# QC_analysis

2023-11-19

protoplast_gene_file="reference/proto_genes.txt"

# default parameters

min_genes=200; min.cells = 3 min_UMIs=500; MAX percent_mito = 5 MAX percent_chloro = 5 PCA_dimensions=20 clustering_resolution=.6 nof_integration_features=5000

```
set.seed(12345)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

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

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

```
library(Seurat)
```

```
## Loading required package: SeuratObject
```

```
## Loading required package: sp
```

```
##
## Attaching package: 'SeuratObject'
```

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

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ forcats   1.0.0     ✔ readr     2.1.4
## ✔ ggplot2   3.4.4     ✔ stringr   1.5.1
## ✔ lubridate 1.9.3     ✔ tibble    3.2.1
## ✔ purrr     1.0.2     ✔ tidyr     1.3.0
```

```
## ── Conflicts ─────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(cowplot)
```

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

```
library(patchwork)
```

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

```
library(Rcpp)
library(Matrix)
```

```
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(harmony)
library(monocle3)
```

```
## Loading required package: Biobase
## Loading required package: BiocGenerics
##
## Attaching package: 'BiocGenerics'
##
## The following objects are masked from 'package:lubridate':
##
##     intersect, setdiff, union
##
## The following object is masked from 'package:SeuratObject':
##
##     intersect
##
## The following objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union
##
## 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
##
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
##
## Loading required package: SingleCellExperiment
```

```
## Warning: package 'SingleCellExperiment' was built under R version 4.3.2
```

```
## Loading required package: SummarizedExperiment
```

```
## Warning: package 'SummarizedExperiment' was built under R version 4.3.2
```

```
## Loading required package: MatrixGenerics
## Loading required package: matrixStats
##
## Attaching package: 'matrixStats'
##
## The following objects are masked from 'package:Biobase':
##
##     anyMissing, rowMedians
##
## 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
##
## The following object is masked from 'package:Biobase':
##
##     rowMedians
##
## Loading required package: GenomicRanges
## Loading required package: stats4
## Loading required package: S4Vectors
##
## Attaching package: 'S4Vectors'
##
## The following objects are masked from 'package:Matrix':
##
##     expand, unname
##
## 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 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 object is masked from 'package:sp':
##
##     %over%
##
## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice
##
## Loading required package: GenomeInfoDb
```

```
## Warning: package 'GenomeInfoDb' was built under R version 4.3.2
```

```
##
## Attaching package: 'SummarizedExperiment'
##
## The following object is masked from 'package:Seurat':
##
##     Assays
##
## The following object is masked from 'package:SeuratObject':
##
##     Assays
##
##
## Attaching package: 'monocle3'
##
## The following objects are masked from 'package:Biobase':
##
##     exprs, fData, fData<-, pData, pData<-
```

QC & filter

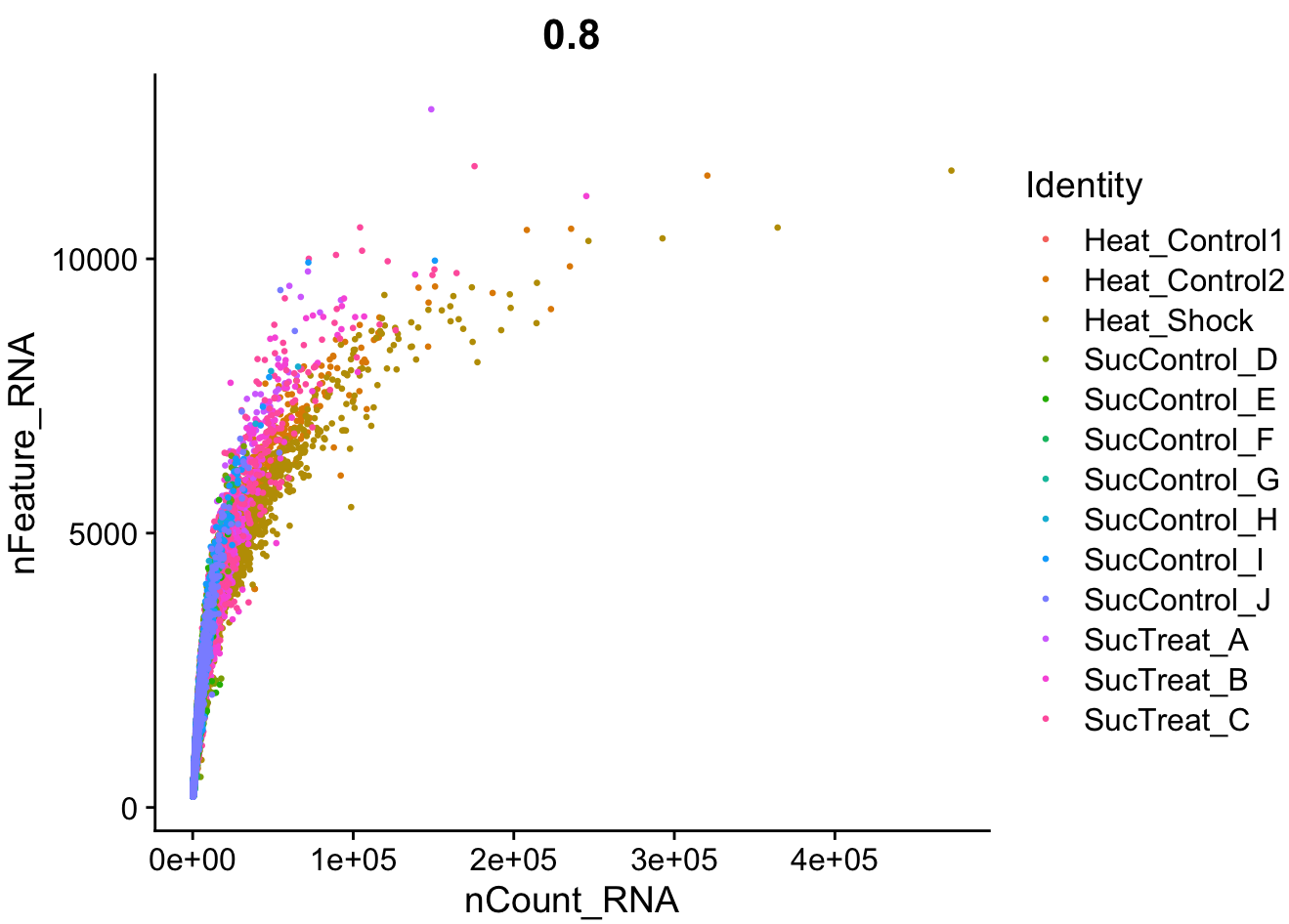
```
##only consider cells with at least 200 detected genes and genes need to be expressed in at least 3 cells.

alldata <- readRDS("alldata.rds")
selected_c <- WhichCells(alldata, expression = nFeature_RNA > 200)
selected_f <- rownames(alldata)[Matrix::rowSums(alldata) > 3]
data.filt <- subset(alldata, features = selected_f, cells = selected_c)
dim(data.filt)
```

```
## [1] 33907 47066
```

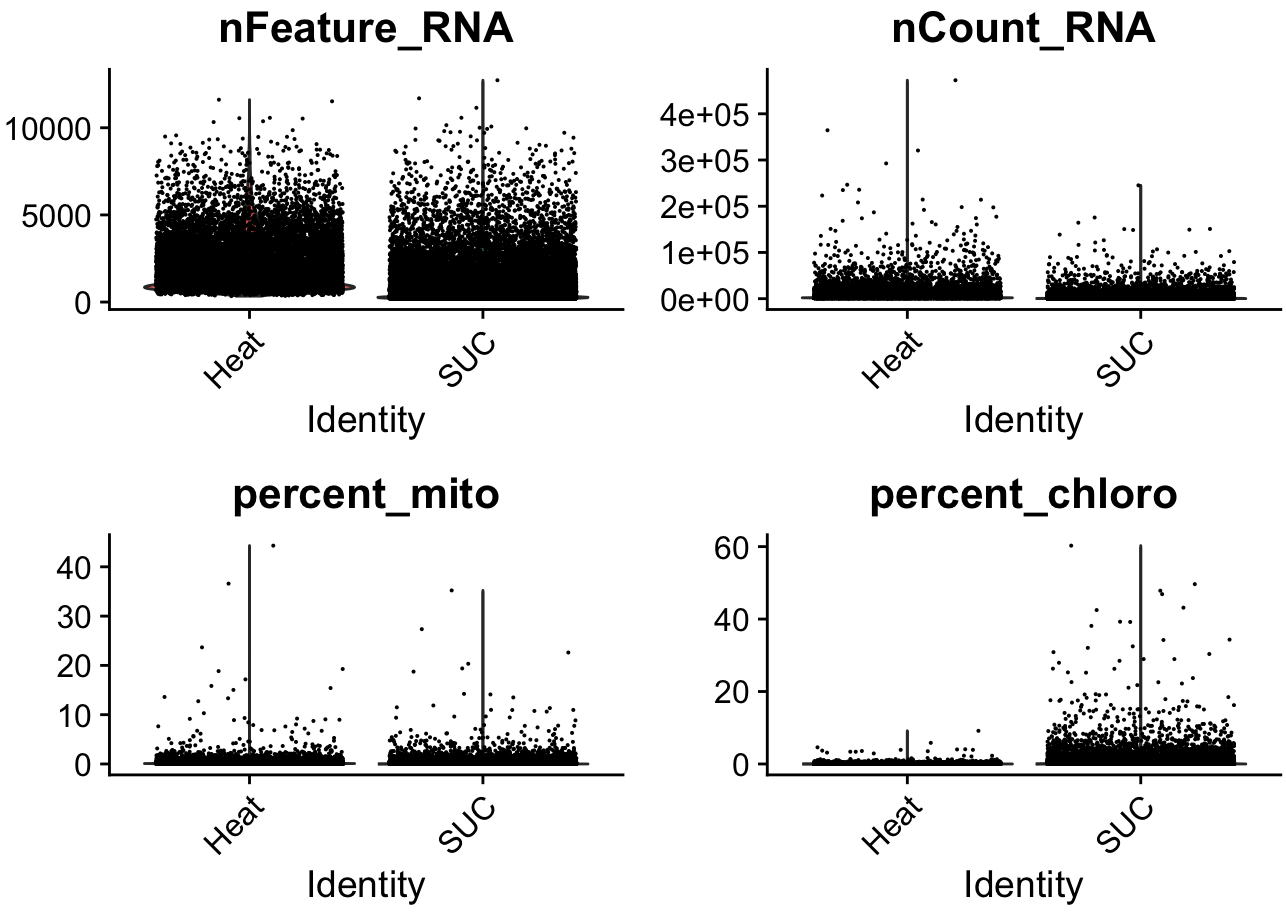
```
## remove protoplast-response genes, mitochondrial and chloroplast genes from integration features
alldata <- data.filt
proto_genes <- read.csv("./proto_genes.txt")
proto_genes %in% rownames(alldata)
```

```
## [1] FALSE
```

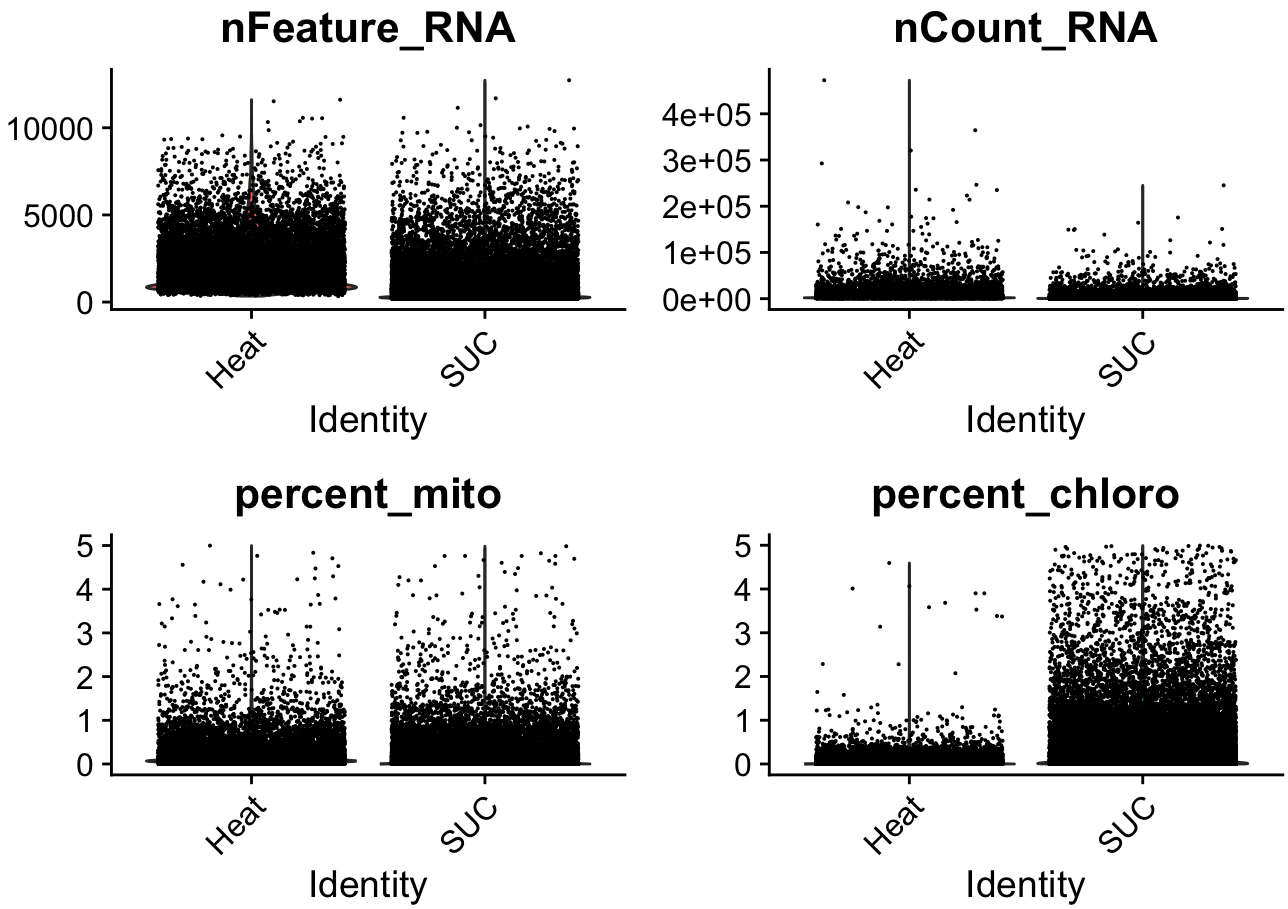
```
alldata_proto_rm=alldata[!rownames(alldata) %in% proto_genes,]
alldata <- alldata_proto_rm

## Find out the Mitchondria & chloroplast high expression genes
alldata <- PercentageFeatureSet(alldata, "ATMG", col.name = "percent_mito")
alldata <- PercentageFeatureSet(alldata, "ATCG", col.name = "percent_chloro")

feats <- c("nFeature_RNA", "nCount_RNA", "percent_mito", "percent_chloro")
FeatureScatter(alldata, "nCount_RNA", "nFeature_RNA", group.by = "sample", pt.size = 0.5)
```




```
VlnPlot(alldata, features = c("nFeature_RNA", "nCount_RNA", "percent_mito", "percent_chloro"), nc
ol = 2)
```



```
aldata_MTCTRM <- subset(alldata, subset = percent_mito < 5 & percent_chloro < 5)

VlnPlot(aldata_MTCTRM, features = c("nFeature_RNA", "nCount_RNA", "percent_mito", "percent_chlor
o"), ncol = 2)
```




```
saveRDS(aldata_MTCTRM , "./aldata_MTCTRM.rds")
```

```
## rm doublets from each sample separetly
rm(alldata_proto_rm)
rm(alldata)
rm(data.filt)
rm(sobj,alldata_MTCTRM)
```

```
## Warning in rm(sobj, alldata_MTCTRM): object 'sobj' not found
```

```
## Warning in rm(sobj, alldata_MTCTRM): object 'alldata_MTCTRM' not found
```

```
sub_object <- as.list(SplitObject(aldata_MTCTRM, split.by = "sample"))

suppressMessages(require(DoubletFinder))

all_data_doublets_filter <- list()
for(i in 1: length(sub_object)){
  data.filt1 = FindVariableFeatures(sub_object[[i]], verbose = F)
  data.filt1 = NormalizeData(data.filt1)
  data.filt1 = ScaleData(data.filt1, vars.to.regress = c("nFeature_RNA", "percent_mito"),
    verbose = F)
  data.filt1 = RunPCA(data.filt1, verbose = F, npcs = 20)
  data.filt1 = RunUMAP(data.filt1, dims = 1:10, verbose = F)
  nExp <- round(ncol(data.filt1) * 0.08)  # expect 8% doublets
  data.filt1 <- doubletFinder_v3(data.filt1, pN = 0.25, pK = 0.09, nExp = nExp, PCs = 1:10)
  all_data_doublets_filter[[i]] <- data.filt1
  }
```

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

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

```
## Also defined by 'BiocGenerics'
```

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

```
## Also defined by 'BiocGenerics'
```

```
## Loading required package: fields
```

```
## Loading required package: spam
```

```
## Spam version 2.10-0 (2023-10-23) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.
```

```
##
## Attaching package: 'spam'
```

```
## The following object is masked from 'package:stats4':
##
##     mle
```

```
## The following object is masked from 'package:Matrix':
##
##     det
```

```
## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve
```

```
## Loading required package: viridisLite
```

```
##
## Try help(fields) to get started.
```

```
## Loading required package: KernSmooth
```

```
## KernSmooth 2.23 loaded
## Copyright M. P. Wand 1997-2009
```

```
## [1] "Creating 1684 artificial doublets..."
## [1] "Creating Seurat object..."
## [1] "Normalizing Seurat object..."
```

```
## Normalizing layer: counts
```

```
## [1] "Finding variable genes..."
```

```
## Finding variable features for layer counts
```

```
## [1] "Scaling data..."
```

```
## Centering and scaling data matrix
```

```
## [1] "Running PCA..."
## [1] "Calculating PC distance matrix..."
## [1] "Computing pANN..."
## [1] "Classifying doublets.."
```

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

```
## Also defined by 'BiocGenerics'
```

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

```
## Also defined by 'BiocGenerics'
```

```
## [1] "Creating 2266 artificial doublets..."
## [1] "Creating Seurat object..."
## [1] "Normalizing Seurat object..."
```

```
## Normalizing layer: counts
```

```
## [1] "Finding variable genes..."
```

```
## Finding variable features for layer counts
```

```
## [1] "Scaling data..."
```

```
## Centering and scaling data matrix
```

```
## [1] "Running PCA..."
## [1] "Calculating PC distance matrix..."
## [1] "Computing pANN..."
## [1] "Classifying doublets.."
```

```
## Found more than one class "dist" in cache; using the first, from namespace 'spam'
## Also defined by 'BiocGenerics'
```

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

```
## Also defined by 'BiocGenerics'
```

```
## [1] "Creating 1279 artificial doublets..."
## [1] "Creating Seurat object..."
## [1] "Normalizing Seurat object..."
```

```
## Normalizing layer: counts
```

```
## [1] "Finding variable genes..."
```

```
## Finding variable features for layer counts
```

```
## [1] "Scaling data..."
```

```
## Centering and scaling data matrix
```

```
## [1] "Running PCA..."
## [1] "Calculating PC distance matrix..."
## [1] "Computing pANN..."
## [1] "Classifying doublets.."
```

```
## Found more than one class "dist" in cache; using the first, from namespace 'spam'
## Also defined by 'BiocGenerics'
```

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

```
## Also defined by 'BiocGenerics'
```

```
## [1] "Creating 1394 artificial doublets..."
## [1] "Creating Seurat object..."
## [1] "Normalizing Seurat object..."
```

```
## Normalizing layer: counts
```

```
## [1] "Finding variable genes..."
```

```
## Finding variable features for layer counts
```

```
## [1] "Scaling data..."
```

```
## Centering and scaling data matrix
```

```
## [1] "Running PCA..."
## [1] "Calculating PC distance matrix..."
## [1] "Computing pANN..."
## [1] "Classifying doublets.."
```

```
## Found more than one class "dist" in cache; using the first, from namespace 'spam'
## Also defined by 'BiocGenerics'
```

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

```
## Also defined by 'BiocGenerics'
```

```
## [1] "Creating 1225 artificial doublets..."
## [1] "Creating Seurat object..."
## [1] "Normalizing Seurat object..."
```

```
## Normalizing layer: counts
```

```
## [1] "Finding variable genes..."
```

```
## Finding variable features for layer counts
```

```
## [1] "Scaling data..."
```

```
## Centering and scaling data matrix
```

```
## [1] "Running PCA..."
## [1] "Calculating PC distance matrix..."
## [1] "Computing pANN..."
## [1] "Classifying doublets.."
```

```
## Found more than one class "dist" in cache; using the first, from namespace 'spam'
## Also defined by 'BiocGenerics'
```

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

```
## Also defined by 'BiocGenerics'
```

```
## [1] "Creating 3023 artificial doublets..."
## [1] "Creating Seurat object..."
## [1] "Normalizing Seurat object..."
```

```
## Normalizing layer: counts
```

```
## [1] "Finding variable genes..."
```

```
## Finding variable features for layer counts
```

```
## [1] "Scaling data..."
```

```
## Centering and scaling data matrix
```

```
## [1] "Running PCA..."
## [1] "Calculating PC distance matrix..."
## [1] "Computing pANN..."
## [1] "Classifying doublets.."
```

```
## Found more than one class "dist" in cache; using the first, from namespace 'spam'
## Also defined by 'BiocGenerics'
```

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

```
## Also defined by 'BiocGenerics'
```

```
## [1] "Creating 555 artificial doublets..."
## [1] "Creating Seurat object..."
## [1] "Normalizing Seurat object..."
```

```
## Normalizing layer: counts
```

```
## [1] "Finding variable genes..."
```

```
## Finding variable features for layer counts
```

```
## [1] "Scaling data..."
```

```
## Centering and scaling data matrix
```

```
## [1] "Running PCA..."
## [1] "Calculating PC distance matrix..."
## [1] "Computing pANN..."
## [1] "Classifying doublets.."
```

```
## Found more than one class "dist" in cache; using the first, from namespace 'spam'
## Also defined by 'BiocGenerics'
```

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

```
## Also defined by 'BiocGenerics'
```

```
## [1] "Creating 402 artificial doublets..."
## [1] "Creating Seurat object..."
## [1] "Normalizing Seurat object..."
```

```
## Normalizing layer: counts
```

```
## [1] "Finding variable genes..."
```

```
## Finding variable features for layer counts
```

```
## [1] "Scaling data..."
```

```
## Centering and scaling data matrix
```

```
## [1] "Running PCA..."
## [1] "Calculating PC distance matrix..."
## [1] "Computing pANN..."
## [1] "Classifying doublets.."
```

```
## Found more than one class "dist" in cache; using the first, from namespace 'spam'
## Also defined by 'BiocGenerics'
```

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

```
## Also defined by 'BiocGenerics'
```

```
## [1] "Creating 447 artificial doublets..."
## [1] "Creating Seurat object..."
## [1] "Normalizing Seurat object..."
```

```
## Normalizing layer: counts
```

```
## [1] "Finding variable genes..."
```

```
## Finding variable features for layer counts
```

```
## [1] "Scaling data..."
```

```
## Centering and scaling data matrix
```

```
## [1] "Running PCA..."
## [1] "Calculating PC distance matrix..."
## [1] "Computing pANN..."
## [1] "Classifying doublets.."
```

```
## Found more than one class "dist" in cache; using the first, from namespace 'spam'
## Also defined by 'BiocGenerics'
```

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

```
## Also defined by 'BiocGenerics'
```

```
## [1] "Creating 544 artificial doublets..."
## [1] "Creating Seurat object..."
## [1] "Normalizing Seurat object..."
```

```
## Normalizing layer: counts
```

```
## [1] "Finding variable genes..."
```

```
## Finding variable features for layer counts
```

```
## [1] "Scaling data..."
```

```
## Centering and scaling data matrix
```

```
## [1] "Running PCA..."
## [1] "Calculating PC distance matrix..."
## [1] "Computing pANN..."
## [1] "Classifying doublets.."
```

```
## Found more than one class "dist" in cache; using the first, from namespace 'spam'
## Also defined by 'BiocGenerics'
```

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

```
## Also defined by 'BiocGenerics'
```

```
## [1] "Creating 450 artificial doublets..."
## [1] "Creating Seurat object..."
## [1] "Normalizing Seurat object..."
```

```
## Normalizing layer: counts
```

```
## [1] "Finding variable genes..."
```

```
## Finding variable features for layer counts
```

```
## [1] "Scaling data..."
```

```
## Centering and scaling data matrix
```

```
## [1] "Running PCA..."
## [1] "Calculating PC distance matrix..."
## [1] "Computing pANN..."
## [1] "Classifying doublets.."
```

```
## Found more than one class "dist" in cache; using the first, from namespace 'spam'
## Also defined by 'BiocGenerics'
```

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

```
## Also defined by 'BiocGenerics'
```

```
## [1] "Creating 1257 artificial doublets..."
## [1] "Creating Seurat object..."
## [1] "Normalizing Seurat object..."
```

```
## Normalizing layer: counts
```

```
## [1] "Finding variable genes..."
```

```
## Finding variable features for layer counts
```

```
## [1] "Scaling data..."
```

```
## Centering and scaling data matrix
```

```
## [1] "Running PCA..."
## [1] "Calculating PC distance matrix..."
## [1] "Computing pANN..."
## [1] "Classifying doublets.."
```

```
## Found more than one class "dist" in cache; using the first, from namespace 'spam'
## Also defined by 'BiocGenerics'
```

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

```
## Also defined by 'BiocGenerics'
```

```
## [1] "Creating 996 artificial doublets..."
## [1] "Creating Seurat object..."
## [1] "Normalizing Seurat object..."
```

```
## Normalizing layer: counts
```

```
## [1] "Finding variable genes..."
```

```
## Finding variable features for layer counts
```

```
## [1] "Scaling data..."
```

```
## Centering and scaling data matrix
```

```
## [1] "Running PCA..."
## [1] "Calculating PC distance matrix..."
## [1] "Computing pANN..."
## [1] "Classifying doublets.."
```

```
#### name of the DF prediction can change, so extract the correct column name.
for(i in 1:13){
  colnames(all_data_doublets_filter[[i]]@meta.data)[10:11] <- c("pANN", "DF.classifications")
}

saveRDS(all_data_doublets_filter, "all_data_doublets_filter.rds")


all_data_doublets <- readRDS("all_data_doublets_filter.rds")

###remove the doublets
all_data_doublets_RM <- list()
for(i in 1: length(all_data_doublets)){
  data.filt <- all_data_doublets[[i]]
  DF.name = colnames(data.filt@meta.data)[grepl("DF.classification", colnames(data.filt@meta.dat
a))]
  data.filt = data.filt[, data.filt@meta.data[, DF.name] == "Singlet"]
  dim(data.filt)
  print(dim(data.filt))
  all_data_doublets_RM[i] <- data.filt
}
```

```
## [1] 33907  4648
```

```
## Warning in `[<-`(`*tmp*`, i, value = data.filt): implicit list embedding of S4
## objects is deprecated
```

```
## [1] 33907  6254
```

```
## Warning in `[<-`(`*tmp*`, i, value = data.filt): implicit list embedding of S4
## objects is deprecated
```

```
## [1] 33907  3531
```

```
## Warning in `[<-`(`*tmp*`, i, value = data.filt): implicit list embedding of S4
## objects is deprecated
```

```
## [1] 33907  3847
```

```
## Warning in `[<-`(`*tmp*`, i, value = data.filt): implicit list embedding of S4
## objects is deprecated
```

```
## [1] 33907  3380
```

```
## Warning in `[<-`(`*tmp*`, i, value = data.filt): implicit list embedding of S4
## objects is deprecated
```

```
## [1] 33907  8343
```

```
## Warning in `[<-`(`*tmp*`, i, value = data.filt): implicit list embedding of S4
## objects is deprecated
```

```
## [1] 33907  1531
```

```
## Warning in `[<-`(`*tmp*`, i, value = data.filt): implicit list embedding of S4
## objects is deprecated
```

```
## [1] 33907  1110
```

```
## Warning in `[<-`(`*tmp*`, i, value = data.filt): implicit list embedding of S4
## objects is deprecated
```

```
## [1] 33907  1234
```

```
## Warning in `[<-`(`*tmp*`, i, value = data.filt): implicit list embedding of S4
## objects is deprecated
```

```
## [1] 33907  1501
```

```
## Warning in `[<-`(`*tmp*`, i, value = data.filt): implicit list embedding of S4
## objects is deprecated
```

```
## [1] 33907  1242
```

```
## Warning in `[<-`(`*tmp*`, i, value = data.filt): implicit list embedding of S4
## objects is deprecated
```

```
## [1] 33907  3470
```

```
## Warning in `[<-`(`*tmp*`, i, value = data.filt): implicit list embedding of S4
## objects is deprecated
```

```
## [1] 33907  2749
```

```
## Warning in `[<-`(`*tmp*`, i, value = data.filt): implicit list embedding of S4
## objects is deprecated
```

```
saveRDS(all_data_doublets_RM, "all_data_doublets_RM.rds")
```

```
###SCT Transformation for each sample

rm(all_data_doublets)
rm(all_data_doublets_RM)
rm(sub_object)
rm(all_data_doublets_filter)


all_data_doublets_RM <- readRDS("all_data_doublets_RM.rds")

merged_seurat <- all_data_doublets_RM[[1]]

for(i in 2: length(all_data_doublets_RM)){
  colnames(all_data_doublets_RM[[1]]@meta.data)[10:11] <- c("pANN", "DF.classifications")
  merged_seurat <- merge(merged_seurat, all_data_doublets_RM[[i]])
}
rm(all_data_doublets_RM)

merged_seurat.1 <- merged_seurat %>%
    FindVariableFeatures(selection.method = "vst", nfeatures = 1000) %>%
    SCTransform(vst.flavor = "v2", variable.features.n = 1000, conserve.memory=TRUE)
```

```
## Running SCTransform on assay: RNA
```

```
## vst.flavor='v2' set. Using model with fixed slope and excluding poisson genes.
```

```
## Calculating cell attributes from input UMI matrix: log_umi
```

```
## Variance stabilizing transformation of count matrix of size 31263 by 42840
```

```
## Model formula is y ~ log_umi
```

```
## Get Negative Binomial regression parameters per gene
```

```
## Using 2000 genes, 5000 cells
```

```
## Found 114 outliers - those will be ignored in fitting/regularization step
```

```
## Skip calculation of full residual matrix
```

```
## Will not return corrected UMI because residual type is not set to 'pearson'
```

```
## Calculating gene attributes
```

```
## Wall clock passed: Time difference of 6.879779 secs
```

```
## Setting min_variance based on median UMI:  0.04
```

```
## Calculating variance for residuals of type pearson for 31263 genes
```

```
## Determine variable features
```

```
## Setting min_variance based on median UMI:  0.16
```

```
## Calculating residuals of type pearson for 1000 genes
```

```
##
  |
  |                                                                      |   0%
  |
  |==================                                                    |  25%
  |
  |==================================                                    |  50%
  |
  |====================================================                  |  75%
  |
  |======================================================================| 100%
```

```
## Computing corrected UMI count matrix
```

```
## Centering data matrix
```

```
## Place corrected count matrix in counts slot
```

```
## Set default assay to SCT
```

```
saveRDS(merged_seurat.1, "merged_seurat.1_SCTtransform2.rds")
```

```
###Integerarion using Harmony
merged_seurat.1@meta.data$orig.type[is.na(merged_seurat.1@meta.data$orig.type)] = "Heat"

merged_seurat.1 <- RunPCA(merged_seurat.1, assay = "SCT", npcs = 20)
```

```
## PC_ 1
## Positive:  AT1G54050, AT3G12580, AT5G12020, AT5G59720, AT1G07400, AT4G25200, AT1G16030, AT2G29
500, AT2G26150, AT1G59860
##     AT5G12030, AT3G46230, AT1G62480, AT1G74310, AT1G12080, AT1G53540, AT5G56030, AT3G59370, AT
1G75750, AT4G26320
##     AT2G47180, AT3G10020, AT2G02130, AT5G56540, AT5G51440, AT1G71000, AT5G52640, AT1G55330, AT
2G19310, AT4G11210
## Negative:  AT5G60530, AT1G52070, AT1G47600, AT1G51470, AT3G16440, AT5G54370, AT1G06090, AT1G52
060, AT1G52050, AT3G48340
##     AT3G06460, AT3G49190, AT3G19430, AT3G03500, AT1G26820, AT1G17180, AT2G43610, AT5G04200, AT
1G15385, AT5G55110
##     AT2G23410, AT2G37540, AT5G16230, AT3G22740, AT5G35735, AT1G50060, AT4G04460, AT4G27400, AT
1G06080, AT4G22640
## PC_ 2
## Positive:  AT1G54050, AT3G12580, AT5G12020, AT5G59720, AT1G07400, AT1G16030, AT4G25200, AT2G29
500, AT3G46230, AT2G26150
##     AT1G59860, AT5G12030, AT1G74310, AT1G53540, AT5G56030, AT2G47180, AT3G10020, AT5G51440, AT
3G09440, AT1G71000
##     AT5G52640, AT2G19310, AT2G32120, AT3G09350, AT2G36460, AT4G10250, AT5G48570, AT5G10695, AT
5G25450, AT3G24500
## Negative:  AT3G22620, AT2G36100, AT2G28670, AT5G66390, AT3G22600, AT5G42180, AT2G32300, AT3G11
550, AT3G24020, AT3G55230
##     AT2G27370, AT5G15290, AT4G13580, AT3G56240, AT1G30750, AT1G75750, AT2G39430, AT4G34050, AT
1G05260, AT4G26320
##     AT5G06200, AT2G30210, AT1G71740, AT3G59370, AT2G40113, AT3G19450, AT5G40450, AT5G46890, AT
5G46900, AT4G11290
## PC_ 3
## Positive:  AT2G36100, AT2G28670, AT5G66390, AT5G42180, AT3G22620, AT3G11550, AT3G24020, AT3G55
230, AT2G32300, AT2G27370
##     AT4G13580, AT5G15290, AT1G30750, AT2G39430, AT3G22600, AT1G71740, AT2G30210, AT5G06200, AT
2G40113, AT4G02090
##     AT3G56240, AT5G65530, AT1G44970, AT4G17215, AT1G61590, AT3G53260, AT1G75750, AT4G26140, AT
1G05260, AT2G38400
## Negative:  NM-008302.3, NM-010480.5, NM-018853.3, NM-009093.2, NM-008972.2, NP-904328.1, NM-01
1295.6, NM-009098.2, NM-025974.2, NM-027015.4
##     NM-010106.2, NM-013765.2, NM-016738.5, NM-025814.2, NM-170669.2, NM-011029.4, NM-018860.4,
NM-026055.2, NM-207523.2, NM-001033865.1
##     NM-026030.2, NM-172086.2, NM-007393.5, NM-025274.3, NM-009084.4, NM-026147.6, NM-009092.3,
NM-026069.3, NM-020600.4, NM-024266.3
## PC_ 4
## Positive:  AT3G59370, AT1G12080, AT2G02130, AT1G62480, AT4G12550, AT3G62680, AT1G10682, AT3G54
580, AT5G40730, AT3G28550
##     AT5G17820, AT4G12545, AT4G40090, AT4G11210, AT3G54590, AT5G46900, AT4G22666, AT5G14330, AT
5G56540, AT3G09260
##     AT5G46890, AT1G65310, AT1G30870, AT1G23720, AT5G05500, AT4G25820, AT3G09925, AT3G01190, AT
5G57625, AT1G01750
## Negative:  AT2G36100, AT2G28670, AT5G66390, AT5G42180, AT3G22620, AT3G11550, AT3G24020, AT3G55
230, AT2G27370, AT4G13580
##     AT2G32300, AT1G30750, AT5G15290, AT2G39430, AT3G22600, AT1G71740, AT2G30210, AT5G06200, AT
2G40113, NM-008302.3
##     NM-010480.5, NM-018853.3, NM-009093.2, NM-008972.2, NP-904328.1, NM-011295.6, NM-009098.2,
NM-025974.2, NM-027015.4, NM-013765.2
## PC_ 5
## Positive:  AT3G59370, AT1G12080, AT1G62480, AT2G02130, AT4G26320, AT1G10682, AT4G11210, AT5G56
540, AT2G31083, AT2G13820
##     AT1G75750, AT5G59090, AT1G77690, AT1G47600, AT1G51470, AT1G55330, AT3G21770, AT2G45470, AT
```

```
4G14130, AT1G72230
##     AT3G16440, AT1G26450, AT1G06090, AT2G31085, AT4G18510, AT4G23690, AT2G46890, AT4G34050, AT
4G11190, AT1G26820
## Negative:  AT3G62680, AT4G40090, AT3G54580, AT3G54590, AT4G25820, AT1G30870, AT5G57625, AT5G05
500, AT3G28550, AT5G14330
##     AT5G17820, AT3G09925, AT4G26010, AT4G00680, AT4G12550, AT4G02270, AT5G67400, AT1G01750, AT
1G12560, AT1G62980
##     AT1G23720, AT5G35190, AT4G12545, AT4G22666, AT3G49960, AT1G52070, AT5G04960, AT4G01480, AT
```

```
harmonized_seurat <- RunHarmony(merged_seurat.1,
                                group.by.vars = c("orig.type", "sample", "sample_index"),
                                lambda = c(1,1,1),
                                reduction = "pca", assay.use = "SCT", reduction.save = "harmony")
```

```
## Transposing data matrix
```

```
## Initializing state using k-means centroids initialization
```

```
## Harmony 1/10
```

```
## Harmony 2/10
```

```
## Harmony 3/10
```

```
## Harmony 4/10
```

```
## Harmony 5/10
```

```
## Harmony 6/10
```

```
## Harmony 7/10
```

```
## Harmony 8/10
```

```
## Harmony converged after 8 iterations
```

```
harmonized_seurat <- RunUMAP(harmonized_seurat, reduction = "harmony", assay = "SCT", dims = 1:1
0)
```

```
## 21:45:33 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'
```

```
## 21:45:33 Read 42840 rows and found 10 numeric columns
```

```
## 21:45:33 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'
```

```
## 21:45:33 Building Annoy index with metric = cosine, n_trees = 50
```

```
## 0%   10   20   30   40   50   60   70   80   90   100%
```

```
## [----|----|----|----|----|----|----|----|----|----|
```

```
## **************************************************|
## 21:45:36 Writing NN index file to temp file /var/folders/gt/w451x6dd2xs29hm6r8jx9bph0000gn/T//
RtmpUtc8me/file107b50033bc0
## 21:45:37 Searching Annoy index using 1 thread, search_k = 3000
## 21:45:46 Annoy recall = 100%
## 21:45:47 Commencing smooth kNN distance calibration using 1 thread with target n_neighbors = 3
0
## 21:45:48 Initializing from normalized Laplacian + noise (using RSpectra)
## 21:45:50 Commencing optimization for 200 epochs, with 1711102 positive edges
## 21:46:02 Optimization finished
```

```
harmonized_seurat <- FindNeighbors(object = harmonized_seurat, reduction = "harmony")
```

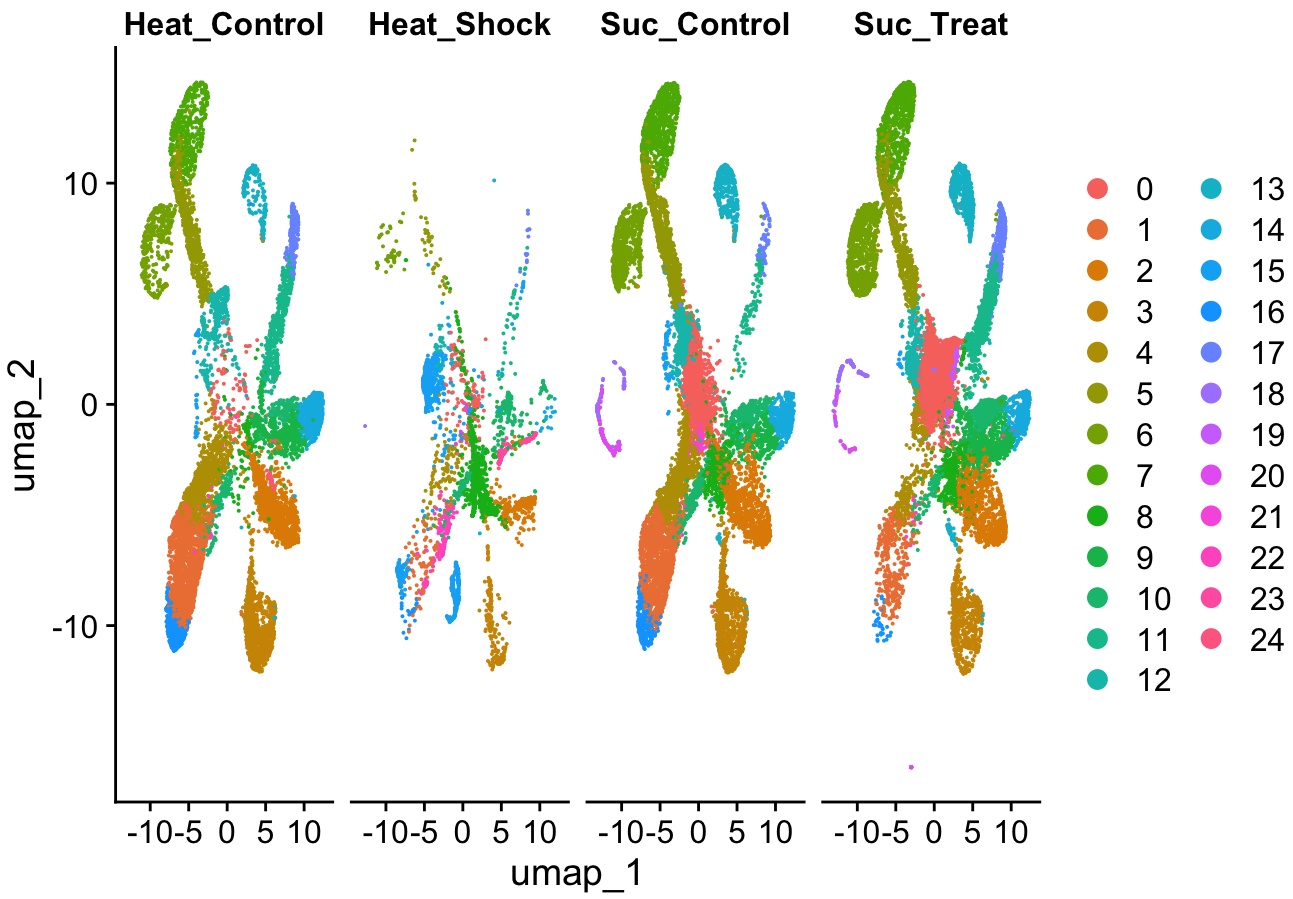
```
## Computing nearest neighbor graph
## Computing SNN
```

```
harmonized_seurat <- FindClusters(harmonized_seurat, resolution = c(0.6))
```

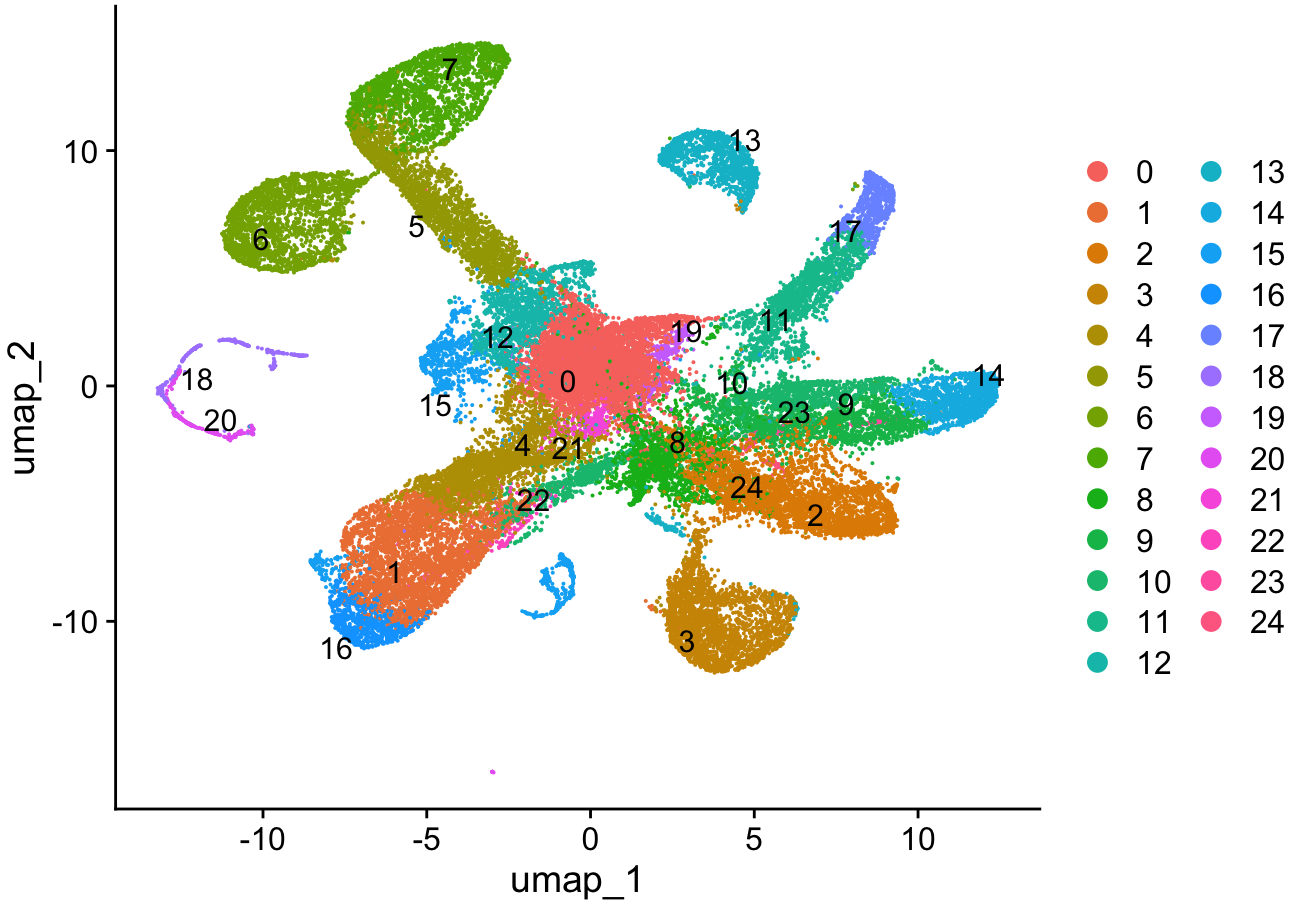
```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 42840
## Number of edges: 1305065
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.9324
## Number of communities: 25
## Elapsed time: 7 seconds
```

```
saveRDS(harmonized_seurat, "harmonized_seurat2.rds")
```

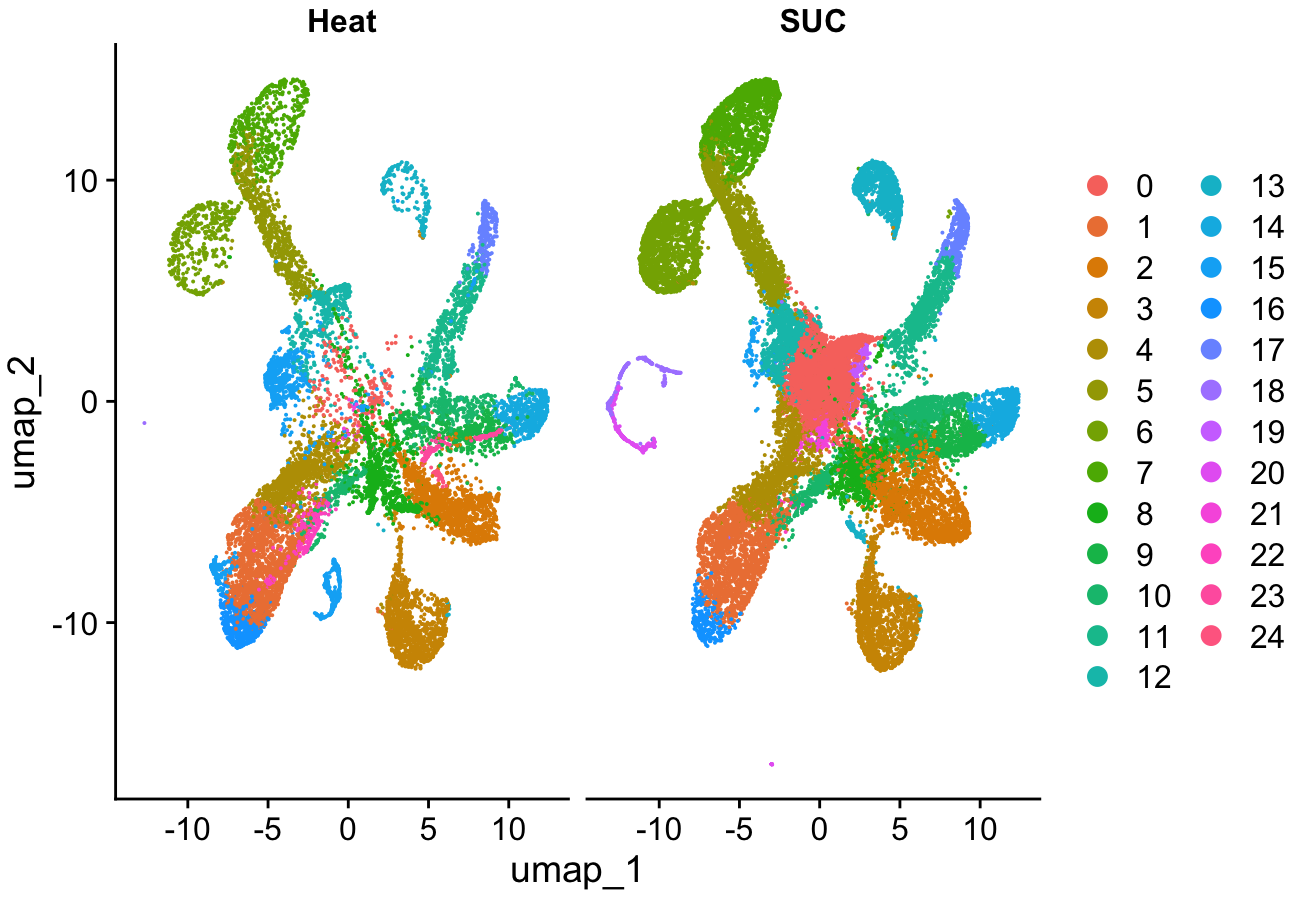
```
rm(merged_seurat)
rm(merged_seurat.1)
p1 <- DimPlot(harmonized_seurat, reduction = "umap", split.by = "sample_index")
p2 <- DimPlot(harmonized_seurat, reduction = "umap", label = TRUE, repel = TRUE)
p3 <- DimPlot(harmonized_seurat, reduction = "umap", split.by = "orig.type")

p1
```




```
p2
```


umap_2
umap_1
10
0
-10
-5
0
5
10
7
13
6
5
17
12
19
11
18
0
10
14
15
23
9
20
4
21
8
24
22
2
1
16
3
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

p3


Heat
SUC
umap_2
umap_1
10
0
-10
-5
0
5
10
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

```
sobj <-  readRDS("harmonized_seurat2.rds")
rm(aldata_MTCTRM)
rm(alldatea)
```

```
## Warning in rm(alldatea): object 'alldatea' not found
```

```
rm(all_seurats)
```

```
## Warning in rm(all_seurats): object 'all_seurats' not found
```

```
rm(harmonized_seurat)

### Run label transfer

options(Seurat.memsafe=TRUE)
mB=max(30000,ceiling(as.numeric(object.size(sobj))/100000000)*100)
print(paste("----- increasing max mem to",mB))
```

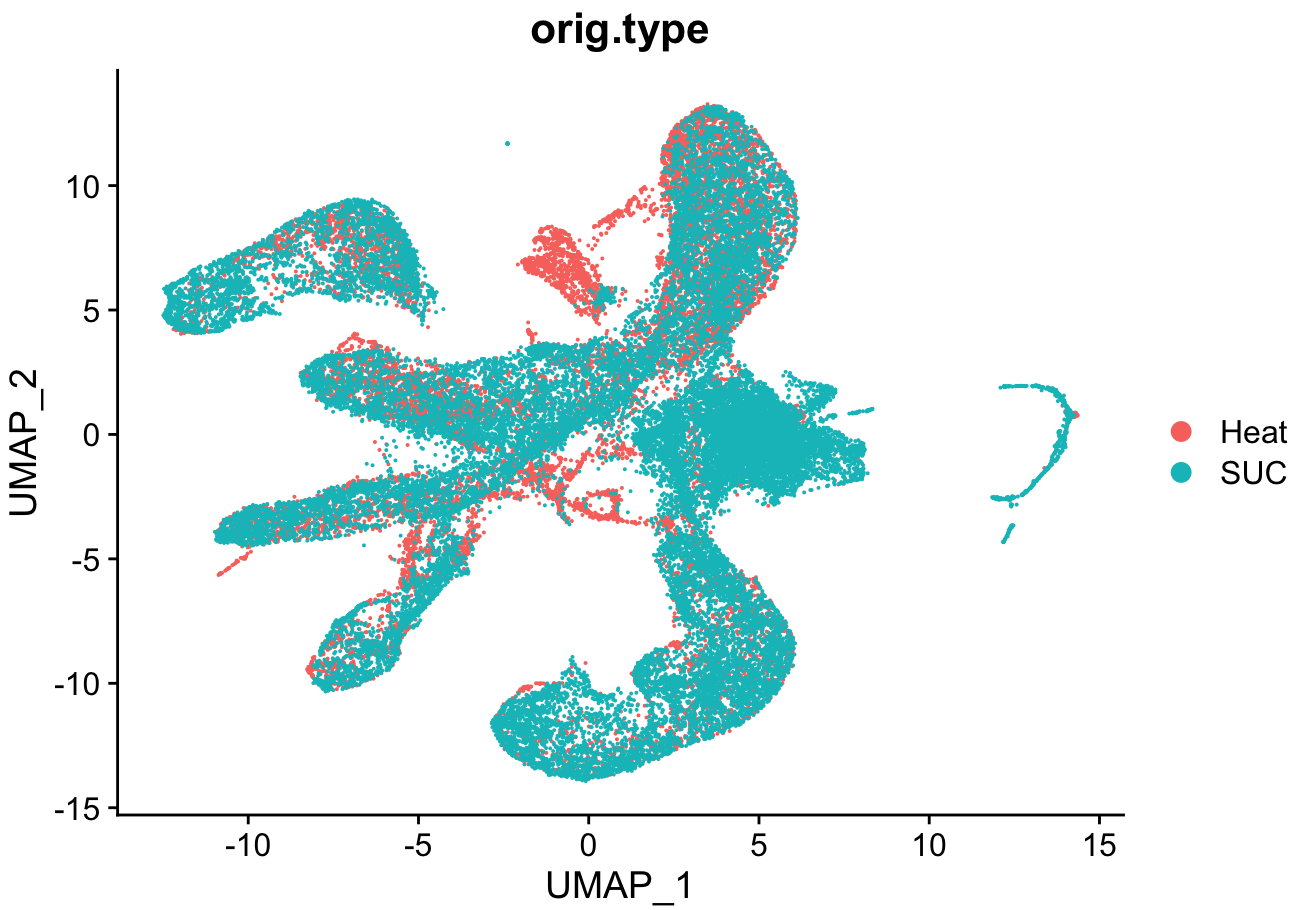
```
## [1] "----- increasing max mem to 30000"
```

```
maxSize=mB*1024^2
```

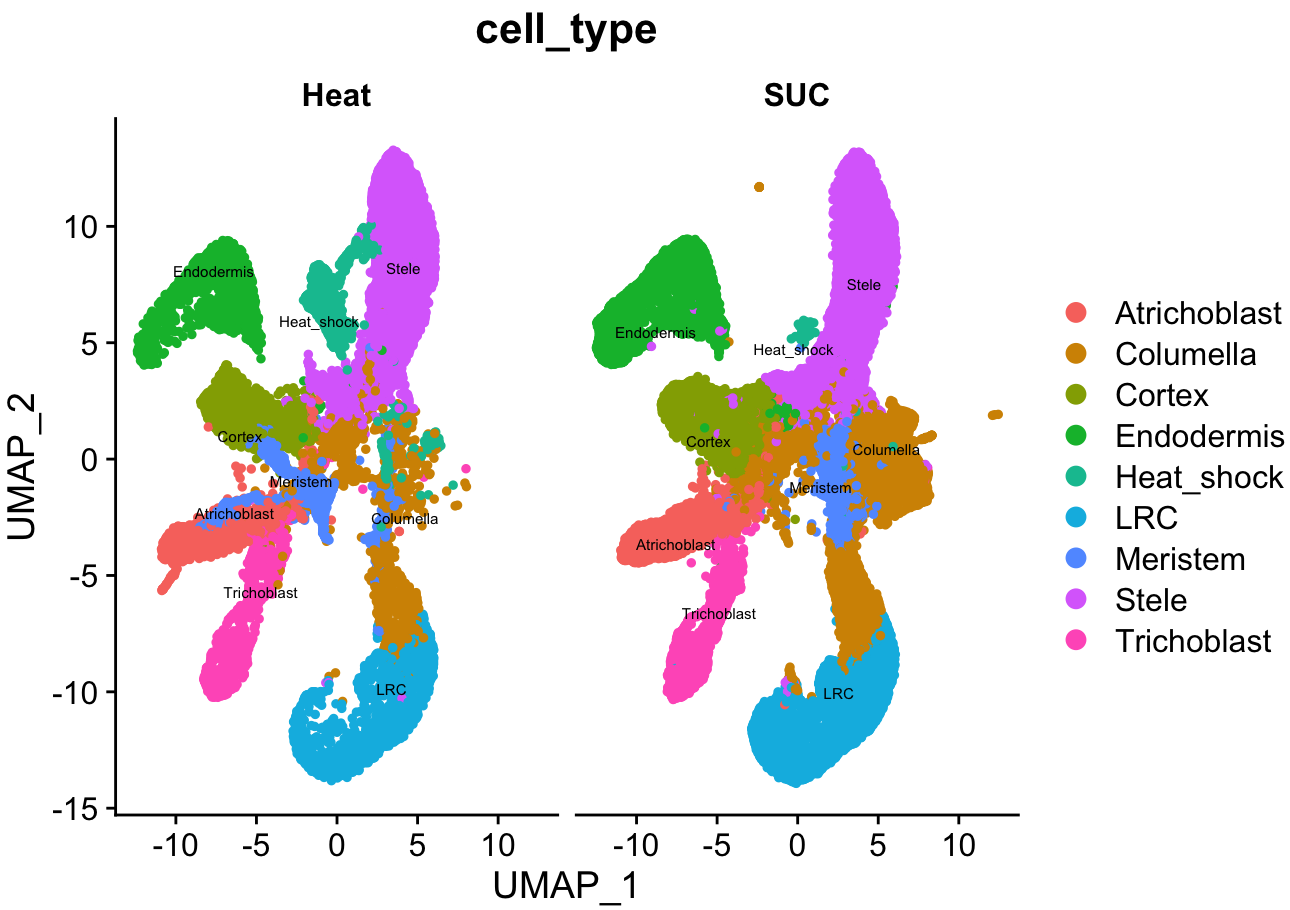
```
##### functions
chop=function(myStr,mySep,myField){
  choppedString=sapply(strsplit(myStr,mySep),"[",myField)
  if(length(myField)>1){
    choppedString=apply(choppedString,2,function(x){paste0(x[!is.na(x)],collapse=mySep)})
  }
  return(choppedString)
}

chop=function(myStr,mySep,myField){
  choppedString=sapply(strsplit(myStr,mySep),"[",myField)
  if(length(myField)>1){
    choppedString=apply(choppedString,2,function(x){paste0(x[!is.na(x)],collapse=mySep)})
  }
  return(choppedString)
}

parse_predictions <- function(predictions){
  predictions %>%
    tibble::rownames_to_column("cell")%>%
    mutate(numIDs=round(1/prediction.score.max))%>%
    rowwise()%>%
    mutate(prediction.score.second=sort(c_across(3:(ncol(.)-2)),decreasing=T)[numIDs],
           second.id=gsub("\\.","-",gsub(",NA","",gsub("prediction.score.","",paste0(colnames(pre
dictions)[-1][c_across(3:(ncol(.)-2))>=prediction.score.second][1:numIDs],collapse=","))))
    )%>%
    ungroup()%>%
    dplyr::select(predicted.id,prediction.score.max,prediction.score.second,second.id,numIDs,cel
l)%>%
    tibble::column_to_rownames("cell")
}
gc()
```

```
##             used   (Mb) gc trigger   (Mb) limit (Mb)   max used   (Mb)
## Ncells   9544537  509.8   15897987  849.1         NA   15897987  849.1
## Vcells 604736036 4613.8 1167598896 8908.1     102400 1167598896 8908.1
```

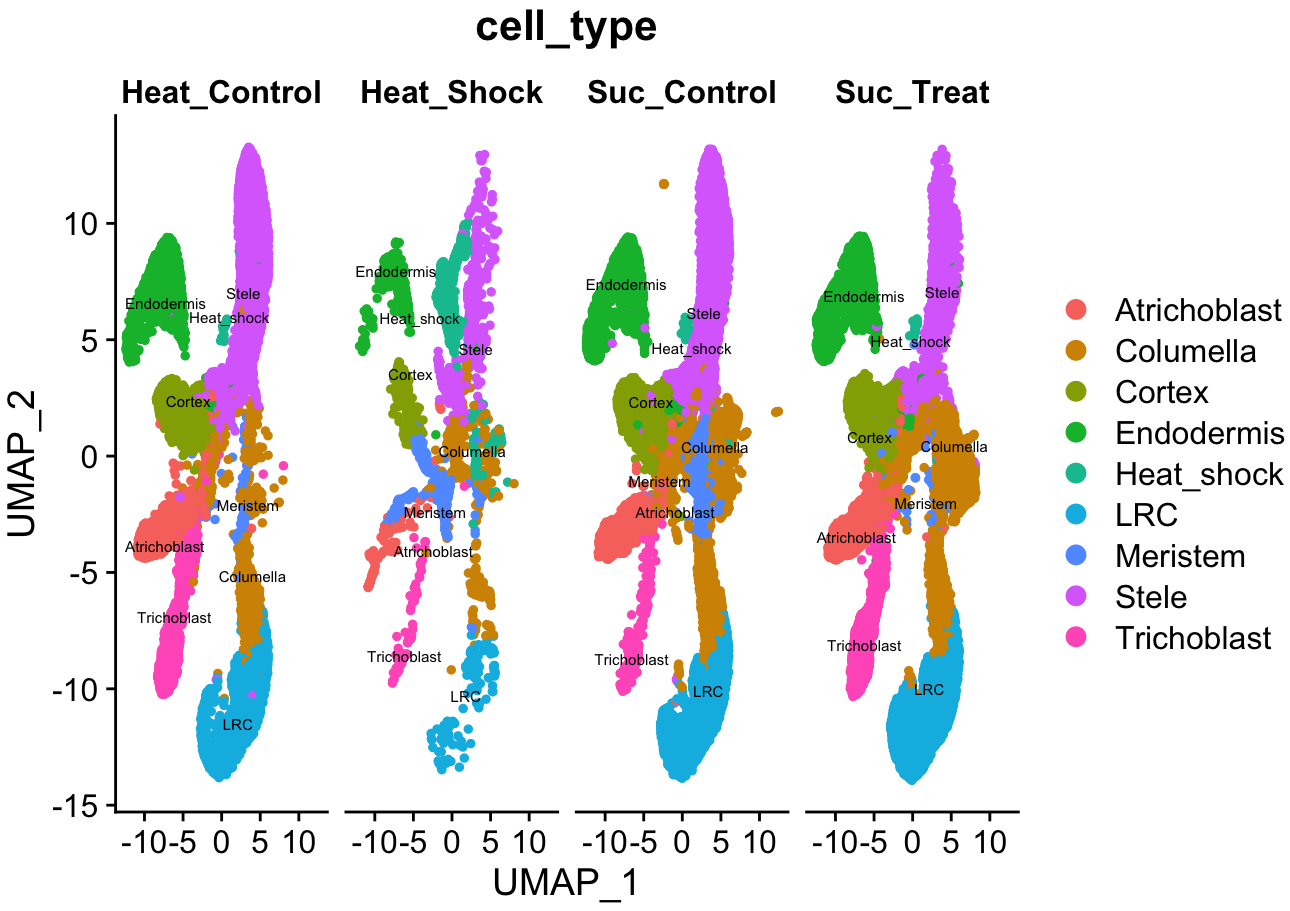
```
sobj.lt <- readRDS("./sobj.lt.rds")
p1 <- DimPlot(sobj.lt, reduction = "umap", group.by = "orig.type")
p2 <- DimPlot(sobj.lt, reduction = "umap", group.by = "predicted.anno", label = TRUE, repel = TRU
E) +
    NoLegend()
p1
```




```
###Label each cluster based on the top cluster labels

library(dplyr)
library(tidyverse)


cell_type <- read.csv("Top_label.csv")

sobj.lt$cell_type <- cell_type$cell_type[match(sobj.lt@meta.data$seurat_clusters, cell_type$seura
t_clusters)]
###Remove the cluster with the cell number less than 10
sobj.lt.filter <- subset(sobj.lt, !(subset=seurat_clusters %in% c('19', '20', '21')))

p1 <- DimPlot(sobj.lt.filter, reduction = "umap", group.by = "cell_type", split.by = "orig.type",
label = TRUE, repel = TRUE, pt.size = 1, label.size = 2)
p2 <- DimPlot(sobj.lt.filter, reduction = "umap", group.by = "cell_type", split.by = "sample_inde
x", label = TRUE, repel = TRUE, pt.size = 1, label.size = 2)


p1
```


cell_type
Heat
SUC
Endodermis
Stele
Heat_shock
Cortex
Meristem
Atrichoblast
Columella
Trichoblast
LRC
UMAP_2
UMAP_1
Atrichoblast
Columella
Cortex
Endodermis
Heat_shock
LRC
Meristem
Stele
Trichoblast

p2


cell_type
Heat_Control
Heat_Shock
Suc_Control
Suc_Treat
Endodermis
Stele
Heat_shock
Cortex
Meristem
Atrichoblast
Columella
Trichoblast
LRC
UMAP_2
UMAP_1
Atrichoblast
Columella
Cortex
Endodermis
Heat_shock
LRC
Meristem
Stele
Trichoblast

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

```
##
## Attaching package: 'data.table'
```

```
## The following object is masked from 'package:SummarizedExperiment':
##
##     shift
```

```
## The following object is masked from 'package:GenomicRanges':
##
##     shift
```

```
## The following object is masked from 'package:IRanges':
##
##     shift
```

```
## The following objects are masked from 'package:S4Vectors':
##
##     first, second
```

```
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
```

```
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
library(magrittr)
```

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:GenomicRanges':
##
##     subtract
```

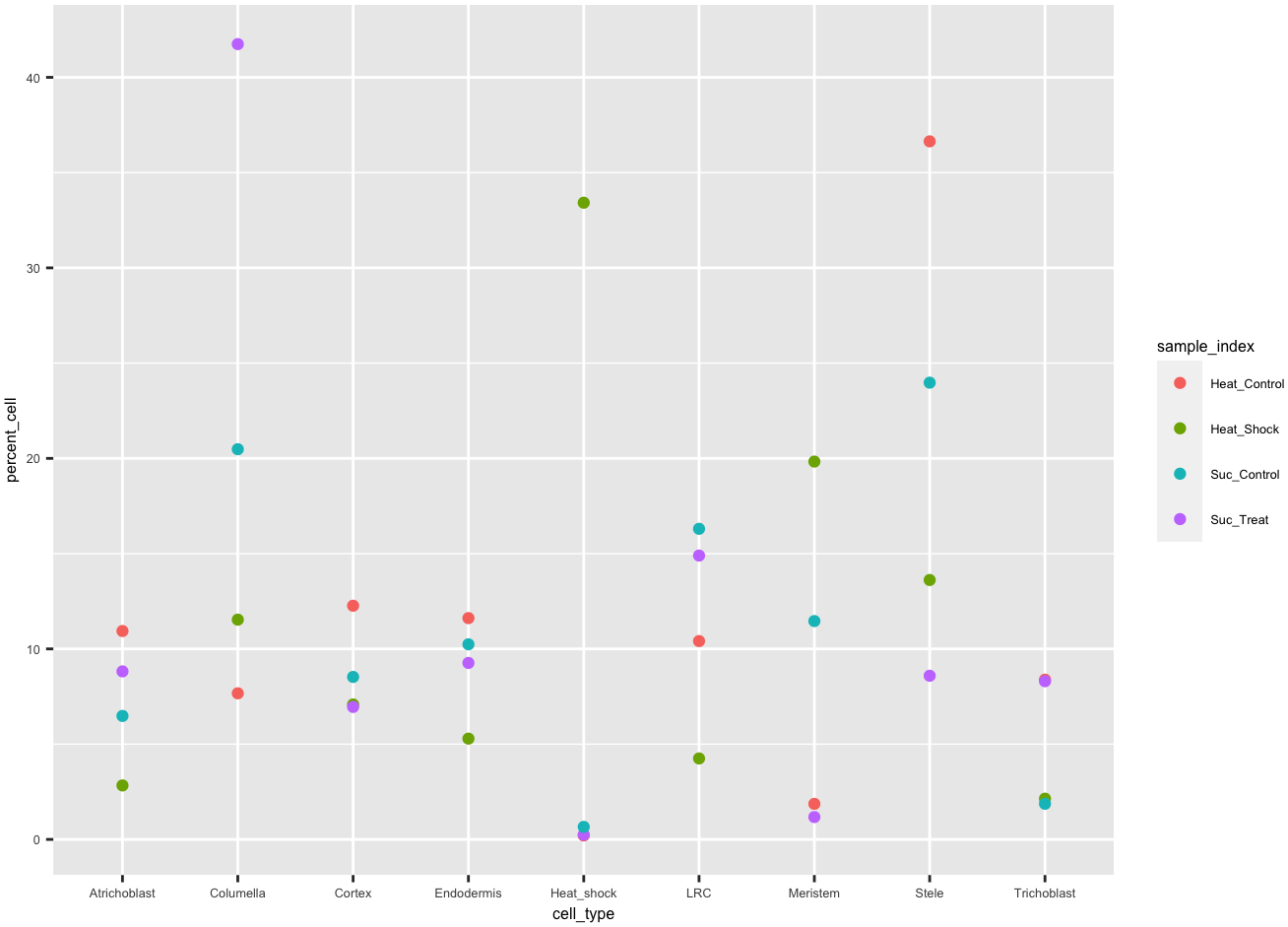
```
## The following object is masked from 'package:purrr':
##
##     set_names
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

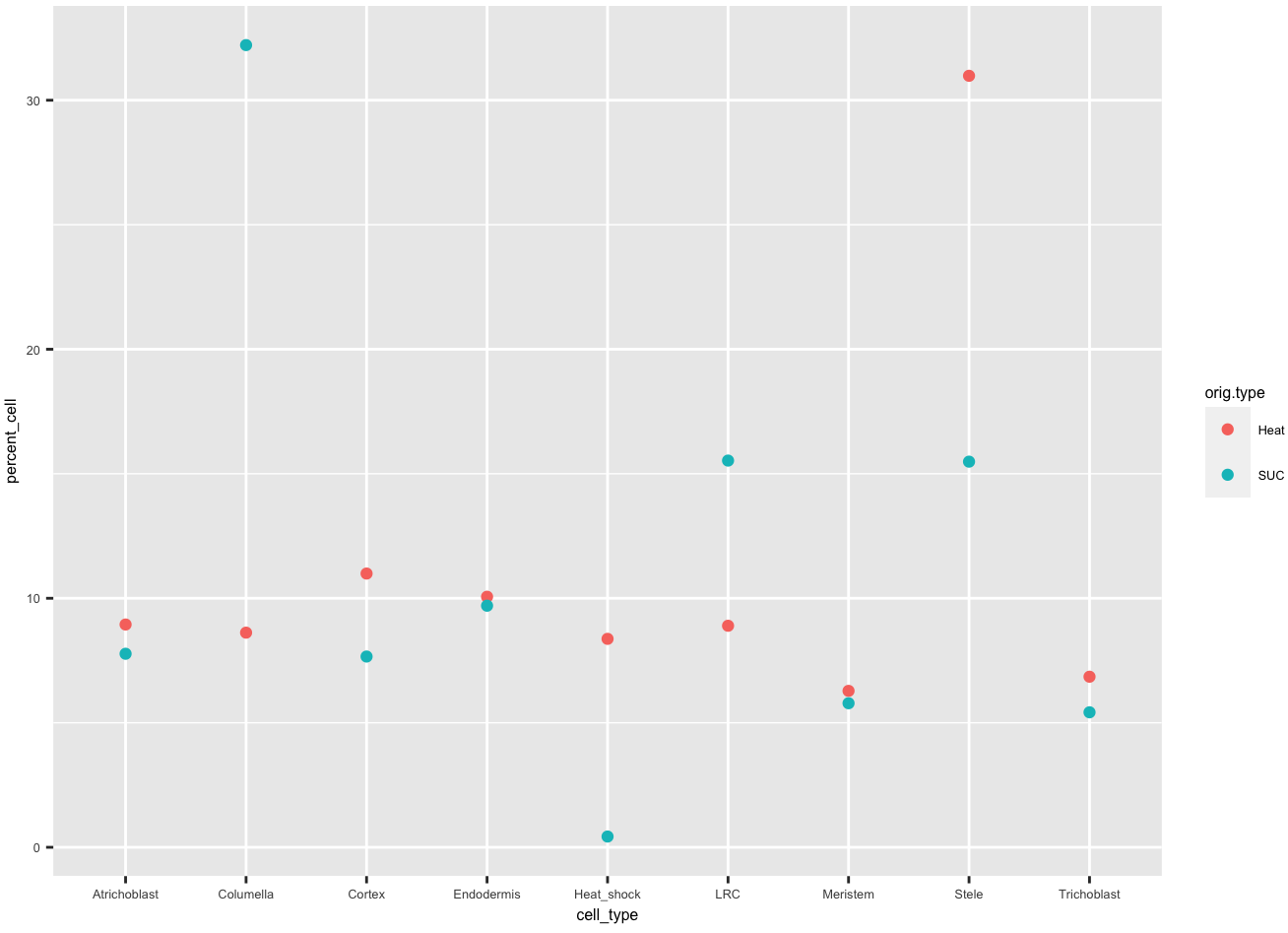
```
library(dplyr)
md <- sobj.lt.filter@meta.data %>% as.data.table
count_cell <- md[, .N, by = c("sample_index", "cell_type")]

as.numeric(count_cell$N)
```

```
##  [1] 3889 1105  814  890 1302 1233  198 1161   22 1156  245  686  183  471  147
## [16]  399   98   74 6399 2284 1273 1066 1420 1317  180 1352   38 2985 1275 1427
## [31]   82 2550 2030  807 1062  233
```

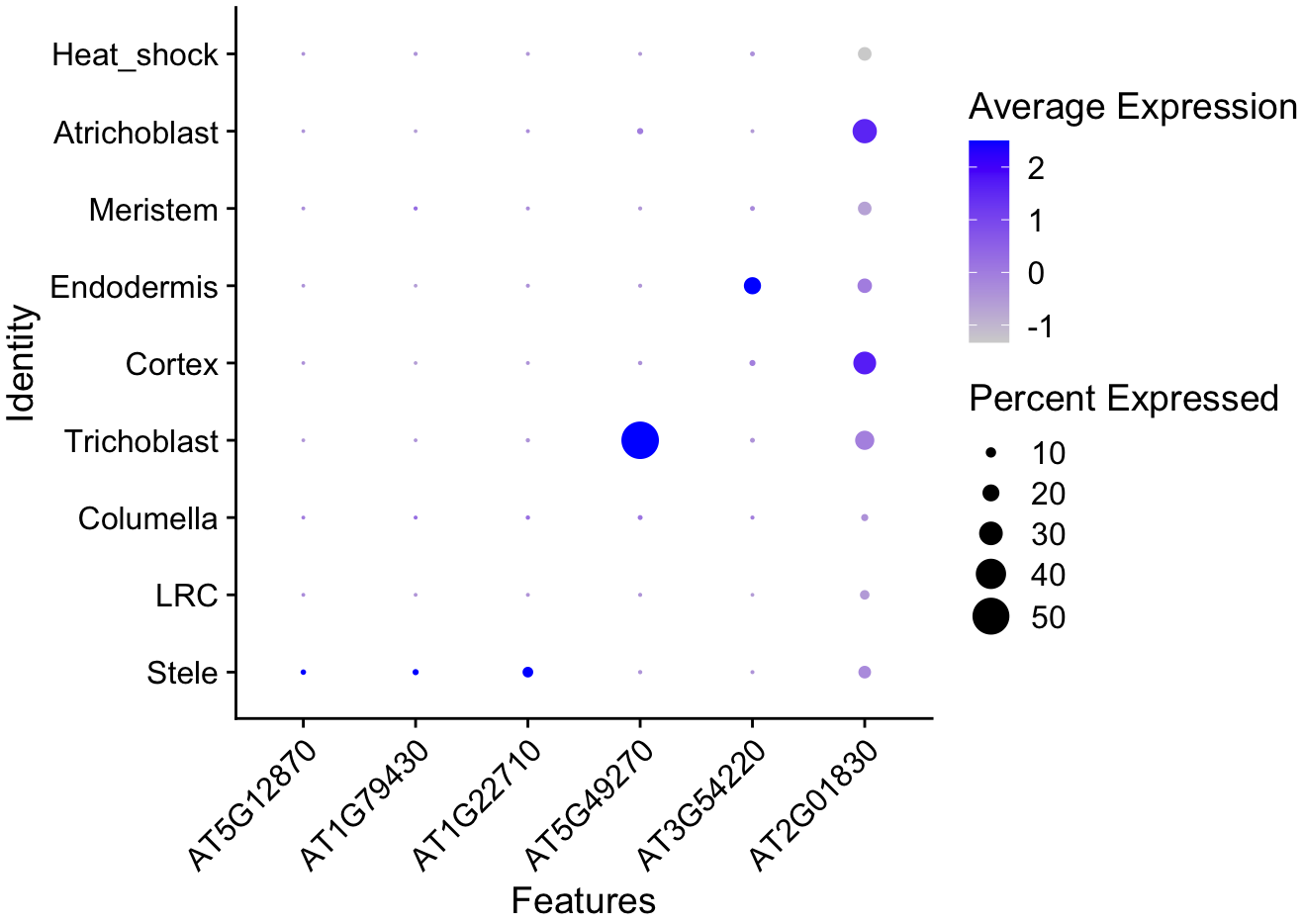
```
count_cell <- count_cell %>%
  group_by(sample_index) %>%
  mutate(total_cell = sum(N)) %>%
  ungroup() %>%
  mutate(percent_cell = count_cell$N/total_cell*100)

p1 <- ggplot(count_cell, aes(cell_type, percent_cell, col=
sample_index))+
  geom_point() +
  theme(text=element_text(size=6))

count_cell <- md[, .N, by = c("orig.type", "cell_type")]

count_cell <- count_cell %>%
  group_by(orig.type) %>%
  mutate(total_cell2 = sum(N)) %>%
  ungroup() %>%
  mutate(percent_cell = count_cell$N/total_cell2*100)
p2 <- ggplot(count_cell, aes(cell_type, percent_cell, col=
orig.type))+
  geom_point() +
  theme(text=element_text(size=6))

p1
```


40
30
20
10
0
percent_cell
Atrichoblast
Columella
Cortex
Endodermis
Heat_shock
LRC
Meristem
Stele
Trichoblast
cell_type
sample_index
Heat_Control
Heat_Shock
Suc_Control
Suc_Treat

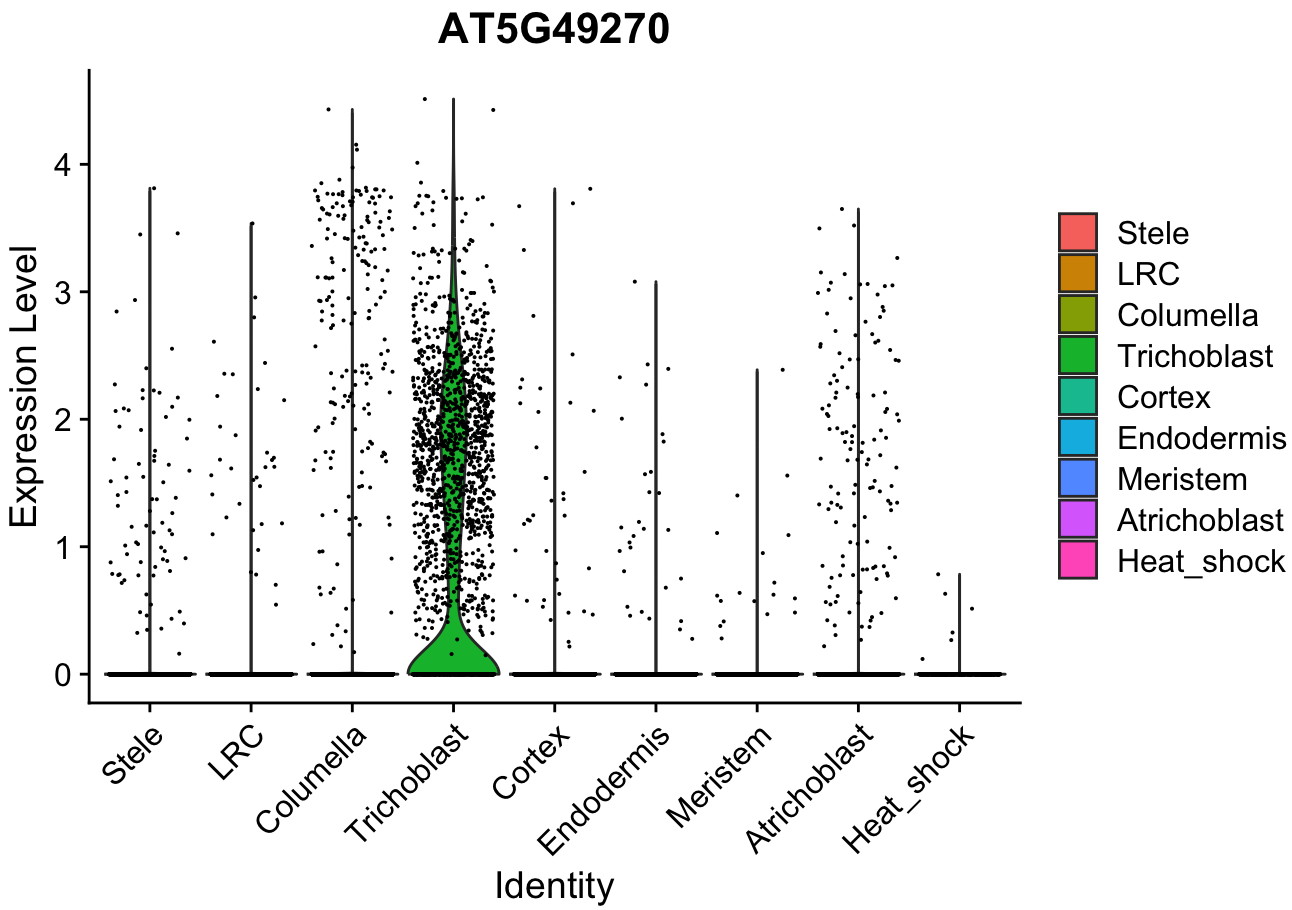
p2


30
20
10
0
percent_cell
Atrichoblast
Columella
Cortex
Endodermis
Heat_shock
LRC
Meristem
Stele
Trichoblast
cell_type
orig.type
Heat
SUC

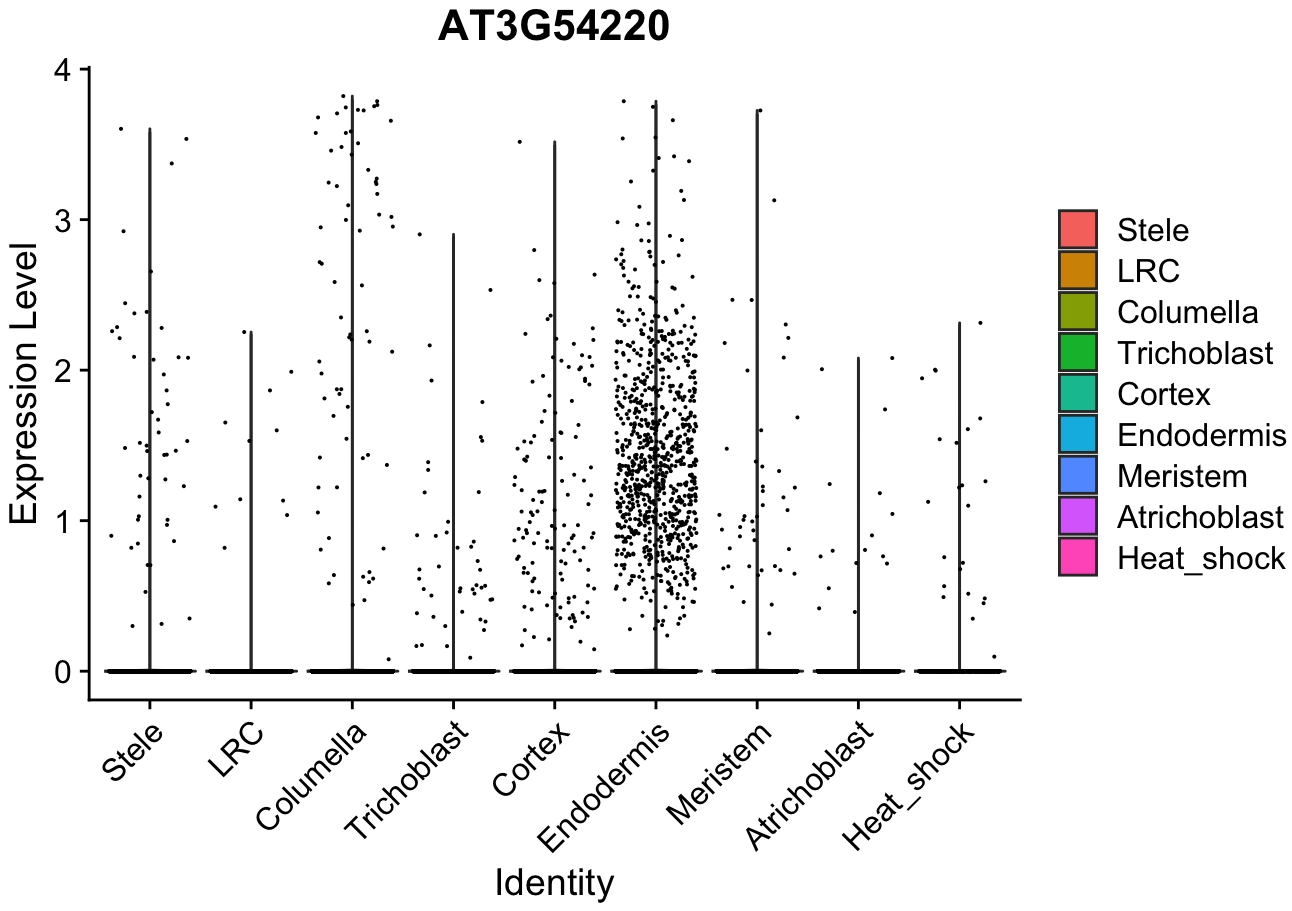
```
#### To identify the known marker from previous literatures
####stele_specific markers

features = c("AT5G12870","AT1G79430","AT1G22710", "AT5G49270", "AT3G54220", "AT2G01830")
Idents(sobj.lt.filter) = "cell_type"
DotPlot(sobj.lt.filter, features = features) + RotatedAxis()
```



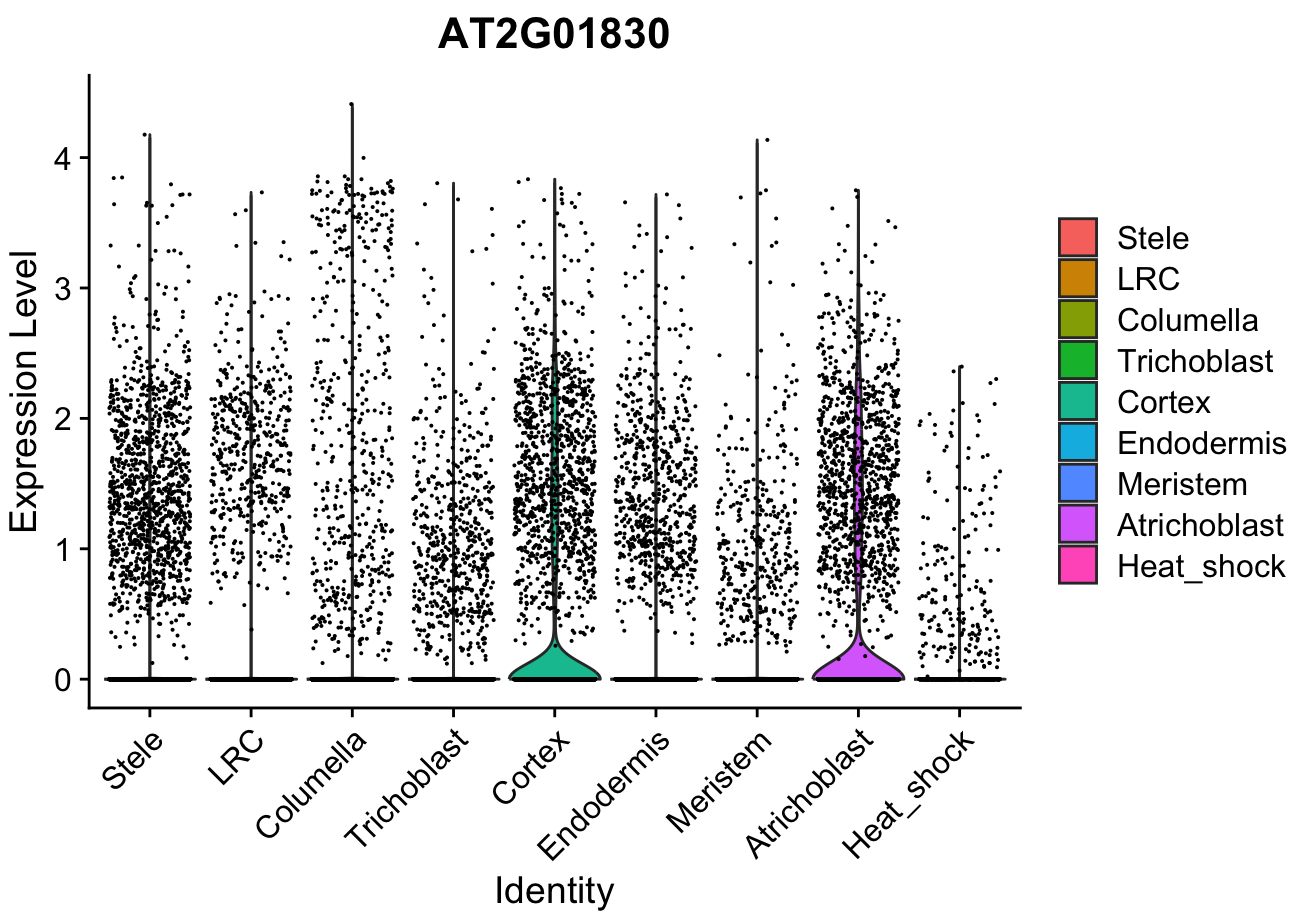

```
###ROOT HAIR specific
VlnPlot(sobj.lt.filter, features ="AT5G49270")
```




```
###SCR, endodermis
VlnPlot(sobj.lt.filter, features ="AT3G54220")
```



```
###The nonspecific expression of WOODENLEG (WOL)
VlnPlot(sobj.lt.filter, features =c("AT2G01830"))
```



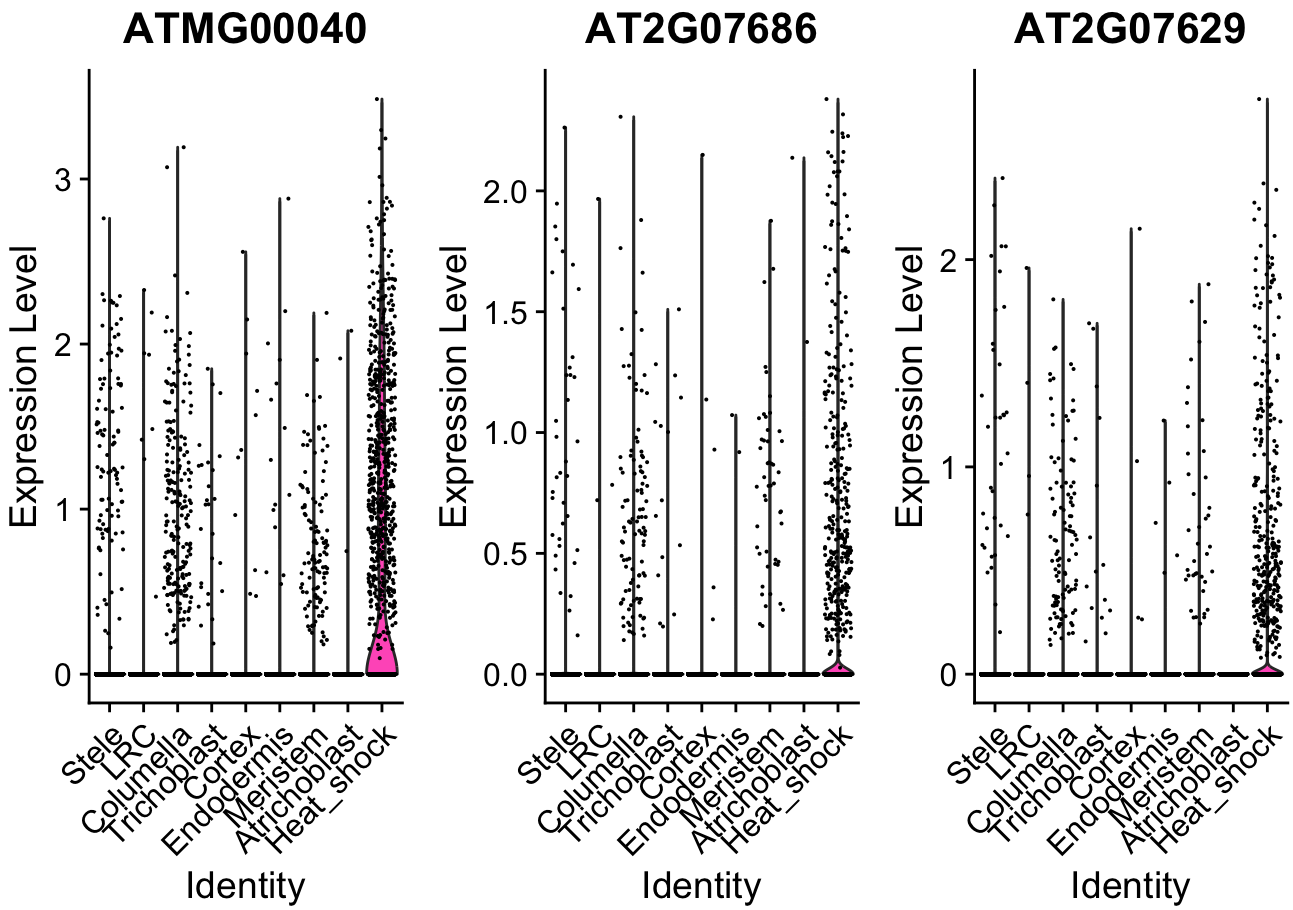

```
##find marker for each cluster
sobj.lt.filter
```

```
## An object of class Seurat
## 65168 features across 41853 samples within 4 assays
## Active assay: RNA (33907 features, 1000 variable features)
##  2 layers present: counts, data
##  3 other assays present: SCT, prediction.score.anno, prediction.score.cluster
##  5 dimensional reductions calculated: pca, harmony, umap, ref., ref.umap
```

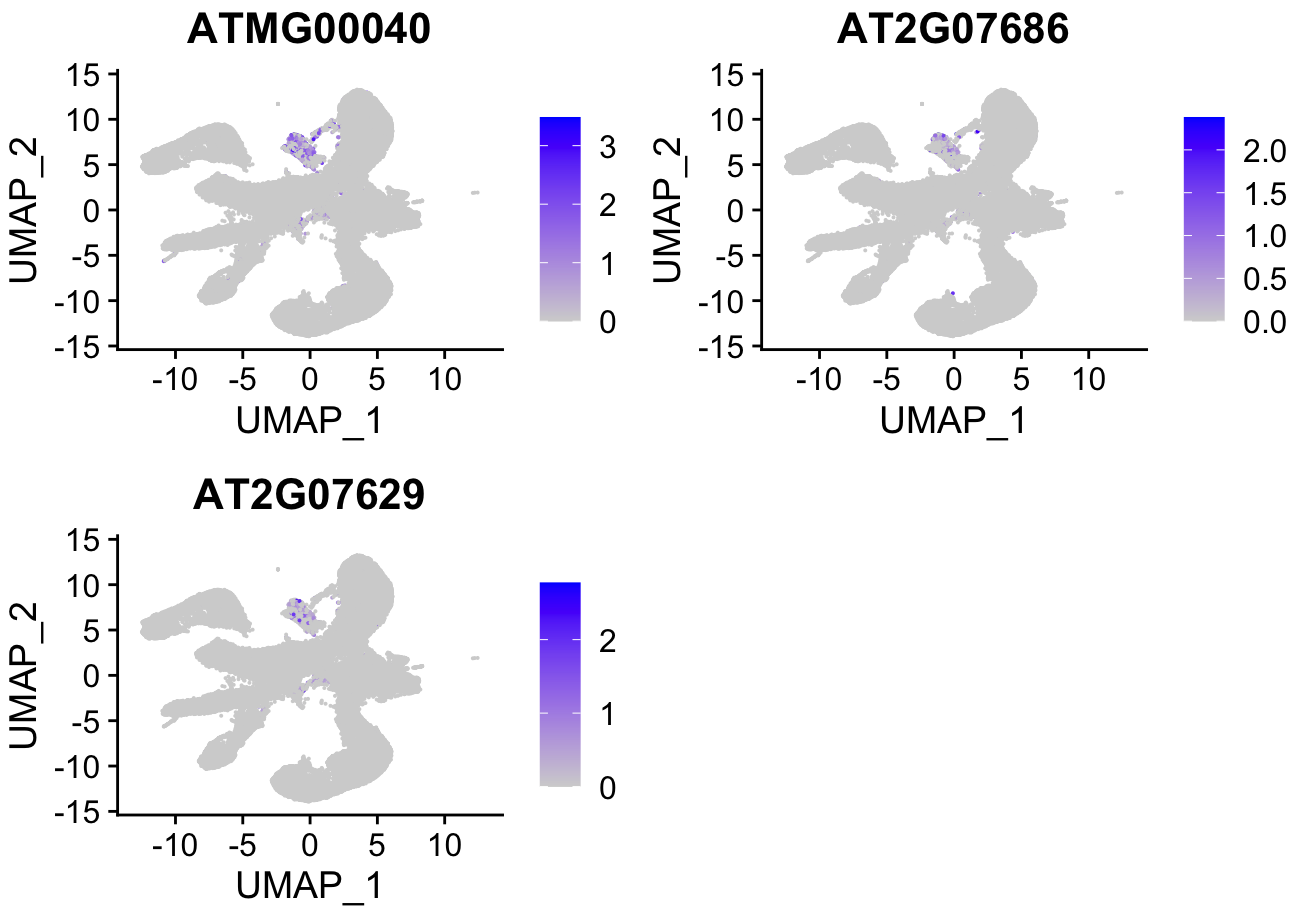
```
Idents(sobj.lt.filter)="cell_type"

markers_cluster=FindAllMarkers(object = sobj.lt.filter,min.pct = .25,logfc.threshold = log2(1.5),
verbose=FALSE) %>% filter(p_val_adj<.05)
```

```
## 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
```

```
Top_maker_cluster <- markers_cluster %>%
    group_by(cluster) %>%
    slice_max(n = 3, order_by = avg_log2FC)

Top_maker_cluster <- filter(Top_maker_cluster, cluster == "Heat_shock")

p1 <- VlnPlot(sobj.lt.filter, features = Top_maker_cluster$gene, ncol =3)
p2 <- FeaturePlot(sobj.lt.filter, features = Top_maker_cluster$gene)
p1
```




```
p2
```




```
write.csv(markers_cluster, "markers_cell_type")
```

```
####Finding differentially expressed features (cluster biomarkers)
library(clusterProfiler)
```

```
## Warning: package 'clusterProfiler' was built under R version 4.3.2
```

```
##
```

```
## 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. clusterProfiler 4.0: A universal enrichment tool for interpreting omics data. The Innovatio
n. 2021, 2(3):100141
```

```
##
## 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(enrichplot)
```

```
## Warning: package 'enrichplot' was built under R version 4.3.2
```

```
library(ggplot2)

df <- markers_cluster
# SET THE DESIRED ORGANISM HERE
organism = "org.At.tair.db"

# Continue detaching other packages

library(organism, character.only = TRUE)
```

```
## 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
```

```
##
```

```
# Continue for other packages


library(organism, character.only = TRUE)

df <- filter(df, cluster == "Heat_shock")
# we want the log2 fold change
original_gene_list <- df$avg_log2FC
# name the vector
unique(df$cluster)
```

```
## [1] Heat_shock
## 9 Levels: Stele LRC Columella Trichoblast Cortex Endodermis ... Heat_shock
```

```
names(original_gene_list) <- df$gene

# omit any NA values
gene_list <- na.omit(original_gene_list)

# sort the list in decreasing order (required for clusterProfiler)
gene_list = sort(gene_list, decreasing = TRUE)

keytypes(org.At.tair.db)
```

```
##  [1] "ARACYC"       "ARACYCENZYME" "ENTREZID"     "ENZYME"       "EVIDENCE"
##  [6] "EVIDENCEALL"  "GENENAME"     "GO"           "GOALL"        "ONTOLOGY"
## [11] "ONTOLOGYALL"  "PATH"         "PMID"         "REFSEQ"       "SYMBOL"
## [16] "TAIR"
```

```
gse <- gseGO(geneList=gene_list,
             ont ="ALL",
             keyType = "TAIR",
             nPerm = 10000,
             minGSSize = 3,
             maxGSSize = 800,
             pvalueCutoff = 0.05,
             verbose = TRUE,
             OrgDb = organism,
             pAdjustMethod = "none")
```

```
## preparing geneSet collections...
```

```
## GSEA analysis...
```

```
## Warning in .GSEA(geneList = geneList, exponent = exponent, minGSSize =
## minGSSize, : We do not recommend using nPerm parameter incurrent and future
## releases
```

```
## Warning in fgsea(pathways = geneSets, stats = geneList, nperm = nPerm, minSize
## = minGSSize, : You are trying to run fgseaSimple. It is recommended to use
## fgseaMultilevel. To run fgseaMultilevel, you need to remove the nperm argument
## in the fgsea function call.
```

```
## leading edge analysis...
```

```
## done...
```

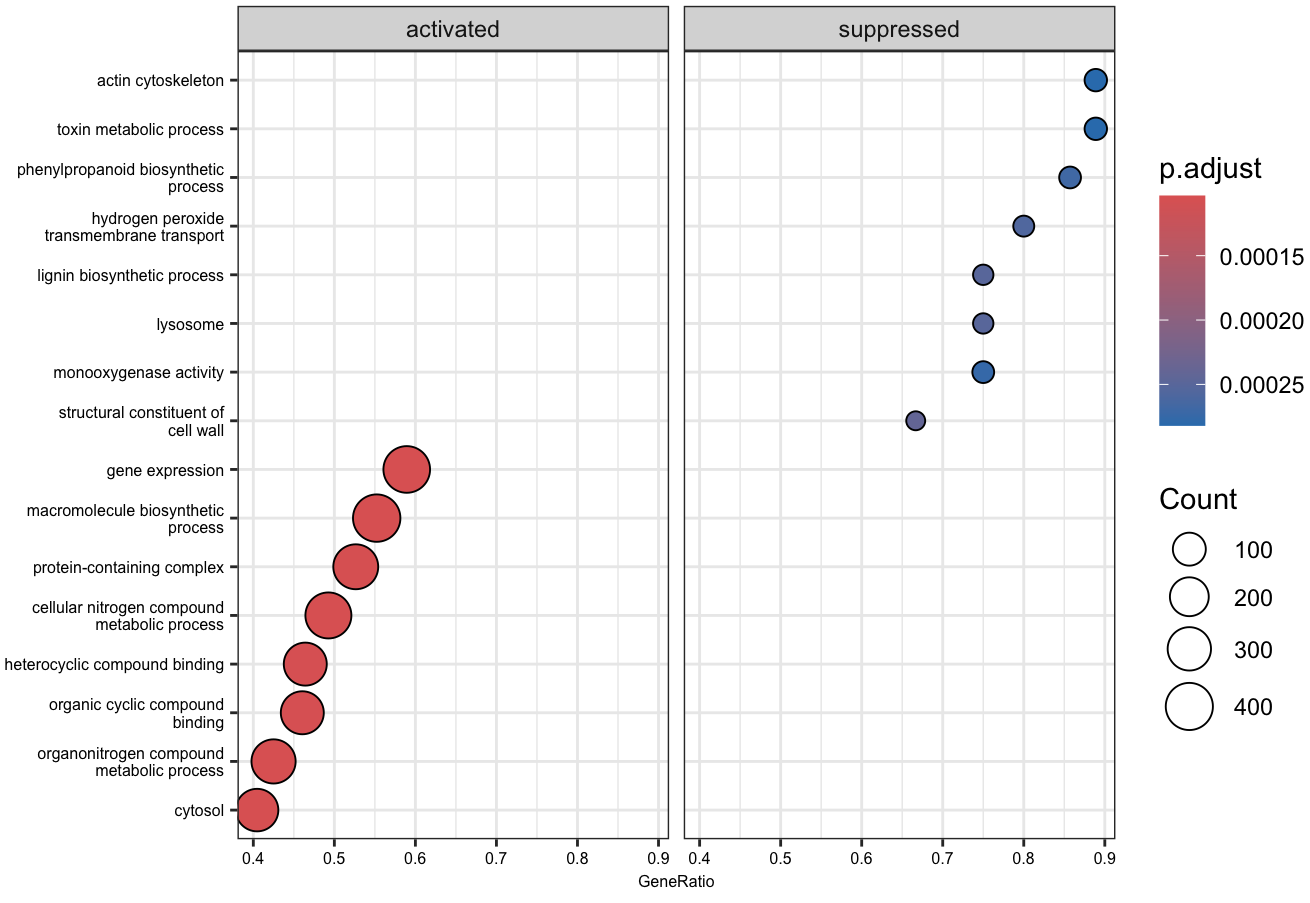
```
###Dotplot
require(DOSE)
```

```
## Loading required package: DOSE
```

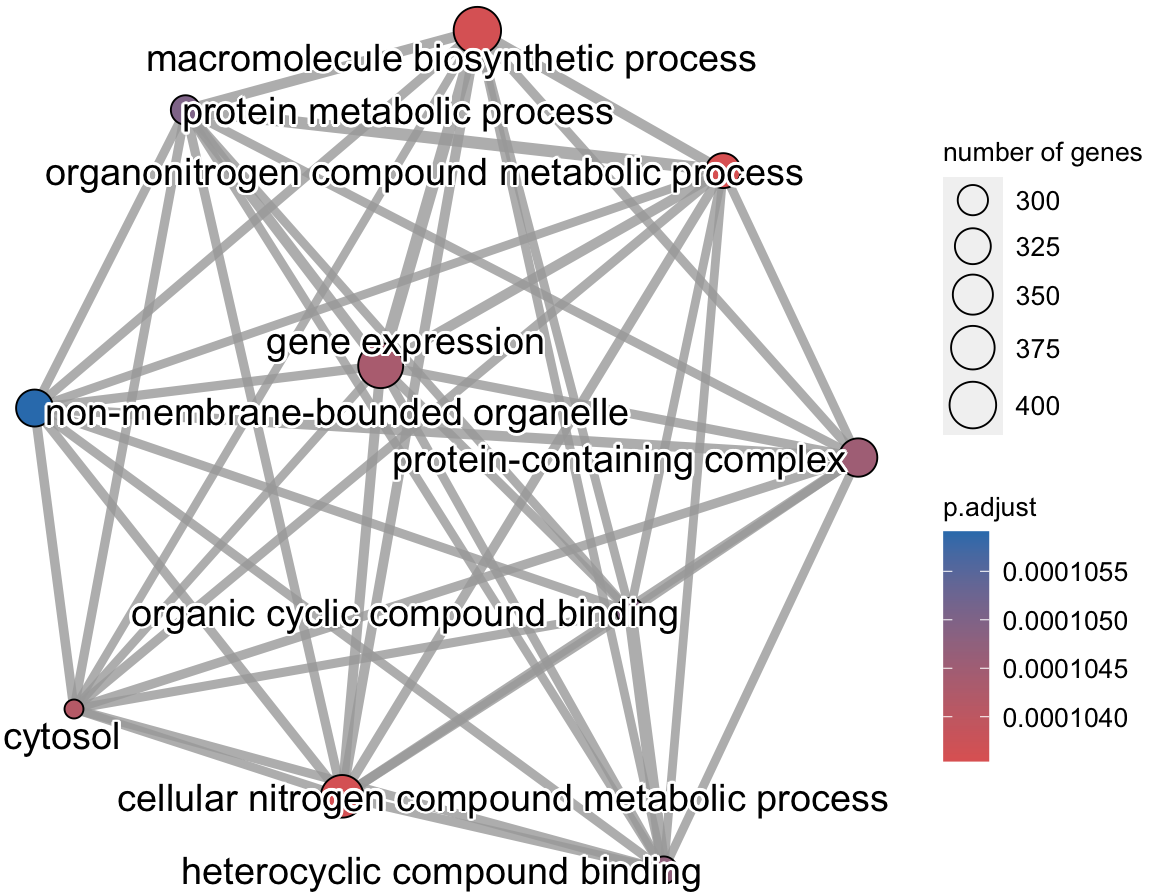
```
## Warning: package 'DOSE' was built under R version 4.3.2
```

```
## DOSE v3.28.1  For help: https://yulab-smu.top/biomedical-knowledge-mining-book/
##
## If you use DOSE in published research, please cite:
## Guangchuang Yu, Li-Gen Wang, Guang-Rong Yan, Qing-Yu He. DOSE: an R/Bioconductor package for D
isease Ontology Semantic and Enrichment analysis. Bioinformatics 2015, 31(4):608-609
```

```
Goenrichment <- dotplot(gse, showCategory=8, split=".sign", font.size = 6) + facet_grid(.~.sign)
Goenrichment
```



```
gse.1 <- pairwise_termsim(gse)
p1 <- emapplot(gse.1, showCategory = 10, font.size = 2)
p1
```




KEGG Gene Set Enrichment Analysis

```
# Convert gene IDs for gseKEGG function
# We will lose some genes here because not all IDs will be converted
ids<-bitr(names(original_gene_list), fromType = "TAIR", toType = "ENTREZID", OrgDb=organism)
```

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

```
## Warning in bitr(names(original_gene_list), fromType = "TAIR", toType =
## "ENTREZID", : 1.95% of input gene IDs are fail to map...
```

```
# remove duplicate IDS (here I use "ENSEMBL", but it should be whatever was selected as keyType)
dedup_ids = ids[!duplicated(ids[c("TAIR")]),]


# Create a new dataframe df2 which has only the genes which were successfully mapped using the bi
tr function above
df2 = df[df$gene %in% dedup_ids$TAIR,]

# Create a new column in df2 with the corresponding ENTREZ IDs
df2$Y = dedup_ids$ENTREZID

# Create a vector of the gene unuiverse
kegg_gene_list <- df2$avg_log2FC

# Name vector with ENTREZ ids
names(kegg_gene_list) <- df2$Y

# omit any NA values
kegg_gene_list<-na.omit(kegg_gene_list)

# sort the list in decreasing order (required for clusterProfiler)
kegg_gene_list = sort(kegg_gene_list, decreasing = TRUE)

kegg_organism = "ath"

kk2 <- gseKEGG(geneList     = kegg_gene_list,
               organism     = kegg_organism,
               nPerm        = 10000,
               minGSSize    = 3,
               maxGSSize    = 800,
               pvalueCutoff = 0.05,
               pAdjustMethod = "none",
               keyType       = "ncbi-geneid")
```

```
## Reading KEGG annotation online: "https://rest.kegg.jp/link/ath/pathway"...
```

```
## Reading KEGG annotation online: "https://rest.kegg.jp/list/pathway/ath"...
```

```
## Reading KEGG annotation online: "https://rest.kegg.jp/conv/ncbi-geneid/ath"...
```

```
## preparing geneSet collections...
```

```
## GSEA analysis...
```

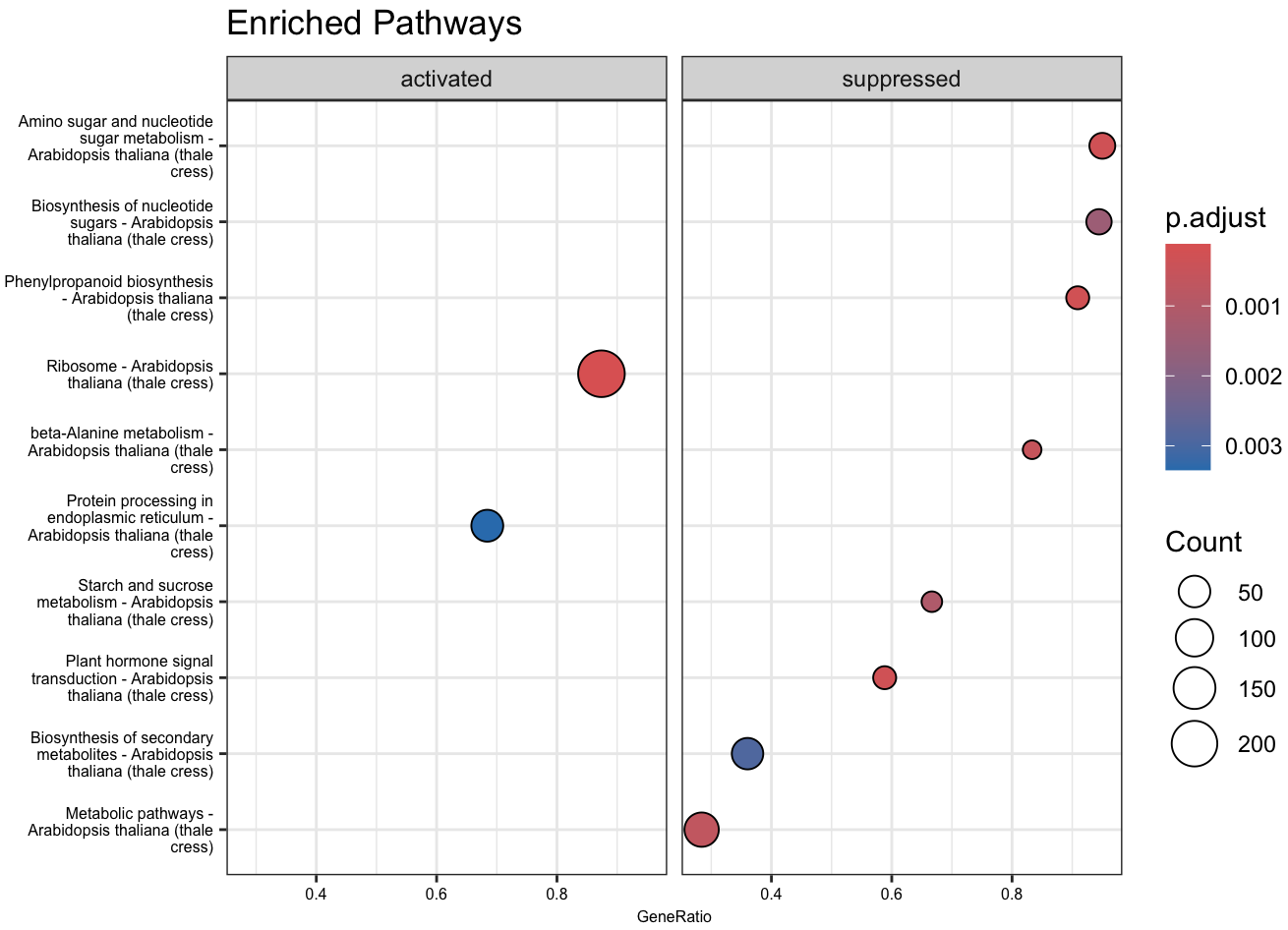
```
## Warning in .GSEA(geneList = geneList, exponent = exponent, minGSSize =
## minGSSize, : We do not recommend using nPerm parameter incurrent and future
## releases
```

```
## Warning in fgsea(pathways = geneSets, stats = geneList, nperm = nPerm, minSize
## = minGSSize, : You are trying to run fgseaSimple. It is recommended to use
## fgseaMultilevel. To run fgseaMultilevel, you need to remove the nperm argument
## in the fgsea function call.
```

```
## leading edge analysis...
```

```
## done...
```

```
p1 <- dotplot(kk2, showCategory = 8, title = "Enriched Pathways" , split=".sign", font.size = 6)
+ facet_grid(.~.sign)

p1
```




```
kk2.1 <- pairwise_termsim(kk2)

p2 <- emapplot(kk2.1)
```