RNA-Seq data analysis

Denis Puthier

11 janvier 2016

Introduction

The "Tuxedo Suite" has been developed for RNA-Seq data analysis. It is mainly composed of Bowtie, Tophat, Cufflinks, CuffDiff. It is intended to provide powerful solutions for read mapping, discovery of novel gene structures and differential expression analysis. In the practical session we will use this suite to analyz e samples obtained from P5424 thymocyte cell line.

Galaxy servers

The Galaxy server motto is "Data intensive biology for everyone". Galaxy is a web-based framework for data intensive biomedical research. Galaxy can be install easily on any computer but is also proposed through remote access by numerous research groups. It is intended to ease the development of complex workflows to analyze various types of biological data. Although it has been historically oriented toward NGS data analysis (ChIP-Seq, RNA-Seq, Ribosome profiling...), lots of public servers are also proposing set of tools dedicated to genomics, proteomics, image analysis, cancer stem cell analysis... The public server web page of the galaxy team lists all the publicly available servers throughout the world (n=70 at the time of writing).

Connecting to the pedagogix Galaxy server

NB: Note that the pedagogix server is only maintained for pedagogic purposes to propose privileged access to students from M2 BBSG and Polytech. It is not intended to be a production server as it is not heavily maintained.

- Open a connection to pedagogix Galaxy server. If this is your first connection, use the Register command. Otherwise, enter your login (use Login in the User menu at the top of the Galaxy window).

Loading fastq files in galaxy

Analysis of the whole dataset would be time consuming. To make the analysis feasible within a reasonable time, data were previously mapped to the mouse genome (version mm9). A subset of reads that aligned onto chromosome 18 was extracted and will be used for this tutorial. The following dataset are available.

File name	Experiment	Description
DM1_chr18-20Mto50M_R1.fq.gz	Control DMSO, replicate 1	Right end read.
DM1_chr18-20Mto50M_R2.fq.gz	Control DMSO, replicate 1	Left end read..
DM2_chr18-20Mto50M_R1.fq.gz	Control DMSO, replicate 2	Right end read.
DM2_chr18-20Mto50M_R2.fq.gz	Control DMSO, replicate 2	Left end read.

File name	Experiment	Description
DM3_chr18-20Mto50M_R1.fq.gz	Control DMSO, replicate 3	Right end read.
DM3_chr18-20Mto50M_R2.fq.gz	Control DMSO, replicate 3	Left end read.
PI1_chr18-20Mto50M_R1.fq.gz	PMA/Ionomycine treated, replicate 1	Right end read.
PI1_chr18-20Mto50M_R2.fq.gz	PMA/Ionomycine treated, replicate 1	Left end read..
PI2_chr18-20Mto50M_R1.fq.gz	PMA/Ionomycine treated, replicate 2	Right end read.
PI2_chr18-20Mto50M_R2.fq.gz	PMA/Ionomycine treated, replicate 2	Left end read.
PI3_chr18-20Mto50M_R1.fq.gz	PMA/Ionomycine treated, replicate 3	Right end read.
PI3_chr18-20Mto50M_R2.fq.gz	PMA/Ionomycine treated, replicate 3	Left end read.

These datasets are available directly in Galaxy to avoid network issues. We will start by analyzing the DM1 sample (control DMSO, replicate 1).

- In the upper left corner, click on **Unnamed history** and rename this workspace to **DM1**.
- Select **Shared Data > Data Libraries > TlemCen 2016 > DM1 > DM1_chr18-20Mto50M_R1.fq > Import this dataset into selected history**. In the new window select **DM1** as **Destination history**. Click on **Import library dataset**.
- Select **Analyze Data** in the upper menu.
- Using the pencil, rename the dataset to **DM1_R1**.
- Select **Shared Data > Data Libraries > TlemCen 2016 > DM1 > DM1_chr18-20Mto50M_R2.fq > Import this dataset into selected history**. In the new window select **DM1** as **Destination history**. Click on **Import library dataset**.
- Select **Analyse Data** in the upper menu.
- Using the pencil, rename the dataset to **DM1_R2**.
- Click the eye icon to display the content of DM1_R1 file.

- How is the quality encoded?
- What can you say about the quality of the first encountered reads?

Quality control with FastQC

FastQC aims to provide a simple way to do some quality control checks on raw sequence data coming from high throughput sequencing pipelines. It provides a modular set of analyses that you can use to give a quick impression of whether your data has any problems of which you should be aware before doing any further analysis. FastQC can be run as a stand alone interactive application (for the immediate analysis of small numbers of FastQ files) or in a non-interactive mode (through shell commands) where it would be suitable for processing large numbers of files.

It is important to stress that although the analysis results appear to give a pass/fail result, these evaluations must be taken in the context of what you expect from your library. A 'normal' sample as far as FastQC is concerned is random and diverse. Some experiments may be expected to produce libraries that are biased in particular ways. You should treat the summary evaluations therefore as pointers to where you should concentrate your attention and understand why your library may not look random and diverse.

This web site provides you with some nice example of sequencing failures and may help you in analyzing fastqc outputs.

- Use **NGS: QC and manipulation > FastQC:Read QC**.
- Select the first fastq file (**DM1_R1**) and press **Execute**.

- Display the data for the corresponding fastqc result (use the view (eyes) icon above the dataset name in the right panel).
- Carefully inspect all the statistics.

Perform the same operation for **DM1_R2** file.

- What do you think of the overall quality of the sequencing?
- Carefully inspect all diagrams. The FastQC documentation contains a section that explains the meaning of each diagram/
- What is the format of quality encoding? You need to know it to perform next step (read trimming).

Read trimming

Read trimming is a pre-processing step in which input read ends are cut (most generally the right end). One should keep in mind that this step may be crucial depending on the aligner used. Indeed most aligners will be unable to align a large fraction of the dataset when poor quality ends are kept. Several programs may be used to perform sequence trimming:

- FASTX-Toolkit
- sickle
- the ShortRead Bioconductor package
- ...

Here we will use sickle.

- Search for the sickle tool using the galaxy search engine (upper left corner).
- Select sickle tool.
- Set **Single-End or Paired-End reads** to **Paired-end**.
- From **Paired-End Forward Strand FastQ Reads** dropdown list select 'DM1_R1'.
- From **Paired-End reverse Strand FastQ Reads** dropdown list select 'DM1_R2'.
- Set **Quality Threshold to 20, Length Threshold to 25** and press execute.
- Rename **Paired-End forward strand output of Sickle** to DM1_R1_trim
- Rename **Paired-End reverse strand output of Sickle** to DM1_R2_trim
- Perform a new fastqc analysis using the trimmed read as input.
- The number of reads should be reduced.

- What does the **Singletons from Paired-End** file contain?
- Delete **Singletons from Paired-End** dataset.
- How many read to you retrieve after trimming?
- How does it compare with the input fastq files?

Getting the sequence of mouse chromosome 18 at UCSC

Most of the time the galaxy server will provide you with an already indexed genome that can be used by tophat to perform read alignment. In this practical, we would like to restrict the alignment to mouse chromosome 18 (this will be faster). We thus need to download the sequence of mouse chromosome 18. This sequence will be provided to tophat in the subsequent steps (tophat will perform sequence indexing internally by calling bowtie-build).

The sequence of the chr18 (mm9 build) can be downloaded through the UCSC web site.

- Go to the UCSC ftp web site. Copy the link address to **chr18.fa.gz**.
- Select **Tools > Get Data > Upload File**. In the text area (**URL/Text**) paste the link to the chr18 sequence.
- Select **fasta** as **File Format** and **mm9** as a reference genome. Press Execute to import the sequence into your history.
- Rename the record in the history to **chr18_mm9.fa**.

- Check the first lines and last line of the file using head and tail respectively (**Text Manipulation > head-or-tail**).
- How to you explained the N stretch inside the sequence ?

NB: the chromosome sequence can also be obtained from ensembl ftp web site.

Getting the size of the chromosomes

Several programs need to know about chromosome length to perform dedicated task. Chromosome information can be obtained using **UCSC** whose **table-browser** is interfaced in Galaxy.

- Use **Get Data > UCSC Main table browser**.
- Set : **Clade** to Mammal, **Genome** to Mouse, **assembly** to "July 2007 (NCBI37/mm9)", **group** to **All tables**, **database** to mm9 and table to **chromInfo**.
- Set **output format** to **all fields from selected table** and **Send output** to **Galaxy**.
- In the new web page press **Send query to galaxy**.
- Rename the dataset to **mm9_chrom_info_txt**.

- What does this file content?
- Use **Text Manipulation > Cut** and **Statistics > Summary Statistics** to compute the median size of a mouse chromosome.

Getting transcript annotation in gtf format

In order to provide topHat with the location of known exons in the human genome, we will download a file in GTF format (Gene transfer format). You can get more information about this format on UCSC web site or GENCODE web site.

GTF file can be obtained both from **UCSC** table browser or **ensembl** ftp web site.

NB: it is very important at this step to ensure that the fasta file and the GTF file are obtained from the genome release (here mouse genome version mm9/GRCm37). The chromosome sequences and gene positions vary between genome releases.

Here we will use a GTF file containing information related to transcripts from mouse chromosome 18. This GTF file **was obtained from GENCODE** web site (Version M1 July 2011). Annotations are based on Ensembl server version 65.

- Select **Shared Data > Data Libraries > TlemCen 2016 > GTF > chr18_20M-50M_gencode_vM1.gtf > Import this dataset into selected history**. In the new window select **DM1** as **Destination history**. Click on **Import library dataset**.

- Select **Analyze Data** in the upper menu.**.

- Check the first lines of the GTF file. What kind of information is enclosed in this file?

Mapping read with TopHat

TopHat is a fast splice-aware junction mapper for RNA-Seq reads. It aligns RNA-Seq reads to mammalian-sized genomes using the ultra high-throughput short read aligner Bowtie, and then analyzes the mapping results to identify splice junctions between exons.

We will start by mapping the reads corresponding to control sample.

- Select **NGS: Mapping > Tophat2** from the toolbox.
- Set "Is this library mate-paired?:" to "Paired-end".
- Set **RNA-Seq FASTQ file, forward reads** to "DM1_R1_trim".
- Set **RNA-Seq FASTQ file, reverse reads** to "DM1_R2_trim".
- Set **Use a built in reference genome or one from your history** to "Use a genome from history".
- Set **Select the reference genome** to **chr18_mm9.fa**.
- Set **TopHat settings to use** to **Full parameter list**.
- Set **Maximum number of alignments to be allowed** to 1.
- Set **Library Type** to **FR First strand**.
- Set **Use Own Junctions** to **yes**.
- Set **Use Gene Annotation Model** to **yes**.
- Set **Gene Model Annotations** to ** chr18_20M-50M_gencode_vM1.gtf**.
- Press **Execute**.
- Rename the **accepted_hits** dataset to **DM1_alignments**.
- Rename the 'splice-junction' bed file to DM1_splice_junctions.bed.

NB: By default tophat will accept reads whose genomic mapping is ambiguous. This multi-mapped reads may be problematic in the downstream analysis. Indeed, keeping them may introduce spurious transcript models when trying to reconstruct underlying transcripts. However discarding them may also be problematic when computing expression levels of gene families. The **Maximum number of alignments to be allowed** argument instructs TopHat to allow up to this many alignments to the reference for a given read, and choose the alignments based on their alignment scores if there are more than this number. The default is 20 for read mapping. Depending on the need, more stringent policy may be chosen (e.g setting "Maximum number of alignments to be allowed" to 1 indicates that muti-mapped reads should be discarded). Additional arguments are available in the command line version of tophat including -x/–transcriptome-max-hits and -M/–prefilter-multihits.

- Is this GTF mandatory for tophat?
- What is the benefit of providing tophat with a GTF?
- What are the benefits and drawbacks when selecting 1 for argument **Maximum number of alignments to be allowed**? What is your feeling?

Checking the number of aligned reads

We will used **samtools flagstat** to assess the number of aligned read available in the bam file.

- Select **Statistics > flagstat**.
- Select the **BAM** file and press **Execute**.

- Check the statistics. Is that expected?
- How does it compare with the input right-end and left-end fastq files?

- How does it compare with the number of trimmed reads?
- How does it compare with the number of raw reads?

Viewing the results with Integrated Genome Browser (IGV).

The Integrative Genomics Viewer (IGV) is a high-performance visualization tool for interactive exploration of large, integrated genomic datasets. It supports a wide variety of data types, including array-based and next-generation sequence data, and genomic annotations.

- Create an IGV account here.
- Download IGV and launch it with 750 MB or 1.2 Gb depending of your machine.
- Select **mm9** as a genome and browse to **chromosome 18**.
- In galaxy select tophat result (bam format) and download the **bam** file.
- In galaxy select tophat result (bam format) and download the **bai** file.
- In IGV, go to the **Egr1** gene (by typing 'Egr1' in the **GO** text area).
- Zoom to view alignments.
- In the left panel right click on the bam file name and select **View as pairs**.
- In the left panel right click on the bam file name and set **Color Alignments by > Read strand**.
- In galaxy select tophat result (**control_splice_junctions.bed**) and download the **bed** file. If this file does not contain a **bed** extension, rename it to add **.bed**.
- Load the **control_splice_junctions.bed** into IGV (File > Load from file).
- Unzoom to view the number of alignments supporting exon junctions.
- Mouse over a junction on of **control_splice_junctions.bed** track. What is the **Depth** about ?

- What is the strand of gene 'Egr1'?
- In the Egr1 gene some exonic region are larger than others ? What are they?
- Where are the 3' and 5' UTR regions?
- Regarding reads, what does the blue and pink color indicate?
- Mouse over a paired read. What are the meanings of the following tags/keys:
  - CIGAR?
  - Mapped?
  - Mapping quality?
  - Secondary?
  - Duplicate?
  - Mate-is mapped?
  - Insert-size?
  - Pair-orientation?
  - First in pair?
  - Second in pair?
- What are the meaning of :
- NH?
- NM?
- Mouse over **several paired alignments** on Egr1. What are the values of the **pair-orientation** keys?

- Go to internal exons of Etf1 (this gene is located just 40kb away on the 3' side of Egr1).
  - What is the strand of Etf1?
  - What are the values of **pair-orientation** key on **paired alignments**?
  - Look at additional gene examples.
  - What can you conclude regarding **paired alignments** values?
  - How would you isolate the signal emitted from the plus and minus strands?
- Looking at **Nr3c1** you will find some signal extending from the 5' region?
  - Is it produced by the plus or minus strand?

Creating a bigwig track

As you may have notice the user needs to zoom inside IGV to visualize the alignments. This is due to the fact that BAM files may be very large (tens of Gb or more). Loading all information from the file would thus saturate the computer memory. We will thus create a more lightweight file that will just provide us with the mean coverage of each genomic region. This file in bigWig format will be compressed and indexed as the BAM files.

- Use the **NGS: RNA Analysis > BAM to Wiggle** tool to convert the BAM file to a wiggle format (the uncompressed and unindexed version of the BgigWig format).
- Set **Strand-specific** to **Paired-end**.
- Set **Pair-End Read Type** to **read1 (positive –> negative; negative –> positive), read2 (positive –> positive; negative –> negative)**.
- Set **Chromosome size file** to **mm9_chrom_info_txt**.
- Press **Execute**.

- Why does the output contain two files?

For the two wiggle files:

- Select **Convert Formats > Wig/BedGraph-to-bigWig**.
- Click on **Execute**.
- Rename the output obtained from ** Wiggle on Forward Reads** to **DM1_plus.bigwig**.
- Rename the output obtained from ** Wiggle on Reverse Reads** to **DM1_minus.bigwig**.
- Download the subsequent bigwig file and load it into **IGV**.
- In **IGV**, on the **left panel**, right click on the bigwig track name. Use **Set data range** and set the value **min, mid and max** value to **-200, 0, 200** respectively.
- Unzoom.

Searching for novel transcript with cufflinks

We can now use the cufflinks software to try to discover new transcripts inside the dataset. We will also provide cufflinks with the set of known transcript.

- In the toolbox, select **NGS: RNA Analysis > Cufflinks**.
- Select the bam file in the **SAM or BAM file of aligned RNA-Seq reads** menu.

- Set **Set advanced Cufflinks options** to **yes**.
- Set **Use Reference Annotation** to **Use reference annotation as guide**.
- Set **Reference Annotation** to **chr18_20M-50M_gencode_vM1.gtf**.
- Set **Library prep used for input read** to **fr-firststranded**.
- Press **execute**.
- Rename **assembled transcript** dataset to **DM1_transcripts**.

- Have a look at the **assembled transcripts** file produced by cufflinks. What are the gene_ids, transcript_ids?
- What additional information is provided?
- Load the **assembled transcript** file produced by cufflinks into IGV.
- What can we say about the transcripts produced by the **Pura** gene?

Extracting a workflow

Galaxy allows user to apply the developed pipeline to another set of sample. To this aim, the user must create a **workflow**.

- In the history menu, select **history options**.
- Click on **Extract workflow**.
- Set the name of the new workflow to **RNA-Seq mapping and transcript discovery.**.

- Using the menu go to **workflow > RNA-Seq mapping and transcript discovery > edit**.
- Have a look at the workflow.
- Rename the input elements to **Read_1, Read_2, CHROM_SIZE and GTF** according to their connections in the workflow.

Apply the workflow to PI1_chr18_20M-50M

We will now apply this workflow to the sample corresponding to **the activated thymocyte (replicate 1)**.

- Create a new history: **History > Create new**.
- Rename this workspace : **PI1**.
- Select **Shared Data > Data Libraries > TlemCen 2016** .
- Open all folder and select (radio button): **mm9_chr_size.txt**, **chr18_mm9_fa**, **chr18_20M-50M_gencode_vM1.gtf**, and the two fastq files from **PI1** sample.
- Use **For selected datasets > import to current history** and click **GO**.
- Click on **Galaxy** (top left) to go back to your history (PM1). You should see the five datasets.
- In the top menu select **workflow > RNA-Seq mapping and transcript discovery > edit**. Have a look at your new workflow. Check the input files.

- Select **workflow > RNA-Seq mapping and transcript discovery > run**. Set the proper input files.
- Click **Run workflow** at the bottom of the page.
- Rename **assembled transcript** dataset to **PI1_transcripts**.
- Rename the **accepted_hits** dataset to **PI1_alignments**.
- Rename the 'splice-junction' bed file to **PI1_splice_junctions.bed**.

- Load the **PI1_splice_junctions.bed** and **PI1_alignments** files into IGV.
- Go to the **Egr1** gene. What can you see?

Apply the worflow to all other samples (facultative)

You may apply the workflow to all replicates. Do to that, create successively an history for storing DM2, DM3, PI2 and PI3. Import the corresponding files into the history and run the workflow. Don't forget to import **mm9_chr_size.txt**, **chr18_mm9_fa** and **chr18_20M-50M_gencode_vM1.gtf** files.

Creating a workspace to compare samples from both classes

Create a new history entitled **DM versus PI**. Copy the datasets below in this history (use **history > Copy datasets**).

- The bam and bai files (accepted_hits that should have been renamed **PI1_alignments** and **DM1_alignments**) for all samples.
- The **assembled transcripts** files (cufflinks results) from all samples.
- The gtf file (**chr18_20M-50M_gencode_vM1.gtf**).
- The **mm9_chr_size.txt** file.

Merging the reference and inferred genomic annotations

We now have at least three different GTF files (depending on whether you have processed DM2,DM3,PI2,PI3):

- The reference annotation
- The discovered transcripts in the control sample(s).
- The discovered transcripts in the activated sample(s).

We will ask **cuffmerge** to merge the novel annotations (obtained through cufflinks) with the reference (known annotation) and to classify the transcripts. It will annotate transcripts by producing a GTF file containing flags. Some of this flags may indicate that:

- The transcript is unknown (class code "u").
- The transcript is a novel isoform of a known transcript (class code "j").
- The transcript is the same as the original/known transcript ((class code "=").
- ...

For a full description of all possible flags ("class code"), please refer to the cuffmerge web site (section 'Transfrag class codes').

Here we will concentrate on retrieving the position of novel transcripts.

- Now select **NGS: RNA Analysis > cuffmerge**. Set **GTF file(s) produced by Cufflink** to the two **assembled transcript** files (**DM1__transcripts**...).
- Select **Use Reference Annotation**. Set **Reference Annotation** to **chr18__20M-50M__gencode__vM1.gtf**.
- Press **Execute**.
- Use **Filter and sort > Select lines that match an expression**. Select line containing **class_code "u"**. The 'u' indicate they are unknown genes (not present in the reference annotation).
- Merge this unknown transcript with the reference annotation (**chr18__20M-50M__gencode__vM1.gtf**) using **Text Manipulation > Concatenate datasets**.
- Rename the file to **assembly.gtf**.

- How many transcripts were classified as unknown?

Quantification

The objective of quantification is to estimate the expression level of each gene by counting the number of reads overlapping each gene model. Several programs have been developed for this task (cuffdiff, featureCount, HTSeq-count,...). The FeatureCounts software is a lightweight read counting program written entirely in the C programming language. It has a variety of advanced parameters but its major strength is its outstanding performance (10GB SE BAM file takes about 7 minutes on a single average CPU).

- Copy the two bam files in the **assembly and quantification** history.
- Select **NGS: RNA Analysis > featureCounts**.
- Select the two bam files in **Alignment file**.
- Set **GFF/GTF Source** to Use reference from history.
- Select **assembly.gtf** as Gene annotation file.
- Set **featureCounts parameters** to **extended settings**.
- Set **GFF feature type filter** to exon (we want to count inside exonic regions).
- Set **GFF gene identifier** to **gene__id** (all exons of all transcripts of a given gene will be summed up to get the final expression value).
- Set **Strand specific protocol** to **Stranded (reverse)"**.
- Set **Minimum read quality** to 12.
- Select **PE Count fragments instead of reads:**.
- Click **Execute**.
- Check the **Summary file**. What are **Unassigned__MultiMapping**, **Unassigned__NoFeatures**, **Unassigned__MappingQuality**, **Unassigned__Chimera** ?

Descritive statistics with R

- Download the gene expression table produced by featureCount.
- Rename the file to **raw__counts.txt**.
- Open **RStudio**.

```
## First we read gene counts
count <- read.table("raw_counts.txt" ,sep="\t", head=T,row=1)
#head(count)
#head(rownames(count))
#colnames(count)
```

```r
# Change column names
colnames(count) <- c("control", "Treated")
## Values are log2 transformed
## (a pseudo-count is added in case one of the sample is equal or close to zero)
count <- log2(count +1)

## Checking distribution of FPKM values
hist(as.matrix(count), main="Distribution of count values")
```

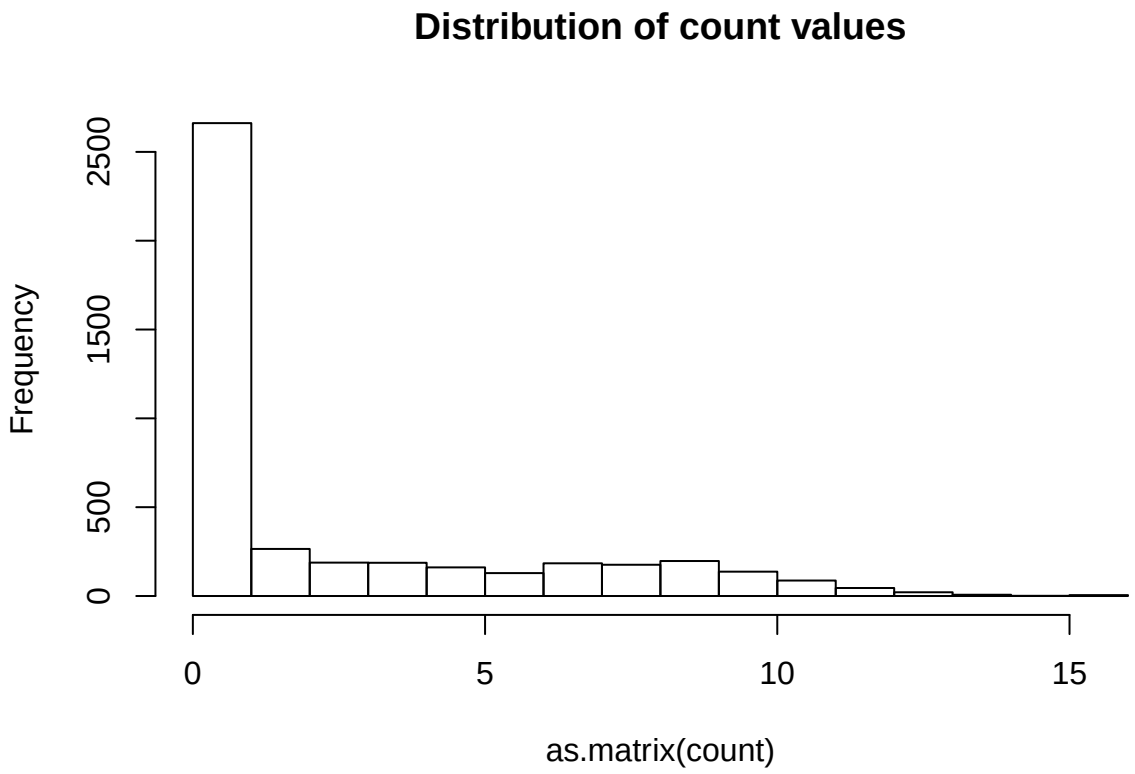

Figure 1:

```r
boxplot(count, col=c("red","gray"), pch=16, main="Boxplot for count valaues")
```

```r
## Scatter plot comparing expression levels in sample 1 and 2
par(xaxs='i',yaxs='i')
plot(count, pch=20, panel.first=grid(col="darkgray"))
identify(count[,1], count[,2],lab=rownames(count))
```

```
## integer(0)
```

What's about ENSMUSG00000038418 ?

The gene ENSMUSG00000038418 seems to be strongly induced during T-cell activation. Go to Ensembl genome Browser and use the search area to get some information about it.

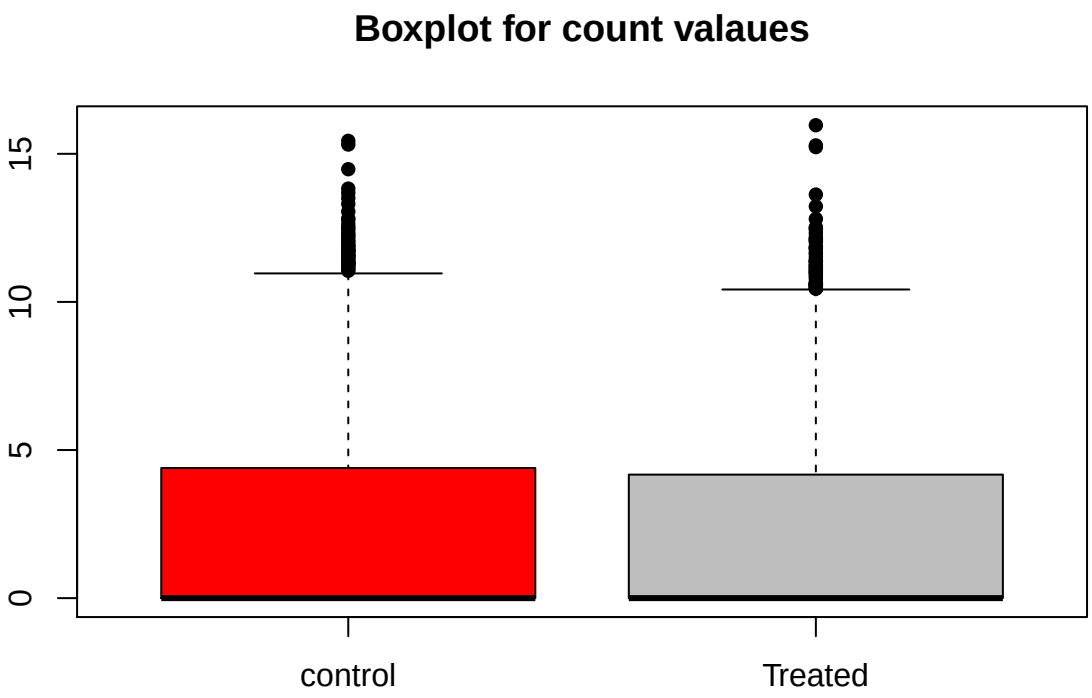

Figure 2:

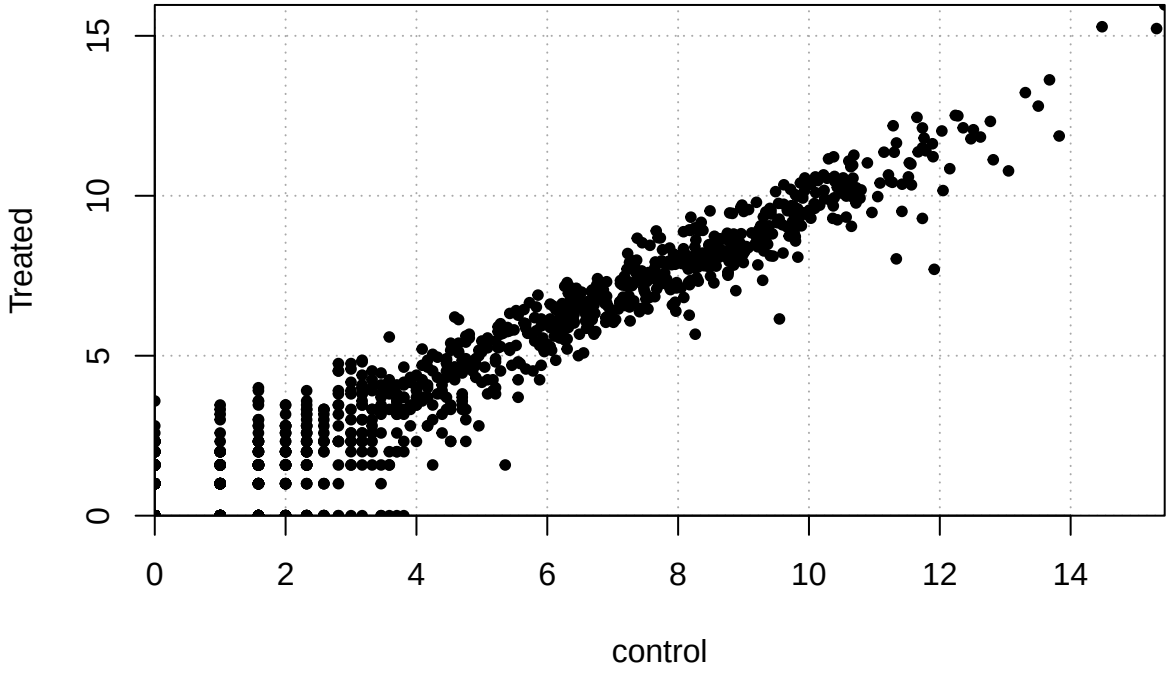

Figure 3:

- Go to the ensembl genome browser.
- In the **search panel** select **Mouse**. In the text area of the search panel type **ENSMUSG00000038418**.
- Click on entry **ENSMUSG00000038418**. What is the meaning of the **ENSG** prefix ?
- What is the **description** associated to EGR1 ? What does it mean ?
- Ask for **Show transcript table**. How many transcripts are associated with EGR1 ?
- What about the **ENST** prefix ?
- What is the **biotype of this transcript** ?
- Some links are provided toward **RefSeq** and **UniprotDB**. What kind of information do they store ?

Differential expression analysis with DESeq2

We have seen that gene Egr1 is strongly induced upon PMA/Ionomycin treatment. Our purpose is now to define a list of differentially expressed genes. To this aim we will use DESeq2 R library (interfaced in Galaxy server). DESeq2 will perform a statistical test that will point out some genes whose expression differs between both conditions. At these step we will work with the full dataset that can be obtained through **Shared Data > Data Libraries > TlemCen 2016 > COUNTS > DM_vs_PI_gene_counts.txt > Import this dataset into selected history**. In the new window select **DM versus PI** as **Destination history**. Click on **Import library dataset**.

- From the left menu, select **NGS: RNA Analysis > Differential_Count**
- Set **DMSO** as the first treatment name (and select columns **DM1, DM2, DM3**).
- Set **PMA/Inomycin** as the second treatment name (and select columns **PI1, PI2, PI3**).
- Set **Run this model using edgeR** to **Do not run edgeR**.
- Set **Run Voom** to **Do not run Voom**.
- Set **Do not run DESeq2** to **Run DESeq2**.
- Set **fdr** as **FDR (Type II error) control method**.
- Click **Execute**.

- What is the first plot ?
- What can you say from the second plot ?
- Look at the produced **html file**. What can you guess from the heatmap ?

Selecting differentially expressed genes

We will now select differentially expressed genes by applying a threshold on the **adjusted p-value (padj) and log2-fold change.

- Select **Filter and Sort > Filter data on any column using simple expressions**.
- Set **Number of header lines to skip** to 1.
- Apply the following filter **c7 < 0.01** (i.e adjusted p-value < 0.01) to dataset **Differential-Counts_topTable_DESeq2.xls**.
- How many lines (genes) do you obtain as output ?
- Rename the output as **differentially_expressed_info**.

- Retrieve the list of upregulated genes by performing the following steps:

  - Select **Filter and Sort > Filter data on any column using simple expressions**.
  - Set **Number of header lines to skip** to 1.
  - Apply the following filter **c3 >= 1** (i.e log2 fold change >= 1) to dataset **differentially_expressed_info**.
  - Rename the dataset **Upregulated_info**.
  - Cut the column 1 from **Upregulated_info** (gene name) using **Text Manipulation > cut** and save it in a dataset called **Upregulated_gene_list**.
  - How many genes are called as 'upregulated' using these criteria.

- Retrieve the list of downregulated genes by performing the following steps:
  - Select **Filter and Sort > Filter data on any column using simple expressions**.
  - Set **Number of header lines to skip** to 1.
  - Apply the following filter **c3 <= -1** (i.e log2 fold change >= 1) to dataset **differentially_expressed_info**.
  - Rename the dataset **Downregulated_info**.
  - Cut the column 1 from **Downregulated_info** (gene name) using **Text Manipulation > cut** and save it in a dataset called **Downregulated_gene_list**.
  - How many genes are called as 'Downregulated' using these criteria.

Now we will extract the count for differentially expressed genes from the count matrix (DM_vs_PI_gene_counts.txt).

- Use **Concatenate datasets tail-to-head** to merge **Downregulated_gene_list** and **Upregulated_gene_list**. Rename this new dataset **Up_Down_gene_list**.
- Use **Join, Subtract and Group > Join two Datasets.**
- Set **Join** to **Up_Down_gene_list**.
- Set **using column** to **c1**.
- Set **with** to **DM_vs_PI_gene_counts.txt**.
- Set **and column** to **c1**.
- Click **Execute**.
- Rename the dataset **join_dataset**.
- Select **Text Manipulation > Cut**.
- Set **Cut columns** to **c2,c3,c4,c5,c6,c7,c8**.
- Set **From** to **join_dataset**.
- Rename the dataset **join_dataset_nr**.
- To retrieve the header, perform the following operations.
  - Use **Text Manipulation > Select first lines from a dataset** set **Select first** to 1 and from to **DM_vs_PI_gene_counts.txt**. Press **Execute**. Rename the output to **header**.
  - Use **Text Manipulation > Concatenate datasets tail-to-head** and concatenate the **header** dataset and **join_dataset_nr** dataset.
- Rename this dataset as **Up_Down_counts**.

Hierarchical clustering with cluster 3.0.

We will now apply a hierarchical clustering (class discovery) to check whether our selected genes are able to distinguish between PI and DM samples. First we will need to transform data into log2 scale.

- Select **Convert Formats > Add a pseudocount and perform log2 transformation**.

- Choose **Up_Down_counts** as input.
- Rename the dataset to **Up_Down_counts_log2**.
- Save the matrix onto disk.

- What is the objective of logarithmic transformation ?
- Why base 2 ?

- The **cluster 3.0** program should be already installed on your computer.
- Launch **cluster 3.0** and select.
- Use **Up_Down_counts_log2** as input file.
- Center each row (median).
- Distance measure for genes/rows clustering : **Pearson**.
- Distance measure for samples/columns clustering : **Pearson**.
- Hierarchical clustering method: **average**.
- Press **run**.

Now we can load the result in **Java Treeview**.

- launch **Java Treeview**.
- Use **File > Open** and browse to the .cdt file.
- Go to **Settings > Pixel settings** and set **Global > Y > Fill** and **contrast** to 3.

- What do you think about sample clustering?
- What do you think about gene clustering?
- What would you propose?

Test alternative methods for differential analysis (facultative)

Try to use edgeR or Voom to get a list of differentially expressed genes. Which one seems the most conservative ?

Performing functional enrichment analysis using DAVID

We will now use the DAVID database to perform functionnal enrichment analysis using our list of differentially expressed genes as input.

- Go to Database for Annotation, Visualization and Integrated Discovery (DAVID) web site
- In the left menu select **Functional annotation**.
- In the left menu select **Upload** tab.
- Copy and paste the list **Upregulated_gene_list** or load it into DAVID by using **B:Choose From a File**.
- In **Step 2: Select Identifier** set identifiers to **OFFICIAL_GENE_SYMBOL**.
- In **Step 3: List Type** select **Gene list**.
- Click on **Submit list**.
- If DAVID warns you about identifiers use option 1: **Continue to submit the IDs That DAVID Could map**.

- In the left panel set **Select to limit annotations by one or more species** to **Mus musculus**.
- Click **select species**.
- Select **Functional Annotation chart**.
- In **Combined View for Selected Annotation** select **Functional Annotation chart**.

- What are the terms that are enriched in the list of genes you provided?
- Does it seem biologically meaningful?
- How do you interprete the p-value?
- What could be the Benjamini column?
