Transgenerational study - Gene expression analysis of F2 liver

J.Beil and J. Perner

LOAD DATA

```r
counts <- read.csv("./data/GSE229524_liver_RNA_counts.csv.gz", row.names = 1)
colnames(counts) <- gsub("animal|_liver_RNA", "", colnames(counts))

avgTx_length <- read.csv("./data/GSE229524_liver_RNA_avgTxLength.csv.gz", row.names = 1)
colnames(avgTx_length) <- gsub("animal|_liver_RNA", "", colnames(avgTx_length))

animal_metadata <- read.csv2("./data/animal_metadata.csv", sep=",")
row.names(animal_metadata) <- animal_metadata$animal
```

```r
# 100: https://apr2020.archive.ensembl.org
ensembl100 <- useEnsembl(biomart = "ENSEMBL_MART_ENSEMBL",
                         dataset = "rnorvegicus_gene_ensembl",
                         version = 100)
anno <- getBM(attributes = c("ensembl_gene_id_version", "ensembl_gene_id", "rgd_symbol"),
              filters = "ensembl_gene_id_version",
              values = row.names(counts),
              mart = ensembl100)

# collapse duplicated gene names
tmp1 <- split(anno$rgd_symbol, anno$ensembl_gene_id_version)
tmp1 <- sapply(tmp1, paste0, collapse=";")
tmp2 <- unique(anno[,1:2])[, 2]
names(tmp2) <- unique(anno[,1:2])[, 1]
anno <- data.frame(ensembl_gene_id = tmp2,
                   rgd_symbol = tmp1[names(tmp2)],
                   row.names = names(tmp2))
```

```r
dds <- DESeqDataSetFromMatrix(counts,
                              colData=DataFrame(animal_metadata[colnames(counts), ]),
                              design=~ F0 + F1 + F2)
assay(dds, "avgTxLength") <- avgTx_length[, colnames(dds)]
```

FILTER DATA

```r
# filter genes by group: samples in each group should together have at least 10 reads
idx <- split(row.names(colData(dds)), colData(dds)[, c("F0andF1", "F2")])
keep <- sapply(idx, function(i) rowSums(counts(dds)[, i])>10)
dds <- dds[rowSums(keep)>0, ]
```

F0 AND F1 DIFFERENTIAL EXPRESSION ANALYSIS

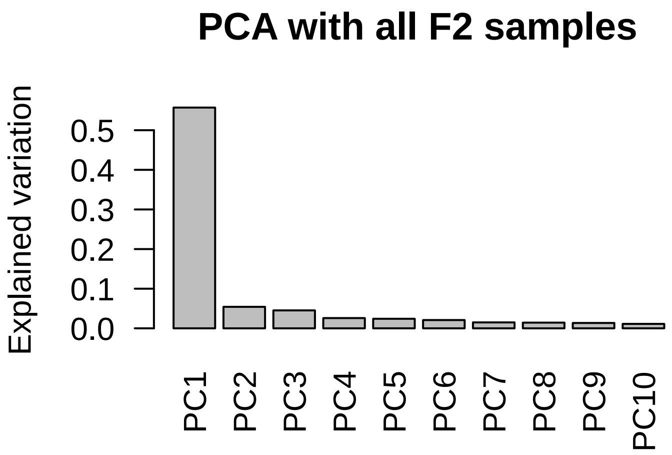
```
dds <- DESeq(dds)
res_f0 <- results(dds, name="F0_treated_vs_control")
res_f1 <- results(dds, name="F1_treated_vs_control")
```

EXPLORATORY ANALYSIS

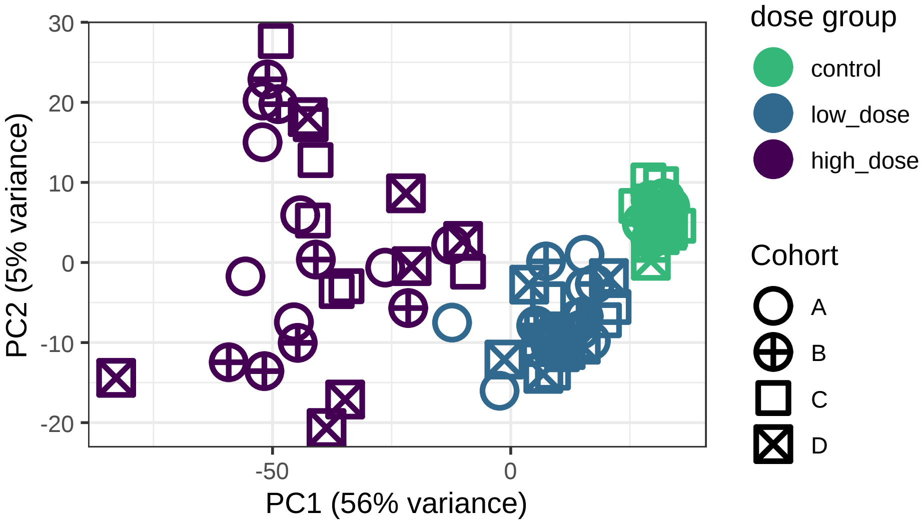
```
# variance stabilizing transformation for PCA plots
vsd <- vst(dds)
```

all groups

```
pca <- runPCA_topN(assay(vsd), 3000)
vars <- plot_ExplainedVar(pca, "PCA with all F2 samples")
```

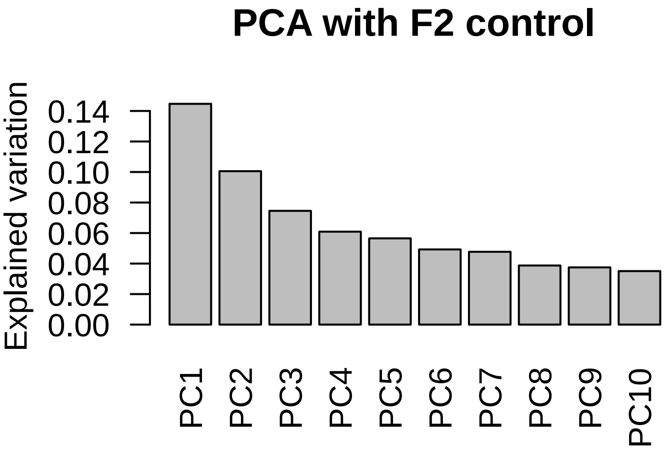


```
# prepare data frame for plotting
toplot <- data.frame(PC1=pca$x[,1],
                     PC2=pca$x[,2],
                     PC3=pca$x[,3],
                     PC4=pca$x[,4],
                     PC5=pca$x[,5])
toplot <- cbind(toplot, colData(dds)[row.names(toplot), ])
toplot <- as.data.frame(toplot)

# set aesthetics;
setcols <- rev(viridis(4)[-c(4)])
names(setcols) <- c("control", "low_dose", "high_dose")

# single plot for PC1 and PC2
plot_PC_1to2(toplot, setcols, "", vars=vars,
             col_var = "F2", col_var_name="dose group",
             shape_var="F0andF1")
```

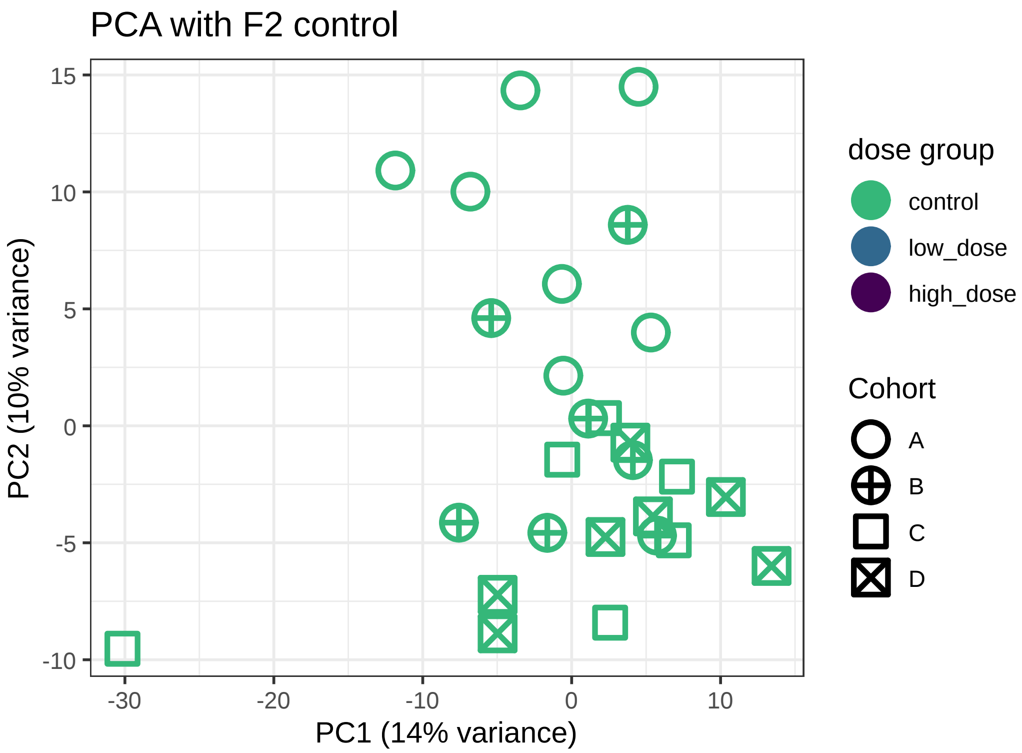


F2 treatment groups only (example: control group)

```
# subselect samples for PCA
vsd_sub <- assay(vsd)[, colData(vsd)$F2%in%"control"]
pca <- runPCA_topN(vsd_sub)
vars <- plot_ExplainedVar(pca, "PCA with F2 control")
```



```
toplot <- data.frame(PC1=pca$x[,1],
                     PC2=pca$x[,2],
                     PC3=pca$x[,3],
                     PC4=pca$x[,4],
                     PC5=pca$x[,5])
toplot <- cbind(toplot, colData(dds)[row.names(toplot), ])
toplot <- as.data.frame(toplot)

plot_PC_1to2(toplot, setcols, "PCA with F2 control", vars,
             col_var = "F2", col_var_name="dose group",
             shape_var="F0andF1")
```

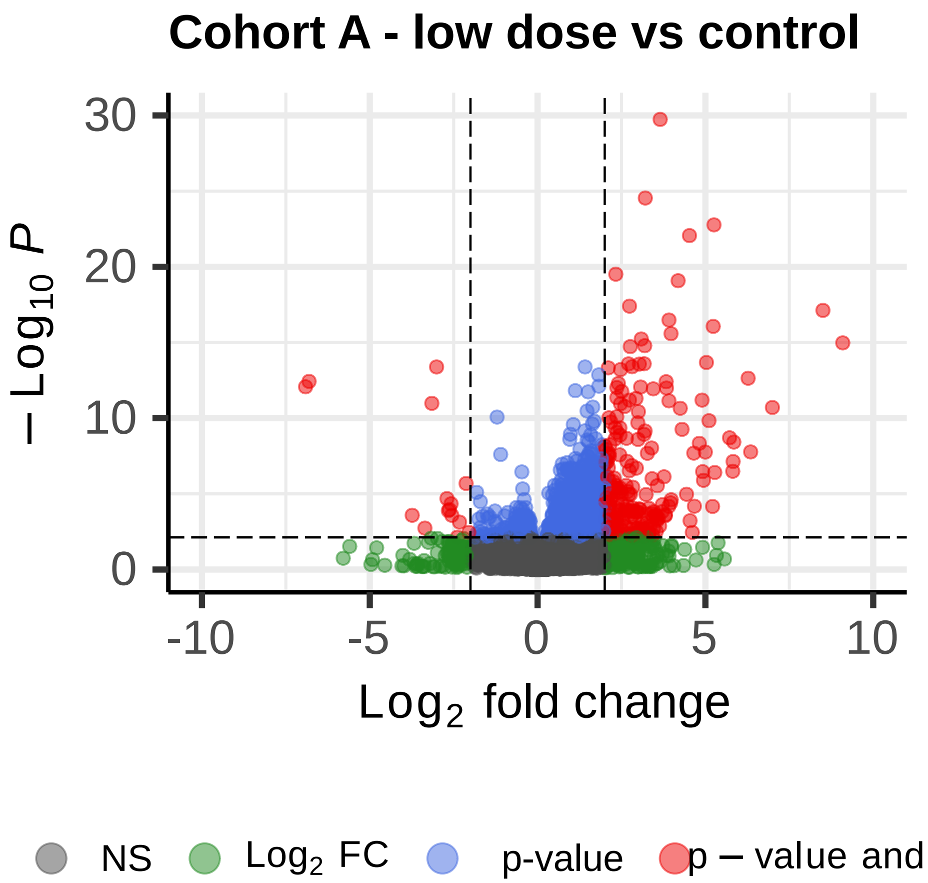


DE ANALYSIS BETWEEN TREATMENT GROUP WITHIN COHORT

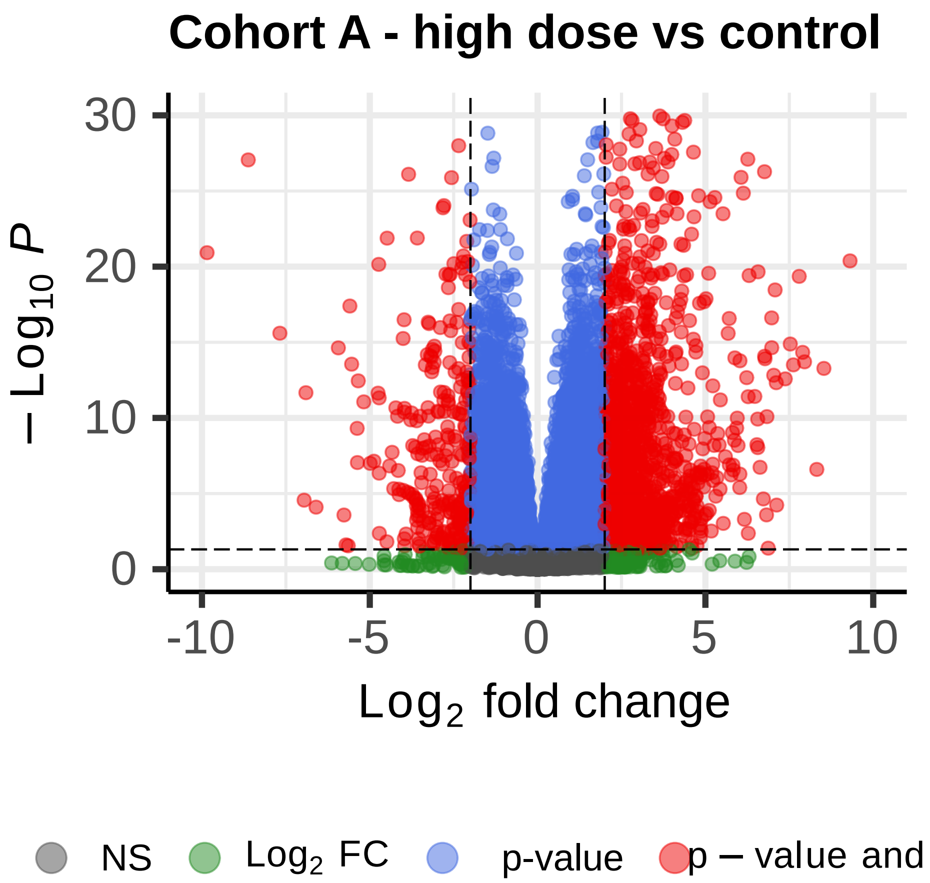
```r
# build new grouping variable for easier referencing based on cohort and F2 treatment
tmp <- unite(as.data.frame(colData(dds)), Group, c(F0andF1, F2)) %>% pull(Group)
colData(dds)$Group <- factor(tmp, sort(unique(tmp)))

# rerun DE analysis
design(dds) <- ~ Group
dds <- DESeq(dds)
```

```r
# keep statistics for each group
resA_LDvsCtrl <- results(dds, contrast =c("Group", "A_low_dose", "A_control"))
resA_HDvsCtrl <- results(dds, contrast =c("Group", "A_high_dose", "A_control"))

resB_LDvsCtrl <- results(dds, contrast =c("Group", "B_low_dose", "B_control"))
resB_HDvsCtrl <- results(dds, contrast =c("Group", "B_high_dose", "B_control"))

resC_LDvsCtrl <- results(dds, contrast =c("Group", "C_low_dose", "C_control"))
resC_HDvsCtrl <- results(dds, contrast =c("Group", "C_high_dose", "C_control"))

resD_LDvsCtrl <- results(dds, contrast =c("Group", "D_low_dose", "D_control"))
resD_HDvsCtrl <- results(dds, contrast =c("Group", "D_high_dose", "D_control"))
```

```r
# volcano plots for different contrasts - example cohort A
plot_Volcano(resA_LDvsCtrl, "Cohort A - low dose vs control" )
```

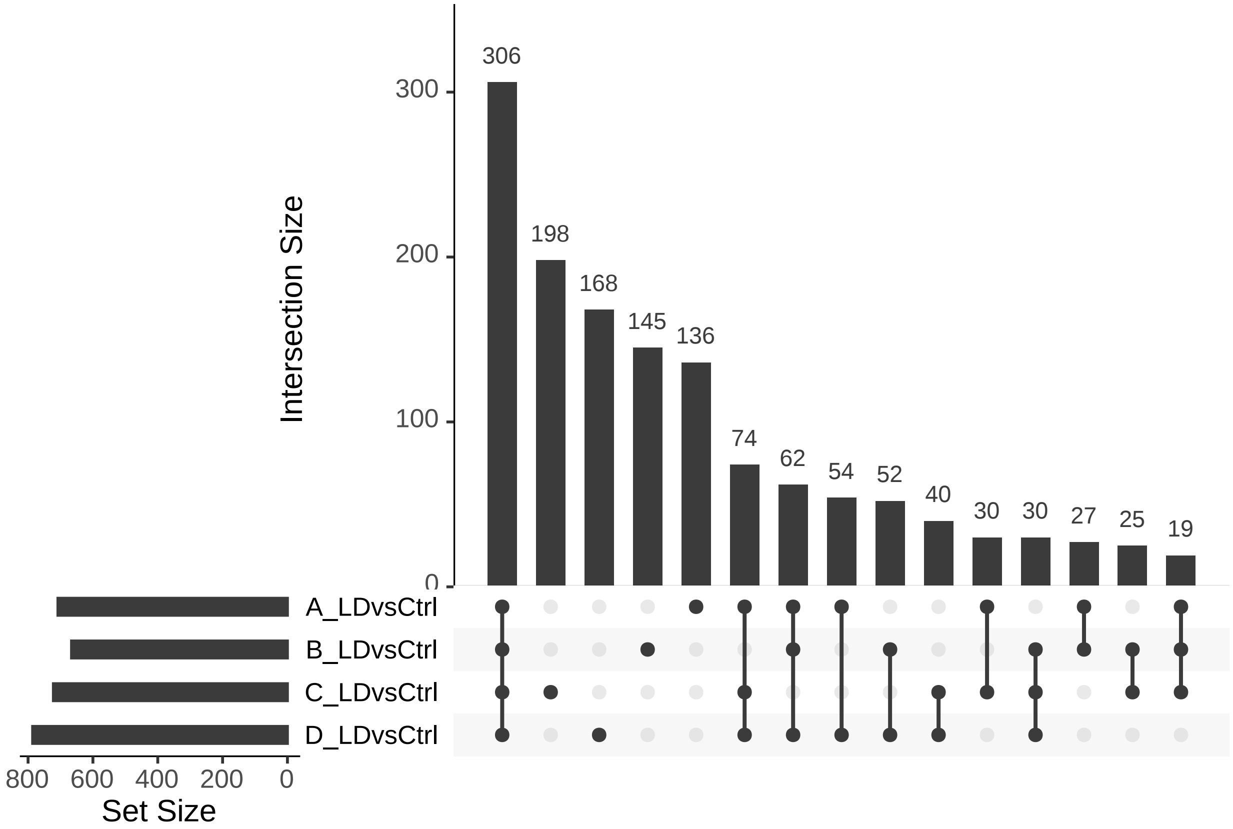


```r
plot_Volcano(resA_HDvsCtrl, "Cohort A - high dose vs control")
```

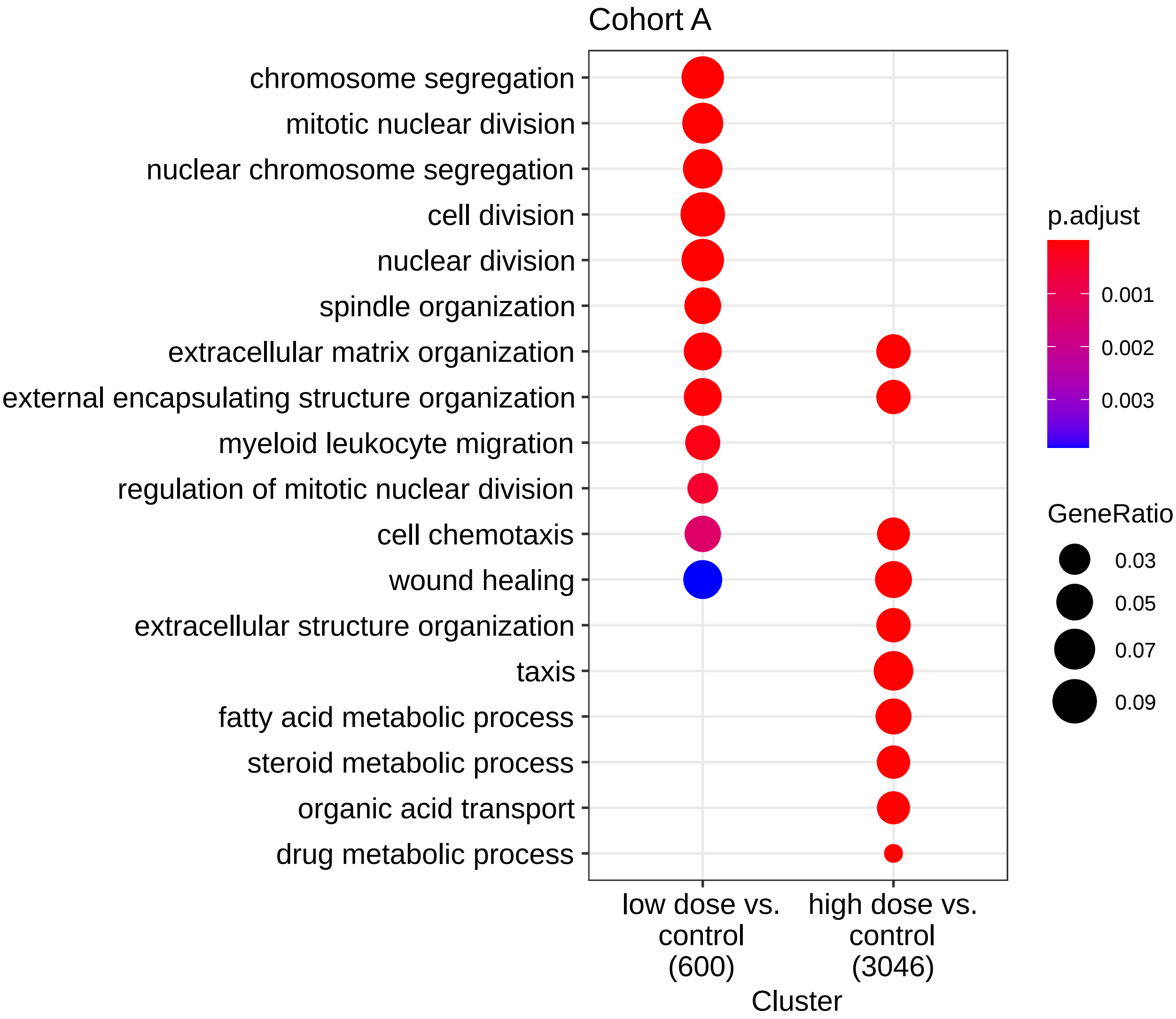


```
# select differentially expressed genes
resA_LDvsCtrl_sig <- selectSignGenes(resA_LDvsCtrl)
resA_HDvsCtrl_sig <- selectSignGenes(resA_HDvsCtrl)

resB_LDvsCtrl_sig <- selectSignGenes(resB_LDvsCtrl)
resB_HDvsCtrl_sig <- selectSignGenes(resB_HDvsCtrl)

resC_LDvsCtrl_sig <- selectSignGenes(resC_LDvsCtrl)
resC_HDvsCtrl_sig <- selectSignGenes(resC_HDvsCtrl)

resD_LDvsCtrl_sig <- selectSignGenes(resD_LDvsCtrl)
resD_HDvsCtrl_sig <- selectSignGenes(resD_HDvsCtrl)
```

```
# check overlap of DE genes for different contrasts - example for low dose vs control contrasts
pp <- upset(fromList(list(A_LDvsCtrl=row.names(resA_LDvsCtrl_sig),
                          B_LDvsCtrl=row.names(resB_LDvsCtrl_sig),
                          C_LDvsCtrl=row.names(resC_LDvsCtrl_sig),
                          D_LDvsCtrl=row.names(resD_LDvsCtrl_sig))),
            sets = c("D_LDvsCtrl", "C_LDvsCtrl", "B_LDvsCtrl", "A_LDvsCtrl"),
            keep.order=TRUE,
            empty.intersections = FALSE,
            order.by = c("degree", "freq"),
            decreasing = c(FALSE, TRUE),
            text.scale = 1.4)
pp
```

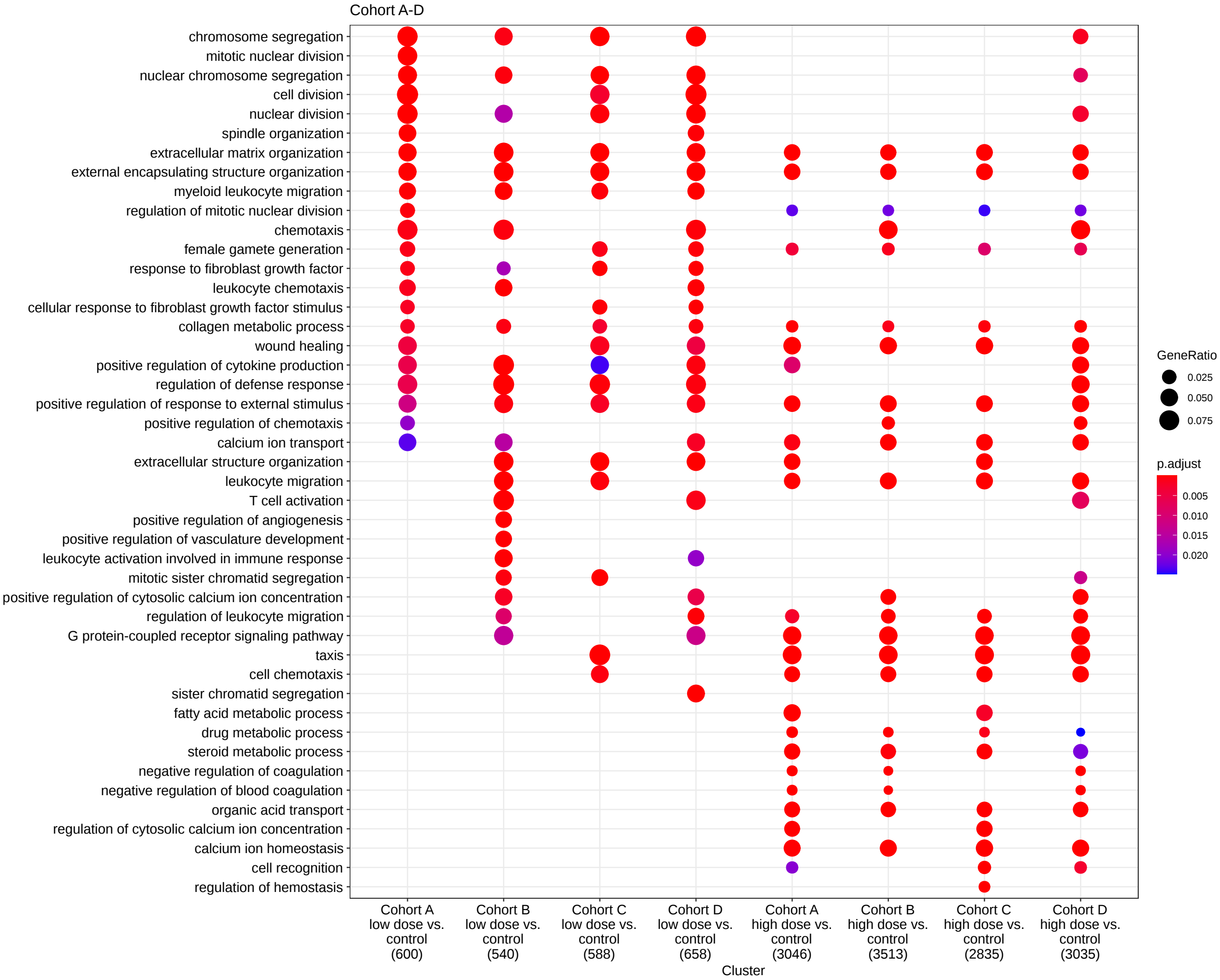


```
# define gene universe = all genes for which we could calculate DE statistics in the two contrasts
res_list <- list(resA_LDvsCtrl, resA_HDvsCtrl)
common_genes <- row.names(resA_HDvsCtrl)[rowSums(is.na(sapply(res_list, function(x) x$padj))) <1]

# list of selected DE genes by contrast
de_genes <- list(`low dose vs.\ncontrol` = row.names(resA_LDvsCtrl_sig),
                 `high dose vs.\ncontrol` = row.names(resA_HDvsCtrl_sig))

# map to Ensemble gene ID
common_genes <- sapply(common_genes, function(x) anno[x, "ensembl_gene_id"])
de_genes <- lapply(de_genes, function(x) anno[x, "ensembl_gene_id"])

# GO term enrichment analysis (biological processes only) comparing the two contrasts
go_term_A <- compareCluster(geneClusters= de_genes,
                            fun = "enrichGO",
                            universe = common_genes,
                            OrgDb = "org.Rn.eg.db",
                            keyType = "ENSEMBL",
                            ont = "BP",
                            pvalueCutoff = 0.01,
                            qvalueCutoff = 0.05,
                            pAdjustMethod = "fdr",
                            minGSSize = 10,
                            maxGSSize = 500)

#cut some redundant GO terms
go_term_A_simple <- clusterProfiler::simplify(go_term_A,
                                              cutoff=0.7,
                                              by="p.adjust",
                                              select_fun=min)

# plot results
dotplot(go_term_A_simple, showCategory=10, title="Cohort A")
```



```
### common_genes based on all groups
res_list <- list(resA_HDvsCtrl, resB_HDvsCtrl, resC_HDvsCtrl, resD_HDvsCtrl,
                 resA_LDvsCtrl, resB_LDvsCtrl, resC_LDvsCtrl, resD_LDvsCtrl)
common_genes <- row.names(resA_HDvsCtrl)[rowSums(is.na(sapply(res_list, function(x) x$padj))) <1]
common_genes <- sapply(common_genes, function(x) anno[x, "ensembl_gene_id"])

#compare significant genes in one de_genes list as different geneClusters
de_genes <- list( `Cohort A\nlow dose vs.\ncontrol` =row.names(resA_LDvsCtrl_sig),
                  `Cohort B\nlow dose vs.\ncontrol`=row.names(resB_LDvsCtrl_sig),
                  `Cohort C\nlow dose vs.\ncontrol`=row.names(resC_LDvsCtrl_sig),
                  `Cohort D\nlow dose vs.\ncontrol`=row.names(resD_LDvsCtrl_sig),
```

```r
                  `Cohort A\nhigh dose vs.\ncontrol`=row.names(resA_HDvsCtrl_sig),
                  `Cohort B\nhigh dose vs.\ncontrol`=row.names(resB_HDvsCtrl_sig),
                  `Cohort C\nhigh dose vs.\ncontrol`=row.names(resC_HDvsCtrl_sig),
                  `Cohort D\nhigh dose vs.\ncontrol`=row.names(resD_HDvsCtrl_sig) )
de_genes <- lapply(de_genes, function(x) anno[x, "ensembl_gene_id"])

go_term_all_HD <- compareCluster(geneClusters= de_genes,
               fun = "enrichGO",
               universe = common_genes, #intersection of all noNA genes
                             OrgDb = "org.Rn.eg.db",
                             keyType = "ENSEMBL",
                             ont = "BP",
                             pvalueCutoff = 0.025,
                             qvalueCutoff = 0.04,
                             pAdjustMethod = "fdr",
                             minGSSize = 10,
                             maxGSSize = 500)
#cut some redundant Go terms
go_term_all_simple <- clusterProfiler::simplify(go_term_all_HD,
                              cutoff=0.7,
                              by="p.adjust",
                              select_fun=min)

dotplot(go_term_all_simple, showCategory=12, title="Cohort A-D")
```



DE ANALYSIS BETWEEN COHORTS WITHIN TREATMENT GROUP

```
res_AvsC_ctrl <- results(dds, contrast =c("Group", "C_control", "A_control"))
res_AvsD_ctrl <- results(dds, contrast =c("Group", "D_control", "A_control"))
```

SELECT AND ANALYSE F0-ASSOCIATED DIFFERENTIALLY EXPRESSED GENES

```
all_res <- cbind(res_f0[, c("baseMean", "log2FoldChange", "padj")],
                 res_f1[, c("log2FoldChange", "padj")],
                 res_AvsC_ctrl[, c("log2FoldChange", "padj")],
                 res_AvsD_ctrl[, c("log2FoldChange", "padj")])

all_res <- all_res[!apply(all_res[, grepl("padj", colnames(all_res))],
                          1, function(x) sum(is.na(x))>2), ]
all_res <- as.data.frame(all_res)
```

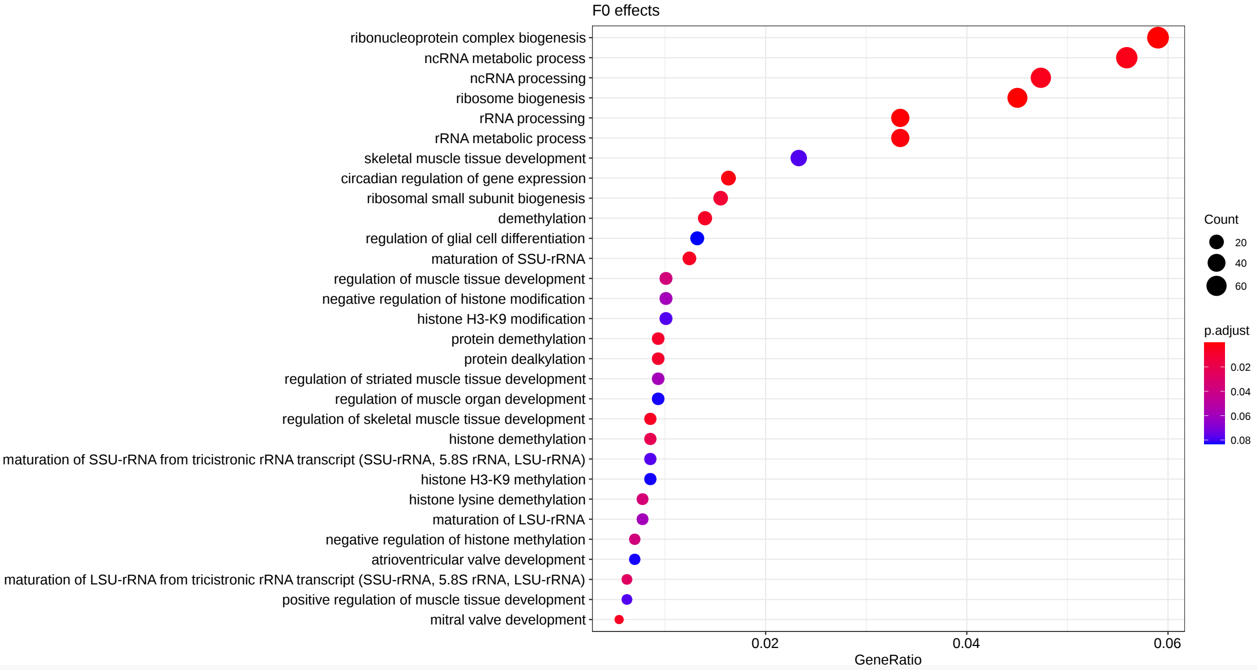
```r
all_res$genes <- anno[row.names(all_res), "rgd_symbol"]
colnames(all_res) <- c("baseMean",
                       unlist(lapply(c("F0", "F1", "AvsC", "AvsD"),
                                     paste, c("log2FC", "FDR"), sep="_")),
                       "gene")
knitr::kable(head(all_res))
```

	baseMean	F0_log2FC	F0_FDR	F1_log2FC	F1_FDR	AvsC_log2FC	AvsC_FDR	AvsD_log2FC	AvsD_FDR	gene
ENSRNOG00000000001.5	6.852368	0.0863101	0.8874273	0.1791090	0.8107811	0.5869153	0.8142304	0.1563366	0.9489577	
ENSRNOG00000000007.7	5.511898	0.1967154	0.6127445	-0.0527750	NA	0.0266432	0.9958033	-0.0727502	NA	Gad1
ENSRNOG00000000017.7	14.918171	-0.1621037	0.7247667	0.4411666	0.4398289	-0.4584782	0.8168477	0.1181633	0.9474644	Steap1
ENSRNOG00000000021.13	119.988896	-0.0355114	0.8426598	0.0626019	0.7701166	0.0960125	0.8986485	0.1929687	0.6649111	
ENSRNOG00000000024.2	7622.067315	0.0469581	0.8411652	-0.1855178	0.4692397	0.3504180	0.5923682	-0.4505608	0.3099140	Hebp1
ENSRNOG00000000033.15	43.009410	0.0246413	0.9293735	0.0888970	0.7816401	-0.2129287	0.8166284	-0.2692640	0.6702070	Tmcc2

```r
selectedGenes <- all_res$F0_FDR < 0.1 &  all_res$baseMean>50
selectedGenes[is.na(selectedGenes)] <- FALSE
print(table(selectedGenes))
```

```
## selectedGenes
## FALSE  TRUE
## 14157  1523
```

```r
de_genes <- row.names(all_res[selectedGenes, ])
de_genes <- anno[de_genes, "ensembl_gene_id"]
universe <- anno[row.names(all_res), "ensembl_gene_id"]
go <- enrichGO(de_genes,
               universe = universe,
               OrgDb = "org.Rn.eg.db",
               keyType = "ENSEMBL",
               ont = "ALL",
               pvalueCutoff = 0.1,
               qvalueCutoff = 0.2,
               pAdjustMethod = "fdr",
               minGSSize = 10,
               maxGSSize = 500)

dotplot(go, showCategory=30, title="F0 effects")
```



```
tmp <- data.frame(go)
tmp$geneID <- sapply(strsplit(tmp$geneID, "/"),
                     function(x) paste0(anno[anno$ensembl_gene_id%in%x, 2],
                                        collapse="/ "))
knitr::kable(head(tmp))
```

	ONTOLOGY	ID	Description	GeneRatio	BgRatio	pvalue	p.adjust	qvalue	geneID	Count
GO:0042254	BP	GO:0042254	ribosome biogenesis	58/12885/12373+0.000007830007762					Rpf2/ Rrp1b/ Rrp1/ Pwp2/ Lsg1/ Rrp15/ Aatf/ Tsr1/ Rrn3/ Utp3/ Dcaf13/ Utp23/ Pes1/ Utp25/ Pa2g4/ Surf6/ Nol10/ Shq1/ Nop56/ Rrs1/ Nmd3/ Ftsj3/ Nol9/ Mrps2/ Nop14/ Ddx10/ Srfbp1/ Heatr3/ Xrcc5/ Gtpbp4/ Mphosph10/ Nop58/ Wdr12/ Brix1/ Nop2/ Rrp8/ Wdr3/ Utp4/ Ppan/ Bop1/ Nob1/ Rrp7a/ Ddx18/ Mettl17/ Wdr43/ Naf1/ Wdr36/ Mrm1/ Usp36/ Mrm3/ Ddx21/ Nol8;NEWGENE_1309551/ LOC108348069;Heatr1;Heatr1l1/ Bysl/ Glul/ Rsl24d1/ Nol11/ Gar1	58
GO:0022613	BP	GO:0022613	ribonucleoprotein complex biogenesis	76/12887/12373+0.000007830007762					Prmt7/ Srpk1/ Rpf2/ Ddx39b/ Sfswap/ Nufip1/ Rrp1b/ Rrp1/ Pwp2/ Eif3b/ Lsg1/ Rrp15/ Aatf/ Luc7l3/ Tsr1/ Rrn3/ Utp3/ Dcaf13/ Utp23/ Pes1/ Prpf39/ Utp25/ Pa2g4/ Surf6/ Nol10/ Shq1/ Srsf6/ Nop56/ Strap/ Rrs1/ Nmd3/ Ftsj3/ Nol9/ Mrps2/ Prmt5/ Nop14/ Ddx10/ Ruvbl1/ Mbnl1/ Srfbp1/ Heatr3/ Xrcc5/ Gtpbp4/ Mphosph10/ Nop58/ Wdr12/ Brix1/ Smn1/ Nop2/ Rrp8/ Wdr3/ Khdc4/ Utp4/ Ppan/ Cpsf7/ Prpf19/ Bop1/ Nob1/ Rrp7a/ Ddx18/ Mettl17/ Wdr43/ Naf1/ Wdr36/ Mrm1/ Usp36/ Mrm3/ Nudt21/ Ddx21/ Nol8;NEWGENE_1309551/ LOC108348069;Heatr1;Heatr1l1/ Bysl/ Glul/ Rsl24d1/ Nol11/ Gar1	76
GO:0006364	BP	GO:0006364	rRNA processing	43/12886/12373e-		07	0.000621406052		Rpf2/ Rrp1b/ Rrp1/ Pwp2/ Rrp15/ Tsr1/ Utp3/ Dcaf13/ Utp23/ Pes1/ Utp25/ Pa2g4/ Nol10/ Shq1/ Rrs1/ Ftsj3/ Nol9/ Nop14/ Ddx10/ Srfbp1/ Gtpbp4/ Mphosph10/ Wdr12/ Nop2/ Rrp8/ Wdr3/ Utp4/ Bop1/ Nob1/ Rrp7a/ Ddx18/ Wdr43/ Naf1/ Wdr36/ Mrm1/ Usp36/ Mrm3/ Ddx21/ Nol8;NEWGENE_1309551/ LOC108348069;Heatr1;Heatr1l1/ Bysl/ Nol11/ Gar1	43
GO:0016072	BP	GO:0016072	rRNA metabolic process	43/12895/12373e-		06	0.001810981724		Rpf2/ Rrp1b/ Rrp1/ Pwp2/ Rrp15/ Tsr1/ Utp3/ Dcaf13/ Utp23/ Pes1/ Utp25/ Pa2g4/ Nol10/ Shq1/ Rrs1/ Ftsj3/ Nol9/ Nop14/ Ddx10/ Srfbp1/ Gtpbp4/ Mphosph10/ Wdr12/ Nop2/ Rrp8/ Wdr3/ Utp4/ Bop1/ Nob1/ Rrp7a/ Ddx18/ Wdr43/ Naf1/ Wdr36/ Mrm1/ Usp36/ Mrm3/ Ddx21/ Nol8;NEWGENE_1309551/ LOC108348069;Heatr1;Heatr1l1/ Bysl/ Nol11/ Gar1	43
GO:0032922	BP	GO:0032922	circadian regulation of gene expression	21/12887/12373e-		06	0.002510524497		Hnf1b/ Ogt/ Ahr/ Ppargc1a/ Ppp1cb/ Cry1/ Usp2/ Per1/ Cry2/ Ncoa2/ Nr1d1/ Nampt/ Noct/ Prmt5/ Arntl/ Zfhx3/ Per3/ Egr1/ Id1/ Csnk1d/ Top1	21
GO:0034660	BP	GO:0034660	ncRNA metabolic process	72/12898/12373e-		06	0.004060539546		Rpf2/ Rrp1b/ Rrp1/ Pwp2/ Rrp15/ Tsr1/ Utp3/ Dcaf13/ Utp23/ Pes1/ Utp25/ Rtcb/ Pa2g4/ Nol10/ Shq1/ Yars1/ Rrs1/ Rars1/ Dus4l/ Thumpd2/ Ftsj3/ Nol9/ Trmt61a;LOC103692719/ Nop14/ Ddx10/ Pus3/ Farsb/ Trmt11/ Srfbp1/ Gtpbp4/ Mphosph10/ Wdr12/ Toe1/ Ctu1/ Nop2/ Nolc1/ Rrp8/ Kars/ Wdr3/ Tut1/ Utp4/ Cars/ Bop1/ Nob1/ Rrp7a/ Tars3/ Ddx18/ Mars1/ Ints5/ Wdr43/ Naf1/ Wdr36/ Mrm1/ Hars1/ Usp36/ LOC685619/ Mrm3/ Pus1/ Rpp38/ Ddx21/ Nol8;NEWGENE_1309551/ LOC108348069;Heatr1;Heatr1l1/ Mettl1/ Bysl/ Mars2;LOC100911305/ Dus3l/ Ddx17/ Nol11/ Pus10/ Tsen2/ Qtrt2/ Gar1	72

```
selectedGenes <- selectedGenes & sign(all_res$F0_log2FC)==sign(all_res$F1_log2FC)
selectedGenes[is.na(selectedGenes)] <- FALSE
table(selectedGenes)
```

```
## selectedGenes
## FALSE  TRUE
## 14376  1304
```

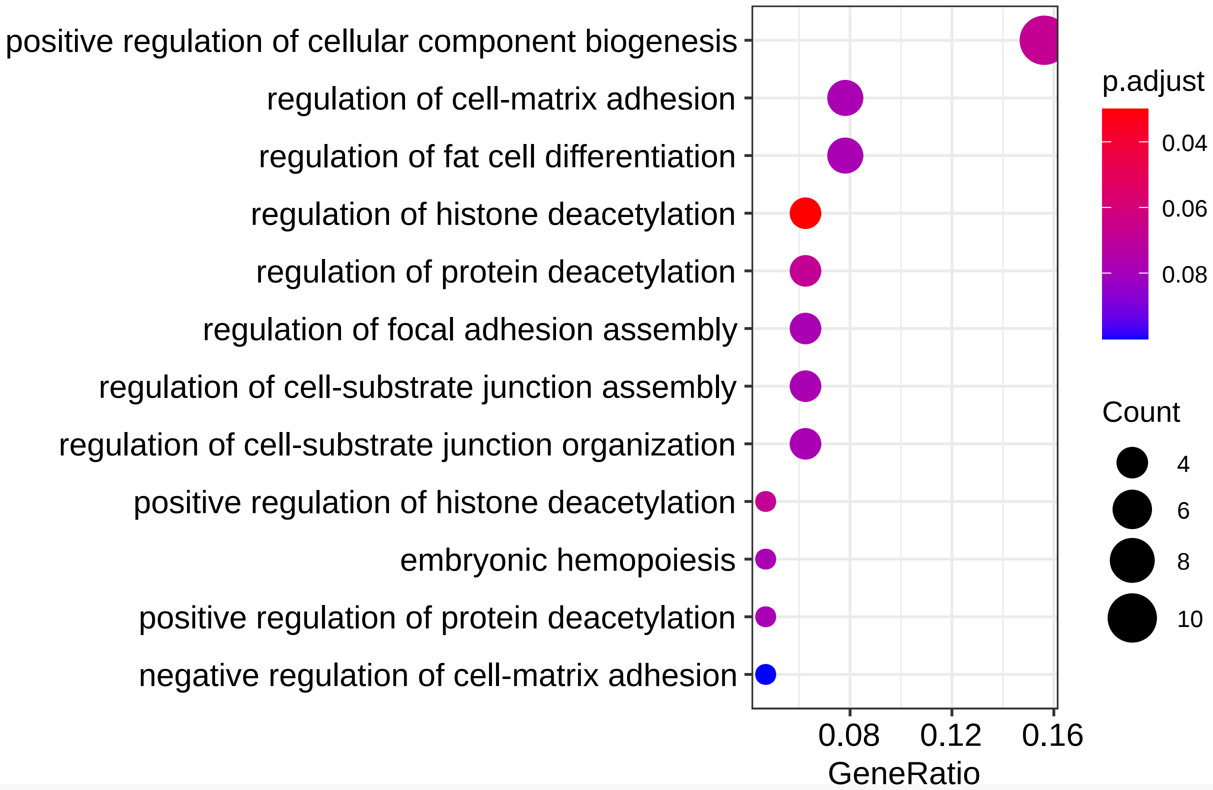
```r
selectedGenes <- selectedGenes & all_res$AvsC_FDR <0.2 & all_res$AvsD_FDR <0.1
selectedGenes[is.na(selectedGenes)] <- FALSE
print(table(selectedGenes))
```

```
## selectedGenes
## FALSE  TRUE
## 15611    69
```

```r
knitr::kable(head(all_res[selectedGenes, ]))
```

	baseMean	F0_log2FC	F0_FDR	F1_log2FC	F1_FDR	AvsC_log2FC	AvsC_FDR	AvsD_log2FC	AvsD_FDR	gene
ENSRNOG00000016622	744.5328	-0.2800426	0.0000149	-0.0173940	0.9265521	-0.4935879	0.0457818	-0.3473848	0.0887203	Apex2
ENSRNOG00000008246	605.2422	-0.2219823	0.0012845	-0.0320448	0.8504547	-0.3837241	0.1956533	-0.5443889	0.0023141	Dse
ENSRNOG00000011233	400.1715	-0.6228360	0.0000047	-0.2107163	0.4716270	-0.9831938	0.0998556	-0.7744223	0.0878276	RGD1562310
ENSRNOG00000011891	1691.5009	0.3153347	0.0000972	0.1039722	0.5256438	0.5506939	0.0998556	0.4963029	0.0393689	Sik1
ENSRNOG00000015162	2778.0068	-0.2110136	0.0485317	-0.1206075	0.5230320	-0.5886480	0.1289949	-0.5984118	0.0220845	Rapgef4
ENSRNOG00000017622	2457.7067	-0.3794000	0.0000000	-0.1368098	0.3169068	-0.5916710	0.0045183	-0.5944619	0.0004003	Pcyt1a

```r
de_genes <- row.names(all_res[selectedGenes, ])
de_genes <- anno[de_genes, "ensembl_gene_id"]
universe <- anno[row.names(all_res), "ensembl_gene_id"]
go <- enrichGO(de_genes,
               universe = universe,
               OrgDb = "org.Rn.eg.db",
               keyType = "ENSEMBL",
               ont = "ALL",
               pvalueCutoff = 0.1,
               qvalueCutoff = 0.2,
               pAdjustMethod = "fdr",
               minGSSize = 10,
               maxGSSize = 500)

dotplot(go, showCategory=30)
```



```
tmp <- data.frame(go)
tmp$geneID <- sapply(strsplit(tmp$geneID, "/"),
                    function(x) paste0(anno[anno$ensembl_gene_id%in%x, 2],
                                       collapse="/ "))
knitr::kable(head(tmp))
```

	ONTOLOGY	ID	Description	GeneRatio	BgRatio	pvalue	p.adjust	qvalue	geneID	Count
GO:0031063	BP	GO:0031063	regulation of histone deacetylation	4/64	30/12373	0.0000161	0.0300108	0.0277031	Lpin1/ Jdp2/ Vegfa/ Ski	4
GO:0044089	BP	GO:0044089	positive regulation of cellular component biogenesis	10/64	468/12373	0.0001317	0.0680612	0.0628275	Phldb2/ Kit/ Vps4b/ Ahr/ Wasl/ Sdc4/ Vegfa/ Cdc42ep4/ Arhgef10l/	10
GO:0031065	BP	GO:0031065	positive regulation of histone deacetylation	3/64	20/12373	0.0001413	0.0680612	0.0628275	Lpin1/ Jdp2/ Vegfa	3
GO:0090311	BP	GO:0090311	regulation of protein deacetylation	4/64	52/12373	0.0001463	0.0680612	0.0628275	Lpin1/ Jdp2/ Vegfa/ Ski	4
GO:0051893	BP	GO:0051893	regulation of focal adhesion assembly	4/64	63/12373	0.0003085	0.0777950	0.0718128	Phldb2/ Sdc4/ Vegfa/	4
GO:0090109	BP	GO:0090109	regulation of cell-substrate junction assembly	4/64	63/12373	0.0003085	0.0777950	0.0718128	Phldb2/ Sdc4/ Vegfa/	4

```
sessionInfo()
```

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: CentOS Linux 7 (Core)
##
## Matrix products: default
## BLAS/LAPACK: /usr/prog/OpenBLAS/0.2.20-GCC-6.4.0-2.28/lib/libopenblas_haswellp-r0.2.20.so
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C               LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8    LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C             LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
```

```
## [1] parallel  stats4    stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] readxl_1.3.1                pheatmap_1.0.12             reshape2_1.4.4              knitr_1.33
##  [5] viridis_0.6.1               viridisLite_0.4.0           UpSetR_1.4.0                forcats_0.5.1
##  [9] stringr_1.4.0               dplyr_1.0.7                 purrr_0.3.4                 readr_2.0.1
## [13] tidyr_1.1.3                 tibble_3.1.3                tidyverse_1.3.1             org.Rn.eg.db_3.13.0
## [17] AnnotationDbi_1.54.1        clusterProfiler_4.0.4       BisqueRNA_1.0.5             sp_1.4-5
## [21] SeuratObject_4.1.0          Seurat_4.0.3                goseq_1.44.0                geneLenDataBase_1.28.0
## [25] BiasedUrn_1.07              biomaRt_2.48.3              ggpubr_0.4.0                EnhancedVolcano_1.10.0
## [29] ggrepel_0.9.1               GGally_2.1.2                ggplot2_3.3.5               DESeq2_1.32.0
## [33] SummarizedExperiment_1.22.0 Biobase_2.52.0              MatrixGenerics_1.4.3        matrixStats_0.61.0
## [37] GenomicRanges_1.44.0        GenomeInfoDb_1.28.1         IRanges_2.26.0              S4Vectors_0.30.0
## [41] BiocGenerics_0.38.0
##
## loaded via a namespace (and not attached):
##   [1] rappdirs_0.3.3           rtracklayer_1.52.1       scattermore_0.8          bit64_4.0.5
##   [5] irlba_2.3.3              DelayedArray_0.18.0      rpart_4.1.16             data.table_1.14.0
##   [9] KEGGREST_1.32.0          RCurl_1.98-1.4           generics_0.1.0           GenomicFeatures_1.44.1
##  [13] cowplot_1.1.1            RSQLite_2.2.7            shadowtext_0.0.8         RANN_2.6.1
##  [17] future_1.27.0            tzdb_0.1.2               bit_4.0.4                enrichplot_1.12.2
##  [21] lubridate_1.7.10         spatstat.data_2.2-0      httpuv_1.6.2             xml2_1.3.2
##  [25] assertthat_0.2.1         xfun_0.31                hms_1.1.0                evaluate_0.14
##  [29] promises_1.2.0.1         fansi_0.5.0              restfulr_0.0.13          progress_1.2.2
##  [33] dbplyr_2.1.1             igraph_1.3.4             DBI_1.1.1                geneplotter_1.70.0
##  [37] htmlwidgets_1.5.3        reshape_0.8.8            spatstat.geom_2.4-0      ellipsis_0.3.2
##  [41] backports_1.2.1          annotate_1.70.0          deldir_1.0-6             vctrs_0.3.8
##  [45] ROCR_1.0-11              abind_1.4-5              cachem_1.0.6             withr_2.4.2
##  [49] ggforce_0.3.3            progressr_0.10.1         sctransform_0.3.2        GenomicAlignments_1.28.0
##  [53] treeio_1.16.2            prettyunits_1.1.1        goftest_1.2-3            cluster_2.1.2
##  [57] DOSE_3.18.2              ape_5.5                  lazyeval_0.2.2           crayon_1.4.1
##  [61] genefilter_1.74.0        labeling_0.4.2           pkgconfig_2.0.3          tweenr_1.0.2
##  [65] nlme_3.1-152             vipor_0.4.5              rlang_0.4.11             globals_0.15.1
##  [69] miniUI_0.1.1.1           lifecycle_1.0.0          downloader_0.4           filelock_1.0.2
##  [73] extrafontdb_1.0          BiocFileCache_2.0.0      modelr_0.1.8             ggrastr_0.2.3
##  [77] cellranger_1.1.0         polyclip_1.10-0          lmtest_0.9-40            Matrix_1.3-4
##  [81] aplot_0.0.6              carData_3.0-4            zoo_1.8-9                reprex_2.0.1
##  [85] beeswarm_0.4.0           ggridges_0.5.3           png_0.1-7                rjson_0.2.20
##  [89] bitops_1.0-7             KernSmooth_2.23-20       Biostrings_2.60.2        blob_1.2.2
##  [93] qvalue_2.24.0            spatstat.random_2.2-0    parallelly_1.32.1        rstatix_0.7.0
##  [97] ggsignif_0.6.2           scales_1.1.1             memoise_2.0.0            magrittr_2.0.1
## [101] plyr_1.8.6               ica_1.0-3                zlibbioc_1.38.0          compiler_4.1.1
## [105] scatterpie_0.1.7         BiocIO_1.2.0             RColorBrewer_1.1-2       ash_1.0-15
## [109] fitdistrplus_1.1-8       Rsamtools_2.8.0          cli_3.0.1                XVector_0.32.0
## [113] listenv_0.8.0            pbapply_1.5-0            patchwork_1.1.1          MASS_7.3-54
## [117] mgcv_1.8-36              tidyselect_1.1.1         stringi_1.7.3            highr_0.9
## [121] proj4_1.0-10.1           yaml_2.2.1               GOSemSim_2.18.1          locfit_1.5-9.4
## [125] grid_4.1.1               fastmatch_1.1-3          tools_4.1.1              future.apply_1.9.0
## [129] rio_0.5.27               rstudioapi_0.13          foreign_0.8-81           gridExtra_2.3
## [133] farver_2.1.0             Rtsne_0.15               ggraph_2.0.5             digest_0.6.27
## [137] rvcheck_0.1.8            BiocManager_1.30.16      rgeos_0.5-9              shiny_1.6.0
## [141] Rcpp_1.0.7               car_3.0-11               broom_0.7.9              ggalt_0.4.0
## [145] later_1.3.0              RcppAnnoy_0.0.19         httr_1.4.2               colorspace_2.0-2
## [149] rvest_1.0.1              fs_1.5.0                 tensor_1.5               XML_3.99-0.7
## [153] reticulate_1.20          splines_4.1.1            uwot_0.1.10              spatstat.utils_2.3-1
## [157] tidytree_0.3.4           graphlayouts_0.7.1       plotly_4.9.4.1           xtable_1.8-4
## [161] jsonlite_1.7.2           ggtree_3.0.3             tidygraph_1.2.0          ggfun_0.0.3
## [165] R6_2.5.1                 mime_0.11                pillar_1.6.2             htmltools_0.5.1.1
## [169] glue_1.4.2               fastmap_1.1.0            BiocParallel_1.26.1      codetools_0.2-18
## [173] maps_3.3.0             fgsea_1.18.0           utf8_1.2.2             spatstat.sparse_2.1-1
## [177] lattice_0.20-44        curl_4.3.2             ggbeeswarm_0.6.0       leiden_0.4.2
## [181] zip_2.2.0              GO.db_3.13.0           openxlsx_4.2.4         Rttf2pt1_1.3.9
## [185] survival_3.2-12        rmarkdown_2.14         munsell_0.5.0          DO.db_2.9
## [189] GenomeInfoDbData_1.2.6 haven_2.4.2            gtable_0.3.0           spatstat.core_2.4-4
## [193] extrafont_0.17
```