Exploring RNA-seq data from Gierlinski, Schurch and Barton

Mustafa Abu El-Qumsan & Jacques van Helden

2016-02-01

## Parameters for the execution of this tutorial

## Introduction

The goal of this tutorial is to get familiar with RNA-seq data:

* load a table of read counts per gene;
* do some exploratory statistics about these counts per gene (distributions, histograms, ...)

### Data source

1. Gierliński,M., Cole,C., Schofield,P., Schurch,N.J., Sherstnev,A., Singh,V., Wrobel,N., Gharbi,K., Simpson,G., Owen-Hughes,T., et al. (2015) Statistical models for RNA-seq data derived from a two-condition 48-replicate experiment. Bioinformatics, 10.1093/bioinformatics/btv425.
2. Schurch,N.J., Schofield,P., Gierliński,M., Cole,C., Sherstnev,A., Singh,V., Wrobel,N., Gharbi,K., Simpson,G.G., Owen-Hughes,T., et al. (2015) Evaluation of tools for differential gene expression analysis by RNA-seq on a 48 biological replicate experiment. arXiv.

#### Full original datasets (>300 fastq files !)

Data shared at ENA: <https://figshare.com/articles/Metadata_for_a_highly_replicated_two_condition_yeast_RNAseq_experiment_/1416210>

Fastq files: <http://www.ebi.ac.uk/ena/data/view/ERP004763>

#### Read count tables

* Wild-type yeast strains (WT): <https://dx.doi.org/10.6084/m9.figshare.1425503>
* Snf2 mutant: <https://dx.doi.org/10.6084/m9.figshare.1425502>

## Data loading

We will now load the two read count tables (WT and Snf2 mutants respectively).

## Data exploration

We will first check the dimension of the two tables.

[1] 7125 48

[1] 7125 48

The tables *counts.wt* and *counts.snf2* contain the read counts per gene for the Wild-Type and the Snf2 strains, respectively. Each table contains one row per gene and one column per sample.

|  |  |  |
| --- | --- | --- |
| Genotype | Rows | Columns |
| WT | 7125 | 48 |
| Snf2 | 7125 | 48 |
| All | 7125 | 96 |
|  |  | |

## Summary of counts per sample (column)

We will use the function *summary()* to compute descriptie statistics for each column of the count table.

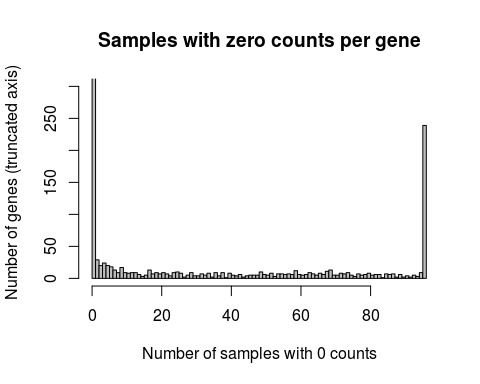
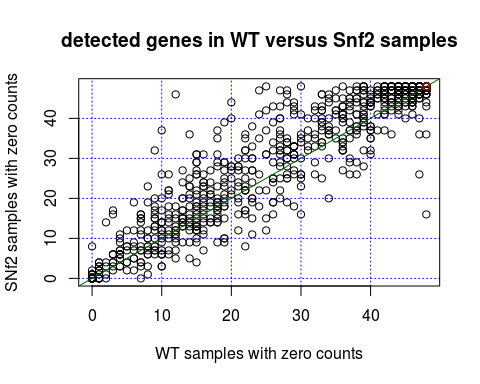
WT.1 WT.2 WT.3 WT.4   
 Min. : 0.0 Min. : 0 Min. : 0.0 Min. : 0   
 1st Qu.: 49.0 1st Qu.: 88 1st Qu.: 75.0 1st Qu.: 105   
 Median : 224.0 Median : 371 Median : 297.0 Median : 430   
 Mean : 837.1 Mean : 1107 Mean : 903.6 Mean : 1464   
 3rd Qu.: 561.0 3rd Qu.: 853 3rd Qu.: 670.0 3rd Qu.: 1019   
 Max. :188825.0 Max. :196804 Max. :172119.0 Max. :328674   
 WT.5 WT.6 WT.7 WT.8   
 Min. : 0 Min. : 0 Min. : 0 Min. : 0   
 1st Qu.: 84 1st Qu.: 153 1st Qu.: 78 1st Qu.: 80   
 Median : 353 Median : 667 Median : 350 Median : 360   
 Mean : 1124 Mean : 2052 Mean : 1244 Mean : 1381   
 3rd Qu.: 819 3rd Qu.: 1599 3rd Qu.: 858 3rd Qu.: 900   
 Max. :225435 Max. :357247 Max. :261535 Max. :343455   
 WT.9 WT.10 WT.11 WT.12   
 Min. : 0 Min. : 0 Min. : 0.0 Min. : 0   
 1st Qu.: 77 1st Qu.: 81 1st Qu.: 55.0 1st Qu.: 76   
 Median : 344 Median : 360 Median : 244.0 Median : 332   
 Mean : 1243 Mean : 1112 Mean : 953.8 Mean : 1153   
 3rd Qu.: 841 3rd Qu.: 867 3rd Qu.: 611.0 3rd Qu.: 802   
 Max. :268394 Max. :188415 Max. :255994.0 Max. :257336   
 WT.13 WT.14 WT.15 WT.16   
 Min. : 0 Min. : 0 Min. : 0 Min. : 0.0   
 1st Qu.: 92 1st Qu.: 103 1st Qu.: 71 1st Qu.: 50.0   
 Median : 403 Median : 432 Median : 300 Median : 216.0   
 Mean : 1275 Mean : 1348 Mean : 1069 Mean : 781.7   
 3rd Qu.: 971 3rd Qu.: 1021 3rd Qu.: 736 3rd Qu.: 526.0   
 Max. :228628 Max. :253155 Max. :245011 Max. :173478.0   
 WT.17 WT.18 WT.19 WT.20   
 Min. : 0 Min. : 0 Min. : 0 Min. : 0.0   
 1st Qu.: 82 1st Qu.: 80 1st Qu.: 84 1st Qu.: 57.0   
 Median : 357 Median : 345 Median : 364 Median : 245.0   
 Mean : 1008 Mean : 1080 Mean : 1309 Mean : 883.1   
 3rd Qu.: 831 3rd Qu.: 821 3rd Qu.: 879 3rd Qu.: 595.0   
 Max. :170161 Max. :194935 Max. :310977 Max. :208697.0   
 WT.21 WT.22 WT.23 WT.24   
 Min. : 0 Min. : 0 Min. : 0 Min. : 0   
 1st Qu.: 91 1st Qu.: 128 1st Qu.: 79 1st Qu.: 86   
 Median : 420 Median : 584 Median : 345 Median : 368   
 Mean : 995 Mean : 1435 Mean : 1072 Mean : 1116   
 3rd Qu.: 1015 3rd Qu.: 1330 3rd Qu.: 822 3rd Qu.: 841   
 Max. :127734 Max. :162107 Max. :185052 Max. :197120   
 WT.25 WT.26 WT.27 WT.28   
 Min. : 0 Min. : 0 Min. : 0 Min. : 0   
 1st Qu.: 104 1st Qu.: 87 1st Qu.: 92 1st Qu.: 99   
 Median : 478 Median : 375 Median : 415 Median : 475   
 Mean : 1155 Mean : 1296 Mean : 1347 Mean : 1132   
 3rd Qu.: 1123 3rd Qu.: 914 3rd Qu.: 1006 3rd Qu.: 1097   
 Max. :115918 Max. :291273 Max. :242148 Max. :114079   
 WT.29 WT.30 WT.31 WT.32   
 Min. : 0 Min. : 0 Min. : 0 Min. : 0.0   
 1st Qu.: 83 1st Qu.: 118 1st Qu.: 83 1st Qu.: 60.0   
 Median : 353 Median : 495 Median : 369 Median : 248.0   
 Mean : 1145 Mean : 1622 Mean : 1208 Mean : 917.9   
 3rd Qu.: 816 3rd Qu.: 1196 3rd Qu.: 912 3rd Qu.: 601.0   
 Max. :242182 Max. :332378 Max. :210862 Max. :241648.0   
 WT.33 WT.34 WT.35 WT.36   
 Min. : 0 Min. : 0 Min. : 0.0 Min. : 0   
 1st Qu.: 91 1st Qu.: 74 1st Qu.: 69.0 1st Qu.: 74   
 Median : 391 Median : 307 Median : 303.0 Median : 300   
 Mean : 1334 Mean : 1200 Mean : 934.2 Mean : 1102   
 3rd Qu.: 923 3rd Qu.: 721 3rd Qu.: 732.0 3rd Qu.: 699   
 Max. :274478 Max. :314904 Max. :162505.0 Max. :247516   
 WT.37 WT.38 WT.39 WT.40   
 Min. : 0 Min. : 0 Min. : 0 Min. : 0   
 1st Qu.: 75 1st Qu.: 54 1st Qu.: 61 1st Qu.: 118   
 Median : 344 Median : 245 Median : 281 Median : 491   
 Mean : 1545 Mean : 1010 Mean : 1040 Mean : 1674   
 3rd Qu.: 901 3rd Qu.: 620 3rd Qu.: 689 3rd Qu.: 1210   
 Max. :467550 Max. :270499 Max. :246621 Max. :358116   
 WT.41 WT.42 WT.43 WT.44   
 Min. : 0 Min. : 0 Min. : 0 Min. : 0   
 1st Qu.: 92 1st Qu.: 71 1st Qu.: 74 1st Qu.: 54   
 Median : 420 Median : 312 Median : 291 Median : 259   
 Mean : 1400 Mean : 1027 Mean : 930 Mean : 1138   
 3rd Qu.: 1020 3rd Qu.: 750 3rd Qu.: 676 3rd Qu.: 674   
 Max. :278531 Max. :209062 Max. :191739 Max. :318398   
 WT.45 WT.46 WT.47 WT.48   
 Min. : 0 Min. : 0.0 Min. : 0 Min. : 0   
 1st Qu.: 100 1st Qu.: 55.0 1st Qu.: 115 1st Qu.: 79   
 Median : 457 Median : 233.0 Median : 500 Median : 334   
 Mean : 1625 Mean : 783.4 Mean : 1696 Mean : 1122   
 3rd Qu.: 1124 3rd Qu.: 559.0 3rd Qu.: 1228 3rd Qu.: 790   
 Max. :338511 Max. :169415.0 Max. :348635 Max. :244192   
 SNF2.1 SNF2.2 SNF2.3 SNF2.4   
 Min. : 0 Min. : 0 Min. : 0 Min. : 0   
 1st Qu.: 137 1st Qu.: 98 1st Qu.: 98 1st Qu.: 137   
 Median : 540 Median : 422 Median : 408 Median : 563   
 Mean : 1436 Mean : 1220 Mean : 1231 Mean : 1496   
 3rd Qu.: 1203 3rd Qu.: 960 3rd Qu.: 954 3rd Qu.: 1268   
 Max. :205731 Max. :199983 Max. :210124 Max. :204494   
 SNF2.5 SNF2.6 SNF2.7 SNF2.8   
 Min. : 0.0 Min. : 0 Min. : 0 Min. : 0   
 1st Qu.: 84.0 1st Qu.: 109 1st Qu.: 162 1st Qu.: 92   
 Median : 336.0 Median : 555 Median : 668 Median : 377   
 Mean : 966.1 Mean : 1216 Mean : 1752 Mean : 1144   
 3rd Qu.: 763.0 3rd Qu.: 1340 3rd Qu.: 1523 3rd Qu.: 862   
 Max. :158243.0 Max. :64926 Max. :242692 Max. :209080   
 SNF2.9 SNF2.10 SNF2.11 SNF2.12   
 Min. : 0 Min. : 0 Min. : 0 Min. : 0   
 1st Qu.: 116 1st Qu.: 89 1st Qu.: 109 1st Qu.: 97   
 Median : 495 Median : 366 Median : 459 Median : 377   
 Mean : 1407 Mean : 1156 Mean : 1391 Mean : 1039   
 3rd Qu.: 1106 3rd Qu.: 850 3rd Qu.: 1046 3rd Qu.: 841   
 Max. :210681 Max. :178901 Max. :246736 Max. :152647   
 SNF2.13 SNF2.14 SNF2.15 SNF2.16   
 Min. : 0 Min. : 0 Min. : 0 Min. : 0   
 1st Qu.: 115 1st Qu.: 126 1st Qu.: 80 1st Qu.: 121   
 Median : 500 Median : 498 Median : 345 Median : 493   
 Mean : 1008 Mean : 1335 Mean : 1182 Mean : 1274   
 3rd Qu.: 1120 3rd Qu.: 1128 3rd Qu.: 815 3rd Qu.: 1080   
 Max. :95955 Max. :173286 Max. :252213 Max. :181744   
 SNF2.17 SNF2.18 SNF2.19 SNF2.20   
 Min. : 0 Min. : 0 Min. : 0.0 Min. : 0   
 1st Qu.: 126 1st Qu.: 108 1st Qu.: 87.0 1st Qu.: 102   
 Median : 493 Median : 439 Median : 353.0 Median : 443   
 Mean : 1429 Mean : 1173 Mean : 995.9 Mean : 1278   
 3rd Qu.: 1104 3rd Qu.: 985 3rd Qu.: 799.0 3rd Qu.: 1004   
 Max. :229969 Max. :159994 Max. :149249.0 Max. :214569   
 SNF2.21 SNF2.22 SNF2.23 SNF2.24   
 Min. : 0 Min. : 0.0 Min. : 0 Min. : 0   
 1st Qu.: 112 1st Qu.: 85.0 1st Qu.: 96 1st Qu.: 159   
 Median : 463 Median : 338.0 Median : 391 Median : 634   
 Mean : 1338 Mean : 945.4 Mean : 1039 Mean : 1956   
 3rd Qu.: 1057 3rd Qu.: 751.0 3rd Qu.: 875 3rd Qu.: 1462   
 Max. :201864 Max. :140481.0 Max. :146200 Max. :308470   
 SNF2.25 SNF2.26 SNF2.27 SNF2.28   
 Min. : 0 Min. : 0 Min. : 0.0 Min. : 0   
 1st Qu.: 79 1st Qu.: 125 1st Qu.: 75.0 1st Qu.: 93   
 Median : 352 Median : 507 Median : 317.0 Median : 374   
 Mean : 1181 Mean : 1377 Mean : 931.8 Mean : 1078   
 3rd Qu.: 855 3rd Qu.: 1142 3rd Qu.: 727.0 3rd Qu.: 850   
 Max. :231916 Max. :210356 Max. :142615.0 Max. :180213   
 SNF2.29 SNF2.30 SNF2.31 SNF2.32   
 Min. : 0 Min. : 0 Min. : 0 Min. : 0   
 1st Qu.: 108 1st Qu.: 90 1st Qu.: 74 1st Qu.: 124   
 Median : 422 Median : 370 Median : 335 Median : 491   
 Mean : 1089 Mean : 1014 Mean : 1067 Mean : 1322   
 3rd Qu.: 917 3rd Qu.: 825 3rd Qu.: 788 3rd Qu.: 1093   
 Max. :161886 Max. :155304 Max. :199024 Max. :195303   
 SNF2.33 SNF2.34 SNF2.35 SNF2.36   
 Min. : 0 Min. : 0 Min. : 0.0 Min. : 0   
 1st Qu.: 146 1st Qu.: 104 1st Qu.: 93.0 1st Qu.: 86   
 Median : 616 Median : 441 Median : 376.0 Median : 374   
 Mean : 1653 Mean : 1254 Mean : 894.8 Mean : 1258   
 3rd Qu.: 1368 3rd Qu.: 993 3rd Qu.: 842.0 3rd Qu.: 891   
 Max. :233323 Max. :194648 Max. :96395.0 Max. :249423   
 SNF2.37 SNF2.38 SNF2.39 SNF2.40   
 Min. : 0 Min. : 0 Min. : 0 Min. : 0   
 1st Qu.: 102 1st Qu.: 88 1st Qu.: 178 1st Qu.: 99   
 Median : 428 Median : 382 Median : 759 Median : 420   
 Mean : 1120 Mean : 1180 Mean : 2131 Mean : 1161   
 3rd Qu.: 954 3rd Qu.: 878 3rd Qu.: 1737 3rd Qu.: 942   
 Max. :152411 Max. :214801 Max. :323457 Max. :180745   
 SNF2.41 SNF2.42 SNF2.43 SNF2.44   
 Min. : 0 Min. : 0.0 Min. : 0 Min. : 0   
 1st Qu.: 116 1st Qu.: 75.0 1st Qu.: 119 1st Qu.: 128   
 Median : 481 Median : 325.0 Median : 476 Median : 494   
 Mean : 1398 Mean : 967.7 Mean : 1343 Mean : 1246   
 3rd Qu.: 1098 3rd Qu.: 752.0 3rd Qu.: 1057 3rd Qu.: 1084   
 Max. :239701 Max. :163673.0 Max. :219417 Max. :167478   
 SNF2.45 SNF2.46 SNF2.47 SNF2.48   
 Min. : 0 Min. : 0 Min. : 0 Min. : 0   
 1st Qu.: 95 1st Qu.: 108 1st Qu.: 124 1st Qu.: 96   
 Median : 386 Median : 450 Median : 488 Median : 383   
 Mean : 1103 Mean : 1236 Mean : 1229 Mean : 1026   
 3rd Qu.: 858 3rd Qu.: 1005 3rd Qu.: 1058 3rd Qu.: 839   
 Max. :177709 Max. :170179 Max. :160083 Max. :142435

We can immediately draw some observations from these summaries.

* **min**: in each column, the mean is 0. This corresponds to genes that were undetected: apparently these genes are either not transcribed, or they are transcribed at a weak level so that they were not sequenced in the library.
* **Mean**: in this experiment the mean counts per gene are of the order of 1000 counts per gene for each biological sample.
* \*\*max\* The max count is immensely larger than the mean counts, and this is true for each column. This observation is very general with RNA-seq data: in all the cases I have studied so far, we detect a very few genes having a huge number of counts.
* **median** is the count value that splits the sorted values in two. Half of the values of a column are smaller than or equal to the median.

## Number of undetected genes per sample

We would like to count the number of undetected genes per sample, i.e. genes having zero counts.

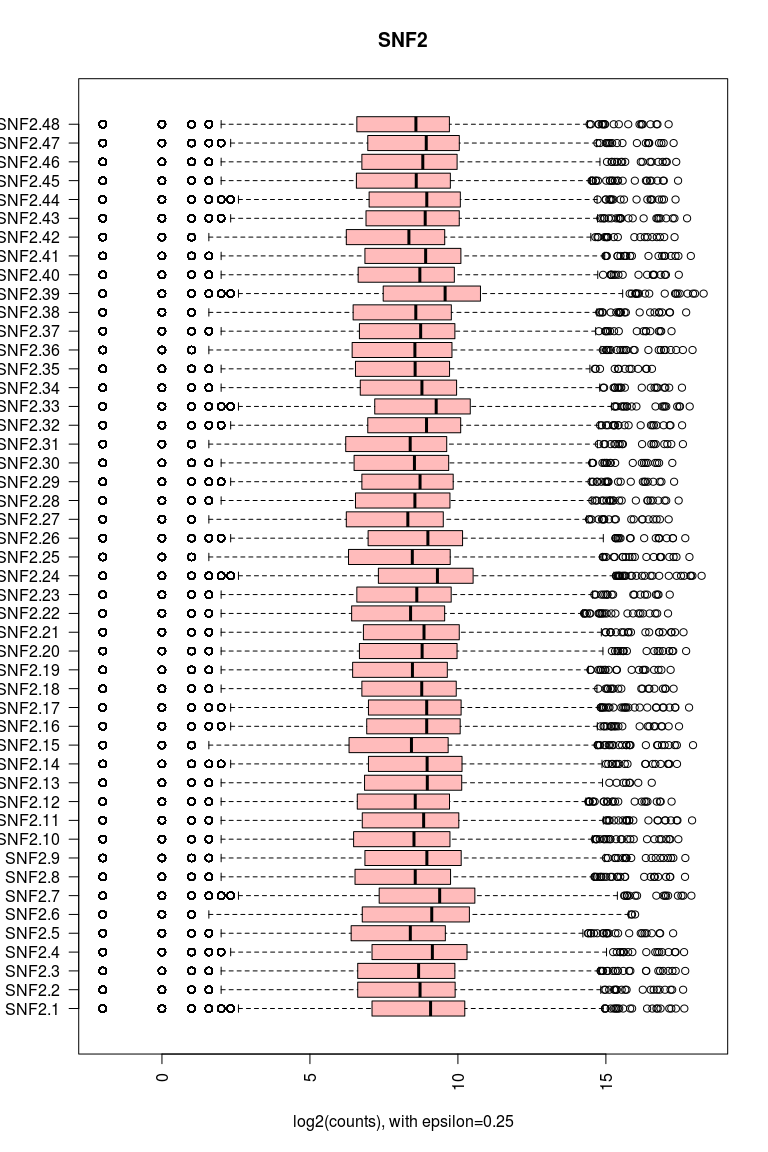
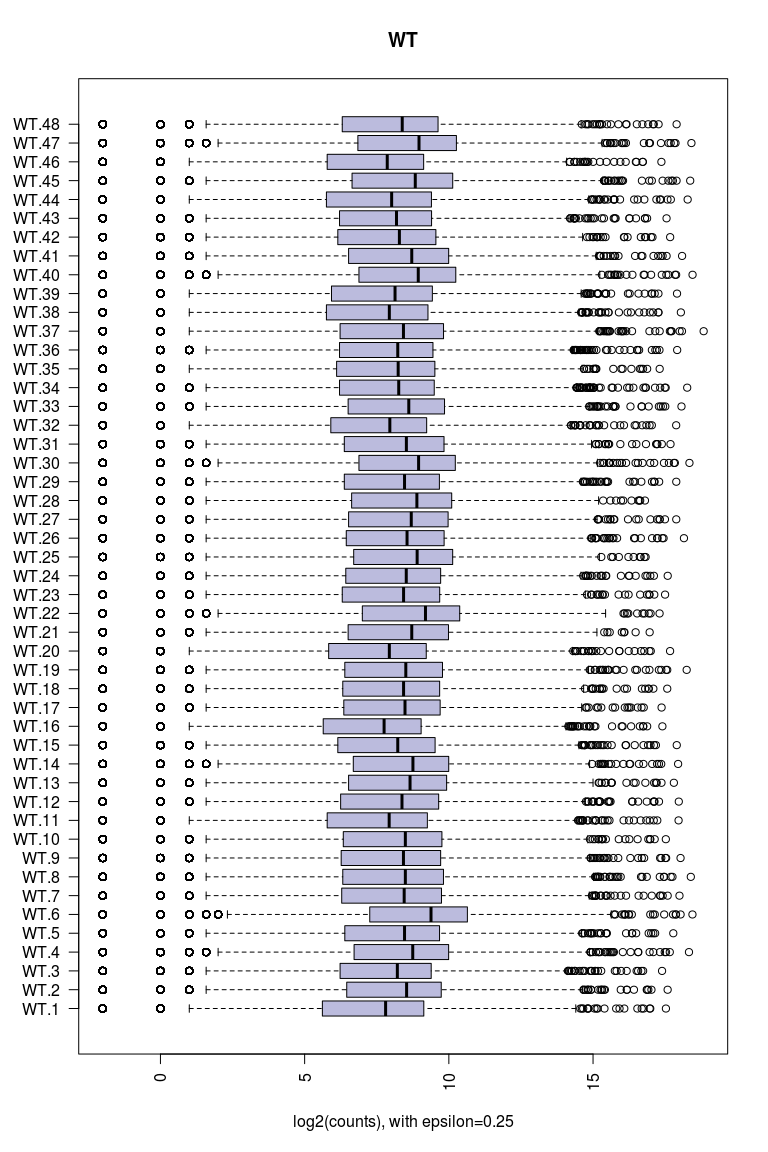
## Compute the log-transformed counts

RNA-seq count data are characterized by a very wide range of values, with small count numbers of the majority of the genes, and a small number of genes having hundreds fo thousands of counts. For many purposes (visualisation, summarization, ...), it is more relevant to work with log2-transformed counts.

However, we first must treate a problem: since many genes have 0 counts, the logarithmic transformation would give -Infinite, which creates problems for both computation and display. To circumvent this, we arbitrarily convert the zero counts in a small epsilon value (smaller than 1, so we can distinguish the genes with 1 counts from genes with 0 counts). we will choose an epsilon of 0.01.

## Box plots

We would like to draw box plots of the counts per genes.



## Normalization of RNA-seq counts

The total number of counts (``libsum'') can show wide variations from sample to sample, resulting from various sources (biological sample, sequencing, mapping, ...). Before comparing the counts per gene between different samples, it is important to perform a between-sample normalization.

Several methods have been proposed for this.

### Counts per million (CPM)

Counts are converted to counts per million by a simple scaling rule.

We dispose of a raw count table containing 7125 rows (one row per gene) and 96 columns (one column per sample).

Let us define as the count of reads for gene ( row) in sample ( column).

We can compute the sum of counts for each sample (column), which is also called the ***libsum***.

where is the total number of genes (rows in the table).

The basic way to compute counts per million reads is to divide each count by the libsum of the corresponding sample (column), and multiplying by 1 million for scaling purposes.

### Median-based standardization

A well-known problem with this simple approach is that the libsum can be strongly influenced by outliers, i.e. a handful of genes which are represented by hundreds of millions of reads.

To circumvent this, one possibility is to use a more robust scaling factor for each sample, for example the median count, or the percentile.

### When should we use / not use normalized counts?

Normalized log2-transformed counts are very convenient for display purposes (e.g. to compare the read density between samples or between conditions).

**However**, they cannot be used for all purposes. Indeed, the normalization procedure converts integer counts into fractional numbers, which do not comply with the probabilistic models underlying manysome of the statistical tests that are performed on these datasets. In particular, the most popular packages for differential analysis with RNA-seq expect to take as input a table with raw counts.

The log2 transformation completely modifies the nature of the data, and completely change the scale of the counts. Consequently, log2-transformed data should not be used either as input for the classical differential analysis packages.

Log2-transformed standardized counts may also be interesting for some clustering or classification purposes, because they will strongly reduce the dynamic range of the data, and the *log2 transformation has a normalizing effect* (this is a general effect) that may contribute to increase the suitedness of the data for some analysis methods (e.g. PCA, clustering, discriminant analysis, ...). We will investigate this very soon.

## Exercises

### Exercise 1. Summary statistics

Compute a table with summary statistics per samples (columns of the count table). In particular, include in this table the following statistics. All statistics should be computed on both the raw counts and log2-transformed counts.

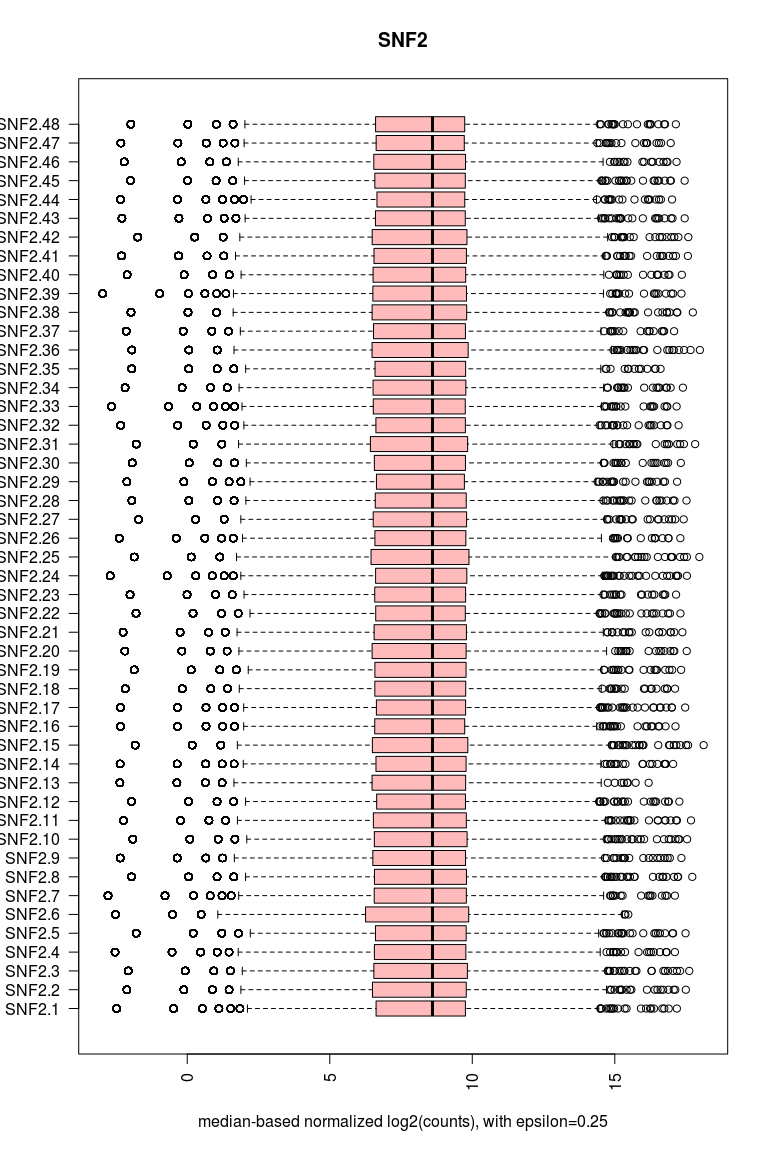
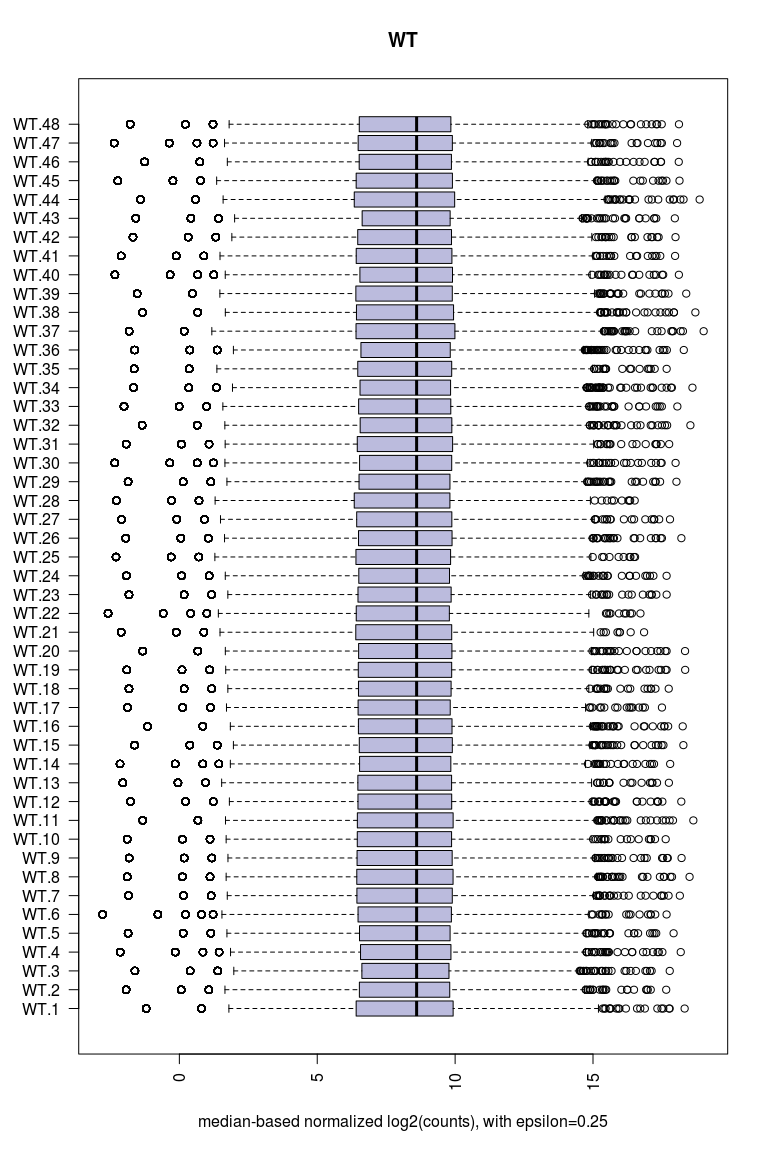
1. Sum of counts in the sample.
2. Median count per sample.
3. Some illustrative percentiles (0=min, 5, 10, 25, 50, 75, 90, 95, 100=max)
4. Inter-Quartile Range (IQR).
5. Classical statistical estimates (mean, variance, standard deviation).
6. Any other statistics that you moght find relevant.

The result table must contain one row per sample (column of the original count table), and one column per statistics (points a to f above).

### Exercise 2. Sample-wise standardization

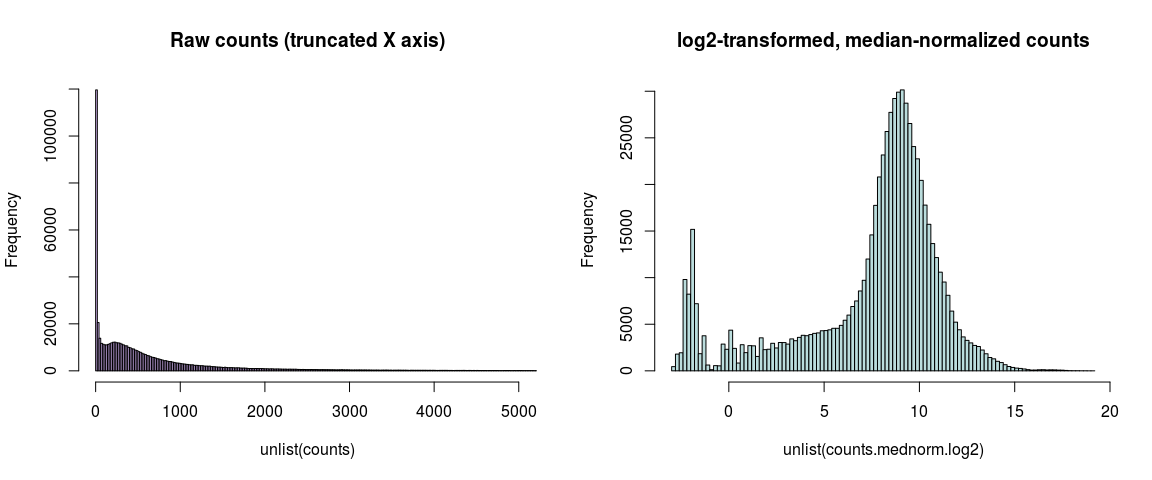
1. Standardize the count table by scaling all values accoding to the median per sample (do it for the counts and for the log2-transformed counts).
2. Compare the scaling factors that would be applied with libsum and median-based standardization, respectively (draw a plot comparing the scaling factors for each sample).

#### Boxplots of sample-wise standardized counts (log2-transformed)



### Exercise 3. Normalizing effect of the log2 transformation

Plot an histogram of the raw counts (all the table) versus log2-transformed median-based normalized counts.



This distribution deos not look like any conventional distribution, but this is not surprizing. Indeed, each gene can be considered as a separate object for which we measure the level of expression in two different samples, and which follows its own distribution. The histograms above can thus be understood as a mixture of ~7000 (number of genes) times 2 (nuber of genotypes) independent distributions, each of which is represented by 48 observations (replicates).

### Exercise 4. Statistics per gene (rows of the count table)

Based on the standardized log2-transformed counts, compute a table with statistics per gene: for each gene, compute the mean, sd, var, median, percentiles 25 and 75, iqr, ...

### Exercise 5. Comparison between WT and mutant genotypes

1. Compute the mean counts per gene for each genotype, using the median-based standardized log2-transformed counts.
2. Draw a plot to compare the mean expression per gene between WT and SNF2 genotypes.

## Next steps

1. Do the exercises above (exploration of the data)
2. Learn to use the *DESeq2* and *edgeR* packages by using the Vignettes.
3. Run DESeq2 and edgeR on the yeast dataset, in order to detect differentially expressed genes between WT and Snf2.
4. After that, you will analyse the robustness of the lists of differentially expressed genes by running resampling approaches (bootstrap, jacknife, subsampling).

Regularly we discuss about the results and interpret.