

# Common Marmoset Gut Microbiome Profiles in Health and Intestinal Disease

Alex Sheh

September 21, 2020

**This is the 3rd part of the RNAseq analysis picking up after FASTQ files were processed and aligned**

We load rdata file with the feature counts and libraries.

```
load("ibd_cj_feature_counts.RData")
load("ibd_hs_feature_counts.RData")

# for ML algorithms
library(Rsubread)
library(edgeR)

## Loading required package: limma

library(gplots)

##
## Attaching package: 'gplots'

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

library(org.Hs.eg.db)

## Loading required package: AnnotationDbi

## Loading required package: stats4

## Loading required package: BiocGenerics

## Loading required package: parallel

##
## Attaching package: 'BiocGenerics'
```

```

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

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

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

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

## Loading required package: Biobase

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

## Loading required package: IRanges

## Loading required package: S4Vectors

##
## Attaching package: 'S4Vectors'

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

## The following object is masked from 'package:base':
##
##   expand.grid

##
## Attaching package: 'IRanges'

## The following object is masked from 'package:grDevices':
##
##   windows

```

```

##



library(AnnotationDbi)
library(GO.db)

##



library(mygene)

## Loading required package: GenomicFeatures

## Loading required package: GenomeInfoDb

## Loading required package: GenomicRanges

library(topGO)

## Loading required package: graph

## Loading required package: SparseM

## 
## Attaching package: 'SparseM'

## The following object is masked from 'package:base':
## 
##      backsolve

## 
## groupGOTerms:      GOBPTerm, GOMFTerm, GOCCTerm environments built.

## 
## Attaching package: 'topGO'

## The following object is masked from 'package:GenomicFeatures':
## 
##      genes

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

data("geneList")
library(Rgraphviz)

## Loading required package: grid

## 
## Attaching package: 'grid'

```

```

## The following object is masked from 'package:topGO':
##
##      depth

##
## Attaching package: 'Rgraphviz'

## The following objects are masked from 'package:IRanges':
##
##      from, to

## The following objects are masked from 'package:S4Vectors':
##
##      from, to

library(ggplot2)
library(colorspace)
library(ggVennDiagram)

sessionInfo()

## R version 3.6.3 (2020-02-29)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 18363)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] grid      parallel   stats4    stats     graphics  grDevices utils
## [8] datasets  methods   base
##
## other attached packages:
## [1] ggVennDiagram_0.3      colorspace_1.4-1      ggplot2_3.3.2
## [4] Rgraphviz_2.30.0       topGO_2.38.1       SparseM_1.78
## [7] graph_1.64.0          mygene_1.22.0       GenomicFeatures_1.38.2
## [10] GenomicRanges_1.38.0   GenomeInfoDb_1.22.1  GO.db_3.10.0
## [13] org.Hs.eg.db_3.10.0   AnnotationDbi_1.48.0 IRanges_2.20.2
## [16] S4Vectors_0.24.4       Biobase_2.46.0       BiocGenerics_0.32.0
## [19] gplots_3.0.4           edgeR_3.28.1       limma_3.42.2
## [22] Rsubread_2.0.1
##
## loaded via a namespace (and not attached):
## [1] class_7.3-15            ellipsis_0.3.1
## [3] htmlTable_2.0.1         futile.logger_1.4.3
## [5] XVector_0.26.0          base64enc_0.1-3
## [7] rstudioapi_0.11         bit64_0.9-7

```

```

## [9] sqldf_0.4-11                  splines_3.6.3
## [11] knitr_1.29                    Formula_1.2-3
## [13] jsonlite_1.7.0                Rsamtools_2.2.3
## [15] cluster_2.1.0                 dbplyr_1.4.4
## [17] png_0.1-7                     compiler_3.6.3
## [19] httr_1.4.2                     backports_1.1.7
## [21] assertthat_0.2.1              Matrix_1.2-18
## [23] formatR_1.7                   acepack_1.4.1
## [25] htmltools_0.5.0              prettyunits_1.1.1
## [27] tools_3.6.3                   gtable_0.3.0
## [29] glue_1.4.1                     GenomeInfoDbData_1.2.2
## [31] dplyr_1.0.0                   rappdirs_0.3.1
## [33] Rcpp_1.0.5                     vctrs_0.3.1
## [35] Biostrings_2.54.0             gdata_2.18.0
## [37] rtracklayer_1.46.0            xfun_0.16
## [39] stringr_1.4.0                proto_1.0.0
## [41] lifecycle_0.2.0               gtools_3.8.2
## [43] XML_3.99-0.3                 zlibbioc_1.32.0
## [45] scales_1.1.1                 hms_0.5.3
## [47] SummarizedExperiment_1.16.1   lambda.r_1.2.4
## [49] RColorBrewer_1.1-2            yaml_2.2.1
## [51] curl_4.3                      memoise_1.1.0
## [53] gridExtra_2.3                 biomaRt_2.42.1
## [55] rpart_4.1-15                  latticeExtra_0.6-29
## [57] stringi_1.4.6                RSQLite_2.2.0
## [59] e1071_1.7-3                  checkmate_2.0.0
## [61] caTools_1.18.0                BiocParallel_1.20.1
## [63] chron_2.3-55                 rlang_0.4.7
## [65] pkgconfig_2.0.3              matrixStats_0.56.0
## [67] bitops_1.0-6                  evaluate_0.14
## [69] lattice_0.20-38              sf_0.9-5
## [71] purrrr_0.3.4                 GenomicAlignments_1.22.1
## [73] htmlwidgets_1.5.1            bit_1.1-15.2
## [75] tidyselect_1.1.0              plyr_1.8.6
## [77] magrittr_1.5                  R6_2.4.1
## [79] generics_0.0.2               Hmisc_4.4-0
## [81] DelayedArray_0.12.3           DBI_1.1.0
## [83] withr_2.2.0                  gsubfn_0.7
## [85] pillar_1.4.6                  foreign_0.8-75
## [87] units_0.6-7                  survival_3.1-8
## [89] RCurl_1.98-1.2                nnet_7.3-12
## [91] tibble_3.0.3                  crayon_1.3.4
## [93] futile.options_1.0.1          KernSmooth_2.23-16
## [95] BiocFileCache_1.10.2          rmarkdown_2.3
## [97] jpeg_0.1-8.1                  progress_1.2.2
## [99] locfit_1.5-9.4                data.table_1.12.8
## [101] blob_1.2.1                   classInt_0.4-3
## [103] digest_0.6.25                VennDiagram_1.6.20
## [105] openssl_1.4.2                munsell_0.5.0
## [107] askpass_1.1

#####function getgeneid
getgeneid<-function(strGO,isde){
  require(org.Hs.eg.db)

```

```

require(mygene)
require(AnnotationDbi)
x <- org.Hs.egGO2ALLEGS
Rkeys(x) <- strGO
EG <- mappedLkeys(x)

is.deh.keep<-isde@.Data!=0
is.deh.kept <- as.matrix(isde[is.deh.keep])
rnames <- rownames(isde)
rnames <- rnames[is.deh.keep]
rownames(is.deh.kept)<-rnames
isde_G0 <- intersect(EG, rownames(is.deh.kept))
GOI<-as.matrix(is.deh.kept[isde_G0,])
return(getGenes(rownames(GOI), fields = c("symbol","name","summary")))
}

```

## Compare IBD cases

### Supplementary Figure 8a

```

#tabulate
ibd <- cbind(fc_2$counts, fc_4$counts, fc_8$counts, fc_6$counts, fc_10$counts, fc_15$counts)
#write.table(ibd, file="20200824_ibd_3v3_raw_feature_counts.txt")
g_ibd <- c("ibd","ibd","ibd","non_ibd","non_ibd","non_ibd")
#trim lowly expressed exons
y <- DGEList(counts = ibd[,1:6], group = g_ibd)

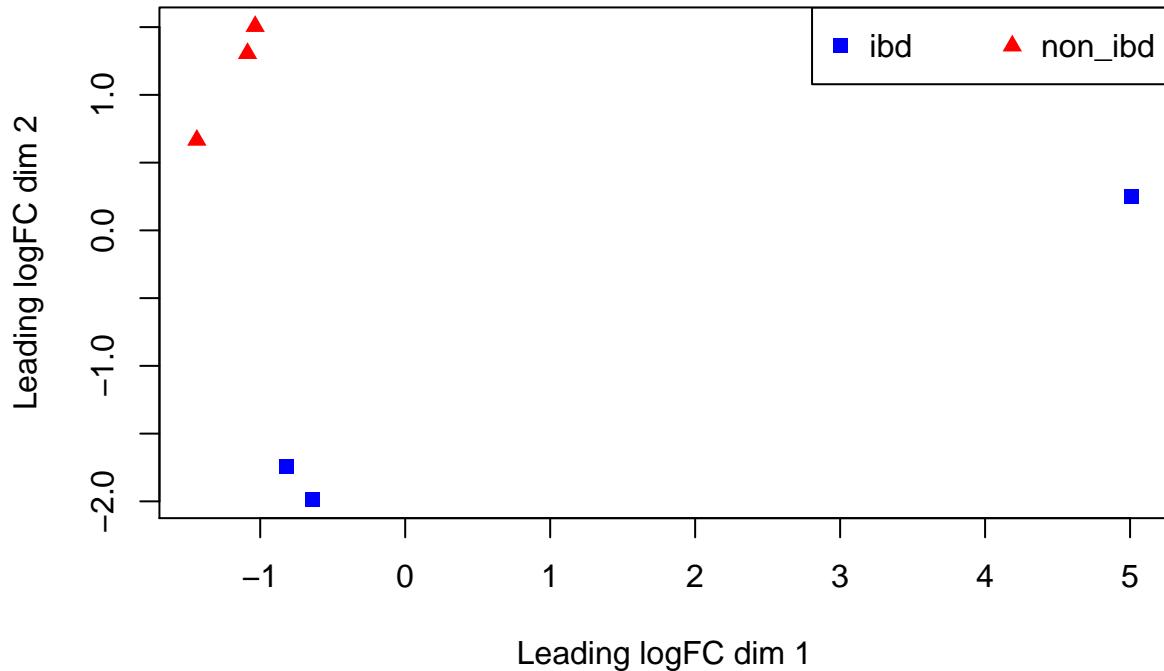
#counts per million have to have counts greater than X per million
# and be found in at least Y samples
cpm(10, mean(y$samples$lib.size))
#this value is used in the rowsum inequality
keep <-rowSums(cpm(y)>2) >= 2
y <- y[keep,]

#TMM normalization
y.norm <-calcNormFactors(y, method = "TMM")
y.norm$samples

# #plot MA plot - mean and difference plot based on the first sample.
# plotMD(cpm(y.norm, log=TRUE), column=1)
# abline(h=0, col="red", lty=2, lwd=2)

#data exploration of dataset
plotMDS(y.norm,pch = c(15,15,15,17,17,17), col = c("blue", "blue","blue", "red","red","red"))
legend("topright", legend=c("ibd", "non_ibd"), pch=c(15,17), col=c("blue","red"), ncol=2)

```



```
# dev.copy(png, paste0("ibd_3v3_plotMDS.png"))
# dev.off()
```

## Ordination of IBD data in the jejunum

After evaluating PCA we decided to exclude one sample for subsequent analysis

### Supplementary Figure 8b

```
#tabulate without sample 2
ibd2 <- cbind(fc_4$counts, fc_8$counts, fc_6$counts, fc_10$counts, fc_15$counts)
g_ibd2 <- c("ibd","ibd","non_ibd","non_ibd","non_ibd")

#trim lowly expressed exons
y2 <- DGEList(counts = ibd2[,1:5], group = g_ibd2)

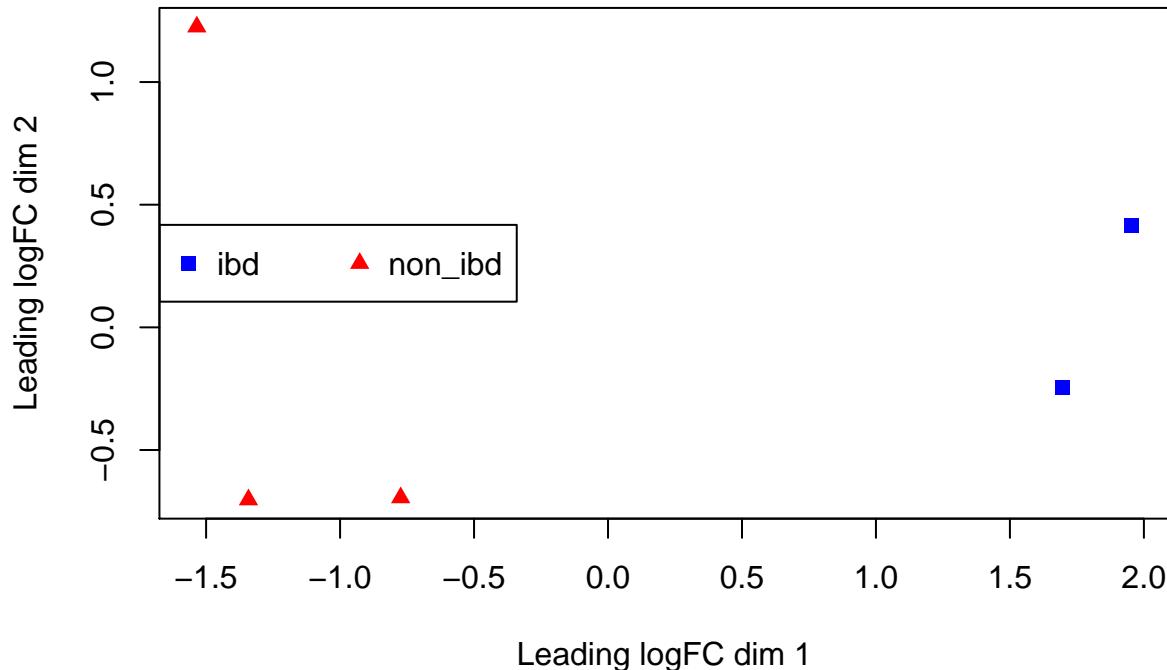
#counts per million have to have counts greater than X per million
#and be found in at least Y samples
cpm(10, mean(y2$samples$lib.size))
#this value is used in the rowsum inequality
keep2 <-rowSums(cpm(y2)>1.6) >= 2
y2 <- y2[keep2,]
```

```

#TMM normalization
y2.norm <- calcNormFactors(y2, method = "TMM")
y2.norm$samples

#data exploration SUPPLEMENTARY FIGURE 8b
plotMDS(y2.norm, pch = c(15,15,17,17,17), col = c("blue","blue", "red", "red", "red"))
legend("left", legend=c("ibd", "non_ibd"), pch=c(15,17), col=c("blue", "red"), ncol=2)

```

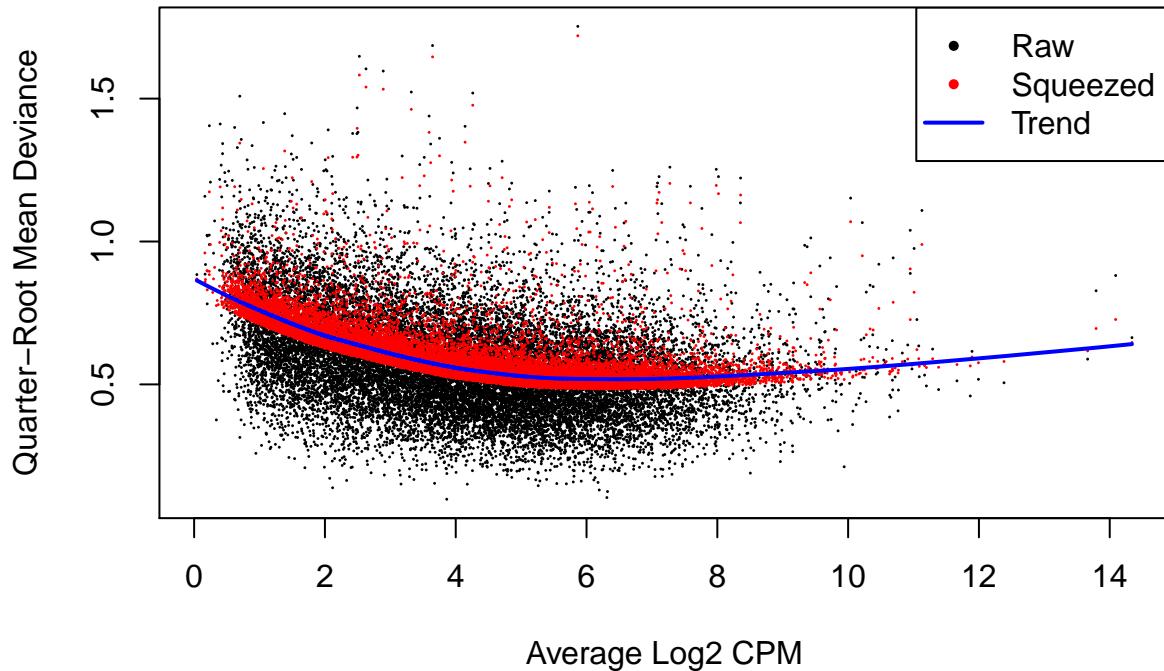


```

# dev.copy(png, paste0("ibd_3v2_plotMDS.png"))
# dev.off()

#Estimate common dispersion
design2 <- model.matrix(~ 0 + g_ibd2)
colnames(design2) <- c("ibd", "non_ibd")
y2.norm <- estimateCommonDisp(y2.norm, design2, verbose = TRUE, robust= TRUE)
fit2 <- glmQLFit(y2.norm, design2, robust=TRUE)
plotQLDisp(fit2)

```



```

# outliers from the mean-NB dispersion trend. Outliers are marked by small prior.df values:
o2 <- order(fit2$df.prior)
y2.norm$counts[o2[1:6],] #genes tag used in the tutorial but our labels had names
con2 <- makeContrasts(non_ibd - ibd, levels=design2)
res2 <- glmQLFTest(fit2, contrast=con2)
topTags(res2)

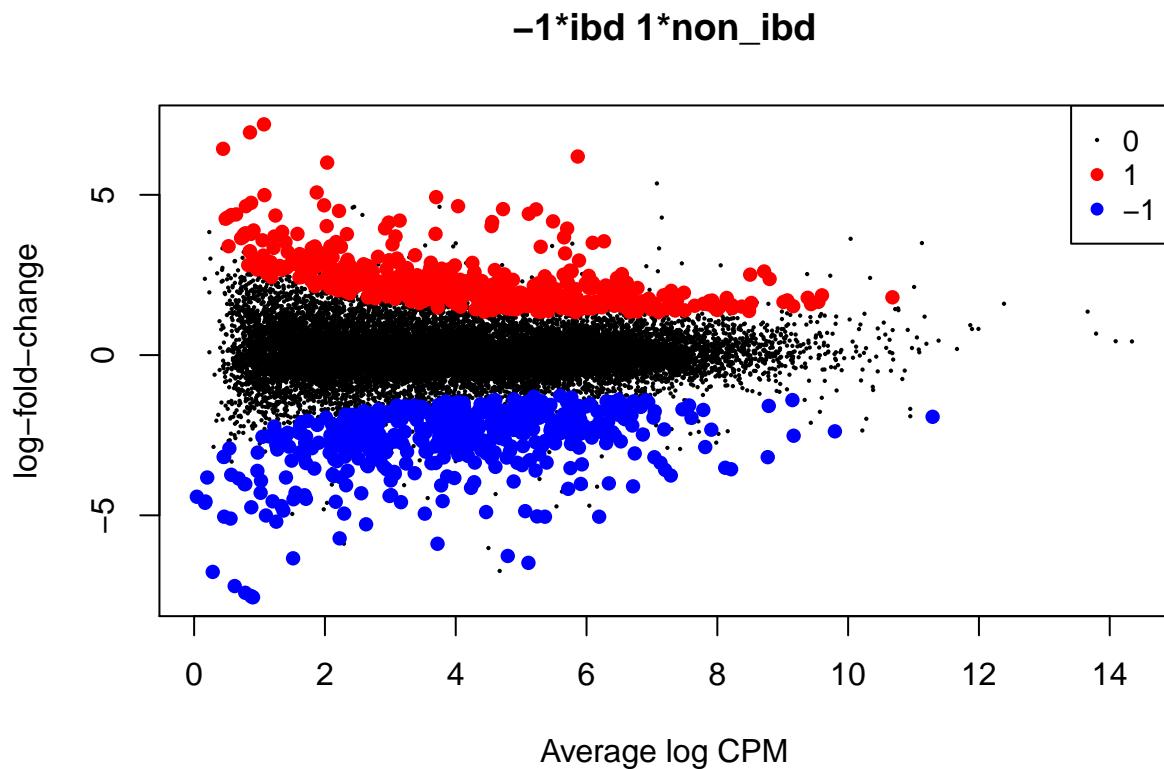
is.de2 <- decideTestsDGE(res2, p.value=0.05)
# summary(is.de2)
# plotSmear(res2, de.tags=rownames(res2)[is.de2!=0])
# dev.copy(png, paste0("ibd_3v2_plotsmear.png"))
# dev.off()

#supplementary table 7
write.table(topTags(res2, n=100000)$table, file="supplementary_table_7.txt")

#imposing the 1.5 fold change
tr2 <- glmTreat(fit2, contrast = con2, lfc = log2(1.5))
topTags(tr2)

is.de.fc2 <- decideTestsDGE(tr2)
summary(is.de.fc2)
plotMD(tr2, status=is.de.fc2, values=c(1,-1), col=c("red","blue"), legend="topright")

```



```

# dev.copy(png, paste0("ibd_3v2_plotMD15.png"))
# dev.off()

# 4C Venn diagram IBD -----
ibd_toptags_3v2 <- topTags(res2, n = 2000, p.value = 0.05)
ibd_toptags_3v2 <- as.data.frame(ibd_toptags_3v2[,c(1,5)])
ibd_toptags_3v2 <- ibd_toptags_3v2[order(ibd_toptags_3v2$logFC),]
ibd_toptags_genes_3v2 <- rownames(ibd_toptags_3v2)
ibd_genes_3v2 <- rownames(y2.norm$pseudo.counts)
ibd_commongenes_3v2 <- setdiff(ibd_genes_3v2,ibd_toptags_genes_3v2)

ibd_toptags_3v2_up <- ibd_toptags_3v2$logFC>0
ibd_toptags_3v2_up <- ibd_toptags_3v2[ibd_toptags_3v2_up,]
ibd_toptags_3v2_dn <- ibd_toptags_3v2$logFC<0
ibd_toptags_3v2_dn <- ibd_toptags_3v2[ibd_toptags_3v2_dn,]
ibd_genes_3v2_up <- rownames(ibd_toptags_3v2_up)
ibd_genes_3v2_dn <- rownames(ibd_toptags_3v2_dn)
ibd_genes_3v2_com_up <- append(ibd_genes_3v2_up,ibd_commongenes_3v2)
ibd_genes_3v2_com_dn <- append(ibd_genes_3v2_dn,ibd_commongenes_3v2)

# draw venn
ibd_3v2_venn <- list(NonIBD=ibd_genes_3v2_com_up, IBD=ibd_genes_3v2_com_dn)
ibd_3v2_venn_plot <- ggVennDiagram(ibd_3v2_venn,color="black",
                                     size=1, category.names = c("Non-IBD", "IBD")) +
  scale_fill_gradient(low="lightyellow",high = "red") +

```

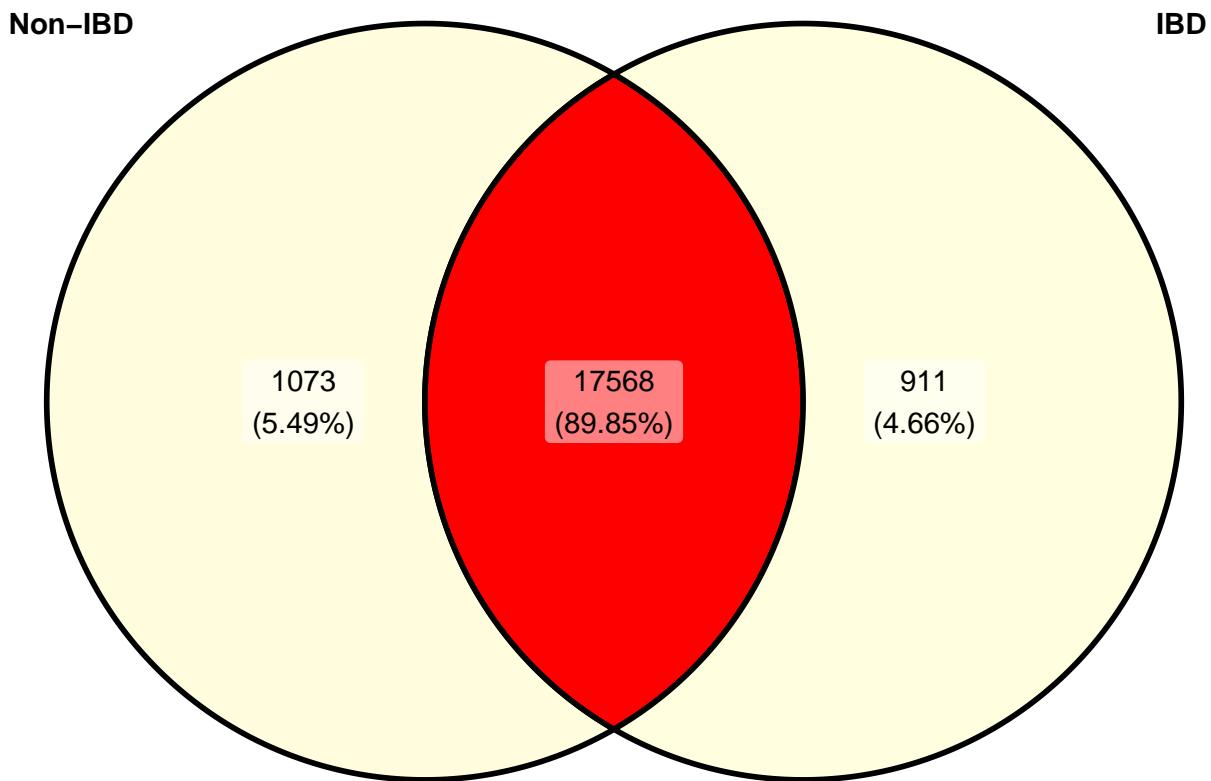
```

theme(legend.position = "none")

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

ibd_3v2_venn_plot

```



```

# ggsave(
#   "ibd_3v2_venn.png",
#   ibd_3v2_venn_plot,
#   width = 8.5,
#   height = 6,
#   dpi = 1200
# )

# 5 3 vs 2 ibd using annotation for GO -----
#tabulate
ibd_h2 <- cbind(fc_4h$counts, fc_8h$counts, fc_6h$counts, fc_10h$counts, fc_15h$counts)
g_ibdh2 <- c("ibd", "ibd", "non_ibd", "non_ibd", "non_ibd")
#trim lowly expressed exons
yh2 <- DGEList(counts = ibd_h2[,1:5], group = g_ibdh2)
#match symbols to numbers
Symbol <- mapIds(org.Hs.eg.db, rownames(yh2), keytype = "ENTREZID", column="SYMBOL")

```

```

## 'select()' returned 1:1 mapping between keys and columns

#remove the ones that were NA
yh2 <- yh2[!is.na(Symbol), ]
a<-dim(yh2)
yh2 <- yh2[1:a[1]-1, ]

#counts per million have to have counts greater than X per million
# and be found in at least Y samples
cpn(10, mean(yh2$samples$lib.size))
#this value is used in the rowsum inequality
keeph2 <-rowSums(cpn(yh2)>1.6) >= 2
yh2 <- yh2[keeph2,]

#TMM normalization
yh2.norm <-calcNormFactors(yh2, method = "TMM")
yh2.norm$samples

#data exploration of stricture
# plotMDS(yh2.norm, pch = c(15,15,17,17,17), col = c("blue","blue","red","red","red"))
# legend("right", legend=c("ibd", "non_ibd"), pch=c(15,17), col=c("blue","red"), ncol=2)
# dev.copy(png, paste0("ibd_3v2h_plotMDS.png"))
# dev.off()

#Estimate common dispersion
design_h2 <- model.matrix(~ 0 + g_ibdh2)
colnames(design_h2) <- c("ibd", "non_ibd")
yh2.norm <- estimateCommonDisp(yh2.norm, design_h2, verbose = TRUE, robust= TRUE)

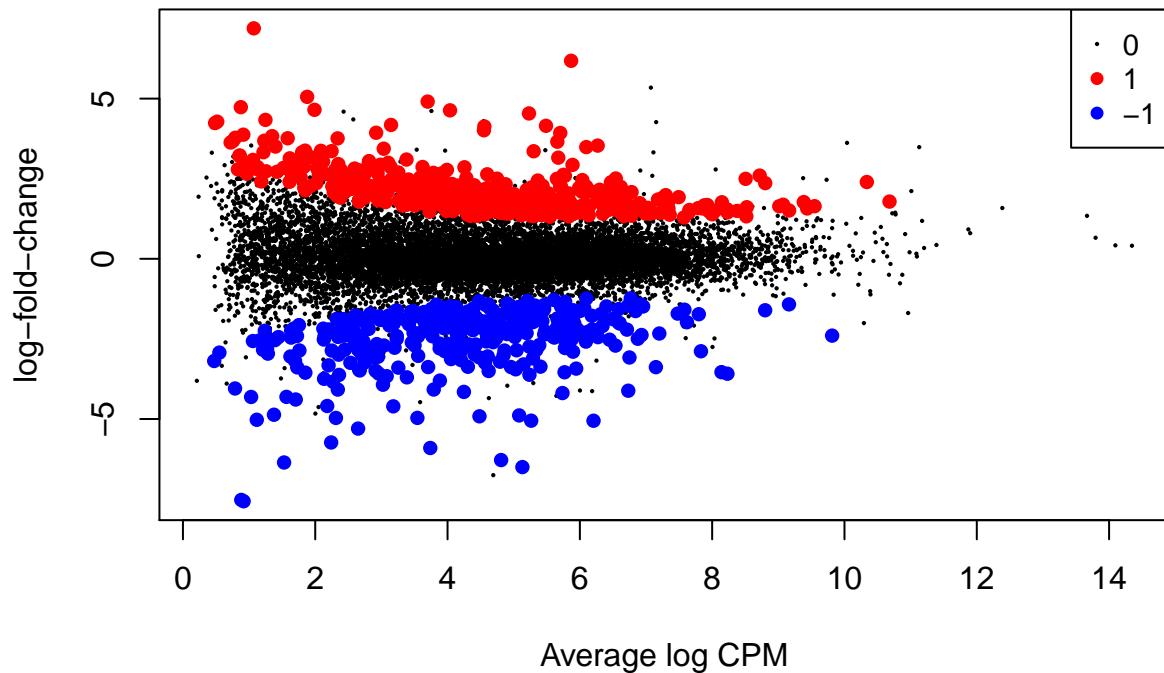
fit_h2 <- glmQLFit(yh2.norm, design_h2, robust=TRUE)
head(fit_h2$coefficients)
# plotQLDisp(fit_h2)
# dev.copy(png, paste0("ibd_3v2h_plotQLDisp.png"))
# dev.off()
# summary(fit_h2$df.prior)

# outliers from the mean-NB dispersion trend. Outliers are marked by small prior.df values:
o_h2 <- order(fit_h2$df.prior)
yh2.norm$counts[o_h2[1:6],] #genes tag used in the tutorial but our labels had names
conh2 <- makeContrasts(non_ibd - ibd, levels=design_h2)
resh2 <- glmQLFTest(fit_h2, contrast=conh2)
topTags(resh2)
is.deh2 <- decideTestsDGE(resh2, p.value=0.05)
# summary(is.deh2)

#imposing the 1.5 fold change
trh2 <- glmTreat(fit_h2, contrast = conh2, lfc = log2(1.5))
topTags(trh2)
is.de.h2 <- decideTestsDGE(trh2)
summary(is.de.h2)
plotMD(trh2, status=is.de.h2, values=c(1,-1), col=c("red","blue"), legend="topright")

```

### -1\*ibd 1\*non\_ibd



```

# dev.copy(png, paste0("ibd_3v2h_plotMD15.png"))
# dev.off()

#using goana
goh2 <-goana(resh2, species = "Hs", FDR = 0.05)
# topGO(goh2, n=15)

#Supplementary table 8
topGO(goh2, n=15, sort="down")
write.table(topGO(goh2,n=2500, sort = "down"), file = "supplementary_table_8a_IBD.txt")
topGO(goh2, n=15, sort="up")
write.table(topGO(goh2,n=2500, sort = "up"), file = "supplementary_table_8b_IBD.txt")

topGO(goh2, n=15, sort="up", ontology = "BP")
topGO(goh2, n=15, sort="down", ontology = "BP")
topGO(goh2, n=15, sort="up", ontology = "CC")
topGO(goh2, n=15, sort="down", ontology = "CC")
topGO(goh2, n=15, sort="up", ontology = "MF")
topGO(goh2, n=15, sort="down", ontology = "MF")
kegh2 <- kegga(resh2, species="Hs", FDR = 0.15)
topKEGG(kegh2, n=15, truncate=34, sort = "up")
topKEGG(kegh2, n=15, truncate=34, sort = "down")

# Table 3 Top Gene Ontology sets in IBD and non-IBD
write.table(topGO(goh2, n=15, sort="up", ontology = "BP"), file = "Table3_nonibd.txt")
write.table(topGO(goh2, n=15, sort="down", ontology = "BP"), file = "Table3_IBD.txt")

```

```

# end table 3

topTagsibd_3v2h <- topTags(resh2, n=100000)$table
#neg
negFCibd_3v2h <- topTagsibd_3v2h[,1]<0
neg_geneibd_3v2h <- as.numeric(topTagsibd_3v2h[,5][negFCibd_3v2h])
names(neg_geneibd_3v2h) <-rownames(topTagsibd_3v2h)[negFCibd_3v2h]
G0data_negibd_3v2h <- new("topG0data", description = "Significant GO",
                           ontology = "BP", allGenes = neg_geneibd_3v2h,
                           geneSel = topDiffGenes, nodeSize = 10, annot = annFUN.org ,
                           mapping = "org.Hs.eg.db", ID="entrez")

## Building most specific GOs .....

## ( 8912 GO terms found. )

## Build GO DAG topology .....

## ( 13075 GO terms and 30857 relations. )

## Annotating nodes .....

## ( 6235 genes annotated to the GO terms. )

resultFisher_negibd_3v2h <- runTest(G0data_negibd_3v2h, algorithm = "classic", statistic = "fisher")

##
##          -- Classic Algorithm --
##
##          the algorithm is scoring 3172 nontrivial nodes
##          parameters:
##                  test statistic: fisher

resultKS_negibd_3v2h <- runTest(G0data_negibd_3v2h, algorithm = "classic", statistic = "ks")

##
##          -- Classic Algorithm --
##
##          the algorithm is scoring 4537 nontrivial nodes
##          parameters:
##                  test statistic: ks
##                  score order: increasing

resultKS.elim_negibd_3v2h <- runTest(G0data_negibd_3v2h, algorithm = "elim", statistic = "ks")

```

```

##          -- Elim Algorithm --
##
##      the algorithm is scoring 4537 nontrivial nodes
##      parameters:
##          test statistic: ks
##          cutOff: 0.01
##          score order: increasing

##
##      Level 18: 1 nodes to be scored      (0 eliminated genes)

##
##      Level 17: 3 nodes to be scored      (0 eliminated genes)

##
##      Level 16: 11 nodes to be scored     (0 eliminated genes)

##
##      Level 15: 24 nodes to be scored     (26 eliminated genes)

##
##      Level 14: 67 nodes to be scored     (89 eliminated genes)

##
##      Level 13: 136 nodes to be scored    (157 eliminated genes)

##
##      Level 12: 193 nodes to be scored    (302 eliminated genes)

##
##      Level 11: 312 nodes to be scored    (767 eliminated genes)

##
##      Level 10: 471 nodes to be scored    (1083 eliminated genes)

##
##      Level 9:  610 nodes to be scored    (1481 eliminated genes)

##
##      Level 8:  637 nodes to be scored    (1698 eliminated genes)

##
##      Level 7:  696 nodes to be scored    (2347 eliminated genes)

##
##      Level 6:  621 nodes to be scored    (2677 eliminated genes)

##
##      Level 5:  406 nodes to be scored    (2932 eliminated genes)

```

```

## Level 4: 225 nodes to be scored (3121 eliminated genes)

## Level 3: 103 nodes to be scored (3506 eliminated genes)

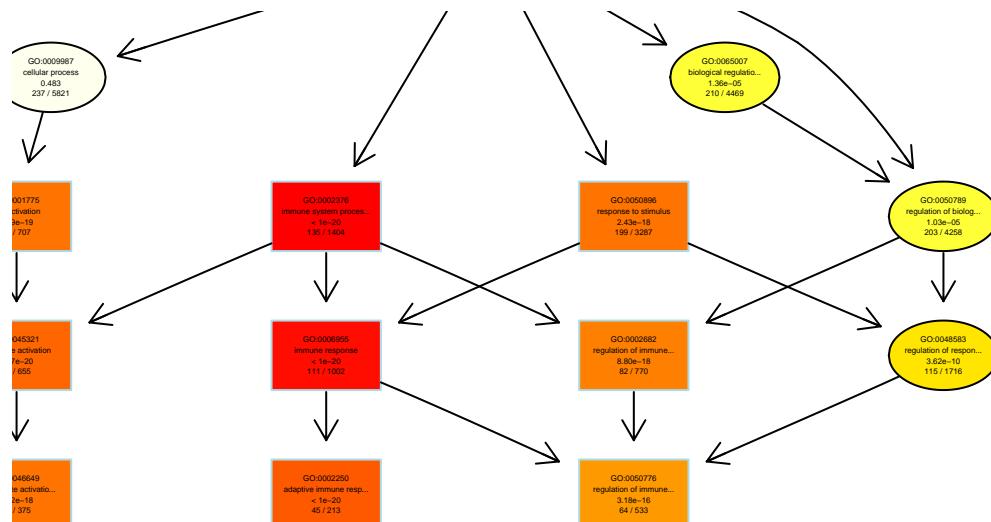
## Level 2: 20 nodes to be scored (3570 eliminated genes)

## Level 1: 1 nodes to be scored (3873 eliminated genes)

allRes_negibd_3v2h <- GenTable(GOdata_negibd_3v2h, classicFisher = resultFisher_negibd_3v2h,
                                classicKS = resultKS_negibd_3v2h,
                                elimKS = resultKS.elim_negibd_3v2h,
                                orderBy = "classicFisher",
                                ranksOf = "classicFisher", topNodes = 10)
# allRes_negibd_3v2h

#Supplementary Figure 9
showSigOfNodes(GOdata_negibd_3v2h, score(resultFisher_negibd_3v2h),
               firstSigNodes = 10, useInfo = 'all')

```



```

printGraph(G0data_negibd_3v2h, resultFisher_negibd_3v2h, firstSigNodes = 10,
           fn.prefix = "SuppFig9", useInfo = "all", pdfSW = TRUE)

#pos
posFCibd_3v2h <- topTagsibd_3v2h[,1]>0
pos_geneibd_3v2h <- as.numeric(topTagsibd_3v2h[,5][posFCibd_3v2h])
names(pos_geneibd_3v2h) <-rownames(topTagsibd_3v2h)[posFCibd_3v2h]
G0data_posibd_3v2h <- new("topG0data", description = "Significant GO",
                           ontology = "BP", allGenes = pos_geneibd_3v2h,
                           geneSel = topDiffGenes, nodeSize = 10, annot = annFUN.org ,
                           mapping = "org.Hs.eg.db", ID="entrez")

## 
## Building most specific GOs .....

## ( 8661 GO terms found. )

## 
## Build GO DAG topology .....

## ( 12816 GO terms and 30194 relations. )

## 
## Annotating nodes .....

## ( 5938 genes annotated to the GO terms. )

resultFisher_posibd_3v2h <- runTest(G0data_posibd_3v2h, algorithm = "classic", statistic = "fisher")

## 
## -- Classic Algorithm --
## 
## the algorithm is scoring 3232 nontrivial nodes
## parameters:
##   test statistic: fisher

resultKS_posibd_3v2h <- runTest(G0data_posibd_3v2h, algorithm = "classic", statistic = "ks")

## 
## -- Classic Algorithm --
## 
## the algorithm is scoring 4292 nontrivial nodes
## parameters:
##   test statistic: ks
##   score order: increasing

resultKS.elim_posibd_3v2h <- runTest(G0data_posibd_3v2h, algorithm = "elim", statistic = "ks")

```

```

##          -- Elim Algorithm --
##
##      the algorithm is scoring 4292 nontrivial nodes
##      parameters:
##          test statistic: ks
##          cutOff: 0.01
##          score order: increasing

##
##      Level 19: 1 nodes to be scored      (0 eliminated genes)

##
##      Level 18: 1 nodes to be scored      (12 eliminated genes)

##
##      Level 17: 4 nodes to be scored      (12 eliminated genes)

##
##      Level 16: 10 nodes to be scored     (12 eliminated genes)

##
##      Level 15: 23 nodes to be scored     (12 eliminated genes)

##
##      Level 14: 52 nodes to be scored     (12 eliminated genes)

##
##      Level 13: 96 nodes to be scored     (224 eliminated genes)

##
##      Level 12: 152 nodes to be scored    (539 eliminated genes)

##
##      Level 11: 281 nodes to be scored    (849 eliminated genes)

##
##      Level 10: 439 nodes to be scored    (1007 eliminated genes)

##
##      Level 9: 563 nodes to be scored     (1331 eliminated genes)

##
##      Level 8: 610 nodes to be scored     (1853 eliminated genes)

##
##      Level 7: 686 nodes to be scored     (2213 eliminated genes)

##
##      Level 6: 614 nodes to be scored     (2550 eliminated genes)

```

```

##      Level 5:    410 nodes to be scored  (3064 eliminated genes)

##      Level 4:    225 nodes to be scored  (3397 eliminated genes)

##      Level 3:    104 nodes to be scored  (3504 eliminated genes)

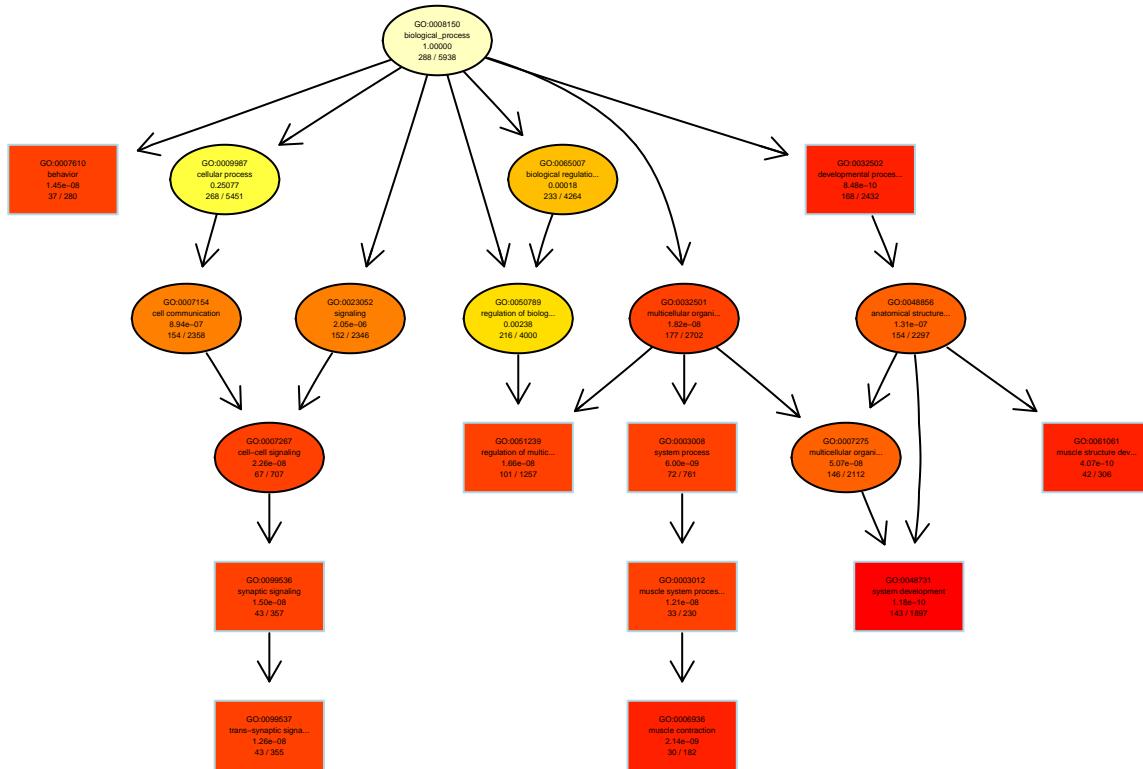
##      Level 2:    20 nodes to be scored   (3544 eliminated genes)

##      Level 1:    1 nodes to be scored    (3546 eliminated genes)

allRes_posibd_3v2h <- GenTable(GOdata_posibd_3v2h, classicFisher = resultFisher_posibd_3v2h,
                                classicKS = resultKS_posibd_3v2h,
                                elimKS = resultKS.elim_posibd_3v2h,
                                orderBy = "classicFisher",
                                ranksOf = "classicFisher", topNodes = 10)
# allRes_posibd_3v2h

# Supplementary Figure 10
showSigOfNodes(GOdata_posibd_3v2h, score(resultFisher_posibd_3v2h),
               firstSigNodes = 10, useInfo = 'all')

```



```

printGraph(G0data_posibd_3v2h, resultFisher_posibd_3v2h, firstSigNodes = 10,
           fn.prefix = "SuppFig10", useInfo = "all", pdfSW = TRUE)

#heatmap table
tbl_up_bp_3v2h <- topGO(goh2, n=15, sort="up", ontology = "BP")
tbl_dn_bp_3v2h <- topGO(goh2, n=15, sort="down", ontology = "BP")
log.tbl_up_bp_3v2h <- append(-log2(tbl_up_bp_3v2h[,6]), -log2(tbl_dn_bp_3v2h[,6]))
log.tbl_dn_bp_3v2h <- append(-log2(tbl_up_bp_3v2h[,7]), -log2(tbl_dn_bp_3v2h[,7]))

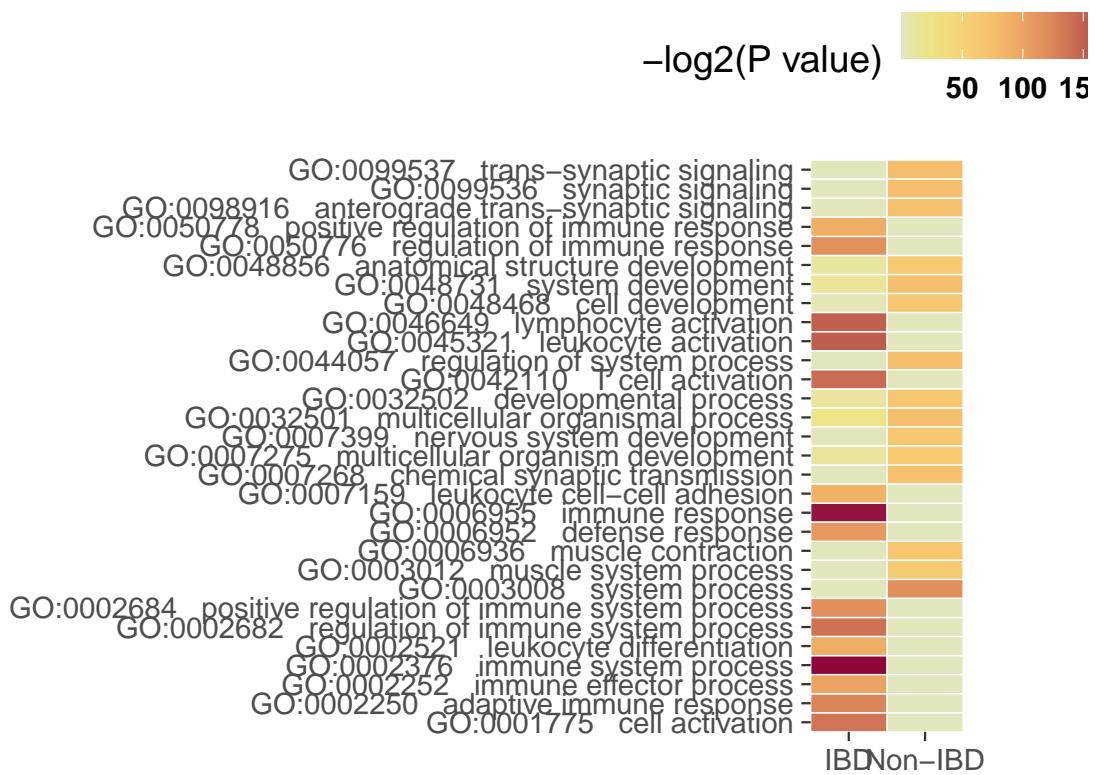
rowname <- rep(c(paste(rownames(tbl_up_bp_3v2h), " ",tbl_up_bp_3v2h$Term),
                  paste(rownames(tbl_dn_bp_3v2h), " ",tbl_dn_bp_3v2h$Term)),2)
colname <- c(rep("Non-IBD",30),rep("IBD",30))
P_value <- append(log.tbl_up_bp_3v2h, log.tbl_dn_bp_3v2h)

#create dataframe and remove 0's
log.tbl_all_bp_3v2h <- as.data.frame(cbind(rowname,colname,as.numeric(P_value)))
log.tbl_all_bp_3v2h[log.tbl_all_bp_3v2h == 0] <- NA
# write.table(log.tbl_all_bp_3v2h, file = "log_tbl_all_ibd_3v2h.txt")

#Supplementary Figure 4d
ibd_3v2h_heatmap_G0 <- ggplot(log.tbl_all_bp_3v2h,
                                 mapping = aes(x = colname, y = rowname, fill = P_value)) +
  geom_tile(colour="white",size=0.25) +
  scale_y_discrete(expand=c(0,0)) +
  scale_x_discrete(expand=c(0,0)) +
  theme_grey(base_size = 14) +
  labs(x = "", y = "") +
  coord_fixed(ratio=0.25) +
  theme(
    #bold font for legend text
    legend.text=element_text(face="bold"),
    legend.position="top",
    #set thickness of axis ticks
    axis.ticks=element_line(size=0.4),
    #remove plot background
    plot.background=element_blank(),
    #remove plot border
    panel.border=element_blank()) +
  scale_fill_continuous_sequential(palette = "Heat", name="-log2(P value)")

ibd_3v2h_heatmap_G0

```



```
# ggsave(
#   "ibd_3v2h_heatmap_GO_BP.png",
#   ibd_3v2h_heatmap_GO,
#   width = 8.5,
#   height = 11,
#   dpi = 1200
# )
```