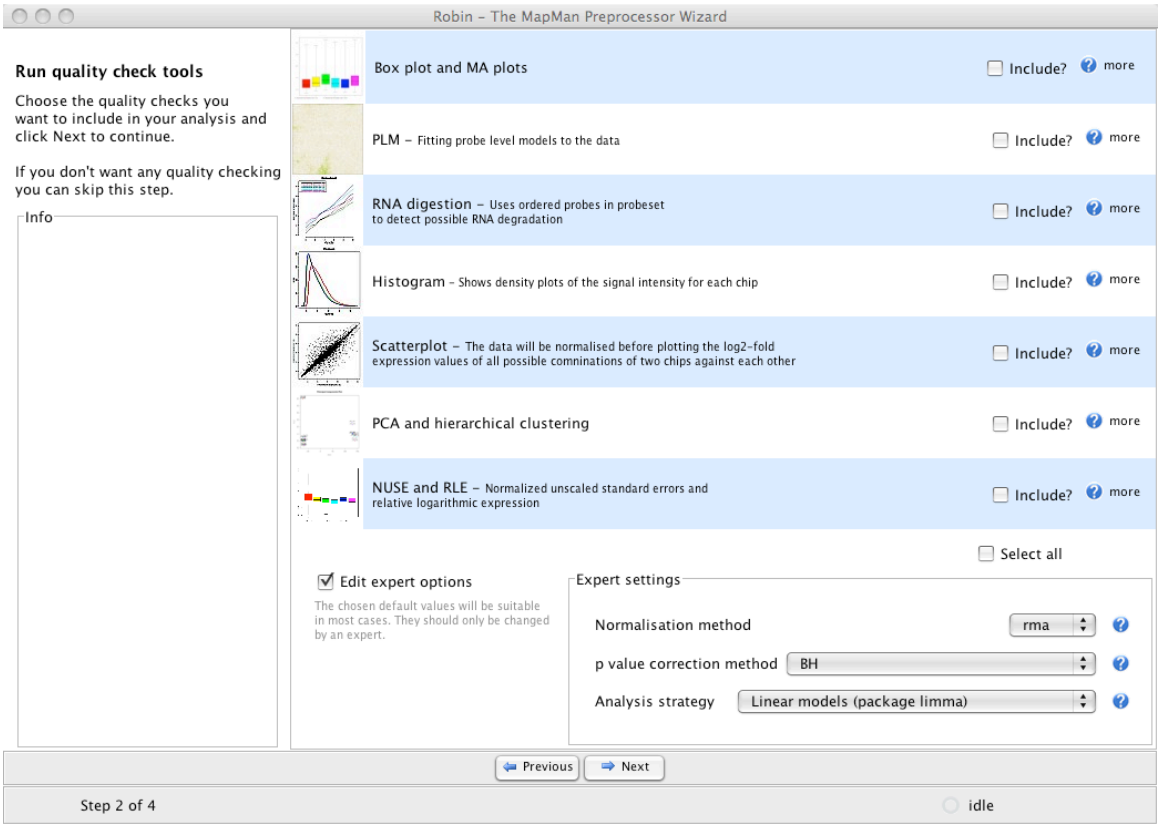

Figure 21: Quality control options available for Affymetrix(r) arrays in Robin.

The ”expert options“ box is not shown by default – the preselected values there can be used to correctly analyze most standard experiments. If you activate the expert settings box you can explicitly choose which normalization method, p-value correction and general analysis strategy is to be used on your data.

4.1.1 Quality Control

After running the chosen quality control methods on your data, Robin will present a summary page showing thumbnails of the generated plots (see Figure 22). Clicking on the individual rows will open the images in full size and offer a possibility to save the image. **PLEASE NOTE**: You don’t have to open each image individually and save them manually – all generated quality control plots will automatically be saved together with the results of your analysis.

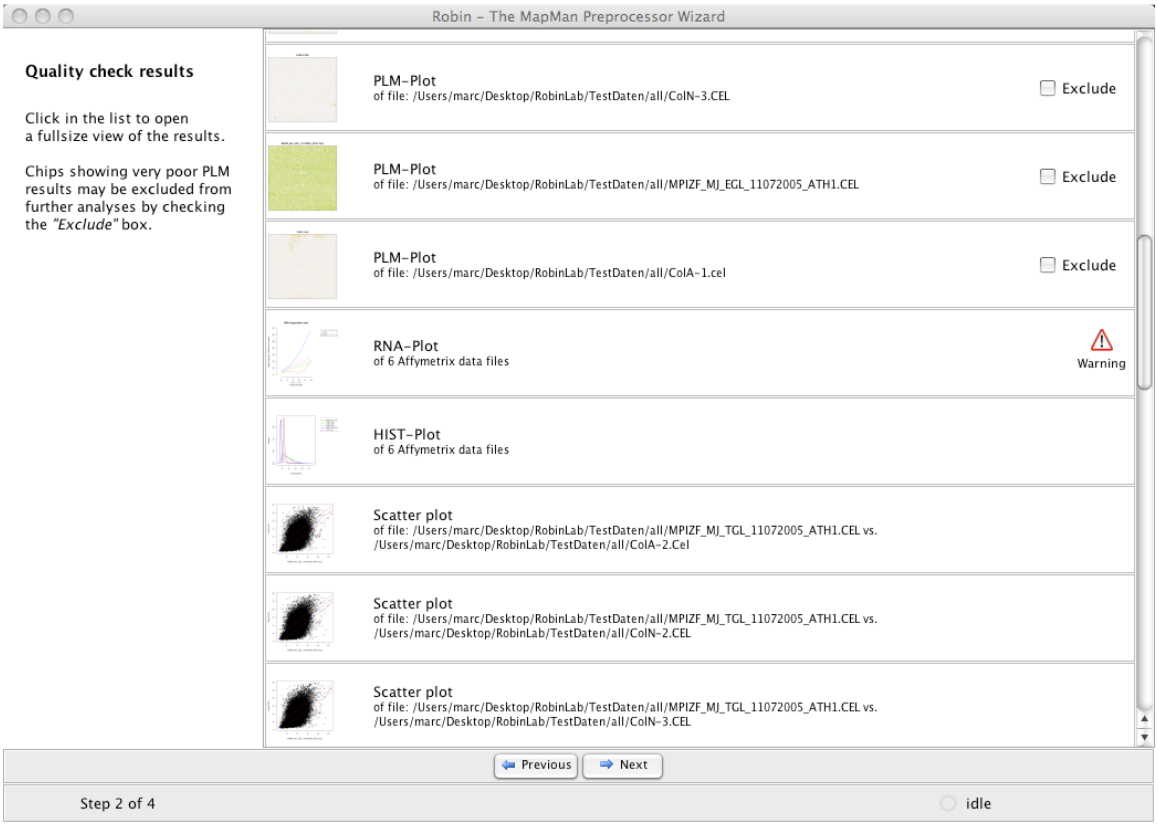

Figure 22: Quality analysis summary page.

Some of the quality assessments functions may have issued warnings – clicking on the small warning icon will open an info panel that tells you more specifically why the warning was generated. For example the RNA degradation analysis may have identified chips that display slopes higher than the accepted threshold or whose slopes deviate by more than 10 per cent from the median slope (see section 5.1.5 for details). Individual chips displaying an extraordinarily bad quality in the PLM-Plot (see 5.1.3) or MA Plot (see 5.1.2) can be excluded from further analyses by checking the "Exclude" box. Section 5 describes all available quality control methods in detail and gives examples of good and bad quality check results.

4.1.2 Experiment design and statistical analysis

The next step in the analysis workflow is the assignment of the chips to groups of biological replicates. NOTE: Robin analyses all replicates as biological replicates – there is no way implemented yet that allows for proper consideration of technical replicates. Please be aware that if technical replicates are imported the statistical test outputs will not be sound any more.

You can choose a descriptive unique name for each group of replicates (like "mutant", "wildtpye" etc : see Figure 23). After sorting the chips, clicking "next" will proceed to the graphical experiment designer. Here the user can set up the comparisons that are to be made by CTRL-click-dragging connections between the groups (see Figure 24).

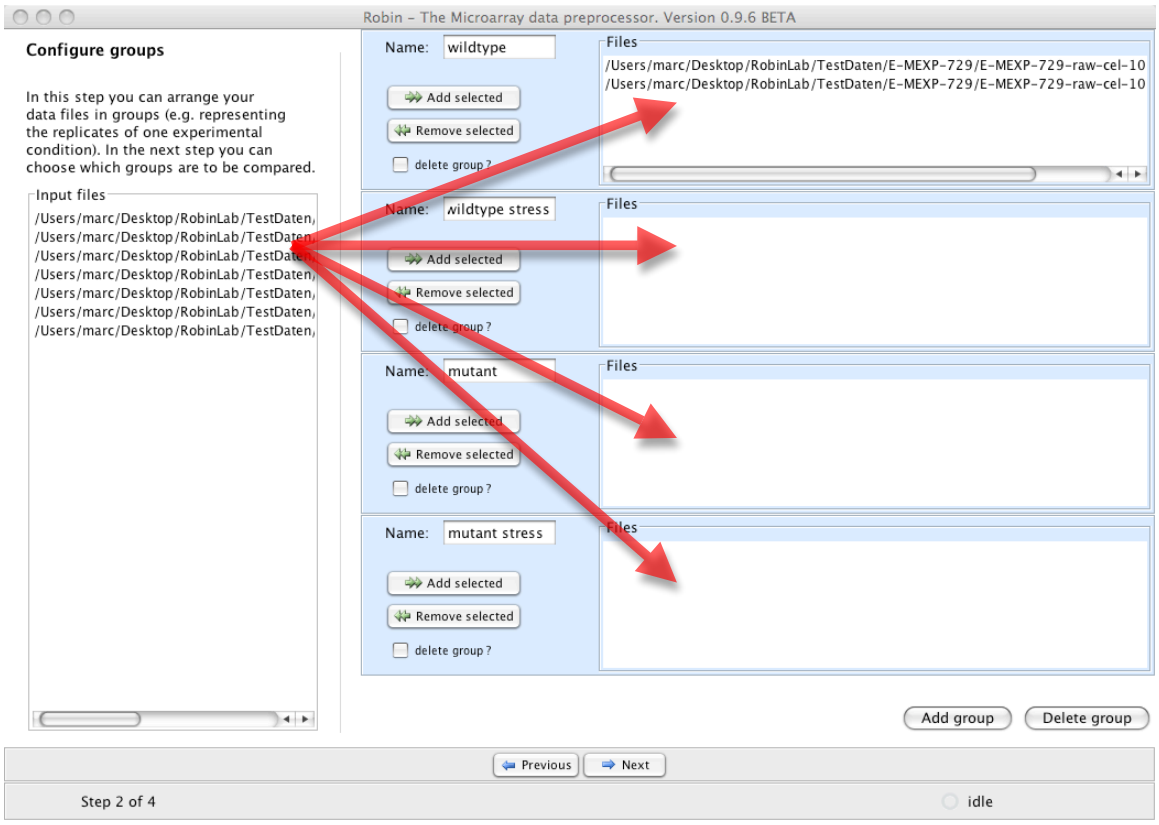

Figure 23: Sorting of replicate experiments into named groups.

4.1.2.1 Single factor analysis

If only a single experimental factor is varied in the experiment (e.g. when comparing "wildtype" and a "mutant" genotype under identical environmental conditions), direct comparisons between the groups are defined by simply dragging an arrow from the "wildtype" to the "mutants" box on the left panel of the graphical designer screen (Figure 24.1).

4.1.2.2 Two factor analysis

If experiments with more than one varying experimental condition are to be analysed the user can combine groups into "meta groups" and define comparisons of meta groups by dragging connections between them. First, the user has to link two groups of replicate chips by an arrow defining the direction of the comparison between them. To define a "meta group", the user has to select these two simple groups of replicates by clicking or dragging a box around them and then click "create metagroup". An orange box that is named after the comparison between the two selected groups will be added to the designer pane (see Figure 24, panel 3). Subsequently, comparisons can be defined between "meta groups" by simply drawing arrows accordingly.
In the example experiment shown in Figure 24, mutant and wild type plants were compared both under stress and normal conditions - so the experiment varies in two dimensions with genotype (wild type or mutant) being one factor and treatment (stress,

no stress) being the other. The first four direct comparisons (Figure 24.2) will yield the genes that are responding to the treatment in the wild type ("wildtype – wildtype stressed) and the mutant ("mutant – mutant stressed"), which genes respond differently between the genotypes under normal conditions ("wildtype – mutant") and stress conditions. The fifth comparison defined between the meta groups named "widtype – wildtyope stressed" and "mutant – mutant stressed" will extract genes that generally respond differentially to stress in the two genotypes ("(wildtype – wildtype stressed) – (mutant – mutant stressed") – this is also referred to as the *interaction term;* see Figure 24.3).

PLEASE NOTE: The direction of the arrow specifies the direction of the comparison. When the arrow points from the wildtype to the mutant this should be read as "wildtype minus mutant". Genes showing a higher expression in the mutant when compared to the wildtype will accordingly yield a negative log2-fold change value as a result!

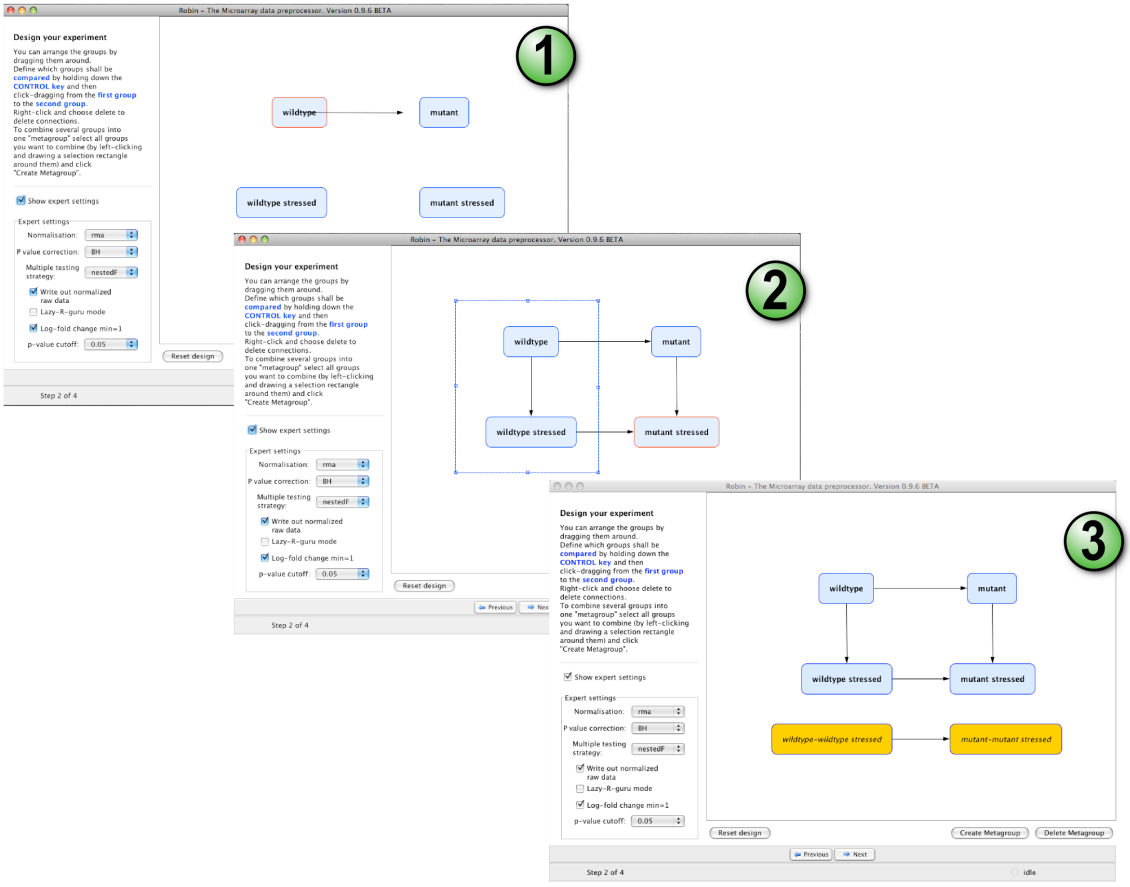

Figure 24: Setting up the experiment using Robins graphical designer.

The experiment designer panel also offers an expert option box that enables the experienced user to influence specific parameters of the statistical inference. By default, the normalization method used for the main analysis will be the same that was chosen on the quality check panel (see Figure 24; if nothing was changed the default will be robust

multi array averaging - RMA) to ensure consistency between the quality check and differential expression statistics. The user can define significance cut-offs like discarding genes that show a log2 fold change in expression lesser than 1 (i.e. less than 2-fold up- or down regulation) and genes showing a p-value greater than e.g. 0.05 (i.e. 5% false discovery rate is accepted). A choice of multiple testing methods is available for the inference of differentially expressed genes:

1) "separate" – Does the multiple testing for each comparison (contrast) separately. Using this method, each specific comparison will always give the same result irrespective of the set of comparisons being made in the analysis. It is the simplest method available as it does not consider multiple testing adjustment between the comparisons and assumes the same raw p-value cut-off for all comparisons (which might be very different).

2) "global" - Implements multiple testing correction across all comparisons and probes simultaneously ensuring a consistent p-value cut-off across all comparisons.

3) "hierarchical" – Does p-value adjustment first for all genes and then across comparisons, which offers more statistical power to control the family-wise error rate when using the method described by [8] for p-value adjustment.

4) "nestedF" – First does p-value adjustment for all genes and uses a nested F test to classify the comparisons as significant or not for the selected genes.

Users that are familiar with R programming can activate the "preview R script"-mode in which all scripts generated by Robin are shown in an internal editor for review and modification prior to execution. Even if this option was not chosen, all R scripts generated by Robin will be written to the "source" folder in the final output directory. When the design step is completed, clicking the "Next" button will first open a file browser asking for a location to save the results to and then move on to the execution of the analysis. After completing the calculations, Robin will show a summary of the warnings generated during the workflow (if any) and offer options to exit, restart or modify the current experiment.

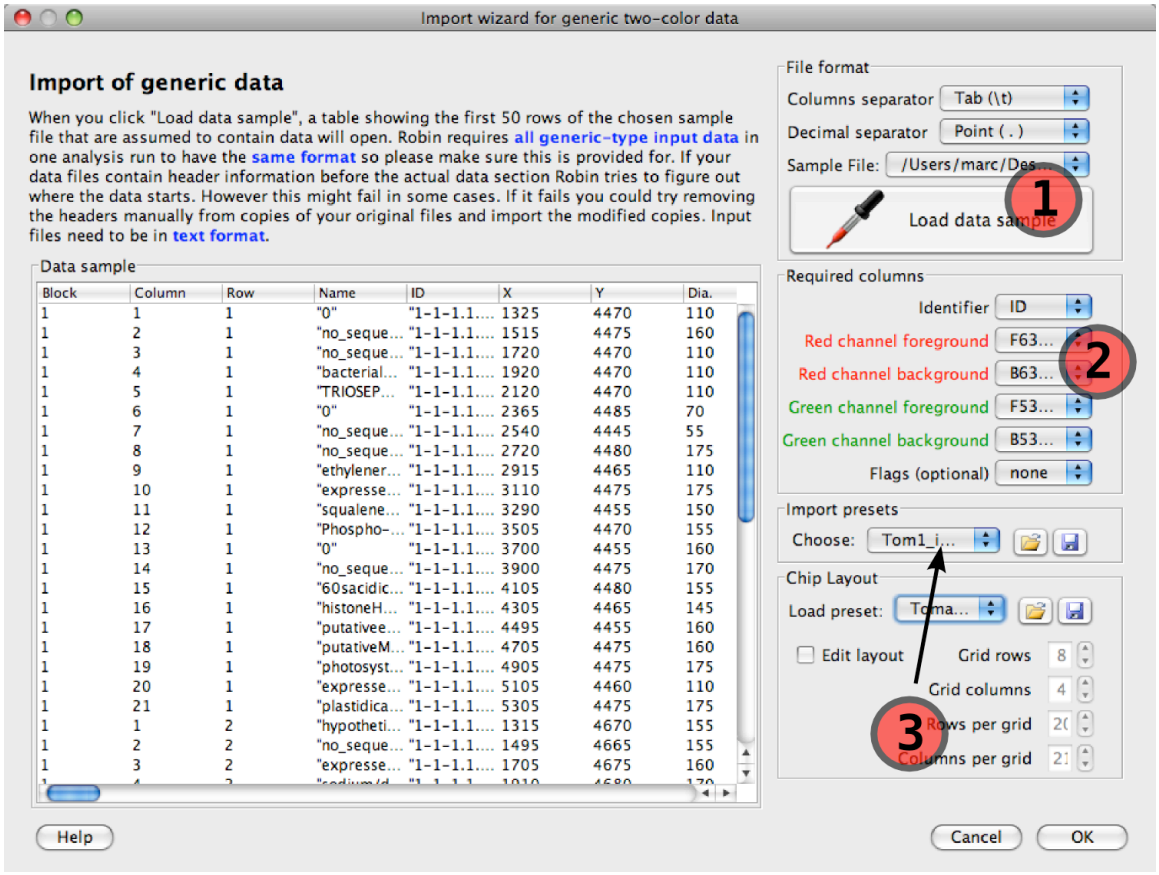

Figure 25: Two color data import wizard. Robin automatically removes header sections from different tabular file formats and extracts the column headers. The user has to define which column contains which data (1) by assigning the proper column names to the required column fields. After choosing a chip layout from the list of layout presets (or defining a new layout and saving it as a preset; see 2), the user can save the import settings (3) and reuse them later when importing data of the same format.

4.2 Analysing two-colour microarray data

The first step when working with two colour microarray data is data import. Robin provides a wizard dialog that helps the user to import various import formats with the only restriction that the data has to be provided in plain text format (.csv, .tab etc). Loading MS Excel worksheets directly is not supported (yet). Aside from this any kind of tabular data can be used. When importing data, the user only needs to know which column separator was used. Layouts of frequently used microarray types are included as clickable presets in the layout preset list – if the layout of your favorite chips is not included in the list, you can define a new layout and save it for later use. The minimal data required to analyze two color chips is an identifier uniquely identifying the oligos / cDNAs spotted on the chip and the red and green channel foreground and background signal intensities. The table view on the left half of the import dialog facilitates choosing the column containing the required data, and after specifying the column names under "Required columns" the information needed to import the data is complete. Robin will create copies of the input files that are stripped off any header text and checked row-wise

for data format consistency. The processed input files will be placed in a separate folder in the output folder.

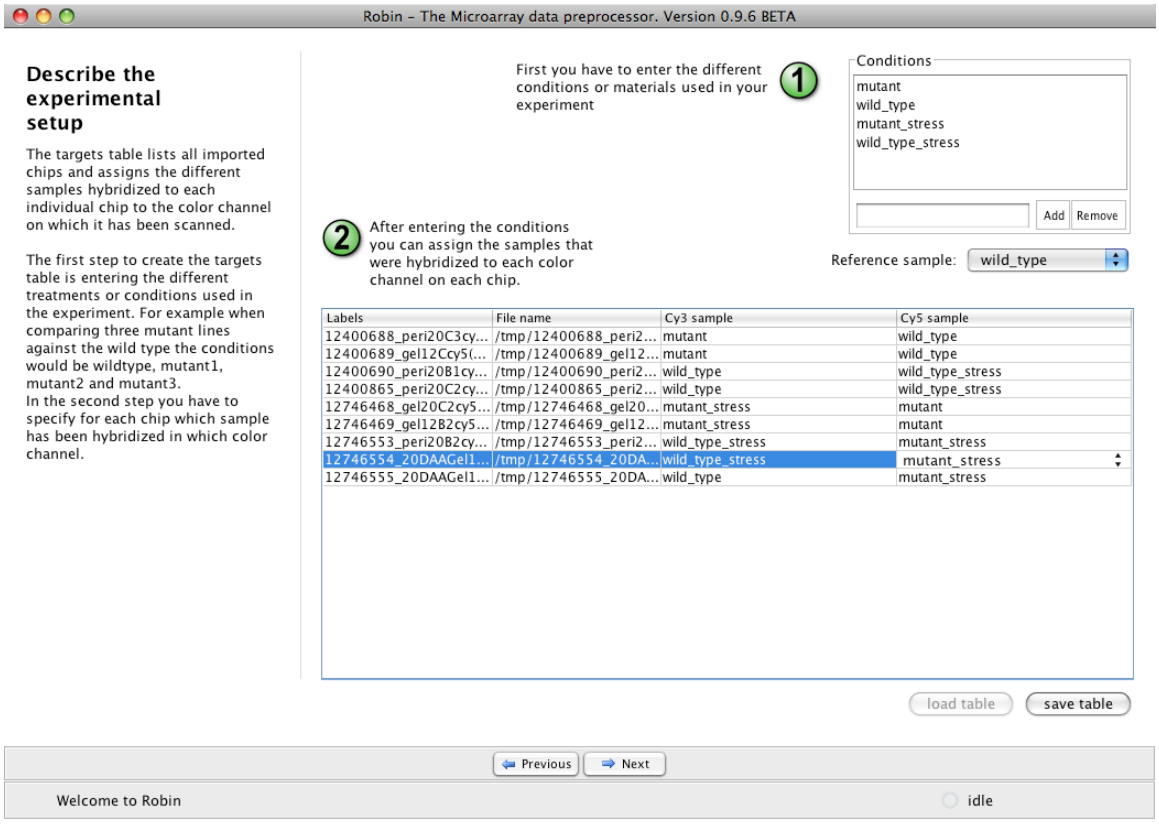

Figure 26: Defining the RNA targets table for two-color microarrays

The next piece of information Robin needs is which different RNA samples (RNA targets) have been hybridized to which channel on which chip. This can be conveniently entered on the targets table panel (see Figure 26). Robin will run some checks on the input to assert consistency. Analogous to Affymetrix data analysis the next panel provides a choice of quality check methods adapted to two color arrays and an expert settings box granting deeper control of the analysis parameters (See Figure 27).
Each step of the normalization process, namely background correction, within-array normalization and between-array normalization can be configured separately to o

Quality check results will be summarized in a list resembling Figure 22. Depending on the amount of factors being varied in the experiment (i.e. the amount of different RNA samples hybridized) clicking “Next” on the quality check panel will either directly start the main analysis (e.g. in a simple two sample comparison) or open the graphical experiment designer panel (see Figure 24). Experiment layout is done exactly as for Affymetrix arrays – please refer to section “Experiment design“ for a detailed description.

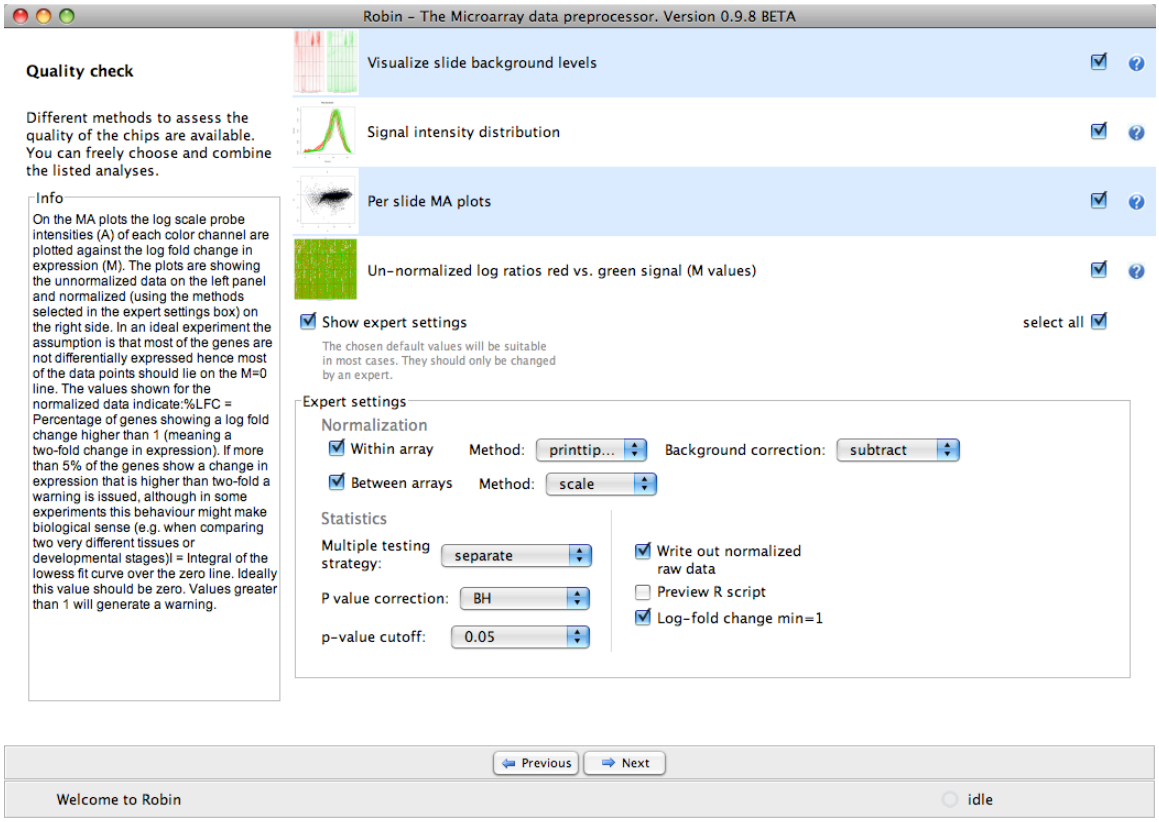

Figure 27: Quality check and expert settings for two color microarrays

4.3 Analysis of generic single channel arrays (e.g. Agilent)

Analysis of generic single channel arrays resembles the workflow for two color chip analysis in the largest part. The flexible import dialog (see Figure 28) allows for configuration of any tabular text file based data. Please note that you have to specify whether the data originates from an Agilent scanner prior to import to make sure that Robin can correctly remove the header section of the data files. Robin will process the input according the configuration chosen in the import dialog and create cleaned-up working copies, leaving the original data untouched. In the next step, analogous to Affymetrix data analysis, several assessment methods can be chosen to investigate into the chips' quality. Since most generic single channel chips are not based on a probeset design (several probes per target) but only contain one probe per target transcript, the probeset specific quality checks available for Affymetrix arrays (i.e. PLM-based analyses, RNA degradation plot) cannot be used.

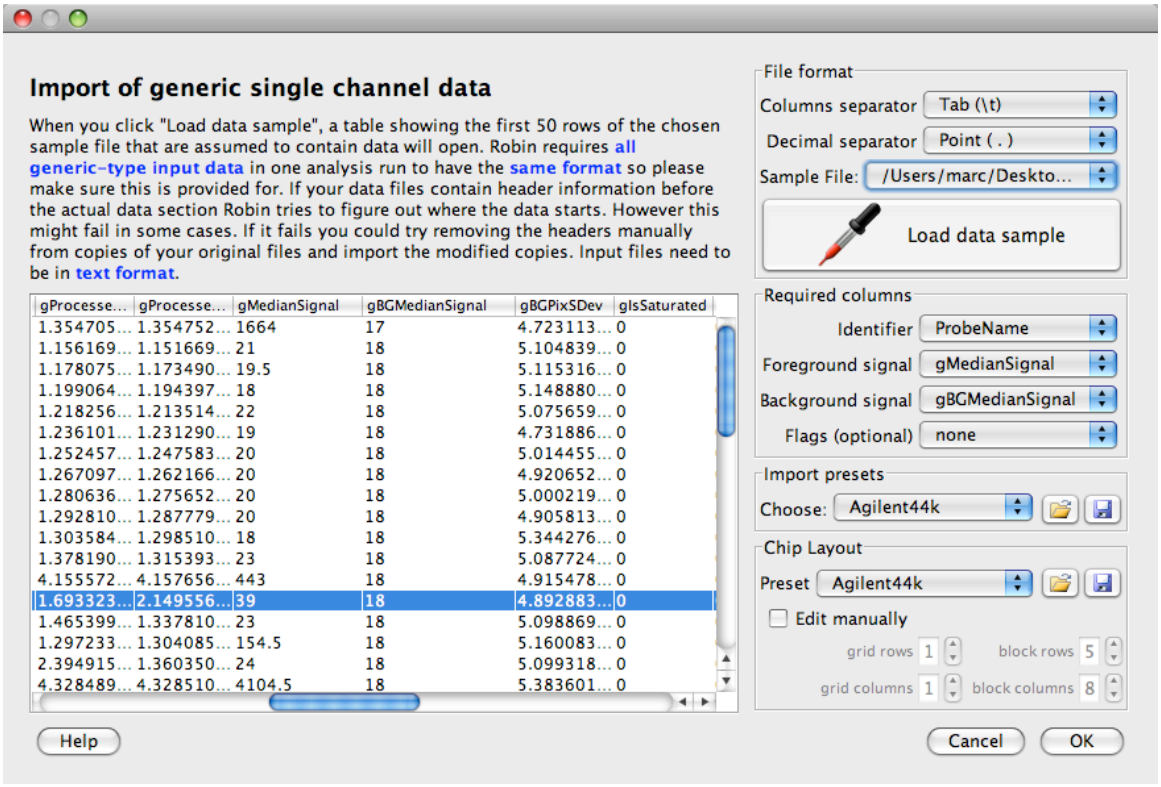

Figure 28: Import dialog for generic single channel microarray data.

Following the review of quality check results as depicted on Figure 22 and described in section 4.1.1, the individual chips have to be organized into groups of biological replicates. Depending on the statistical analysis strategy chosen (rank product- or linear model-based) two or more groups can

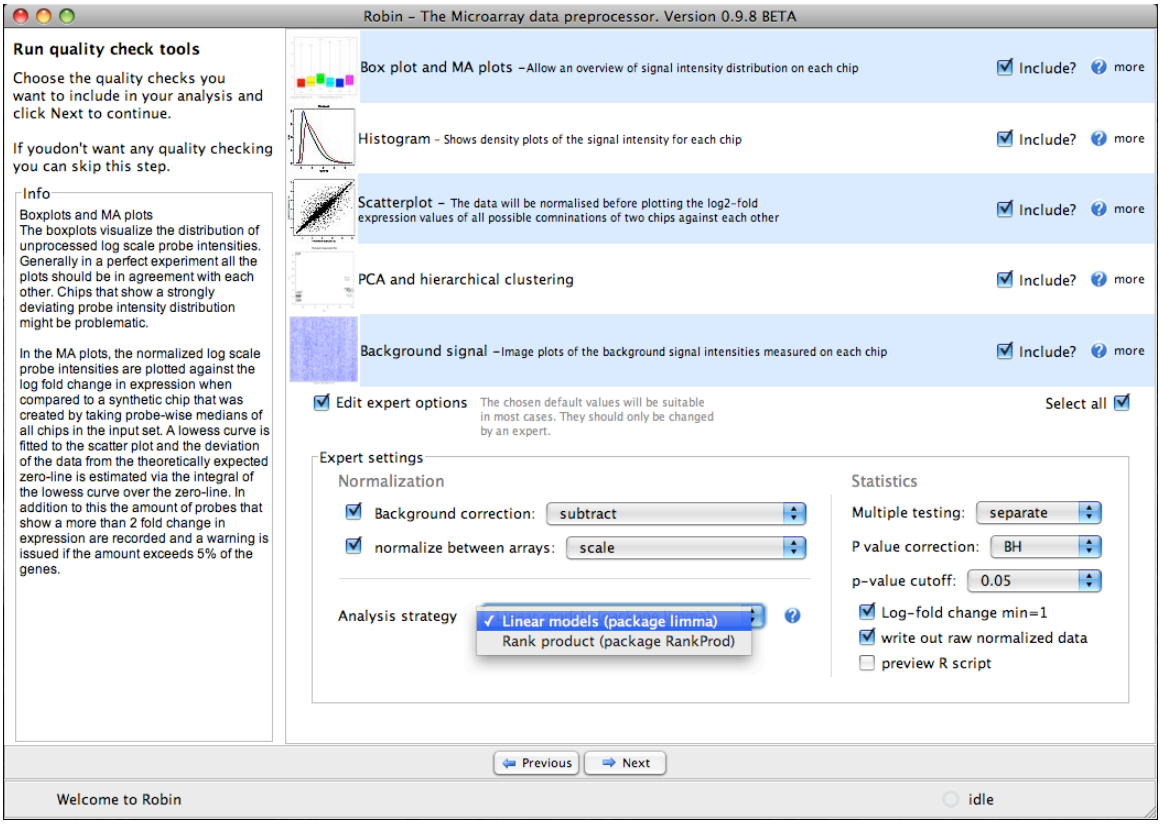

Figure 29: Quality check and expert settings panel for non-Affymetrix single channel microarrays

4.4 Functional annotation of the results

The MapMan project provides functional classifications, so called "Bins", for a wide range of completed genomes and microarray platforms. These classifications are available in the form of mapping files that contain entries for each gene (probe; probeset) assigning them to functional Bins. Multiple assignments are possible in cases where e.g. a transcription factor is member of a certain family of transcription factors (e.g. MADS box factors) and is known to be involved in a specific biological process (e.g. flower development). All available mapping files can be freely downloaded by academic users via the MapManStore webpage (http://mapman.gabipd.org/web/guest/mapmanstore). At the end of the analysis workflow, Robin will show the result annotation dialog (see Figure 30) asking the user whether functional annotation information should be merged into the results. To integrate the annotation, the user simply has to choose the correct mapping file from the drop-down list and click "Annotate". If an appropriate file matching the microarrays platform used is not included in the list, it might be available for download in the MapManStore. In case the MapManStore is also not containing a mapping file for your favorite platform please contact us via the MapMen website's support forum (http://mapman.gabipd.org/web/guest/forum). Robin tries to interleave the chosen mapping file with the results and finally displays an informative summary giving the numbers of identifiers found in the mapping and the results file that could be matched. If the mapping file that was chosen is not compatible with the microarray platform that was used to generate the data Robin will issue a warning. Theoretically, if

mapping file and chip type match perfectly, 100% of the identifiers in the result file should be contained in the mapping, however raw result files often contain several control spots/probesets that are not included in the mapping files (that are only containing functional information for genes). This might lead to numbers lower than 100% but usually above 90%.

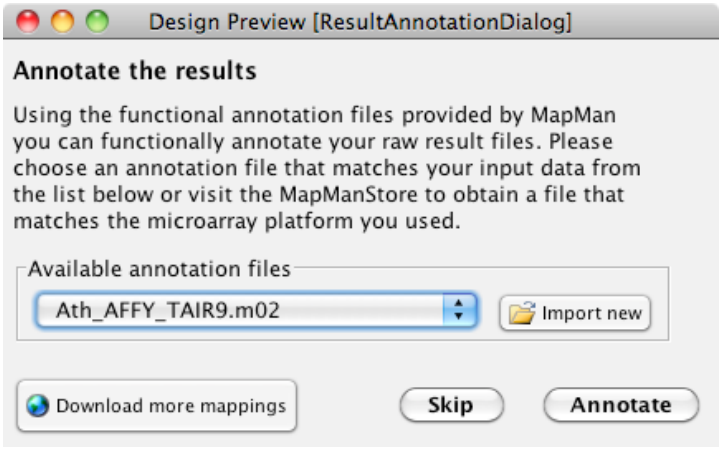

Figure 30: Result annotation dialog. If the microarray platform is supported by the MapMan project, the results can be augmented by adding MapMan functional classification information. The Robin installer package contains a variety of annotation files for common microarray platforms – more can be downloaded from the MapManStore repository

5 Chip quality assessment

When analysing your own primary microarray data or reanalysing data that is publicly available the first step is quality assessment. Individual chips displaying a very bad quality might strongly impact the final results of your microarray experiment and hence lead to incorrect biological assumptions. Chip quality can be affected on different levels and Robin offers a range of informative plots that cover many different aspects of the chip data quality. In the following section these methods will be described in detail.

5.1 Affymetrix chip quality checks

5.1.1 Analysis of signal intensity distribution

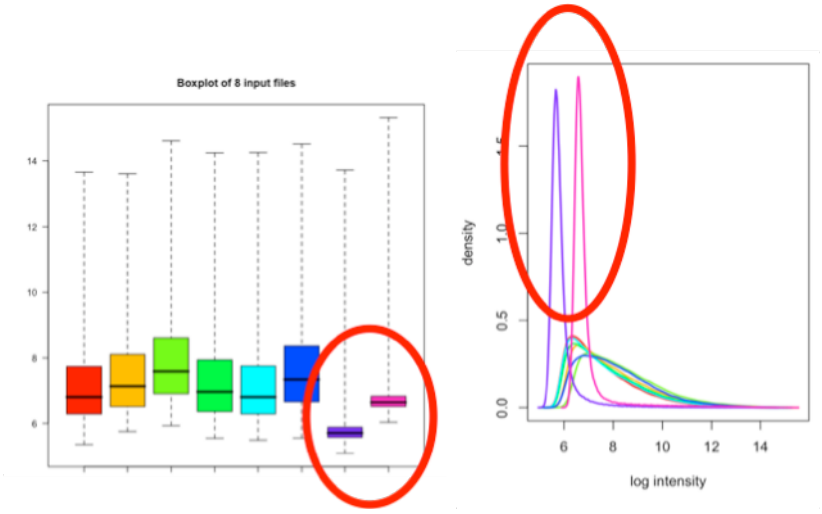

Figure 31: Box plots (left panel) and smoothed signal intensity densities (right panel). The red circles highlight individual chips that show strong outlier behaviour indicating low quality.

Box plots of the unnormalized expression values on each chip give a global overview of the signal intensity distributions. Ideally all chips should have a comparable distribution already before normalization (see Figure 31, left panel). Another way to visualize the distribution of signal intensities is plotting smoothed histograms of the (log2) signal intensity of all perfect match (PM) probes (see Figure 31, right panel). The red circles point out outliers.

5.1.2 MA plots

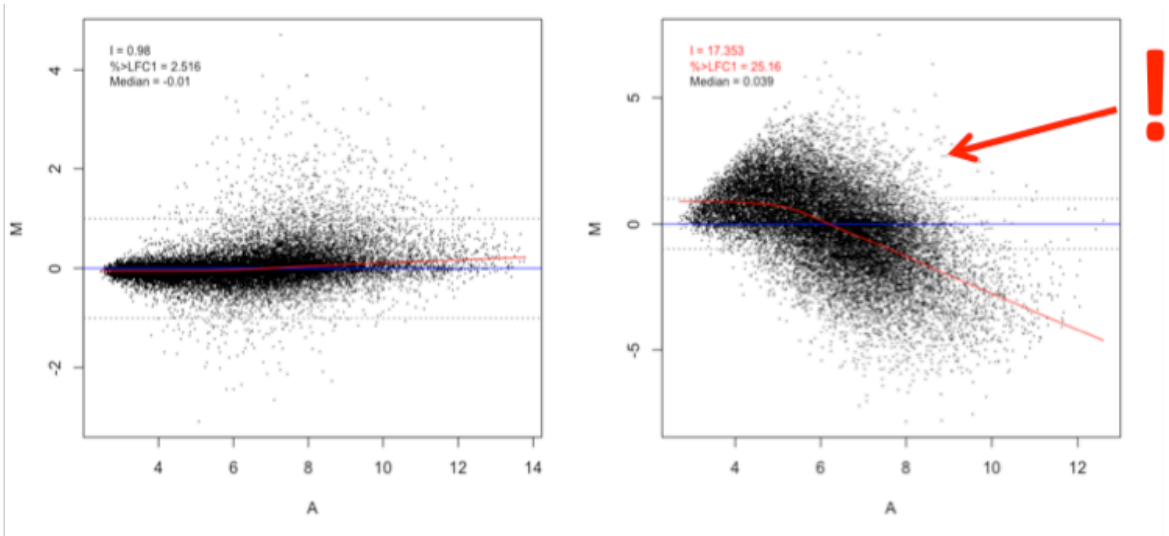

Figure 32: MA plots and box plots. The left panel shows an unobjectionable behavior while the data displayed on the right panel strongly deviates from normal values. In the box plot (see Figure 31) the two highlighted chips are also clearly showing an outlying intensity distribution.

On the MA plots, the average log2 probe signal intensity **A = ½ * (logR + logG)** is plotted against the log2 fold change in expression **M = logR – logG.** In the case of Affymetrix and other single channel chips G is a synthetic chip created from the median expression values of all chips in the input. For two-color chips the M values are calculated as the log2-fold ratio of the normalized red and green signal intensity. Based on the assumption that most of the genes will not show differential expression. Robin will issue a warning if more than 10% of the genes show a greater than two-fold change (log2 fold change of 1, resp. **M >= 1** or **M <= -1**) in expression. The actual percentage of genes showing a higher than two-fold change in expression is shown on the plot as "%>LFC1". To capture artifacts that are related to the signal intensity **A**, a lowess fit curve over the data points is calculated (see Figure 32). If the integral of the absolute values of the lowess curve over the zero line is greater than 1 another warning is generated indicating that there seems to be a signal intensity-dependent artifactual effect (the integrals value is shown on the plot as "I"). The median **M** value also given on each plot is usually less informative as can be seen on the right panel of Figure 32 where the median shows a normal value while the data quality is severely affected. MA plots are available for all microarray types.

5.1.3 False color images of probe level model weights

A linear model is fitted (using RMA style, more later) to your probeset (i.e. your 11 probes), using the boundary, that the effect of all probes in each probeset is zero.
Weights are attached to the different probes in each probeset, low weights are colored in green (i.e. they were not important for the model), and high values in white.

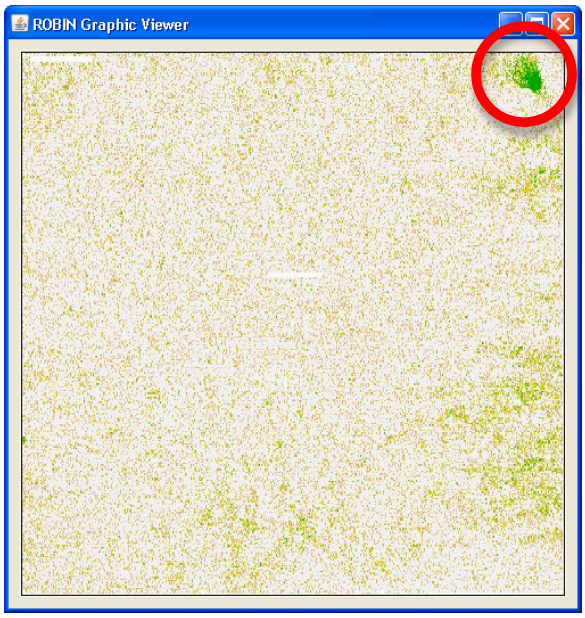

Figure 33: PLM weight image. Here a potential artifact is visible in the upper right corner.

For some examples of probe level model (PLM) image plots showing different artifact have a look at: http://plmimagegallery.bmbolstad.com/. The weights applied to each probe are visualized as pseudo chip images (see Figure 33). Areas on the chip that show consistently low probe weights might indicate technical problems cause e.g. by washing, dust on the chips or scanner malfunction. PLM-based analyses (pseudo images, NUSE and RLE, see next section) are only available for Affymetrix chips.

5.1.4 Normalized unscaled standard error and relative logarithmic expression

The normalized unscaled standard error (NUSE) plots show the standard error estimates of probe level models for each probeset standardized across all chips so that the median standard error for each probeset is 1. The NUSE plot visualizes the distribution of the standard errors for each individual chip. Chips showing a consistently increased standard error are probably of lower quality. The relative logarithmic expression (RLE) is computed by comparing the logarithmic expression of each probeset on each chip to the median expression of this probeset across all chips. According to the assumption that most of the genes are not differentially expressed under a given treatment, the median RLE value should be zero. Individual arrays showing a deviation of the median from the zero line and/or increased spread on a box plot of the RLE values are presumably of low quality.

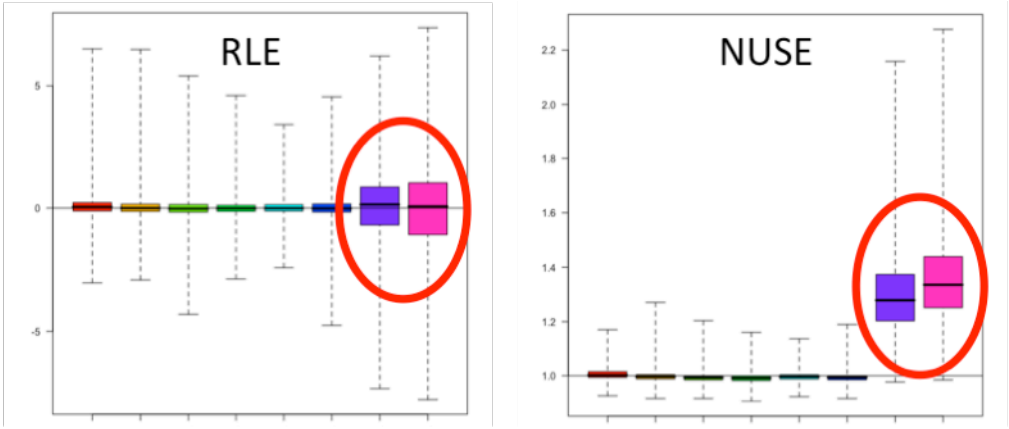

Figure 34: Relative logarithmic expression and normalized unscaled standard error plots. Note the two arrays that are consistently showing low quality behaviour across both plots

5.1.5 RNA degradation

In each probeset the probes are ordered directionally from the 5' to the 3' end. Average probe intensities are plotted by probe number. The resulting plot visualizes the global RNA degradation state of the samples used. Generally, RNA degradation is more active at the 5' terminus - signal intensities of the probes closer to this terminus are accordingly lower. If the slope of the probe intensity curve is exceeding a certain threshold value or the slopes of individual chips are deviating from the median by more than 10% Robin issues a warning (see Figure 35). As this kind of analysis relies on probesets consisting of more than one probe, it is only available for Affymetrix arrays.

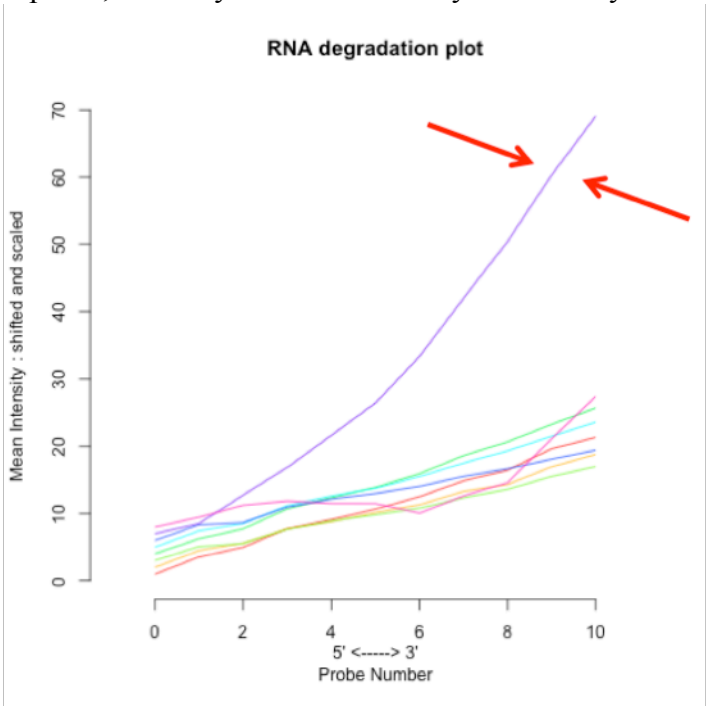

Figure 35: RNA degradation plot.

5.1.6 Scatter plots

If the scatter plot option is chosen, Robin plots pair wise comparisons of the normalized expression values of all possible combinations of two chips. NOTE: Using this feature on a large number of input chips will generate a lot of images and might increase calculation time and memory demand significantly. The scatter plots are useful for assessing whether two replicate chips are showing similar behavior. If they do, the points should lie on a perfect diagonal line. Replicate samples that are not showing this behavior strongly suggest a problem (e.g. accidentally swapped or mislabeled samples, technical problems on one individual chip, strong RNA degradation effects etc.)

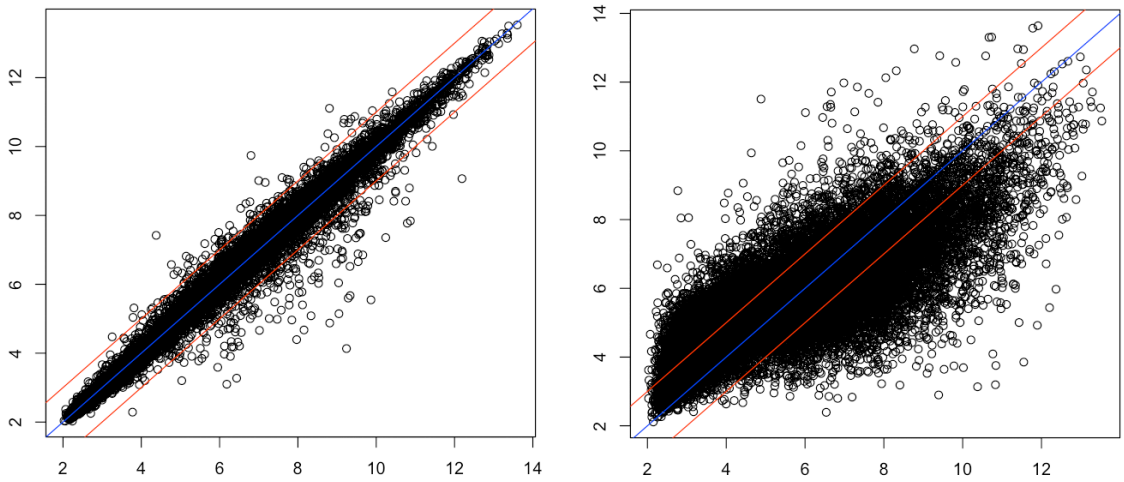

Figure 36: Scatter plots of normalized expression values. The left panel shows two biological replicates of acceptable reproducibility plotted against each other while the right panel shows two chips with very different expression profiles. Identical values are plotted on the blue (0) line; The red lines indicate a log2-fold difference of 1.

5.1.7 Principal component analysis and hierarchical clustering

The data generated in a microarray experiment can be understood as a matrix of ***p*** columns, where ***p*** is the number of chips used, and ***n*** rows, where ***n*** is the number of genes (probesets, probes) measured. Such a dataset could be visualized as a set of ***n*** points in a ***p-dimensional*** space. The principal component analysis reduces the dimensionality of the dataset by finding a small number of linear combinations of the data that explain most of the variance in the dataset. These are the principal components (PCs). The principal components are ordered by the amount of variance explained and subsequently the first two PCs are plotted against each other. The example on the right panel of Figure 37 shows a PCA on eight samples, six of which are grouping closely together on two groups of three replicates while the last two are completely unrelated.

The hierarchical clustering method performs a clustering of the Pearson correlation of raw normalized expression estimates for each chip a the distance measure for the

clustering. Chips that show similar expression profiles should cluster together when using this approach. The results are shown as a dendrogram where the branch length depicts 1-correlation score. The hierarchical clustering gives an overview of the internal structure of the data and identifies experimental conditions that generate similar global responses in gene expression. Replicate chips should always cluster closely. Accordingly, the samples six samples belonging to two groups of three replicates form distinct clusters while the last two are very distant from them and each other. The PCA and hierarchical clustering analyses are only available for Affymetrix and generic single channel experiments.

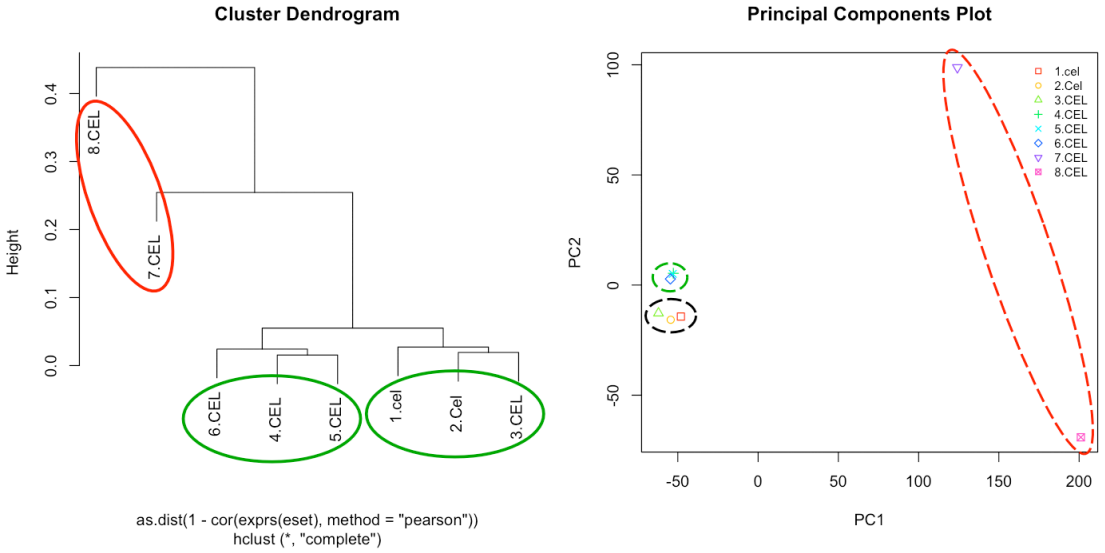

Figure 37: Principal component analysis and hierarchical clustering of normalized expression values. The red circles highlight chips with strongly deviating behavior.

5.2 Two color microarray quality checks

Quality check methods that are specific to two color arrays are described in the following section. Some quality checks that can be run for all chip types – these will not be described again below (e.g. MA plots).

5.2.1 Image plots of two-color background intensities and unnormalized M values

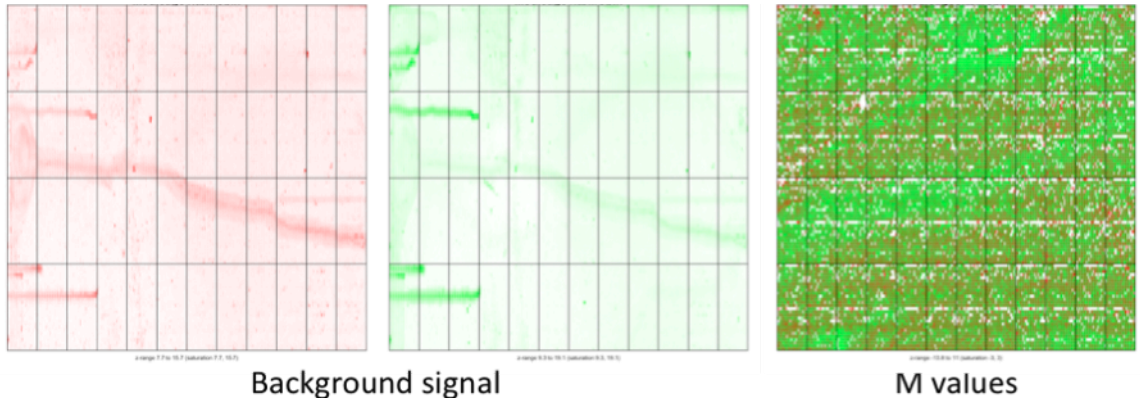

Figure 38: Two-color microarray background signal intensities and unnormalized M value plots.

The background signal intensities measured on two-color and generic single channel chips (not shown here) can be visualized as false-color images. This is very useful for the identification of washing artifacts like those visible on the two left plots in Figure 38. Both color channels display obvious traces of droplets, so called washing artifacts. In the worst case these artifacts carry over to the foreground signal and cannot be eliminated by background subtraction. If this happens they would also be visible on the M value plot shown on the right side of Figure 38 (in the example given, however, this is not the case). The M value plots is simply a false-color image of the merged red and green foreground signal intensities measured on the chip prior to normalization.

5.2.2 Overview of two color signal intensity distribution

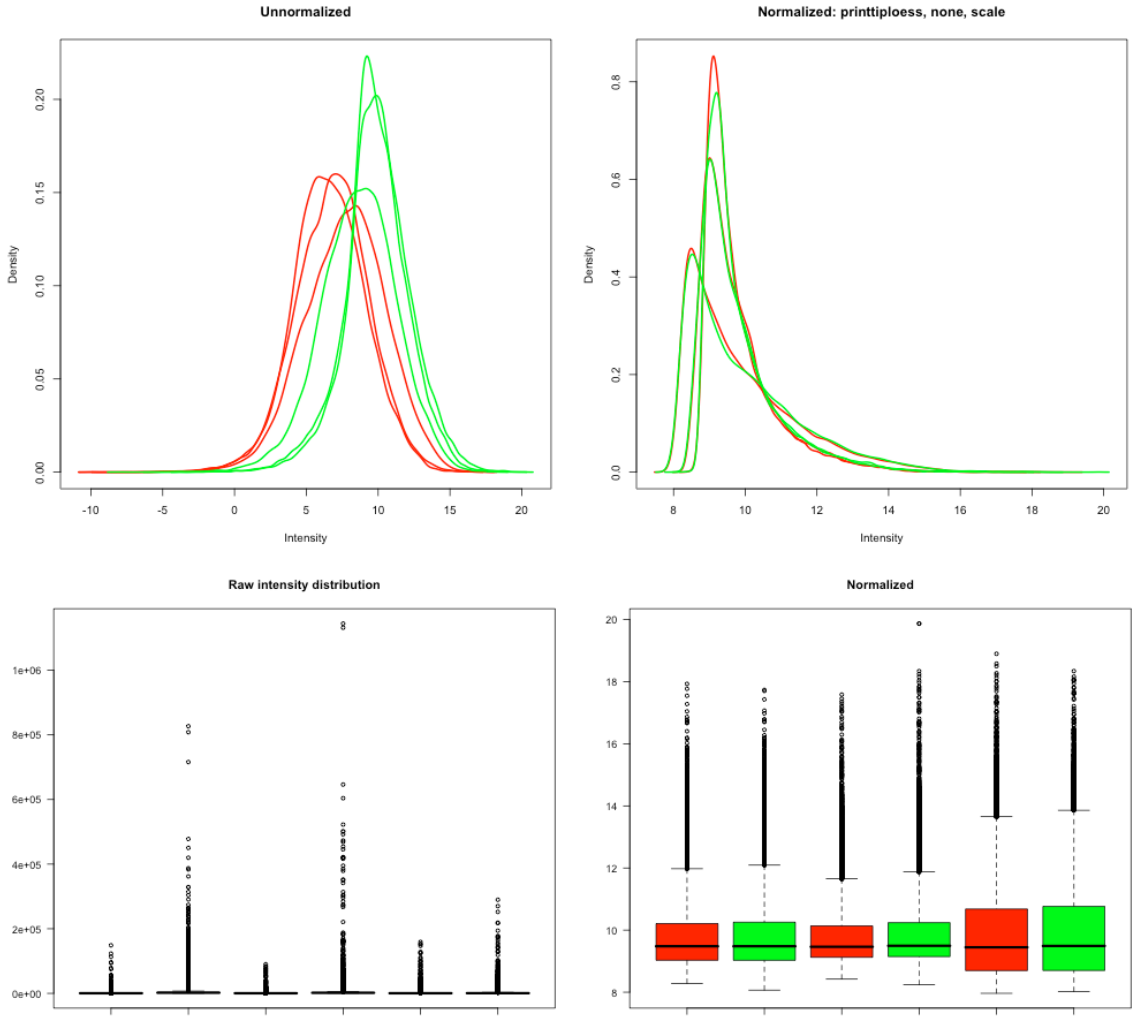

Figure 39: Two color microarray signal intensity distribution assessment. Upper left: Smoothed signal intensity distributions are shown for the red and green channel separately for each chip. Lower left: Box plots of the raw foreground signal intensities for each chip and color channel. The left hand plots show data prior to normalization while the plots on the right half show normalized data. The title of the right hand intensity distribution plot reflects the chosen normalization settings. For the shown example within-array printtiploess normalization without background correction and between-array scaling were performed.

Analogous to the box plots and smoothed histograms that are generated for Affymetrix arrays (see section “Analysis of signal intensity distribution” and Figure 31).

6 Data normalization

When analyzing microarray experiments, the raw data obtained by scanning probe intensities on the chips can be strongly influenced by different technical effects. These can be different levels of background signal due to inhomogeneous washing, systematically deviating probe signal intensities due to different scanner settings (or even same settings on different devices), probe-specific hybridization affinity effects etc.
To make sure that the microarrays you are going to analyze in a differential expression experiment can actually be compared it is very important to eliminate these effects. This process is called normalization. Since the first application of microarray technology many different normalization techniques have been proposed - the most widely used ones are available in Robin. If your favorite method is not among them feel free to contact us.

Generally, all normalization methods consist of two (three in the case of Affymetrix GeneChip microarrays) major steps: (I) background correction, (II) normalization of background-corrected probe level data and (III) summarization of probe-level data to yield one expression measure per probeset.

6.1 Single channel microarray normalization

6.1.1 Normalization methods for Affymetrix arrays

6.1.1.1 RMA [9]

The robust multi-array average (RMA) normalization method proposed by [9] has been widely used and accepted as a well-performing approach for inference of differential gene expression from Affymetrix GeneChip(R)-based experiments. The RMA procedure first does background correction based on the assumption that the background signal is normally distributed while the real probe signal is exponentially distributed (convolution model). The background-corrected data is then quantile normalized. Quantile normalization assumes that the distribution of gene abundances is nearly the same across all chips. A reference distribution is created using the pooled intensity probe distribution on all chips. To normalize each chip, the quantile of each intensity value is computed and then the original value is transformed to the corresponding quantile's value on the pooled reference chip (that is created by averaging the values of each probe across all chips in the experiment). In the last step, a linear model is fitted to the corrected, normalized and log2-transformed probe intensities: [ ] with [ ] being a probe affinity effect, μ_i representing the log2 expression level on array I and [ ] representing a noise error with mean = 0. The model parameters are estimated using the median polish procedure that is robust against outliers.

6.1.1.2 GCRMA [10]

The GCRMA method adds a more refined background adjustment to the standard RMA normalization. This background adjustment method models the different hybridization affinities for each PM-MM probe pair based on its nucleotide sequence which results in a more precise estimate of the background. While the standard RMA approach ignores the

MM probe-derived signal, GCRMA subtracts a shrunken MM value that was corrected for its binding affinity from the PM signal. More specifically, the model assumes:
[ ] and [ ] with O being the optical noise, N being the non-specific binding effect and S being proportional to the real concentration of the target transcript. Hence, the model takes into account the observation, that the MM signal may contain real transcript signal.

6.1.1.3 MAS 5.0 (Affymetrix Microarray Analysis Suite 5.0)

In contrast to the other normalization methods described here, MAS 5.0 works on a single chip basis. Briefly, each chip is divided into 16 (4x4) equally sized grid regions and a background and noise signal value is calculated based on the lowest 2% of measured probe intensities for each grid region. The probe intensities in each grid block are adjusted to the weighted average of the background signal where the weight is dependent on the (euclidean) distance of the probe to the centroid of the grid block. In the next step the perfect match (PM) and mismatch (MM) probe pairs are considered. The original purpose of the PM/MM probe pair design was to use the MM probe signal intensity as unspecific signal intensity and subtract it from the PM probe to generate a reliable probe signal. However it turned out that up to 30% of the MM probes display a signal intensity that is higher than the corresponding PM probe so that a simple subtraction would yield negative values. To work around this problem, the so called ideal mismatch (IM) was introduced. If the PM intensity is larger than MM, IM equals the MM value. In cases where PM=MM or PM<MM, IM is calculated using the PM value and a specific background (SB) value that is computed by taking a robust average of the log ratios of PM and MM. The summarized expression measure is computed using a Tukey biweight of PM and IM values in each probe set on the $\log_2$ scale. In MAS 5.0, the normalization is performed after summarization. A scaling normalization is used to adjust intensity values on each array. MAS 5.0 provides final expression values on the original scale. The Robin analysis workflow takes this into account and logarithmizes the values prior to statistical analysis to provide uniform output independent of the normalization method chosen. For a more detailed description of the method, please see the Affymetrix technical documentation (Affymetrix GeneChip® Expression Analysis, 2004).

6.1.1.4 PLIER (Affymetrix, Probe Logarithmic Intensity Error Estimation, 2005)

The PLIER method was developed by Affymetrix as an improved estimator of signal intensity. It is, unlike MAS 5.0, a multi-array method but includes the summarization algorithm that is also used in the MAS 5.0 method. Like RMA it uses a global model but bases this on a different set of assumptions. Unlike RMA it takes the MM probe signal into account when computing expression values. The observed PM and MM probe signal intensities for the ith probe on the jth array are assumed to be [ ] and [ ] with μ_{ij} being the binding level of probe i and array j, a_i being the probe specific binding affinity, c_j the RNA concentration in the sample hybridized to array j and B_{ij} the background binding intensity of probe I on array j.

PLIER also assumes that the error of the PM and MM probe signals are reciprocal (while MAS 5.0 assumes them to be equal): [ ] with [ ] being the error of the *i*th perfect match probe in the *j*th array and [ ] the error of the corresponding mismatch probe. This results in the following equation: [ ].

Based on the above assumptions, the PLIER algorithm computes the values of a and c by setting the residual $r = \log(\varepsilon)$ to zero using a minimization of a robust average of the r^2 values. PLIER performs slightly better than MAS 5.0 when comparing the analysis of spike-in experiments where RNA of known concentration was added to the sample, possibly due to a better error estimation procedure. For further details and an in-depth discussion of the PLIER algorithms and performance, please see Affymetrix's Guide to PLIER esitimation, 2005 and Therneau and Ballmann, 2008.

6.1.2 Normalization of generic single channel and two color arrays

Since most of the non-Affymetrix microarrays do not adopt a probeset design where multiple probes are matching one target transcript, the summarization step necessary for Affymetrix raw data is omitted. The two remaining steps, background correction and normalization, can be flexibly configured according to the experiments' requirements and users' preferences.

6.1.2.1 Background correction

Several methods to correct the measured probe intensity for background signal intensity are available. The background signal intensity values themselves have to be provided in a separate column in the raw data file and have to be specified upon import of the data (please see 4.2 and 4.3). Aside from "subtract" all background correction procedures are designed to produce positive corrected signal intensities. All methods listed below are implemented in the limma package – please refers to [11] for further details.

1) "subtract"– Simply subtracts the background intensities from the foreground intensity values.

2) "half" – All foreground signal intensities that are less than 0.5 of the original intensity after background subtraction will be set to 0.5 of the uncorrected value.

3) "minimum" – Values that are zero or negative after simple background subtraction are set to 0.5 times the smallest positive corrected value.

4) "edwards" – Uses a log-linear interpolation to adjust low intensity values (see Edwards, 2003)

5) "normexp" – Uses the same convolution model that is applied in the RMA method to model the background intensity with two modifications to make it better applicable for two color arrays. First, the model is fitted to the background

subtracted foreground values of each color channel separately and second, instead of using a kernel density parameter estimator for the model parameters, a maximum-likelihood estimator is used See (Ritchie et al., 2007) for details.

6) "rma" – Employs the background correction step of the RMA method for Affymetrix arrays.

6.1.2.2 Within-array normalization

This option is only available for two color microarrays and normalizes the log2 ratios of expression of the red and green channel signals so that the average log2-ratio is zero. This is again based on the assumption that most of the genes do not show differential expression in a given experiment. The options made available are implemented in the limma package and will be described in the following:

1) "median" – Simply subtracts the median from the calculated M values.

2) "loess" – Uses global loess regression (a robust smoothing algorithm based on local polynomial regression) to compute a trend in the data. Each M value is normalized by subtracting the corresponding the corresponding value of the loess curve from it according to [ ], where N is the normalized value, M the raw value and *loess(A)* the loess curve as a function of the average signal intensity A.

3) "printtiploess" – Performs the loess normalization separately for each print tip group. This approach accounts better for local spatial variation in background signal intensity and it therefore used as the default method for within-array normalization in Robin.

4) "robustspline" – This method does also normalize print tip group-wise but uses regression splines and empirical Bayes-based shrinkage instead of loess curves for normalization.

6.1.2.3 Between-array normalization

In addition to normalizing within each two color array, the user can choose to also perform between array normalization. When analyzing single channel arrays, this is the only normalization approach available. Applying between array normalization makes sure that the expression intensities (resp. log2 ratios on two color arrays) have equal distributions across a series of chips. Several options are available for two color arrays while the list is limited to scale and quantile normalization for single channel arrays (again, the methods listed are provided by the limma Bioconductor package).

1) "scale" – Log2 ratios of expression are scaled to have the same median absolute deviation (MAD) across arrays.

2) "quantile" – Adjusts to intensities to have the same empirical distribution across chips. This is the normalization method that is also used by the RMA procedure for Affymetrix chips.

3) "Aquantile" – Is a variation of the quantile method that only adjusts the A values to display the same distribution.

4) "Tquantile" – Does a quantile normalization separately for each of the target groups defined on the targets designer panel in the two color chip analysis workflow (see 4.2).

5) "Gquantile" and "Rquantile" – Quantile normalization is performed for the green ("G") or red ("R") color channel only. This approach makes sense if a common reference design has been employed in the experiment that is being analyzed and the reference sample was always hybridized in the same color channel.

Both normalization approaches can be combined when working with two color channels. In this case, within array normalization and background correction are performed prior to between array normalization steps. The preset default settings should give robust expression estimates in most cases. However, given the heterogeneity of two color and single color technical chip platforms, different settings may perform better for individual chip types. When trying to assess whether the chosen settings give decent results in a given experiment, it helps to inspect the shape of the MA plots after normalization. If the distribution of values displays the expected (often "trumpet"-like) shape and the plot is centered on the M = 0 line (see Figure 32), the settings seem to be sound. If in doubt, please seek advice from an experienced statistician.
For more in-depth information on the normalization methods in general, please refer to [12]

7 Analysis of differential gene expression

The statistical methods Robin employs to identify differentially expressed genes are based on two different approaches: Linear modeling (limma, [13]) and rank product-based analysis (RankProd, [14, 15]). When analyzing Affymetrix data, the user can choose between these two options with the restriction that rank product-based inference of differential expression is only available when two groups are to be compared. When working with two color microarrays, rank product-based analysis is not available yet. The two methods differ in that they take two completely different approaches to the detection of differentially expressed genes. While the linear model-based method relies on advanced statistical modelling and Bayesian inference, the rank product approach more resembles biological reasoning on the data. More specifically, limma assumes a linear model $y_j = X\alpha_j + \varepsilon_j$ where y_j contains the expression data for each gene, X is the design matrix describing the systematic part of the data and α_j is a vector of coefficients (representing the response level for gene j on chip g). The biologically interesting

contrasts of the coefficients are defined by [ ], where *C* is the contrast matrix (for a more detailed in-depth discussion please refer to [13]).

The rank product approach, on the other hand, assumes for an experiment in which *n* genes are investigated in *k* replicates, that the probability to find a gene at the top position of a ranked list of up- or down regulated genes is exactly [ ]. The combined probability of finding a gene at a certain position in the ranked list, when *k* replicates *i* and n_i genes are measured can be expressed as the rank product [ ], where [ ] is the position of gene *g* in the ranked list of decreasing (*up*) or increasing (*down*) fold changes in the *i*th replicate (see [14] for further details not to be reproduced here).
Since rank product-based analysis is limited to comparing two experimental conditions, the linear model-based analysis offers far more options and flexibility with respect to the available settings and design of the experiment (e.g. if two factors, like genotype and treatment, are being varied in an experiment and the user is interested in the interaction effect).

8 Output

At the end of each analysis run, Robin asks for a directory to save all files that are relevant to the experiment. These include processed raw input data files (only in the case of two color and generic single channel analysis), R source code for quality assessment and main analysis, various informative plots illustrating the quality check results and main analysis results and tabular text data files containing the full results in all detail. The following table lists all files that are generated. The "Type" column refers to the microarray type for which this kind of output file can be generated (G = all platforms, A=Affymetrix, T=two color microarrays, S=generic single channel arrays). Parts of the file names written in italics refer to variable text: *EXP_NAME*: The name of the experiment as entered by the user when choosing the name of the output folder. *TMP*: An automatically generated unique identifier used for temporary files (the quality check output files are first stored in the system's temporary folder and are later copied to the quality checks folder of the output directory). *GRP*: Reference to the group names as assigned by the user when sorting individual raw files (e.g. .cel files) into groups of biological replicates.

Filename	Folder	Description	Type
*EXP_NAME*_results.txt	.	This file contains the normalized log2 fold change in expression values for all comparisons defined in the design step of the experiment. In addition, a second column containing a flag value denoting the statistical significance of each log fold change is generated for each gene. A value of 0 means not significant, while -1 and 1 mean significantly up- or down-regulated.	G

*EXP_NAME*_summary.txt	.	A text file summarizing and documenting the analysis inputs, program settings and warnings generated during the workflow.	G
*EXP_NAME*_design.png	.	PNG representation of the experiment design as configured on the graphical designer panel in the last step of the analysis workflow.	G
redundant.probes.info.txt	detailed_results	If redundant probe names are found in the input data of the generic single channel rank product analysis, this file is generated. It contains the redundant identifiers, number of spots found and the median values for each of the identifiers on each chip.	S
full_table_*GRPa-GRPb*.txt	detailed_results	Tables giving the complete statistical results for each of the comparisons made. The columns contain from left to right: (Feature.ID) A unique identifier for the oligonucleotide probes or probe sets on the chips; (logFC) the log2-fold change in expression; (AveExpr) average normalized expression value; (t) t-statistic; (P.Value, adj.P.Val) raw and Benjamini-Hochberg-corrected p-values for differential expression; (B) the log-odds for differential expression.	G
top100table_*GRPa-GRPb*.txt	detailed_results	Contains the same data columns as the full tables but excludes probes / probesets that do not fulfill the chosen p-value and or minimal log2-fold change cut offs.	
EXP_NAME.PAcalls.table.txt	detailed_results	Only generated when analyzing Affymetrix chips. Table containing the present / absent calls for each probeset on each chip in the experiment plus the attached p-values that are calculated using the MAS5calls function.	A
raw_*METHOD*_normalized_expression_values.txt	.	Expression estimates for each probe/probeset on each chip after normalization.	A
*TMP*_hclust.png	qualitychecks	Hierarchical clustering of the normalized expression values. The clustering is based on 1-pearson correlation of expression as the distance measure. Full linkage hierarchical clustering is performed.	A, S
*TMP*_pcaplot.png	qualitychecks	Scatter plot of the first two components obtained in a principal component analysis of the normalized expression values.	A, S

*TMP*_boxplot.png	qualitychecks	Boxplots of the unnormalized signal intensities on each chip	G
*TMP*_hist.png	qualitychecks	Smoothed density plots showing the signal intensity distribution on each chip prior to normalization.	A, S
*TMP*_density.png	qualitychecks	These plots display the signal intensity distribution for two color arrays analogous to the ”hist” plots for Affymetrix and other single channel arrays. Smoothed distributions are plotted separately for both color channels	T
*TMP*_maplot*1..n*.png	qualitychecks	MA plots of chip 1 to n. When analyzing single channel chips, these plots show the log2-fold change in expression of each individual chip when compared to a synthetic chip constructed from the median expression values of all chips in the experiment. In the case of two color arrays the M values correspond to log log2 ratio between the green and red channel signal intensities prior to and after normalization. Each plot also shows the following quality-associated parameters: “I” – Absolute value of the numerical integral of the lowess fit curve over the M=0 line. Values greater than 1 are considered to indicate lower quality. “%>LFC1” - Percentage of probes/probesets displaying a log2-fold change greater than 1. Based on the assumption that most of the genes will not show differential expression, a warning will be issued of more than 5% of the probes show an absolute log2 fold change higher than 1 (meaning 2-fold up- or downregulation). “median” – Gives the median value of M. In an ideal experiment this should be zero.	G
*TMP*_plm*1..n*.png	qualitychecks	Shows pseudo images of the model weights for each probe after fitting linear probe level models. Low weights are indicated by stronger red or green color	A
*TMP*_rle.png	qualitychecks	Boxplots of the relative logarithmic expression (RLE) values on each chip. The boxes should be centered around zero.	A

*TMP*_nuse.png	qualitychecks	Boxplots of the normalized unscaled standard errors (NUSE) of the probe level models on each chip. The plots should be centered around zero and display comparable spread.	A
*TMP*_scat*1..n1..m*.png	qualitychecks	Scatter plots of all possible combinations of two chips. The normalized expression values are plotted against each other.	A, S
*TMP*_rna.png	qualitychecks	RNA degradation plot (only available for Affymetrix arrays). Shows mean intensities of probes in all probesets ordered from the 5' to the 3' end of the target sequence. This plot allows a good overview of the global RNA quality on the chips.	A
*TMP*_bground.png	qualitychecks	Pseudo images of the background signal intensities measured on two color or non-Affymetrix single channel arrays.	T, S
*TMP*_mvalues.png	qualitychecks	Pseudo image plots of the unnormalized M (= log2 ratios of green and red signal) values of two color chips.	T
*XYZ*_robin	input	Cleaned-up copies of the input raw data files	T, S
EXP_NAME.main.analysis.R	source	The R script file containing code for the main analysis. The file can be used as a starting point for customizations of the analysis. Please note that the file contains some hard coded paths.	G
qualityChecks.R	source	Quality checks R source code file.	G
MAplot_*GRPa-GRPb*.png	plots	The plots folder contains some informative plots on the results of the main analysis. MA plots are generated for each contrast that was defined on the experiment designer panel. Genes that are significantly differentially expressed according to the statistical analysis are highlighted by red circles.	G
vennDiagram_down/total/up.png	plots	Venn diagrams showing the number of significantly up- down- and total regulated genes for up to four contrasts.	G
PCAplot.png	plots	Principal component analysis plot analogous to the plots generated in the quality checks section. This plot does in addition highlight the groups of replicate experiments as defined on the groups panel.	A, S

REFERENCES

1. Cock, P.J., et al., *The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants.* Nucleic Acids Res, 2010. **38**(6): p. 1767-71.
2. Langmead, B., et al., *Ultrafast and memory-efficient alignment of short DNA sequences to the human genome.* Genome Biol, 2009. **10**(3): p. R25.
3. Robinson, M.D., D.J. McCarthy, and G.K. Smyth, *edgeR: a Bioconductor package for differential expression analysis of digital gene expression data.* Bioinformatics, 2010. **26**(1): p. 139-40.
4. Robinson, M.D. and G.K. Smyth, *Moderated statistical tests for assessing differences in tag abundance.* Bioinformatics, 2007. **23**(21): p. 2881-7.
5. Robinson, M.D. and G.K. Smyth, *Small-sample estimation of negative binomial dispersion, with applications to SAGE data.* Biostatistics, 2008. **9**(2): p. 321-32.
6. McCarthy, D.J., Y. Chen, and G.K. Smyth, *Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation.* Nucleic Acids Research, 2012.
7. Anders, S. and W. Huber, *Differential expression analysis for sequence count data.* Genome Biol, 2010. **11**(10): p. R106.
8. Holm, S., *A stagewise rejective multiple test procedure based on a modified Bonferroni test.* Scandinavian Journal of Statistics, 1979(6): p. 65-70.
9. Irizarry, R.A., et al., *Exploration, normalization, and summaries of high density oligonucleotide array probe level data.* Biostatistics (Oxford, England), 2003. **4**(2): p. 249-64.
10. Wu, Z., et al., *A Model-Based Background Adjustment for Oligonucleotide Expression Arrays.* Journal of the American Statistical Association, 2004. **99**(468): p. 909-917.
11. Ritchie, M.E., et al., *A comparison of background correction methods for two-colour microarrays.* Bioinformatics, 2007. **23**(20): p. 2700-7.
12. Smyth, G.K. and T. Speed, *Normalization of cDNA microarray data.* Methods, 2003. **31**(4): p. 265-73.
13. Smyth, G.K., *Linear models and empirical bayes methods for assessing differential expression in microarray experiments.* Statistical applications in genetics and molecular biology, 2004. **3**: p. Article3.
14. Breitling, R., et al., *Rank products: a simple, yet powerful, new method to detect differentially regulated genes in replicated microarray experiments.* FEBS Lett, 2004. **573**(1-3): p. 83-92.
15. Hong, F., et al., *RankProd: a bioconductor package for detecting differentially expressed genes in meta-analysis.* Bioinformatics, 2006. **22**(22): p. 2825-7.