

Proteomics Muscle Cox7a KO- Garcia-Poyotas, 2024

Code ▾

AUTHOR

Prateek Arora

PUBLISHED

January 24, 2024

1 Clear memory, set working directory, load packages

▼ Code

```
rm(list=ls())
gc()
setwd("/mnt/Data_8TB/Carolina_data/Cell_paper/")
getwd()
```

A matrix: 2 × 6 of type dbl

	used	(Mb)	gc trigger	(Mb)	max used	(Mb)
Ncells	637193	34.1	1411248	75.4	1411248	75.4
Vcells	1183427	9.1	8388608	64.0	1815676	13.9

'/mnt/Data_8TB/Carolina_data/Cell_paper'

▼ Code

```
library(tidyverse) # metapackage of all tidyverse packages
library(clusterProfiler) # for GO analysis
library(org.Mm.eg.db) # for GO analysis
library(org.Hs.eg.db) # for GO analysis
library(ComplexHeatmap) # for heatmap
library(paletteer) # for color palettes
library(openxlsx) # for writing excel files
library(limma) # for DE analysis
library(ggVennDiagram) # for venn diagrams
```

```
— Attaching core tidyverse packages ————— tidyverse 2.0.0 —
✓ dplyr     1.1.2      ✓ readr     2.1.4
✓forcats   1.0.0      ✓ stringr   1.5.0
✓ ggplot2   3.4.2      ✓ tibble    3.2.1
✓ lubridate 1.9.2      ✓ tidyverse  1.3.0
✓ purrr    1.0.1
— 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
```

```
clusterProfiler v4.8.1 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 Innovation. 2021, 2(3):100141
```

```
Attaching package: 'clusterProfiler'
```

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

```
simplify
```

```
The following object is masked from 'package:stats':
```

```
filter
```

```
Loading required package: AnnotationDbi
```

```
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 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: Biobase

Welcome to Bioconductor

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

Loading required package: IRanges

Loading required package: S4Vectors

Attaching package: 'S4Vectors'

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

```
rename
```

The following objects are masked from 'package:lubridate':

```
second, second<-
```

The following objects are masked from 'package:dplyr':

```
first, rename
```

The following object is masked from 'package:tidyR':

```
expand
```

The following object is masked from 'package:utils':

```
findMatches
```

The following objects are masked from 'package:base':

```
expand.grid, I, unname
```

Attaching package: 'IRanges'

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

slice

The following object is masked from 'package:lubridate':

%within%

The following objects are masked from 'package:dplyr':

collapse, desc, slice

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

reduce

Attaching package: 'AnnotationDbi'

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

select

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

select

Loading required package: grid

=====

ComplexHeatmap version 2.16.0

Bioconductor page: <http://bioconductor.org/packages/ComplexHeatmap/>

Github page: <https://github.com/jokergoo/ComplexHeatmap>

Documentation: <http://jokergoo.github.io/ComplexHeatmap-reference>

If you use it in published research, please cite either one:

- Gu, Z. Complex Heatmap Visualization. iMeta 2022.
- Gu, Z. Complex heatmaps reveal patterns and correlations in multidimensional genomic data. Bioinformatics 2016.

The new InteractiveComplexHeatmap package can directly export static complex heatmaps into an interactive Shiny app with zero effort. Have a try!

```
This message can be suppressed by:  
 suppressPackageStartupMessages(library(ComplexHeatmap))  
=====
```

Attaching package: 'limma'

The following object is masked from 'package:BiocGenerics':

plotMA

2 Import the data and perform limma analysis

▼ Code

```
# Load data  
data <- read.xlsx("./LIMMA_Muscle.xlsx")  
data <- data[-c(1:3),] # remove first 3 rows  
colnames(data) <- data[1,] # set column names  
data <- data[-1,] # remove first row  
data$gene[is.na(data$gene)] <- paste0("NA_", data$protein[is.na(data$gene)]) #  
# replace NA gene names with NA_protein names  
data$gene[is.na(data$gene)]  
data$gene <- make.unique(data$gene, sep = "_") # make gene names unique  
rownames(data) <- data$gene # set row names  
head(data) # check data
```

	protein	AKO1	AKO2	AKO3
	<chr>	<chr>	<chr>	<chr>
ppp4r3b	AOA0G2KC70	2.0724701109999999	5.3395652289999997	1.8406234159999999
ehmt2	AOA0G2KCN9	-1.657059426	2.4307019670000001	1.7876982029999999
col28a1b	AOA0G2KCR2	1.9575273870000001	2.6430229430000001	1.700726089
ddb1	AOA0G2KCW6	-0.3990389409999998	-0.94651125300000005	-0.79190209700000003
comp	AOA0G2KDJ5	-0.57219328000000003	0.47831322799999998	0.47560477499999998

	protein	AKO1	AKO2	AKO3
	<chr>	<chr>	<chr>	<chr>
ptprd	A0A0G2KEA8	-0.1248520719999999	-0.4801183629999999	1.9367150689999999

▼ Code

```
data_selected <- data[,c(2:10)] # select only the columns with protein expression
data
data_selected <- data_selected %>%
  mutate_at(c(1:9), as.numeric) # convert each column to numeric
data_selected <- as.matrix(data_selected) # convert to matrix
head(data_selected)
```

A matrix: 6 × 9 of type dbl

	AKO1	AKO2	AKO3	AWT1	AWT2	AWT3	JWT1
ppp4r3b	2.0724701	5.3395652	1.8406234	1.6077886	1.33812441	-1.02774047	0.7308176
ehmt2	-1.6570594	2.4307020	1.7876982	1.7745410	2.35141282	2.91844745	-3.7952009
col28a1b	1.9575274	2.6430229	1.7007261	1.2173278	1.36063600	2.04355012	-1.7788666
ddb1	-0.3990389	-0.9465113	-0.7919021	-1.5153913	-0.60943715	-0.56973959	0.7213614
comp	-0.5721933	0.4783132	0.4756048	0.2236121	0.09977164	-0.05334311	-0.3563766
ptprd	-0.1248521	-0.4801184	1.9367151	1.6849042	-0.76818314	1.01717945	-0.4363621

▼ Code

```
# Create metadata
metadata <- data.frame(samples = colnames(data_selected),
                        group = c(rep("Adult_KO", 3), rep("Adult_WT", 3),
                        rep("Juvenile_WT", 3))) # create metadata dataframe
metadata
```

A data.frame: 9 × 2

samples	group
<chr>	<chr>
AKO1	Adult_KO
AKO2	Adult_KO
AKO3	Adult_KO
AWT1	Adult_WT
AWT2	Adult_WT

samples	group
<chr>	<chr>
AWT3	Adult_WT
JWT1	Juvenile_WT
JWT2	Juvenile_WT
JWT3	Juvenile_WT

▼ Code

```
# perform limma analysis between scaf and wt to get p-values
design <- model.matrix(~0+metadata$group) # create design matrix
colnames(design) <- c("Adult_KO", "Adult_WT", "Juvenile_WT") # set column names
design
fit <- lmFit(data_selected, design)
fit
```

A matrix: 9 × 3 of type dbl

	Adult_KO	Adult_WT	Juvenile_WT
1	1	0	0
2	1	0	0
3	1	0	0
4	0	1	0
5	0	1	0
6	0	1	0
7	0	0	1
8	0	0	1
9	0	0	1

```
An object of class "MArrayLM"
$coefficients
    Adult_KO   Adult_WT Juvenile_WT
ppp4r3b  3.0842196  0.63939084  0.4968233
ehmt2    0.8537802  2.34813376 -4.2422847
col28a1b 2.1004255  1.54050465 -1.7416860
ddb1     -0.7124841 -0.89818935  0.7835218
comp      0.1272416  0.09001354 -0.5229877
2097 more rows ...
```

```
$rank
[1] 3
```

```
$assign
[1] 1 1 1
```

```

$qr
$qr
  Adult_KO  Adult_WT Juvenile_WT
1 -1.7320508  0.0000000  0.0000000
2  0.5773503 -1.7320508  0.0000000
3  0.5773503  0.0000000 -1.7320508
4  0.0000000  0.5773503  0.0000000
5  0.0000000  0.5773503  0.0000000
6  0.0000000  0.5773503  0.0000000
7  0.0000000  0.0000000  0.5773503
8  0.0000000  0.0000000  0.5773503
9  0.0000000  0.0000000  0.5773503
attr("assign")
[1] 1 1 1
attr("contrasts")
attr("contrasts")$`metadata$group`
[1] "contr.treatment"

$qraux
[1] 1.57735 1.00000 1.00000

$pivot
[1] 1 2 3

$tol
[1] 1e-07

$rank
[1] 3

$df.residual
[1] 6 6 6 6 6
2097 more elements ...

$\sigma
  ppp4r3b    ehmt2  col28a1b      ddb1      comp
1.4573700 1.3409254 0.4634475 0.6125249 0.3894319
2097 more elements ...

$cov.coefficients
  Adult_KO  Adult_WT Juvenile_WT
Adult_KO  0.3333333  0.0000000  0.0000000
Adult_WT   0.0000000  0.3333333  0.0000000
Juvenile_WT 0.0000000  0.0000000  0.3333333

$stdev.unscaled
  Adult_KO  Adult_WT Juvenile_WT
ppp4r3b  0.5773503  0.5773503  0.5773503
ehmt2    0.5773503  0.5773503  0.5773503
col28a1b 0.5773503  0.5773503  0.5773503
ddb1     0.5773503  0.5773503  0.5773503

```

```

comp      0.5773503 0.5773503   0.5773503
2097 more rows ...

$pivot
[1] 1 2 3

$Amean
  ppp4r3b     ehmt2   col28a1b      ddb1      comp
1.4068112 -0.3467902  0.6330814 -0.2757172 -0.1019109
2097 more elements ...

$method
[1] "ls"

$design
  Adult_KO Adult_WT Juvenile_WT
1       1       0       0
2       1       0       0
3       1       0       0
4       0       1       0
5       0       1       0
6       0       1       0
7       0       0       1
8       0       0       1
9       0       0       1

attr("assign")
[1] 1 1 1
attr("contrasts")
attr("contrasts")$`metadata$group`
[1] "contr.treatment"

```

▼ Code

```

# "Adult_KO", "Adult_WT", "Juvenile_WT"
cont_matrix <- makeContrasts( # create contrast matrix
adult_ko_vs_adult_wt=Adult_KO-Adult_WT,
adult_ko_vs_juvenile_wt=Adult_KO-Juvenile_WT,
adult_wt_vs_juvenile_wt=Adult_WT-Juvenile_WT,
levels=design)
cont_matrix # check contrast matrix
fit2 <- contrasts.fit(fit, cont_matrix) # fit contrast matrix
fit2 <- eBayes(fit2) # perform empirical bayes analysis
# fit2

```

A matrix: 3 × 3 of type dbl

	adult_ko_vs_adult_wt	adult_ko_vs_juvenile_wt	adult_wt_vs_juvenile_wt
Adult_KO	1	1	0
Adult_WT	-1	0	1
Juvenile_WT	0	-1	-1

▼ Code

```

adult_ko_vs_adult_wt <- topTable(fit2, coef = 1, adjust.method = "BH", sort.by = "p",
                                    number = Inf) # get full table
adult_ko_vs_adult_wt <- adult_ko_vs_adult_wt %>% dplyr::select(logFC, P.Value,
                                                               adj.P.Val) # select only logFC, P.Value, adj.P.Val columns
colnames(adult_ko_vs_adult_wt) <-
  paste0("adult_ko_vs_adult_wt_", colnames(adult_ko_vs_adult_wt)) # rename
  columns
adult_ko_vs_adult_wt <- adult_ko_vs_adult_wt %>% rownames_to_column("gene") # add
  gene column
adult_ko_vs_adult_wt %>%
  
```



```

adult_ko_vs_juvenile_wt <- topTable(fit2, coef = 2, adjust.method = "BH", sort.by =
  "p", number = Inf) # get full table
adult_ko_vs_juvenile_wt <- adult_ko_vs_juvenile_wt %>% dplyr::select(logFC, P.Value,
                                                               adj.P.Val) # select only logFC, P.Value, adj.P.Val columns
colnames(adult_ko_vs_juvenile_wt) <-
  paste0("adult_ko_vs_juvenile_wt_", colnames(adult_ko_vs_juvenile_wt)) # rename
  columns
adult_ko_vs_juvenile_wt <- adult_ko_vs_juvenile_wt %>% rownames_to_column("gene") # add
  gene column
adult_ko_vs_juvenile_wt %>% head
  
```



```

adult_wt_vs_juvenile_wt <- topTable(fit2, coef = 3, adjust.method = "BH", sort.by =
  "p", number = Inf) # get full table
adult_wt_vs_juvenile_wt <- adult_wt_vs_juvenile_wt %>% dplyr::select(logFC, P.Value,
                                                               adj.P.Val) # select only logFC, P.Value, adj.P.Val columns
colnames(adult_wt_vs_juvenile_wt) <-
  paste0("adult_wt_vs_juvenile_wt_", colnames(adult_wt_vs_juvenile_wt)) # rename
  columns
adult_wt_vs_juvenile_wt <- adult_wt_vs_juvenile_wt %>% rownames_to_column("gene") # add
  gene column
adult_wt_vs_juvenile_wt %>% head # check data
  
```

A data.frame: 6 × 4

gene	adult_ko_vs_adult_wt_logFC	adult_ko_vs_adult_wt_PValue	adult_ko_vs_adult_wt_adj.PVal
<chr>	<dbl>	<dbl>	<dbl>
1 cox7a1	-7.090382	8.497674e-08	0.0001786211
2 dnm2a	3.418267	1.055365e-05	0.0071750944
3 cox5aa	-5.075714	1.332007e-05	0.0071750944
4 cox5ab	-6.305060	1.365384e-05	0.0071750944
5 decr1	3.735197	1.728903e-05	0.0072683078
6 ptgr1.1	6.343476	2.480240e-05	0.0086891062

A data.frame: 6 × 4

gene	adult_ko_vs_juvenile_wt_logFC	adult_ko_vs_juvenile_wt_PValue	adult_ko_vs_juvenile_wt_a
<chr>	<dbl>	<dbl>	<dbl>
1			
2			
3			
4			
5			
6			

gene	adult_ko_vs_juvenile_wt_logFC	adult_ko_vs_juvenile_wt_PValue	adult_ko_vs_juvenile_wt_a
<chr>	<dbl>	<dbl>	<dbl>
1 gstr	8.173132	2.016344e-09	4.238356e-06
2 rpl23	-6.813533	6.696831e-09	7.038369e-06
3 rpl23a	-7.704851	1.930364e-08	1.352542e-05
4 rps8	-9.093287	3.722932e-08	1.369146e-05
5 myha	-12.315355	4.688831e-08	1.369146e-05
6 eef1a1l1	-7.219879	5.401093e-08	1.369146e-05

A data.frame: 6 × 4

gene	adult_wt_vs_juvenile_wt_logFC	adult_wt_vs_juvenile_wt_PValue	adult_wt_vs_juvenile_wt_a
<chr>	<dbl>	<dbl>	<dbl>
1 rpl23	-7.979944	1.647233e-09	1.898943e-06
2 gstr	8.274604	1.806796e-09	1.898943e-06
3 rpl23a	-8.594055	7.360111e-09	3.896835e-06
4 aars1	-5.974487	1.128217e-08	3.896835e-06
5 rps8	-10.404265	1.137604e-08	3.896835e-06
6 myha	-14.393014	1.189957e-08	3.896835e-06

◀ ▶

▼ Code

```
# merge data, adult_ko_vs_adult_wt, adult_ko_vs_juvenile_wt, adult_wt_vs_juvenile_wt
# by column gene
data_merged <- Reduce(function(x, y) merge(x, y, by = "gene", all = TRUE),
list(data,adult_ko_vs_adult_wt, adult_ko_vs_juvenile_wt,
adult_wt_vs_juvenile_wt)) # merge data
data_merged <- data_merged %>% dplyr::select(-contains("limma")) # remove columns
# with limma
# remove _* from gene column
data_merged$gene <- gsub("_", "", data_merged$gene) # remove _* from gene column

data_merged %>% head
```

2102 · 21
2102 · 4
2102 · 4
2102 · 4

gene	protein	AKO1	AKO2	AKO3
<chr>	<chr>	<chr>	<chr>	<chr>

gene	protein	AKO1	AKO2	AKO3
<chr>	<chr>	<chr>	<chr>	<chr>
1 aamdc	Q502H1	1.466751474999999	-0.5739484550000002	-1.865304979
2 aars1	A0A286YAH8	-1.3534255310000001	-1.5746633350000001	-1.794735197999999
3 aass	E7FD66	0.7297958780000001	1.3124486870000001	0.2894110169999999
4 abcb10	F1Q5K6	0.28526026100000001	1.243046010999999	0.6226767509999997
5 abcb11a	F6NLY2	0.6139335449999997	2.1746057840000002	1.228077936
6 abcb11b	A0A0R4IVE7	0.3971807560000002	-0.2657105779999998	-0.6573270559999997

3 Add annotation

▼ Code

```
# pull description of genes from biomart and merge it to data_merged
# get the description of the genes
library(biomart)
zf_mart <- useMart("ensembl", dataset = "drerio_gene_ensembl") # set biomart
description_df <- getBM(attributes = c("ensembl_gene_id", # get ensemble gene id
                                         "zfin_id_symbol", # get zfin id
                                         "description", # get description
                                         "definition_1006", # get definition from GO
                                         "name_1006", # get name from GO
                                         "namespace_1003", # get namespace from GO
                                         (molecular function, biological process, cellular component)
                                         "phenotype_description"), # get phenotype
                                         description
                                         filters = "zfin_id_symbol", # filter by zfin id
                                         values = data_merged$gene, # values as input for filter
                                         mart = zf_mart) # set biomart
head(description_df)
```

A data.frame: 6 × 7

	ensembl_gene_id	zfin_id_symbol	description	definition_1006	name_1006	nan
	<chr>	<chr>	<chr>	<chr>	<chr>	<ch
1	ENSDARG00000103957	aamdc	adipogenesis associated, Mth938 domain containing [Source:ZFIN;Acc:ZDB- GENE-050522-64]			

ensembl_gene_id	zfin_id_symbol	description	definition_1006	name_1006	nan
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
2	ENSDARG00000069142	aars1	alanyl-tRNA synthetase 1 [Source:ZFIN;Acc:ZDB-GENE-030131-3663]	Binding to a metal ion.	metal ion binding
3	ENSDARG00000069142	aars1	alanyl-tRNA synthetase 1 [Source:ZFIN;Acc:ZDB-GENE-030131-3663]	Binding to ATP, adenosine 5'-triphosphate, a universally important coenzyme and enzyme regulator.	ATP binding
4	ENSDARG00000069142	aars1	alanyl-tRNA synthetase 1 [Source:ZFIN;Acc:ZDB-GENE-030131-3663]	Binding to a nucleic acid.	nucleic acid binding
5	ENSDARG00000069142	aars1	alanyl-tRNA synthetase 1 [Source:ZFIN;Acc:ZDB-GENE-030131-3663]	Binding to a nucleotide, any compound consisting of a nucleoside that is esterified with (ortho)phosphate or an oligophosphate at any hydroxyl group on the ribose or deoxyribose.	nucleotide binding
6	ENSDARG00000069142	aars1	alanyl-tRNA synthetase 1 [Source:ZFIN;Acc:ZDB-GENE-030131-3663]	Binding to an RNA molecule or a portion thereof.	RNA binding

▼ Code

```
# collapse duplicated genes(zfin_id_symbol) to one row
description_df <- description_df %>%
  group_by(zfin_id_symbol) %>%
  summarise(description = paste0(unique(description), collapse = "; "),
            definition_1006 = paste0(unique(definition_1006), collapse = "; "),
            phenotype_description = paste0(unique(phenotype_description), collapse =
= "; ")),
```

```

    name_1006 = paste0(unique(name_1006), collapse = "; "),
    namespace_1003 = paste0(unique(namespace_1003), collapse = "; ")
  )
# rename zfin_id_symbol to gene and definition_1006 to GO_description, description to
# gene_full_name, phenotype_description to associated_phenotype using
dplyr::rename
description_df <- description_df %>%
  dplyr::rename(gene = zfin_id_symbol,
                "GO_description" = definition_1006,
                "gene_full_name" = description,
                "associated_phenotype" = phenotype_description,
                "GO_name" = name_1006,
                "GO_family" = namespace_1003)
head(description_df, 2)

```

A tibble: 2 × 6

gene	gene_full_name	GO_description	associated_phenotype	GO_name	GO_family
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
aamdc	adipogenesis associated, Mth938 domain containing [Source:ZFIN;Acc:ZDB- GENE-050522-64]				
aars1	alanyl-tRNA synthetase 1 [Source:ZFIN;Acc:ZDB- GENE-030131-3663]	Binding to a metal ion.; Binding to ATP, adenosine 5'- triphosphate, a universally important coenzyme and enzyme regulator.; Binding to a nucleic acid.; Binding to a nucleotide, any compound consisting of a nucleoside that is esterified with (ortho)phosphate or an oligophosphate at any hydroxyl group on the	caudal fin morphology; eye decreased size; motor neuron spatial pattern; whole organism curved; whole organism decreased length; whole organism morphology	metal ion binding; ATP binding; nucleic acid binding; nucleotide binding; RNA binding; mitochondrion; aminoacyl- tRNA ligase activity; ligase activity; translation; tRNA aminoacylation; tRNA binding; cytoplasm; amino acid binding; zinc ion binding; aminoacyl- tRNA metabolism	molecular_fu cellular_com biological_pr acid binding; nucleotide binding; RNA binding; mitochondrion; aminoacyl- tRNA ligase activity; ligase activity; translation; tRNA aminoacylation; tRNA binding; cytoplasm; amino acid binding; zinc ion binding; aminoacyl- tRNA metabolism

gene	gene_full_name	GO_description	associated_phenotype	GO_name	GO_family
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
		<p>ribose or deoxyribose.; Binding to an RNA molecule or a portion thereof.; A semiautonomous, self replicating organelle that occurs in varying numbers, shapes, and sizes in the cytoplasm of virtually all eukaryotic cells.</p> <p>It is notably the site of tissue respiration.;</p> <p>Catalysis of the formation of aminoacyl-tRNA from ATP, amino acid, and tRNA with the release of diphosphate and AMP.;</p> <p>Catalysis of the joining of two molecules, or two groups within a single molecule, using the energy from the hydrolysis of ATP, a similar triphosphate, or a pH gradient.;</p> <p>The cellular metabolic process in which a protein is formed, using the sequence of a mature mRNA or</p>		<p>involved in translational fidelity; tRNA modification; aminoacyl-tRNA editing activity; alanine-tRNA ligase activity; alanyl-tRNA aminoacylation; tRNA metabolic process; catalytic activity, acting on a tRNA</p>	

gene	gene_full_name	GO_description	associated_phenotype	GO_name	GO_family
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
		<p>circRNA</p> <p>molecule to</p> <p>specify the</p> <p>sequence of</p> <p>amino acids in a</p> <p>polypeptide</p> <p>chain. Translation</p> <p>is mediated by</p> <p>the ribosome,</p> <p>and begins with</p> <p>the formation of</p> <p>a ternary</p> <p>complex between</p> <p>aminoacylated</p> <p>initiator</p> <p>methionine</p> <p>tRNA, GTP, and</p> <p>initiation factor</p> <p>2, which</p> <p>subsequently</p> <p>associates with</p> <p>the small subunit</p> <p>of the ribosome</p> <p>and an mRNA or</p> <p>circRNA.</p> <p>Translation ends</p> <p>with the release</p> <p>of a polypeptide</p> <p>chain from the</p> <p>ribosome.; The</p> <p>chemical</p> <p>reactions and</p> <p>pathways by</p> <p>which the various</p> <p>amino acids</p> <p>become bonded</p> <p>to their</p> <p>corresponding</p> <p>tRNAs. The most</p> <p>common route</p> <p>for synthesis of</p> <p>aminoacyl tRNA</p> <p>is by the</p> <p>formation of an</p>			

gene	gene_full_name	GO_description	associated_phenotype	GO_name	GO_family
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
		ester bond between the 3'-hydroxyl group of the most 3' adenosine of the tRNA and the alpha carboxylic acid group of an amino acid, usually catalyzed by the cognate aminoacyl-tRNA ligase. A given aminoacyl-tRNA ligase aminoacylates all species of an isoaccepting group of tRNA molecules.; Binding to a transfer RNA.; The contents of a cell excluding the plasma membrane and nucleus, but including other subcellular structures.; Binding to an amino acid, organic acids containing one or more amino substituents.; Binding to a zinc ion (Zn).; Any process which detects an amino-acid acetylated tRNA is charged with the correct			

gene	gene_full_name	GO_description	associated_phenotype	GO_name	GO_family
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
		<p>amino acid, or removes incorrect amino acids from a charged tRNA.</p> <p>This process can be performed by tRNA synthases, or by subsequent reactions after tRNA aminoacylation.;</p> <p>The covalent alteration of one or more nucleotides within a tRNA molecule to produce a tRNA molecule with a sequence that differs from that coded genetically.; The hydrolysis of an incorrectly aminoacylated tRNA.; Catalysis of the reaction: ATP + L-alanine + tRNA(Ala) = AMP + diphosphate + L-alanyl-tRNA(Ala).; The process of coupling alanine to alanyl-tRNA, catalyzed by alanyl-tRNA synthetase. The alanyl-tRNA synthetase is a class-II synthetases. The</p>			

gene	gene_full_name	GO_description	associated_phenotype	GO_name	GO_family
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
		<p>activated amino acid is transferred to the 3'-OH group of an alanine accepting tRNA.; The chemical reactions and pathways involving tRNA, transfer RNA, a class of relatively small RNA molecules responsible for mediating the insertion of amino acids into the sequence of nascent polypeptide chains during protein synthesis.</p> <p>Transfer RNA is characterized by the presence of many unusual minor bases, the function of which has not been completely established.;</p> <p>Catalytic activity that acts to modify a tRNA.</p>			

▼ Code

```
df_merged_description <- merge(data_merged, description_df, by.x = "gene", by.y = "gene", all.x = TRUE)
head(df_merged_description, 1)
```

	gene	protein	AKO1	AKO2	AKO3	AWT1
1	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
	aamdc	Q502H1	1.466751474999999	-0.5739484550000002	-1.865304979	0.2672544270000

▼ Code

```
write.csv(df_merged_description, file = "./LIMMA_Muscle_logfc_description.csv",
          row.names = FALSE)
```

▼ Code

```
write.csv(data_merged, file = "./LIMMA_Muscle_logfc.csv", row.names = FALSE)
```

4 Perform Pathway analysis

4.1 GO analysis

▼ Code

```
GO_function <- function(gene_list, pval = 0.05, onto= "MF", prefix="", org="mouse"){
library(clusterProfiler)
library(org.Mm.eg.db)
library(org.Hs.eg.db)
library(ReactomePA)
if(org=="mouse"){
  orgdb <- "org.Mm.eg.db"
  org_reactome <- "mouse"
} else if(org=="human"){
  orgdb <- "org.Hs.eg.db"
  org_reactome <- "human"
} else{
  message("Please enter a valid organism (mouse or human)")
}

if(onto %in% c("MF", "CC", "BP")){
  compGO <- enrichGO(gene = gene_list, pvalueCutoff = pval, keyType = "SYMBOL",
                      pAdjustMethod = "BH", OrgDb = orgdb, ont = onto)
} else if(onto=="reactome"){
  gene_list <- bitr(gene_list, fromType = "SYMBOL", toType = "ENTREZID", OrgDb
  = orgdb)
  gene_list <- gene_list$ENTREZID
```

```

compGO <- enrichPathway(gene = gene_list, pvalueCutoff = 0.05, organism=
org_reactome,           readable = TRUE)

}else{
  message("Please enter a valid GO term")
}

if(is.null(compGO)){
  message(paste0("No GO:",onto, " obtained"))
  message(paste0("*****"))
  message(paste0("\n"))

}else {

  compGO_df <- as.data.frame(compGO)
  compGO_df$GeneRatio_decimal <- compGO_df$GeneRatio
  compGO_df$GeneRatio_decimal <- sapply(compGO_df$GeneRatio_decimal,
                                         function(x) (eval(parse(text =
as.character(x)))))

  compGO_df$BgRatio_decimal <- compGO_df$BgRatio
  compGO_df$BgRatio_decimal <- sapply(compGO_df$BgRatio_decimal,
                                         function(x) (eval(parse(text =
as.character(x)))))

  compGO_df <- compGO_df %>% tidyverse::separate_rows(geneID, sep = "/", convert =
FALSE) %>%
    arrange(desc(GeneRatio_decimal))
  compGO_df %>% head

  if(nrow(compGO_df)==0){
    message(paste0("No GP:",onto, " obtained"))

    message(paste0("*****"))
    message(paste0("\n"))

  } else{

    write.csv(compGO_df, paste0(prefix,"_GO_",onto, "_pathways.csv"))

    full_name= switch(onto,
                      MF= "Molecular Function",
                      CC= "Cellular Components",
                      BP= "Biological Pathways",
                      reactome= "Reactome Pathways"
                      )

    print(dotplot(compGO, showCategory = 15, title = paste0("GO Pathway
Enrichment Analysis \n",full_name),
                  font.size = 12))
    dev.copy(
      pdf,
      file = paste0(prefix,"_GO_",onto, "_pathways.pdf"),
      width = 10,
      height = 12
    )
  }
}

```

```

        )
dev.off()

message(paste0("Pathway analysis GO:",onto, " for Cell Type : ", " done"))
message(paste0("*****"))
message(paste0("\n"))

}

}

}

```

▼ Code

```

# Function to convert zebrafish gene names to human gene names
convertDanioGeneList_Mouse <- function(x){
  require("biomaRt") # load biomaRt package
  mouse = useEnsembl("ensembl", dataset = "mmusculus_gene_ensembl", host =
    "https://dec2021.archive.ensembl.org") # use human mart
  danio = useEnsembl("ensembl", dataset = "drerio_gene_ensembl", host =
    "https://dec2021.archive.ensembl.org") # use zebrafish mart

  genesV2 = getLDS(attributes = c("ensembl_gene_id", "zfin_id_symbol"),
    filters = "zfin_id_symbol", # get zebrafish gene names
    values = x, # use the zebrafish gene names
    mart = danio, # use the zebrafish mart
    attributesL = c("mgi_symbol", "ensembl_gene_id", "description"
      ), # get human gene names
    martL = mouse, uniqueRows=T) # use the human mart

  colnames(genesV2)[colnames(genesV2)== "Gene.stable.ID"] <- "EnsmbIID_Zebrafish" # rename columns
  colnames(genesV2)[colnames(genesV2)== "Gene.stable.ID.1"] <- "EnsmbIID_Mouse" # rename columns

  # Print the first 6 genes found to the screen
  # print(head(genesV2))
  return(genesV2) # return the genes
}

```

▼ Code

```

mouse_genes <- convertDanioGeneList_Mouse(df_merged_description$gene)
colnames(mouse_genes)

```

'EnsmbIID_Zebrafish' · 'ZFIN.symbol' · 'MGI.symbol' · 'EnsmbIID_Mouse' · 'Gene.description'

▼ Code

```

df_merged_description[grep("kif",df_merged_description$gene),]

```

gene	protein	AKO1	AKO2	AKO3	
<chr>	<chr>	<chr>	<chr>	<chr>	
836	kif13bb	E7F084	-1.903475904	-3.1895115280000002	-2.904640949
837	kif16ba	AOA0R4ILH1	2.569494693999999	-0.9892400869999999	0.2412342839999999
838	kif18a	B0V302	0.7411851050000001	0.9515150169999996	1.387779482

▼ Code

```
mouse_genes[grepl("kif", mouse_genes$ZFIN.symbol), ]
```

A data.frame: 1 × 5

EnsmbIID_Zebrafish	ZFIN.symbol	MGI.symbol	EnsmbIID_Mouse	Gene.description
<chr>	<chr>	<chr>	<chr>	<chr>
100 ENSDARG00000073859	kif13bb	Kif13b	ENSMUSG00000060012	kinesin family mem [Source:MGI Symbol;Acc:MGI:100]

▼ Code

```
mouse_genes$MGI.symbol %>% tail
```

'Fyco1' · 'Ppip5k2' · 'Tubgcp2' · 'Edf1' · 'Furin' · 'Acadl'

▼ Code

```
biomaRt::listAttributes(mmu_mart)[grep("GO",  
biomaRt::listAttributes(mmu_mart)$description), ]
```

A data.frame: 7 × 3

	name	description	page
	<chr>	<chr>	<chr>
42	go_id	GO term accession	feature_page
43	name_1006	GO term name	feature_page
44	definition_1006	GO term definition	feature_page
45	go_linkage_type	GO term evidence code	feature_page
46	namespace_1003	GO domain	feature_page
47	goslim_goa_accession	GOSlim GOA Accession(s)	feature_page
48	goslim_goa_description	GOSlim GOA Description	feature_page

▼ Code

```
# pull description of genes from biomart and merge it to data_merged  
# get the description of the genes  
  
# Create an empty dataframe to store the results  
description_df_mouse <- data.frame()  
  
# Loop through the values in batches of 10  
for (i in seq(1, length(unique(mouse_genes$MGI.symbol)), by = 10)) {
```

```

# Get the current batch of values
batch_values <- unique(mouse_genes$MGI.symbol)[i:(i+9)]
mmu_mart <- useMart("ensembl", dataset = "mmusculus_gene_ensembl")
# Query biomart for the description of the genes
results_df <- getBM(attributes = c("mgi_symbol", "description",
                                    "definition_1006",
                                    "name_1006", "namespace_1003",
                                    "phenotype_description"),
                     filters = "mgi_symbol",
                     values = batch_values,
                     mart = mmu_mart,
                     uniqueRows = TRUE)

# Append the results to the dataframe
description_df_mouse <- rbind(results_df, description_df_mouse)
}

# Print the results
head(description_df_mouse)

```

A data.frame: 6 × 6

	mgi_symbol	description	definition_1006	name_1006	namespace_1003	phenotyp
	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
1	Acadl	acyl-Coenzyme A dehydrogenase, long-chain [Source:MGI Symbol;Acc:MGI:87866]	Binding to FAD, flavin-adenine dinucleotide, the coenzyme or the prosthetic group of various flavoprotein oxidoreductase enzymes, in either the oxidized form, FAD, or the reduced form, FADH2.	flavin adenine dinucleotide binding	molecular_function	abnormal thermoge
2	Acadl	acyl-Coenzyme A dehydrogenase, long-chain [Source:MGI Symbol;Acc:MGI:87866]	Catalysis of an oxidation-reduction (redox) reaction	oxidoreductase activity, acting on the CH-CH group of in which a CH- CH group acts as a hydrogen or electron	molecular_function	abnormal thermoge

mgi_symbol	description	definition_1006	name_1006	namespace_1003	phenotyp
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
donor and reduces a hydrogen or electron acceptor.					
3	Acadl	acyl-Coenzyme A dehydrogenase, long-chain [Source:MGI Symbol;Acc:MGI:87866]	Catalysis of the reaction: acyl-CoA + oxidized [electron-transfer flavoprotein]= 2,3-dehydroacyl-CoA + reduced [electron-transfer flavoprotein] + H+.	acyl-CoA dehydrogenase activity	molecular_function abnormal thermoge
4	Acadl	acyl-Coenzyme A dehydrogenase, long-chain [Source:MGI Symbol;Acc:MGI:87866]	Catalysis of the reaction: a long-chain 2,3-saturated fatty acyl-CoA + H+ + oxidized [electron-transfer flavoprotein]= a long-chain (2E)-enoyl-CoA + reduced [electron-transfer flavoprotein].	long-chain-acyl-CoA dehydrogenase activity	molecular_function abnormal thermoge
5	Acadl	acyl-Coenzyme A dehydrogenase, long-chain [Source:MGI Symbol;Acc:MGI:87866]	The chemical reactions and pathways resulting in the breakdown of long-chain fatty acids, a fatty acid with a chain length	long-chain fatty acid catabolic process	biological_process abnormal thermoge

mgi_symbol	description	definition_1006	name_1006	namespace_1003	phenotyp
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
		between C13 and C22.			
6	Acadl	acyl-Coenzyme A dehydrogenase, long- chain [Source:MGI Symbol;Acc:MGI:87866] (redox)	Catalysis of an oxidation- reduction reaction, a reversible chemical reaction in which the oxidation state of an atom or atoms within a molecule is altered. One substrate acts as a hydrogen or electron donor and becomes oxidized, while the other acts as hydrogen or electron acceptor and becomes reduced.	oxidoreductase activity	molecular_function abnormal thermoge

▼ Code

```
mouse_genes %>% colnames()
description_df_mouse %>% colnames()
```

'EnsmbIID_Zebrafish' · 'ZFIN.symbol' · 'MGI.symbol' · 'EnsmbIID_Mouse' · 'Gene.description'
'MGI.symbol' · 'description' · 'definition_1006' · 'phenotype_description' · 'name_1006' ·
'namespace_1003'

▼ Code

```
mouse_genes_description <- merge(mouse_genes, description_df_mouse, by.x =
  "MGI.symbol", by.y = "MGI.symbol", all.x = TRUE)
head(mouse_genes_description, 1)
```

MGI.symbol	EnsmblID_Zebrafish	ZFIN.symbol	EnsmblID_Mouse	Gene.description
<chr>	<chr>	<chr>	<chr>	<chr>
1 1700009N14Rik	ENSDARG00000057026	ran	ENSMUSG00000028287	RIKEN cDNA 1700009N14 ger [Source:MGI Symbol;Acc:MGI:

MGI.symbol	EnsemblID_Zebrafish	ZFIN.symbol	EnsemblID_Mouse	Gene.description
<chr>	<chr>	<chr>	<chr>	<chr>

▼ Code

```
# collapse duplicated genes(zfin_id_symbol) to one row
mouse_genes_description <- merge(mouse_genes, description_df_mouse, by.x =
  "MGI.symbol", by.y = "MGI.symbol", all.x = TRUE)
```

```

mouse_genes_description %>% colnames()
mouse_genes_description <- mouse_genes_description %>% dplyr::select(-
  Gene.description)
mouse_genes_description <- mouse_genes_description %>%
  group_by(ZFIN.symbol,MGI.symbol) %>%
  summarise(description = paste0(unique(description), collapse = "; "),
            definition_1006 = paste0(unique(definition_1006), collapse = "; "),
            phenotype_description = paste0(unique(phenotype_description), collapse
              = "; "),
            name_1006 = paste0(unique(name_1006), collapse = "; "),
            namespace_1003 = paste0(unique(namespace_1003), collapse = "; ")

  )
mouse_genes_description %>% colnames()
# rename zfin_id_symbol to gene and definition_1006 to GO_description, description to
# gene_full_name, phenotype_description to associated_phenotype using
# dplyr::rename
mouse_genes_description <- mouse_genes_description %>%
  dplyr::rename(gene_zf = ZFIN.symbol,
                gene_mouse = MGI.symbol,
                "GO_description_mouse" = definition_1006,
                "gene_full_name_mouse" = description,
                "associated_phenotype_mouse" = phenotype_description,
                "GO_name_mouse" = name_1006,
                "GO_family_mouse" = namespace_1003)
head(mouse_genes_description,2)

```

'MGI.symbol' · 'EnsmbIID_Zebrafish' · 'ZFIN.symbol' · 'EnsmbIID_Mouse' · 'Gene.description' · 'description' · 'definition_1006' · 'phenotype_description' · 'name_1006' · 'namespace_1003'
`summarise()` has grouped output by 'ZFIN.symbol'. You can override using the
`.groups` argument.

'ZFIN.symbol' · 'MGI.symbol' · 'description' · 'definition_1006' · 'phenotype_description' · 'name_1006' · 'namespace_1003'

A grouped_df: 2 × 7

gene_zf	gene_mouse	gene_full_name_mouse	GO_description_mouse	associated_phenotype_mous
<chr>	<chr>	<chr>	<chr>	<chr>
aamdc	Aamdc	adipogenesis associated Mth938 domain containing [Source:MGI Symbol;Acc:MGI:1913523]	The synthesis of RNA from a DNA template by RNA polymerase II (RNAP II), originating at an RNA polymerase II promoter. Includes transcription of messenger RNA (mRNA) and certain small nuclear RNAs (snRNAs).; A lipid bilayer along with all the proteins and	

gene_zf	gene_mouse	gene_full_name_mouse	GO_description_mouse	associated_phenotype_mous
<chr>	<chr>	<chr>	<chr>	<chr>
			<p>protein complexes embedded in it and attached to it.; Any process that stops, prevents, or reduces the frequency, rate or extent of cell death by apoptotic process.; The contents of a cell excluding the plasma membrane and nucleus, but including other subcellular structures.; A molecular process that can be carried out by the action of a single macromolecular machine, usually via direct physical interactions with other molecular entities.</p> <p>Function in this sense denotes an action, or activity, that a gene product (or a complex) performs. These actions are described from two distinct but related perspectives: (1) biochemical activity, and (2) role as a component in a larger system/process.; Any process that activates or increases the frequency, rate or extent of transcription from an RNA polymerase II promoter.; Any process that activates or increases the frequency, rate or</p>	

gene_zf	gene_mouse	gene_full_name_mouse	GO_description_mouse	associated_phenotype_mous
<chr>	<chr>	<chr>	<chr>	<chr>
			extent of adipocyte differentiation.	
aars1	Aars	alanyl-tRNA synthetase [Source:MGI Symbol;Acc:MGI:2384560]	The process of coupling alanine to alanyl-tRNA, catalyzed by alanyl-tRNA synthetase. The alanyl-tRNA synthetase is a class-II synthetases. The activated amino acid is transferred to the 3'-OH group of an alanine accepting tRNA.; Binding to a protein.; Any process that stops, prevents, or reduces the frequency, rate or extent of cell death by apoptotic process in neurons.; Any apoptotic process in a neuron, the basic cellular unit of nervous tissue. Each neuron consists of a body, an axon, and dendrites. Their purpose is to receive, conduct, and transmit impulses in the nervous system.; Catalysis of the joining of two molecules, or two groups within a single molecule, using the energy from the hydrolysis of ATP, a similar triphosphate, or a pH gradient.; The cellular metabolic process in which a protein is formed, using the sequence of a mature mRNA or circRNA molecule to	thin ventricular wall; Purkinje cell degeneration; abnormal autophagy; tremors; embryonic lethality prior to organogenesis; decreased ventricle muscle contractility disheveled coat; focal hair loss; hair follicle degeneration hair follicle outer root sheath hyperplasia; impaired balance myocardial fiber disarray; small heart; decreased body size; decreased body weight; abnormal cardiovascular system physiology; abnormal hair shaft morphology; abnormal mitochondrial morphology; abnormal myocardial fiber morphology; abnormal myocardial fiber physiology; abnormal sarcoplasmic reticulum morphology; alopecia; ataxia; cardiac fibrosis; cardiac interstitial fibrosis; decreased cardiac output

gene_zf	gene_mouse	gene_full_name_mouse	GO_description_mouse	associated_phenotype_mous
<chr>	<chr>	<chr>	<chr>	<chr>
			<p>specify the sequence of amino acids in a polypeptide chain.</p> <p>Translation is mediated by the ribosome, and begins with the formation of a ternary complex between aminoacylated initiator methionine tRNA, GTP, and initiation factor 2, which subsequently associates with the small subunit of the ribosome and an mRNA or circRNA. Translation ends with the release of a polypeptide chain from the ribosome.;</p> <p>Catalysis of the reaction: ATP + L-alanine + tRNA(Ala) = AMP + diphosphate + L-alanyl-tRNA(Ala);</p> <p>The part of the cytoplasm that does not contain organelles but which does contain other particulate matter, such as protein complexes.;</p> <p>Binding to a metal ion.;</p> <p>Binding to an RNA molecule or a portion thereof.;</p> <p>Binding to a transfer RNA.;</p> <p>Binding to a zinc ion (Zn).;</p> <p>Catalytic activity that acts to modify a tRNA.;</p> <p>The chemical reactions and pathways involving tRNA, transfer RNA, a class of relatively small RNA molecules</p>	

gene_zf	gene_mouse	gene_full_name_mouse	GO_description_mouse	associated_phenotype_mous
<chr>	<chr>	<chr>	<chr>	<chr>
			<p>responsible for mediating the insertion of amino acids into the sequence of nascent polypeptide chains during protein synthesis. Transfer RNA is characterized by the presence of many unusual minor bases, the function of which has not been completely established.; The covalent alteration of one or more nucleotides within a tRNA molecule to produce a tRNA molecule with a sequence that differs from that coded genetically.; Any process which detects an amino-acid acetylated tRNA is charged with the correct amino acid, or removes incorrect amino acids from a charged tRNA. This process can be performed by tRNA synthases, or by subsequent reactions after tRNA aminoacylation.; A semiautonomous, self replicating organelle that occurs in varying numbers, shapes, and sizes in the cytoplasm of virtually all eukaryotic cells. It is</p>	

gene_zf	gene_mouse	gene_full_name_mouse	GO_description_mouse	associated_phenotype_mous
<chr>	<chr>	<chr>	<chr>	<chr>
			<p>notably the site of tissue respiration.; Any process that an organism uses to control its balance, the orientation of the organism (or the head of the organism) in relation to the source of gravity. In humans and animals, balance is perceived through visual cues, the labyrinth system of the inner ears and information from skin pressure receptors and muscle and joint receptors.; Binding to an amino acid, organic acids containing one or more amino substituents.; Any process pertaining to the functions of the nervous and muscular systems of an organism.; The process whose specific outcome is the progression of the cerebellar Purkinje cell layer over time, from its formation to the mature structure. The Purkinje cell layer lies just underneath the molecular layer of the cerebellar cortex. It contains the neuronal cell bodies of the Purkinje cells that are arranged side by side in a single layer.</p>	

gene_zf	gene_mouse	gene_full_name_mouse	GO_description_mouse	associated_phenotype_mous
<chr>	<chr>	<chr>	<chr>	<chr>
			<p>Candelabrum interneurons are vertically oriented between the Purkinje cells. Purkinje neurons are inhibitory and provide the output of the cerebellar cortex through axons that project into the white matter. Extensive dendritic trees from the Purkinje cells extend upward in a single plane into the molecular layer where they synapse with parallel fibers of granule cells.; The hydrolysis of an incorrectly aminoacylated tRNA.; Catalysis of the hydrolysis of misacylated Ser-tRNA(Ala).; Any process that modulates the ability of the cytoplasmic translational apparatus to interpret the genetic code.; Binding to ATP, adenosine 5'-triphosphate, a universally important coenzyme and enzyme regulator.; The contents of a cell excluding the plasma membrane and nucleus, but including other subcellular structures.; Binding to a nucleic acid.; Binding to a</p>	

gene_zf	gene_mouse	gene_full_name_mouse	GO_description_mouse	associated_phenotype_mous
<chr>	<chr>	<chr>	<chr>	<chr>
			<p>nucleotide, any compound consisting of a nucleoside that is esterified with (ortho)phosphate or an oligophosphate at any hydroxyl group on the ribose or deoxyribose.; Catalysis of the formation of aminoacyl-tRNA from ATP, amino acid, and tRNA with the release of diphosphate and AMP.; The chemical reactions and pathways by which the various amino acids become bonded to their corresponding tRNAs. The most common route for synthesis of aminoacyl tRNA is by the formation of an ester bond between the 3'-hydroxyl group of the most 3' adenosine of the tRNA and the alpha carboxylic acid group of an amino acid, usually catalyzed by the cognate aminoacyl-tRNA ligase. A given aminoacyl-tRNA ligase aminoacylates all species of an isoaccepting group of tRNA molecules.;</p>	

▼ Code

```
description_df_mouse <- merge(data_merged, mouse_genes_description, by.x = "gene",
                                by.y = "gene_zf", all.x = TRUE)
head(description_df_mouse, 1)
```

	gene	protein	AKO1	AKO2	AKO3	AWT1
	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
1	aamdc	Q502H1	1.466751474999999	-0.5739484550000002	-1.865304979	0.2672544270000

gene	protein	AKO1	AKO2	AKO3	AWT1
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>

▼ Code

```
# select up regulated genes in adult_ko_vs_adult_wt from data_merged
adult_ko_vs_adult_wt_up <- data_merged %>% filter(adult_ko_vs_adult_wt_P.Value < 0.05
    & adult_ko_vs_adult_wt_logFC >= 0)
adult_ko_vs_adult_wt_up %>% dim
adult_ko_vs_adult_wt_up %>% pull(gene) %>% unique %>% length
adult_ko_vs_adult_wt_up %>% pull(gene) %>% unique %>% head()
# select down regulated genes in adult_ko_vs_adult_wt from data_merged
adult_ko_vs_adult_wt_down <- data_merged %>% filter(adult_ko_vs_adult_wt_P.Value <
    0.05 & adult_ko_vs_adult_wt_logFC <= 0)
adult_ko_vs_adult_wt_down %>% dim
adult_ko_vs_adult_wt_down %>% pull(gene) %>% unique %>% length()
adult_ko_vs_adult_wt_down %>% pull(gene) %>% unique %>% head()
adult_ko_vs_adult_wt_all <- data_merged %>% filter(adult_ko_vs_adult_wt_P.Value <
    0.05)
adult_ko_vs_adult_wt_all %>% dim
adult_ko_vs_adult_wt_all %>% pull(gene) %>% unique %>% length()
adult_ko_vs_adult_wt_all %>% pull(gene) %>% unique %>% head()
```

173 · 21

170

'aars1' · 'abcb11a' · 'abcd2' · 'aclya' · 'aco2' · 'acta1b'

173 · 21

166

'aldh9a1a' · 'aldh9a1a.2' · 'aldoaa' · 'aldoab' · 'ank2a' · 'ankrd10b'

346 · 21

336

'aars1' · 'abcb11a' · 'abcd2' · 'aclya' · 'aco2' · 'acta1b'

▼ Code

```
# write an excel sheet with one sheet for up genes, down genes, and all genes
write.xlsx(list("adult_ko_vs_adult_wt_up"= adult_ko_vs_adult_wt_up,
"adult_ko_vs_adult_wt_down"=adult_ko_vs_adult_wt_down,
"adult_ko_vs_adult_wt_all"=adult_ko_vs_adult_wt_all

),
          file = "./LIMMA_Muscle_logfc_up_down_PA_adult_wt_adult_ko.xlsx", append =
TRUE)
```

▼ Code

```
ls()
```

```
'adult_ko_vs_adult_wt' · 'adult_ko_vs_adult_wt_all' · 'adult_ko_vs_adult_wt_all_mouse' ·
'adult_ko_vs_adult_wt_down' · 'adult_ko_vs_adult_wt_down_mouse' · 'adult_ko_vs_adult_wt_up' ·
'adult_ko_vs_adult_wt_up_mouse' · 'adult_ko_vs_juvenile_wt' · 'adult_ko_vs_juvenile_wt_all' ·
'adult_ko_vs_juvenile_wt_all_mouse' · 'adult_ko_vs_juvenile_wt_down' ·
'adult_ko_vs_juvenile_wt_down_mouse' · 'adult_ko_vs_juvenile_wt_up' ·
'adult_ko_vs_juvenile_wt_up_mouse' · 'adult_wt_vs_juvenile_wt' · 'adult_wt_vs_juvenile_wt_all' ·
'adult_wt_vs_juvenile_wt_all_mouse' · 'adult_wt_vs_juvenile_wt_down' ·
'adult_wt_vs_juvenile_wt_down_mouse' · 'adult_wt_vs_juvenile_wt_up' ·
'adult_wt_vs_juvenile_wt_up_mouse' · 'cont_matrix' · 'convertDanioGeneList_Mouse' ·
'createVennDiagram' · 'data' · 'data_merged' · 'data_selected' · 'design' · 'fit' · 'fit2' · 'GO_function' ·
'items' · 'items_df' · 'ItemsList' · 'metadata' · 'PCA' · 'pca_matrix' · 'plot' · 'venn_list'
```

▼ Code

```
# write an excel sheet with one sheet for up genes, down genes, and all genes
write.xlsx(list("adult_ko_vs_juvenile_wt_up"= adult_ko_vs_juvenile_wt_up,
"adult_ko_vs_juvenile_wt_down"=adult_ko_vs_juvenile_wt_down,
"adult_ko_vs_juvenile_wt_all"=adult_ko_vs_juvenile_wt_all

),
          file = "./LIMMA_Muscle_logfc_up_down_PA_adult_ko_vs_juvenile_wt.xlsx",
append = TRUE)
```

▼ Code

```
adult_ko_vs_adult_wt_up_mouse <-
  convertDanioGeneList_Mouse(adult_ko_vs_adult_wt_up$gene) # convert zebrafish
  gene names to mouse gene names

adult_ko_vs_adult_wt_up_mouse %>% head()
adult_ko_vs_adult_wt_down_mouse <-
  convertDanioGeneList_Mouse(adult_ko_vs_adult_wt_down$gene)
adult_ko_vs_adult_wt_down_mouse %>% head()
adult_ko_vs_adult_wt_all_mouse <-
  convertDanioGeneList_Mouse(adult_ko_vs_adult_wt_all$gene)
```

```
adult_ko_vs_adult_wt_all_mouse %>% head()
```

ERROR: Error in curl::curl_fetch_memory(url, handle = handle): Operation was aborted by an application callback

▼ Code

```
adult_ko_vs_adult_wt_up_mouse$MGI.symbol %>% head
```

'Rps14' · 'Rbm31y' · 'Myoz3' · 'Rps18-ps6' · 'Mug2' · 'BC048507'

▼ Code

```
# perform GO analysis for adult_ko_vs_adult_wt_up_mouse
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/up_pval05/Molecular_Functions",
           recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/up_pval05/Cellular_Processes",
           recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/up_pval05/Biological_Processes",
           recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/up_pval05/Reactome",
           recursive = TRUE)

GO_function(adult_ko_vs_adult_wt_up_mouse$MGI.symbol, pval = 0.05, onto= "MF",
            prefix="./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/up_pval05/Molecular_Functions",
            org="mouse")
GO_function(adult_ko_vs_adult_wt_up_mouse$MGI.symbol, pval = 0.05, onto= "CC",
            prefix="./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/up_pval05/Cellular_Processes",
            org="mouse")
GO_function(adult_ko_vs_adult_wt_up_mouse$MGI.symbol, pval = 0.05, onto= "BP",
            prefix="./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/up_pval05/Biological_Processes",
            org="mouse")
GO_function(adult_ko_vs_adult_wt_up_mouse$MGI.symbol, pval = 0.05, onto= "reactome",
            prefix="./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/up_pval05/Reactome",
            org="mouse")
```

▼ Code

```
# perform GO analysis for adult_ko_vs_adult_wt_down_mouse
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/down_pval05/Molecular_Functions",
           recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/down_pval05/Cellular_Processes",
           recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/down_pval05/Biological_Processes",
           recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/down_pval05/Reactome",
           recursive = TRUE)

GO_function(adult_ko_vs_adult_wt_down_mouse$MGI.symbol, pval = 0.05, onto= "MF",
            prefix="./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/down_pval05/Molecular_Functions",
            org="mouse")
GO_function(adult_ko_vs_adult_wt_down_mouse$MGI.symbol, pval = 0.05, onto= "CC",
            prefix="./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/down_pval05/Cellular_Processes",
            org="mouse")
GO_function(adult_ko_vs_adult_wt_down_mouse$MGI.symbol, pval = 0.05, onto= "BP",
            prefix="./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/down_pval05/Biological_Processes",
            org="mouse")
GO_function(adult_ko_vs_adult_wt_down_mouse$MGI.symbol, pval = 0.05, onto= "reactome",
            prefix="./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/down_pval05/Reactome",
            org="mouse")
```

```

GO_function(adult_ko_vs_adult_wt_down_mouse$MGI.symbol, pval = 0.05, onto= "BP",
            prefix=".~/proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/down_pval05/B
            org="mouse")
GO_function(adult_ko_vs_adult_wt_down_mouse$MGI.symbol, pval = 0.05, onto= "reactome"
            prefix=".~/proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/down_pval05/R
            org="mouse")

```

▼ Code

```

# perform GO analysis for adult_ko_vs_adult_wt_all_mouse
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/all_pval05/Molecul
            recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/all_pval05/Cellula
            recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/all_pval05/Biolog
            recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/all_pval05/Reactor
            recursive = TRUE)

GO_function(adult_ko_vs_adult_wt_all_mouse$MGI.symbol, pval = 0.05, onto= "MF",
            prefix=".~/proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/all_pval05/Mo
            org="mouse")
GO_function(adult_ko_vs_adult_wt_all_mouse$MGI.symbol, pval = 0.05, onto= "CC",
            prefix=".~/proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/all_pval05/Ce
            org="mouse")
GO_function(adult_ko_vs_adult_wt_all_mouse$MGI.symbol, pval = 0.05, onto= "BP",
            prefix=".~/proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/all_pval05/Bi
            org="mouse")
GO_function(adult_ko_vs_adult_wt_all_mouse$MGI.symbol, pval = 0.05, onto= "reactome",
            prefix=".~/proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_adult_wt/all_pval05/Re
            org="mouse")

```

▼ Code

```

# select up regulated genes in adult_ko_vs_juvenile_wt from data_merged
adult_ko_vs_juvenile_wt_up <- data_merged %>% filter(adult_ko_vs_juvenile_wt_P.Value
              < 0.05 & adult_ko_vs_juvenile_wt_logFC >= 0)
adult_ko_vs_juvenile_wt_up %>% dim
adult_ko_vs_juvenile_wt_up %>% pull(gene) %>% unique %>% length
adult_ko_vs_juvenile_wt_up %>% pull(gene) %>% unique %>% head()
# select down regulated genes in adult_ko_vs_juvenile_wt from data_merged
adult_ko_vs_juvenile_wt_down <- data_merged %>%
  filter(adult_ko_vs_juvenile_wt_P.Value < 0.05 &
         adult_ko_vs_juvenile_wt_logFC <= 0)
adult_ko_vs_juvenile_wt_down %>% dim
adult_ko_vs_juvenile_wt_down %>% pull(gene) %>% unique %>% length()
adult_ko_vs_juvenile_wt_down %>% pull(gene) %>% unique %>% head()
adult_ko_vs_juvenile_wt_all <- data_merged %>% filter(adult_ko_vs_juvenile_wt_P.Value
              < 0.05)
adult_ko_vs_juvenile_wt_all %>% dim
adult_ko_vs_juvenile_wt_all %>% pull(gene) %>% unique %>% length()
adult_ko_vs_juvenile_wt_all %>% pull(gene) %>% unique %>% head()

```

487 · 21

478

'abcd2' · 'acaa2' · 'acads' · 'acat1' · 'acbd7' · 'ache'

452 · 21

438

'aars1' · 'aclya' · 'adh5' · 'adh8b' · 'adrm1' · 'afp4'

939 · 21

914

'aars1' · 'abcd2' · 'acaa2' · 'acads' · 'acat1' · 'acbd7'

▼ Code

```
# write an excel sheet with one sheet for up genes, down genes, and all genes
adult_ko_vs_juvenile_wt_up_mouse <-
  convertDanioGeneList_Mouse(adult_ko_vs_juvenile_wt_up$gene)

adult_ko_vs_juvenile_wt_up_mouse %>% head()
adult_ko_vs_juvenile_wt_down_mouse <-
  convertDanioGeneList_Mouse(adult_ko_vs_juvenile_wt_down$gene)
adult_ko_vs_juvenile_wt_down_mouse %>% head()
adult_ko_vs_juvenile_wt_all_mouse <-
  convertDanioGeneList_Mouse(adult_ko_vs_juvenile_wt_all$gene)
adult_ko_vs_juvenile_wt_all_mouse %>% head()
```

A data.frame: 6 × 5

	EnsmbIID_Zebrafish	ZFIN.symbol	MGI.symbol	EnsmbIID_Mouse	Gene.description
	<chr>	<chr>	<chr>	<chr>	<chr>
1	ENSDARG00000063924	mt-cyb	mt-Cytb	ENSMUSG00000064370	mitochondrially encoded cytochrome b [Source:Symbol;Acc:MGI:1024]
2	ENSDARG00000040712	adprh	Adprh	ENSMUSG00000002844	ADP-ribosylarginine hydrolase [Source:MCGSymbol;Acc:MGI:1098]
3	ENSDARG00000014230	dlst	Dlst	ENSMUSG00000004789	dihydrolipoamide S-succinyltransferase (E component of 2-oxo-glutarate complex) [Source:MGI Symbol;Acc:MGI:1926]
4	ENSDARG00000054191	pgk1	Pgk1	ENSMUSG00000062070	phosphoglycerate kinase [Source:MGI Symbol;Acc:MGI:9751]
5	ENSDARG00000097819	znf576.1	Zfp92	ENSMUSG00000031374	zinc finger protein 92 [Source:MGI Symbol;Acc:MGI:1080]

EnsemblID_Zebrafish	ZFIN.symbol	MGI.symbol	EnsemblID_Mouse	Gene.description
<chr>	<chr>	<chr>	<chr>	<chr>
6 ENSDARG00000011665	aldoaa	Aldoart2	ENSMUSG0000063129	aldolase 1 A, retrogen [Source:MGI Symbol;Acc:MGI:193]

A data.frame: 6 × 5

EnsemblID_Zebrafish	ZFIN.symbol	MGI.symbol	EnsemblID_Mouse	Gene.description
<chr>	<chr>	<chr>	<chr>	<chr>
1 ENSDARG0000003429	hnrnndl	Rbm31y	ENSMUSG0000095365	RNA binding motif 31 linked [Source:MGI Symbol;Acc:MGI:192]
2 ENSDARG00000044299	lmbn1	Lmnb1	ENSMUSG0000024590	lamin B1 [Source:MGI Symbol;Acc:MGI:967]
3 ENSDARG00000036675	hnrnpa1b	Hnrnpa1	ENSMUSG0000046434	heterogeneous nuclear ribonucleoprotein A1 [Source:MGI Symbol;Acc:MGI:1048]
4 ENSDARG00000056334	mlh3	Mlh3	ENSMUSG0000021245	mutL homolog 3 [Source:MGI Symbol;Acc:MGI:135]
5 ENSDARG00000036629	rps14	Rps14	ENSMUSG0000024608	ribosomal protein S14 [Source:MGI Symbol;Acc:MGI:9810]
6 ENSDARG0000008292	eif2s3	Eif2s3y	ENSMUSG0000069049	eukaryotic translation initiation factor 2, sub 3, structural gene Y-lir [Source:MGI Symbol;Acc:MGI:1349]

A data.frame: 6 × 5

EnsemblID_Zebrafish	ZFIN.symbol	MGI.symbol	EnsemblID_Mouse	Gene.description
<chr>	<chr>	<chr>	<chr>	<chr>
1 ENSDARG00000063924	mt-cyb	mt-Cytb	ENSMUSG0000064370	mitochondrially encoded cytochrome b [Source: Symbol;Acc:MGI:1029]
2 ENSDARG00000011665	aldoaa	Aldoart2	ENSMUSG0000063129	aldolase 1 A, retrogen [Source:MGI Symbol;Acc:MGI:193]

EnsemblID_Zebrafish	ZFIN.symbol	MGI.symbol	EnsemblID_Mouse	Gene.description
<chr>	<chr>	<chr>	<chr>	<chr>
3 ENSDARG00000003429	hnrnndl	Rbm31y	ENSMUSG00000095365	RNA binding motif 31 linked [Source:MGI Symbol;Acc:MGI:192]
4 ENSDARG00000056334	mlh3	Mlh3	ENSMUSG00000021245	mutL homolog 3 [Source:MGI Symbol;Acc:MGI:135]
5 ENSDARG00000014230	dlst	Dlst	ENSMUSG00000004789	dihydrolipoamide S-succinyltransferase (E component of 2-oxo-glutarate complex) [Source:MGI Symbol;Acc:MGI:192]
6 ENSDARG00000044299	lmbn1	Lmnb1	ENSMUSG00000024590	lamin B1 [Source:MGI Symbol;Acc:MGI:967]

▼ Code

```
# perform GO analysis for adult_ko_vs_juvenile_wt_up_mouse
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_juvenile_wt/up_pval05/Molecular_Function")
  recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_juvenile_wt/up_pval05/Cellular_Component")
  recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_juvenile_wt/up_pval05/Biological_Process")
  recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_juvenile_wt/up_pval05/Reactome")
  recursive = TRUE)

GO_function(adult_ko_vs_juvenile_wt_up_mouse$MGI.symbol, pval = 0.05, onto= "MF",
            prefix="./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_juvenile_wt/up_pval05/",
            org="mouse")
GO_function(adult_ko_vs_juvenile_wt_up_mouse$MGI.symbol, pval = 0.05, onto= "CC",
            prefix="./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_juvenile_wt/up_pval05/",
            org="mouse")
GO_function(adult_ko_vs_juvenile_wt_up_mouse$MGI.symbol, pval = 0.05, onto= "BP",
            prefix="./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_juvenile_wt/up_pval05/",
            org="mouse")
GO_function(adult_ko_vs_juvenile_wt_up_mouse$MGI.symbol, pval = 0.05, onto= "reactome",
            prefix="./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_juvenile_wt/up_pval05/",
            org="mouse")

# perform GO analysis for adult_ko_vs_juvenile_wt_down_mouse
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_juvenile_wt/down_pval05/Molecular_Function")
  recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_juvenile_wt/down_pval05/Cellular_Component")
  recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_juvenile_wt/down_pval05/Biological_Process")
  recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_juvenile_wt/down_pval05/Reactome")
  recursive = TRUE)
```

```

GO_function(adult_ko_vs_juvenile_wt_down_mouse$MGI.symbol, pval = 0.05, onto= "MF",
            prefix=".~/proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_juvenile_wt/down_pval0"
            org="mouse")
GO_function(adult_ko_vs_juvenile_wt_down_mouse$MGI.symbol, pval = 0.05, onto= "CC",
            prefix=".~/proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_juvenile_wt/down_pval0"
            org="mouse")
GO_function(adult_ko_vs_juvenile_wt_down_mouse$MGI.symbol, pval = 0.05, onto= "BP",
            prefix=".~/proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_juvenile_wt/down_pval0"
            org="mouse")
GO_function(adult_ko_vs_juvenile_wt_down_mouse$MGI.symbol, pval = 0.05, onto= "reactome",
            prefix=".~/proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_juvenile_wt/down_pval0"
            org="mouse")

# perform GO analysis for adult_ko_vs_juvenile_wt_all_mouse
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_juvenile_wt/all_pval05/Molecular_Functions",
           recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_juvenile_wt/all_pval05/Cellosaurus",
           recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_juvenile_wt/all_pval05/BioProcesses",
           recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_juvenile_wt/all_pval05/Reactomes",
           recursive = TRUE)

GO_function(adult_ko_vs_juvenile_wt_all_mouse$MGI.symbol, pval = 0.05, onto= "MF",
            prefix=".~/proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_juvenile_wt/all_pval05"
            org="mouse")
GO_function(adult_ko_vs_juvenile_wt_all_mouse$MGI.symbol, pval = 0.05, onto= "CC",
            prefix=".~/proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_juvenile_wt/all_pval05"
            org="mouse")
GO_function(adult_ko_vs_juvenile_wt_all_mouse$MGI.symbol, pval = 0.05, onto= "BP",
            prefix=".~/proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_juvenile_wt/all_pval05"
            org="mouse")
GO_function(adult_ko_vs_juvenile_wt_all_mouse$MGI.symbol, pval = 0.05, onto= "reactome",
            prefix=".~/proteomics_n_3_muscle/GO_Analysis/adult_ko_vs_juvenile_wt/all_pval05"
            org="mouse")

```

▼ Code

```

# select up regulated genes in adult_wt_vs_juvenile_wt from data_merged
adult_wt_vs_juvenile_wt_up <- data_merged %>% filter(adult_wt_vs_juvenile_wt_P.Value
              < 0.05 & adult_wt_vs_juvenile_wt_logFC >= 0)
adult_wt_vs_juvenile_wt_up %>% dim
adult_wt_vs_juvenile_wt_up %>% pull(gene) %>% unique %>% length
adult_wt_vs_juvenile_wt_up %>% pull(gene) %>% unique %>% head()
# select down regulated genes in adult_wt_vs_juvenile_wt from data_merged
adult_wt_vs_juvenile_wt_down <- data_merged %>%
  filter(adult_wt_vs_juvenile_wt_P.Value < 0.05 &
         adult_wt_vs_juvenile_wt_logFC <= 0)
adult_wt_vs_juvenile_wt_down %>% dim
adult_wt_vs_juvenile_wt_down %>% pull(gene) %>% unique %>% length()
adult_wt_vs_juvenile_wt_down %>% pull(gene) %>% unique %>% head()
adult_wt_vs_juvenile_wt_all <- data_merged %>% filter(adult_wt_vs_juvenile_wt_P.Value
              < 0.05)

```

```
adult_wt_vs_juvenile_wt_all %>% dim
adult_wt_vs_juvenile_wt_all %>% pull(gene) %>% unique %>% length()
adult_wt_vs_juvenile_wt_all %>% pull(gene) %>% unique %>% head()
```

569 · 21

559

'aamdc' · 'acaa2' · 'acads' · 'acadvl' · 'acat1' · 'acbd7'

493 · 21

479

'aars1' · 'abcg2d' · 'acadsb' · 'acly'a' · 'actn3a' · 'adgrv1'

1062 · 21

1035

'aamdc' · 'aars1' · 'abcg2d' · 'acaa2' · 'acads' · 'acadsb'

▼ Code

```
adult_wt_vs_juvenile_wt_up_mouse <-
  convertDanioGeneList_Mouse(adult_wt_vs_juvenile_wt_up$gene)

adult_wt_vs_juvenile_wt_up_mouse %>% head()
adult_wt_vs_juvenile_wt_down_mouse <-
  convertDanioGeneList_Mouse(adult_wt_vs_juvenile_wt_down$gene)
adult_wt_vs_juvenile_wt_down_mouse %>% head()
adult_wt_vs_juvenile_wt_all_mouse <-
  convertDanioGeneList_Mouse(adult_wt_vs_juvenile_wt_all$gene)
adult_wt_vs_juvenile_wt_all_mouse %>% head()
```

A data.frame: 6 × 5

	EnsmbIID_Zebrafish	ZFIN.symbol	MGI.symbol	EnsmbIID_Mouse	Gene.description
	<chr>	<chr>	<chr>	<chr>	<chr>
1	ENSDARG00000063908	mt-co2	mt-Co2	ENSMUSG00000064354	mitochondrially encoded cytochrome c oxidase [Source:MGI Symbol;Acc:MGI:102!]
2	ENSDARG00000063924	mt-cyb	mt-Cytb	ENSMUSG00000064370	mitochondrially encoded cytochrome b [Source: MGI Symbol;Acc:MGI:102!]
3	ENSDARG00000054191	pgk1	Pgk1	ENSMUSG00000062070	phosphoglycerate kinase [Source:MGI Symbol;Acc:MGI:975!]
4	ENSDARG00000043848	sod1	Sod1	ENSMUSG00000022982	superoxide dismutase soluble [Source:MGI Symbol;Acc:MGI:983!]
5	ENSDARG00000060069	isoc1	Isoc1	ENSMUSG00000024601	isochorismatase domain containing 1 [Source:MGI Symbol;Acc:MGI:191!]

EnsemblID_Zebrafish	ZFIN.symbol	MGI.symbol	EnsemblID_Mouse	Gene.description
<chr>	<chr>	<chr>	<chr>	<chr>
6 ENSDARG00000040712	adprh	Adprh	ENSMUSG0000002844	ADP-ribosylarginine hydrolase [Source:MGI Symbol;Acc:MGI:1098]

A data.frame: 6 × 5

EnsemblID_Zebrafish	ZFIN.symbol	MGI.symbol	EnsemblID_Mouse	Gene.description
<chr>	<chr>	<chr>	<chr>	<chr>
1 ENSDARG0000003429	hnrnndl	Rbm31y	ENSMUSG00000095365	RNA binding motif 31 linked [Source:MGI Symbol;Acc:MGI:1921]
2 ENSDARG00000019398	psma6a	Psma6	ENSMUSG00000021024	proteasome subunit a 6 [Source:MGI Symbol;Acc:MGI:1341]
3 ENSDARG00000013755	actn3a	Actn3	ENSMUSG00000006457	actinin alpha 3 [Source:MGI Symbol;Acc:MGI:9961]
4 ENSDARG00000036629	rps14	Rps14	ENSMUSG00000024608	ribosomal protein S14 [Source:MGI Symbol;Acc:MGI:9810]
5 ENSDARG00000019332	ndufb4	Ndufb4	ENSMUSG00000022820	NADH:ubiquinone oxidoreductase subunit [Source:MGI Symbol;Acc:MGI:1919]
6 ENSDARG00000036675	hnrrnpa1b	Hnrnpa1	ENSMUSG00000046434	heterogeneous nuclear ribonucleoprotein A1 [Source:MGI Symbol;Acc:MGI:1048]



A data.frame: 6 × 5

EnsemblID_Zebrafish	ZFIN.symbol	MGI.symbol	EnsemblID_Mouse	Gene.description
<chr>	<chr>	<chr>	<chr>	<chr>
1 ENSDARG00000063924	mt-cyb	mt-Cytb	ENSMUSG00000064370	mitochondrially encoded cytochrome b [Source:MGI Symbol;Acc:MGI:1029]
2 ENSDARG00000063908	mt-co2	mt-Co2	ENSMUSG00000064354	mitochondrially encoded cytochrome c oxidase

EnsemblID_Zebrafish	ZFIN.symbol	MGI.symbol	EnsemblID_Mouse	Gene.description
<chr>	<chr>	<chr>	<chr>	<chr>
				[Source:MGI Symbol;Acc:MGI:102]
3 ENSDARG00000034470	aldoab	Aldoart2	ENSMUSG00000063129	aldolase 1 A, retrogen [Source:MGI Symbol;Acc:MGI:193]
4 ENSDARG00000103917	znf185	Zfp185	ENSMUSG00000031351	zinc finger protein 185 [Source:MGI Symbol;Acc:MGI:108]
5 ENSDARG00000014106	cfl2	Cfl2	ENSMUSG00000062929	cofilin 2, muscle [Source:MGI Symbol;Acc:MGI:101]
6 ENSDARG00000001431	actn3b	Actn3	ENSMUSG00000006457	actinin alpha 3 [Source:MGI Symbol;Acc:MGI:996]

▼ Code

```
# perform GO analysis for adult_wt_vs_juvenile_wt_up_mouse
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_wt_vs_juvenile_wt/up_pval05/Molecular_Function")
  recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_wt_vs_juvenile_wt/up_pval05/Cellular_Process")
  recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_wt_vs_juvenile_wt/up_pval05/Biological_Process")
  recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_wt_vs_juvenile_wt/up_pval05/Reactome")
  recursive = TRUE)

GO_function(adult_wt_vs_juvenile_wt_up_mouse$MGI.symbol, pval = 0.05, onto= "MF",
            prefix="./proteomics_n_3_muscle/GO_Analysis/adult_wt_vs_juvenile_wt/up_pval05/",
            org="mouse")
GO_function(adult_wt_vs_juvenile_wt_up_mouse$MGI.symbol, pval = 0.05, onto= "CC",
            prefix="./proteomics_n_3_muscle/GO_Analysis/adult_wt_vs_juvenile_wt/up_pval05/",
            org="mouse")
GO_function(adult_wt_vs_juvenile_wt_up_mouse$MGI.symbol, pval = 0.05, onto= "BP",
            prefix="./proteomics_n_3_muscle/GO_Analysis/adult_wt_vs_juvenile_wt/up_pval05/",
            org="mouse")
GO_function(adult_wt_vs_juvenile_wt_up_mouse$MGI.symbol, pval = 0.05, onto= "reactome",
            prefix="./proteomics_n_3_muscle/GO_Analysis/adult_wt_vs_juvenile_wt/up_pval05/",
            org="mouse")

# perform GO analysis for adult_wt_vs_juvenile_wt_down_mouse
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_wt_vs_juvenile_wt/down_pval05/Molecular_Function")
  recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_wt_vs_juvenile_wt/down_pval05/Cellular_Process")
  recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_wt_vs_juvenile_wt/down_pval05/Biological_Process")
  recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_wt_vs_juvenile_wt/down_pval05/Reactome")
  recursive = TRUE)
```

```

GO_function(adult_wt_vs_juvenile_wt_down_mouse$MGI.symbol, pval = 0.05, onto= "MF",
            prefix=".~/proteomics_n_3_muscle/GO_Analysis/adult_wt_vs_juvenile_wt/down_pval05"
            org="mouse")
GO_function(adult_wt_vs_juvenile_wt_down_mouse$MGI.symbol, pval = 0.05, onto= "CC",
            prefix=".~/proteomics_n_3_muscle/GO_Analysis/adult_wt_vs_juvenile_wt/down_pval05"
            org="mouse")
GO_function(adult_wt_vs_juvenile_wt_down_mouse$MGI.symbol, pval = 0.05, onto= "BP",
            prefix=".~/proteomics_n_3_muscle/GO_Analysis/adult_wt_vs_juvenile_wt/down_pval05"
            org="mouse")
GO_function(adult_wt_vs_juvenile_wt_down_mouse$MGI.symbol, pval = 0.05, onto= "reactome",
            prefix=".~/proteomics_n_3_muscle/GO_Analysis/adult_wt_vs_juvenile_wt/down_pval05"
            org="mouse")

# perform GO analysis for adult_wt_vs_juvenile_wt_all_mouse
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_wt_vs_juvenile_wt/all_pval05/Molecular_Function"
           recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_wt_vs_juvenile_wt/all_pval05/Cellular_Process"
           recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_wt_vs_juvenile_wt/all_pval05/Biological_Process"
           recursive = TRUE)
dir.create("./proteomics_n_3_muscle/GO_Analysis/adult_wt_vs_juvenile_wt/all_pval05/Reactome"
           recursive = TRUE)

GO_function(adult_wt_vko_wt_pval05_mouse
ko_wt_pval05_mouse
ko_wt_pval05_mouse
ko_wt_pval05_mouses_juvenile_wt_all_mouse$MGI.symbol, pval = 0.05, onto= "MF",
            prefix=".~/proteomics_n_3_muscle/GO_Analysis/adult_wt_vs_juvenile_wt/all_pval05"
            org="mouse")
GO_function(adult_wt_vs_juvenile_wt_all_mouse$MGI.symbol, pval = 0.05, onto= "CC",
            prefix=".~/proteomics_n_3_muscle/GO_Analysis/adult_wt_vs_juvenile_wt/all_pval05"
            org="mouse")
GO_function(adult_wt_vs_juvenile_wt_all_mouse$MGI.symbol, pval = 0.05, onto= "BP",
            prefix=".~/proteomics_n_3_muscle/GO_Analysis/adult_wt_vs_juvenile_wt/all_pval05"
            org="mouse")
GO_function(adult_wt_vs_juvenile_wt_all_mouse$MGI.symbol, pval = 0.05, onto= "reactome",
            prefix=".~/proteomics_n_3_muscle/GO_Analysis/adult_wt_vs_juvenile_wt/all_pval05"
            org="mouse")

```

▼ Code

```

# make bigger graph
options(repr.plot.width=18, repr.plot.height=10)
# make venn diagram between the significant genes in the three comparisons
venn_list <- list( wt_v_juv=adult_wt_vs_juvenile_wt_all$gene,
                    ko_v_juv=adult_ko_vs_juvenile_wt_all$gene,
                    ko_v_wt= adult_ko_vs_adult_wt_all$gene
                  )

plot <- ggVennDiagram(venn_list,
                      label_size = NA,
                      # set_size=8

```

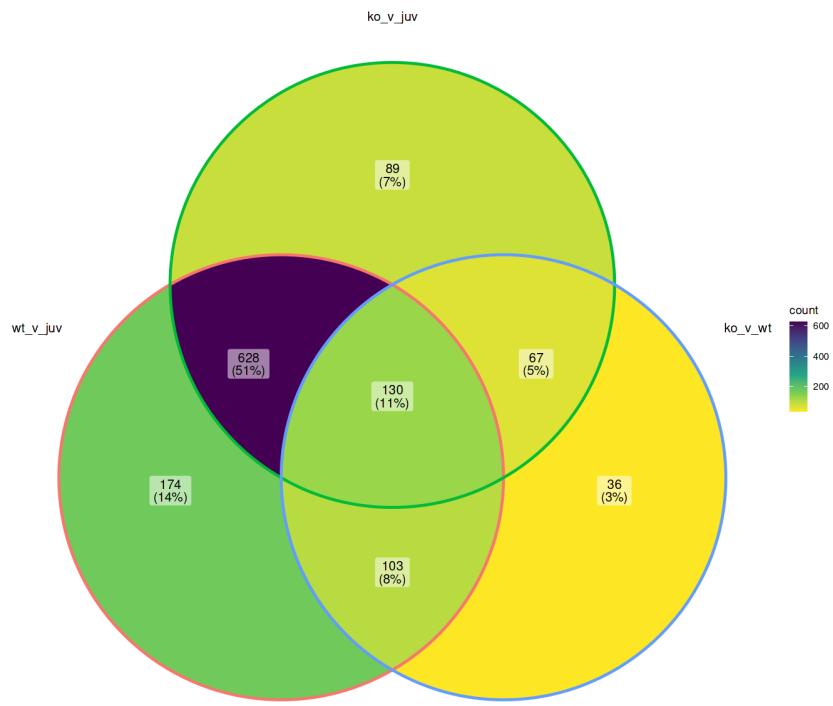
```

) +
  scale_fill_viridis_c(option = "viridis", direction = -1) +
  theme(plot.title = element_text(hjust = 0.5),
    )
NULL

plot

```

NULL



▼ Code

```

# get the genes in the intersection of the three comparisons
ItemsList <- gplots::venn(venn_list, show.plot = FALSE)

items <- attributes(ItemsList)$intersections # get the items in the intersection
items_df <- tibble::enframe(items) # convert the items to a dataframe
items_df %>% head
items_df$value <- gsub("^\\"\\(\\"|\\"\\\$)", "", items_df$value) # remove the c() and ""
# from the items
items_df$value <- gsub('["\\"]', '', items_df$value) # remove the " and ) from the
# items

items_df <- separate_wider_delim(items_df, delim = ",",
  too_few = "align_start") # separate the items into columns
items_df <- items_df %>% t %>% as.data.frame # transpose the dataframe

colnames(items_df) <- items_df[1,] # set the first row as column names

items_df <- items_df[-1,] # remove the first row
rownames(items_df) <- NULL # remove the row names
items_df <- items_df %>% mutate_all(str_trim) # remove white spaces from all columns
items_df %>% head

```

A tibble: 6 × 2

name	value
<chr>	<list>
ko_v_juv:ko_v_wt	abcd2, a....
wt_v_juv:ko_v_wt	adgrv1,
wt_v_juv:ko_v_juv	acaa2, a....
wt_v_juv:ko_v_juv:ko_v_wt	aars1, a....
ko_v_wt	abcb11a,....
ko_v_juv	ache, ac....

A data.frame: 6 × 7

	ko_v_juv:ko_v_wt	wt_v_juv:ko_v_wt	wt_v_juv:ko_v_juv	wt_v_juv:ko_v_juv:ko_v_wt	ko_v_wt	ko_v_juv
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
1	abcd2	adgrv1	acaa2	aars1	abcb11a	ache
2	acta1b	aldh9a1a	acads	aclya	actbb	actc1a
3	anxa2a	aldh9a1a.2	acat1	aco2	adh8a	adrm1
4	apobec2a	apom	acbd7	adh8b	ahnak	anxa1a
5	bcar3	arhgap4a	acot13	aimp1a	aldh4a1	atp1b1
6	btr05	atp2a1l	actn2b	aldoaa	ankrd52a	atp5if1



4.2 GSEA

▼ Code

```
GSEA_function <- function(df, pval_deg = 0.05, pval_enrich=0.2, onto= "MF",
prefix=""){

  gene_list_df <- df[df$adult_ko_vs_adult_wt_P.Value<=pval_deg,]
  gene_list_df <- gene_list_df %>% arrange(desc(adult_ko_vs_adult_wt_logFC))
  gene_list_df <- gene_list_df[!is.na(gene_list_df$gene_mouse),]
  gene_list_df <- gene_list_df[!duplicated(gene_list_df$gene_mouse),]
  gene_list <- gene_list_df %>% pull(adult_ko_vs_adult_wt_logFC)
  names(gene_list) <- gene_list_df %>% pull(gene_mouse)

  gene_list <- gene_list[!duplicated(gene_list)]
  print(head(gene_list))}
```

```

compGO <- gseGO(gene = gene_list, pvalueCutoff = pval_enrich,,keyType =
  "SYMBOL",
                        pAdjustMethod = "BH",OrgDb = "org.Mm.eg.db", ont = onto)

# print(compGO)
if(is.null(compGO)|nrow(compGO@result)==0){
  message(paste0("No GSEA:",onto))
  message(paste0("*****"))
  message(paste0("\n"))

} else {

  compGO_df <- as.data.frame(compGO)
  compGO_df <- compGO_df %>% tidyverse::separate_rows(core_enrichment, sep = "/",
convert = FALSE) %>% arrange((p.adjust))

  if(nrow(compGO_df)==0){
    message(paste0("No GSEA:",onto))

    message(paste0("*****"))
    message(paste0("\n"))

  } else{

    write.csv(compGO_df, paste0(prefix,"_GSEA_",onto, "_pathways.csv"))

    full_name= switch(onto,
      MF= "Molecular Function",
      CC= "Cellular Components",
      BP= "Biological Processes"
      )
    print(paste(full_name, onto))
    print(dotplot(compGO, showCategory = 15, title = paste0("DEG GO Pathway
Enrichment Analysis \n",full_name ),
      font.size = 12) + facet_grid(.~.sign))
    dev.copy(
      pdf,
      file = paste0(prefix,"_GSEA_",onto, "_pathways.pdf"),
      width = 10,
      height = 8
      )
    dev.off ()

    message(paste0("DEG Pathway analysis GO:",onto, " done"))
    message(paste0("*****"))
    message(paste0("\n"))

  }
}

}

```

▼ Code

```
getwd()
```

```
'/mnt/Data_8TB/Carolina_data/Cell_paper'
```

▼ Code

```
dir.create("proteomics_n_3_muscle/GSEA_subcluster", showWarnings = FALSE)
dir.create("proteomics_n_3_muscle/GSEA_subcluster/Molecular_function", showWarnings =
  FALSE, recursive = TRUE)
dir.create("proteomics_n_3_muscle/GSEA_subcluster/Cellular_component", showWarnings =
  FALSE, recursive = TRUE)
dir.create("proteomics_n_3_muscle/GSEA_subcluster/Biological_processes", showWarnings =
  FALSE, recursive = TRUE)

GSEA_function(df = description_df_mouse_zf, onto = "MF",
  prefix="proteomics_n_3_muscle/GSEA_subcluster/Molecular_function/muscle_n_3_GS"
GSEA_function(df = description_df_mouse_zf, onto = "CC",
  prefix="proteomics_n_3_muscle/GSEA_subcluster/Cellular_component/muscle_n_3_GS"
GSEA_function(df = description_df_mouse_zf, onto = "BP",
  prefix="proteomics_n_3_muscle/GSEA_subcluster/Biological_processes/muscle_n_3_
```

```
Col1a1  Arhgap4   Col6a3   Adgrv1     Decr1      Dnm2
4.609072 4.524977 4.040525 3.784338 3.735197 3.418267
```

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

```
GSEA analysis...
```

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

```
done...
```

```
[1] "Moleuclar Function MF"
```

```
DEG Pathway analysis GO:MF done
```

```
*****
**
```

```
Col1a1  Arhgap4   Col6a3   Adgrv1     Decr1      Dnm2
4.609072 4.524977 4.040525 3.784338 3.735197 3.418267
```

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

```
GSEA analysis...
```

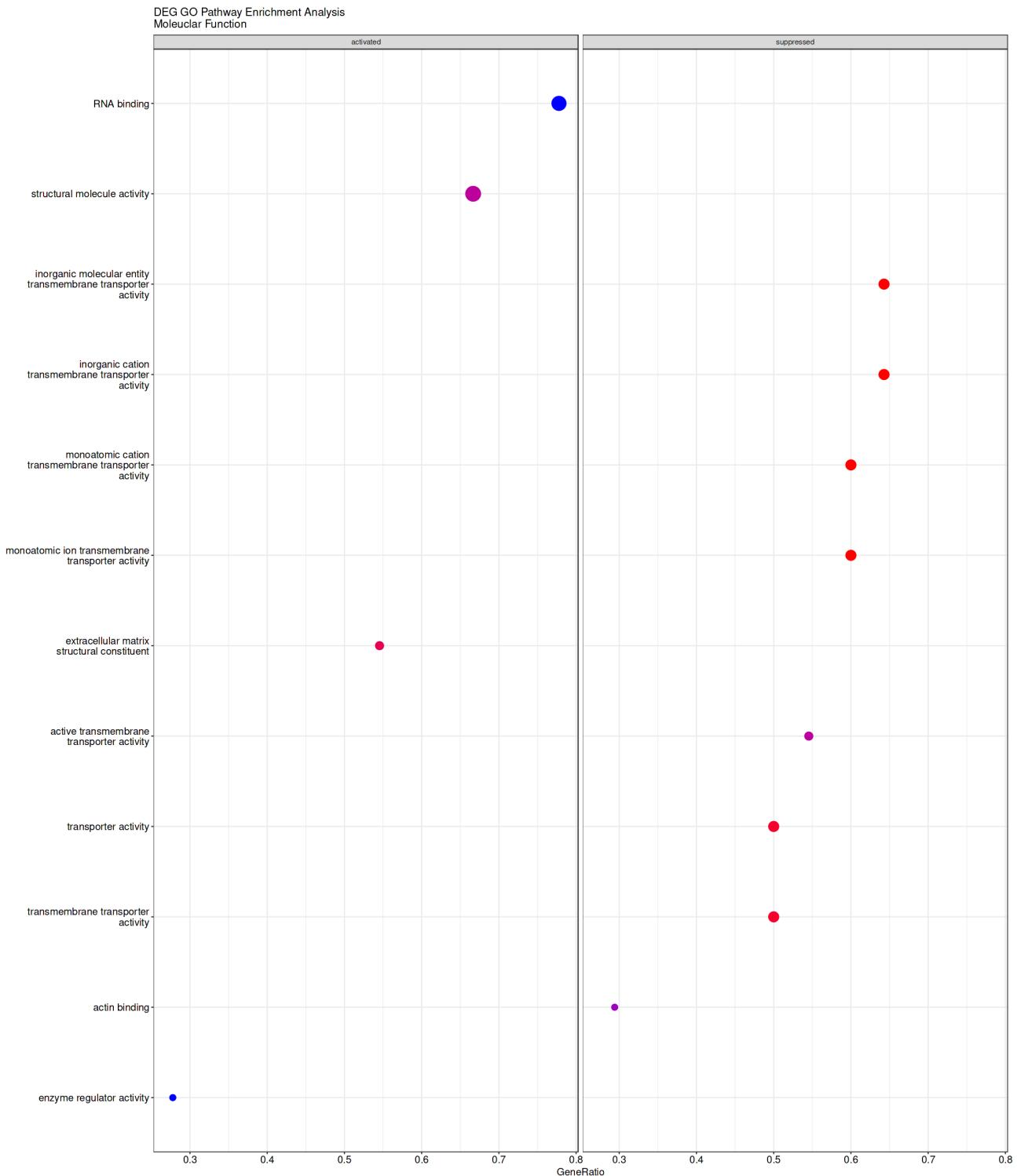
```
leading edge analysis...
```

done...

[1] "Cellular Components CC"

DEG Pathway analysis GO:CC done

**



Col1a1 Argap4 Col6a3 Adgrv1 Decr1 Dnm2
4.609072 4.524977 4.040525 3.784338 3.735197 3.418267

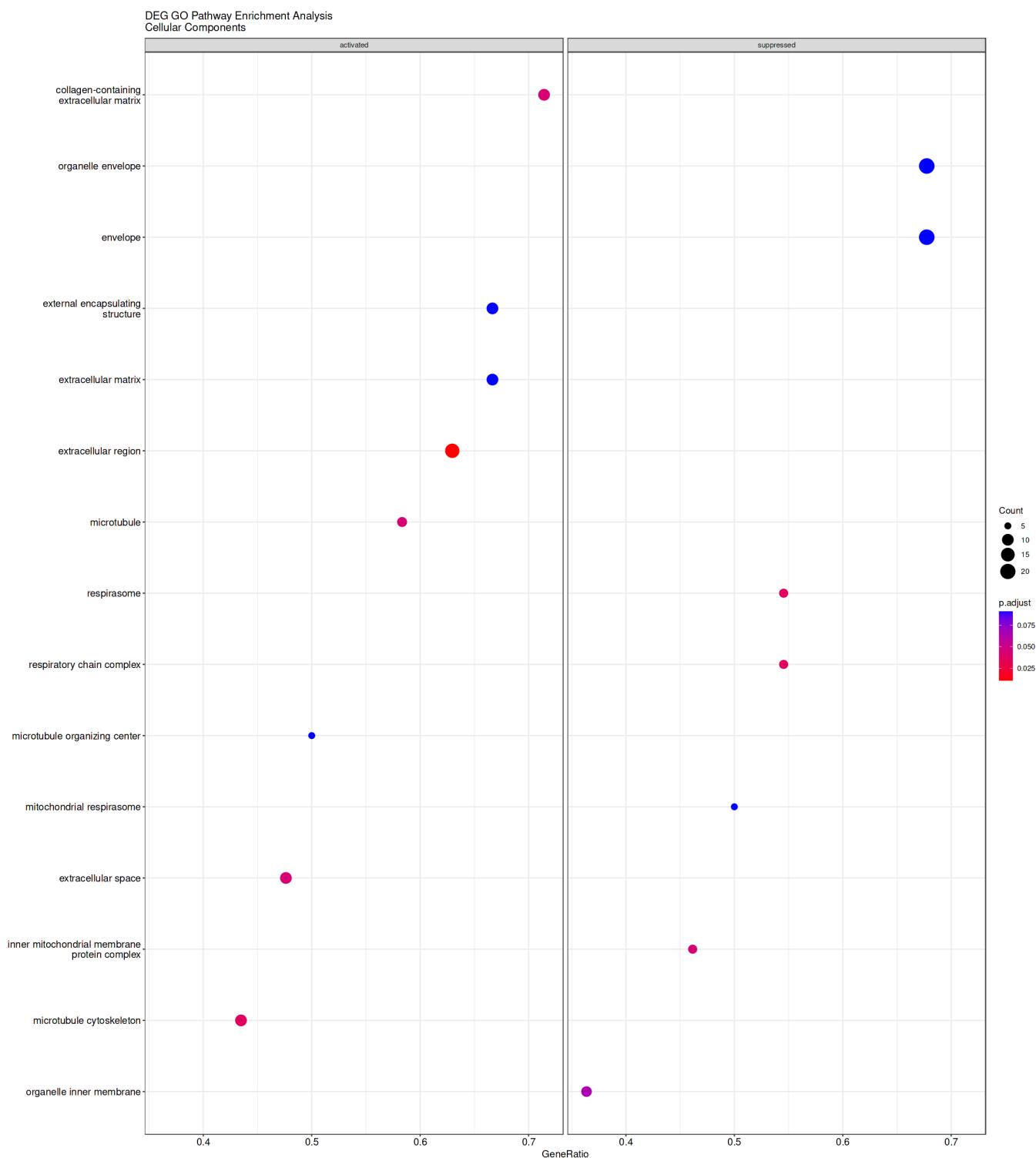
preparing geneSet collections...

GSEA analysis...

no term enriched under specific pvalueCutoff...

No GSEA:BP

* *



▼ Code

```
mitocarta_mouse_pathways <- readxl::read_excel("Mouse.MitoCarta3.0.xls", sheet = 4)

mitocarta_mouse_pathways <- mitocarta_mouse_pathways %>% dplyr::select(MitoPathway,
    Genes)
# mitocarta_mouse_pathways %>% head
mitocarta_mouse_pathways_rows_list <- mitocarta_mouse_pathways %>%
    tidyverse::separate_rows(Genes, sep = ",", convert = FALSE) %>%
    arrange(MitoPathway) %>% dplyr::rename(term= MitoPathway, gene= Genes)
# mitocarta_mouse_pathways_rows_list <- mitocarta_mouse_pathways_rows_list
# remove trailing and leading spaces
mitocarta_mouse_pathways_rows_list$gene <- gsub("^\\s+|\\s+$", "", mitocarta_mouse_pathways_rows_list$gene)
mitocarta_mouse_pathways_rows_list %>% head
```

A tibble: 6 × 2

term	gene
<chr>	<chr>
ABC transporters	Abca9
ABC transporters	Abcb10
ABC transporters	Abcb6
ABC transporters	Abcb7
ABC transporters	Abcb8
ABC transporters	Abcd1

▼ Code

```
mitocarta_pathways <- read.delim("./MitoPathways3.0.gmx", sep = "\t", header = TRUE)
mitocarta_pathways %>% head
mitocarta_pathways %>% dim
```

Mitochondrial_central_dogma	mtDNA_maintenance	mtDNA_repli
<chr>	<chr>	<chr>
1 Mitochondrial_central_dogma	Mitochondrial_central_dogma.mtDNA_maintenance	Mitochondrial_ce
2 AARS2	APEX1	DNA2
3 ALKBH1	ATAD3A	EXOG
4 ANGEL2	ATAD3B	LIG3
5 APEX1	DNA2	MGME1
6 ATAD3A	ENDOG	POLG

▼ Code

```
gene_list_df <-
  description_df_mouse_zf[description_df_mouse_zf$adult_ko_vs_adult_wt_P.Value <=
  gene_list_df <- gene_list_df %>% arrange(desc(adult_ko_vs_adult_wt_logFC))
  gene_list_df <- gene_list_df[!is.na(gene_list_df$gene_mouse),]
  gene_list_df <- gene_list_df[!duplicated(gene_list_df$gene_mouse),]
  gene_list <- gene_list_df %>% pull(adult_ko_vs_adult_wt_logFC)
  names(gene_list) <- gene_list_df %>% pull(gene_mouse)

  gene_list <- gene_list[!duplicated(gene_list)]
  print(head(gene_list))
```

Col1a1 Argap4 Col6a3 Adgrv1 Decr1 Dnm2
 4.609072 4.524977 4.040525 3.784338 3.735197 3.418267

▼ Code

```
gene_list %>% length
names(gene_list)
```

224

'Col1a1' · 'Arhgap4' · 'Col6a3' · 'Adgrv1' · 'Decr1' · 'Dnm2' · 'Hbb-y' · 'Slc5a12' · 'Ltf' · 'Phldb1' ·
 'Hbb-bh1' · 'Ndc80' · 'Col1a2' · 'Klhl4' · 'Rcor2' · 'Dynll1' · '4930415L06Rik' · 'Postn' · 'Orm1' · 'Cct2' ·
 'Mdh1' · 'Snd1' · 'Rps28' · 'Larp1b' · 'Abcd2' · 'Hspb1' · 'Rps17' · 'Cct3' · 'Aco2' · 'Rps10' · 'Fga' · 'Ppia' ·
 'Opa1' · 'Cltc' · 'Traf1' · 'Hsp90b1' · 'Mpz' · 'G6pc3' · 'Fgg' · 'Ndafb7' · 'Tktl2' · 'Serpinb8' · 'Ssr3' ·
 'Synpo2' · 'Mtx2' · 'Cd99l2' · 'Endou' · 'Ube2k' · 'Klhl41' · 'Rps14' · 'Tmf1' · 'Cox7a2' · 'Fxr1' · 'Gm5096' ·
 'Tmem232' · 'Acly' · 'Serbp1' · 'Hsp90ab1' · 'Anxa5' · 'Eif3j1' · 'Rrbp1' · 'Hnrnpab' · 'Ppp5c' · 'Adh7' ·
 'Lamb2' · 'Col12a1' · 'Jup' · 'Myoz3' · 'Ndufa8' · 'Oxct2b' · 'Rps26' · 'Mybpc2' · 'Anxa2' · 'Rps5' ·
 'Gm5244' · 'Rpl23' · 'Wfs1' · 'Glud1' · 'Myh9' · 'Camk2b' · 'Aars' · 'Pabpc4' · 'Ilk' · 'Chrna1' · 'Tln2' ·
 'Col11a1' · 'Ndufa3' · 'Tcap' · 'Rpl12' · 'Ddx39a' · 'Dhtkd1' · 'Ptges3' · 'Edf1' · 'Psmb6' · 'Acta1' · 'Aimp1' ·
 'Pdlim5' · 'Prkcsh' · 'Cap2' · 'Rbm31y' · 'Prodh' · 'Aldh4a1' · 'Col4a2' · 'Pank4' · 'Cct7' · 'Klhl31' · 'Rpl27' ·
 'Col6a2' · 'Rps18-ps6' · 'Mrpl12' · 'Apoo' · 'Pdk1' · 'Ywhah' · 'Ccde88c' · 'Vapb' · 'Psmd2' · 'Mlh3' ·
 'Psmd4' · 'Tnnt1' · 'Myl9' · 'Pkrl' · 'Eci3' · 'Des' · 'Recql' · 'Eno3' · 'Zdhhc5' · 'Trdn' · 'Pfkp' · 'Bin1' ·
 'Eif5a2' · 'Ndufc2' · 'Aldoart2' · 'Txndc17' · 'Em13' · 'Edrf1' · 'Rpn2' · 'Tpi1' · 'Atp1a4' · 'Got1' · 'Spata7' ·
 'Tmem38a' · 'Ppk1' · 'Slc25a20' · 'Kif13b' · 'G3bp1' · 'Fibp' · 'Dnah3' · 'Chchd10' · 'Pkm' · 'Pebp1' ·
 'Chchd2' · 'Krt19' · 'Rps6ka1' · 'Rtn4' · 'Ank2' · 'Tpm1' · 'Etfdh' · 'Rab7' · 'Myl2' · 'Prdx3' · 'Pfkm' ·
 'Slc25a4' · 'Smyd1' · 'Top2a' · 'Cfap58' · 'Pygm' · 'Rpl34' · 'Slk' · 'Gpd1' · 'Gpd1l' · 'Dvl3' · 'Sorbs1' ·
 'Shprh' · 'Shank2' · 'Ankrd10' · 'Chd9' · 'Ldha' · 'Slc6a21' · 'Ccn5' · 'Mns1' · 'Nav1' · 'Gpi1' · 'Golga4' ·
 'Ankrd52' · 'Flnc' · 'Hibadh' · 'Prmt2' · 'Igdcc3' · 'Mt-Co2' · 'Ptprb' · 'Uqcrh' · 'Pygb' · 'Apobec2' · 'Myl10' ·
 'Scn10a' · 'Cops4' · 'Ddx59' · 'Sh3gl1' · 'Gm10358' · 'Hdac7' · 'Glo1' · 'Cox7c' · 'Sin3b' · 'Pcdhac2' ·
 'P2rx4' · 'Ikzf5' · 'Bcar3' · 'Sbf2' · 'Slc6a2' · 'Cox6b2' · 'Cox4i2' · 'Atp5d' · 'Twf1' · 'Bloc1s6' · 'H6pd' ·
 'Tpm2' · 'Epc1' · 'Micall2' · 'Mybph' · 'Cox5b-ps' · 'Ugt8a' · 'Pvalb' · 'Cox5a' · 'Cox4i1'

▼ Code

```
names(gene_list)[grepl("Cox7", names(gene_list))]
(mitocarta_mouse_pathways_rows_list$gene)[grepl("Cox7",
(mitocarta_mouse_pathways_rows_list$gene))]
```

```
'Cox7a2' · 'Cox7c'  
'Cox7a1' · 'Cox7a2' · 'Cox7a2l' · 'Cox7b' · 'Cox7c' · 'Cox7a1' · 'Cox7a2' · 'Cox7a2l' · 'Cox7b' · 'Cox7c' ·  
'Cox7a1' · 'Cox7a2' · 'Cox7a2l' · 'Cox7b' · 'Cox7c' · 'Cox7a2l' · 'Cox7a1' · 'Cox7a2' · 'Cox7a2l' · 'Cox7b' ·  
'Cox7c' · 'Cox7a2l'
```

▼ Code

```
names(gene_list)[names(gene_list) %in% mitocarta_mouse_pathways_rows_list$gene]
```

```
'Decr1' · 'Abcd2' · 'Aco2' · 'Opa1' · 'Ndufb7' · 'Mtx2' · 'Cox7a2' · 'Acly' · 'Ndufa8' · 'Glud1' · 'Ndufa3' ·  
'Dhtkd1' · 'Prodh' · 'Aldh4a1' · 'Mrpl12' · 'Apoo' · 'Pdk1' · 'Ndufc2' · 'Slc25a20' · 'Chchd2' · 'Etfdh' ·  
'Prdx3' · 'Slc25a4' · 'Hibadh' · 'mt-Co2' · 'Uqcrh' · 'Cox7c' · 'Cox6b2' · 'Cox4i2' · 'Atp5d' · 'Cox5a' · 'Cox4i1'
```

▼ Code

```
intersect(names(gene_list), mitocarta_mouse_pathways_rows_list$gene)
```

```
'Decr1' · 'Abcd2' · 'Aco2' · 'Opa1' · 'Ndufb7' · 'Mtx2' · 'Cox7a2' · 'Acly' · 'Ndufa8' · 'Glud1' · 'Ndufa3' ·  
'Dhtkd1' · 'Prodh' · 'Aldh4a1' · 'Mrpl12' · 'Apoo' · 'Pdk1' · 'Ndufc2' · 'Slc25a20' · 'Chchd2' · 'Etfdh' ·  
'Prdx3' · 'Slc25a4' · 'Hibadh' · 'mt-Co2' · 'Uqcrh' · 'Cox7c' · 'Cox6b2' · 'Cox4i2' · 'Atp5d' · 'Cox5a' · 'Cox4i1'
```

▼ Code

```
library(viridis)
```

Loading required package: viridisLite

▼ Code

```
gsea_mito <- GSEA(gene_list, TERM2GENE = mitocarta_mouse_pathways_rows_list,  
                    pvalueCutoff = 0.05)  
dotplot(gsea_mito, showCategory = 15, title = paste0("Mitopathway Geneset Enrichment  
Analysis"),  
        font.size = 12) + facet_grid(.~.sign)+ scale_size_area(limits = c(0, 60))+  
        scale_colour_viridis(option = "plasma", direction = 1, limits=c(0, 0.2))+  
        NULL  
# Save dotplot as pdf file  
dev.copy(  
  pdf,  
  file = paste0("proteomics_n_3_muscle/GSEA_subcluster/Mitopathway_GSEA.pdf"),  
  width = 22,  
  height = 8  
)  
dev.off ()  
  
gsea_mito_df <- as.data.frame(gsea_mito)  
gsea_mito_df  
  
gsea_mito_df <- gsea_mito_df %>% tidyverse::separate_rows(core_enrichment, sep = "/",  
                           convert = FALSE) %>%  
                           arrange((p.adjust))  
  
# Save enriched pathways data frame as CSV file  
write.csv(gsea_mito_df,  
          paste0("proteomics_n_3_muscle/GSEA_subcluster/Mitopathway_GSEA.csv"))
```

```

# Print message indicating that analysis for the current cell type is complete
#   message(paste0("Cell type: ", i, " done"))
message(paste0("*****"))
message(paste0("\n"))

```

preparing geneSet collections...

GSEA analysis...

leading edge analysis...

done...

Scale for size is already present.

Adding another scale for size, which will replace the existing scale.

Scale for colour is already present.

Adding another scale for colour, which will replace the existing scale.

pdf: 3

png: 2

A data.frame: 2

ID	Description	setSize	enrichmentScore	NES	pvalue	p.adjust
<chr>	<chr>	<int>	<dbl>	<dbl>	<dbl>	<dbl>
OXPHOS	OXPHOS	OXPHOS	13	-0.6880534	-2.210767	0.0002748124
OXPHOS	OXPHOS	OXPHOS	13	-0.6880534	-2.210767	0.0002748124
subunits	subunits	subunits				0.0004122186

◀ ▶

**



5 PCA

▼ Code

```
# Function to plot PCA
PCA <- function(mat,color_pca="",shape_pca= "", label_pca= "", save_plot= "no",
                 name_of_plot= "PCA", comp1=1, comp2=2, height=10, width=10){
  # mat: matrix of counts
  # color_pca: vector of colors for each sample
  # shape_pca: vector of shapes for each sample
  # label_pca: vector of labels for each sample
  # save_plot: yes or no
```

```

# name_of_plot: name of the plot
# comp1: component 1
# comp2: component 2
# height: height of the plot
# width: width of the plot

#1. Extract the counts.
dt <- mat

#2. Perform pca
pca_dt <- prcomp(t(dt))
cat("PCA running...\n")

#3. Extract percentVar data.
percentVar_dt <- pca_dt$sdev^2/sum(pca_dt$sdev^2)
cat("Percents calculated...\n")

#4. Create the new dataframe to plot.
dt_f <- data.frame(PC1=pca_dt$x[,comp1],
                     PC2=pca_dt$x[,comp2],
                     color_pca=color_pca,
                     shape_pca=shape_pca,
                     label_pca= label_pca)
cat("Data frame built...\n")

#5. Plot it
cat("Plotting...\n")

print(save_plot)
require(ggplot2)
require(ggrepel)
if (save_plot== "no") {
  pca_p <- ggplot(data = dt_f, aes_string(x = paste0("PC1"),
                                             y = paste0("PC2"),
                                             color = "color_pca",
                                             shape= "shape_pca", label="label_pca")) +
    geom_point(size = 3) +
    geom_text_repel(size= 3, max.overlaps = 50,
                   box.padding = 1.5, point.padding = 0.5, force = 50) +
    xlab(paste0("PC", comp1,": ",
                round(percentVar_dt[comp1] * 100), "% variance")) +
    ylab(paste0("PC",comp2,": ",
                round(percentVar_dt[comp2] * 100), "% variance")) +
    theme_classic()
  NULL
}
if (save_plot== "yes"){
  dev.copy(pdf,paste0(name_of_plot,".pdf"),width = width,height = height)
  cat("Saving plot as: ",paste0(name_of_plot,"...\n"))
}

```

```

pca_p <- ggplot(data = dt_f, aes_string(x = paste0("PC",comp1),
                                         y = paste0("PC",comp2),
                                         color = "color_pca",
                                         shape= "shape_pca",
                                         label="label_pca"
                                         )) +
  geom_text_repel(size= 3, max.overlaps = 50,
                  box.padding    = 1.5,
                  point.padding = 0.5, force = 50) +
  geom_point(size = 3) +
  xlab(paste0("PC", comp1,": ", round(percentVar_dt[comp1] * 100), "% variance")) +
  ylab(paste0("PC",comp2,": ", round(percentVar_dt[comp2] * 100), "% variance")) +
  theme_classic()

NULL
print(pca_p)
dev.off()
}

cat("Done")
print(pca_p)

#return(pca_p)
}

```

▼ Code

```

# prepare the data for PCA
pca_matrix <- data_merged %>% dplyr::select(c('AK01', 'AK02', 'AK03', 'AWT1', 'AWT2',
                                                'AWT3', 'JWT1', 'JWT2', 'JWT3', "gene")) # select only the counts columns
rownames(pca_matrix) <- make.unique(pca_matrix$gene) # set the gene names as row names
pca_matrix <- pca_matrix %>% dplyr::select(-gene) # remove the gene column
# make all columns numeric
pca_matrix <- pca_matrix %>% mutate_all(as.numeric) # make all columns numeric
pca_matrix <- as.matrix(pca_matrix) # convert to matrix
pca_matrix %>% head

```

A matrix: 6 × 9 of type dbl

	AKO1	AKO2	AKO3	AWT1	AWT2	AWT3	JWT1	J
aamdc	1.4667515	-0.5739485	-1.8653050	0.26725443	-0.23076900	0.9212840	-1.5362111	-
aars1	-1.3534255	-1.5746633	-1.7947352	-2.83197660	-2.54200323	-2.6800583	3.3682986	3
aass	0.7297959	1.3124487	0.2894110	0.53375739	0.05744185	-0.9853736	-0.5110688	-
abcb10	0.2852603	1.2430460	0.6226768	0.21875262	0.09663467	0.7806194	1.5690452	1
abcb11a	0.6139335	2.1746058	1.2280779	0.54923656	-5.56276938	-2.2331746	-0.3427620	-
abcb11b	0.3971808	-0.2657106	-0.6573271	-0.01452689	0.54376251	-0.5148912	1.2912932	-

▼ Code

```

# prepare the metadata for PCA
metadata <- data.frame(samples = colnames(pca_matrix),
                       condition= NA) # create a dataframe with samples and
                           condition columns
# Where samples is AKO1/AKO2/AKO3 condition with Cox7a1
metadata$condition[metadata$samples %in% c("AKO1", "AKO2", "AKO3")] <- "Cox7a1"
# Where samples is AWT1/AWT2/AWT3 condition with WT
metadata$condition[metadata$samples %in% c("AWT1", "AWT2", "AWT3")] <- "WT"
# Where samples is JWT1/JWT2/JWT3 condition with Juvenile
metadata$condition[metadata$samples %in% c("JWT1", "JWT2", "JWT3")] <- "Juvenile"

metadata$condition <- factor(metadata$condition, levels = c( "Cox7a1", "WT",
    "Juvenile")) # set the levels of the condition column

metadata

```

A data.frame: 9 × 2

samples	condition
<chr>	<fct>
AKO1	Cox7a1
AKO2	Cox7a1
AKO3	Cox7a1
AWT1	WT
AWT2	WT
AWT3	WT
JWT1	Juvenile
JWT2	Juvenile
JWT3	Juvenile

▼ Code

```

dir.create("./proteomics_n_3_muscle/figures", recursive = TRUE) # create a directory
to save the plots
PCA(pca_matrix, color_pca = metadata$condition,
shape_pca = metadata$condition,
save_plot = "yes", name_of_plot =
"./proteomics_n_3_muscle/figures/PCA_proteomics_n_3_muscle_nolabels", height
= 6, width = 8) # perform PCA and save the plot

```

Warning message in dir.create("./proteomics_n_3_muscle/figures", recursive = TRUE):
'./proteomics_n_3_muscle/figures' already exists"

PCA running...
Percents calculated...
Data frame built...
Plotting...
[1] "yes"

```
Loading required package: ggplot2
```

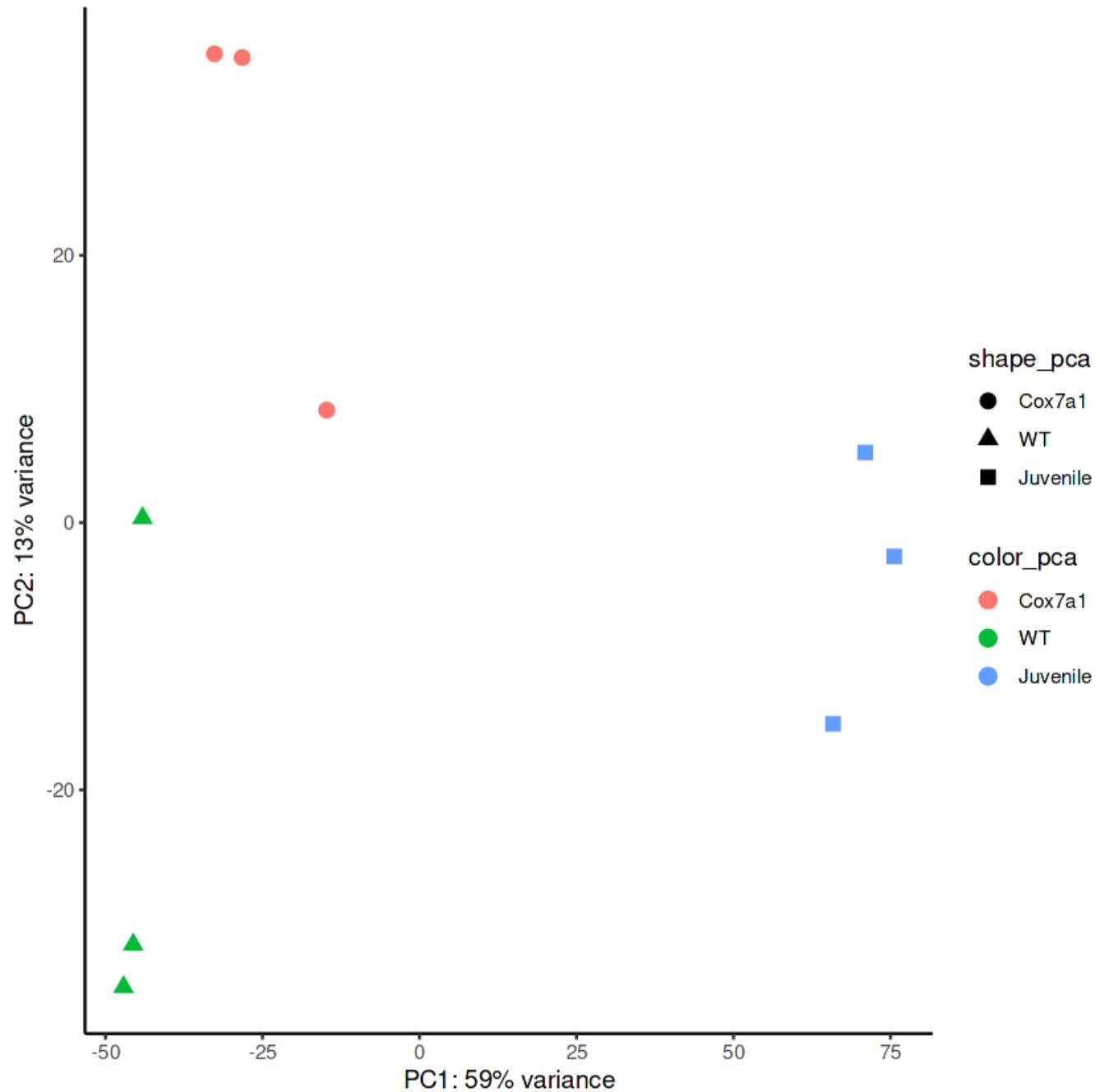
```
Loading required package: ggrepel
```

```
Saving plot as: ./proteomics_n_3_muscle/figures/PCA_proteomics_n_3_muscle_nolabels...
```

Warning message:

```
"`aes_string()` was deprecated in ggplot2 3.0.0.  
i Please use tidy evaluation idioms with `aes()``.  
i See also `vignette("ggplot2-in-packages")` for more information."
```

Done



6 Add more annotations and manual curation

▼ Code

```
# in the column gene remove all the numbers after the dot and the dot
df_merged_description$gene <- gsub("\\..*", "", df_merged_description$gene)
```

▼ Code

```
description_df_mouse_zf <-
  merge(df_merged_description, description_df_mouse[,c("gene", # merge the
  description dataframe with the merged dataframe
  colnames(description_df_mouse)[!(colnames(description_df_mouse) %in%
  colnames(df_merged_description))]]], by="gene", all.x = TRUE)
description_df_mouse_zf <- description_df_mouse_zf %>% dplyr::rename(
  gene_full_name_zf = gene_full_name,
  GO_description_zf = GO_description,
  associated_phenotype_zf = associated_phenotype,
  GO_name_zf = GO_name,
  GO_family_zf = GO_family,
)
description_df_mouse_zf %>% colnames()
```

'gene' · 'protein' · 'AKO1' · 'AKO2' · 'AKO3' · 'AWT1' · 'AWT2' · 'AWT3' · 'JWT1' · 'JWT2' · 'JWT3' ·
 'NOP' · 'adult_ko_vs_adult_wt_logFC' · 'adult_ko_vs_adult_wt_PValue' · 'adult_ko_vs_adult_wt_adj.PVal' ·
 'adult_ko_vs_juvenile_wt_logFC' · 'adult_ko_vs_juvenile_wt_PValue' · 'adult_ko_vs_juvenile_wt_adj.PVal' ·
 'adult_wt_vs_juvenile_wt_logFC' · 'adult_wt_vs_juvenile_wt_PValue' · 'adult_wt_vs_juvenile_wt_adj.PVal' ·
 'gene_full_name_zf' · 'GO_description_zf' · 'associated_phenotype_zf' · 'GO_name_zf' · 'GO_family_zf' ·
 'gene_mouse' · 'gene_full_name_mouse' · 'GO_description_mouse' · 'associated_phenotype_mouse' ·
 'GO_name_mouse' · 'GO_family_mouse'

▼ Code

```
# Import the mouse mitocarta genes
mito_genes <- readxl::read_excel("Mouse.MitoCarta3.0.xls", sheet = 2)
# head(mito_genes)
mito_genes <- mito_genes %>% dplyr::select('Symbol', "Description",
  'MitoCarta3.0_SubMitoLocalization', 'MitoCarta3.0_MitoPathways')
colnames(mito_genes)
mito_genes %>% dim
head(mito_genes)
```

'Symbol' · 'Description' · 'MitoCarta3.0_SubMitoLocalization' · 'MitoCarta3.0_MitoPathways'
 1140 · 4

A tibble: 6 × 4

Symbol	Description	MitoCarta3.0_SubMitoLocalization	MitoCarta3.0_MitoPathways
<chr>	<chr>	<chr>	<chr>
Cyc1	cytochrome c-1	MIM	OXPHOS > Complex III > CIII subunits Metabolism > Metals and cofactors > Heme-containing proteins Metabolism > Electron carriers > Cytochromes OXPHOS > OXPHOS subunits

Symbol	Description	MitoCarta3.0_SubMitoLocalization	MitoCarta3.0_MitoPathways
<chr>	<chr>	<chr>	<chr>
Pdha1	pyruvate dehydrogenase E1 alpha 1	Matrix	Metabolism > Carbohydrate metabolism > Pyruvate metabolism
Atp5d	ATP synthase, H+ transporting, mitochondrial F1 complex, delta subunit	MIM	OXPHOS > Complex V > CV subunits OXPHOS > OXPHOS subunits
Isca2	iron-sulfur cluster assembly 2	Matrix	Metabolism > Metals and cofactors > Fe-S cluster biosynthesis Metabolism > Metals and cofactors > Fe-S-containing proteins
Pdhb	pyruvate dehydrogenase (lipoamide) beta	Matrix	Metabolism > Carbohydrate metabolism > Pyruvate metabolism
Uqcrc1	ubiquinol-cytochrome c reductase core protein 1	MIM	Protein import, sorting and homeostasis > Protein import and sorting > Preprotein cleavage OXPHOS > Complex III > CIII subunits OXPHOS > OXPHOS subunits

▼ Code

```
# merge the mito genes with the description dataframe
description_df_mouse_zf <- merge(description_df_mouse_zf,mito_genes,
    by.x="gene_mouse", by.y="Symbol", all.x = TRUE)
description_df_mouse_zf %>% dim
colnames(description_df_mouse_zf)
```

2511 · 35

'gene_mouse' · 'gene' · 'protein' · 'AKO1' · 'AKO2' · 'AKO3' · 'AWT1' · 'AWT2' · 'AWT3' · 'JWT1' · 'JWT2' · 'JWT3' · 'NOP' · 'adult_ko_vs_adult_wt_logFC' · 'adult_ko_vs_adult_wt_PValue' · 'adult_ko_vs_adult_wt_AdjPVal' · 'adult_ko_vs_juvenile_wt_logFC' · 'adult_ko_vs_juvenile_wt_PValue' · 'adult_ko_vs_juvenile_wt_AdjPVal' · 'adult_wt_vs_juvenile_wt_logFC' · 'adult_wt_vs_juvenile_wt_PValue' · 'adult_wt_vs_juvenile_wt_AdjPVal' · 'gene_full_name_zf' · 'GO_description_zf' · 'associated_phenotype_zf' · 'GO_name_zf' · 'GO_family_zf' · 'gene_full_name_mouse' · 'GO_description_mouse' · 'associated_phenotype_mouse' · 'GO_name_mouse' · 'GO_family_mouse' · 'Description' · 'MitoCarta3.0_SubMitoLocalization' · 'MitoCarta3.0_MitoPathways'

▼ Code

```
nadia_annotations_manual <- readxl::read_excel("new_proteomics_muslce_and
heart_NM.xlsx", sheet = 1) # read the manual annotations
colnames(nadia_annotations_manual)[22] <- 'confirmation_status_muscle'
colnames(nadia_annotations_manual)[23] <- 'info_genecards_uniprot_nadia'
```

```

colnames(nadia_annotations_manual)
nadia_annotations_manual <- nadia_annotations_manual %>%
  dplyr::select(gene, confirmation_status_muscle, info_genecards_uniprot_nadia)
nadia_annotations_manual$gene <- gsub("\\..*", "", nadia_annotations_manual$gene)
nadia_annotations_manual_2 <- readxl::read_excel("new_proteomics_muslce_and
  heart_NM.xlsx", sheet = 4)
colnames(nadia_annotations_manual_2)
  [colnames(nadia_annotations_manual_2)=="COMPARISON"] <-
  'confirmation_status_ventricle'
colnames(nadia_annotations_manual_2)[colnames(nadia_annotations_manual_2)=="info_gene
  cards:uniprot"] <- 'info_genecards_uniprot_nadia'
colnames(nadia_annotations_manual_2)
nadia_annotations_manual_2 <- nadia_annotations_manual_2 %>%
  dplyr::select(gene, confirmation_status_ventricle,
  info_genecards_uniprot_nadia)
nadia_annotations_manual_2$gene <- gsub("\\..*", "", nadia_annotations_manual_2$gene)
nadia_annotations_manual <-
  merge(nadia_annotations_manual, nadia_annotations_manual_2, by="gene", all.x
  = TRUE)
nadia_annotations_manual$confirmation_status_muscle[is.na(nadia_annotations_manual$confi
  <- ""
nadia_annotations_manual$confirmation_status_ventricle[is.na(nadia_annotations_manual$confi
  <- ""
nadia_annotations_manual$info_genecards_uniprot_nadia.x[is.na(nadia_annotations_manual$in
  <- ""
nadia_annotations_manual$info_genecards_uniprot_nadia.y[is.na(nadia_annotations_manual$in
  <- ""

# merge the columns info_genecards_uniprot_nadia.x and info_genecards_uniprot_nadia.y
# into one column info_genecards_uniprot_nadia only if there is an entry in
# either of the two columns, else take the only value available
nadia_annotations_manual$info_genecards_uniprot_nadia <-
  ifelse(nadia_annotations_manual$info_genecards_uniprot_nadia.x == "" &
  nadia_annotations_manual$info_genecards_uniprot_nadia.y == "",
  nadia_annotations_manual$info_genecards_uniprot_nadia.x,
  paste(nadia_annotations_manual$info_genecards_uniprot_nadia.x,
  nadia_annotations_manual$info_genecards_uniprot_nadia.y, sep = ";"))
nadia_annotations_manual$info_genecards_uniprot_nadia <- gsub("^;|;$", "", 
  nadia_annotations_manual$info_genecards_uniprot_nadia)

nadia_annotations_manual <- nadia_annotations_manual %>%
  dplyr::select(gene, confirmation_status_muscle,
  confirmation_status_ventricle, info_genecards_uniprot_nadia)
# remove ; at the beginning of the string or at the end of the string

head(nadia_annotations_manual)

```

New names:

- ` ` -> `...22`

```
'gene' · 'protein' · 'AKO1' · 'AKO2' · 'AKO3' · 'AWT1' · 'AWT2' · 'AWT3' · 'JWT1' · 'JWT2' · 'JWT3' · ▶
'NOP' · 'adult_ko_vs_adult_wt_logFC' · 'adult_ko_vs_adult_wt_PValue' · 'adult_ko_vs_adult_wt_adj.PVal' ·
'adult_ko_vs_juvenile_wt_logFC' · 'adult_ko_vs_juvenile_wt_PValue' · 'adult_ko_vs_juvenile_wt_adj.PVal' ·
'adult_wt_vs_juvenile_wt_logFC' · 'adult_wt_vs_juvenile_wt_PValue' · 'adult_wt_vs_juvenile_wt_adj.PVal' ·
'confirmation_status_muscle' · 'info_genecards_uniprot_nadia'
'gene' · 'protein' · 'KO1' · 'KO2' · 'KO3' · 'WT1' · 'WT2' · 'WT3' · 'NOP' · 'ko_vs_wt_logFC' ·
'ko_vs_wt_PValue' · 'ko_vs_wt_adj.PVal' · 'confirmation_status_ventricle' · 'info_genecards_uniprot_nadia'
```

A data.frame: 6 × 4

gene	confirmation_status_muscle	confirmation_status_ventricle	info_genecards_uniprot_nadia
<chr>	<chr>	<chr>	<chr>
1 aamdc			
2 aars1			
3 aass			
4 abcb10			
5 abcb11a			ATP Binding Cassette Subfamily B Member 11,transporter activity and ATPase-coupled transmembrane transporter activity.liver
6 abcb11b			

▼ Code

```
description_df_mouse_zf <- merge(description_df_mouse_zf, nadia_annotations_manual,
    by="gene", all.x = TRUE)
description_df_mouse_zf %>% dim
colnames(description_df_mouse_zf)
```

3896 · 38

'gene' · 'gene_mouse' · 'protein' · 'AKO1' · 'AKO2' · 'AKO3' · 'AWT1' · 'AWT2' · 'AWT3' · 'JWT1' · 'JWT2' ·
'JWT3' · 'NOP' · 'adult_ko_vs_adult_wt_logFC' · 'adult_ko_vs_adult_wt_P.Value' ·
'adult_ko_vs_adult_wt_Adj.PVal' · 'adult_ko_vs_juvenile_wt_logFC' · 'adult_ko_vs_juvenile_wt_P.Value' ·
'adult_ko_vs_juvenile_wt_Adj.PVal' · 'adult_wt_vs_juvenile_wt_logFC' · 'adult_wt_vs_juvenile_wt_P.Value' ·
'adult_wt_vs_juvenile_wt_Adj.PVal' · 'gene_full_name_zf' · 'GO_description_zf' ·
'associated_phenotype_zf' · 'GO_name_zf' · 'GO_family_zf' · 'gene_full_name_mouse' ·
'GO_description_mouse' · 'associated_phenotype_mouse' · 'GO_name_mouse' · 'GO_family_mouse' ·
'Description' · 'MitoCarta3.0_SubMitoLocalization' · 'MitoCarta3.0_MitoPathways' ·
'confirmation_status_muscle' · 'confirmation_status_ventricle' · 'info_genecards_uniprot_nadia'

▼ Code

```
description_df_mouse_zf <- description_df_mouse_zf %>% dplyr::select(-
    contains("juv")) # remove juvenile columns as they are not needed
# remove ; from the beginning or the end of the string across all columns
description_df_mouse_zf <- description_df_mouse_zf %>% mutate(across(everything(),
    ~str_replace(., "^[ ;]| ; $", "")))
description_df_mouse_zf %>% colnames() %>% length()
```

32

▼ Code

```
# mitribosome genes as per
# https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4501431/#:~:text=The%20model%20of
# none found
#
```

```

df_merged_description[grepl("Malsu", df_merged_description$gene, ignore.case = TRUE),] %>% pull(gene)

df_merged_description[grepl("mt-Tv", df_merged_description$gene, ignore.case = TRUE),] %>% pull(gene)
mitoribosome <- df_merged_description[grepl("Mitochondrial Ribosomal", df_merged_description$gene_full_name, ignore.case = TRUE)|grepl("Mitochondrial Ribosomal", df_merged_description$GO_description, ignore.case = TRUE),]

mitoribosome_2 <- df_merged_description[grepl("mitochondrial", df_merged_description$gene_full_name, ignore.case = TRUE)|grepl("mitochondrial", df_merged_description$GO_description, ignore.case = TRUE),] %>% filter(grepl("mitochondria", gene_full_name, ignore.case = TRUE)&grepl("ribosom", GO_description, ignore.case = TRUE))

mitoribosome_combined <- rbind(mitoribosome, mitoribosome_2)
mitoribosome_combined %>% dim
mitoribosome_combined %>% pull(gene) %>% unique()
mitoribosome_combined %>% write.csv("./proteomics_n_3_muscle/mitoribosome_check.csv")

```

11 · 26

'mrpl12' · 'mrps15' · 'mrps36' · 'mrps9' · 'mul1a' · 'tufm' · 'vars2'

▼ Code

```

description_df_mouse[grepl("Malsu", description_df_mouse$gene, ignore.case = TRUE)|grepl("Malsu", description_df_mouse$gene_mouse),] %>% pull(gene)

description_df_mouse[grepl("mt-Tv", description_df_mouse$gene, ignore.case = TRUE)|grepl("mt-Tv", description_df_mouse$gene_mouse),] %>% pull(gene)

mitoribosome_mouse <- description_df_mouse_zf[grepl("Mitochondrial Ribosomal", description_df_mouse_zf$gene_full_name_mouse, ignore.case = TRUE)|grepl("Mitochondrial Ribosomal", description_df_mouse_zf$GO_description_mouse, ignore.case = TRUE)|grepl("Mitochondrial Ribosomal", description_df_mouse_zf$gene_full_name_zf, ignore.case = TRUE)|grepl("Mitochondrial Ribosomal", description_df_mouse_zf$GO_description_zf, ignore.case = TRUE),]

colnames(mitoribosome_mouse)
mitoribosome_2_mouse <- description_df_mouse_zf[grepl("mitochondrial", description_df_mouse_zf$gene_full_name_mouse, ignore.case = TRUE)|grepl("mitochondrial", description_df_mouse_zf$GO_description_mouse, ignore.case = TRUE),] %>% filter(grepl("mitochondria", gene_full_name_mouse, ignore.case = TRUE)&grepl("ribosom", GO_description_mouse, ignore.case = TRUE)|grepl("mitochondria", gene_full_name_zf, ignore.case = TRUE)&grepl("ribosom", GO_description_zf, ignore.case = TRUE))

colnames(mitoribosome_2_mouse)

mitoribosome_combined_mouse <- rbind(mitoribosome_mouse, mitoribosome_2_mouse) %>% filter(adult_ko_vs_adult_wt_P.Value<0.05) %>% unique
colnames(mitoribosome_combined_mouse)
mitoribosome_combined_mouse %>% dim

mitoribosome_combined_mouse %>% filter(adult_ko_vs_adult_wt_P.Value<0.05) %>%

```

```
pull(gene) %>% unique  
mitoribosome_combined_mouse %>% colnames
```

```
'gene' · 'gene_mouse' · 'protein' · 'AKO1' · 'AKO2' · 'AKO3' · 'AWT1' · 'AWT2' · 'AWT3' · 'JWT1' · 'JWT2' ·  
'JWT3' · 'NOP' · 'adult_ko_vs_adult_wt_logFC' · 'adult_ko_vs_adult_wt_P.Value' ·  
'adult_ko_vs_adult_wt_adj.P.Val' · 'gene_full_name_zf' · 'GO_description_zf' · 'associated_phenotype_zf' ·  
'GO_name_zf' · 'GO_family_zf' · 'gene_full_name_mouse' · 'GO_description_mouse' ·  
'associated_phenotype_mouse' · 'GO_name_mouse' · 'GO_family_mouse' · 'Description' ·  
'MitoCarta3.0_SubMitoLocalization' · 'MitoCarta3.0_MitoPathways' · 'confirmation_status_muscle' ·  
'confirmation_status_ventricle' · 'info_genecards_uniprot_nadia'  
'gene' · 'gene_mouse' · 'protein' · 'AKO1' · 'AKO2' · 'AKO3' · 'AWT1' · 'AWT2' · 'AWT3' · 'JWT1' · 'JWT2' ·  
'JWT3' · 'NOP' · 'adult_ko_vs_adult_wt_logFC' · 'adult_ko_vs_adult_wt_P.Value' ·  
'adult_ko_vs_adult_wt_adj.P.Val' · 'gene_full_name_zf' · 'GO_description_zf' · 'associated_phenotype_zf' ·  
'GO_name_zf' · 'GO_family_zf' · 'gene_full_name_mouse' · 'GO_description_mouse' ·  
'associated_phenotype_mouse' · 'GO_name_mouse' · 'GO_family_mouse' · 'Description' ·  
'MitoCarta3.0_SubMitoLocalization' · 'MitoCarta3.0_MitoPathways' · 'confirmation_status_muscle' ·  
'confirmation_status_ventricle' · 'info_genecards_uniprot_nadia'  
'gene' · 'gene_mouse' · 'protein' · 'AKO1' · 'AKO2' · 'AKO3' · 'AWT1' · 'AWT2' · 'AWT3' · 'JWT1' · 'JWT2' ·  
'JWT3' · 'NOP' · 'adult_ko_vs_adult_wt_logFC' · 'adult_ko_vs_adult_wt_P.Value' ·  
'adult_ko_vs_adult_wt_adj.P.Val' · 'gene_full_name_zf' · 'GO_description_zf' · 'associated_phenotype_zf' ·  
'GO_name_zf' · 'GO_family_zf' · 'gene_full_name_mouse' · 'GO_description_mouse' ·  
'associated_phenotype_mouse' · 'GO_name_mouse' · 'GO_family_mouse' · 'Description' ·  
'MitoCarta3.0_SubMitoLocalization' · 'MitoCarta3.0_MitoPathways' · 'confirmation_status_muscle' ·  
'confirmation_status_ventricle' · 'info_genecards_uniprot_nadia'
```

7 · 32

```
'mrpl12' · 'mrps15' · 'mrps36' · 'mrps9' · 'shmt2' · 'tufm' · 'vars2'  
'gene' · 'gene_mouse' · 'protein' · 'AKO1' · 'AKO2' · 'AKO3' · 'AWT1' · 'AWT2' · 'AWT3' · 'JWT1' · 'JWT2' ·  
'JWT3' · 'NOP' · 'adult_ko_vs_adult_wt_logFC' · 'adult_ko_vs_adult_wt_P.Value' ·  
'adult_ko_vs_adult_wt_adj.P.Val' · 'gene_full_name_zf' · 'GO_description_zf' · 'associated_phenotype_zf' ·  
'GO_name_zf' · 'GO_family_zf' · 'gene_full_name_mouse' · 'GO_description_mouse' ·  
'associated_phenotype_mouse' · 'GO_name_mouse' · 'GO_family_mouse' · 'Description' ·  
'MitoCarta3.0_SubMitoLocalization' · 'MitoCarta3.0_MitoPathways' · 'confirmation_status_muscle' ·  
'confirmation_status_ventricle' · 'info_genecards_uniprot_nadia'
```

▼ Code

```
description_df_mouse_zf[grepl("mitosis", description_df_mouse_zf$GO_name_zf,  
    ignore.case = TRUE)|grepl("mitosis", description_df_mouse_zf$GO_name_mouse,  
    ignore.case = TRUE),] %>% dplyr::select(gene, contains("mouse")) %>%  
    pull(gene)
```

'phb2a' · 'phb2b' · 'rpl24'

▼ Code

```
cell_cycle_genes <- df_merged_description[grepl("cell cycle",  
    df_merged_description$GO_name, ignore.case = TRUE),] %>% unique  
  
# cell_cycle_genes %>% pull(gene) %>% unique
```

```

cell_cycle_genes_mouse <- description_df_mouse_zf[grepl("cell
    cycle|proliferation|mitotic|mitosis|spindle|kinesin",
description_df_mouse_zf$GO_description_mouse, ignore.case = TRUE)|grepl("cell
    cycle|proliferation|mitotic|mitosis|spindle|kinesin",
description_df_mouse_zf$GO_description_zf, ignore.case = TRUE)|grepl("cell
    cycle|proliferation|mitotic|mitosis|spindle|kinesin",
description_df_mouse_zf$GO_name_mouse, ignore.case = TRUE)|grepl("cell
    cycle|proliferation|mitotic|mitosis|spindle|kinesin",
description_df_mouse_zf$GO_name_zf, ignore.case = TRUE),] %>%
    filter(adult_ko_vs_adult_wt_P.Value<0.05) %>% unique()
cell_cycle_genes_mouse %>% pull(gene)

```

'aimp1a' · 'ank2a' · 'anxa2a' · 'arhgap4a' · 'bin1b' · 'camk2b1' · 'ccdc88c' · 'cct2' · 'cct2' · 'cct3' · 'cct7' ·
 'cfap58' · 'cltcb' · 'col4a2' · 'dnah3' · 'dynll1' · 'dynll1' · 'edf1' · 'eml3' · 'epc1b' · 'got1' · 'gpib' · 'hsp90ab1' ·
 'hspb1' · 'ilk' · 'jupa' · 'kif13bb' · 'klhl41b' · 'mdh1aa' · 'mlh3' · 'mns1' · 'mtx2' · 'myh9b' · 'nav1b' · 'ndc80' ·
 'opa1' · 'pdk1' · 'pepb1' · 'ppp4r3b' · 'prdx3' · 'ptges3b' · 'ptprb' · 'rab7a' · 'rpl12' · 'rpl23' · 'rpl27' ·
 'rpl34' · 'rps10' · 'rps14' · 'rps17' · 'rps28' · 'rps6ka1' · ' rtn4a' · ' rtn4a' · ' rtn4a' · ' shank2a' · ' sin3b' ·
 'smyd1b' · 'snd1' · 'tfa' · 'top2a' · 'tpm1' · 'trdn' · 'trdn' · 'tubb4bl' · 'wfs1a' · 'ywhah'

▼ Code

```

ecm_genes <- df_merged_description[(grepl("extracellular matrix",
    df_merged_description$GO_name, ignore.case = TRUE))|(grepl("col",
    df_merged_description$gene, ignore.case = TRUE)),] %>%
    filter(adult_ko_vs_adult_wt_P.Value<0.05) %>% unique
ecm_genes %>% pull(gene) %>% unique %>% length
ecm_genes %>% pull(gene) %>% unique()

```

11

'atp1a1a' · 'col11a1b' · 'col12a1a' · 'col12a1b' · 'col1a1a' · 'col1a1b' · 'col1a2' · 'col4a2' · 'col6a2' ·
 'col6a3' · 'postnb'

▼ Code

```

ecm_genes_mouse <- description_df_mouse_zf[(grepl("extracellular matrix",
    description_df_mouse_zf$GO_name_mouse, ignore.case = TRUE))|
    (grepl("extracellular matrix", description_df_mouse_zf$GO_name_zf,
    ignore.case = TRUE)),] %>% filter(adult_ko_vs_adult_wt_P.Value<0.05) %>% unique()
ecm_genes_mouse %>% pull(gene) #>% unique %>% length
ecm_genes_mouse %>% colnames

```

'anxa2a' · 'anxa5b' · 'atp1a1a' · 'col11a1b' · 'col12a1a' · 'col12a1b' · 'col1a1a' · 'col1a1b' · 'col1a2' ·
 'col4a2' · 'col6a2' · 'col6a3' · 'col6a3' · 'fga' · 'fga' · 'fgg' · 'lamb2' · 'pkmb' · 'postnb' · 'postnb'
 'gene' · 'gene_mouse' · 'protein' · 'AKO1' · 'AKO2' · 'AKO3' · 'AWT1' · 'AWT2' · 'AWT3' · 'JWT1' · 'JWT2' ·
 'JWT3' · 'NOP' · 'adult_ko_vs_adult_wt_logFC' · 'adult_ko_vs_adult_wt_PValue' ·
 'adult_ko_vs_adult_wt_adj.PVal' · 'gene_full_name_zf' · 'GO_description_zf' · 'associated_phenotype_zf' ·
 'GO_name_zf' · 'GO_family_zf' · 'gene_full_name_mouse' · 'GO_description_mouse' ·
 'associated_phenotype_mouse' · 'GO_name_mouse' · 'GO_family_mouse' · 'Description' ·
 'MitoCarta3.0_SubMitoLocalization' · 'MitoCarta3.0_MitoPathways' · 'confirmation_status_muscle' ·
 'confirmation_status_ventricle' · 'info_genecards_uniprot_nadia'

▼ Code

```

actin_genes_mouse <- description_df_mouse_zf %>% filter(grepl("actin",
  description_df_mouse_zf$GO_name_mouse, ignore.case = TRUE)|grepl("actin",
  description_df_mouse_zf$GO_name_zf, ignore.case = TRUE)) %>%
  filter(adult_ko_vs_adult_wt_P.Value<0.05) %>% unique() #%%>
  dplyr::select(gene, gene_mouse, gene_full_name_mouse, GO_name_mouse,
  GO_description_mouse)
actin_genes_mouse %>% pull(gene)    %>% unique() %>% length
actin_genes_mouse %>% pull(gene)    %>% unique()

# actin_genes_mouse[grepl("aldo", actin_genes_mouse$gene_mouse, ignore.case = TRUE), ]

```

45

```
'aars1' · 'acta1b' · 'adh8a' · 'adh8b' · 'aldh4a1' · 'aldh9a1a' · 'aldoaa' · 'aldoab' · 'anxa5b' · 'bin1b' ·
'camk2b1' · 'cap2' · 'dhtkd1' · 'flnca' · 'gapdh' · 'glud1a' · 'gpd1a' · 'gpd1b' · 'gpd1l' · 'h6pd' · 'hibadhb' ·
'hsp90b1' · 'ilk' · 'jupa' · 'ldha' · 'mdh1aa' · 'micall2b' · 'mybpc2a' · 'myh9b' · 'myl2a' · 'pdlim5a' · 'pfen2b' ·
'ptges3b' · 'rpl23' · 'scinla' · 'slc6a2' · 'synpo2b' · 'tln2a' · 'top2a' · 'tpm1' · 'tpm2' · 'tpm3' · 'tpm4a' ·
'twf1b' · 'ywhah'
```

▼ Code

```

muscle_genes_mouse <- description_df_mouse_zf %>% filter(grepl("muscle",
  description_df_mouse_zf$GO_name_mouse, ignore.case = TRUE)|grepl("muscle",
  description_df_mouse_zf$GO_name_zf, ignore.case = TRUE)) %>%
  filter(adult_ko_vs_adult_wt_P.Value<0.05) %>% unique##>%
  dplyr::select(gene, gene_mouse, gene_full_name_mouse, GO_name_mouse,
  GO_description_mouse)
muscle_genes_mouse %>% pull(gene)    %>% unique() %>% length
muscle_genes_mouse %>% pull(gene)    %>% unique()

```

46

```
'acta1b' · 'aldoaa' · 'aldoab' · 'ank2a' · 'bin1b' · 'chrna1' · 'col11a1b' · 'desma' · 'epc1b' · 'fga' · 'flnca' ·
'fxr1' · 'gapdh' · 'gpd1l' · 'gpib' · 'hdac7b' · 'ilk' · 'jupa' · 'klhl41b' · 'krt94' · 'lamb2' · 'mybpc2a' · 'myh9b' ·
'myl2a' · 'ndc80' · 'p2rx4b' · 'pdlim5a' · 'pfkma' · 'pfkmb' · 'pkmb' · 'postnb' · 'pygmb' · 'scn5lab' · 'sin3b' ·
'slc25a4' · 'smyd1b' · 'synpo2b' · 'tcap' · 'tmem38a' · 'tnnt1' · 'tpm1' · 'tpm2' · 'tpm3' · 'tpm4a' · 'trdn' ·
'twf1b'
```

▼ Code

```
intersect(actin_genes_mouse$gene, muscle_genes_mouse$gene) %>% length
```

19

▼ Code

```

glycolysis_genes_mouse <- description_df_mouse_zf %>% filter(grepl("glycolysis",
  description_df_mouse_zf$GO_name_mouse, ignore.case =
  TRUE)|grepl("glycolysis", description_df_mouse_zf$GO_name_zf, ignore.case =
  TRUE)) %>% filter(adult_ko_vs_adult_wt_P.Value<0.05) %>% unique()##>%
  dplyr::select(gene, gene_mouse, gene_full_name_mouse, GO_name_mouse,
  GO_description_mouse)
glycolysis_genes_mouse %>% pull(gene)    %>% unique() %>% length
glycolysis_genes_mouse %>% pull(gene)    %>% unique()

```

11

```
'aldoaa' · 'aldoab' · 'eno3' · 'gapdh' · 'gpib' · 'pfkma' · 'pfkmb' · 'pfkpb' · 'pgk1' · 'pkmb' · 'tpi1b'
```

▼ Code

```
gluconeogenesis_mouse <- description_df_mouse_zf %>% filter(grepl("gluconeogenesis",  
    description_df_mouse_zf$GO_name_mouse, ignore.case = TRUE)|grepl("gluconeogenesis",  
    description_df_mouse_zf$GO_name_zf, ignore.case = TRUE)) %>%  
    filter(adult_ko_vs_adult_wt_P.Value<0.05) %>% unique()##>%  
    dplyr::select(gene, gene_mouse, gene_full_name_mouse, GO_name_mouse,  
    GO_description_mouse)  
gluconeogenesis_mouse %>% pull(gene)    %>% unique() %>% length  
gluconeogenesis_mouse %>% pull(gene)    %>% unique()
```

10

'g6pc3' · 'gapdh' · 'got1' · 'gpd1a' · 'gpd1b' · 'gpib' · 'mdh1aa' · 'pgk1' · 'ppp4r3b' · 'tpi1b'

▼ Code

```
fatty_genes_mouse <- description_df_mouse_zf %>% filter(grepl("fatty",  
    description_df_mouse_zf$GO_name_mouse, ignore.case = TRUE)|grepl("fatty",  
    description_df_mouse_zf$GO_name_zf, ignore.case = TRUE)) %>%  
    filter(adult_ko_vs_adult_wt_P.Value<0.05)  %>% unique()  
fatty_genes_mouse %>% dplyr::select(gene, gene_mouse, gene_full_name_mouse,  
    GO_name_mouse, GO_description_mouse)  %>% pull(gene)    %>% unique() %>%  
    length  
fatty_genes_mouse %>% pull(gene)    %>% unique()  
# fatty_genes_mouse[grepl("c3a.1", fatty_genes_mouse$gene, ignore.case = TRUE), ]
```

8

'abcd2' · 'aclya' · 'adh8a' · 'adh8b' · 'eci2' · 'etfdh' · 'got1' · 'ptges3b'

▼ Code

```
proteolysis_genes_mouse <- description_df_mouse_zf %>% filter(grepl("proteolysis",  
    description_df_mouse_zf$GO_name_mouse, ignore.case =  
    TRUE)|grepl("proteolysis", description_df_mouse_zf$GO_name_zf, ignore.case =  
    TRUE)) %>% filter(adult_ko_vs_adult_wt_P.Value<0.05)  %>% unique()  
proteolysis_genes_mouse %>% dplyr::select(gene, gene_mouse, gene_full_name_mouse,  
    GO_name_mouse, GO_description_mouse)  %>% pull(gene)    %>% unique() %>%  
    length  
proteolysis_genes_mouse %>% pull(gene)    %>% unique()  
# proteolysis_genes_mouse %>% colnames
```

5

'adgrv1' · 'endou' · 'myh9b' · 'psmb6' · 'tfa'

▼ Code

```
tca_genes_mouse <- description_df_mouse_zf %>% filter(grepl("citric|tca",  
    description_df_mouse_zf$GO_name_mouse, ignore.case =  
    TRUE)|grepl("citric|tca", description_df_mouse_zf$GO_name_zf, ignore.case =  
    TRUE)) %>% filter(adult_ko_vs_adult_wt_P.Value<0.05)  %>% unique()  
tca_genes_mouse %>% dplyr::select(gene, gene_mouse, gene_full_name_mouse,  
    GO_name_mouse, GO_description_mouse)  %>% pull(gene)    %>% unique() %>%  
    length  
tca_genes_mouse %>% pull(gene)    %>% unique()
```

0

▼ Code

```

epigenetic_genes_mouse <- description_df_mouse_zf %>% filter(grepl("chromatin
    remodel|histone|methylation|acetylation",
    description_df_mouse_zf$GO_name_mouse, ignore.case = TRUE)|grepl("chromatin
    remodel|histone|methylation|acetylation",
    description_df_mouse_zf$GO_description_mouse, ignore.case = TRUE)|

grepl("chromatin remodel|histone|methylation|acetylation",
    description_df_mouse_zf$GO_name_zf, ignore.case = TRUE)|grepl("chromatin
    remodel|histone|methylation|acetylation",
    description_df_mouse_zf$GO_description_zf, ignore.case = TRUE)) %>%
filter(adult_ko_vs_adult_wt_P.Value<0.05) %>% unique()
epigenetic_genes_mouse %>% dplyr::select(gene, gene_mouse, gene_full_name_mouse,
    GO_name_mouse, GO_description_mouse) %>% pull(gene) %>% unique() %>%
length
epigenetic_genes_mouse %>% pull(gene) %>% unique()
# epigenetic_genes_mouse %>% colnames

```

21

'apobec2a' · 'bhmt' · 'chd9' · 'ddx39ab' · 'edf1' · 'epc1b' · 'hdac7b' · 'hmga1a' · 'hnrrnpabb' · 'hsp90ab1' ·
'jupa' · 'hdc80' · 'pkmb' · 'prmt2' · 'rcor2' · 'recql' · 'shprh' · 'sin3b' · 'slka' · 'smyd1b' · 'tmf1'

▼ Code

```

one_carbon_genes_mouse <- description_df_mouse_zf %>%
    filter(grepl("tetrahydrofolate|one-carbon|carbonic",
    description_df_mouse_zf$GO_name_mouse, ignore.case =
    TRUE)|grepl("tetrahydrofolate|one-carbon|carbonic",
    description_df_mouse_zf$GO_description_mouse, ignore.case = TRUE)|

grepl("tetrahydrofolate|one-carbon|carbonic", description_df_mouse_zf$GO_name_zf,
    ignore.case = TRUE)|grepl("tetrahydrofolate|one-carbon|carbonic",
    description_df_mouse_zf$GO_description_zf, ignore.case =
    TRUE)|grepl("tetrahydrofolate|one-carbon|carbonic",
    description_df_mouse_zf$gene_full_name_zf, ignore.case =
    TRUE)|grepl("tetrahydrofolate|one-carbon|carbonic",
    description_df_mouse_zf$gene_full_name_mouse, ignore.case = TRUE)

) %>% filter(adult_ko_vs_adult_wt_P.Value<0.05) %>% unique()
one_carbon_genes_mouse %>% pull(gene) %>% unique() %>% length
one_carbon_genes_mouse %>% pull(gene) %>% unique()
# one_carbon_genes_mouse %>% colnames

```

2

'bhmt' · 'ca15b'

▼ Code

```

insulin_genes_mouse <- description_df_mouse_zf %>% filter(grepl("insulin",
    description_df_mouse_zf$GO_name_mouse, ignore.case = TRUE)|

grepl("insulin", description_df_mouse_zf$GO_name_zf, ignore.case =
    TRUE)|grepl("insulin", description_df_mouse_zf$gene_full_name_zf,
    ignore.case = TRUE)|grepl("insulin",
    description_df_mouse_zf$gene_full_name_mouse, ignore.case = TRUE)

) %>% filter(adult_ko_vs_adult_wt_P.Value<0.05) %>% unique()
insulin_genes_mouse %>% pull(gene) %>% unique() %>% length
insulin_genes_mouse %>% pull(gene) %>% unique()
# insulin_genes_mouse %>% colnames

```

'bcar3' · 'col1a1a' · 'col1a1b' · 'dynll1' · 'glud1a' · 'got1' · 'hsp90b1' · 'myh9b' · 'opa1' · 'oxct1a' · 'pfkma' · 'pfkmb' · 'pklr' · 'pkmb'

▼ Code

```
microtubule_genes_mouse <- description_df_mouse_zf %>% filter(grepl("microtubule",
  description_df_mouse_zf$GO_name_mouse, ignore.case = TRUE) |
grepl("microtubule", description_df_mouse_zf$GO_name_zf, ignore.case =
  TRUE)|grepl("microtubule", description_df_mouse_zf$gene_full_name_zf,
  ignore.case = TRUE)|grepl("microtubule",
  description_df_mouse_zf$gene_full_name_mouse, ignore.case = TRUE)
) %>% filter(adult_ko_vs_adult_wt_P.Value<0.05) %>% unique()
microtubule_genes_mouse %>% pull(gene) %>% unique() %>% length
microtubule_genes_mouse %>% pull(gene) %>% unique()
# microtubule_genes_mouse %>% colnames
```

28

'arhgap4a' · 'bin1b' · 'camk2b1' · 'ccdc88c' · 'cct2' · 'cct3' · 'cct7' · 'cfap58' · 'cltcb' · 'dnah3' · 'dynll1' · 'eml3' · 'gapdh' · 'kif13bb' · 'kif18a' · 'klhl4' · 'mns1' · 'nav1b' · 'ndc80' · 'opa1' · 'phldb1a' · 'ppp4r3b' · 'ppp5c' · 'slka' · 'spata7' · 'trdn' · 'tubb4bl' · 'vapb'

▼ Code

```
gpcr_genes_mouse <- description_df_mouse_zf %>% filter(grepl("G Protein-Coupled
  Receptor", description_df_mouse_zf$GO_name_mouse, ignore.case = TRUE) |
grepl("G Protein-Coupled Receptor", description_df_mouse_zf$GO_name_zf, ignore.case =
  TRUE)|grepl("G Protein-Coupled Receptor",
  description_df_mouse_zf$gene_full_name_zf, ignore.case = TRUE)|grepl("G
  Protein-Coupled Receptor", description_df_mouse_zf$gene_full_name_mouse,
  ignore.case = TRUE)
) %>% filter(adult_ko_vs_adult_wt_P.Value<0.05) %>% unique()
gpcr_genes_mouse %>% pull(gene) %>% unique() %>% length
gpcr_genes_mouse %>% pull(gene) %>% unique()
```

4

'adgrv1' · 'camk2b1' · 'myh9b' · 'tas1r2'

▼ Code

```
description_df_mouse_zf[grepl("nme", description_df_mouse_zf$gene, ignore.case =
  TRUE)|grepl("pyrimidine", description_df_mouse_zf$gene_mouse, ignore.case =
  TRUE),] %>% pull(gene) %>% unique()
```

'nme2b'

▼ Code

```
nucleotide_genes_mouse <- description_df_mouse_zf %>%
  filter(grepl("nucleotide|pyrimidine|purine|nucleoside",
  description_df_mouse_zf$GO_name_mouse, ignore.case = TRUE) |
grepl("nucleotide|pyrimidine|purine|nucleoside", description_df_mouse_zf$GO_name_zf,
  ignore.case = TRUE)|grepl("nucleotide|pyrimidine|purine|nucleoside",
  description_df_mouse_zf$gene_full_name_zf, ignore.case =
  TRUE)|grepl("nucleotide|pyrimidine|purine|nucleoside",
  description_df_mouse_zf$gene_full_name_mouse, ignore.case = TRUE)) %>%
  filter(adult_ko_vs_adult_wt_P.Value<0.05) %>% unique()
nucleotide_genes_mouse %>% pull(gene) %>% unique() %>% length
```

```
nucleotide_genes_mouse %>% pull(gene)    %>% unique()
# nucleotide_genes_mouse %>% colnames
```

48

```
'aars1' · 'abcd2' · 'aclya' · 'acta1b' · 'atp1a1a' · 'atp2a1l' · 'bcar3' · 'camk2b1' · 'ccdc88c' · 'cct2' · 'cct3' ·
'cct7' · 'chd9' · 'ddx39ab' · 'ddx59' · 'dhdh' · 'dnah3' · 'eef2b' · 'g3bp1' · 'gapdh' · 'glud1a' · 'hsp90ab1' ·
'hsp90b1' · 'ilk' · 'kif13bb' · 'myh9b' · 'nme2b' · 'opa1' · 'p2rx4b' · 'pank4' · 'pdk1' · 'pebp1' · 'pfkma' ·
'pfkmb' · 'pfkpb' · 'pgk1' · 'pklr' · 'pkmb' · 'pygmb' · 'rab7a' · 'recql' · 'rps6ka1' · 'sbf2' · 'shprh' · 'slc25a4' ·
'slka' · 'top2a' · 'ube2kb'
```

▼ Code

```
biosynthetic_genes_mouse <- description_df_mouse_zf %>% filter(grepl("biosynthetic",
  description_df_mouse_zf$GO_name_mouse, ignore.case = TRUE) |
grepl("biosynthetic", description_df_mouse_zf$GO_name_zf, ignore.case =
  TRUE)|grepl("biosynthetic", description_df_mouse_zf$gene_full_name_zf,
  ignore.case = TRUE)|grepl("biosynthetic",
  description_df_mouse_zf$gene_full_name_mouse, ignore.case = TRUE)) %>%
  filter(adult_ko_vs_adult_wt_P.Value<0.05) %>% unique()
biosynthetic_genes_mouse %>% pull(gene)    %>% unique() %>% length
biosynthetic_genes_mouse %>% pull(gene)    %>% unique()
```

```
# biosynthetic_genes_mouse %>% colnames
```

30

```
'abcd2' · 'aclya' · 'adh8a' · 'adh8b' · 'aldoaa' · 'aldoab' · 'atp5f1d' · 'bhmt' · 'cl tcb' · 'col1a1a' · 'col1a1b' ·
'dynll1' · 'got1' · 'h6pd' · 'hsp90ab1' · 'ldha' · 'mdh1aa' · 'mt-co2' · 'myh9b' · 'ndufc2' · 'nme2b' · 'pank4' ·
'pbldp' · 'pdk1' · 'pklr' · 'pkmb' · 'ptges3b' · 'tfa' · 'tpi1b' · 'ugt8'
```

▼ Code

```
arginine_genes_mouse <- description_df_mouse_zf %>% filter(grepl("arginine",
  description_df_mouse_zf$GO_name_mouse, ignore.case = TRUE) |
grepl("arginine", description_df_mouse_zf$GO_name_zf, ignore.case =
  TRUE)|grepl("arginine", description_df_mouse_zf$gene_full_name_zf,
  ignore.case = TRUE)|grepl("arginine",
  description_df_mouse_zf$gene_full_name_mouse, ignore.case = TRUE)) %>%
  filter(adult_ko_vs_adult_wt_P.Value<0.05) %>% unique()
arginine_genes_mouse %>% pull(gene)    %>% unique() %>% length
arginine_genes_mouse %>% pull(gene)    %>% unique()
```

```
# arginine_genes_mouse %>% colnames
```

1

```
'prmt2'
```

▼ Code

```
description_df_mouse_zf[grepl("ndufa4a", description_df_mouse_zf$gene, ignore.case =
  TRUE)|grepl("ndufa4a", description_df_mouse_zf$gene_mouse, ignore.case =
  TRUE),] ##%>% pull(gene) %>% unique()
```

	gene	gene_mouse	protein	AKO1	AKO2	AKO3
	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
2018	ndufa4a	NA	A0A2R8QLN3	-1.699492725999999	-6.801685516	-6.975045720999

▼ Code

```
NADH_genes_mouse <- description_df_mouse_zf %>% filter(grepl("NADH",
  description_df_mouse_zf$GO_name_mouse, ignore.case = TRUE) |
grepl("NADH", description_df_mouse_zf$GO_name_zf, ignore.case = TRUE)|grepl("NADH",
  description_df_mouse_zf$gene_full_name_zf, ignore.case = TRUE)|grepl("NADH",
  description_df_mouse_zf$gene_full_name_mouse, ignore.case = TRUE)) %>%
  filter(adult_ko_vs_adult_wt_P.Value<0.05) %>% unique()

NADH_genes_mouse %>% pull(gene) %>% unique() %>% length
NADH_genes_mouse %>% pull(gene) %>% unique()

# NADH_genes_mouse %>% colnames
```

8

'gpd1a' · 'gpd1b' · 'gpd1l' · 'mdh1aa' · 'ndufa3' · 'ndufa8' · 'ndufb7' · 'ndufc2'

▼ Code

```
complement_genes_mouse <- description_df_mouse_zf %>% filter(grepl("complement",
  description_df_mouse_zf$GO_name_mouse, ignore.case = TRUE) |
grepl("complement", description_df_mouse_zf$GO_name_zf, ignore.case =
  TRUE)|grepl("complement", description_df_mouse_zf$gene_full_name_zf,
  ignore.case = TRUE)|grepl("complement",
  description_df_mouse_zf$gene_full_name_mouse, ignore.case = TRUE)) %>%
  filter(adult_ko_vs_adult_wt_P.Value<0.05) %>% unique()

complement_genes_mouse %>% pull(gene) %>% unique() %>% length
complement_genes_mouse %>% pull(gene) %>% unique()

# complement_genes_mouse %>% colnames
```

2

'c3a' · 'hsp90ab1'

▼ Code

```
telomere_genes_mouse <- description_df_mouse_zf %>% filter(grepl("telomere",
  description_df_mouse_zf$GO_name_mouse, ignore.case = TRUE) |
grepl("telomere", description_df_mouse_zf$GO_name_zf, ignore.case =
  TRUE)|grepl("telomere", description_df_mouse_zf$gene_full_name_zf,
  ignore.case = TRUE)|grepl("telomere",
  description_df_mouse_zf$gene_full_name_mouse, ignore.case = TRUE)) %>%
  filter(adult_ko_vs_adult_wt_P.Value<0.05) %>% unique()

telomere_genes_mouse %>% pull(gene) %>% unique() %>% length
telomere_genes_mouse %>% pull(gene) %>% unique()
```

```
# telomere_genes_mouse %>% colnames
```

5

```
'cct2' · 'cct3' · 'cct7' · 'hsp90ab1' · 'ptges3b'
```

▼ Code

```
epiderm_genes_mouse <- description_df_mouse_zf %>% filter(grepl("epiderm",
  description_df_mouse_zf$GO_name_mouse, ignore.case = TRUE) |
grepl("epiderm", description_df_mouse_zf$GO_name_zf, ignore.case =
  TRUE)|grepl("epiderm", description_df_mouse_zf$gene_full_name_zf,
  ignore.case = TRUE)|grepl("epiderm",
  description_df_mouse_zf$gene_full_name_mouse, ignore.case = TRUE)) %>%
  filter(adult_ko_vs_adult_wt_P.Value<0.05) %>% unique()
epiderm_genes_mouse %>% pull(gene) %>% unique() %>% length
epiderm_genes_mouse %>% pull(gene) %>% unique()
```

```
# epiderm_genes_mouse %>% colnames
```

4

```
'bcar3' · 'col1a1a' · 'col1a1b' · 'rab7a'
```

▼ Code

```
golgi_genes_mouse <- description_df_mouse_zf %>% filter(grepl("golgi",
  description_df_mouse_zf$GO_name_mouse, ignore.case = TRUE) |
grepl("golgi", description_df_mouse_zf$GO_name_zf, ignore.case = TRUE)|grepl("golgi",
  description_df_mouse_zf$gene_full_name_zf, ignore.case =
  TRUE)|grepl("golgi", description_df_mouse_zf$gene_full_name_mouse,
  ignore.case = TRUE)) %>% filter(adult_ko_vs_adult_wt_P.Value<0.05) %>%
  unique()
golgi_genes_mouse %>% pull(gene) %>% unique() %>% length
golgi_genes_mouse %>% pull(gene) %>% unique()
```

```
# golgi_genes_mouse %>% colnames
```

12

```
'aimp1a' · 'apooa' · 'arhgap4a' · 'cltcb' · 'col1a1a' · 'col1a1b' · 'golga4' · 'pebp1' · 'postnb' · 'rab7a' · 'tmf1' ·
'vepb'
```

▼ Code

```
adhesion_genes_mouse <- description_df_mouse_zf %>% filter(grepl("adhesion",
  description_df_mouse_zf$GO_name_mouse, ignore.case = TRUE) |
grepl("adhesion", description_df_mouse_zf$GO_name_zf, ignore.case =
  TRUE)|grepl("adhesion", description_df_mouse_zf$gene_full_name_zf,
  ignore.case = TRUE)|grepl("adhesion",
  description_df_mouse_zf$gene_full_name_mouse, ignore.case = TRUE)) %>%
  filter(adult_ko_vs_adult_wt_P.Value<0.05) %>% unique()
adhesion_genes_mouse %>% pull(gene) %>% unique() %>% length
adhesion_genes_mouse %>% pull(gene) %>% unique()
```

```
# adhesion_genes_mouse %>% colnames
```

29

```
'adgrv1' · 'bcar3' · 'cd99l2' · 'col12a1a' · 'col12a1b' · 'col1a1a' · 'col1a1b' · 'col6a2' · 'col6a3' · 'fga' · 'fgg' ·  
'igdcc3' · 'ilk' · 'jupa' · 'lamb2' · 'micall2b' · 'mpz' · 'mybpc2a' · 'myh9b' · 'pcdh2ab9' · 'postnb' · 'ppiaa' ·  
'ppiab' · 'rtn4a' · 'serpinb14' · 'slka' · 'synpo2b' · 'tln2a' · 'tpm1'
```

▼ Code

```
mit_genes_mouse <- description_df_mouse_zf[description_df_mouse_zf$gene_mouse %in%  
  mito_genes$Symbol,] %>% filter(adult_ko_vs_adult_wt_P.Value<0.05) %>%  
  unique  
mit_genes_mouse %>% dim  
mit_genes_mouse %>% pull(gene) %>% unique() %>% length  
mit_genes_mouse %>% pull(gene) %>% unique()
```

36 · 32

34

```
'abcd2' · 'acly'a' · 'aco2' · 'aldh4a1' · 'apooa' · 'atp5f1d' · 'chchd10' · 'chchd2' · 'cox5b2' · 'cox6b1' ·  
'cox6b2' · 'cox7a2a' · 'cox7c' · 'dhtkd1' · 'eci2' · 'etfdh' · 'glud1a' · 'hibadh' · 'mrpl12' · 'mt-co2' · 'mtx2' ·  
'ndufa3' · 'ndufa8' · 'ndufb7' · 'ndufc2' · 'opa1' · 'oxct1a' · 'pdk1' · 'prdx3' · 'prodhb' · 'slc25a20' ·  
'slc25a4' · 'snd1' · 'uqcrh'
```

▼ Code

```
calcium_genes_mouse <- description_df_mouse_zf %>% filter(grepl("calcium",  
  description_df_mouse_zf$GO_name_mouse, ignore.case = TRUE) |  
  grepl("calcium", description_df_mouse_zf$GO_name_zf, ignore.case =  
  TRUE) | grepl("calcium", description_df_mouse_zf$gene_full_name_zf,  
  ignore.case = TRUE) | grepl("calcium",  
  description_df_mouse_zf$gene_full_name_mouse, ignore.case = TRUE)) %>%  
  filter(adult_ko_vs_adult_wt_P.Value<0.05) %>% unique()  
calcium_genes_mouse %>% pull(gene) %>% unique() %>% length  
calcium_genes_mouse %>% pull(gene) %>% unique()
```

```
# calcium_genes_mouse %>% colnames
```

29

```
'adgrv1' · 'ank2a' · 'anxa2a' · 'anxa5b' · 'atp2a1l' · 'bin1b' · 'calr3a' · 'camk2b1' · 'desma' · 'fga' · 'fgg' ·  
'got1' · 'myh9b' · 'myl10' · 'myl13' · 'myl2a' · 'myl9b' · 'opa1' · 'p2rx4b' · 'pcdh2ab9' · 'prkcsh' · 'pvalb4' ·  
'pvalb7' · 'pygmb' · 'tmem38a' · 'tnnc1a' · 'trdn' · 'vapb' · 'wfs1a'
```

▼ Code

```
cardiac_genes_mouse <- description_df_mouse_zf %>% filter(grepl("cardiac",  
  description_df_mouse_zf$GO_name_mouse, ignore.case = TRUE) |  
  grepl("cardiac", description_df_mouse_zf$GO_name_zf, ignore.case =  
  TRUE) | grepl("cardiac", description_df_mouse_zf$gene_full_name_zf,  
  ignore.case = TRUE) | grepl("cardiac",  
  description_df_mouse_zf$gene_full_name_mouse, ignore.case = TRUE)) %>%  
  filter(adult_ko_vs_adult_wt_P.Value<0.05) %>% unique()  
cardiac_genes_mouse %>% pull(gene) %>% unique() %>% length
```

```
cardiac_genes_mouse %>% pull(gene) %>% unique()
```

```
# cardiac_genes_mouse %>% colnames
```

22

```
'ank2a' · 'bin1b' · 'col11a1b' · 'desma' · 'gpd1l' · 'hspb1' · 'ilk' · 'jupa' · 'myl2a' · 'p2rx4b' · 'pdlim5a' ·  
'rtn4a' · 'scn5lab' · 'sin3b' · 'slc25a4' · 'smyd1b' · 'tcap' · 'tmem38a' · 'tpm1' · 'tpm4a' · 'trdn' · 'twf1b'
```

▼ Code

```
# write an excel file with all the genes in different tabs  
library(openxlsx)  
wb <- createWorkbook()  
addWorksheet(wb, "mitoribosome")  
writeData(wb, "mitoribosome", mitoribosome_combined_mouse)  
addWorksheet(wb, "cell_cycle")  
writeData(wb, "cell_cycle", cell_cycle_genes_mouse)  
addWorksheet(wb, "ecm")  
writeData(wb, "ecm", ecm_genes_mouse)  
addWorksheet(wb, "actin")  
writeData(wb, "actin", actin_genes_mouse)  
addWorksheet(wb, "muscle")  
writeData(wb, "muscle", muscle_genes_mouse)  
addWorksheet(wb, "glycolysis")  
writeData(wb, "glycolysis", glycolysis_genes_mouse)  
addWorksheet(wb, "gluconeo")  
writeData(wb, "gluconeo", gluconeo_genes_mouse)  
addWorksheet(wb, "fatty")  
writeData(wb, "fatty", fatty_genes_mouse)  
addWorksheet(wb, "proteolysis")  
writeData(wb, "proteolysis", proteolysis_genes_mouse)  
addWorksheet(wb, "tca")  
writeData(wb, "tca", tca_genes_mouse)  
addWorksheet(wb, "epigenetic")  
writeData(wb, "epigenetic", epigenetic_genes_mouse)  
addWorksheet(wb, "one_carbon")  
writeData(wb, "one_carbon", one_carbon_genes_mouse)  
addWorksheet(wb, "insulin")  
writeData(wb, "insulin", insulin_genes_mouse)  
addWorksheet(wb, "microtubule")  
writeData(wb, "microtubule", microtubule_genes_mouse)  
addWorksheet(wb, "gpcr")  
writeData(wb, "gpcr", gpcr_genes_mouse)  
addWorksheet(wb, "nucleotide")  
writeData(wb, "nucleotide", nucleotide_genes_mouse)  
addWorksheet(wb, "biosynthetic")  
writeData(wb, "biosynthetic", biosynthetic_genes_mouse)  
addWorksheet(wb, "arginine")  
writeData(wb, "arginine", arginine_genes_mouse)  
addWorksheet(wb, "NADH")  
writeData(wb, "NADH", NADH_genes_mouse)  
addWorksheet(wb, "complement")  
writeData(wb, "complement", complement_genes_mouse)
```

```

addWorksheet(wb, "telomere")
writeData(wb, "telomere", telomere_genes_mouse)
addWorksheet(wb, "epiderm")
writeData(wb, "epiderm", epiderm_genes_mouse)
addWorksheet(wb, "golgi")
writeData(wb, "golgi", golgi_genes_mouse)
addWorksheet(wb, "adhesion")
writeData(wb, "adhesion", adhesion_genes_mouse)
addWorksheet(wb, "mitocarta")
writeData(wb, "mitocarta", mit_genes_mouse)
addWorksheet(wb, "calcium")
writeData(wb, "calcium", calcium_genes_mouse)
addWorksheet(wb, "cardiac")
writeData(wb, "cardiac", cardiac_genes_mouse)

saveWorkbook(wb,
    "./proteomics_n_3_muscle/annotation_genes_proteomics_n_3_muscle_calcium_cardia
overwrite = TRUE)

```

7 Heatmaps

▼ Code

```

description_df_mouse_adult_ko_vs_adult_wt <- description_df_mouse %>%
  filter(adult_ko_vs_adult_wt_P.Value<0.05) %>%
dplyr::select("gene",'protein','AKO1','AKO2', 'AKO3', 'AWT1', 'AWT2', 'AWT3')
description_df_mouse_adult_ko_vs_adult_wt %>% dim()
description_df_mouse_adult_ko_vs_adult_wt <-
  description_df_mouse_adult_ko_vs_adult_wt %>% distinct()

description_df_mouse_adult_ko_vs_adult_wt$unique_rownames <-
  make.unique(description_df_mouse_adult_ko_vs_adult_wt$gene, sep="_")
description_df_mouse_adult_ko_vs_adult_wt <-description_df_mouse_adult_ko_vs_adult_wt
  %>% dplyr::select("unique_rownames", everything())
description_df_mouse_adult_ko_vs_adult_wt %>% dim()
description_df_mouse_adult_ko_vs_adult_wt[duplicated(description_df_mouse_adult_ko_vs_a
= TRUE),]

```

389 · 8

346 · 9

unique_rownames	gene	protein	AKO1	A
<chr>	<chr>	<chr>	<chr>	<
46 cct2	cct2	A0A1L1QZG0	-1.1897050179999999	-2
47 cct2_1	cct2	A0A0G2KUJ0	-0.16838325100000001	-1
66 col6a3	col6a3	F1Q4X1	2.2490576189999998	2

unique_rownames	gene	protein	AKO1	A
<chr>	<chr>	<chr>	<chr>	<
67 col6a3_1	col6a3	F1QKE8	3.754599029	5
187 obscnb	obschnb	A0A140LFP6	-1.0027332579999999	-1
188 obscnb_1	obschnb	X1WFS0	-1.531034577	-1
189 obscnb_2	obschnb	A0A140LGX4	-2.4297810740000001	-1
190 obscnb_3	obschnb	A0A140LFP0	-2.0188504140000001	-1
195 ORFNames=zgc:162509 {ECO:0000313 ZFIN:ZDB-GENE-050809-100};	ORFNames=zgc:162509 {ECO:0000313 ZFIN:ZDB-GENE-050809-100};	F1QY25	1.986491735	3
196 ORFNames=zgc:162509 {ECO:0000313 ZFIN:ZDB-GENE-050809-100};_1	ORFNames=zgc:162509 {ECO:0000313 ZFIN:ZDB-GENE-050809-100};	F1QNU2	-0.20985749400000001	0
200 ORFNames=zgc:66156 {ECO:0000313 ZFIN:ZDB-GENE-040426-1554};	ORFNames=zgc:66156 {ECO:0000313 ZFIN:ZDB-GENE-040426-1554};	F1Q9G9	1.8547040880000001	-0
201 ORFNames=zgc:66156 {ECO:0000313 ZFIN:ZDB-GENE-040426-1554};_1	ORFNames=zgc:66156 {ECO:0000313 ZFIN:ZDB-GENE-040426-1554};	A0A0R4IHN7	5.8871461E-2	-0
329 ttn.2	ttn.2	B0S757	-0.72246685499999996	-2
330 ttn.2_1	ttn.2	B0S758	-0.7310128719999998	-2
331 ttn.2_2	ttn.2	F1QZQ0	-2.9815558809999998	-8
332 ttn.2_3	ttn.2	F1R7N8	-3.2637919009999998	-8

▼ Code

```

adult_ko_vs_adult_wt_mat <- description_df_mouse_adult_ko_vs_adult_wt
rownames(adult_ko_vs_adult_wt_mat) <-
  description_df_mouse_adult_ko_vs_adult_wt$unique_rownames
adult_ko_vs_adult_wt_mat <- adult_ko_vs_adult_wt_mat %>% dplyr::select(-
  unique_rownames, -gene, -protein)
# make all columns numeric
adult_ko_vs_adult_wt_mat <- adult_ko_vs_adult_wt_mat %>% mutate_all(as.numeric)
adult_ko_vs_adult_wt_mat <- as.matrix(adult_ko_vs_adult_wt_mat)
adult_ko_vs_adult_wt_mat %>% head

```

A matrix: 6 × 6 of type dbl

	AKO1	AKO2	AKO3	AWT1	AWT2	AWT3
aars1	-1.3534255	-1.57466334	-1.79473520	-2.8319766	-2.5420032	-2.68005827

	AKO1	AKO2	AKO3	AWT1	AWT2	AWT3
abcb11a	0.6139335	2.17460578	1.22807794	0.5492366	-5.5627694	-2.23317459
abcd2	1.9115116	4.89578943	1.93462670	1.7566192	0.8851672	-0.21348279
aclya	-0.7835549	-0.02263724	-0.04294958	-1.9327479	-1.1844248	-2.04807657
aco2	3.3997626	2.33558737	3.02802187	1.5014695	-0.1329592	1.77908961
acta1b	0.7253859	1.07200672	0.92540279	-0.5457693	0.1424129	0.08422017

▼ Code

```
heatmap_list <- read.xlsx("muscle_for_heat_map.xlsx", sheet = "Combined", colNames =
    TRUE, detectDates = FALSE, skipEmptyRows = TRUE)
heatmap_list$NOP[is.na(heatmap_list$NOP)] <- 0
# Make the 1st letter of each word in the column OXPHOS.Complex capital without using
# stringr::str_to_title
heatmap_list$OXPHOS.Complex <-
    sapply(strsplit(as.character(heatmap_list$OXPHOS.Complex), " "), function(x)
        paste(toupper(substring(x, 1, 1)), substring(x, 2), sep="", collapse=" "))

heatmap_list %>% head
heatmap_list %>% dim
```

A data.frame: 6

Protein	AKO1	AKO2	AKO3	AWT1	AWT2
	<chr>	<chr>	<chr>	<chr>	<chr>
1	ndufa3	1.1054797	0.012593261	1.137783118	-0.326734128999999
2	ndufa8	1.594717603	1.501586096	0.150963366999999	0.621726396999999
3	ndufb7	1.856295654	-0.061710999	-1.235054364	-1.317484943
4	ndufc2	-0.99685305	-1.17236300099999	-0.994636413	0.567605263
5	uqcrh	-2.320787564	1.053185371	-0.206646588999999	2.29753711399999
6	cox5b2	-1.239462898	-4.12699089799999	-4.23119880099999	1.29610152599999

204 · 11

▼ Code

```
# if a protein is duplicated in a OXPHOS.Complex, select the one with the highest NOP
heatmap_list$OXPHOS.Complex %>% unique
heatmap_list$OXPHOS.Complex %>% unique %>% length
heatmap_list_unique <- heatmap_list %>% group_by(OXPHOS.Complex, Protein) %>%
    filter(NOP == max(NOP)) %>% ungroup()
# if two proteins are duplicated in a OXPHOS.Complex still and have same NOP then
# select the top one
heatmap_list_unique <- heatmap_list_unique %>% group_by(OXPHOS.Complex, Protein) %>%
    filter(adult_ko_vs_adult_wt_logFC == max(adult_ko_vs_adult_wt_logFC)) %>%
    ungroup()
```

```

# if still two proteins are duplicated in a OXPHOS.Complex, then select the first one
heatmap_list_unique <- heatmap_list_unique %>% group_by(OXPHOS.Complex, Protein) %>%
  filter(row_number() == 1) %>% ungroup()
heatmap_list_unique$OXPHOS.Complex %>% unique# %>% dim
heatmap_list_unique$OXPHOS.Complex %>% unique %>% length
heatmap_list_unique[duplicated(heatmap_list_unique$Protein)|duplicated(heatmap_list_unique$OXPHOS.Complex, = TRUE),] %>% dim
# arrange by first alphabetically OXPHOS.Complexs then by log2fc
heatmap_list_unique <- heatmap_list_unique %>% arrange(OXPHOS.Complex,
  adult_ko_vs_adult_wt_logFC)

heatmap_list_unique$unique_rownames <-
  make.unique(heatmap_list_unique$Protein, sep="_")
heatmap_list_unique <- heatmap_list_unique %>% dplyr::select("unique_rownames",
  everything())
heatmap_list_unique %>% head
# make a matrix with the unique proteins
heatmap_mat <- heatmap_list_unique %>% as.data.frame()
rownames(heatmap_mat) <- heatmap_list_unique$unique_rownames
heatmap_mat <- heatmap_mat %>% dplyr::select(-unique_rownames, -Protein, -
  OXPHOS.Complex, -NOP, -adult_ko_vs_adult_wt_logFC, -Label)
# make all columns numeric
heatmap_mat <- heatmap_mat %>% mutate_all(as.numeric)
heatmap_mat <- heatmap_mat %>% dplyr::select(contains("WT"), contains("KO"))
heatmap_mat <- as.matrix(heatmap_mat)
# make columns WT and then KO

heatmap_mat %>% head

```

'Complex I' · 'Complex III' · 'Complex IV' · 'Complex V' · 'BCAA Metabolism' · 'Glu Metabolism' ·
 'Lys Metabolism' · 'Pro Metabolism' · 'Keton Metabolism' · 'Pyr Metabolism' · 'TCA' · 'ROS/Gluthathione' ·
 'Fatty Acid Oxidation' · 'Nucleotide Metabolism' · 'Mitoribosome' · 'Apoptosis' · 'Cristae Formation' ·
 'Fusion' · 'Protein Import' · 'ABC Transporter' · 'Glycolysis' · 'Gluconeogenesis' · 'Fatty Acid Metabolism' ·
 'Methionin Biosynthesis/1C' · 'Valin Catabolism' · '1C' · 'NADH Pathway' · 'Proteolysis' ·
 'Extra Cellular Membrane' · 'Cell Cycle' · 'Calcium Signalling/Related'

31

'Complex I' · 'Complex III' · 'Complex IV' · 'Complex V' · 'BCAA Metabolism' · 'Glu Metabolism' · ▶
 'Lys Metabolism' · 'Pro Metabolism' · 'Keton Metabolism' · 'Pyr Metabolism' · 'TCA' · 'ROS/Gluthathione' ·
 'Fatty Acid Oxidation' · 'Nucleotide Metabolism' · 'Mitoribosome' · 'Apoptosis' · 'Cristae Formation' ·
 'Fusion' · 'Protein Import' · 'ABC Transporter' · 'Glycolysis' · 'Gluconeogenesis' · 'Fatty Acid Metabolism' ·
 'Methionin Biosynthesis/1C' · 'Valin Catabolism' · '1C' · 'NADH Pathway' · 'Proteolysis' ·
 'Extra Cellular Membrane' · 'Cell Cycle' · 'Calcium Signalling/Related'

31

73 · 11

unique_rownames	Protein	AKO1	AKO2	AKO3
<chr>	<chr>	<chr>	<chr>	<chr>
ca15b	ca15b	-3.042537333999999	-3.684703129999999	-4.330819399000000

unique_rownames	Protein	AKO1	AKO2	AKO3
<chr>	<chr>	<chr>	<chr>	<chr>
abcd2	abcd2	1.91151158	4.8957894270000004	1.9346266990000001
chchd2	chchd2	-0.8041605240000004	-1.2628537980000001	-1.513433253000000
hibadhb	hibadhb	-0.6264784599999996	-1.103564789	-1.596652081
vapb	vapb	0.2159768039999999	-2.929394999999999E-2	-6.816005299999999
myl9b	myl9b	-1.567959699	-0.8856575479999996	-1.105193684999999

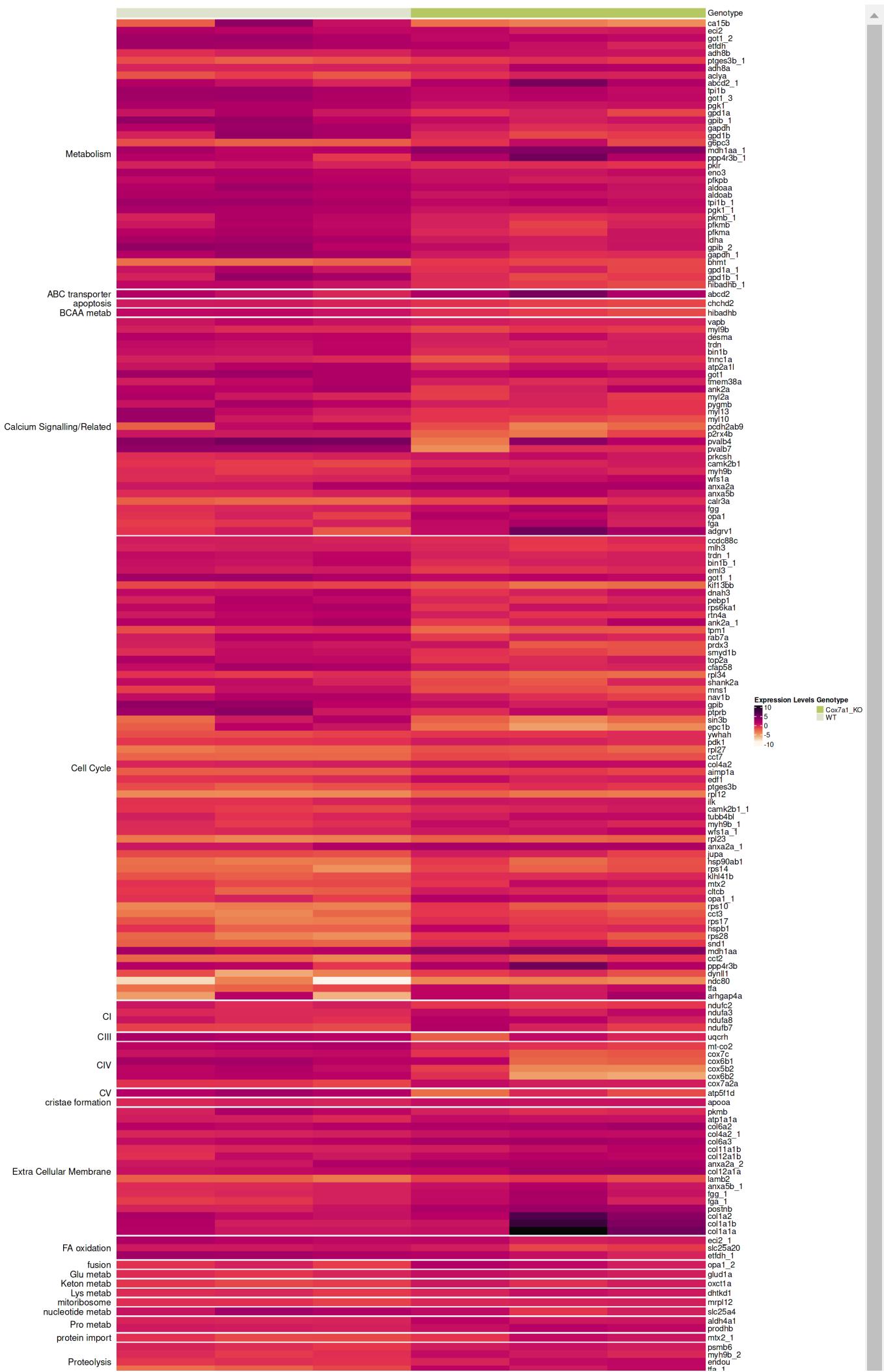
A matrix: 6 × 6 of type dbl

	AWT1	AWT2	AWT3	AKO1	AKO2	AKO3
ca15b	-2.3226962	3.6440017	0.8211058	-3.0425373	-3.68470313	-4.33081940
abcd2	1.7566192	0.8851672	-0.2134828	1.9115116	4.89578943	1.93462670
chchd2	0.3553867	0.3861939	-0.1166660	-0.8041605	-1.26285380	-1.51343325 ▶
hibadhb	1.1568960	1.2233540	0.7122335	-0.6264785	-1.10356479	-1.59665208
vapb	0.6340176	1.2565735	0.7602536	0.2159768	-0.02929395	-0.06816005
myl9b	0.1093725	-0.5347866	-0.4617509	-1.5679597	-0.88565755	-1.10519368

▼ Code

```
# make bigger graph
options(repr.plot.width=18, repr.plot.height=30)
ha = HeatmapAnnotation(
  Genotype = rep(c("WT", "Cox7a1_KO" ), each=3),
  col = list(Genotype = c(
    "WT"=paletteer::paletteer_d("calecopal::arbutus", n=2),
    [1],
    "Cox7a1_KO"=paletteer::paletteer_d("calecopal::arbutus",
    n=2)[2]))
  )
Heatmap(heatmap_mat, name="Expression Levels",
  cluster_columns = FALSE,
  cluster_rows = FALSE,
  row_dend_width = unit(0, "cm"),
# col = viridis::viridis(100, direction = -1),
col= paletteer::paletteer_c("grDevices::Rocket", n=100, direction=-1),
show_column_names = FALSE,
top_annotation = ha,
row_split = factor(heatmap_list_unique$Label, levels =
  unique(heatmap_list_unique$Label)),
# order the splits by the order of the labels
```

```
# row_split_order = factor(heatmap_list_unique$Label),  
row_title_rot = 0,  
# turn off the dendrogram  
  
bottom_annotation = ha)
```



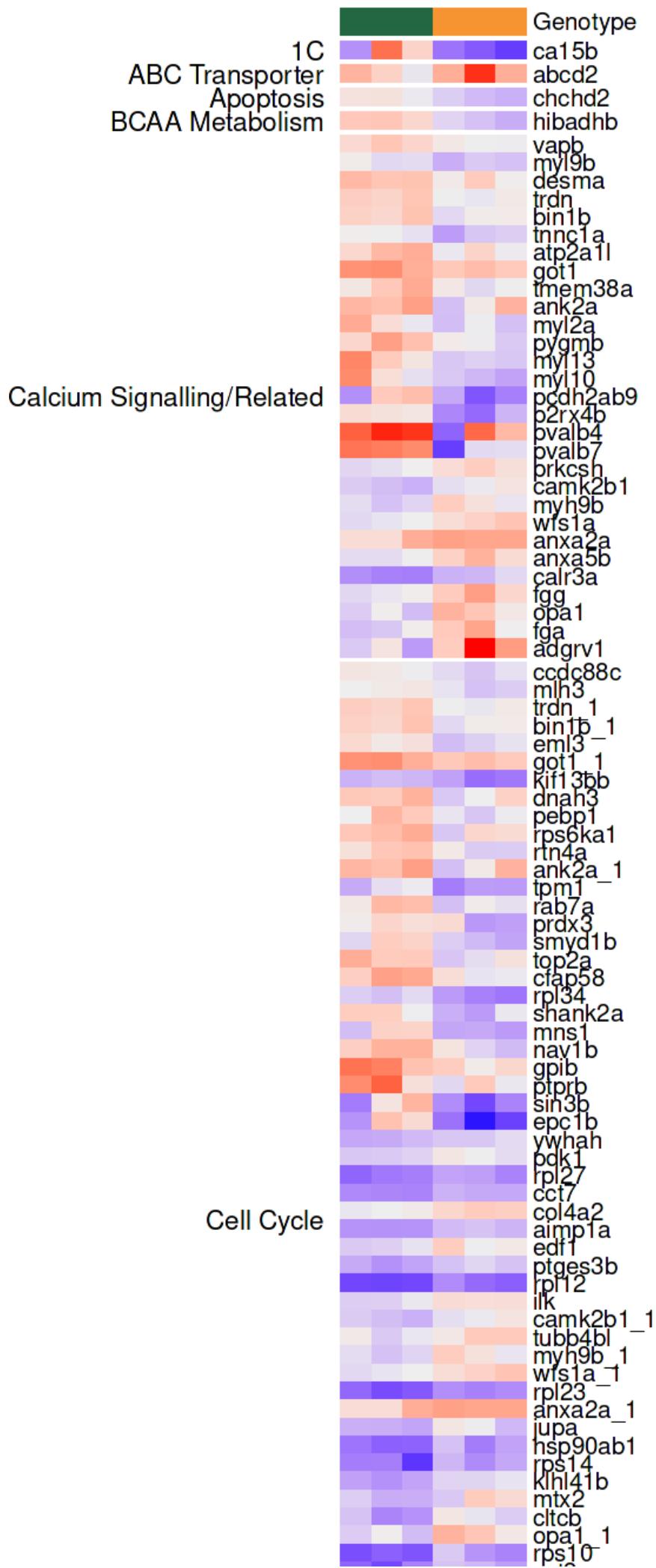


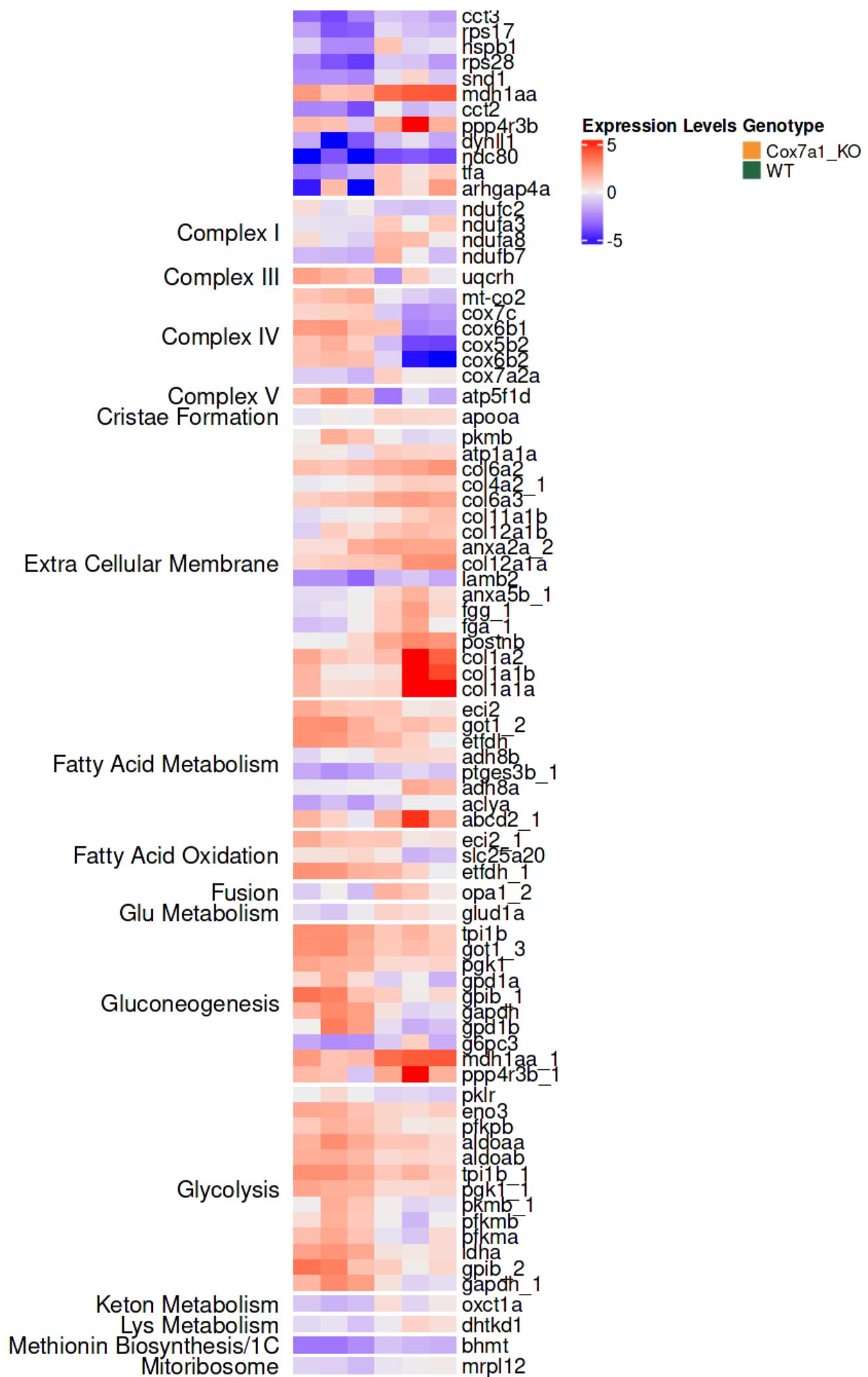
▼ Code

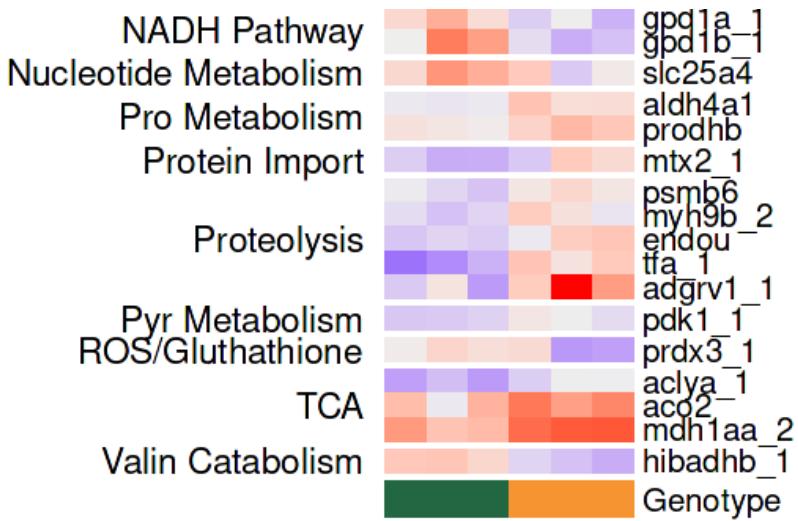
```
# make bigger graph
options(repr.plot.width=7, repr.plot.height=25)
ha = HeatmapAnnotation(
  Genotype = rep(c("WT", "Cox7a1_KO"), each=3),
  col = list(Genotype = c(
    "WT"="#226741",
    "Cox7a1_KO"="#f69431"
  )))
Heatmap(heatmap_mat, name="Expression Levels",
  cluster_columns = FALSE,
  cluster_rows = FALSE,
  row_dend_width = unit(0, "cm"),
# col = viridis::viridis(100, direction = -1),
# col= paletteer::paletteer_c("grDevices::Rocket", n=100, direction=-1),
  show_column_names = FALSE,
  top_annotation = ha,
  row_split = factor(heatmap_list_unique$OXPHOS.Complex, levels =
    unique(heatmap_list_unique$OXPHOS.Complex)),
  row_title_rot = 0,
  bottom_annotation = ha)
dev.copy(pdf, file = "proteomics_n_3_muscle/muscle_heatmap_annotated.pdf", width = 7,
         height = 25)
dev.off()
```

pdf: 3

png: 2







▼ Code

```
heatmap_list_v2 <- read.xlsx("muscle for heat map_v2.xlsx", sheet = "Mitocarta",
                             colNames = TRUE, detectDates = FALSE, skipEmptyRows = TRUE)
heatmap_list_v2$NOP[is.na(heatmap_list_v2$NOP)] <- 0
# Make the 1st letter of each word in the column OXPHOS.Complex capital without using
# stringr::str_to_title
heatmap_list_v2$OXPHOS.Complex <-
  sapply(strsplit(as.character(heatmap_list_v2$OXPHOS.Complex), " "),
        function(x) paste(toupper(substring(x, 1, 1)), substring(x, 2), sep = "", collapse = " "))

heatmap_list_v2 %>% head
heatmap_list_v2 %>% dim
```

A data.frame: 6

	Protein	AKO1	AKO2	AKO3	AWT1	AWT2
	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
1	ndufa3	1.1054797	0.012593261	1.137783118	-0.326734128999999	-0.41951
2	ndufa8	1.594717603	1.501586096	0.150963366999999	0.621726396999999	-0.37613
3	ndufb7	1.856295654	-0.061710999	-1.235054364	-1.317484943	-1.36181
4	ndufc2	-0.99685305	-1.17236300099999	-0.994636413	0.567605263	-0.48391
5	uqcrh	-2.320787564	1.053185371	-0.206646588999999	2.29753711399999	1.78187
6	cox5b2	-1.239462898	-4.12699089799999	-4.23119880099999	1.29610152599999	1.90180

35 · 11

▼ Code

```
# if a protein is duplicated in a OXPHOS.Complex, select the one with the highest NOP
heatmap_list_v2$OXPHOS.Complex %>% unique
heatmap_list_v2$OXPHOS.Complex %>% unique %>% length
heatmap_list_v2_unique <- heatmap_list_v2 %>% group_by(OXPHOS.Complex, Protein) %>%
  filter(NOP == max(NOP)) %>% ungroup()
```

```

# if two proteins are duplicated in a OXPHOS.Complex still and have same NOP then
# select the top one
heatmap_list_v2_unique <- heatmap_list_v2_unique %>% group_by(OXPHOS.Complex,
  Protein) %>% filter(adult_ko_vs_adult_wt_logFC ==
  max(adult_ko_vs_adult_wt_logFC)) %>% ungroup()
# if still two proteins are duplicated in a OXPHOS.Complex, then select the first one
heatmap_list_v2_unique <- heatmap_list_v2_unique %>% group_by(OXPHOS.Complex,
  Protein) %>% filter(row_number() == 1) %>% ungroup()
order_heatmap <- heatmap_list_v2_unique$OXPHOS.Complex %>% unique# %>% dim
unique(heatmap_list_v2_unique$OXPHOS.Complex)
heatmap_list_v2_unique$OXPHOS.Complex %>% unique %>% length
heatmap_list_v2_unique[duplicated(heatmap_list_v2_unique$Protein)|duplicated(heatmap_l:
= TRUE),] %>% dim

# arrange by first alphabetically OXPHOS.Complex as in order_heatmap and then by
# log2fc
heatmap_list_v2_unique <- heatmap_list_v2_unique %>% arrange(factor(OXPHOS.Complex,
  levels = order_heatmap), adult_ko_vs_adult_wt_logFC)

# heatmap_list_v2_unique <- heatmap_list_v2_unique %>%
#   arrange(adult_ko_vs_adult_wt_logFC)

heatmap_list_v2_unique$unique_rownames <-
  make.unique(heatmap_list_v2_unique$Protein, sep="_")
heatmap_list_v2_unique <- heatmap_list_v2_unique %>% dplyr::select("unique_rownames",
  everything())
heatmap_list_v2_unique %>% head
# make a matrix with the unique proteins
heatmap_mat_v2 <- heatmap_list_v2_unique %>% as.data.frame()
rownames(heatmap_mat_v2) <- heatmap_list_v2_unique$unique_rownames
heatmap_mat_v2 <- heatmap_mat_v2 %>% dplyr::select(-unique_rownames, -Protein, -
  OXPHOS.Complex, -NOP, -adult_ko_vs_adult_wt_logFC, -Label)

# make all columns numeric
heatmap_mat_v2 <- heatmap_mat_v2 %>% mutate_all(as.numeric)
heatmap_mat_v2 <- heatmap_mat_v2 %>% dplyr::select(contains("WT"), contains("KO"))
heatmap_mat_v2 <- as.matrix(heatmap_mat_v2)
# make columns WT and then KO

# heatmap_mat_v2 %>% head

```

'Complex I' · 'Complex III' · 'Complex IV' · 'Complex V' · 'BCAA Metabolism' · 'Glu Metabolism' ·
 'Lys Metabolism' · 'Pro Metabolism' · 'Ketone Metabolism' · 'Pyr Metabolism' · 'TCA' ·
 'ROS/Gluthathione' · 'Fatty Acid Oxidation' · 'Nucleotide Metabolism' · 'Mitoribosome' · 'Apoptosis' ·
 'Cristae Formation' · 'Fusion' · 'Protein Import' · 'ABC Transporter'

'Complex I' · 'Complex III' · 'Complex IV' · 'Complex V' · 'BCAA Metabolism' · 'Glu Metabolism' ·
'Lys Metabolism' · 'Pro Metabolism' · 'Ketone Metabolism' · 'Pyr Metabolism' · 'TCA' ·
'ROS/Gluthathione' · 'Fatty Acid Oxidation' · 'Nucleotide Metabolism' · 'Mitoribosome' · 'Apoptosis' ·
'Cristae Formation' · 'Fusion' · 'Protein Import' · 'ABC Transporter'

20

0 · 11

unique_rownames	Protein	AKO1	AKO2	AKO3	AWT1
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
ndufc2	ndufc2	-0.99685305	-1.17236300099999	-0.994636413	0.56760
ndufa3	ndufa3	1.1054797	0.012593261	1.137783118	-0.3267
ndufa8	ndufa8	1.594717603	1.501586096	0.150963366999999	0.62172
ndufb7	ndufb7	1.856295654	-0.061710999	-1.235054364	-1.3174
uqcrh	uqcrh	-2.320787564	1.053185371	-0.206646588999999	2.29753
cox7a1	cox7a1	-4.97990809499999	-5.99686505499999	-6.519969016	0.85754

◀ ▶

▼ Code

```
unique(heatmap_list_v2_unique$OXPHOS.Complex)
```

'Complex I' · 'Complex III' · 'Complex IV' · 'Complex V' · 'BCAA Metabolism' · 'Glu Metabolism' ·
'Lys Metabolism' · 'Pro Metabolism' · 'Ketone Metabolism' · 'Pyr Metabolism' · 'TCA' ·
'ROS/Gluthathione' · 'Fatty Acid Oxidation' · 'Nucleotide Metabolism' · 'Mitoribosome' · 'Apoptosis' ·
'Cristae Formation' · 'Fusion' · 'Protein Import' · 'ABC Transporter'

▼ Code

```
# make bigger graph
options(repr.plot.width=6, repr.plot.height=8)
ha = HeatmapAnnotation(
  Genotype = rep(c("WT", "Cox7a1_KO"), each=3),
  col = list(Genotype = c(
    # "WT"=paletteer::paletteer_d("calecopal::arbutus", n=2)[1],
    "WT"="#226741",
    #
    "Cox7a1_KO"=paletteer::paletteer_d("calecopal::arbutus", n=2)[2])
    "Cox7a1_KO"="#f69431"))

),
# change font size of the annotation
annotation_name_gp = grid::gpar(fontsize = 4),
annotation_legend_param = list(
# title = "Expression Levels",
title_gp = gpar(fontsize = 4),
labels_gp = gpar(fontsize = 4),
grid_width = unit(5,"point"),
grid_height = unit(5,"point")),
```

```

height=unit(0, "point")
)

Heatmap(heatmap_mat_v2, name="Expression Levels",
cluster_columns = FALSE,
cluster_rows = FALSE,
row_dend_width = unit(0, "point"),

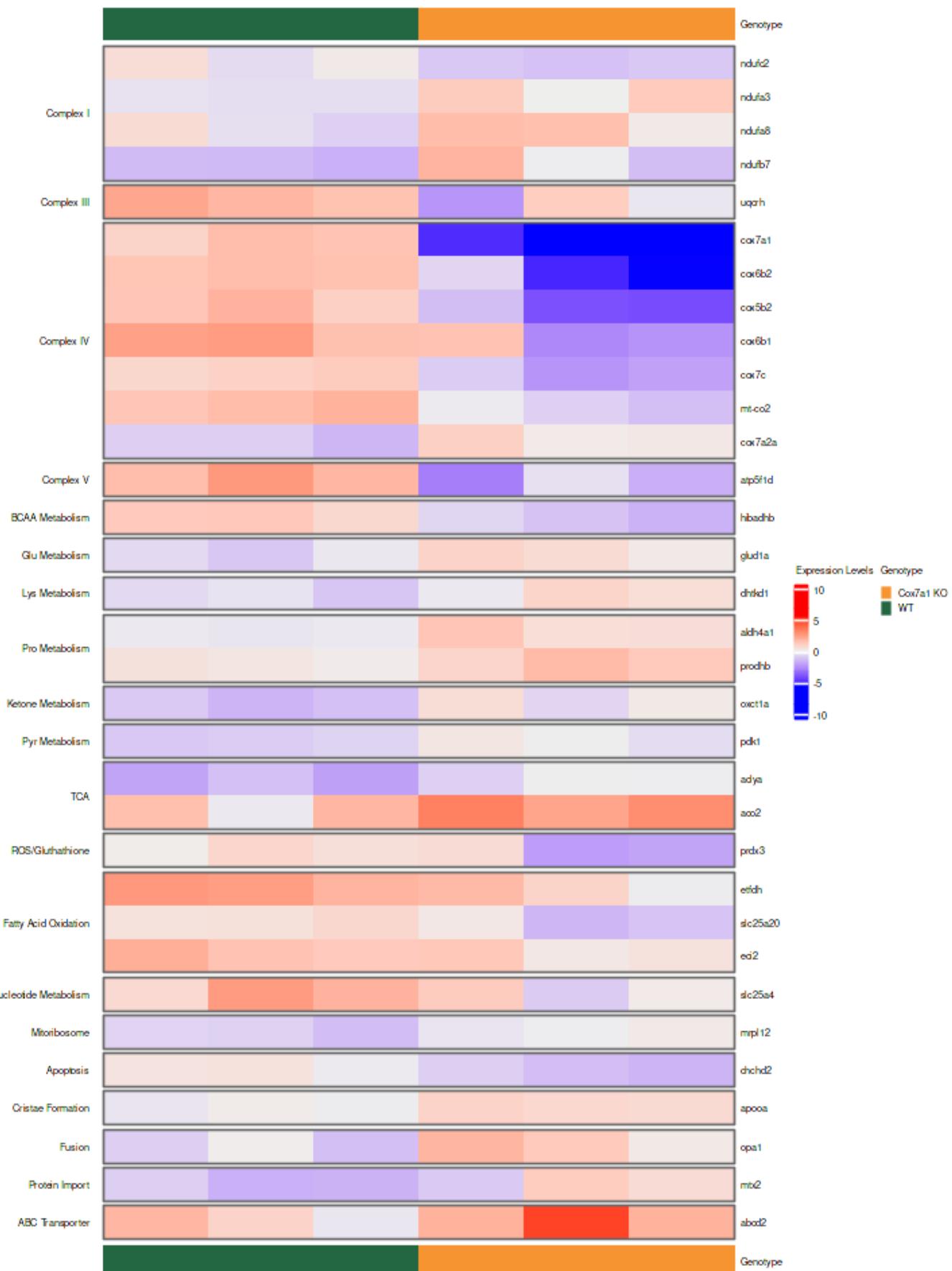
show_column_names = FALSE,
top_annotation = ha,
row_split = factor(heatmap_list_v2_unique$OXPHOS.Complex, levels =
  unique(heatmap_list_v2_unique$OXPHOS.Complex)),
gap=unit(2, "point"),
border= TRUE,
border_gp = gpar(col = "grey40"),
row_title_gp = gpar(fontsize = 4), # change row split font size
row_title_rot = 0,
bottom_annotation = ha,
heatmap_legend_param = list(
  title = "Expression Levels",
  title_gp = gpar(fontsize = 4),
  labels_gp = gpar(fontsize = 4),
  grid_width = unit(5, "point"),
  grid_height = unit(5, "point")
),
row_names_gp = grid::gpar(fontsize = 4))

dev.copy(pdf, file =
  "proteomics_n_3_muscle/muscle_heatmap_annotated_v5_mitocarta.pdf", width =
  2.5, height = 3)
dev.off()

```

pdf: 3

png: 2



▼ Code

```

heatmap_list_v2_others <- read.xlsx("muscle for heat map_v2.xlsx", sheet = "other",
  colNames = TRUE, detectDates = FALSE, skipEmptyRows = TRUE)
heatmap_list_v2_others$NOP[is.na(heatmap_list_v2_others$NOP)] <- 0
# rename column X10 to Oxphos.Complex
  
```

```

heatmap_list_v2_others <- heatmap_list_v2_others %>% dplyr::rename(OXPHOS.Complex =
  X10)
# Make the 1st letter of each word in the column OXPHOS.Complex capital without using
# stringr::str_to_title
heatmap_list_v2_others$OXPHOS.Complex <-
  sapply(strsplit(as.character(heatmap_list_v2_others$OXPHOS.Complex), " "),
    function(x) paste(toupper(substring(x, 1, 1)), substring(x, 2), sep="", collapse=" "))

heatmap_list_v2_others %>% head
heatmap_list_v2_others %>% dim

```

Protein AKO1		AKO2	AKO3	AWT1
<chr>	<chr>	<chr>	<chr>	<chr>
1 ca15b	-3.0425373339999999	-3.6847031299999999	-4.3308193990000001	-2.322696168991
2 prodhb	0.825963545	1.645838796	1.194595321	0.416171817
3 got1	1.1399147060000001	1.514717678	1.119664945	2.738369061999
4 eci2	1.2319295400000001	0.23109229000000001	0.3979523119999997	2.045460442
5 eci2	1.2319295400000001	0.23109229000000001	0.3979523119999997	2.045460442
6 adh8b	0.807751412	0.7574062550000003	0.6550960719999997	-0.633961589991

151 · 11

▼ Code

```

# if a protein is duplicated in a OXPHOS.Complex, select the one with the highest NOP
heatmap_list_v2_others$OXPHOS.Complex %>% unique

heatmap_list_v2_others$OXPHOS.Complex %>% unique %>% length
heatmap_list_v2_others_unique <- heatmap_list_v2_others %>% group_by(OXPHOS.Complex,
  Protein) %>% filter(NOP == max(NOP)) %>% ungroup()
# if two proteins are duplicated in a OXPHOS.Complex still and have same NOP then
# select the top one
heatmap_list_v2_others_unique <- heatmap_list_v2_others_unique %>%
  group_by(OXPHOS.Complex, Protein) %>% filter(adult_ko_vs_adult_wt_logFC ==
    max(adult_ko_vs_adult_wt_logFC)) %>% ungroup()
# if still two proteins are duplicated in a OXPHOS.Complex, then select the first one
heatmap_list_v2_others_unique <- heatmap_list_v2_others_unique %>%
  group_by(OXPHOS.Complex, Protein) %>% filter(row_number() == 1) %>%
  ungroup()
order_heatmap <- heatmap_list_v2_others_unique$OXPHOS.Complex %>% unique# %>% dim

```

```

unique(heatmap_list_v2_others_unique$OXPHOS.Complex)
heatmap_list_v2_others_unique$OXPHOS.Complex %>% unique %>% length
heatmap_list_v2_others_unique[duplicated(heatmap_list_v2_others_unique$Protein) | duplicated(heatmap_list_v2_others_unique$OXPHOS.Complex, TRUE), ] %>% dim

# arrange by first alphabetically OXPHOS.Complex as in order_heatmap and then by log2fc
heatmap_list_v2_others_unique <- heatmap_list_v2_others_unique %>%
  arrange(factor(OXPHOS.Complex, levels = order_heatmap),
         adult_ko_vs_adult_wt_logFC)

# heatmap_list_v2_others_unique <- heatmap_list_v2_others_unique %>%
#   arrange(adult_ko_vs_adult_wt_logFC)

heatmap_list_v2_others_unique$unique_rownames <-
  make.unique(heatmap_list_v2_others_unique$Protein, sep = "_")
heatmap_list_v2_others_unique <- heatmap_list_v2_others_unique %>%
  dplyr::select("unique_rownames", everything())
heatmap_list_v2_others_unique %>% head
# make a matrix with the unique proteins
heatmap_mat_v2_others <- heatmap_list_v2_others_unique %>% as.data.frame()
rownames(heatmap_mat_v2_others) <- heatmap_list_v2_others_unique$unique_rownames
heatmap_mat_v2_others <- heatmap_mat_v2_others %>% dplyr::select(-unique_rownames, -Protein, -OXPHOS.Complex, -NOP, -adult_ko_vs_adult_wt_logFC, -Label)

# make all columns numeric
heatmap_mat_v2_others <- heatmap_mat_v2_others %>% mutate_all(as.numeric)
heatmap_mat_v2_others <- heatmap_mat_v2_others %>% dplyr::select(contains("WT"), contains("KO"))
heatmap_mat_v2_others <- as.matrix(heatmap_mat_v2_others)
# make columns WT and then KO

# heatmap_mat_v2_others %>% head

```

'Methionine Biosynthesis/1 Carbon Cycle' · 'Amino Acid Metabolism' · 'Fatty Acid Metabolism' · 'Gluconeogenesis' · 'Glycogenolysis' · 'Glycolysis' · 'NADH Pathway' · 'TCA' · 'ECM' · 'Muscle' · 'Proteosomal Degradation/Proteolysis' · 'Calcium Signaling And Related'

12

'Methionine Biosynthesis/1 Carbon Cycle' · 'Amino Acid Metabolism' · 'Fatty Acid Metabolism' · 'Gluconeogenesis' · 'Glycogenolysis' · 'Glycolysis' · 'NADH Pathway' · 'TCA' · 'ECM' · 'Muscle' · 'Proteosomal Degradation/Proteolysis' · 'Calcium Signaling And Related'

12

66 · 11

unique_rownames	Protein	AKO1	AKO2	AKO3
<chr>	<chr>	<chr>	<chr>	<chr>
ca15b	ca15b	-3.042537333999999	-3.684703129999999	-4.3308193990000001
bhmt	bhmt	-1.0651798960000001	-1.4252517730000001	-1.546969397
got1	got1	1.1399147060000001	1.514717678	1.119664945
hibadhb	hibadhb	-0.6264784599999996	-1.103564789	-1.596652081
prodhb	prodhb	0.825963545	1.645838796	1.194595321
eci2	eci2	1.2319295400000001	0.23109229000000001	0.3979523119999997

▼ Code

```
unique(heatmap_list_v2_unique$OXPHOS.Complex)
```

'Complex I' · 'Complex III' · 'Complex IV' · 'Complex V' · 'BCAA Metabolism' · 'Glu Metabolism' · 'Lys Metabolism' · 'Pro Metabolism' · 'Ketone Metabolism' · 'Pyr Metabolism' · 'TCA' · 'ROS/Gluthathione' · 'Fatty Acid Oxidation' · 'Nucleotide Metabolism' · 'Mitoribosome' · 'Apoptosis' · 'Cristae Formation' · 'Fusion' · 'Protein Import' · 'ABC Transporter'

▼ Code

```
# make bigger graph
options(repr.plot.width=6, repr.plot.height=8)
ha = HeatmapAnnotation(
  Genotype = rep(c("WT", "Cox7a1_KO"), each=3),
  col = list(Genotype = c(
    # "WT"=paletteer::paletteer_d("calecopal::arbutus", n=2)[1],
    "WT"="#226741",
    #
    "Cox7a1_KO"=paletteer::paletteer_d("calecopal::arbutus", n=2)[2])
    "Cox7a1_KO"="#f69431"))

  ),
  # change font size of the annotation
  annotation_name_gp = grid::gpar(fontsize = 5),
  annotation_legend_param = list(
  # title = "Expression Levels",
  title_gp = gpar(fontsize = 5),
  labels_gp = gpar(fontsize = 5),
  grid_width = unit(5,"point"),
  
```

```

grid_height = unit(5,"point")

))

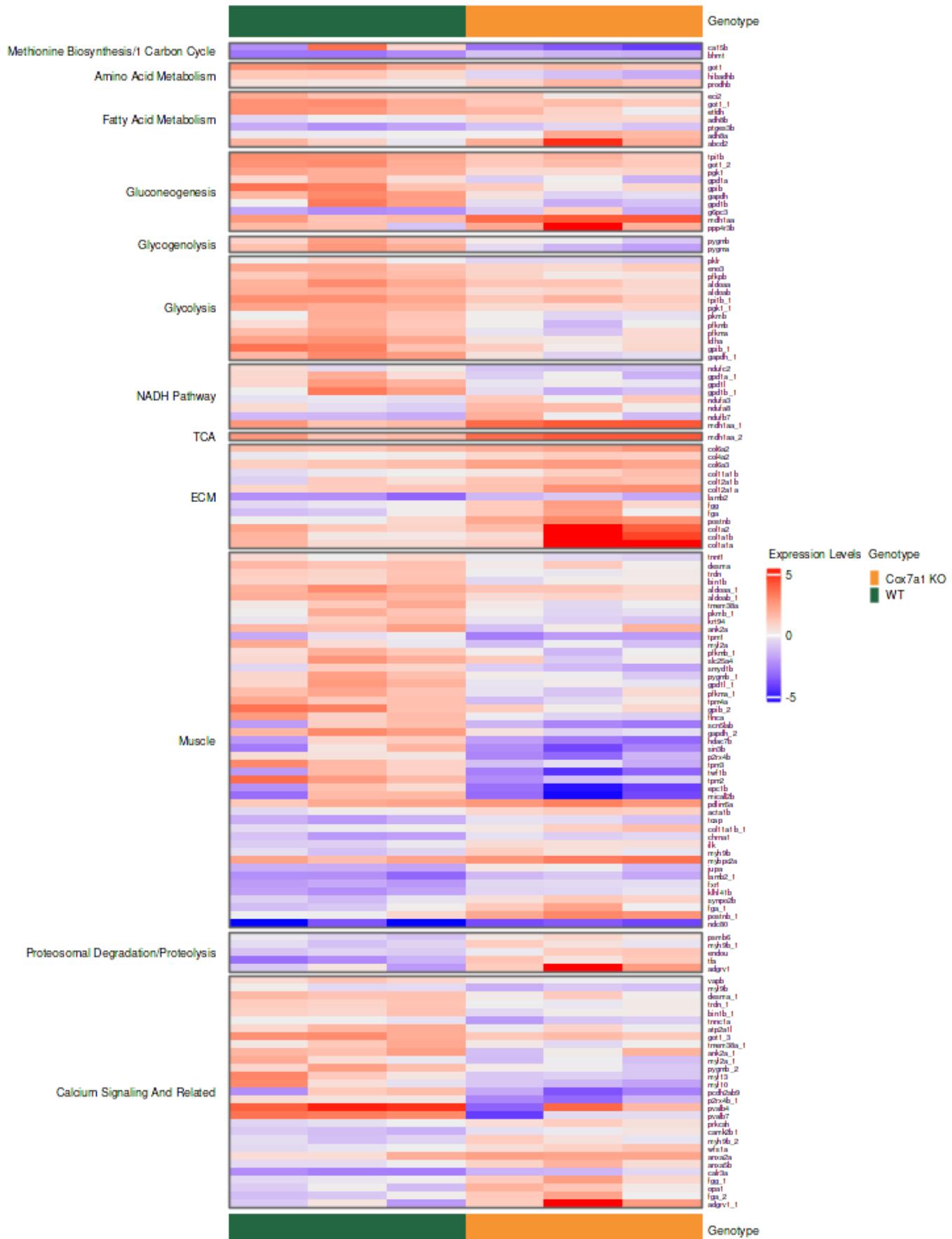
Heatmap(heatmap_mat_v2_others, name="Expression Levels",
  cluster_columns = FALSE,
  cluster_rows = FALSE,
  row_dend_width = unit(0, "cm"),
# col = viridis::viridis(100, direction = -1),
# col= paletteer::paletteer_c("grDevices::Rocket", n=100, direction=-1),
  show_column_names = FALSE,
  top_annotation = ha,
  row_split = factor(heatmap_list_v2_others_unique$OXPHOS.Complex, levels =
    unique(heatmap_list_v2_others_unique$OXPHOS.Complex)),
  gap=unit(2, "point"),
  border= TRUE,
  border_gp = gpar(col = "grey40"),
  row_title_gp = gpar(fontsize = 5), # change row split font size
  row_title_rot = 0,
  bottom_annotation = ha,
  heatmap_legend_param = list(
    title = "Expression Levels",
    title_gp = gpar(fontsize = 5),
    labels_gp = gpar(fontsize = 5),
    grid_width = unit(5,"point"),
    grid_height = unit(5,"point")
  ),
  row_names_gp = grid::gpar(fontsize = 3))
)

dev.copy(pdf, file = "proteomics_n_3_muscle/muscle_heatmap_annotated_v5_others.pdf",
  width = 4, height = 8)
dev.off()

```

pdf: 3

png: 2



8 Volcano Plot

▼ Code

```

# description_df_mouse_zf %>% colnames
description_df_mouse_zf_volcano <- description_df_mouse_zf %>% dplyr::select(gene,
    AKO1, AKO2, AKO3, AWT1, AWT2, AWT3, adult_ko_vs_adult_wt_P.Value,
    adult_ko_vs_adult_wt_logFC, NOP)
# rename adult_ko_vs_adult_wt_P.Value to pval adult_ko_vs_adult_wt_logFC to
# logFoldChange
description_df_mouse_zf_volcano <- description_df_mouse_zf_volcano %>%
    dplyr::rename(pval = adult_ko_vs_adult_wt_P.Value, logFoldChange =
        adult_ko_vs_adult_wt_logFC)
description_df_mouse_zf_volcano <- description_df_mouse_zf_volcano %>% distinct
description_df_mouse_zf_volcano <- description_df_mouse_zf_volcano %>% group_by(gene) %>%
    filter(NOP == max(NOP)) %>% ungroup()
description_df_mouse_zf_volcano <- description_df_mouse_zf_volcano %>% group_by(gene) %>%
    filter(logFoldChange == max(logFoldChange)) %>% ungroup()
# convert columns except gene to numeric
description_df_mouse_zf_volcano <- description_df_mouse_zf_volcano %>%
    mutate_at(vars(contains("AKO")), as.numeric)
description_df_mouse_zf_volcano <- description_df_mouse_zf_volcano %>%
    mutate_at(vars(contains("AWT")), as.numeric)
description_df_mouse_zf_volcano <- description_df_mouse_zf_volcano %>%
    mutate_at(vars(contains("NOP")), as.numeric)
description_df_mouse_zf_volcano <- description_df_mouse_zf_volcano %>%
    mutate_at(vars(contains("logFoldChange")), as.numeric)
description_df_mouse_zf_volcano <- description_df_mouse_zf_volcano %>%
    mutate_at(vars(contains("pval")), as.numeric)

description_df_mouse_zf_volcano %>% colnames()
description_df_mouse_zf_volcano %>% dim()
description_df_mouse_zf_volcano %>% str
# description_df_mouse_zf_volcano %>% arrange(desc(abs(logFoldChange))) %>% head()

```

'gene' · 'AKO1' · 'AKO2' · 'AKO3' · 'AWT1' · 'AWT2' · 'AWT3' · 'pval' · 'logFoldChange' · 'NOP'

1357 · 10

```

tibble [1,357 x 10] (S3: tbl_df/tbl/data.frame)
$ gene      : chr [1:1357] "aamdc" "aars1" "aass" "abcb10" ...
$ AKO1      : num [1:1357] 1.467 -1.353 0.73 0.285 0.397 ...
$ AKO2      : num [1:1357] -0.574 -1.575 1.312 1.243 -0.266 ...
$ AKO3      : num [1:1357] -1.865 -1.795 0.289 0.623 -0.657 ...
$ AWT1      : num [1:1357] 0.2673 -2.832 0.5338 0.2188 -0.0145 ...
$ AWT2      : num [1:1357] -0.2308 -2.542 0.0574 0.0966 0.5438 ...
$ AWT3      : num [1:1357] 0.921 -2.68 -0.985 0.781 -0.515 ...
$ pval      : num [1:1357] 0.42827 0.00561 0.20987 0.39661 0.72744 ...
$ logFoldChange: num [1:1357] -0.643 1.11 0.909 0.352 -0.18 ...
$ NOP       : num [1:1357] 2 8 10 6 1 5 4 29 14 27 ...

```

▼ Code

```

# make a volcano plot for description_df_mouse_zf only for AKO and AWT columns
#####
#
# VOLCANO FUNCTION #
#
#####
require(ggplot2)
require(ggrepel)

```

```

require(clusterProfiler)
require(tidyverse)

draw_volcano<- function(fileinput, title) {
  # read input file
  # drawing plots
  ggplot(data =fileinput , aes(x = logFoldChange, y = -log10(pval))) +
    # draw lines
    geom_hline(yintercept = -log10(0.05), linetype = "dashed", col = "gray50") +
    geom_hline(yintercept = -log10(0.05), linetype = "dotted", col = "gray50") +
    geom_vline(xintercept = 0, linetype = "dashed")+
    # draw points
    geom_point(x = fileinput$logFoldChange, y = -log10(fileinput$pval),alpha = 0.5,
               size = 2,color="grey51") +
    # draw coloured points
    geom_point(data = fileinput[which(fileinput$pval < 0.05 & fileinput$logFoldChange
                                     < -1),],
               aes(x=logFoldChange, y = -log10(pval)), shape = 21, color =
               "#226741", fill = "#226741",
               "green", fill = "green",
               alpha = 0.8, size = 2) +
    geom_point(data = fileinput[which(fileinput$pval < 0.05 & fileinput$logFoldChange
                                     > 1),],
               aes(x=logFoldChange, y = -log10(pval)), shape = 21, color =
               "#f69431", fill = "#f69431",
               "magenta", fill = "magenta",alpha = 0.8, size = 2) +
    # x axis scale
    scale_x_continuous(breaks = seq(round(min(fileinput$logFoldChange)-
                                         0.5),round(max(fileinput$logFoldChange)+ 0.5),by = 1), limits =
    c(round(min(fileinput$logFoldChange)-1),round(max(fileinput$logFoldChange)+1)) +
    + xlab("logFoldChange") + ylab("-Log10(p.value)") +
    + # ylab("-log10(p.value)")+

    # set title
    ggtitle(title) +
    # x and y axis limits
    # black and white theme
    theme_bw() +
    # center title
    theme(plot.title = element_text(hjust = 0.5), axis.text = element_text(size =
      10), axis.title.x = element_text(size = 10),
          axis.title.y = element_text(size = 10))
}

#####

```

Loading required package: ggrepel

▼ Code

```
description_df_mouse_zf_volcano %>% head
```

A tibble: 6 × 10

gene	AKO1	AKO2	AKO3	AWT1	AWT2	AWT3	pval
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
aamdc	1.4667515	-0.5739485	-1.8653050	0.26725443	-0.23076900	0.9212840	0.428266461
aars1	-1.3534255	-1.5746633	-1.7947352	-2.83197660	-2.54200323	-2.6800583	0.005605508
aass	0.7297959	1.3124487	0.2894110	0.53375739	0.05744185	-0.9853736	0.209867021
abcb10	0.2852603	1.2430460	0.6226768	0.21875262	0.09663467	0.7806194	0.396607169
abcb11b	0.3971808	-0.2657106	-0.6573271	-0.01452689	0.54376251	-0.5148912	0.727444898
ablim2	-0.2003520	0.6691284	-0.4681791	-0.94017916	0.52021203	0.8917863	0.798302666

▼ Code

```
description_df_mouse_zf_volcano %>% filter(gene=="cox7a1")
```

A tibble: 1 × 10

gene	AKO1	AKO2	AKO3	AWT1	AWT2	AWT3	pval	logFoldChange
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
cox7a1	-4.979908	-5.996865	-6.519969	0.8575421	1.551887	1.364975	8.497674e-08	-7.090382

▼ Code

```
# change graph size
options(repr.plot.width=10, repr.plot.height=10)
b = draw_volcano(description_df_mouse_zf_volcano, "")
print(b)
# Set genes for marking
# take top and bottom 20 genes according to logFoldChange
top_20_up <- description_df_mouse_zf_volcano %>% arrange(desc(logFoldChange)) %>%
  filter(pval<0.05) %>% head(15)
top_20_up
top_20_down <- description_df_mouse_zf_volcano %>% arrange(logFoldChange) %>%
  filter(pval<0.05) %>% head(15)
# top_20_down
top_20_updown <- rbind(top_20_up,top_20_down)
# top_20_updown

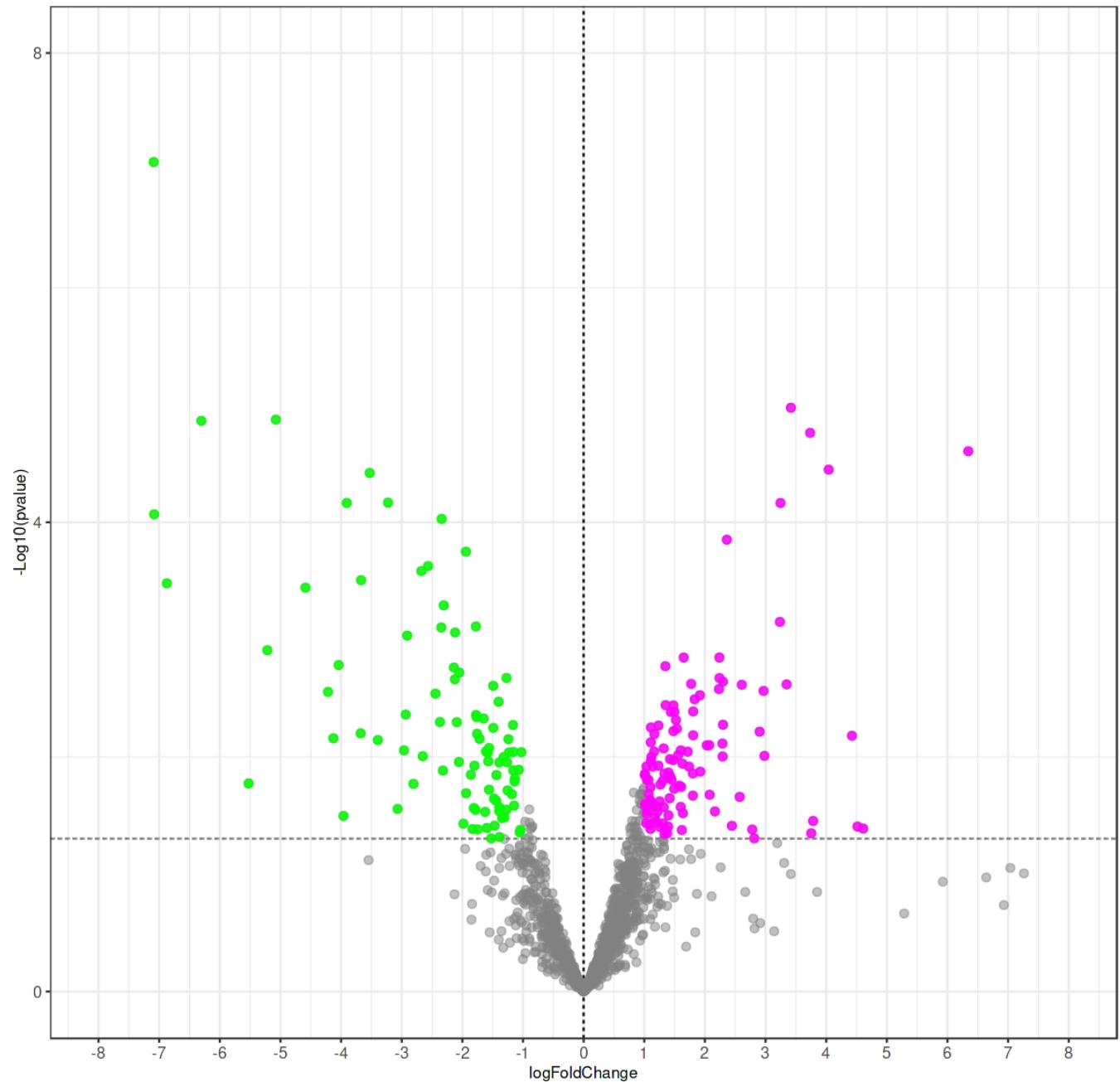
require(ggrepel)
#Paint the genes in the plot
c= b + geom_point(data=description_df_mouse_zf_volcano[description_df_mouse_zf_volcano$gene%in%top_20,,"red","midnightblue"],size=2) +
  geom_text_repel(data =
    description_df_mouse_zf_volcano[description_df_mouse_zf_volcano$gene%in%top_20,],nudge_x = 0.5,
```

```
# # #Save the plot
dev.copy(
  pdf,
  file = "proteomics_n_3_muscle/muscle_volcano_magenta_green.pdf",
  width = 10,
  height = 10
)
dev.off()
```

A tibble: 15 × 10

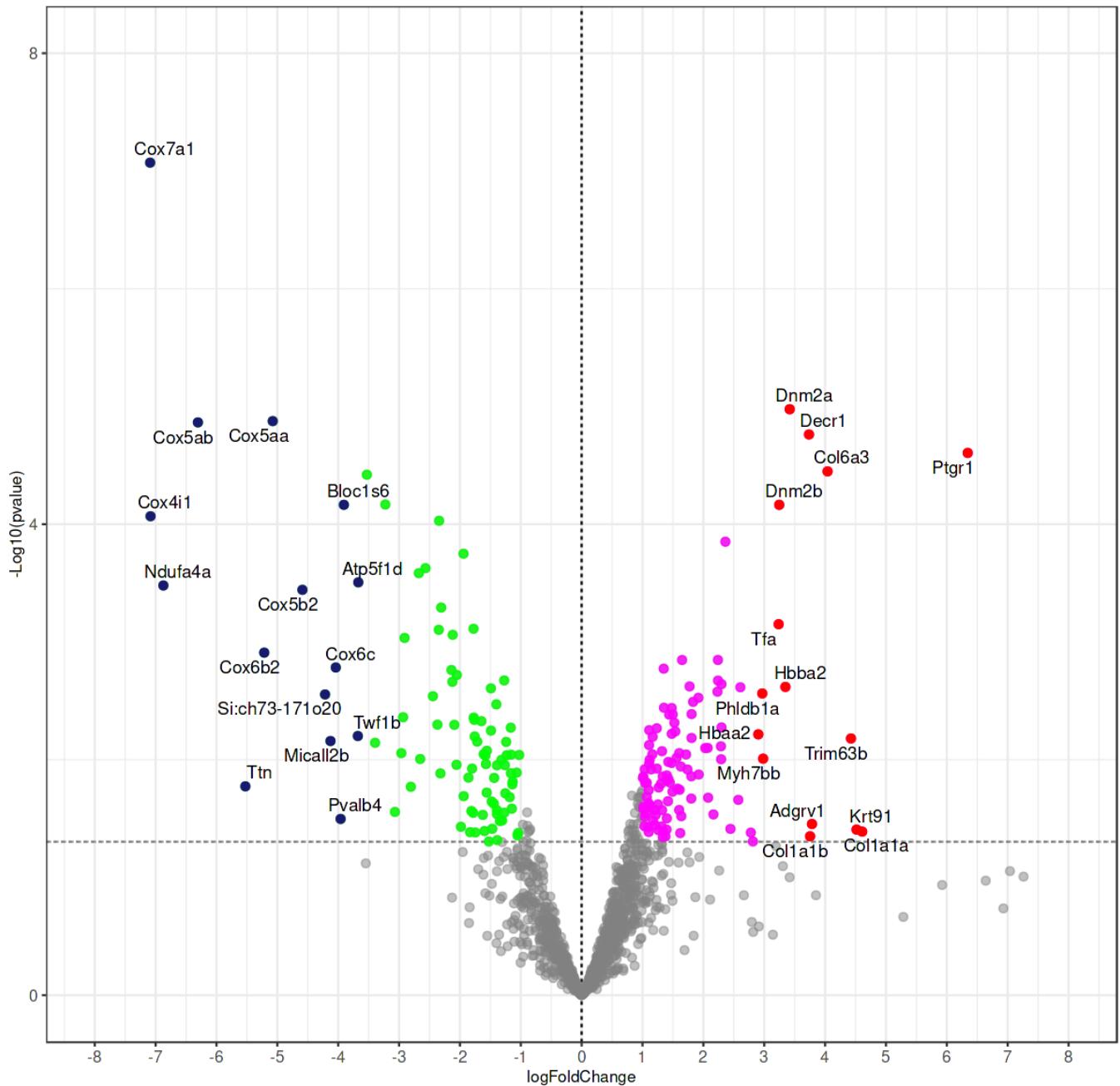
gene	AKO1	AKO2	AKO3	AWT1	AWT2	AWT3	pval
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
ptgr1	2.605453673	2.4928937	5.8000390	-2.2985536	-2.80149612	-3.0319917	2.480240e-05
col1a1a	0.900544733	10.5000921	5.3215073	1.6969782	0.59543050	0.6025185	4.078683e-02
krt91	2.787074278	10.2051594	2.4975397	1.9635349	0.90497268	-0.9285402	3.921087e-02
trim63b	-0.003286546	3.4160337	0.5978636	-2.7471222	-2.27077778	-4.2470609	6.601635e-03
col6a3	3.754599029	5.5415015	4.4753234	0.9305037	0.47072754	0.2486165	3.558867e-05
adgrv1	1.010816887	5.2496166	2.4561205	-0.8947797	0.33333846	-2.0750193	3.514955e-02
col1a1b	0.612985860	8.3097809	4.5309082	1.7562204	0.23147725	0.2070651	4.472143e-02
decr1	3.023227765	3.2854124	3.4034284	-0.2918824	-0.09239048	-1.1092500	1.728903e-05
dnm2a	0.305747091	1.4828109	1.1581865	-2.5327060	-2.17898648	-2.5963649	1.055365e-05
hbba2	4.320792218	3.0300190	4.1329728	0.2657386	0.79754711	0.3774827	2.408032e-03
dnm2b	0.889583393	1.7725652	1.4690079	-1.6170149	-1.20088676	-2.7887759	6.843193e-05
tfa	1.268289501	0.3553622	1.1084034	-3.0646117	-2.43084217	-1.4816043	7.066263e-04

gene	AKO1	AKO2	AKO3	AWT1	AWT2	AWT3	pval	
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
myh7bb	-1.978173192	1.4181456	2.0326610	-1.9429883	-2.70860570	-2.8198197	9.797421e- 2	
							03	
phldb1a	-1.013970204	1.6010596	1.6326834	-1.4359231	-2.19553561	-3.0514311	2.737509e- 2	
							03	
hbac2	4.763062970	3.1230884	3.9600492	0.9110557	1.37230753	0.8557127	6.089900e- 2	
							03	



pdf: 3

png: 2



9 Save data and sessioninfo

▼ Code

```
save.image("./muscle_proteomics_n_3_12122023.RData")
```

▼ Code

```
load("./muscle_proteomics_n_3_12122023.RData")
```

▼ Code

```
sessionInfo()
```

```
R version 4.3.2 (2023-10-31)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 22.04.3 LTS
```

```
Matrix products: default
BLAS:    /usr/lib/x86_64-linux-gnublas/libblas.so.3.10.0
LAPACK:  /usr/lib/x86_64-linux-gnulapack/liblapack.so.3.10.0
```

```
locale:
[1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
[3] LC_TIME=de_CH.UTF-8       LC_COLLATE=en_US.UTF-8
[5] LC_MONETARY=de_CH.UTF-8   LC_MESSAGES=en_US.UTF-8
[7] LC_PAPER=de_CH.UTF-8      LC_NAME=C
[9] LC_ADDRESS=C              LC_TELEPHONE=C
[11] LC_MEASUREMENT=de_CH.UTF-8 LC_IDENTIFICATION=C
```

```
time zone: Europe/Zurich
tzcode source: system (glibc)
```

```
attached base packages:
[1] grid      stats4     stats      graphics  grDevices utils      datasets
[8] methods    base
```

```
other attached packages:
```

```
[1] biomaRt_2.56.1      viridis_0.6.3      viridisLite_0.4.2
[4] ggrepel_0.9.3       ggVennDiagram_1.2.3 limma_3.56.2
[7] openxlsx_4.2.5.2    paletteer_1.5.0    ComplexHeatmap_2.16.0
[10] org.Hs.eg.db_3.17.0 org.Mm.eg.db_3.17.0 AnnotationDbi_1.62.1
[13] IRanges_2.34.1      S4Vectors_0.38.1   Biobase_2.60.0
[16] BiocGenerics_0.46.0 clusterProfiler_4.8.1 lubridate_1.9.2
[19]forcats_1.0.0       stringr_1.5.0      dplyr_1.1.2
[22] purrr_1.0.1        readr_2.1.4       tidyverse_2.0.0
[25] tibble_3.2.1       ggplot2_3.4.2    tidyverse_2.0.0
```

```
loaded via a namespace (and not attached):
```

```
[1] RColorBrewer_1.1-3    shape_1.4.6      jsonlite_1.8.7
[4] magrittr_2.0.3        farver_2.1.1    GlobalOptions_0.1.2
[7] zlibbioc_1.46.0      vctrs_0.6.3     memoise_2.0.1
[10] RCurl_1.98-1.12     ggtree_3.8.0    base64enc_0.1-3
[13] progress_1.2.2       htmltools_0.5.5 curl_5.0.1
[16] gridGraphics_0.5-1   plyr_1.8.8      cachem_1.0.8
[19] uuid_1.1-0          igraph_1.5.0    lifecycle_1.0.3
[22] iterators_1.0.14    pkgconfig_2.0.3 Matrix_1.5-3
[25] R6_2.5.1            fastmap_1.1.1   gson_0.1.0
[28] clue_0.3-64         GenomeInfoDbData_1.2.10 digest_0.6.32
[31] aplot_0.1.10        enrichplot_1.20.0 colorspace_2.1-0
[34] rematch2_2.1.2       patchwork_1.1.2 RSQLite_2.3.1
[37] filelock_1.0.2       fansi_1.0.4     timechange_0.2.0
[40] httr_1.4.6          polyclip_1.10-4 compiler_4.3.2
[43] bit64_4.0.5         withr_2.5.0     doParallel_1.0.17
[46] downloader_0.4       BiocParallel_1.34.2 DBI_1.1.3
[49] ggforce_0.4.1        MASS_7.3-60     rappdirs_0.3.3
[52] rjson_0.2.21        HDO.db_0.99.1   tools_4.3.2
[55] ape_5.7-1           scatterpie_0.2.1 zip_2.3.0
```

```
[58] glue_1.6.2                  nlme_3.1-162                 GOSemSim_2.26.0
[61] shadowtext_0.1.2            pbdZMQ_0.3-9                  cluster_2.1.6
[64] reshape2_1.4.4              fgsea_1.26.0                  generics_0.1.3
[67] gtable_0.3.3               tzdb_0.4.0                   data.table_1.14.8
[70] hms_1.1.3                  xml2_1.3.4                  tidygraph_1.2.3
[73] utf8_1.2.3                 XVector_0.40.0                foreach_1.5.2
[76] pillar_1.9.0               yulab.utils_0.0.6              IRdisplay_1.1
[79] circlize_0.4.15             splines_4.3.2                 tweenr_2.0.2
[82] BiocFileCache_2.8.0         treeio_1.24.1                lattice_0.22-5
[85] bit_4.0.5                   tidyselect_1.2.0              GO.db_3.17.0
[88] Biostrings_2.68.1           gridExtra_2.3                 graphlayouts_1.0.0
[91] matrixStats_1.0.0           stringi_1.7.12                lazyeval_0.2.2
[94] ggfun_0.0.9                 evaluate_0.21                codetools_0.2-19
[97] ggraph_2.1.0                qvalue_2.32.0                RVenn_1.1.0
[100] ggplotify_0.1.0            cli_3.6.1                   IRkernel_1.3.2
[103] repr_1.1.6                 munsell_0.5.0                Rcpp_1.0.11
[106] GenomeInfoDb_1.36.1        dbplyr_2.3.2                 png_0.1-8
[109] XML_3.99-0.14              parallel_4.3.2               blob_1.2.4
[112] prettyunits_1.1.1           DOSE_3.26.1                 bitops_1.0-7
[115] tidytree_0.4.2              scales_1.2.1                 crayon_1.5.2
[118] GetoptLong_1.0.5            rlang_1.1.1                  cowplot_1.1.1
[121] fastmatch_1.1-3            KEGGREST_1.40.0
```