ANALYSIS of RNA-seq datasets regarding SAC and CELL CYCLE INHIBITION

Oct 2nd, 2018

Here we describe the following steps that we had taken in the RNA-seq analysis of the samples :

****

```
library("ggplot2")
library("reshape2")
library("data.table")
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:reshape2':
##
##     dcast, melt
```

```
library("limma")
library("Glimma")
library("edgeR")
library("DESeq2")
```

```
## Loading required package: S4Vectors

## Loading required package: stats4

## Loading required package: BiocGenerics

## Loading required package: parallel

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB

## The following object is masked from 'package:limma':
##
##     plotMA

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind,
##     colMeans, colnames, colSums, dirname, do.call, duplicated,
##     eval, evalq, Filter, Find, get, grep, grepl, intersect,
##     is.unsorted, lapply, lengths, Map, mapply, match, mget, order,
##     paste, pmax, pmax.int, pmin, pmin.int, Position, rank, rbind,
##     Reduce, rowMeans, rownames, rowSums, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which, which.max,
##     which.min

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:data.table':
##
##     first, second
## The following object is masked from 'package:base':
##
##     expand.grid

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

## The following object is masked from 'package:data.table':
##
##     shift

## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb

## Loading required package: SummarizedExperiment

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

## Loading required package: DelayedArray

## Loading required package: matrixStats

##
## Attaching package: 'matrixStats'

## The following objects are masked from 'package:Biobase':
##
##     anyMissing, rowMedians

## Loading required package: BiocParallel

##
## Attaching package: 'DelayedArray'

## The following objects are masked from 'package:matrixStats':
##
##     colMaxs, colMins, colRanges, rowMaxs, rowMins, rowRanges

## The following objects are masked from 'package:base':
##
##     aperm, apply
```

```r
library("gplots")
```

```
##
## Attaching package: 'gplots'

## The following object is masked from 'package:IRanges':
##
##     space

## The following object is masked from 'package:S4Vectors':
##
##     space
## The following object is masked from 'package:stats':
##
##     lowess
```

```r
library("pheatmap")
library("ComplexHeatmap")
```

```
## Loading required package: grid
```

```
## ========================================
## ComplexHeatmap version 1.18.1
## Bioconductor page: http://bioconductor.org/packages/ComplexHeatmap/
## Github page: https://github.com/jokergoo/ComplexHeatmap
## Documentation: http://bioconductor.org/packages/ComplexHeatmap/
##
## If you use it in published research, please cite:
## Gu, Z. Complex heatmaps reveal patterns and correlations in multidimensional
##   genomic data. Bioinformatics 2016.
## ========================================
```

```r
library("scatterplot3d")
library("enrichR")
library("tidyr")
```

```
##
## Attaching package: 'tidyr'
```

```
## The following object is masked from 'package:S4Vectors':
##
##     expand
```

```
## The following object is masked from 'package:reshape2':
##
##     smiths
```

```r
library("plyr")
```

```
##
## Attaching package: 'plyr'
```

```
## The following object is masked from 'package:matrixStats':
##
##     count
```

```
## The following object is masked from 'package:IRanges':
##
##     desc
```

```
## The following object is masked from 'package:S4Vectors':
##
##     rename
```

```r
library("dplyr")
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
## The following object is masked from 'package:matrixStats':
##
##     count
## The following object is masked from 'package:Biobase':
##
##     combine
## The following objects are masked from 'package:GenomicRanges':
##
##     intersect, setdiff, union
## The following object is masked from 'package:GenomeInfoDb':
##
##     intersect
## The following objects are masked from 'package:IRanges':
##
##     collapse, desc, intersect, setdiff, slice, union
## The following objects are masked from 'package:S4Vectors':
##
##     first, intersect, rename, setdiff, setequal, union
## The following objects are masked from 'package:BiocGenerics':
##
##     combine, intersect, setdiff, union
## The following objects are masked from 'package:data.table':
##
##     between, first, last
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library("RColorBrewer")
```

****

O. THE SAMPLES that we are working on

```
We are working with PAIRED-END RNA-seq data; the sequencing has been done at GENEWIZ.

The files are located in the following folder : /labs/jlgoldbe/evan_RNAseq_aug2018

Aph-1_R1_001.fastq
Aph-1_R2_001.fastq
Aph-2_R1_001.fastq
Aph-2_R2_001.fastq
Aph-3_R1_001.fastq
Aph-3_R2_001.fastq

Aph-KH7-1_R1_001.fastq
Aph-KH7-1_R2_001.fastq
Aph-KH7-2_R1_001.fastq
Aph-KH7-2_R2_001.fastq
Aph-KH7-3_R1_001.fastq
Aph-KH7-3_R2_001.fastq

DMSO-1-lane1_R1_001.fastq
DMSO-1-lane1_R2_001.fastq
DMSO-1-lane2_R1_001.fastq
DMSO-1-lane2_R2_001.fastq
DMSO-2-lane1_R1_001.fastq
DMSO-2-lane1_R2_001.fastq
DMSO-2-lane2_R1_001.fastq
DMSO-2-lane2_R2_001.fastq
DMSO-3-lane1_R1_001.fastq
DMSO-3-lane1_R2_001.fastq
DMSO-3-lane2_R1_001.fastq
DMSO-3-lane2_R2_001.fastq

KH7-1_R1_001.fastq
KH7-1_R2_001.fastq
KH7-2_R1_001.fastq
KH7-2_R2_001.fastq
KH7-3_R1_001.fastq
KH7-3_R2_001.fastq

Noc-1_R1_001.fastq
Noc-1_R2_001.fastq
Noc-2_R1_001.fastq
Noc-2_R2_001.fastq
Noc-3_R1_001.fastq
Noc-3_R2_001.fastq

md5sum_list.txt
```

****

I. SEQUENCE ALIGNMENT to HG38 GENOME by using STAR aligner

1. THE QUALITY of the RNA-seq DATA

```
Here checking the QUALITY of the RNA-seq data by using FASTQC : FASTQ file for the READ1 :

#!/bin/bash

module load fastqc/0.11.2

## we reads a FASTQ file
## we make a folder for the FASTQC results

## an example is :

## FILE="/labs/jlgoldbe/evan_RNAseq_aug2018/KH7-1_R1_001.fastq"
## FASTQ="KH7-1_R1_001.fastq"

FILE="/labs/jlgoldbe/evan_RNAseq_aug2018/Aph-1_R1_001.fastq"
FASTQ="Aph-1_R1_001.fastq"

mkdir "${FASTQ}.report.fastqc"

fastqc -t 12 \
-o "${FASTQ}.report.fastqc" \
$FILE
```

```
Here checking the QUALITY of the RNA-seq data by using FASTQC : FASTQ file for the READ2 :

#!/bin/bash

module load fastqc/0.11.2

## we reads a FASTQ file
## we make a folder for the FASTQC results

## an example is :

## FILE="/labs/jlgoldbe/evan_RNAseq_aug2018/KH7-1_R1_001.fastq"
## FASTQ="KH7-1_R1_001.fastq"

FILE="/labs/jlgoldbe/evan_RNAseq_aug2018/Aph-1_R2_001.fastq"
FASTQ="Aph-1_R2_001.fastq"

mkdir "${FASTQ}.report.fastqc"

fastqc -t 12 \
-o "${FASTQ}.report.fastqc" \
$FILE
```

****

2. TRIMMING the ADAPTORS by using TRIMMOMATIC

```
Here trimming the adaptors by using TRIMMOMATIC :

#!/bin/bash

module load fastqc/0.11.2
module load trim_galore/0.4.5
module load cutadapt/1.8.1
module load trimmomatic/0.36

##################################################################

TRIMMOMATIC="/labs/jlgoldbe/btanasa/software/Trimmomatic-0.38"

EVAN_DATA="/labs/jlgoldbe/evan_RNAseq_aug2018/"

##################################################################

## an example is shown below :

## INPUT1="DMSO-1-lane1_R1_001.fastq"
## INPUT2="DMSO-1-lane1_R2_001.fastq"
## OUTPUT="DMSO-1-lane1.fastq.gz"

## in the OUTPUT file name : SHORT_NAME + fastq.gz

###################################################################

INPUT1="Aph-1_R1_001.fastq"
INPUT2="Aph-1_R2_001.fastq"
OUTPUT="Aph-1.fastq.gz"

###################################################################

java -jar $TRIMMOMATIC/trimmomatic-0.38.jar PE \
-threads 12 \
-validatePairs \
$EVAN_DATA$INPUT1 \
$EVAN_DATA$INPUT2 \
-baseout $OUTPUT \
-summary "${OUTPUT}.summary" \
ILLUMINACLIP:$TRIMMOMATIC/adapters/TruSeq_adapters_GENEWIZ.txt:2:30:10 \
SLIDINGWINDOW:4:15  \
LEADING:6 \
TRAILING:4 \
MINLEN:36
```

```
After trimming the adaptors by using TRIMMOMATIC, the results are listed in the files ".summary" :

        input_read_pairs    surviving_read_percent  dropped_read_percent

Aph-1.fastq.gz.summary  49595469    98.18   0.13
Aph-2.fastq.gz.summary  57625833    98.31   0.11
Aph-3.fastq.gz.summary  68651545    98.41   0.11

Aph-KH7-1.fastq.gz.summary  60256714    98.61   0.11
Aph-KH7-2.fastq.gz.summary  64961842    98.46   0.14
Aph-KH7-3.fastq.gz.summary  59076760    98.43   0.12

DMSO-1-lane1.fastq.gz.summary   57344343    98.62   0.11
DMSO-1-lane2.fastq.gz.summary   54329985    98.29   0.12
DMSO-2-lane1.fastq.gz.summary   53922668    98.59   0.14
DMSO-2-lane2.fastq.gz.summary   50703732    98.39   0.14
DMSO-3-lane1.fastq.gz.summary   57424274    98.37   0.13
DMSO-3-lane2.fastq.gz.summary   54223979    98.05   0.14

KH7-1.fastq.gz.summary  56630753    98.52   0.13
KH7-2.fastq.gz.summary  52335900    98.29   0.13
KH7-3.fastq.gz.summary  52666519    98.47   0.1

Noc-1.fastq.gz.summary  59799359    98.05   0.13
Noc-2.fastq.gz.summary  54333273    98.23   0.1
Noc-3.fastq.gz.summary  55772753    98.44   0.12
```

****

3. DOING the SEQUENCE ALIGNMENT by using STAR ALIGNER

```
#!/bin/bash

STAR="/labs/jlgoldbe/btanasa/software/STAR_2.6.0a/bin/Linux_x86_64/STAR"
HG38="/labs/jlgoldbe/btanasa/genomes_STAR_from_UCSF/hg38_genome_simple_index_STAR"
GENES="/labs/jlgoldbe/btanasa/genes_GENCODE/gencode.v28.basic.annotation.gtf"

IN="/labs/jlgoldbe/evan_RNAseq_aug2018"
OUT="/labs/jlgoldbe/evan_RNAseq_aug2018_results"

$STAR \
--runMode alignReads \
--runThreadN  12 \
--genomeDir $HG38 \
--sjdbGTFfile $GENES \
--sjdbOverhang 99 \
--quantMode TranscriptomeSAM \
--outSAMtype BAM SortedByCoordinate \
--outSAMorder paired \
--outWigType wiggle \
--outWigStrand Unstranded \
--outWigNorm RPM \
--limitBAMsortRAM 32000000000 \
--chimSegmentMin 20 \
--outFilterType BySJout \
--outFilterMultimapNmax 20 \
--alignSJoverhangMin 8 \
--alignSJDBoverhangMin 1 \
--outSAMattributes All \
--outFilterMismatchNmax 999 \
--outFilterMismatchNoverLmax 0.04 \
--alignIntronMin 20 \
--alignIntronMax 1000000 \
--alignMatesGapMax 1000000 \
--readFilesIn $IN/Aph-3_R1_001.fastq $IN/Aph-3_R2_001.fastq \
--outFileNamePrefix $OUT/Aph3/
```

****

II. READ COUNTING with RSEM on GENCODE GENES of hg38 GENOME

```
The script was run from each folder for each SAMPLE that contains the ALIGNMENTS :

#!/bin/bash

RSEM="/labs/jlgoldbe/btanasa/software/RSEM_1.3.1_bin_stanford/bin"
STAR="/labs/jlgoldbe/btanasa/software/STAR_2.6.0a/bin/Linux_x86_64/STAR"
HG38="/labs/jlgoldbe/btanasa/genomes_STAR_from_UCSF/hg38_genome_simple_index_STAR"
GENES="/labs/jlgoldbe/btanasa/genes_GENCODE/gencode.v28.basic.annotation.gtf"
HG38_FASTA="/labs/jlgoldbe/btanasa/genomes_STAR_from_UCSF/hg38_genome_from_Marcus/hg38_genome.fa"
HG38_RSEM="/labs/jlgoldbe/btanasa/genomes_STAR_from_UCSF/hg38_genome_from_Marcus_by_RSEM/hg38_genome"

module load rsem/1.2.30
module load samtools/1.9
mkdir rsem

$RSEM/rsem-calculate-expression --bam --no-bam-output -p 12 --paired-end --forward-prob 0.5 \
Aligned.toTranscriptome.out.bam \
$HG38_RSEM \
./rsem >& \
./rsem/rsem.log
```

```
In each folder, we have the following files (that will have to be renamed depending on the name of the

Aligned.sortedByCoord.out.bam
Aligned.toTranscriptome.out.bam

rsem.genes.results
rsem.isoforms.results

Also, we have to define a header for the WIG files, and an example is shown below :

track type=wiggle_0 name="Aph1" description="Aph1" visibility=full autoScale=off
viewLimits=0.0:25.0 color=255,0,0 yLineMark=11.76 yLineOnOff=on priority=10
```

****

III. INTEGRATING all the FILES that we have obtained with RSEM :

```
Here we are integrating the files from RSEM that contain the gene expression data with the GENCODE genes

sample.Aph1.rsem.genes.results
sample.Aph2.rsem.genes.results
sample.Aph3.rsem.genes.results

sample.Aph_KH7_1.rsem.genes.results
sample.Aph_KH7_2.rsem.genes.results
sample.Aph_KH7_3.rsem.genes.results

sample.DMSO1_lane1.rsem.genes.results
sample.DMSO1_lane2.rsem.genes.results
sample.DMSO2_lane1.rsem.genes.results
sample.DMSO2_lane2.rsem.genes.results
sample.DMSO3_lane1.rsem.genes.results
sample.DMSO3_lane2.rsem.genes.results

sample.KH7_1.rsem.genes.results
sample.KH7_2.rsem.genes.results
sample.KH7_3.rsem.genes.results
sample.Noc_1.rsem.genes.results
sample.Noc_2.rsem.genes.results
sample.Noc_3.rsem.genes.results

the_GENES.58381_genes.gencode.v28.basic.annotation.28aug2018.txt

The files from RSEM contain the following information :

**COUNTS**
**TPM**
**FPKM**
```

****

```
############################################################################################
############################################################################################
###### reading the files with the GENE EXPRESSION COUNTS :

genes <- read.delim("the_GENES.58381_genes.gencode.v28.basic.annotation.28aug2018.txt",
                     sep="\t", header=T, stringsAsFactors=F)

# head(genes)
dim(genes)

genes.dt <- as.data.table(genes)

# head(genes.dt)
dim(genes.dt)

###### to integrate these files : reading the files and changing the names of the columns

name <- "the_GENES.58381_genes.gencode.v28.basic.annotation.28aug2018.txt"

############################################################################################
############################################################################################
Aph1 <- read.delim("sample.Aph1.rsem.genes.results", sep="\t",
                                                     header=T, stringsAsFactors=F)

Aph1.simple <- data.frame( Aph1.gene =  Aph1$gene_id,
                           Aph1.count = Aph1$expected_count,
                           Aph1.TPM =   Aph1$TPM,
                           Aph1.FPKM =  Aph1$FPKM,
                           stringsAsFactors=F)

# head(Aph1)
dim(Aph1)

# head(Aph1.simple)
dim(Aph1.simple)

############################################################################################
############################################################################################
Aph2 <- read.delim("sample.Aph2.rsem.genes.results", sep="\t",
                                                     header=T, stringsAsFactors=F)

Aph2.simple <- data.frame( Aph2.gene =  Aph2$gene_id,
                           Aph2.count = Aph2$expected_count,
                           Aph2.TPM =   Aph2$TPM,
                           Aph2.FPKM =  Aph2$FPKM,
                           stringsAsFactors=F)

# head(Aph2)
dim(Aph2)

# head(Aph2.simple)
dim(Aph2.simple)
```

```
############################################################################################
############################################################################################
Aph3 <- read.delim("sample.Aph3.rsem.genes.results", sep="\t",
                                                     header=T, stringsAsFactors=F)

Aph3.simple <- data.frame( Aph3.gene =  Aph3$gene_id,
                           Aph3.count = Aph3$expected_count,
                           Aph3.TPM =   Aph3$TPM,
                           Aph3.FPKM =  Aph3$FPKM,
                           stringsAsFactors=F)

# head(Aph3)
dim(Aph3)

# head(Aph3.simple)
dim(Aph3.simple)

############################################################################################
############################################################################################
Aph_KH7_1 <- read.delim("sample.Aph_KH7_1.rsem.genes.results", sep="\t",
                                                              header=T, stringsAsFactors=F)

Aph_KH7_1.simple <- data.frame( Aph_KH7_1.gene =  Aph_KH7_1$gene_id,
                                Aph_KH7_1.count = Aph_KH7_1$expected_count,
                                Aph_KH7_1.TPM =   Aph_KH7_1$TPM,
                                Aph_KH7_1.FPKM =  Aph_KH7_1$FPKM,
                                stringsAsFactors=F)

# head(Aph_KH7_1)
dim(Aph_KH7_1)

# head(Aph_KH7_1.simple)
dim(Aph_KH7_1.simple)

############################################################################################
############################################################################################
Aph_KH7_2 <- read.delim("sample.Aph_KH7_2.rsem.genes.results", sep="\t",
                                                              header=T, stringsAsFactors=F)

Aph_KH7_2.simple <- data.frame( Aph_KH7_2.gene =  Aph_KH7_2$gene_id,
                                Aph_KH7_2.count = Aph_KH7_2$expected_count,
                                Aph_KH7_2.TPM =   Aph_KH7_2$TPM,
                                Aph_KH7_2.FPKM =  Aph_KH7_2$FPKM,
                                stringsAsFactors=F)

# head(Aph_KH7_2)
dim(Aph_KH7_2)

# head(Aph_KH7_2.simple)
dim(Aph_KH7_2.simple)

############################################################################################
############################################################################################
```

```
Aph_KH7_3 <- read.delim("sample.Aph_KH7_3.rsem.genes.results", sep="\t",
                                                              header=T, stringsAsFactors=F)

Aph_KH7_3.simple <- data.frame( Aph_KH7_3.gene =  Aph_KH7_3$gene_id,
                                Aph_KH7_3.count = Aph_KH7_3$expected_count,
                                Aph_KH7_3.TPM =   Aph_KH7_3$TPM,
                                Aph_KH7_3.FPKM =  Aph_KH7_3$FPKM,
                                stringsAsFactors=F)

# head(Aph_KH7_3)
dim(Aph_KH7_3)

# head(Aph_KH7_3.simple)
dim(Aph_KH7_3.simple)

############################################################################################
############################################################################################
DMSO1_lane1 <- read.delim("sample.DMSO1_lane1.rsem.genes.results", sep="\t",
                                                               header=T, stringsAsFactors=F)

DMSO1_lane1.simple <- data.frame( DMSO1_lane1.gene =  DMSO1_lane1$gene_id,
                                  DMSO1_lane1.count = DMSO1_lane1$expected_count,
                                  DMSO1_lane1.TPM =   DMSO1_lane1$TPM,
                                  DMSO1_lane1.FPKM =  DMSO1_lane1$FPKM,
                                  stringsAsFactors=F)

# head(DMSO1_lane1)
dim(DMSO1_lane1)

# head(DMSO1_lane1.simple)
dim(DMSO1_lane1.simple)

############################################################################################
############################################################################################
DMSO1_lane2 <- read.delim("sample.DMSO1_lane2.rsem.genes.results", sep="\t",
                                                              header=T, stringsAsFactors=F)

DMSO1_lane2.simple <- data.frame( DMSO1_lane2.gene =  DMSO1_lane2$gene_id,
                                  DMSO1_lane2.count = DMSO1_lane2$expected_count,
                                  DMSO1_lane2.TPM =   DMSO1_lane2$TPM,
                                  DMSO1_lane2.FPKM =  DMSO1_lane2$FPKM,
                                  stringsAsFactors=F)

# head(DMSO1_lane2)
dim(DMSO1_lane2)

# head(DMSO1_lane2.simple)
dim(DMSO1_lane2.simple)

############################################################################################
############################################################################################
DMSO2_lane1 <- read.delim("sample.DMSO2_lane1.rsem.genes.results", sep="\t",
                                                               header=T, stringsAsFactors=F)
```

```
DMSO2_lane1.simple <- data.frame( DMSO2_lane1.gene =  DMSO2_lane1$gene_id,
                                  DMSO2_lane1.count = DMSO2_lane1$expected_count,
                                  DMSO2_lane1.TPM =   DMSO2_lane1$TPM,
                                  DMSO2_lane1.FPKM =  DMSO2_lane1$FPKM,
                                  stringsAsFactors=F)

# head(DMSO2_lane1)
dim(DMSO2_lane1)

# head(DMSO2_lane1.simple)
dim(DMSO2_lane1.simple)

############################################################################################
############################################################################################
DMSO2_lane2 <- read.delim("sample.DMSO2_lane2.rsem.genes.results", sep="\t",
                                                              header=T, stringsAsFactors=F)

DMSO2_lane2.simple <- data.frame( DMSO2_lane2.gene =  DMSO2_lane2$gene_id,
                                  DMSO2_lane2.count = DMSO2_lane2$expected_count,
                                  DMSO2_lane2.TPM =   DMSO2_lane2$TPM,
                                  DMSO2_lane2.FPKM =  DMSO2_lane2$FPKM,
                                  stringsAsFactors=F)

# head(DMSO2_lane2)
dim(DMSO2_lane2)

# head(DMSO2_lane2.simple)
dim(DMSO2_lane2.simple)

############################################################################################
############################################################################################
DMSO3_lane1 <- read.delim("sample.DMSO3_lane1.rsem.genes.results", sep="\t",
                                                              header=T, stringsAsFactors=F)

DMSO3_lane1.simple <- data.frame( DMSO3_lane1.gene =  DMSO3_lane1$gene_id,
                                  DMSO3_lane1.count = DMSO3_lane1$expected_count,
                                  DMSO3_lane1.TPM =   DMSO3_lane1$TPM,
                                  DMSO3_lane1.FPKM =  DMSO3_lane1$FPKM,
                                  stringsAsFactors=F)

# head(DMSO3_lane1)
dim(DMSO3_lane1)

# head(DMSO3_lane1.simple)
dim(DMSO3_lane1.simple)

############################################################################################
############################################################################################
DMSO3_lane2 <- read.delim("sample.DMSO3_lane2.rsem.genes.results", sep="\t",
                                                               header=T, stringsAsFactors=F)

DMSO3_lane2.simple <- data.frame( DMSO3_lane2.gene =  DMSO3_lane2$gene_id,
                                  DMSO3_lane2.count = DMSO3_lane2$expected_count,
                                  DMSO3_lane2.TPM =   DMSO3_lane2$TPM,
                                  DMSO3_lane2.FPKM =  DMSO3_lane2$FPKM,
                                  stringsAsFactors=F)

# head(DMSO3_lane2)
dim(DMSO3_lane2)

# head(DMSO3_lane2.simple)
dim(DMSO3_lane2.simple)

############################################################################################
############################################################################################
KH7_1 <- read.delim("sample.KH7_1.rsem.genes.results", sep="\t",
                                                               header=T, stringsAsFactors=F)

KH7_1.simple <- data.frame( KH7_1.gene =  KH7_1$gene_id,
                            KH7_1.count = KH7_1$expected_count,
                            KH7_1.TPM =   KH7_1$TPM,
                            KH7_1.FPKM =  KH7_1$FPKM,
                                stringsAsFactors=F)

# head(KH7_1)
dim(KH7_1)

# head(KH7_1.simple)
dim(KH7_1.simple)

############################################################################################
############################################################################################
KH7_2 <- read.delim("sample.KH7_2.rsem.genes.results", sep="\t",
                                                              header=T, stringsAsFactors=F)

KH7_2.simple <- data.frame( KH7_2.gene =  KH7_2$gene_id,
                            KH7_2.count = KH7_2$expected_count,
                            KH7_2.TPM =   KH7_2$TPM,
                            KH7_2.FPKM =  KH7_2$FPKM,
                                stringsAsFactors=F)

# head(KH7_2)
dim(KH7_2)

# head(KH7_2.simple)
dim(KH7_2.simple)

############################################################################################
############################################################################################
KH7_3 <- read.delim("sample.KH7_3.rsem.genes.results", sep="\t",
                                                               header=T, stringsAsFactors=F)

KH7_3.simple <- data.frame( KH7_3.gene =  KH7_3$gene_id,
                            KH7_3.count = KH7_3$expected_count,
                            KH7_3.TPM =   KH7_3$TPM,
                            KH7_3.FPKM =  KH7_3$FPKM,
                                stringsAsFactors=F)

# head(KH7_3)
dim(KH7_3)

# head(KH7_3.simple)
dim(KH7_3.simple)

############################################################################################
############################################################################################
Noc_1 <- read.delim("sample.Noc_1.rsem.genes.results", sep="\t",
                                                               header=T, stringsAsFactors=F)

Noc_1.simple <- data.frame( Noc_1.gene =  Noc_1$gene_id,
                            Noc_1.count = Noc_1$expected_count,
                            Noc_1.TPM =   Noc_1$TPM,
                            Noc_1.FPKM =  Noc_1$FPKM,
                                stringsAsFactors=F)

# head(Noc_1)
dim(Noc_1)

# head(Noc_1.simple)
dim(Noc_1.simple)

############################################################################################
############################################################################################
Noc_2 <- read.delim("sample.Noc_2.rsem.genes.results", sep="\t",
                                                              header=T, stringsAsFactors=F)

Noc_2.simple <- data.frame( Noc_2.gene =  Noc_2$gene_id,
                            Noc_2.count = Noc_2$expected_count,
                            Noc_2.TPM =   Noc_2$TPM,
                            Noc_2.FPKM =  Noc_2$FPKM,
                                stringsAsFactors=F)

# head(Noc_2)
dim(Noc_2)

# head(Noc_2.simple)
dim(Noc_2.simple)

############################################################################################
############################################################################################
Noc_3 <- read.delim("sample.Noc_3.rsem.genes.results", sep="\t",
                                                              header=T, stringsAsFactors=F)

Noc_3.simple <- data.frame( Noc_3.gene =  Noc_3$gene_id,
                            Noc_3.count = Noc_3$expected_count,
                            Noc_3.TPM =   Noc_3$TPM,
                            Noc_3.FPKM =  Noc_3$FPKM,
                                stringsAsFactors=F)
```

```
# head(Noc_3)
dim(Noc_3)

# head(Noc_3.simple)
dim(Noc_3.simple)

#############################################################################################
#############################################################################################
#############################################################################################
#############################################################################################
```

****

```
### now integrating these data; we can make DATA TABLES :

Aph1.simple.dt <- as.data.table(Aph1.simple)
Aph2.simple.dt <- as.data.table(Aph2.simple)
Aph3.simple.dt <- as.data.table(Aph3.simple)

Aph_KH7_1.simple.dt <- as.data.table(Aph_KH7_1.simple)
Aph_KH7_2.simple.dt <- as.data.table(Aph_KH7_2.simple)
Aph_KH7_3.simple.dt <- as.data.table(Aph_KH7_3.simple)

DMSO1_lane1.simple.dt <- as.data.table(DMSO1_lane1.simple)
DMSO1_lane2.simple.dt <- as.data.table(DMSO1_lane2.simple)

DMSO2_lane1.simple.dt <- as.data.table(DMSO2_lane1.simple)
DMSO2_lane2.simple.dt <- as.data.table(DMSO2_lane2.simple)

DMSO3_lane1.simple.dt <- as.data.table(DMSO3_lane1.simple)
DMSO3_lane2.simple.dt <- as.data.table(DMSO3_lane2.simple)

KH7_1.simple.dt <- as.data.table(KH7_1.simple)
KH7_2.simple.dt <- as.data.table(KH7_2.simple)
KH7_3.simple.dt <- as.data.table(KH7_3.simple)

Noc_1.simple.dt <- as.data.table(Noc_1.simple)
Noc_2.simple.dt <- as.data.table(Noc_2.simple)
Noc_3.simple.dt <- as.data.table(Noc_3.simple)

############################################################################################
############################################################################################

library(data.table)

setkeyv(genes.dt, c('GENE_ID'))

setkeyv(Aph1.simple.dt, c('Aph1.gene'))
setkeyv(Aph2.simple.dt, c('Aph2.gene'))
setkeyv(Aph3.simple.dt, c('Aph3.gene'))

setkeyv(Aph_KH7_1.simple.dt, c('Aph_KH7_1.gene'))
setkeyv(Aph_KH7_2.simple.dt, c('Aph_KH7_2.gene'))
setkeyv(Aph_KH7_3.simple.dt, c('Aph_KH7_3.gene'))

setkeyv(DMSO1_lane1.simple.dt, c('DMSO1_lane1.gene'))
setkeyv(DMSO1_lane2.simple.dt, c('DMSO1_lane2.gene'))

setkeyv(DMSO2_lane1.simple.dt, c('DMSO2_lane1.gene'))
setkeyv(DMSO2_lane2.simple.dt, c('DMSO2_lane2.gene'))

setkeyv(DMSO3_lane1.simple.dt, c('DMSO3_lane1.gene'))
setkeyv(DMSO3_lane2.simple.dt, c('DMSO3_lane2.gene'))

setkeyv(KH7_1.simple.dt, c('KH7_1.gene'))
setkeyv(KH7_2.simple.dt, c('KH7_2.gene'))
setkeyv(KH7_3.simple.dt, c('KH7_3.gene'))

setkeyv(Noc_1.simple.dt, c('Noc_1.gene'))
setkeyv(Noc_2.simple.dt, c('Noc_2.gene'))
setkeyv(Noc_3.simple.dt, c('Noc_3.gene'))

####################### to integrate ALL the dataframes :

# expression.Aph123 <- genes.dt[Aph1.simple.dt,][Aph2.simple.dt,][Aph3.simple.dt,]

# expression.Aph_KH7_123 <- genes.dt[Aph_KH7_1.simple.dt,][Aph_KH7_2.simple.dt,][Aph_KH7_3.simple.dt,]

# expression.DMSO <- genes.dt[DMSO1_lane1.simple.dt,][DMSO1_lane2.simple.dt,][DMSO2_lane1.simple.dt,][D

# expression.KH7_123 <- genes.dt[KH7_1.simple.dt,][KH7_2.simple.dt,][KH7_3.simple.dt,]

# expression.Noc_123 <- genes.dt[Noc_1.simple.dt,][Noc_2.simple.dt,][Noc_3.simple.dt,]

expression.all.samples <- genes.dt[DMSO1_lane1.simple.dt,][DMSO1_lane2.simple.dt,][DMSO2_lane1.simple.dt

expression.all.samples
dim(expression.all.samples)

############################################################################################
############################################################################################
###################### to print the RESULTS, where we have integrated ALL the data frames :

name <- "the_GENES.58381_genes.gencode.v28.basic.annotation.28aug2018.txt"

write.table(expression.all.samples,
            file=paste(name, ".INTEGRATED.file.ALL.samples.txt", sep=""),
            sep="\t", quote=FALSE,
            row.names = FALSE, col.names = TRUE)

############################################################################################
############################################################################################
############################################################################################
############################################################################################
```

****

OLD_ANALYSIS. DIFFERENTIAL EXPRESSION with edgeR (an OLD EXAMPLE):

```
### Here it is a very OLD PIECE of R CODE that we have used in the past for CSC and non-CSC data.

eset <- read.delim("mm10.expression.NOADJ.for-samples-DMSO-G9ai.only-samples-245.v.to-use.txt",
                   row.names="Symbol")

group <- factor(c("G9ai","G9ai","G9ai","DMSO","DMSO","DMSO"))
group <- relevel(group,ref="DMSO")

subject <- factor(c(1,2,3,1,2,3))
design <- model.matrix(~group+subject)

y <- DGEList(counts=eset,group=group)

keep <- rowSums(cpm(y) > 0.5) >= 6

y <- y[keep,,keep.lib.sizes=FALSE]
y <- calcNormFactors(y)

logCPM <- cpm(y,log=TRUE,prior.count=3)
fit <- lmFit(logCPM, design)
fit <- eBayes(fit,trend=TRUE, robust=TRUE)

pdf("mm10.expression.NOADJ.for-samples-DMSO-G9ai.only-samples-245.v.to-use.txt.SA.fit.with.edgeR.pdf")
plotSA(fit)
dev.off()

results_edgeR <- topTable(fit, coef=2, adjust="fdr", number=Inf)

write.table(results_edgeR, file="mm10.expression.NOADJ.for-samples-DMSO-G9ai.only-samples-245.v.to-use.
            sep="\t", eol="\n", row.names=TRUE, col.names=TRUE)
```

****

OLD_ANALYSIS. DIFFERENTIAL EXPRESSION with LIMMA (an OLD EXAMPLE):

```
### Here it is a very OLD PIECE of R CODE that we have used in the past for CSC and non-CSC data.

### reading the expression dataset

eset <- read.delim("mm10.expression.NOADJ.for-samples-DMSO-G9ai.only-samples-245.v.to-use.txt",
                   row.names="Symbol")

##########################################

### setting up the groups and the subjects

group <- factor(c("G9ai","G9ai","G9ai","DMSO","DMSO","DMSO"))
subject <- factor(c(1,2,3,1,2,3))

### setting up the design and the contrast matrix

design <- model.matrix(~0+group+subject)
contrast.matrix <- makeContrasts(groupG9ai-groupDMSO, levels=design)

### filtering the genes based on CPM :

y <- DGEList(counts=eset,group=group)

### keep <- rowSums(cpm(y, lib.size=libsize)>1) >= 3

keep <- rowSums(cpm(y)>0.5) >= 6
y <- y[keep,]

y$samples$lib.size <- colSums(y$counts)

### computing the normalization factors :

y <- calcNormFactors(y)

### using the VOOM transformation :

v <- voom(y,design,plot=FALSE)

pdf("mm10.expression.NOADJ.for-samples-DMSO-G9ai.only-samples-245.v.to-use.txt.limma.with.mean-variance
v <- voom(y,design,plot=TRUE)
dev.off()

### doing the LINEAR FIT in LIMMA :

fit <- lmFit(v, design)

fit2 <- contrasts.fit(fit, contrast.matrix)
fit2 <- eBayes(fit2)

### obtaining and writing the results :
```

```r
results_limma <- topTable(fit2, coef=1, adjust="fdr", number=Inf)

write.table(results_limma, file="mm10.expression.NOADJ.for-samples-DMSO-G9ai.only-samples-245.v.to-use.t
                          sep="\t", eol="\n", row.names=TRUE, col.names=TRUE)
```

****

IV. DIFFERENTIAL EXPRESSION with LIMMA (the DATASET of SAC and CELL CYCLE inhibition) :

1. Reading the dataframe and preparing it for the step of DE analysis with LIMMA

```
###### reading the files with the GENE EXPRESSION COUNTS from the previous step (not running the previou

genes <- read.delim("the_GENES.58381_genes.gencode.v28.basic.annotation.28aug2018.txt.INTEGRATED.file.A
                     sep="\t", header=T, stringsAsFactors=F)

### head(genes)
dim(genes)
```

```
## [1] 58381    61
```

```
###### transforming the DATA FRAME into a DATA TABLE :

genes.dt <- as.data.table(genes)

### head(genes.dt)
dim(genes.dt)
```

```
## [1] 58381    61
```

```
############################################################################################
############################################################################################
############################################################################################
############################################################################################

###### in the next section, we are going to select the COLUMNS with COUNTS for DEG analysis :

# > colnames(genes)
# [1] "CHR"               "START"             "END"
# [4] "STRAND"            "GENE_ID"           "GENE_NAME"
# [7] "GENE_TYPE"         "DMSO1_lane1.count" "DMSO1_lane1.TPM"
#[10] "DMSO1_lane1.FPKM"  "DMSO1_lane2.count" "DMSO1_lane2.TPM"
#[13] "DMSO1_lane2.FPKM"  "DMSO2_lane1.count" "DMSO2_lane1.TPM"
#[16] "DMSO2_lane1.FPKM"  "DMSO2_lane2.count" "DMSO2_lane2.TPM"
#[19] "DMSO2_lane2.FPKM"  "DMSO3_lane1.count" "DMSO3_lane1.TPM"
#[22] "DMSO3_lane1.FPKM"  "DMSO3_lane2.count" "DMSO3_lane2.TPM"
#[25] "DMSO3_lane2.FPKM"  "Aph1.count"        "Aph1.TPM"
#[28] "Aph1.FPKM"         "Aph2.count"        "Aph2.TPM"
#[31] "Aph2.FPKM"         "Aph3.count"        "Aph3.TPM"
#[34] "Aph3.FPKM"         "Aph_KH7_1.count"   "Aph_KH7_1.TPM"
#[37] "Aph_KH7_1.FPKM"    "Aph_KH7_2.count"   "Aph_KH7_2.TPM"
#[40] "Aph_KH7_2.FPKM"    "Aph_KH7_3.count"   "Aph_KH7_3.TPM"
#[43] "Aph_KH7_3.FPKM"    "KH7_1.count"       "KH7_1.TPM"
#[46] "KH7_1.FPKM"        "KH7_2.count"       "KH7_2.TPM"
#[49] "KH7_2.FPKM"        "KH7_3.count"       "KH7_3.TPM"
#[52] "KH7_3.FPKM"        "Noc_1.count"       "Noc_1.TPM"
#[55] "Noc_1.FPKM"        "Noc_2.count"       "Noc_2.TPM"
#[58] "Noc_2.FPKM"        "Noc_3.count"       "Noc_3.TPM"
#[61] "Noc_3.FPKM"

################ here we would have to make a special ROWNAME,
```

```r
### as some genes are present in multiple isoforms ..

genes$ID <- rownames(genes)
genes$GENE_NAME_ID <- paste(genes$GENE_NAME,
                            genes$ID, sep=":")

### head(genes)
dim(genes)
```

```
## [1] 58381    63
```

```r
############################################################################################
############################################################################################
############################################################################################
############################################################################################
################# making a DATAFRAME of GENES COUNTS :

genes.counts <- subset(genes, select=c("GENE_NAME_ID",
                       "DMSO1_lane1.count", "DMSO1_lane2.count",
                       "DMSO2_lane1.count", "DMSO2_lane2.count",
                       "DMSO3_lane1.count", "DMSO3_lane2.count",
                       "Aph1.count", "Aph2.count", "Aph3.count",
                       "Aph_KH7_1.count","Aph_KH7_2.count","Aph_KH7_3.count",
                       "KH7_1.count", "KH7_2.count", "KH7_3.count",
                       "Noc_1.count", "Noc_2.count", "Noc_3.count" ))

rownames(genes.counts) <- genes.counts$GENE_NAME_ID
genes.counts <- genes.counts[,-1]

### head(genes.counts)
dim(genes.counts)
```

```
## [1] 58381    18
```

```r
############################################################################
################# making a DATAFRAME based on TPM :

genes.tpm <- subset(genes, select=c("GENE_NAME_ID",
                       "DMSO1_lane1.TPM", "DMSO1_lane2.TPM",
                       "DMSO2_lane1.TPM", "DMSO2_lane2.TPM",
                       "DMSO3_lane1.TPM", "DMSO3_lane2.TPM",
                       "Aph1.TPM", "Aph2.TPM", "Aph3.TPM",
                       "Aph_KH7_1.TPM","Aph_KH7_2.TPM","Aph_KH7_3.TPM",
                       "KH7_1.TPM", "KH7_2.TPM", "KH7_3.TPM",
                       "Noc_1.TPM", "Noc_2.TPM", "Noc_3.TPM" ))

rownames(genes.tpm) <- genes.tpm$GENE_NAME_ID
genes.tpm <- genes.tpm[,-1]

### head(genes.tpm)
dim(genes.tpm)
```

```
## [1] 58381    18
```

```r
############################################################################
################# making a DATAFRAME based on FPKM :
```

```r
genes.fpkm <- subset(genes, select=c("GENE_NAME_ID",
                       "DMSO1_lane1.FPKM", "DMSO1_lane2.FPKM",
                       "DMSO2_lane1.FPKM", "DMSO2_lane2.FPKM",
                       "DMSO3_lane1.FPKM", "DMSO3_lane2.FPKM",
                       "Aph1.FPKM", "Aph2.FPKM", "Aph3.FPKM",
                       "Aph_KH7_1.FPKM","Aph_KH7_2.FPKM","Aph_KH7_3.FPKM",
                       "KH7_1.FPKM", "KH7_2.FPKM", "KH7_3.FPKM",
                       "Noc_1.FPKM", "Noc_2.FPKM", "Noc_3.FPKM" ))

rownames(genes.fpkm) <- genes.fpkm$GENE_NAME_ID
genes.fpkm <- genes.fpkm[,-1]

### head(genes.fpkm)
dim(genes.fpkm)
```

```
## [1] 58381    18
```

```r
############################################################################################
############################################################################################
############################################################################################
############################################################################################
```

```
*****************************************
```

```
#### continuing to work with the DATAFRAME containing the COUNTS : genes.counts
#### in order to assess the DIFFERENTIAL EXPRESSION

### head(genes.counts)
dim(genes.counts)
```

```
## [1] 58381    18
```

```
############################################################################################
############################################################################################
################## and SUBSETING by SPECIFIC SAMPLES :

genes.counts.Aph <- subset(genes.counts, select=c(
                           "DMSO1_lane1.count",
                           "DMSO2_lane1.count",
                           "DMSO3_lane1.count",
                           "Aph1.count", "Aph2.count", "Aph3.count" ))

dim(genes.counts.Aph)
```

```
## [1] 58381     6
```

```
### head(genes.counts.Aph)

###############################################################################
###############################################################################

genes.counts.Aph_KH7 <- subset(genes.counts, select=c(
                               "DMSO1_lane1.count",
                               "DMSO2_lane1.count",
                               "DMSO3_lane1.count",
                               "Aph_KH7_1.count", "Aph_KH7_2.count", "Aph_KH7_3.count" ))
dim(genes.counts.Aph_KH7)
```

```
## [1] 58381     6
```

```
### head(genes.counts.Aph_KH7)

###############################################################################
###############################################################################

genes.counts.KH7 <- subset(genes.counts, select=c(
                           "DMSO1_lane1.count",
                           "DMSO2_lane1.count",
                           "DMSO3_lane1.count",
                           "KH7_1.count", "KH7_2.count", "KH7_3.count" ))
dim(genes.counts.KH7)
```

```
## [1] 58381     6
```

```
### head(genes.counts.KH7)

###############################################################################
###############################################################################
```

```
genes.counts.Noc <- subset(genes.counts, select=c(
                           "DMSO1_lane1.count",
                           "DMSO2_lane1.count",
                           "DMSO3_lane1.count",
                           "Noc_1.count", "Noc_2.count", "Noc_3.count" ))
dim(genes.counts.Noc)
```

```
## [1] 58381     6
```

```
### head(genes.counts.Noc)

############################################################################################
############################################################################################
############################################################################################
############################################################################################
############################################################################################

#### STARTING TO ASSESS THE DIFFERENTIAL EXPRESSION : using LIMMA :

############################################################################################
############################################################################################
############################################################################################
############################################################################################

#### using LIMMA for each individual MATRIX :

#### genes.counts.Aph
#### genes.counts.Aph_KH7
#### genes.counts.KH7
#### genes.counts.Noc

############################################################################################
############################################################################################
############################################################################################
############################################################################################

#### having a model in an OLD PIECE of CODE :

#### setting up the groups and the subjects
# group <- factor(c("csc","csc","csc","csc","csc","non","non","non","non","non"))
# subject <- factor(c(1,2,3,4,5,1,2,3,4,5))

#### setting up the design and the contrast matrix
# design <- model.matrix(~0+group+subject)
# contrast.matrix <- makeContrasts(groupcsc-groupnon, levels=design)

############################################################################################
############################################################################################
############################################################################################
############################################################################################
```

****

2. Performing the DEG analysis : DMSO vs Aph

```
eset <- genes.counts.Aph
eset_name <- deparse(substitute(genes.counts.Aph)) ### in order to get the name of the DF

#### genes.counts.Aph

group <- factor(c("DMSO", "DMSO", "DMSO", "Aph", "Aph","Aph"))
subject <- factor(c(1,2,3, 1,2,3))

#### setting up the design and the contrast matrix

design <- model.matrix(~0+group+subject)
contrast.matrix <- makeContrasts(groupAph-groupDMSO, levels=design)

design
```

```
##   groupAph groupDMSO subject2 subject3
## 1        0         1        0        0
## 2        0         1        1        0
## 3        0         1        0        1
## 4        1         0        0        0
## 5        1         0        1        0
## 6        1         0        0        1
## attr(,"assign")
## [1] 1 1 2 2
## attr(,"contrasts")
## attr(,"contrasts")$group
## [1] "contr.treatment"
##
## attr(,"contrasts")$subject
## [1] "contr.treatment"
```

```
contrast.matrix
```

```
##              Contrasts
## Levels         groupAph - groupDMSO
##   groupAph                        1
##   groupDMSO                      -1
##   subject2                        0
##   subject3                        0
```

```
############################################################################################

### filtering the genes based on CPM :
y <- DGEList(counts=eset, group=group)

### keep <- rowSums(cpm(y, lib.size=libsize)>1) >= 3
keep <- rowSums( cpm(y) > 0.5) >= 6
y <- y[keep,]
y$samples$lib.size <- colSums(y$counts)

####################################################### we can use y$counts for PCA analysis

### computing the normalization factors :
```

```
y <- calcNormFactors(y)

### using the VOOM transformation :
v <- voom(y, design, plot=FALSE)

### the LINEAR FIT in LIMMA :
fit <- lmFit(v, design)
fit2 <- contrasts.fit(fit, contrast.matrix)
fit2 <- eBayes(fit2)

### obtaining and writing the results :
results_limma <- topTable(fit2, coef=1, adjust="fdr", number=Inf)

### adding the rownames as columns

results_limma$Gene <- rownames(results_limma)

### separating the names of the GENES into 1st_PART and NUMBER :

results_limma$GENE <- results_limma$Gene

results_limma.sep <- separate(data=results_limma, col=Gene, into = c("Gene", "ID"), sep = ":")

head(results_limma.sep)
```

```
##                   logFC  AveExpr         t      P.Value    adj.P.Val
## CDKN1A:5578   2.457753 9.372328  55.50985 3.289764e-16 4.541190e-12
## GDF15:6279    4.610037 4.785359  39.96728 1.872510e-14 1.292406e-10
## CDC20:4702   -1.976256 6.343352 -33.60487 1.569775e-13 7.223060e-10
## EPS8L2:14227  2.622750 5.302664  32.17467 2.672951e-13 9.224353e-10
## PLK1:11956   -2.092451 6.163241 -30.94403 4.306089e-13 1.188825e-09
## CCNB1:6830   -1.516280 6.837604 -29.26005 8.528710e-13 1.681862e-09
##                     B   Gene    ID         GENE
## CDKN1A:5578  27.48847 CDKN1A  5578  CDKN1A:5578
## GDF15:6279   22.18700  GDF15  6279   GDF15:6279
## CDC20:4702   21.51309  CDC20  4702   CDC20:4702
## EPS8L2:14227 20.80344 EPS8L2 14227 EPS8L2:14227
## PLK1:11956   20.54143   PLK1 11956   PLK1:11956
## CCNB1:6830   19.91992  CCNB1  6830   CCNB1:6830
```

```
dim(results_limma.sep)
```

```
## [1] 13804     9
```

```
### writing the results to a file :

write.table(results_limma.sep, file=paste("analysis.LIMMA.", eset_name, sep=""),
                               sep="\t",
                               quote=FALSE, eol="\n",
                               row.names=FALSE, col.names=TRUE)

### computing the number of DEG for FDR < 0.05 :

results_limma.deg <- results_limma.sep[results_limma.sep$adj.P.Val < 0.05,]
```

```
### head(results_limma.deg)
dim(results_limma.deg)
```

```
## [1] 5619    9
```

```
write.table(results_limma.deg, file=paste("analysis.LIMMA.", eset_name, ".only.DEG", sep=""),
                               sep="\t",
                               quote=FALSE, eol="\n",
                               row.names=FALSE, col.names=TRUE)


### computing the number of DEG for FDR < 0.05 and FC > 1.2  : UP-REGULATED GENES :

results_limma.deg.up <- results_limma.sep[(results_limma.sep$adj.P.Val < 0.05) &
                                          (results_limma.sep$logFC > log2(1.2) )  ,]

### head(results_limma.deg.up)
dim(results_limma.deg.up)
```

```
## [1] 2259    9
```

```
write.table(results_limma.deg.up, file=paste("analysis.LIMMA.", eset_name, ".only.DEG.and.UP", sep=""),
                                  sep="\t",
                                  quote=FALSE, eol="\n",
                                  row.names=FALSE, col.names=TRUE)

### computing the number of DEG for FDR < 0.05 and FC < -1.2 : DOWN-REGULATED GENES :

results_limma.deg.down <- results_limma.sep[(results_limma.sep$adj.P.Val < 0.05) &
                                            (results_limma.sep$logFC < -log2(1.2) )  ,]

### head(results_limma.deg.down)
dim(results_limma.deg.down)
```

```
## [1] 1844    9
```

```
write.table(results_limma.deg.down, file=paste("analysis.LIMMA.", eset_name, ".only.DEG.and.DOWN", sep=
                                    sep="\t",
                                    quote=FALSE, eol="\n",
                                    row.names=FALSE, col.names=TRUE)


############################################################################################
############################################################################################
### saving the results into another DATAFRAME to be used later :

eset <- genes.counts.Aph
eset_name <- deparse(substitute(genes.counts.Aph)) ### in order to get the name of the DF
genes.counts.Aph.results.limma <- results_limma.sep

### head(genes.counts.Aph.results.limma)
dim(genes.counts.Aph.results.limma)
```

```
## [1] 13804     9
```

```
genes.counts.Aph.results.limma.deg <- results_limma.deg
genes.counts.Aph.results.limma.deg.up <- results_limma.deg.up
genes.counts.Aph.results.limma.deg.down <- results_limma.deg.down

dim(genes.counts.Aph.results.limma.deg.up)
```

```
## [1] 2259    9
```

```
dim(genes.counts.Aph.results.limma.deg.down)
```

```
## [1] 1844    9
```

```
############################################################################################
############################################################################################
############################################################################################
############################################################################################
```

****

3. Performing the DEG analysis : DMSO vs Aph_KH7

```
eset <- genes.counts.Aph_KH7
eset_name <- deparse(substitute(genes.counts.Aph_KH7))

#### genes.counts.Aph_KH7

group <- factor(c("DMSO", "DMSO", "DMSO", "Aph_KH7", "Aph_KH7", "Aph_KH7"))
subject <- factor(c(1,2,3, 1,2,3))

### setting up the design and the contrast matrix

design <- model.matrix(~0+group+subject)
contrast.matrix <- makeContrasts(groupAph_KH7-groupDMSO, levels=design)

design
```

```
##   groupAph_KH7 groupDMSO subject2 subject3
## 1            0         1        0        0
## 2            0         1        1        0
## 3            0         1        0        1
## 4            1         0        0        0
## 5            1         0        1        0
## 6            1         0        0        1
## attr(,"assign")
## [1] 1 1 2 2
## attr(,"contrasts")
## attr(,"contrasts")$group
## [1] "contr.treatment"
##
## attr(,"contrasts")$subject
## [1] "contr.treatment"
```

```
contrast.matrix
```

```
##               Contrasts
## Levels          groupAph_KH7 - groupDMSO
##   groupAph_KH7                        1
##   groupDMSO                          -1
##   subject2                            0
##   subject3                            0
```

```
############################################################################################

### filtering the genes based on CPM :
y <- DGEList(counts=eset, group=group)

### keep <- rowSums(cpm(y, lib.size=libsize)>1) >= 3
keep <- rowSums( cpm(y) > 0.5) >= 6
y <- y[keep,]
y$samples$lib.size <- colSums(y$counts)

####################################################### we can use y$counts for PCA analysis

### computing the normalization factors :
```

```
y <- calcNormFactors(y)

### using the VOOM transformation :
v <- voom(y, design, plot=FALSE)

### the LINEAR FIT in LIMMA :
fit <- lmFit(v, design)
fit2 <- contrasts.fit(fit, contrast.matrix)
fit2 <- eBayes(fit2)

### obtaining and writing the results :
results_limma <- topTable(fit2, coef=1, adjust="fdr", number=Inf)

### adding the rownames as columns

results_limma$Gene <- rownames(results_limma)

### separating the names of the GENES into 1st_PART and NUMBER :

results_limma$GENE <- results_limma$Gene

results_limma.sep <- separate(data=results_limma, col=Gene, into = c("Gene", "ID"), sep = ":")

head(results_limma.sep)
```

```
##                logFC  AveExpr        t      P.Value    adj.P.Val        B
## CDKN1A:5578 3.428973 9.772611 74.76790 2.254575e-22 3.049087e-18 41.45484
## IGFBP3:8871 3.506961 7.489578 66.19848 1.679475e-21 8.152833e-18 39.35310
## P4HA1:5325  3.569850 7.324514 65.90194 1.808526e-21 8.152833e-18 39.24382
## PGK1:2636   2.707239 9.788683 60.94376 6.565632e-21 2.219840e-17 38.26289
## TFRC:1187   2.997478 8.011044 58.35578 1.342062e-20 3.630008e-17 37.49174
## PLOD2:9600  2.707661 8.364187 55.25961 3.293689e-20 7.423975e-17 36.65417
##               Gene   ID        GENE
## CDKN1A:5578 CDKN1A 5578 CDKN1A:5578
## IGFBP3:8871 IGFBP3 8871 IGFBP3:8871
## P4HA1:5325   P4HA1 5325  P4HA1:5325
## PGK1:2636     PGK1 2636   PGK1:2636
## TFRC:1187     TFRC 1187   TFRC:1187
## PLOD2:9600   PLOD2 9600  PLOD2:9600
```

```
dim(results_limma.sep)
```

```
## [1] 13524     9
```

```
### writing the results to a file :

write.table(results_limma.sep, file=paste("analysis.LIMMA.", eset_name, sep=""),
                               sep="\t",
                               quote=FALSE, eol="\n",
                               row.names=FALSE, col.names=TRUE)

### computing the number of DEG for FDR < 0.05 :

results_limma.deg <- results_limma.sep[results_limma.sep$adj.P.Val < 0.05,]
```

```
### head(results_limma.deg)
dim(results_limma.deg)
```

```
## [1] 8439    9
```

```
write.table(results_limma.deg, file=paste("analysis.LIMMA.", eset_name, ".only.DEG", sep=""),
                               sep="\t",
                               quote=FALSE, eol="\n",
                               row.names=FALSE, col.names=TRUE)

### computing the number of DEG for FDR < 0.05 and FC > 1.2  : UP-REGULATED GENES :

results_limma.deg.up <- results_limma.sep[(results_limma.sep$adj.P.Val < 0.05) &
                                          (results_limma.sep$logFC > log2(1.2) )  ,]

### head(results_limma.deg.up)
dim(results_limma.deg.up)
```

```
## [1] 3387    9
```

```
write.table(results_limma.deg.up, file=paste("analysis.LIMMA.", eset_name, ".only.DEG.and.UP", sep=""),
                                  sep="\t",
                                  quote=FALSE, eol="\n",
                                  row.names=FALSE, col.names=TRUE)

### computing the number of DEG for FDR < 0.05 and FC < -1.2 : DOWN-REGULATED GENES :

results_limma.deg.down <- results_limma.sep[(results_limma.sep$adj.P.Val < 0.05) &
                                            (results_limma.sep$logFC < -log2(1.2) )  ,]

### head(results_limma.deg.down)
dim(results_limma.deg.down)
```

```
## [1] 3677    9
```

```
write.table(results_limma.deg.down, file=paste("analysis.LIMMA.", eset_name, ".only.DEG.and.DOWN", sep=
                                    sep="\t",
                                    quote=FALSE, eol="\n",
                                    row.names=FALSE, col.names=TRUE)

############################################################################################
############################################################################################
### saving the results into another DATAFRAME to be used later :

eset <- genes.counts.Aph_KH7
eset_name <- deparse(substitute(genes.counts.Aph_KH7))
genes.counts.Aph_KH7.results.limma <- results_limma.sep

### head(genes.counts.Aph_KH7.results.limma)
dim(genes.counts.Aph_KH7.results.limma)
```

```
## [1] 13524     9
```

```
genes.counts.Aph_KH7.results.limma.deg <- results_limma.deg
genes.counts.Aph_KH7.results.limma.deg.up <- results_limma.deg.up
genes.counts.Aph_KH7.results.limma.deg.down <- results_limma.deg.down
```

```r
dim(genes.counts.Aph_KH7.results.limma.deg.up)
```

```
## [1] 3387    9
```

```r
dim(genes.counts.Aph_KH7.results.limma.deg.down)
```

```
## [1] 3677    9
```

```r
#############################################################################################
#############################################################################################
#############################################################################################
#############################################################################################
```

****

4. Performing the DEG analysis : DMSO vs KH7

```
eset <- genes.counts.KH7
eset_name <- deparse(substitute(genes.counts.KH7))

#### genes.counts.KH7

group <- factor(c("DMSO","DMSO","DMSO", "KH7","KH7","KH7"))
subject <- factor(c(1,2,3, 1,2,3))

### setting up the design and the contrast matrix

design <- model.matrix(~0+group+subject)
contrast.matrix <- makeContrasts(groupKH7-groupDMSO, levels=design)

design
```

```
##   groupDMSO groupKH7 subject2 subject3
## 1         1        0        0        0
## 2         1        0        1        0
## 3         1        0        0        1
## 4         0        1        0        0
## 5         0        1        1        0
## 6         0        1        0        1
## attr(,"assign")
## [1] 1 1 2 2
## attr(,"contrasts")
## attr(,"contrasts")$group
## [1] "contr.treatment"
##
## attr(,"contrasts")$subject
## [1] "contr.treatment"
```

```
contrast.matrix
```

```
##            Contrasts
## Levels      groupKH7 - groupDMSO
##   groupDMSO                   -1
##   groupKH7                     1
##   subject2                     0
##   subject3                     0
```

```
############################################################################################

### filtering the genes based on CPM :
y <- DGEList(counts=eset, group=group)

### keep <- rowSums(cpm(y, lib.size=libsize)>1) >= 3
keep <- rowSums( cpm(y) > 0.5) >= 6
y <- y[keep,]
y$samples$lib.size <- colSums(y$counts)

####################################################### we can use y$counts for PCA analysis

### computing the normalization factors :
```

```r
y <- calcNormFactors(y)

### using the VOOM transformation :
v <- voom(y, design, plot=FALSE)

### the LINEAR FIT in LIMMA :
fit <- lmFit(v, design)
fit2 <- contrasts.fit(fit, contrast.matrix)
fit2 <- eBayes(fit2)

### obtaining and writing the results :
results_limma <- topTable(fit2, coef=1, adjust="fdr", number=Inf)

### adding the rownames as columns

results_limma$Gene <- rownames(results_limma)

### separating the names of the GENES into 1st_PART and NUMBER :

results_limma$GENE <- results_limma$Gene

results_limma.sep <- separate(data=results_limma, col=Gene, into = c("Gene", "ID"), sep = ":")

### head(results_limma.sep)
dim(results_limma.sep)
```

```
## [1] 13607     9
```

```r
### writing the results to a file :

write.table(results_limma.sep, file=paste("analysis.LIMMA.", eset_name, sep=""),
                               sep="\t",
                               quote=FALSE, eol="\n",
                               row.names=FALSE, col.names=TRUE)

### computing the number of DEG for FDR < 0.05 :

results_limma.deg <- results_limma.sep[results_limma.sep$adj.P.Val < 0.05,]

### head(results_limma.deg)
dim(results_limma.deg)
```

```
## [1] 8765    9
```

```r
write.table(results_limma.deg, file=paste("analysis.LIMMA.", eset_name, ".only.DEG", sep=""),
                               sep="\t",
                               quote=FALSE, eol="\n",
                               row.names=FALSE, col.names=TRUE)

### computing the number of DEG for FDR < 0.05 and FC > 1.2  : UP-REGULATED GENES :

results_limma.deg.up <- results_limma.sep[(results_limma.sep$adj.P.Val < 0.05) &
                                          (results_limma.sep$logFC > log2(1.2) )  ,]

### head(results_limma.deg.up)
```

```
dim(results_limma.deg.up)
```

```
## [1] 3455    9
```

```
write.table(results_limma.deg.up, file=paste("analysis.LIMMA.", eset_name, ".only.DEG.and.UP", sep=""),
                                  sep="\t",
                                  quote=FALSE, eol="\n",
                                  row.names=FALSE, col.names=TRUE)

### computing the number of DEG for FDR < 0.05 and FC < -1.2 : DOWN-REGULATED GENES :

results_limma.deg.down <- results_limma.sep[(results_limma.sep$adj.P.Val < 0.05) &
                                            (results_limma.sep$logFC < -log2(1.2) )  ,]

### head(results_limma.deg.down)
dim(results_limma.deg.down)
```

```
## [1] 3428    9
```

```
write.table(results_limma.deg.down, file=paste("analysis.LIMMA.", eset_name, ".only.DEG.and.DOWN", sep=
                                    sep="\t",
                                    quote=FALSE, eol="\n",
                                    row.names=FALSE, col.names=TRUE)

############################################################################################

eset <- genes.counts.KH7
eset_name <- deparse(substitute(genes.counts.KH7))
genes.counts.KH7.results.limma <- results_limma.sep

### head(genes.counts.KH7.results.limma)
dim(genes.counts.KH7.results.limma)
```

```
## [1] 13607     9
```

```
genes.counts.KH7.results.limma.deg <- results_limma.deg
genes.counts.KH7.results.limma.deg.up <- results_limma.deg.up
genes.counts.KH7.results.limma.deg.down <- results_limma.deg.down

dim(genes.counts.KH7.results.limma.deg.up)
```

```
## [1] 3455    9
```

```
dim(genes.counts.KH7.results.limma.deg.down)
```

```
## [1] 3428    9
```

```
############################################################################################
############################################################################################
############################################################################################
############################################################################################
```

****

5. Performing the DEG analysis : DMSO vs Noc

```
eset <- genes.counts.Noc
eset_name <- deparse(substitute(genes.counts.Noc))

#### genes.counts.Noc

group <- factor(c("DMSO", "DMSO", "DMSO", "Noc", "Noc", "Noc"))
subject <- factor(c(1,2,3, 1,2,3))

### setting up the design and the contrast matrix

design <- model.matrix(~0+group+subject)
contrast.matrix <- makeContrasts(groupNoc-groupDMSO, levels=design)

design
```

```
##   groupDMSO groupNoc subject2 subject3
## 1         1        0        0        0
## 2         1        0        1        0
## 3         1        0        0        1
## 4         0        1        0        0
## 5         0        1        1        0
## 6         0        1        0        1
## attr(,"assign")
## [1] 1 1 2 2
## attr(,"contrasts")
## attr(,"contrasts")$group
## [1] "contr.treatment"
##
## attr(,"contrasts")$subject
## [1] "contr.treatment"
```

```
contrast.matrix
```

```
##            Contrasts
## Levels      groupNoc - groupDMSO
##   groupDMSO                   -1
##   groupNoc                     1
##   subject2                     0
##   subject3                     0
```

```
############################################################################################

### filtering the genes based on CPM :
y <- DGEList(counts=eset, group=group)

### keep <- rowSums(cpm(y, lib.size=libsize)>1) >= 3
keep <- rowSums( cpm(y) > 0.5) >= 6
y <- y[keep,]
y$samples$lib.size <- colSums(y$counts)

####################################################### we can use y$counts for PCA analysis

### computing the normalization factors :
```

```
y <- calcNormFactors(y)

### using the VOOM transformation :
v <- voom(y, design, plot=FALSE)

### the LINEAR FIT in LIMMA :
fit <- lmFit(v, design)
fit2 <- contrasts.fit(fit, contrast.matrix)
fit2 <- eBayes(fit2)

### obtaining and writing the results :
results_limma <- topTable(fit2, coef=1, adjust="fdr", number=Inf)

### adding the rownames as columns

results_limma$Gene <- rownames(results_limma)

### separating the names of the GENES into 1st_PART and NUMBER :

results_limma$GENE <- results_limma$Gene

results_limma.sep <- separate(data=results_limma, col=Gene, into = c("Gene", "ID"), sep = ":")

head(results_limma.sep)
```

```
##                        logFC   AveExpr         t      P.Value    adj.P.Val
## IGFBP5:4432       3.243658 10.137931 102.50657 1.408471e-23 1.865802e-19
## IL11:2042         2.889583  7.811912  71.32141 4.092890e-21 1.807284e-17
## SGK1:4823         4.017163  6.468127  73.39507 2.615456e-21 1.732347e-17
## AP000892.6:56438  3.028821  7.381289  69.47551 6.165119e-21 2.041733e-17
## FABP7:11356      -3.141010  6.895168 -65.84429 1.425824e-20 3.777577e-17
## SLC7A5:2791       2.354418  7.862691  58.47411 9.094516e-20 1.721072e-16
##                         B       Gene    ID             GENE
## IGFBP5:4432      44.14693     IGFBP5  4432      IGFBP5:4432
## IL11:2042        38.63660       IL11  2042        IL11:2042
## SGK1:4823        38.51849       SGK1  4823        SGK1:4823
## AP000892.6:56438 38.18597 AP000892.6 56438 AP000892.6:56438
## FABP7:11356      37.31229      FABP7 11356      FABP7:11356
## SLC7A5:2791      35.70521     SLC7A5  2791      SLC7A5:2791
```

```
dim(results_limma.sep)
```

```
## [1] 13247     9
```

```
### writing the results to a file :

write.table(results_limma.sep, file=paste("analysis.LIMMA.", eset_name, sep=""),
                               sep="\t",
                               quote=FALSE, eol="\n",
                               row.names=FALSE, col.names=TRUE)

### computing the number of DEG for FDR < 0.05 :

results_limma.deg <- results_limma.sep[results_limma.sep$adj.P.Val < 0.05,]
```

```
### head(results_limma.deg)
dim(results_limma.deg)
```

```
## [1] 8770    9
```

```
write.table(results_limma.deg, file=paste("analysis.LIMMA.", eset_name, ".only.DEG", sep=""),
                               sep="\t",
                               quote=FALSE, eol="\n",
                               row.names=FALSE, col.names=TRUE)

### computing the number of DEG for FDR < 0.05 and FC > 1.2  : UP-REGULATED GENES :

results_limma.deg.up <- results_limma.sep[(results_limma.sep$adj.P.Val < 0.05) &
                                          (results_limma.sep$logFC > log2(1.2) )  ,]

### head(results_limma.deg.up)
dim(results_limma.deg.up)
```

```
## [1] 3357    9
```

```
write.table(results_limma.deg.up, file=paste("analysis.LIMMA.", eset_name, ".only.DEG.and.UP", sep=""),
                                  sep="\t",
                                  quote=FALSE, eol="\n",
                                  row.names=FALSE, col.names=TRUE)

### computing the number of DEG for FDR < 0.05 and FC < -1.2 : DOWN-REGULATED GENES :

results_limma.deg.down <- results_limma.sep[(results_limma.sep$adj.P.Val < 0.05) &
                                            (results_limma.sep$logFC < -log2(1.2) )  ,]

### head(results_limma.deg.down)
dim(results_limma.deg.down)
```

```
## [1] 3542    9
```

```
write.table(results_limma.deg.down, file=paste("analysis.LIMMA.", eset_name, ".only.DEG.and.DOWN", sep=
                                    sep="\t",
                                    quote=FALSE, eol="\n",
                                    row.names=FALSE, col.names=TRUE)

############################################################################################

eset <- genes.counts.Noc
eset_name <- deparse(substitute(genes.counts.Noc))
genes.counts.Noc.results.limma <- results_limma.sep

### head(genes.counts.Noc.results.limma)
dim(genes.counts.Noc.results.limma)
```

```
## [1] 13247     9
```

```
genes.counts.Noc.results.limma.deg <- results_limma.deg
genes.counts.Noc.results.limma.deg.up <- results_limma.deg.up
genes.counts.Noc.results.limma.deg.down <- results_limma.deg.down

dim(genes.counts.Noc.results.limma.deg.up)
```

```
## [1] 3357    9
```

```
dim(genes.counts.Noc.results.limma.deg.down)
```

```
## [1] 3542    9
```

```
#############################################################################################
#############################################################################################
#############################################################################################
#############################################################################################
#############################################################################################
```

****

6. INTEGRATING all the DATAFRAMES that contain DEG

```
#### AT THIS MOMENT, we would like to INTEGRATE all the DATAFILES from LIMMA that we have :

#### genes OR genes.counts

#### genes.counts.Aph.results.limma
#### genes.counts.Aph_KH7.results.limma
#### genes.counts.KH7.results.limma
#### genes.counts.Noc.results.limma

dim(genes)
```

```
## [1] 58381    63
```

```
dim(genes.counts.Aph.results.limma)
```

```
## [1] 13804     9
```

```
dim(genes.counts.Aph_KH7.results.limma)
```

```
## [1] 13524     9
```

```
dim(genes.counts.KH7.results.limma)
```

```
## [1] 13607     9
```

```
dim(genes.counts.Noc.results.limma)
```

```
## [1] 13247     9
```

```
############################################################################################
############################################################################################
############################################################################################
############################################################################################
############################################################################################
############################################################################################

################################## we will have to change the names of columns,
################################## because all the dataframes have the same COLUMN NAMES

################################## here working with a DATAFRAME : Aph

colnames(genes.counts.Aph.results.limma)[1] <- paste("logFC" ,"Aph" , sep=":")
colnames(genes.counts.Aph.results.limma)[2] <- paste("AveExpr" ,"Aph" , sep=":")
colnames(genes.counts.Aph.results.limma)[3] <- paste("t" ,"Aph" , sep=":")
colnames(genes.counts.Aph.results.limma)[4] <- paste("P.Value" ,"Aph" , sep=":")
colnames(genes.counts.Aph.results.limma)[5] <- paste("adj.P.Val" ,"Aph" , sep=":")
colnames(genes.counts.Aph.results.limma)[6] <- paste("B" ,"Aph" , sep=":")
colnames(genes.counts.Aph.results.limma)[7] <- paste("Gene" ,"Aph" , sep=":")
colnames(genes.counts.Aph.results.limma)[8] <- paste("ID" ,"Aph" , sep=":")
colnames(genes.counts.Aph.results.limma)
```

```
## [1] "logFC:Aph"     "AveExpr:Aph"   "t:Aph"         "P.Value:Aph"
## [5] "adj.P.Val:Aph" "B:Aph"         "Gene:Aph"      "ID:Aph"
## [9] "GENE"
```

```
# colnames(genes.counts.Aph.results.limma)[9] <- paste("GENE" ,"Aph" , sep=":")

################################### here working with a DATAFRAME : KH7

colnames(genes.counts.KH7.results.limma)[1] <- paste("logFC" ,"KH7" , sep=":")
colnames(genes.counts.KH7.results.limma)[2] <- paste("AveExpr" ,"KH7" , sep=":")
colnames(genes.counts.KH7.results.limma)[3] <- paste("t" ,"KH7" , sep=":")
colnames(genes.counts.KH7.results.limma)[4] <- paste("P.Value" ,"KH7" , sep=":")
colnames(genes.counts.KH7.results.limma)[5] <- paste("adj.P.Val" ,"KH7" , sep=":")
colnames(genes.counts.KH7.results.limma)[6] <- paste("B" ,"KH7" , sep=":")
colnames(genes.counts.KH7.results.limma)[7] <- paste("Gene" ,"KH7" , sep=":")
colnames(genes.counts.KH7.results.limma)[8] <- paste("ID" ,"KH7" , sep=":")
colnames(genes.counts.KH7.results.limma)
```

```
## [1] "logFC:KH7"     "AveExpr:KH7"   "t:KH7"         "P.Value:KH7"
## [5] "adj.P.Val:KH7" "B:KH7"         "Gene:KH7"      "ID:KH7"
## [9] "GENE"
```

```
# colnames(genes.counts.KH7.results.limma)[9] <- paste("GENE" ,"KH7" , sep=":")

################################### here working with a DATAFRAME : Aph_KH7

colnames(genes.counts.Aph_KH7.results.limma)[1] <- paste("logFC" ,"Aph_KH7" , sep=":")
colnames(genes.counts.Aph_KH7.results.limma)[2] <- paste("AveExpr" ,"Aph_KH7" , sep=":")
colnames(genes.counts.Aph_KH7.results.limma)[3] <- paste("t" ,"Aph_KH7" , sep=":")
colnames(genes.counts.Aph_KH7.results.limma)[4] <- paste("P.Value" ,"Aph_KH7" , sep=":")
colnames(genes.counts.Aph_KH7.results.limma)[5] <- paste("adj.P.Val" ,"Aph_KH7" , sep=":")
colnames(genes.counts.Aph_KH7.results.limma)[6] <- paste("B" ,"Aph_KH7" , sep=":")
colnames(genes.counts.Aph_KH7.results.limma)[7] <- paste("Gene" ,"Aph_KH7" , sep=":")
colnames(genes.counts.Aph_KH7.results.limma)[8] <- paste("ID" ,"Aph_KH7" , sep=":")
colnames(genes.counts.Aph_KH7.results.limma)
```

```
## [1] "logFC:Aph_KH7"     "AveExpr:Aph_KH7"   "t:Aph_KH7"
## [4] "P.Value:Aph_KH7"   "adj.P.Val:Aph_KH7" "B:Aph_KH7"
## [7] "Gene:Aph_KH7"      "ID:Aph_KH7"        "GENE"
```

```
# colnames(genes.counts.Aph_KH7.results.limma)[9] <- paste("GENE" ,"Aph_KH7" , sep=":")

################################### here working with a DATAFRAME : Noc

colnames(genes.counts.Noc.results.limma)[1] <- paste("logFC" ,"Noc" , sep=":")
colnames(genes.counts.Noc.results.limma)[2] <- paste("AveExpr" ,"Noc" , sep=":")
colnames(genes.counts.Noc.results.limma)[3] <- paste("t" ,"Noc" , sep=":")
colnames(genes.counts.Noc.results.limma)[4] <- paste("P.Value" ,"Noc" , sep=":")
colnames(genes.counts.Noc.results.limma)[5] <- paste("adj.P.Val" ,"Noc" , sep=":")
colnames(genes.counts.Noc.results.limma)[6] <- paste("B" ,"Noc" , sep=":")
colnames(genes.counts.Noc.results.limma)[7] <- paste("Gene" ,"Noc" , sep=":")
colnames(genes.counts.Noc.results.limma)[8] <- paste("ID" ,"Noc" , sep=":")
colnames(genes.counts.Noc.results.limma)
```

```
## [1] "logFC:Noc"     "AveExpr:Noc"   "t:Noc"         "P.Value:Noc"
## [5] "adj.P.Val:Noc" "B:Noc"         "Gene:Noc"      "ID:Noc"
## [9] "GENE"
```

```
# colnames(genes.counts.Noc.results.limma)[9] <- paste("GENE" ,"Noc" , sep=":")
```

```
############################################################################################
############################################################################################
############################################################################################
############################################################################################
############################################################################################
############################################################################################
### library(data.table)
### now integrating these data structures ; we can make DATA TABLES :

genes.dt <- as.data.table(genes)

genes.counts.Aph.results.limma.dt <- as.data.table(genes.counts.Aph.results.limma)
genes.counts.Aph_KH7.results.limma.dt <- as.data.table(genes.counts.Aph_KH7.results.limma)
genes.counts.KH7.results.limma.dt <- as.data.table(genes.counts.KH7.results.limma)
genes.counts.Noc.results.limma.dt <- as.data.table(genes.counts.Noc.results.limma)

#################################### setting up the KEYS :

setkeyv(genes.dt, c('GENE_NAME_ID'))

setkeyv(genes.counts.Aph.results.limma.dt, c('GENE'))
setkeyv(genes.counts.Aph_KH7.results.limma.dt, c('GENE'))
setkeyv(genes.counts.KH7.results.limma.dt, c('GENE'))
setkeyv(genes.counts.Noc.results.limma.dt, c('GENE'))

########################################################################################################
########################################################################################################
########################################################################################################

integration.all.samples.dt <- genes.dt[genes.counts.Aph.results.limma.dt,][genes.counts.Aph_KH7.results

head(integration.all.samples.dt)
```

```
##      CHR     START       END STRAND            GENE_ID GENE_NAME
## 1: chr19  58347751  58355183      +  ENSG00000268895.5  A1BG-AS1
## 2: chr19  58346850  58353499      - ENSG00000121410.11      A1BG
## 3: chr12   9067712   9116157      - ENSG00000175899.14       A2M
## 4: chr22  42692122  42695633      - ENSG00000128274.16    A4GALT
## 5: chr12  53307456  53321631      - ENSG00000094914.12      AAAS
## 6: chr12 125065379 125143320      + ENSG00000081760.16      AACS
##         GENE_TYPE DMSO1_lane1.count DMSO1_lane1.TPM DMSO1_lane1.FPKM
## 1:      antisense             70.26            2.27             1.25
## 2: protein_coding            128.87            4.32             2.38
## 3: protein_coding           1186.00           12.94             7.13
## 4: protein_coding             79.00            2.13             1.17
## 5: protein_coding           1343.00           42.10            23.19
## 6: protein_coding           1885.00           30.70            16.91
##    DMSO1_lane2.count DMSO1_lane2.TPM DMSO1_lane2.FPKM DMSO2_lane1.count
## 1:             73.87            2.61             1.43             60.07
## 2:             83.00            2.95             1.62            121.84
## 3:           1049.00           12.14             6.68            968.00
## 4:             82.00            2.46             1.35             74.00
## 5:           1261.00           41.89            23.05           1305.00
## 6:           1720.00           29.71            16.35           1653.00
##    DMSO2_lane1.TPM DMSO2_lane1.FPKM DMSO2_lane2.count DMSO2_lane2.TPM
## 1:            1.98             1.14             47.39            1.72
## 2:            4.16             2.39            108.94            3.97
## 3:           10.70             6.15            914.00           10.80
## 4:            2.08             1.20             99.00            2.95
## 5:           41.96            24.12           1158.00           39.56
## 6:           27.30            15.69           1511.00           26.69
##    DMSO2_lane2.FPKM DMSO3_lane1.count DMSO3_lane1.TPM DMSO3_lane1.FPKM
## 1:             0.98             92.48            2.61             1.50
## 2:             2.28            100.72            2.97             1.71
## 3:             6.19           1068.00           10.21             5.88
## 4:             1.69            100.00            2.36             1.36
## 5:            22.67           1573.00           43.53            25.08
## 6:            15.30           1928.00           27.56            15.88
##    DMSO3_lane2.count DMSO3_lane2.TPM DMSO3_lane2.FPKM Aph1.count Aph1.TPM
## 1:             68.09            2.08             1.19     100.30     3.48
## 2:             81.00            2.54             1.46     148.96     5.42
## 3:           1158.00           11.78             6.78    1038.00    12.19
## 4:             85.00            2.13             1.22     200.00     5.96
## 5:           1417.00           41.58            23.92     784.00    26.86
## 6:           1818.00           27.64            15.90    1858.00    32.63
##    Aph1.FPKM Aph2.count Aph2.TPM Aph2.FPKM Aph3.count Aph3.TPM Aph3.FPKM
## 1:      1.93      85.23     2.48      1.35     121.79     3.05      1.66
## 2:      3.01     150.89     4.58      2.49     136.89     3.57      1.95
## 3:      6.77    1450.00    14.27      7.77    1670.00    14.16      7.73
## 4:      3.31     205.00     5.07      2.76     178.00     3.60      1.96
## 5:     14.93     993.00    28.26     15.38    1056.00    25.82     14.09
## 6:     18.14    2047.00    30.10     16.38    2371.00    30.03     16.39
##    Aph_KH7_1.count Aph_KH7_1.TPM Aph_KH7_1.FPKM Aph_KH7_2.count
## 1:          108.96          3.26           1.86          131.61
## 2:          197.00          6.02           3.43          191.00
## 3:          860.00          8.51           4.85          832.00
## 4:          140.00          3.57           2.04          141.00
## 5:          701.00         20.22          11.53          766.00
## 6:         1706.00         25.22          14.38         1801.00
##    Aph_KH7_2.TPM Aph_KH7_2.FPKM Aph_KH7_3.count Aph_KH7_3.TPM
## 1:          3.82           2.10          145.94          4.47
## 2:          5.63           3.09          201.00          6.23
## 3:          7.93           4.36          858.00          8.59
## 4:          3.36           1.84          122.00          3.09
## 5:         21.40          11.76          721.00         21.06
## 6:         25.67          14.10         1683.00         25.20
##    Aph_KH7_3.FPKM KH7_1.count KH7_1.TPM KH7_1.FPKM KH7_2.count KH7_2.TPM
## 1:           2.49      129.81      4.01       2.22       72.60      2.41
## 2:           3.48      144.95      4.63       2.56      151.92      5.25
## 3:           4.79     1009.00     10.45       5.77      978.00     10.89
## 4:           1.72      141.00      3.70       2.05      189.00      5.22
## 5:          11.74      796.00     23.86      13.19      738.00     23.99
## 6:          14.05     1936.00     29.95      16.56     1687.00     28.12
##    KH7_2.FPKM KH7_3.count KH7_3.TPM KH7_3.FPKM Noc_1.count Noc_1.TPM
## 1:       1.34      100.99      3.14       1.77       48.01      1.44
## 2:       2.92      167.00      5.46       3.08      106.00      3.16
## 3:       6.06      911.00      9.64       5.44      945.00      9.09
## 4:       2.90      139.00      3.73       2.10      459.00     10.96
## 5:      13.35      798.00     24.57      13.86     1278.00     35.58
## 6:      15.65     1980.00     31.32      17.66     2314.00     33.31
##    Noc_1.FPKM Noc_2.count Noc_2.TPM Noc_2.FPKM Noc_3.count Noc_3.TPM
## 1:       0.79       43.17      1.32       0.73       47.46      1.63
## 2:       1.73       84.00      2.67       1.48       72.00      2.47
## 3:       4.98      840.00      8.66       4.78      865.00      9.64
## 4:       6.00      433.00     11.10       6.12      437.00     12.15
## 5:      19.48     1159.00     34.43      19.00     1029.00     33.20
## 6:      18.24     2076.00     31.98      17.65     2011.00     33.45
##    Noc_3.FPKM    ID   GENE_NAME_ID   logFC:Aph AveExpr:Aph       t:Aph
## 1:       0.90 50224 A1BG-AS1:50224  0.29410626    1.189233   1.4536497
## 2:       1.37  5165      A1BG:5165  0.15414695    1.783208   0.9427268
## 3:       5.35 13981      A2M:13981  0.18596443    4.990345   2.0685938
## 4:       6.74  6005    A4GALT:6005  1.03743913    1.752208   5.2756291
## 5:      18.43  2010      AAAS:2010 -0.75248881    4.916487 -10.0061737
## 6:      18.57  1543      AACS:1543  0.02859842    5.679395   0.4762444
##     P.Value:Aph adj.P.Val:Aph      B:Aph Gene:Aph ID:Aph logFC:Aph_KH7
## 1: 1.709433e-01  2.706080e-01 -5.5310759 A1BG-AS1  50224    0.81515119
## 2: 3.638715e-01  4.826446e-01 -6.2862436     A1BG   5165    0.75566999
## 3: 6.015749e-02  1.187585e-01 -5.6058546      A2M  13981   -0.32055627
## 4: 1.771840e-04  1.194845e-03  0.9524451   A4GALT   6005    0.68603291
## 5: 2.749308e-07  8.801931e-06  7.0549541     AAAS   2010   -0.93145454
## 6: 6.422029e-01  7.336729e-01 -7.6070032     AACS   1543   -0.05954757
##    AveExpr:Aph_KH7   t:Aph_KH7 P.Value:Aph_KH7 adj.P.Val:Aph_KH7
## 1:        1.360675   4.0115612    9.483713e-04      2.188325e-03
## 2:        2.003132   4.3277969    4.836075e-04      1.193051e-03
## 3:        4.655879  -4.0151390    9.411459e-04      2.172765e-03
## 4:        1.490768   3.5160155    2.742924e-03      5.702583e-03
## 5:        4.739835 -12.1252408    1.186132e-09      1.423358e-08
## 6:        5.550024  -0.9272695    3.671140e-01      4.367775e-01
##     B:Aph_KH7 Gene:Aph_KH7 ID:Aph_KH7   logFC:KH7 AveExpr:KH7      t:KH7
## 1: -1.0636776     A1BG-AS1      50224  0.48798805    1.232588   2.339598
## 2: -0.5759844         A1BG       5165  0.44487929    1.890710   2.408672
## 3: -1.9667728          A2M      13981 -0.10801334    4.807986  -1.511183
## 4: -2.1614380       A4GALT       6005  0.93385390    1.654405   4.628395
## 5: 12.0886962         AAAS       2010 -0.80559752    4.846259 -11.914604
## 6: -7.6475664         AACS       1543  0.08307658    5.664225   1.434602
##     P.Value:KH7 adj.P.Val:KH7      B:KH7 Gene:KH7 ID:KH7  logFC:Noc
## 1: 2.998760e-02  4.687435e-02 -4.4590145 A1BG-AS1  50224 -0.6343953
## 2: 2.595905e-02  4.118278e-02 -4.4769824     A1BG   5165 -0.4109608
## 3: 1.466555e-01  1.931980e-01 -6.9436716      A2M  13981 -0.2561607
## 4: 1.691506e-04  4.314130e-04  0.4949528   A4GALT   6005  2.4226068
## 5: 1.921999e-10  2.157809e-09 13.6296328     AAAS   2010 -0.2564614
## 6: 1.671238e-01  2.171142e-01 -7.2341983     AACS   1543  0.2553436
##    AveExpr:Noc     t:Noc  P.Value:Noc adj.P.Val:Noc      B:Noc Gene:Noc
## 1:   0.6594847 -2.875710 1.115945e-02  1.867710e-02 -3.2823852 A1BG-AS1
## 2:   1.4411810 -2.337104 3.306120e-02  4.995001e-02 -4.5298272     A1BG
## 3:   4.7118382 -3.407259 3.696586e-03  6.850682e-03 -3.3599036      A2M
## 4:   2.3800295 17.113076 1.471227e-11  2.883040e-10 16.8251876   A4GALT
## 5:   5.0982177 -4.084189 8.994261e-04  1.909101e-03 -2.0224340     AAAS
## 6:   5.7293178  4.802849 2.065482e-04  5.040795e-04 -0.6620337     AACS
##    ID:Noc
## 1:  50224
## 2:   5165
## 3:  13981
## 4:   6005
## 5:   2010
## 6:   1543
```

```
dim(integration.all.samples.dt)
```

```
## [1] 13247    95
```

```
############################# I dunno why the size of the data frame is : dim(integration.all.samples)
############################# it seems that it may have selected only the COMMON genes ... possibly ...
############################# anyway, we are going to write the file for computing the fold changes ...

write.table(integration.all.samples.dt, file=paste("analysis.LIMMA.integrating.all.samples.with.data.ta
                               sep="\t",
                               quote=FALSE, eol="\n",
                               row.names=FALSE, col.names=TRUE)

#######################################################################################################
#######################################################################################################
#######################################################################################################
#######################################################################################################
#######################################################################################################
#######################################################################################################

################### starting from the dataframe "genes" :

dim(genes)
```

```
## [1] 58381    63
```

```
### head(genes)

################### integrate with : genes.counts.Aph.results.limma

integration.step1 <- merge(genes,
                           genes.counts.Aph.results.limma,
                           by.x = "GENE_NAME_ID" ,
                           by.y = "GENE",
                           all.x = TRUE)

### head(integration.step1)
dim(integration.step1)
```

```
## [1] 58381    71
```

```
################### integrate with : genes.counts.Aph_KH7.results.limma

integration.step2 <- merge(integration.step1,
                           genes.counts.Aph_KH7.results.limma,
                           by.x = "GENE_NAME_ID" ,
                           by.y = "GENE",
                           all.x = TRUE)

### head(integration.step2)
dim(integration.step2)
```

```
## [1] 58381    79
```

```
################### integrate with : genes.counts.KH7.results.limma

integration.step3 <- merge(integration.step2,
                           genes.counts.KH7.results.limma,
                           by.x = "GENE_NAME_ID" ,
                           by.y = "GENE",
                           all.x = TRUE)

### head(integration.step3)
dim(integration.step3)
```

```
## [1] 58381    87
```

```
################### integrate with : genes.counts.Noc.results.limma

integration.step4 <- merge(integration.step3,
                           genes.counts.Noc.results.limma,
                           by.x = "GENE_NAME_ID" ,
                           by.y = "GENE",
                           all.x = TRUE)

### head(integration.step4)
dim(integration.step4)
```

```
## [1] 58381    95
```

```
############################################################################################
############################################################################################

########################## we are going to write the file for computing the fold changes ...

write.table(integration.step4, file=paste("analysis.LIMMA.integrating.all.samples.all.genes.in.4.STEPS"
                               sep="\t",
                               quote=FALSE, eol="\n",
                               row.names=FALSE, col.names=TRUE)

############################################################################################
############################################################################################

########### starting from the dataframe "integration.step4", to separate the DEG, function of FDR, and

# x <- integration.step4

# > colnames(x)
# [1] "GENE_NAME_ID"      "CHR"                "START"
# [4] "END"               "STRAND"             "GENE_ID"
# [7] "GENE_NAME"         "GENE_TYPE"          "DMSO1_lane1.count"
#[10] "DMSO1_lane1.TPM"   "DMSO1_lane1.FPKM"   "DMSO1_lane2.count"
#[13] "DMSO1_lane2.TPM"   "DMSO1_lane2.FPKM"   "DMSO2_lane1.count"
#[16] "DMSO2_lane1.TPM"   "DMSO2_lane1.FPKM"   "DMSO2_lane2.count"
#[19] "DMSO2_lane2.TPM"   "DMSO2_lane2.FPKM"   "DMSO3_lane1.count"
#[22] "DMSO3_lane1.TPM"   "DMSO3_lane1.FPKM"   "DMSO3_lane2.count"
#[25] "DMSO3_lane2.TPM"   "DMSO3_lane2.FPKM"   "Aph1.count"
```

```
#[28] "Aph1.TPM"          "Aph1.FPKM"         "Aph2.count"
#[31] "Aph2.TPM"          "Aph2.FPKM"         "Aph3.count"
#[34] "Aph3.TPM"          "Aph3.FPKM"         "Aph_KH7_1.count"
#[37] "Aph_KH7_1.TPM"     "Aph_KH7_1.FPKM"    "Aph_KH7_2.count"
#[40] "Aph_KH7_2.TPM"     "Aph_KH7_2.FPKM"    "Aph_KH7_3.count"
#[43] "Aph_KH7_3.TPM"     "Aph_KH7_3.FPKM"    "KH7_1.count"
#[46] "KH7_1.TPM"         "KH7_1.FPKM"        "KH7_2.count"
#[49] "KH7_2.TPM"         "KH7_2.FPKM"        "KH7_3.count"
#[52] "KH7_3.TPM"         "KH7_3.FPKM"        "Noc_1.count"
#[55] "Noc_1.TPM"         "Noc_1.FPKM"        "Noc_2.count"
#[58] "Noc_2.TPM"         "Noc_2.FPKM"        "Noc_3.count"
#[61] "Noc_3.TPM"         "Noc_3.FPKM"        "ID"
#[64] "logFC:Aph"         "AveExpr:Aph"       "t:Aph"
#[67] "P.Value:Aph"       "adj.P.Val:Aph"     "B:Aph"
#[70] "Gene:Aph"          "ID:Aph"            "logFC:Aph_KH7"
#[73] "AveExpr:Aph_KH7"   "t:Aph_KH7"         "P.Value:Aph_KH7"
#[76] "adj.P.Val:Aph_KH7" "B:Aph_KH7"         "Gene:Aph_KH7"
#[79] "ID:Aph_KH7"        "logFC:KH7"         "AveExpr:KH7"
#[82] "t:KH7"             "P.Value:KH7"       "adj.P.Val:KH7"
#[85] "B:KH7"             "Gene:KH7"          "ID:KH7"
#[88] "logFC:Noc"         "AveExpr:Noc"       "t:Noc"
#[91] "P.Value:Noc"       "adj.P.Val:Noc"     "B:Noc"
#[94] "Gene:Noc"          "ID:Noc"

#############################################################################################
############### using as criteria for filtering the following fields :

# "DMSO1_lane1.FPKM"
# "DMSO1_lane2.FPKM"
# "DMSO2_lane1.FPKM"
# "DMSO2_lane2.FPKM"
# "DMSO3_lane1.FPKM"
# "DMSO3_lane2.FPKM"
# "Aph1.FPKM"
# "Aph2.FPKM"
# "Aph3.FPKM"
# "Aph_KH7_1.FPKM"
# "Aph_KH7_2.FPKM"
# "Aph_KH7_3.FPKM"
# "KH7_1.FPKM"
# "KH7_2.FPKM"
# "KH7_3.FPKM"
# "Noc_1.FPKM"
# "Noc_2.FPKM"

################ considering the FPKM, FC, and FDR :

#"logFC:Aph"
#"adj.P.Val:Aph"
#"logFC:Aph_KH7"
#"adj.P.Val:Aph_KH7"
#"logFC:KH7"
#"adj.P.Val:KH7"
```

```
#"logFC:Noc"
#"adj.P.Val:Noc"

x <- integration.step4

head(x$"logFC:Aph")
```

```
## [1] 0.1541469 0.2941063        NA 0.1859644        NA        NA
```

```
head(x$"adj.P.Val:Aph")
```

```
## [1] 0.4826446 0.2706080        NA 0.1187585        NA        NA
```

```
head(x$"logFC:Aph_KH7")
```

```
## [1]  0.7556700  0.8151512         NA -0.3205563         NA         NA
```

```
head(x$"adj.P.Val:Aph_KH7")
```

```
## [1] 0.001193051 0.002188325          NA 0.002172765          NA          NA
```

```
head(x$"logFC:KH7")
```

```
## [1]  0.4448793  0.4879880         NA -0.1080133         NA         NA
```

```
head(x$"adj.P.Val:KH7")
```

```
## [1] 0.04118278 0.04687435         NA 0.19319797         NA         NA
```

```
head(x$"logFC:Noc")
```

```
## [1] -0.4109608 -0.6343953         NA -0.2561607         NA         NA
```

```
head(x$"adj.P.Val:Noc")
```

```
## [1] 0.049950011 0.018677102          NA 0.006850682          NA          NA
```

```
###  for some reason that I do not know, the R code for subsetting the BIG DATA FRAME is not working !
###  to try again at some time point ..

### head(x)
### tail(x)

dim(x)
```

```
## [1] 58381    95
```

```
############################################################################################
############################################################################################
############################################################################################
############################################################################################
############################################################################################
############################################################################################
```

****

7. PRINTING the LISTS of DEG

```
### to integrate these DATAFRAMES with X, and to print it :

############################################################################################
############################################################################################
####################### considering the comparisons DMSO:Aph :

eset_name <- deparse(substitute(genes.counts.Aph))

# genes.counts.Aph.results.limma.deg
# genes.counts.Aph.results.limma.deg.up
# genes.counts.Aph.results.limma.deg.down

genes.counts.Aph.results.limma.deg.up.and.x <- merge(genes.counts.Aph.results.limma.deg.up,
                                                     x,
                                                     by.x="GENE",
                                                     by.y="GENE_NAME_ID",
                                                     all.x = TRUE)



write.table(genes.counts.Aph.results.limma.deg.up.and.x,
            file=paste("analysis.LIMMA.", eset_name, ".only.DEG.and.UP.info.all.samples", sep=""),
            sep="\t",
            quote=FALSE, eol="\n",
            row.names=FALSE, col.names=TRUE)

# head(genes.counts.Aph.results.limma.deg.up.and.x)
# tail(genes.counts.Aph.results.limma.deg.up.and.x)
dim(genes.counts.Aph.results.limma.deg.up.and.x)
```

```
## [1] 2259  103
```

```
genes.counts.Aph.results.limma.deg.down.and.x <- merge(genes.counts.Aph.results.limma.deg.down,
                                                     x,
                                                     by.x="GENE",
                                                     by.y="GENE_NAME_ID",
                                                     all.x = TRUE)

write.table(genes.counts.Aph.results.limma.deg.down.and.x,
            file=paste("analysis.LIMMA.", eset_name, ".only.DEG.and.DOWN.info.all.samples", sep=""),
            sep="\t",
            quote=FALSE, eol="\n",
            row.names=FALSE, col.names=TRUE)

# head(genes.counts.Aph.results.limma.deg.down.and.x)
# tail(genes.counts.Aph.results.limma.deg.down.and.x)
dim(genes.counts.Aph.results.limma.deg.down.and.x)
```

```
## [1] 1844  103
```

```
############################################################################################
############################################################################################
####################### considering the comparisons DMSO:Aph_KH7 :
```

```r
eset_name <- deparse(substitute(genes.counts.Aph_KH7))

# genes.counts.Aph_KH7.results.limma.deg
# genes.counts.Aph_KH7.results.limma.deg.up
# genes.counts.Aph_KH7.results.limma.deg.down

genes.counts.Aph_KH7.results.limma.deg.up.and.x <- merge(genes.counts.Aph_KH7.results.limma.deg.up,
                                                         x,
                                                         by.x="GENE",
                                                         by.y="GENE_NAME_ID",
                                                         all.x = TRUE)

write.table(genes.counts.Aph_KH7.results.limma.deg.up.and.x,
            file=paste("analysis.LIMMA.", eset_name, ".only.DEG.and.UP.info.all.samples", sep=""),
            sep="\t",
            quote=FALSE, eol="\n",
            row.names=FALSE, col.names=TRUE)

# head(genes.counts.Aph_KH7.results.limma.deg.up.and.x)
# tail(genes.counts.Aph_KH7.results.limma.deg.up.and.x)
dim(genes.counts.Aph_KH7.results.limma.deg.up.and.x)
```

```
## [1] 3387  103
```

```r
genes.counts.Aph_KH7.results.limma.deg.down.and.x <- merge(genes.counts.Aph_KH7.results.limma.deg.down,
                                                         x,
                                                         by.x="GENE",
                                                         by.y="GENE_NAME_ID",
                                                         all.x = TRUE)

write.table(genes.counts.Aph_KH7.results.limma.deg.down.and.x,
            file=paste("analysis.LIMMA.", eset_name, ".only.DEG.and.DOWN.info.all.samples", sep=""),
            sep="\t",
            quote=FALSE, eol="\n",
            row.names=FALSE, col.names=TRUE)

# head(genes.counts.Aph_KH7.results.limma.deg.down.and.x)
# tail(genes.counts.Aph_KH7.results.limma.deg.down.and.x)
dim(genes.counts.Aph_KH7.results.limma.deg.down.and.x)
```

```
## [1] 3677  103
```

```r
############################################################################################
############################################################################################
####################### considering the comparisons DMSO:KH7 :

eset_name <- deparse(substitute(genes.counts.KH7))

# genes.counts.KH7.results.limma.deg
# genes.counts.KH7.results.limma.deg.up
# genes.counts.KH7.results.limma.deg.down

genes.counts.KH7.results.limma.deg.up.and.x <- merge(genes.counts.KH7.results.limma.deg.up,
                                                         x,
                                                         by.x="GENE",
                                                         by.y="GENE_NAME_ID",
                                                         all.x = TRUE)

write.table(genes.counts.KH7.results.limma.deg.up.and.x,
            file=paste("analysis.LIMMA.", eset_name, ".only.DEG.and.UP.info.all.samples", sep=""),
            sep="\t",
            quote=FALSE, eol="\n",
            row.names=FALSE, col.names=TRUE)

# head(genes.counts.KH7.results.limma.deg.up.and.x)
# tail(genes.counts.KH7.results.limma.deg.up.and.x)
dim(genes.counts.KH7.results.limma.deg.up.and.x)
```

```
## [1] 3455  103
```

```
genes.counts.KH7.results.limma.deg.down.and.x  <- merge(genes.counts.KH7.results.limma.deg.down,
                                                         x,
                                                         by.x="GENE",
                                                         by.y="GENE_NAME_ID",
                                                         all.x = TRUE)

write.table(genes.counts.KH7.results.limma.deg.down.and.x,
            file=paste("analysis.LIMMA.", eset_name, ".only.DEG.and.DOWN.info.all.samples", sep=""),
            sep="\t",
            quote=FALSE, eol="\n",
            row.names=FALSE, col.names=TRUE)

# head(genes.counts.KH7.results.limma.deg.down.and.x)
# tail(genes.counts.KH7.results.limma.deg.down.and.x)
dim(genes.counts.KH7.results.limma.deg.down.and.x)
```

```
## [1] 3428  103
```

```
############################################################################################
############################################################################################
####################### considering the comparisons DMSO:Noc :

eset_name <- deparse(substitute(genes.counts.Noc))

# genes.counts.Noc.results.limma.deg
# genes.counts.Noc.results.limma.deg.up
# genes.counts.Noc.results.limma.deg.down


genes.counts.Noc.results.limma.deg.up.and.x <- merge(genes.counts.Noc.results.limma.deg.up,
                                                         x,
                                                         by.x="GENE",
                                                         by.y="GENE_NAME_ID",
                                                         all.x = TRUE)

write.table(genes.counts.Noc.results.limma.deg.up.and.x,
            file=paste("analysis.LIMMA.", eset_name, ".only.DEG.and.UP.info.all.samples", sep=""),
            sep="\t",
            quote=FALSE, eol="\n",
            row.names=FALSE, col.names=TRUE)

# head(genes.counts.Noc.results.limma.deg.up.and.x)
# tail(genes.counts.Noc.results.limma.deg.up.and.x)
dim(genes.counts.Noc.results.limma.deg.up.and.x)
```

```
## [1] 3357  103
```

```
genes.counts.Noc.results.limma.deg.down.and.x  <- merge(genes.counts.Noc.results.limma.deg.down,
                                                         x,
                                                         by.x="GENE",
                                                         by.y="GENE_NAME_ID",
                                                         all.x = TRUE)

write.table(genes.counts.Noc.results.limma.deg.down.and.x,
            file=paste("analysis.LIMMA.", eset_name, ".only.DEG.and.DOWN.info.all.samples", sep=""),
            sep="\t",
            quote=FALSE, eol="\n",
            row.names=FALSE, col.names=TRUE)

# head(genes.counts.Noc.results.limma.deg.down.and.x)
# tail(genes.counts.Noc.results.limma.deg.down.and.x)
dim(genes.counts.Noc.results.limma.deg.down.and.x)
```

```
## [1] 3542  103
```

****

V. PERFORMING the GENE SET ENRICHMENT ANALYSIS by using "enrichR" library :

```
For the following R code below, typically we start from a list of DEG :

for example, starting from these 2 lists of genes :
"analysis.LIMMA.genes.counts.Aph.only.DEG.and.DOWN.info.all.samples"
"analysis.LIMMA.genes.counts.Aph.only.DEG.and.UP.info.all.samples"
```

```
library("enrichR")

################################################################################################
################################################################################################
################################################################################################
################################################################################################

args <- commandArgs(TRUE)

FILE <- args[1]

name <- basename(FILE)

###### reading the FILE :

file <- read.delim(FILE, sep="\t", header=T, stringsAsFactors=T)

list_genes <- paste("", file$GENE_NAME, "", sep="")

######### FILE <- "analysis.LIMMA.genes.counts.Aph_KH7.only.DEG.and.DOWN.info.all.samples"
######### FILE <- "analysis.LIMMA.genes.counts.Aph_KH7.only.DEG.and.UP.info.all.samples"
####################################################################################################
####################################################################################################

dbs <- listEnrichrDbs()

### here chosing specific databases :
# dbs <- c("GO_Molecular_Function_2015",
#          "GO_Cellular_Component_2015",
#          "GO_Biological_Process_2015")

# enriched <- enrichr(list_genes, dbs)

dbs <- c("GO_Biological_Process_2018",
         "GO_Cellular_Component_2018",
         "GO_Molecular_Function_2018",
         "DSigDB",
         "Genome_Browser_PWMs",
         "TRANSFAC_and_JASPAR_PWMs",
         "ENCODE_TF_ChIP-seq_2014",
         "ENCODE_TF_ChIP-seq_2015",
         "ChEA_2016",
         "ENCODE_and_ChEA_Consensus_TFs_from_ChIP-X",
         "KEGG_2016",
         "WikiPathways_2016",
         "Reactome_2016",
         "BioCarta_2016",
         "Panther_2016",
         "NCI-Nature_2016",
         "OMIM_Disease",
         "OMIM_Expanded",
         "MSigDB_Computational",
         "MSigDB_Oncogenic_Signatures",
         "Chromosome_Location")

enriched <- enrichr(list_genes, dbs)

#######################################################################################################
#######################################################################################################
########## printing the SELECTED databases :

CATEGORY <- "GO_Biological_Process_2018"

results.db <- enriched[[CATEGORY]]

results.db.fdr <- subset(results.db, Adjusted.P.value < 0.05)

head(results.db.fdr[,1:6])
tail(results.db.fdr[,1:6])
dim(results.db.fdr[,1:6])

write.table(results.db.fdr,
            file=paste(basename(FILE), ".enrichR.fdr0.05.", CATEGORY, ".txt", sep=""),
            sep="\t", quote = FALSE,
            row.names = FALSE, col.names = TRUE)

#######################################################################################################
#######################################################################################################

CATEGORY <- "GO_Cellular_Component_2018"

results.db <- enriched[[CATEGORY]]

results.db.fdr <- subset(results.db, Adjusted.P.value < 0.05)

head(results.db.fdr[,1:6])
tail(results.db.fdr[,1:6])
dim(results.db.fdr[,1:6])

write.table(results.db.fdr,
            file=paste(basename(FILE), ".enrichR.fdr0.05.", CATEGORY, ".txt", sep=""),
            sep="\t", quote = FALSE,
            row.names = FALSE, col.names = TRUE)

#######################################################################################################
######################################################################################################

CATEGORY <- "GO_Molecular_Function_2018"
```

```
results.db <- enriched[[CATEGORY]]

results.db.fdr <- subset(results.db, Adjusted.P.value < 0.05)

head(results.db.fdr[,1:6])
tail(results.db.fdr[,1:6])
dim(results.db.fdr[,1:6])

write.table(results.db.fdr,
            file=paste(basename(FILE), ".enrichR.fdr0.05.", CATEGORY, ".txt", sep=""),
            sep="\t", quote = FALSE,
            row.names = FALSE, col.names = TRUE)

######################################################################################################
#####################################################################################################

CATEGORY <- "DSigDB"

results.db <- enriched[[CATEGORY]]

results.db.fdr <- subset(results.db, Adjusted.P.value < 0.05)

head(results.db.fdr[,1:6])
tail(results.db.fdr[,1:6])
dim(results.db.fdr[,1:6])

write.table(results.db.fdr,
            file=paste(basename(FILE), ".enrichR.fdr0.05.", CATEGORY, ".txt", sep=""),
            sep="\t", quote = FALSE,
            row.names = FALSE, col.names = TRUE)

######################################################################################################
#####################################################################################################

CATEGORY <- "Genome_Browser_PWMs"

results.db <- enriched[[CATEGORY]]

results.db.fdr <- subset(results.db, Adjusted.P.value < 0.05)

head(results.db.fdr[,1:6])
tail(results.db.fdr[,1:6])
dim(results.db.fdr[,1:6])

write.table(results.db.fdr,
            file=paste(basename(FILE), ".enrichR.fdr0.05.", CATEGORY, ".txt", sep=""),
            sep="\t", quote = FALSE,
            row.names = FALSE, col.names = TRUE)

######################################################################################################
#####################################################################################################

CATEGORY <- "TRANSFAC_and_JASPAR_PWMs"
```

```
results.db <- enriched[[CATEGORY]]

results.db.fdr <- subset(results.db, Adjusted.P.value < 0.05)

head(results.db.fdr[,1:6])
tail(results.db.fdr[,1:6])
dim(results.db.fdr[,1:6])

write.table(results.db.fdr,
            file=paste(basename(FILE), ".enrichR.fdr0.05.", CATEGORY, ".txt", sep=""),
            sep="\t", quote = FALSE,
            row.names = FALSE, col.names = TRUE)

#####################################################################################################
####################################################################################################

CATEGORY <- "ENCODE_TF_ChIP-seq_2014"

results.db <- enriched[[CATEGORY]]

results.db.fdr <- subset(results.db, Adjusted.P.value < 0.05)

head(results.db.fdr[,1:6])
tail(results.db.fdr[,1:6])
dim(results.db.fdr[,1:6])

write.table(results.db.fdr,
            file=paste(basename(FILE), ".enrichR.fdr0.05.", CATEGORY, ".txt", sep=""),
            sep="\t", quote = FALSE,
            row.names = FALSE, col.names = TRUE)

#####################################################################################################
####################################################################################################

CATEGORY <- "ENCODE_TF_ChIP-seq_2015"

results.db <- enriched[[CATEGORY]]

results.db.fdr <- subset(results.db, Adjusted.P.value < 0.05)

head(results.db.fdr[,1:6])
tail(results.db.fdr[,1:6])
dim(results.db.fdr[,1:6])

write.table(results.db.fdr,
            file=paste(basename(FILE), ".enrichR.fdr0.05.", CATEGORY, ".txt", sep=""),
            sep="\t", quote = FALSE,
            row.names = FALSE, col.names = TRUE)

#####################################################################################################
####################################################################################################

CATEGORY <- "ChEA_2016"
```

```
results.db <- enriched[[CATEGORY]]

results.db.fdr <- subset(results.db, Adjusted.P.value < 0.05)

head(results.db.fdr[,1:6])
tail(results.db.fdr[,1:6])
dim(results.db.fdr[,1:6])

write.table(results.db.fdr,
            file=paste(basename(FILE), ".enrichR.fdr0.05.", CATEGORY, ".txt", sep=""),
            sep="\t", quote = FALSE,
            row.names = FALSE, col.names = TRUE)

#######################################################################################################
######################################################################################################

CATEGORY <- "ENCODE_and_ChEA_Consensus_TFs_from_ChIP-X"

results.db <- enriched[[CATEGORY]]

results.db.fdr <- subset(results.db, Adjusted.P.value < 0.05)

head(results.db.fdr[,1:6])
tail(results.db.fdr[,1:6])
dim(results.db.fdr[,1:6])

write.table(results.db.fdr,
            file=paste(basename(FILE), ".enrichR.fdr0.05.", CATEGORY, ".txt", sep=""),
            sep="\t", quote = FALSE,
            row.names = FALSE, col.names = TRUE)

#######################################################################################################
######################################################################################################

CATEGORY <- "KEGG_2016"

results.db <- enriched[[CATEGORY]]

results.db.fdr <- subset(results.db, Adjusted.P.value < 0.05)

head(results.db.fdr[,1:6])
tail(results.db.fdr[,1:6])
dim(results.db.fdr[,1:6])

write.table(results.db.fdr,
            file=paste(basename(FILE), ".enrichR.fdr0.05.", CATEGORY, ".txt", sep=""),
            sep="\t", quote = FALSE,
            row.names = FALSE, col.names = TRUE)

#######################################################################################################
######################################################################################################

CATEGORY <- "WikiPathways_2016"
```

```
results.db <- enriched[[CATEGORY]]

results.db.fdr <- subset(results.db, Adjusted.P.value < 0.05)

head(results.db.fdr[,1:6])
tail(results.db.fdr[,1:6])
dim(results.db.fdr[,1:6])

write.table(results.db.fdr,
            file=paste(basename(FILE), ".enrichR.fdr0.05.", CATEGORY, ".txt", sep=""),
            sep="\t", quote = FALSE,
            row.names = FALSE, col.names = TRUE)

######################################################################################################
#####################################################################################################

CATEGORY <- "Reactome_2016"

results.db <- enriched[[CATEGORY]]

results.db.fdr <- subset(results.db, Adjusted.P.value < 0.05)

head(results.db.fdr[,1:6])
tail(results.db.fdr[,1:6])
dim(results.db.fdr[,1:6])

write.table(results.db.fdr,
            file=paste(basename(FILE), ".enrichR.fdr0.05.", CATEGORY, ".txt", sep=""),
            sep="\t", quote = FALSE,
            row.names = FALSE, col.names = TRUE)

######################################################################################################
#####################################################################################################

CATEGORY <- "BioCarta_2016"

results.db <- enriched[[CATEGORY]]

results.db.fdr <- subset(results.db, Adjusted.P.value < 0.05)

head(results.db.fdr[,1:6])
tail(results.db.fdr[,1:6])
dim(results.db.fdr[,1:6])

write.table(results.db.fdr,
            file=paste(basename(FILE), ".enrichR.fdr0.05.", CATEGORY, ".txt", sep=""),
            sep="\t", quote = FALSE,
            row.names = FALSE, col.names = TRUE)

######################################################################################################
#####################################################################################################

CATEGORY <- "Panther_2016"
```

```
results.db <- enriched[[CATEGORY]]

results.db.fdr <- subset(results.db, Adjusted.P.value < 0.05)

head(results.db.fdr[,1:6])
tail(results.db.fdr[,1:6])
dim(results.db.fdr[,1:6])

write.table(results.db.fdr,
            file=paste(basename(FILE), ".enrichR.fdr0.05.", CATEGORY, ".txt", sep=""),
            sep="\t", quote = FALSE,
            row.names = FALSE, col.names = TRUE)

######################################################################################################
#####################################################################################################

CATEGORY <- "NCI-Nature_2016"

results.db <- enriched[[CATEGORY]]

results.db.fdr <- subset(results.db, Adjusted.P.value < 0.05)

head(results.db.fdr[,1:6])
tail(results.db.fdr[,1:6])
dim(results.db.fdr[,1:6])

write.table(results.db.fdr,
            file=paste(basename(FILE), ".enrichR.fdr0.05.", CATEGORY, ".txt", sep=""),
            sep="\t", quote = FALSE,
            row.names = FALSE, col.names = TRUE)

######################################################################################################
#####################################################################################################

CATEGORY <- "OMIM_Disease"

results.db <- enriched[[CATEGORY]]

results.db.fdr <- subset(results.db, Adjusted.P.value < 0.05)

head(results.db.fdr[,1:6])
tail(results.db.fdr[,1:6])
dim(results.db.fdr[,1:6])

write.table(results.db.fdr,
            file=paste(basename(FILE), ".enrichR.fdr0.05.", CATEGORY, ".txt", sep=""),
            sep="\t", quote = FALSE,
            row.names = FALSE, col.names = TRUE)

######################################################################################################
#####################################################################################################

CATEGORY <- "OMIM_Expanded"
```

```
results.db <- enriched[[CATEGORY]]

results.db.fdr <- subset(results.db, Adjusted.P.value < 0.05)

head(results.db.fdr[,1:6])
tail(results.db.fdr[,1:6])
dim(results.db.fdr[,1:6])

write.table(results.db.fdr,
            file=paste(basename(FILE), ".enrichR.fdr0.05.", CATEGORY, ".txt", sep=""),
            sep="\t", quote = FALSE,
            row.names = FALSE, col.names = TRUE)

######################################################################################################
#####################################################################################################

CATEGORY <- "MSigDB_Computational"

results.db <- enriched[[CATEGORY]]

results.db.fdr <- subset(results.db, Adjusted.P.value < 0.05)

head(results.db.fdr[,1:6])
tail(results.db.fdr[,1:6])
dim(results.db.fdr[,1:6])

write.table(results.db.fdr,
            file=paste(basename(FILE), ".enrichR.fdr0.05.", CATEGORY, ".txt", sep=""),
            sep="\t", quote = FALSE,
            row.names = FALSE, col.names = TRUE)

######################################################################################################
#####################################################################################################

CATEGORY <- "MSigDB_Oncogenic_Signatures"

results.db <- enriched[[CATEGORY]]

results.db.fdr <- subset(results.db, Adjusted.P.value < 0.05)

head(results.db.fdr[,1:6])
tail(results.db.fdr[,1:6])
dim(results.db.fdr[,1:6])

write.table(results.db.fdr,
            file=paste(basename(FILE), ".enrichR.fdr0.05.", CATEGORY, ".txt", sep=""),
            sep="\t", quote = FALSE,
            row.names = FALSE, col.names = TRUE)

######################################################################################################
#####################################################################################################

CATEGORY <- "Chromosome_Location"
```

```
results.db <- enriched[[CATEGORY]]

results.db.fdr <- subset(results.db, Adjusted.P.value < 0.05)

head(results.db.fdr[,1:6])
tail(results.db.fdr[,1:6])
dim(results.db.fdr[,1:6])

write.table(results.db.fdr,
            file=paste(basename(FILE), ".enrichR.fdr0.05.", CATEGORY, ".txt", sep=""),
            sep="\t", quote = FALSE,
            row.names = FALSE, col.names = TRUE)

######################################################################################################
######################################################################################################

######################################################################################################
######################################################################################################

# CATEGORY <- ""

# results.db <- enriched[[CATEGORY]]

# results.db.fdr <- subset(results.db, Adjusted.P.value < 0.05)

# head(results.db.fdr[,1:6])
# tail(results.db.fdr[,1:6])
# dim(results.db.fdr[,1:6])

# write.table(results.db.fdr,
#             file=paste(basename(FILE), ".enrichR.fdr0.05.", CATEGORY, ".txt", sep=""),
#             sep="\t", quote = FALSE,
#             row.names = FALSE, col.names = TRUE)

######################################################################################################
#####################################################################################################

#                                 Genome_Browser_PWMs      615        13362
#                            TRANSFAC_and_JASPAR_PWMs      326        27884
#                          Transcription_Factor_PPIs      290         6002
#                                           ChEA_2013      353        47172
#                    Drug_Perturbations_from_GEO_2014      701        47107
#                             ENCODE_TF_ChIP-seq_2014      498        21493
#                                       BioCarta_2013      249         1295
#                                       Reactome_2013       78         3185
#                                   WikiPathways_2013      199         2854
#                Disease_Signatures_from_GEO_up_2014      142        15057
#                                          KEGG_2013      200         4128
#                         TF-LOF_Expression_from_GEO      269        34061
#                                TargetScan_microRNA      222         7504
#                                   PPI_Hub_Proteins      385        16399
#                         GO_Molecular_Function_2015     1136        12753
#                                          GeneSigDB     2139        23726
#                               Chromosome_Location      386        32740
#                                  Human_Gene_Atlas       84        13373
#                                  Mouse_Gene_Atlas       96        19270
#                        GO_Cellular_Component_2015      641        13236
#                        GO_Biological_Process_2015     5192        14264
#                          Human_Phenotype_Ontology     1779         3096
#                   Epigenomics_Roadmap_HM_ChIP-seq      383        22288
#                                          KEA_2013      474         4533
#                 NURSA_Human_Endogenous_Complexome     1796        10231
#                                             CORUM     1658         2741
#                           SILAC_Phosphoproteomics       84         5655
#                   MGI_Mammalian_Phenotype_Level_3       71        10406
#                   MGI_Mammalian_Phenotype_Level_4      476        10493
#                                       Old_CMAP_up     6100        11251
#                                     Old_CMAP_down     6100         8695
#                                      OMIM_Disease       90         1759
#                                     OMIM_Expanded      187         2178
#                                         VirusMINT       85          851
#                              MSigDB_Computational      858        10061
#                       MSigDB_Oncogenic_Signatures      189        11250
#             Disease_Signatures_from_GEO_down_2014      142        15406
#                   Virus_Perturbations_from_GEO_up      323        17711
#                 Virus_Perturbations_from_GEO_down      323        17576
#                     Cancer_Cell_Line_Encyclopedia      967        15797
#                          NCI-60_Cancer_Cell_Lines       93        12232
#       Tissue_Protein_Expression_from_ProteomicsDB      207        13572
#  Tissue_Protein_Expression_from_Human_Proteome_Map       30         6454
#                                  HMDB_Metabolites     3906         3723
#                             Pfam_InterPro_Domains      311         7588
#                        GO_Biological_Process_2013      941         7682
#                        GO_Cellular_Component_2013      205         7324
#                        GO_Molecular_Function_2013      402         8469
#                              Allen_Brain_Atlas_up     2192        13121
#                           ENCODE_TF_ChIP-seq_2015      816        26382
#                 ENCODE_Histone_Modifications_2015      412        29065
#                 Phosphatase_Substrates_from_DEPOD       59          280
#                            Allen_Brain_Atlas_down     2192        13877
#                 ENCODE_Histone_Modifications_2013      109        15852
#                         Achilles_fitness_increase      216         4320
#                         Achilles_fitness_decrease      216         4271
#                      MGI_Mammalian_Phenotype_2013      476        10496
#                                     BioCarta_2015      239         1678
#                                     HumanCyc_2015      125          756
#                                         KEGG_2015      179         3800
#                                   NCI-Nature_2015      209         2541
#                                      Panther_2015      104         1918
#                                 WikiPathways_2015      404         5863
#                                     Reactome_2015     1389         6768
#                                            ESCAPE      315        25651
#                                        HomoloGene       12        19129
#               Disease_Perturbations_from_GEO_down      839        23939
#                 Disease_Perturbations_from_GEO_up      839        23561
#                  Drug_Perturbations_from_GEO_down      906        23877
#                   Genes_Associated_with_NIH_Grants    32876        15886
#                     Drug_Perturbations_from_GEO_up      906        24350
#                                           KEA_2015      428         3102
#              Single_Gene_Perturbations_from_GEO_up     2460        31132
#            Single_Gene_Perturbations_from_GEO_down     2460        30832
#                                          ChEA_2015      395        48230
#                                              dbGaP      345         5613
#                           LINCS_L1000_Chem_Pert_up    33132         9559
#                         LINCS_L1000_Chem_Pert_down    33132         9448
#   GTEx_Tissue_Sample_Gene_Expression_Profiles_down     2918        16725
#     GTEx_Tissue_Sample_Gene_Expression_Profiles_up     2918        19249
#                 Ligand_Perturbations_from_GEO_down      261        15090
#                  Aging_Perturbations_from_GEO_down      286        16129
#                    Aging_Perturbations_from_GEO_up      286        15309
#                   Ligand_Perturbations_from_GEO_up      261        15103
#                   MCF7_Perturbations_from_GEO_down      401        15022
#                     MCF7_Perturbations_from_GEO_up      401        15676
#                Microbe_Perturbations_from_GEO_down      312        15854
#                  Microbe_Perturbations_from_GEO_up      312        15015
#              LINCS_L1000_Ligand_Perturbations_down       96         3788
#                LINCS_L1000_Ligand_Perturbations_up       96         3357
#              LINCS_L1000_Kinase_Perturbations_down     3644        12668
#                LINCS_L1000_Kinase_Perturbations_up     3644        12638
#                                      Reactome_2016     1530         8973
#                                          KEGG_2016      293         7010
#                                  WikiPathways_2016      437         5966
#          ENCODE_and_ChEA_Consensus_TFs_from_ChIP-X      104        15562
#                 Kinase_Perturbations_from_GEO_down      285        17850
#                   Kinase_Perturbations_from_GEO_up      285        17660
#                                      BioCarta_2016      237         1348
#                                     HumanCyc_2016      152          934
#                                   NCI-Nature_2016      209         2541
#                                      Panther_2016      112         2041
#                                        DrugMatrix     7876         5209
#                                         ChEA_2016      645        49238
#                                             huMAP      995         2243
#                                    Jensen_TISSUES     1842        19586
# RNA-Seq_Disease_Gene_and_Drug_Signatures_from_GEO     1302        22440
#                      MGI_Mammalian_Phenotype_2017     5231         8184
#                               Jensen_COMPARTMENTS     2283        18329
#                                   Jensen_DISEASES     1811        15755
#                                      BioPlex_2017     3915        10271
#                        GO_Cellular_Component_2017      636        10427
#                        GO_Molecular_Function_2017      972        10601
#                        GO_Biological_Process_2017     3166        13822
#                       GO_Cellular_Component_2017b      816         8002
#                       GO_Molecular_Function_2017b     3271        10089
#                       GO_Biological_Process_2017b    10125        13247
#                                    ARCHS4_Tissues      108        21809
#                                 ARCHS4_Cell-lines      125        23601
#                                  ARCHS4_IDG_Coexp      352        20883
#                              ARCHS4_Kinases_Coexp      498        19612
#                                  ARCHS4_TFs_Coexp     1724        25983
#                           SysMyo_Muscle_Gene_Sets     1135        19500
#                                   miRTarBase_2017     3240        14893
#                          TargetScan_microRNA_2017      683        17598
#              Enrichr_Libraries_Most_Popular_Genes      121         5902
#           Enrichr_Submissions_TF-Gene_Coocurrence     1722        12486
#        Data_Acquisition_Method_Most_Popular_Genes       12         1073
#                                            DSigDB     4026        19513
#                        GO_Biological_Process_2018     5103        14433
#                        GO_Cellular_Component_2018      446         8655
#                        GO_Molecular_Function_2018     1151        114

###################################################################################################
####################################################################################################
######### SELECTED DATABASES : ################

# GO_Biological_Process_2018
# GO_Cellular_Component_2018
# GO_Molecular_Function_2018
# DSigDB
# Genome_Browser_PWMs
# TRANSFAC_and_JASPAR_PWMs
# ENCODE_TF_ChIP-seq_2014
# ENCODE_TF_ChIP-seq_2015
# ChEA_2016
# ENCODE_and_ChEA_Consensus_TFs_from_ChIP-X
# KEGG_2016
# WikiPathways_2016
# Reactome_2016
# BioCarta_2016
# Panther_2016
# NCI-Nature_2016
# OMIM_Disease
# OMIM_Expanded
# MSigDB_Computational
# MSigDB_Oncogenic_Signatures
# Chromosome_Location
```

****

VI. DATA VISUALIZATION : PCA and MDS

1. INITIALLY preparing a large data frame with all the EXPRESSION DATA :

```
###### Here reusing an OLD PIECE of R CODE :

NAME <- "z.analysis.results"

#################################################################################################
##################### here to upload the data where we did integrate all the files with ALL GENES
###### the results from RSEM
###### the results from LIMMA

genes.expression.large <- read.delim("analysis.LIMMA.integrating.all.samples.all.genes.in.4.STEPS",
                     sep="\t", header=T, stringsAsFactors=F)

# head(genes.expression.large)
dim(genes.expression.large)
```

```
## [1] 58381    95
```

```
################ here we would have to make a special ROWNAME,
### as some genes are present in multiple isoforms ..

genes.expression.large$ID <- rownames(genes.expression.large)
genes.expression.large$GENE_NAME_ID <- paste(genes.expression.large$GENE_NAME,
                                             genes.expression.large$ID, sep=":")

# head(genes.expression.large)
dim(genes.expression.large)
```

```
## [1] 58381    95
```

```
############################################################################################
############################################################################################
############################################################################################
############################################################################################

###### transforming the DATA FRAME into a DATA TABLE :

genes.expression.large.dt <- as.data.table(genes.expression.large)

# head(genes.expression.large.dt)
dim(genes.expression.large.dt)
```

```
## [1] 58381    95
```

```
############################################################################################
############################################################################################
############################################################################################
############################################################################################

###### here selecting the following fields below in order to make
###### the PCA plots
###### the MDS plots
###### the BOXPLOTS
```

```r
###### the SCATTER PLOTS
###### the VOLCANO PLOTS
###### the HEATMAPS

############################################################################################
############################################################################################
############################################################################################
############################################################################################

############################################################################
################# making a DATAFRAME of GENES COUNTS :

genes.expression.large.counts <- subset(genes.expression.large,
                                               select=c("GENE_NAME_ID",
                                               "DMSO1_lane1.count", "DMSO1_lane2.count",
                                               "DMSO2_lane1.count", "DMSO2_lane2.count",
                                               "DMSO3_lane1.count", "DMSO3_lane2.count",
                                               "Aph1.count", "Aph2.count", "Aph3.count",
                                               "Aph_KH7_1.count","Aph_KH7_2.count","Aph_KH7_3.count",
                                               "KH7_1.count", "KH7_2.count", "KH7_3.count",
                                               "Noc_1.count", "Noc_2.count", "Noc_3.count"),
                                                na.rm = TRUE)

rownames(genes.expression.large.counts) <- genes.expression.large.counts$GENE_NAME_ID
genes.expression.large.counts <- genes.expression.large.counts[,-1]

# head(genes.expression.large.counts)
dim(genes.expression.large.counts)
```

```
## [1] 58381    18
```

```r
############################################################################
############################################################################
################# making a DATAFRAME based on TPM :

genes.expression.large.tpm <- subset(genes.expression.large,
                                     select=c("GENE_NAME_ID",
                                     "DMSO1_lane1.TPM", "DMSO1_lane2.TPM",
                                     "DMSO2_lane1.TPM", "DMSO2_lane2.TPM",
                                     "DMSO3_lane1.TPM", "DMSO3_lane2.TPM",
                                     "Aph1.TPM", "Aph2.TPM", "Aph3.TPM",
                                     "Aph_KH7_1.TPM","Aph_KH7_2.TPM","Aph_KH7_3.TPM",
                                     "KH7_1.TPM", "KH7_2.TPM", "KH7_3.TPM",
                                     "Noc_1.TPM", "Noc_2.TPM", "Noc_3.TPM" ),
                                      na.rm = TRUE)

rownames(genes.expression.large.tpm) <- genes.expression.large.tpm$GENE_NAME_ID
genes.expression.large.tpm <- genes.expression.large.tpm[,-1]

# head(genes.expression.large.tpm)
dim(genes.expression.large.tpm)
```

```
## [1] 58381    18
```

```r
###################################################################################
#################################################### to look at the MEDIAN : it is 0 !!!1

#############################################################################
################# making a DATAFRAME based on FPKM :

genes.expression.large.fpkm <- subset(genes.expression.large,
                                      select=c("GENE_NAME_ID",
                                      "DMSO1_lane1.FPKM", "DMSO1_lane2.FPKM",
                                      "DMSO2_lane1.FPKM", "DMSO2_lane2.FPKM",
                                      "DMSO3_lane1.FPKM", "DMSO3_lane2.FPKM",
                                      "Aph1.FPKM", "Aph2.FPKM", "Aph3.FPKM",
                                      "Aph_KH7_1.FPKM","Aph_KH7_2.FPKM","Aph_KH7_3.FPKM",
                                      "KH7_1.FPKM", "KH7_2.FPKM", "KH7_3.FPKM",
                                      "Noc_1.FPKM", "Noc_2.FPKM", "Noc_3.FPKM" ),
                                       na.rm = TRUE)

rownames(genes.expression.large.fpkm) <- genes.expression.large.fpkm$GENE_NAME_ID
genes.expression.large.fpkm <- genes.expression.large.fpkm[,-1]

# head(genes.expression.large.fpkm)
dim(genes.expression.large.fpkm)
```

```
## [1] 58381    18
```

```r
###################################################################################
#################################################### to look at the MEDIAN : hmmm ... it is 0 !!!

#########################################################################################################
#########################################################################################################

### so we will have somehow to exclude the genes that are NA, <NA>
### or use a smaller dataframe that we have obtained with the DATA.TABLE

#########################################################################################################
#########################################################################################################
#########################################################################################################
#########################################################################################################
#########################################################################################################
```

```r
##################################################################################################
##################### here to upload the data where we did integrate all the files with ALL GENES
###### the results from RSEM
###### the results from LIMMA

genes.expression.small <- read.delim("analysis.LIMMA.integrating.all.samples.with.data.table",
                                      sep="\t", header=T, stringsAsFactors=F)

# head(genes.expression.small)
dim(genes.expression.small)
```

```
## [1] 13247    95
```

```r
################ here we would have to make a special ROWNAME,
### as some genes are present in multiple isoforms ..
```

```r
genes.expression.small$ID <- rownames(genes.expression.small)
genes.expression.small$GENE_NAME_ID <- paste(genes.expression.small$GENE_NAME,
                                             genes.expression.small$ID, sep=":")

# head(genes.expression.small)
dim(genes.expression.small)
```

```
## [1] 13247    95
```

```r
############################################################################################
############################################################################################
############################################################################################
############################################################################################

### for some reasons that I do not know yet, there are some lines with NA in the file
### to exclude these lines from the files with TPM or FPKM, or here below :

genes.expression.small.NA <- subset(genes.expression.small, is.na(CHR))
dim(genes.expression.small.NA)     ### 291  95
```

```
## [1] 291  95
```

```r
genes.expression.small.non.NA <- subset(genes.expression.small, !is.na(CHR))
dim(genes.expression.small.non.NA) ### 12956    95
```

```
## [1] 12956    95
```

```r
genes.expression.small <- genes.expression.small.non.NA
dim(genes.expression.small)
```

```
## [1] 12956    95
```

```r
############################################################################################
############################################################################################
############################################################################################
############################################################################################

###### transforming the DATA FRAME into a DATA TABLE :

genes.expression.small.dt <- as.data.table(genes.expression.small)

# head(genes.expression.small.dt)
dim(genes.expression.small.dt)
```

```
## [1] 12956    95
```

```r
############################################################################################
############################################################################################
############################################################################################
############################################################################################
############################################################################################
```

```r
############################################################################
################# making a DATAFRAME of GENES COUNTS :

genes.expression.small.counts <- subset(genes.expression.small,
                                               select=c("GENE_NAME_ID",
                                               "DMSO1_lane1.count", "DMSO1_lane2.count",
                                               "DMSO2_lane1.count", "DMSO2_lane2.count",
                                               "DMSO3_lane1.count", "DMSO3_lane2.count",
                                               "Aph1.count", "Aph2.count", "Aph3.count",
                                               "Aph_KH7_1.count","Aph_KH7_2.count","Aph_KH7_3.count",
                                               "KH7_1.count", "KH7_2.count", "KH7_3.count",
                                               "Noc_1.count", "Noc_2.count", "Noc_3.count"),
                                                na.rm = TRUE)

rownames(genes.expression.small.counts) <- genes.expression.small.counts$GENE_NAME_ID
genes.expression.small.counts <- genes.expression.small.counts[,-1]

# head(genes.expression.small.counts)
dim(genes.expression.small.counts)
```

```
## [1] 12956    18
```

```
##################################################################################
##################################################################################
############################################################################
############################################################################
################# making a DATAFRAME based on TPM :

genes.expression.small.tpm <- subset(genes.expression.small,
                        select=c("GENE_NAME_ID",
                       "DMSO1_lane1.TPM", "DMSO1_lane2.TPM",
                       "DMSO2_lane1.TPM", "DMSO2_lane2.TPM",
                       "DMSO3_lane1.TPM", "DMSO3_lane2.TPM",
                       "Aph1.TPM", "Aph2.TPM", "Aph3.TPM",
                       "Aph_KH7_1.TPM","Aph_KH7_2.TPM","Aph_KH7_3.TPM",
                       "KH7_1.TPM", "KH7_2.TPM", "KH7_3.TPM",
                       "Noc_1.TPM", "Noc_2.TPM", "Noc_3.TPM" ),
                        na.rm = TRUE)

rownames(genes.expression.small.tpm) <- genes.expression.small.tpm$GENE_NAME_ID
genes.expression.small.tpm <- genes.expression.small.tpm[,-1]

# head(genes.expression.small.tpm)
dim(genes.expression.small.tpm)
```

```
## [1] 12956    18
```

```
##################################################################################
##################################################################################
#################################################### to look at the MEDIAN :

median(genes.expression.small.tpm[,1],na.rm=T)
```

```
## [1] 18.49
```

```
median(genes.expression.small.tpm[,2],na.rm=T)
```

```
## [1] 18.53
```

```
median(genes.expression.small.tpm[,3],na.rm=T)
```

```
## [1] 17.855
```

```
median(genes.expression.small.tpm[,4],na.rm=T)
```

```
## [1] 17.77
```

```
median(genes.expression.small.tpm[,5],na.rm=T)
```

```
## [1] 17.72
```

```
median(genes.expression.small.tpm[,6],na.rm=T)
```

```
## [1] 17.75
```

```
median(genes.expression.small.tpm[,7],na.rm=T)
```

```
## [1] 19.11
```

```
median(genes.expression.small.tpm[,8],na.rm=T)
```

```
## [1] 19.79
```

```
median(genes.expression.small.tpm[,9],na.rm=T)
```

```
## [1] 19.455
```

```
median(genes.expression.small.tpm[,10],na.rm=T)
```

```
## [1] 15.56
```

```
median(genes.expression.small.tpm[,11],na.rm=T)
```

```
## [1] 16.38
```

```
median(genes.expression.small.tpm[,12],na.rm=T)
```

```
## [1] 15.885
```

```
median(genes.expression.small.tpm[,13],na.rm=T)
```

```
## [1] 17.945
```

```
median(genes.expression.small.tpm[,14],na.rm=T)
```

```
## [1] 17.865
```

```
median(genes.expression.small.tpm[,15],na.rm=T)
```

```
## [1] 17.66
```

```
median(genes.expression.small.tpm[,16],na.rm=T)
```

```
## [1] 17.445
```

```
median(genes.expression.small.tpm[,17],na.rm=T)
```

```
## [1] 17.31
```

```
median(genes.expression.small.tpm[,18],na.rm=T)
```

```
## [1] 17.19
```

```
###### making the BOXPLOTS for the genes

pdf(paste(NAME, ".boxplot.TPM.pdf", sep=""))
    par(las=2)
    par(mar=c(8,4,2,2))

    boxplot(genes.expression.small.tpm,
            ylim=c(0,60),
            col=c(rep("red",6), rep("orange",3), rep("green",3),
                            rep("blue",3), rep("violet",3)),
            ylab="TPM",
            main="TPM values of ~13240 genes",
            cex.main=0.8, cex.lab=0.8)
dev.off()
```

```
## pdf
##   2
```

```
####### printing the FILE with TPM values :

write.table(genes.expression.small.tpm,
            file=paste(NAME, ".file.TPM.txt", sep=""),
            sep="\t")

##################################################################################
##################################################################################
############################################################################
############################################################################
################# making a DATAFRAME based on FPKM :

genes.expression.small.fpkm <- subset(genes.expression.small,
                        select=c("GENE_NAME_ID",
                       "DMSO1_lane1.FPKM", "DMSO1_lane2.FPKM",
                       "DMSO2_lane1.FPKM", "DMSO2_lane2.FPKM",
                       "DMSO3_lane1.FPKM", "DMSO3_lane2.FPKM",
                       "Aph1.FPKM", "Aph2.FPKM", "Aph3.FPKM",
                       "Aph_KH7_1.FPKM","Aph_KH7_2.FPKM","Aph_KH7_3.FPKM",
                       "KH7_1.FPKM", "KH7_2.FPKM", "KH7_3.FPKM",
                       "Noc_1.FPKM", "Noc_2.FPKM", "Noc_3.FPKM" ),
                        na.rm = TRUE)

rownames(genes.expression.small.fpkm) <- genes.expression.small.fpkm$GENE_NAME_ID
genes.expression.small.fpkm <- genes.expression.small.fpkm[,-1]

# head(genes.expression.small.fpkm)
dim(genes.expression.small.fpkm)
```

```
## [1] 12956    18
```

```
##################################################################################
##################################################################################
#################################################### to look at the MEDIAN :

median(genes.expression.small.fpkm[,1],na.rm=T)
```

```
## [1] 10.185
```

```
median(genes.expression.small.fpkm[,2],na.rm=T)
```

```
## [1] 10.2
```

```r
median(genes.expression.small.fpkm[,3],na.rm=T)
```

```
## [1] 10.26
```

```r
median(genes.expression.small.fpkm[,4],na.rm=T)
```

```
## [1] 10.19
```

```r
median(genes.expression.small.fpkm[,5],na.rm=T)
```

```
## [1] 10.21
```

```r
median(genes.expression.small.fpkm[,6],na.rm=T)
```

```
## [1] 10.21
```

```r
median(genes.expression.small.fpkm[,7],na.rm=T)
```

```
## [1] 10.63
```

```r
median(genes.expression.small.fpkm[,8],na.rm=T)
```

```
## [1] 10.77
```

```r
median(genes.expression.small.fpkm[,9],na.rm=T)
```

```
## [1] 10.62
```

```r
median(genes.expression.small.fpkm[,10],na.rm=T)
```

```
## [1] 8.87
```

```r
median(genes.expression.small.fpkm[,11],na.rm=T)
```

```
## [1] 9
```

```r
median(genes.expression.small.fpkm[,12],na.rm=T)
```

```
## [1] 8.855
```

```r
median(genes.expression.small.fpkm[,13],na.rm=T)
```

```
## [1] 9.915
```

```r
median(genes.expression.small.fpkm[,14],na.rm=T)
```

```
## [1] 9.945
```

```r
median(genes.expression.small.fpkm[,15],na.rm=T)
```

```
## [1] 9.96
```

```r
median(genes.expression.small.fpkm[,16],na.rm=T)
```

```
## [1] 9.55
```

```r
median(genes.expression.small.fpkm[,17],na.rm=T)
```

```
## [1] 9.55
```

```r
median(genes.expression.small.fpkm[,18],na.rm=T)
```

```
## [1] 9.54
```

```
###### making the BOXPLOTS for the genes :

pdf(paste(NAME, ".boxplot.FPKM.pdf", sep=""))
    par(las=2)
    par(mar=c(8,4,2,2))

    boxplot(genes.expression.small.fpkm,
            ylim=c(0,60),
            col=c(rep("red",6), rep("orange",3), rep("green",3),
                            rep("blue",3), rep("violet",3)),
            ylab="FPKM",
            main="FPKM values of ~13240 genes",
            cex.main=0.8, cex.lab=0.8)
dev.off()
```

```
## pdf
##   2
```

```
####### printing the FILE with FPKM values :

write.table(genes.expression.small.fpkm,
            file=paste(NAME, ".file.FPKM.txt", sep=""),
            sep="\t")
```

****

2. PCA ANALYSIS :

```
library(scatterplot3d)

###### THE PCA ANALYSIS :

# colnames(genes.expression.small.fpkm)
# [1] "DMSO1_lane1.FPKM" "DMSO1_lane2.FPKM" "DMSO2_lane1.FPKM" "DMSO2_lane2.FPKM"
# [5] "DMSO3_lane1.FPKM" "DMSO3_lane2.FPKM" "Aph1.FPKM"        "Aph2.FPKM"
# [9] "Aph3.FPKM"        "Aph_KH7_1.FPKM"   "Aph_KH7_2.FPKM"   "Aph_KH7_3.FPKM"
# [13] "KH7_1.FPKM"      "KH7_2.FPKM"       "KH7_3.FPKM"       "Noc_1.FPKM"
# [17] "Noc_2.FPKM"      "Noc_3.FPKM"

group <- factor( c(rep("DMSO",6), rep("Aph", 3), rep("Aph_KH7", 3),
                   rep("KH7", 3), rep("Noc", 3)) )

##### library(scatterplot3d)

pca <- prcomp(t(genes.expression.small.fpkm))

###### plotting the pca$x :

# s3D <-
pdf(paste(NAME, ".PCA.display.in.3D.pdf", sep=""))
       scatterplot3d(pca$x[,1:3],
                            color = c(rep("red",6), rep("orange",3), rep("green",3),
                                      rep("blue",3), rep("violet",3)),
                            pch=18,
                            main="PCA analysis of ~13000 genes",
                            grid=TRUE,
                            box=TRUE)
dev.off()
```

```
## pdf
##   2
```

```
###### to get some inspiration from :
# http://www.sthda.com/english/wiki/scatterplot3d-3d-graphics-r-software-and-data-visualization

# legend(s3D$xyz.convert(7.5, 3, 4.5),
#       legend = row.names(pca$x[,1:3]),
#       color = c(rep("red",6), rep("orange",3), rep("green",3),
#                                    rep("blue",3), rep("violet",3)),
#       pch = 16)

######################################################################################
######################################################################################
######################################################################################
######################################################################################

pca.df <- data.frame(PCA1=pca$x[,1],
                     PCA2=pca$x[,2],
                     PCA3=pca$x[,3],
                     group=group)
```

```r
## Here we are plotting PCA1 vs PCA2

pdf(paste(NAME, ".PCA.display.PC1.vs.PC2.pdf", sep=""))
ggplot(pca.df,
       aes(x=PCA1, y=PCA2, color=group, label=rownames(pca.df))) +
       geom_point(size=3) +
       # geom_text(col='black', size=4) +
       theme_bw() +
       theme(legend.position="top",
             legend.title=element_blank(),
             legend.key = element_blank()) +
             labs(x="PC1", y="PC2") +
        ggtitle("PCA analysis : PC1 vs PC2")
dev.off()
```

```
## pdf
##   2
```

```r
## Here we are plotting PCA1 vs PCA3

pdf(paste(NAME, ".PCA.display.PC1.vs.PC3.pdf", sep=""))
ggplot(pca.df,
       aes(x=PCA1, y=PCA3, color=group, label=rownames(pca.df))) +
       geom_point(size=3) +
       # geom_text(col='black', size=4) +
       theme_bw() +
       theme(legend.position="top",
             legend.title=element_blank(),
             legend.key = element_blank()) +
             labs(x="PC1", y="PC3") +
        ggtitle("PCA analysis : PC1 vs PC3")
dev.off()
```

```
## pdf
##   2
```

```r
## Here we are plotting PCA2 vs PCA3

pdf(paste(NAME, ".PCA.display.PC2.vs.PC3.pdf", sep=""))
ggplot(pca.df,
       aes(x=PCA2, y=PCA3, color=group, label=rownames(pca.df))) +
       geom_point(size=3) +
       # geom_text(col='black', size=4) +
       theme_bw() +
       theme(legend.position="top",
             legend.title=element_blank(),
             legend.key = element_blank()) +
             labs(x="PC2", y="PC3") +
        ggtitle("PCA analysis : PC2 vs PC3")
dev.off()
```

```
## pdf
##   2
```

****

3. MDS ANALYSIS :

```
library(scatterplot3d)

###### THE MDS ANALYSIS :

group <- factor( c(rep("DMSO",6), rep("Aph", 3), rep("Aph_KH7", 3),
                   rep("KH7", 3), rep("Noc", 3)) )

### We can use the function plotMDS from LIMMA or we can use the function cmdscale :
### mds <- plotMDS(genes.expression.small.fpkm)
### mds.df <- data.frame(MDSx=mds$x, MDSy=mds$y, group=group)

mds <- cmdscale(dist(t(genes.expression.small.fpkm)))
mds.df <- data.frame(MDSx=mds[,1], MDSy=mds[,2], group=group)

### plot(cmdscale(dist(t(genes.expression.small.fpkm))))
### text(cmdscale(dist(t(genes.expression.small.fpkm))),
###               labels=colnames(genes.expression.small.fpkm))

pdf(paste(NAME, ".MDS.display.MDS1.vs.MDS2.pdf", sep=""))
ggplot(mds.df, aes(x=MDSx,
                   y=MDSy,
                   color=group,
                   label=rownames(mds.df))) +
       geom_point(size=3) +
       # geom_text(col='black', size=4) +
       theme_bw() +
       theme(legend.position="top", legend.title=element_blank(),
                                    legend.key = element_blank()) +
       labs(x="MDS dimension 1", y="MDS dimension 2") +
       ggtitle("MDS display")
dev.off()
```

```
## pdf
##   2
```

****

VII. DATA VISUALIZATION : HEATMAPS

```
# Here making a few HEATMAPS, considering either the CELL CYCLE GENES,
# or the genes that are reactive in ASTROGLIOSIS

NAME <- "z.analysis.results"
```

****

1. Here considering the CELL CYCLE GENES :

```
####### to use two datasets in order to retrieve the genes :
####### genes.expression.large.tpm
####### genes.expression.large.fpkm
####### genes.expression.large

# head(genes.expression.large) ### using GENE_NAME
dim(genes.expression.large)  ### using GENE_NAME
```

```
## [1] 58381    95
```

```
# tail(genes.expression.large) ### using GENE_NAME

##################################################################################################
##################################################################################################

###### before we do the intersection, I believe that we shall take the 1st occurence of the genes:
# > length(genes.expression.large$GENE_NAME)
# [1] 58381
# > length(unique(genes.expression.large$GENE_NAME))
# [1] 56832
#
# X <- genes.expression.large
# Y <- subset(X, !duplicated(X$GENE_NAME))
# dim(Y) ### [1] 56832

 genes.expression.large.unique <- subset(genes.expression.large,
                                         !duplicated(genes.expression.large$GENE_NAME))

 dim(genes.expression.large.unique)
```

```
## [1] 56832    95
```

```
#################################################################################################
#################################################################################################
### here doing the PCA/MDS analysis on CELL_CYCLE_GENES

genes_cell_cyle <- read.delim("genes.KEGG_Cell_Cycle_GENES.txt",
                               header=TRUE, sep="\t", stringsAsFactors=F)

genes_cell_cyle$GENE_NAME <- genes_cell_cyle$Gene

# head(genes_cell_cyle)
# tail(genes_cell_cyle)
dim(genes_cell_cyle)
```

```
## [1] 128   2
```

```
# genes_cell_cycle_and_info <- merge(genes_cell_cyle,
#                                   genes.expression.large.unique,
#                                   by.x = Gene,
#                                   by.y = GENE_NAME,
#                                   all.x = TRUE)
```

```r
genes_cell_cycle_and_info <- join(genes_cell_cyle, genes.expression.large.unique, type = "inner")
```

```
## Joining by: GENE_NAME
```

```r
# head(genes_cell_cycle_and_info)
# tail(genes_cell_cycle_and_info)
dim(genes_cell_cycle_and_info)
```

```
## [1] 124  96
```

```r
write.table( genes_cell_cycle_and_info,
             file=paste(NAME, "genes.KEGG_Cell_Cycle_GENES.with.info.expression.txt", sep="."),
             sep="\t",
             quote = FALSE,
             row.names = FALSE,
             col.names = TRUE)

##################################################################################################

genes_cell_cycle_and_info.tpm <- subset(genes_cell_cycle_and_info,
                                     select=c("GENE_NAME",
                                     "DMSO1_lane1.TPM", "DMSO1_lane2.TPM",
                                     "DMSO2_lane1.TPM", "DMSO2_lane2.TPM",
                                     "DMSO3_lane1.TPM", "DMSO3_lane2.TPM",
                                     "Aph1.TPM", "Aph2.TPM", "Aph3.TPM",
                                     "Aph_KH7_1.TPM","Aph_KH7_2.TPM","Aph_KH7_3.TPM",
                                     "KH7_1.TPM", "KH7_2.TPM", "KH7_3.TPM",
                                     "Noc_1.TPM", "Noc_2.TPM", "Noc_3.TPM" ),
                                      na.rm = TRUE)

rownames(genes_cell_cycle_and_info.tpm) <- genes_cell_cycle_and_info.tpm$GENE_NAME
genes_cell_cycle_and_info.tpm <- genes_cell_cycle_and_info.tpm[,-1]

# head(genes_cell_cycle_and_info.tpm)
dim(genes_cell_cycle_and_info.tpm)
```

```
## [1] 124  18
```

```r
#####################################################################################################
#### dunno why R introduces the NA

# genes_cell_cycle_and_info.NA <- subset(genes_cell_cycle_and_info, is.na(CHR))
# dim(genes_cell_cycle_and_info.NA)      ###

# genes_cell_cycle_and_info.non.NA <- subset(genes_cell_cycle_and_info, !is.na(CHR))
# dim(genes_cell_cycle_and_info.non.NA)  ###

# genes_cell_cycle_and_info <- genes_cell_cycle_and_info.non.NA
# dim(genes_cell_cycle_and_info)

##############################################################################################
##################################################################################################
##################################################################################################
##### doing the HEATMAP analysis :

pdf(paste("genes.KEGG_Cell_Cycle_GENES.with.info.expression.heatmap.pdf", sep="."))
```

```
par(las=2)
par(mar=c(8,4,2,2))
par(cex.main=0.6)
    heatmap.2(as.matrix(genes_cell_cycle_and_info.tpm), col=bluered(149),
                        scale="row",trace="none",
                        cexRow=0.6, cexCol=0.6, cex.main=0.6,
                        Rowv=FALSE, symkey=FALSE, labRow=NA,
                        key=T, keysize=1.5, density.info="none",
                        main="heatmap of KEGG_Cell_Cycle genes")
```

```
## Warning in heatmap.2(as.matrix(genes_cell_cycle_and_info.tpm), col =
## bluered(149), : Discrepancy: Rowv is FALSE, while dendrogram is `both'.
## Omitting row dendogram.
```

```
dev.off()
```

```
## pdf
##   2
```

```
###############################################################################################
###################################################################################################
###################################################################################################
##### doing the PCA analysis :

group <- factor( c(rep("DMSO",6), rep("Aph", 3), rep("Aph_KH7", 3),
                   rep("KH7", 3), rep("Noc", 3)) )

pca <- prcomp(t(genes_cell_cycle_and_info.tpm))

###### plotting the pca$x :

pdf(paste(NAME, "genes.KEGG_Cell_Cycle_GENES.with.info.expression.PCA.display.in.3D.pdf", sep="."))
                           scatterplot3d(pca$x[,1:3],
                           color = c(rep("red",6), rep("orange",3), rep("green",3),
                                     rep("blue",3), rep("violet",3)),
                           pch=18,
                           main="PCA analysis of Cell Cycle Genes",
                           grid=TRUE,
                           box=TRUE)
dev.off()
```

```
## pdf
##   2
```

```
#######################################################################################
#######################################################################################

pca.df <- data.frame(PCA1=pca$x[,1],
                     PCA2=pca$x[,2],
                     PCA3=pca$x[,3],
                     group=group)

## Here we are plotting PCA1 vs PCA2

pdf(paste(NAME, "genes.KEGG_Cell_Cycle_GENES.with.info.expression.PCA.display.PC1.vs.PC2.pdf", sep=".")
ggplot(pca.df,
       aes(x=PCA1, y=PCA2, color=group, label=rownames(pca.df))) +
       geom_point(size=3) +
       # geom_text(col='black', size=4) +
       theme_bw() +
       theme(legend.position="top",
             legend.title=element_blank(),
             legend.key = element_blank()) +
             labs(x="PC1", y="PC2") +
        ggtitle("PCA analysis : PC1 vs PC2")
dev.off()
```

```
## pdf
##   2
```

****

2. Here considering LPS-reactive genes :

```
### here doing the PCA/MDS analysis on genes that are LPS_reactive

genes_LPS_reactive <- read.delim("genes.from_EVAN_Top50changes_in_LPS_reactive_astrocytes_symbol_HUGO",
                                  header=T, sep="\t", stringsAsFactors=F)

genes_LPS_reactive$GENE_NAME <- genes_LPS_reactive$Approved_symbol

dim(genes_LPS_reactive)
```

```
## [1] 40  7
```

```
# head(genes_LPS_reactive)
# tail(genes_LPS_reactive)

# genes_LPS_reactive_and_info <- merge(genes_LPS_reactive,
#                                      genes.expression.large.unique,
#                                      by.x = GENE_NAME,
#                                      by.y = GENE_NAME,
#                                      all.x = TRUE)

genes_LPS_reactive_and_info <- join(genes_LPS_reactive, genes.expression.large.unique, type = "inner")
```

```
## Joining by: GENE_NAME
```

```
# head(genes_LPS_reactive_and_info)
# tail(genes_LPS_reactive_and_info)
dim(genes_LPS_reactive_and_info)
```

```
## [1]  39 101
```

```
write.table( genes_LPS_reactive_and_info,
             file=paste(NAME, "genes.LPS_reactive.with.info.expression.txt", sep="."),
             sep="\t",
             quote = FALSE,
             row.names = FALSE,
             col.names = TRUE)

#####################################################################################################
#### dunno why R introduces the NA

# genes_LPS_reactive_and_info.NA <- subset(genes_LPS_reactive_and_info, is.na(CHR))
# dim(genes_LPS_reactive_and_info.NA)      ###

# genes_LPS_reactive_and_info.non.NA <- subset(genes_LPS_reactive_and_info, !is.na(CHR))
# dim(genes_LPS_reactive_and_info.non.NA)  ###

# genes_LPS_reactive_and_info <- genes_LPS_reactive_and_info.non.NA
# dim(genes_LPS_reactive_and_info)

##################################################################################################

genes_LPS_reactive_and_info.tpm <-   subset(genes_LPS_reactive_and_info,
                                     select=c("GENE_NAME",
                                     "DMSO1_lane1.TPM", "DMSO1_lane2.TPM",
                                     "DMSO2_lane1.TPM", "DMSO2_lane2.TPM",
                                     "DMSO3_lane1.TPM", "DMSO3_lane2.TPM",
                                     "Aph1.TPM", "Aph2.TPM", "Aph3.TPM",
                                     "Aph_KH7_1.TPM","Aph_KH7_2.TPM","Aph_KH7_3.TPM",
                                     "KH7_1.TPM", "KH7_2.TPM", "KH7_3.TPM",
                                     "Noc_1.TPM", "Noc_2.TPM", "Noc_3.TPM" ),
                                      na.rm = TRUE)

rownames(genes_LPS_reactive_and_info.tpm) <- genes_LPS_reactive_and_info.tpm$GENE_NAME
genes_LPS_reactive_and_info.tpm <- genes_LPS_reactive_and_info.tpm[,-1]

# head(genes_LPS_reactive_and_info.tpm)
dim(genes_LPS_reactive_and_info.tpm)
```

```
## [1] 39 18
```

```
#################################################################################################
#####################################################################################################
#####################################################################################################
##### doing the HEATMAP analysis :

pdf(paste("genes.LPS_reactive.with.info.expression.heatmap.pdf", sep=""))
par(las=2)
par(mar=c(8,4,2,2))
par(cex.main=0.6)
    heatmap.2(as.matrix(genes_LPS_reactive_and_info.tpm), col=bluered(149),
                         scale="row",trace="none",
                         cexRow=0.6, cexCol=0.6, cex.main=0.6,
                         Rowv=FALSE, symkey=FALSE, labRow=NA,
                         key=T, keysize=1.5, density.info="none",
                         main="heatmap of LPS reactive genes")
```

```
## Warning in heatmap.2(as.matrix(genes_LPS_reactive_and_info.tpm), col =
## bluered(149), : Discrepancy: Rowv is FALSE, while dendrogram is `both'.
## Omitting row dendogram.
```

```
dev.off()
```

```
## pdf
##   2
```

```
#####################################################################################################
#####################################################################################################
#####################################################################################################
##### doing the PCA analysis :

group <- factor( c(rep("DMSO",6), rep("Aph", 3), rep("Aph_KH7", 3),
                   rep("KH7", 3), rep("Noc", 3)) )

pca <- prcomp(t(genes_LPS_reactive_and_info.tpm))

###### plotting the pca$x :

pdf(paste(NAME, "genes.LPS_reactive.with.info.expression.PCA.display.in.3D.pdf", sep="."))
                            scatterplot3d(pca$x[,1:3],
                           color = c(rep("red",6), rep("orange",3), rep("green",3),
                                     rep("blue",3), rep("violet",3)),
                           pch=18,
                           main="PCA analysis of LPS reactive genes",
                           grid=TRUE,
                           box=TRUE)
dev.off()
```

```
## pdf
##   2
```

```
########################################################################################
########################################################################################

pca.df <- data.frame(PCA1=pca$x[,1],
                     PCA2=pca$x[,2],
                     PCA3=pca$x[,3],
                     group=group)

## Here we are plotting PCA1 vs PCA2

pdf(paste(NAME, "genes.LPS_reactive.with.info.expression.PCA.display.PC1.vs.PC2.pdf", sep="."))
ggplot(pca.df,
       aes(x=PCA1, y=PCA2, color=group, label=rownames(pca.df))) +
       geom_point(size=3) +
       # geom_text(col='black', size=4) +
       theme_bw() +
       theme(legend.position="top",
             legend.title=element_blank(),
             legend.key = element_blank()) +
             labs(x="PC1", y="PC2") +
        ggtitle("PCA analysis : PC1 vs PC2")
dev.off()
```

```
## pdf
##   2
```

```
#########################################################################################################
#########################################################################################################
```

****

3. Here considering MCAO-reactive genes :

```
### here doing the PCA/MDS analysis on genes that are MCAO_reactive

genes_MCAO_reactive <- read.delim("genes.from_EVAN_Top50changes_in_MCAO_reactive_astrocytes_symbol_HUGO
                                  header=T, sep="\t", stringsAsFactors=F)

genes_MCAO_reactive$GENE_NAME <- genes_MCAO_reactive$Approved_symbol

dim(genes_MCAO_reactive)
```

```
## [1] 45  7
```

```
# head(genes_MCAO_reactive)
# tail(genes_MCAO_reactive)

# genes_MCAO_reactive_and_info <- merge(genes_MCAO_reactive,
#                                      genes.expression.large.unique,
#                                      by.x = GENE_NAME,
#                                      by.y = GENE_NAME,
#                                      all.x = TRUE)

genes_MCAO_reactive_and_info <- join(genes_MCAO_reactive, genes.expression.large.unique, type = "inner"
```

```
## Joining by: GENE_NAME
```

```
dim(genes_MCAO_reactive_and_info)
```

```
## [1]  45 101
```

```
# head(genes_MCAO_reactive_and_info)
# tail(genes_MCAO_reactive_and_info)

################################################################

write.table( genes_MCAO_reactive_and_info,
             file=paste(NAME, "genes.MCAO_reactive.with.info.expression.txt", sep="."),
             sep="\t",
             quote = FALSE,
             row.names = FALSE,
             col.names = TRUE)

#####################################################################################################
#### dunno why R introduces the NA

# genes_MCAO_reactive_and_info.NA <- subset(genes_MCAO_reactive_and_info, is.na(CHR))
# dim(genes_MCAO_reactive_and_info.NA)      ###

# genes_MCAO_reactive_and_info.non.NA <- subset(genes_MCAO_reactive_and_info, !is.na(CHR))
# dim(genes_MCAO_reactive_and_info.non.NA)  ###

# genes_MCAO_reactive_and_info <- genes_MCAO_reactive_and_info.non.NA
# dim(genes_MCAO_reactive_and_info)

##################################################################################################
```

```r
genes_MCAO_reactive_and_info.tpm <-   subset(genes_MCAO_reactive_and_info,
                                             select=c("GENE_NAME",
                                            "DMSO1_lane1.TPM", "DMSO1_lane2.TPM",
                                            "DMSO2_lane1.TPM", "DMSO2_lane2.TPM",
                                            "DMSO3_lane1.TPM", "DMSO3_lane2.TPM",
                                            "Aph1.TPM", "Aph2.TPM", "Aph3.TPM",
                                            "Aph_KH7_1.TPM","Aph_KH7_2.TPM","Aph_KH7_3.TPM",
                                            "KH7_1.TPM", "KH7_2.TPM", "KH7_3.TPM",
                                            "Noc_1.TPM", "Noc_2.TPM", "Noc_3.TPM" ),
                                             na.rm = TRUE)

rownames(genes_MCAO_reactive_and_info.tpm) <- genes_MCAO_reactive_and_info.tpm$GENE_NAME
genes_MCAO_reactive_and_info.tpm <- genes_MCAO_reactive_and_info.tpm[,-1]

# head(genes_MCAO_reactive_and_info.tpm)
dim(genes_MCAO_reactive_and_info.tpm)
```

```
## [1] 45 18
```

```r
##############################################################################################
##################################################################################################
##################################################################################################
##### doing the HEATMAP analysis :

pdf(paste(NAME, "genes.MCAO_reactive.with.info.expression.heatmap.pdf", sep="."))
par(las=2)
par(mar=c(8,4,2,2))
par(cex.main=0.6)
    heatmap.2(as.matrix(genes_MCAO_reactive_and_info.tpm), col=bluered(149),
                        scale="row",trace="none",
                        cexRow=0.6, cexCol=0.6, cex.main=0.6,
                        Rowv=FALSE, symkey=FALSE, labRow=NA,
                        key=T, keysize=1.5, density.info="none",
                        main="heatmap of MCAO reactive genes")
```

```
## Warning in heatmap.2(as.matrix(genes_MCAO_reactive_and_info.tpm), col =
## bluered(149), : Discrepancy: Rowv is FALSE, while dendrogram is `both'.
## Omitting row dendogram.
```

```r
dev.off()
```

```
## pdf
##   2
```

```r
##################################################################################################
##################################################################################################
##################################################################################################
##### doing the PCA analysis :

group <- factor( c(rep("DMSO",6), rep("Aph", 3), rep("Aph_KH7", 3),
                   rep("KH7", 3), rep("Noc", 3)) )

pca <- prcomp(t(genes_LPS_reactive_and_info.tpm))

###### plotting the pca$x :
```

```
pdf(paste("genes.MCAO_reactive.with.info.expression.PCA.display.in.3D.pdf", sep=""))
                           scatterplot3d(pca$x[,1:3],
                           color = c(rep("red",6), rep("orange",3), rep("green",3),
                                     rep("blue",3), rep("violet",3)),
                           pch=18,
                           main="PCA analysis of MCAO reactive genes",
                           grid=TRUE,
                           box=TRUE)
dev.off()
```

```
## pdf
##   2
```

```
######################################################################################
######################################################################################

pca.df <- data.frame(PCA1=pca$x[,1],
                     PCA2=pca$x[,2],
                     PCA3=pca$x[,3],
                     group=group)

## Here we are plotting PCA1 vs PCA2

pdf(paste(NAME, "genes.MCAO_reactive.with.info.expression.PCA.display.PC1.vs.PC2.pdf", sep="."))
ggplot(pca.df,
       aes(x=PCA1, y=PCA2, color=group, label=rownames(pca.df))) +
       geom_point(size=3) +
       # geom_text(col='black', size=4) +
       theme_bw() +
       theme(legend.position="top",
             legend.title=element_blank(),
             legend.key = element_blank()) +
             labs(x="PC1", y="PC2") +
        ggtitle("PCA analysis : PC1 vs PC2")
dev.off()
```

```
## pdf
##   2
```

```
##############################################
```

****

VIII. DATA VISUALIZATION :

A. SCATTER PLOTS

B. VOLCANO PLOTS

for each pair-wise comparison

```
### Here reading again the files with all the info (RPKM, TPM, EXPRESSION and DE genes)

### Here using the DATA TABLE that contains the information on ~13 000 genes
```

****

1. SETTING UP the DATAFRAMES

```
#################################################################################################
##################### here to upload the data where we did integrate all the files with ALL GENES
###### the results from RSEM
###### the results from LIMMA

genes.expression.small <- read.delim("analysis.LIMMA.integrating.all.samples.with.data.table",
                                     sep="\t", header=T, stringsAsFactors=F)

# head(genes.expression.small)
dim(genes.expression.small)
```

```
## [1] 13247    95
```

```
################ here we would have to make a special ROWNAME,
### as some genes are present in multiple isoforms ..

genes.expression.small$ID <- rownames(genes.expression.small)
genes.expression.small$GENE_NAME_ID <- paste(genes.expression.small$GENE_NAME,
                                             genes.expression.small$ID, sep=":")

# head(genes.expression.small)
dim(genes.expression.small)
```

```
## [1] 13247    95
```

```
############################################################################################
############################################################################################
############################################################################################
############################################################################################

### for some reasons that I do not know yet, there are some lines with NA in the file
### to exclude these lines from the files with TPM or FPKM, or here below :

genes.expression.small.NA <- subset(genes.expression.small, is.na(CHR))
dim(genes.expression.small.NA)     ### 291   95
```

```
## [1] 291  95
```

```
genes.expression.small.non.NA <- subset(genes.expression.small, !is.na(CHR))
dim(genes.expression.small.non.NA) ### 12956    95
```

```
## [1] 12956    95
```

```
genes.expression.small <- genes.expression.small.non.NA
dim(genes.expression.small)
```

```
## [1] 12956    95
```

```
############################################################################################
############################################################################################
############################################################################################
############################################################################################

###### transforming the DATA FRAME into a DATA TABLE :
```

```
genes.expression.small.dt <- as.data.table(genes.expression.small)

# head(genes.expression.small.dt)
dim(genes.expression.small.dt)
```

```
## [1] 12956    95
```

```
# colnames(genes.expression.small)
# [1] "CHR"               "START"             "END"
# [4] "STRAND"            "GENE_ID"           "GENE_NAME"
# [7] "GENE_TYPE"         "DMSO1_lane1.count" "DMSO1_lane1.TPM"
#[10] "DMSO1_lane1.FPKM"  "DMSO1_lane2.count" "DMSO1_lane2.TPM"
#[13] "DMSO1_lane2.FPKM"  "DMSO2_lane1.count" "DMSO2_lane1.TPM"
#[16] "DMSO2_lane1.FPKM"  "DMSO2_lane2.count" "DMSO2_lane2.TPM"
#[19] "DMSO2_lane2.FPKM"  "DMSO3_lane1.count" "DMSO3_lane1.TPM"
#[22] "DMSO3_lane1.FPKM"  "DMSO3_lane2.count" "DMSO3_lane2.TPM"
#[25] "DMSO3_lane2.FPKM"  "Aph1.count"        "Aph1.TPM"
#[28] "Aph1.FPKM"         "Aph2.count"        "Aph2.TPM"
#[31] "Aph2.FPKM"         "Aph3.count"        "Aph3.TPM"
#[34] "Aph3.FPKM"         "Aph_KH7_1.count"   "Aph_KH7_1.TPM"
#[37] "Aph_KH7_1.FPKM"    "Aph_KH7_2.count"   "Aph_KH7_2.TPM"
#[40] "Aph_KH7_2.FPKM"    "Aph_KH7_3.count"   "Aph_KH7_3.TPM"
#[43] "Aph_KH7_3.FPKM"    "KH7_1.count"       "KH7_1.TPM"
#[46] "KH7_1.FPKM"        "KH7_2.count"       "KH7_2.TPM"
#[49] "KH7_2.FPKM"        "KH7_3.count"       "KH7_3.TPM"
#[52] "KH7_3.FPKM"        "Noc_1.count"       "Noc_1.TPM"
#[55] "Noc_1.FPKM"        "Noc_2.count"       "Noc_2.TPM"
#[58] "Noc_2.FPKM"        "Noc_3.count"       "Noc_3.TPM"
#[61] "Noc_3.FPKM"        "ID"                "GENE_NAME_ID"
#[64] "logFC.Aph"         "AveExpr.Aph"       "t.Aph"
#[67] "P.Value.Aph"       "adj.P.Val.Aph"     "B.Aph"
#[70] "Gene.Aph"          "ID.Aph"            "logFC.Aph_KH7"
#[73] "AveExpr.Aph_KH7"   "t.Aph_KH7"         "P.Value.Aph_KH7"
#[76] "adj.P.Val.Aph_KH7" "B.Aph_KH7"         "Gene.Aph_KH7"
#[79] "ID.Aph_KH7"        "logFC.KH7"         "AveExpr.KH7"
#[82] "t.KH7"             "P.Value.KH7"       "adj.P.Val.KH7"
#[85] "B.KH7"             "Gene.KH7"          "ID.KH7"
#[88] "logFC.Noc"         "AveExpr.Noc"       "t.Noc"
#[91] "P.Value.Noc"       "adj.P.Val.Noc"     "B.Noc"
#[94] "Gene.Noc"          "ID.Noc"

####################################################################################################
####################################################################################################

# genes.expression.small %>%
#                transmute(GENE_NAME,
#                          Aph_Mean = rowMeans(select(., c(Aph_KH7_1.FPKM,
#                                                          Aph_KH7_2.FPKM,
#                                                          Aph_KH7_3.FPKM ))))

### to add some DATA with the COMPUTED AVERAGES :

genes.expression.small$DMSO_lane1.FPKM.average <- rowMeans(subset(genes.expression.small,
                                                  select = c(DMSO1_lane1.FPKM,
                                                             DMSO2_lane1.FPKM,
                                                             DMSO3_lane1.FPKM)),
                                                  na.rm = TRUE)

genes.expression.small$DMSO_lane2.FPKM.average <- rowMeans(subset(genes.expression.small,
                                                  select = c(DMSO1_lane2.FPKM,
                                                             DMSO2_lane2.FPKM,
                                                             DMSO3_lane2.FPKM)),
                                                  na.rm = TRUE)

genes.expression.small$Aph.FPKM.average <- rowMeans(subset(genes.expression.small,
                                                  select = c(Aph1.FPKM,
                                                             Aph2.FPKM,
                                                             Aph3.FPKM)),
                                                  na.rm = TRUE)

genes.expression.small$Aph_KH7.FPKM.average <- rowMeans(subset(genes.expression.small,
                                                  select = c(Aph_KH7_1.FPKM,
                                                             Aph_KH7_2.FPKM,
                                                             Aph_KH7_3.FPKM)),
                                                  na.rm = TRUE)

genes.expression.small$KH7.FPKM.average <- rowMeans(subset(genes.expression.small,
                                                  select = c(KH7_1.FPKM,
                                                             KH7_2.FPKM,
                                                             KH7_3.FPKM)),
                                                  na.rm = TRUE)

genes.expression.small$Noc.FPKM.average <- rowMeans(subset(genes.expression.small,
                                                  select = c(Noc_1.FPKM,
                                                             Noc_2.FPKM,
                                                             Noc_3.FPKM)),
                                                  na.rm = TRUE)

### writing for verification :

write.table(genes.expression.small,
            file=paste(NAME, "analysis.LIMMA.integrating.all.samples.with.data.table.printing.FPKM.avera
            quote=F,
            sep="\t",
            row.names = FALSE,
            col.names = TRUE)

############################################################################################
############################################################################################
```

****

```
### here selecting the sets of DEG that we will do the displays on : here the FC > 1.5
###
### genes regulated by Aph
### genes regulated by Aph_KH7
### genes regulated by KH7
### genes regulated by Noc

### here we have started to place the DEG in separate dataframes,
### although we shall possibly keep the initial big and large DATAFRAME !


############################################################################################
############################################################################################

### in the BIG LARGE DATAFRAME
### to LABEL THE GENES based on the REGULATION : "", "U", or"D"

############################################################################################
############################################################################################
```

****

2. DISPLAYS of Aph-regulated genes

```
#### SHOWING THE GENES that are REGULATED by APH
#### as SCATTER PLOT
#### as VOLCANO PLOT

########################################## making another COLUMN that depends on REGULATION

genes.expression.small$Aph.regulated <- ""

genes.expression.small$Aph.regulated[( (genes.expression.small$logFC.Aph > 0.58) &
                                       (genes.expression.small$adj.P.Val.Aph < 0.05) &
                                       (genes.expression.small$Aph.FPKM.average > 1) ) ] <- "U"

genes.expression.small$Aph.regulated[( (genes.expression.small$logFC.Aph < -0.58) &
                                       (genes.expression.small$adj.P.Val.Aph < 0.05) &
                                       (genes.expression.small$DMSO_lane1.FPKM.average > 1) )] <- "D"

write.table(genes.expression.small,
            file=paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
                       "Aph", sep=""),
            quote=F,
            sep="\t",
            row.names = FALSE,
            col.names = TRUE)

######################################### computing the number of GENES in each category :

table(genes.expression.small$Aph.regulated)
```

```
##
##           D     U
## 11837   340   779
```

```
write.table(table(genes.expression.small$Aph.regulated),
            file=paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
                       "Aph", ".a.summary", sep=""),
            quote=F,
            sep="\t",
            row.names = FALSE,
            col.names = TRUE)

# table(genes.expression.small$Aph.regulated)[[]]
# [1] 11837
# table(genes.expression.small$Aph.regulated)[["U"]]
# 779
# table(genes.expression.small$Aph.regulated)[["D"]]
# 340

####### making some smaller dataframes only of the genes that are reg by Aph :

 genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.UP.by.Aph <- subset(genes.expression.small,
                                                 ( (logFC.Aph > 0.58) &
                                                   (adj.P.Val.Aph < 0.05) &
                                                   (Aph.FPKM.average > 1) ) )

 genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.DOWN.by.Aph <- subset(genes.expression.small,
                                                 ( (logFC.Aph < -0.58) &
                                                   (adj.P.Val.Aph < 0.05) &
                                                   (DMSO_lane1.FPKM.average > 1) ) )

 genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.REG.by.Aph <- rbind(genes.expression.small.DEG.FDR0p05.F
                                                                    genes.expression.small.DEG.FDR0p05.F


 dim(genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.UP.by.Aph)
```

```
## [1] 779 102
```

```
 dim(genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.DOWN.by.Aph)
```

```
## [1] 340 102
```

```
 dim(genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.REG.by.Aph)
```

```
## [1] 1119  102
```

```
 write.table(genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.REG.by.Aph,
             file=paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.", "Aph
                        ".only.DEG",  sep=""),
             quote=F,
             sep="\t",
             row.names = FALSE,
             col.names = TRUE)

###############################################################################################
###############################################################################################

### MAKING the DISPLAYS as SCATTER PLOTS
### PDF
### PNG
### with limma
### with ggplot2

###############################################################################################
###############################################################################################

pdf(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "Aph",
           ".display.limma.SCATTER.pdf", sep=""))

   plotWithHighlights(log2(genes.expression.small$DMSO_lane1.FPKM.average),
                      log2(genes.expression.small$Aph.FPKM.average),
                      status=genes.expression.small$Aph.regulated,
                   values=c("U","D"),
                   bg.col="grey",
                   xlim=c(-2,12), ylim=c(-2,12),
                   hl.cex=0.6, cex.main=0.8, cex.lab =0.8,
                   xlab="log2 average FPKM in DMSO",
                   ylab="log2 average FPKM in Aph treatment",
                   legend= "topright",
                   main=paste("Aph", " regulated genes", sep=""))

dev.off()
```

```
## pdf
##   2
```

```
#############################################################################################

png(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "Aph",
           ".display.limma.SCATTER.png", sep=""))

   plotWithHighlights(log2(genes.expression.small$DMSO_lane1.FPKM.average),
                      log2(genes.expression.small$Aph.FPKM.average),
                      status=genes.expression.small$Aph.regulated,
                   values=c("U","D"),
                   bg.col="grey",
                   xlim=c(-2,12), ylim=c(-2,12),
                   hl.cex=0.6, cex.main=0.8, cex.lab =0.8,
                   xlab="log2 average FPKM in DMSO",
                   ylab="log2 average FPKM in Aph treatment",
                   legend= "topright",
                   main=paste("Aph", " regulated genes", sep=""))

dev.off()
```

```
## pdf
##   2
```

```
############################################################################################
############################################################################################

pdf(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "Aph",
           ".display.ggplot2.SCATTER.pdf", sep=""))

ggplot(genes.expression.small,
       aes(x=log2(DMSO_lane1.FPKM.average),
           y=log2(Aph.FPKM.average),
           color=Aph.regulated)) +
           geom_point(size=1) +
           theme_bw() +
           xlim(-2, 12) +
           ylim(-2, 12) +
           scale_colour_manual(values = c("grey","D"="green", "U"="red")) +
           labs(x="log2 average FPKM in DMSO",
                y="log2 average FPKM in Aph") +
           ggtitle(paste("Aph", " regulated genes", sep="")) +
           theme(legend.position="bottom",
                 legend.title=element_blank(),
                 legend.key = element_blank())
```

```
## Warning: Removed 88 rows containing missing values (geom_point).
```

```
dev.off()
```

```
## pdf
##   2
```

```
############################################################################################

png(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "Aph",
           ".display.ggplot2.SCATTER.png", sep=""))

ggplot(genes.expression.small,
       aes(x=log2(DMSO_lane1.FPKM.average),
           y=log2(Aph.FPKM.average),
           color=Aph.regulated)) +
           geom_point(size=1) +
           theme_bw() +
           xlim(-2, 12) +
           ylim(-2, 12) +
           scale_colour_manual(values = c("grey","D"="green", "U"="red")) +
           labs(x="log2 average FPKM in DMSO",
                y="log2 average FPKM in Aph") +
           ggtitle(paste("Aph", " regulated genes", sep="")) +
           theme(legend.position="bottom",
                 legend.title=element_blank(),
                 legend.key = element_blank())
```

```
## Warning: Removed 88 rows containing missing values (geom_point).
```

```
dev.off()
```

```
## pdf
##   2
```

```
############################################################################################
############################################################################################
############################################################################################
############################################################################################

### MAKING the DISPLAYS as VOLCANO PLOTS
### PDF
### PNG
### with limma
### with ggplot2

############################################################################################
############################################################################################

pdf(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "Aph",
           ".display.limma.VOLCANO.pdf", sep=""))

plotWithHighlights(       genes.expression.small$logFC.Aph,
                   -log10(genes.expression.small$adj.P.Val.Aph),
                   status=genes.expression.small$Aph.regulated,
                   values=c("U","D"),
                   bg.col="grey",
                   xlim=c(-3,3),
                   ylim=c(0,10),
                   hl.cex=0.6, cex.main=0.8, cex.lab =0.8,
                   xlab="log2FC",
                   ylab="-log10 adj.P.Val",
                   legend= "topright",
                   main=paste("Aph", " regulated genes", sep=""))

dev.off()
```

```
## pdf
##   2
```

```
############################################################################################

png(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "Aph",
           ".display.limma.VOLCANO.png", sep=""))

plotWithHighlights(       genes.expression.small$logFC.Aph,
                   -log10(genes.expression.small$adj.P.Val.Aph),
                   status=genes.expression.small$Aph.regulated,
                   values=c("U","D"),
                   bg.col="grey",
                   xlim=c(-3,3),
                   ylim=c(0,10),
                   hl.cex=0.6, cex.main=0.8, cex.lab =0.8,
                   xlab="log2FC",
                   ylab="-log10 adj.P.Val",
                   legend= "topright",
                   main=paste("Aph", " regulated genes", sep=""))

dev.off()
```

```
## pdf
##   2
```

```
############################################################################################
############################################################################################

pdf(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "Aph",
           ".display.ggplot2.VOLCANO.pdf", sep=""))

    ggplot(genes.expression.small,
           aes(x=genes.expression.small$logFC.Aph,
               y=-log10(genes.expression.small$adj.P.Val.Aph),
           color=Aph.regulated)) +
           geom_point(size=1) +
           theme_bw() +
           xlim(-4, 4) +
           ylim(0, 10) +
           scale_colour_manual(values = c("grey","D"="green", "U"="red")) +
           labs(x="log2FC",
```

```r
                y="-log10 adj.P.Val") +
           ggtitle(paste("Aph", " regulated genes", sep="")) +
           theme(legend.position="bottom",
                 legend.title=element_blank(),
                 legend.key = element_blank())
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```

```r
dev.off()
```

```
## pdf
##   2
```

```r
############################################################################################

png(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "Aph",
           ".display.ggplot2.VOLCANO.png", sep=""))

    ggplot(genes.expression.small,
           aes(x=genes.expression.small$logFC.Aph,
               y=-log10(genes.expression.small$adj.P.Val.Aph),
           color=Aph.regulated)) +
           geom_point(size=1) +
           theme_bw() +
           xlim(-4, 4) +
           ylim(0, 10) +
           scale_colour_manual(values = c("grey","D"="green", "U"="red")) +
           labs(x="log2FC",
                y="-log10 adj.P.Val") +
           ggtitle(paste("Aph", " regulated genes", sep="")) +
           theme(legend.position="bottom",
                 legend.title=element_blank(),
                 legend.key = element_blank())
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```

```r
dev.off()
```

```
## pdf
##   2
```

****

2. DISPLAYS of Aph_KH7-regulated genes

```
############################################################################################
############################################################################################
#### SHOWING THE GENES that are REGULATED by Aph_KH7
#### as SCATTER PLOT
#### as VOLCANO PLOT

########################################## making another COLUMN that depends on REGULATION

genes.expression.small$Aph_KH7.regulated <- ""

genes.expression.small$Aph_KH7.regulated[( (genes.expression.small$logFC.Aph_KH7 > 0.58) &
                                       (genes.expression.small$adj.P.Val.Aph_KH7 < 0.05) &
                                       (genes.expression.small$Aph_KH7.FPKM.average > 1) ) ] <- "U"

genes.expression.small$Aph_KH7.regulated[( (genes.expression.small$logFC.Aph_KH7 < -0.58) &
                                       (genes.expression.small$adj.P.Val.Aph_KH7 < 0.05) &
                                       (genes.expression.small$DMSO_lane1.FPKM.average > 1) )] <- "D"

write.table(genes.expression.small,
            file=paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
                        "Aph_KH7", sep=""),
            quote=F,
            sep="\t",
            row.names = FALSE,
            col.names = TRUE)

######################################### computing the number of GENES in each category :

table(genes.expression.small$Aph_KH7.regulated)
```

```
##
##         D    U
## 9807 1761 1388
```

```
write.table(table(genes.expression.small$Aph_KH7.regulated),
            file=paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
                       "Aph_KH7",
                       ".a.summary", sep=""),
            quote=F,
            sep="\t",
            row.names = FALSE,
            col.names = TRUE)

# table(genes.expression.small$Aph_KH7.regulated)[[]]
# [1] 11837
# table(genes.expression.small$Aph_KH7.regulated)[["U"]]
# 779
# table(genes.expression.small$Aph_KH7.regulated)[["D"]]
# 340

############################################################################################
############################################################################################
```

```
####### making some smaller dataframes only of the genes that are reg by Aph_KH7 :

 genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.UP.by.Aph_KH7 <- subset(genes.expression.small,
                                                 ( (logFC.Aph_KH7 > 0.58) &
                                                   (adj.P.Val.Aph_KH7 < 0.05) &
                                                   (Aph_KH7.FPKM.average > 1) ) )

 genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.DOWN.by.Aph_KH7 <- subset(genes.expression.small,
                                                 ( (logFC.Aph_KH7 < -0.58) &
                                                   (adj.P.Val.Aph_KH7 < 0.05) &
                                                   (DMSO_lane1.FPKM.average > 1) ) )

 genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.REG.by.Aph_KH7 <- rbind(genes.expression.small.DEG.FDR0p
                                                                    genes.expression.small.DEG.FDR0p05.


 dim(genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.UP.by.Aph_KH7)
```

```
## [1] 1388  103
```

```
 dim(genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.DOWN.by.Aph_KH7)
```

```
## [1] 1761  103
```

```
 dim(genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.REG.by.Aph_KH7)
```

```
## [1] 3149  103
```

```
 write.table(genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.REG.by.Aph_KH7,
             file=paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.", "Ap
                        ".only.DEG",  sep=""),
             quote=F,
             sep="\t",
             row.names = FALSE,
             col.names = TRUE)

############################################################################################
############################################################################################

### MAKING the DISPLAYS as SCATTER PLOTS
### PDF
### PNG
### with limma
### with ggplot2

############################################################################################
############################################################################################

pdf(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "Aph_KH7",
           ".display.limma.SCATTER.pdf", sep=""))

   plotWithHighlights(log2(genes.expression.small$DMSO_lane1.FPKM.average),
                      log2(genes.expression.small$Aph_KH7.FPKM.average),
                      status=genes.expression.small$Aph_KH7.regulated,
                   values=c("U","D"),
                   bg.col="grey",
                   xlim=c(-2,12), ylim=c(-2,12),
                   hl.cex=0.6, cex.main=0.8, cex.lab =0.8,
                   xlab="log2 average FPKM in DMSO",
                   ylab="log2 average FPKM in Aph_KH7 treatment",
                   legend= "topright",
                   main=paste("Aph_KH7", " regulated genes", sep=""))

dev.off()
```

```
## pdf
##   2
```

```
#############################################################################################

png(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "Aph_KH7",
           ".display.limma.SCATTER.png", sep=""))

   plotWithHighlights(log2(genes.expression.small$DMSO_lane1.FPKM.average),
                      log2(genes.expression.small$Aph_KH7.FPKM.average),
                      status=genes.expression.small$Aph_KH7.regulated,
                   values=c("U","D"),
                   bg.col="grey",
                   xlim=c(-2,12), ylim=c(-2,12),
                   hl.cex=0.6, cex.main=0.8, cex.lab =0.8,
                   xlab="log2 average FPKM in DMSO",
                   ylab="log2 average FPKM in Aph_KH7 treatment",
                   legend= "topright",
                   main=paste("Aph_KH7", " regulated genes", sep=""))

dev.off()
```

```
## pdf
##   2
```

```
############################################################################################
############################################################################################

pdf(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "Aph_KH7",
           ".display.ggplot2.SCATTER.pdf", sep=""))

ggplot(genes.expression.small,
       aes(x=log2(DMSO_lane1.FPKM.average),
           y=log2(Aph_KH7.FPKM.average),
           color=Aph_KH7.regulated)) +
           geom_point(size=1) +
           theme_bw() +
           xlim(-2, 12) +
           ylim(-2, 12) +
           scale_colour_manual(values = c("grey","D"="green", "U"="red")) +
           labs(x="log2 average FPKM in DMSO",
                y="log2 average FPKM in Aph_KH7") +
           ggtitle(paste("Aph_KH7", " regulated genes", sep="")) +
           theme(legend.position="bottom",
                 legend.title=element_blank(),
                 legend.key = element_blank())
```

```
## Warning: Removed 115 rows containing missing values (geom_point).
```

```
dev.off()
```

```
## pdf
##   2
```

```
############################################################################################

png(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "Aph_KH7",
           ".display.ggplot2.SCATTER.png", sep=""))

ggplot(genes.expression.small,
       aes(x=log2(DMSO_lane1.FPKM.average),
           y=log2(Aph_KH7.FPKM.average),
           color=Aph_KH7.regulated)) +
           geom_point(size=1) +
           theme_bw() +
           xlim(-2, 12) +
           ylim(-2, 12) +
           scale_colour_manual(values = c("grey","D"="green", "U"="red")) +
           labs(x="log2 average FPKM in DMSO",
                y="log2 average FPKM in Aph_KH7") +
           ggtitle(paste("Aph_KH7", " regulated genes", sep="")) +
           theme(legend.position="bottom",
                 legend.title=element_blank(),
                 legend.key = element_blank())
```

```
## Warning: Removed 115 rows containing missing values (geom_point).
```

```
dev.off()
```

```
## pdf
##   2
```

```
############################################################################################
############################################################################################
############################################################################################
############################################################################################

### MAKING the DISPLAYS as VOLCANO PLOTS
### PDF
### PNG
### with limma
### with ggplot2

############################################################################################
############################################################################################

pdf(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "Aph_KH7",
           ".display.limma.VOLCANO.pdf", sep=""))

plotWithHighlights(       genes.expression.small$logFC.Aph_KH7,
                   -log10(genes.expression.small$adj.P.Val.Aph_KH7),
                   status=genes.expression.small$Aph_KH7.regulated,
                   values=c("U","D"),
                   bg.col="grey",
                   xlim=c(-3,3),
                   ylim=c(0,14),
                   hl.cex=0.6, cex.main=0.8, cex.lab =0.8,
                   xlab="log2FC",
                   ylab="-log10 adj.P.Val",
                   legend= "topright",
                   main=paste("Aph_KH7", " regulated genes", sep=""))

dev.off()
```

```
## pdf
##   2
```

```
############################################################################################

png(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "Aph_KH7",
           ".display.limma.VOLCANO.png", sep=""))

plotWithHighlights(       genes.expression.small$logFC.Aph_KH7,
                   -log10(genes.expression.small$adj.P.Val.Aph_KH7),
                   status=genes.expression.small$Aph_KH7.regulated,
                   values=c("U","D"),
                   bg.col="grey",
                   xlim=c(-3,3),
                   ylim=c(0,10),
                   hl.cex=0.6, cex.main=0.8, cex.lab =0.8,
                   xlab="log2FC",
                   ylab="-log10 adj.P.Val",
                   legend= "topright",
                   main=paste("Aph_KH7", " regulated genes", sep=""))

dev.off()
```

```
## pdf
##   2
```

```
############################################################################################
############################################################################################

pdf(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "Aph_KH7",
           ".display.ggplot2.VOLCANO.pdf", sep=""))

    ggplot(genes.expression.small,
           aes(x=genes.expression.small$logFC.Aph_KH7,
               y=-log10(genes.expression.small$adj.P.Val.Aph_KH7),
           color=Aph_KH7.regulated)) +
           geom_point(size=1) +
           theme_bw() +
           xlim(-4, 4) +
           ylim(0, 10) +
           scale_colour_manual(values = c("grey","D"="green", "U"="red")) +
           labs(x="log2FC",
                y="-log10 adj.P.Val") +
           ggtitle(paste("Aph_KH7", " regulated genes", sep="")) +
           theme(legend.position="bottom",
                 legend.title=element_blank(),
                 legend.key = element_blank())
```

```
## Warning: Removed 486 rows containing missing values (geom_point).
```

```
dev.off()
```

```
## pdf
##   2
```

```
#############################################################################################

png(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "Aph_KH7",
           ".display.ggplot2.VOLCANO.png", sep=""))

    ggplot(genes.expression.small,
           aes(x=genes.expression.small$logFC.Aph_KH7,
               y=-log10(genes.expression.small$adj.P.Val.Aph_KH7),
           color=Aph_KH7.regulated)) +
           geom_point(size=1) +
           theme_bw() +
           xlim(-4, 4) +
           ylim(0, 10) +
           scale_colour_manual(values = c("grey","D"="green", "U"="red")) +
           labs(x="log2FC",
                y="-log10 adj.P.Val") +
           ggtitle(paste("Aph_KH7", " regulated genes", sep="")) +
           theme(legend.position="bottom",
                 legend.title=element_blank(),
                 legend.key = element_blank())
```

```
## Warning: Removed 486 rows containing missing values (geom_point).
```

```
dev.off()
```

```
## pdf
##   2
```

```
#############################################################################################
#############################################################################################
```

****

4. DISPLAYS of KH7-regulated genes

```
############################################################################################
############################################################################################
#### SHOWING THE GENES that are REGULATED by KH7
#### as SCATTER PLOT
#### as VOLCANO PLOT

########################################## making another COLUMN that depends on REGULATION

genes.expression.small$KH7.regulated <- ""

genes.expression.small$KH7.regulated[( (genes.expression.small$logFC.KH7 > 0.58) &
                                       (genes.expression.small$adj.P.Val.KH7 < 0.05) &
                                       (genes.expression.small$KH7.FPKM.average > 1) ) ] <- "U"

genes.expression.small$KH7.regulated[( (genes.expression.small$logFC.KH7 < -0.58) &
                                       (genes.expression.small$adj.P.Val.KH7 < 0.05) &
                                       (genes.expression.small$DMSO_lane1.FPKM.average > 1) )] <- "D"

write.table(genes.expression.small,
            file=paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
                        "KH7", sep=""),
            quote=F,
            sep="\t",
            row.names = FALSE,
            col.names = TRUE)

######################################### computing the number of GENES in each category :

table(genes.expression.small$KH7.regulated)
```

```
##
##           D     U
## 10192  1324  1440
```

```
# head(genes.expression.small$KH7.regulated)

write.table(table(genes.expression.small$KH7.regulated),
            file=paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
                       "KH7", ".a.summary", sep=""),
            quote=F,
            sep="\t",
            row.names = FALSE,
            col.names = TRUE)

############################################################################################
############################################################################################

####### making some smaller dataframes only of the genes that are reg by KH7 :

 genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.UP.by.KH7 <- subset(genes.expression.small,
                                                 ( (logFC.KH7 > 0.58) &
                                                   (adj.P.Val.KH7 < 0.05) &
                                                   (KH7.FPKM.average > 1) ) )

 genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.DOWN.by.KH7 <- subset(genes.expression.small,
                                                 ( (logFC.KH7 < -0.58) &
                                                   (adj.P.Val.KH7 < 0.05) &
                                                   (DMSO_lane1.FPKM.average > 1) ) )

 genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.REG.by.KH7 <- rbind(genes.expression.small.DEG.FDR0p05.F
                                                                    genes.expression.small.DEG.FDR0p05.F


 dim(genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.UP.by.KH7)
```

```
## [1] 1440  104
```

```
 dim(genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.DOWN.by.KH7)
```

```
## [1] 1324  104
```

```
 dim(genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.REG.by.KH7)
```

```
## [1] 2764  104
```

```
 write.table(genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.REG.by.KH7,
             file=paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.", "KH
                        ".only.DEG",  sep=""),
             quote=F,
             sep="\t",
             row.names = FALSE,
             col.names = TRUE)

#############################################################################################
#############################################################################################

### MAKING the DISPLAYS as SCATTER PLOTS
### PDF
### PNG
### with limma
### with ggplot2

#############################################################################################
#############################################################################################

pdf(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "KH7",
           ".display.limma.SCATTER.pdf", sep=""))

   plotWithHighlights(log2(genes.expression.small$DMSO_lane1.FPKM.average),
                      log2(genes.expression.small$KH7.FPKM.average),
                      status=genes.expression.small$KH7.regulated,
                   values=c("U","D"),
                   bg.col="grey",
                   xlim=c(-2,12), ylim=c(-2,12),
                   hl.cex=0.6, cex.main=0.8, cex.lab =0.8,
                   xlab="log2 average FPKM in DMSO",
                   ylab="log2 average FPKM in KH7 treatment",
                   legend= "topright",
                   main=paste("KH7", " regulated genes", sep=""))

dev.off()
```

```
## pdf
##   2
```

```
#############################################################################################

png(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "KH7",
           ".display.limma.SCATTER.png", sep=""))

   plotWithHighlights(log2(genes.expression.small$DMSO_lane1.FPKM.average),
                      log2(genes.expression.small$KH7.FPKM.average),
                      status=genes.expression.small$KH7.regulated,
                   values=c("U","D"),
                   bg.col="grey",
                   xlim=c(-2,12), ylim=c(-2,12),
                   hl.cex=0.6, cex.main=0.8, cex.lab =0.8,
                   xlab="log2 average FPKM in DMSO",
                   ylab="log2 average FPKM in KH7 treatment",
                   legend= "topright",
                   main=paste("KH7", " regulated genes", sep=""))

dev.off()
```

```
## pdf
##   2
```

```
############################################################################################
############################################################################################

pdf(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "KH7",
           ".display.ggplot2.SCATTER.pdf", sep=""))

ggplot(genes.expression.small,
       aes(x=log2(DMSO_lane1.FPKM.average),
           y=log2(KH7.FPKM.average),
           color=KH7.regulated)) +
           geom_point(size=1) +
           theme_bw() +
           xlim(-2, 12) +
           ylim(-2, 12) +
           scale_colour_manual(values = c("grey","D"="green", "U"="red")) +
           labs(x="log2 average FPKM in DMSO",
                y="log2 average FPKM in KH7") +
           ggtitle(paste("KH7", " regulated genes", sep="")) +
           theme(legend.position="bottom",
                 legend.title=element_blank(),
                 legend.key = element_blank())
```

```
## Warning: Removed 102 rows containing missing values (geom_point).
```

```r
dev.off()
```

```
## pdf
##   2
```

```r
#############################################################################################

png(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "KH7",
           ".display.ggplot2.SCATTER.png", sep=""))

ggplot(genes.expression.small,
       aes(x=log2(DMSO_lane1.FPKM.average),
           y=log2(KH7.FPKM.average),
           color=KH7.regulated)) +
           geom_point(size=1) +
           theme_bw() +
           xlim(-2, 12) +
           ylim(-2, 12) +
           scale_colour_manual(values = c("grey","D"="green", "U"="red")) +
           labs(x="log2 average FPKM in DMSO",
                y="log2 average FPKM in KH7") +
           ggtitle(paste("KH7", " regulated genes", sep="")) +
           theme(legend.position="bottom",
                 legend.title=element_blank(),
                 legend.key = element_blank())
```

```
## Warning: Removed 102 rows containing missing values (geom_point).
```

```r
dev.off()
```

```
## pdf
##   2
```

```r
#############################################################################################
#############################################################################################
#############################################################################################
#############################################################################################

### MAKING the DISPLAYS as VOLCANO PLOTS
### PDF
### PNG
### with limma
### with ggplot2

#############################################################################################
#############################################################################################

pdf(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "KH7",
           ".display.limma.VOLCANO.pdf", sep=""))

plotWithHighlights(       genes.expression.small$logFC.KH7,
                   -log10(genes.expression.small$adj.P.Val.KH7),
                   status=genes.expression.small$KH7.regulated,
                   values=c("U","D"),
                   bg.col="grey",
                   xlim=c(-3,3),
                   ylim=c(0,14),
                   hl.cex=0.6, cex.main=0.8, cex.lab =0.8,
                   xlab="log2FC",
                   ylab="-log10 adj.P.Val",
                   legend= "topright",
                   main=paste("KH7", " regulated genes", sep=""))

dev.off()
```

```
## pdf
##   2
```

```
############################################################################################

png(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "KH7",
           ".display.limma.VOLCANO.png", sep=""))

plotWithHighlights(       genes.expression.small$logFC.KH7,
                    -log10(genes.expression.small$adj.P.Val.KH7),
                    status=genes.expression.small$KH7.regulated,
                    values=c("U","D"),
                    bg.col="grey",
                    xlim=c(-3,3),
                    ylim=c(0,14),
                    hl.cex=0.6, cex.main=0.8, cex.lab =0.8,
                    xlab="log2FC",
                    ylab="-log10 adj.P.Val",
                    legend= "topright",
                    main=paste("KH7", " regulated genes", sep=""))

dev.off()
```

```
## pdf
##   2
```

```
############################################################################################
############################################################################################

pdf(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "KH7",
           ".display.ggplot2.VOLCANO.pdf", sep=""))

    ggplot(genes.expression.small,
           aes(x=genes.expression.small$logFC.KH7,
               y=-log10(genes.expression.small$adj.P.Val.KH7),
           color=KH7.regulated)) +
           geom_point(size=1) +
           theme_bw() +
           xlim(-4, 4) +
           ylim(0, 14) +
           scale_colour_manual(values = c("grey","D"="green", "U"="red")) +
           labs(x="log2FC",
                y="-log10 adj.P.Val") +
           ggtitle(paste("KH7", " regulated genes", sep="")) +
           theme(legend.position="bottom",
                 legend.title=element_blank(),
                 legend.key = element_blank())
```

```
## Warning: Removed 185 rows containing missing values (geom_point).
```

```
dev.off()
```

```
## pdf
##   2
```

```
#############################################################################################

png(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "KH7",
           ".display.ggplot2.VOLCANO.png", sep=""))

    ggplot(genes.expression.small,
           aes(x=genes.expression.small$logFC.KH7,
               y=-log10(genes.expression.small$adj.P.Val.KH7),
           color=KH7.regulated)) +
           geom_point(size=1) +
           theme_bw() +
           xlim(-4, 4) +
           ylim(0, 14) +
           scale_colour_manual(values = c("grey","D"="green", "U"="red")) +
           labs(x="log2FC",
                y="-log10 adj.P.Val") +
           ggtitle(paste("KH7", " regulated genes", sep="")) +
           theme(legend.position="bottom",
                 legend.title=element_blank(),
                 legend.key = element_blank())
```

```
## Warning: Removed 185 rows containing missing values (geom_point).
```

```
dev.off()
```

```
## pdf
##   2
```

```
#############################################################################################
#############################################################################################
```

****

5. DISPLAYS of Noc-regulated genes

```
############################################################################################
############################################################################################
#### SHOWING THE GENES that are REGULATED by Noc
#### as SCATTER PLOT
#### as VOLCANO PLOT

########################################## making another COLUMN that depends on REGULATION

genes.expression.small$Noc.regulated <- ""

genes.expression.small$Noc.regulated[( (genes.expression.small$logFC.Noc > 0.58) &
                                       (genes.expression.small$adj.P.Val.Noc < 0.05) &
                                       (genes.expression.small$Noc.FPKM.average > 1) ) ] <- "U"

genes.expression.small$Noc.regulated[( (genes.expression.small$logFC.Noc < -0.58) &
                                       (genes.expression.small$adj.P.Val.Noc < 0.05) &
                                       (genes.expression.small$DMSO_lane1.FPKM.average > 1) )] <- "D"

write.table(genes.expression.small,
            file=paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
                       "Noc", sep=""),
            quote=F,
            sep="\t",
            row.names = FALSE,
            col.names = TRUE)

######################################### computing the number of GENES in each category :

table(genes.expression.small$Noc.regulated)
```

```
##
##         D    U
## 9877 1744 1335
```

```
# head(genes.expression.small$Noc.regulated)

write.table(table(genes.expression.small$Noc.regulated),
            file=paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
                       "Noc", ".a.summary", sep=""),
            quote=F,
            sep="\t",
            row.names = FALSE,
            col.names = TRUE)

############################################################################################
############################################################################################

####### making some smaller dataframes only of the genes that are reg by Noc :

 genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.UP.by.Noc <- subset(genes.expression.small,
                                                 ( (logFC.Noc > 0.58) &
                                                   (adj.P.Val.Noc < 0.05) &
                                                   (Noc.FPKM.average > 1) ) )

 genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.DOWN.by.Noc <- subset(genes.expression.small,
                                                 ( (logFC.Noc < -0.58) &
                                                   (adj.P.Val.Noc < 0.05) &
                                                   (DMSO_lane1.FPKM.average > 1) ) )

 genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.REG.by.Noc <- rbind(genes.expression.small.DEG.FDR0p05.
                                                                    genes.expression.small.DEG.FDR0p05.


 dim(genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.UP.by.Noc)
```

```
## [1] 1335  105
```

```
 dim(genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.DOWN.by.Noc)
```

```
## [1] 1744  105
```

```
 dim(genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.REG.by.Noc)
```

```
## [1] 3079  105
```

```
 write.table(genes.expression.small.DEG.FDR0p05.FC1p5.FPKM1.REG.by.Noc,
             file=paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
                         "Noc",
                        ".only.DEG",  sep=""),
             quote=F,
             sep="\t",
             row.names = FALSE,
             col.names = TRUE)

##############################################################################################
##############################################################################################

### MAKING the DISPLAYS as SCATTER PLOTS
### PDF
### PNG
### with limma
### with ggplot2

##############################################################################################
##############################################################################################

pdf(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "Noc",
           ".display.limma.SCATTER.pdf", sep=""))

   plotWithHighlights(log2(genes.expression.small$DMSO_lane1.FPKM.average),
                      log2(genes.expression.small$Noc.FPKM.average),
                      status=genes.expression.small$Noc.regulated,
                   values=c("U","D"),
                   bg.col="grey",
                   xlim=c(-2,12), ylim=c(-2,12),
                   hl.cex=0.6, cex.main=0.8, cex.lab =0.8,
                   xlab="log2 average FPKM in DMSO",
                   ylab="log2 average FPKM in Noc treatment",
                   legend= "topright",
                   main=paste("Noc", " regulated genes", sep=""))

dev.off()
```

```
## pdf
##   2
```

```
##############################################################################################

png(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "Noc",
           ".display.limma.SCATTER.png", sep=""))

   plotWithHighlights(log2(genes.expression.small$DMSO_lane1.FPKM.average),
                      log2(genes.expression.small$Noc.FPKM.average),
                      status=genes.expression.small$Noc.regulated,
                   values=c("U","D"),
                   bg.col="grey",
                   xlim=c(-2,12), ylim=c(-2,12),
                   hl.cex=0.6, cex.main=0.8, cex.lab =0.8,
                   xlab="log2 average FPKM in DMSO",
                   ylab="log2 average FPKM in Noc treatment",
                   legend= "topright",
                   main=paste("Noc", " regulated genes", sep=""))

dev.off()
```

```
## pdf
##   2
```

```
#############################################################################################
#############################################################################################

pdf(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "Noc",
           ".display.ggplot2.SCATTER.pdf", sep=""))

ggplot(genes.expression.small,
       aes(x=log2(DMSO_lane1.FPKM.average),
           y=log2(Noc.FPKM.average),
           color=Noc.regulated)) +
           geom_point(size=1) +
           theme_bw() +
           xlim(-2, 12) +
           ylim(-2, 12) +
           scale_colour_manual(values = c("grey","D"="green", "U"="red")) +
           labs(x="log2 average FPKM in DMSO",
                y="log2 average FPKM in Noc") +
           ggtitle(paste("Noc", " regulated genes", sep="")) +
           theme(legend.position="bottom",
                 legend.title=element_blank(),
                 legend.key = element_blank())
```

```
## Warning: Removed 187 rows containing missing values (geom_point).
```

```
dev.off()
```

```
## pdf
##   2
```

```
##########################################################################################

png(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "Noc",
           ".display.ggplot2.SCATTER.png", sep=""))

ggplot(genes.expression.small,
       aes(x=log2(DMSO_lane1.FPKM.average),
           y=log2(Noc.FPKM.average),
           color=Noc.regulated)) +
           geom_point(size=1) +
           theme_bw() +
           xlim(-2, 12) +
           ylim(-2, 12) +
           scale_colour_manual(values = c("grey","D"="green", "U"="red")) +
           labs(x="log2 average FPKM in DMSO",
                y="log2 average FPKM in Noc") +
           ggtitle(paste("Noc", " regulated genes", sep="")) +
           theme(legend.position="bottom",
                 legend.title=element_blank(),
                 legend.key = element_blank())
```

```
## Warning: Removed 187 rows containing missing values (geom_point).
```

```
dev.off()
```

```
## pdf
##   2
```

```
##########################################################################################
##########################################################################################
##########################################################################################
##########################################################################################

### MAKING the DISPLAYS as VOLCANO PLOTS
### PDF
### PNG
### with limma
### with ggplot2

##########################################################################################
##########################################################################################

pdf(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "Noc",
           ".display.limma.VOLCANO.pdf", sep=""))

plotWithHighlights(       genes.expression.small$logFC.Noc,
                   -log10(genes.expression.small$adj.P.Val.Noc),
                   status=genes.expression.small$Noc.regulated,
                   values=c("U","D"),
                   bg.col="grey",
                   xlim=c(-3,3),
                   ylim=c(0,14),
                   hl.cex=0.6, cex.main=0.8, cex.lab =0.8,
                   xlab="log2FC",
                   ylab="-log10 adj.P.Val",
                   legend= "topright",
                   main=paste("Noc", " regulated genes", sep=""))

dev.off()
```

```
## pdf
##   2
```

```
############################################################################################

png(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "Noc",
           ".display.limma.VOLCANO.png", sep=""))

plotWithHighlights(       genes.expression.small$logFC.Noc,
                   -log10(genes.expression.small$adj.P.Val.Noc),
                   status=genes.expression.small$Noc.regulated,
                   values=c("U","D"),
                   bg.col="grey",
                   xlim=c(-3,3),
                   ylim=c(0,14),
                   hl.cex=0.6, cex.main=0.8, cex.lab =0.8,
                   xlab="log2FC",
                   ylab="-log10 adj.P.Val",
                   legend= "topright",
                   main=paste("Noc", " regulated genes", sep=""))

dev.off()
```

```
## pdf
##   2
```

```
############################################################################################
############################################################################################

pdf(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "Noc",
           ".display.ggplot2.VOLCANO.pdf", sep=""))

    ggplot(genes.expression.small,
           aes(x=genes.expression.small$logFC.Noc,
               y=-log10(genes.expression.small$adj.P.Val.Noc),
           color=Noc.regulated)) +
           geom_point(size=1) +
           theme_bw() +
           xlim(-4, 4) +
           ylim(0, 14) +
           scale_colour_manual(values = c("grey","D"="green", "U"="red")) +
           labs(x="log2FC",
                y="-log10 adj.P.Val") +
           ggtitle(paste("Noc", " regulated genes", sep="")) +
           theme(legend.position="bottom",
                 legend.title=element_blank(),
                 legend.key = element_blank())
```

```
## Warning: Removed 61 rows containing missing values (geom_point).
```

```
dev.off()
```

```
## pdf
##   2
```

```
#############################################################################################

png(paste("analysis.LIMMA.integrating.all.samples.with.data.table.the.genes.REG.by.",
           "Noc",
           ".display.ggplot2.VOLCANO.png", sep=""))

    ggplot(genes.expression.small,
           aes(x=genes.expression.small$logFC.Noc,
               y=-log10(genes.expression.small$adj.P.Val.Noc),
           color=Noc.regulated)) +
           geom_point(size=1) +
           theme_bw() +
           xlim(-4, 4) +
           ylim(0, 14) +
           scale_colour_manual(values = c("grey","D"="green", "U"="red")) +
           labs(x="log2FC",
                y="-log10 adj.P.Val") +
           ggtitle(paste("Noc", " regulated genes", sep="")) +
           theme(legend.position="bottom",
                 legend.title=element_blank(),
                 legend.key = element_blank())
```

```
## Warning: Removed 61 rows containing missing values (geom_point).
```

```
dev.off()
```

```
## pdf
##   2
```

```
#############################################################################################
#############################################################################################
```

****

IX. OTHER ANALYSES by using ENRICHMENT BROWSER

****

X. OTHER ANALYSES considering all the ISOFORMS

```
### Here it is a script that I have used in order to INTEGRATE the files with ISOFORMS

name <- "the_ISOFORMS.100985_isoforms.gencode.v28.basic.annotation.28aug2018.txt"

############################################################################################
############################################################################################

###### reading the files with the GENE EXPRESSION COUNTS :

genes <- read.delim("the_ISOFORMS.100985_isoforms.gencode.v28.basic.annotation.28aug2018.txt",
                     sep="\t", header=T, stringsAsFactors=F)

head(genes)
dim(genes)

genes.dt <- as.data.table(genes)

head(genes.dt)
dim(genes.dt)   ## 100985      8

###### to integrate these files : reading the files and changing the names of the columns

############################################################################################
############################################################################################
Aph1 <- read.delim("sample.Aph1.rsem.isoforms.results", sep="\t", header=T, stringsAsFactors=F)

Aph1.simple <- data.frame( Aph1.transcript =  Aph1$transcript_id,
                           Aph1.count = Aph1$expected_count,
                           Aph1.TPM =   Aph1$TPM,
                           Aph1.FPKM =  Aph1$FPKM,
                           stringsAsFactors=F)

head(Aph1)
dim(Aph1)

head(Aph1.simple)
dim(Aph1.simple)

############################################################################################
############################################################################################
Aph2 <- read.delim("sample.Aph2.rsem.isoforms.results", sep="\t", header=T, stringsAsFactors=F)

Aph2.simple <- data.frame( Aph2.transcript =  Aph2$transcript_id,
                           Aph2.count = Aph2$expected_count,
                           Aph2.TPM =   Aph2$TPM,
                           Aph2.FPKM =  Aph2$FPKM,
                           stringsAsFactors=F)

head(Aph2)
dim(Aph2)
```

```
head(Aph2.simple)
dim(Aph2.simple)

############################################################################################
############################################################################################
Aph3 <- read.delim("sample.Aph3.rsem.isoforms.results", sep="\t", header=T, stringsAsFactors=F)

Aph3.simple <- data.frame( Aph3.transcript =  Aph3$transcript_id,
                           Aph3.count = Aph3$expected_count,
                           Aph3.TPM =   Aph3$TPM,
                           Aph3.FPKM =  Aph3$FPKM,
                           stringsAsFactors=F)

head(Aph3)
dim(Aph3)

head(Aph3.simple)
dim(Aph3.simple)

############################################################################################
############################################################################################
Aph_KH7_1 <- read.delim("sample.Aph_KH7_1.rsem.isoforms.results", sep="\t", header=T, stringsAsFactors=

Aph_KH7_1.simple <- data.frame( Aph_KH7_1.transcript =  Aph_KH7_1$transcript_id,
                                Aph_KH7_1.count = Aph_KH7_1$expected_count,
                                Aph_KH7_1.TPM =   Aph_KH7_1$TPM,
                                Aph_KH7_1.FPKM =  Aph_KH7_1$FPKM,
                                stringsAsFactors=F)

head(Aph_KH7_1)
dim(Aph_KH7_1)

head(Aph_KH7_1.simple)
dim(Aph_KH7_1.simple)

############################################################################################
############################################################################################
Aph_KH7_2 <- read.delim("sample.Aph_KH7_2.rsem.isoforms.results", sep="\t", header=T, stringsAsFactors=

Aph_KH7_2.simple <- data.frame( Aph_KH7_2.transcript =  Aph_KH7_2$transcript_id,
                                Aph_KH7_2.count = Aph_KH7_2$expected_count,
                                Aph_KH7_2.TPM =   Aph_KH7_2$TPM,
                                Aph_KH7_2.FPKM =  Aph_KH7_2$FPKM,
                                stringsAsFactors=F)

head(Aph_KH7_2)
dim(Aph_KH7_2)

head(Aph_KH7_2.simple)
dim(Aph_KH7_2.simple)

############################################################################################
############################################################################################
```

```
Aph_KH7_3 <- read.delim("sample.Aph_KH7_3.rsem.isoforms.results", sep="\t", header=T, stringsAsFactors=

Aph_KH7_3.simple <- data.frame( Aph_KH7_3.transcript =  Aph_KH7_3$transcript_id,
                                Aph_KH7_3.count = Aph_KH7_3$expected_count,
                                Aph_KH7_3.TPM =   Aph_KH7_3$TPM,
                                Aph_KH7_3.FPKM =  Aph_KH7_3$FPKM,
                                stringsAsFactors=F)

head(Aph_KH7_3)
dim(Aph_KH7_3)

head(Aph_KH7_3.simple)
dim(Aph_KH7_3.simple)

############################################################################################
############################################################################################
DMSO1_lane1 <- read.delim("sample.DMSO1_lane1.rsem.isoforms.results", sep="\t", header=T, stringsAsFact

DMSO1_lane1.simple <- data.frame( DMSO1_lane1.transcript =  DMSO1_lane1$transcript_id,
                                  DMSO1_lane1.count = DMSO1_lane1$expected_count,
                                  DMSO1_lane1.TPM =   DMSO1_lane1$TPM,
                                  DMSO1_lane1.FPKM =  DMSO1_lane1$FPKM,
                                  stringsAsFactors=F)

head(DMSO1_lane1)
dim(DMSO1_lane1)

head(DMSO1_lane1.simple)
dim(DMSO1_lane1.simple)

############################################################################################
############################################################################################
DMSO1_lane2 <- read.delim("sample.DMSO1_lane2.rsem.isoforms.results", sep="\t", header=T, stringsAsFact

DMSO1_lane2.simple <- data.frame( DMSO1_lane2.transcript =  DMSO1_lane2$transcript_id,
                                  DMSO1_lane2.count = DMSO1_lane2$expected_count,
                                  DMSO1_lane2.TPM =   DMSO1_lane2$TPM,
                                  DMSO1_lane2.FPKM =  DMSO1_lane2$FPKM,
                                  stringsAsFactors=F)

head(DMSO1_lane2)
dim(DMSO1_lane2)

head(DMSO1_lane2.simple)
dim(DMSO1_lane2.simple)

############################################################################################
############################################################################################
DMSO2_lane1 <- read.delim("sample.DMSO2_lane1.rsem.isoforms.results", sep="\t", header=T, stringsAsFact

DMSO2_lane1.simple <- data.frame( DMSO2_lane1.transcript =  DMSO2_lane1$transcript_id,
                                  DMSO2_lane1.count = DMSO2_lane1$expected_count,
                                  DMSO2_lane1.TPM =   DMSO2_lane1$TPM,
                                  DMSO2_lane1.FPKM =  DMSO2_lane1$FPKM,
                                  stringsAsFactors=F)

head(DMSO2_lane1)
dim(DMSO2_lane1)

head(DMSO2_lane1.simple)
dim(DMSO2_lane1.simple)

#############################################################################################
#############################################################################################
DMSO2_lane2 <- read.delim("sample.DMSO2_lane2.rsem.isoforms.results", sep="\t", header=T, stringsAsFacto

DMSO2_lane2.simple <- data.frame( DMSO2_lane2.transcript =  DMSO2_lane2$transcript_id,
                                  DMSO2_lane2.count = DMSO2_lane2$expected_count,
                                  DMSO2_lane2.TPM =   DMSO2_lane2$TPM,
                                  DMSO2_lane2.FPKM =  DMSO2_lane2$FPKM,
                                  stringsAsFactors=F)

head(DMSO2_lane2)
dim(DMSO2_lane2)

head(DMSO2_lane2.simple)
dim(DMSO2_lane2.simple)

#############################################################################################
#############################################################################################
DMSO3_lane1 <- read.delim("sample.DMSO3_lane1.rsem.isoforms.results", sep="\t", header=T, stringsAsFacto

DMSO3_lane1.simple <- data.frame( DMSO3_lane1.transcript =  DMSO3_lane1$transcript_id,
                                  DMSO3_lane1.count = DMSO3_lane1$expected_count,
                                  DMSO3_lane1.TPM =   DMSO3_lane1$TPM,
                                  DMSO3_lane1.FPKM =  DMSO3_lane1$FPKM,
                                  stringsAsFactors=F)

head(DMSO3_lane1)
dim(DMSO3_lane1)

head(DMSO3_lane1.simple)
dim(DMSO3_lane1.simple)

#############################################################################################
#############################################################################################
DMSO3_lane2 <- read.delim("sample.DMSO3_lane2.rsem.isoforms.results", sep="\t", header=T, stringsAsFacto

DMSO3_lane2.simple <- data.frame( DMSO3_lane2.transcript =  DMSO3_lane2$transcript_id,
                                  DMSO3_lane2.count = DMSO3_lane2$expected_count,
                                  DMSO3_lane2.TPM =   DMSO3_lane2$TPM,
                                  DMSO3_lane2.FPKM =  DMSO3_lane2$FPKM,
                                  stringsAsFactors=F)

head(DMSO3_lane2)
dim(DMSO3_lane2)
```

```
head(DMSO3_lane2.simple)
dim(DMSO3_lane2.simple)

############################################################################################
############################################################################################
KH7_1 <- read.delim("sample.KH7_1.rsem.isoforms.results", sep="\t", header=T, stringsAsFactors=F)

KH7_1.simple <- data.frame( KH7_1.transcript =  KH7_1$transcript_id,
                            KH7_1.count = KH7_1$expected_count,
                            KH7_1.TPM =   KH7_1$TPM,
                            KH7_1.FPKM =  KH7_1$FPKM,
                                stringsAsFactors=F)

head(KH7_1)
dim(KH7_1)

head(KH7_1.simple)
dim(KH7_1.simple)

############################################################################################
############################################################################################
KH7_2 <- read.delim("sample.KH7_2.rsem.isoforms.results", sep="\t", header=T, stringsAsFactors=F)

KH7_2.simple <- data.frame( KH7_2.transcript =  KH7_2$transcript_id,
                            KH7_2.count = KH7_2$expected_count,
                            KH7_2.TPM =   KH7_2$TPM,
                            KH7_2.FPKM =  KH7_2$FPKM,
                                stringsAsFactors=F)

head(KH7_2)
dim(KH7_2)

head(KH7_2.simple)
dim(KH7_2.simple)

############################################################################################
############################################################################################
KH7_3 <- read.delim("sample.KH7_3.rsem.isoforms.results", sep="\t", header=T, stringsAsFactors=F)

KH7_3.simple <- data.frame( KH7_3.transcript =  KH7_3$transcript_id,
                            KH7_3.count = KH7_3$expected_count,
                            KH7_3.TPM =   KH7_3$TPM,
                            KH7_3.FPKM =  KH7_3$FPKM,
                                stringsAsFactors=F)

head(KH7_3)
dim(KH7_3)

head(KH7_3.simple)
dim(KH7_3.simple)

############################################################################################
############################################################################################
```

```
Noc_1 <- read.delim("sample.Noc_1.rsem.isoforms.results", sep="\t", header=T, stringsAsFactors=F)

Noc_1.simple <- data.frame( Noc_1.transcript =  Noc_1$transcript_id,
                            Noc_1.count = Noc_1$expected_count,
                            Noc_1.TPM =   Noc_1$TPM,
                            Noc_1.FPKM =  Noc_1$FPKM,
                                stringsAsFactors=F)

head(Noc_1)
dim(Noc_1)

head(Noc_1.simple)
dim(Noc_1.simple)

############################################################################################
############################################################################################
Noc_2 <- read.delim("sample.Noc_2.rsem.isoforms.results", sep="\t", header=T, stringsAsFactors=F)

Noc_2.simple <- data.frame( Noc_2.transcript =  Noc_2$transcript_id,
                            Noc_2.count = Noc_2$expected_count,
                            Noc_2.TPM =   Noc_2$TPM,
                            Noc_2.FPKM =  Noc_2$FPKM,
                                stringsAsFactors=F)

head(Noc_2)
dim(Noc_2)

head(Noc_2.simple)
dim(Noc_2.simple)

############################################################################################
############################################################################################
Noc_3 <- read.delim("sample.Noc_3.rsem.isoforms.results", sep="\t", header=T, stringsAsFactors=F)

Noc_3.simple <- data.frame( Noc_3.transcript =  Noc_3$transcript_id,
                            Noc_3.count = Noc_3$expected_count,
                            Noc_3.TPM =   Noc_3$TPM,
                            Noc_3.FPKM =  Noc_3$FPKM,
                                stringsAsFactors=F)

head(Noc_3)
dim(Noc_3)

head(Noc_3.simple)
dim(Noc_3.simple)

############################################################################################
############################################################################################
############################################################################################
############################################################################################

library(data.table)
```

```
### now integrating these data structures ; we can make DATA TABLES :

Aph1.simple.dt <- as.data.table(Aph1.simple)
Aph2.simple.dt <- as.data.table(Aph2.simple)
Aph3.simple.dt <- as.data.table(Aph3.simple)

Aph_KH7_1.simple.dt <- as.data.table(Aph_KH7_1.simple)
Aph_KH7_2.simple.dt <- as.data.table(Aph_KH7_2.simple)
Aph_KH7_3.simple.dt <- as.data.table(Aph_KH7_3.simple)

DMSO1_lane1.simple.dt <- as.data.table(DMSO1_lane1.simple)
DMSO1_lane2.simple.dt <- as.data.table(DMSO1_lane2.simple)

DMSO2_lane1.simple.dt <- as.data.table(DMSO2_lane1.simple)
DMSO2_lane2.simple.dt <- as.data.table(DMSO2_lane2.simple)

DMSO3_lane1.simple.dt <- as.data.table(DMSO3_lane1.simple)
DMSO3_lane2.simple.dt <- as.data.table(DMSO3_lane2.simple)

KH7_1.simple.dt <- as.data.table(KH7_1.simple)
KH7_2.simple.dt <- as.data.table(KH7_2.simple)
KH7_3.simple.dt <- as.data.table(KH7_3.simple)

Noc_1.simple.dt <- as.data.table(Noc_1.simple)
Noc_2.simple.dt <- as.data.table(Noc_2.simple)
Noc_3.simple.dt <- as.data.table(Noc_3.simple)

##############################################################################################
##############################################################################################
### library(data.table)

setkeyv(genes.dt, c('TRANSCRIPT_ID'))

setkeyv(Aph1.simple.dt, c('Aph1.transcript'))
setkeyv(Aph2.simple.dt, c('Aph2.transcript'))
setkeyv(Aph3.simple.dt, c('Aph3.transcript'))

setkeyv(Aph_KH7_1.simple.dt, c('Aph_KH7_1.transcript'))
setkeyv(Aph_KH7_2.simple.dt, c('Aph_KH7_2.transcript'))
setkeyv(Aph_KH7_3.simple.dt, c('Aph_KH7_3.transcript'))

setkeyv(DMSO1_lane1.simple.dt, c('DMSO1_lane1.transcript'))
setkeyv(DMSO1_lane2.simple.dt, c('DMSO1_lane2.transcript'))

setkeyv(DMSO2_lane1.simple.dt, c('DMSO2_lane1.transcript'))
setkeyv(DMSO2_lane2.simple.dt, c('DMSO2_lane2.transcript'))

setkeyv(DMSO3_lane1.simple.dt, c('DMSO3_lane1.transcript'))
setkeyv(DMSO3_lane2.simple.dt, c('DMSO3_lane2.transcript'))

setkeyv(KH7_1.simple.dt, c('KH7_1.transcript'))
setkeyv(KH7_2.simple.dt, c('KH7_2.transcript'))
setkeyv(KH7_3.simple.dt, c('KH7_3.transcript'))
```

```r
setkeyv(Noc_1.simple.dt, c('Noc_1.transcript'))
setkeyv(Noc_2.simple.dt, c('Noc_2.transcript'))
setkeyv(Noc_3.simple.dt, c('Noc_3.transcript'))

####################### to integrate ALL the dataframes :

# expression.Aph123 <- genes.dt[Aph1.simple.dt,][Aph2.simple.dt,][Aph3.simple.dt,]

# expression.Aph_KH7_123 <- genes.dt[Aph_KH7_1.simple.dt,][Aph_KH7_2.simple.dt,][Aph_KH7_3.simple.dt,]

# expression.DMSO <- genes.dt[DMSO1_lane1.simple.dt,][DMSO1_lane2.simple.dt,][DMSO2_lane1.simple.dt,][D

# expression.KH7_123 <- genes.dt[KH7_1.simple.dt,][KH7_2.simple.dt,][KH7_3.simple.dt,]

# expression.Noc_123 <- genes.dt[Noc_1.simple.dt,][Noc_2.simple.dt,][Noc_3.simple.dt,]

expression.all.samples <- genes.dt[DMSO1_lane1.simple.dt,][DMSO1_lane2.simple.dt,][DMSO2_lane1.simple.dt

expression.all.samples
dim(expression.all.samples)
############################################################################################
############################################################################################
###################### to print the RESULTS, where we have integrated ALL the data frames :

name <- "the_ISOFORMS.100985_isoforms.gencode.v28.basic.annotation.28aug2018.txt"

write.table(expression.all.samples,
            file=paste(name, ".INTEGRATED.file.ALL.samples.txt", sep=""),
            sep="\t", quote=FALSE,
            row.names = FALSE, col.names = TRUE)

############################################################################################
############################################################################################
############################################################################################
############################################################################################
```

****