Quantification

In this exercise we will work with RNA-seq data from a study by Serin et al (2017). RNA-seq was performed on Arabidopsis seeds matured at standard temperature (ST, 22°C day/18°C night) or at high temperature (HT, 25°C day/23°C night). Both conditions have three biological replicates. The RNA-seq reads are available from the Sequence Read Archive (SRA) at the NCBI: https://www.ncbi.nlm.nih.gov/Traces/study/?acc=SRP125020

The reads were downloaded as fastq files using the fastq-dump program from the SRA toolkit: https://www.ncbi.nlm.nih.gov/sra/docs/toolkitsoft/

The reads were mapped to the Arabidopsis genome using the **Hisat2** program and quantified using **stringtie** and **prepDE.py** to get the read counts per gene (Pertea et al. 2016). A text file called **ST_vs_HT.csv** with read counts per gene for each of the samples is available at: http://www.bioinformatics.nl/courses/RNAseq/

Part I, using Excel

Download the file (using “save target as…” from the right mouse button menu)

Open this file with Excel. You should get a sheet with data in seven columns. The first column contains Arabidopsis gene IDs, columns B-G contain read count data for three replicates from two experimental conditions.

Let us first clean up the data a bit by removing genes that have no counts. In cell H1 type **max** and in cell H2 =MAX(B2:G2) This will give the maximum count for each of the six samples for the first gene.

Now move your mouse to the right bottom corner of cell H2 until the cursor changes to a black +. If you double click the +, the formula will be repeated for all rows that have data.

Select columns A-H and from the Data tab click on the Filter button. Now you should be able to sort the max column on highest to lowest. Remove all rows that have a max count of 0.

Next we will add some descriptions to the gene IDs to find out what the genes with the highest counts acctually are.

Browse to http://plants.ensembl.org/ and click on ‘BioMart’ Choose database ‘Ensembl Plants Genes ’, dataset ‘Arabidopsis thaliana genes’. In the menu on the left click on ‘Filters’ and in the center pane expand ‘GENE’. For “Limit to genes (external references)” select ‘With TAIR ID(s)’.

In the menu on the left click on ‘ ‘Attributes’ and in the center pane expand ‘GENE’. Here, make sure only ‘Gene stable ID’ and ‘Gene description’ are selected (i.e.

deselect ‘Transcript stable ID’). If you now click on the ‘Results’ button you should get a small table with gene IDs and corresponding descriptions. Press the ‘Go’ button next to ‘Export all results to File TSV’ and a file called ‘mart_export.txt’ should be downloaded by your browser.

Going back to Excel, select the tab for the (empty) Sheet1 and select ‘Import’ from the ‘File’ menu to import a Text file. Import the gene IDs with their descriptions in Sheet1.

Going back to the sheet with the counts, in cell I1 type **description** and in I2 put the following formula:

=VLOOKUP(A2,Sheet1!A:B,2,FALSE)

This will lookup the value of the gene ID for each row with count data in the list of descriptions.

Again double click the right bottom corner of the cell to copy the formula to the other rows as well.

Given the origin of the mRNA, do the genes with the highest counts make sense? Now let’s have a look at the data in R, using R Studio.

Part II, using R

We will again have a look at the count data, but this time in R.

Open R Studio, create a new R script, paste the following line in it (without the line breaks):

```
counts =
read.table("http://www.bioinformatics.nl/courses/RNAseq/ST_vs_HT.csv",
sep=",", row.names=1, header=TRUE, stringsAsFactors=FALSE)
```

Select the line and run it.

This will load a table with the count data, you can view it using:

```
View(counts)
```

This should look similar to the Excel spreadsheet. To get the number of lines:

```
nrow(counts)
```

To view the counts for a specific gene in a specific sample:

```
counts["AT4G22890","ST_1"]
```

Now, we will remove genes without counts using the maximum counts per row:

```
mx = apply(counts,1,max)
counts = counts[mx>0,]
```

mx is a list with the maximum counts for each row, and every row for which this value is larger than 0 is selected in the second statement.

To check how many rows (genes) we have left:

```
nrow(counts)
```

We will now zoom in on the first two samples, which are biological replicates. First we select the columns:

```
ST_1 = counts[,"ST_1"]
ST_2 = counts[,"ST_2"]
```

To get an idea of the values we can look at a summary:

```
summary(ST_1)
```

Or plot the values sorted from low to high:

```
plot(sort(ST_1))
```

Or look at the histogram:

```
hist(ST_1)
```

You will probably notice that the plots are not very informative, because of a few genes with many counts. To get a better overview of the counts per gene it helps to first take the logarithm of the counts:

```
plot(sort(log2(ST_1)))
hist(log2(ST_1))
```

To assess how similar the biological replicates are, we can use a scatterplot

```
plot(log2(ST_1),log2(ST_2))
```

In this plot every dot represents a gene, with the (log2 transformed) counts in one sample on the x-axis and the counts in the other sample on the y-axis.

How similar are the replicates, based on this plot?

The data we have worked with were produced by mapping reads onto the genome using the Hisat2 program, now we will have a look at counts produced by mapping reads onto the transcriptome by the **kallisto** program. Kallisto performs pseudomapping and directly returns the counts normalized as **tpm**, transcripts per million. It also returns estimated counts that should be comparable to the counts we got from Hisat2. However, the counts are now per transcript and not per gene.

To load the kallisto data run (again without the line breaks):

```
kallisto =
read.table("http://www.bioinformatics.nl/courses/RNAseq/ST_1_kallisto.t
sv",sep="\t",row.names = 1, header=TRUE,stringsAsFactors = FALSE)
```

You can again view it:

```
View(kallisto)
```

To get the number of transcripts that were quantified:

```
nrow(kallisto)
```

The tpm value for a transcript is actually the fraction multiplied by a million to get a more readable number (i.e. 7 instead of 0.000007). What would you expect the sum of the tpm values for all transcripts in a sample to be?

Let’s find out:

```
sum(kallisto$tpm)
```

Both kallisto and Hisat2 used the same reads, so the counts produced by these tools should (hopefully) be comparable. To check that they are, we first have to aggregate kallisto's transcript counts to gene counts. A simple way to do that is to just sum up the counts for all transcripts that belong to the same gene.

To add a column to the kallisto table that represents the gene for a transcript, we can strip off the last part of the transcript identifier, i.e.: AT3G18710.1 -> AT3G18710

The substring function can be used for this:

```
kallisto$gene = substring(rownames(kallisto),1,9)
```

Now we can use the aggregate function to sum up the counts:

```
kallisto_gene = aggregate(est_counts ~ gene, data=kallisto, sum)
```

To create a scatterplot of the kallisto and Hisat2 counts we can first combine the data into one table using the merge function:

```
combined_counts =
merge(kallisto_gene,counts,by.x="gene",by.y=0,all=FALSE)
```

With the **by.x** and **by.y** values we instruct merge to use the 'gene' column fo the kallisto table and the row names of the Hisat table to combine rows for the same gene. The **all=FALSE** option removes all genes that do are not present in both tables.

Now we can plot the data for the first sample ST_1, again as log2 values:

```
with(combined_counts,plot(log2(est_counts),log2(ST_1)))
```

How well do both methods agree?

Let's find out tomorrow how this effects the subsequent identification of differentially expressed genes...

References

Pertea, M., D. Kim, G. M. Pertea, J. T. Leek and S. L. Salzberg (2016). "Transcript-level expression analysis of RNA-seq experiments with HISAT, StringTie and Ballgown." Nature Protocols 11: 1650.

Serin, E. A. R., L. B. Snoek, H. Nijveen, L. A. J. Willems, J. M. Jiménez-Gómez, H. W. M. Hilhorst and W. Ligterink (2017). "Construction of a High-Density Genetic Map from RNA-Seq Data for an Arabidopsis Bay-0 × Shahdara RIL Population." Frontiers in Genetics 8: 201.

The counts were produced using these Linux commands:

Hisat2

```
hisat2 -x TAIR11 -U SRR6293018.fastq -S SRR6293018.sam samtools sort -o SRR6293018.bam
SRR6293018.sam stringtie -e -B -G genes.gtf -o SRR6293018.gtf SRR6293018.bam python prepDE.py
```

Kallisto

```
kallisto quant -i TAIR.idx -o SRR6293018 -b 100 --single -l 200 -s 20 -t 10 > SRR6293018.fastq
```