RNAseq analysis CCS

Etienne Loire

23/1/2020

Introduction

This document has been generated with a R notebook. It's purpose is to describe the analysis steps necessary for the results presented in a scientific publication

Our goal is to search for commonly differentialy expressed genes in Aedes aegypti in response to several arbovirus. We will thus used comparison with mock infection responses, at two different infection stages in cell cultures derived from *Aedes aegypti*.

3 replicates of 3 differents controls (mock) and 2 viral infection (dengue, RVF) in cell cultures (lines derived from Aedes aegypti) has been performed. Early (24H) and late (6days) response have been measured by RNAseq sequencing. Fastq reads have been analyzed (Cleaning, Mapping on reference and coverage analysis) have been performed by a third party (Montpellier Genomix Platefrom).

Dataset exploration and quality control

Raw counts tables are present in the "Data" directory under the name "Raw_Counts_RNA-Seq_CetreSossah.txt" Samples are described in the file "sample.csv" in the "Data" directory.

First step is looking at the complete dataset to assess the quality of results

```r
require(tidyverse)
require(edgeR)
require(ggrepel)
require(ggpubr)
require(xlsx)
mytheme = theme_bw()
infos = read.table("Data/sample.csv",sep=",",header=T)
infos = infos %>% mutate(subtype = substring(name,1,nchar(as.character(name))-1))
data  = read.csv("Data/Raw_Counts_RNA-Seq_CetreSossah.txt",sep=",",header=T,row.names = 1)
data %>% dim
```

```
## [1] 19610    30
```

We see that we have raw counts for 19610 genes in 30 samples. First let's filter all genes with expression values not above 0.5 count per millions reads (cpm) in at least three of the samples.

```r
mdata = as.matrix(data)
mdatacpm = cpm(mdata)
abovecpm = mdatacpm > 0.5
table(rowSums(abovecpm))
```

```
##
##    0    1    2    3    4    5    6    7    8    9   10   11   12   13   14
## 8215 1249  566  378  269  215  209  172  157  151  146  120  103  118  114
##   15   16   17   18   19   20   21   22   23   24   25   26   27   28   29
##  116  125  117  123  129  141  137  141  172  167  181  243  321  445  751
##   30
## 4119
```

```
keep = rowSums(abovecpm) >= 3
summary(keep)
```

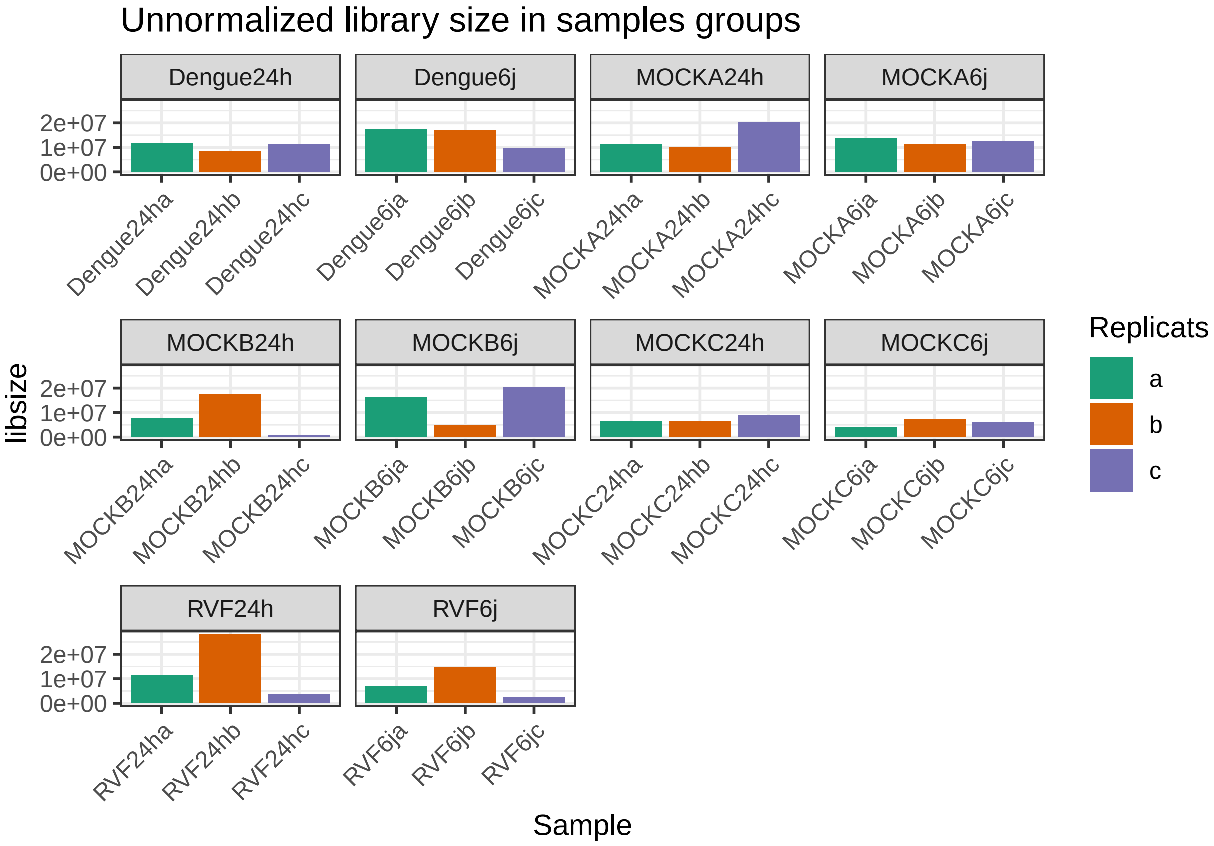
```
##     Mode   FALSE    TRUE
## logical   10030    9580
```

```
filtmdata = mdata[keep,]
```

9580 genes satisfy this threshold

Now we will look at the library size of each samples and look at a multidimensional scaling plot (MDS) to see if genes expression is less variable among replicates than among groups of samples.

```
DG = DGEList(counts = filtmdata)
ggplot(data.frame(name = colnames(DG),libsize = DG$samples$lib.size,type = infos$subtype,time=infos$tim
         arrange(.,sample,time) ) + geom_bar(aes(x=name,y=libsize,fill=sample),stat="identity") +
  facet_wrap(~ type,scale="free_x")  + scale_fill_brewer(name="Replicats",palette ="Dark2") + xlab("Sam
  mytheme + theme(axis.text.x = element_text(angle=45,hjust =1 )) + ggtitle("Unnormalized library size
```



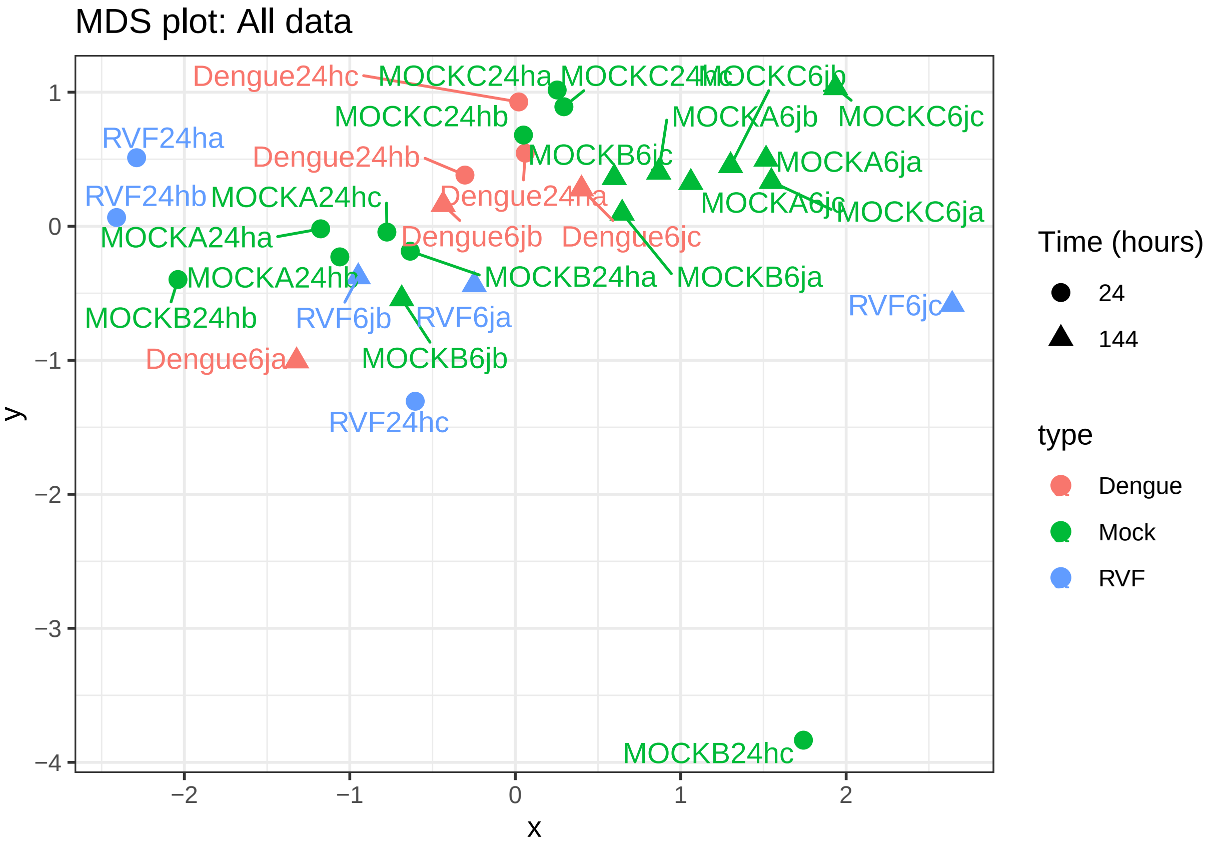
```
ggsave("Figures/library_size.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

We can already see that somes samples seems to have a lower depth of sequencing when compared to others (Notably MOCKB24hc, MOCKB6jb,MOCKC6ja,RVF24hc,RVF6jc). We will see in the MDS plot if this seems to be a problem.

```
mdata = plotMDS(DG,plot=FALSE)
dfmdf=data.frame(x=mdata$x,y=mdata$y)
dfmdf %>% mutate(name = rownames(dfmdf)) %>% left_join(infos,by="name") %>%
ggplot() +
  geom_point(aes(x=x,y=y,color=type,shape=as.factor(time)),size=3) +
  geom_text_repel(aes(x=x,y=y,color = type,label= name)) +
  scale_shape_discrete("Time (hours)") +
  mytheme + ggtitle("MDS plot: All data")
```

```
## Warning: Column `name` joining character vector and factor, coercing into
## character vector
```



```
ggsave("Figures/MDS_All_DATA.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

We see that there is indeed a problem with some of the cited Samples

Fitlering of unreliable samples

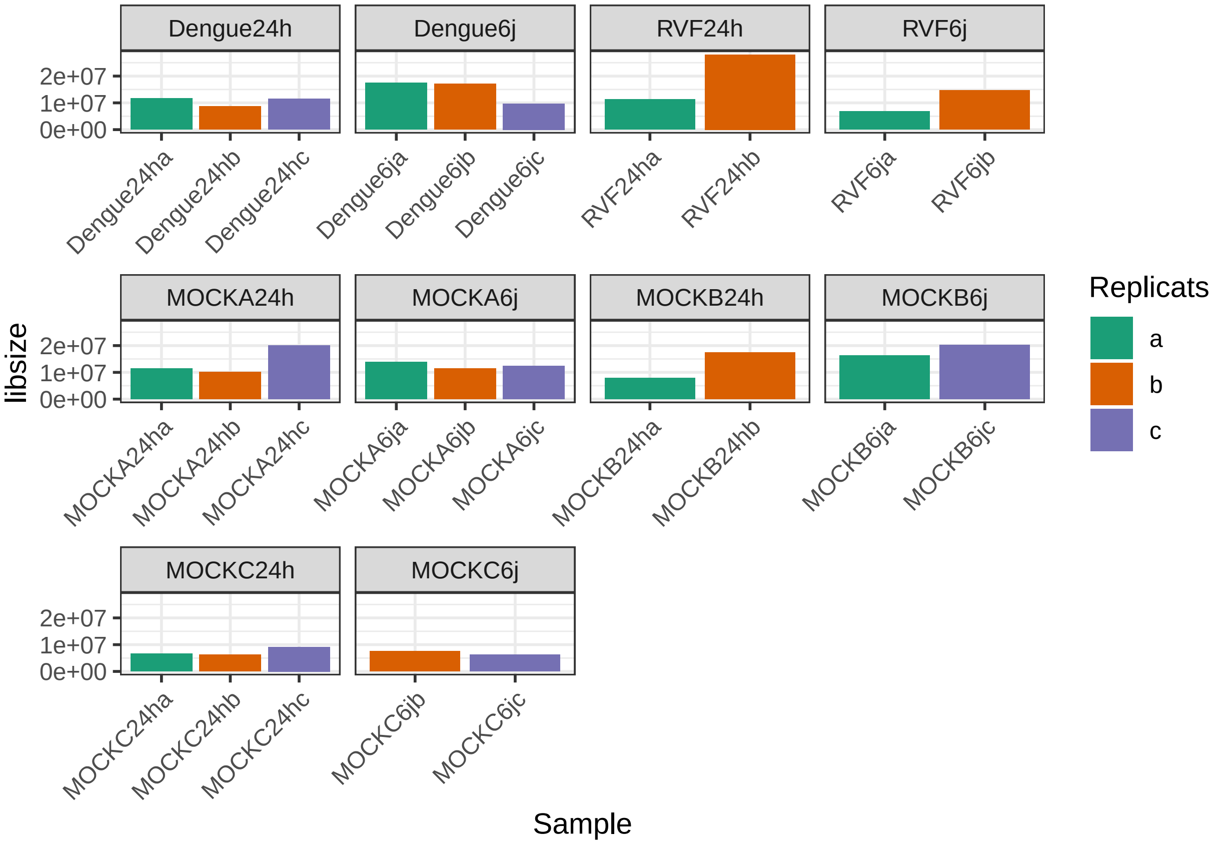
```
toremove  = DG$samples %>% mutate(sample=rownames(.)) %>% filter(lib.size<5000000) %>% select(sample)
toremove
```

```
##      sample
## 1 MOCKB24hc
## 2  MOCKB6jb
## 3  MOCKC6ja
## 4   RVF24hc
## 5    RVF6jc
```

```
fdata = data %>% select(-c("MOCKB6jb","MOCKC6ja","RVF24hc","RVF6jc","MOCKB24hc"))
mdata = as.matrix(fdata)
mdatacpm = cpm(mdata)
abovecpm = mdatacpm > 0.5
keep = rowSums(abovecpm) >= 3
summary(keep)
```

```
##    Mode   FALSE    TRUE
## logical   10294    9316
```

```
filtmdata = mdata[keep,]
DG = DGEList(counts = filtmdata)
DG = calcNormFactors(DG)
infos = infos %>% filter(!(name %in% c("MOCKB6jb","MOCKC6ja","RVF24hc","RVF6jc","MOCKB24hc")))
# Reorder factor
infos$type = factor(infos$type,levels=c("Dengue","RVF","Mock"))
infos$subtype = factor(infos$subtype,levels=c("Dengue24h","Dengue6j","RVF24h","RVF6j",
                                              "MOCKA24h","MOCKA6j","MOCKB24h","MOCKB6j","MOCKC24h","MOC
ggplot(data.frame(name = colnames(DG),libsize = DG$samples$lib.size,type = infos$subtype,time=infos$tim
         arrange(.,sample,time) ) + geom_bar(aes(x=name,y=libsize,fill=sample),stat="identity") +
  facet_wrap(~ type,scale="free_x")  + scale_fill_brewer(name="Replicats",palette ="Dark2") + xlab("Sam
  mytheme + theme(axis.text.x = element_text(angle=45,hjust =1 ))
```



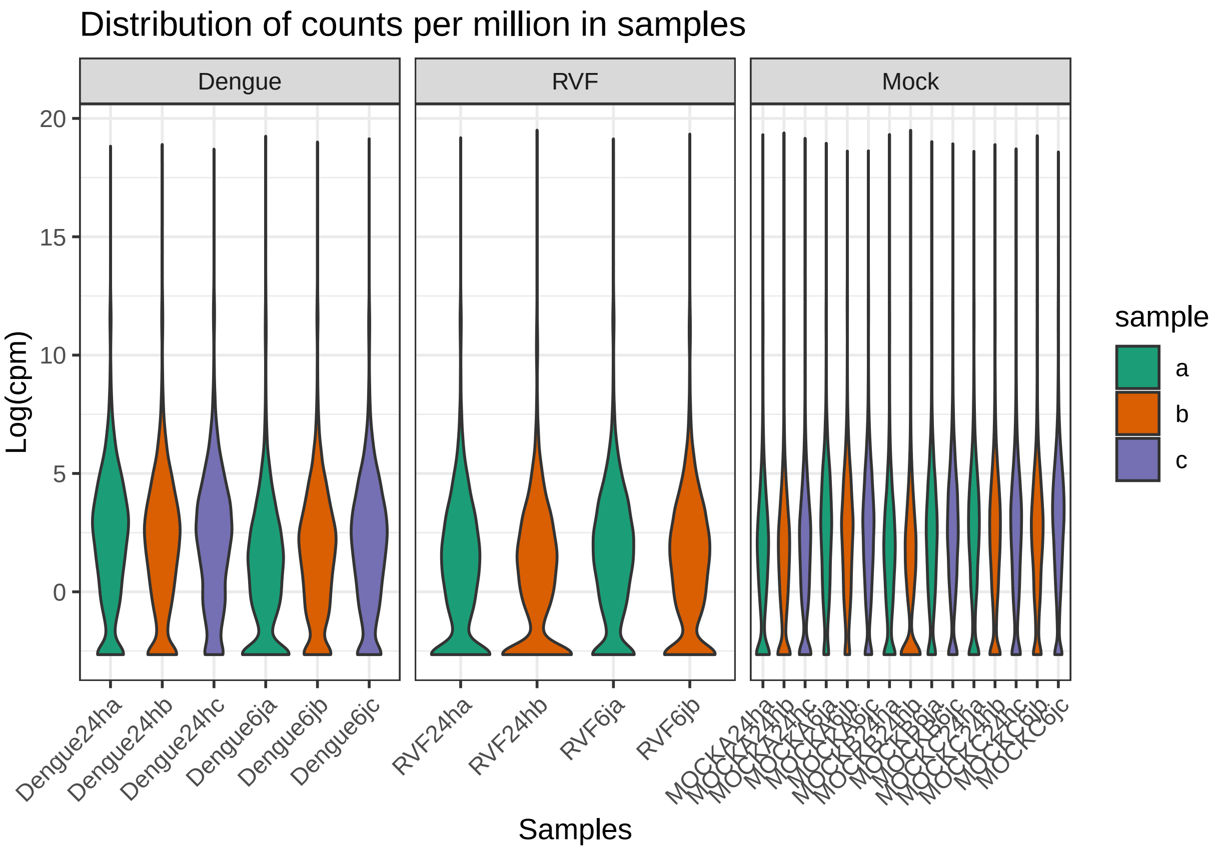
```
ggsave("Figures/filtered_lib_size.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
logcount = cpm(DG$counts,log=T)
infos$name=as.factor(infos$name)
datalogcpm = data.frame(logcount) %>% gather(name,count) %>% left_join(infos,by = "name")
```

```
## Warning: Column `name` joining character vector and factor, coercing into
## character vector
```

```
ggplot(datalogcpm %>% arrange(.,sample,time)) + geom_violin(aes(x=name,y=count,fill=sample))  + facet_w
  ggtitle("Distribution of counts per million in samples")
```

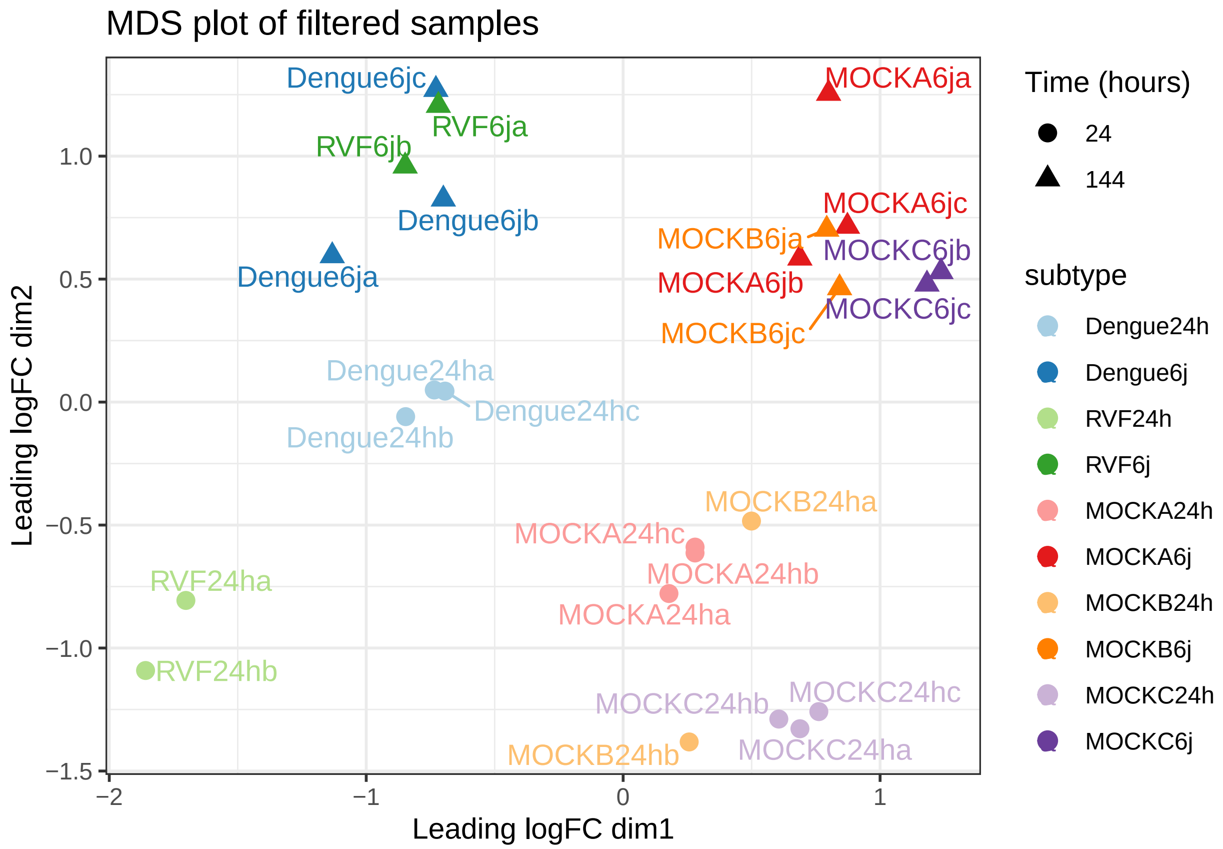


```
ggsave("Figures/LogCPM_violin_count.pdf",height =7,width =15)
```

Selected samples exhibit a good homogenenity among samples after normalization.

```
mdata = plotMDS(DG,top=500,plot=FALSE)
dfmdf=data.frame(x=mdata$x,y=mdata$y)
ggplot(dfmdf %>% mutate(name = rownames(dfmdf)) %>% left_join(infos,by="name" )) +
  geom_point(aes(x=x,y=y,color=subtype,shape=as.factor(time)),size=3) +
  geom_text_repel(aes(x=x,y=y,color = subtype,label= name)) +
  scale_shape_discrete("Time (hours)") +
  scale_color_brewer(type="qual",palette="Paired")  + xlab("Leading logFC dim1") +
  ylab("Leading logFC dim2") + mytheme + ggtitle("MDS plot of filtered samples")
```

```
## Warning: Column `name` joining character vector and factor, coercing into
## character vector
```



```
ggsave("Figures/MDS_GOOD_DATA.pdf")
```

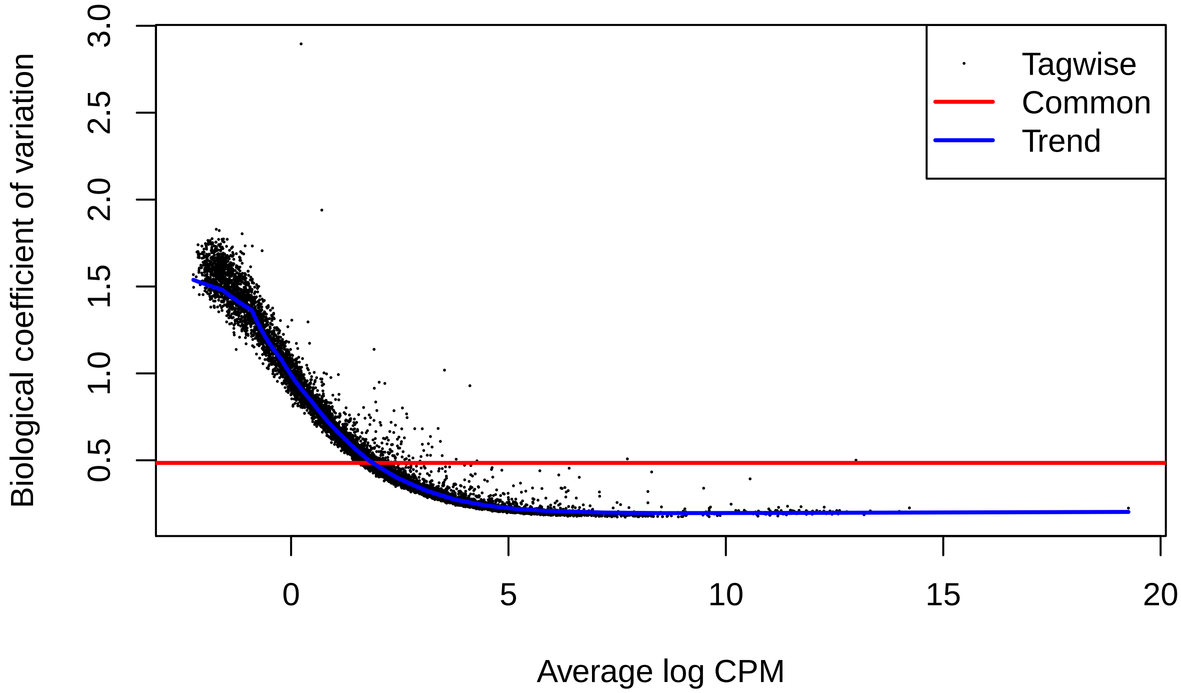
```
## Saving 6.5 x 4.5 in image
```

Here we see a nice dataset, with replicates well grouped and a net seperation of groups of samples. The first dimension separates mock infection from viral infection, and the second dimension separates early (24H) and late (6j) respones. Additionnaly, we see that late responses to viral and mock infections are similar, indicating the possibility to conduct a direct comparison between them to search for common differential expression of genes in response to both viruses. For early response, we need to analyse both viruses separatly and then search for overlap in list of differentially expressed genes.

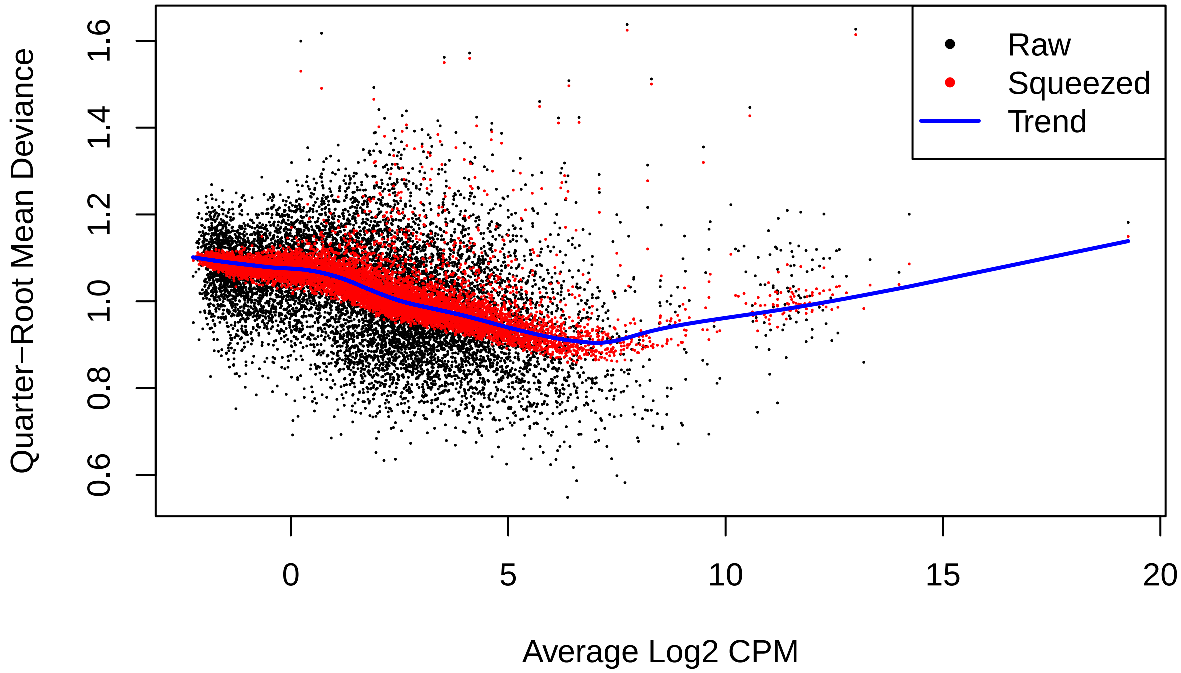
Differential expression analysis

GLM fit

```
# First get gene annotations
Desc = read.csv("Data/Gene_description.txt",sep="\t",header=T)
Long_description = Desc %>% group_by(NCBI.gene.ID,Gene.name,Gene.description) %>% summarize(GOslims = t
colnames(Long_description)[1] = "geneID"
Long_description$geneID =  as.character(Long_description$geneID)
# Design with all biological replicates:
subtype =  as.factor(as.vector(infos$subtype))
design1 = model.matrix(~0+subtype)
DG = estimateDisp(DG,design1,robust = T)
plotBCV(DG)
```



```
fit <- glmQLFit(DG, design1, robust=TRUE)
plotQLDisp(fit)
```

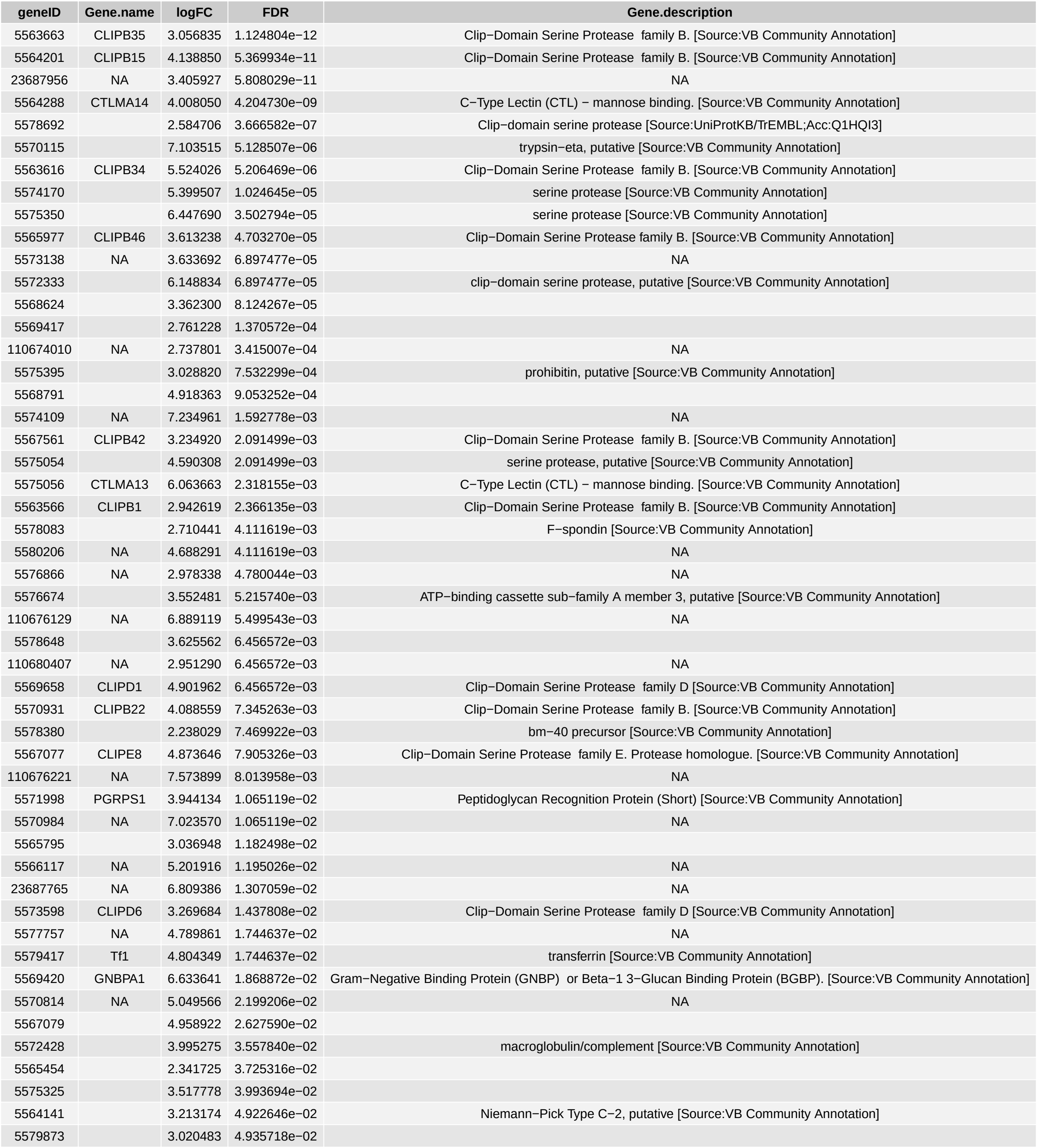


Here we see that the biological variation (among samples in the same groups) is quite low, suggesting that our selection of samples lead to a clean datasets. Trended variation along gene expression is correct (high, then low as expression values increase). The GLM fit for each genes shows the levels of variation among group, with the empirical Bayes shrinkage around trended variation in red. The fit, from my experience, seems quite good.

Early viral response

We will compare the expression values at 24H in virus versus mock infection

```
Early <- makeContrasts((0.5*subtypeDengue24h + 0.5*subtypeRVF24h)-(1/3*subtypeMOCKA24h + 1/3*subtypeMOC
#DG = estimateDisp(DG,design1,robust = T)
fit <- glmQLFit(DG, design1, robust=TRUE)
tr <- glmTreat(fit, contrast=Early, lfc=log2(3))
tmp  = topTags(tr,n=1000,p.value=0.05)
tmp$table$geneID = rownames(tmp$table)
tmp$table %>% left_join(.,Long_description,by="geneID") %>%
  filter(logFC>0) %>%
  write.xlsx(.,file="DE_results.xlsx", sheetName = "UP_DE_genes_earlyvirus_vs_earlymock",row.names=F)
tmp$table %>% left_join(.,Long_description,by="geneID") %>%
  filter(logFC<0) %>%
  write.xlsx(.,file="DE_results.xlsx", sheetName =" DOWN_DE_genes_earlyvirus_vs_earlymock.csv",append=T
upearly = tmp$table %>% left_join(.,Long_description,by="geneID") %>%
  filter(logFC>0) %>% select(c(6,7,1,5,6,8))
upearly %>%  ggtexttable(rows = NULL)
```



geneID	Gene.name	logFC	FDR	Gene.description
5563663	CLIPB35	3.056835	1.124804e−12	Clip−Domain Serine Protease family B. [Source:VB Community Annotation]
5564201	CLIPB15	4.138850	5.369934e−11	Clip−Domain Serine Protease family B. [Source:VB Community Annotation]
23687956	NA	3.405927	5.808029e−11	NA
5564288	CTLMA14	4.008050	4.204730e−09	C−Type Lectin (CTL) − mannose binding. [Source:VB Community Annotation]
5578692		2.584706	3.666582e−07	Clip−domain serine protease [Source:UniProtKB/TrEMBL;Acc:Q1HQI3]
5570115		7.103515	5.128507e−06	trypsin−eta, putative [Source:VB Community Annotation]
5563616	CLIPB34	5.524026	5.206469e−06	Clip−Domain Serine Protease family B. [Source:VB Community Annotation]
5574170		5.399507	1.024645e−05	serine protease [Source:VB Community Annotation]
5575350		6.447690	3.502794e−05	serine protease [Source:VB Community Annotation]
5565977	CLIPB46	3.613238	4.703270e−05	Clip−Domain Serine Protease family B. [Source:VB Community Annotation]
5573138	NA	3.633692	6.897477e−05	NA
5572333		6.148834	6.897477e−05	clip−domain serine protease, putative [Source:VB Community Annotation]
5568624		3.362300	8.124267e−05	
5569417		2.761228	1.370572e−04	
110674010	NA	2.737801	3.415007e−04	NA
5575395		3.028820	7.532299e−04	prohibitin, putative [Source:VB Community Annotation]
5568791		4.918363	9.053252e−04	
5574109	NA	7.234961	1.592778e−03	NA
5567561	CLIPB42	3.234920	2.091499e−03	Clip−Domain Serine Protease family B. [Source:VB Community Annotation]
5575054		4.590308	2.091499e−03	serine protease, putative [Source:VB Community Annotation]
5575056	CTLMA13	6.063663	2.318155e−03	C−Type Lectin (CTL) − mannose binding. [Source:VB Community Annotation]
5563566	CLIPB1	2.942619	2.366135e−03	Clip−Domain Serine Protease family B. [Source:VB Community Annotation]
5578083		2.710441	4.111619e−03	F−spondin [Source:VB Community Annotation]
5580206	NA	4.688291	4.111619e−03	NA
5576866	NA	2.978338	4.780044e−03	NA
5576674		3.552481	5.215740e−03	ATP−binding cassette sub−family A member 3, putative [Source:VB Community Annotation]
110676129	NA	6.889119	5.499543e−03	NA
5578648		3.625562	6.456572e−03	
110680407	NA	2.951290	6.456572e−03	NA
5569658	CLIPD1	4.901962	6.456572e−03	Clip−Domain Serine Protease family D [Source:VB Community Annotation]
5570931	CLIPB22	4.088559	7.345263e−03	Clip−Domain Serine Protease family B. [Source:VB Community Annotation]
5578380		2.238029	7.469922e−03	bm−40 precursor [Source:VB Community Annotation]
5567077	CLIPE8	4.873646	7.905326e−03	Clip−Domain Serine Protease family E. Protease homologue. [Source:VB Community Annotation]
110676221	NA	7.573899	8.013958e−03	NA
5571998	PGRPS1	3.944134	1.065119e−02	Peptidoglycan Recognition Protein (Short) [Source:VB Community Annotation]
5570984	NA	7.023570	1.065119e−02	NA
5565795		3.036948	1.182498e−02	
5566117	NA	5.201916	1.195026e−02	NA
23687765	NA	6.809386	1.307059e−02	NA
5573598	CLIPD6	3.269684	1.437808e−02	Clip−Domain Serine Protease family D [Source:VB Community Annotation]
5577757	NA	4.789861	1.744637e−02	NA
5579417	Tf1	4.804349	1.744637e−02	transferrin [Source:VB Community Annotation]
5569420	GNBPA1	6.633641	1.868872e−02	Gram−Negative Binding Protein (GNBP) or Beta−1 3−Glucan Binding Protein (BGBP). [Source:VB Community Annotation]
5570814	NA	5.049566	2.199206e−02	NA
5567079		4.958922	2.627590e−02	
5572428		3.995275	3.557840e−02	macroglobulin/complement [Source:VB Community Annotation]
5565454		2.341725	3.725316e−02	
5575325		3.517778	3.993694e−02	
5564141		3.213174	4.922646e−02	Niemann−Pick Type C−2, putative [Source:VB Community Annotation]
5579873		3.020483	4.935718e−02	

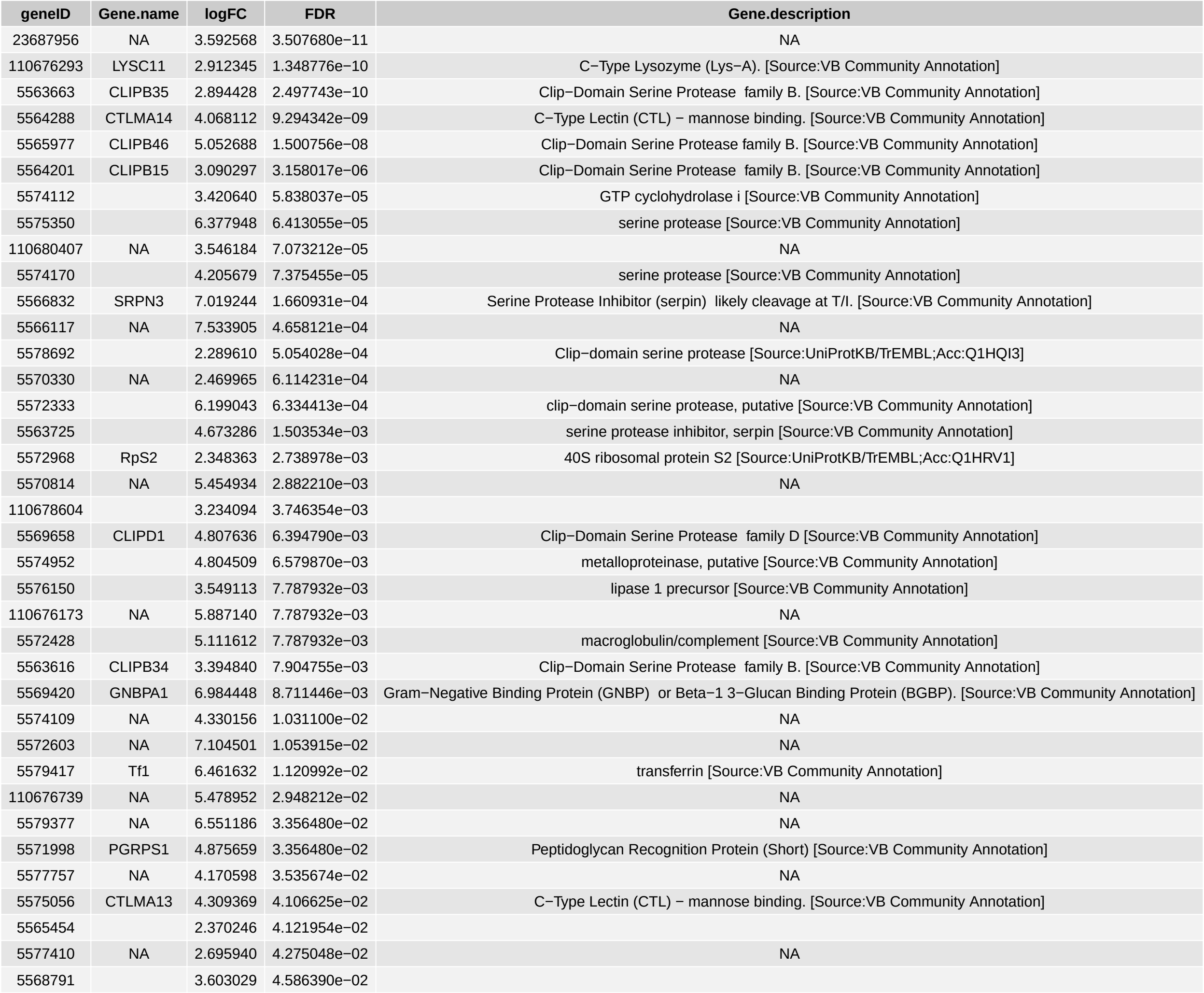
```
ggsave("Figures/UP_EARLY.pdf")
```

```
## Saving 15 x 20 in image
```

Up-relgulated genes following viral infection are, for the most part, related to native immune defense. Clip-domain Serine protease are clearly overrepresented (see https://www.ncbi.nlm.nih.gov/pubmed/26688791) as well as the prohibitin (https://www.ncbi.nlm.nih.gov/pubmed/20674955) and C-type lectin (https://www.ncbi.nlm.nih.gov/pubmed/20674955). Awesome.

Late viral response:

```
Late <- makeContrasts((0.5*subtypeDengue6j + 0.5*subtypeRVF6j)-(1/3*subtypeMOCKA6j + 1/3*subtypeMOCKB6j
tr <- glmTreat(fit, contrast=Late, lfc=log2(3))
tmp  = topTags(tr,n=1000,p.value=0.05)
tmp$table$geneID = rownames(tmp$table)
tmp$table %>% left_join(.,Long_description,by="geneID") %>%
  filter(logFC>0) %>%
  write.xlsx(.,file="DE_results.xlsx", sheetName = "UP_DE_genes_latevirus_vs_latemock",append=T,row.name
tmp$table %>% left_join(.,Long_description,by="geneID") %>%
  filter(logFC<0) %>%
  write.xlsx(.,file="DE_results.xlsx", sheetName ="DOWN_DE_genes_latevirus_vs_latemock",append=T,row.nam
uplate = tmp$table %>% left_join(.,Long_description,by="geneID") %>%
  filter(logFC>0) %>% select(c(6,7,1,5,6,8))
uplate %>%  ggtexttable(rows = NULL)
```



geneID	Gene.name	logFC	FDR	Gene.description
23687956	NA	3.592568	3.507680e−11	NA
110676293	LYSC11	2.912345	1.348776e−10	C−Type Lysozyme (Lys−A). [Source:VB Community Annotation]
5563663	CLIPB35	2.894428	2.497743e−10	Clip−Domain Serine Protease family B. [Source:VB Community Annotation]
5564288	CTLMA14	4.068112	9.294342e−09	C−Type Lectin (CTL) − mannose binding. [Source:VB Community Annotation]
5565977	CLIPB46	5.052688	1.500756e−08	Clip−Domain Serine Protease family B. [Source:VB Community Annotation]
5564201	CLIPB15	3.090297	3.158017e−06	Clip−Domain Serine Protease family B. [Source:VB Community Annotation]
5574112		3.420640	5.838037e−05	GTP cyclohydrolase i [Source:VB Community Annotation]
5575350		6.377948	6.413055e−05	serine protease [Source:VB Community Annotation]
110680407	NA	3.546184	7.073212e−05	NA
5574170		4.205679	7.375455e−05	serine protease [Source:VB Community Annotation]
5566832	SRPN3	7.019244	1.660931e−04	Serine Protease Inhibitor (serpin) likely cleavage at T/I. [Source:VB Community Annotation]
5566117	NA	7.533905	4.658121e−04	NA
5578692		2.289610	5.054028e−04	Clip−domain serine protease [Source:UniProtKB/TrEMBL;Acc:Q1HQI3]
5570330	NA	2.469965	6.114231e−04	NA
5572333		6.199043	6.334413e−04	clip−domain serine protease, putative [Source:VB Community Annotation]
5563725		4.673286	1.503534e−03	serine protease inhibitor, serpin [Source:VB Community Annotation]
5572968	RpS2	2.348363	2.738978e−03	40S ribosomal protein S2 [Source:UniProtKB/TrEMBL;Acc:Q1HRV1]
5570814	NA	5.454934	2.882210e−03	NA
110678604		3.234094	3.746354e−03	
5569658	CLIPD1	4.807636	6.394790e−03	Clip−Domain Serine Protease family D [Source:VB Community Annotation]
5574952		4.804509	6.579870e−03	metalloproteinase, putative [Source:VB Community Annotation]
5576150		3.549113	7.787932e−03	lipase 1 precursor [Source:VB Community Annotation]
110676173	NA	5.887140	7.787932e−03	NA
5572428		5.111612	7.787932e−03	macroglobulin/complement [Source:VB Community Annotation]
5563616	CLIPB34	3.394840	7.904755e−03	Clip−Domain Serine Protease family B. [Source:VB Community Annotation]
5569420	GNBPA1	6.984448	8.711446e−03	Gram−Negative Binding Protein (GNBP) or Beta−1 3−Glucan Binding Protein (BGBP). [Source:VB Community Annotation]
5574109	NA	4.330156	1.031100e−02	NA
5572603	NA	7.104501	1.053915e−02	NA
5579417	Tf1	6.461632	1.120992e−02	transferrin [Source:VB Community Annotation]
110676739	NA	5.478952	2.948212e−02	NA
5579377	NA	6.551186	3.356480e−02	NA
5571998	PGRPS1	4.875659	3.356480e−02	Peptidoglycan Recognition Protein (Short) [Source:VB Community Annotation]
5577757	NA	4.170598	3.535674e−02	NA
5575056	CTLMA13	4.309369	4.106625e−02	C−Type Lectin (CTL) − mannose binding. [Source:VB Community Annotation]
5565454		2.370246	4.121954e−02	
5577410	NA	2.695940	4.275048e−02	NA
5568791		3.603029	4.586390e−02	

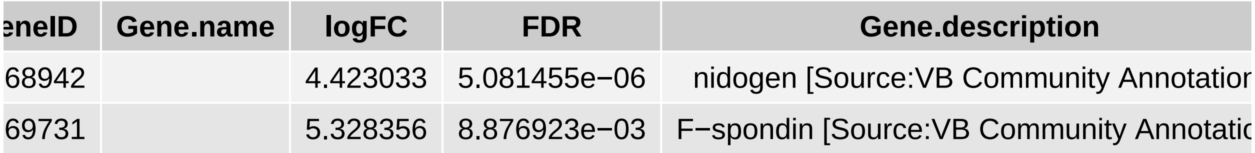
```
ggsave("Figures/UP_LATE.pdf")
```

```
## Saving 15 x 20 in image
```

Very intersingly, many up regulated genes in the early response are also upregulated 6 days post infection relative to control: Clip-domain serine protease, prohibitin, C-type lectin notably. We see the additional presence of Niemann-Pick type C family genes - already shown as related to dengue infections https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3935818/ and macroglobulin/complement (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4767563/ !) Transferrin is also well known to be involved in viral response ! https://www.annualreviews.org/doi/abs/10.1146/annurev-nutr-082117-051749

Late versus early response

```
Early <- makeContrasts((0.5*subtypeDengue6j + 0.5*subtypeRVF6j)-(0.5*subtypeDengue24h + 0.5*subtypeRVF24
DG = estimateDisp(DG,design1,robust = T)
fit <- glmQLFit(DG, design1, robust=TRUE)
tr <- glmTreat(fit, contrast=Early, lfc=log2(3))
tmp  = topTags(tr,n=1000,p.value=0.05)
tmp$table$geneID = rownames(tmp$table)
tmp$table %>% left_join(.,Long_description,by="geneID") %>%
  filter(logFC>0) %>%
  write.xlsx(.,file="DE_results.xlsx", sheetName = "UP_DE_genes_late_virus_vs_early_virus",append=T,row
tmp$table %>% left_join(.,Long_description,by="geneID") %>%
  filter(logFC<0) %>%
  write.xlsx(.,file="DE_results.xlsx", sheetName =" DOWN_DE_genes_late_virus_vs_early_virus",append=T,ro
tmp$table %>% left_join(.,Long_description,by="geneID") %>%
  filter(logFC>0) %>% select(c(6,7,1,5,6,8)) %>%  ggtexttable(rows = NULL)
```



eneID	Gene.name	logFC	FDR	Gene.description
68942		4.423033	5.081455e−06	nidogen [Source:VB Community Annotatior
69731		5.328356	8.876923e−03	F−spondin [Source:VB Community Annotatic

```
ggsave("Figures/UP_virus_early-vs-late.pdf")
```

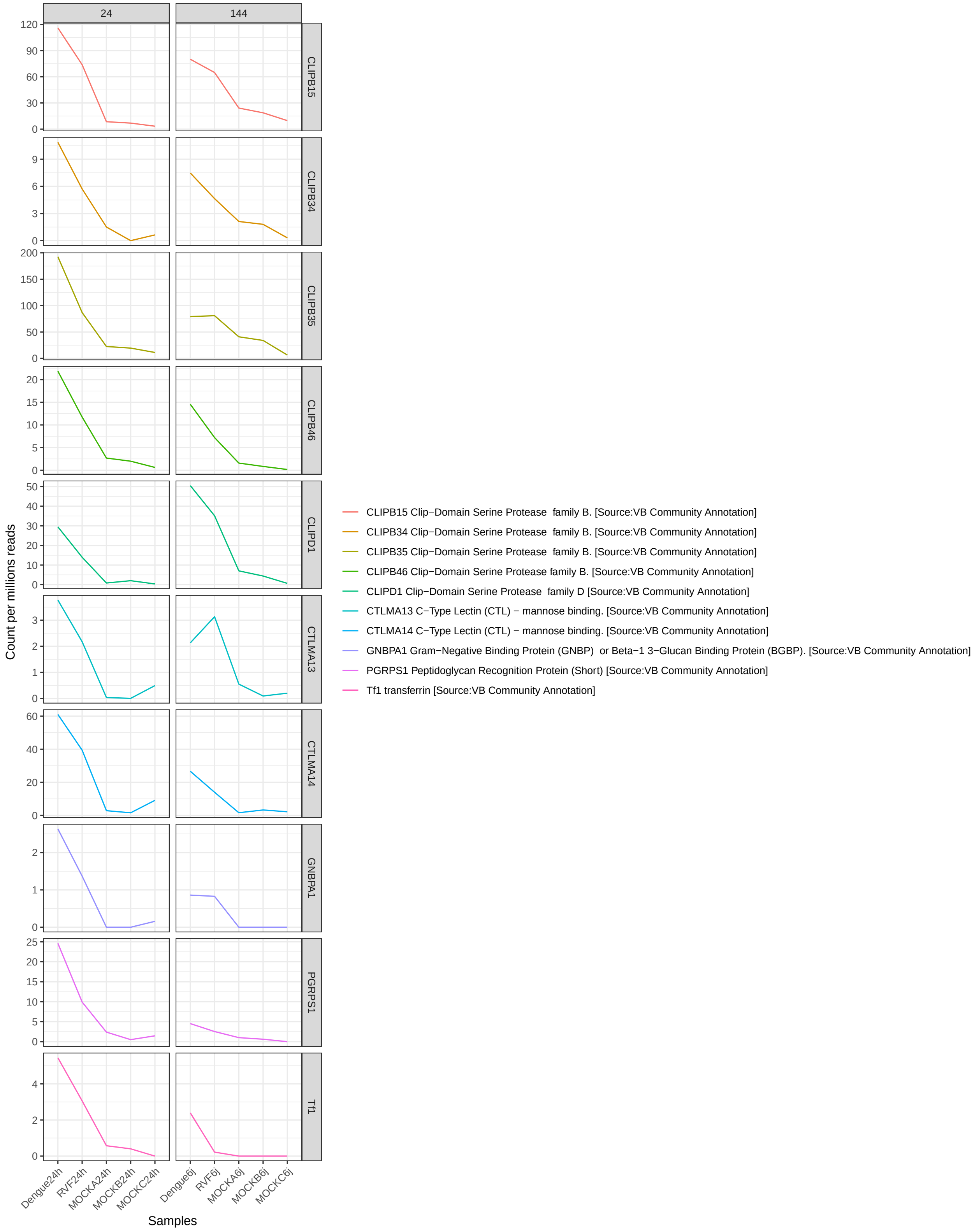
```
## Saving 6.5 x 4.5 in image
```

Only two genes significantly upregulated between early and late response. F-spondine (already seen in previous results but I am not able to) and nidogene. The latter is also related to viral infection (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5636170/ see that spondin modifs are also cited in the article !) but is not upregulated when compared to mock infections in our data.

Figures to check level of expression of UP genes (early and late)

```
eg = upearly %>% filter(Gene.name !="NA" & Gene.name !="") %>% select(geneID)
el = uplate %>% filter(Gene.name !="NA" & Gene.name !="") %>% select(geneID)
liste = intersect(eg,el)
#liste = union(eg,el)
liste = liste %>% mutate(geneID = as.character(geneID))
counts = cpm(DG$counts) %>% as.data.frame
counts$geneID = rownames(counts)
counts %>% filter(geneID %in% liste$geneID) %>% left_join(Long_description %>% select(geneID,Gene.name,G
  gather(sample,cpm,1:25) %>% left_join(infos %>% mutate(sample=name),by="sample") %>%
  group_by(geneID,Gene.name,Gene.description,subtype,time) %>% summarize(mcpm = mean(cpm)) %>%
  mutate(alldesc = paste(Gene.name, Gene.description," ")) %>%
  ggplot() + geom_line(aes(group=Gene.name,x=subtype,y=mcpm,color=alldesc)) + theme_bw() +
  facet_grid(Gene.name ~ time, scale = "free") + theme(axis.text.x = element_text(angle=45,hjust=1)) +
  scale_color_discrete(name="") + ylab("Count per millions reads") + xlab("Samples")
```

```
## Warning: Column `sample` joining character vector and factor, coercing into
## character vector
```



```
ggsave("Figures/Expression_levels_of_up_regulated_genes.pdf")
```

```
## Saving 12 x 15 in image
```

Here I kept list of genes up regulated in viral versus mock infection, late and early, and looked for the intersection of both lists. I filtered to keep only “known” genes. Counts per million in each samples is used to check for actual overexpression in samples infected by a virus.