

Untitled

Núria Rivera Brugués

2023-11-14

```
library(data.table)
library(Seurat)

## Loading required package: SeuratObject

## Loading required package: sp

##
## Attaching package: 'SeuratObject'

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

library(sp)
library(Matrix)
library(SeuratObject)
library(patchwork)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':
## 
##     between, first, last

## The following objects are masked from 'package:stats':
## 
##     filter, lag

## The following objects are masked from 'package:base':
## 
##     intersect, setdiff, setequal, union

library(tidyverse)
```

```

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## vforcats    1.0.0     vreadr      2.1.4
## vggplot2    3.4.4     vstringr    1.5.1
## vlubridate  1.9.3     vtibble     3.2.1
## vpurrr      1.0.2     vtidyrm    1.3.0

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::between()    masks data.table::between()
## x tidyrm::expand()    masks Matrix::expand()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x lubridate::hour()   masks data.table::hour()
## x lubridate::isoweek() masks data.table::isoweek()
## x dplyr::lag()        masks stats::lag()
## x dplyr::last()       masks data.table::last()
## x lubridate::mday()   masks data.table::mday()
## x lubridate::minute() masks data.table::minute()
## x lubridate::month()  masks data.table::month()
## x tidyrm::pack()      masks Matrix::pack()
## x lubridate::quarter() masks data.table::quarter()
## x lubridate::second() masks data.table::second()
## x purrr::transpose()  masks data.table::transpose()
## x tidyrm::unpack()    masks Matrix::unpack()
## x lubridate::wday()   masks data.table::wday()
## x lubridate::week()   masks data.table::week()
## x lubridate::yday()   masks data.table::yday()
## x lubridate::year()   masks data.table::year()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(ggplot2) # plots
library(magrittr) # %>% operator

## 
## Attaching package: 'magrittr'
##
## The following object is masked from 'package:purrr':
## 
##     set_names
## 
## The following object is masked from 'package:tidyrm':
## 
##     extract

library(reticulate) # required for "leiden" clustering
library(enrichR) # functional enrichment

## Welcome to enrichR
## Checking connection ...
## No internet connection could be found.

```

```

library(future) # multicore support for Seurat
library(cowplot)

## 
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:lubridate':
## 
##     stamp
## 
## The following object is masked from 'package:patchwork':
## 
##     align_plots

library(SingleCellExperiment)

## Loading required package: SummarizedExperiment
## Loading required package: MatrixGenerics
## Loading required package: matrixStats
##
## Attaching package: 'matrixStats'
##
## The following object is masked from 'package:dplyr':
## 
##     count
## 
## 
## Attaching package: 'MatrixGenerics'
##
## The following objects are masked from 'package:matrixStats':
## 
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars
## 
## Loading required package: GenomicRanges
## Loading required package: stats4
## Loading required package: BiocGenerics
## 
## Attaching package: 'BiocGenerics'

```

```

##
## The following objects are masked from 'package:lubridate':
##
##     intersect, setdiff, union
##
## The following objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union
##
## The following object is masked from 'package:SeuratObject':
##
##     intersect
##
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
##
## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, 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.max, which.min
##
## Loading required package: S4Vectors
##
## Attaching package: 'S4Vectors'
##
## The following objects are masked from 'package:lubridate':
##
##     second, second<-
##
## The following object is masked from 'package:tidyR':
##
##     expand
##
## The following objects are masked from 'package:dplyr':
##
##     first, rename
##
## The following objects are masked from 'package:Matrix':
##
##     expand, unname
##
## The following objects are masked from 'package:data.table':
##
##     first, second
##
## The following object is masked from 'package:utils':
##
##     findMatches
##

```

```

## The following objects are masked from 'package:base':
##
##      expand.grid, I, unname
##
## Loading required package: IRanges
##
## Attaching package: 'IRanges'
##
## The following object is masked from 'package:lubridate':
##
##      %within%
##
## The following object is masked from 'package:purrr':
##
##      reduce
##
## The following objects are masked from 'package:dplyr':
##
##      collapse, desc, slice
##
## The following object is masked from 'package:sp':
##
##      %over%
##
## The following object is masked from 'package:data.table':
##
##      shift
##
## Loading required package: GenomeInfoDb
##
## Attaching package: 'GenomicRanges'
##
## The following object is masked from 'package:magrittr':
##
##      subtract
##
## 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")'.
##
##
## Attaching package: 'Biobase'
##
## The following object is masked from 'package:MatrixGenerics':
##
##      rowMedians
##
## The following objects are masked from 'package:matrixStats':
##
##      anyMissing, rowMedians
##

```

```

## 
## Attaching package: 'SummarizedExperiment'
##
## The following object is masked from 'package:Seurat':
## 
##     Assays
## 
## The following object is masked from 'package:SeuratObject':
## 
##     Assays

library(readr)

healthy<-read.delim("/home/nrb/Escritorio/perAnna/20_sc_kidney/healthy.dge.txt")

annotation<-read.delim("/home/nrb/Escritorio/perAnna/20_sc_kidney/GSE119531_Healthy.combined.cell.annotation.txt")

matrix<-read.delim("/home/nrb/Escritorio/perAnna/20_sc_kidney/GSE119531_series_matrix.txt")

# Eliminar las 11 últimas letras en cada cadena
annotation$group <- sapply(annotation$CellBarcode, function(x) sub("_.*", "", x))
annotation$group <- sub("\\-", ".", annotation$group)
annotation$CellBarcode <- sub("\\-", ".", annotation$CellBarcode)
annotation$group<-as.factor(annotation$group)
table(annotation$group)

sNuc10x<-subset(annotation, group=="sNuc.10x")
sCellDropseq<-subset(annotation, group=="sCellDropseq")
DroNcSeq<-subset(annotation, group=="DroNcSeq")
sNucDropseq<-subset(annotation, group=="sNucDropseq")

healthy_DroNcSeq <- healthy[,DroNcSeq$CellBarcode]
write.csv2(healthy_DroNcSeq, "healthy_DroNcSeq.csv")
healthy_sNuc10x <- healthy[,sNuc10x$CellBarcode]
write.csv2(healthy_sNuc10x, "healthy_sNuc10x.csv")
healthy_sCellDropseq <- healthy[,sCellDropseq$CellBarcode]
write.csv2(healthy_sCellDropseq, "healthy_sCellDropseq.csv")
write_tsv(healthy_sCellDropseq, "healthy_sCellDropseq.tsv")
healthy_sNucDropseq <- healthy[,sNucDropseq$CellBarcode]
write.csv2(healthy_sNucDropseq, "healthy_sNucDropseq.csv")
write_tsv(healthy_sNucDropseq, "healthy_sNucDropseq.tsv")

```

Utilitzo només sNucDropseq i sCellDropseq

```

healthy_sNucDropseq<-as(as.matrix(healthy_sNucDropseq) , "sparseMatrix") # counts
healthy_sNucDropseq[1:10, 1:10]
healthy_sCellDropseq<-as(as.matrix(healthy_sCellDropseq), "sparseMatrix") # counts
healthy_sCellDropseq[1:10, 1:10]

sNucDropseq<-sNucDropseq[,-3] # barcodes

```

```

sNucDropseq<-as.matrix(sNucDropseq)
sCellDropseq<-sCellDropseq[,-3] # barcodes
sCellDropseq<-as.matrix(sCellDropseq)

features_sNucDropseq<-rownames(healthy_sNucDropseq) # features
features_sNucDropseq<-as.matrix(features_sNucDropseq)
features_sCellDropseq<-rownames(healthy_sCellDropseq) # features
features_sCellDropseq<-as.matrix(features_sCellDropseq)

seurat_Cell <- CreateSeuratObject(counts=healthy_sCellDropseq, project = "Cell", min.cells = 5, min.features = 2)
str(seurat_Cell)
seurat_Nuc <- CreateSeuratObject(counts=healthy_sNucDropseq, project = "Nuc", min.cells = 5, min.features = 2)

sc.sn <- merge(seurat_Cell, y = seurat_Nuc, project = "sc.sn")
head(colnames(sc.sn))
table(sc.sn$orig.ident)

# Get cell names
sCellDropseq_names <- grep(pattern = "^sCell", x = colnames(sc.sn), value = T)
sNucDropseq_names <- grep(pattern = "^sNuc", x = colnames(sc.sn), value = T)

# Create new meta.data column
sc.sn@meta.data$Fraction[colnames(sc.sn) %in% sNucDropseq_names] <- "Nuc"
sc.sn@meta.data$Fraction[colnames(sc.sn) %in% sCellDropseq_names] <- "Cell"
saveRDS(sc.sn, file = "sc.sn.rds")

sc.sn <- readRDS("sc.sn.rds")

```

QC and selecting cells for further analysis

```

pbmc<-sc.sn
pbmc[["percent.mt"]] <- PercentageFeatureSet(pbmc, pattern = "^\mt-")

# Visualize QC metrics as a violin plot
pdf(paste("QC",sep=""))
VlnPlot(pbmc, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3)

## Warning: Default search for "data" layer in "RNA" assay yielded no results;
## utilizing "counts" layer instead.

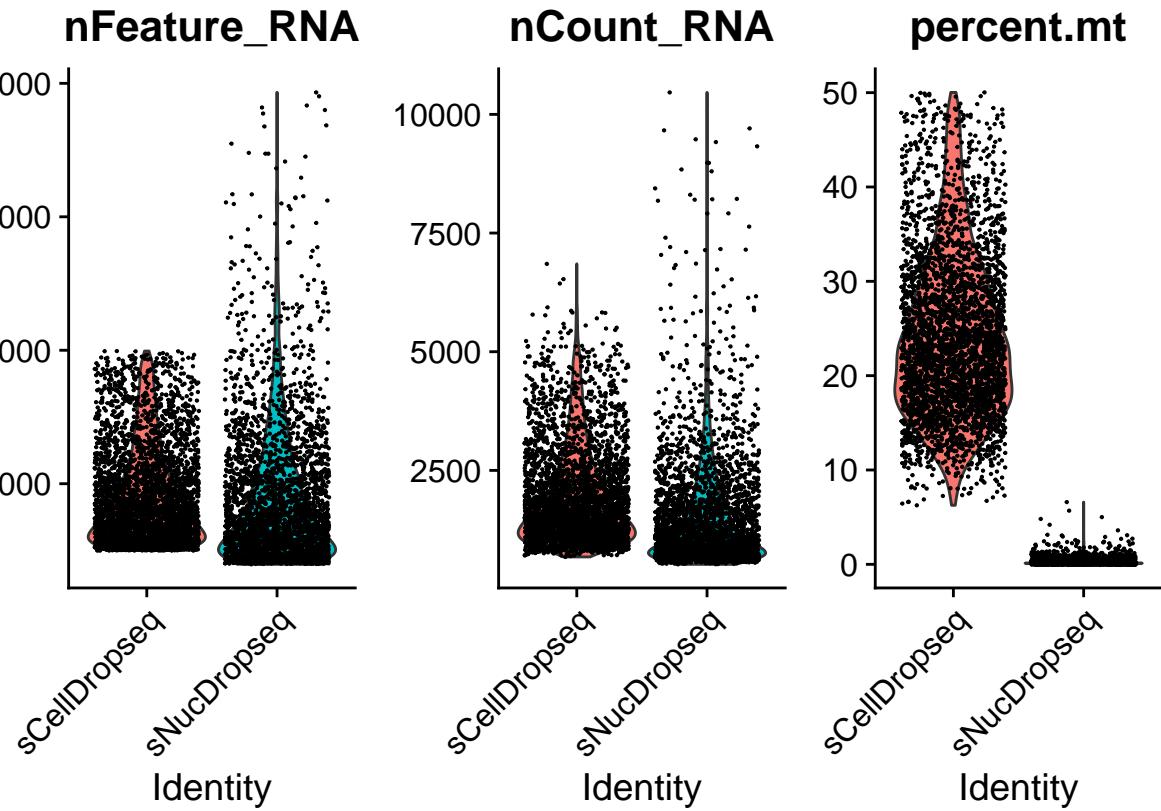
dev.off()

## pdf
## 2

VlnPlot(pbmc, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3)

## Warning: Default search for "data" layer in "RNA" assay yielded no results;
## utilizing "counts" layer instead.

```

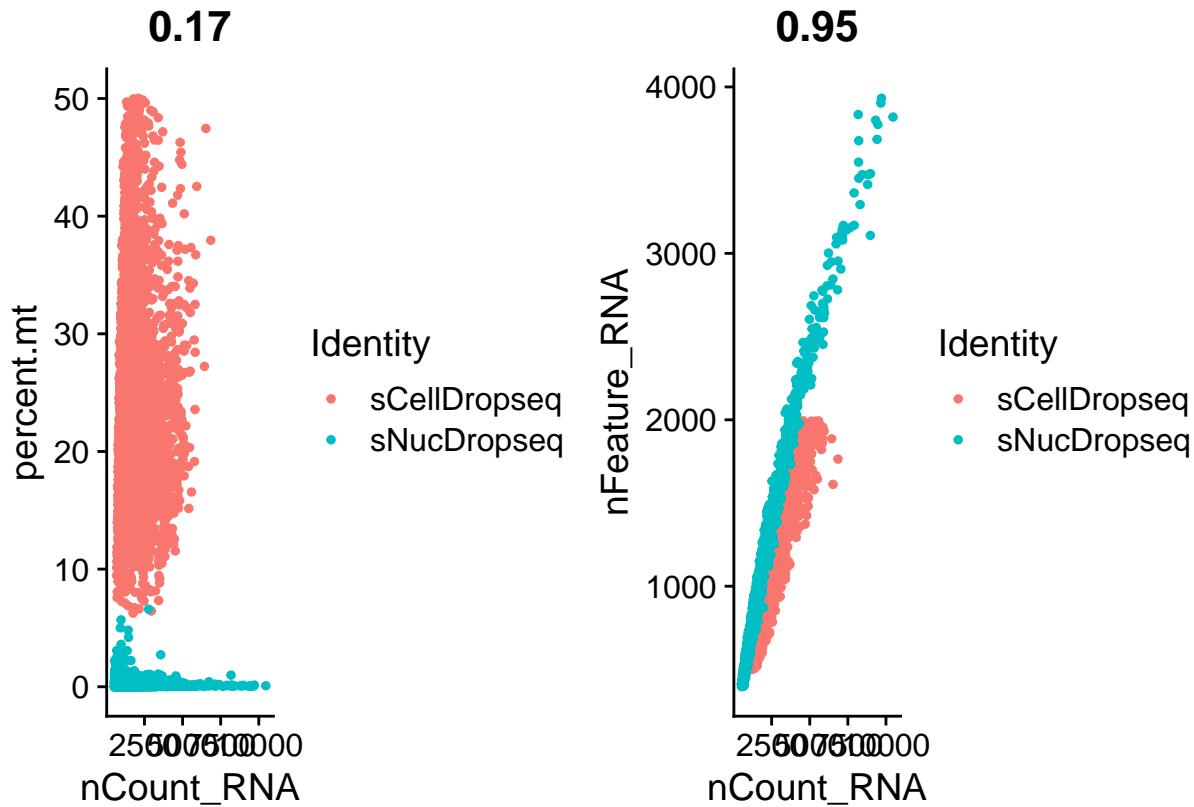


```
# FeatureScatter is typically used to visualize feature-feature relationships, but can be used
# for anything calculated by the object, i.e. columns in object metadata, PC scores etc.
```

```
plot1 <- FeatureScatter(pbmc, feature1 = "nCount_RNA", feature2 = "percent.mt")
plot2 <- FeatureScatter(pbmc, feature1 = "nCount_RNA", feature2 = "nFeature_RNA")
pdf(paste("scatter_features",sep=""))
plot_grid(plot1 + plot2)
dev.off()
```

```
## pdf
## 2

plot_grid(plot1 + plot2)
```



```
pbmc <- subset(pbmc, subset = nFeature_RNA > 200 & nFeature_RNA < 4000 & percent.mt < 50 )
```

Normalizing the data

```
pbmc <- NormalizeData(pbmc, normalization.method = "LogNormalize", scale.factor = 10000)
```

```
## Normalizing layer: counts.Cell  
## Normalizing layer: counts.Nuc
```

Identification of highly variable features (feature selection)

```
pbmc <- FindVariableFeatures(pbmc, selection.method = "vst", nfeatures = 2000)
```

```
## Finding variable features for layer counts.Cell  
## Finding variable features for layer counts.Nuc
```

```

# Identify the 10 most highly variable genes
top10 <- head(VariableFeatures(pbmc), 10)

# plot variable features with and without labels
plot1 <- VariableFeaturePlot(pbmc) + NoLegend()
plot2 <- LabelPoints(plot = plot1, points = top10, repel = TRUE) + NoLegend()

## When using repel, set xnudge and ynudge to 0 for optimal results

pdf(paste("variable.features", sep=""))
plot_grid(plot1 + plot2)

```

```

## Warning: Removed 2248 rows containing missing values ('geom_point()').
## Removed 2248 rows containing missing values ('geom_point()').

```

```
dev.off()
```

```

## pdf
## 2

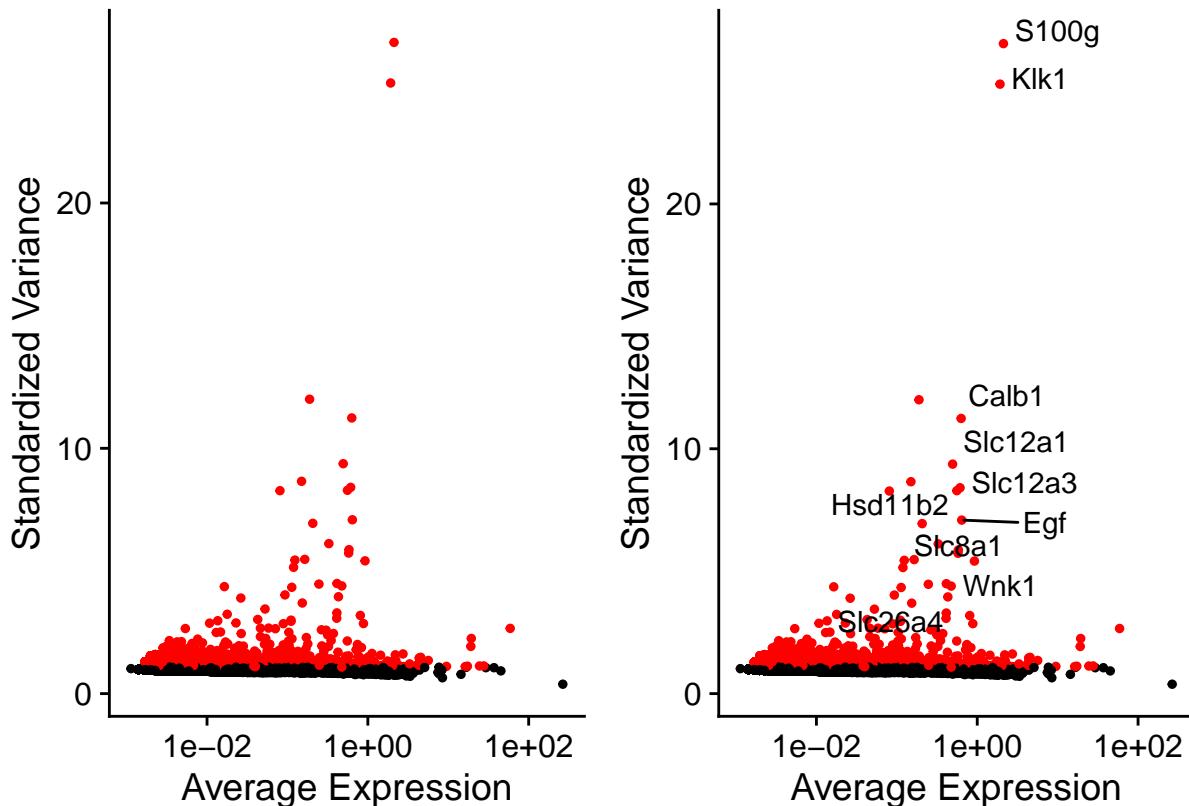
```

```
plot_grid(plot1 + plot2)
```

```

## Warning: Removed 2248 rows containing missing values ('geom_point()').
## Removed 2248 rows containing missing values ('geom_point()').

```



```

sc.sn.list <- SplitObject(sc.sn, split.by = "Fraction")
features <- SelectIntegrationFeatures(object.list = sc.sn.list)

## No variable features found for object1 in the object.list. Running FindVariableFeatures ...

## Finding variable features for layer counts.Cell

## No variable features found for object2 in the object.list. Running FindVariableFeatures ...

## Finding variable features for layer counts.Nuc

sc.sn.list <- lapply(X = sc.sn.list, FUN = function(x) {
  x <- NormalizeData(x, normalization.method = "LogNormalize", scale.factor = 10000)
  x <- ScaleData(x, features = features, verbose = FALSE)
  x <- RunPCA(x, features = features, verbose = FALSE)
})

## Normalizing layer: counts.Cell

## Normalizing layer: counts.Nuc

kidney.anchors <- FindIntegrationAnchors(object.list = sc.sn.list, dims = 1:30)

## Computing 2000 integration features

## No variable features found for object1 in the object.list. Running FindVariableFeatures ...

## Finding variable features for layer counts.Cell

## No variable features found for object2 in the object.list. Running FindVariableFeatures ...

## Finding variable features for layer counts.Nuc

## Scaling features for provided objects

## Finding all pairwise anchors

## Running CCA

## Merging objects

## Finding neighborhoods

## Finding anchors

## Found 8687 anchors

## Filtering anchors

## Retained 2374 anchors

```

```

kidney.integrated <- IntegrateData(anchorset = kidney.anchors, dims = 1:30)

## Merging dataset 2 into 1

## Extracting anchors for merged samples

## Finding integration vectors

## Finding integration vector weights

## Integrating data

# switch to integrated assay. The variable features of this assay are automatically
# set during IntegrateData
DefaultAssay(kidney.integrated) <- "integrated"

# Run the standard workflow for visualization and clustering
kidney.integrated <- ScaleData(kidney.integrated, verbose = FALSE)
kidney.integrated <- RunPCA(kidney.integrated, npcs = 30, verbose = FALSE)
kidney.integrated <- RunUMAP(kidney.integrated, reduction = "pca", dims = 1:30)

## Warning: The default method for RunUMAP has changed from calling Python UMAP via reticulate to the R
## To use Python UMAP via reticulate, set umap.method to 'umap-learn' and metric to 'correlation'
## This message will be shown once per session

## 20:07:22 UMAP embedding parameters a = 0.9922 b = 1.112

## Found more than one class "dist" in cache; using the first, from namespace 'spam'

## Also defined by 'BiocGenerics'

## 20:07:22 Read 6542 rows and found 30 numeric columns

## 20:07:22 Using Annoy for neighbor search, n_neighbors = 30

## Found more than one class "dist" in cache; using the first, from namespace 'spam'

## Also defined by 'BiocGenerics'

## 20:07:22 Building Annoy index with metric = cosine, n_trees = 50

## 0%   10    20    30    40    50    60    70    80    90   100%
## [----|----|----|----|----|----|----|----|----|----|
## ****|*****|*****|*****|*****|*****|*****|*****|*****|*****|
## 20:07:23 Writing NN index file to temp file /tmp/Rtmppe07dsx/file6cd2230811f6
## 20:07:23 Searching Annoy index using 1 thread, search_k = 3000
## 20:07:26 Annoy recall = 100%
## 20:07:26 Commencing smooth kNN distance calibration using 1 thread with target n_neighbors = 30
## 20:07:27 Initializing from normalized Laplacian + noise (using RSpectra)
## 20:07:28 Commencing optimization for 500 epochs, with 292248 positive edges
## 20:07:37 Optimization finished

```

```

kidney.integrated <- RunTSNE(kidney.integrated, dims.use = 1:10, do.fast = TRUE)
p1 <- DimPlot(kidney.integrated, reduction = "umap", group.by = "Fraction", label = TRUE)
p2 <- DimPlot(kidney.integrated, reduction = "tsne", group.by = "Fraction", label = TRUE)
pdf(paste("umap.tsne",sep=""))
plot_grid(p1 + p2)
dev.off()

```

```

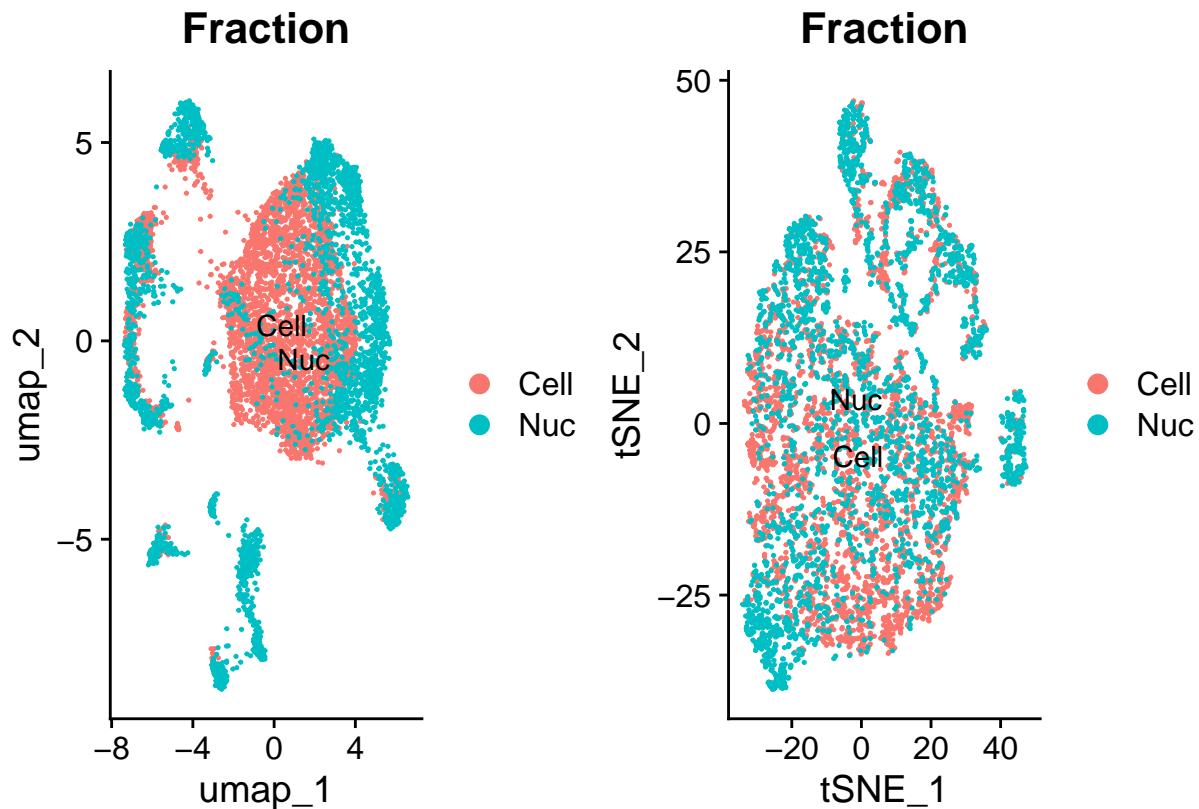
## pdf
## 2

```

```

plot_grid(p1 + p2)

```



```

pdf(paste("umap",sep=""))
DimPlot(kidney.integrated, reduction = "umap", group.by = "Fraction")
dev.off()

```

```

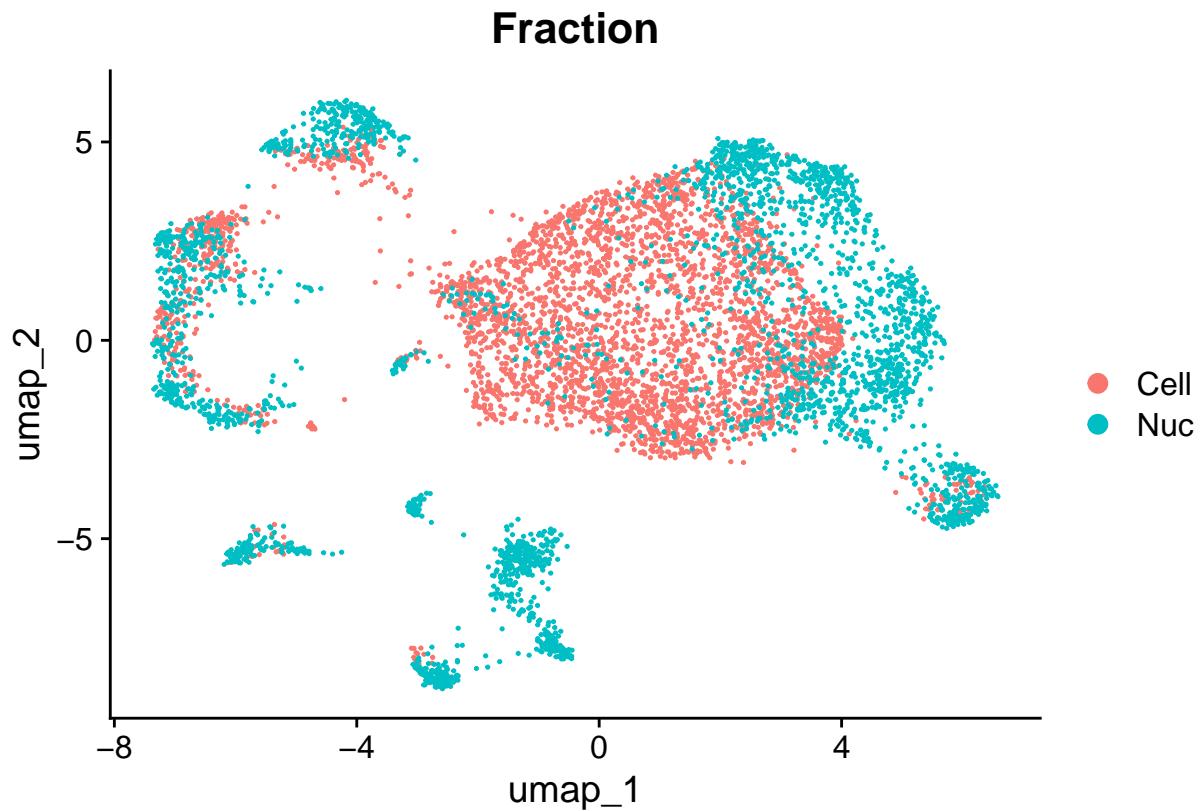
## pdf
## 2

```

```

DimPlot(kidney.integrated, reduction = "umap", group.by = "Fraction")

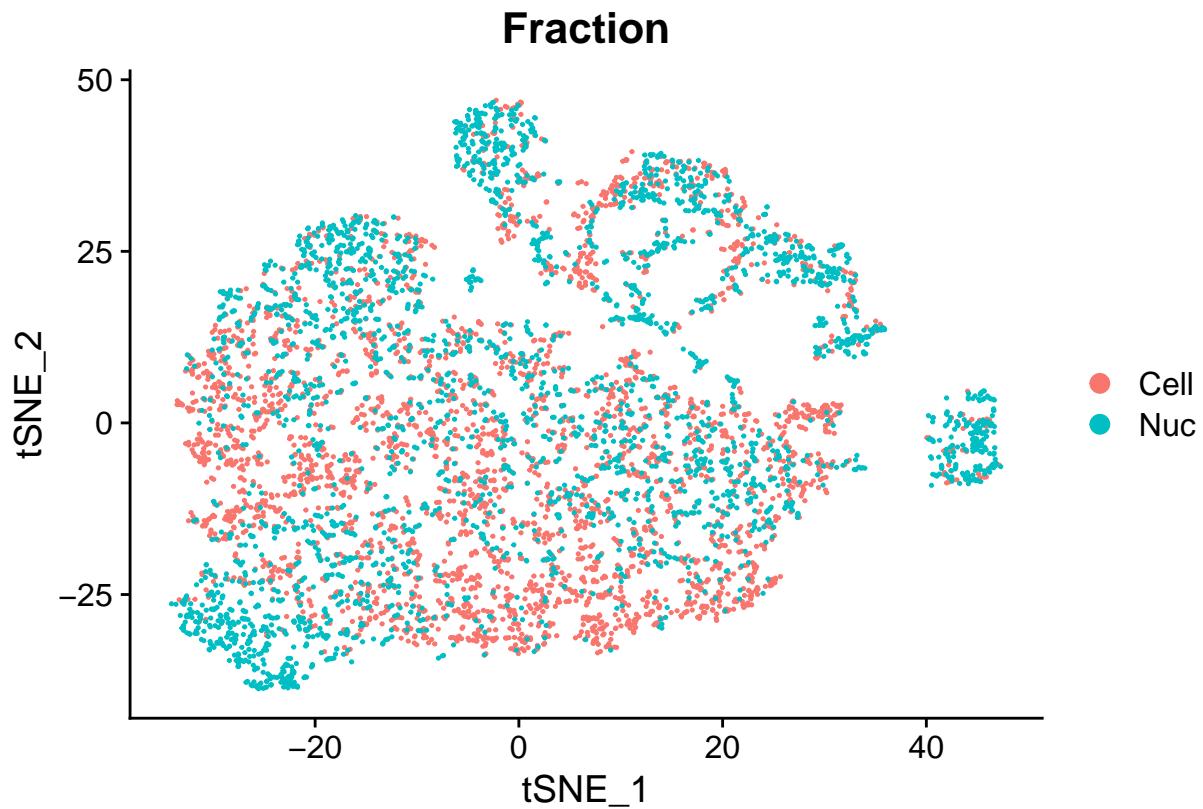
```



```
pdf(paste("tsne",sep=""))
DimPlot(kidney.integrated, reduction = "tsne", group.by = "Fraction")
dev.off()
```

```
## pdf
## 2

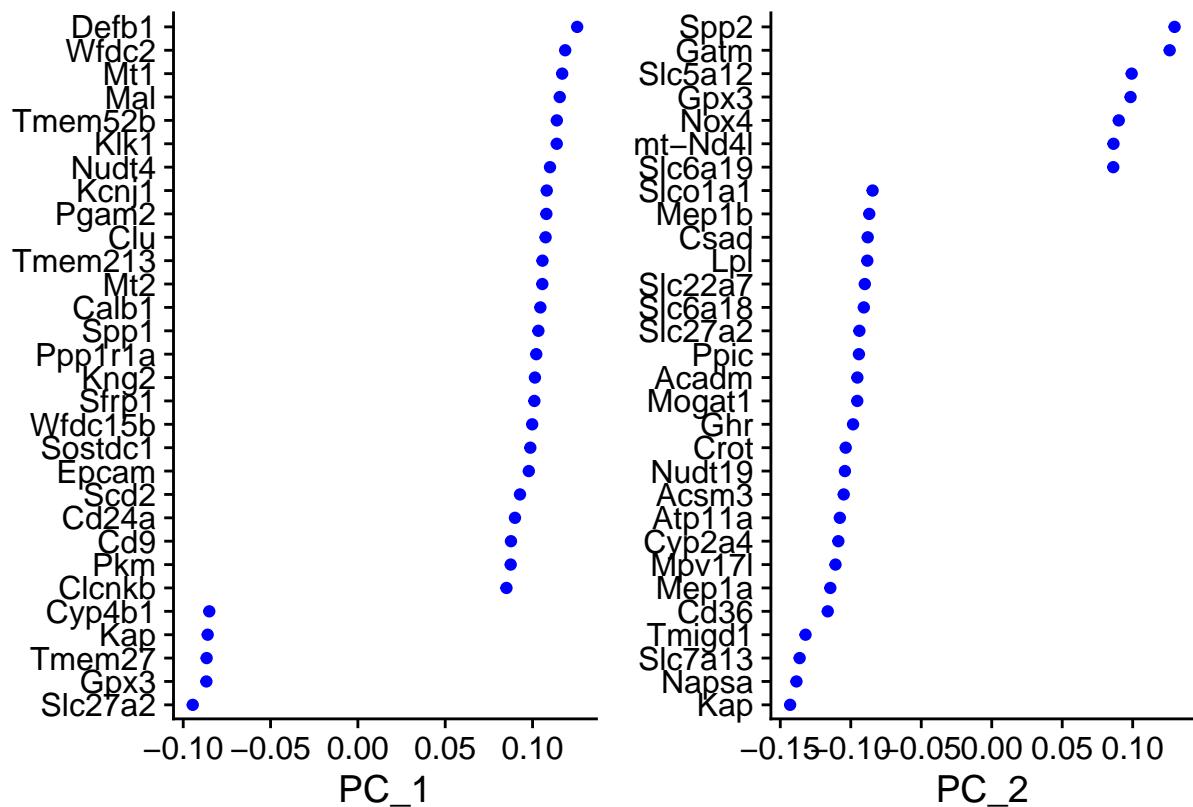
DimPlot(kidney.integrated, reduction = "tsne", group.by = "Fraction")
```



```
pdf(paste("Vizdim",sep=""))
VizDimLoadings(kidney.integrated, dims = 1:2, reduction = "pca")
dev.off()
```

```
## pdf
## 2

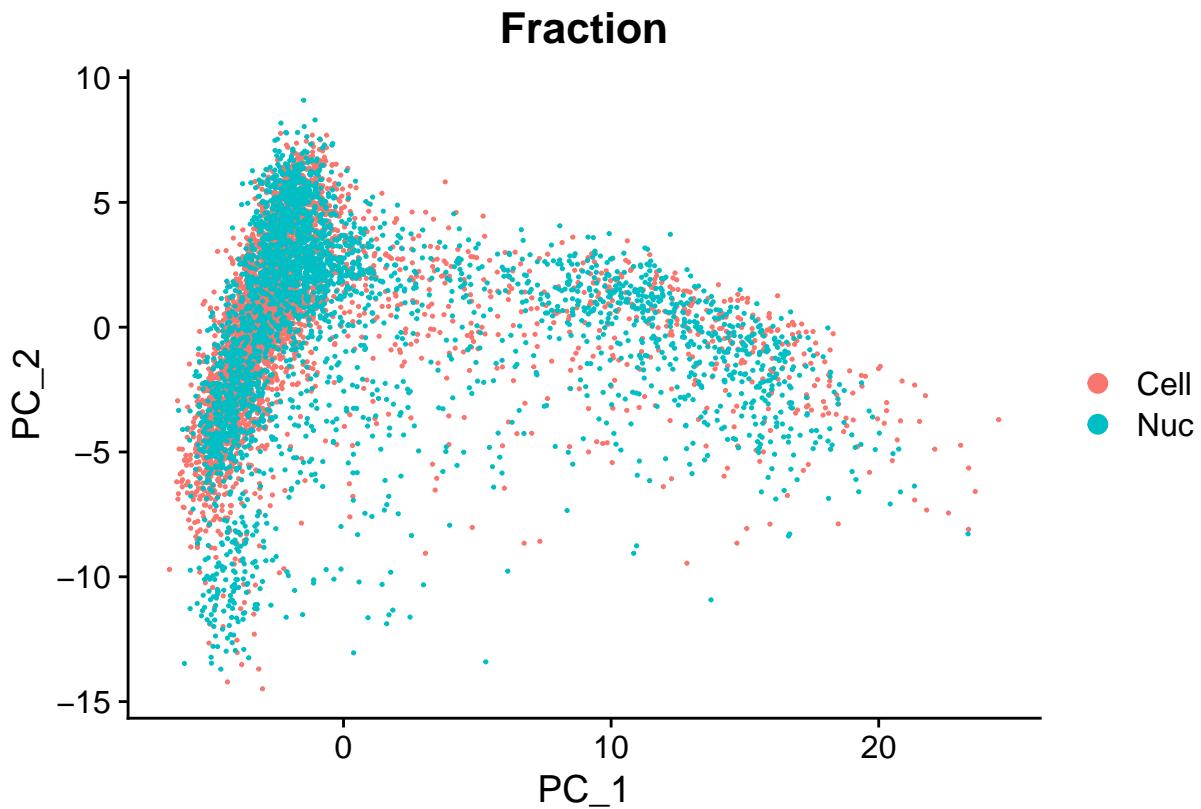
VizDimLoadings(kidney.integrated, dims = 1:2, reduction = "pca")
```



```
pdf(paste("pca.pdf",sep=""))
DimPlot(kidney.integrated, reduction = "pca",group.by = "Fraction")
dev.off()
```

```
## pdf
## 2

DimPlot(kidney.integrated, reduction = "pca",group.by = "Fraction")
```

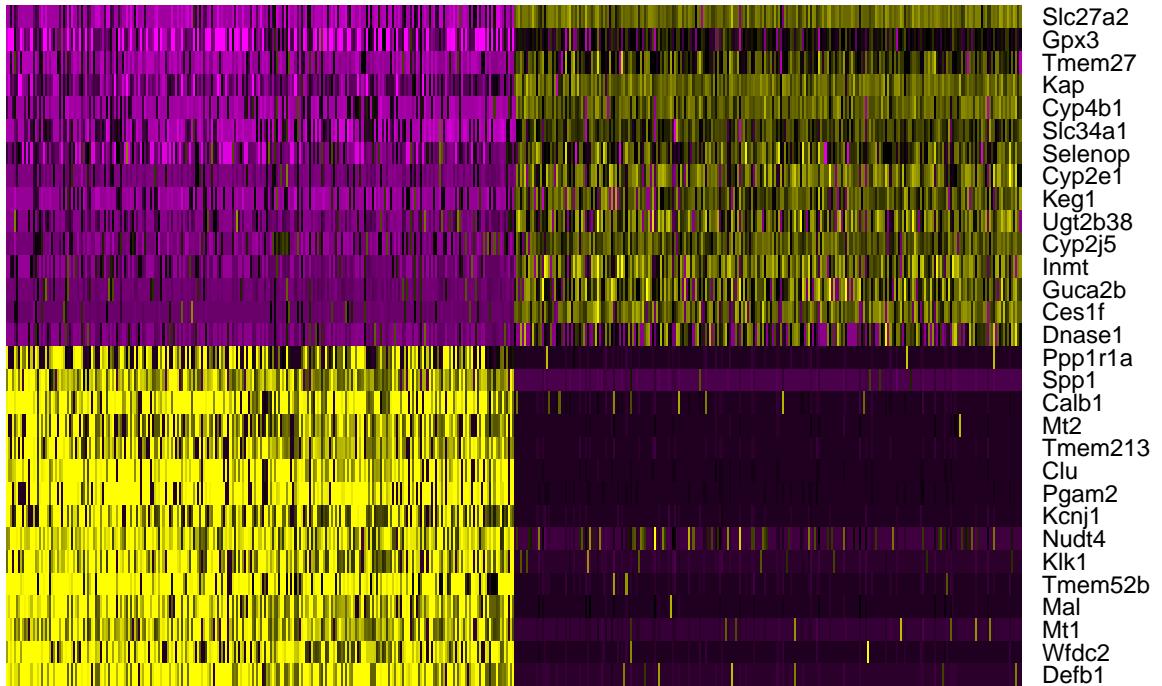


```
pdf(paste("heatmap_PCA1.pdf",sep=""))
DimHeatmap(kidney.integrated, dims = 1, cells = 500, balanced = TRUE)
dev.off()
```

```
## pdf
## 2

DimHeatmap(kidney.integrated, dims = 1, cells = 500, balanced = TRUE)
```

PC_1

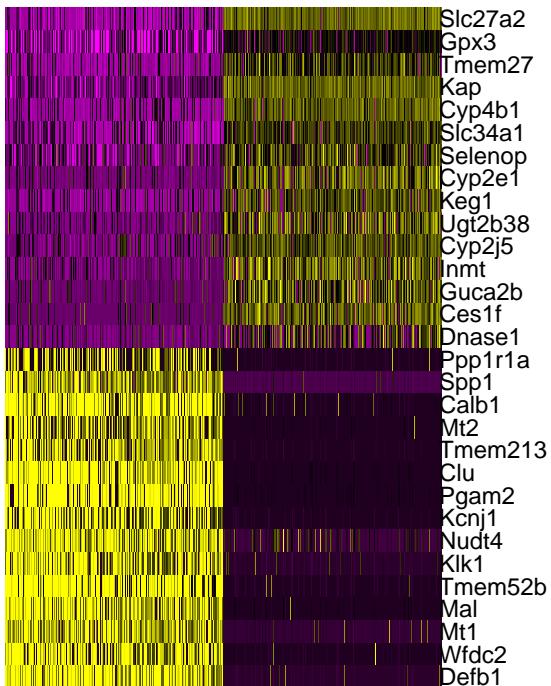


```
pdf(paste("heatmap_PCA2.pdf",sep=""))
DimHeatmap(kidney.integrated, dims = 1:2, cells = 500, balanced = TRUE)
dev.off()
```

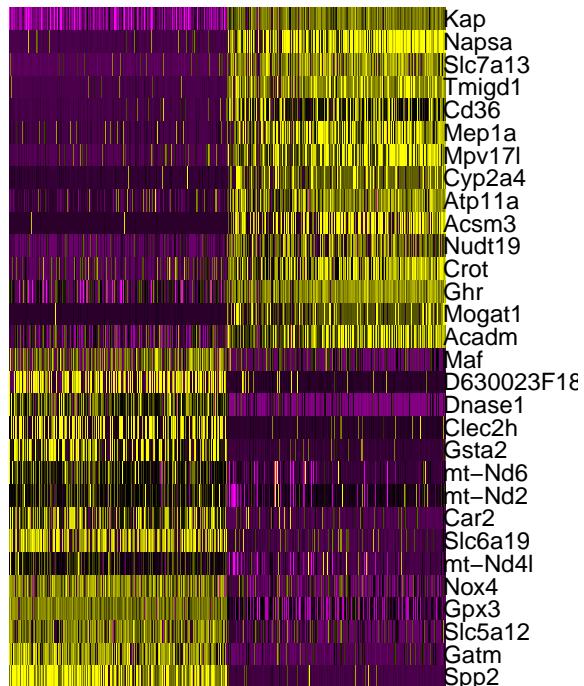
```
## pdf
## 2

DimHeatmap(kidney.integrated, dims = 1:2, cells = 500, balanced = TRUE)
```

PC_1



PC_2



```
# Examine and visualize PCA results a few different ways
print(kidney.integrated[["pca"]], dims = 1:5, nfeatures = 5)
```

```
## PC_ 1
## Positive: Defb1, Wfdc2, Mt1, Mal, Tmem52b
## Negative: Slc27a2, Gpx3, Tmem27, Kap, Cyp4b1
## PC_ 2
## Positive: Spp2, Gatm, Slc5a12, Gpx3, Nox4
## Negative: Kap, Napsa, Slc7a13, Tmigd1, Cd36
## PC_ 3
## Positive: Hsd11b2, Aqp2, Spink8, Cav2, Aqp3
## Negative: Wfdc15b, Egf, Umod, Ppp1r1a, Slc12a1
## PC_ 4
## Positive: Malat1, Snhg11, Slc5a12, Neat1, Slc34a1
## Negative: Hsd11b2, Spink8, Tmsb4x, Aqp2, Cav2
## PC_ 5
## Positive: Clu, Calb1, Pgam2, Slc12a3, Tmem52b
## Negative: Atp6v1g3, Slc4a9, Atp6v1c2, Atp6v0d2, Plet1
```

```
# NOTE: This process can take a long time for big datasets, comment out for expediency. More
# approximate techniques such as those implemented in ElbowPlot() can be used to reduce
# computation time
kidney.integrated <- JackStraw(kidney.integrated, num.replicate = 100)
kidney.integrated <- ScoreJackStraw(kidney.integrated, dims = 1:20)
```

```
pdf(paste("JackStrawPlot.pdf",sep=""))
JackStrawPlot(kidney.integrated, dims = 1:20)
```

```
## Warning: Removed 28078 rows containing missing values ('geom_point()').
```

```

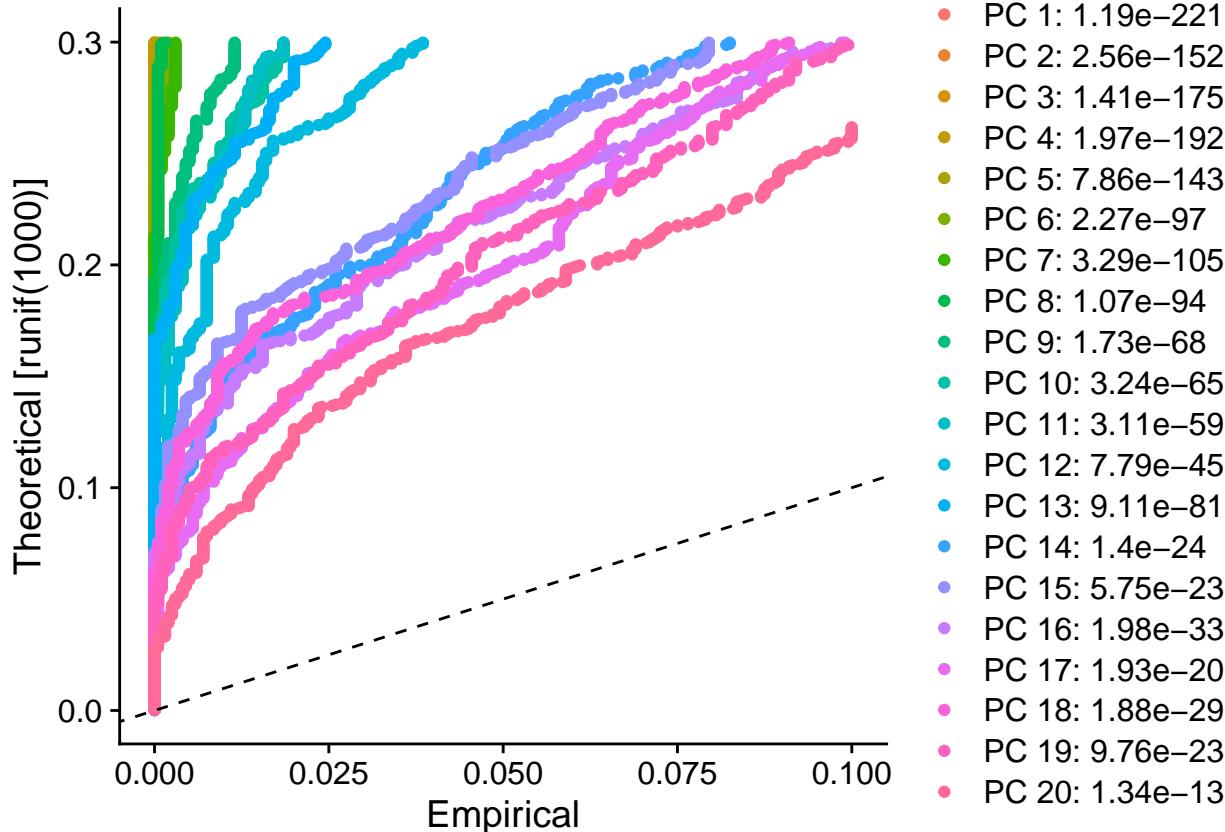
dev.off()

## pdf
## 2

JackStrawPlot(kidney.integrated, dims = 1:20)

## Warning: Removed 28078 rows containing missing values ('geom_point()').

```



```

pdf(paste("elbowplot.pdf", sep=""))
ElbowPlot(kidney.integrated)
dev.off()

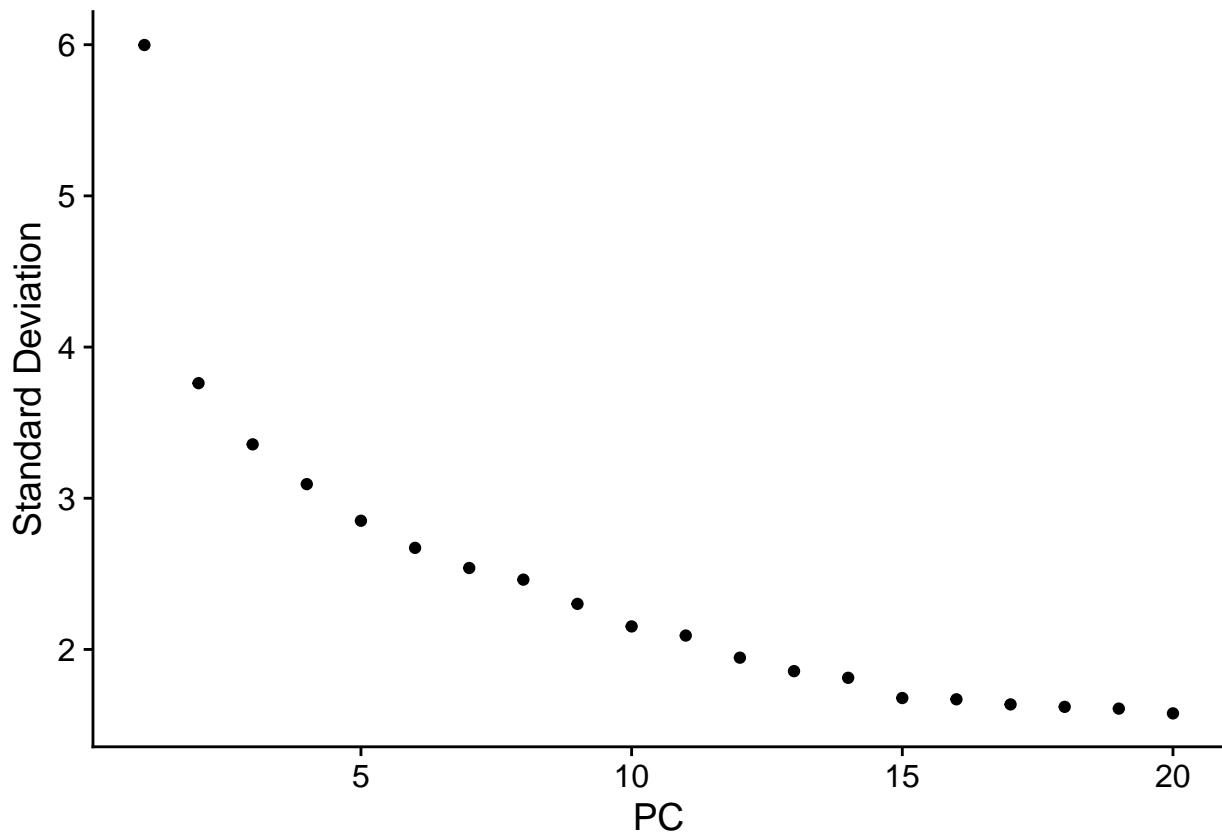
```

```

## pdf
## 2

ElbowPlot(kidney.integrated)

```



Finding differentially expressed features (cluster biomarkers)

```
# find markers for every cluster compared to all remaining cells, report only the positive ones
kidney.integrated.markers <- FindAllMarkers(kidney.integrated, only.pos = F, min.pct = 0.25, logfc.thre...
```

```
## Calculating cluster sCellDropseq
```

```
## Warning in mean.fxn(object[features, cells.2, drop = FALSE]): Se han producido
## NaNs
```

```
## For a (much!) faster implementation of the Wilcoxon Rank Sum Test,
## (default method for FindMarkers) please install the presto package
## -----
## install.packages('devtools')
## devtools::install_github('immunogenomics/presto')
## -----
## After installation of presto, Seurat will automatically use the more
## efficient implementation (no further action necessary).
## This message will be shown once per session
```

```
## Calculating cluster sNucDropseq
```

```
## Warning in mean.fxn(object[features, cells.1, drop = FALSE]): Se han producido
## NaNs
```

```

?FindAllMarkers()
kidney.integrated.markers %>% group_by(cluster) %>% top_n(n = 5, wt = avg_log2FC)

## # A tibble: 10 x 7
## # Groups:   cluster [2]
##       p_val avg_log2FC pct.1 pct.2 p_val_adj cluster      gene
##       <dbl>     <dbl> <dbl> <dbl>    <dbl> <fct>      <chr>
## 1 0.151 1.63 0.151 1.0 0       sCellDropseq Jun
## 2 0.085 1.63 0.085 0.932 0     sCellDropseq Car3
## 3 0.174 1.50 0.174 0.929 0     sCellDropseq Ttr
## 4 0.024 2.67 0.024 0.695 0     sCellDropseq Gm14064
## 5 0.028 1.50 0.028 0.646 0     sCellDropseq Ier2
## 6 7.63e-298 3.82 0.259 0.017 1.53e-294 sNucDropseq 9530026P05Rik
## 7 1.17e-138 3.36 0.27 0.007 2.34e-135 sNucDropseq Eda
## 8 8.32e- 29 3.24 0.314 0.006 1.66e- 25 sNucDropseq Col14a1
## 9 4.55e- 28 3.31 0.322 0.004 9.09e- 25 sNucDropseq Ttn
## 10 4.39e- 21 3.53 0.357 0.006 8.77e- 18 sNucDropseq Slit3

write.csv2(kidney.integrated.markers, "kidney.integrated.markers.csv")

clusterCell.markers <- FindMarkers(kidney.integrated, ident.1 = "sCellDropseq", logfc.threshold = 0.25,
                                    ## Warning in mean.fxn(object[features, cells.2, drop = FALSE]): Se han producido
                                    ## NaNs

head(clusterCell.markers, n = 5)

##       myAUC avg_diff power avg_log2FC pct.1 pct.2
## Cyp3a13 0.063 0.06626718 0.874 0.8267248 0.033 0.938
## Mmp20  0.069 0.07825942 0.862 1.2795033 0.028 0.936
## Slc15a5 0.070 0.06472888 0.860 0.4377045 0.059 0.988
## Fut2   0.076 0.01103067 0.848 0.2550506 0.014 0.888
## Ociad2 0.085 0.14289776 0.830 0.7538613 0.080 0.993

clusterNuc.markers <- FindMarkers(kidney.integrated, ident.1 = "sNucDropseq", logfc.threshold = 0.25, t

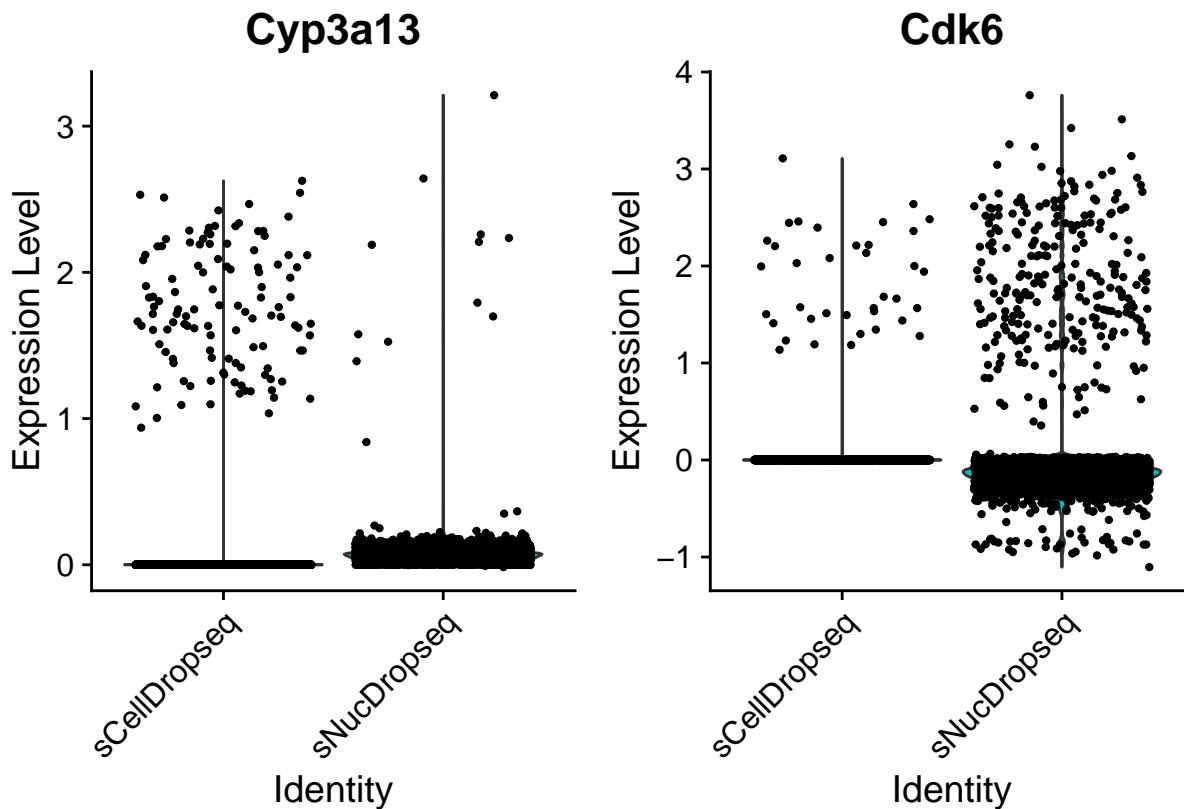
## Warning in mean.fxn(object[features, cells.1, drop = FALSE]): Se han producido
## NaNs

head(clusterNuc.markers, n = 5)

##       myAUC avg_diff power avg_log2FC pct.1 pct.2
## Cyp3a13 0.937 -0.06626718 0.874 -0.8267248 0.938 0.033
## Mmp20  0.931 -0.07825942 0.862 -1.2795033 0.936 0.028
## Slc15a5 0.930 -0.06472888 0.860 -0.4377045 0.988 0.059
## Fut2   0.924 -0.01103067 0.848 -0.2550506 0.888 0.014
## Ociad2 0.915 -0.14289776 0.830 -0.7538613 0.993 0.080

```

```
VlnPlot(kidney.integrated, features = c("Cyp3a13", "Cdk6"))
```



```
# you can plot raw counts as well
VlnPlot(kidney.integrated, features = c("Cdk6", "Wnk3"), log = TRUE)

## Warning in self$trans$transform(x): Se han producido NaNs

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning in self$trans$transform(x): Se han producido NaNs

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 1 rows containing non-finite values ('stat_ydensity()').

## Warning: Removed 1 rows containing missing values ('geom_point()').

## Warning in self$trans$transform(x): Se han producido NaNs

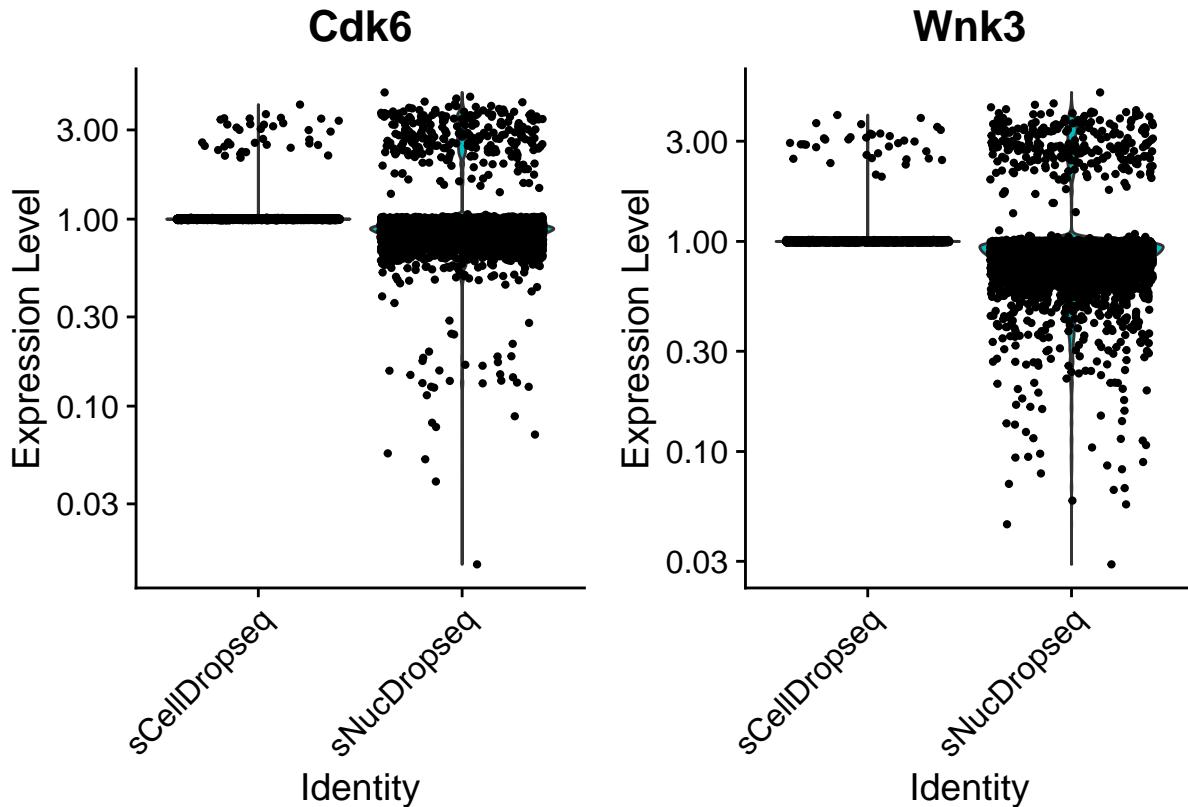
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning in self$trans$transform(x): Se han producido NaNs

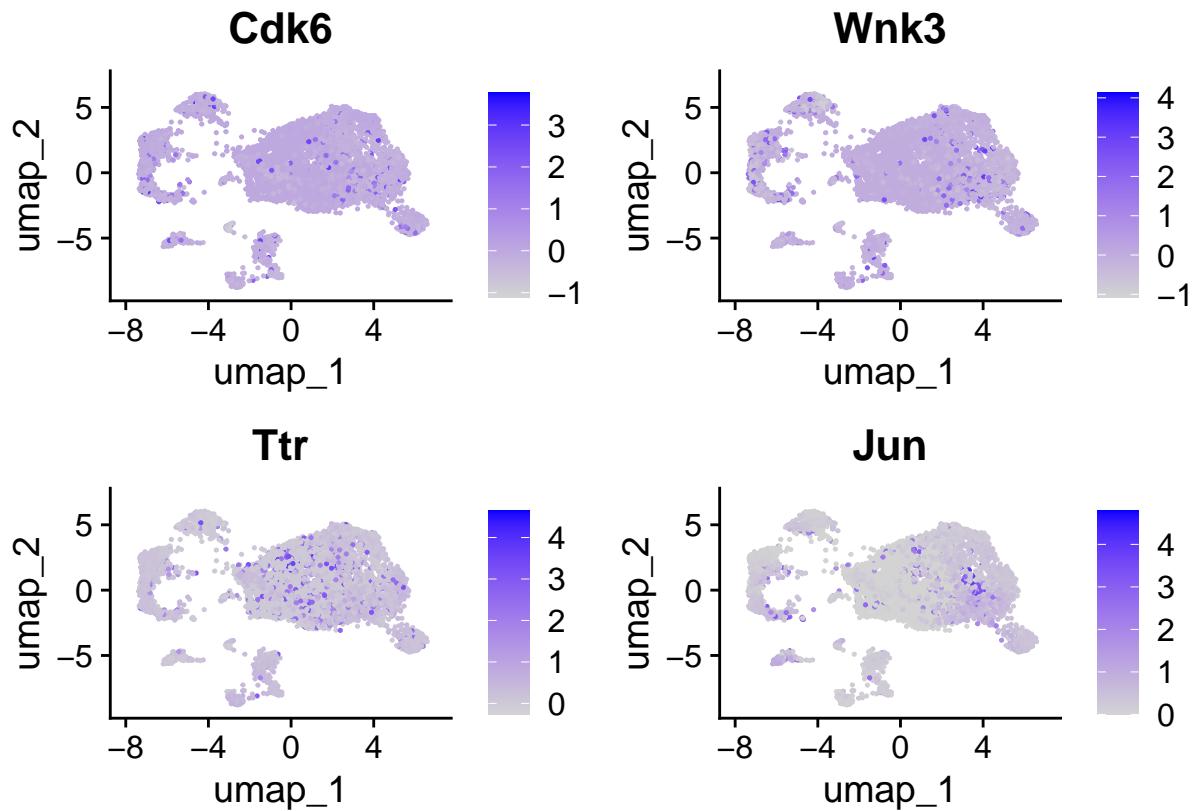
## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 3 rows containing non-finite values ('stat_ydensity()').
```

```
## Warning: Removed 3 rows containing missing values ('geom_point()').
```



```
FeaturePlot(kidney.integrated, features = c("Cdk6", "Wnk3", "Ttr", "Jun"))
```

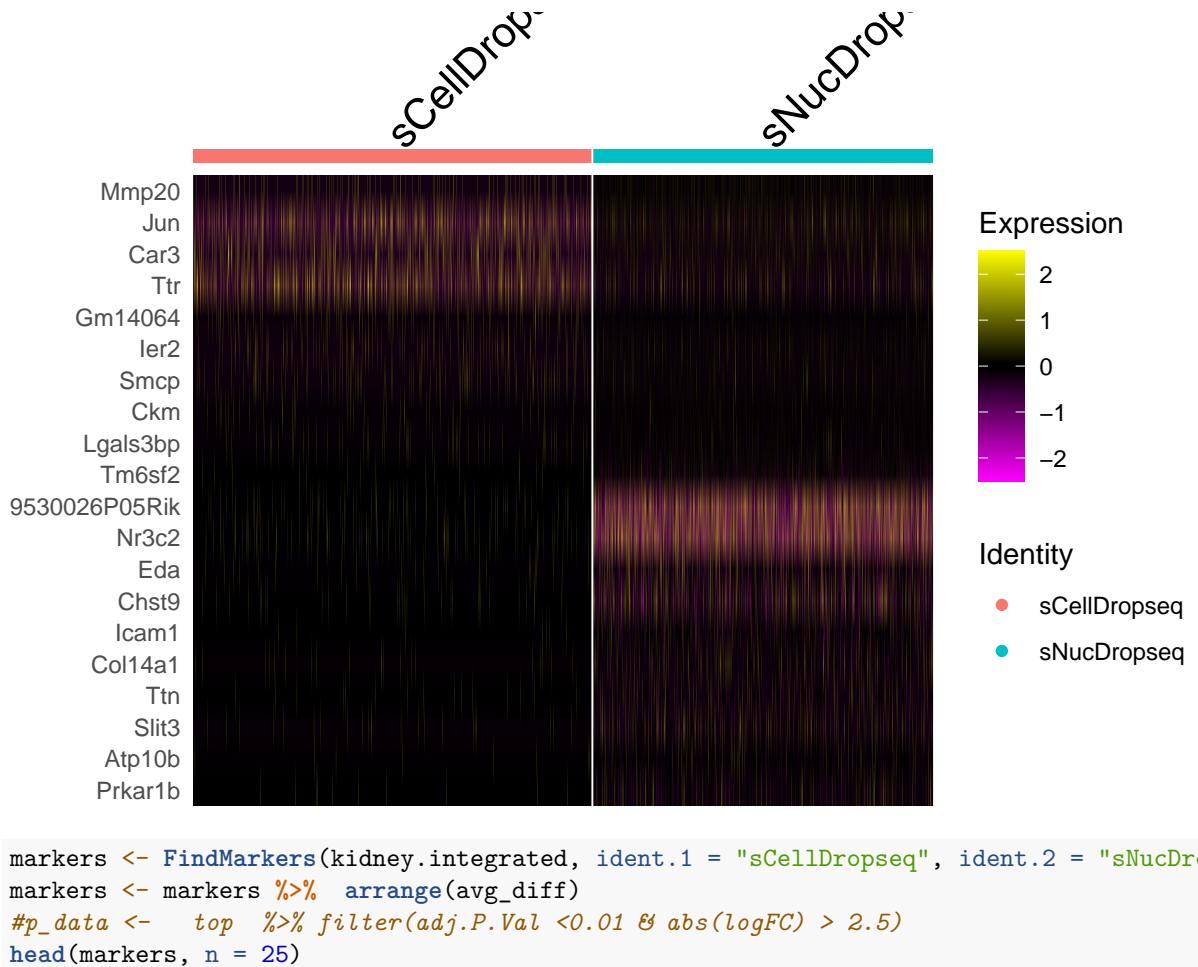


```
top10 <- kidney.integrated.markers %>% group_by(cluster) %>% top_n(n = 10, wt = avg_log2FC)

pdf(paste("heatmap_clusters.pdf", sep=""))
DoHeatmap(kidney.integrated, features = top10$gene)
dev.off()
```

```
## pdf
## 2

DoHeatmap(kidney.integrated, features = top10$gene)
```



```
##          p_val    avg_diff pct.1   pct.2    p_val_adj
## Mecom 1.735360e-25 -0.2796612 0.103 0.360 3.470719e-22
## Mt2 6.393839e-304 -0.2350503 0.065 0.251 1.278768e-300
## Snhg11 5.850032e-55 -0.2229048 0.171 0.426 1.170006e-51
## Kng2 1.834181e-145 -0.2189941 0.137 0.285 3.668363e-142
## Wfdc2 9.060712e-111 -0.2178616 0.089 0.259 1.812142e-107
## Slc8a1 2.471224e-03 -0.2166285 0.039 0.485 1.000000e+00
## Mt1 8.771755e-227 -0.2128257 0.170 0.291 1.754351e-223
## Pkm 2.342084e-46 -0.2121228 0.043 0.253 4.684167e-43
## Kcnj1 1.682105e-04 -0.2090757 0.082 0.263 3.364210e-01
## Egf 2.222853e-77 -0.2060994 0.156 0.397 4.445705e-74
## Slc12a1 1.339821e-57 -0.2051441 0.098 0.415 2.679643e-54
## Klk1 8.165313e-27 -0.2045173 0.168 0.323 1.633063e-23
## Nudt4 4.228941e-110 -0.2039009 0.314 0.330 8.457882e-107
## Mal 7.204670e-120 -0.2011654 0.099 0.258 1.440934e-116
## Erbb4 7.348635e-88 -0.1985328 0.040 0.367 1.469727e-84
## Wnk1 5.753176e-59 -0.1944373 0.166 0.384 1.150635e-55
## Junb 0.000000e+00 -0.1823453 0.022 0.343 0.000000e+00
## Dnase2a 0.000000e+00 -0.1818017 0.023 0.453 0.000000e+00
## Trim56 2.386999e-71 -0.1817064 0.045 0.281 4.773999e-68
## Tmem213 5.294626e-01 -0.1800904 0.069 0.265 1.000000e+00
## Kcnn2 0.000000e+00 -0.1778942 0.082 0.429 0.000000e+00
## PnISR 3.964388e-23 -0.1760151 0.289 0.652 7.928775e-20
```

```

## Rnase4  3.559444e-241 -0.1739652 0.004 0.324 7.118887e-238
## Spp1    2.405022e-103 -0.1729338 0.301 0.421 4.810044e-100
## Epcam   7.469090e-209 -0.1692687 0.074 0.255 1.493818e-205

```

```
write.csv2(markers, "markers.csv")
```

ORA

```

markers<-read.csv2("./markers.csv" )
healthy<-read.delim("/home/nrb/Escritorio/perAnna/20_sc_kidney/healthy.dge.txt")

allEntrezs <- rownames(healthy)
rownames(markers)<-markers$X
selectedEntrezsUP <- markers$X
length(allEntrezs); length(selectedEntrezsUP)

## [1] 19713

## [1] 528

library(clusterProfiler)

## 

## clusterProfiler v4.10.0 For help: https://yulab-smu.top/biomedical-knowledge-mining-book/
## 
## If you use clusterProfiler in published research, please cite:
## T Wu, E Hu, S Xu, M Chen, P Guo, Z Dai, T Feng, L Zhou, W Tang, L Zhan, X Fu, S Liu, X Bo, and G Yu.

## 
## Attaching package: 'clusterProfiler'

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

## The following object is masked from 'package:S4Vectors':
## 
##     rename

## The following object is masked from 'package:purrr':
## 
##     simplify

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

```

```

library(org.Mm,eg.db)

## Loading required package: AnnotationDbi

##
## Attaching package: 'AnnotationDbi'

## The following object is masked from 'package:clusterProfiler':
##
##      select

## The following object is masked from 'package:dplyr':
##
##      select

##


ego <- enrichGO(gene = selectedEntrezsUP,
                 universe = allEntrezs,
                 keyType = "SYMBOL",
                 OrgDb = org.Mm,eg.db,
                 ont = "BP",
                 pAdjustMethod = "BH",
                 qvalueCutoff = 0.01,
                 readable = TRUE)

```

```
head(ego)
```

	ID	Description	GeneRatio	BgRatio
##	GO:0009410	response to xenobiotic stimulus	28/477	295/16635
##	GO:0055064	chloride ion homeostasis	8/477	24/16635
##	GO:0006805	xenobiotic metabolic process	15/477	109/16635
##	GO:0055081	monoatomic anion homeostasis	8/477	27/16635
##	GO:0019373	epoxyxygenase P450 pathway	8/477	32/16635
##	GO:0055075	potassium ion homeostasis	8/477	33/16635
##	pvalue	p.adjust	qvalue	
##	GO:0009410	3.086993e-08	0.0001304872	0.0001161684
##	GO:0055064	2.116189e-07	0.0004472565	0.0003981776
##	GO:0006805	5.014773e-07	0.0006257544	0.0005570884
##	GO:0055081	5.921499e-07	0.0006257544	0.0005570884
##	GO:0019373	2.472867e-06	0.0019941915	0.0017753625
##	GO:0055075	3.182900e-06	0.0019941915	0.0017753625
##				
##	GO:0009410	Umod/Atp1a1/Gstm2/Cyp2j6/Mgst1/Jun/Cyp3a13/S1co1a6/Npas2/Crot/Akr1c13/Prkcb/Gclc/Mat2a/Kcr		
##	GO:0055064			
##	GO:0006805			Gstm2/Cyp2j6/C
##	GO:0055081			
##	GO:0019373			
##	GO:0055075			
##	Count			
##	GO:0009410	28		

```

## GO:0055064      8
## GO:0006805     15
## GO:0055081      8
## GO:0019373      8
## GO:0055075      8

ego_results <- data.frame(ego)
write.csv(ego_results, "clusterProfiler_ORAresults_UpGO.csv")

```

```

pdf(paste("dotplot.pdf",sep=""))
dotplot(ego, showCategory=10)
dev.off()

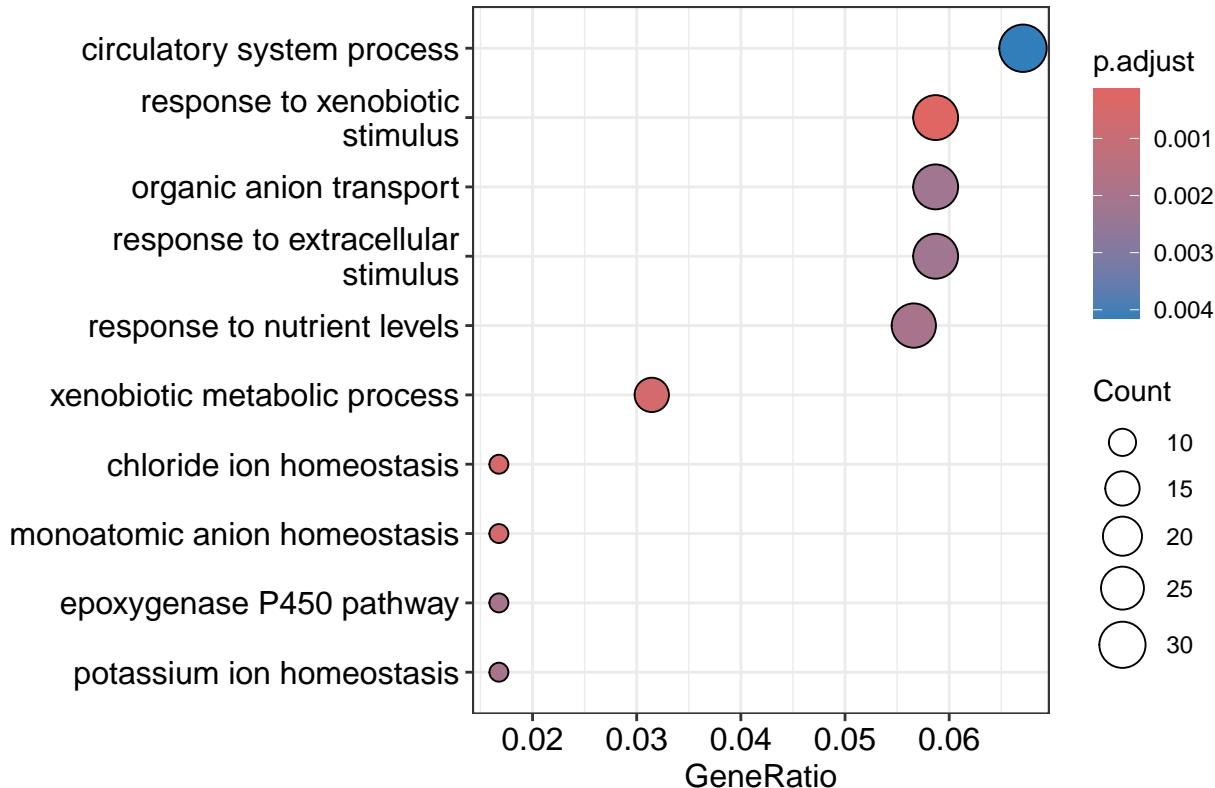
```

```

## pdf
## 2

```

```
dotplot(ego, showCategory=10)
```



```

pdf(paste("GO.pdf",sep=""))
goplot(ego, showCategory=5, cex=0.5)
dev.off()

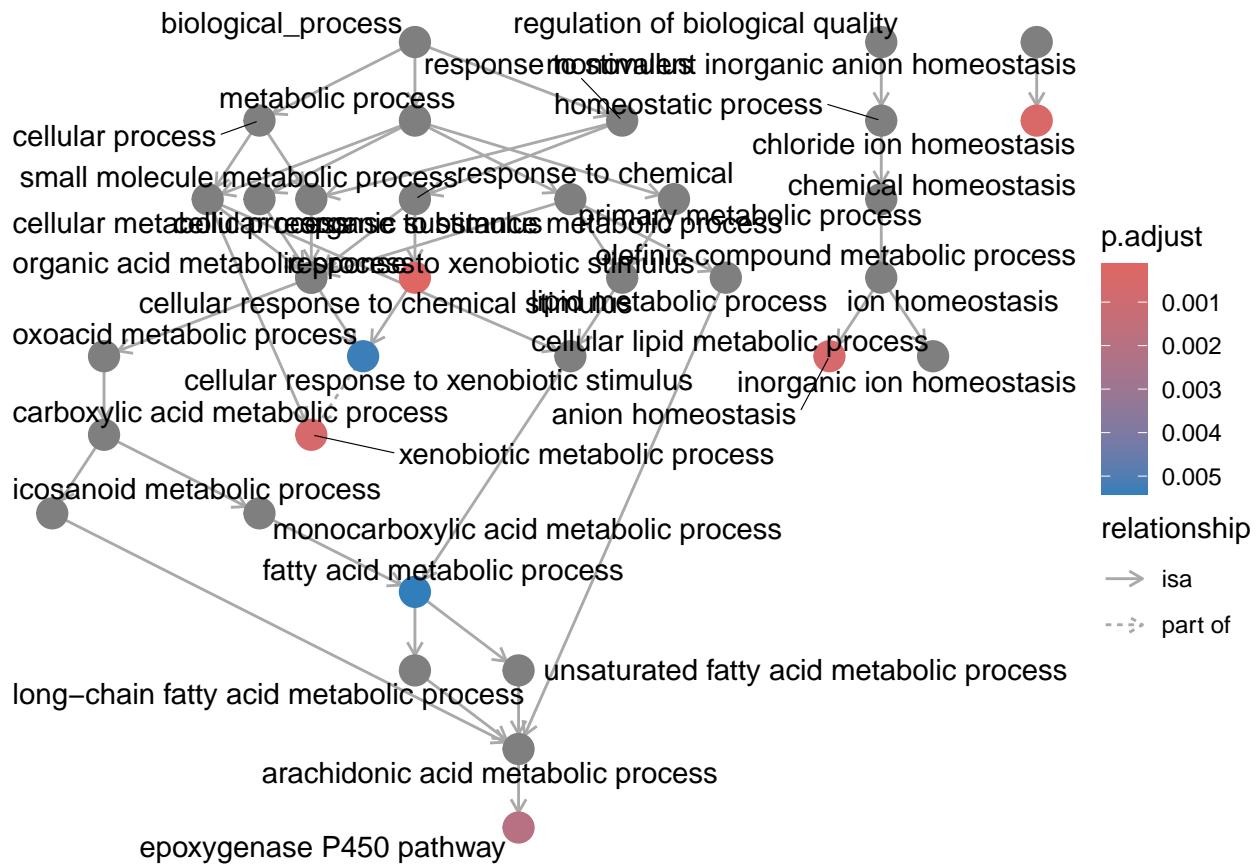
```

```

## pdf
## 2

```

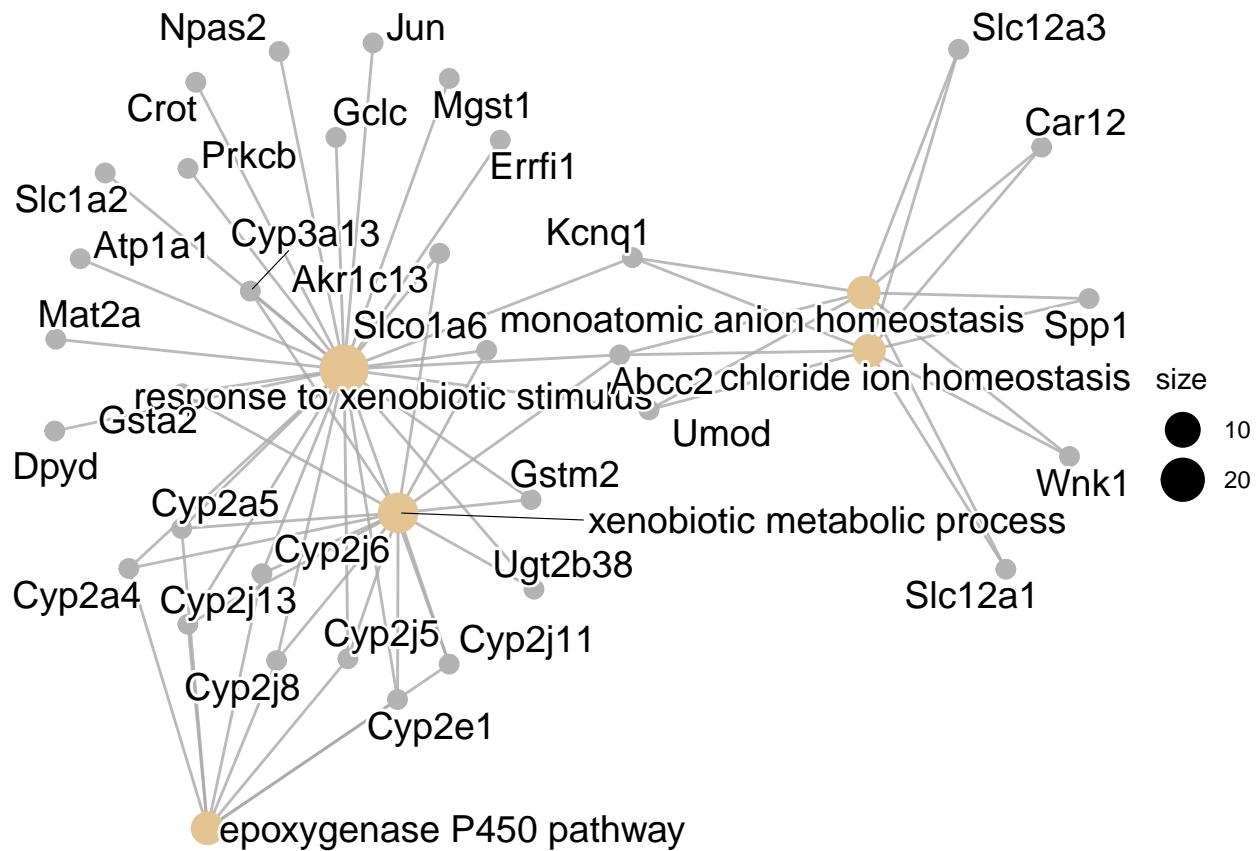
```
goplot(ego, showCategory=5, cex=0.5)
```



```
## Gene network para los términos seleccionados  
pdf(paste("cneplot.pdf",sep=""))  
cnetplot(ego)  
dev.off()
```

```
## pdf  
## 2
```

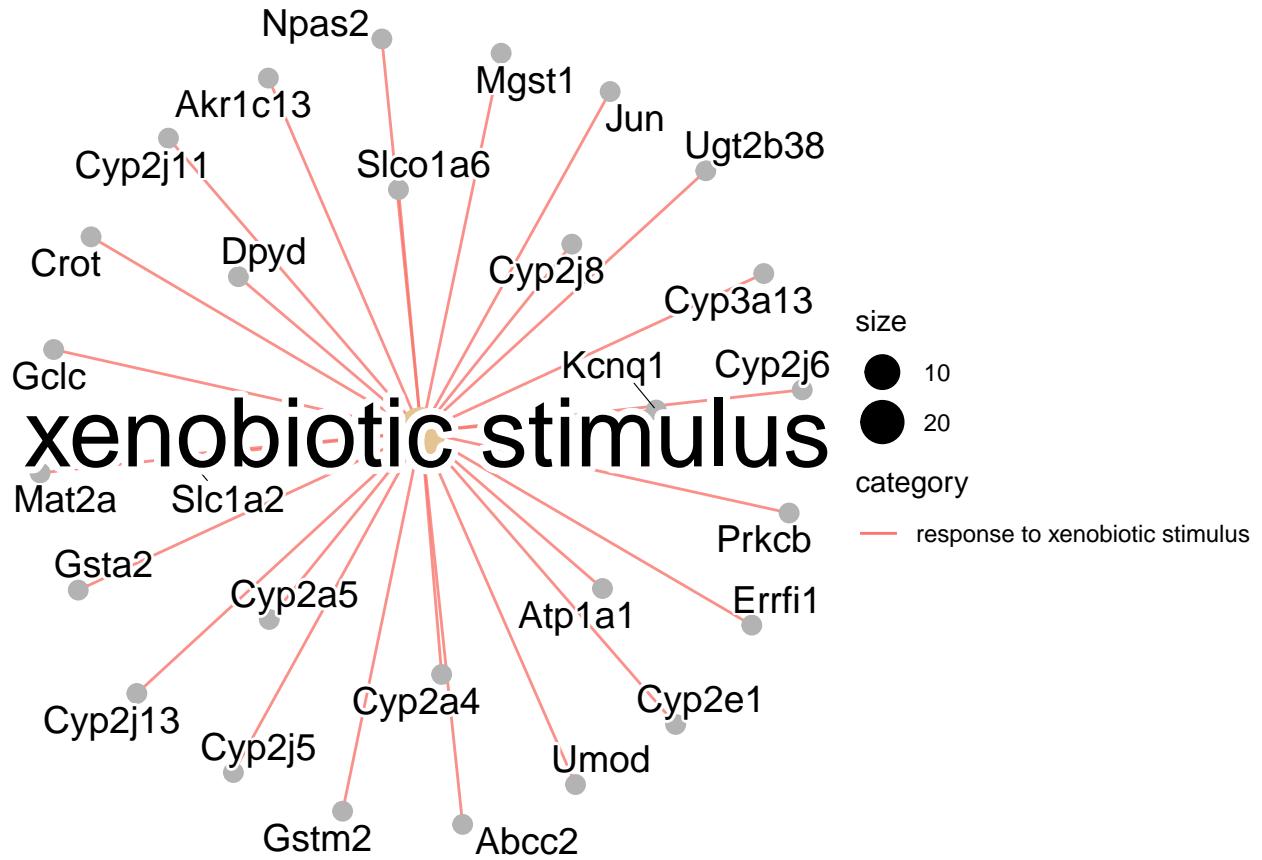
```
cnetplot(ego)
```



```

library(clusterProfiler)
library(ggplot2)
ego2 = clusterProfiler:::simplify(ego, cutoff = 0.01, by = "p.adjust")
png("./cnetplot_transp.png", units = "in", width = 24, height = 16, res = 600,
  bg = "transparent")
par(bg = NA)
a <- cnetplot(ego2, showCategory = 5, cex_category = 1, cex_label_category = 2.5,
  cex_gene = 1, cex_label_gene = 1, circular = FALSE, colorEdge = TRUE)
a
invisible(dev.off())
a

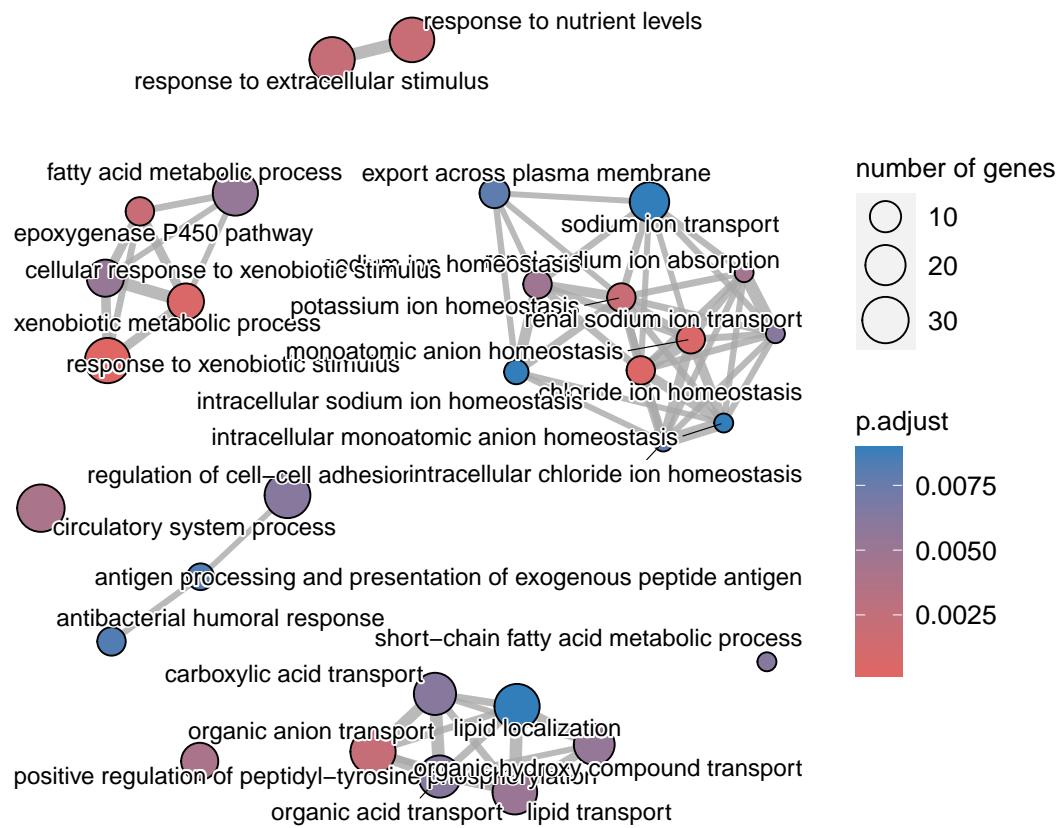
```



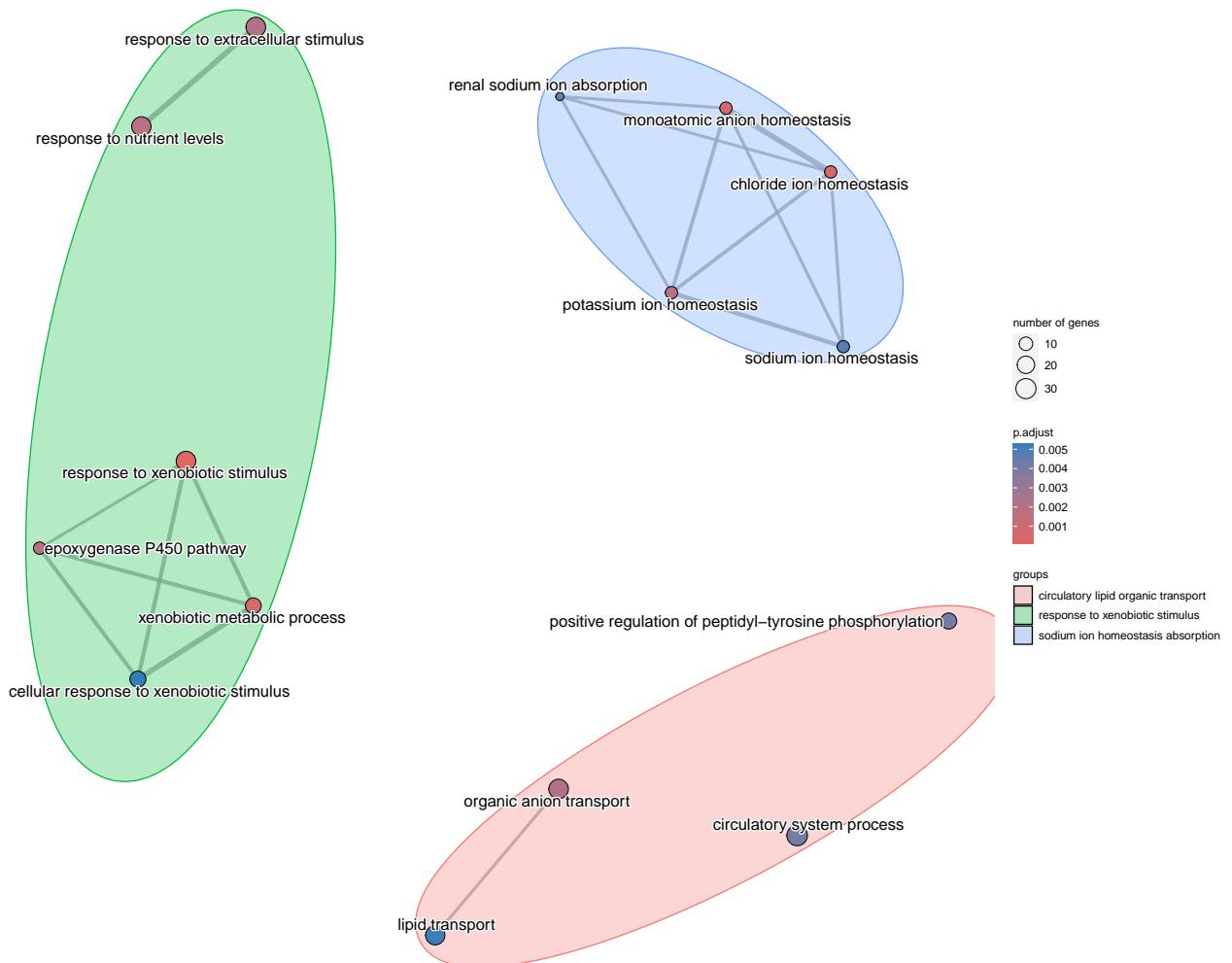
```
## Enrichment Map
library(enrichplot)
ego_sim <- pairwise_termsim(ego)
pdf(paste("emaplot.pdf", sep = ""))
emappplot(ego_sim, cex_label_category=0.6)
dev.off()

## pdf
## 2

emappplot(ego_sim, cex_label_category=0.6)
```



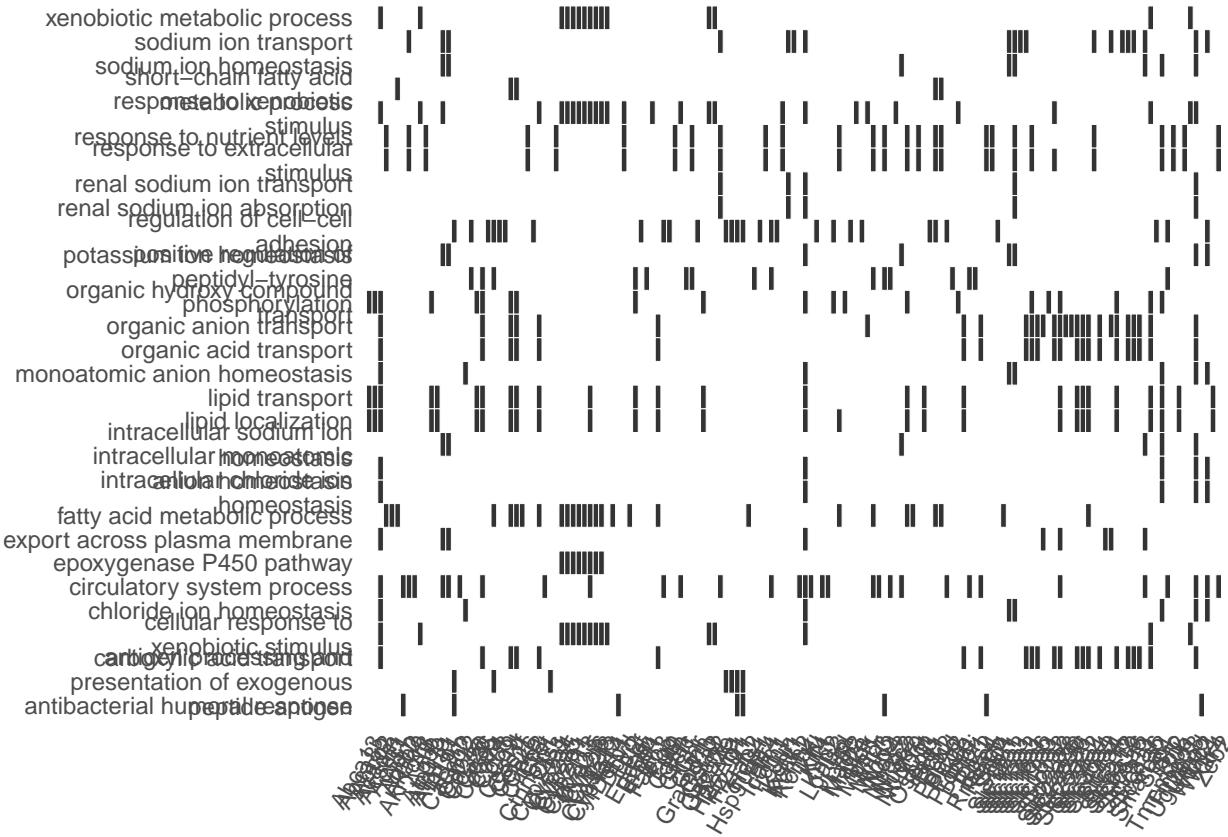
```
term_similarity_matrix = pairwise_termsim(ego)
emapplot(term_similarity_matrix, showCategory = 15, group_category = TRUE, group_legend = TRUE)
```



```
pdf(paste("emaplot_grouped.pdf",sep=""),width = 10, height = 10)
emapplot(term_similarity_matrix, showCategory = TRUE, group_category = TRUE, group_legend = TRUE)
dev.off()
```

```
## pdf
## 2

library(enrichplot)
heatplot(ego)
```



```
pdf(paste("heatplot_ego.pdf",sep=""),width = 18, height = 20)
heatplot(ego)
dev.off()
```

```
## pdf
## 2
```

```
library(geneset)
library(genekitr)
```

```
## Welcome to use genekitr! (Vignette: https://www.genekitr.fun)
## Citation for genekitr:
## Liu, Y., Li, G. Empowering biologists to decode omics data: the Genekitr R package and web server. B

##
## Attaching package: 'genekitr'

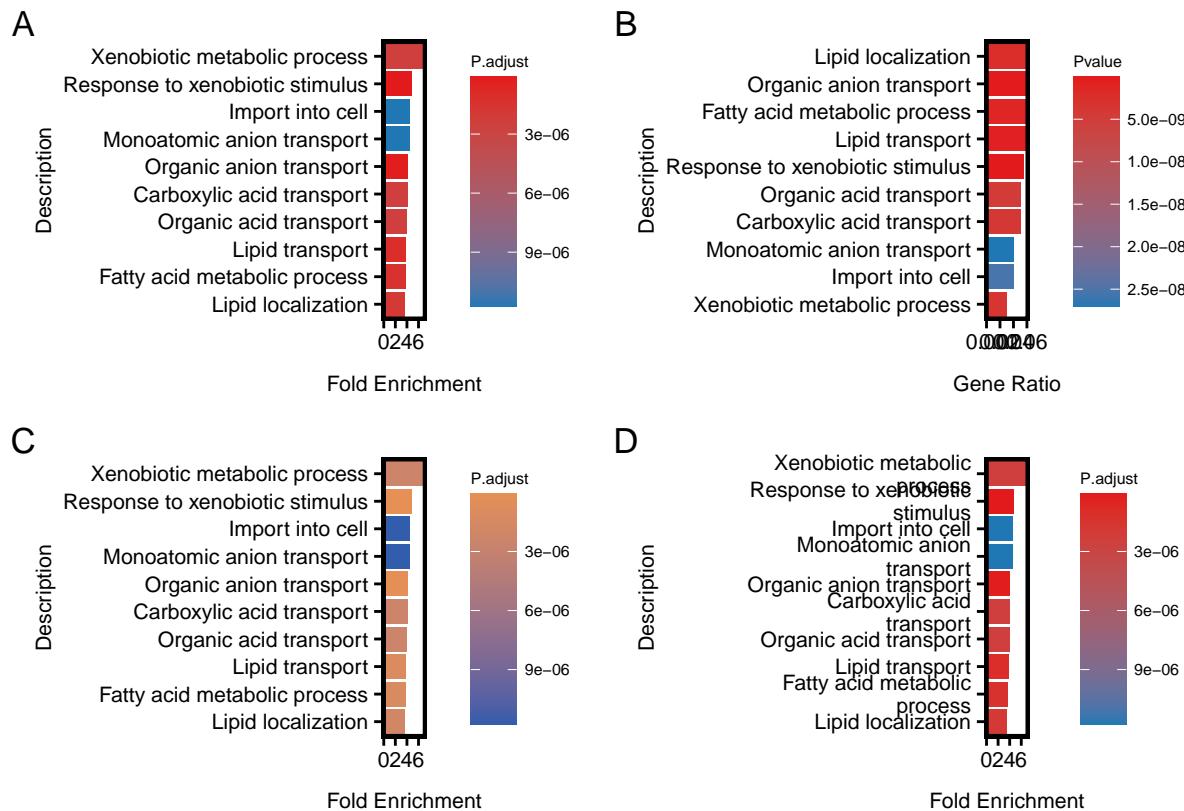
## The following object is masked from 'package:enrichR':
##
##     plotEnrich

mm10_gs <- geneset::getGO(org = "mouse",ont = "bp")

# ORA analysis
ego3 <- genORA(selectedEntrezsUP, geneset = mm10_gs, p_cutoff = 0.01, q_cutoff = 0.01)
```


Bar Plot

```
library(patchwork)
p1 <- plotEnrich(ego3, plot_type = "bar")
p2 <- plotEnrich(ego3, plot_type = "bar", term_metric = "GeneRatio", stats_metric = "pvalue")
p3 <- plotEnrich(ego3, plot_type = "bar", up_color = "#E69056", down_color = "#325CAC")
p4 <- plotEnrich(ego3, plot_type = "bar", wrap_length = 25)
p1 + p2 + p3 + p4 + plot_annotation(tag_levels = "A")
```



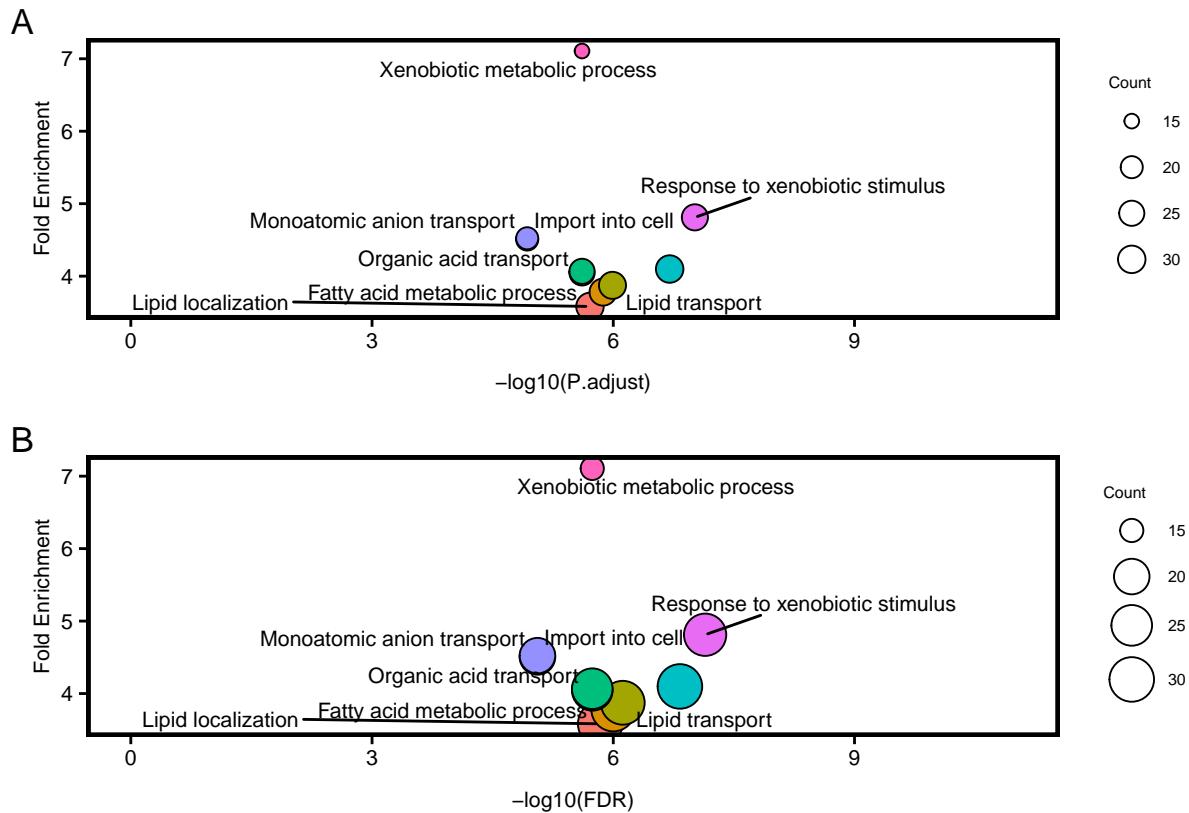
```
pdf(paste("BarPlot.pdf", sep = ""))
p1 + p2 + p3 + p4 + plot_annotation(tag_levels = "A")
dev.off()
```

```
## pdf
## 2
```

Bubble Plot

```
library(patchwork)
p1 <- plotEnrich(ego3, plot_type = "bubble")
p2 <- plotEnrich(ego3, plot_type = "bubble",
                  scale_ratio = 0.5, stats_metric = "qvalue")
p1 / p2 + plot_annotation(tag_levels = "A")
```

```
## Warning: ggrepel: 2 unlabeled data points (too many overlaps). Consider increasing max.overlaps
## ggrepel: 2 unlabeled data points (too many overlaps). Consider increasing max.overlaps
```

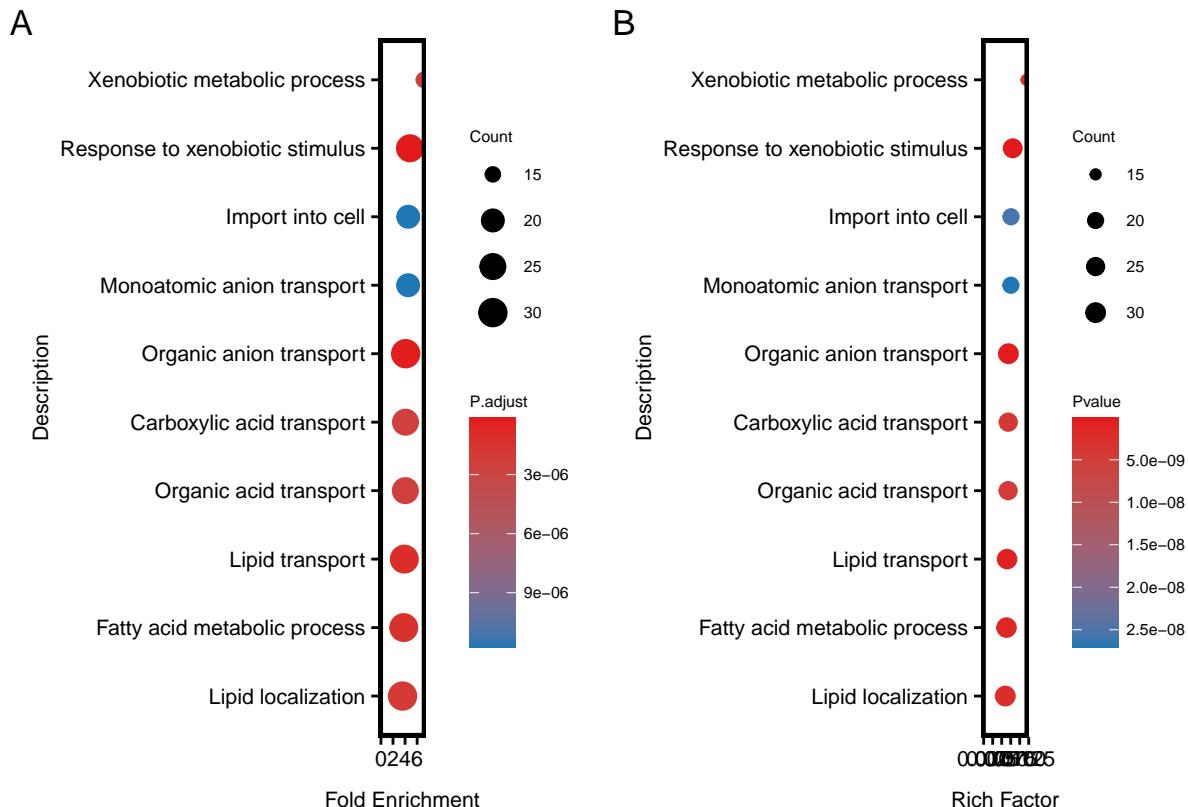


```
pdf(paste("BubblePlot.pdf", sep = ""))
p1 / p2 + plot_annotation(tag_levels = "A")
dev.off()
```

```
## pdf
## 2
```

Dot Plot

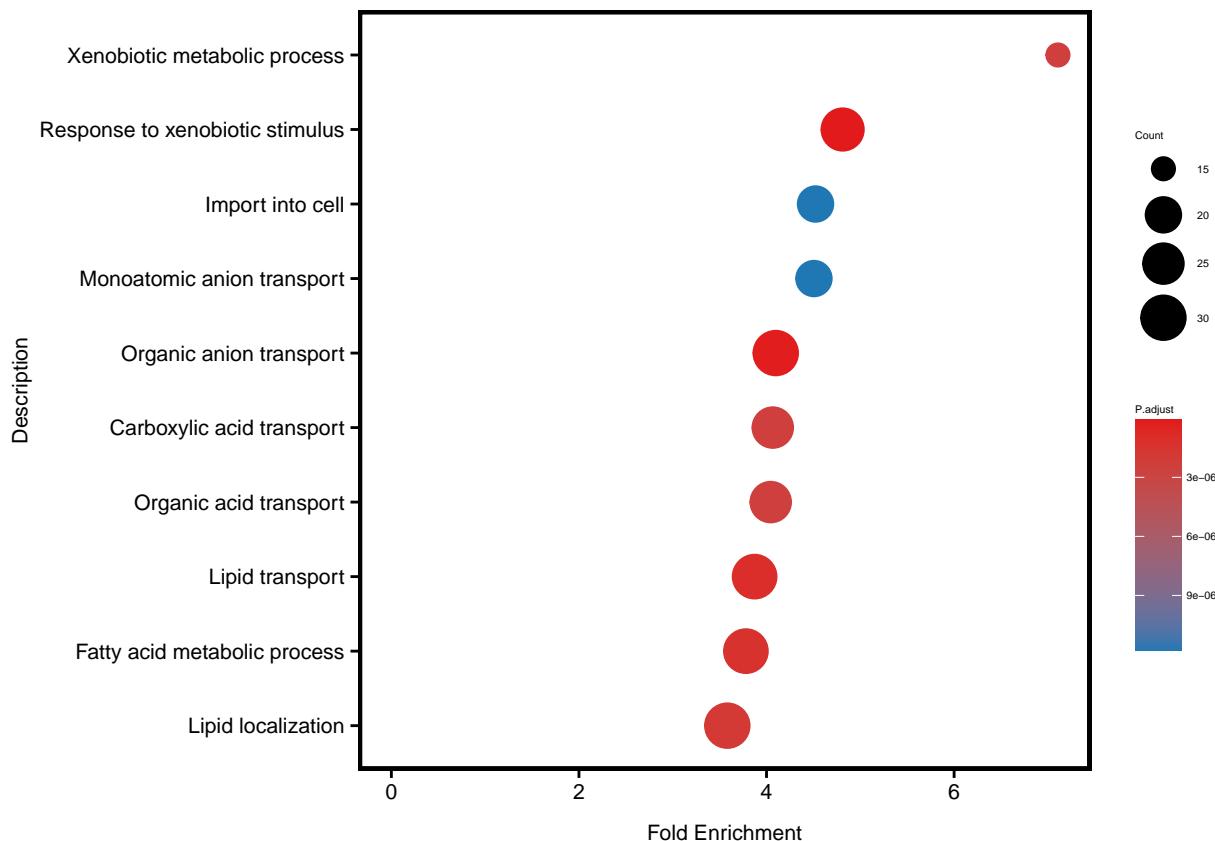
```
library(patchwork)
p1 <- plotEnrich(ego3, plot_type = "dot")
p2 <- plotEnrich(ego3,
  plot_type = "dot",
  scale_ratio = 0.2,
  stats_metric = "pvalue",
  term_metric = "RichFactor"
)
p1 + p2 + plot_annotation(tag_levels = "A")
```



```
pdf(paste("DOTPlot.pdf",sep=""))
p1 + p2 + plot_annotation(tag_levels = "A")
dev.off()
```

```
## pdf
## 2

plotEnrich(ego3,
           plot_type = 'dot',
           scale_ratio = 0.5, # dot size
           main_text_size = 8,
           legend_text_size = 4,
           n_term = 6) # show terms
```



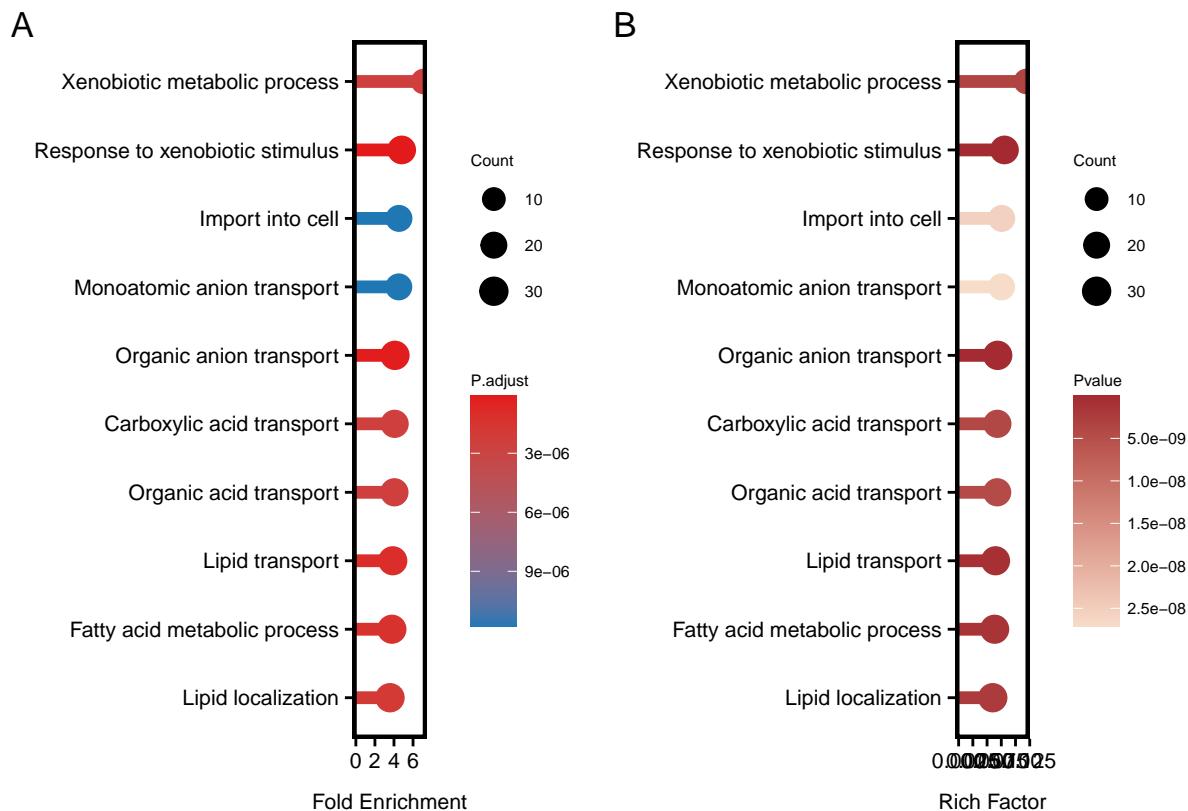
```
pdf(paste("PlotEnrich.pdf",sep=""))
plotEnrich(ego3,
  plot_type = 'dot',
  scale_ratio = 0.5, # dot size
  main_text_size = 8,
  legend_text_size =4,
  n_term = 6) # show terms
dev.off()
```

```
## pdf
## 2
```

Lollipop Plot

```
library(patchwork)
p1 <- plotEnrich(ego3, plot_type = "lollipop")
p2 <- plotEnrich(ego3,
  plot_type = "lollipop",
  scale_ratio = .3,
  stats_metric = "pvalue",
  term_metric = "RichFactor",
  up_color = "#a32a31",
  down_color = "#f7dcca"
```

```
)
p1 + p2 + plot_annotation(tag_levels = "A")
```



```

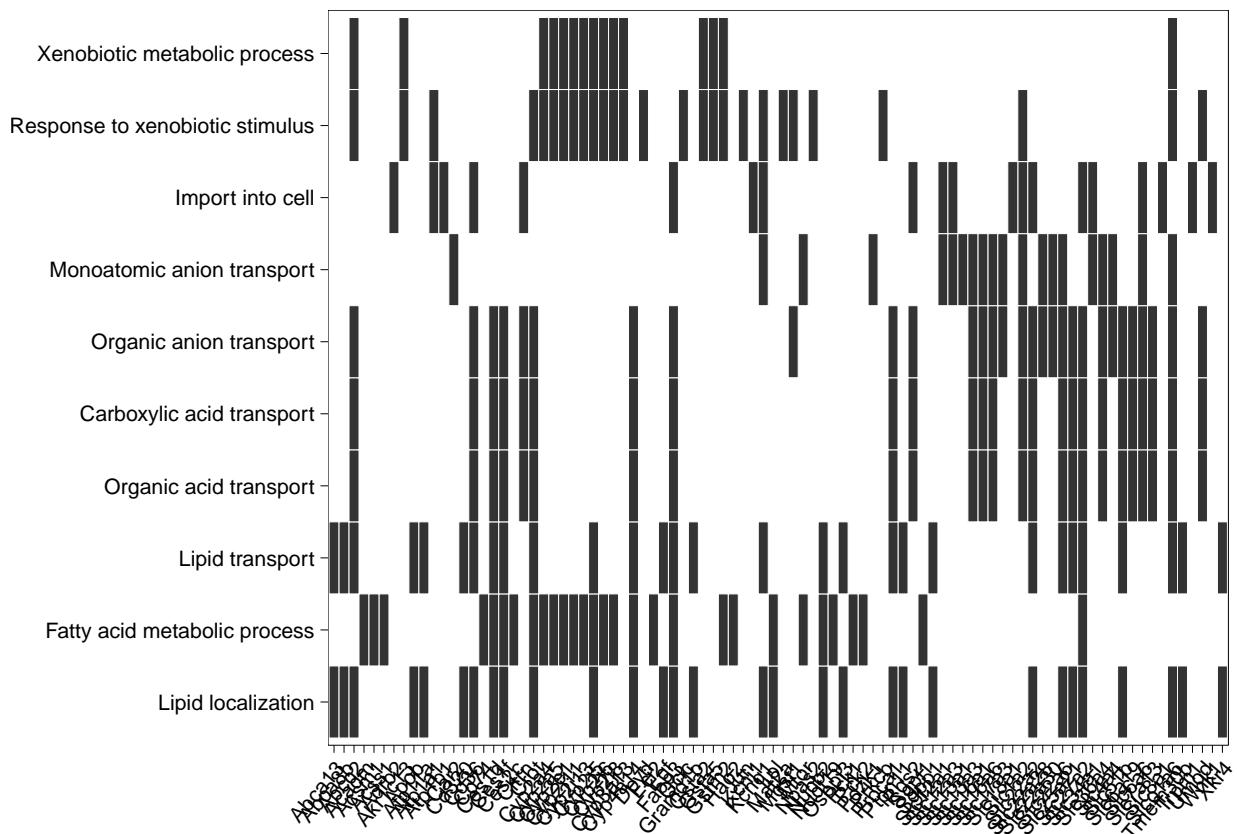
pdf(paste("Lollipop.pdf", sep=""))
p1 + p2 + plot_annotation(tag_levels = "A")
dev.off()
```

```
## pdf
## 2
```

Heatmap Plot

```

library(patchwork)
p1 <- plotEnrich(ego3, plot_type = "geneheat")
show_gene = c('MALAT1', 'ABCC2', 'ABCC2', 'ABCC2')
p2 <- plotEnrich(ego3, plot_type = "geneheat", show_gene = show_gene)
p3 <- plotEnrich(ego3, plot_type = "geneheat", show_gene = show_gene)
p1
```



```
#/ p2 / p3 + plot_annotation(tag_levels = "A")
pdf(paste("Heatmappplot.pdf",sep=""))
p1 + plot_annotation(tag_levels = "A")
dev.off()
```

```
## pdf
## 2
```

Wordcloud Plot

```
plotEnrich(ego3, plot_type = "wordcloud")
```

transport

metabolic
anion stimulus
lipid fatty
cell response
xenobiotic
organic import
carboxylic monoatomic
localization
process

```
## NULL

pdf(paste("Wordcloudplot.pdf",sep=""))
plotEnrich(ego3, plot_type = "wordcloud")
```

```
## NULL
```

```
dev.off()
```

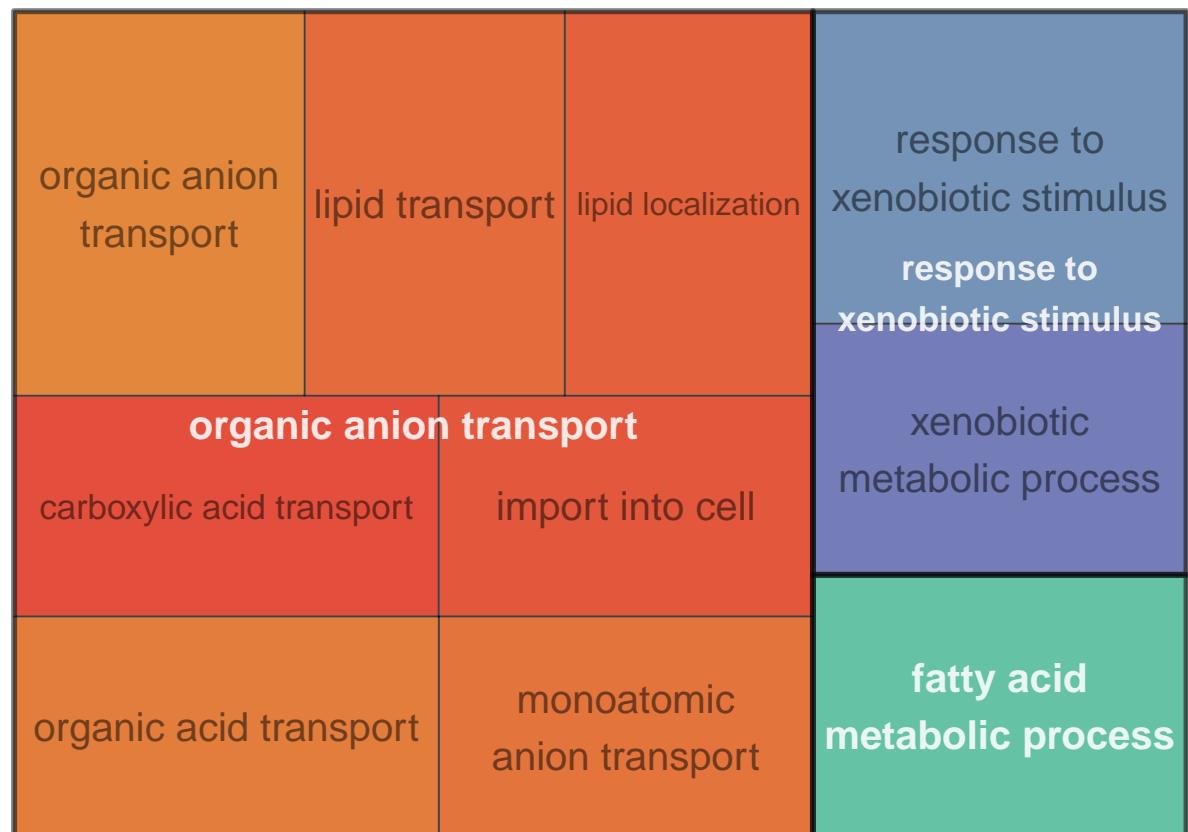
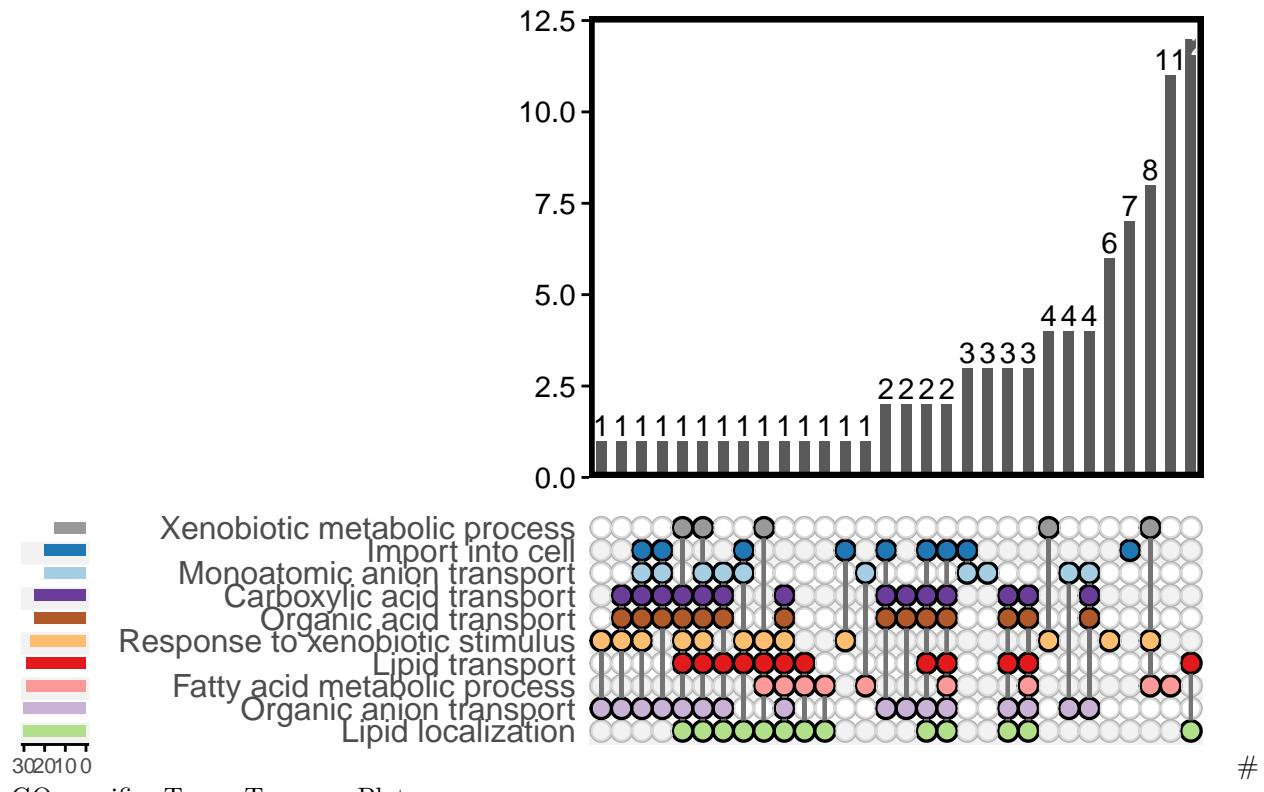
```
## pdf
## 2
```

Upset Plot

```
pdf(paste("Upsetplot.pdf",sep=""))
plotEnrich(ego3, plot_type = "upset",main_text_size = 15,legend_text_size = 8)
dev.off()
```

```
## pdf
## 2
```

```
plotEnrich(ego3, plot_type = "upset",main_text_size = 15,legend_text_size = 8)
```



```

#"bar", "wego", "dot", "bubble", "lollipop", "geneheat", "genechord", "network", "gomap", "goheat", "go

pdf(paste("gotangram.pdf",sep=""))
plotEnrich(ego3, plot_type = "gotangram",main_text_size = 15,legend_text_size = 8, scale_ratio = 0.5, s
dev.off()

```

```

## pdf
## 2

```

Network Plot

```

library(patchwork)
library(igraph)

##
## Attaching package: 'igraph'

## The following object is masked from 'package:clusterProfiler':
##
##      simplify

## The following object is masked from 'package:GenomicRanges':
##
##      union

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

## The following object is masked from 'package:S4Vectors':
##
##      union

## The following objects are masked from 'package:BiocGenerics':
##
##      normalize, path, union

## The following objects are masked from 'package:future':
##
##      %>%, %<-%

## The following objects are masked from 'package:lubridate':
##
##      %--%, union

## The following objects are masked from 'package:purrr':
##
##      compose, simplify

```

```

## The following object is masked from 'package:tidyverse':
##
##     crossing

## The following object is masked from 'package:tibble':
##
##     as_data_frame

## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union

## The following object is masked from 'package:Seurat':
##
##     components

## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

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

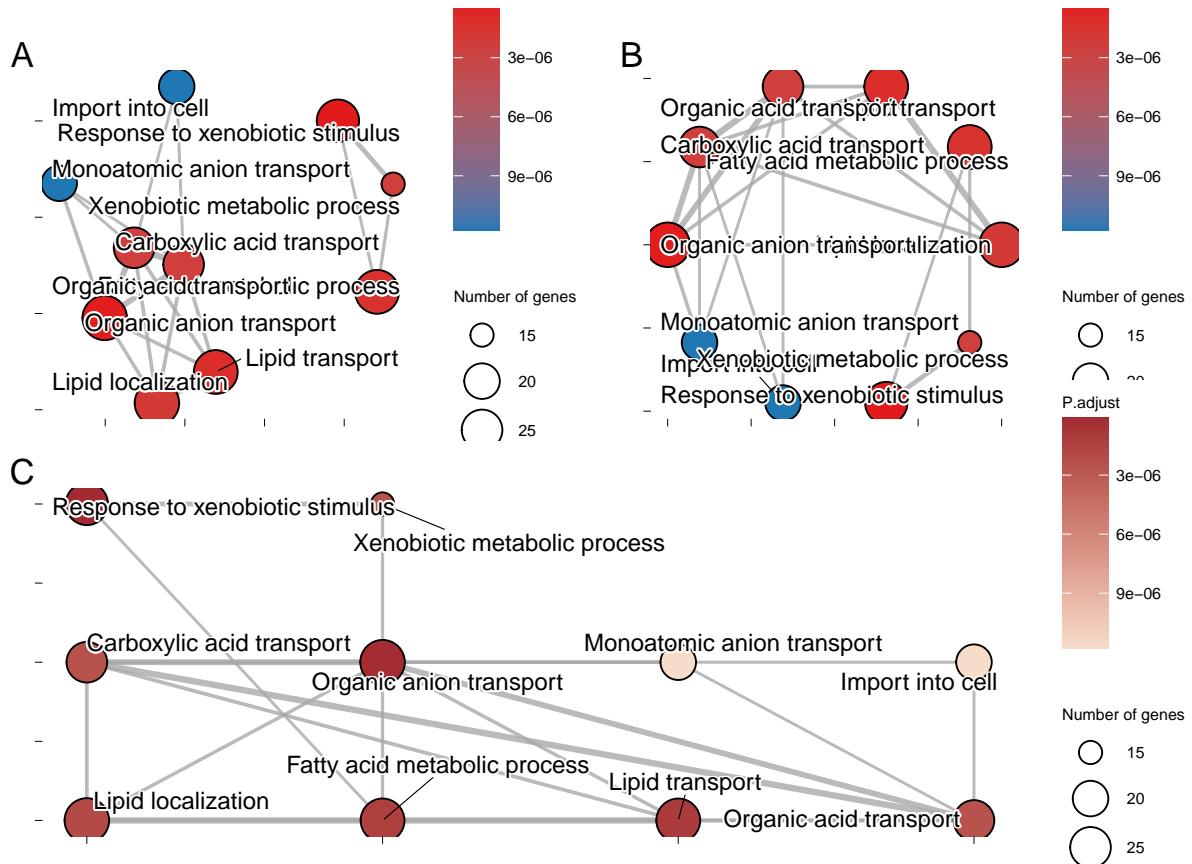
library(ggraph)

##
## Attaching package: 'ggraph'

## The following object is masked from 'package:sp':
##
##     geometry

p1 <- plotEnrich(ego3, plot_type = "network", scale_ratio = 0.5)
p2 <- plotEnrich(ego3, plot_type = "network",
                  layout = "circle", scale_ratio = 0.5)
p3 <- plotEnrich(ego3, plot_type = "network",
                  layout = "grid", sim_method = "Wang",
                  up_color = "#a32a31", down_color = "#f7dcca",
                  scale_ratio = 0.5)
(p1 + p2) / p3 + plot_annotation(tag_levels = "A")

```



```
pdf(paste("Network.pdf", sep=""))
(p1 + p2) / p3 + plot_annotation(tag_levels = "A")
dev.off()
```

```
## pdf
## 2
```

GO-specific: WEGO Plot

```
# 1st step: prepare input IDs
data(geneList, package = "genekitr")
id <- names(geneList)[abs(geneList) > 2]
id<-rownames(markers)

# 2nd step: prepare CC and MF gene sets
go_cc <- geneset::getGO(org = "mouse", ont = "cc")
go_mf <- geneset::getGO(org = "mouse", ont = "mf")

# 3rd step: analysis
ego_cc <- genORA(id, geneset = go_cc)

## Warning in max(n_ent): ningun argumento finito para max; retornando -Inf
```

```

## Warning in min(n_ent): ningún argumento finito para min; retornando Inf
## Warning in min(n_ent): ningún argumento finito para min; retornando Inf

## Warning in max(n_ent): ningun argumento finito para max; retornando -Inf
## Warning in min(n_ent): ningún argumento finito para min; retornando Inf
## Warning in min(n_ent): ningún argumento finito para min; retornando Inf

ego_mf <- genORA(id, geneset = go_mf)

## Warning in max(n_ent): ningun argumento finito para max; retornando -Inf
## Warning in max(n_ent): ningún argumento finito para min; retornando Inf
## Warning in max(n_ent): ningún argumento finito para min; retornando Inf

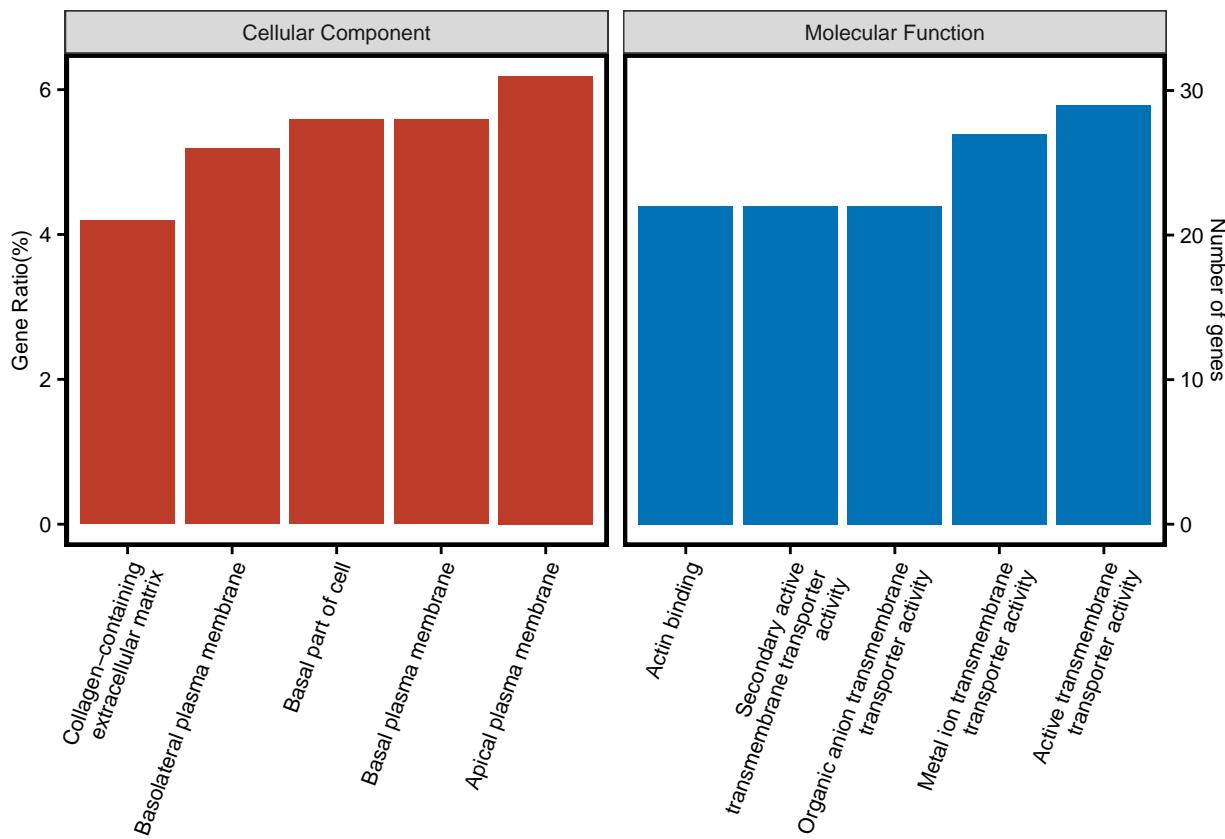
## Warning in max(n_ent): ningun argumento finito para max; retornando -Inf
## Warning in min(n_ent): ningún argumento finito para min; retornando Inf
## Warning in min(n_ent): ningún argumento finito para min; retornando Inf

# 4th step: merge two data frames
# Note: each data frame should add new column "Ontology"
ego_cc <- ego_cc %>% dplyr::mutate(Ontology = "cc") %>% dplyr::rename(ID = 1)
ego_mf <- ego_mf %>% dplyr::mutate(Ontology = "mf") %>% dplyr::rename(ID = 1)

all_ego <- rbind(ego_cc,ego_mf)

plotEnrich(all_ego, plot_type = "wego", n_term = 5)

```



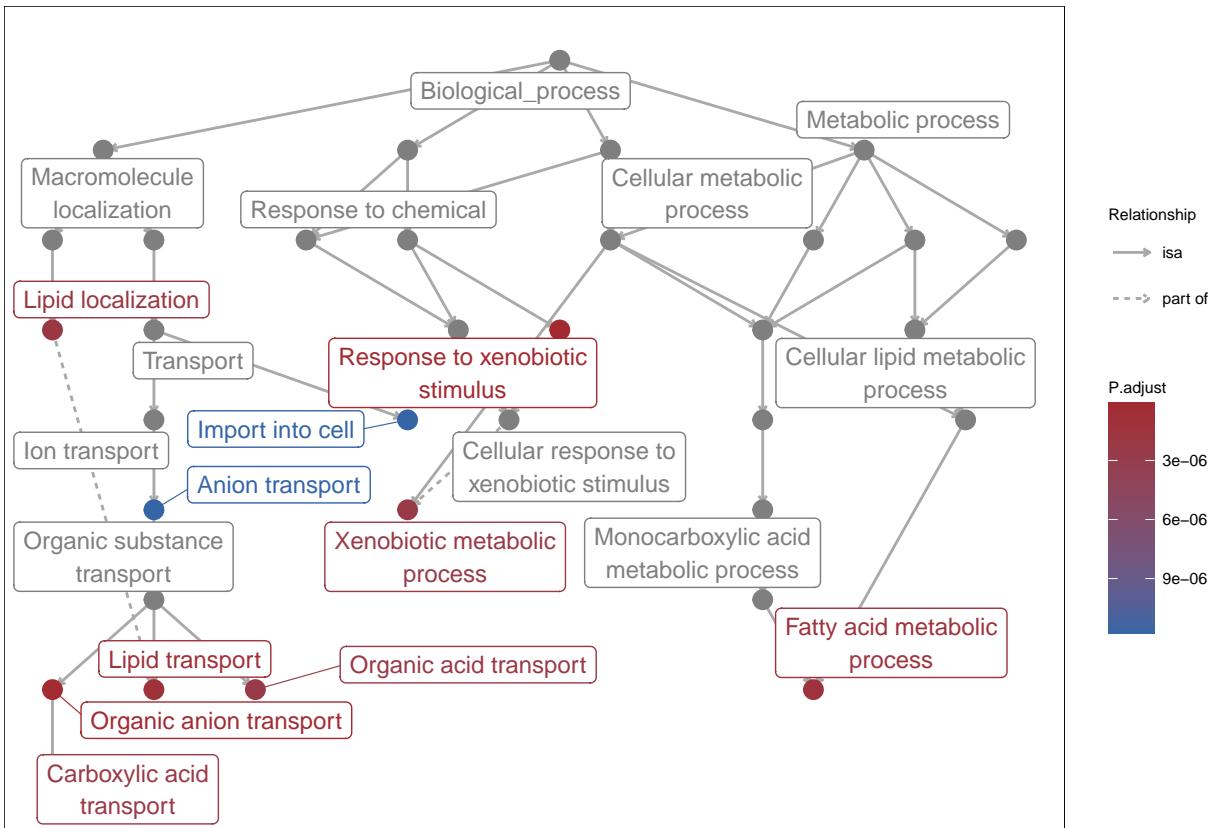
```
pdf(paste("WEGOPlot.pdf", sep=""))
plotEnrich(all_ego, plot_type = "wego", n_term = 5)
dev.off()
```

```
## pdf
## 2
```

GO-specific: Map Plot

```
library(igraph)
library(ggraph)
plotEnrich(ego3, plot_type = "gomap", wrap_length = 25,
           up_color = '#a32a31', down_color = '#3665a6')
```

```
## Warning: Removed 13 rows containing missing values ('geom_label_repel()').
```



```
pdf(paste("MapPlot.pdf", sep=""))
plotEnrich(ego3, plot_type = "gomap", wrap_length = 25,
           up_color = '#a32a31', down_color = '#3665a6')
```

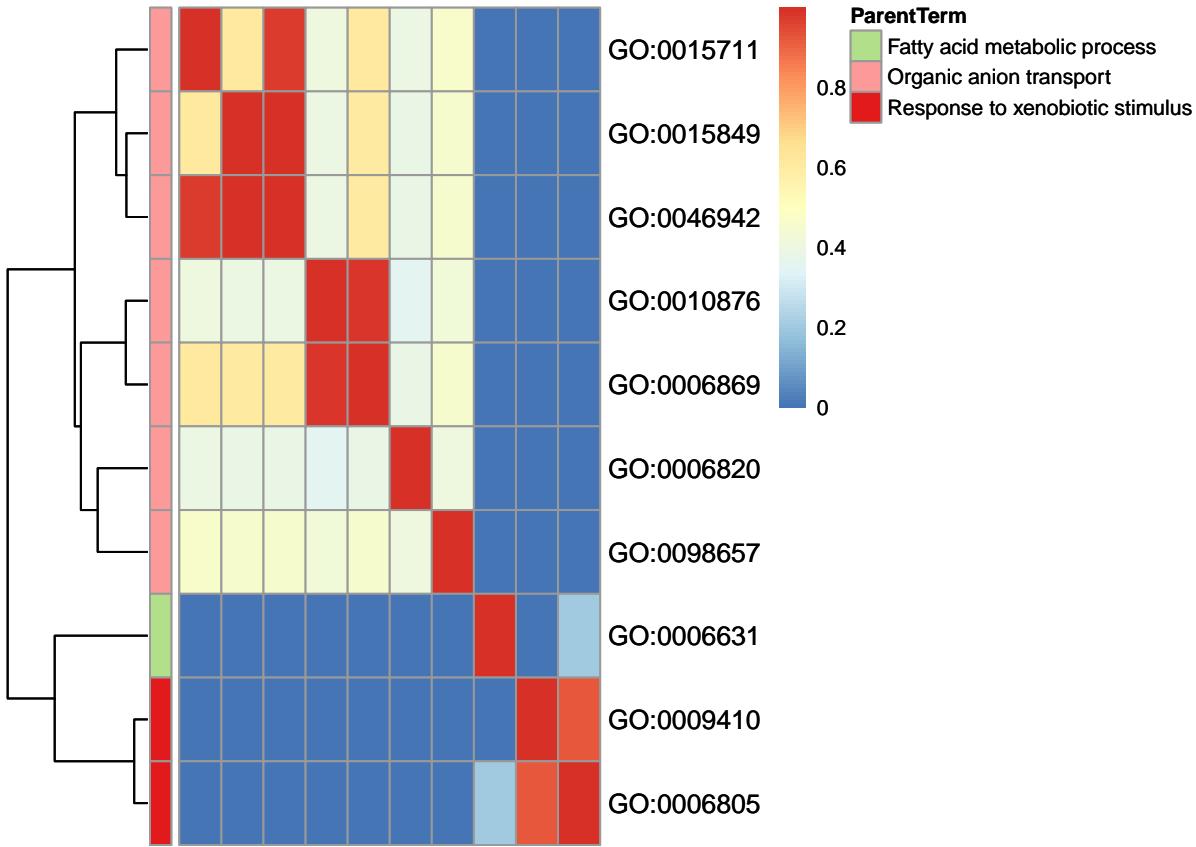
```
## Warning: Removed 13 rows containing missing values ('geom_label_repel()').
```

```
dev.off()
```

```
## pdf
## 2
```

GO-specific: Terms Heatmap Plot

```
plotEnrich(ego3, plot_type = "goheat", sim_method = "Rel")
```



```
pdf(paste("TermsHeatmap.pdf", sep=""))
plotEnrich(ego3, plot_type = "goheat", sim_method = "Rel")
dev.off()
```

```
## pdf
## 3
```

Plot Theme

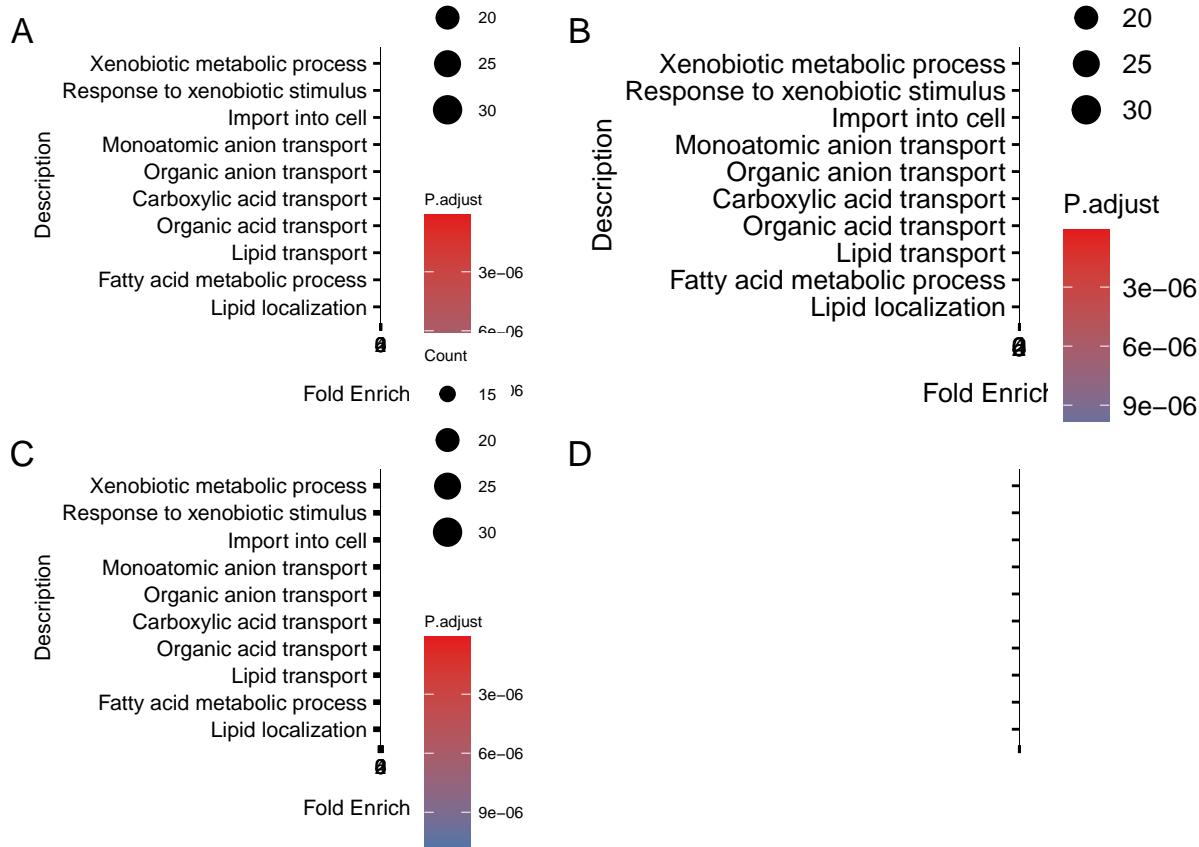
```
library(patchwork)
p1 <- plotEnrich(ego3, plot_type = "dot")
p2 <- plotEnrich(ego3,
  plot_type = "dot",
  main_text_size = 10,
  legend_text_size = 10
)

p3 <- plotEnrich(ego3,
  plot_type = "dot",
  border_thick = 3,
  remove_grid = F
)
```

```

p4 <- plotEnrich(ego3,
  plot_type = "dot",
  remove_main_text = T,
  remove_legend_text = T,
  remove_legend = T
)
p1 + p2 + p3 + p4 + plot_annotation(tag_levels = "A")

```



```

pdf(paste("PlotTheme.pdf", sep=""))
p1 + p2 + p3 + p4 + plot_annotation(tag_levels = "A")
dev.off()

```

```

## pdf
## 2

```

Advanced Plot

```

# 1st step: prepare input IDs
# Since the geneList is logFC decreasing ordered, we could take first 100 as up-regulated genes and vice versa
markers <- markers[order(markers$avg_diff, decreasing = TRUE), ]
up_genes <- rownames(markers[markers$avg_diff > 0.3, ])
down_genes <- rownames(markers[markers$avg_diff < -0.2, ])

```

```

# 2nd step: prepare gene set
mm_gs <- geneset::getGO(org = "mouse", ont = "bp")

# 3rd step: ORA analysis separately
up_go <- genORA(up_genes, geneset = mm_gs)
down_go <- genORA(down_genes, geneset = mm_gs)

dim(up_go)

```

```
## [1] 12 12
```

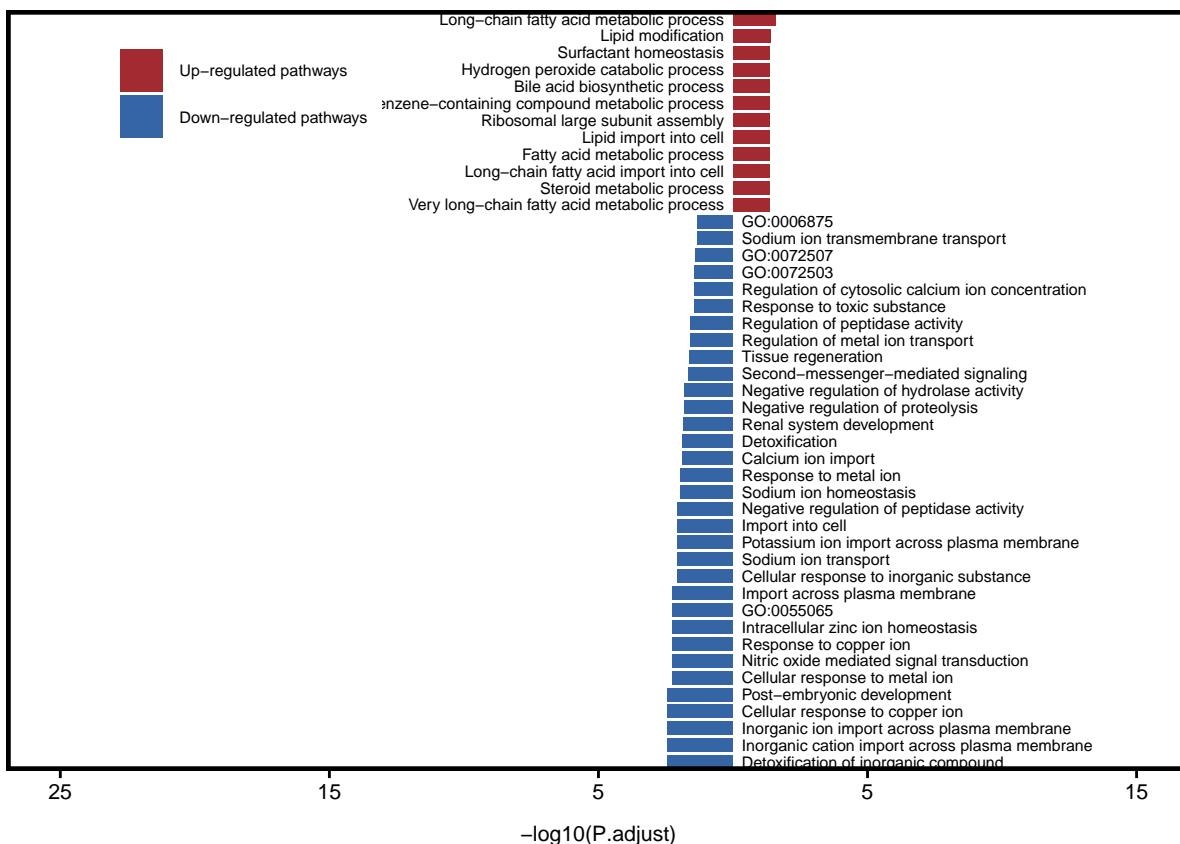
```
dim(down_go)
```

```
## [1] 33 12
```

```

plotEnrichAdv(up_go, down_go,
              plot_type = "one",
              term_metric = "FoldEnrich",
              stats_metric = "p.adjust",
              xlim_left = 25, xlim_right = 15) +
  theme(legend.position = c(0.2, 0.9))

```



```

pdf(paste("Advanced.pdf",sep=""))
plotEnrichAdv(up_go, down_go,
              plot_type = "one",
              term_metric = "FoldEnrich",
              stats_metric = "p.adjust",
              xlim_left = 25, xlim_right = 15) +
  theme(legend.position = c(0.2, 0.9))
dev.off()

```

```

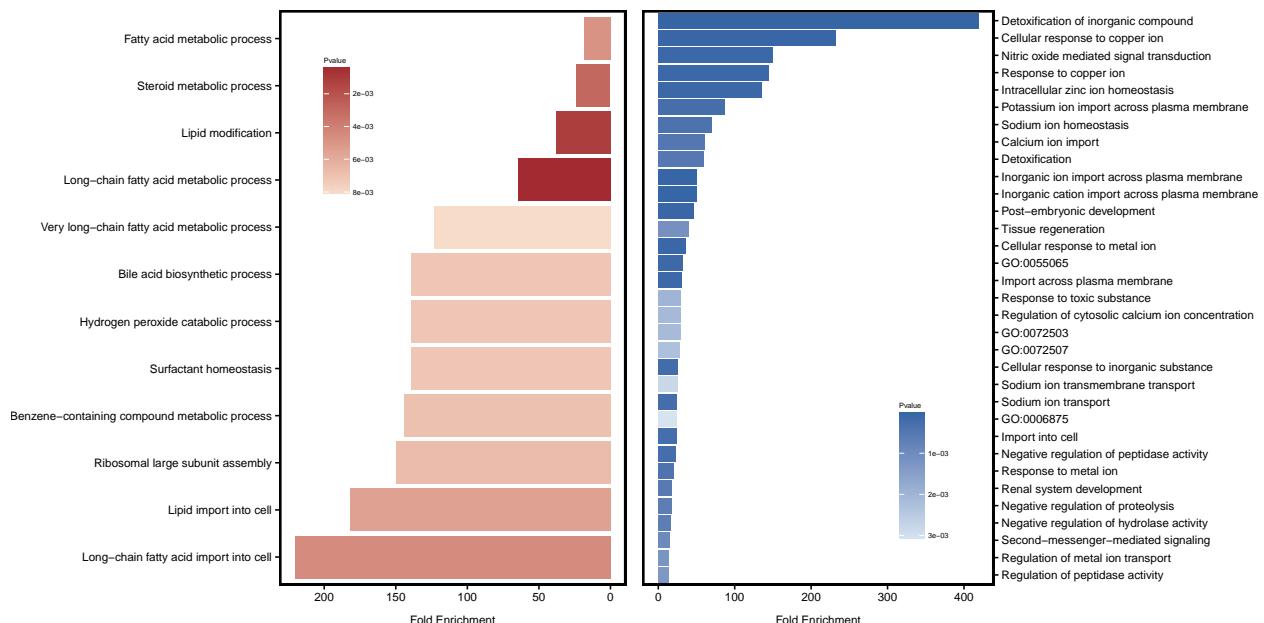
## pdf
## 2

```

```

plotEnrichAdv(up_go, down_go,
              plot_type = "two",
              term_metric = "FoldEnrich",
              stats_metric = "pvalue",
              legend_text_size = 5
) +
  theme(legend.position = "none")

```



```

pdf(paste("Advanced2.pdf",sep=""))
plotEnrichAdv(up_go, down_go,
              plot_type = "two",
              term_metric = "FoldEnrich",
              stats_metric = "pvalue",
              legend_text_size = 5
) +
  theme(legend.position = "none")
dev.off()

```

```

## pdf
## 2

```