six_donor_DGE_limma

Sarah Hp

2023-08-18

```
library(limma)
library(edgeR)
library(biomaRt)
library(ggplot2)
library(ggrepel)
library(tidyr)
library(RUVSeq)
library(clusterProfiler)
```

```
library(here)
```

```
## here() starts at /projects/imb-pkbphil/sp/rnaseq/six_donor_trans/splicing_paper
```

```
i_am("R/01_six_donor_DGE_limma.Rmd")
```

```
## here() starts at /projects/imb-pkbphil/sp/rnaseq/six_donor_trans/splicing_paper
```

```
knitr::opts_chunk$set(echo = TRUE, dev = c("pdf"), fig.path = "plots/six_donor_DGE_limma/")
```

Load experiment 1; select relevant samples

```
la = read.delim(here("02featureCounts/late_adipo_s4_s6.nomm.feature.counts"), skip=1)
colnames(la) = gsub("output.01hisat.","",gsub(".sorted.bam","", colnames(la)))

# select the bulk day15 samples
la = la[,c(1:6,grep("^(19|2[01234])", colnames(la)))]
head(la)
```

```
##              Geneid                                                          Chr
## 1 ENSG00000223972.5           chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1
## 2 ENSG00000227232.5 chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1
## 3 ENSG00000278267.1                                                        chr1
## 4 ENSG00000243485.5                                  chr1;chr1;chr1;chr1;chr1
## 5 ENSG00000284332.1                                                        chr1
## 6 ENSG00000237613.2                                  chr1;chr1;chr1;chr1;chr1
##                                                                  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 19.21423_S40 20.21424_S47 21.21425_S54
## 1     +;+;+;+;+;+;+;+;+;+   1735           0           0           0
## 2 -;-;-;-;-;-;-;-;-;-;-   1351           5          15          27
## 3                      -     68           6           6           5
## 4              +;+;+;+;+   1021           0           0           0
## 5                      +    138           0           0           0
## 6              -;-;-;-;-   1219           1           2           1
##   22.21426_S3 23.21427_S10 24.21428_S17
## 1           0            0            0
## 2          13           13           37
## 3           8            3            9
## 4           0            0            0
## 5           0            0            0
## 6           0            0            0
```

Load experiment 2; select relevant samples

```
fc = read.delim(here("02featureCounts/beige_rnaseq.nomm.feature.counts"), skip=1)
colnames(fc) = gsub("output.01hisat.","",gsub(".sorted.bam","", colnames(fc)))
head(fc)[1:9]
```

```
##              Geneid                                                          Chr
## 1 ENSG00000223972.5           chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1
## 2 ENSG00000227232.5 chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1;chr1
## 3 ENSG00000278267.1                                                        chr1
## 4 ENSG00000243485.5                                  chr1;chr1;chr1;chr1;chr1
## 5 ENSG00000284332.1                                                        chr1
## 6 ENSG00000237613.2                                  chr1;chr1;chr1;chr1;chr1
##                                                              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 1.22589_S146 2.22590_S149 3.22591_S154
```

```
## 1     +;+;+;+;+;+;+;+;+;+   1735           0           0           0
## 2 -;-;-;-;-;-;-;-;-;-;-;-   1351         323         193         171
## 3                       -     68          45          39          26
## 4              +;+;+;+;+;+   1021           0           0           0
## 5                       +    138           0           0           0
## 6              -;-;-;-;-;-   1219           0           0           0
```

Match experiments

```
nrow(la); nrow(fc)
```

```
## [1] 60668
```

```
## [1] 60668
```

```
counts = merge(fc, la, by=colnames(fc)[1:6])
#head(counts)
colnames(counts)
```

```
##  [1] "Geneid"         "Chr"            "Start"          "End"
##  [5] "Strand"         "Length"         "1.22589_S146"   "2.22590_S149"
##  [9] "3.22591_S154"   "4.22592_S128"   "5.22593_S132"   "6.22594_S136"
## [13] "7.22595_S140"   "8.22596_S143"   "9.22597_S147"   "10.22598_S150"
## [17] "11.22599_S155"  "12.22600_S129"  "13.22601_S133"  "14.22602_S137"
## [21] "15.22603_S141"  "16.22604_S144"  "17.22605_S148"  "18.22606_S151"
## [25] "19.22607_S156"  "20.22608_S130"  "21.22609_S134"  "22.22610_S138"
## [29] "23.22611_S142"  "24.22612_S145"  "25.22613_S153"  "26.22614_S152"
## [33] "27.22615_S157"  "28.22616_S131"  "29.22617_S135"  "30.22618_S139"
## [37] "19.21423_S40"   "20.21424_S47"   "21.21425_S54"   "22.21426_S3"
## [41] "23.21427_S10"   "24.21428_S17"
```

```
nrow(counts)
```

```
## [1] 60668
```

Save this to file for GEO submission later. [Reanalysis of s4/s6 white]

```
write.table(counts, here( "02featureCounts/six_donors.merged.nomm.counts"), sep="\t", quote=F, row.names
```

formatting

```
counts$Geneid = gsub("\\..*", "", counts$Geneid)
counts[counts$Geneid == "ENSG00000132170",7:ncol(counts)]# double check gene matching
```

```
##      1.22589_S146 2.22590_S149 3.22591_S154 4.22592_S128 5.22593_S132
## 6545        11069        13502        17008         7451        14564
##      6.22594_S136 7.22595_S140 8.22596_S143 9.22597_S147 10.22598_S150
## 6545         8291        11652        14165        17663         20856
##      11.22599_S155 12.22600_S129 13.22601_S133 14.22602_S137 15.22603_S141
## 6545         12708         10109         16121         17043         12563
```

```
##      16.22604_S144 17.22605_S148 18.22606_S151 19.22607_S156 20.22608_S130
## 6545          7407          8276         16290         10702         12934
##      21.22609_S134 22.22610_S138 23.22611_S142 24.22612_S145 25.22613_S153
## 6545         13346         10671          9431         12668         12939
##      26.22614_S152 27.22615_S157 28.22616_S131 29.22617_S135 30.22618_S139
## 6545          7333          5670          8552          5035          9837
##      19.21423_S40 20.21424_S47 21.21425_S54 22.21426_S3 23.21427_S10
## 6545        10822        12847        13837        8180        16506
##      24.21428_S17
## 6545        18386
```

Load info

```
info = readxl::read_xlsx(here("sample_info/publication_ids.xlsx"))
info$frac_assigned_to_genes = as.double(gsub("%","", info$percent_assigned_to_genes))/100
info$donor.condition = paste(info$donor, info$condition, sep=".")
head(info)
```

```
## # A tibble: 6 x 11
##   sample       time    rep donor    condition name               dataset fc_file
##   <chr>        <chr> <dbl> <chr>    <chr>     <chr>              <chr>   <chr>
## 1 19-21423_S40 day15     1 subject4 white     day15_subject4_wh~ late_a~ beige_~
## 2 20-21424_S47 day15     2 subject4 white     day15_subject4_wh~ late_a~ beige_~
## 3 21-21425_S54 day15     3 subject4 white     day15_subject4_wh~ late_a~ beige_~
## 4 22-21426_S3  day15     1 subject6 white     day15_subject6_wh~ late_a~ beige_~
## 5 23-21427_S10 day15     2 subject6 white     day15_subject6_wh~ late_a~ beige_~
## 6 24-21428_S17 day15     3 subject6 white     day15_subject6_wh~ late_a~ beige_~
## # i 3 more variables: percent_assigned_to_genes <chr>,
## #   frac_assigned_to_genes <dbl>, donor.condition <chr>
```

```
counts$Geneid = gsub("\\..*", "", counts$Geneid)
counts[counts$Geneid == "ENSG00000132170",7:ncol(counts)]# double check gene matching
```

```
##      1.22589_S146 2.22590_S149 3.22591_S154 4.22592_S128 5.22593_S132
## 6545        11069        13502        17008         7451        14564
##      6.22594_S136 7.22595_S140 8.22596_S143 9.22597_S147 10.22598_S150
## 6545         8291        11652        14165        17663         20856
##      11.22599_S155 12.22600_S129 13.22601_S133 14.22602_S137 15.22603_S141
## 6545         12708         10109         16121         17043         12563
##      16.22604_S144 17.22605_S148 18.22606_S151 19.22607_S156 20.22608_S130
## 6545          7407          8276         16290         10702         12934
##      21.22609_S134 22.22610_S138 23.22611_S142 24.22612_S145 25.22613_S153
## 6545         13346         10671          9431         12668         12939
##      26.22614_S152 27.22615_S157 28.22616_S131 29.22617_S135 30.22618_S139
## 6545          7333          5670          8552          5035          9837
##      19.21423_S40 20.21424_S47 21.21425_S54 22.21426_S3 23.21427_S10
## 6545        10822        12847        13837        8180        16506
##      24.21428_S17
## 6545        18386
```

Make object

```
rownames(info) = gsub("-",".", info$sample)
```

```
## Warning: Setting row names on a tibble is deprecated.
```

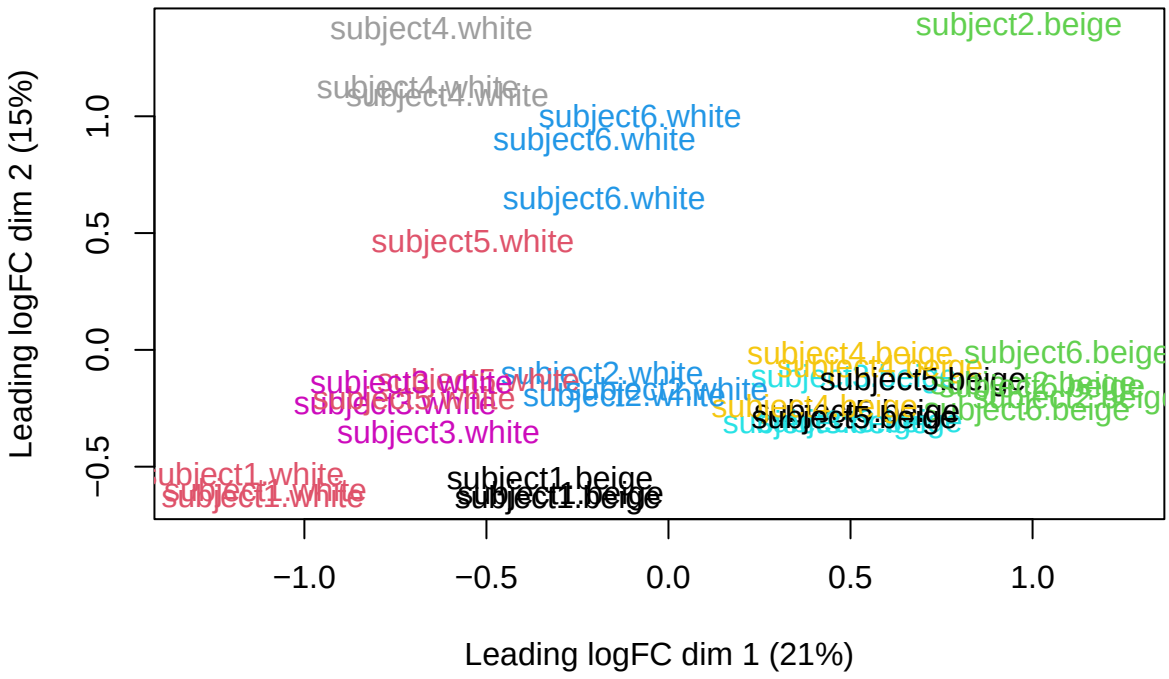
```
info = info[colnames(counts)[7:ncol(counts)],]#make sure order is the same
summary(rownames(info) == colnames(counts)[7:ncol(counts)])
```

```
##    Mode   FALSE
## logical      36
```

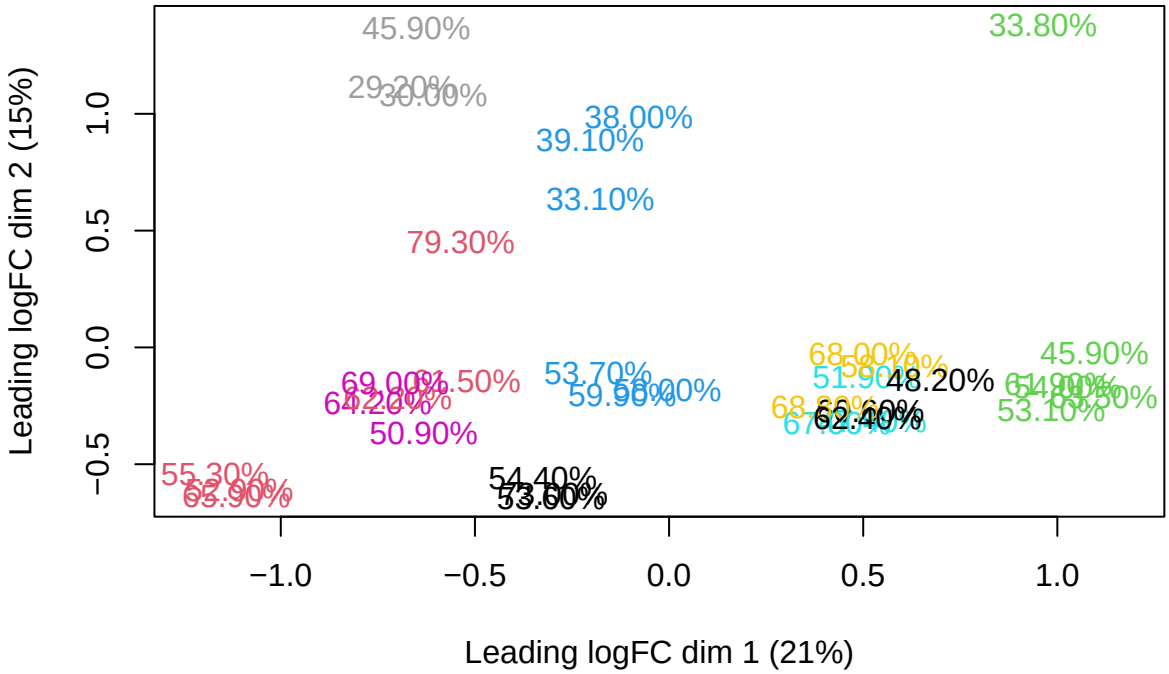
```
ob = DGEList(counts = data.matrix(counts[7:ncol(counts)]),
             samples = info,
             group = info$donor.condition,
             genes = counts[c(1,6)])
rownames(ob$counts) = ob$genes$Geneid
#head(ob)

summary(ob$counts[grep("ENSG00000132170", rownames(ob$counts)),c(1,5:ncol(ob$counts))]) ##PPARG check
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5035    9431   12563   12075   14165   20856
```

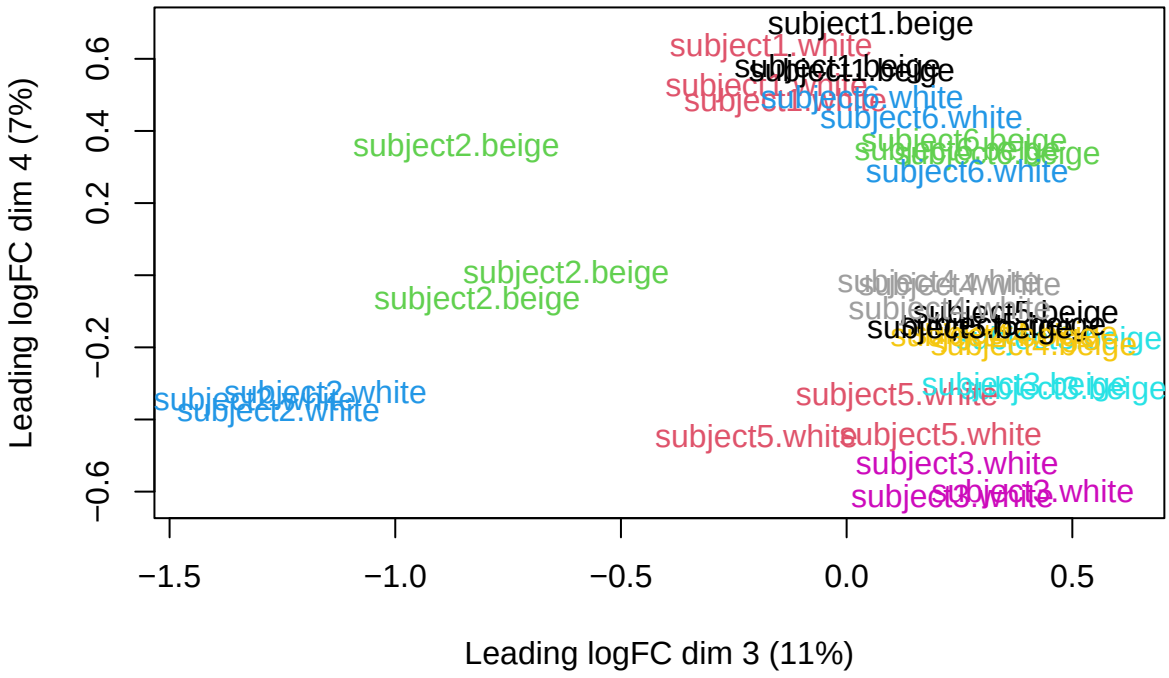
```
plotMDS(ob, top = 5000, labels =ob$samples$group, col = c(1:12)[ob$samples$group])
```



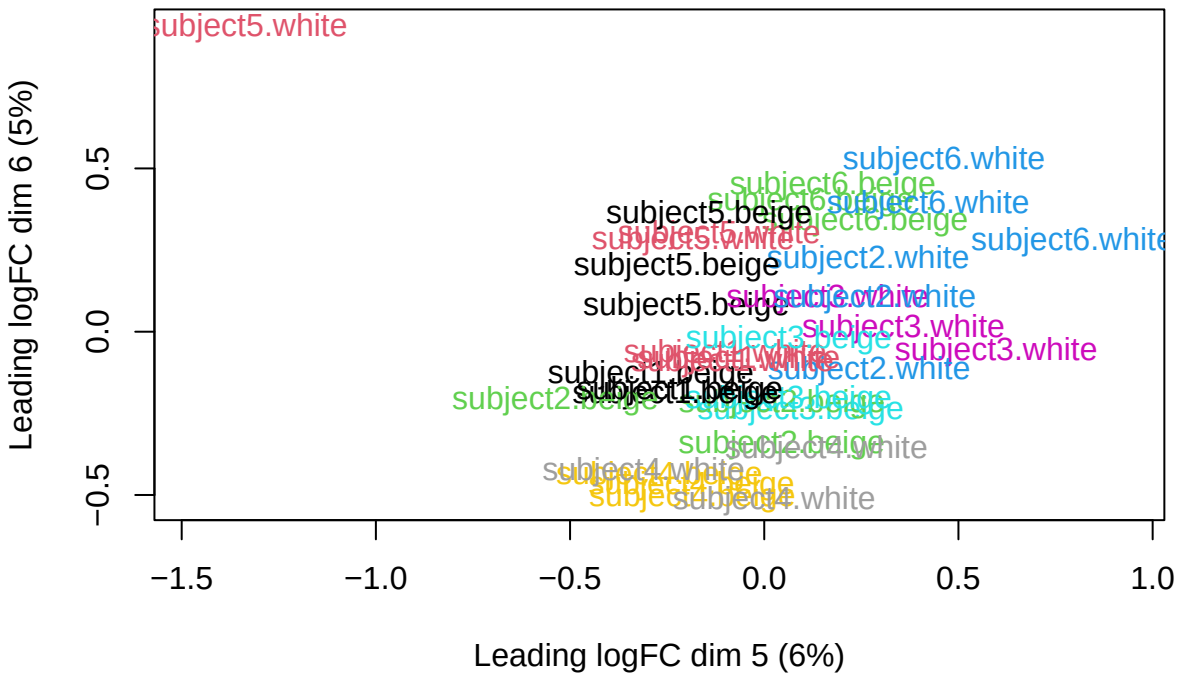
```
library(RColorBrewer)
to_col = colorRampPalette(c("blue","red"))(25)
plotMDS(ob, top = 5000, labels =ob$samples$percent_assigned_to_genes, col = c(1:12)[ob$samples$group])
```



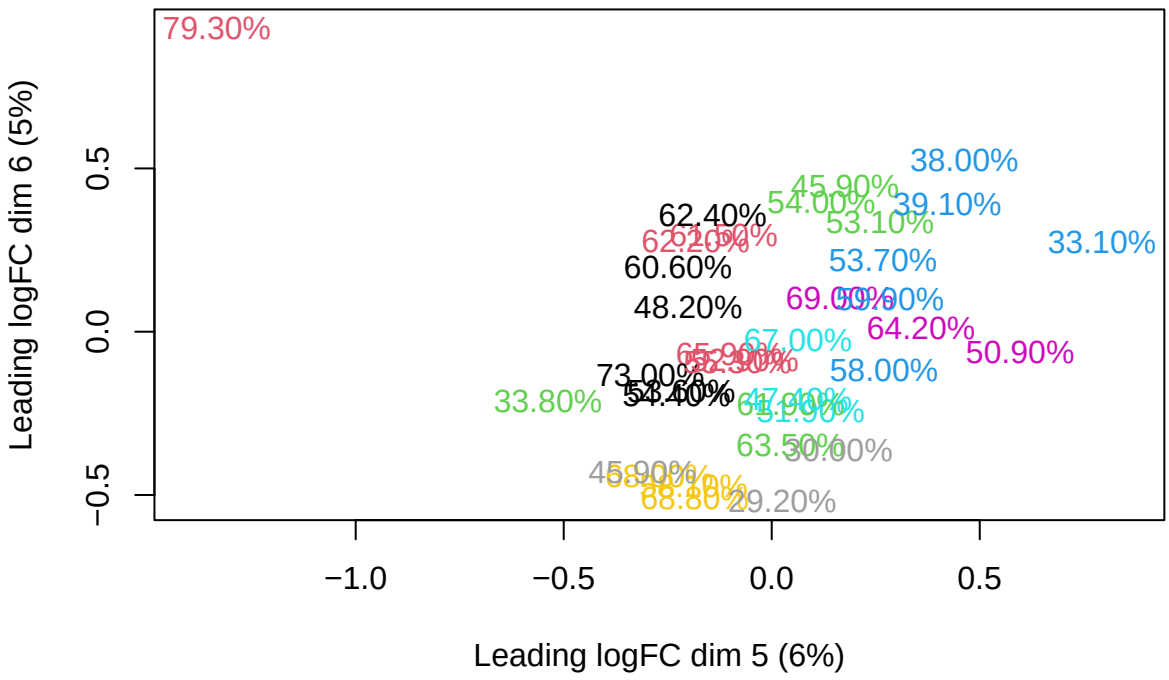

```
plotMDS(ob, top = 5000, labels =ob$samples$group, col = c(1:12)[ob$samples$group],
        dim.plot = c(3,4))
```




```
plotMDS(ob, top = 5000, labels =ob$samples$group, col = c(1:12)[ob$samples$group],
        dim.plot = c(5,6))
```

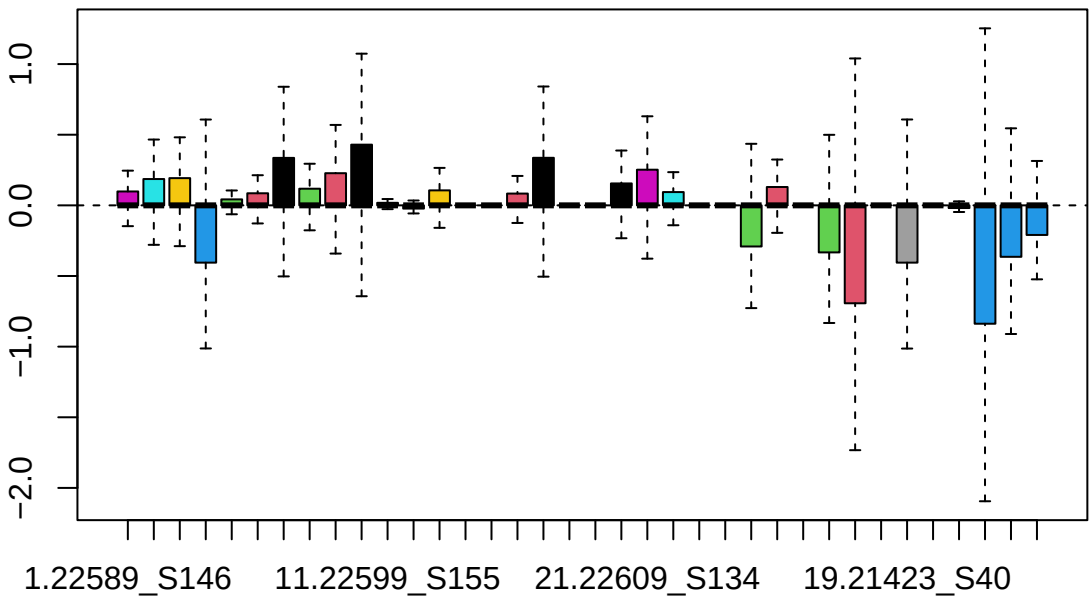



```
plotMDS(ob, top = 5000, labels =ob$samples$percent_assigned_to_genes, col = c(1:12)[ob$samples$group],
        dim.plot = c(5,6))
```

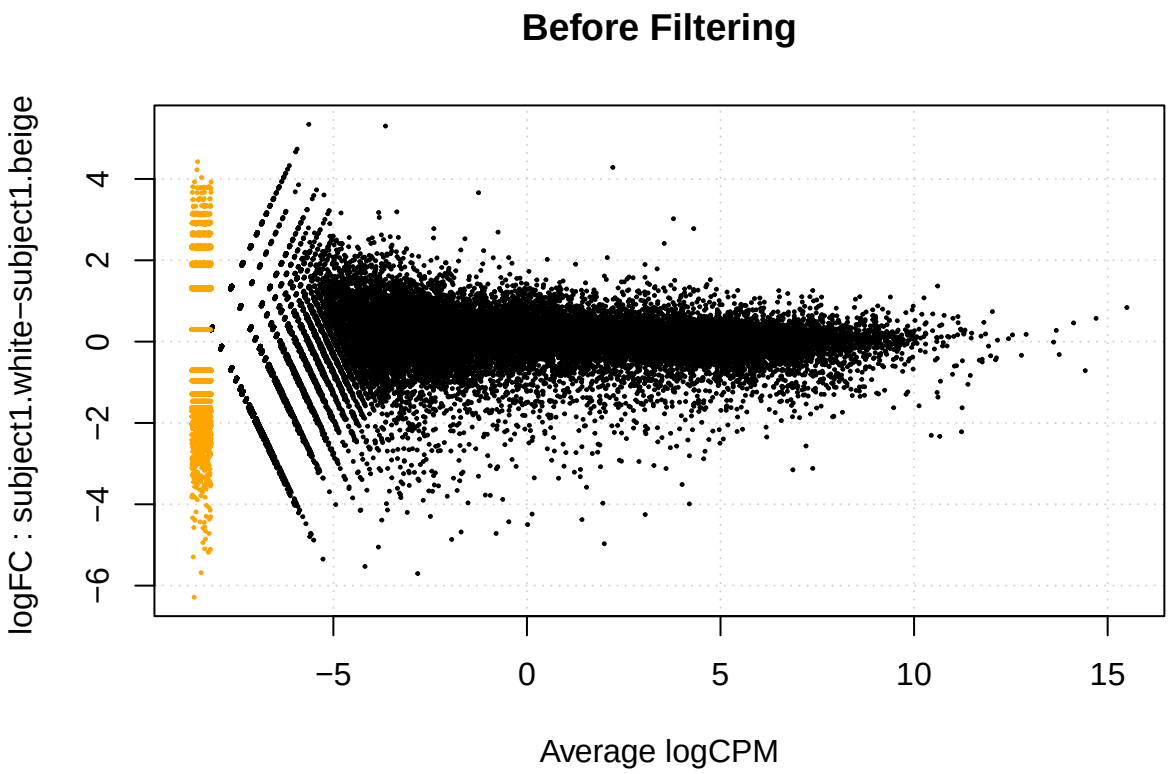


Filter

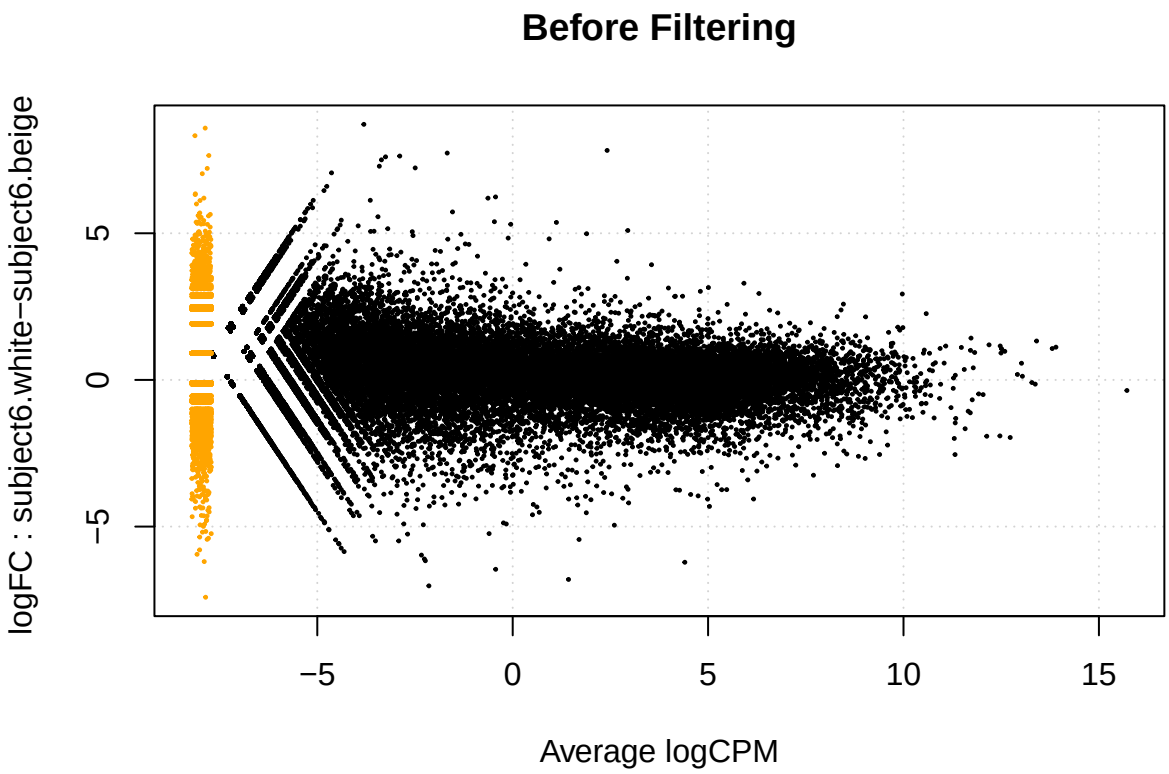
```
plotRLE(ob$counts, outline=FALSE, col=ob$samples$group, main="Before Filtering")
```



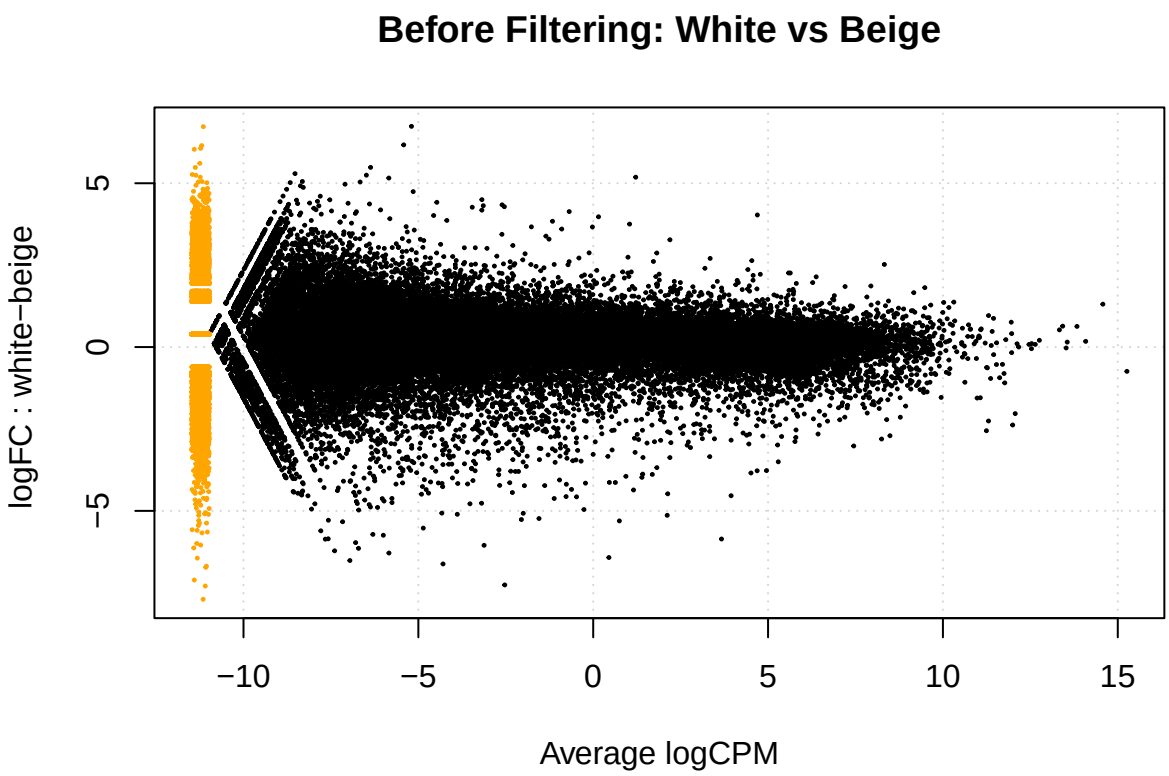

```
plotSmear(ob, main = "Before Filtering")
```




```
plotSmear(ob, main = "Before Filtering", pair=c("subject6.beige","subject6.white"))
```




```
ob$samples$group = as.factor(ob$samples$condition)
plotSmear(ob, main = "Before Filtering: White vs Beige")
```




```
nrow(ob) #60 668 genes before filtering
```

```
## [1] 60668
```

Slight shift in mean between white and beige for s6.

Conduct a filtering using white and beige min.count and min.prop are default I just thought I'd list them here

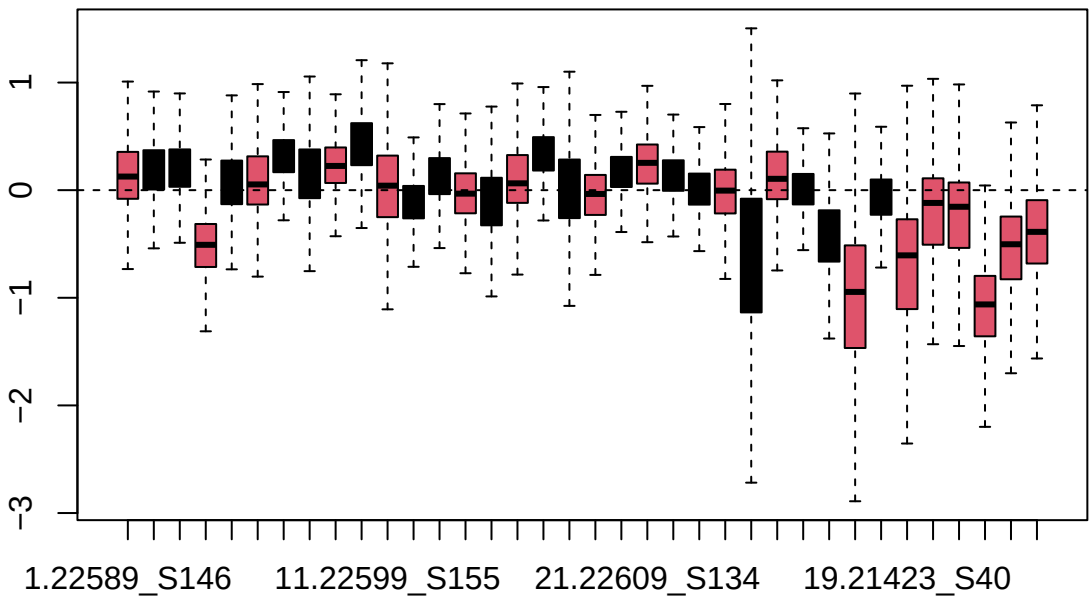
```
levels(ob$samples$group)
```

```
## [1] "beige" "white"
```

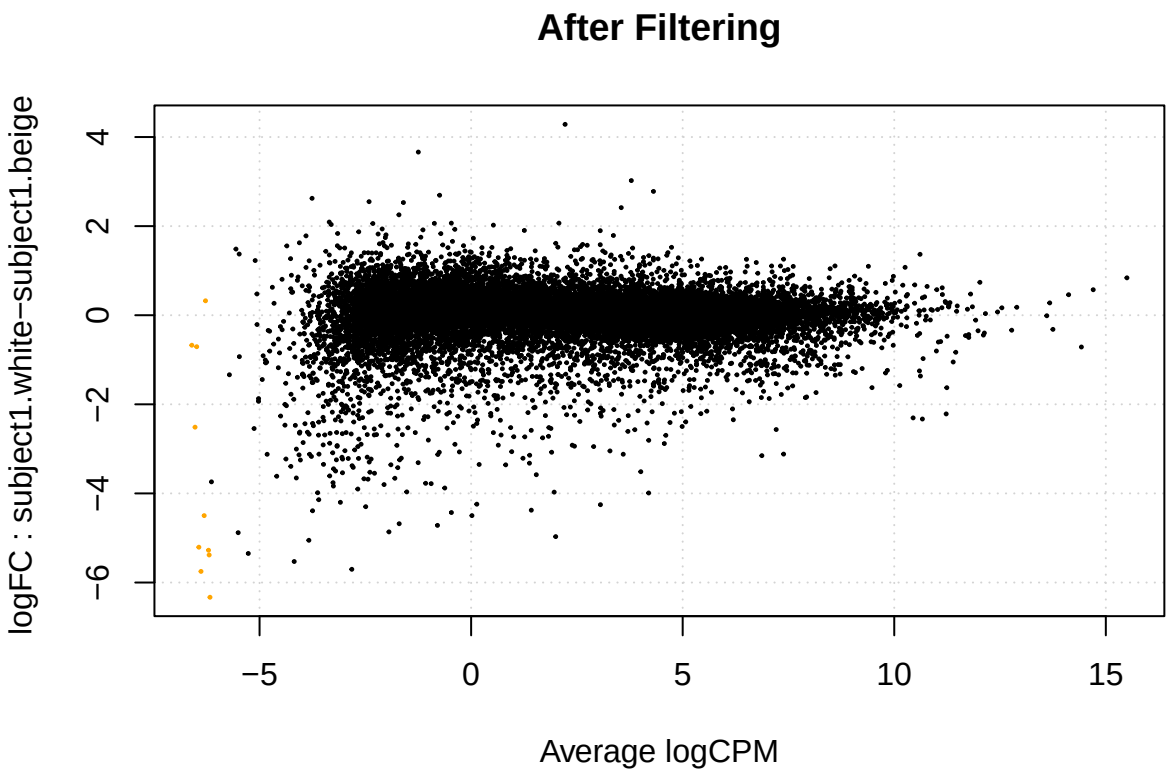
```
filt = ob[filterByExpr(ob, min.count=10, min.prop=0.7),, keep.lib.sizes=FALSE]
nrow(filt) #18061
```

```
## [1] 18061
```

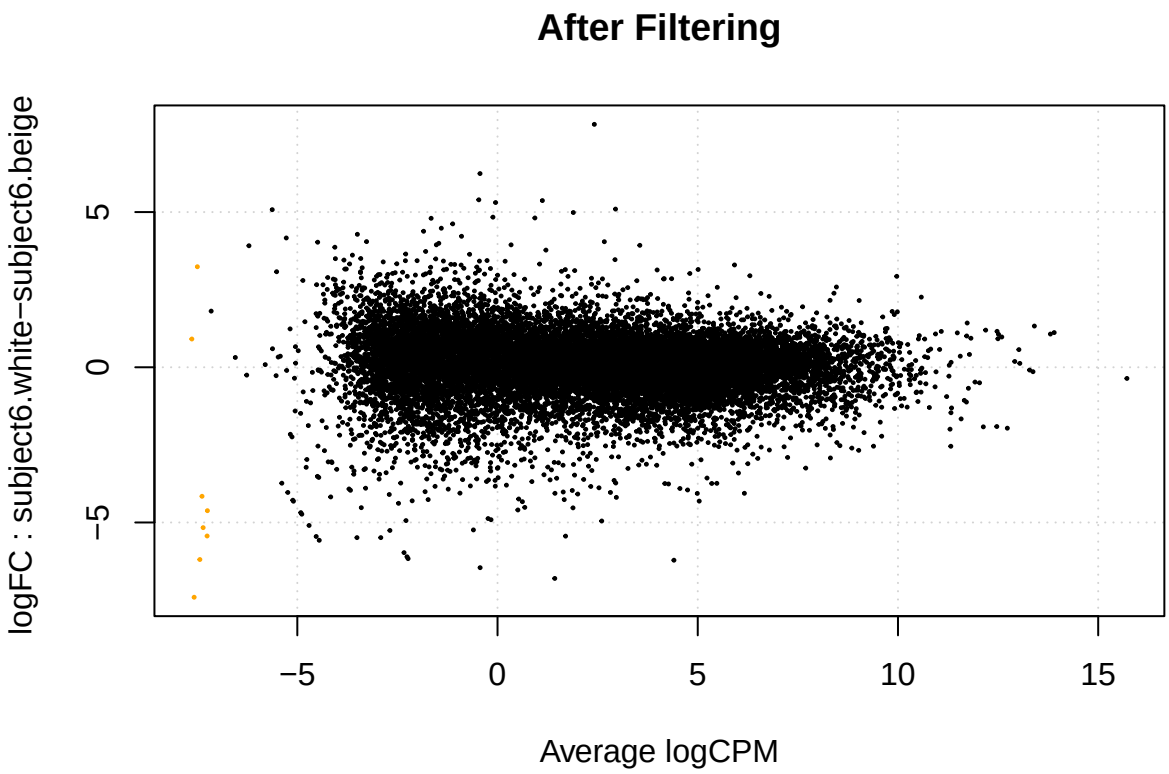
```
plotRLE(filt$counts, outline=FALSE, col=filt$samples$group, main="After Filtering")
```



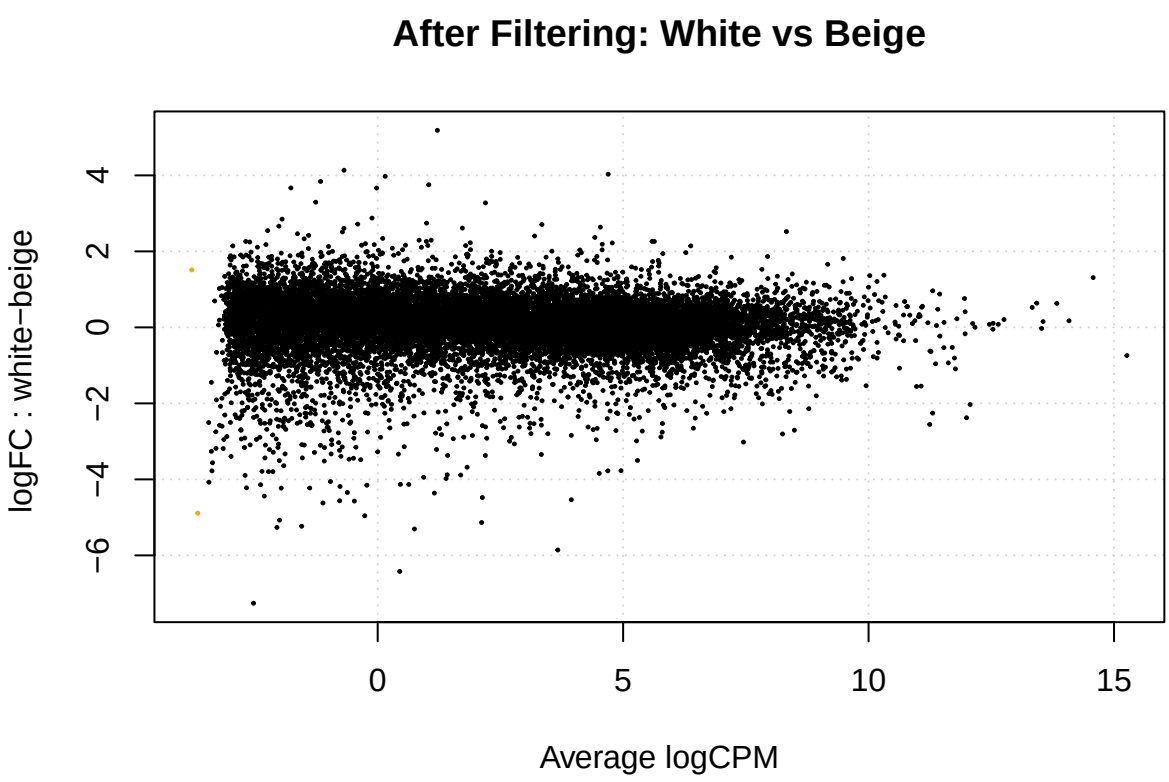

```
filt$samples$group = as.factor(filt$samples$donor.condition)
plotSmear(filt, main = "After Filtering")
```




```
plotSmear(filt, main = "After Filtering", pair=c("subject6.beige","subject6.white"))
```




```
filt$samples$group = as.factor(filt$samples$condition)
plotSmear(filt, main = "After Filtering: White vs Beige")
```



Sanity checks

```
summary(filt$counts[grep("ENSG00000132170", rownames(filt$counts)),]) #PPARG is included and looks high
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5035    9211   12616   12123   14265   20856
```

```
summary(filt$counts[grep("ENSG00000228630", rownames(filt$counts)),]) #HOTAIR is still detected at low
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   26.00   91.25  146.50  339.61  454.00 1520.00
```

```
rownames(filt$counts)[grep("ENSG00000223972.5", rownames(filt$counts))] #This lowly expr gene is filter
```

```
## character(0)
```

```
summary(filt$counts[grep("ENSG00000176194", rownames(filt$counts)),]) #CIDEA at low levels in beige sam
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0    17.0    46.5   310.0   521.8  1901.0
```

Norm Factors

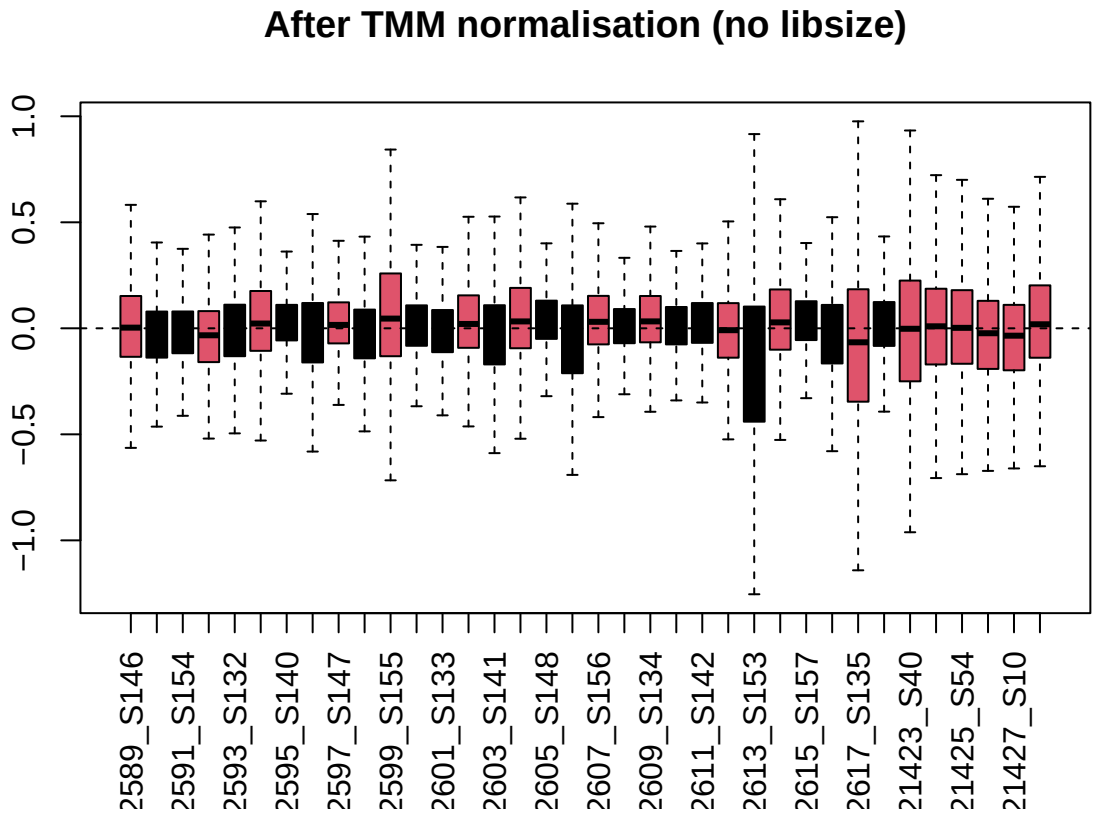
```
filt = calcNormFactors(filt, method="TMM")
filt$samples[c("norm.factors")]
```

```
##               norm.factors
## 1.22589_S146     1.0048258
## 2.22590_S149     0.9551677
## 3.22591_S154     0.9775318
## 4.22592_S128     0.9512721
## 5.22593_S132     0.9729840
## 6.22594_S136     1.0364256
## 7.22595_S140     1.0386764
## 8.22596_S143     0.9575702
## 9.22597_S147     1.0409671
## 10.22598_S150    0.9547073
## 11.22599_S155    1.1037450
## 12.22600_S129    1.0151597
## 13.22601_S133    0.9888540
## 14.22602_S137    1.0428647
## 15.22603_S141    0.9422844
## 16.22604_S144    1.0496733
## 17.22605_S148    1.0612113
## 18.22606_S151    0.9162526
## 19.22607_S156    1.0529346
## 20.22608_S130    1.0062590
## 21.22609_S134    1.0560136
## 22.22610_S138    1.0151836
## 23.22611_S142    1.0381196
## 24.22612_S145    0.9904936
## 25.22613_S153    0.8176175
## 26.22614_S152    1.0443173
## 27.22615_S157    1.0504565
## 28.22616_S131    0.9522480
## 29.22617_S135    0.9044118
## 30.22618_S139    1.0246099
## 19.21423_S40     1.0362388
## 20.21424_S47     1.0329218
## 21.21425_S54     1.0242432
## 22.21426_S3      0.9804359
## 23.21427_S10     0.9537317
## 24.21428_S17     1.0674308
```

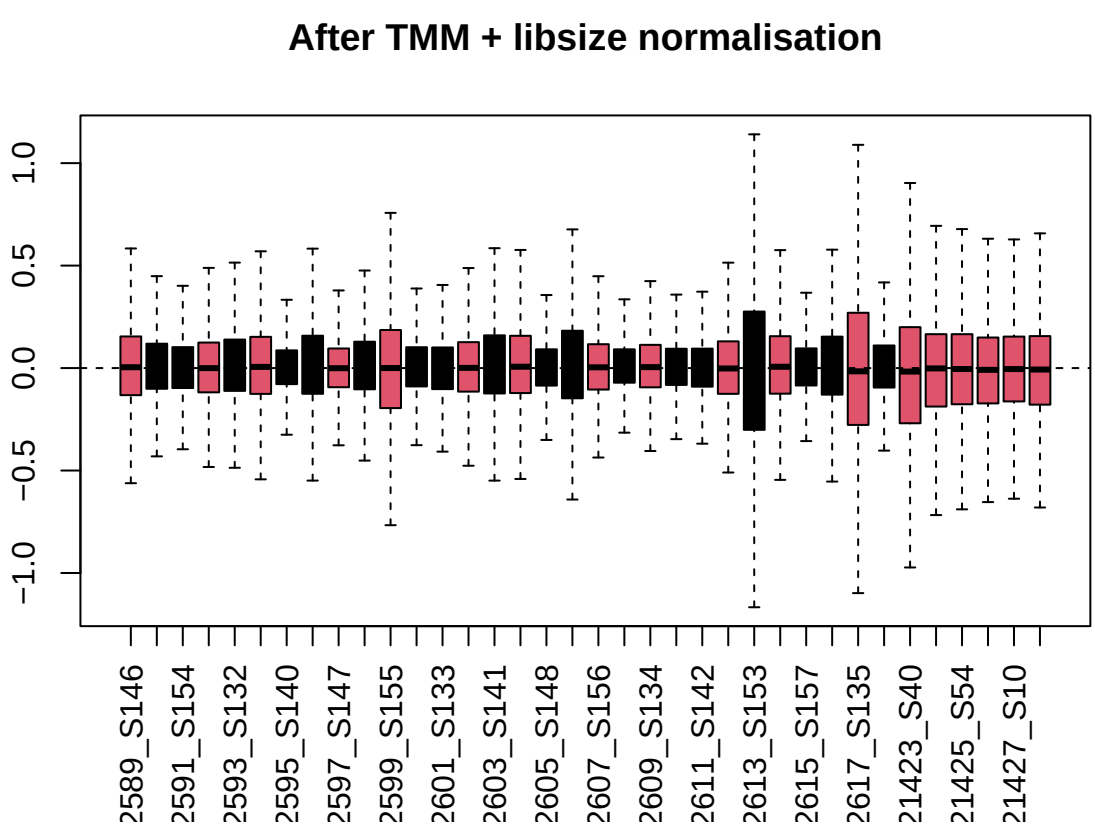
```
summary(filt$samples$norm.factors)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.8176  0.9570  1.0152  1.0016  1.0414  1.1037
```

```
plotRLE(cpm(filt, normalized.lib.sizes = F), outline=F, col=filt$samples$group, las=3,
        main="After TMM normalisation (no libsize)")
```




```
plotRLE(cpm(filt, normalized.lib.sizes = T), outline=F, col=filt$samples$group, las=3,
        main="After TMM + libsize normalisation")
```



Annotate gene lists

```
mart <- biomaRt::useMart(biomart = "ensembl",
  dataset = "hsapiens_gene_ensembl",
   host = "https://sep2019.archive.ensembl.org")

#searchFilters(mart, pattern="ensembl")

annot = getBM(c("external_gene_name", "description", "gene_biotype","ensembl_gene_id", "ensembl_gene_id
              filters = "ensembl_gene_id",
              values = filt$genes$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]
##     gene_biotype ensembl_gene_id ensembl_gene_id_version
## 1 protein_coding ENSG00000000003      ENSG00000000003.15
## 2 protein_coding ENSG00000000005       ENSG00000000005.6
```

```
## [1] 18058     5
```

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

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

#match the gene lists from genes and counts
nrow(filt$counts); nrow(filt$genes)
```

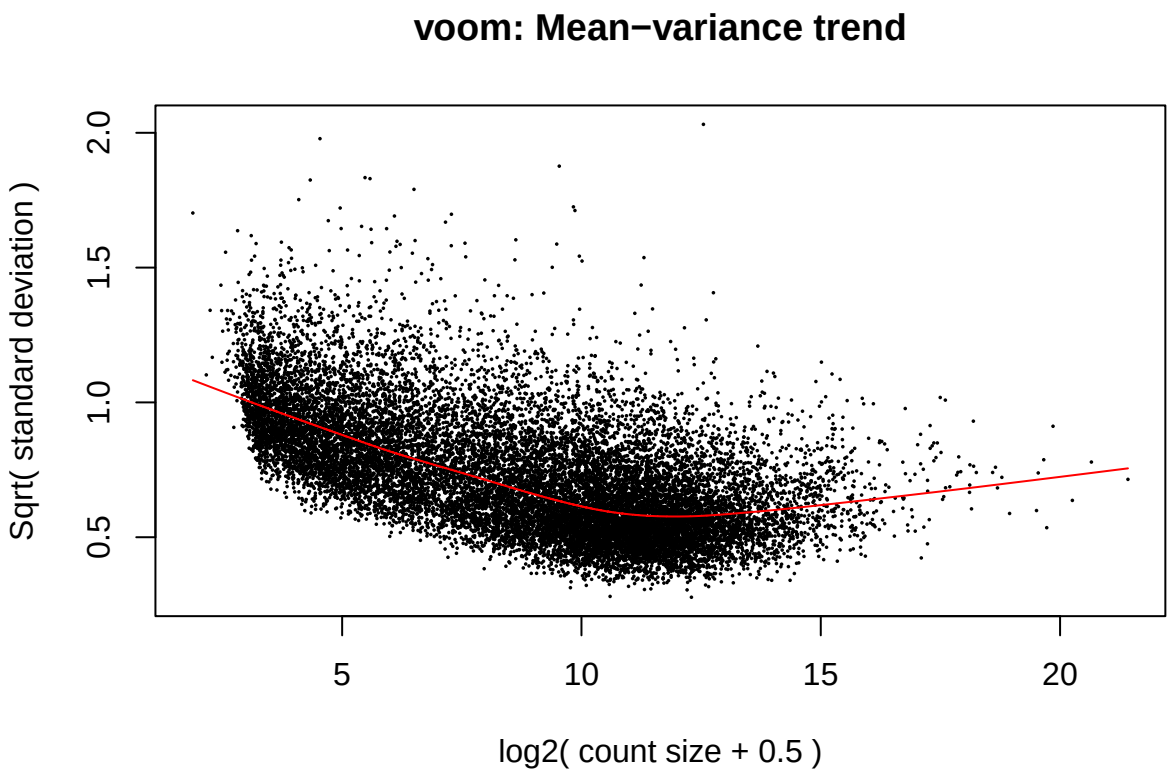
```
## [1] 18061
```

```
## [1] 18061
```

```
save(filt, file = here("03limma/DGElist_ob_limma_filt.RData"))
```

voom normlisation

```
norm = voom(filt, plot=T)
```



Limma

```
#head(norm$design)

#set some factor levels
norm$targets$group = paste(norm$targets$condition, norm$targets$donor, sep=".") #explicit
#simplify names for intercepting
norm$targets$group = gsub("subject","s", norm$targets$group)
norm$targets$group = factor(gsub("eige|hite","", norm$targets$group))
norm$targets$group
```

```
##  [1] w.s3 b.s3 b.s4 w.s2 b.s2 w.s1 b.s1 b.s6 w.s5 b.s5 w.s3 b.s3 b.s4 w.s2 b.s2
## [16] w.s1 b.s1 b.s6 w.s5 b.s5 w.s3 b.s3 b.s4 w.s2 b.s2 w.s1 b.s1 b.s6 w.s5 b.s5
## [31] w.s4 w.s4 w.s4 w.s6 w.s6 w.s6
## Levels: b.s1 b.s2 b.s3 b.s4 b.s5 b.s6 w.s1 w.s2 w.s3 w.s4 w.s5 w.s6
```

```
design = model.matrix(~0 +frac_assigned_to_genes + group, data=norm$targets)
colnames(design) = c("frac_assigned_to_genes",levels(norm$targets$group))
head(design)
```

```
##              frac_assigned_to_genes b.s1 b.s2 b.s3 b.s4 b.s5 b.s6 w.s1 w.s2
## 1.22589_S146                  0.642    0    0    0    0    0    0    0    0
## 2.22590_S149                  0.519    0    0    1    0    0    0    0    0
## 3.22591_S154                  0.581    0    0    0    1    0    0    0    0
## 4.22592_S128                  0.537    0    0    0    0    0    0    0    1
## 5.22593_S132                  0.619    0    1    0    0    0    0    0    0
## 6.22594_S136                  0.659    0    0    0    0    0    0    1    0
##              w.s3 w.s4 w.s5 w.s6
## 1.22589_S146    1    0    0    0
## 2.22590_S149    0    0    0    0
## 3.22591_S154    0    0    0    0
## 4.22592_S128    0    0    0    0
## 5.22593_S132    0    0    0    0
## 6.22594_S136    0    0    0    0
```

```
explicit_fit = lmFit(norm, design=design)
coi = makeContrasts(s4 = b.s4- w.s4,
                    s3 = b.s3 - w.s3,
                    s2 = b.s2 - w.s2,
                    s1 = b.s1 - w.s1,
                    s6 = b.s6 - w.s6,
                    s5 = b.s5 - w.s5,
                      levels = design)
cfit = contrasts.fit(explicit_fit, coi)
efit = eBayes(cfit, robust=T)

results= decideTests(efit, method="separate", adjust.method = "BH", p.value = 0.01)
summary(results)
```

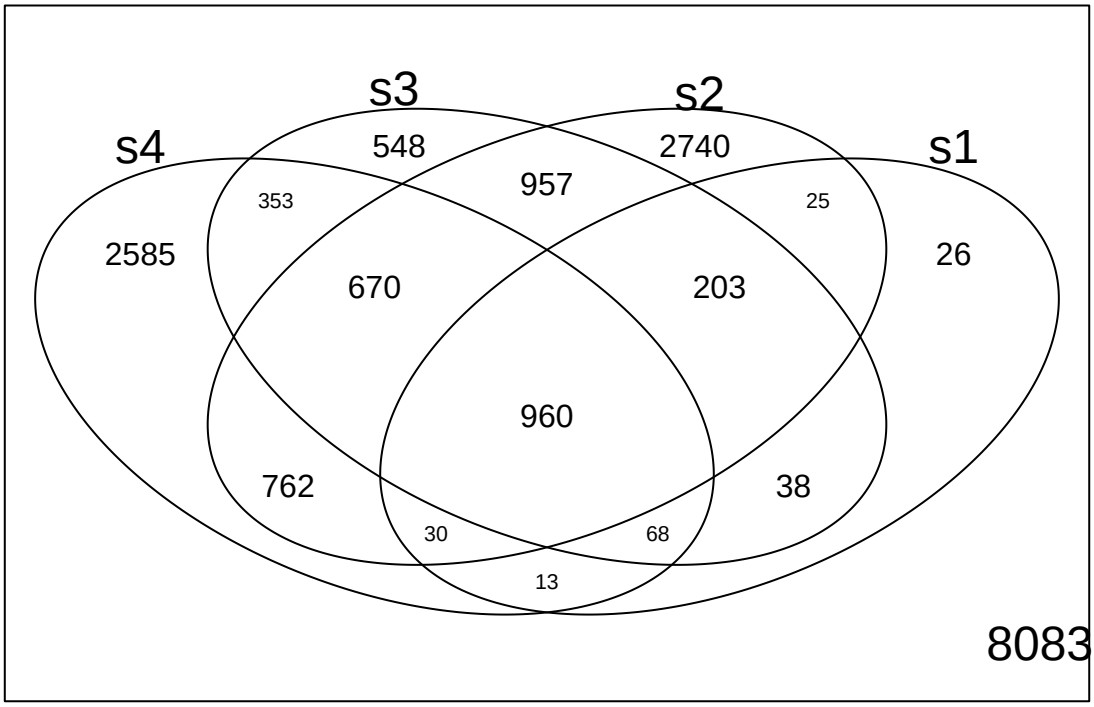
```
##           s4    s3    s2    s1    s6    s5
## Down    1155   798  1941    84  1520   971
```

```
## NotSig 15319 15656 13868 17223 14562 15465
## Up      1587  1607  2252   754  1979  1625
```

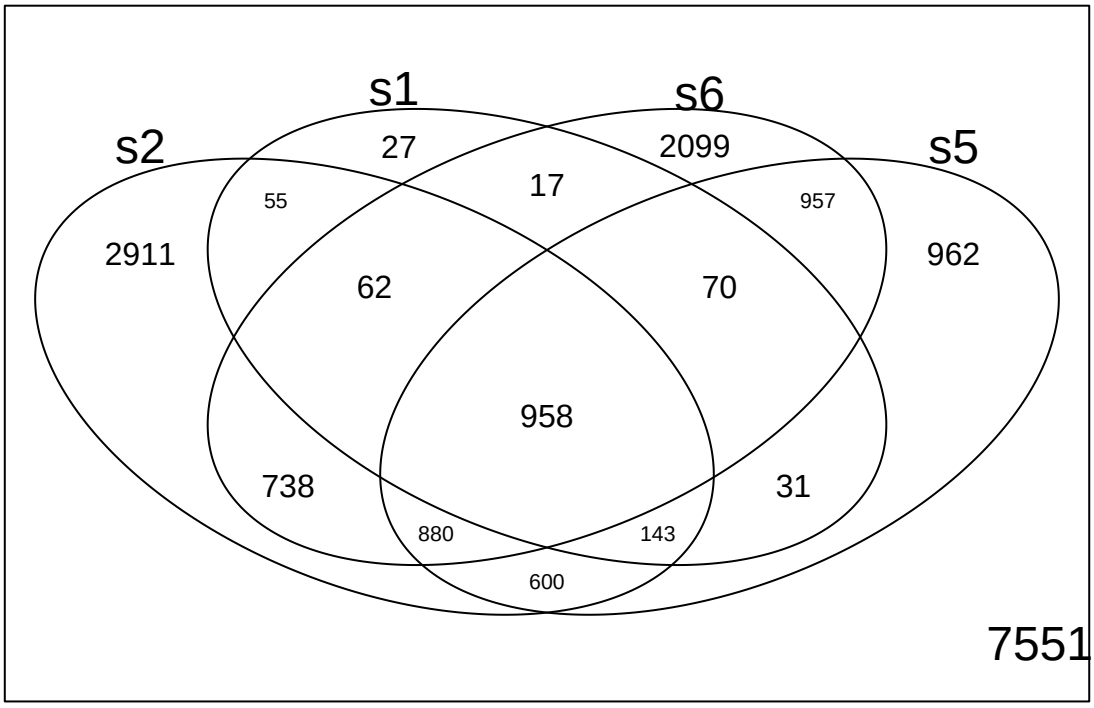
```
results= decideTests(efit, method="separate", adjust.method = "BH", p.value = 0.05)
summary(results)
```

```
##           s4    s3    s2    s1    s6    s5
## Down    2430  1514  3189   276  2668  2039
## NotSig 12620 14264 11714 16698 12280 13460
## Up      3011  2283  3158  1087  3113  2562
```

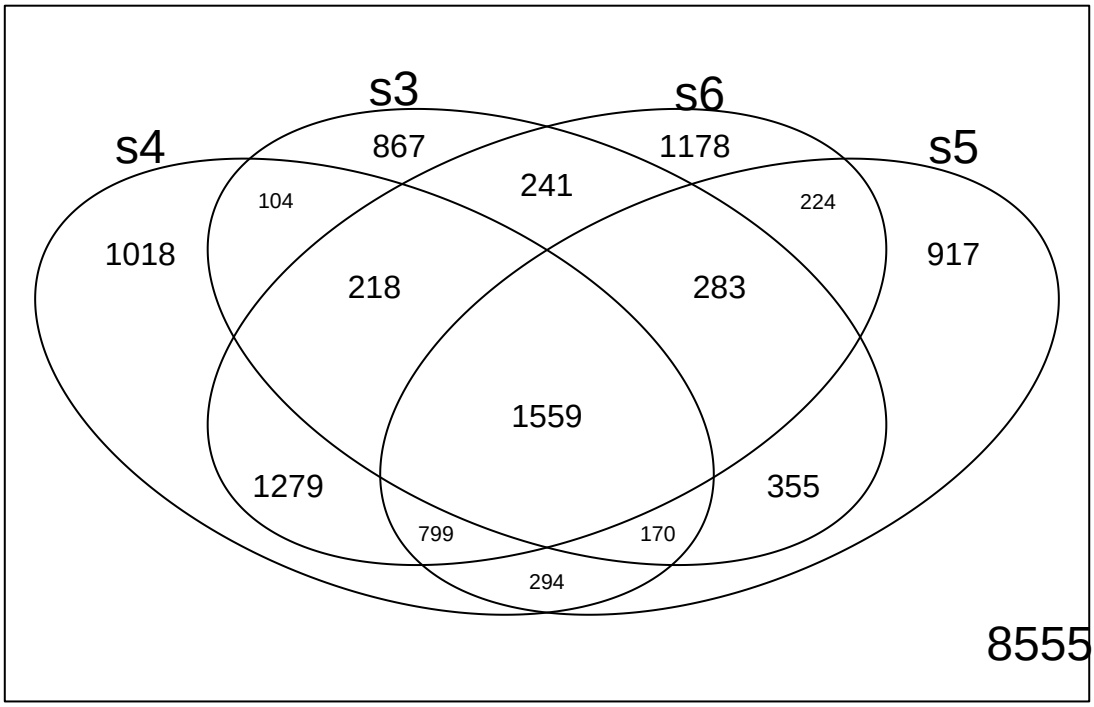
```
vennDiagram(results[,1:4])
```




```
vennDiagram(results[,c(3:6)])
```




```
vennDiagram(results[,c(1:2,5:6)])
```



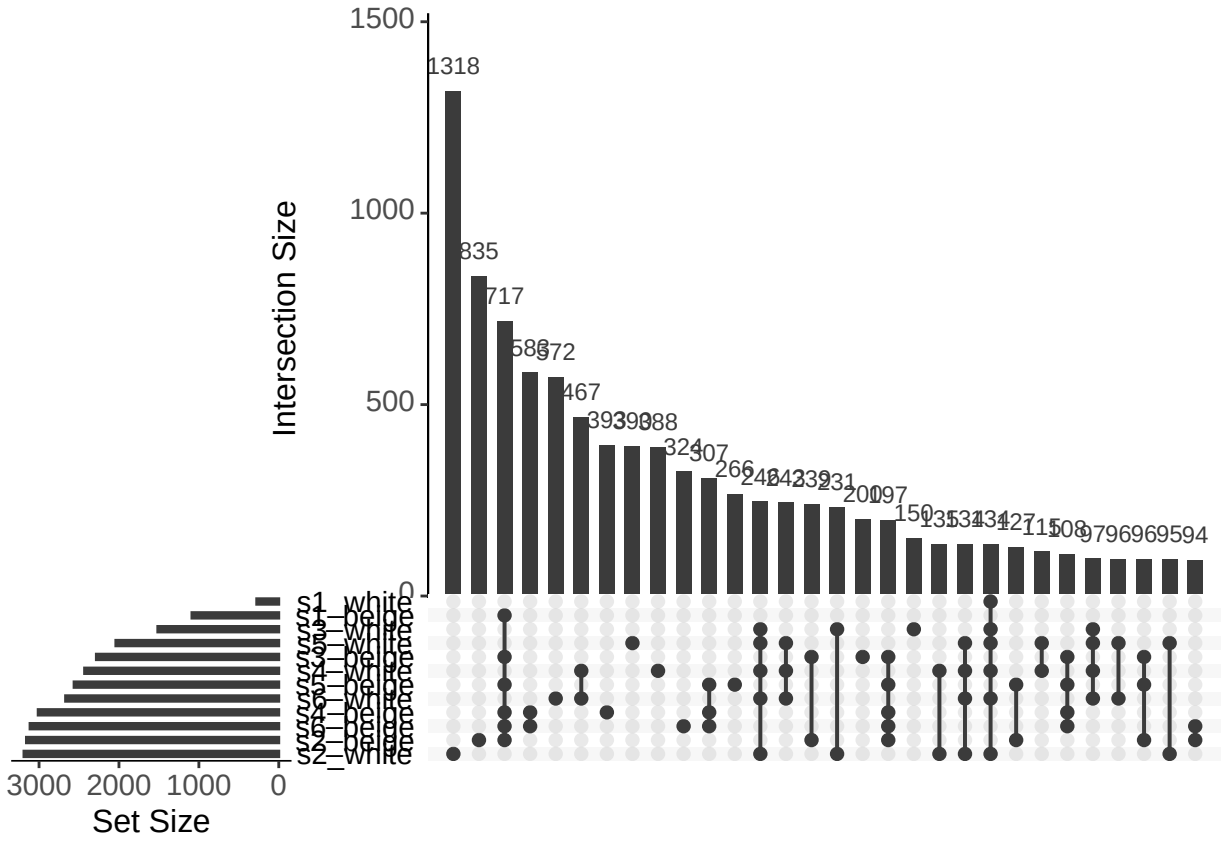
s4 has a small amount of independent genes, but not more than subject2.

Upsets

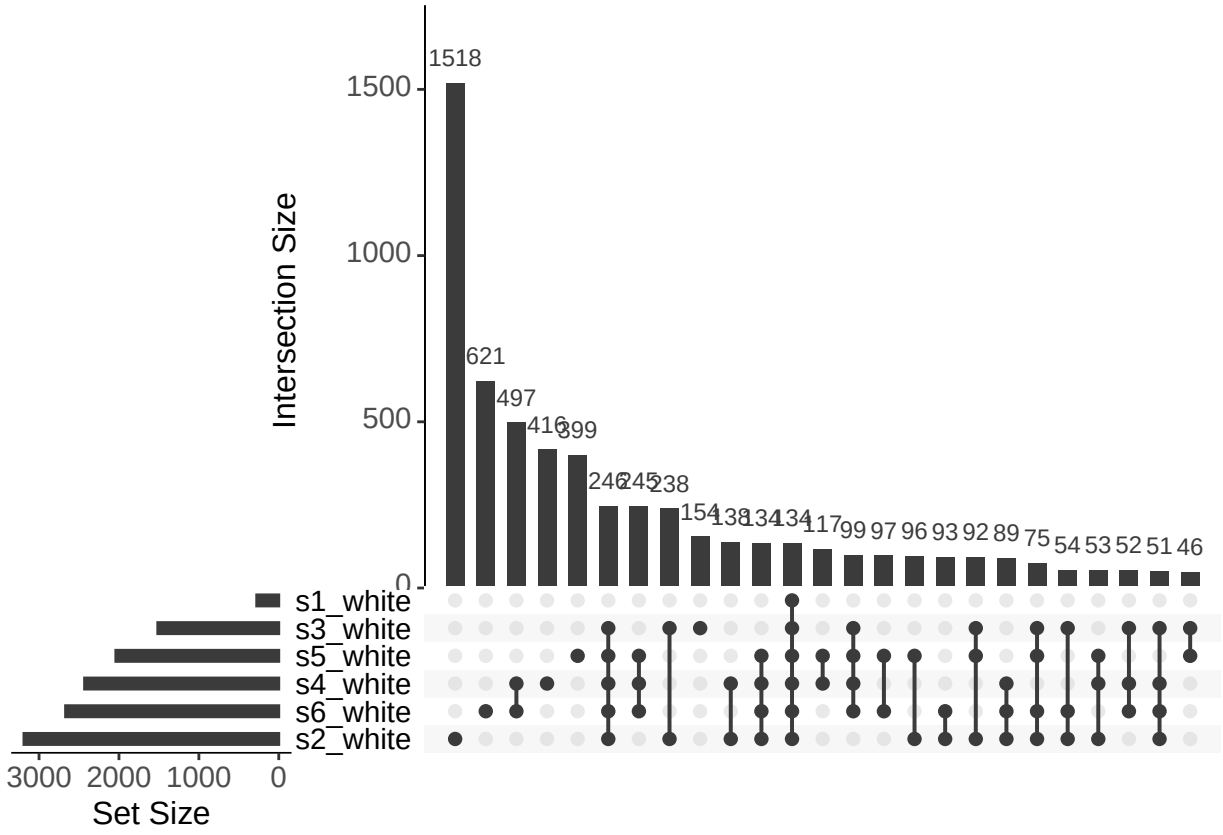
```
make_upset_table = function(res) {
  library(UpSetR)

  res= data.frame(res)
  new_bin = matrix(NA,ncol=ncol(res)*2, nrow=nrow(res))
  colnames(new_bin) = paste(rep(colnames(res),each=2), c("beige","white"), sep= "_")
  rownames(new_bin) = rownames(res)
  for (i in 1:nrow(res)){
    jcount=0
    for (j in 1:length(res[i,])){
        nj = j + jcount
        v = res[i,j]
        if (v == 0){
            new_bin[i,nj] = 0
            new_bin[i,nj+1] = 0
        } else if (v == 1){
            new_bin[i,nj] = 1
            new_bin[i,nj+1] = 0
        } else if (v == -1){
            new_bin[i,nj] = 0
```

```
            new_bin[i,nj+1] = 1
        } else {print("Error")}

        jcount = jcount + 1
    }
  }
  summary(new_bin)
  new_bin = as.data.frame(new_bin)
  #print(upset(new_bin,
   #   order.by = "freq", nintersects=25))
  return(new_bin)
}
```

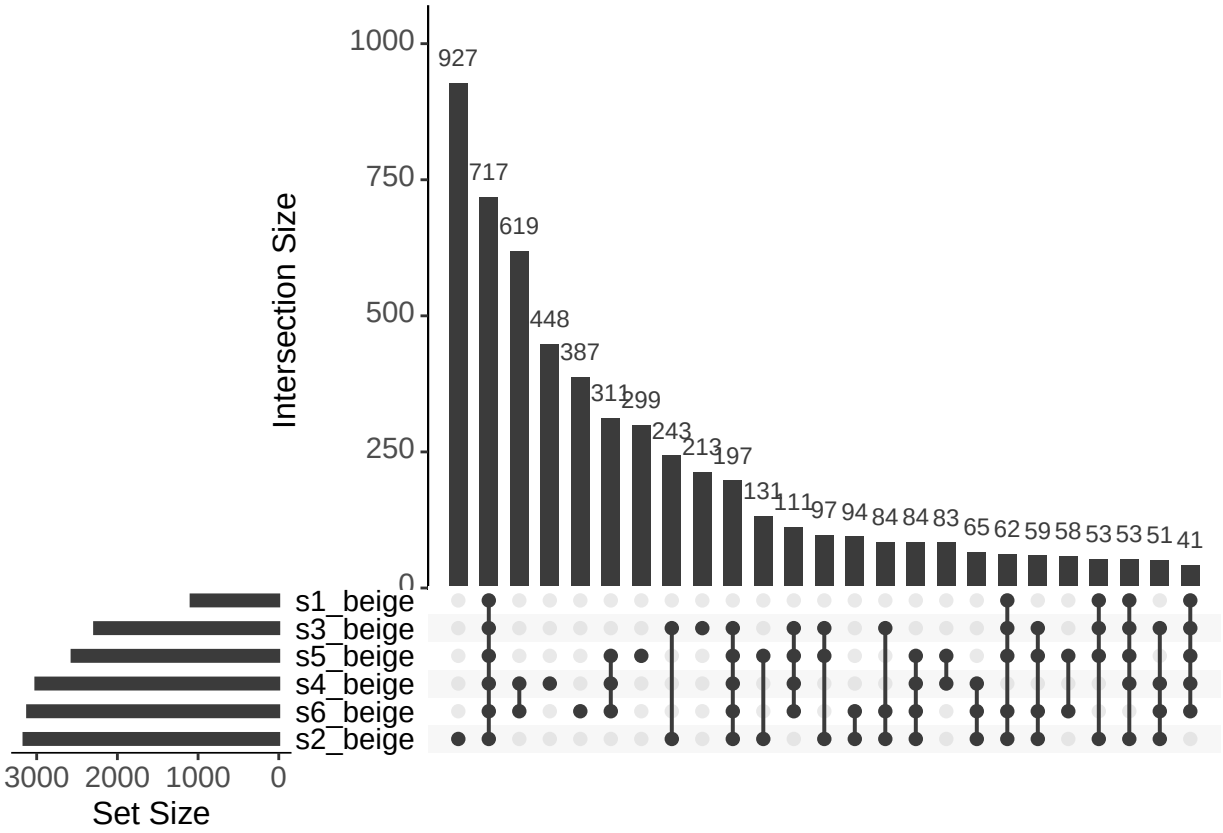
```
results_bin = make_upset_table(results)
upset(results_bin,nsets=14,
      order.by = "freq", nintersects=30, text.scale=1.5)
```




```
upset(results_bin[,grep("white", colnames(results_bin))],nsets=14,
      order.by = "freq", nintersects=25, text.scale=1.5)
```




```
upset(results_bin[,grep("beige", colnames(results_bin))],nsets=14,
      order.by = "freq", nintersects=25, text.scale=1.5)
```



Overall pvalue

We can extract merely like this

```
anytable = topTable(efit, number = Inf, adjust.method = "BH",
                    coef = c('s3','s4','s2','s1','s6','s5'))
anytable$AvelogFC = rowMeans(anytable[c('s3','s4','s2','s1','s6','s5')])
summary(anytable$adj.P.Val < 0.01) #much more conserative than previous tries
```

```
##    Mode   FALSE    TRUE
## logical   10502    7559
```

Donor tables

```
donortabs = list()
donors = c('s3','s4','s2','s1','s6','s5')
base_matrix = matrix(NA, dimnames = list(efit$genes$gene_name,donors),
                                    nrow = nrow(efit$genes), ncol=length(donors))
donorlists = list(adj.p.val = base_matrix, logFC = base_matrix,
                  genes= efit$genes )
for ( co in donors){
  onedonor = topTable(efit, number = Inf, adjust.method = "BH",
                    coef = co)
```

```
  donortabs[[co]] = onedonor
  donorlists$adj.p.val[,co] = onedonor$adj.P.Val[order(onedonor$Geneid)]
  donorlists$logFC[,co] = onedonor$logFC[order(onedonor$Geneid)]
}
summary(donortabs)
```

```
##    Length Class      Mode
## s3 13     data.frame list
## s4 13     data.frame list
## s2 13     data.frame list
## s1 13     data.frame list
## s6 13     data.frame list
## s5 13     data.frame list
```

```
ann_cols = c("Geneid", "gene_name", "description")
alltab = anytable[,!colnames(anytable) %in%c("ensembl_gene_id_version", "ID")]
for (d in donors){
  formatted = donortabs[[d]]
  formatted = formatted[,c(ann_cols,"P.Value","adj.P.Val","AveExpr")]
  alltab = merge(alltab, formatted, by=ann_cols,
                 suffixes = c("",paste(".", d, sep="")))
  alltab = alltab[!duplicated(alltab$gene_name),] #remove duplicates because they tend to propogate
}
head(alltab); nrow(alltab)
```

```
##            Geneid gene_name
## 1 ENSG00000000003    TSPAN6
## 2 ENSG00000000005      TNMD
## 3 ENSG00000000419      DPM1
## 4 ENSG00000000457     SCYL3
## 5 ENSG00000000460  C1orf112
## 6 ENSG00000000938       FGR
##                                                    description Length
## 1                                                tetraspanin 6    4536
## 2                                                  tenomodulin    1476
## 3 dolichyl-phosphate mannosyltransferase subunit 1, catalytic    1207
## 4                                   SCY1 like pseudokinase 3    6883
## 5                          chromosome 1 open reading frame 112    5970
## 6              FGR proto-oncogene, Src family tyrosine kinase    3382
##     gene_biotype          s4          s3         s2          s1         s6
## 1 protein_coding  0.37343465  0.57278654  0.8603088  0.49422154  0.6439646
## 2 protein_coding  1.81772689  2.58660597 -0.2720670  2.38787369  1.9785258
## 3 protein_coding  0.06689829 -0.06185359  0.1194288 -0.11790050  0.1710766
## 4 protein_coding  0.32546768  0.49896973  0.5484816  0.39439050  0.3469994
## 5 protein_coding -0.17866146  0.13017667  0.3071896 -0.05290602 -0.2082301
## 6 protein_coding  3.24138536  2.88724528  2.3497110  1.91496687  2.7442577
##            s5  AveExpr          F      P.Value    adj.P.Val   AvelogFC
## 1  0.56429720 5.212308 31.7140824 2.969288e-11 7.806158e-10  0.5848355
## 2  1.94504039 1.961126 20.9163666 3.809066e-09 5.434087e-08  1.7406176
## 3 -0.01815705 5.314282  0.4712428 8.237264e-01 8.380174e-01  0.0265821
## 4  0.35682721 3.626998 10.0207970 6.300732e-06 3.633461e-05  0.4118560
## 5 -0.36021700 1.367959  1.3456734 2.705375e-01 3.177382e-01 -0.0604414
```

```
## 6  2.88754674 2.951839 95.9233119 2.476214e-17 5.025045e-15  2.6708521
##     P.Value.s3 adj.P.Val.s3 AveExpr.s3   P.Value.s4 adj.P.Val.s4 AveExpr.s4
## 1 1.180377e-05 2.253571e-04   5.212308 1.164889e-02 4.171991e-02   5.212308
## 2 1.956120e-06 5.218110e-05   1.961126 1.864017e-03 1.155716e-02   1.961126
## 3 6.755915e-01 8.196868e-01   5.314282 7.245401e-01 8.149446e-01   5.314282
## 4 8.348144e-04 6.877477e-03   3.626998 6.456487e-02 1.441980e-01   3.626998
## 5 5.128644e-01 7.062247e-01   1.367959 4.826271e-01 6.174632e-01   1.367959
## 6 3.418267e-11 6.015586e-09   2.951839 1.506080e-10 5.681113e-08   2.951839
##     P.Value.s2 adj.P.Val.s2 AveExpr.s2   P.Value.s1 adj.P.Val.s1 AveExpr.s1
## 1 1.031683e-08 4.254162e-07   5.212308 7.423497e-05 2.280200e-03   5.212308
## 2 5.374467e-01 6.575098e-01   1.961126 4.606466e-06 2.307037e-04   1.961126
## 3 4.196868e-01 5.530398e-01   5.314282 4.216597e-01 7.813318e-01   5.314282
## 4 3.317889e-04 2.112990e-03   3.626998 5.556707e-03 6.606388e-02   3.626998
## 5 1.314679e-01 2.355830e-01   1.367959 7.868527e-01 9.473852e-01   1.367959
## 6 3.346207e-09 1.798686e-07   2.951839 8.493511e-08 8.063249e-06   2.951839
##     P.Value.s6 adj.P.Val.s6 AveExpr.s6   P.Value.s5 adj.P.Val.s5 AveExpr.s5
## 1 5.072093e-06 1.086679e-04   5.212308 1.989400e-05 3.539955e-04   5.212308
## 2 1.584704e-04 1.496932e-03   1.961126 1.460798e-04 1.739187e-03   1.961126
## 3 2.804584e-01 4.204665e-01   5.314282 9.043584e-01 9.430495e-01   5.314282
## 4 2.076997e-02 6.094661e-02   3.626998 1.359659e-02 5.250544e-02   3.626998
## 5 3.288567e-01 4.715745e-01   1.367959 8.133352e-02 1.835518e-01   1.367959
## 6 2.754188e-10 5.292092e-08   2.951839 6.097364e-11 1.596007e-08   2.951839
```

```
## [1] 18043
```

```
summary(rowSums(alltab[,grep("adj.P.Val\\.s", colnames(alltab))] < 0.05) == 6)
```

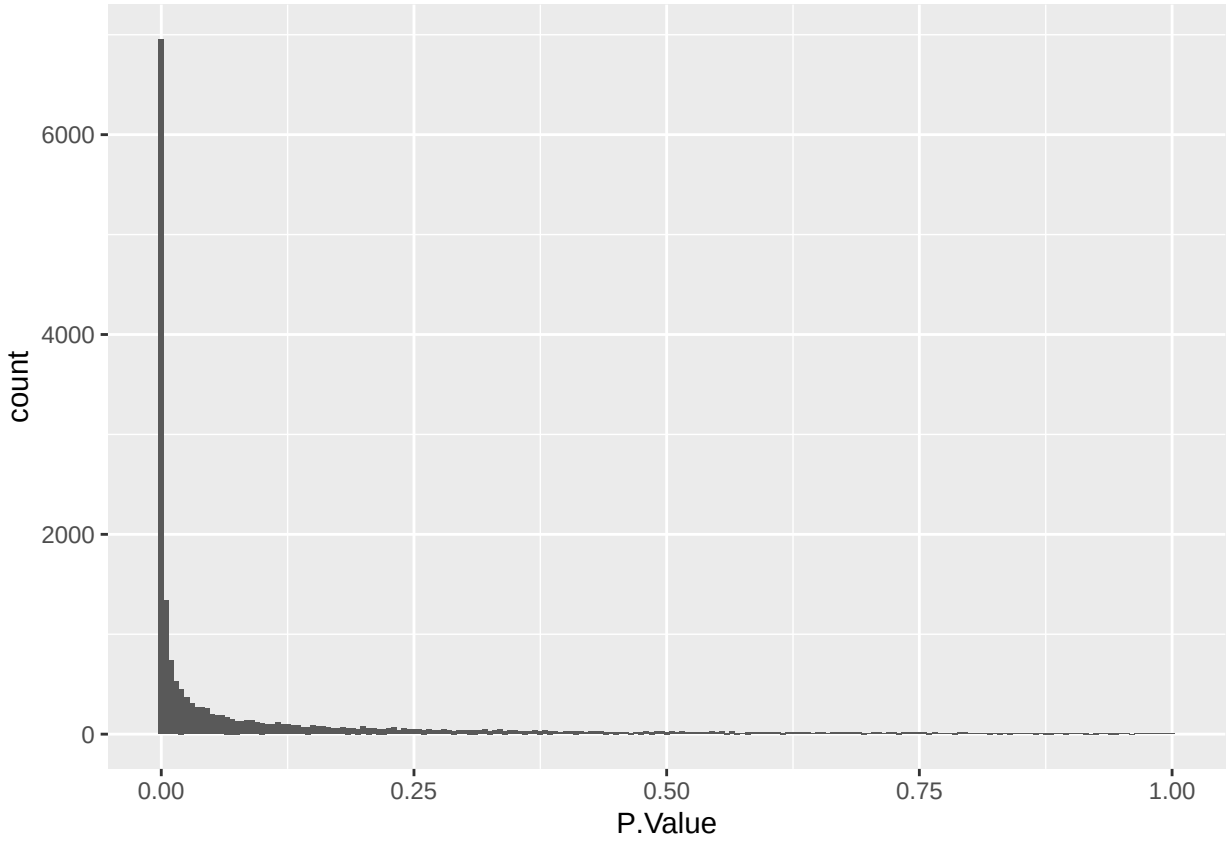
```
##    Mode   FALSE    TRUE
## logical   17190     853
```

We get a different number of significant genes than using decideTests because in order to extract the p.adjust for each contrast we have to forgo the p-value adjustment across contrasts.

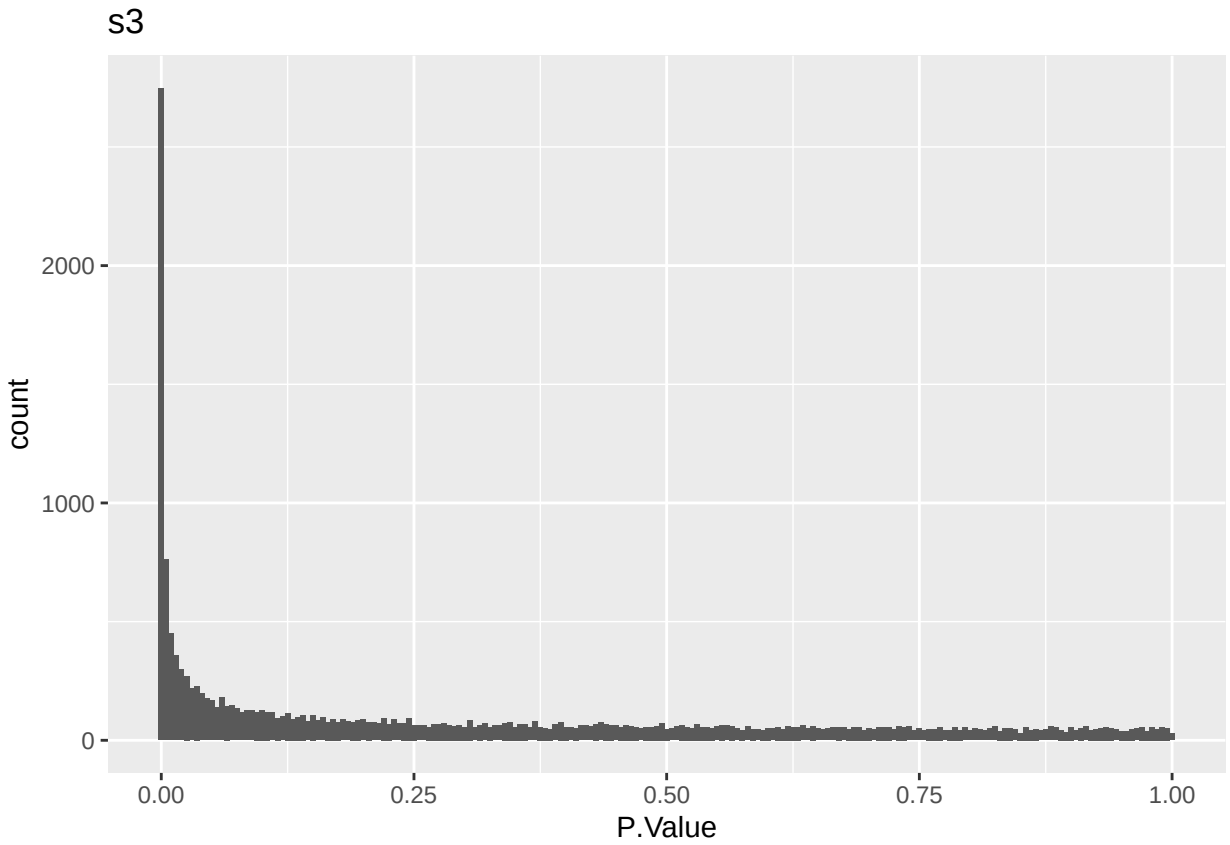
```
logfc_cols = grep("^s[1-6]$", colnames(alltab))
colnames(alltab)[logfc_cols] = paste0("logFC.", colnames(alltab)[logfc_cols])
alltab = rename(alltab, all.donors.P.Value = P.Value,
                all.donors.adj.P.Val = adj.P.Val,
                all.donors.AvelogFC = AvelogFC)


write.table(alltab, here("03limma/any_and_all_donor_DGE.tsv"),
            sep="\t",quote=F, row.names = F)
```

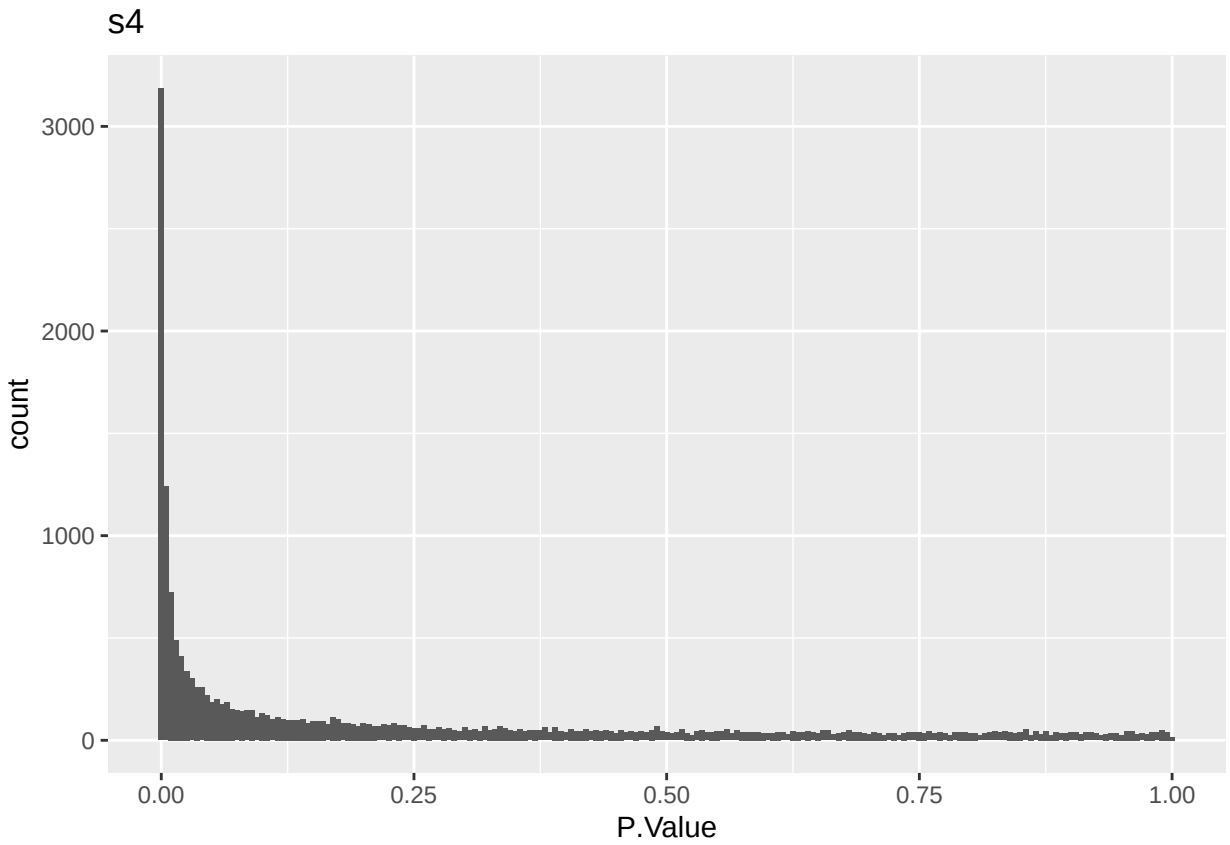
Pvalue histograms

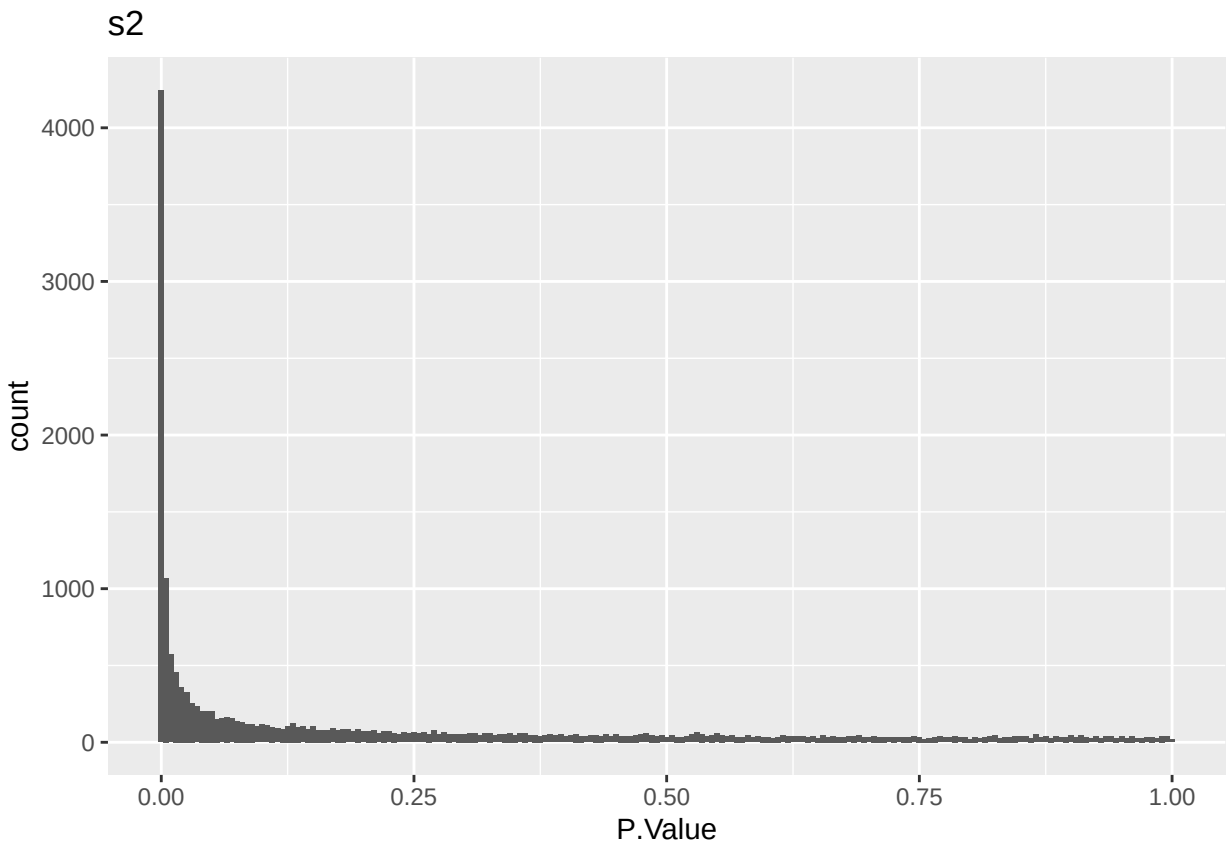
```
ggplot(anytable) + geom_histogram(aes(x=P.Value), binwidth = 0.005)
```

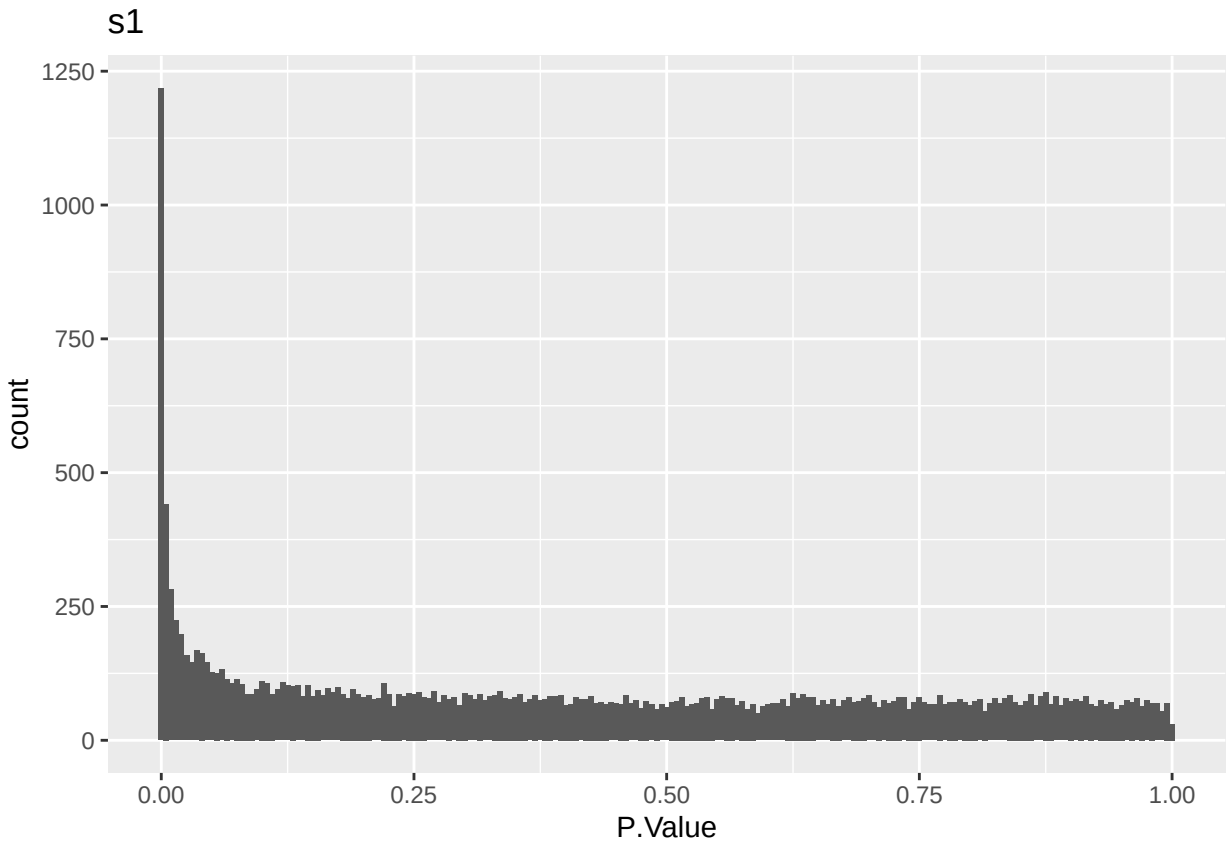


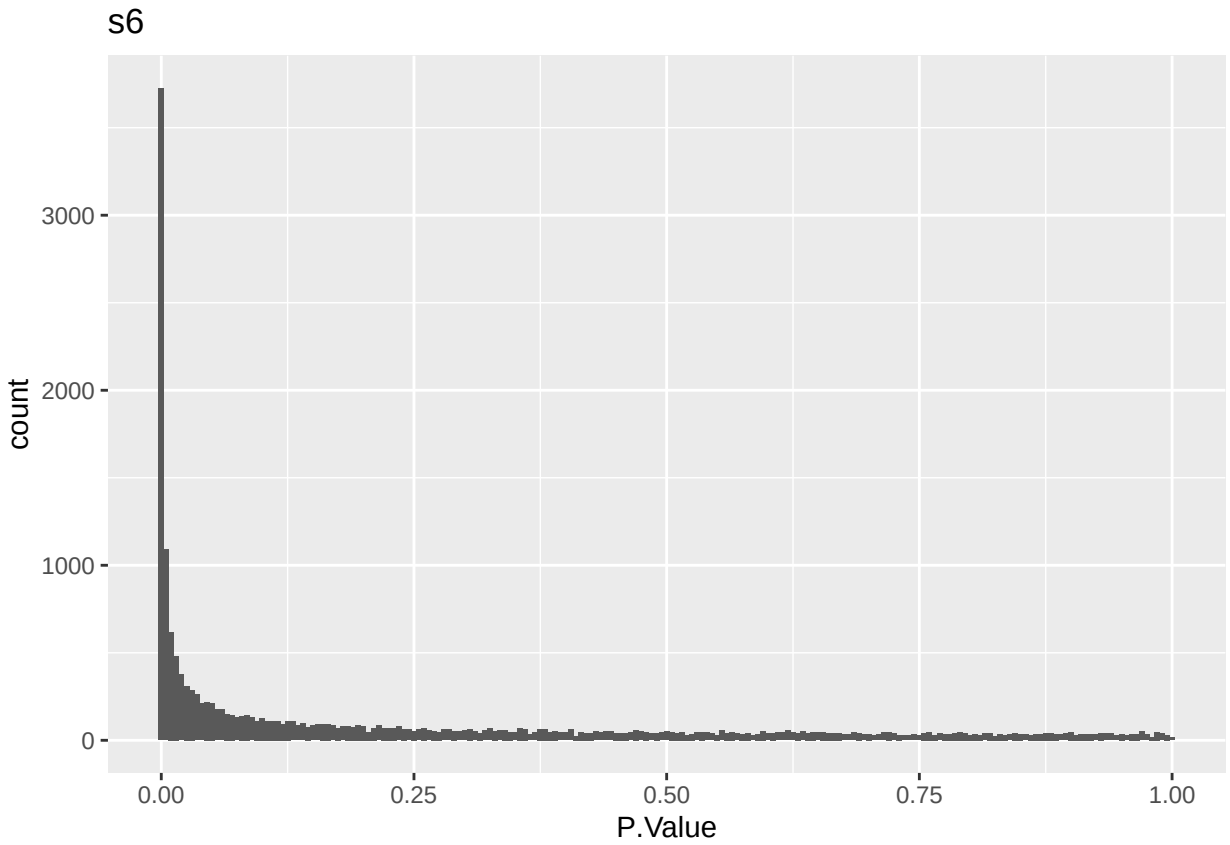

```
for (d in donors){
  p = ggplot(donortabs[[d]]) + geom_histogram(aes(x=P.Value), binwidth = 0.005) + ggtitle(d)
  print(p)
}
```

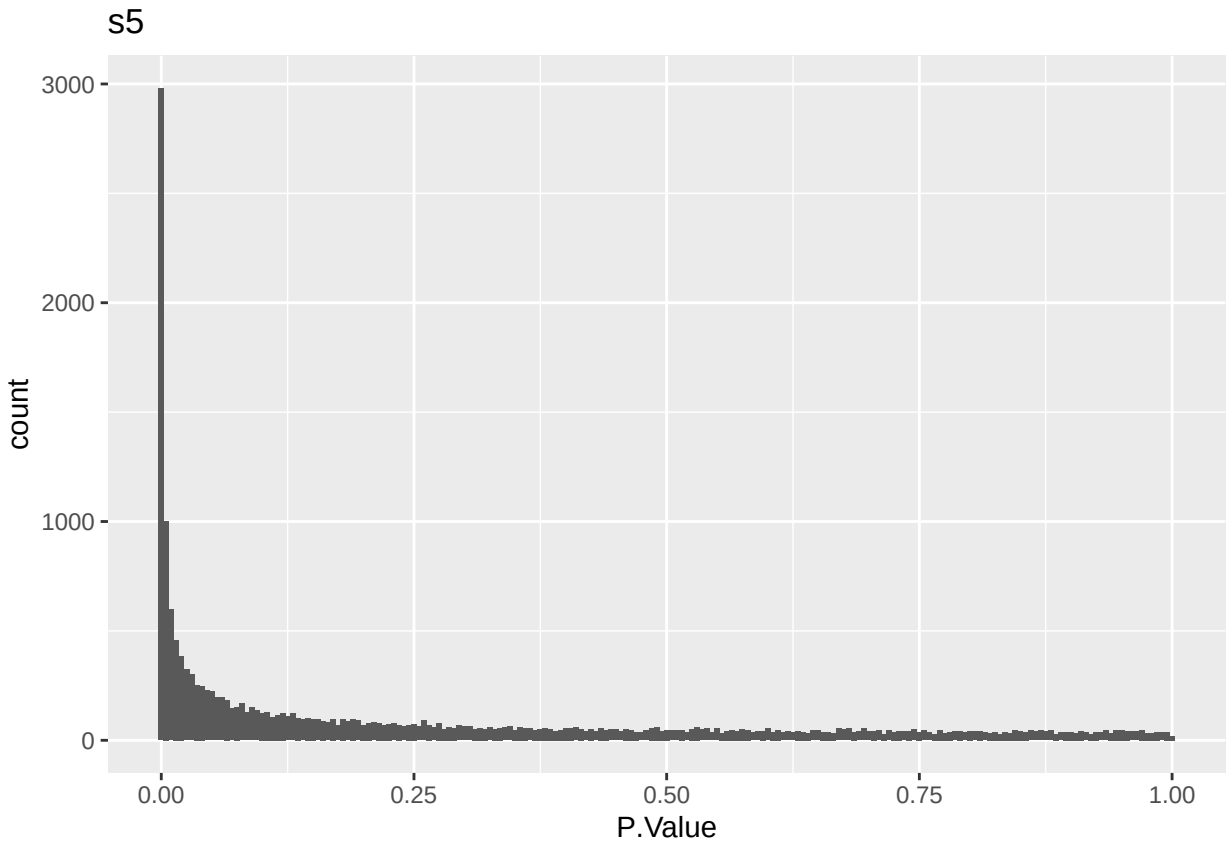

s3
2000
1000
0
count
0.00
0.25
0.50
0.75
1.00
P.Value


s4
3000
2000
1000
0
count
0.00
0.25
0.50
0.75
1.00
P.Value


s2
4000
3000
2000
1000
0
count
0.00
0.25
0.50
0.75
1.00
P.Value


s1
1250
1000
750
500
250
0
count
0.00
0.25
0.50
0.75
1.00
P.Value

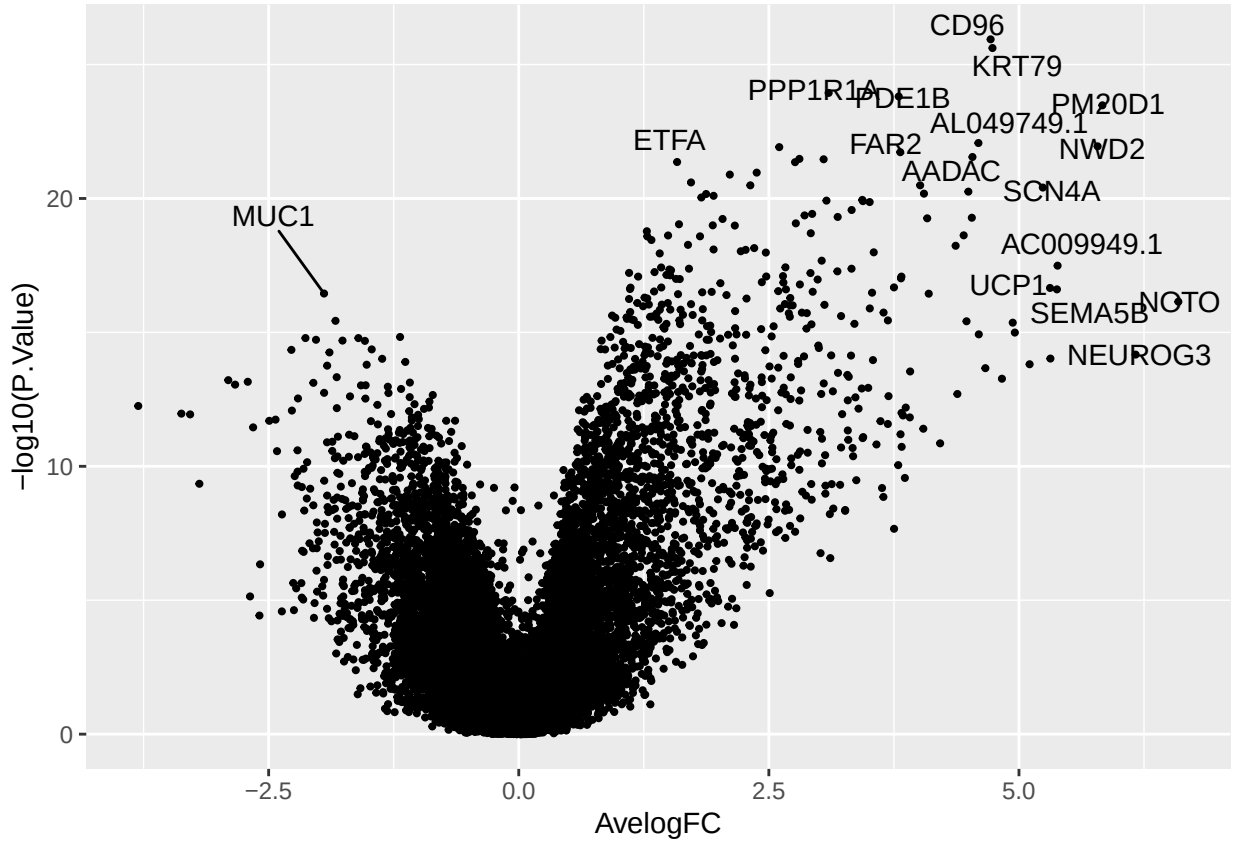

s6
3000
2000
1000
0
count
0.00
0.25
0.50
0.75
1.00
P.Value



Volcano Plots

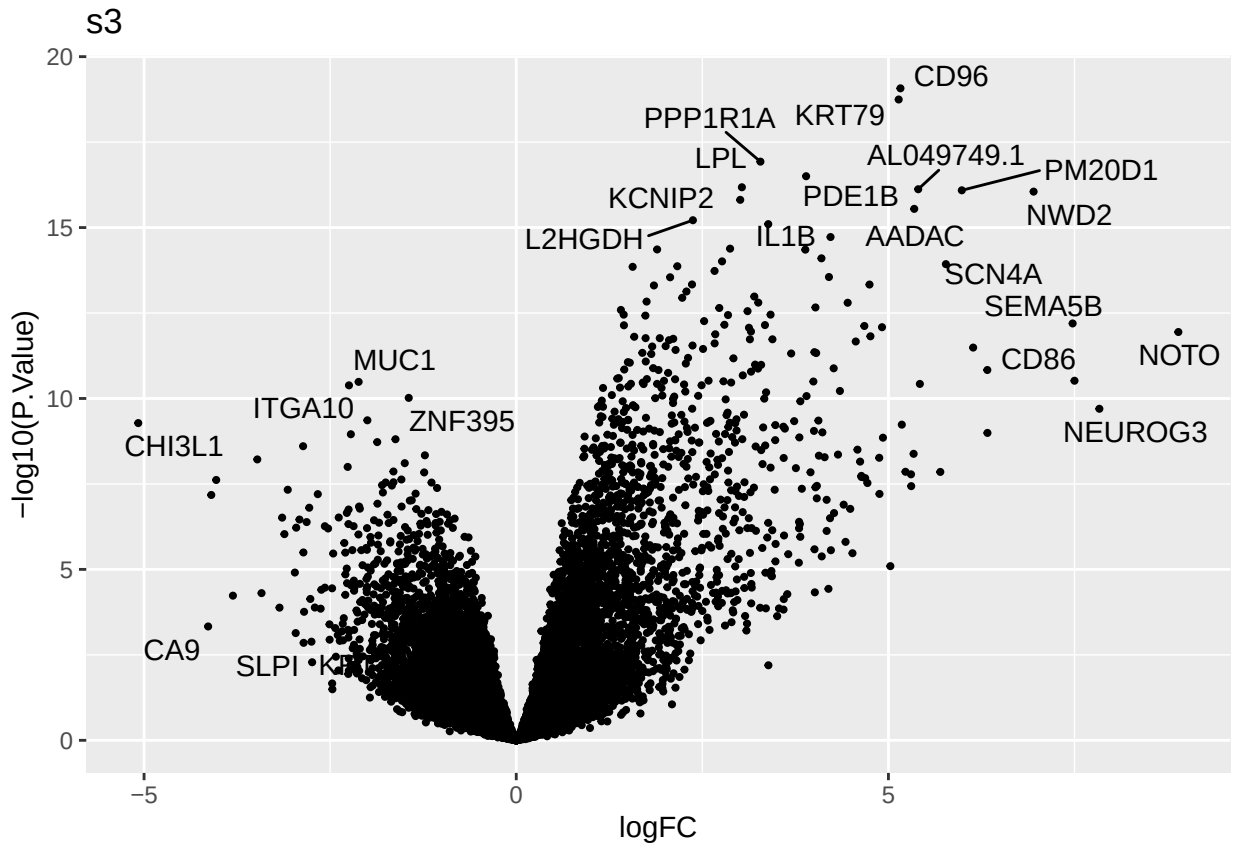
```
ggplot(anytable, aes(x=AvelogFC, y=-log10(P.Value))) +
  geom_point(show.legend = F, size=0.75) +
    geom_text_repel(data=anytable[anytable$AvelogFC > 5 |
                                    anytable$AvelogFC < -2.75 |
                                    anytable$adj.P.Val < 0.0001,],
                    aes(label=gene_name))
```

```
## Warning: ggrepel: 3599 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

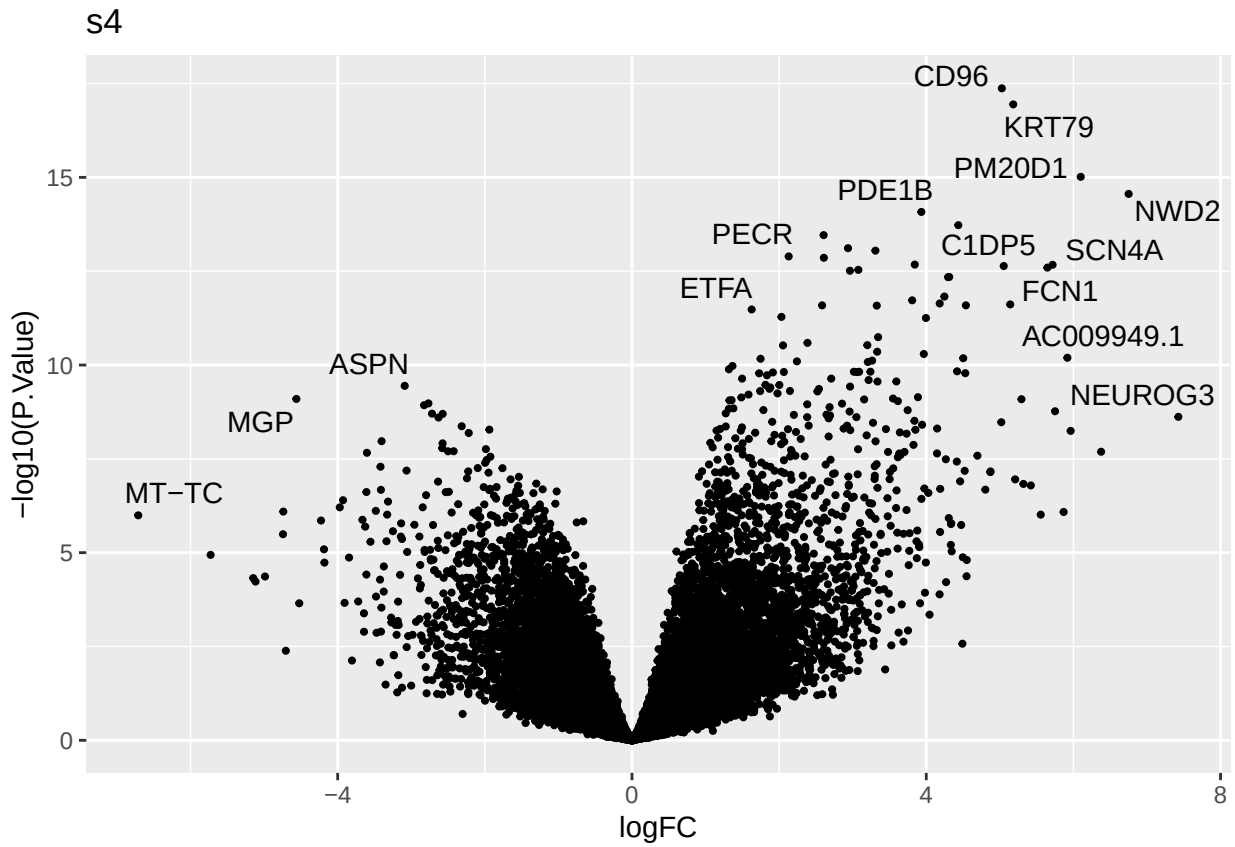



```
for ( d in donors){
    tab =donortabs[[d]]
p = ggplot(tab, aes(x=logFC, y=-log10(P.Value))) +
  geom_point(show.legend = F, size=0.75) +  ggtitle(d) +
    geom_text_repel(data=tab[tab$logFC > 5 |
                               tab$logFC < -2.75 |
                               tab$adj.P.Val < 0.0001,],
                    aes(label=gene_name))

print(p)
}
```

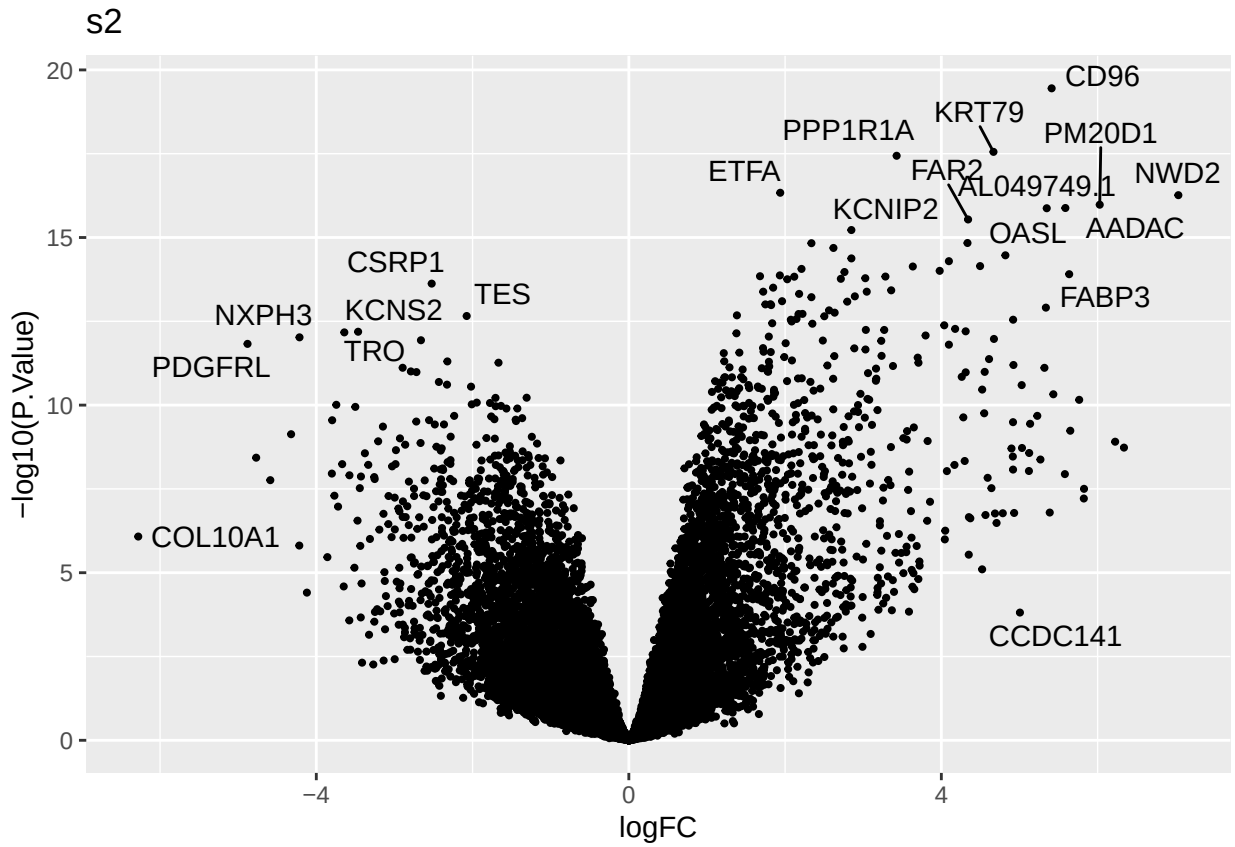
```
## Warning: ggrepel: 767 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



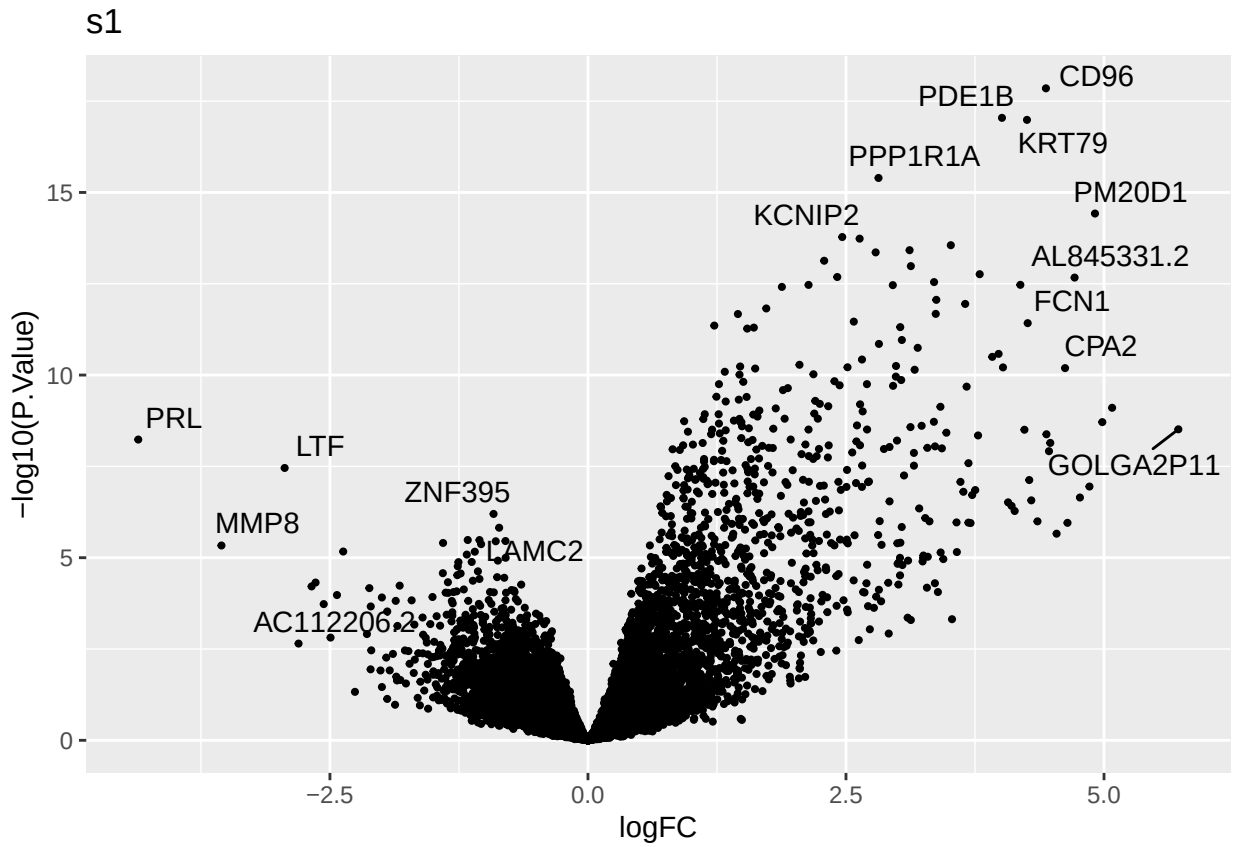

```
## Warning: ggrepel: 581 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



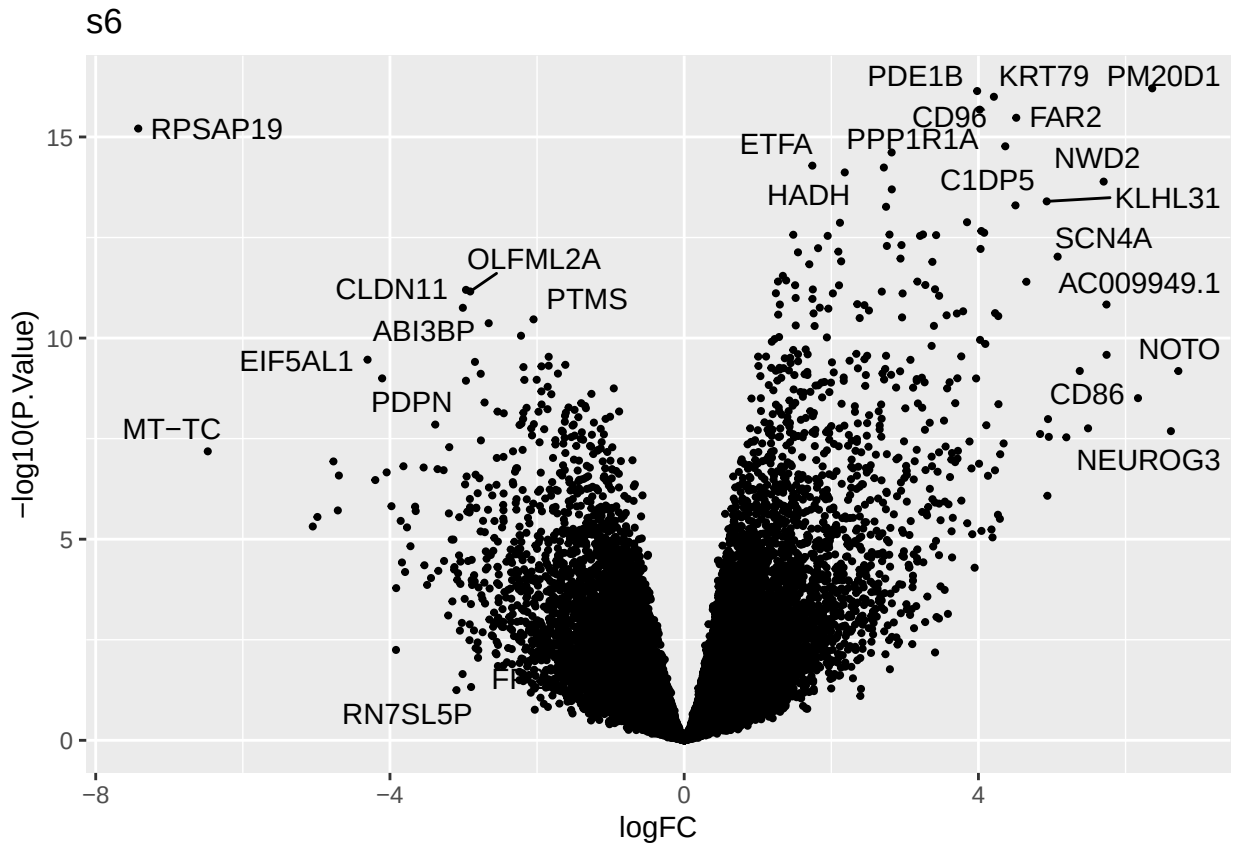

```
## Warning: ggrepel: 1521 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



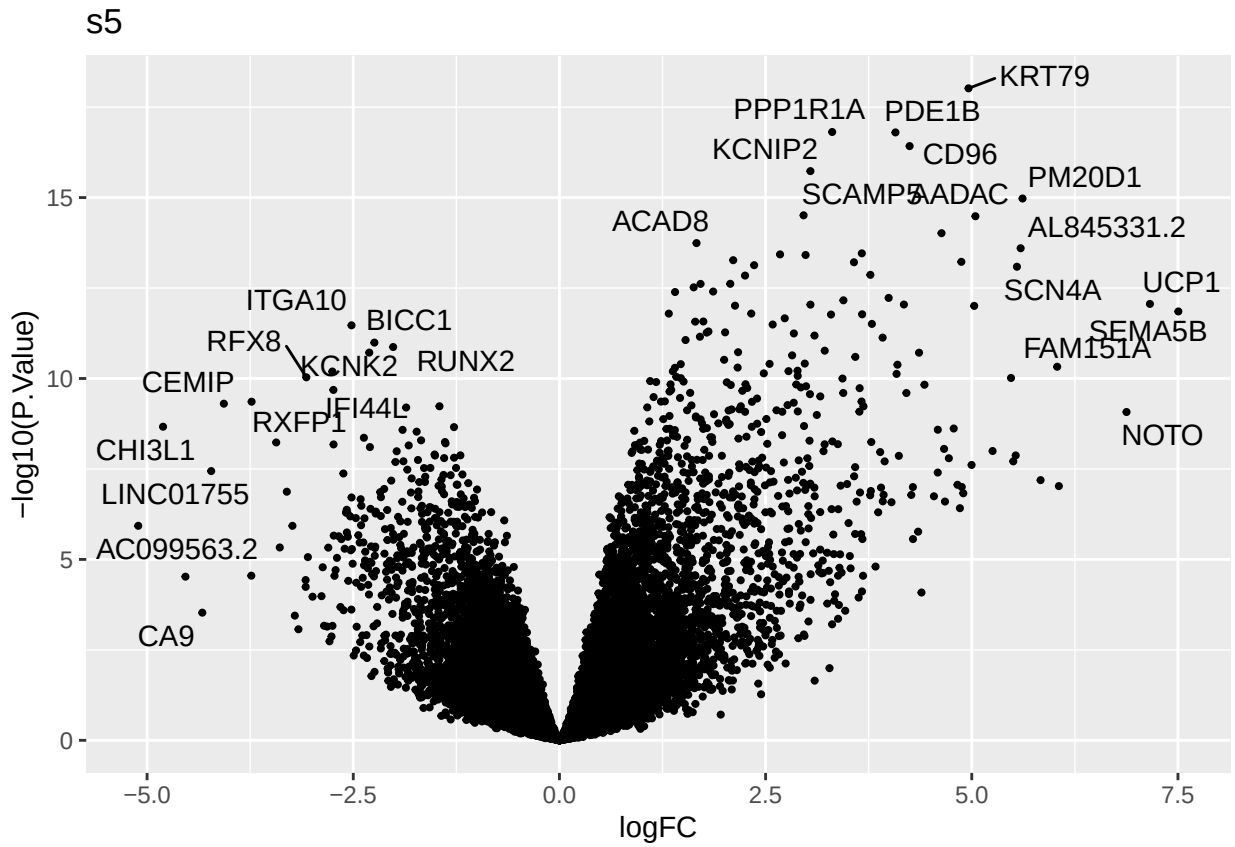

```
## Warning: ggrepel: 290 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```




```
## Warning: ggrepel: 852 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```




```
## Warning: ggrepel: 725 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



GO

```
molsig <- clusterProfiler::read.gmt(here("annotations/msigdb.v2023.1.Hs.symbols.gmt"))
head(molsig); nrow(molsig)
```

```
##      term       gene
## 1 chr1p11  LINC02798
## 2 chr1p11    MTIF2P1
## 3 chr1p11    SRGAP2C
## 4 chr1p11 SRGAP2-AS1
## 5 chr1p11  LINC01691
## 6 chr1p11     NBPF26

## [1] 3961711
```

```
prefixes = c("HALLMARK", "KEGG","REACTOME", "WP", "GOBP", "GOCC","GOMF")
colnames(molsig) = c("term","gene")
some.molsig = molsig[gsub("_.*","", molsig$term) %in% prefixes,]
some.molsig$term = factor(some.molsig$term)
table(gsub("_.*","", some.molsig$term))
```

```
##
##     GOBP     GOCC     GOMF HALLMARK     KEGG REACTOME       WP
##   642656    98915   108833     7322    12796    92769    31635
```

```
rm(molsig)
shorten = function(ont) {
  abbreviate(gsub("_"," ", tolower(ont)),minlength=40, dot=T, named = F)
}
```

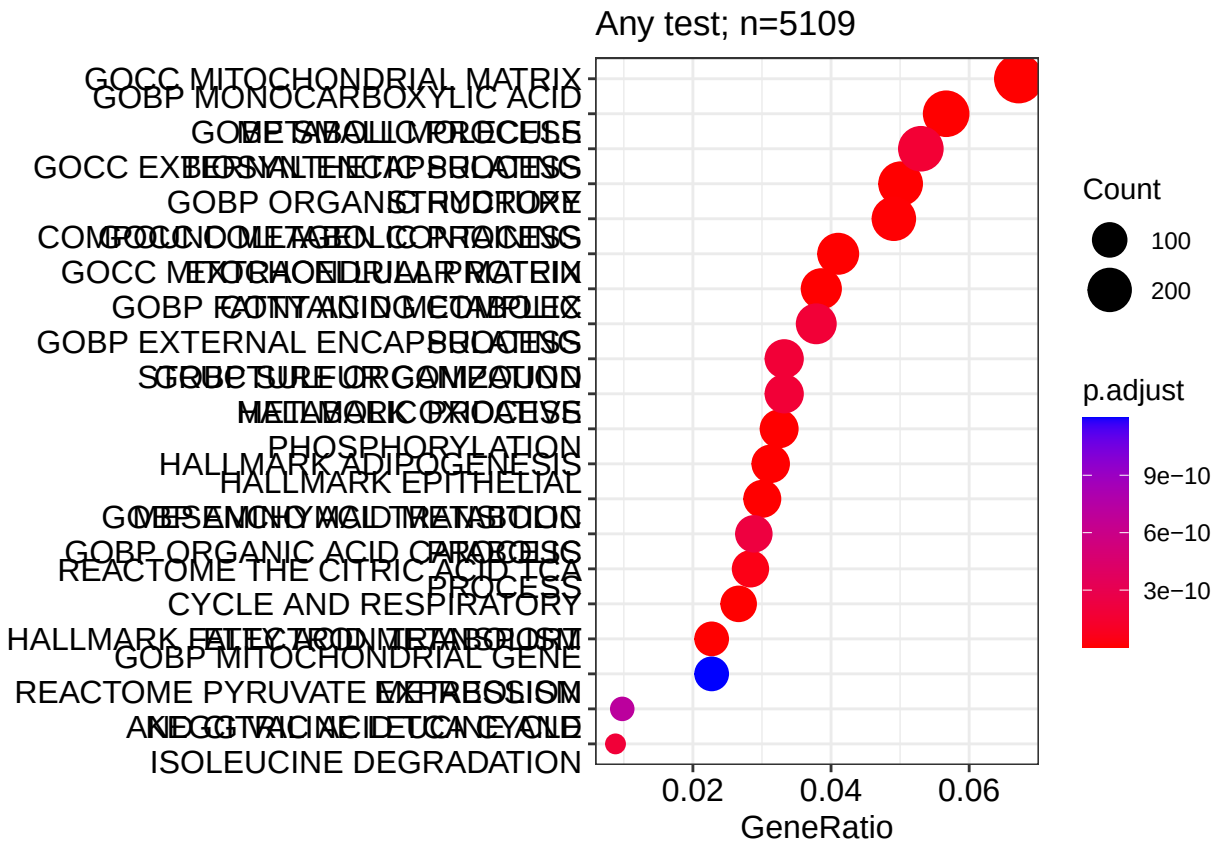
GO: Genes DE in ANY donor (n=5109)

```
genes= anytable$gene_name[anytable$adj.P.Val < 0.001]
any = enricher(genes,
               universe = anytable$gene_name,
            TERM2GENE = some.molsig)
head(any)
```

```
##
## GOCC_MITOCHONDRIAL_MATRIX
## HALLMARK_OXIDATIVE_PHOSPHORYLATION
## HALLMARK_ADIPOGENESIS
## HALLMARK_EPITHELIAL_MESENCHYMAL_TRANSITION                                                    HALL
## GOCC_MITOCHONDRIAL_PROTEIN_CONTAINING_COMPLEX                                                GOCC_MI
## REACTOME_THE_CITRIC_ACID_TCA_CYCLE_AND_RESPIRATORY_ELECTRON_TRANSPORT REACTOME_THE_CITRIC_ACID_TCA_CY
##
## GOCC_MITOCHONDRIAL_MATRIX
## HALLMARK_OXIDATIVE_PHOSPHORYLATION
## HALLMARK_ADIPOGENESIS
## HALLMARK_EPITHELIAL_MESENCHYMAL_TRANSITION                                                    HALL
## GOCC_MITOCHONDRIAL_PROTEIN_CONTAINING_COMPLEX                                                GOCC_MI
## REACTOME_THE_CITRIC_ACID_TCA_CYCLE_AND_RESPIRATORY_ELECTRON_TRANSPORT REACTOME_THE_CITRIC_ACID_TCA_CY
##                                                                    GeneRatio
## GOCC_MITOCHONDRIAL_MATRIX                                           275/4094
## HALLMARK_OXIDATIVE_PHOSPHORYLATION                                  133/4094
## HALLMARK_ADIPOGENESIS                                               128/4094
## HALLMARK_EPITHELIAL_MESENCHYMAL_TRANSITION                          123/4094
## GOCC_MITOCHONDRIAL_PROTEIN_CONTAINING_COMPLEX                       158/4094
## REACTOME_THE_CITRIC_ACID_TCA_CYCLE_AND_RESPIRATORY_ELECTRON_TRANSPORT  109/4094
##                                                                     BgRatio
## GOCC_MITOCHONDRIAL_MATRIX                                         450/13071
## HALLMARK_OXIDATIVE_PHOSPHORYLATION                                200/13071
## HALLMARK_ADIPOGENESIS                                             197/13071
## HALLMARK_EPITHELIAL_MESENCHYMAL_TRANSITION                        197/13071
## GOCC_MITOCHONDRIAL_PROTEIN_CONTAINING_COMPLEX                     282/13071
## REACTOME_THE_CITRIC_ACID_TCA_CYCLE_AND_RESPIRATORY_ELECTRON_TRANSPORT 173/13071
##                                                                         pvalue
## GOCC_MITOCHONDRIAL_MATRIX                                         4.871789e-40
## HALLMARK_OXIDATIVE_PHOSPHORYLATION                                7.953976e-25
## HALLMARK_ADIPOGENESIS                                             1.464833e-22
## HALLMARK_EPITHELIAL_MESENCHYMAL_TRANSITION                        1.419945e-19
## GOCC_MITOCHONDRIAL_PROTEIN_CONTAINING_COMPLEX                     3.907187e-18
## REACTOME_THE_CITRIC_ACID_TCA_CYCLE_AND_RESPIRATORY_ELECTRON_TRANSPORT 6.394579e-18
##                                                                       p.adjust
## GOCC_MITOCHONDRIAL_MATRIX                                         3.964174e-36
## HALLMARK_OXIDATIVE_PHOSPHORYLATION                                3.236075e-21
```

```
## HALLMARK_ADIPOGENESIS                                                 3.973115e-19
## HALLMARK_EPITHELIAL_MESENCHYMAL_TRANSITION                            2.888522e-16
## GOCC_MITOCHONDRIAL_PROTEIN_CONTAINING_COMPLEX                         6.358555e-15
## REACTOME_THE_CITRIC_ACID_TCA_CYCLE_AND_RESPIRATORY_ELECTRON_TRANSPORT 8.672115e-15
##                                                                             qvalue
## GOCC_MITOCHONDRIAL_MATRIX                                             3.307175e-36
## HALLMARK_OXIDATIVE_PHOSPHORYLATION                                    2.699747e-21
## HALLMARK_ADIPOGENESIS                                                 3.314634e-19
## HALLMARK_EPITHELIAL_MESENCHYMAL_TRANSITION                            2.409796e-16
## GOCC_MITOCHONDRIAL_PROTEIN_CONTAINING_COMPLEX                         5.304726e-15
## REACTOME_THE_CITRIC_ACID_TCA_CYCLE_AND_RESPIRATORY_ELECTRON_TRANSPORT 7.234849e-15
##
## GOCC_MITOCHONDRIAL_MATRIX                                             ETFA/ACAA2/HADH/MLYCD/ETFDH/ALI
## HALLMARK_OXIDATIVE_PHOSPHORYLATION
## HALLMARK_ADIPOGENESIS
## HALLMARK_EPITHELIAL_MESENCHYMAL_TRANSITION
## GOCC_MITOCHONDRIAL_PROTEIN_CONTAINING_COMPLEX
## REACTOME_THE_CITRIC_ACID_TCA_CYCLE_AND_RESPIRATORY_ELECTRON_TRANSPORT
##                                                                       Count
## GOCC_MITOCHONDRIAL_MATRIX                                               275
## HALLMARK_OXIDATIVE_PHOSPHORYLATION                                      133
## HALLMARK_ADIPOGENESIS                                                   128
## HALLMARK_EPITHELIAL_MESENCHYMAL_TRANSITION                              123
## GOCC_MITOCHONDRIAL_PROTEIN_CONTAINING_COMPLEX                           158
## REACTOME_THE_CITRIC_ACID_TCA_CYCLE_AND_RESPIRATORY_ELECTRON_TRANSPORT   109
```

```
dotplot(any, showCategory = 20) + ggtitle (paste0("Any test; n=",  length(genes)))
```




```
head(results)
```

```
## TestResults matrix
##                  Contrasts
##                   s4 s3 s2 s1 s6 s5
##   ENSG00000000003  1  1  1  1  1  1
##   ENSG00000000005  1  1  0  1  1  1
##   ENSG00000000419  0  0  0  0  0  0
##   ENSG00000000457  0  1  1  0  0  0
##   ENSG00000000460  0  0  0  0  0  0
##   ENSG00000000938  1  1  1  1  1  1
```

```
decided = results[,donors]
rownames(decided) = filt$genes$gene_name[order(rownames(decided))]
head(decided); nrow(decided)
```

```
## TestResults matrix
##           Contrasts
##            s3 s4 s2 s1 s6 s5
##   TSPAN6    1  1  1  1  1  1
##   TNMD      1  1  0  1  1  1
##   DPM1      0  0  0  0  0  0
##   SCYL3     1  0  1  0  0  0
##   C1orf112  0  0  0  0  0  0
##   FGR       1  1  1  1  1  1
```

```
## [1] 18061
```

```
ddf = as.data.frame(decided)
ddf$str = paste0(ddf$s3,ddf$s4,ddf$s2,ddf$s1,ddf$s6,ddf$s5)

contrasts = list(beige_all =  rownames(ddf)[grep("1{6}", ddf$str)],
                 s2_beige = rownames(ddf)[grep("001000", ddf$str)],
                 s4_6_beige = rownames(ddf)[grep("010010", ddf$str)],
                 s4_beige = rownames(ddf)[grep("010000", ddf$str)],
                 s6_beige = rownames(ddf)[grep("000010", ddf$str)],
                 s5_beige = rownames(ddf)[grep("000001", ddf$str)],

                 s4_6_5_beige = rownames(ddf)[grep("010011", ddf$str)],
                 s3_s2_beige =rownames(ddf)[grep("101000", ddf$str)],

                 s3_beige = rownames(ddf)[grep("100000", ddf$str)],
                 beige_not9 = rownames(ddf)[grep("111011", ddf$str)],

                 white_all = rownames(ddf)[grep("(-1){6}", ddf$str)],
                 s2_white = rownames(ddf)[grep("00-1000", ddf$str)],
                 s4_6_white = rownames(ddf)[grep("0-100-10", ddf$str)],
                 s4_white = rownames(ddf)[grep("0-10000", ddf$str)],
                 s6_white = rownames(ddf)[grep("0000-10", ddf$str)],
                 s5_white =  rownames(ddf)[grep("00000-1", ddf$str)],

                 s4_6_5_white = rownames(ddf)[grep("0-100-1-1", ddf$str)],
                 s3_s2_white =rownames(ddf)[grep("-10-1000", ddf$str)],
                 white_not9 = rownames(ddf)[grep("-1-1-10-1-1", ddf$str)],
                 s3_white = rownames(ddf)[grep("-100000", ddf$str)]
)

str(contrasts)
```

```
## List of 20
##  $ beige_all   : chr [1:717] "TSPAN6" "FGR" "ICA1" "POLDIP2" ...
##  $ s2_beige    : chr [1:835] "CFH" "FUCA2" "AK2" "GCFC2" ...
##  $ s4_6_beige  : chr [1:583] "CYP51A1" "RHBDD2" "SPATA20" "CCDC124" ...
##  $ s4_beige    : chr [1:393] "LAS1L" "DVL2" "ABCB4" "RPUSD1" ...
##  $ s6_beige    : chr [1:324] "EEF1AKNMT" "BID" "GRAMD1B" "RIPOR3" ...
##  $ s5_beige    : chr [1:266] "WNT16" "RHOBTB2" "NUDCD3" "BAK1" ...
##  $ s4_6_5_beige: chr [1:307] "FAM214B" "PDK2" "NADK" "PSMC4" ...
##  $ s3_s2_beige : chr [1:241] "SCYL3" "MAP3K9" "AGPS" "GLRX2" ...
##  $ s3_beige    : chr [1:350] "CASP10" "SLC7A2" "CDC27" "SCIN" ...
##  $ beige_not9  : chr [1:197] "SLC22A16" "HCCS" "ST3GAL1" "POMT2" ...
##  $ white_all   : chr [1:134] "DCN" "LTF" "SLC7A14" "CLDN11" ...
##  $ s2_white    : chr [1:1318] "TMEM176A" "SARM1" "PLXND1" "ARHGAP33" ...
##  $ s4_6_white  : chr [1:467] "HOXA11" "RALA" "MATR3" "BCLAF1" ...
##  $ s4_white    : chr [1:388] "FBXL3" "ZFX" "FARP2" "PAFAH1B1" ...
##  $ s6_white    : chr [1:572] "KRIT1" "HSPB6" "IFRD1" "E2F2" ...
##  $ s5_white    : chr [1:390] "RBM6" "CROT" "AP2B1" "ST7L" ...
##  $ s4_6_5_white: chr [1:243] "RECQL" "RANBP9" "PGM3" "CLK1" ...
##  $ s3_s2_white : chr [1:231] "IFFO1" "MRC2" "PLAUR" "DNASE1L1" ...
##  $ white_not9  : chr [1:246] "ETV1" "RWDD2A" "GLT8D1" "HGF" ...
##  $ s3_white    : chr [1:150] "CASP10" "STAB1" "CDH1" "DTNBP1" ...
```

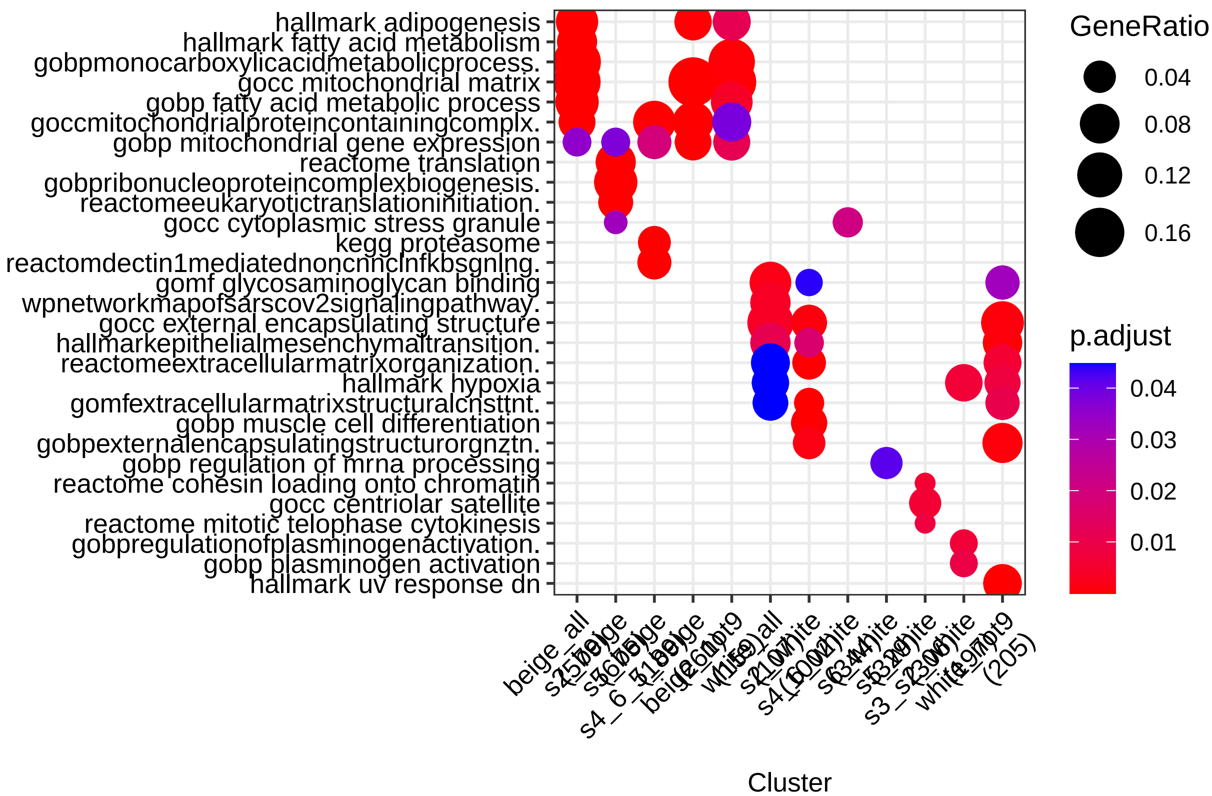
GO for combinations of donors

```
go_all = compareCluster(geneClusters = contrasts,
                        fun = "enricher",
                        TERM2GENE = some.molsig,
                        universe = anytable$gene_name)
head(go_all)
```

```
##     Cluster                                          ID
## 1 beige_all                      HALLMARK_ADIPOGENESIS
## 2 beige_all              HALLMARK_FATTY_ACID_METABOLISM
## 3 beige_all GOBP_MONOCARBOXYLIC_ACID_METABOLIC_PROCESS
## 4 beige_all                   GOCC_MITOCHONDRIAL_MATRIX
## 5 beige_all           GOBP_FATTY_ACID_METABOLIC_PROCESS
## 6 beige_all         GOBP_ORGANIC_ACID_CATABOLIC_PROCESS
##                                   Description GeneRatio   BgRatio       pvalue
## 1                      HALLMARK_ADIPOGENESIS    57/579 197/13071 2.920537e-31
## 2              HALLMARK_FATTY_ACID_METABOLISM    48/579 142/13071 6.681039e-30
## 3 GOBP_MONOCARBOXYLIC_ACID_METABOLIC_PROCESS    84/579 474/13071 8.146883e-29
## 4                   GOCC_MITOCHONDRIAL_MATRIX    81/579 450/13071 2.876342e-28
## 5           GOBP_FATTY_ACID_METABOLIC_PROCESS    62/579 303/13071 7.835987e-25
## 6         GOBP_ORGANIC_ACID_CATABOLIC_PROCESS    47/579 206/13071 4.138292e-21
##       p.adjust       qvalue
## 1 1.590817e-27 1.445820e-27
## 2 1.819581e-26 1.653733e-26
## 3 1.479202e-25 1.344379e-25
## 4 3.916859e-25 3.559852e-25
## 5 8.536524e-22 7.758452e-22
## 6 3.756880e-18 3.414454e-18
##
## 1
## 2
## 3 PDK4/ACSM3/ADIPOR2/ABHD5/NR1H3/MSMO1/ACAA1/ME1/ACSL4/MGAT4A/ALDH3A2/ACADVL/ACAT1/ACACB/MCCC1/SLC1A3
## 4        POLDIP2/PDK4/SLC25A5/ACSM3/LARS2/ALAS1/MRPS35/CS/ACADVL/ACAT1/MCCC1/BCKDHB/HADHA/DLD/ACO2/MI
## 5
## 6
##   Count
## 1    57
## 2    48
## 3    84
## 4    81
## 5    62
## 6    47
```

```
p = dotplot(go_all, showCategory =3, font.size=10) +  theme(axis.text.x = element_text(angle=45, hjust=
p + scale_y_discrete(labels=shorten(levels(p$data$Description)))
```

```
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```



GO beige in all donors

Using the significance table rather than the results test because its more reproducible.

```
beige_all = alltab[rowSums(alltab[grep("adj.P.Val.s[1-6]", colnames(alltab))] < 0.05 ) == 6,]
nrow(beige_all)
```

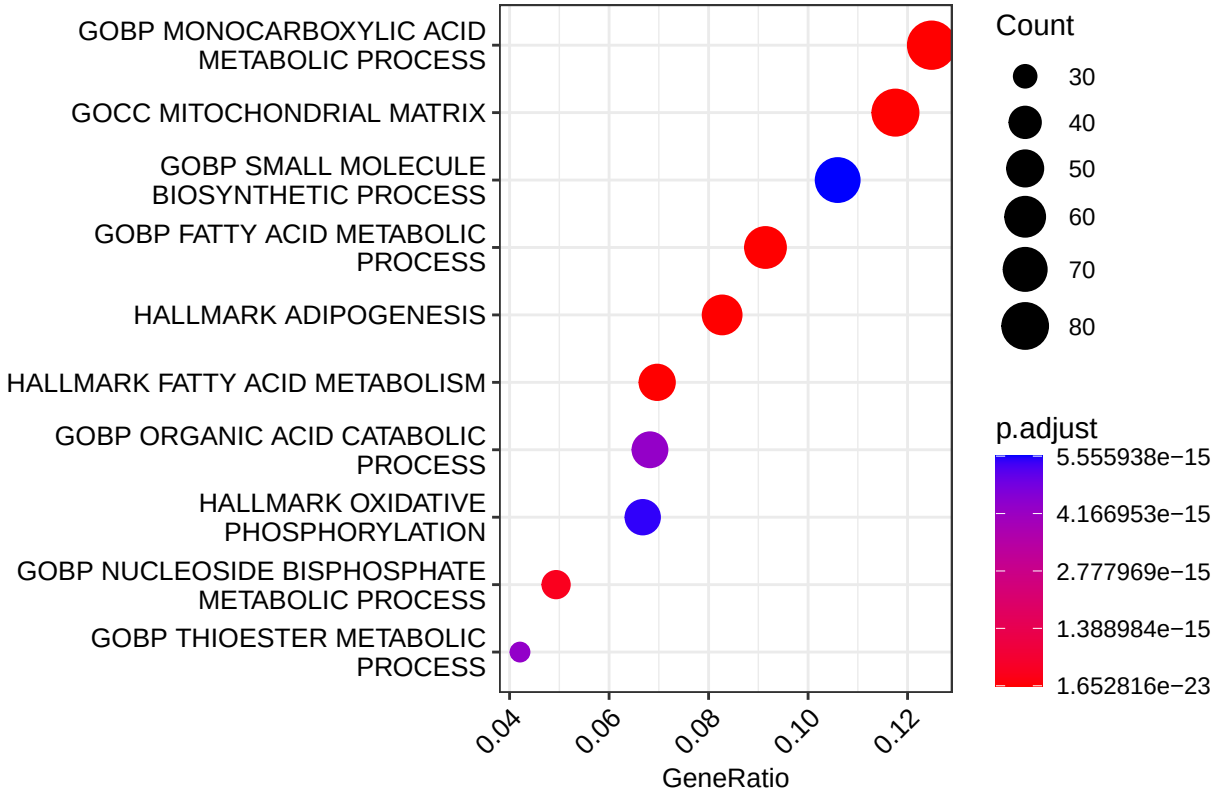
```
## [1] 853
```

```
go_beige_all = enricher(beige_all$gene_name,
                        TERM2GENE = some.molsig,
                        universe = anytable$gene_name)
head(go_beige_all)
```

```
##                                                                                           ID
## HALLMARK_ADIPOGENESIS                                                  HALLMARK_ADIPOGENESIS
## HALLMARK_FATTY_ACID_METABOLISM                                HALLMARK_FATTY_ACID_METABOLISM
## GOBP_MONOCARBOXYLIC_ACID_METABOLIC_PROCESS        GOBP_MONOCARBOXYLIC_ACID_METABOLIC_PROCESS
## GOCC_MITOCHONDRIAL_MATRIX                                          GOCC_MITOCHONDRIAL_MATRIX
## GOBP_FATTY_ACID_METABOLIC_PROCESS                          GOBP_FATTY_ACID_METABOLIC_PROCESS
## GOBP_NUCLEOSIDE_BISPHOSPHATE_METABOLIC_PROCESS GOBP_NUCLEOSIDE_BISPHOSPHATE_METABOLIC_PROCESS
##                                                                                  Description
## HALLMARK_ADIPOGENESIS                                                  HALLMARK_ADIPOGENESIS
## HALLMARK_FATTY_ACID_METABOLISM                                HALLMARK_FATTY_ACID_METABOLISM
```

```
## GOBP_MONOCARBOXYLIC_ACID_METABOLIC_PROCESS         GOBP_MONOCARBOXYLIC_ACID_METABOLIC_PROCESS
## GOCC_MITOCHONDRIAL_MATRIX                                       GOCC_MITOCHONDRIAL_MATRIX
## GOBP_FATTY_ACID_METABOLIC_PROCESS                           GOBP_FATTY_ACID_METABOLIC_PROCESS
## GOBP_NUCLEOSIDE_BISPHOSPHATE_METABOLIC_PROCESS GOBP_NUCLEOSIDE_BISPHOSPHATE_METABOLIC_PROCESS
##                                                GeneRatio   BgRatio       pvalue
## HALLMARK_ADIPOGENESIS                             57/689 197/13071 2.785333e-27
## HALLMARK_FATTY_ACID_METABOLISM                    48/689 142/13071 1.734421e-26
## GOBP_MONOCARBOXYLIC_ACID_METABOLIC_PROCESS        86/689 474/13071 8.604599e-25
## GOCC_MITOCHONDRIAL_MATRIX                         81/689 450/13071 3.861167e-23
## GOBP_FATTY_ACID_METABOLIC_PROCESS                 63/689 303/13071 1.575969e-21
## GOBP_NUCLEOSIDE_BISPHOSPHATE_METABOLIC_PROCESS    34/689 100/13071 3.407372e-19
##                                                  p.adjust       qvalue
## HALLMARK_ADIPOGENESIS                          1.652816e-23 1.495577e-23
## HALLMARK_FATTY_ACID_METABOLISM                 5.146027e-23 4.656464e-23
## GOBP_MONOCARBOXYLIC_ACID_METABOLIC_PROCESS     1.701990e-21 1.540072e-21
## GOCC_MITOCHONDRIAL_MATRIX                      5.728041e-20 5.183108e-20
## GOBP_FATTY_ACID_METABOLIC_PROCESS              1.870360e-18 1.692424e-18
## GOBP_NUCLEOSIDE_BISPHOSPHATE_METABOLIC_PROCESS 3.369891e-16 3.049299e-16
##
## HALLMARK_ADIPOGENESIS
## HALLMARK_FATTY_ACID_METABOLISM
## GOBP_MONOCARBOXYLIC_ACID_METABOLIC_PROCESS     PDK4/ACSM3/ADIPOR2/ABHD5/NR1H3/MSMO1/ACAA1/ME1/PKM/AC
## GOCC_MITOCHONDRIAL_MATRIX                                   POLDIP2/PDK4/SLC25A5/ACSM3/LARS2/ALAS
## GOBP_FATTY_ACID_METABOLIC_PROCESS
## GOBP_NUCLEOSIDE_BISPHOSPHATE_METABOLIC_PROCESS
##                                                Count
## HALLMARK_ADIPOGENESIS                             57
## HALLMARK_FATTY_ACID_METABOLISM                    48
## GOBP_MONOCARBOXYLIC_ACID_METABOLIC_PROCESS        86
## GOCC_MITOCHONDRIAL_MATRIX                         81
## GOBP_FATTY_ACID_METABOLIC_PROCESS                 63
## GOBP_NUCLEOSIDE_BISPHOSPHATE_METABOLIC_PROCESS    34
```

```
p = dotplot(go_beige_all, showCategory =10, font.size=10) +  theme(axis.text.x = element_text(angle=45,
p
```



FPKM tables

```
lib_rpkm = data.frame(rpkm(filt, normalize.lib.sizes=TRUE))
colnames(lib_rpkm) = paste(filt$samples$donor.condition, "_rep",filt$samples$rep, sep="")
lib_rpkm = lib_rpkm[order(colnames(lib_rpkm))]

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

```
##            Geneid Length gene_name
## 1 ENSG00000000003   4536    TSPAN6
## 2 ENSG00000000005   1476      TNMD
## 3 ENSG00000000419   1207      DPM1
## 4 ENSG00000000457   6883     SCYL3
## 5 ENSG00000000460   5970  C1orf112
## 6 ENSG00000000938   3382       FGR
##                                                 description   gene_biotype
## 1                                            tetraspanin 6  protein_coding
## 2                                             tenomodulin  protein_coding
## 3 dolichyl-phosphate mannosyltransferase subunit 1, catalytic  protein_coding
## 4                                SCY1 like pseudokinase 3  protein_coding
## 5                        chromosome 1 open reading frame 112  protein_coding
```

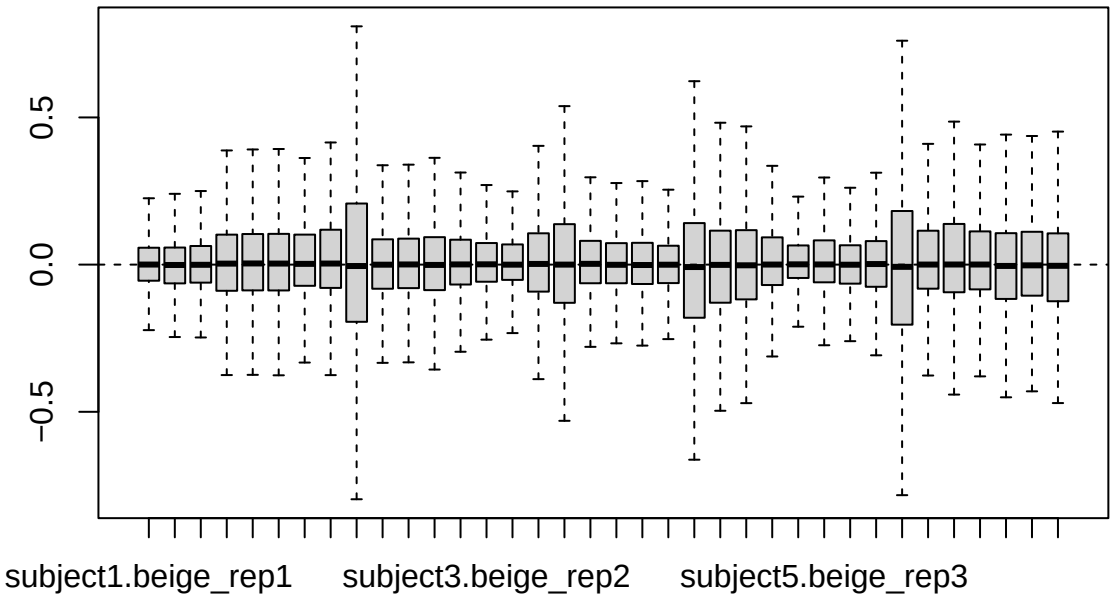
```
## 6              FGR proto-oncogene, Src family tyrosine kinase  protein_coding
##   ensembl_gene_id_version subject1.beige_rep1 subject1.beige_rep2
## 1      ENSG00000000003.15           6.9254899           7.1041492
## 2       ENSG00000000005.6           2.2801229           1.5381684
## 3      ENSG00000000419.12          33.4367039          31.4487083
## 4      ENSG00000000457.14           1.9757409           1.8447962
## 5      ENSG00000000460.17           0.3906274           0.3652596
## 6      ENSG00000000938.13           3.5126295           2.5525319
##   subject1.beige_rep3 subject1.white_rep1 subject1.white_rep2
## 1           7.0164629           4.7128194           5.0708211
## 2           1.0495540           0.4615557           0.1656141
## 3          27.0153634          33.0186300          32.4906301
## 4           1.7676057           1.4624651           1.4171984
## 5           0.4261506           0.3954615           0.4328556
## 6           2.8674982           0.7050256           0.9189715
##   subject1.white_rep3 subject2.beige_rep1 subject2.beige_rep2
## 1           5.4418345          10.9284653            9.774450
## 2           0.3331110          16.3059522           19.540289
## 3          34.7173572          33.3397649           27.154128
## 4           1.3734576           2.1309163            2.059405
## 5           0.3949397           0.4049026            0.418865
## 6           0.7467195           6.3532910            4.664816
##   subject2.beige_rep3 subject2.white_rep1 subject2.white_rep2
## 1          16.8977361           6.4045447            6.887769
## 2          56.2187609          23.9793882           33.149772
## 3          36.4339823          28.3932325           32.103143
## 4           2.9304897           1.5831367            1.618697
## 5           0.6188623           0.3936126            0.406746
## 6           2.9052807           0.8177922            0.756395
##   subject2.white_rep3 subject3.beige_rep1 subject3.beige_rep2
## 1           6.1627964          10.5844425            9.787839
## 2          33.5831338           2.5010606            2.553310
## 3          26.8767081          39.3488030           31.606617
## 4           1.5491843           2.2084212            2.076156
## 5           0.3295161           0.5057674            0.436492
## 6           1.0818338           9.3804593            6.719158
##   subject3.beige_rep3 subject3.white_rep1 subject3.white_rep2
## 1            8.801945           6.3522863           6.0961385
## 2            3.315264           0.2897050           0.5306761
## 3           31.369481          31.2666494          35.9181825
## 4            2.056277           1.5754197           1.2810999
## 5            0.501716           0.3948589           0.4949906
## 6            5.440378           0.8493867           1.1545008
##   subject3.white_rep3 subject4.beige_rep1 subject4.beige_rep2
## 1           6.2487150           9.2558572           9.7210231
## 2           0.6497380           1.8138929           3.4756272
## 3          37.0786698          33.5048504          37.0758923
## 4           1.6494505           2.1226017           2.2437674
## 5           0.4316152           0.4215859           0.5620584
## 6           0.8506919           7.4686360           5.3233254
##   subject4.beige_rep3 subject4.white_rep1 subject4.white_rep2
## 1           8.9397143           8.7135345           8.7486479
## 2           1.6126162           1.0275705           0.6455009
## 3          27.5773901          36.3470915          34.3252630
```

```
## 4           2.0604610           1.8713614           1.5792702
## 5           0.4256922           0.7014885           0.5754954
## 6           4.7395932           0.5689426           0.6487976
##   subject4.white_rep3 subject5.beige_rep1 subject5.beige_rep2
## 1           9.6065895          10.6304243           9.6054108
## 2           0.2907158           5.7198626           5.4055047
## 3          33.3195527          30.3958272          30.0042303
## 4           1.6832203           1.8713287           1.9437106
## 5           0.4114251           0.3833146           0.3504826
## 6           0.6868834           7.3511831           4.6432371
##   subject5.beige_rep3 subject5.white_rep1 subject5.white_rep2
## 1           9.9677634           6.5873342           6.3455039
## 2           7.5206419           1.3830061           1.5991558
## 3          34.5704866          34.8785961          35.4498533
## 4           1.9163335           1.3438945           1.6117483
## 5           0.3321905           0.4575574           0.3690112
## 6           4.8957811           0.9359907           0.7870949
##   subject5.white_rep3 subject6.beige_rep1 subject6.beige_rep2
## 1           5.9682040          11.1349030           12.557102
## 2           2.1362231           7.3100161           11.403522
## 3          22.5609106          34.2276292           40.814843
## 4           1.5871360           1.8944114            2.187790
## 5           0.5975049           0.3548748            0.360928
## 6           0.4520281          12.7224387           13.732668
##   subject6.beige_rep3 subject6.white_rep1 subject6.white_rep2
## 1          11.6781141            7.902057           8.8622896
## 2           7.8596296            2.052549           2.5683056
## 3          39.7817825           37.952305          35.7383505
## 4           2.1737580            1.803030           1.8126365
## 5           0.4118008            0.440210           0.4430891
## 6          11.8919911            1.662068           2.4080463
##   subject6.white_rep3
## 1           8.3762978
## 2           1.7968096
## 3          33.8286255
## 4           1.3539377
## 5           0.4257265
## 6           1.8079666
```

```
rowMeans(format_rpkm[format_rpkm$gene_name == "PPARG",c(7:ncol(format_rpkm))]) #Average RPKM 36
```

```
##     5319
## 36.46861
```

```
plotRLE(data.matrix(format_rpkm[7:ncol(format_rpkm)]), outline=FALSE)  #check normalisation
```




```
write.table(format_rpkm, sep='\t', row.names = FALSE,quote = F,
            file=here("03limma/beige_day15_rpkm_tmm.tab"))
```

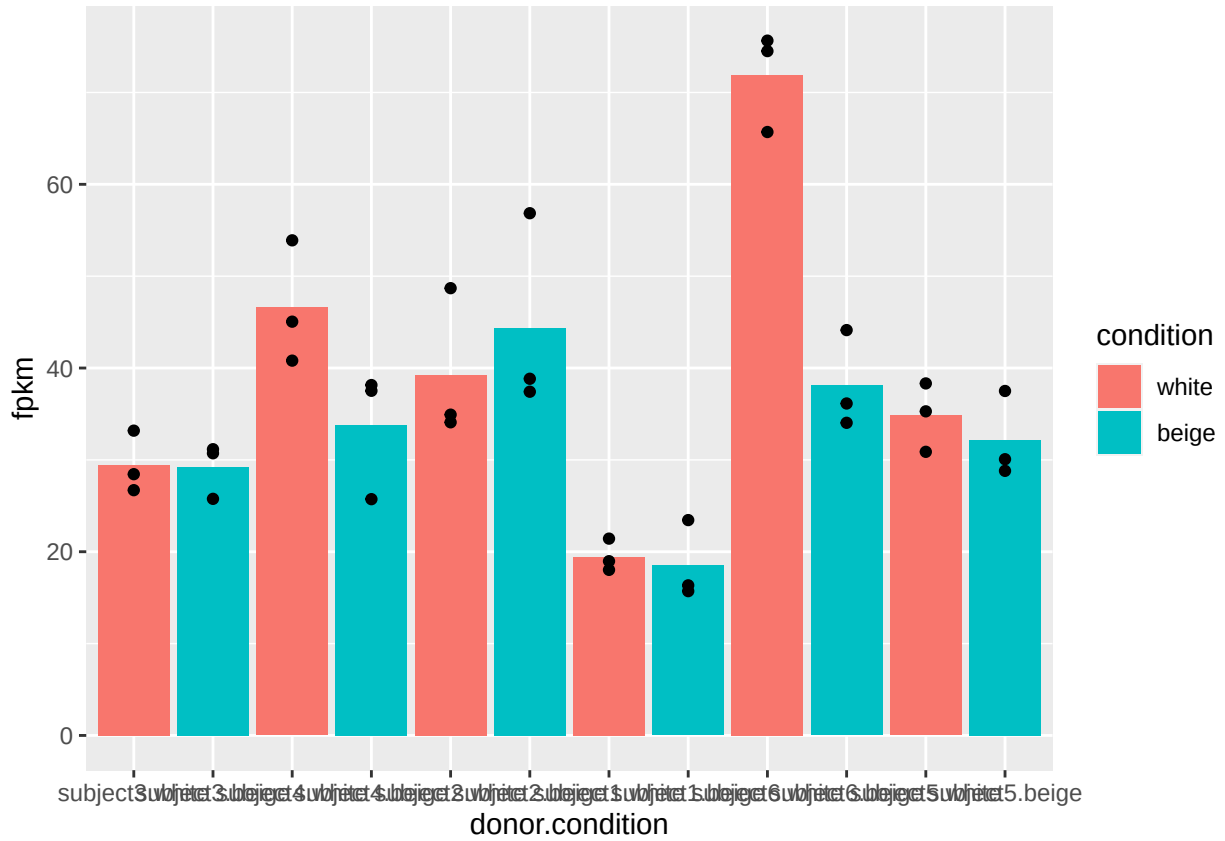
```
library(tidyr)
long = pivot_longer(format_rpkm, 7:ncol(format_rpkm), names_to = "biorep", values_to ="fpkm")
long$donor = factor(gsub("\\..*", "", long$biorep), levels=unique(filt$samples$donor))
long$condition = factor(gsub(".*\\.","", gsub("_rep.","", long$biorep)), levels=c("white","beige"))
long$rep = gsub(".*_","", long$biorep)
long$donor.condition = factor(paste(long$donor, long$condition, sep="."),
                              levels = paste(rep(levels(long$donor),each=2), rep(c("white","beige"), 6)
head(long)
```

```
## # A tibble: 6 x 12
##   Geneid Length gene_name description gene_biotype ensembl_gene_id_vers~1 biorep
##   <chr>   <dbl> <chr>     <chr>       <chr>        <chr>                  <chr>
## 1 ENSG0~   4536 TSPAN6    "tetraspan~ protein_cod~ ENSG00000000003.15     subje~
## 2 ENSG0~   4536 TSPAN6    "tetraspan~ protein_cod~ ENSG00000000003.15     subje~
## 3 ENSG0~   4536 TSPAN6    "tetraspan~ protein_cod~ ENSG00000000003.15     subje~
## 4 ENSG0~   4536 TSPAN6    "tetraspan~ protein_cod~ ENSG00000000003.15     subje~
## 5 ENSG0~   4536 TSPAN6    "tetraspan~ protein_cod~ ENSG00000000003.15     subje~
## 6 ENSG0~   4536 TSPAN6    "tetraspan~ protein_cod~ ENSG00000000003.15     subje~
## # i abbreviated name: 1: ensembl_gene_id_version
## # i 5 more variables: fpkm <dbl>, donor <fct>, condition <fct>, rep <chr>,
## #   donor.condition <fct>
```

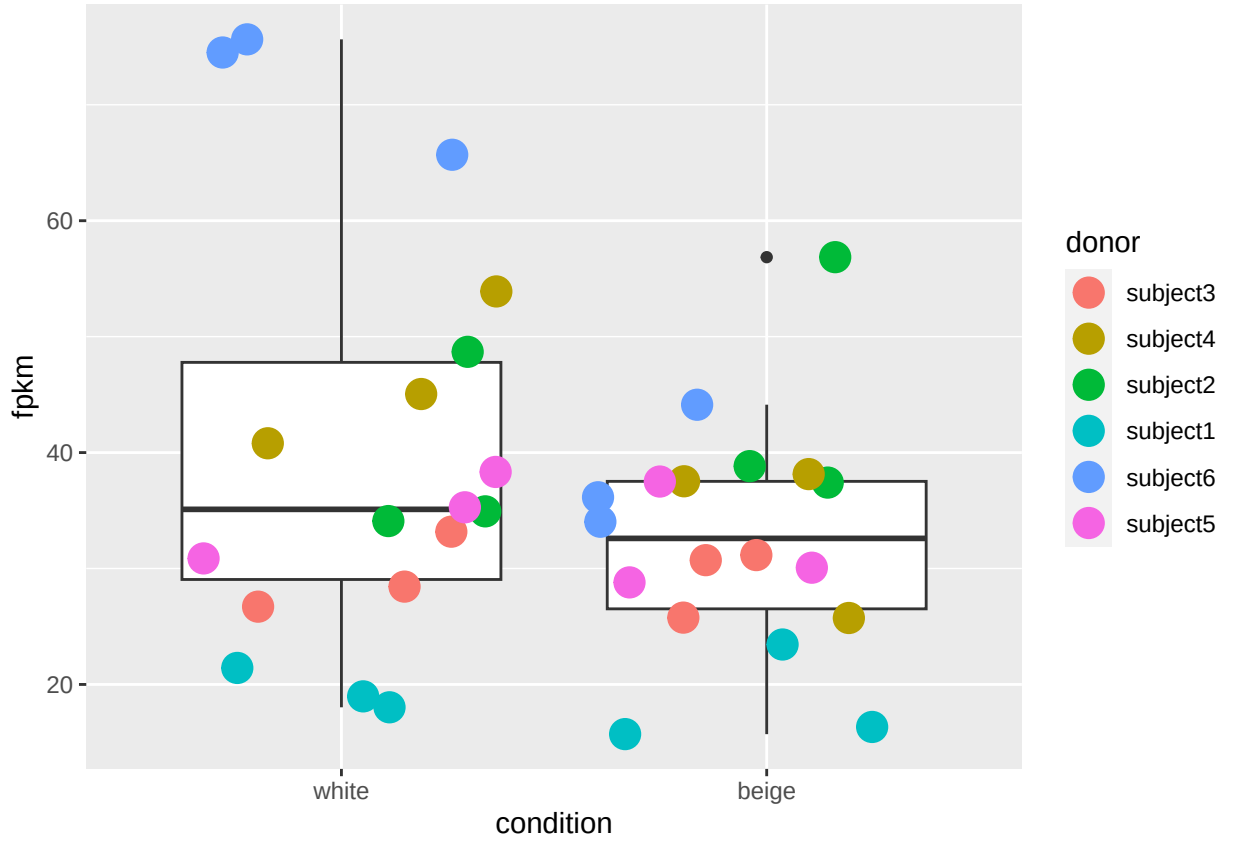
```
long = group_by(long, Geneid) %>% mutate(zscore= scale(fpkm))
head(long)
```

```
## # A tibble: 6 x 13
## # Groups:   Geneid [1]
##   Geneid Length gene_name description gene_biotype ensembl_gene_id_vers~1 biorep
##   <chr>   <dbl> <chr>     <chr>       <chr>        <chr>                  <chr>
## 1 ENSG0~   4536 TSPAN6    "tetraspan~ protein_cod~ ENSG00000000003.15     subje~
## 2 ENSG0~   4536 TSPAN6    "tetraspan~ protein_cod~ ENSG00000000003.15     subje~
## 3 ENSG0~   4536 TSPAN6    "tetraspan~ protein_cod~ ENSG00000000003.15     subje~
## 4 ENSG0~   4536 TSPAN6    "tetraspan~ protein_cod~ ENSG00000000003.15     subje~
## 5 ENSG0~   4536 TSPAN6    "tetraspan~ protein_cod~ ENSG00000000003.15     subje~
## 6 ENSG0~   4536 TSPAN6    "tetraspan~ protein_cod~ ENSG00000000003.15     subje~
## # i abbreviated name: 1: ensembl_gene_id_version
## # i 6 more variables: fpkm <dbl>, donor <fct>, condition <fct>, rep <chr>,
## #   donor.condition <fct>, zscore <dbl[,1]>
```

```
ggplot(long[long$gene_name == "PPARG",]) + geom_bar(stat="summary",fun="mean", aes(x=donor.condition, y=
  geom_point(aes(x=donor.condition, y=fpkm))
```




```
ggplot(long[long$gene_name == "PPARG",]) + geom_boxplot(aes(x=condition, y=fpkm)) +
  geom_jitter(aes(x=condition, y=fpkm, colour=donor), size=5)
```




```
save(long, file=here("03limma/rpkm_rep_for_plotting.RData"))
```

```
filt$samples$group = factor(filt$samples$donor.condition)
group_rpkm = data.frame(rpkmByGroup(filt,normalize.lib.sizes=TRUE))

colSums(group_rpkm)
```

```
## subject1.beige subject1.white subject2.beige subject2.white subject3.beige
##       244925.7       228831.0       349289.1       259712.5       296811.1
## subject3.white subject4.beige subject4.white subject5.beige subject5.white
##       263803.7       288923.2       313523.5       279363.2       259112.1
## subject6.beige subject6.white
##       289573.1       327761.0
```

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

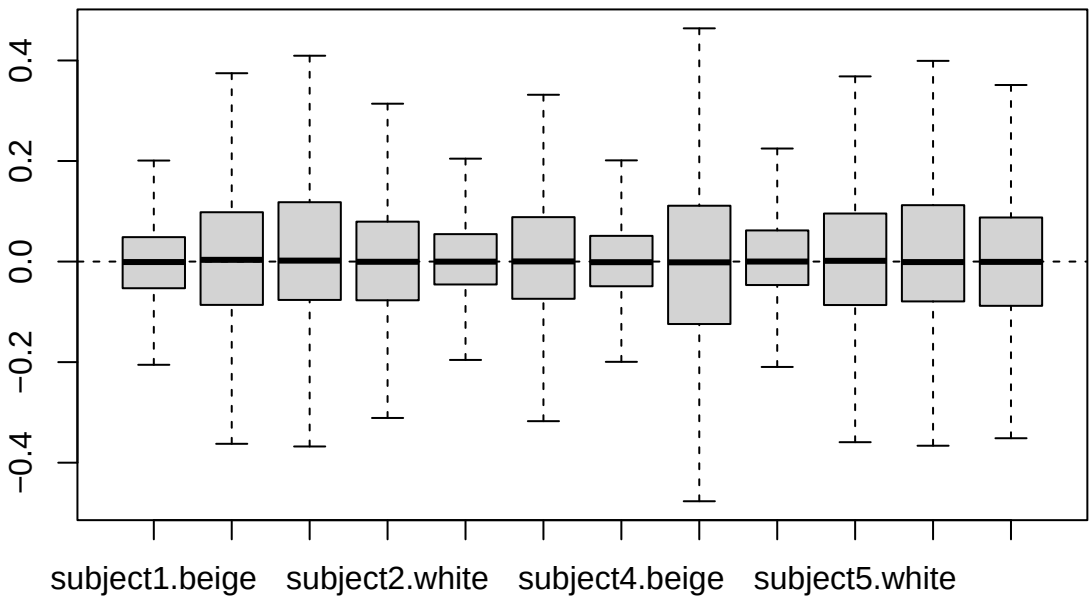
```
##            Geneid Length gene_name
## 1 ENSG00000000003   4536    TSPAN6
## 2 ENSG00000000005   1476      TNMD
## 3 ENSG00000000419   1207      DPM1
## 4 ENSG00000000457   6883     SCYL3
## 5 ENSG00000000460   5970  C1orf112
```

```
## 6 ENSG00000000938   3382       FGR
##                                                    description   gene_biotype
## 1                                              tetraspanin 6  protein_coding
## 2                                                tenomodulin  protein_coding
## 3 dolichyl-phosphate mannosyltransferase subunit 1, catalytic  protein_coding
## 4                                   SCY1 like pseudokinase 3  protein_coding
## 5                          chromosome 1 open reading frame 112  protein_coding
## 6              FGR proto-oncogene, Src family tyrosine kinase  protein_coding
##   ensembl_gene_id_version subject1.beige subject1.white subject2.beige
## 1      ENSG00000000003.15       7.015370      5.0752183     12.5300941
## 2       ENSG00000000005.6       1.627583      0.3200113     30.6612441
## 3      ENSG00000000419.12      30.635853     33.4089794     32.3055705
## 4      ENSG00000000457.14       1.862880      1.4176901      2.3720554
## 5      ENSG00000000460.17       0.393713      0.4077405      0.4788014
## 6      ENSG00000000938.13       2.977751      0.7902128      4.6470874
##   subject2.white subject3.beige subject3.white subject4.beige subject4.white
## 1      6.4851444      9.7247952      6.2324136      9.3056202      9.0232992
## 2     30.2448972      2.7906896      0.4912616      2.3021246      0.6401144
## 3     29.1252400     34.1100563     34.7548040     32.7208903     34.6618668
## 4      1.5836657      2.1136964      1.5022350      2.1423301      1.7105704
## 5      0.3762281      0.4816479      0.4403470      0.4698593      0.5608791
## 6      0.8865716      7.1807572      0.9509210      5.8448505      0.6362375
##   subject5.beige subject5.white subject6.beige subject6.white
## 1     10.0681020      6.3019890     11.7900221      8.3815053
## 2      6.2119587      1.6856768      8.8600561      2.1400068
## 3     31.6549387     30.9930433     38.2724535     35.8339890
## 4      1.9104027      1.5130770      2.0849212      1.6548458
## 5      0.3557019      0.4699746      0.3751493      0.4361585
## 6      5.6321927      0.7380337     12.7832922      1.9632742
```

```
rowMeans(format_grpkm[format_grpkm$gene_name == "PPARG",c(7:ncol(format_grpkm))])#expr matches the repl
```

```
##     5319
## 36.46869
```

```
plotRLE(data.matrix(format_grpkm[7:ncol(format_grpkm)]), outline=FALSE)
```




```
write.table(format_grpkm, sep='\t', row.names = FALSE, quote=F,
            file=here("03limma/beige_day15_rpkm_tmm_means.tab"))
```