

ASC_TIMEcourse_limma_nucleolus_analysis

Sarah Hp

6/4/22

Calculate time effect for adipose stem cell differentiation to adipocytes via RNAseq. Here we extract the differentially expressed genes for Pro - D0, D0 - D3 and D3 - D15 for two donors. Previously Pro, D0, D1, D3, and D9 were sequenced in triplicate for two independent donors (GSE176020). New samples: day15 floating and bulk RNAseq.

output files:

- adipogenesis_rpkm_tmm.tab
- adipogenesis_rpkm_tmm_means.tab
- nucleolus_adipogenesis_DE.tab

intermediate files (merge of several experiments):

- late_adipo_and_D1&D2_native_rnaseq.counts
- late_adipo_and_D1&D2_native_rnaseq_info.tab

```
#DE tools
library(limma)
library(edgeR)
library(biomaRt) #annotation
#plotting
library(ggplot2)
library(RUVSeq)
#Data manipulation
library(tidyr)
library(dplyr)
```

```
read.geo <- function(file_url) {
  con <- gzcon(url(file_url))
  txt <- readLines(con)
  return(read.delim(textConnection(txt), skip=1, header=T))
}
```

Merge experiments

Skip ahead to [Start analysis here](#) for DGE analysis from intermediate files.

Load Donor 2 timecourse from GEO

(with fractionally assigned multimapping alignments). Then format the table.

```
d2_file = "https://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE176020&format=file&file=GSE176020%5Fd13%5F"
d2_tab = read.geo(d2_file)
colnames(d2_tab) = gsub("output.01hisat.", "", gsub(".sorted.bam", "", colnames(d2_tab)))
#head(d2_tab) #table is messy so I won't print it here
```

Load file with sample info and match sample information to columns in FC table

```
d2_info = read.delim("../sample_info/donor2_info.csv", header=TRUE, stringsAsFactors = FALSE, sep = ",")
#head(d2_info)

#remove unnecessary columns from d2_info
names(d2_info)[names(d2_info)=='Additional..fill.in.2.3.digits.code..max.5..construct/etc...'] = 'donor'
names(d2_info)[names(d2_info)=='Replicate..fill.in.number.'] = 'biorep'
names(d2_info)[names(d2_info)=='timept'] = 'time'
d2_info = d2_info[c('sample_id','time','donor','biorep','Tube.Label')]

#Checked that FC table and d2_info are in the same order
#colnames(d2_tab)
#d2_info$sample_id
```

Load the Donor 1 timecourse:

And generate sample information from the bam file names

```
d1_file = "https://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE176020&format=file&file=GSE176020%5Fmar19%5F"
d1_tab = read.geo(d1_file)
colnames(d1_tab) = gsub("output.01hisat.", "", gsub(".sorted.bam", "", colnames(d1_tab)))
head(d1_tab)

##          Geneid      Chr
## 1 ENSG00000223972 1;1;1;1;1;1;1;1;1
## 2 ENSG00000227232 1;1;1;1;1;1;1;1;1
## 3 ENSG00000278267           1
## 4 ENSG00000243485 1;1;1;1;1
## 5 ENSG00000284332           1
## 6 ENSG00000237613 1;1;1;1;1
##                                         Start
## 1             11869;12010;12179;12613;12613;12975;13221;13221;13453
## 2 14404;15005;15796;16607;16858;17233;17606;17915;18268;24738;29534
## 3                               17369
## 4                         29554;30267;30564;30976;30976
## 5                               30366
## 6                         34554;35245;35277;35721;35721
##                                         End
## 1             12227;12057;12227;12721;12697;13052;13374;14409;13670
## 2 14501;15038;15947;16765;17055;17368;17742;18061;18366;24891;29570
## 3                               17436
## 4                         30039;30667;30667;31109;31097
```

```

## 5                                     30503
## 6                               35174;35481;35481;36073;36081
##          Strand Length 1.KD4NT_S1 10.P6D3_S10 11.P6D9_S11 12.KD3NT_S12
## 1      +;+;+;+;+;+;+;+;+ 1735      1.51      3.06      8.41      4.82
## 2      -;-;-;-;-;-;-;-;- 1351     187.13     223.71     171.04     235.24
## 3                  -     68      0.20      4.70      2.03      6.18
## 4      +;+;+;+;+;+ 1021      0.20      1.14      0.00      1.03
## 5                  +    138      0.00      0.00      0.00      0.00
## 6      -;-;-;-;- 1219     37.19     42.05     25.39     25.61
## 13.P8D0_S13 14.P8D1_S14 15.P8D3_S15 16.P8D9_S16 2.KD5NT_S2 3.P5D0_S3
## 1      3.00      3.34      2.58      3.08      2.87      2.41
## 2      287.59     243.51     187.14     138.19     96.13     189.13
## 3      2.98      2.95      1.86      2.86      0.45      0.17
## 4      1.48      0.70      0.00      0.00      0.00      0.43
## 5      0.00      0.00      0.00      0.00      0.00      0.00
## 6      69.98     55.23     33.27     19.05     26.12     25.11
## 4.P5D1_S4 5.P5D3_S5 6.P5D9_S6 7.KD6NT_S7 8.P6D0_S8 9.P6D1_S9
## 1      1.58      1.18      1.65      2.75      0.67      2.50
## 2      159.41     218.31     135.33     106.75     93.00     143.30
## 3      1.68      1.07      0.48      0.67      0.00      1.37
## 4      0.25      0.52      0.00      0.00      1.22      0.54
## 5      0.00      0.00      0.00      0.00      0.00      0.00
## 6      32.61     33.20     24.30     16.86     30.08     27.54

```

```

#Remove the file system artefacts from the count column names to create a list of the libraries:
sample_id = grep("[[:digit:]]", colnames(d1_tab), value = T)
sample_id

```

```

## [1] "1.KD4NT_S1"   "10.P6D3_S10"  "11.P6D9_S11"  "12.KD3NT_S12" "13.P8D0_S13"
## [6] "14.P8D1_S14"  "15.P8D3_S15"  "16.P8D9_S16"  "2.KD5NT_S2"   "3.P5D0_S3"
## [11] "4.P5D1_S4"    "5.P5D3_S5"   "6.P5D9_S6"   "7.KD6NT_S7"   "8.P6D0_S8"
## [16] "9.P6D1_S9"

```

```

#Next create a sample table with information about samples and time points taken from the filenames.
Tube.Label = gsub("\\.*_S[[:digit:]]+", "", sample_id)
time = gsub(".*(KD|P)[43586]D?", "", gsub("_S.*","",sample_id))
time = factor(gsub("NT","-2",time)) #match factors
biorep = gsub("\\.(KD|P)", "P", gsub("(D[0139]|NT)_S.*","",sample_id))
biorep = gsub("P3", "P8", biorep) #fix biorep error
donor = rep("D1G", 16)

d1_info = data.frame(sample_id, time, donor, biorep, Tube.Label)
d1_info

```

```

##      sample_id time donor biorep Tube.Label
## 1      1.KD4NT_S1  -2  D1G    P4      1
## 2      10.P6D3_S10   3  D1G    P6     10
## 3      11.P6D9_S11   9  D1G    P6     11
## 4      12.KD3NT_S12  -2  D1G    P8     12
## 5      13.P8D0_S13   0  D1G    P8     13
## 6      14.P8D1_S14   1  D1G    P8     14
## 7      15.P8D3_S15   3  D1G    P8     15
## 8      16.P8D9_S16   9  D1G    P8     16

```

```

## 9   2.KD5NT_S2   -2   D1G      P5       2
## 10  3.P5D0_S3    0   D1G      P5       3
## 11  4.P5D1_S4    1   D1G      P5       4
## 12  5.P5D3_S5    3   D1G      P5       5
## 13  6.P5D9_S6    9   D1G      P5       6
## 14  7.KD6NT_S7   -2   D1G      P6       7
## 15  8.P6D0_S8    0   D1G      P6       8
## 16  9.P6D1_S9    1   D1G      P6       9

```

Load New Samples (day 15)

These are from the same donors

```

day15_tab = read.delim("../02featureCounts/day15_bulk_v_float.counts", skip=1, header=T, stringsAsFactors=TRUE)
colnames(day15_tab) = gsub("output.01hisat.", "", gsub(".sorted.bam", "", colnames(day15_tab)))
colnames(day15_tab)

## [1] "Geneid"          "Chr"            "Start"           "End"             "Strand"
## [6] "Length"          "13.21417_S53"  "14.21418_S2"   "15.21419_S9"  "16.21420_S16"
## [11] "17.21421_S24"   "18.21422_S32"  "19.21423_S40"  "20.21424_S47" "21.21425_S54"
## [16] "22.21426_S3"   "23.21427_S10" "24.21428_S17"

day15_info = read.delim("../sample_info/day15_bulk_v_float_info.csv", header=T, sep=",")
day15_info$sample_id = gsub("-", ".", day15_info$sample_id)
day15_info$time = factor(gsub("D", "", day15_info$time))
colnames(day15_info) = gsub("rep", "biorep", colnames(day15_info))
#use cell_type column as experiment delimiter
day15_info$exp = 3

day15_info = day15_info[!(colnames(day15_info) %in% c("bio.condition"))]

colnames(day15_info); day15_info$sample_id

## [1] "time"          "biorep"        "donor"         "separation"   "sample_id"
## [6] "exp"

## [1] "13.21417_S53" "14.21418_S2"  "15.21419_S9"  "16.21420_S16" "17.21421_S24"
## [6] "18.21422_S32" "19.21423_S40" "20.21424_S47" "21.21425_S54" "22.21426_S3"
## [11] "23.21427_S10" "24.21428_S17"

```

Edits to day15 file to fit: - remove D from time; add dash to donor - change column names: rep to biorep; cell_type to exp - remove researcher, bio.condition

Merge read counts and sample info

Merging the fc tables directly. Merge sample information for d1 and d2; remove tube.label column, add exp column (since d1 and d2 have overlapping samples) and add separation = bulk for these experiments.

```

early = merge(d2_tab, d1_tab)
colnames(early); dim(early)

```

```

## [1] "Geneid"           "Chr"              "Start"            "End"
## [5] "Strand"           "Length"           "6.19190_S20"     "7.19191_S1"
## [9] "8.19192_S3"       "9.19193_S6"      "10.19194_S9"    "11.19195_S12"
## [13] "12.19196_S15"    "13.19197_S18"    "14.19198_S21"   "15.19199_S2"
## [17] "1.19200_S5"       "2.19201_S8"      "3.19202_S11"    "4.19203_S14"
## [21] "5.19204_S17"     "20.S2.KD5NT_S16" "21.S7.KD6NT_S19" "22.S12.KD3NT_S22"
## [25] "1.KD4NT_S1"       "10.P6D3_S10"      "11.P6D9_S11"    "12.KD3NT_S12"
## [29] "13.P8D0_S13"      "14.P8D1_S14"      "15.P8D3_S15"    "16.P8D9_S16"
## [33] "2.KD5NT_S2"       "3.P5D0_S3"        "4.P5D1_S4"      "5.P5D3_S5"
## [37] "6.P5D9_S6"        "7.KD6NT_S7"       "8.P6D0_S8"      "9.P6D1_S9"

```

```
## [1] 58735    40
```

```
#Add experiment separator
d2_info$exp = 2
d1_info$exp = 1
colnames(d2_info)
```

```
## [1] "sample_id"    "time"          "donor"         "biorep"        "Tube.Label"
## [6] "exp"
```

```
colnames(d1_info)
```

```
## [1] "sample_id"    "time"          "donor"         "biorep"        "Tube.Label"
## [6] "exp"
```

```
early_info = rbind(d2_info, d1_info)
early_info$separation = "bulk"
```

```
early_info = early_info[colnames(early_info) != "Tube.Label"]
```

```
table(paste(early_info$time, early_info$donor, early_info$exp))
```

```
##
## -2 D1G 1 -2 D1G 2 -2 D2A 2  0 D1G 1  0 D2A 2  1 D1G 1  1 D2A 2  3 D1G 1
##        4            3            3            3            3            3            3            3            3
##  3 D2A 2  9 D1G 1  9 D2A 2
##        3            3            3
```

Merging in day15 now

```
whole = merge(early, day15_tab)
dim(whole)
```

```
## [1] 58735    52
```

```
whole_info = rbind(early_info, day15_info)
str(whole_info); tail(whole_info)
```

```

## 'data.frame':   46 obs. of  6 variables:
## $ sample_id : chr  "6.19190_S20" "7.19191_S1" "8.19192_S3" "9.19193_S6" ...
## $ time      : chr  "-2" "0" "1" "3" ...
## $ donor     : chr  "D2A" "D2A" "D2A" "D2A" ...
## $ biorep    : chr  "7" "7" "7" "7" ...
## $ exp       : num  2 2 2 2 2 2 2 2 2 ...
## $ separation: chr  "bulk" "bulk" "bulk" "bulk" ...

##           sample_id time donor biorep exp separation
## 41 19.21423_S40   15   D1G   P7    3      bulk
## 42 20.21424_S47   15   D1G   P8_2   3      bulk
## 43 21.21425_S54   15   D1G   P8_1   3      bulk
## 44 22.21426_S3    15   D2A   P7    3      bulk
## 45 23.21427_S10   15   D2A   P6    3      bulk
## 46 24.21428_S17   15   D2A   P7    3      bulk

```

Save combined read counts so we don't need to do this again.

```

write.table(whole, "../03limma/late_adipo_and_D1&D2_native_rnaseq.counts", sep="\t", row.names = F)
write.table(whole_info, "../03limma/late_adipo_and_D1&D2_native_rnaseq_info.tab", sep="\t", row.names = F)

```

Start Analysis Here

Load combined read counts from above

```

whole = read.delim("../03limma/late_adipo_and_D1&D2_native_rnaseq.counts", sep="\t", check.names = F)
whole_info = read.delim( "../03limma/late_adipo_and_D1&D2_native_rnaseq_info.tab", sep="\t")

```

Make DGE object

remove additional samples.

```

whole_info$time.donor = paste(whole_info$time, whole_info$donor, sep=".")
whole_info$group = paste(whole_info$time.donor, whole_info$separation, sep=".")

whole_ob = DGEList(counts=data.matrix(whole[grep("[[:digit:]]", colnames(whole))]),
                   genes= whole[c("Geneid", "Length")], samples = whole_info)
rownames(whole_ob$counts) = whole_ob$genes$Geneid
summary (whole_ob)

```

```

##           Length Class      Mode
## counts     2701810 -none-    numeric
## samples      10 data.frame list
## genes        2 data.frame list

```

```

#Check column names are the same
whole_ob$samples$sample_id

```

```

## [1] "6.19190_S20"      "7.19191_S1"       "8.19192_S3"       "9.19193_S6"
## [5] "10.19194_S9"     "11.19195_S12"    "12.19196_S15"    "13.19197_S18"
## [9] "14.19198_S21"     "15.19199_S2"     "1.19200_S5"      "2.19201_S8"
## [13] "3.19202_S11"      "4.19203_S14"     "5.19204_S17"     "20.S2.KD5NT_S16"
## [17] "21.S7.KD6NT_S19"   "22.S12.KD3NT_S22"  "1.KD4NT_S1"      "10.P6D3_S10"
## [21] "11.P6D9_S11"       "12.KD3NT_S12"    "13.P8D0_S13"     "14.P8D1_S14"
## [25] "15.P8D3_S15"       "16.P8D9_S16"      "2.KD5NT_S2"      "3.P5D0_S3"
## [29] "4.P5D1_S4"         "5.P5D3_S5"       "6.P5D9_S6"       "7.KD6NT_S7"
## [33] "8.P6D0_S8"         "9.P6D1_S9"       "13.21417_S53"   "14.21418_S2"
## [37] "15.21419_S9"       "16.21420_S16"    "17.21421_S24"   "18.21422_S32"
## [41] "19.21423_S40"     "20.21424_S47"    "21.21425_S54"   "22.21426_S3"
## [45] "23.21427_S10"     "24.21428_S17"

```

```
colnames(whole_ob)
```

```

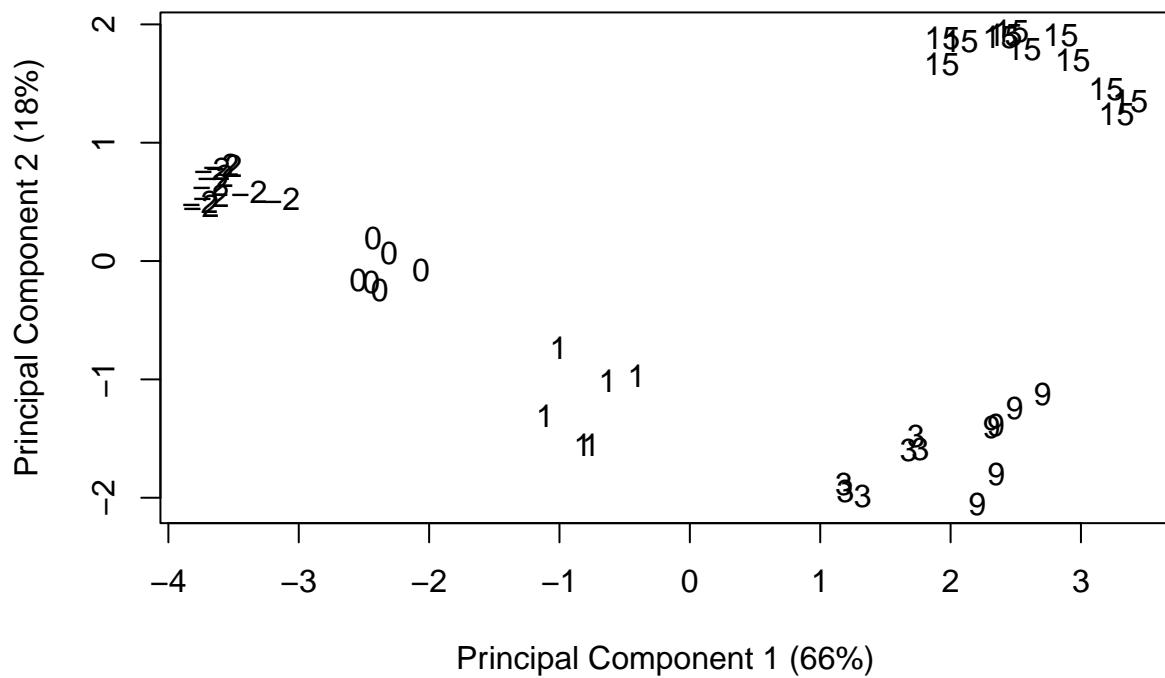
## [1] "6.19190_S20"      "7.19191_S1"       "8.19192_S3"       "9.19193_S6"
## [5] "10.19194_S9"     "11.19195_S12"    "12.19196_S15"    "13.19197_S18"
## [9] "14.19198_S21"     "15.19199_S2"     "1.19200_S5"      "2.19201_S8"
## [13] "3.19202_S11"      "4.19203_S14"     "5.19204_S17"     "20.S2.KD5NT_S16"
## [17] "21.S7.KD6NT_S19"   "22.S12.KD3NT_S22"  "1.KD4NT_S1"      "10.P6D3_S10"
## [21] "11.P6D9_S11"       "12.KD3NT_S12"    "13.P8D0_S13"     "14.P8D1_S14"
## [25] "15.P8D3_S15"       "16.P8D9_S16"      "2.KD5NT_S2"      "3.P5D0_S3"
## [29] "4.P5D1_S4"         "5.P5D3_S5"       "6.P5D9_S6"       "7.KD6NT_S7"
## [33] "8.P6D0_S8"         "9.P6D1_S9"       "13.21417_S53"   "14.21418_S2"
## [37] "15.21419_S9"       "16.21420_S16"    "17.21421_S24"   "18.21422_S32"
## [41] "19.21423_S40"     "20.21424_S47"    "21.21425_S54"   "22.21426_S3"
## [45] "23.21427_S10"     "24.21428_S17"

```

Initial Plots

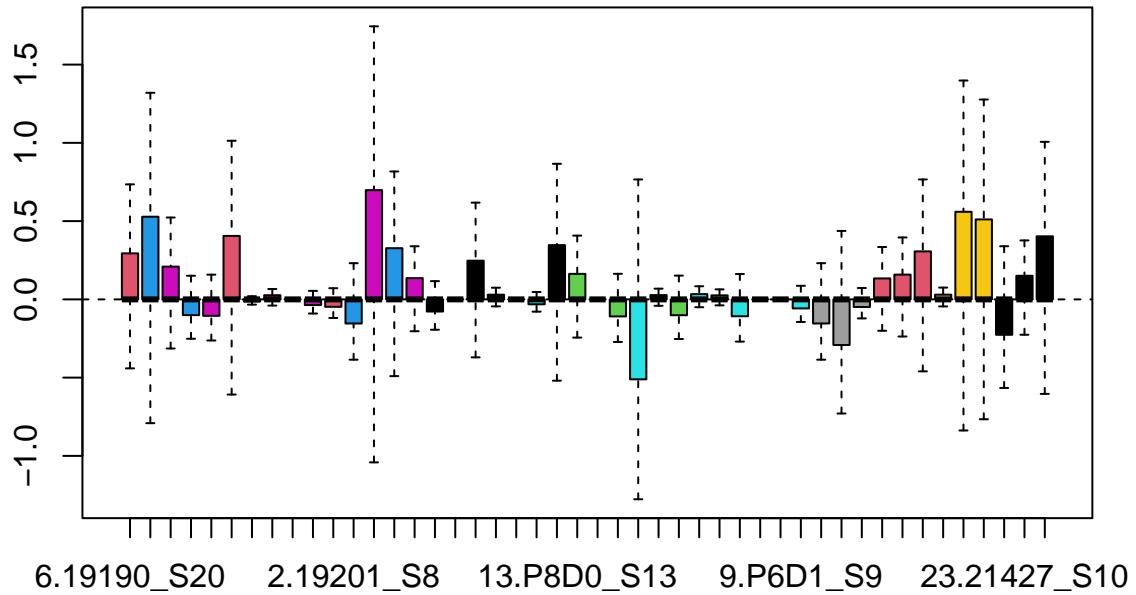
Everything looks good. 1. MDSplot separates timepoints well 2. the RLE plot shows that median counts between libraries are 0, meaning library size and distribution is similar between libraries. Therefore normalisation between experiments is not necessary.

```
plotMDS(whole_ob, labels = whole_ob$samples$time, gene.selection = "common")
```



```
plotRLE(whole_ob$counts, outline=FALSE, col=whole_ob$samples$group, main="Before filtering")
```

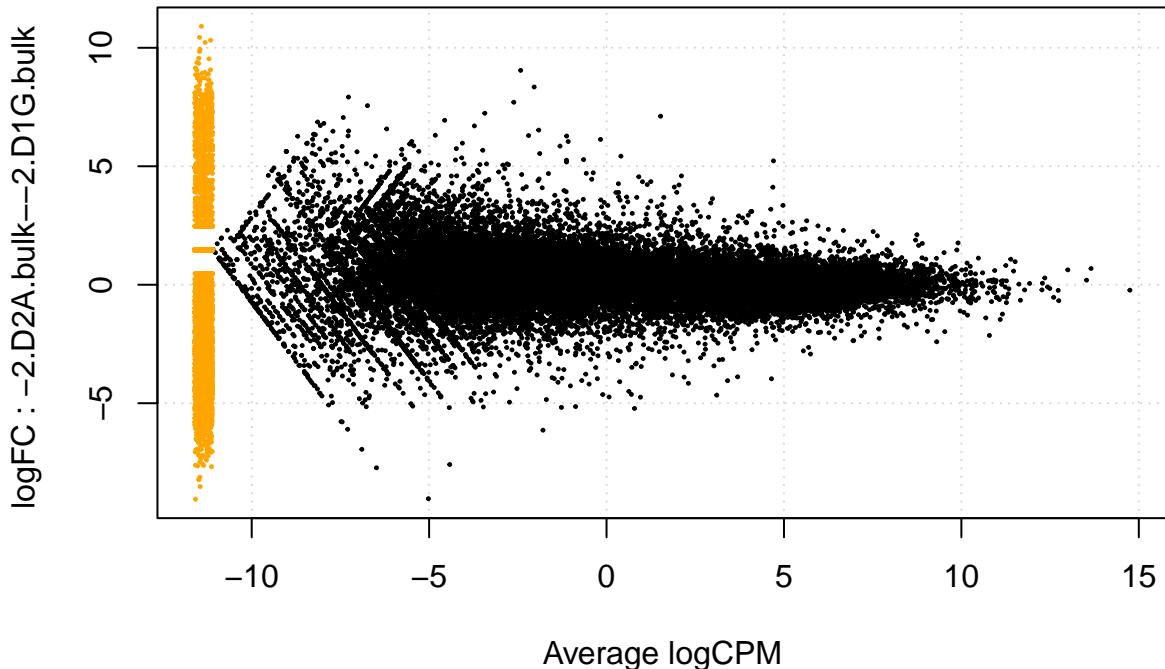
Before filtering



```
## Filtering
```

```
plotSmear(whole_ob, main = "Before Filtering")
```

Before Filtering



```
dim(whole_ob)
```

```
## [1] 58735     46
```

```
filt_series = whole_ob[filterByExpr(whole_ob), , keep.lib.sizes=FALSE]
dim(filt_series) #21174 genes kept (about 1k more genes than the timecourse to day9)
```

```
## [1] 21174     46
```

```
#The filter is approxiamtely 10/ libsize in millions
10/(median(whole_ob$samples$lib.size)/1000000) #0.245 minimum CPM
```

```
## [1] 0.2447707
```

```
#sanity check that gene_names match up
filt_series$counts[filt_series$genes$Geneid == "ENSG00000228630", ] #HOTAIR
```

```
##      6.19190_S20      7.19191_S1      8.19192_S3      9.19193_S6
##      67.00          60.00          61.00          26.00
##      10.19194_S9     11.19195_S12     12.19196_S15     13.19197_S18
##      18.00          39.00          52.00          36.00
##      14.19198_S21    15.19199_S2      1.19200_S5      2.19201_S8
##      41.00          15.00          18.00          46.00
```

```

##      3.19202_S11      4.19203_S14      5.19204_S17 20.S2.KD5NT_S16
##      105.00          37.00          12.00          200.00
## 21.S7.KD6NT_S19 22.S12.KD3NT_S22      1.KD4NT_S1 10.P6D3_S10
##      251.90          313.00          276.00          325.00
##      11.P6D9_S11      12.KD3NT_S12      13.P8D0_S13 14.P8D1_S14
##      96.00           427.00          770.25          462.00
##      15.P8D3_S15      16.P8D9_S16      2.KD5NT_S2  3.P5D0_S3
##      237.00           106.00          196.00          375.00
##      4.P5D1_S4       5.P5D3_S5       6.P5D9_S6  7.KD6NT_S7
##      454.00           438.00          151.00          342.00
##      8.P6D0_S8       9.P6D1_S9       13.21417_S53 14.21418_S2
##      595.00           330.00          71.00           41.00
##      15.21419_S9      16.21420_S16      17.21421_S24 18.21422_S32
##      71.00            26.00           11.00           15.00
##      19.21423_S40     20.21424_S47     21.21425_S54 22.21426_S3
##      339.00           797.12          844.00          26.00
##      23.21427_S10     24.21428_S17
##      78.00            47.00

```

```
filt_series$counts[filt_series$genes$Geneid == "ENSG00000132170",] #PPARG
```

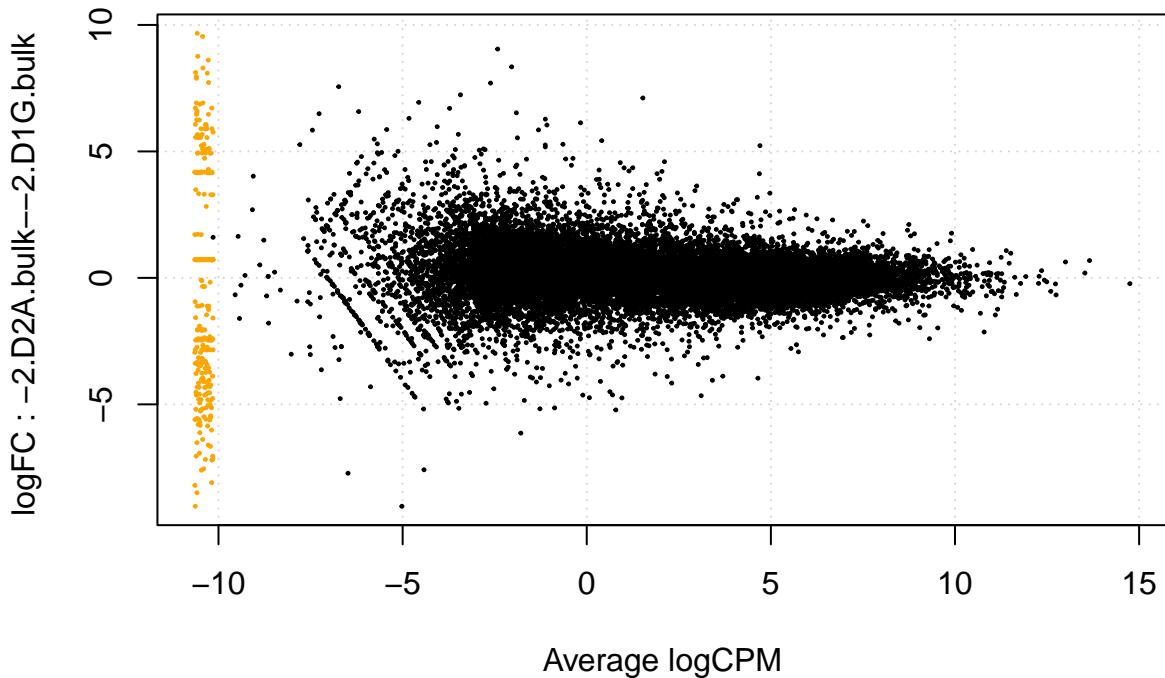
```

##      6.19190_S20      7.19191_S1      8.19192_S3 9.19193_S6
##      1303.10          1667.00          3994.00        6675.10
##      10.19194_S9      11.19195_S12     12.19196_S15 13.19197_S18
##      7365.24           1319.10          557.12          3194.31
##      14.19198_S21     15.19199_S2      1.19200_S5  2.19201_S8
##      7950.67           9278.10          542.00          671.00
##      3.19202_S11      4.19203_S14     5.19204_S17 20.S2.KD5NT_S16
##      4561.33           8161.83          11953.80         392.20
## 21.S7.KD6NT_S19 22.S12.KD3NT_S22      1.KD4NT_S1 10.P6D3_S10
##      1183.37           1248.50          743.00          3307.83
##      11.P6D9_S11      12.KD3NT_S12     13.P8D0_S13 14.P8D1_S14
##      5122.53           976.17           645.00          1255.00
##      15.P8D3_S15      16.P8D9_S16      2.KD5NT_S2  3.P5D0_S3
##      3011.50           3613.41          537.00          262.00
##      4.P5D1_S4       5.P5D3_S5       6.P5D9_S6  7.KD6NT_S7
##      886.00            3387.60          6195.75          793.00
##      8.P6D0_S8       9.P6D1_S9       13.21417_S53 14.21418_S2
##      537.10            1261.10          20687.05        21560.08
##      15.21419_S9      16.21420_S16      17.21421_S24 18.21422_S32
##      19293.64           30703.87          32751.01        38751.89
##      19.21423_S40     20.21424_S47     21.21425_S54 22.21426_S3
##      10886.47           12895.56          13914.89        8218.48
##      23.21427_S10     24.21428_S17
##      16600.64          18475.29

```

```
plotSmear(filt_series, main = "After Filtering")
```

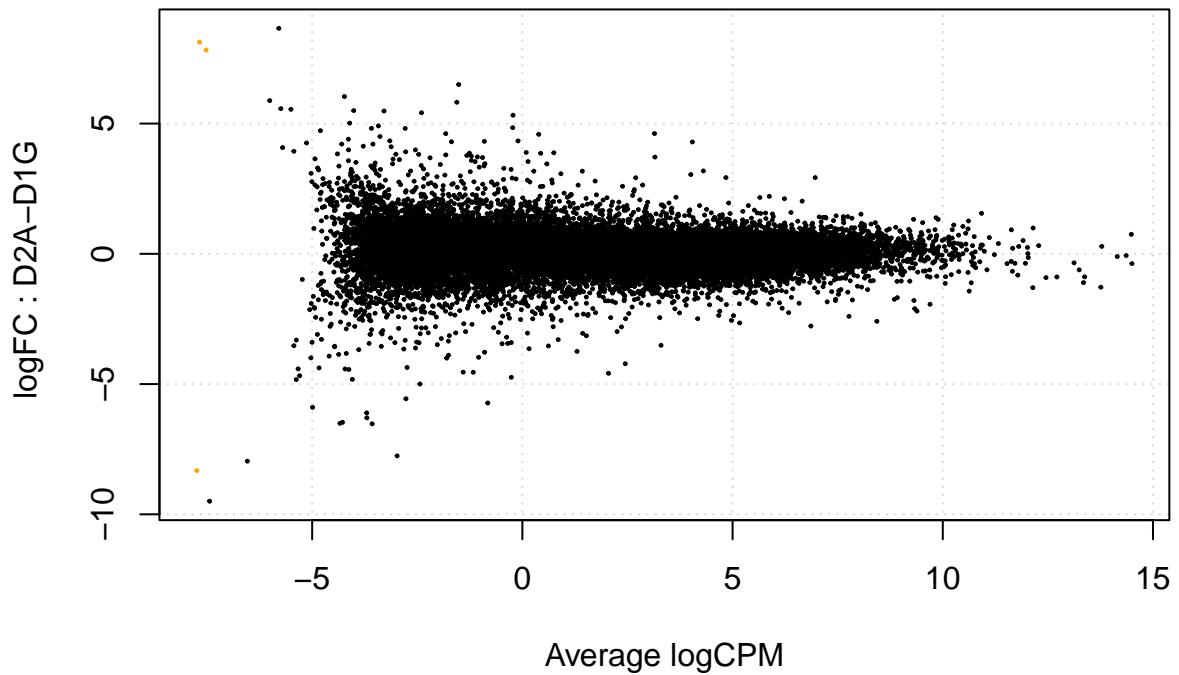
After Filtering



Donor vs donor this looks like very nice filtering (above, at any one time point some genes will be missing).

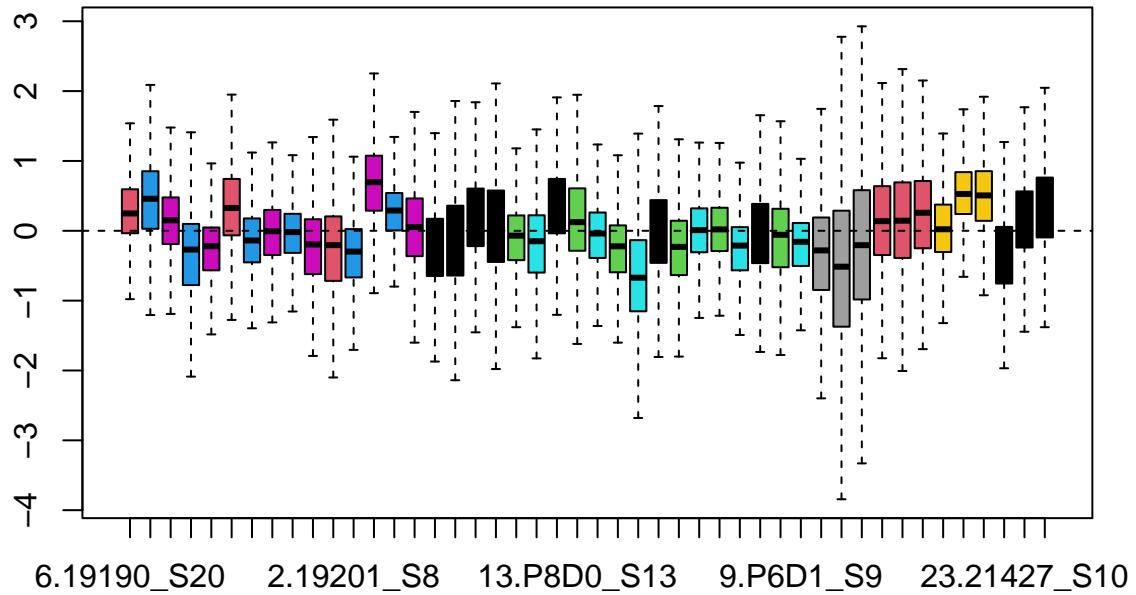
```
#Plot donor v donor -> a few low expression genes still but predominantly good filtering
two_filt = filt_series
two_filt$samples$group = as.factor(filt_series$samples$donor)
plotSmear(two_filt, main = "After Filtering")
```

After Filtering

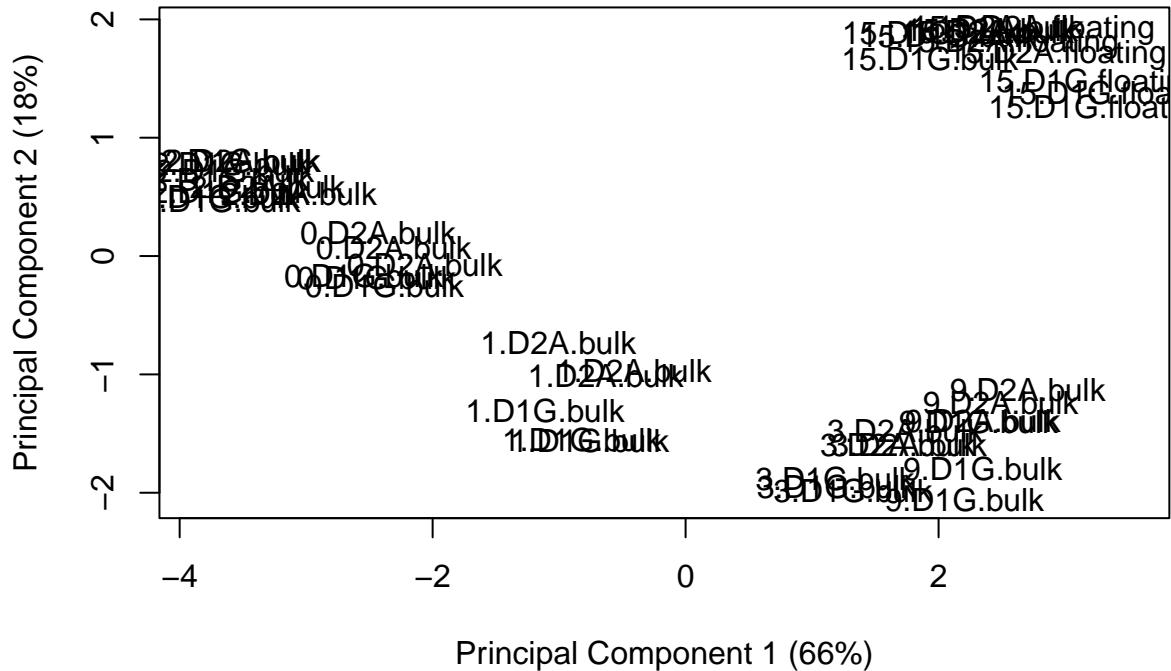


```
#library sizes are different, but relative log expression is
#now ~ normally distributed within libraries
plotRLE(filter_series$counts, outline=FALSE,
        col=filter_series$samples$group, main="After filtering")
```

After filtering



```
plotMDS(filt_series, labels = filt_series$samples$group, gene.selection = "common")
```



Because we're mainly interested in days Pro, 0, 3 and 15 (bulk) I'm going to create a filter vector that eliminates genes that would drop out if we only used those samples. The limma voom method requires adequate filtering of low counts genes to maintain conservative p-values.

```
short_ob = whole_ob[, whole_ob$samples$time %in% c(-2,0,3,15) &
                     whole_ob$samples$separation == "bulk"]
short_ob$samples
```

##	group	lib.size	norm.factors	sample_id	time	donor
## 6.19190_S20	-2.D2A.bulk	43343287	1	6.19190_S20	-2	D2A
## 7.19191_S1	0.D2A.bulk	53954862	1	7.19191_S1	0	D2A
## 9.19193_S6	3.D2A.bulk	27650074	1	9.19193_S6	3	D2A
## 11.19195_S12	-2.D2A.bulk	44069380	1	11.19195_S12	-2	D2A
## 12.19196_S15	0.D2A.bulk	33748947	1	12.19196_S15	0	D2A
## 14.19198_S21	3.D2A.bulk	38990845	1	14.19198_S21	3	D2A
## 1.19200_S5	-2.D2A.bulk	24715231	1	1.19200_S5	-2	D2A
## 2.19201_S8	0.D2A.bulk	25887790	1	2.19201_S8	0	D2A
## 4.19203_S14	3.D2A.bulk	51335376	1	4.19203_S14	3	D2A
## 20.S2.KD5NT_S16	-2.D1G.bulk	28227567	1	20.S2.KD5NT_S16	-2	D1G
## 21.S7.KD6NT_S19	-2.D1G.bulk	30639099	1	21.S7.KD6NT_S19	-2	D1G
## 22.S12.KD3NT_S22	-2.D1G.bulk	39314312	1	22.S12.KD3NT_S22	-2	D1G
## 1.KD4NT_S1	-2.D1G.bulk	40875335	1	1.KD4NT_S1	-2	D1G
## 10.P6D3_S10	3.D1G.bulk	37487858	1	10.P6D3_S10	3	D1G
## 12.KD3NT_S12	-2.D1G.bulk	52250725	1	12.KD3NT_S12	-2	D1G
## 13.P8D0_S13	0.D1G.bulk	50473193	1	13.P8D0_S13	0	D1G
## 15.P8D3_S15	3.D1G.bulk	36821008	1	15.P8D3_S15	3	D1G

```

## 2.KD5NT_S2      -2.D1G.bulk 39256967      1      2.KD5NT_S2      -2      D1G
## 3.P5D0_S3      0.D1G.bulk 35885732      1      3.P5D0_S3      0      D1G
## 5.P5D3_S5      3.D1G.bulk 45252924      1      5.P5D3_S5      3      D1G
## 7.KD6NT_S7      -2.D1G.bulk 37829067      1      7.KD6NT_S7      -2      D1G
## 8.P6D0_S8      0.D1G.bulk 38988386      1      8.P6D0_S8      0      D1G
## 19.21423_S40    15.D1G.bulk 43513597     1      19.21423_S40    15      D1G
## 20.21424_S47    15.D1G.bulk 68529570     1      20.21424_S47    15      D1G
## 21.21425_S54    15.D1G.bulk 67338531     1      21.21425_S54    15      D1G
## 22.21426_S3      15.D2A.bulk 28524813     1      22.21426_S3      15      D2A
## 23.21427_S10    15.D2A.bulk 51405957     1      23.21427_S10    15      D2A
## 24.21428_S17    15.D2A.bulk 52000236     1      24.21428_S17    15      D2A
##
##          biorep exp separation time.donor
## 6.19190_S20      7   2      bulk      -2.D2A
## 7.19191_S1       7   2      bulk      0.D2A
## 9.19193_S6       7   2      bulk      3.D2A
## 11.19195_S12     8   2      bulk      -2.D2A
## 12.19196_S15     8   2      bulk      0.D2A
## 14.19198_S21     8   2      bulk      3.D2A
## 1.19200_S5        5   2      bulk      -2.D2A
## 2.19201_S8        5   2      bulk      0.D2A
## 4.19203_S14       5   2      bulk      3.D2A
## 20.S2.KD5NT_S16    5   2      bulk      -2.D1G
## 21.S7.KD6NT_S19    6   2      bulk      -2.D1G
## 22.S12.KD3NT_S22   8   2      bulk      -2.D1G
## 1.KD4NT_S1         P4  1      bulk      -2.D1G
## 10.P6D3_S10        P6  1      bulk      3.D1G
## 12.KD3NT_S12        P8  1      bulk      -2.D1G
## 13.P8D0_S13        P8  1      bulk      0.D1G
## 15.P8D3_S15        P8  1      bulk      3.D1G
## 2.KD5NT_S2         P5  1      bulk      -2.D1G
## 3.P5D0_S3         P5  1      bulk      0.D1G
## 5.P5D3_S5         P5  1      bulk      3.D1G
## 7.KD6NT_S7         P6  1      bulk      -2.D1G
## 8.P6D0_S8         P6  1      bulk      0.D1G
## 19.21423_S40      P7  3      bulk      15.D1G
## 20.21424_S47      P8_2 3      bulk      15.D1G
## 21.21425_S54      P8_1 3      bulk      15.D1G
## 22.21426_S3        P7  3      bulk      15.D2A
## 23.21427_S10      P6  3      bulk      15.D2A
## 24.21428_S17      P7  3      bulk      15.D2A

strict_filt_genes = short_ob[filterByExpr(short_ob),, keep.lib.sizes=FALSE]$genes
nrow(strict_filt_genes)

## [1] 20294

```

Normalise

```

filt_series = calcNormFactors(filt_series,method = "TMM")
filt_series$samples$norm.factors

```

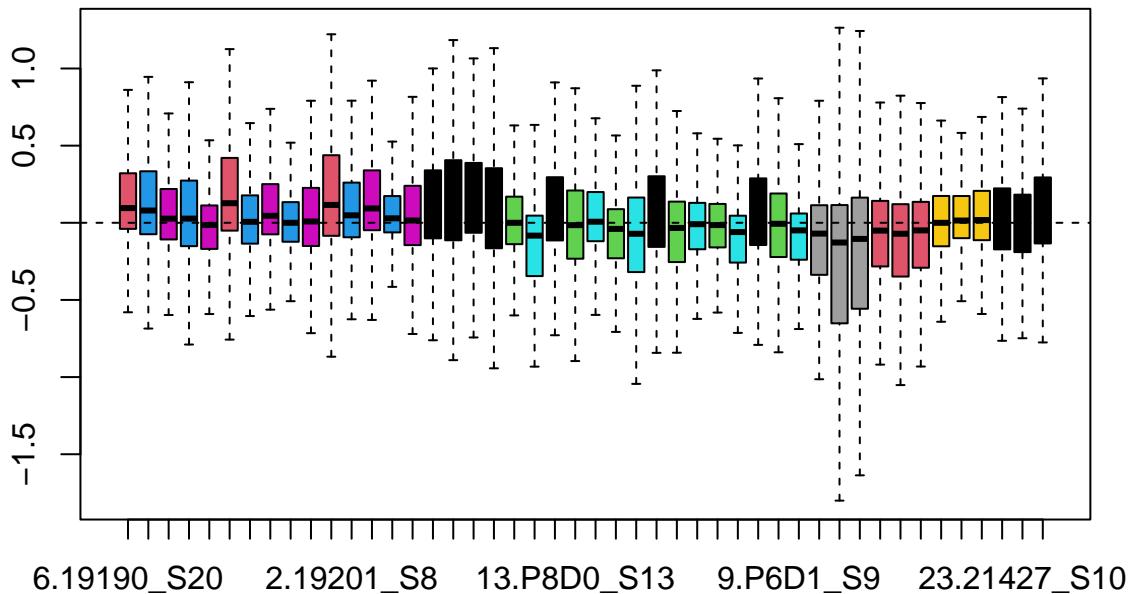
```

## [1] 1.2453947 1.2143844 1.1251937 1.1279176 0.9458378 1.3221153 1.0006515
## [8] 1.1403659 0.9928395 1.0445354 1.3664816 1.1431881 1.2621641 1.0546869
## [15] 1.0419248 1.2097142 1.3116360 1.2920292 1.1244045 0.9951992 0.7803828
## [22] 1.1123288 0.9314547 1.0344064 0.8685081 0.8944916 1.1077081 0.8561931
## [29] 0.9462953 0.9419874 0.8237669 1.1048697 0.9583928 0.8541696 0.7900846
## [36] 0.6108555 0.6834306 0.8374520 0.7701180 0.8412588 0.9839876 1.0048101
## [43] 1.0123225 0.9974551 0.9286849 1.0430259

plotRLE(cpm(filt_series$counts), outline=FALSE,
        col=filt_series$samples$group, main="After TMM normalisation (no libsize)")

```

After TMM normalisation (no libsize)



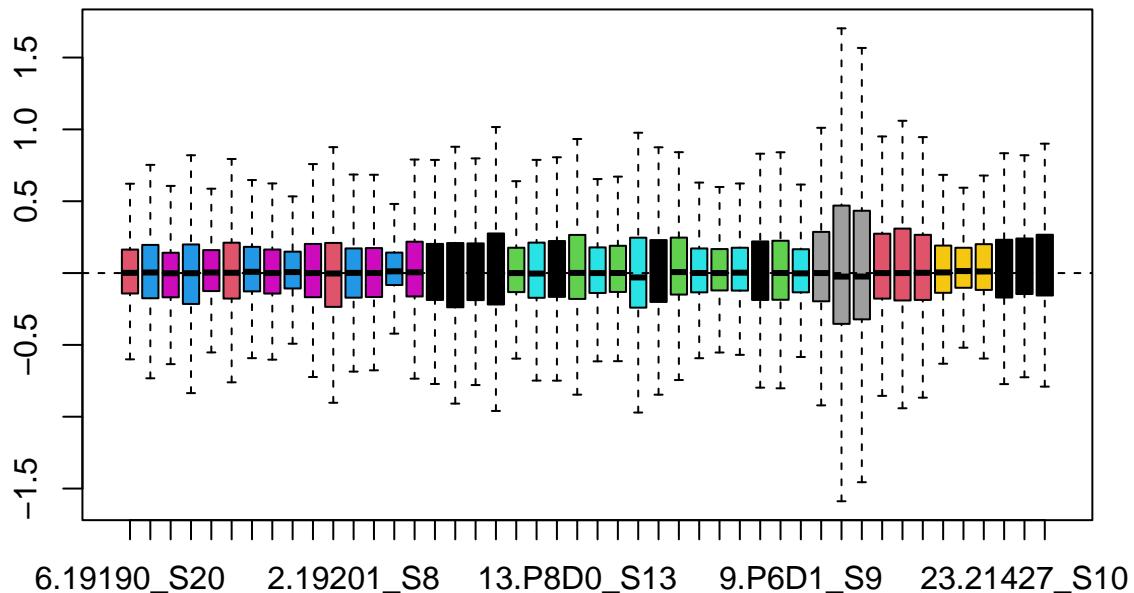
Library size is approximately the same after CPM normalisation, but it still needs to be adjusted directly for library size:

```

plotRLE(cpm(filt_series, normalized.lib.sizes = TRUE), outline=FALSE,
        col=filt_series$samples$group, main="After TMM normalisation (with libsize)")

```

After TMM normalisation (with libsize)



Annotate Gene Lists

```

library(biomaRt)
mart <- biomaRt::useMart(biomart = "ensembl",
  dataset = "hsapiens_gene_ensembl",
  host = "https://jan2019.archive.ensembl.org")
annot = getBM(c("external_gene_name", "description", "ensembl_gene_id"),
  filters = "ensembl_gene_id",
  values = filt_series$genes$Geneid,
  mart = mart)

head(annot, n=2); dim(annot)

##   external_gene_name                               description
## 1           TSPAN6  tetraspanin 6 [Source:HGNC Symbol;Acc:HGNC:11858]
## 2           TNMD    tenomodulin [Source:HGNC Symbol;Acc:HGNC:17757]
##   ensembl_gene_id
## 1 ENSG000000000003
## 2 ENSG000000000005

## [1] 21174      3

```

```

#Tidying up the annot table
annot$description = gsub("\\[Source:.+\\]", "", annot$description)
colnames(annot)[1] = "gene_name"

#Add gene names to filt_series
filt_series$genes = merge(filt_series$genes, annot, by.x= "Geneid",
                           by.y = "ensembl_gene_id", sort=FALSE)
#head(filt_series$genes)
#make sure length is numeric
filt_series$genes$Length = as.numeric(filt_series$genes$Length)

```

Calculate FPKM

```

rpkm = as.data.frame(rpkm(filt_series, normalized.lib.sizes = T))
rpkm$gene = rownames(rpkm)
colnames(rpkm) = paste("day",filt_series$samples$group, "_rep",filt_series$samples$biorep, sep="")
head(rpkm)

```

	day-2.D2A.bulk_rep7	day0.D2A.bulk_rep7	day1.D2A.bulk_rep7
## ENSG000000000003	4.429771	8.156323470	4.38952419
## ENSG000000000005	0.000000	0.009483402	0.04928463
## ENSG00000000419	44.399215	35.471356378	41.58979026
## ENSG00000000457	1.238538	2.004196996	1.73642745
## ENSG00000000460	1.180203	0.616667448	0.79787079
## ENSG00000000938	0.037342	0.039555121	0.06281165
##	day3.D2A.bulk_rep7	day9.D2A.bulk_rep7	day-2.D2A.bulk_rep8
## ENSG000000000003	10.6804534	6.3980668	6.801005854
## ENSG000000000005	0.0597703	1.3456444	0.000000000
## ENSG00000000419	43.5042255	52.6514494	45.991978584
## ENSG00000000457	2.1702937	1.8297088	1.295959535
## ENSG00000000460	0.6343311	0.5946063	2.466087279
## ENSG00000000938	0.1200338	0.2892193	0.004942582
##	day0.D2A.bulk_rep8	day1.D2A.bulk_rep8	day3.D2A.bulk_rep8
## ENSG000000000003	5.34283930	8.97825018	6.20967066
## ENSG000000000005	0.00000000	0.01544879	0.03209898
## ENSG00000000419	39.33891270	49.48751378	50.78492239
## ENSG00000000457	1.57592908	1.85270325	1.76943419
## ENSG00000000460	0.52619404	0.65859945	0.48500820
## ENSG00000000938	0.02557923	0.03579814	0.20826453
##	day9.D2A.bulk_rep8	day-2.D2A.bulk_rep5	day0.D2A.bulk_rep5
## ENSG000000000003	10.8402147	9.06770306	5.32941019
## ENSG000000000005	2.4098978	0.00000000	0.08398167
## ENSG00000000419	42.9456144	33.13432809	39.93582431
## ENSG00000000457	2.0765218	1.48149383	1.92021291
## ENSG00000000460	0.6567233	1.30065327	0.54949797
## ENSG00000000938	0.2264708	0.01705362	0.01946035
##	day1.D2A.bulk_rep5	day3.D2A.bulk_rep5	day9.D2A.bulk_rep5
## ENSG000000000003	5.84949758	6.0114837	9.2254928
## ENSG000000000005	0.60139195	0.2409799	4.4756616
## ENSG00000000419	41.36207525	44.8663886	41.5605299
## ENSG00000000457	1.75979885	1.9199313	2.2089933

```

## ENSG00000000460      0.75463451      0.4644325      0.6545097
## ENSG00000000938      0.03693757      0.1435891      0.6182750
##          day-2.D1G.bulk_rep5 day-2.D1G.bulk_rep6 day-2.D1G.bulk_rep8
## ENSG00000000003      2.91981053      7.24007579     6.04191622
## ENSG00000000005      0.01819565      0.03092287     0.00000000
## ENSG000000000419     53.71154817     39.44174826    45.42236856
## ENSG000000000457     1.02998553      1.53342908    1.72161788
## ENSG000000000460     3.09922167      2.03999184    1.65032753
## ENSG000000000938     0.00000000      0.00000000    0.00566926
##          day-2.D1G.bulk_repP4 day3.D1G.bulk_repP6 day9.D1G.bulk_repP6
## ENSG00000000003      2.721130      3.04182092    2.3619870
## ENSG00000000005      0.000000      0.03330655    0.3159819
## ENSG000000000419     97.377053      55.34687707   61.8625718
## ENSG000000000457     1.638974      2.02064095    1.9257976
## ENSG000000000460     1.159884      0.46371294    0.3505026
## ENSG000000000938     0.000000      0.10033191    0.3091501
##          day-2.D1G.bulk_repP8 day0.D1G.bulk_repP8 day1.D1G.bulk_repP8
## ENSG00000000003      2.007633      2.37629595    2.47737266
## ENSG00000000005      0.000000      0.00000000    0.00000000
## ENSG000000000419     59.451797      42.01958949   48.28312360
## ENSG000000000457     1.452945      1.24041816    1.42477241
## ENSG000000000460     1.110480      0.27099166    0.32147046
## ENSG000000000938     0.000000      0.08574259    0.01490318
##          day3.D1G.bulk_repP8 day9.D1G.bulk_repP8 day-2.D1G.bulk_repP5
## ENSG00000000003      2.40701054     1.4325053     2.17149640
## ENSG00000000005      0.01942692     0.00000000    0.00000000
## ENSG000000000419     55.26450018     30.4398764    68.43831575
## ENSG000000000457     1.99078950     1.9511371    1.14632438
## ENSG000000000460     0.25160256     0.2157656    2.96071334
## ENSG000000000938     0.33312081     0.3322643    0.01986473
##          day0.D1G.bulk_repP5 day1.D1G.bulk_repP5 day3.D1G.bulk_repP5
## ENSG00000000003      2.05304718     2.05323845   2.85095385
## ENSG00000000005      0.00000000     0.07376914   0.11659502
## ENSG000000000419     54.29605373     62.97810749  57.48804753
## ENSG000000000457     1.56093866     1.77664156   1.93550926
## ENSG000000000460     0.46373842     0.40604554   0.46402550
## ENSG000000000938     0.00937088     0.03418777   0.08105267
##          day9.D1G.bulk_repP5 day-2.D1G.bulk_repP6 day0.D1G.bulk_repP6
## ENSG00000000003      3.1988551      2.11611139   3.65396967
## ENSG00000000005      0.5437334      0.01486434   0.00000000
## ENSG000000000419     57.4549878     71.05122695  43.42671695
## ENSG000000000457     2.1379258      1.45307098   1.94983340
## ENSG000000000460     0.5134805      1.34958572   0.38134185
## ENSG000000000938     0.3779840      0.01377754   0.09247046
##          day1.D1G.bulk_repP6 day15.D1G.floating_repP6
## ENSG00000000003      2.11199568     14.2865244
## ENSG00000000005      0.03562279     0.6167761
## ENSG000000000419     75.29026376     36.1991881
## ENSG000000000457     1.81719641     1.9090162
## ENSG000000000460     0.34121346     0.8320844
## ENSG000000000938     0.01650912     3.1995683
##          day15.D1G.floating_repP8_1 day15.D1G.floating_repP8_2
## ENSG00000000003      19.9670327     12.9109419
## ENSG00000000005      0.7768957      0.3267391

```

```

## ENSG00000000419          28.8368126      25.0081755
## ENSG00000000457          1.3601721      1.4215021
## ENSG00000000460          0.9401136      0.6962417
## ENSG00000000938          4.6260542      3.4460243
##                  day15.D2A.floating_repP6_1 day15.D2A.floating_repP6_2
## ENSG00000000003          13.8343917      16.5892979
## ENSG00000000005          3.0427612      2.8011821
## ENSG00000000419          46.1827349      43.6749895
## ENSG00000000457          1.9288800      2.0130346
## ENSG00000000460          0.6710995      0.6981899
## ENSG00000000938          3.6745701      4.3981194
##                  day15.D2A.floating_repP7 day15.D1G.bulk_repP7
## ENSG00000000003          17.1528743      9.0381035
## ENSG00000000005          1.9451814      0.9721878
## ENSG00000000419          41.6803305      38.4294219
## ENSG00000000457          1.9813518      1.9312377
## ENSG00000000460          0.7506485      0.7282125
## ENSG00000000938          1.8029603      0.5715975
##                  day15.D1G.bulk_repP8_2 day15.D1G.bulk_repP8_1
## ENSG00000000003          8.8393788      9.5489457
## ENSG00000000005          0.6045077      0.2642974
## ENSG00000000419          35.1662323      33.6787557
## ENSG00000000457          1.5891691      1.6755817
## ENSG00000000460          0.6000864      0.4106589
## ENSG00000000938          0.6481190      0.6652296
##                  day15.D2A.bulk_repP7 day15.D2A.bulk_repP6 day15.D2A.bulk_repP7
## ENSG00000000003          7.6205880      8.960145      8.4071871
## ENSG00000000005          1.8124435      2.368496      1.6611412
## ENSG00000000419          37.5163734      36.592366      34.4896434
## ENSG00000000457          1.7366544      1.826420      1.3572695
## ENSG00000000460          0.4360015      0.463495      0.4265672
## ENSG00000000938          1.5888471      2.374268      1.7839139
##                         NA
## ENSG00000000003 ENSG00000000003
## ENSG00000000005 ENSG00000000005
## ENSG00000000419 ENSG00000000419
## ENSG00000000457 ENSG00000000457
## ENSG00000000460 ENSG00000000460
## ENSG00000000938 ENSG00000000938

rpkm = rpkm[order(colnames(rpkm))]

format_rpkm = merge(filt_series$genes, rpkm,
                     by.x="Geneid", by.y = 'row.names', sort=FALSE)
head(format_rpkm)

##           Geneid Length gene_name
## 1 ENSG00000000003    4535   TSPAN6
## 2 ENSG00000000005    1610    TNMD
## 3 ENSG00000000419    1207    DPM1
## 4 ENSG00000000457    6883   SCYL3
## 5 ENSG00000000460    5967 C1orf112
## 6 ENSG00000000938    3474    FGR
##                                         description

```

```

## 1                               tetraspanin 6
## 2                               tenomodulin
## 3 dolichyl-phosphate mannosyltransferase subunit 1, catalytic
## 4                               SCY1 like pseudokinase 3
## 5                               chromosome 1 open reading frame 112
## 6 FGR proto-oncogene, Src family tyrosine kinase
##   day-2.D1G.bulk_rep5 day-2.D1G.bulk_rep6 day-2.D1G.bulk_rep8
## 1      2.91981053      7.24007579      6.04191622
## 2      0.01819565      0.03092287      0.00000000
## 3      53.71154817     39.44174826     45.42236856
## 4      1.02998553     1.53342908     1.72161788
## 5      3.09922167     2.03999184     1.65032753
## 6      0.00000000     0.00000000     0.00566926
##   day-2.D1G.bulk_repP4 day-2.D1G.bulk_repP5 day-2.D1G.bulk_repP6
## 1      2.721130      2.17149640      2.11611139
## 2      0.000000      0.00000000      0.01486434
## 3      97.377053     68.43831575     71.05122695
## 4      1.638974      1.14632438     1.45307098
## 5      1.159884      2.96071334     1.34958572
## 6      0.000000      0.01986473     0.01377754
##   day-2.D1G.bulk_repP8 day-2.D2A.bulk_rep5 day-2.D2A.bulk_rep7
## 1      2.007633      9.06770306      4.429771
## 2      0.000000      0.00000000      0.000000
## 3      59.451797     33.13432809     44.399215
## 4      1.452945      1.48149383     1.238538
## 5      1.110480      1.30065327     1.180203
## 6      0.000000      0.01705362     0.037342
##   day-2.D2A.bulk_rep8 day0.D1G.bulk_repP5 day0.D1G.bulk_repP6
## 1      6.801005854    2.05304718      3.65396967
## 2      0.0000000000    0.00000000      0.00000000
## 3      45.991978584    54.29605373     43.42671695
## 4      1.295959535    1.56093866     1.94983340
## 5      2.466087279    0.46373842     0.38134185
## 6      0.004942582    0.00937088     0.09247046
##   day0.D1G.bulk_repP8 day0.D2A.bulk_rep5 day0.D2A.bulk_rep7 day0.D2A.bulk_rep8
## 1      2.37629595     5.32941019     8.156323470    5.34283930
## 2      0.0000000000     0.08398167     0.009483402    0.00000000
## 3      42.01958949     39.93582431     35.471356378    39.33891270
## 4      1.24041816     1.92021291     2.004196996    1.57592908
## 5      0.27099166     0.54949797     0.616667448    0.52619404
## 6      0.08574259     0.01946035     0.039555121    0.02557923
##   day1.D1G.bulk_repP5 day1.D1G.bulk_repP6 day1.D1G.bulk_repP8
## 1      2.05323845     2.11199568     2.47737266
## 2      0.07376914     0.03562279     0.00000000
## 3      62.97810749     75.29026376     48.28312360
## 4      1.77664156     1.81719641     1.42477241
## 5      0.40604554     0.34121346     0.32147046
## 6      0.03418777     0.01650912     0.01490318
##   day1.D2A.bulk_rep5 day1.D2A.bulk_rep7 day1.D2A.bulk_rep8 day15.D1G.bulk_repP7
## 1      5.84949758     4.38952419     8.97825018    9.0381035
## 2      0.60139195     0.04928463     0.01544879    0.9721878
## 3      41.36207525     41.58979026     49.48751378    38.4294219
## 4      1.75979885     1.73642745     1.85270325    1.9312377
## 5      0.75463451     0.79787079     0.65859945    0.7282125

```

```

## 6      0.03693757      0.06281165      0.03579814      0.5715975
## day15.D1G.bulk_repP8_1 day15.D1G.bulk_repP8_2 day15.D1G.floating_repP6
## 1      9.5489457      8.8393788      14.2865244
## 2      0.2642974      0.6045077      0.6167761
## 3      33.6787557      35.1662323      36.1991881
## 4      1.6755817      1.5891691      1.9090162
## 5      0.4106589      0.6000864      0.8320844
## 6      0.6652296      0.6481190      3.1995683
## day15.D1G.floating_repP8_1 day15.D1G.floating_repP8_2 day15.D2A.bulk_repP6
## 1      19.9670327      12.9109419      8.960145
## 2      0.7768957      0.3267391      2.368496
## 3      28.8368126      25.0081755      36.592366
## 4      1.3601721      1.4215021      1.826420
## 5      0.9401136      0.6962417      0.463495
## 6      4.6260542      3.4460243      2.374268
## day15.D2A.bulk_repP7 day15.D2A.bulk_repP7.1 day15.D2A.floating_repP6_1
## 1      7.6205880      8.4071871      13.8343917
## 2      1.8124435      1.6611412      3.0427612
## 3      37.5163734      34.4896434      46.1827349
## 4      1.7366544      1.3572695      1.9288800
## 5      0.4360015      0.4265672      0.6710995
## 6      1.5888471      1.7839139      3.6745701
## day15.D2A.floating_repP6_2 day15.D2A.floating_repP7 day3.D1G.bulk_repP5
## 1      16.5892979      17.1528743      2.85095385
## 2      2.8011821      1.9451814      0.11659502
## 3      43.6749895      41.6803305      57.48804753
## 4      2.0130346      1.9813518      1.93550926
## 5      0.6981899      0.7506485      0.46402550
## 6      4.3981194      1.8029603      0.08105267
## day3.D1G.bulk_repP6 day3.D1G.bulk_repP8 day3.D2A.bulk_rep5 day3.D2A.bulk_rep7
## 1      3.04182092      2.40701054      6.0114837      10.6804534
## 2      0.03330655      0.01942692      0.2409799      0.0597703
## 3      55.34687707      55.26450018      44.8663886      43.5042255
## 4      2.02064095      1.99078950      1.9199313      2.1702937
## 5      0.46371294      0.25160256      0.4644325      0.6343311
## 6      0.10033191      0.33312081      0.1435891      0.1200338
## day3.D2A.bulk_rep8 day9.D1G.bulk_repP5 day9.D1G.bulk_repP6
## 1      6.20967066      3.1988551      2.3619870
## 2      0.03209898      0.5437334      0.3159819
## 3      50.78492239      57.4549878      61.8625718
## 4      1.76943419      2.1379258      1.9257976
## 5      0.48500820      0.5134805      0.3505026
## 6      0.20826453      0.3779840      0.3091501
## day9.D1G.bulk_repP8 day9.D2A.bulk_rep5 day9.D2A.bulk_rep7 day9.D2A.bulk_rep8
## 1      1.4325053      9.2254928      6.3980668      10.8402147
## 2      0.0000000      4.4756616      1.3456444      2.4098978
## 3      30.4398764      41.5605299      52.6514494      42.9456144
## 4      1.9511371      2.2089933      1.8297088      2.0765218
## 5      0.2157656      0.6545097      0.5946063      0.6567233
## 6      0.3322643      0.6182750      0.2892193      0.2264708
## NA
## 1 ENSG00000000003
## 2 ENSG00000000005
## 3 ENSG00000000419

```

```

## 4 ENSG00000000457
## 5 ENSG00000000460
## 6 ENSG00000000938

write.table(format_rpkm, sep='\t', row.names = FALSE, quote = F,
            file="../03limma/adipogenesis_rpkm_tmm.tab")

filt_series$samples$time.donor = filt_series$samples$group
filt_series$samples$group = factor(paste("day",filt_series$samples$group, sep=""))
group_rpkm = data.frame(rpkmByGroup(filt_series,normalize.lib.sizes=TRUE))

colSums(group_rpkm)

##      day.2.D1G.bulk    day.2.D2A.bulk    day0.D1G.bulk    day0.D2A.bulk
##      240021.2          217572.3        368588.3        259177.3
##      day1.D1G.bulk     day1.D2A.bulk    day15.D1G.bulk day15.D1G.floating
##      432303.4          263522.6        337702.1        624718.6
##      day15.D2A.bulk   day15.D2A.floating day3.D1G.bulk   day3.D2A.bulk
##      346766.4          464595.3        475628.3        327786.6
##      day9.D1G.bulk    day9.D2A.bulk
##      590691.7          339424.4

format_grpkm = merge(filt_series$genes, group_rpkm,
                      by.x="Geneid", by.y = 'row.names', sort=FALSE)
head(format_grpkm)

##           Geneid Length gene_name
## 1 ENSG00000000003    4535   TSPAN6
## 2 ENSG00000000005    1610    TNMD
## 3 ENSG00000000419    1207    DPM1
## 4 ENSG00000000457    6883    SCYL3
## 5 ENSG00000000460    5967  C1orf112
## 6 ENSG00000000938    3474     FGR
##                                         description day.2.D1G.bulk
## 1                                     tetraspanin 6  3.601999275
## 2                                     tenomodulin 0.007936845
## 3 dolichyl-phosphate mannosyltransferase subunit 1, catalytic 62.130005013
## 4                                     SCY1 like pseudokinase 3  1.426204636
## 5                                     chromosome 1 open reading frame 112  1.906747475
## 6 FGR proto-oncogene, Src family tyrosine kinase 0.005503088
##      day.2.D2A.bulk day0.D1G.bulk day0.D2A.bulk day1.D1G.bulk day1.D2A.bulk
## 1      6.76071437  2.69543757  6.28332533  2.21416364  6.40180026
## 2      0.00000000  0.00000000  0.02464032  0.03748234  0.25958703
## 3     41.18788108 46.57260499 38.24198388 62.18075805 44.14029328
## 4      1.33753816  1.58269525  1.83452213  1.67277113  1.78269269
## 5      1.65219243  0.36937923  0.56555920  0.35661924  0.73777550
## 6      0.01975517  0.06610962  0.03066300  0.02233808  0.04456323
##      day15.D1G.bulk day15.D1G.floating day15.D2A.bulk day15.D2A.floating
## 1      9.1422795    15.7180139    8.3310883    15.8591145
## 2      0.5985436    0.5635616    1.9493616    2.5943861
## 3     35.7545717    30.0118693    36.1959969    43.8454420
## 4     1.7310483    1.5638762    1.6389382    1.9744310

```

```

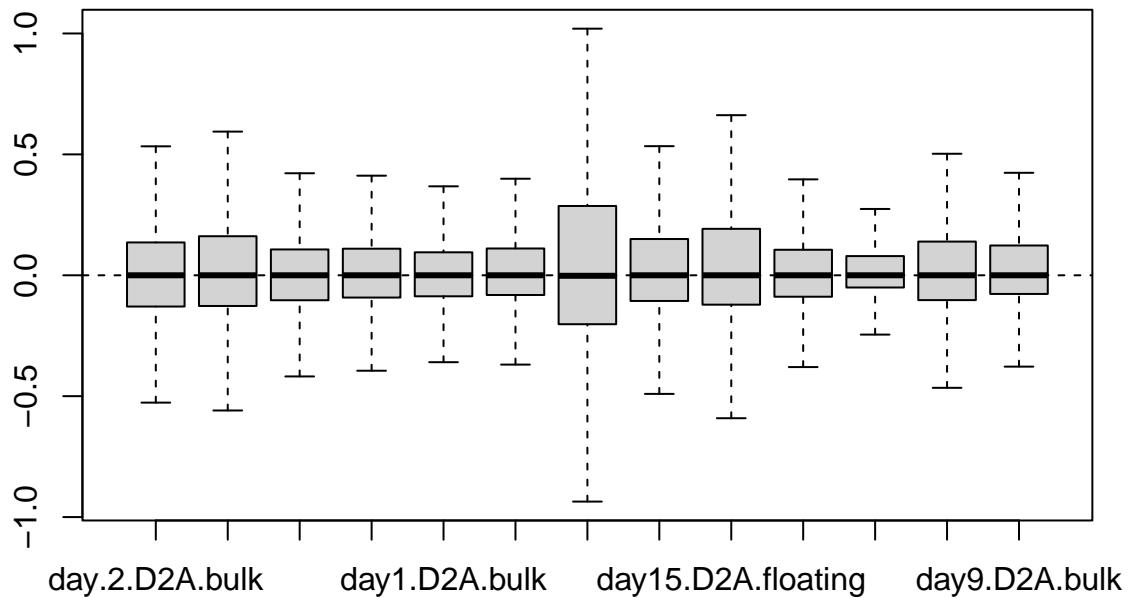
## 5      0.5774263      0.8213003      0.4421079      0.7067799
## 6      0.6295882      3.7540597      1.9196028      3.2901662
## day3.D1G.bulk day3.D2A.bulk day9.D1G.bulk day9.D2A.bulk
## 1      2.76755264     7.6279970     2.3380573     8.8221868
## 2      0.06034418     0.1242015     0.3061671     2.7625195
## 3      56.03395361    46.3859709    49.9509805    45.7146885
## 4      1.98222151    1.9524608    2.0051649    2.0390762
## 5      0.39508980    0.5258706    0.3640591    0.6355628
## 6      0.16554146    0.1578544    0.3397660    0.3845655

rowMeans(format_grpkm[format_grpkm$gene_name == "PPARG",c(6:ncol(format_grpkm))])#expr matches the rep1

## 5456
## 46.428

plotRLE(data.matrix(format_grpkm[6:ncol(format_grpkm)]), outline=FALSE)

```



```

write.table(format_grpkm, sep='\t', row.names = FALSE, quote=F,
           file="../03limma/adipogenesis_rpkm_tmm_means.tab")

```

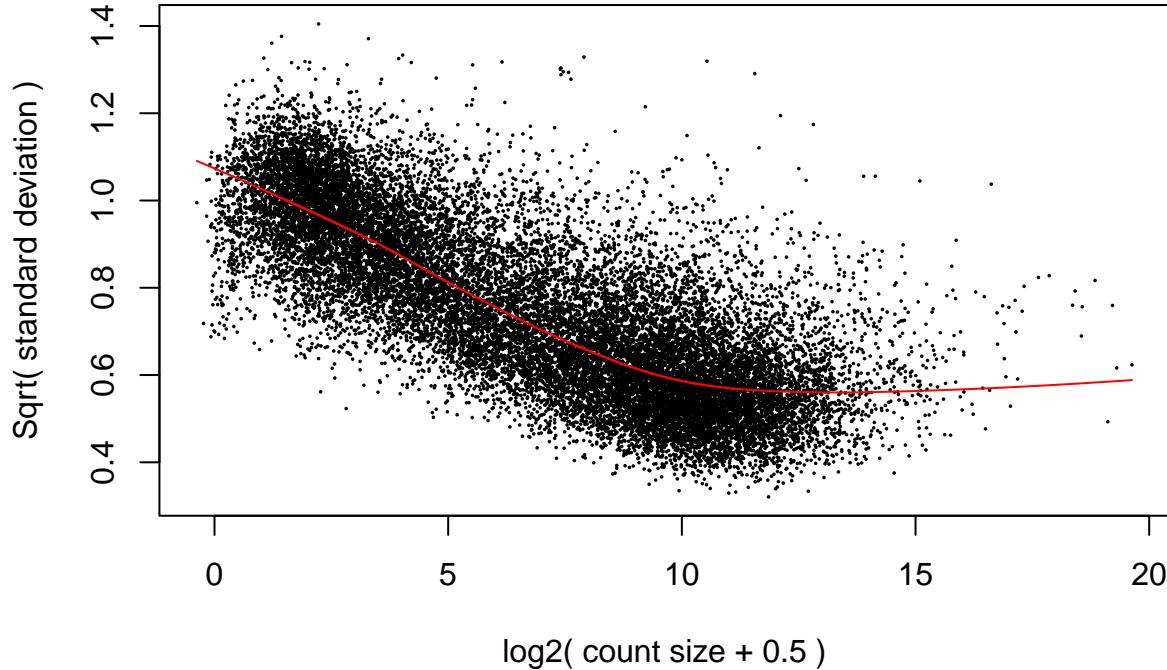
Limma Log Normalisation

```

filt_series$samples$group = filt_series$samples$time.donor
norm_series = voom(filt_series, plot = TRUE)

```

voom: Mean–variance trend



```

head(norm_series$design, n=2) #groups are time.donor

```

```

##          (Intercept) group-2.D2A.bulk group0.D1G.bulk group0.D2A.bulk
## 6.19190_S20           1             1             0             0
## 7.19191_S1           1             0             0             1
##          group1.D1G.bulk group1.D2A.bulk group15.D1G.bulk
## 6.19190_S20           0             0             0
## 7.19191_S1           0             0             0
##          group15.D1G.floating group15.D2A.bulk group15.D2A.floating
## 6.19190_S20           0             0             0
## 7.19191_S1           0             0             0
##          group3.D1G.bulk group3.D2A.bulk group9.D1G.bulk group9.D2A.bulk
## 6.19190_S20           0             0             0             0
## 7.19191_S1           0             0             0             0

```

```

summary(norm_series$E[,1])#log normalisation min -7, max 15

```

```

##      Min. 1st Qu. Median    Mean 3rd Qu.    Max.
## -6.754 -1.468  1.972   1.594  4.635 14.871

```

Design

```
#1 check the default design
colnames(norm_series$design)

## [1] "(Intercept)"           "group-2.D2A.bulk"      "group0.D1G.bulk"
## [4] "group0.D2A.bulk"        "group1.D1G.bulk"      "group1.D2A.bulk"
## [7] "group15.D1G.bulk"       "group15.D1G.floating" "group15.D2A.bulk"
## [10] "group15.D2A.floating"   "group3.D1G.bulk"      "group3.D2A.bulk"
## [13] "group9.D1G.bulk"        "group9.D2A.bulk"

head(filt_series$samples, n=2)

##          group lib.size norm.factors sample_id time donor biorep exp
## 6.19190_S20 -2.D2A.bulk 43327502    1.245395 6.19190_S20 -2   D2A     7   2
## 7.19191_S1   0.D2A.bulk 53932903    1.214384 7.19191_S1   0   D2A     7   2
##               separation time.donor
## 6.19190_S20      bulk -2.D2A.bulk
## 7.19191_S1      bulk  0.D2A.bulk

pair_design = model.matrix(~0+filt_series$samples$group)
colnames(pair_design) = paste("day", gsub("-", ".", levels(filt_series$samples$group)), sep="")
rownames(pair_design) = filt_series$samples$sample_id
head(pair_design, n=2)

##          day.2.D1G.bulk day.2.D2A.bulk day0.D1G.bulk day0.D2A.bulk
## 6.19190_S20            0              1              0              0
## 7.19191_S1            0              0              0              1
##          day1.D1G.bulk day1.D2A.bulk day15.D1G.bulk day15.D1G.floating
## 6.19190_S20            0              0              0                  0
## 7.19191_S1            0              0              0                  0
##          day15.D2A.bulk day15.D2A.floating day3.D1G.bulk day3.D2A.bulk
## 6.19190_S20            0              0              0                  0
## 7.19191_S1            0              0              0                  0
##          day9.D1G.bulk day9.D2A.bulk
## 6.19190_S20            0              0
## 7.19191_S1            0              0
```

Making a large time based model

Using bulk day15 samples time_effect_donor1 + time_effect_donor2 was the pvalue used at 0.01 to select genes for dpgp

```
pair_fit = lmFit(norm_series, design=pair_design)

contrasts = makeContrasts(time_effect_donor1 = (day0.D1G.bulk - day2.D1G.bulk) + #D-2 -> D0 effect
                           (day1.D1G.bulk - day0.D1G.bulk) + #D0 -> D1 effect
                           (day3.D1G.bulk - day1.D1G.bulk) + #D1 -> D3 effect
                           (day9.D1G.bulk - day3.D1G.bulk) + #D3 -> D9 effect
                           (day15.D1G.bulk - day9.D1G.bulk),
```

```

time_effect_donor2 = (day0.D2A.bulk - day.2.D2A.bulk) + #D-2 -> D0 effect
                     (day1.D2A.bulk - day0.D2A.bulk) + #D0 -> D1 effect
                     (day3.D2A.bulk - day1.D2A.bulk) + #D1 -> D3 effect
                     (day9.D2A.bulk - day3.D2A.bulk) + #D3 -> D9 effect
                     (day15.D2A.bulk - day9.D2A.bulk), #D9 -> D15 effect
simple_time_effect = (day0.D1G.bulk - day.2.D1G.bulk) + #D-2 -> D0 effect
                     (day3.D1G.bulk - day0.D1G.bulk) + #D0 -> D3 effect
                     (day15.D1G.bulk - day3.D1G.bulk) + #D3 -> D15 effect
                     (day0.D2A.bulk - day.2.D2A.bulk) + #D-2 -> D0 effect
                     (day3.D2A.bulk - day0.D2A.bulk) + #D1 -> D3 effect
                     (day15.D2A.bulk - day3.D2A.bulk), #D9 -> D15 effect
#Donor 7 changes over time
d.2_to_d0.D1 = day0.D1G.bulk - day.2.D1G.bulk,
d0_to_d1.D1 = day1.D1G.bulk - day0.D1G.bulk,
d1_to_d3.D1 = day3.D1G.bulk - day1.D1G.bulk,
d3_to_d9.D1= day9.D1G.bulk - day3.D1G.bulk,
d9_to_d15.D1.bulk = day15.D1G.bulk - day9.D1G.bulk,
d9_to_d15.D1.floating = day15.D1G.floating - day9.D1G.bulk,
d15.bulk_to_floating.D1 = day15.D1G.floating - day15.D1G.bulk,
#Donor 13 changes over time
d.2_to_d0.D2 = day0.D2A.bulk - day.2.D2A.bulk,
d0_to_d1.D2 = day1.D2A.bulk - day0.D2A.bulk,
d1_to_d3.D2 = day3.D2A.bulk - day1.D2A.bulk,
d3_to_d9.D2 = day9.D2A.bulk - day3.D2A.bulk,
d9_to_d15.D2.bulk = day15.D2A.bulk - day9.D2A.bulk,
d9_to_d15.D2.floating = day15.D2A.floating - day9.D2A.bulk,
d15.bulk_to_floating.D2 = day15.D2A.floating - day15.D2A.bulk,
#And the additional timepoints
d0_to_d3.D1 = day3.D1G.bulk - day0.D1G.bulk,
d0_to_d3.D2 = day3.D2A.bulk - day0.D2A.bulk,
d3_to_d15.D1 = day15.D1G.bulk - day3.D1G.bulk,
d3_to_d15.D2 = day15.D2A.bulk - day3.D2A.bulk,
levels=pair_design)
special_fit = contrasts.fit(pair_fit,contrasts)
special_fit = eBayes(special_fit, robust = TRUE)

```

```
summary(decideTests(special_fit, method="separate", adjust.method = "BH", p.value = 0.01)) #Adjusted for
```

	time_effect_donor1	time_effect_donor2	simple_time_effect	d.2_to_d0.D1
## Down	4563	4558	5677	2489
## NotSig	11929	12246	9577	15507
## Up	4682	4370	5920	3178
	d0_to_d1.D1	d1_to_d3.D1	d3_to_d9.D1	d9_to_d15.D1.bulk
## Down	1437	250	319	3353
## NotSig	18900	20481	20466	14120
## Up	837	443	389	3701
	d9_to_d15.D1.floating	d15.bulk_to_floating.D1	d.2_to_d0.D2	d0_to_d1.D2
## Down		4357	3097	909
## NotSig		12264	14860	19492
## Up		4553	3217	773
	d1_to_d3.D2	d3_to_d9.D2	d9_to_d15.D2.bulk	d9_to_d15.D2.floating
## Down	619	178	3067	3181
## NotSig	19845	20687	15334	14969

```

## Up 710 309 2773 3024
##      d15.bulk_to_floating.D2 d0_to_d3.D1 d0_to_d3.D2 d3_to_d15.D1
## Down 131 2275 2295 3626
## NotSig 20871 17367 17103 13821
## Up 172 1532 1776 3727
##      d3_to_d15.D2
## Down 3120
## NotSig 14923
## Up 3131

summary(decideTests(special_fit ,method="global", adjust.method = "BH"), p.value= 0.01) #Unadjusted for

##      time_effect_donor1 time_effect_donor2 simple_time_effect d.2_to_d0.D1
## Down 5221 5396 6195 3493
## NotSig 10452 10500 8377 13338
## Up 5501 5278 6602 4343
##      d0_to_d1.D1 d1_to_d3.D1 d3_to_d9.D1 d9_to_d15.D1.bulk
## Down 2548 1033 1461 4262
## NotSig 16874 19076 18477 12007
## Up 1752 1065 1236 4905
##      d9_to_d15.D1.floating d15.bulk_to_floating.D1 d.2_to_d0.D2 d0_to_d1.D2
## Down 5132 4255 1885 2463
## NotSig 10613 12584 17453 16828
## Up 5429 4335 1836 1883
##      d1_to_d3.D2 d3_to_d9.D2 d9_to_d15.D2.bulk d9_to_d15.D2.floating
## Down 1474 803 4022 4019
## NotSig 18206 19266 13104 12956
## Up 1494 1105 4048 4199
##      d15.bulk_to_floating.D2 d0_to_d3.D1 d0_to_d3.D2 d3_to_d15.D1
## Down 996 3335 3330 4642
## NotSig 18822 15385 15140 11762
## Up 1356 2454 2704 4770
##      d3_to_d15.D2
## Down 4135
## NotSig 12685
## Up 4354

```

Interesting, the time effect is less significant in donor2. The time effect is similar rather than different between donors -> most genes respond similarly to the adipogenic cocktail, regardless of depot/donor. Removing those extra timepoints (Day1, Day9), leads to ~6k genes found both up and down

Number of DE genes

Adding an extra timepoint adds more genes to the DE list; 11,639 genes at $p < 0.01$ (basically 50% of all detected genes).

```

comb = topTable(special_fit, number = nrow(pair_fit$genes), coef = c("time_effect_donor1","time_effect_donor2"))
dim(comb[comb$adj.P.Val < 0.05,]) #More than half of all genes change; most will be the same

```

```

## [1] 14236    10

```

```

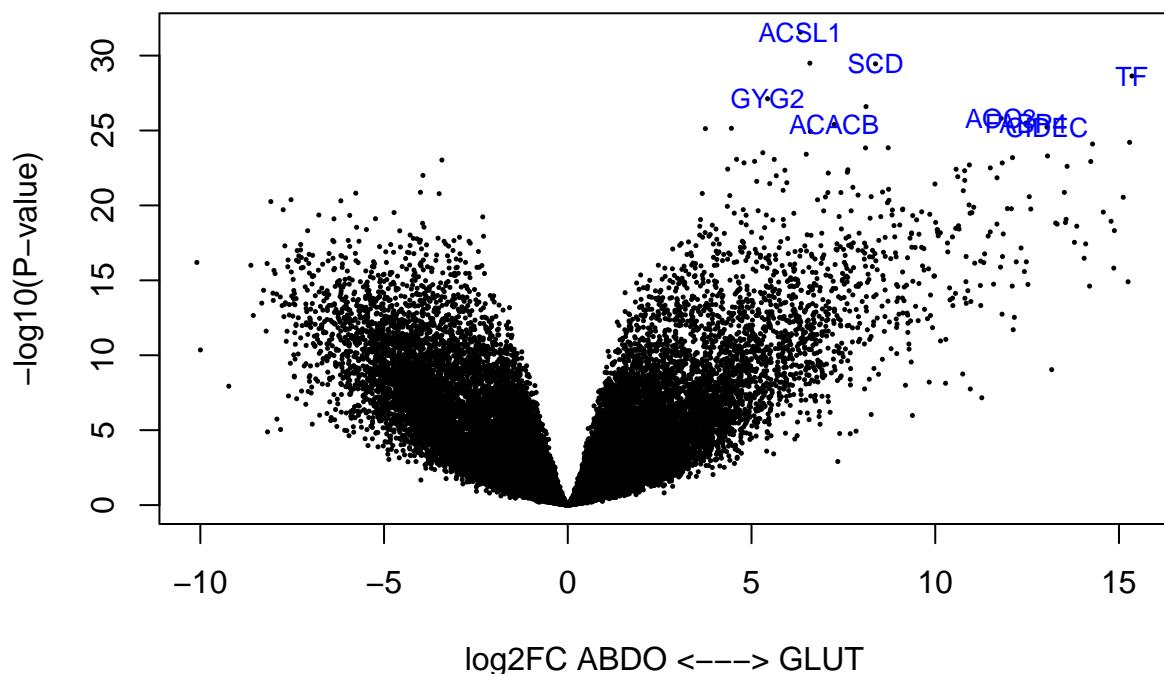
dim(comb[comb$adj.P.Val < 0.01,])

## [1] 11639    10

volcanoplot(special_fit, coef = c(1,2), highlight = 10, names = special_fit$genes$gene_name,
             main="Time effect (combined across donors)", xlab="log2FC ABDO <---> GLUT")

```

Time effect (combined across donors)

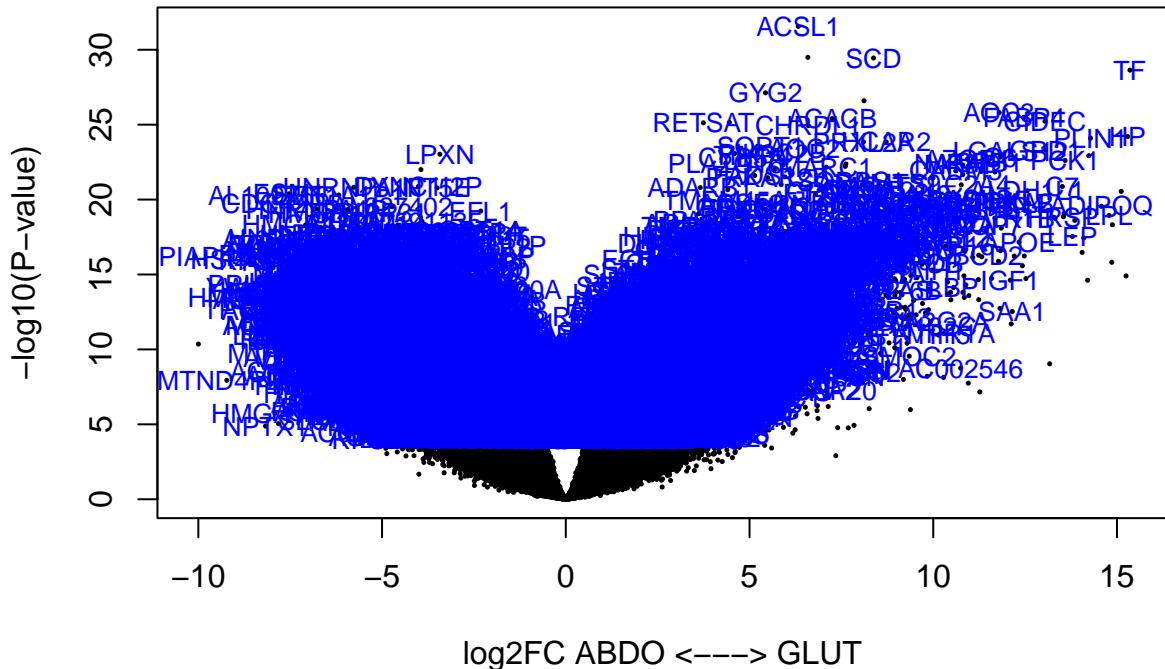


```

volcanoplot(special_fit, coef = c(1,2), highlight = 11639, names = special_fit$genes$gene_name,
             main="Time effect (combined across donors)", xlab="log2FC ABDO <---> GLUT")

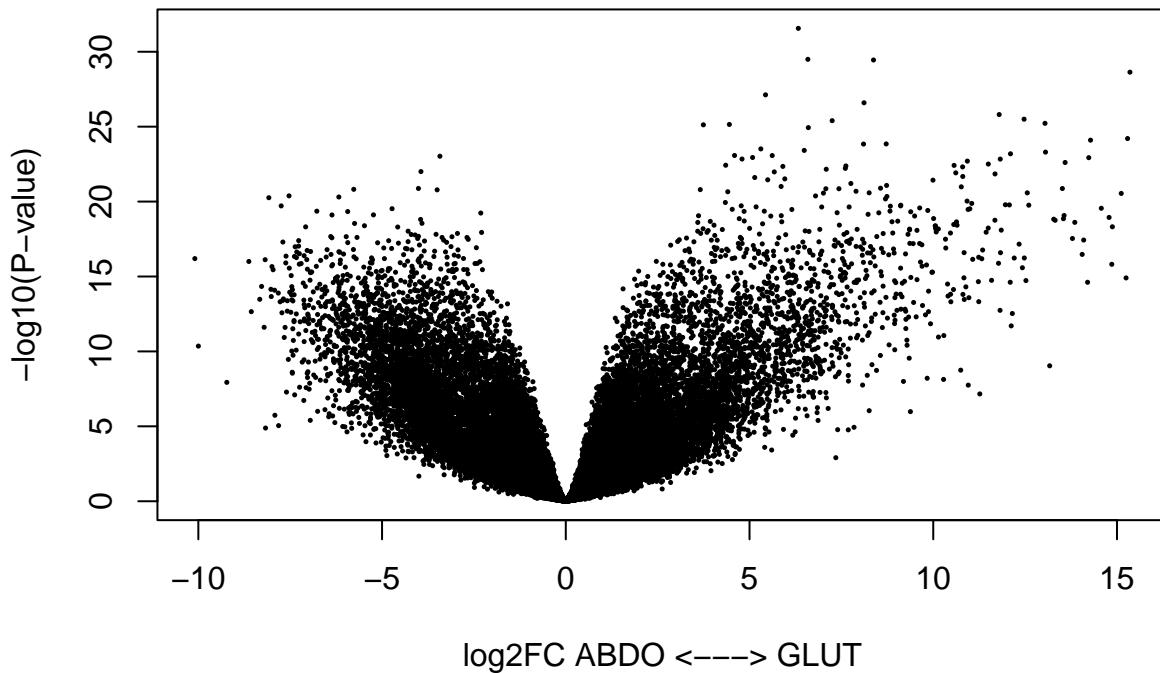
```

Time effect (combined across donors)



```
volcanoplot(special_fit, coef = c(1,2),
             main="Time effect (combined across donors)", xlab="log2FC ABDO <---> GLUT")
```

Time effect (combined across donors)



```
head(comb)
```

```
##           Geneid Length gene_name
## ENSG00000151726 ENSG00000151726    6284      ACSL1
## ENSG00000099194 ENSG00000099194    5362       SCD
## ENSG00000091513 ENSG00000091513   26199        TF
## ENSG00000056998 ENSG00000056998    3655      GYG2
## ENSG00000042445 ENSG00000042445    4005     RETSAT
## ENSG00000131471 ENSG00000131471    4524      AOC3
##                                     description
## ENSG00000151726 acyl-CoA synthetase long chain family member 1
## ENSG00000099194                      stearoyl-CoA desaturase
## ENSG00000091513                      transferrin
## ENSG00000056998                      glycogenin 2
## ENSG00000042445                      retinol saturase
## ENSG00000131471 amine oxidase, copper containing 3
## time_effect_donor1 time_effect_donor2 AveExpr      F
## ENSG00000151726          6.328527      6.587910 8.810891 1311.9493
## ENSG00000099194          8.374125      8.116002 11.807874  977.5091
## ENSG00000091513         15.353000      14.232266 3.105257  772.2546
## ENSG00000056998          5.435924      4.592658 4.980263  682.1097
## ENSG00000042445          3.744336      4.451633 7.100134  664.3586
## ENSG00000131471         11.794312      11.680727 6.379407  583.7300
##             P.Value      adj.P.Val
## ENSG00000151726 4.270636e-36 9.042645e-32
## ENSG00000099194 1.366785e-33 1.447015e-29
```

```

## ENSG00000091513 1.130334e-31 7.977894e-28
## ENSG00000056998 9.522686e-31 5.040834e-27
## ENSG00000042445 1.558533e-30 6.600076e-27
## ENSG00000131471 2.068864e-29 6.793301e-26

nuc = topTable(special_fit, number = nrow(pair_fit$genes), coef = "simple_time_effect")
dim(nuc[nuc$adj.P.Val < 0.05,]) #More than half of all genes change; most will be the same

## [1] 13727    10

dim(nuc[nuc$adj.P.Val < 0.01,])

## [1] 11597    10

nuc = nuc[nuc$Geneid %in% strict_filt_genes$Geneid,] #We can filter out ~450 genes with low expression
nrow(nuc)

## [1] 20294

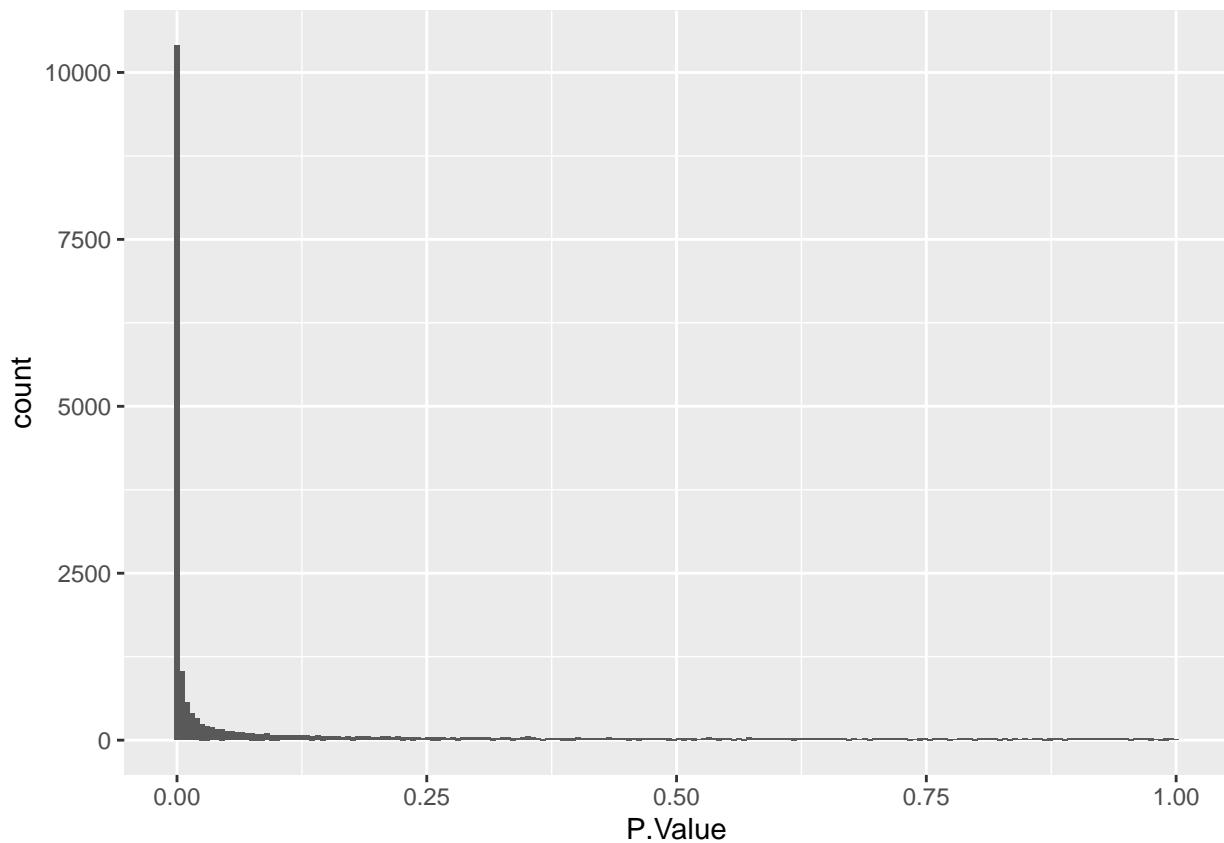
head(nuc)

##                                     Geneid Length gene_name
## ENSG00000151726 ENSG00000151726    6284    ACSL1
## ENSG00000099194 ENSG00000099194    5362      SCD
## ENSG00000042445 ENSG00000042445    4005    RETSAT
## ENSG00000056998 ENSG00000056998    3655     GYG2
## ENSG00000076555 ENSG00000076555   14505    ACACB
## ENSG00000101938 ENSG00000101938    3920    CHRDL1
##                                         description      logFC
## ENSG00000151726 acyl-CoA synthetase long chain family member 1 12.916437
## ENSG00000099194                               stearoyl-CoA desaturase 16.490127
## ENSG00000042445                               retinol saturase 8.195969
## ENSG00000056998                               glycogenin 2 10.028582
## ENSG00000076555 acetyl-CoA carboxylase beta 14.871196
## ENSG00000101938                               chordin like 1 14.207558
##          AveExpr      t      P.Value adj.P.Val      B
## ENSG00000151726 8.810891 50.67161 8.291877e-37 1.755722e-32 68.92770
## ENSG00000099194 11.807874 43.63744 2.897205e-34 3.067271e-30 64.45829
## ENSG00000042445  7.100134 36.32129 2.309287e-31 9.779370e-28 59.33478
## ENSG00000056998  4.980263 36.39842 2.133268e-31 9.779370e-28 58.59981
## ENSG00000076555  7.351638 33.29351 7.038353e-30 2.474282e-26 55.41491
## ENSG00000101938  5.954537 32.38451 1.960845e-29 4.613214e-26 54.48242

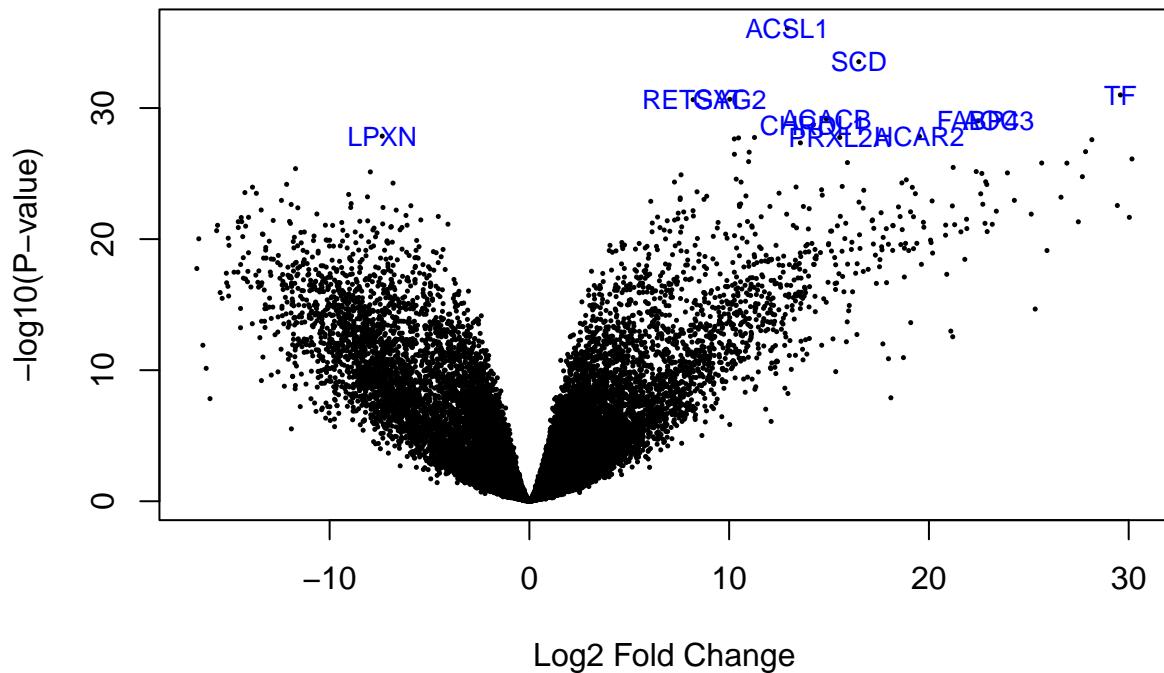
write.table(nuc, sep="\t", row.names=FALSE,
            file.path("../03limma/nucleolus_adipogenesis_DE.tab"))

ggplot(nuc) + geom_histogram(aes(x=P.Value), bins=200)

```

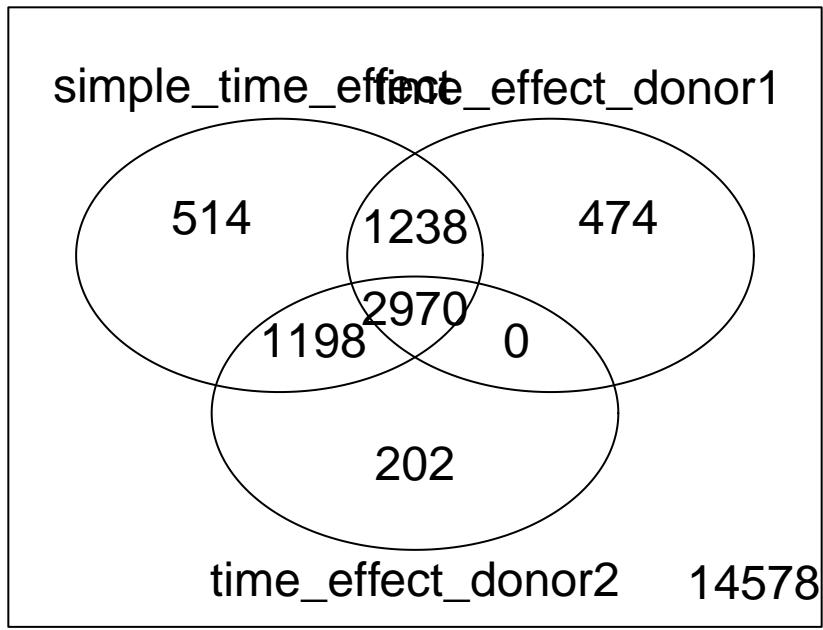


```
volcanoplot(special_fit, coef="simple_time_effect", highlight = 12, names = special_fit$genes$gene_name)
```

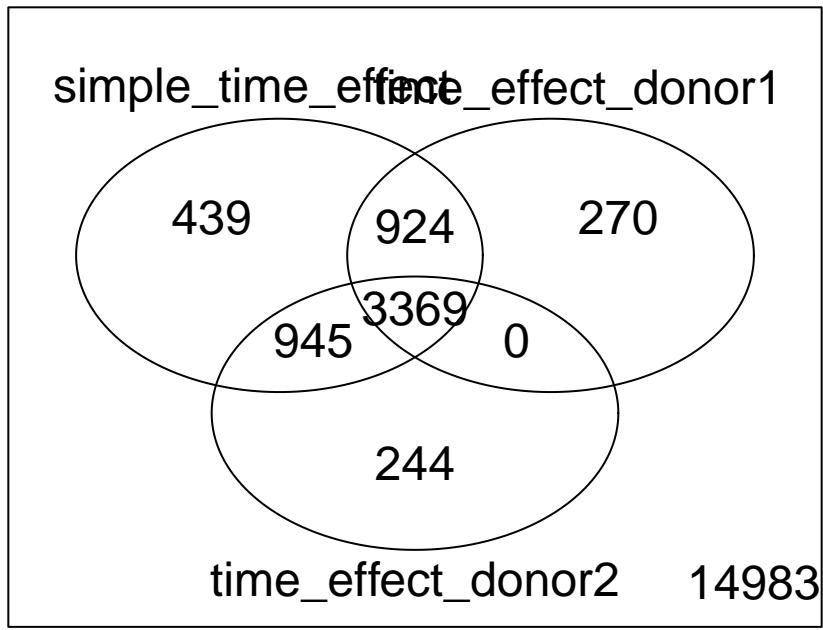


```
## Venn diagram of new and old lists
```

```
res = decideTests(special_fit, method="separate", adjust.method = "BH", p.value = 0.01) #Adjusted for c
vennDiagram(res[,c("simple_time_effect","time_effect_donor1", "time_effect_donor2")], include="up")
```



```
vennDiagram(res[,c("simple_time_effect","time_effect_donor1", "time_effect_donor2")], include="down")
```



Okay, so there's about 500 unique genes is either direction discovered by this test (as opposed to the d1->d3->d9 tests). We lose about the same number from either donor, but none that were in common between the donors, which seems promising. It seems that we capture the bulk of the adipogenic changes with just timepoints Pro, 0, 3 and 15 :)