Differential expression of RNA-seq data for a time course analysis on S. mediterranea worms treated with a 70 volt electric field

M. LeGro

4/1/2021

Filtering and normalizing raw reads

First I imported the raw counts file and the metadata files and created a DGEList object with the data. Then I filtered the raw counts for all transcripts with a collective CPM value > 0.5 to remove lowly expressing and partial transcripts. The data were normalized to CPM for downstream statistical comparisons.

```
library("gplots")
```

```

Attaching package: 'gplots'

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

    lowess
```

```
library(ggplot2)
library(limma)
library(edgeR)
library(dplyr)
```

```

Attaching package: 'dplyr'

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

    filter, lag

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

    intersect, setdiff, setequal, union
```

```
library(RColorBrewer)
library(dplyr)
setwd("/Volumes/Watsys/EF/")

## Importing raw reads and metadata
counts <- read.delim("2018.8.9_EF_counts.txt", row.names = 1)
```

```
meta <- read.delim("2019.11.21_meta.txt", row.names = 1)
meta.f <- factor(meta$Time)

metared <- read.delim("2019.11.16_pDCS_reducedsamps_meta.txt",
    row.names = 1)
meta.red <- factor(metared$Time)

## Setting the column names and sample matrix
colnames(counts) <- c("Length", "15min_1", "15min_2", "15min_3",
    "30min_1", "30min_2", "30min_3", "60min_1", "60min_2", "60min_3",
    "0min_1", "0min_2", "0min_3", "delay_1", "delay_2", "delay_3")
countsmat <- as.matrix(counts[, 2:16])

countsmat15 <- as.matrix(counts[, 2:4])
countsmat60 <- as.matrix(counts[, 8:10])
countsmatdelay <- as.matrix(counts[, 14:16])
countsmatctrl <- as.matrix(counts[, 11:13])

## Creating matrices of samples and a reduced matrix for
## comparisons of 15 and 60 minute timepoints

mat <- cbind(countsmatctrl, countsmat15, countsmat60, countsmatdelay)
colnames(mat)
```

```
 [1] "0min_1"  "0min_2"  "0min_3"  "15min_1" "15min_2" "15min_3" "60min_1"
 [8] "60min_2" "60min_3" "delay_1" "delay_2" "delay_3"
```

```
matreduced <- cbind(countsmatctrl, countsmat15, countsmat60)
colnames(matreduced)
```

```
[1] "0min_1"  "0min_2"  "0min_3"  "15min_1" "15min_2" "15min_3" "60min_1"
[8] "60min_2" "60min_3"
```

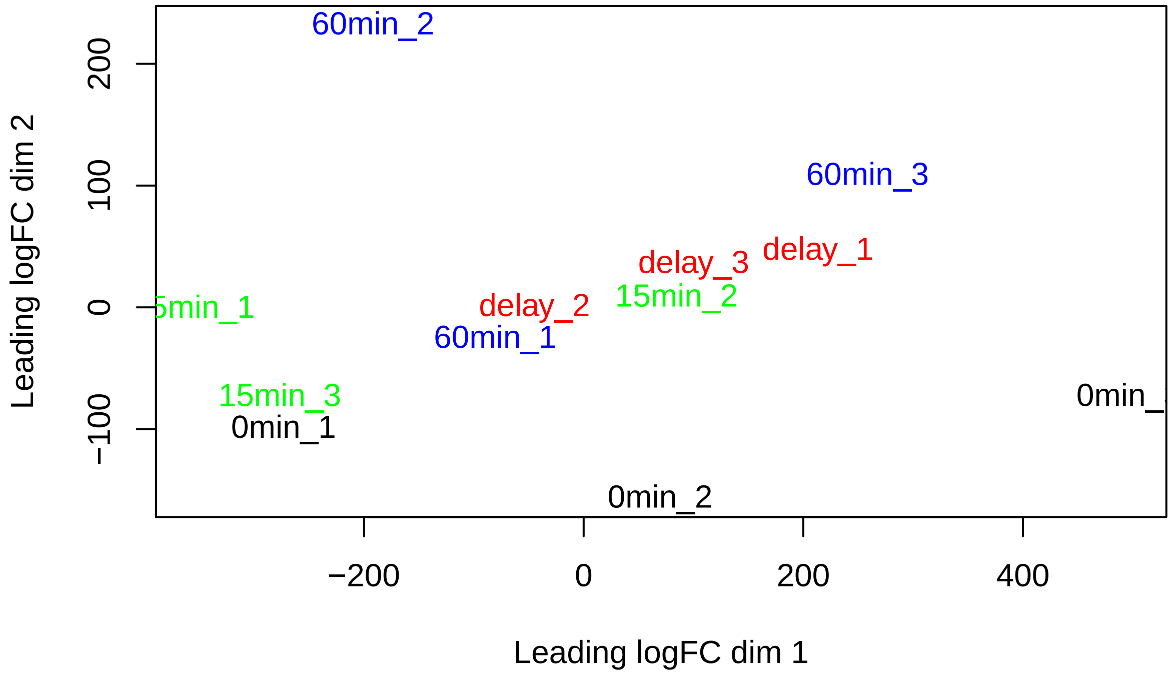
```
# Make an EList object to work with in limma voom
list <- DGEList(mat, group = meta.f)  #Creates a DGE list of the counts dataset
listred <- DGEList(matreduced, group = meta.red)  #Creates a DGE list of the counts dataset

# Calculate Normalization Factors
list <- calcNormFactors(list)
listred <- calcNormFactors(listred)

# Filter samples with less than 0.5 counts per million in all
# samples
keep <- rowSums(cpm(list) > 0.5) >= 1
list <- list[keep, ]
write.csv(list$counts, file = "2019.11.21_pDCS_filterednormdcounts.csv")

keep <- rowSums(cpm(listred) > 0.5) >= 1
listred <- listred[keep, ]
write.csv(listred$counts, file = "2019.11.16_pDCS_filteredcounts_reducedsamps.csv")

#### Writing the cpm counts into files
```

```
CPM <- cpm(list$counts)
write.csv(CPM, file = "2019.11.21_pDCS_countsCPM.csv")

CPMred <- cpm(listred$counts)
write.csv(CPMred, file = "2021.4.15_pDCS_countsCPM_reducedsamps.csv")

countsfiltered <- list$counts
countsfilteredred <- listred$counts

# transpose the matrix so gene rows are columns
CPMt <- t(CPM)
CPMredt <- t(CPMred)

# find the standard score or Z-score of the CPM values for
# downstream graphing and visualization
CPMz <- scale(CPMt, center = TRUE, scale = TRUE)
CPMzscore <- t(CPMz)

CPMredz <- scale(CPMredt, center = TRUE, scale = TRUE)
CPMredzscore <- t(CPMredz)

write.csv(CPMzscore, file = "2021.4.15_pDCS_CPMzscores_allsamps.csv")
write.csv(CPMredzscore, file = "2021.4.15_pDCS_CPMzscores_reducedsamples.csv")
```

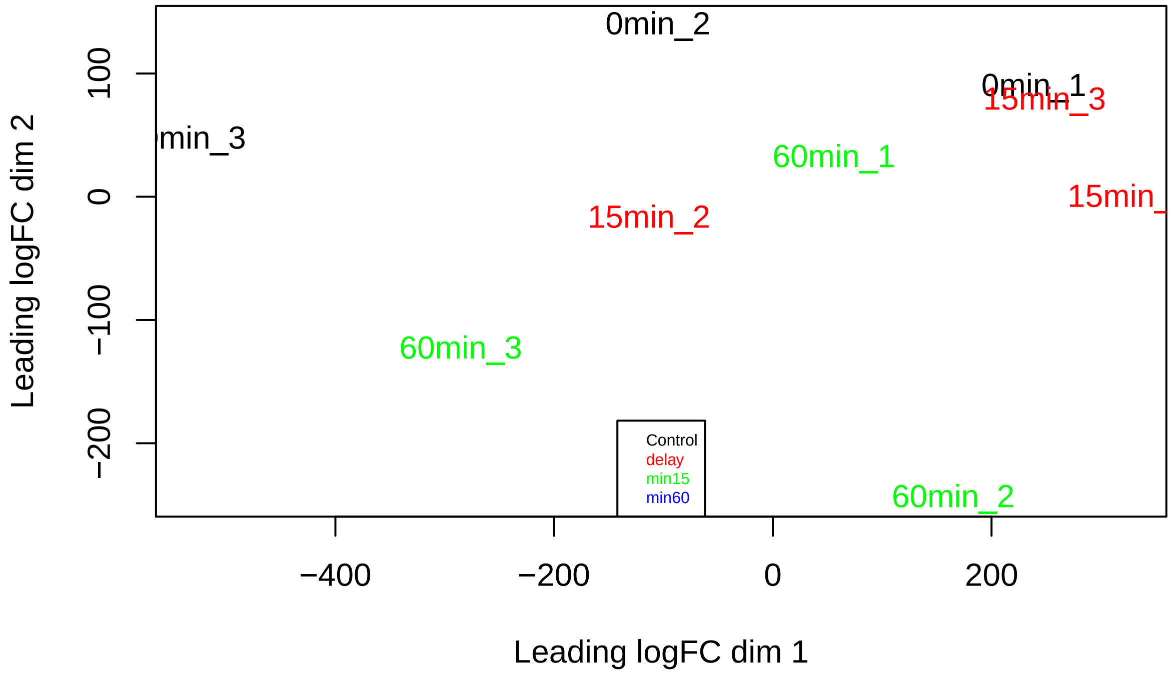
Plotting sample-to-sample variation

I plotted Multi-dimensional scaling plots of all samples to identify sample-to-sample variation.

```
cols <- c("black", "red", "green", "blue", "orange")
MDS <- plotMDS(CPM, col = cols[as.numeric(meta.f)])
```



```
MDS <- plotMDS(CPMred, col = cols[as.numeric(meta.red)])
MDS <- legend("bottom", legend = levels(meta.f), text.col = cols,
    cex = 0.5)
```



Limma-voom method for linear modeling of reads

Next I set up the linear model for statistical comparison based on Chapter 9 of the limma-voom userguide. The Userguides documents can be found here:https://www.bioconductor.org/packages/devel/bioc/vignettes/limma/inst/doc/. I used the 'voomwithQualityweights' function which combines the observational-level weights with sample-specific quality weights. This reduces the sample-to-sample variation and downstream removes likelihood of false positives. Bayesian statistics were used to determine the significance of transcripts between samples. After determining the differential expression of transcripts I merged the topTables of statistics with the CPM z-scores.

```r
# Analyzing as for a single factor, section 9.5.2, making
# design matrix
design <- model.matrix(~0 + meta.f)
colnames(design)
```

```
[1] "meta.fControl" "meta.fdelay"   "meta.fmin15"   "meta.fmin60"
```

```r
colnames(design) <- c("Control", "delay", "min.15", "min.60")
colnames(design)
```

```
[1] "Control" "delay"   "min.15"  "min.60"
```

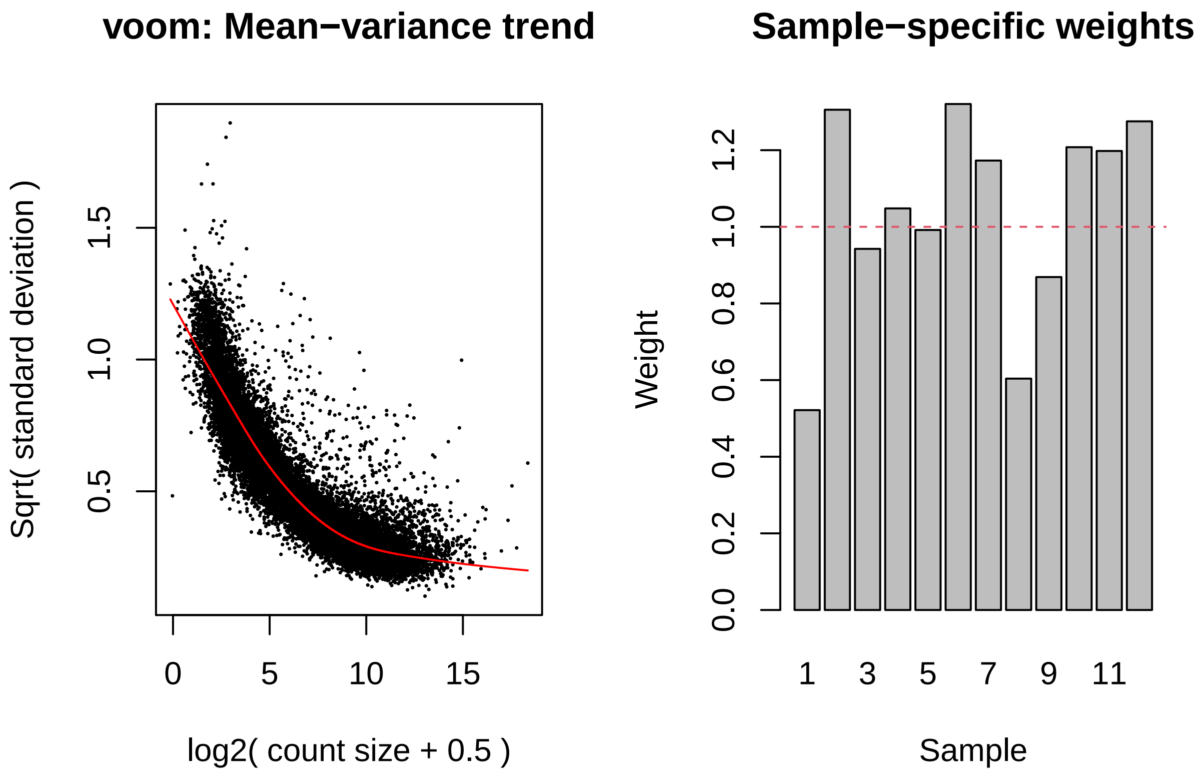
```
designred <- model.matrix(~0 + meta.red)
colnames(designred)
```

```
[1] "meta.redControl" "meta.redmin15"   "meta.redmin60"
```

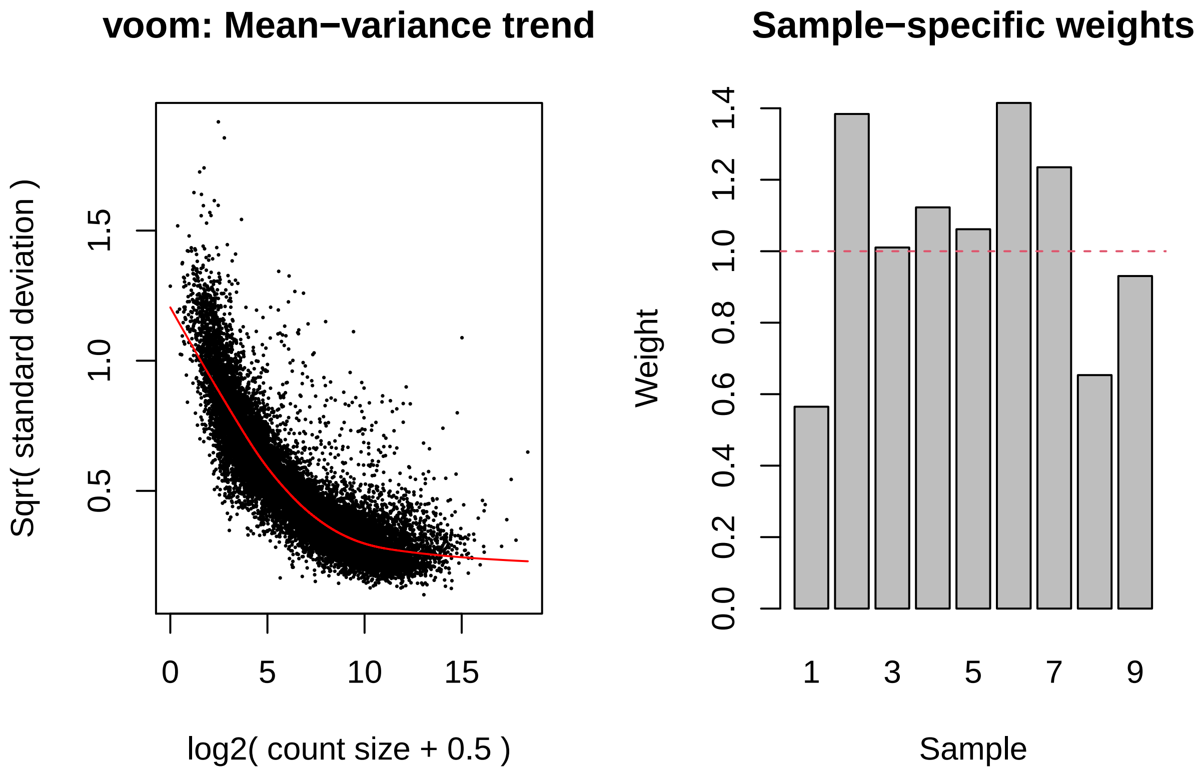
```
colnames(designred) <- c("Control", "min.15", "min.60")
colnames(designred)
```

```
[1] "Control" "min.15"  "min.60"
```

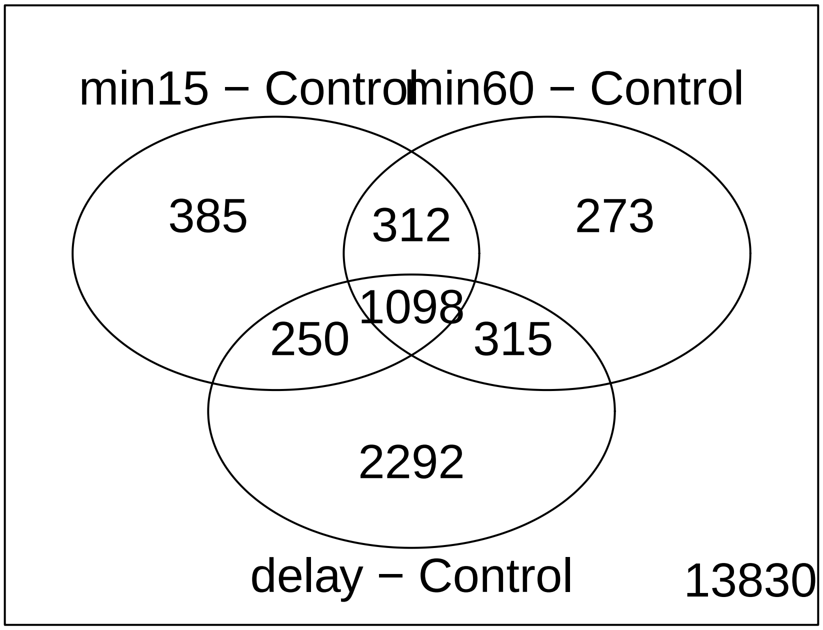
```
# Analyzing as for a single factor, section 9.5.2 and
# adjusting for sample-to-sample variation
v <- voomWithQualityWeights(list, design = design, normalize.method = "none",
    plot = TRUE)
```



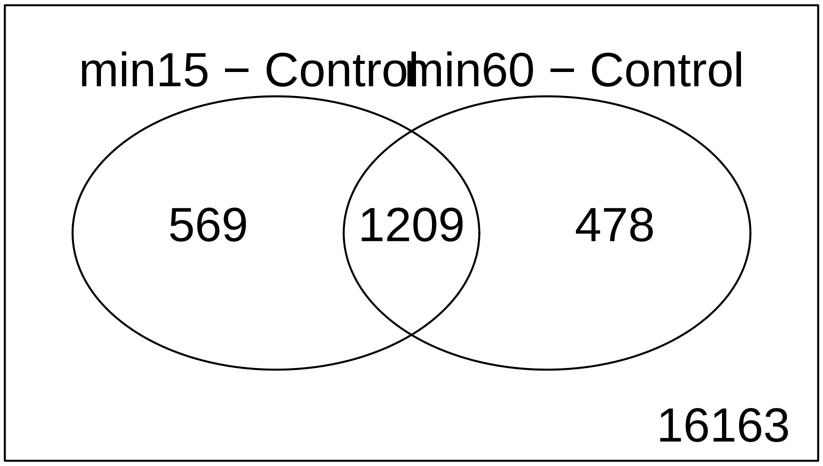
```
vred <- voomWithQualityWeights(listred, design = designred, normalize.method = "none",
    plot = TRUE)
```



```
# fits a linear model to the normalized/filtered dataset
vfit <- lmFit(v, design = design)
vfit <- eBayes(vfit)

vfitred <- lmFit(vred, design = designred)
vfitred <- eBayes(vfitred)

# Setting the statistical contrast matrix
m <- makeContrasts(min15 - Control, min60 - Control, delay -
    Control, levels = meta.f)

mred <- makeContrasts(min15 - Control, min60 - Control, levels = meta.red)

# Fitting the statistical contrasts to the linear model
vfit <- contrasts.fit(vfit, contrasts = m)
vfit <- eBayes(vfit)

vfitred <- contrasts.fit(vfitred, contrasts = mred)
vfitred <- eBayes(vfitred)

# test results from testing a set of contrasts equal to 0
results <- decideTests(vfit)
vennDiagram(results)
```



```
resultsred <- decideTests(vfitred)
vennDiagram(resultsred)
```



```
## printing the t-statistic
tstat <- vfit$t
tstatred <- vfitred$t

# Creating the toptable of test statistics
toptable <- topTable(vfit, number = 3e+05)
toptable_15vctrl <- topTable(vfit, coef = 1, number = 3e+05,
    sort.by = "P")
toptable_60vctrl <- topTable(vfit, coef = 2, number = 3e+05,
    sort.by = "P")
toptable_delayvctrl <- topTable(vfit, coef = 3, number = 3e+05,
    sort.by = "P")

# write.csv(toptable, file = '2018.9.15_EF_toptable.csv')
write.csv(toptable_15vctrl, file = "2019.11.21_pDCS_toptable_coef1_15vctrl.csv")
write.csv(toptable_60vctrl, file = "2019.11.21_pDCS_toptable_coef2_60vctrl.csv")
write.csv(toptable_delayvctrl, file = "2019.11.21_pDCS_toptable_coef3_delayvctrl.csv")

toptablered <- topTable(vfitred, number = 3e+05)
toptable_15vctrlred <- topTable(vfitred, coef = 1, number = 3e+05,
    sort.by = "P")
toptable_60vctrlred <- topTable(vfitred, coef = 2, number = 3e+05,
    sort.by = "P")

write.csv(toptablered, file = "2021.4.16_pDCS_redsamps_toptable.csv")
write.csv(toptable_15vctrlred, file = "2021.4.16_pDCS_toptable_redsamps_15vctrl.csv")
write.csv(toptable_60vctrlred, file = "2021.4.16_pDCS_toptable_redsamps_60vctrl.csv")

# Merging the topTables of statistics with the CPM z-score
# matrices for all genes
merged15vctrl <- merge.data.frame(toptable_15vctrl, CPMzscore,
    by.x = "row.names", by.y = "row.names")
merged60vctrl <- merge.data.frame(toptable_60vctrl, CPMzscore,
    by.x = "row.names", by.y = "row.names")
mergeddelayvctrl <- merge.data.frame(toptable_delayvctrl, CPMzscore,
    by.x = "row.names", by.y = "row.names")

merged15vctrlred <- merge.data.frame(toptable_15vctrlred, CPMredzscore,
    by.x = "row.names", by.y = "row.names")
merged60vctrlred <- merge.data.frame(toptable_60vctrlred, CPMredzscore,
    by.x = "row.names", by.y = "row.names")
```

Row means CPM z-scores for downstream plotting

The following block of code calculate the mean z-scores for each row of the CPM z-scores matrix. This matrix will be used for downstream plotting of heatmaps.

```
# calculating the mean z-scores for z-scores counts file
z0 <- rowMeans(subset(CPMredzscore, select = c("0min_1", "0min_2",
    "0min_3")), na.rm = TRUE)
z15 <- rowMeans(subset(CPMredzscore, select = c("15min_1", "15min_2",
    "15min_3")), na.rm = TRUE)
z60 <- rowMeans(subset(CPMredzscore, select = c("60min_1", "60min_2",
    "60min_3")), na.rm = TRUE)

mat <- cbind(z0, z15, z60)
colnames(mat)
```

```
[1] "z0"  "z15" "z60"
```

```
merged15vctrlred <- merge.data.frame(toptable_15vctrlred, mat,
    by.x = "row.names", by.y = "row.names")
merged60vctrlred <- merge.data.frame(toptable_60vctrlred, mat,
    by.x = "row.names", by.y = "row.names")

# merged zscores with toptable list
write.csv(merged15vctrlred, file = "2021.4.16_pDCS_meanzscores_15vctrl_CPMmeanzscores_wtoptable.csv")
write.csv(merged60vctrlred, file = "2021.4.16_pDCS_meanzscores_60vctrl_CPMmeanzscores_wtoptable.csv")
```

Selecting significantly differentially expressed genes using False Discovery Rate (FDR) and Log Fold Change (LogFC) cutoffs

First the significantly differentially expressed genes are selected using an FDR cutoff <0.05 and <0.01. I decided to use the FDR<0.05 for downstream gene selection since the number of genes is sufficient for downstream plotting. To determine which genes are considered "upregulated" or "downregulated" I'm going to use a logFC cutoff of 0.5 for up or down. However, these cutoffs will only be used for downstream plotting. For the following analysis I used the reduced samples matrices since these time points are needed for publication.

```
### Finding the numbers of sig DE genes correlating with the
### FDR value of 0.01 and 0.05 for both statistical contrasts
### 1778 sidDE at FDR< 0.05 609 sigDE at FDR< 0.01
ctrlv15_0.05 <- merged15vctrlred[which(merged15vctrlred$adj.P.Val <
    0.05), ]
ctrlv15_0.01 <- merged15vctrlred[which(merged15vctrlred$adj.P.Val <
    0.01), ]

## 1687 sidDE at FDR< 0.05 sigDE at FDR< 0.01
ctrlv60_0.05 <- merged60vctrlred[which(merged60vctrlred$adj.P.Val <
    0.05), ]
ctrlv60_0.01 <- merged60vctrlred[which(merged60vctrlred$adj.P.Val <
    0.01), ]

### Finding the upregulated and downregulated genes for each of
### the contrasts and accompanying FDR rates LogFC difference
### of 0.5 and -0.5

# 267 total upregulated genes 315 total downregulated 588
# total updown
ctrlv15_0.05UP_LF0.5 <- ctrlv15_0.05[which(ctrlv15_0.05$logFC >
    0.5), ]
ctrlv15_0.05DOWN_LF0.5 <- ctrlv15_0.05[which(ctrlv15_0.05$logFC <
    -0.5), ]
ctrlv15_0.05UPDOWN <- subset(ctrlv15_0.05, logFC > 0.5 | logFC <
    -0.5)

# 282 total upregulated genes 319 total downregulated 601
# total updown
ctrlv60_0.05UP_LF0.5 <- ctrlv60_0.05[which(ctrlv60_0.05$logFC >
    0.5), ]
ctrlv60_0.05DOWN_LF0.5 <- ctrlv60_0.05[which(ctrlv60_0.05$logFC <
    -0.5), ]
ctrlv60_0.05UPDOWN <- subset(ctrlv60_0.05, logFC > 0.5 | logFC <
    -0.5)
```

Creating Heatmaps of all significantly differentially expressed genes for supplement

Next I created matrices for plotting heatmaps of all significantly differentially expressed genes with an FDR <0.05. I also created complimentary heatmaps with a LogFC cutoff of 0.5 up and down.

```
### Creating the matrices
### ############################################################
ctrlv15_0.05allhm <- cbind(ctrlv15_0.05$z0, ctrlv15_0.05$z15)
rownames(ctrlv15_0.05allhm) <- c(ctrlv15_0.05$Row.names)
colnames(ctrlv15_0.05allhm) <- c("Control", "15_minute")

ctrlv15_0.05UPDOWNhm <- cbind(ctrlv15_0.05UPDOWN$z0, ctrlv15_0.05UPDOWN$z15)
rownames(ctrlv15_0.05UPDOWNhm) <- c(ctrlv15_0.05UPDOWN$Row.names)
```

```r
colnames(ctrlv15_0.05UPDOWNhm) <- c("Control", "15_minute")

ctrlv60_0.05allhm <- cbind(ctrlv60_0.05$z0, ctrlv60_0.05$z60)
rownames(ctrlv60_0.05allhm) <- c(ctrlv60_0.05$Row.names)
colnames(ctrlv60_0.05allhm) <- c("Control", "60_minute")

ctrlv60_0.05UPDOWNhm <- cbind(ctrlv60_0.05UPDOWN$z0, ctrlv60_0.05UPDOWN$z60)
rownames(ctrlv60_0.05UPDOWNhm) <- c(ctrlv60_0.05UPDOWN$Row.names)
colnames(ctrlv60_0.05UPDOWNhm) <- c("Control", "60_minute")


### Creating the heatmaps
### ##########################################################
### Calling the packages and setting the colors
library(ComplexHeatmap)
```

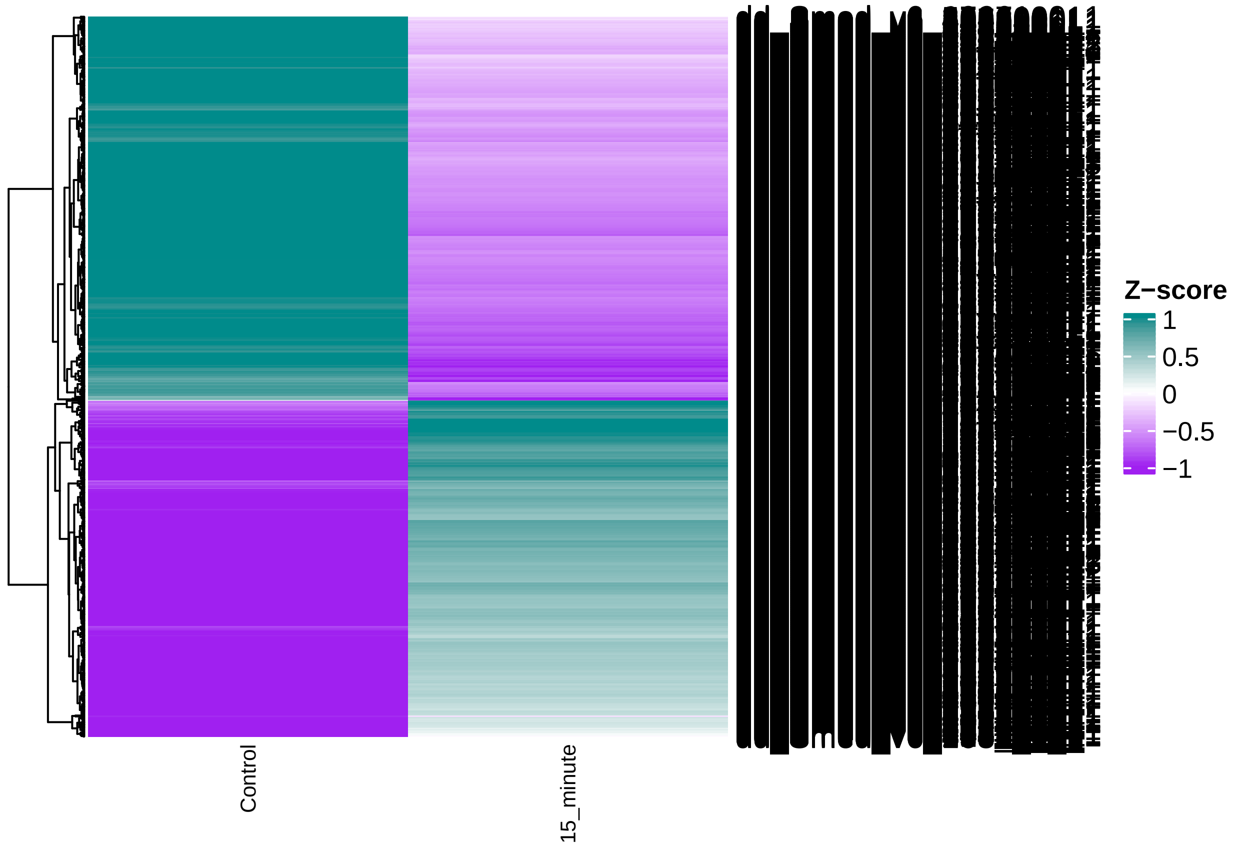
```
Loading required package: grid

========================================
ComplexHeatmap version 2.6.2
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:
Gu, Z. Complex heatmaps reveal patterns and correlations in multidimensional
  genomic data. Bioinformatics 2016.

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

```r
library(circlize)
```

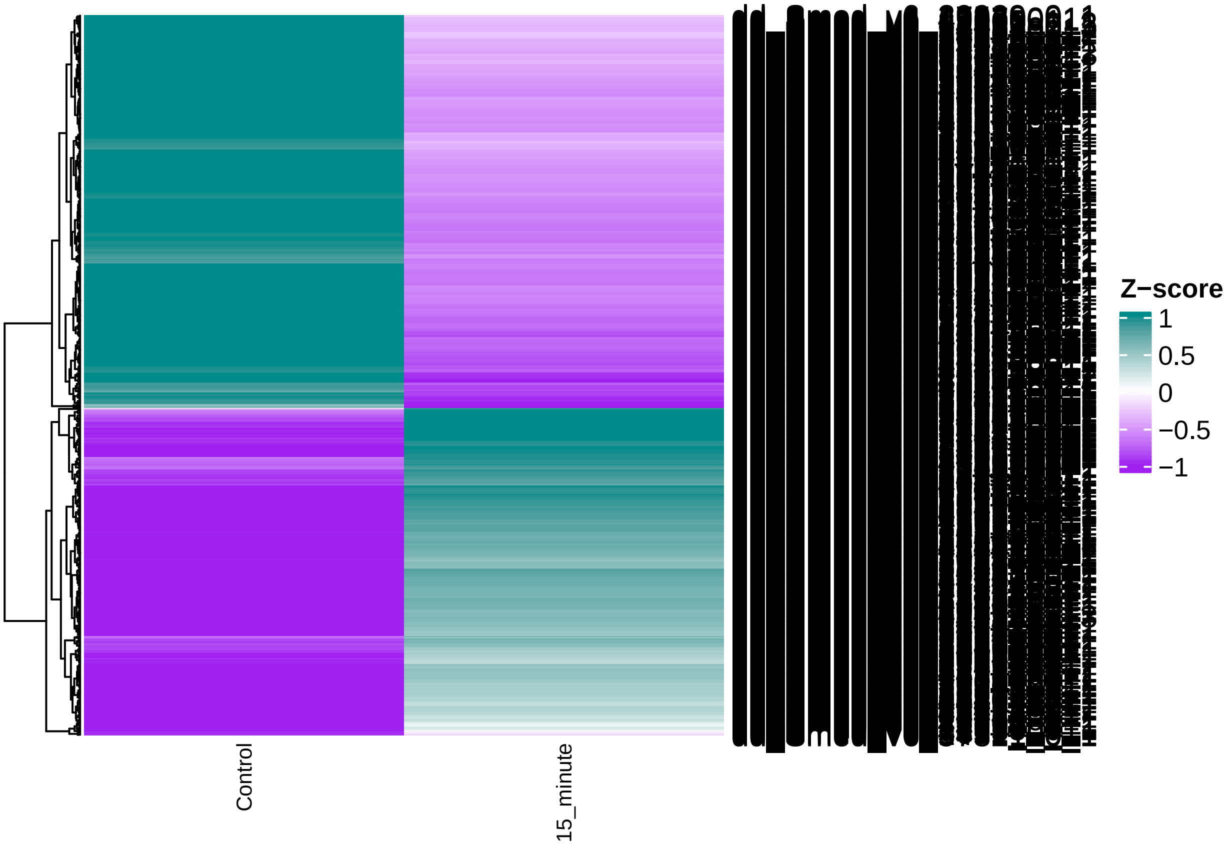
```
========================================
circlize version 0.4.12
CRAN page: https://cran.r-project.org/package=circlize
Github page: https://github.com/jokergoo/circlize
Documentation: https://jokergoo.github.io/circlize_book/book/

If you use it in published research, please cite:
Gu, Z. circlize implements and enhances circular visualization
  in R. Bioinformatics 2014.

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

```r
library(RColorBrewer)

mycolz <- colorRamp2(c(-1, 0, 1), c("purple", "white", "dark cyan"))
```

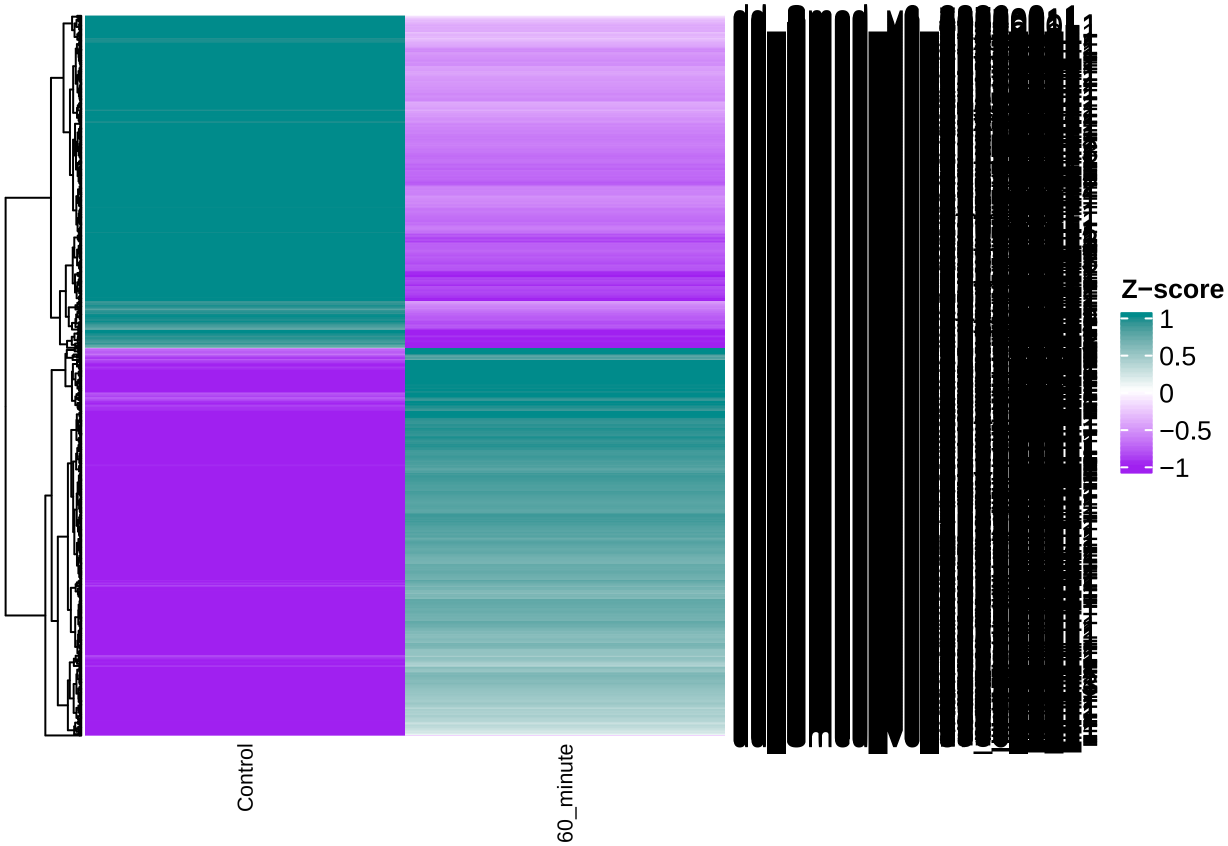
```
##### Heatmaps of sigDE genes for each comparison plotting the
##### mean zscores####
Heatmap(ctrlv15_0.05allhm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



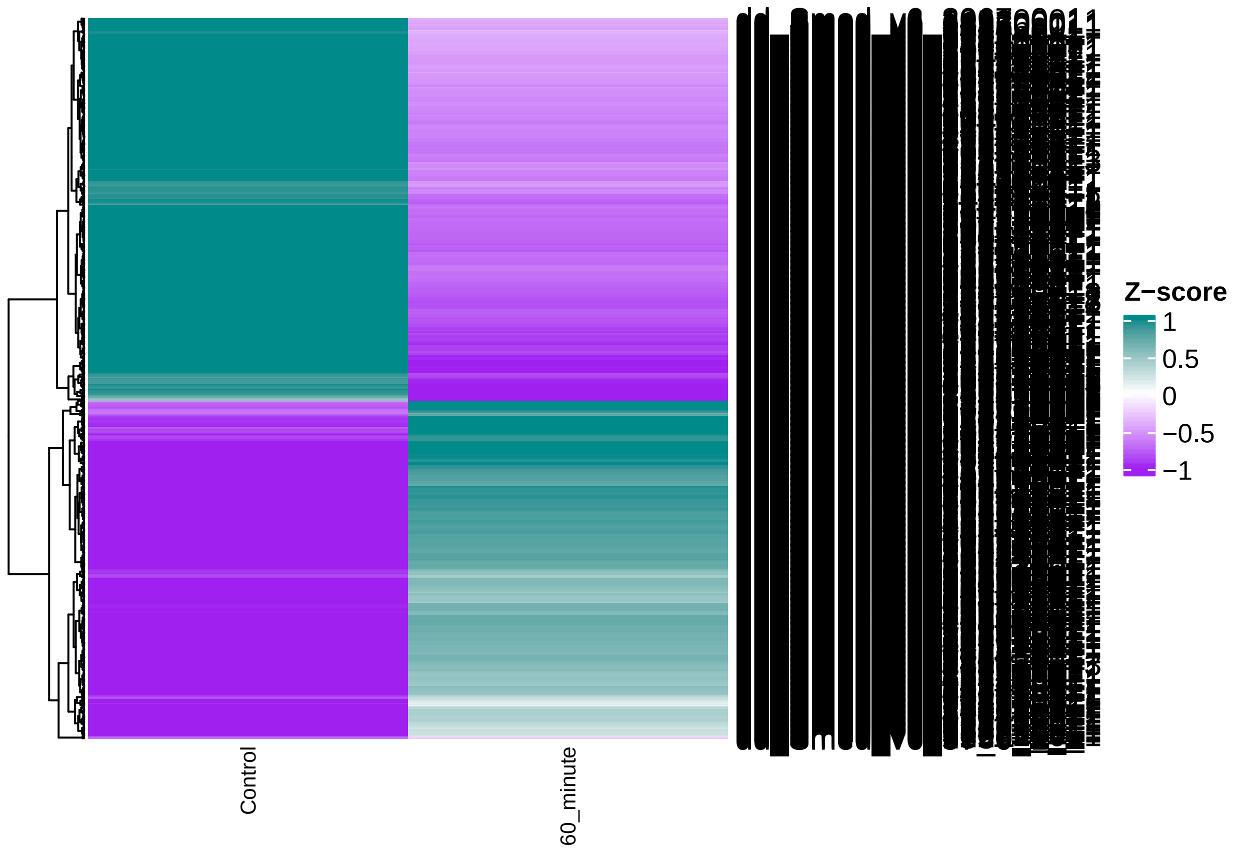
```
Heatmap(ctrlv15_0.05UPDOWNhm, name = "Z-score", col = mycolz,
    column_names_gp = gpar(fontsize = 8), cluster_columns = FALSE,
    cluster_rows = TRUE)
```



```
Heatmap(ctrlv60_0.05allhm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



```
Heatmap(ctrlv60_0.05UPDOWNhm, name = "Z-score", col = mycolz,
    column_names_gp = gpar(fontsize = 8), cluster_columns = FALSE,
    cluster_rows = TRUE)
```



## Merging pathway annotations with mean z-scores for downstream plotting The following lists of pathway annotations are Homo sapien ortholog BLAST annotations that were downloaded from the planmine database here:https://www.bioconductor.org/packages/devel/bioc/vignettes/limma/inst/doc/. I merged the pathway annotations with the significantly differentially expressed transcripts for the 15 minute and 60 minute pairwise comparisons (FDR<0.05 and additional lists with LogFC cutoff of 0.5).

```
celldeath <- read.delim("2020.8.26_celldeath_ddsmedv6_homosapiens_blastannots.txt",
    row.names = 1)
neural <- read.delim("2020.8.26_neural_ddsmedv6_homosapiens_blastannots.txt",
    row.names = 1)
proliferation <- read.delim("2020.8.26_proliferation_ddsmedv6_homosapiens_blastannots.txt",
    row.names = 1)
replication <- read.delim("2020.8.26_replication_ddsmedv6_homosapiens_blastannots.txt",
    row.names = 1)
signaling <- read.delim("2020.8.26_signaling_ddsmedv6_homosapiens_blastannots.txt",
    row.names = 1)
calcium <- read.delim("2020.11.2_calcium_v6homologs_homosapiens_blastannots.txt",
    row.names = 1)
cellmigration <- read.delim("2020.11.2_cellmigration_v6homologs_homosapiens_blastannots.txt",
    row.names = 1)
DNAdamage <- read.delim("2020.11.2_DNAdamage_v6homologs_homosapiens_blastannots.txt",
    row.names = 1)
nbsc <- read.delim("2020.11.2_tspanpaper_markersfulllist_NBSConly_v6homologs.txt",
    row.names = 1)
sublethal <- read.delim("2020.11.2_tspanpaper_markersfulllist_SLonly_v6homologs.txt",
    row.names = 1)
cellcycle <- read.delim("2020.11.16_cellcycle_geneIDs_homosapiens_blastannots.txt",
    row.names = 1)
dedifferentiation <- read.delim("2020.11.16_de-differentiation_geneIDs_homosapiens_blastannots.txt",
    row.names = 1)
DNArepair <- read.delim("2020.11.16_DNArepair_geneIDs_homosapiens_blastannots.txt",
    row.names = 1)
HR <- read.delim("2020.11.16_homologousrecombination_geneIDs_homosapiens_blastannots.txt",
    row.names = 1)
MMR <- read.delim("2020.11.16_mismatchrepair_geneIDs_homosapiens_blastannots.txt",
    row.names = 1)
ieg <- read.delim("2020.11.16_pDCS_ieglist.txt", row.names = 1)
piwi <- read.delim("2020.11.16_piwi_geneIDs_blastannots.txt",
    row.names = 1)
tspan <- read.delim("2020.11.16_t-span_geneIDs_blastannots.txt",
    row.names = 1)
tetraspanin <- read.delim("2020.11.16_tetraspanin_geneIDs_homosapiens_blastannots.txt",
    row.names = 1)
NHEJ <- read.delim("2020.11.16_NHEJ_geneIDs_homosapiens_blastannots.txt",
    row.names = 1)
mex3 <- read.delim("2020.11.22_EF_Hippo_mex3b.txt", row.names = 1)
cellmigS6 <- read.delim("2020.11.29_pDCS_cellmigration_figureS6_homologs_blastannots.txt",
    row.names = 1)
rad51 <- read.delim("2020.11.29_pDCS_rad51_homologs_blastannots.txt",
    row.names = 1)
agat <- read.delim("2021.2.15_agat_blastannots.txt", row.names = 1)
NBFIG <- read.delim("2021.2.22_stemcellmarkers.txt", row.names = 1)

#### MERGING THE gene homologs for pathways lists with the sigDE
#### genes for each timepoint and the specifically upregulated
#### and downregulated ##### cell death ####
celldeath15 <- merge.data.frame(ctrlv15_0.05, celldeath, by.x = "Row.names",
    by.y = "row.names")
celldeath60 <- merge.data.frame(ctrlv60_0.05, celldeath, by.x = "Row.names",
    by.y = "row.names")

celldeath15UPDOWN <- merge.data.frame(ctrlv15_0.05UPDOWN, celldeath,
    by.x = "Row.names", by.y = "row.names")
celldeath60UPDOWN <- merge.data.frame(ctrlv60_0.05UPDOWN, celldeath,
    by.x = "Row.names", by.y = "row.names")

#### neural ####
neural15 <- merge.data.frame(ctrlv15_0.05, neural, by.x = "Row.names",
    by.y = "row.names")
neural60 <- merge.data.frame(ctrlv60_0.05, neural, by.x = "Row.names",
    by.y = "row.names")

neural15UPDOWN <- merge.data.frame(ctrlv15_0.05UPDOWN, neural,
    by.x = "Row.names", by.y = "row.names")
neural60UPDOWN <- merge.data.frame(ctrlv60_0.05UPDOWN, neural,
    by.x = "Row.names", by.y = "row.names")

#### proliferation ####
proliferation15 <- merge.data.frame(ctrlv15_0.05, proliferation,
    by.x = "Row.names", by.y = "row.names")
```

```
proliferation60 <- merge.data.frame(ctrlv60_0.05, proliferation,
    by.x = "Row.names", by.y = "row.names")

proliferation15UPDOWN <- merge.data.frame(ctrlv15_0.05UPDOWN,
    proliferation, by.x = "Row.names", by.y = "row.names")
proliferation60UPDOWN <- merge.data.frame(ctrlv60_0.05UPDOWN,
    proliferation, by.x = "Row.names", by.y = "row.names")

#### replication ####
replication15 <- merge.data.frame(ctrlv15_0.05, replication,
    by.x = "Row.names", by.y = "row.names")
replication60 <- merge.data.frame(ctrlv60_0.05, replication,
    by.x = "Row.names", by.y = "row.names")

replication15UPDOWN <- merge.data.frame(ctrlv15_0.05UPDOWN, replication,
    by.x = "Row.names", by.y = "row.names")
replication60UPDOWN <- merge.data.frame(ctrlv60_0.05UPDOWN, replication,
    by.x = "Row.names", by.y = "row.names")

#### signaling ####
signaling15 <- merge.data.frame(ctrlv15_0.05, signaling, by.x = "Row.names",
    by.y = "row.names")
signaling60 <- merge.data.frame(ctrlv60_0.05, signaling, by.x = "Row.names",
    by.y = "row.names")

signaling15UPDOWN <- merge.data.frame(ctrlv15_0.05UPDOWN, signaling,
    by.x = "Row.names", by.y = "row.names")
signaling60UPDOWN <- merge.data.frame(ctrlv60_0.05UPDOWN, signaling,
    by.x = "Row.names", by.y = "row.names")

#### calcium ####
calcium15 <- merge.data.frame(ctrlv15_0.05, calcium, by.x = "Row.names",
    by.y = "row.names")
calcium60 <- merge.data.frame(ctrlv60_0.05, calcium, by.x = "Row.names",
    by.y = "row.names")

calcium15UPDOWN <- merge.data.frame(ctrlv15_0.05UPDOWN, calcium,
    by.x = "Row.names", by.y = "row.names")
calcium60UPDOWN <- merge.data.frame(ctrlv60_0.05UPDOWN, calcium,
    by.x = "Row.names", by.y = "row.names")

#### cell migration ####
cellmigration15 <- merge.data.frame(ctrlv15_0.05, cellmigration,
    by.x = "Row.names", by.y = "row.names")
cellmigration60 <- merge.data.frame(ctrlv60_0.05, cellmigration,
    by.x = "Row.names", by.y = "row.names")

cellmigration15UPDOWN <- merge.data.frame(ctrlv15_0.05UPDOWN,
    cellmigration, by.x = "Row.names", by.y = "row.names")
cellmigration60UPDOWN <- merge.data.frame(ctrlv60_0.05UPDOWN,
    cellmigration, by.x = "Row.names", by.y = "row.names")

#### DNA damage ####
```

```r
DNAdamage15 <- merge.data.frame(ctrlv15_0.05, DNAdamage, by.x = "Row.names",
    by.y = "row.names")
DNAdamage60 <- merge.data.frame(ctrlv60_0.05, DNAdamage, by.x = "Row.names",
    by.y = "row.names")

DNAdamage15UPDOWN <- merge.data.frame(ctrlv15_0.05UPDOWN, DNAdamage,
    by.x = "Row.names", by.y = "row.names")
DNAdamage60UPDOWN <- merge.data.frame(ctrlv60_0.05UPDOWN, DNAdamage,
    by.x = "Row.names", by.y = "row.names")

#### cellcycle ####
cellcycle15 <- merge.data.frame(ctrlv15_0.05, cellcycle, by.x = "Row.names",
    by.y = "row.names")
cellcycle60 <- merge.data.frame(ctrlv60_0.05, cellcycle, by.x = "Row.names",
    by.y = "row.names")

cellcycle15UPDOWN <- merge.data.frame(ctrlv15_0.05UPDOWN, cellcycle,
    by.x = "Row.names", by.y = "row.names")
cellcycle60UPDOWN <- merge.data.frame(ctrlv60_0.05UPDOWN, cellcycle,
    by.x = "Row.names", by.y = "row.names")

#### DNArepair ####
DNArepair15 <- merge.data.frame(ctrlv15_0.05, DNArepair, by.x = "Row.names",
    by.y = "row.names")
DNArepair60 <- merge.data.frame(ctrlv60_0.05, DNArepair, by.x = "Row.names",
    by.y = "row.names")

DNArepair15UPDOWN <- merge.data.frame(ctrlv15_0.05UPDOWN, DNArepair,
    by.x = "Row.names", by.y = "row.names")
DNArepair60UPDOWN <- merge.data.frame(ctrlv60_0.05UPDOWN, DNArepair,
    by.x = "Row.names", by.y = "row.names")

#### HR ####
HR15 <- merge.data.frame(ctrlv15_0.05, HR, by.x = "Row.names",
    by.y = "row.names")
HR60 <- merge.data.frame(ctrlv60_0.05, HR, by.x = "Row.names",
    by.y = "row.names")

HR15UPDOWN <- merge.data.frame(ctrlv15_0.05UPDOWN, HR, by.x = "Row.names",
    by.y = "row.names")
HR60UPDOWN <- merge.data.frame(ctrlv60_0.05UPDOWN, HR, by.x = "Row.names",
    by.y = "row.names")

#### MMR ####
MMR15 <- merge.data.frame(ctrlv15_0.05, MMR, by.x = "Row.names",
    by.y = "row.names")
MMR60 <- merge.data.frame(ctrlv60_0.05, MMR, by.x = "Row.names",
    by.y = "row.names")

MMR15UPDOWN <- merge.data.frame(ctrlv15_0.05UPDOWN, MMR, by.x = "Row.names",
    by.y = "row.names")
MMR60UPDOWN <- merge.data.frame(ctrlv60_0.05UPDOWN, MMR, by.x = "Row.names",
    by.y = "row.names")
```

```r
#### NHEJ ####
NHEJ15 <- merge.data.frame(ctrlv15_0.05, NHEJ, by.x = "Row.names",
    by.y = "row.names")
NHEJ60 <- merge.data.frame(ctrlv60_0.05, NHEJ, by.x = "Row.names",
    by.y = "row.names")

NHEJ15UPDOWN <- merge.data.frame(ctrlv15_0.05UPDOWN, NHEJ, by.x = "Row.names",
    by.y = "row.names")
NHEJ60UPDOWN <- merge.data.frame(ctrlv60_0.05UPDOWN, NHEJ, by.x = "Row.names",
    by.y = "row.names")

#### cell migration figure S6 ####
figurecellmig15 <- merge.data.frame(ctrlv15_0.05, cellmigS6,
    by.x = "Row.names", by.y = "row.names")
figurecellmig60 <- merge.data.frame(ctrlv60_0.05, cellmigS6,
    by.x = "Row.names", by.y = "row.names")

figurecellmig15UPDOWN <- merge.data.frame(ctrlv15_0.05UPDOWN,
    cellmigS6, by.x = "Row.names", by.y = "row.names")
figurecellmig60UPDOWN <- merge.data.frame(ctrlv60_0.05UPDOWN,
    cellmigS6, by.x = "Row.names", by.y = "row.names")

#### NB markers figure 1 ####
NBFIG15 <- merge.data.frame(ctrlv15_0.05, NBFIG, by.x = "Row.names",
    by.y = "row.names")
NBFIG60 <- merge.data.frame(ctrlv60_0.05, NBFIG, by.x = "Row.names",
    by.y = "row.names")

NBFIG15UPDOWN <- merge.data.frame(ctrlv15_0.05UPDOWN, NBFIG,
    by.x = "Row.names", by.y = "row.names")
NBFIG60UPDOWN <- merge.data.frame(ctrlv60_0.05UPDOWN, NBFIG,
    by.x = "Row.names", by.y = "row.names")

#### rad51 ####
rad5115 <- merge.data.frame(ctrlv15_0.05, rad51, by.x = "Row.names",
    by.y = "row.names")
rad5160 <- merge.data.frame(ctrlv60_0.05, rad51, by.x = "Row.names",
    by.y = "row.names")

rad5115UPDOWN <- merge.data.frame(ctrlv15_0.05UPDOWN, rad51,
    by.x = "Row.names", by.y = "row.names")
rad5160UPDOWN <- merge.data.frame(ctrlv60_0.05UPDOWN, rad51,
    by.x = "Row.names", by.y = "row.names")

#### nbsc ####
nbsc15 <- merge.data.frame(ctrlv15_0.05, nbsc, by.x = "Row.names",
    by.y = "row.names")
nbsc60 <- merge.data.frame(ctrlv60_0.05, nbsc, by.x = "Row.names",
    by.y = "row.names")

nbsc15UPDOWN <- merge.data.frame(ctrlv15_0.05UPDOWN, nbsc, by.x = "Row.names",
    by.y = "row.names")
nbsc60UPDOWN <- merge.data.frame(ctrlv60_0.05UPDOWN, nbsc, by.x = "Row.names",
    by.y = "row.names")

#### sublethal ####
sublethal15 <- merge.data.frame(ctrlv15_0.05, sublethal, by.x = "Row.names",
    by.y = "row.names")
sublethal60 <- merge.data.frame(ctrlv60_0.05, sublethal, by.x = "Row.names",
    by.y = "row.names")

sublethal15UPDOWN <- merge.data.frame(ctrlv15_0.05UPDOWN, sublethal,
    by.x = "Row.names", by.y = "row.names")
sublethal60UPDOWN <- merge.data.frame(ctrlv60_0.05UPDOWN, sublethal,
    by.x = "Row.names", by.y = "row.names")

#### ieg ####
ieg15 <- merge.data.frame(ctrlv15_0.05, ieg, by.x = "Row.names",
    by.y = "row.names")
ieg60 <- merge.data.frame(ctrlv60_0.05, ieg, by.x = "Row.names",
    by.y = "row.names")

ieg15UPDOWN <- merge.data.frame(ctrlv15_0.05UPDOWN, ieg, by.x = "Row.names",
    by.y = "row.names")
ieg60UPDOWN <- merge.data.frame(ctrlv60_0.05UPDOWN, ieg, by.x = "Row.names",
    by.y = "row.names")

#### piwi ####
piwi15 <- merge.data.frame(ctrlv15_0.05, piwi, by.x = "Row.names",
    by.y = "row.names")
piwi60 <- merge.data.frame(ctrlv60_0.05, piwi, by.x = "Row.names",
    by.y = "row.names")

piwi15UPDOWN <- merge.data.frame(ctrlv15_0.05UPDOWN, piwi, by.x = "Row.names",
    by.y = "row.names")
piwi60UPDOWN <- merge.data.frame(ctrlv60_0.05UPDOWN, piwi, by.x = "Row.names",
    by.y = "row.names")
```

Creating matrices for plotting pathway heatmaps

The following chunk of code creates matrices for 15 and 60 minute contrasts using all significantly differentially expressed genes and additional heatmaps using a LogFC cutoff of 0.5.

```
### Creating the matrices
### ##########################################################
### Cell death
### ##################################################
celldeath15$descrip <- paste(celldeath15$blast_description, celldeath15$Row.names,
    sep = "_")
celldeath15$symbol <- paste(celldeath15$blast_symbol, celldeath15$Row.names,
    sep = "_")
celldeath15_hm <- cbind(celldeath15$z0, celldeath15$z15)
rownames(celldeath15_hm) <- c(celldeath15$symbol)
colnames(celldeath15_hm) <- c("Control", "15_minute")
```

```r
celldeath60$descrip <- paste(celldeath60$blast_description, celldeath60$Row.names,
    sep = "_")
celldeath60$symbol <- paste(celldeath60$blast_symbol, celldeath60$Row.names,
    sep = "_")
celldeath60_hm <- cbind(celldeath60$z0, celldeath60$z15)
rownames(celldeath60_hm) <- c(celldeath60$symbol)
colnames(celldeath60_hm) <- c("Control", "60_minute")

celldeath15UPDOWN$descrip <- paste(celldeath15UPDOWN$blast_description,
    celldeath15UPDOWN$Row.names, sep = "_")
celldeath15UPDOWN$symbol <- paste(celldeath15UPDOWN$blast_symbol,
    celldeath15UPDOWN$Row.names, sep = "_")
celldeath15UPDOWN_hm <- cbind(celldeath15UPDOWN$z0, celldeath15UPDOWN$z15)
rownames(celldeath15UPDOWN_hm) <- c(celldeath15UPDOWN$symbol)
colnames(celldeath15UPDOWN_hm) <- c("Control", "15_minute")

celldeath60UPDOWN$descrip <- paste(celldeath60UPDOWN$blast_description,
    celldeath60UPDOWN$Row.names, sep = "_")
celldeath60UPDOWN$symbol <- paste(celldeath60UPDOWN$blast_symbol,
    celldeath60UPDOWN$Row.names, sep = "_")
celldeath60UPDOWN_hm <- cbind(celldeath60UPDOWN$z0, celldeath60UPDOWN$z15)
rownames(celldeath60UPDOWN_hm) <- c(celldeath60UPDOWN$symbol)
colnames(celldeath60UPDOWN_hm) <- c("Control", "60_minute")

################################################################################# neural ###############
neural15_hm <- cbind(neural15$z0, neural15$z15)
rownames(neural15_hm) <- c(neural15$Blast_Symbol)
colnames(neural15_hm) <- c("Control", "15_minute")

neural60_hm <- cbind(neural60$z0, neural60$z15)
rownames(neural60_hm) <- c(neural60$Blast_Symbol)
colnames(neural60_hm) <- c("Control", "60_minute")

neural15UPDOWN_hm <- cbind(neural15UPDOWN$z0, neural15UPDOWN$z15)
rownames(neural15UPDOWN_hm) <- c(neural15UPDOWN$Blast_Symbol)
colnames(neural15UPDOWN_hm) <- c("Control", "15_minute")

neural60UPDOWN_hm <- cbind(neural60UPDOWN$z0, neural60UPDOWN$z15)
rownames(neural60UPDOWN_hm) <- c(neural60UPDOWN$Blast_Symbol)
colnames(neural60UPDOWN_hm) <- c("Control", "60_minute")

################################################################################# proliferation
################################################################################# ######################
proliferation15$descrip <- paste(proliferation15$blast_description,
    proliferation15$Row.names, sep = "_")
proliferation15$symbol <- paste(proliferation15$blast_symbol,
    proliferation15$Row.names, sep = "_")
proliferation15_hm <- cbind(proliferation15$z0, proliferation15$z15)
rownames(proliferation15_hm) <- c(proliferation15$symbol)
colnames(proliferation15_hm) <- c("Control", "15_minute")

proliferation60$descrip <- paste(proliferation60$blast_description,
    proliferation60$Row.names, sep = "_")
```

```
proliferation60$symbol <- paste(proliferation60$blast_symbol,
    proliferation60$Row.names, sep = "_")
proliferation60_hm <- cbind(proliferation60$z0, proliferation60$z15)
rownames(proliferation60_hm) <- c(proliferation60$symbol)
colnames(proliferation60_hm) <- c("Control", "60_minute")

proliferation15UPDOWN$descrip <- paste(proliferation15UPDOWN$blast_description,
    proliferation15UPDOWN$Row.names, sep = "_")
proliferation15UPDOWN$symbol <- paste(proliferation15UPDOWN$blast_symbol,
    proliferation15UPDOWN$Row.names, sep = "_")
proliferation15UPDOWN_hm <- cbind(proliferation15UPDOWN$z0, proliferation15UPDOWN$z15)
rownames(proliferation15UPDOWN_hm) <- c(proliferation15UPDOWN$symbol)
colnames(proliferation15UPDOWN_hm) <- c("Control", "15_minute")

proliferation60UPDOWN$descrip <- paste(proliferation60UPDOWN$blast_description,
    proliferation60UPDOWN$Row.names, sep = "_")
proliferation60UPDOWN$symbol <- paste(proliferation60UPDOWN$blast_symbol,
    proliferation60UPDOWN$Row.names, sep = "_")
proliferation60UPDOWN_hm <- cbind(proliferation60UPDOWN$z0, proliferation60UPDOWN$z15)
rownames(proliferation60UPDOWN_hm) <- c(proliferation60UPDOWN$symbol)
colnames(proliferation60UPDOWN_hm) <- c("Control", "60_minute")

################################################################################ replication
################################################################################ ######################
replication15_hm <- cbind(replication15$z0, replication15$z15)
rownames(replication15_hm) <- c(replication15$Blast_Symbol)
colnames(replication15_hm) <- c("Control", "15_minute")

replication60_hm <- cbind(replication60$z0, replication60$z15)
rownames(replication60_hm) <- c(replication60$Blast_Symbol)
colnames(replication60_hm) <- c("Control", "60_minute")

replication15UPDOWN_hm <- cbind(replication15UPDOWN$z0, replication15UPDOWN$z15)
rownames(replication15UPDOWN_hm) <- c(replication15UPDOWN$Blast_Symbol)
colnames(replication15UPDOWN_hm) <- c("Control", "15_minute")

replication60UPDOWN_hm <- cbind(replication60UPDOWN$z0, replication60UPDOWN$z15)
rownames(replication60UPDOWN_hm) <- c(replication60UPDOWN$Blast_Symbol)
colnames(replication60UPDOWN_hm) <- c("Control", "60_minute")

################################################################################ signaling
################################################################################ ######################
signaling15_hm <- cbind(signaling15$z0, signaling15$z15)
rownames(signaling15_hm) <- c(signaling15$Blast_Symbol)
colnames(signaling15_hm) <- c("Control", "15_minute")

signaling60_hm <- cbind(signaling60$z0, signaling60$z15)
rownames(signaling60_hm) <- c(signaling60$Blast_Symbol)
colnames(signaling60_hm) <- c("Control", "60_minute")

signaling15UPDOWN_hm <- cbind(signaling15UPDOWN$z0, signaling15UPDOWN$z15)
rownames(signaling15UPDOWN_hm) <- c(signaling15UPDOWN$Blast_Symbol)
colnames(signaling15UPDOWN_hm) <- c("Control", "15_minute")
```

```r
signaling60UPDOWN_hm <- cbind(signaling60UPDOWN$z0, signaling60UPDOWN$z15)
rownames(signaling60UPDOWN_hm) <- c(signaling60UPDOWN$Blast_Symbol)
colnames(signaling60UPDOWN_hm) <- c("Control", "60_minute")

################################################################################ calcium ##############
calcium15_hm <- cbind(calcium15$z0, calcium15$z15)
rownames(calcium15_hm) <- c(calcium15$Blast_Symbol)
colnames(calcium15_hm) <- c("Control", "15_minute")

calcium60_hm <- cbind(calcium60$z0, calcium60$z15)
rownames(calcium60_hm) <- c(calcium60$Blast_Symbol)
colnames(calcium60_hm) <- c("Control", "60_minute")

calcium15UPDOWN_hm <- cbind(calcium15UPDOWN$z0, calcium15UPDOWN$z15)
rownames(calcium15UPDOWN_hm) <- c(calcium15UPDOWN$Blast_Symbol)
colnames(calcium15UPDOWN_hm) <- c("Control", "15_minute")

calcium60UPDOWN_hm <- cbind(calcium60UPDOWN$z0, calcium60UPDOWN$z15)
rownames(calcium60UPDOWN_hm) <- c(calcium60UPDOWN$Blast_Symbol)
colnames(calcium60UPDOWN_hm) <- c("Control", "60_minute")

################################################################################ cellmigration
################################################################################ ######################
cellmigration15_hm <- cbind(cellmigration15$z0, cellmigration15$z15)
rownames(cellmigration15_hm) <- c(cellmigration15$Blast_Symbol)
colnames(cellmigration15_hm) <- c("Control", "15_minute")

cellmigration60_hm <- cbind(cellmigration60$z0, cellmigration60$z15)
rownames(cellmigration60_hm) <- c(cellmigration60$Blast_Symbol)
colnames(cellmigration60_hm) <- c("Control", "60_minute")

cellmigration15UPDOWN_hm <- cbind(cellmigration15UPDOWN$z0, cellmigration15UPDOWN$z15)
rownames(cellmigration15UPDOWN_hm) <- c(cellmigration15UPDOWN$Blast_Symbol)
colnames(cellmigration15UPDOWN_hm) <- c("Control", "15_minute")

cellmigration60UPDOWN_hm <- cbind(cellmigration60UPDOWN$z0, cellmigration60UPDOWN$z15)
rownames(cellmigration60UPDOWN_hm) <- c(cellmigration60UPDOWN$Blast_Symbol)
colnames(cellmigration60UPDOWN_hm) <- c("Control", "60_minute")

################################################################################ DNAdamage
################################################################################ ######################
DNAdamage15_hm <- cbind(DNAdamage15$z0, DNAdamage15$z15)
rownames(DNAdamage15_hm) <- c(DNAdamage15$Blast_Symbol)
colnames(DNAdamage15_hm) <- c("Control", "15_minute")

DNAdamage60_hm <- cbind(DNAdamage60$z0, DNAdamage60$z15)
rownames(DNAdamage60_hm) <- c(DNAdamage60$Blast_Symbol)
colnames(DNAdamage60_hm) <- c("Control", "60_minute")

DNAdamage15UPDOWN_hm <- cbind(DNAdamage15UPDOWN$z0, DNAdamage15UPDOWN$z15)
rownames(DNAdamage15UPDOWN_hm) <- c(DNAdamage15UPDOWN$Blast_Symbol)
colnames(DNAdamage15UPDOWN_hm) <- c("Control", "15_minute")
```

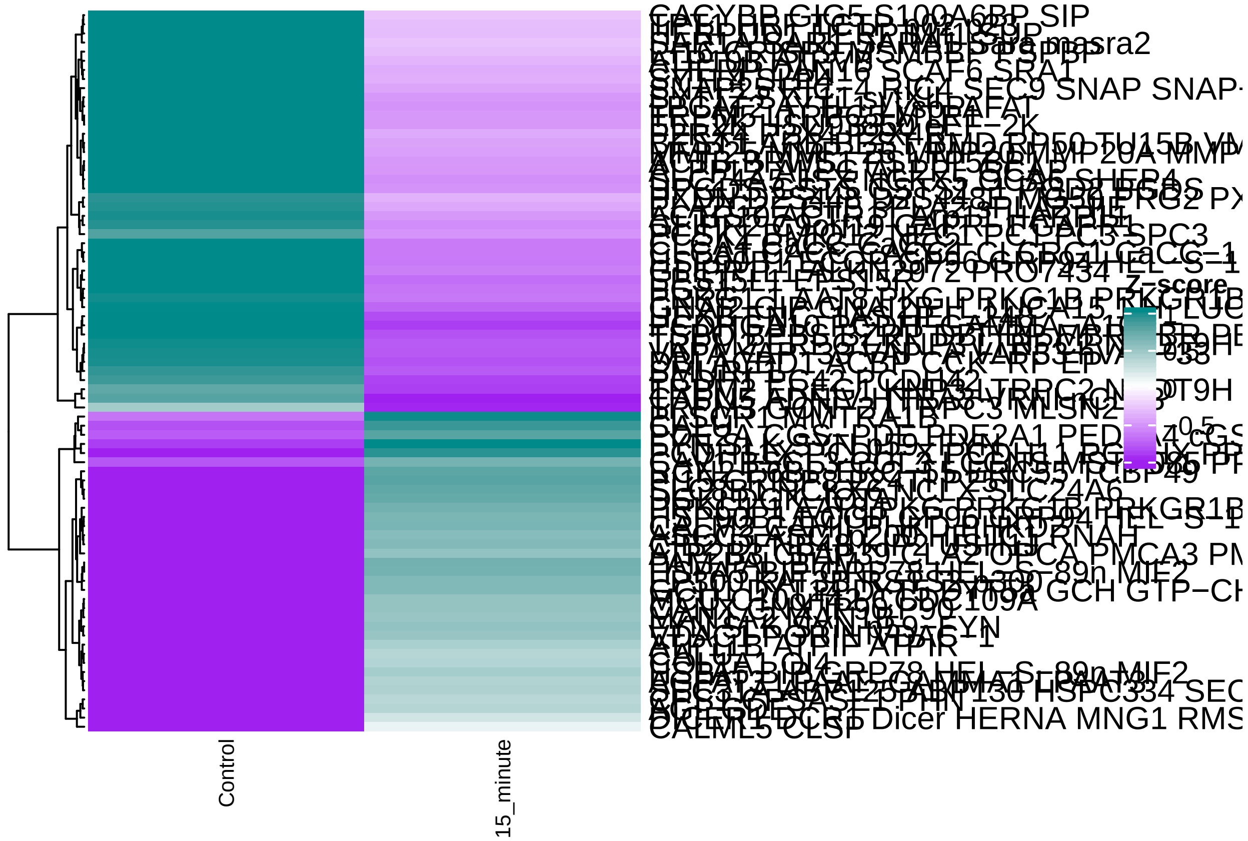
```
DNAdamage60UPDOWN_hm <- cbind(DNAdamage60UPDOWN$z0, DNAdamage60UPDOWN$z15)
rownames(DNAdamage60UPDOWN_hm) <- c(DNAdamage60UPDOWN$Blast_Symbol)
colnames(DNAdamage60UPDOWN_hm) <- c("Control", "60_minute")

################################################################################ nbsc #################
nbsc15_hm <- cbind(nbsc15$z0, nbsc15$z15)
rownames(nbsc15_hm) <- c(nbsc15$Row.names)
colnames(nbsc15_hm) <- c("Control", "15_minute")

nbsc60_hm <- cbind(nbsc60$z0, nbsc60$z15)
rownames(nbsc60_hm) <- c(nbsc60$Row.names)
colnames(nbsc60_hm) <- c("Control", "60_minute")

nbsc15UPDOWN_hm <- cbind(nbsc15UPDOWN$z0, nbsc15UPDOWN$z15)
rownames(nbsc15UPDOWN_hm) <- c(nbsc15UPDOWN$Row.names)
colnames(nbsc15UPDOWN_hm) <- c("Control", "15_minute")

nbsc60UPDOWN_hm <- cbind(nbsc60UPDOWN$z0, nbsc60UPDOWN$z15)
rownames(nbsc60UPDOWN_hm) <- c(nbsc60UPDOWN$Row.names)
colnames(nbsc60UPDOWN_hm) <- c("Control", "60_minute")

################################################################################ sublethal
################################################################################ ######################
sublethal15_hm <- cbind(sublethal15$z0, sublethal15$z15)
rownames(sublethal15_hm) <- c(sublethal15$Row.names)
colnames(sublethal15_hm) <- c("Control", "15_minute")

sublethal60_hm <- cbind(sublethal60$z0, sublethal60$z15)
rownames(sublethal60_hm) <- c(sublethal60$Row.names)
colnames(sublethal60_hm) <- c("Control", "60_minute")

sublethal15UPDOWN_hm <- cbind(sublethal15UPDOWN$z0, sublethal15UPDOWN$z15)
rownames(sublethal15UPDOWN_hm) <- c(sublethal15UPDOWN$Row.names)
colnames(sublethal15UPDOWN_hm) <- c("Control", "15_minute")

sublethal60UPDOWN_hm <- cbind(sublethal60UPDOWN$z0, sublethal60UPDOWN$z15)
rownames(sublethal60UPDOWN_hm) <- c(sublethal60UPDOWN$Row.names)
colnames(sublethal60UPDOWN_hm) <- c("Control", "60_minute")

################################################################################ cellcycle
################################################################################ ######################
cellcycle15_hm <- cbind(cellcycle15$z0, cellcycle15$z15)
rownames(cellcycle15_hm) <- c(cellcycle15$Transcript...Blast.Sequence.Features...Blast.Domain...Symbol)
colnames(cellcycle15_hm) <- c("Control", "15_minute")

cellcycle60_hm <- cbind(cellcycle60$z0, cellcycle60$z15)
rownames(cellcycle60_hm) <- c(cellcycle60$Transcript...Blast.Sequence.Features...Blast.Domain...Symbol)
colnames(cellcycle60_hm) <- c("Control", "60_minute")

cellcycle15UPDOWN_hm <- cbind(cellcycle15UPDOWN$z0, cellcycle15UPDOWN$z15)
rownames(cellcycle15UPDOWN_hm) <- c(cellcycle15UPDOWN$Transcript...Blast.Sequence.Features...Blast.Doma
colnames(cellcycle15UPDOWN_hm) <- c("Control", "15_minute")
```

```
cellcycle60UPDOWN_hm <- cbind(cellcycle60UPDOWN$z0, cellcycle60UPDOWN$z15)
rownames(cellcycle60UPDOWN_hm) <- c(cellcycle60UPDOWN$Transcript...Blast.Sequence.Features...Blast.Domai
colnames(cellcycle60UPDOWN_hm) <- c("Control", "60_minute")


################################################################################ DNArepair
################################################################################ ######################
DNArepair15_hm <- cbind(DNArepair15$z0, DNArepair15$z15)
rownames(DNArepair15_hm) <- c(DNArepair15$Transcript...Blast.Sequence.Features...Blast.Domain...Symbol)
colnames(DNArepair15_hm) <- c("Control", "15_minute")

DNArepair60_hm <- cbind(DNArepair60$z0, DNArepair60$z15)
rownames(DNArepair60_hm) <- c(DNArepair60$Transcript...Blast.Sequence.Features...Blast.Domain...Symbol)
colnames(DNArepair60_hm) <- c("Control", "60_minute")

DNArepair15UPDOWN_hm <- cbind(DNArepair15UPDOWN$z0, DNArepair15UPDOWN$z15)
rownames(DNArepair15UPDOWN_hm) <- c(DNArepair15UPDOWN$Transcript...Blast.Sequence.Features...Blast.Domai
colnames(DNArepair15UPDOWN_hm) <- c("Control", "15_minute")

DNArepair60UPDOWN_hm <- cbind(DNArepair60UPDOWN$z0, DNArepair60UPDOWN$z15)
rownames(DNArepair60UPDOWN_hm) <- c(DNArepair60UPDOWN$Transcript...Blast.Sequence.Features...Blast.Domai
colnames(DNArepair60UPDOWN_hm) <- c("Control", "60_minute")

################################################################################ HR ###################
HR15_hm <- cbind(HR15$z0, HR15$z15)
rownames(HR15_hm) <- c(HR15$Transcript...Blast.Sequence.Features...Blast.Domain...Symbol)
colnames(HR15_hm) <- c("Control", "15_minute")

HR60_hm <- cbind(HR60$z0, HR60$z15)
rownames(HR60_hm) <- c(HR60$Transcript...Blast.Sequence.Features...Blast.Domain...Symbol)
colnames(HR60_hm) <- c("Control", "60_minute")

HR15UPDOWN_hm <- cbind(HR15UPDOWN$z0, HR15UPDOWN$z15)
rownames(HR15UPDOWN_hm) <- c(HR15UPDOWN$Transcript...Blast.Sequence.Features...Blast.Domain...Symbol)
colnames(HR15UPDOWN_hm) <- c("Control", "15_minute")

HR60UPDOWN_hm <- cbind(HR60UPDOWN$z0, HR60UPDOWN$z15)
rownames(HR60UPDOWN_hm) <- c(HR60UPDOWN$Transcript...Blast.Sequence.Features...Blast.Domain...Symbol)
colnames(HR60UPDOWN_hm) <- c("Control", "60_minute")

################################################################################ MMR ##################
MMR15_hm <- cbind(MMR15$z0, MMR15$z15)
rownames(MMR15_hm) <- c(MMR15$Transcript...Blast.Sequence.Features...Blast.Domain...Symbol)
colnames(MMR15_hm) <- c("Control", "15_minute")

MMR60_hm <- cbind(MMR60$z0, MMR60$z15)
rownames(MMR60_hm) <- c(MMR60$Transcript...Blast.Sequence.Features...Blast.Domain...Symbol)
colnames(MMR60_hm) <- c("Control", "60_minute")

MMR15UPDOWN_hm <- cbind(MMR15UPDOWN$z0, MMR15UPDOWN$z15)
rownames(MMR15UPDOWN_hm) <- c(MMR15UPDOWN$Transcript...Blast.Sequence.Features...Blast.Domain...Symbol)
colnames(MMR15UPDOWN_hm) <- c("Control", "15_minute")
```

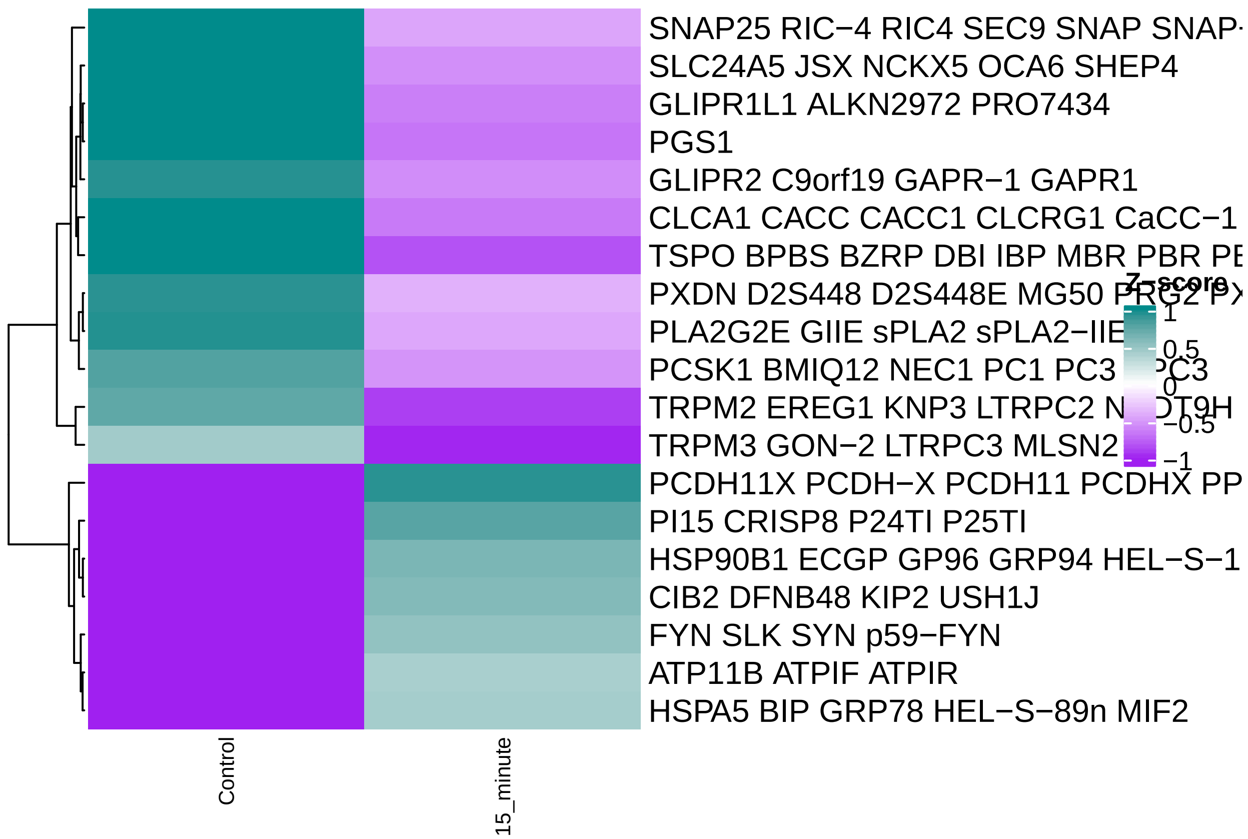
```r
MMR60UPDOWN_hm <- cbind(MMR60UPDOWN$z0, MMR60UPDOWN$z15)
rownames(MMR60UPDOWN_hm) <- c(MMR60UPDOWN$Transcript...Blast.Sequence.Features...Blast.Domain...Symbol)
colnames(MMR60UPDOWN_hm) <- c("Control", "60_minute")

################################################################################ ieg ##################
ieg15_hm <- cbind(ieg15$z0, ieg15$z15)
rownames(ieg15_hm) <- c(ieg15$Symbol)
colnames(ieg15_hm) <- c("Control", "15_minute")

ieg60_hm <- cbind(ieg60$z0, ieg60$z15)
rownames(ieg60_hm) <- c(ieg60$Symbol)
colnames(ieg60_hm) <- c("Control", "60_minute")

ieg15UPDOWN_hm <- cbind(ieg15UPDOWN$z0, ieg15UPDOWN$z15)
rownames(ieg15UPDOWN_hm) <- c(ieg15UPDOWN$Symbol)
colnames(ieg15UPDOWN_hm) <- c("Control", "15_minute")

ieg60UPDOWN_hm <- cbind(ieg60UPDOWN$z0, ieg60UPDOWN$z15)
rownames(ieg60UPDOWN_hm) <- c(ieg60UPDOWN$Symbol)
colnames(ieg60UPDOWN_hm) <- c("Control", "60_minute")

################################################################################ piwi #################
piwi15_hm <- cbind(piwi15$z0, piwi15$z15)
rownames(piwi15_hm) <- c(piwi15$Transcript...Blast.Sequence.Features...Blast.Domain...Symbol)
colnames(piwi15_hm) <- c("Control", "15_minute")

piwi60_hm <- cbind(piwi60$z0, piwi60$z15)
rownames(piwi60_hm) <- c(piwi60$Transcript...Blast.Sequence.Features...Blast.Domain...Symbol)
colnames(piwi60_hm) <- c("Control", "60_minute")

piwi15UPDOWN_hm <- cbind(piwi15UPDOWN$z0, piwi15UPDOWN$z15)
rownames(piwi15UPDOWN_hm) <- c(piwi15UPDOWN$Transcript...Blast.Sequence.Features...Blast.Domain...Symbol
colnames(piwi15UPDOWN_hm) <- c("Control", "15_minute")

piwi60UPDOWN_hm <- cbind(piwi60UPDOWN$z0, piwi60UPDOWN$z15)
rownames(piwi60UPDOWN_hm) <- c(piwi60UPDOWN$Transcript...Blast.Sequence.Features...Blast.Domain...Symbol
colnames(piwi60UPDOWN_hm) <- c("Control", "60_minute")

################################################################################ NHEJ #################
NHEJ15_hm <- cbind(NHEJ15$z0, NHEJ15$z15)
rownames(NHEJ15_hm) <- c(NHEJ15$Transcript...Blast.Sequence.Features...Blast.Domain...Symbol)
colnames(NHEJ15_hm) <- c("Control", "15_minute")

NHEJ60_hm <- cbind(NHEJ60$z0, NHEJ60$z15)
rownames(NHEJ60_hm) <- c(NHEJ60$Transcript...Blast.Sequence.Features...Blast.Domain...Symbol)
colnames(NHEJ60_hm) <- c("Control", "60_minute")

NHEJ15UPDOWN_hm <- cbind(NHEJ15UPDOWN$z0, NHEJ15UPDOWN$z15)
rownames(NHEJ15UPDOWN_hm) <- c(NHEJ15UPDOWN$Transcript...Blast.Sequence.Features...Blast.Domain...Symbol
colnames(NHEJ15UPDOWN_hm) <- c("Control", "15_minute")

NHEJ60UPDOWN_hm <- cbind(NHEJ60UPDOWN$z0, NHEJ60UPDOWN$z15)
rownames(NHEJ60UPDOWN_hm) <- c(NHEJ60UPDOWN$Transcript...Blast.Sequence.Features...Blast.Domain...Symbol
colnames(NHEJ60UPDOWN_hm) <- c("Control", "60_minute")

################################################################################ NBFIG ################
NBFIG15_hm <- cbind(NBFIG15$z0, NBFIG15$z15)
rownames(NBFIG15_hm) <- c(NBFIG15$Marker)
colnames(NBFIG15_hm) <- c("Control", "15_minute")

NBFIG60_hm <- cbind(NBFIG60$z0, NBFIG60$z15)
rownames(NBFIG60_hm) <- c(NBFIG60$Marker)
colnames(NBFIG60_hm) <- c("Control", "60_minute")

NBFIG15UPDOWN_hm <- cbind(NBFIG15UPDOWN$z0, NBFIG15UPDOWN$z15)
rownames(NBFIG15UPDOWN_hm) <- c(NBFIG15UPDOWN$Marker)
colnames(NBFIG15UPDOWN_hm) <- c("Control", "15_minute")

NBFIG60UPDOWN_hm <- cbind(NBFIG60UPDOWN$z0, NBFIG60UPDOWN$z15)
rownames(NBFIG60UPDOWN_hm) <- c(NBFIG60UPDOWN$Marker)
colnames(NBFIG60UPDOWN_hm) <- c("Control", "60_minute")
```

Plotting heatmaps of pathways showing significantly differentially expressed transcipts

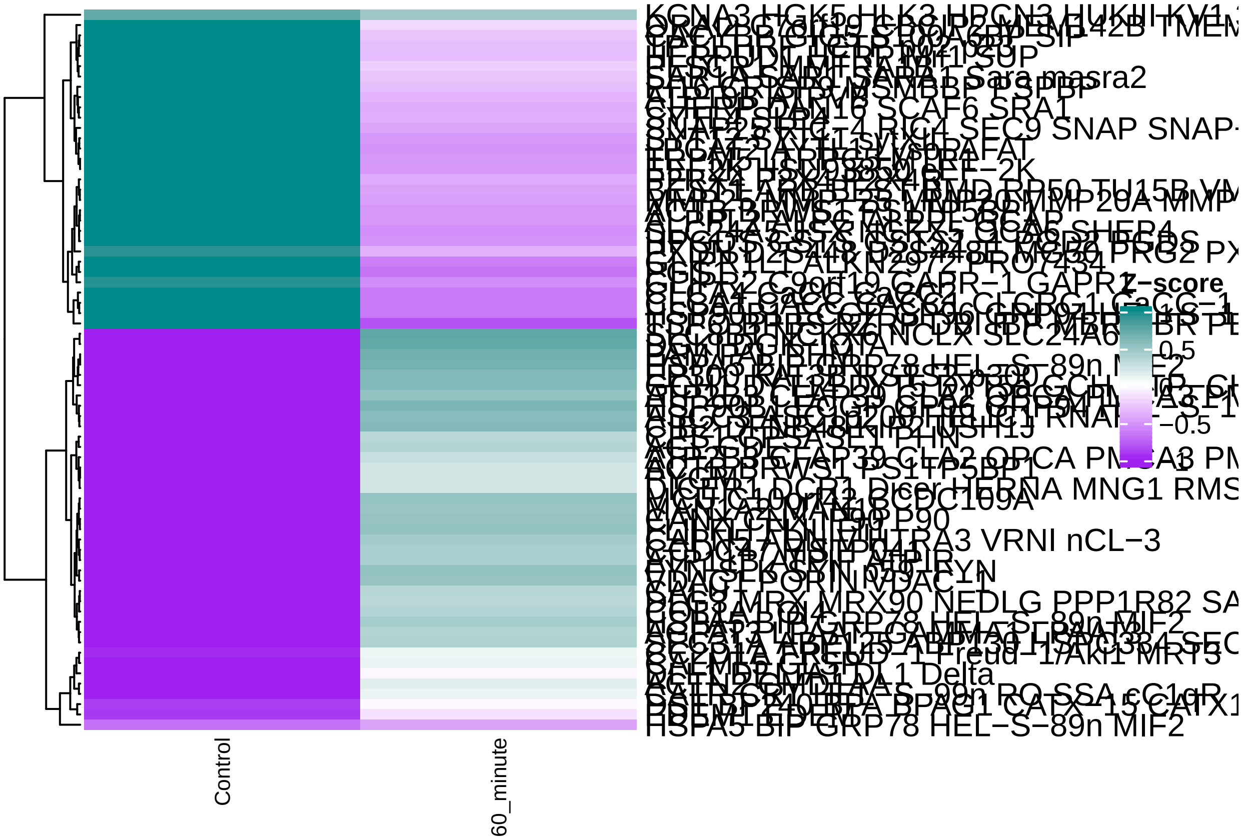
```
##### Heatmaps of sigDE genes for each comparison plotting the
##### mean zscores####
Heatmap(calcium15_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



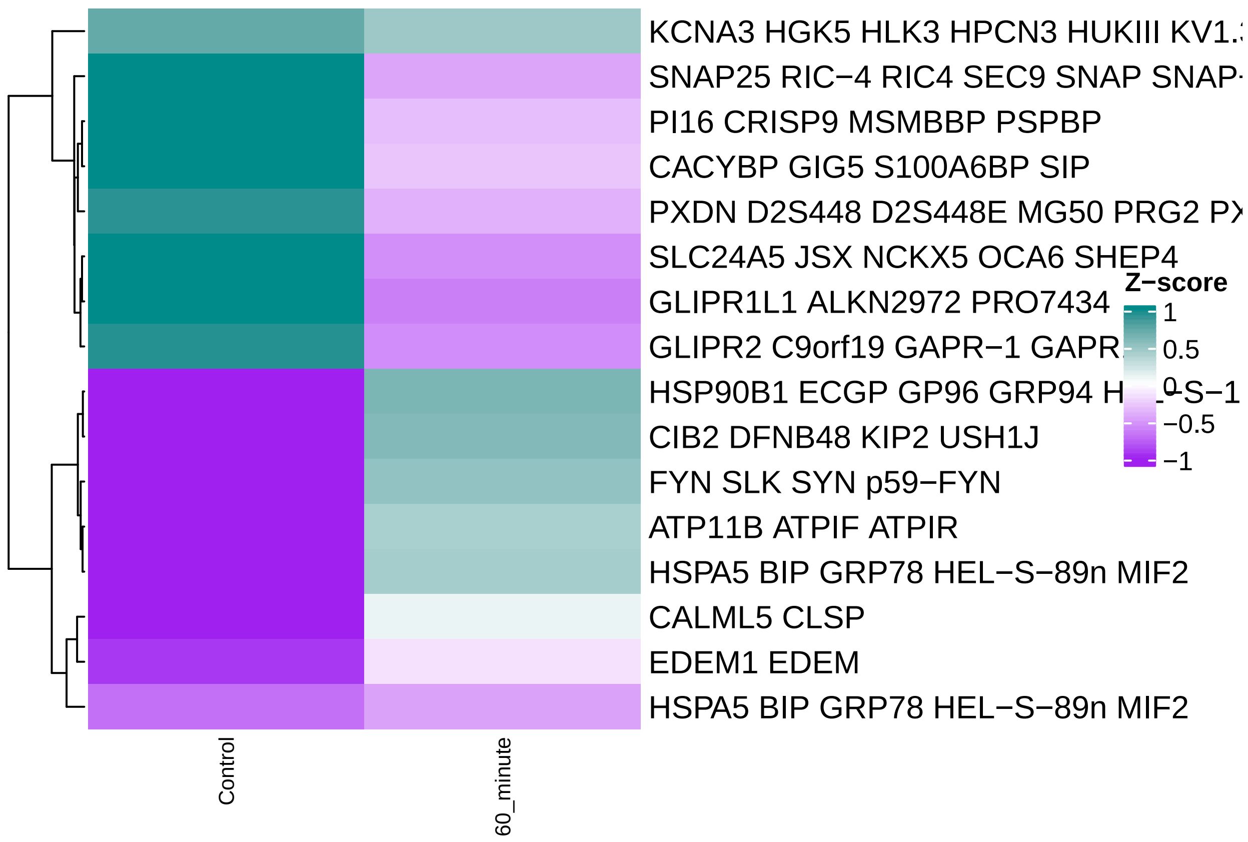
```
Heatmap(calcium15UPDOWN_hm, name = "Z-score", col = myColz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



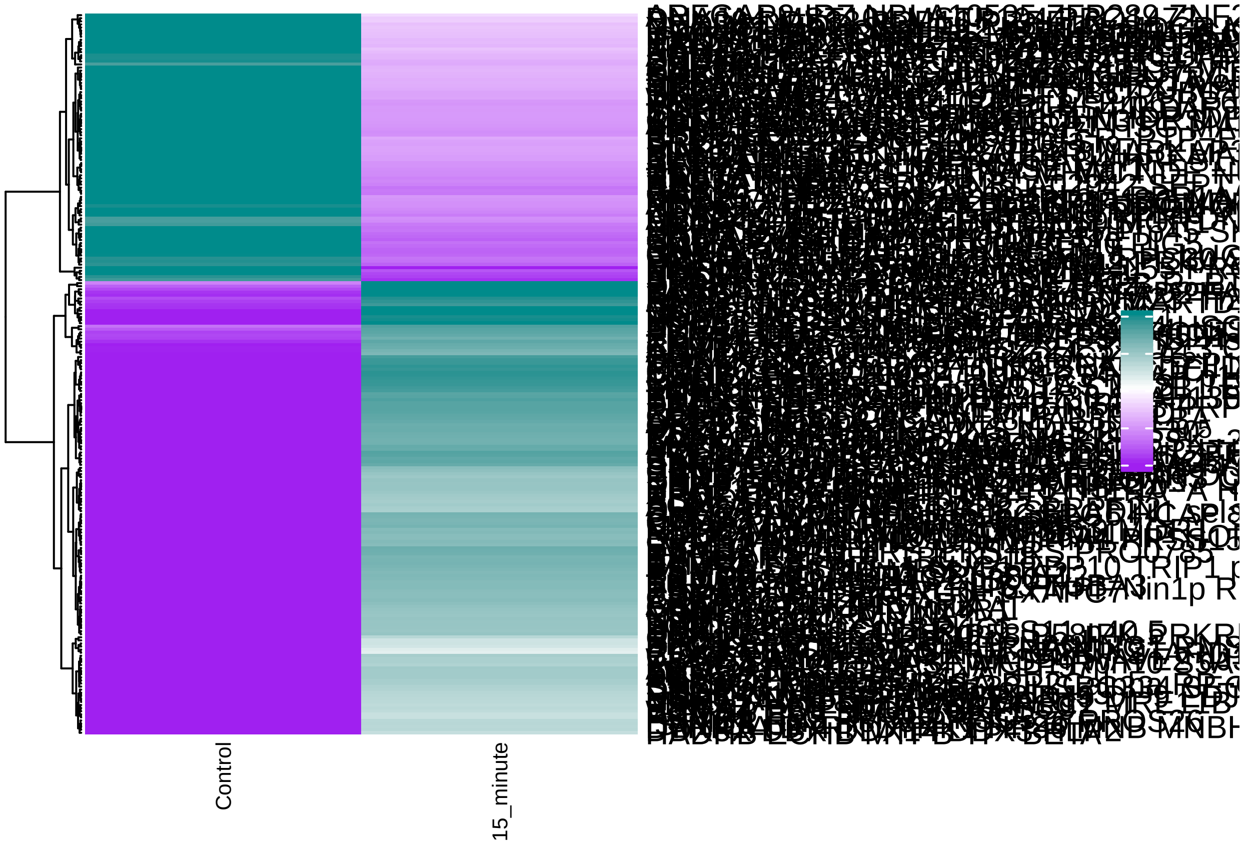
```r
Heatmap(calcium60_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



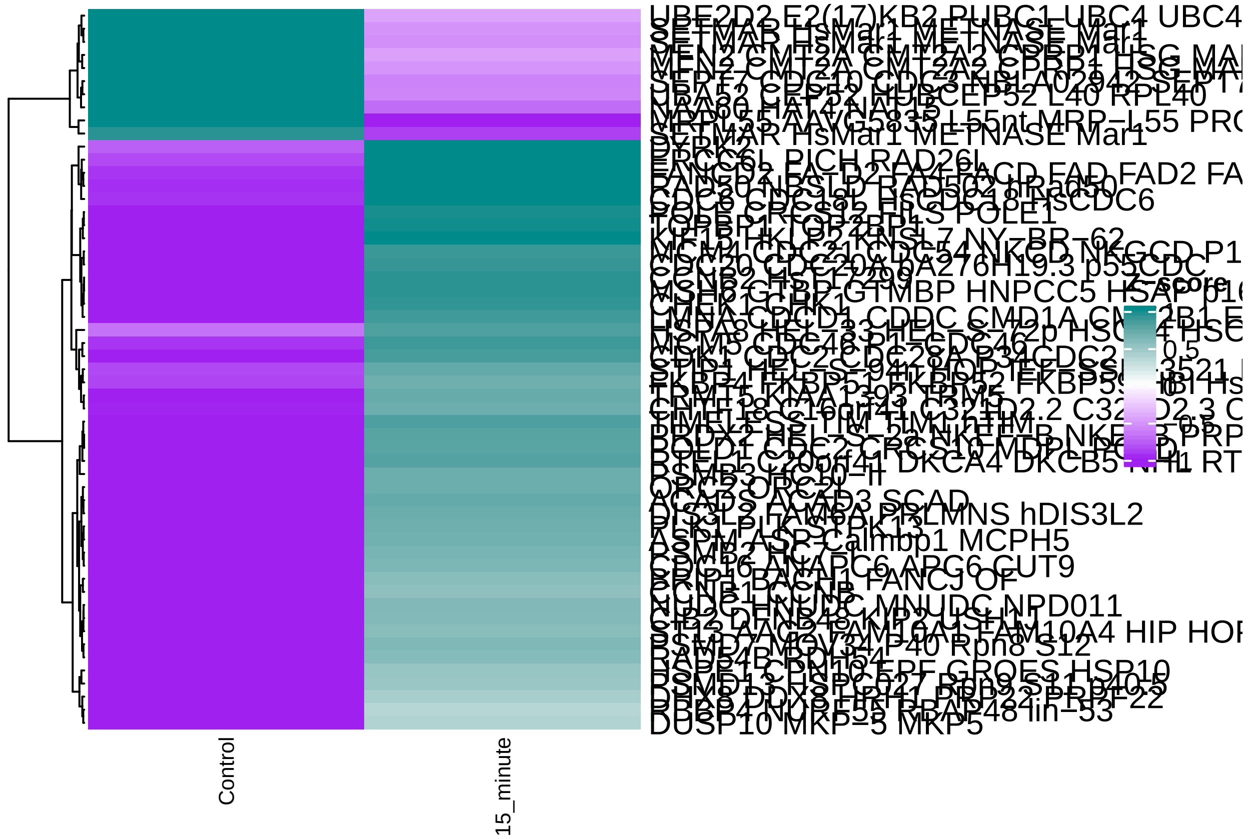
```r
Heatmap(calcium60UPDOWN_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



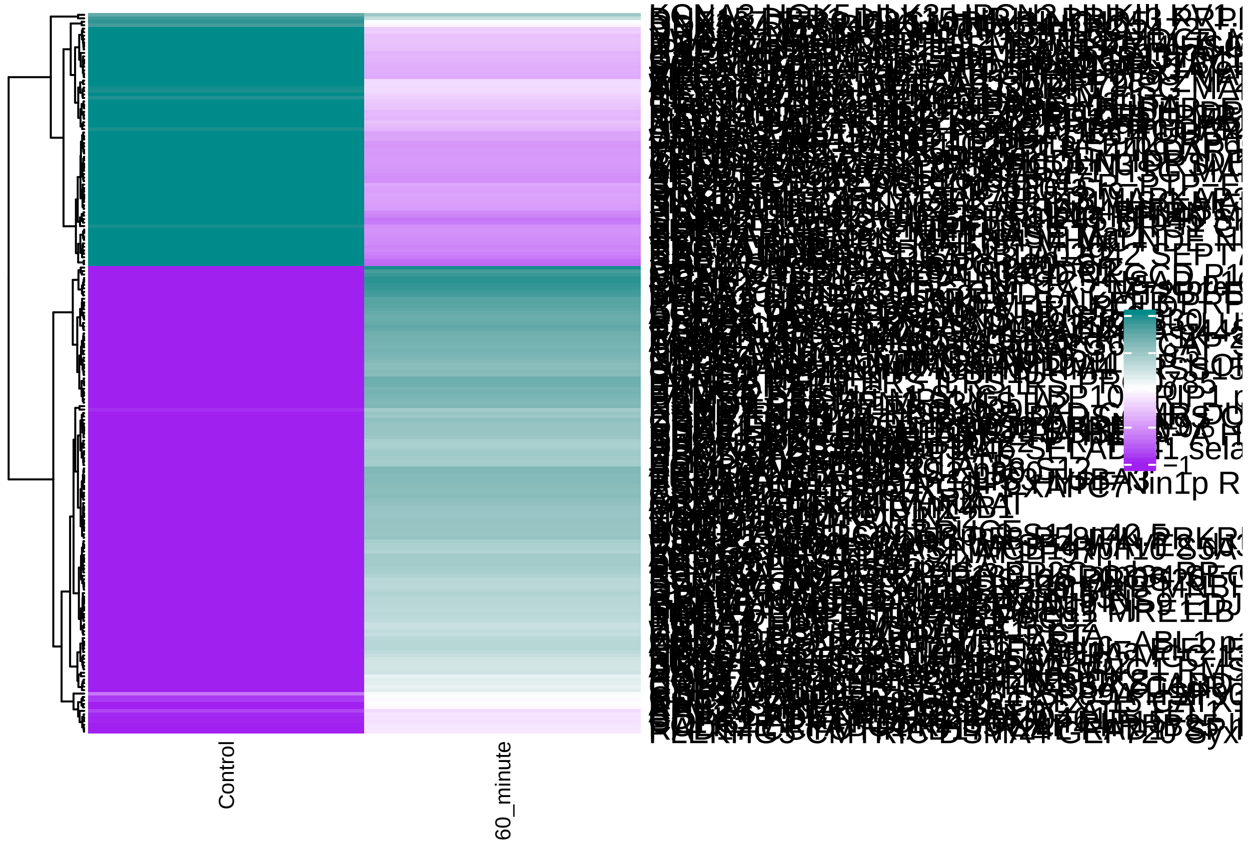
```
Heatmap(cellcycle15_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



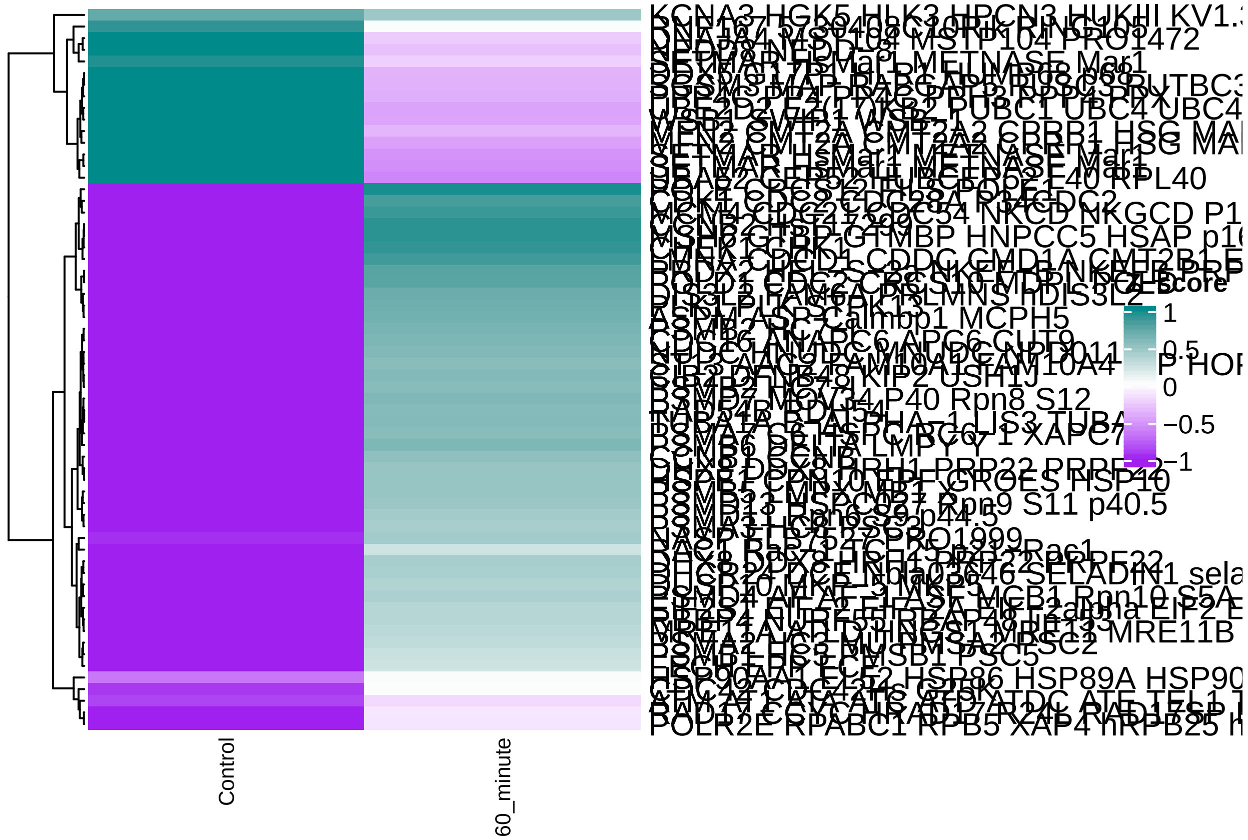
```r
Heatmap(cellcycle15UPDOWN_hm, name = "Z-score", col = mycolz,
    column_names_gp = gpar(fontsize = 8), cluster_columns = FALSE,
    cluster_rows = TRUE)
```



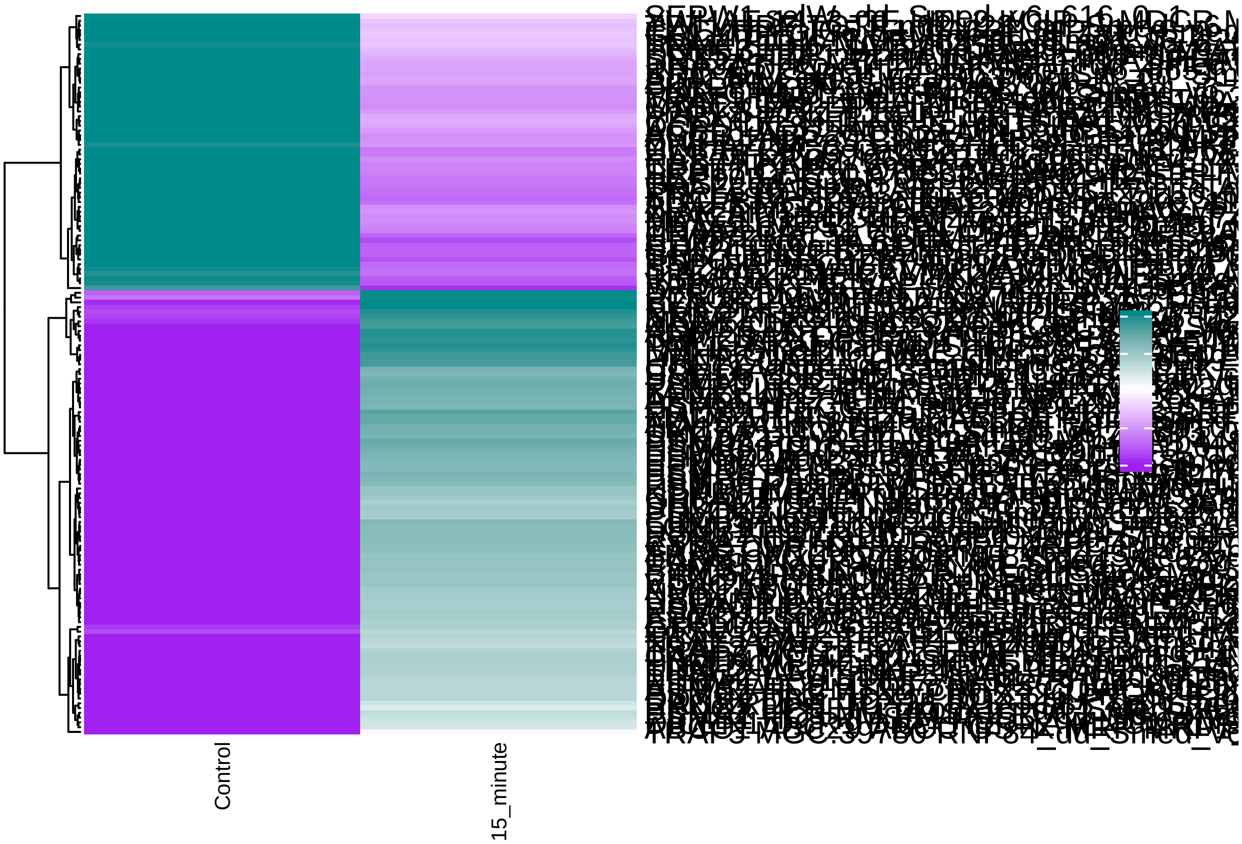
```r
Heatmap(cellcycle60_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



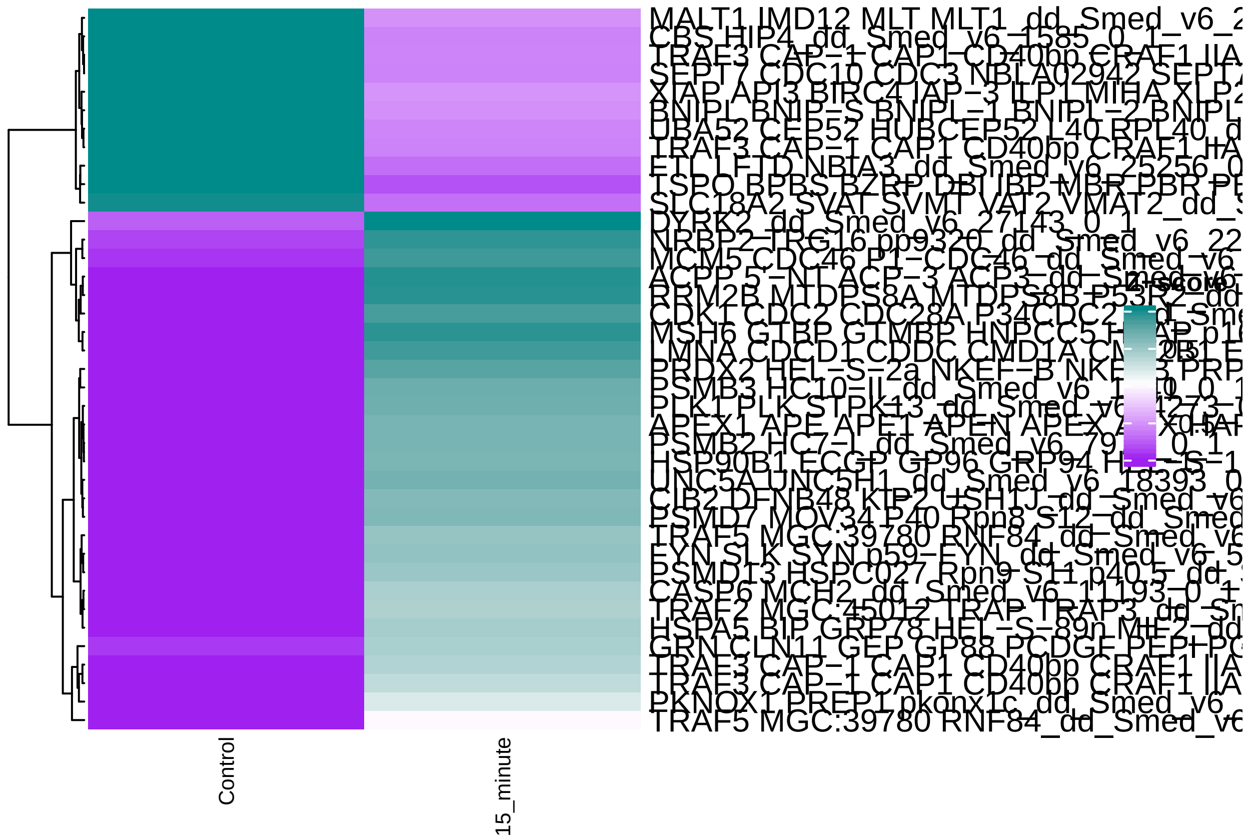
```
Heatmap(cellcycle60UPDOWN_hm, name = "Z-score", col = mycolz,
    column_names_gp = gpar(fontsize = 8), cluster_columns = FALSE,
    cluster_rows = TRUE)
```



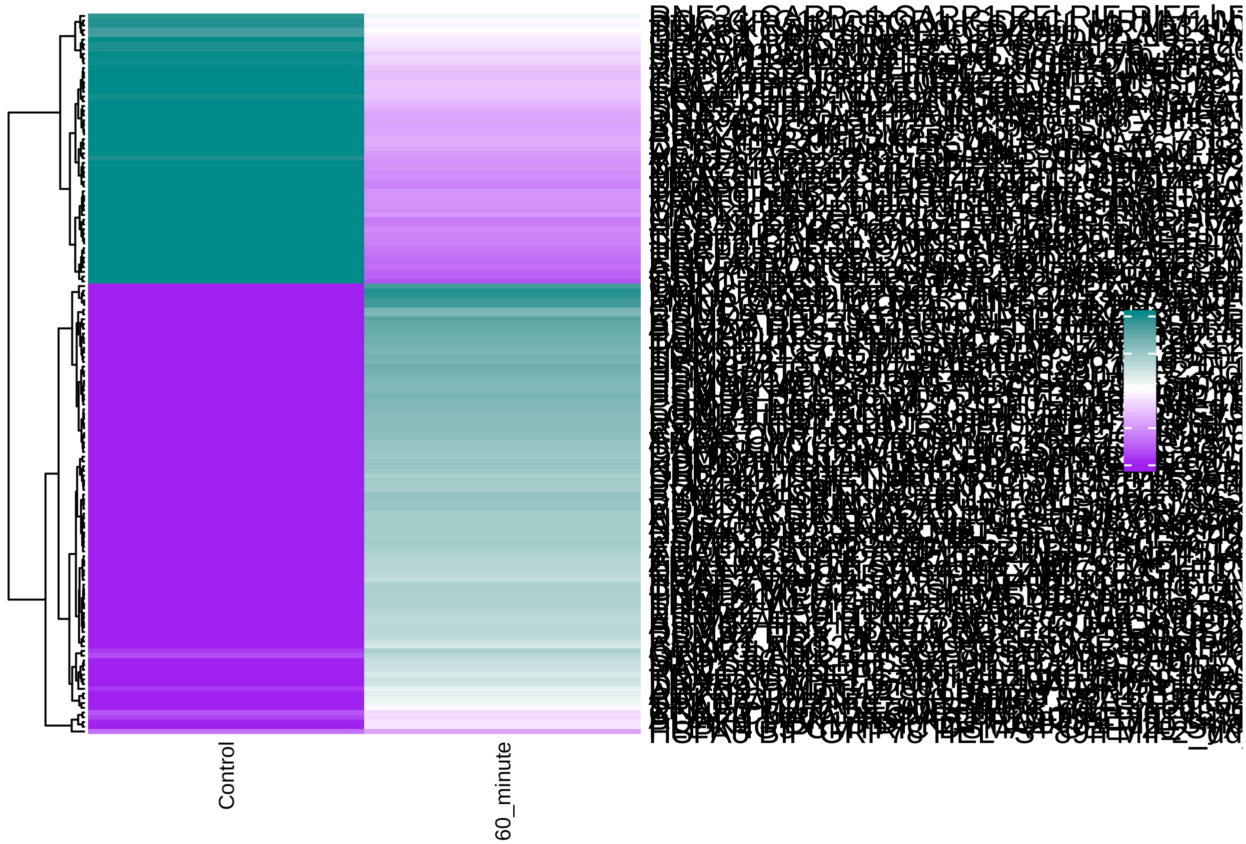
```
Heatmap(celldeath15_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



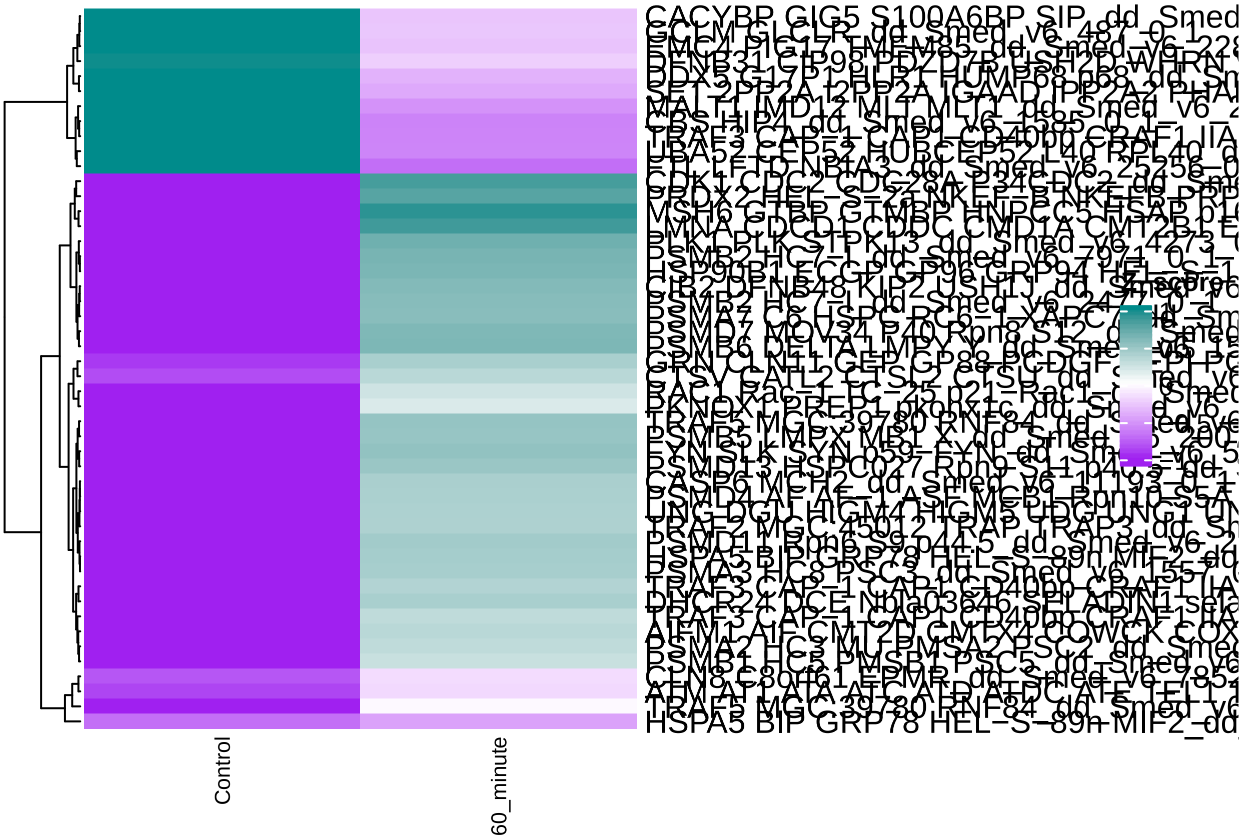
```r
Heatmap(celldeath15UPDOWN_hm, name = "Z-score", col = mycolz,
    column_names_gp = gpar(fontsize = 8), cluster_columns = FALSE,
    cluster_rows = TRUE)
```



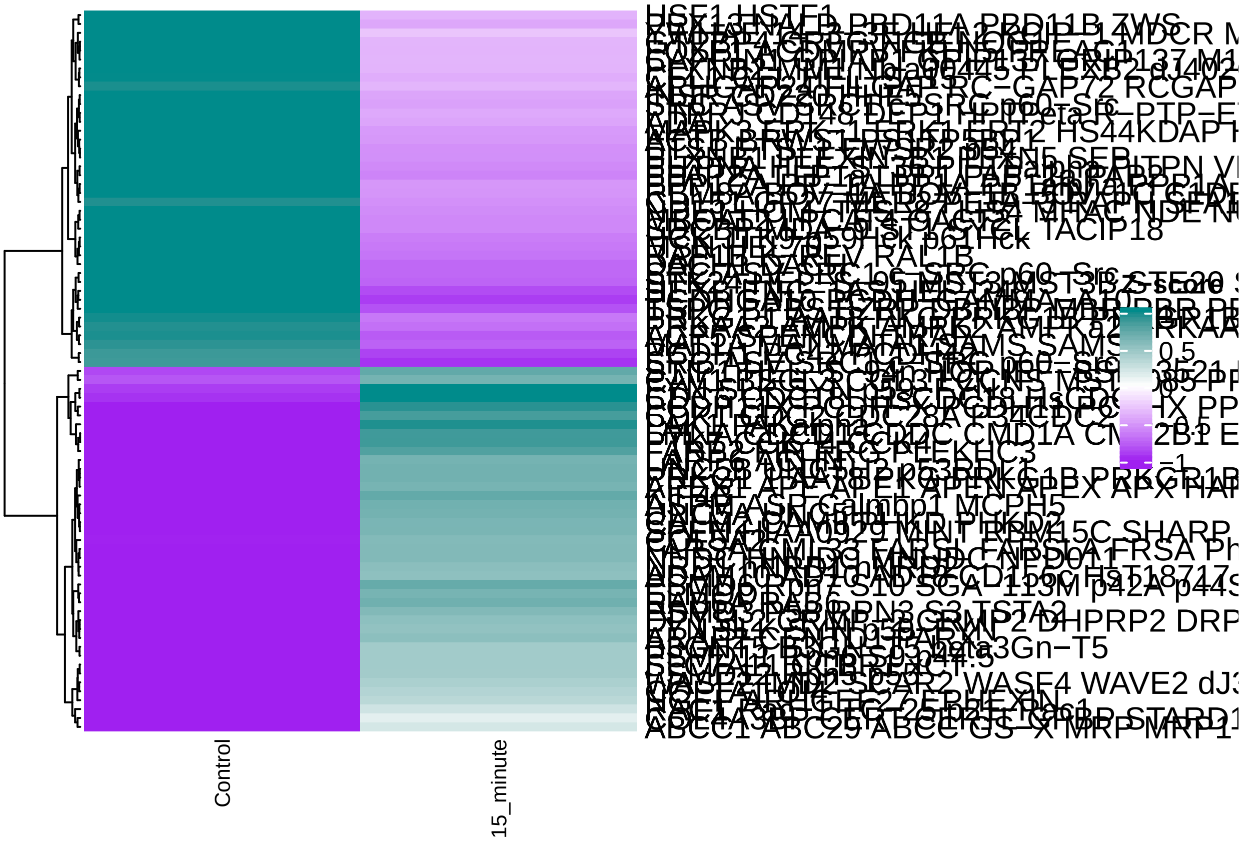
```r
Heatmap(celldeath60_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



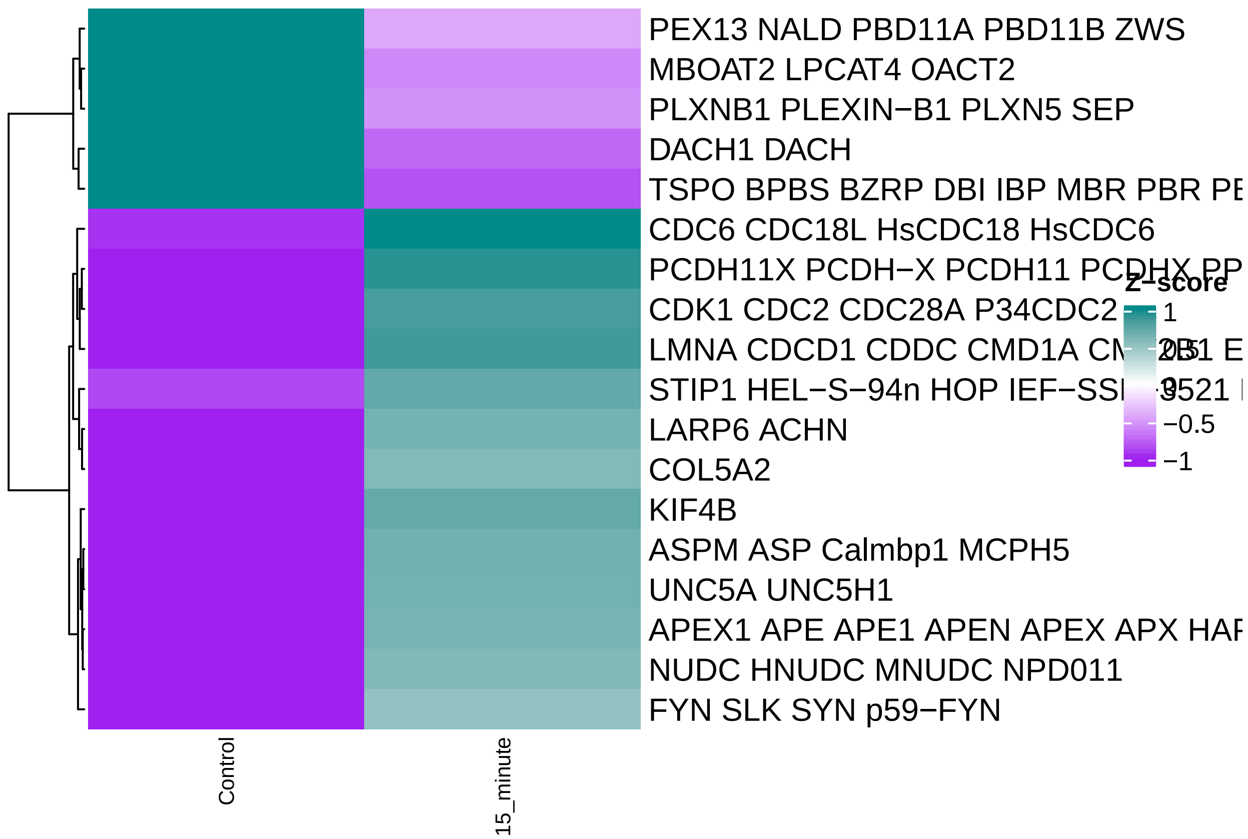
```r
Heatmap(celldeath60UPDOWN_hm, name = "Z-score", col = mycolz,
    column_names_gp = gpar(fontsize = 8), cluster_columns = FALSE,
    cluster_rows = TRUE)
```



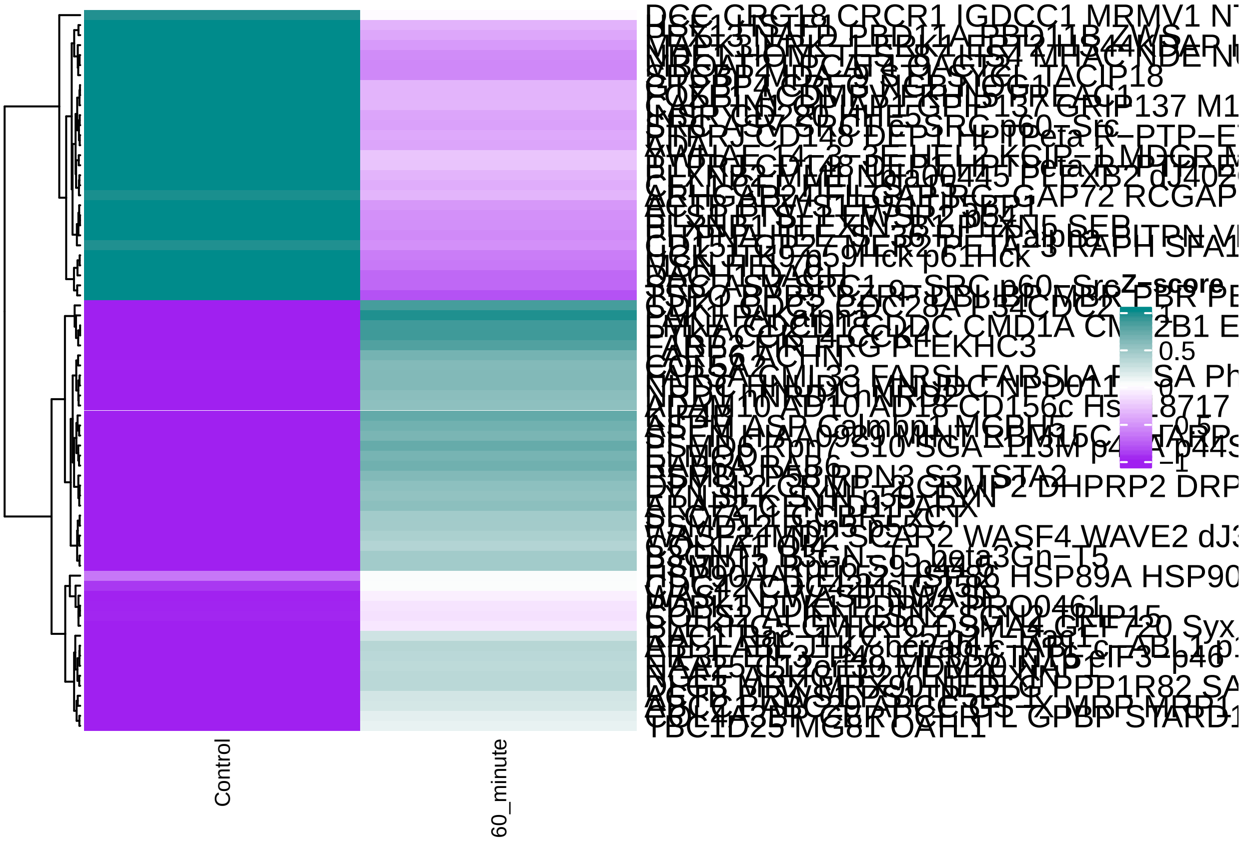
```r
Heatmap(cellmigration15_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



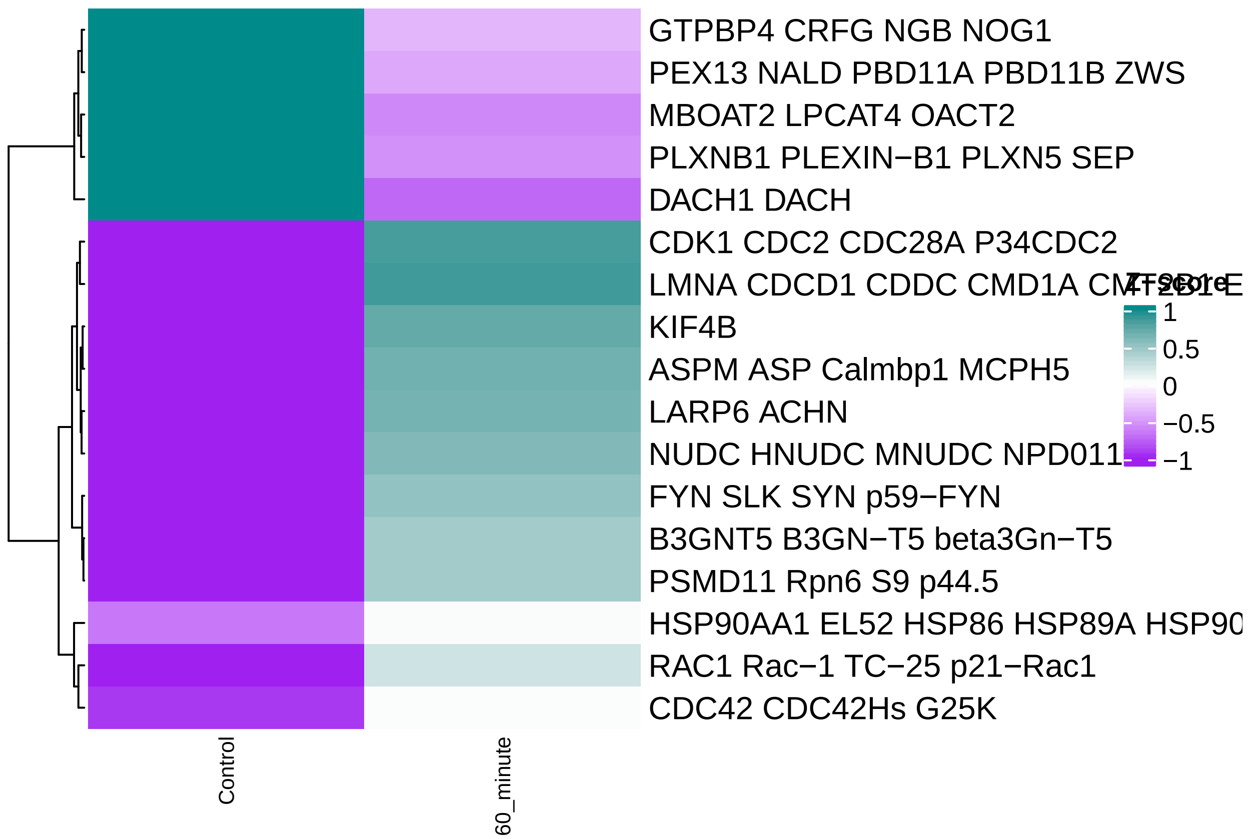
```r
Heatmap(cellmigration15UPDOWN_hm, name = "Z-score", col = mycolz,
    column_names_gp = gpar(fontsize = 8), cluster_columns = FALSE,
    cluster_rows = TRUE)
```



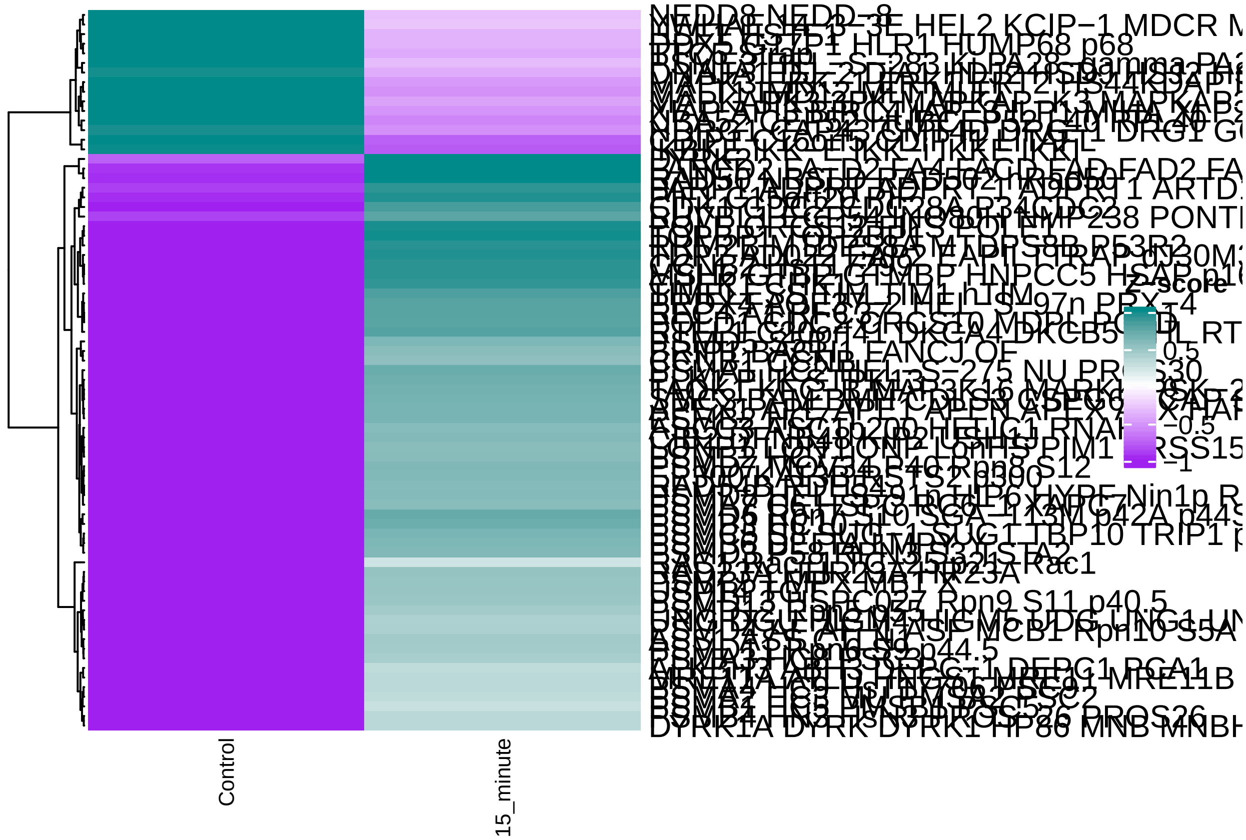
```r
Heatmap(cellmigration60_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



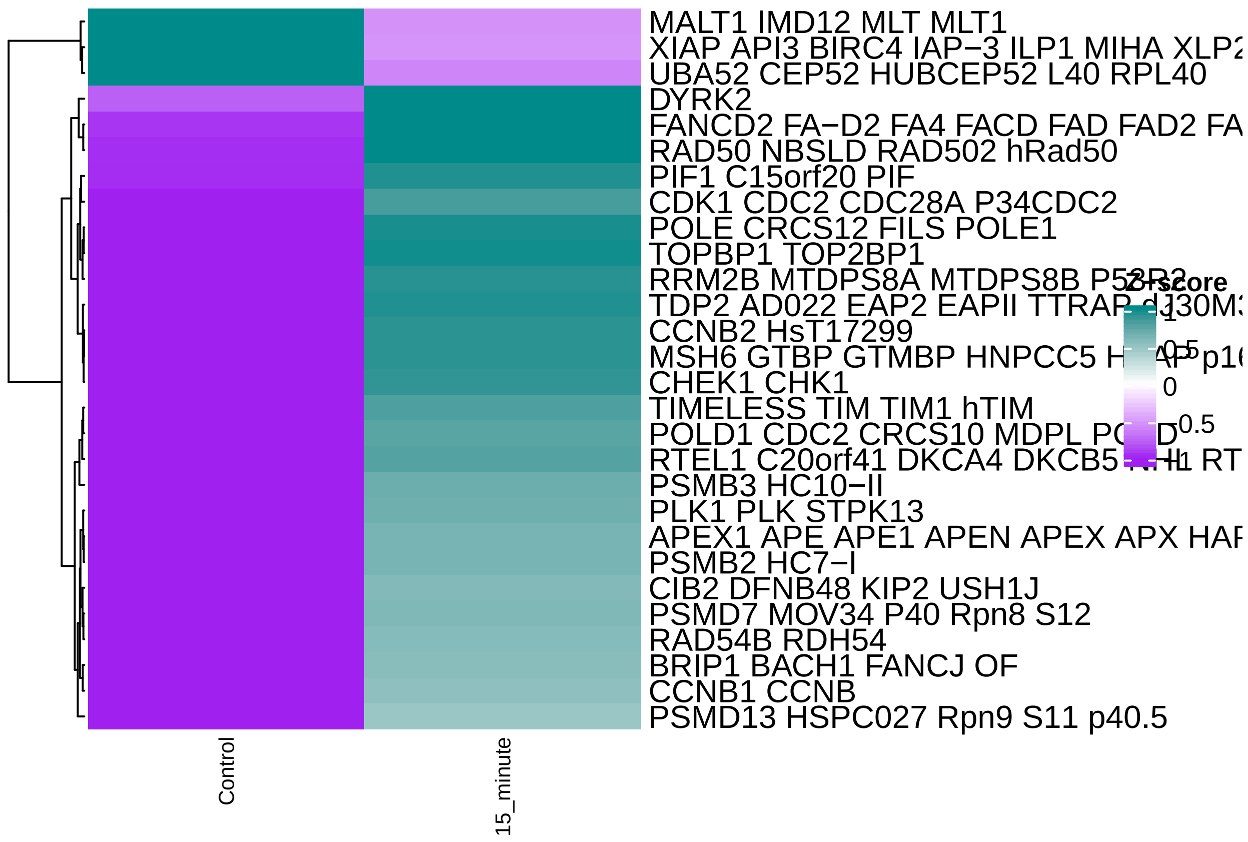
```
Heatmap(cellmigration60UPDOWN_hm, name = "Z-score", col = mycolz,
    column_names_gp = gpar(fontsize = 8), cluster_columns = FALSE,
    cluster_rows = TRUE)
```



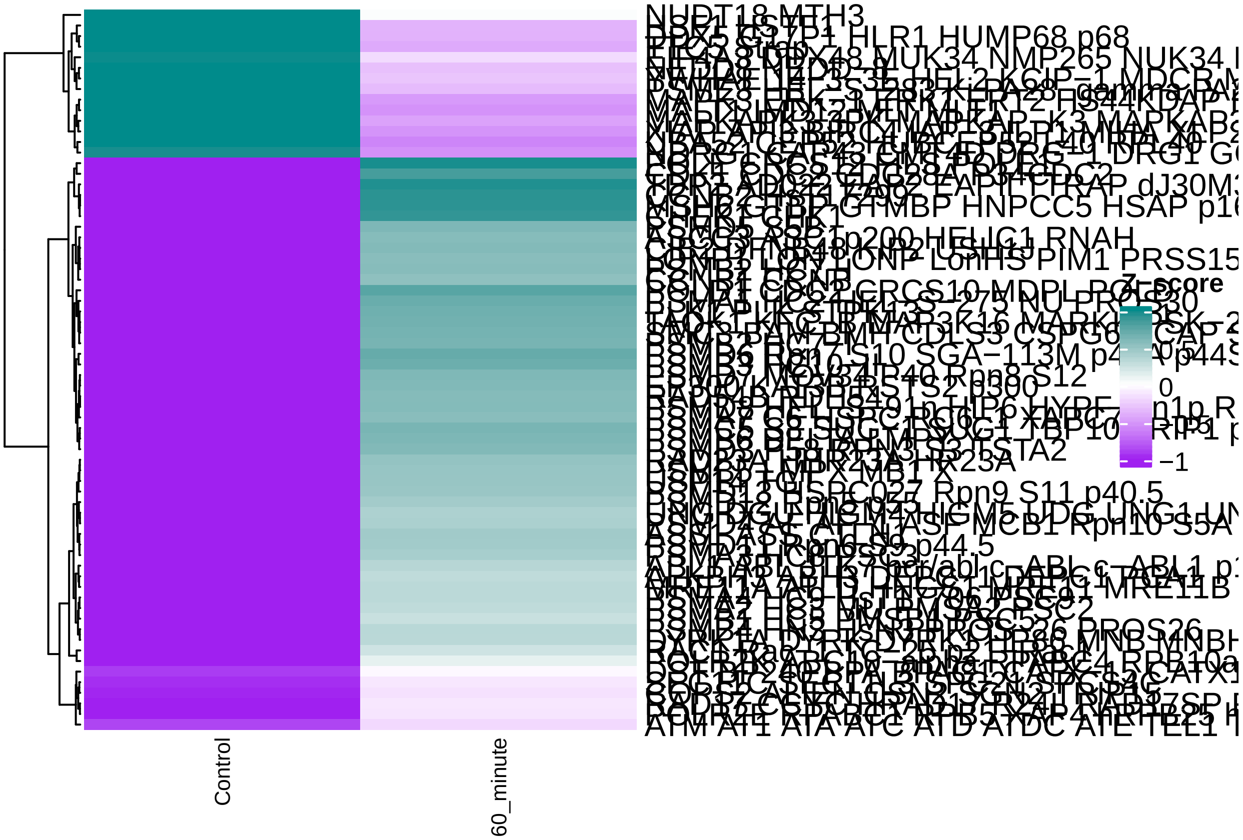
```
Heatmap(DNAdamage15_hm, name = "Z-score", col = myco1z, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



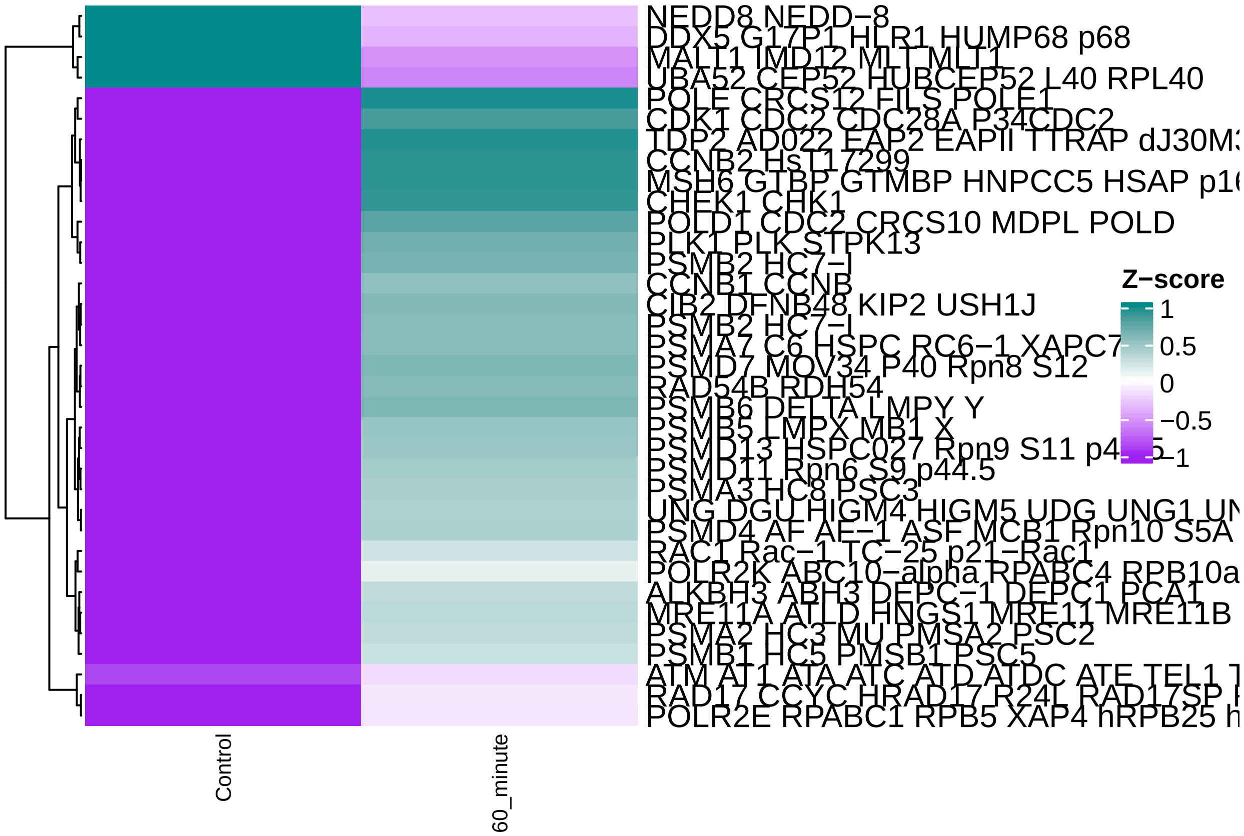
```r
Heatmap(DNAdamage15UPDOWN_hm, name = "Z-score", col = mycolz,
    column_names_gp = gpar(fontsize = 8), cluster_columns = FALSE,
    cluster_rows = TRUE)
```



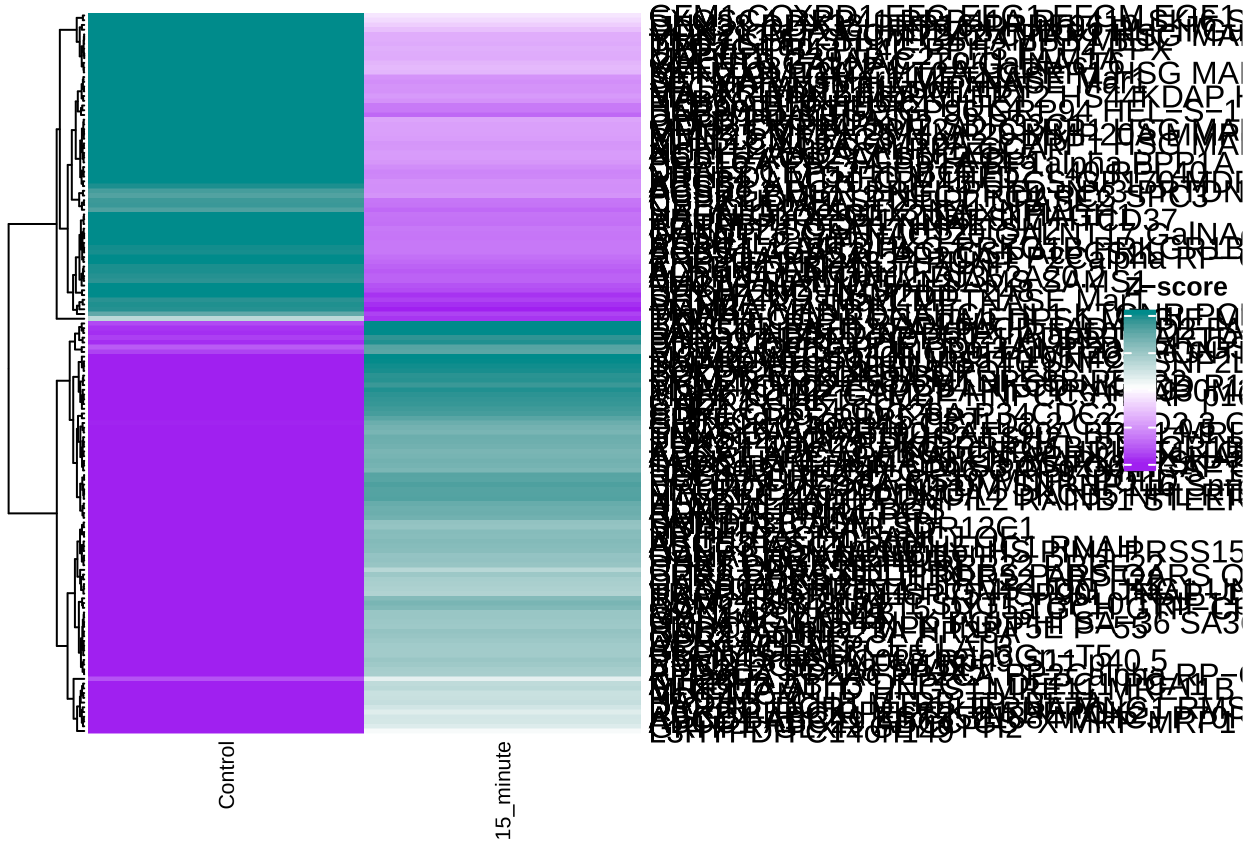
```
Heatmap(DNAdamage60_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



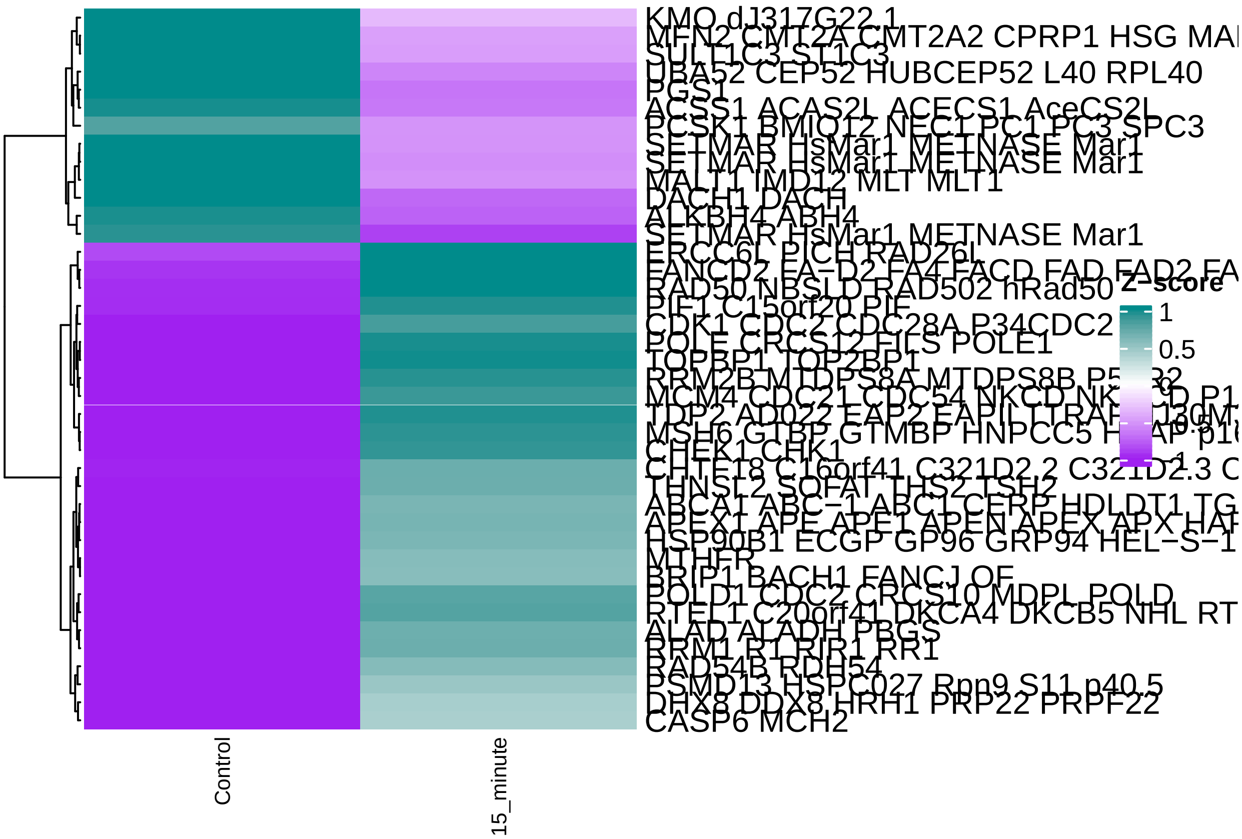
```
Heatmap(DNAdamage60UPDOWN_hm, name = "Z-score", col = mycolz,
    column_names_gp = gpar(fontsize = 8), cluster_columns = FALSE,
    cluster_rows = TRUE)
```



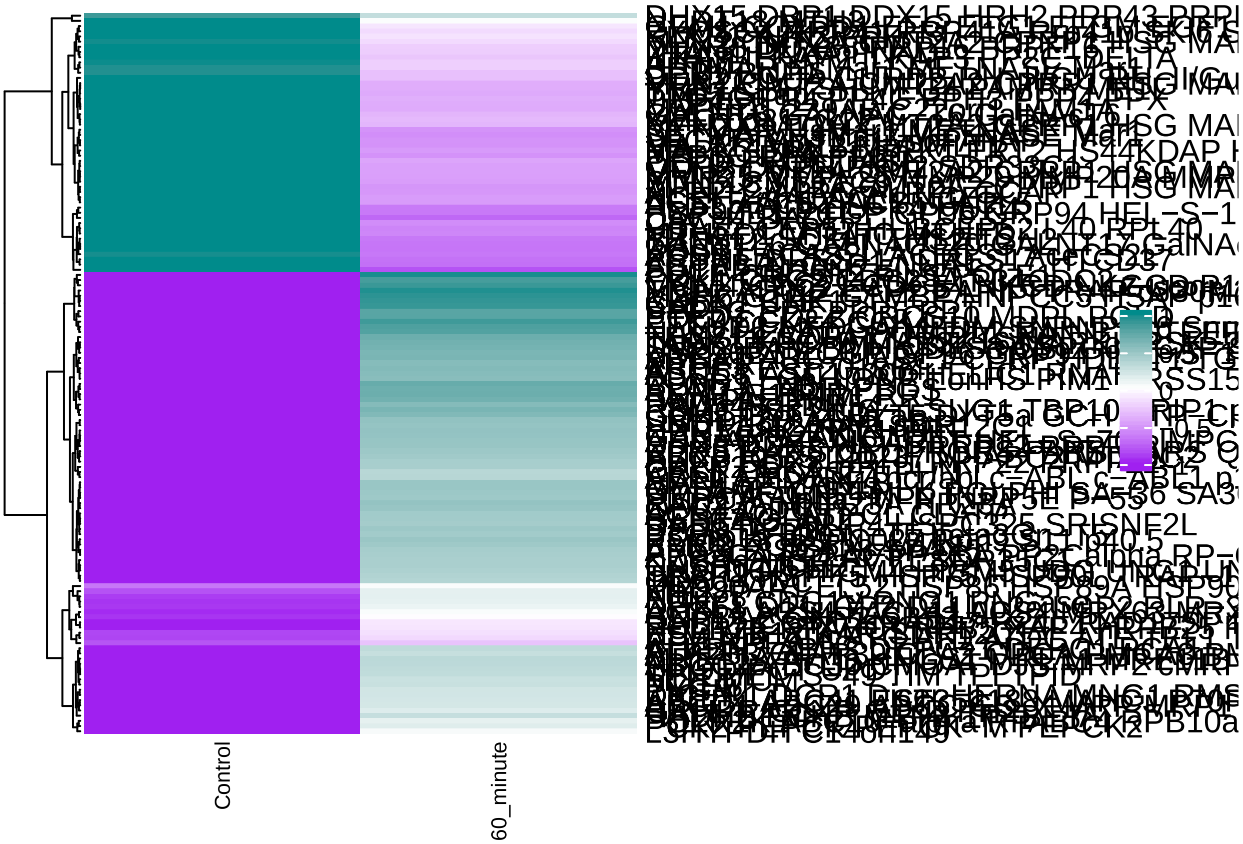
```
Heatmap(DNArepair15_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



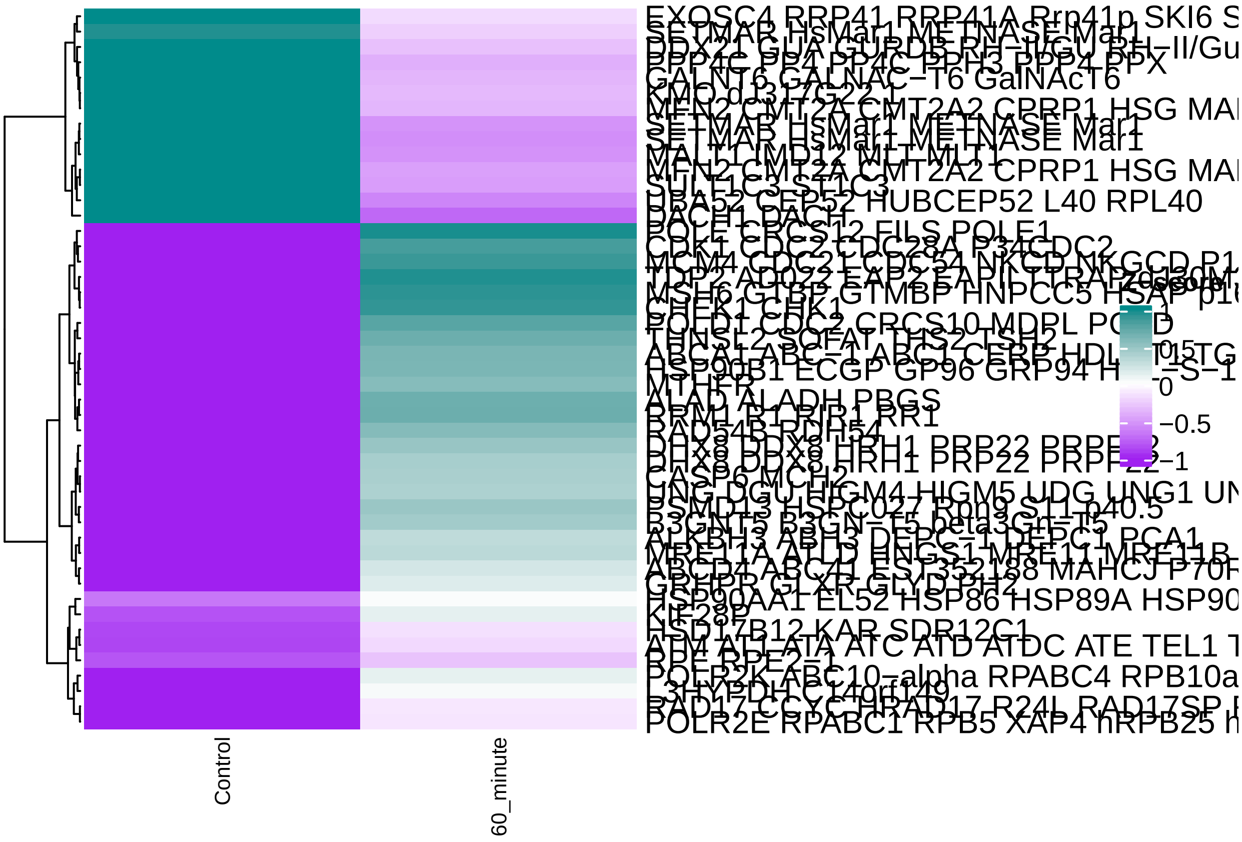
```
Heatmap(DNArepair15UPDOWN_hm, name = "Z-score", col = mycolz,
    column_names_gp = gpar(fontsize = 8), cluster_columns = FALSE,
    cluster_rows = TRUE)
```



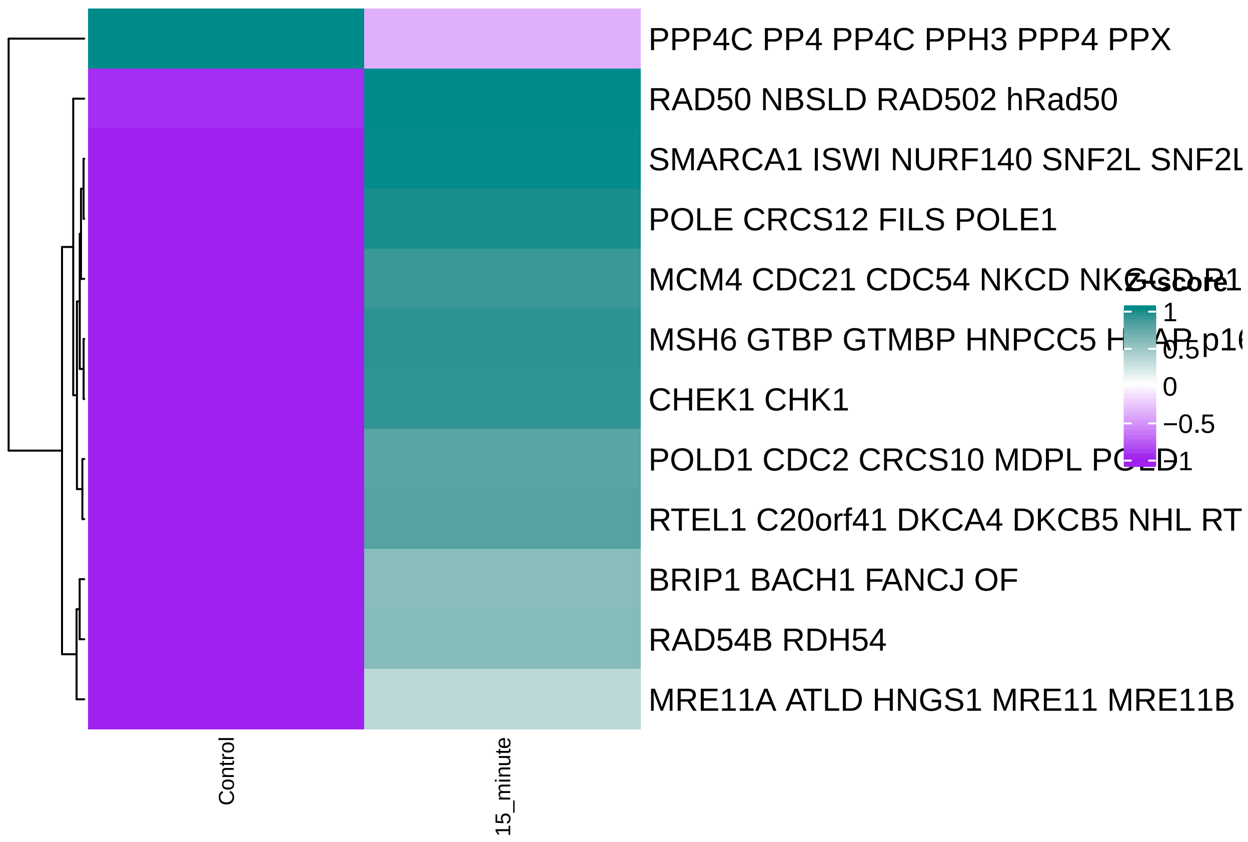
```r
Heatmap(DNArepair60_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



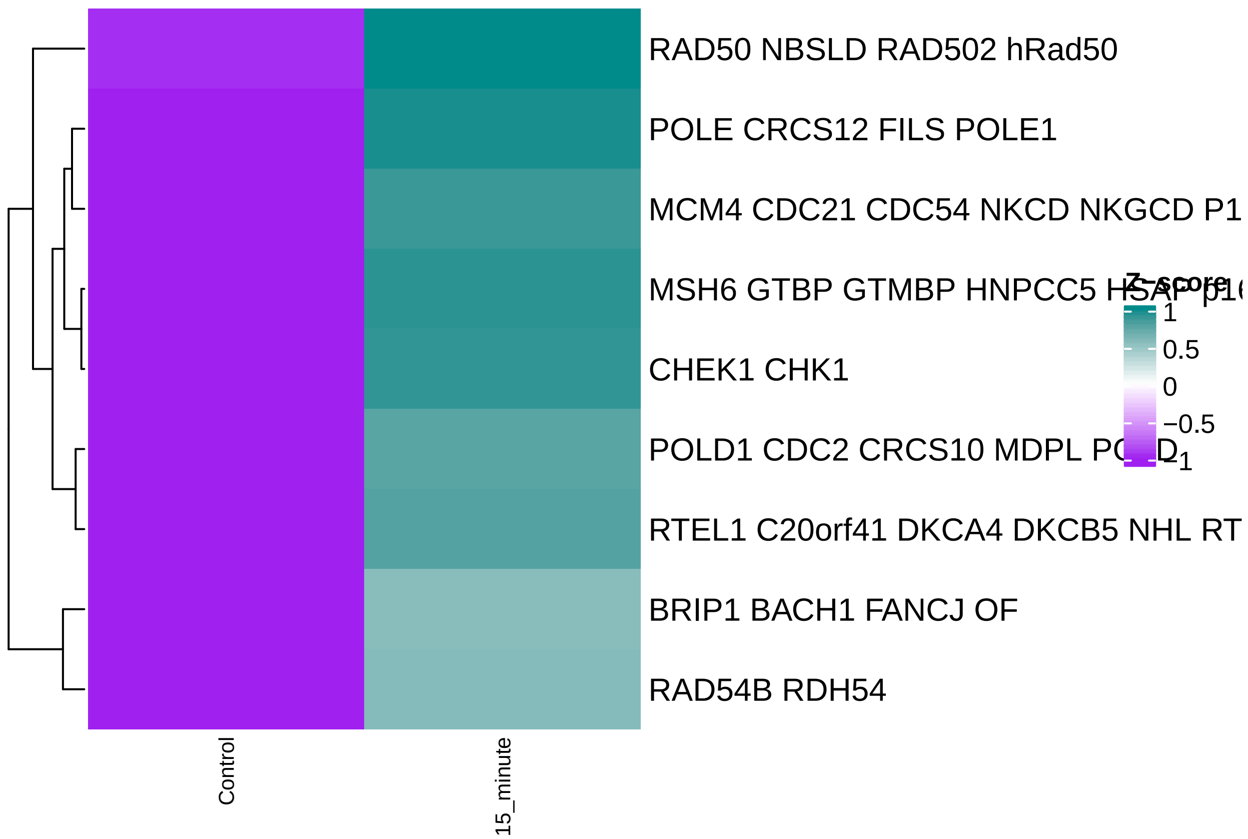
```
Heatmap(DNArepair60UPDOWN_hm, name = "Z-score", col = mycolz,
    column_names_gp = gpar(fontsize = 8), cluster_columns = FALSE,
    cluster_rows = TRUE)
```



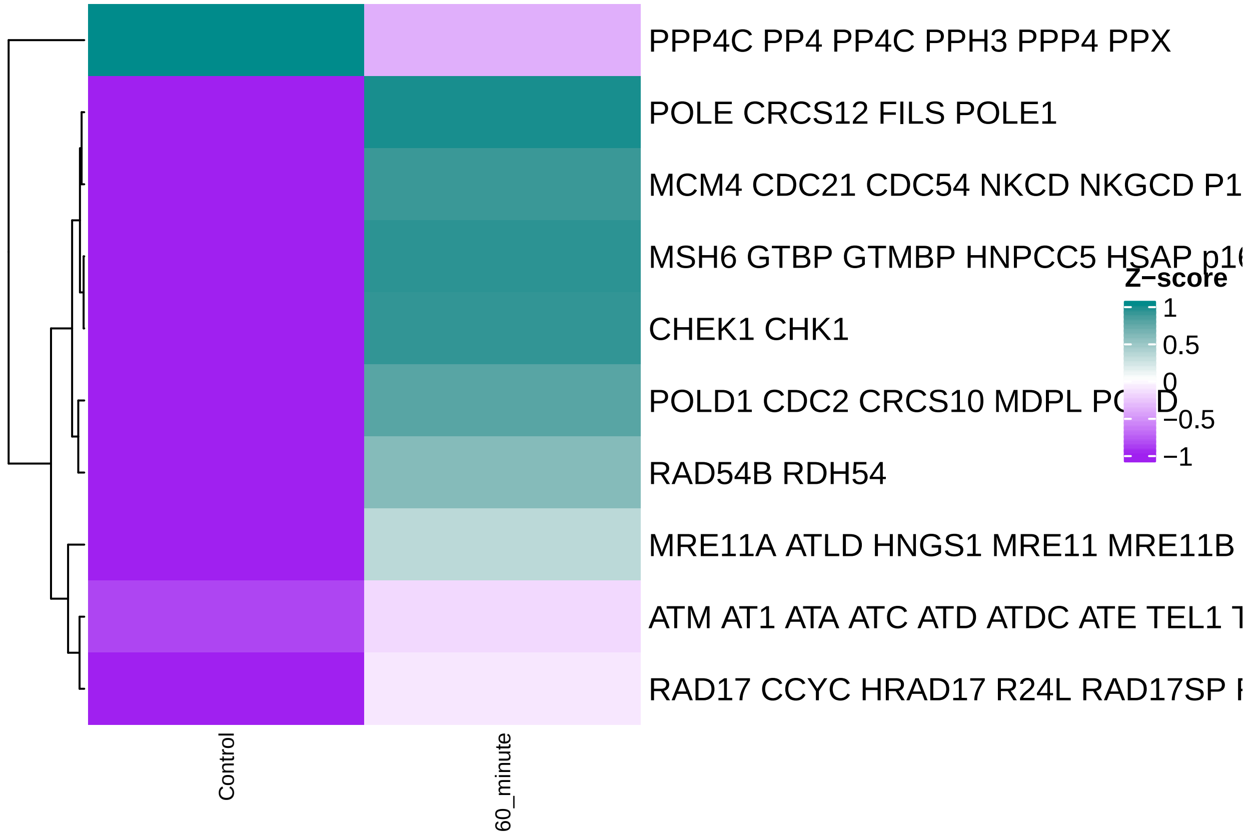
```
Heatmap(HR15_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



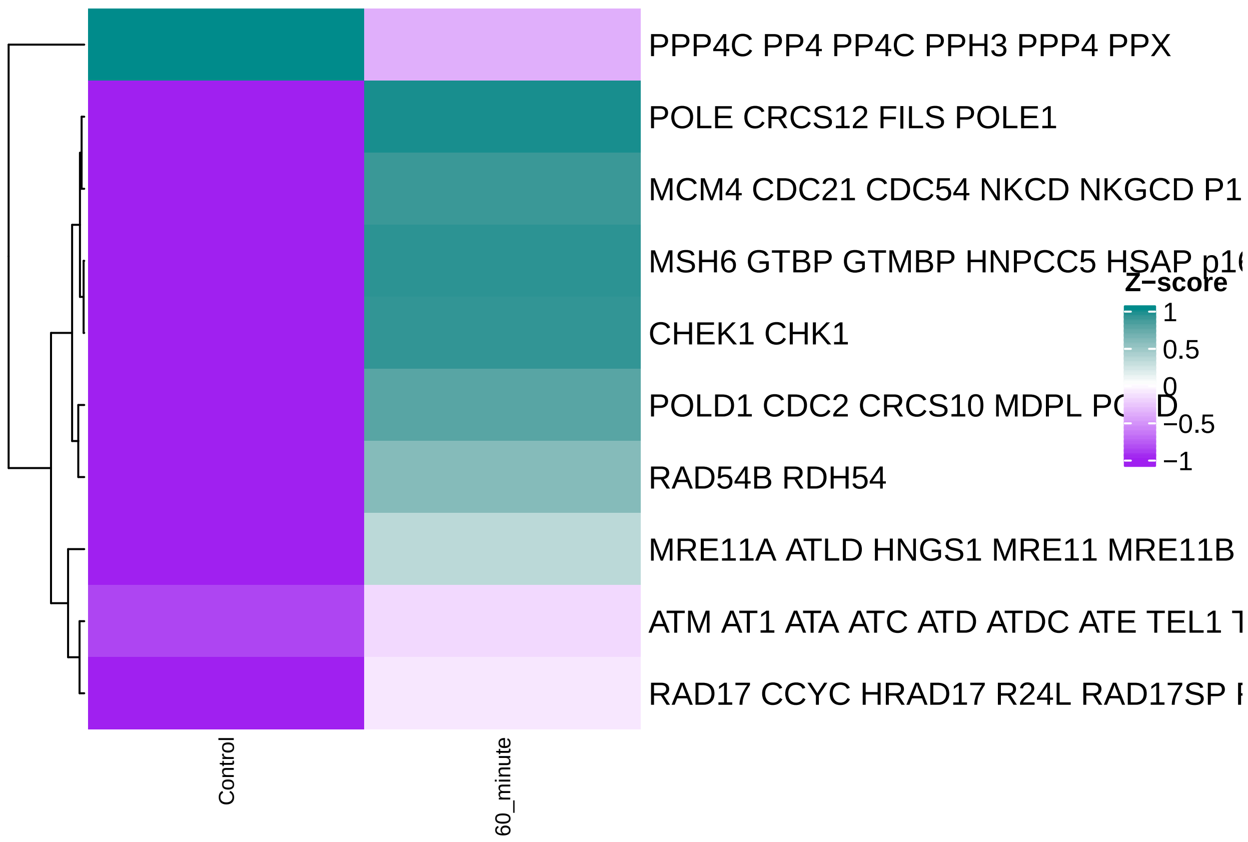
```
Heatmap(HR15UPDOWN_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



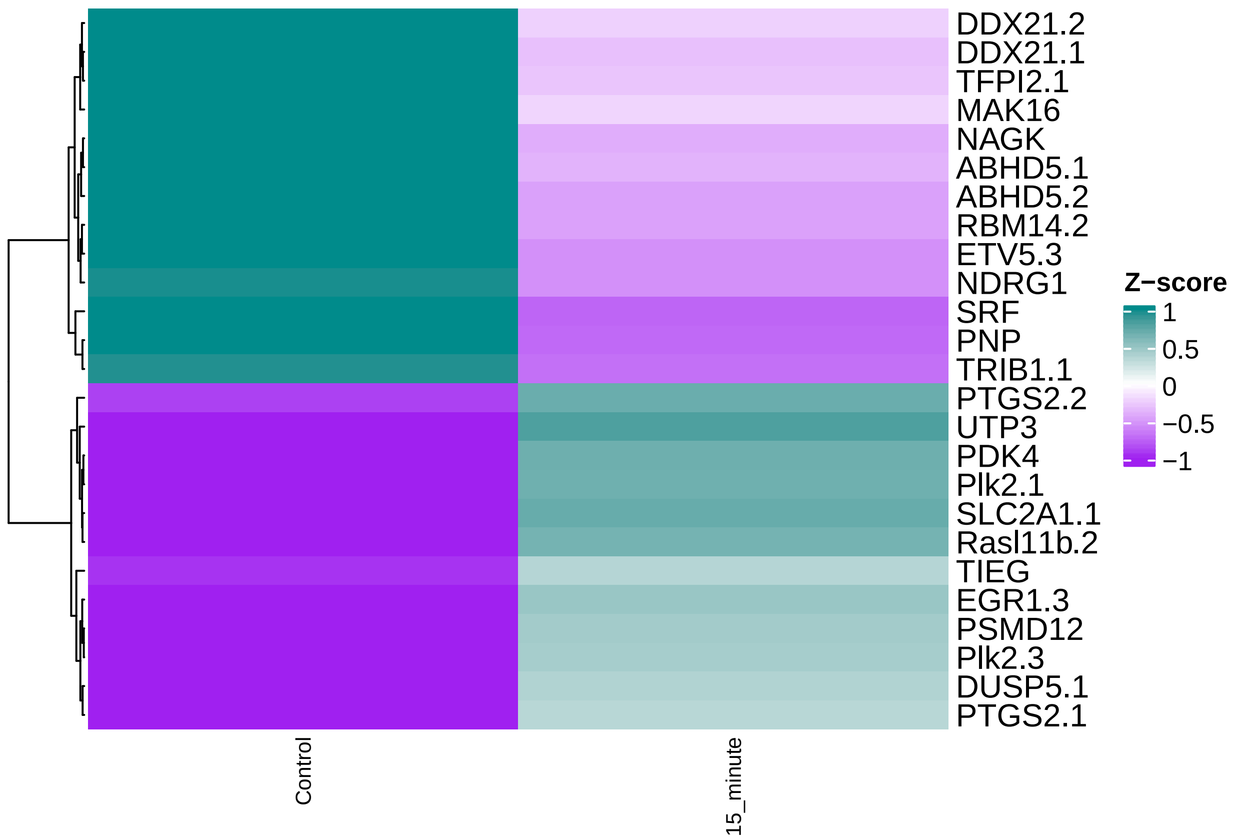
```
Heatmap(HR60_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



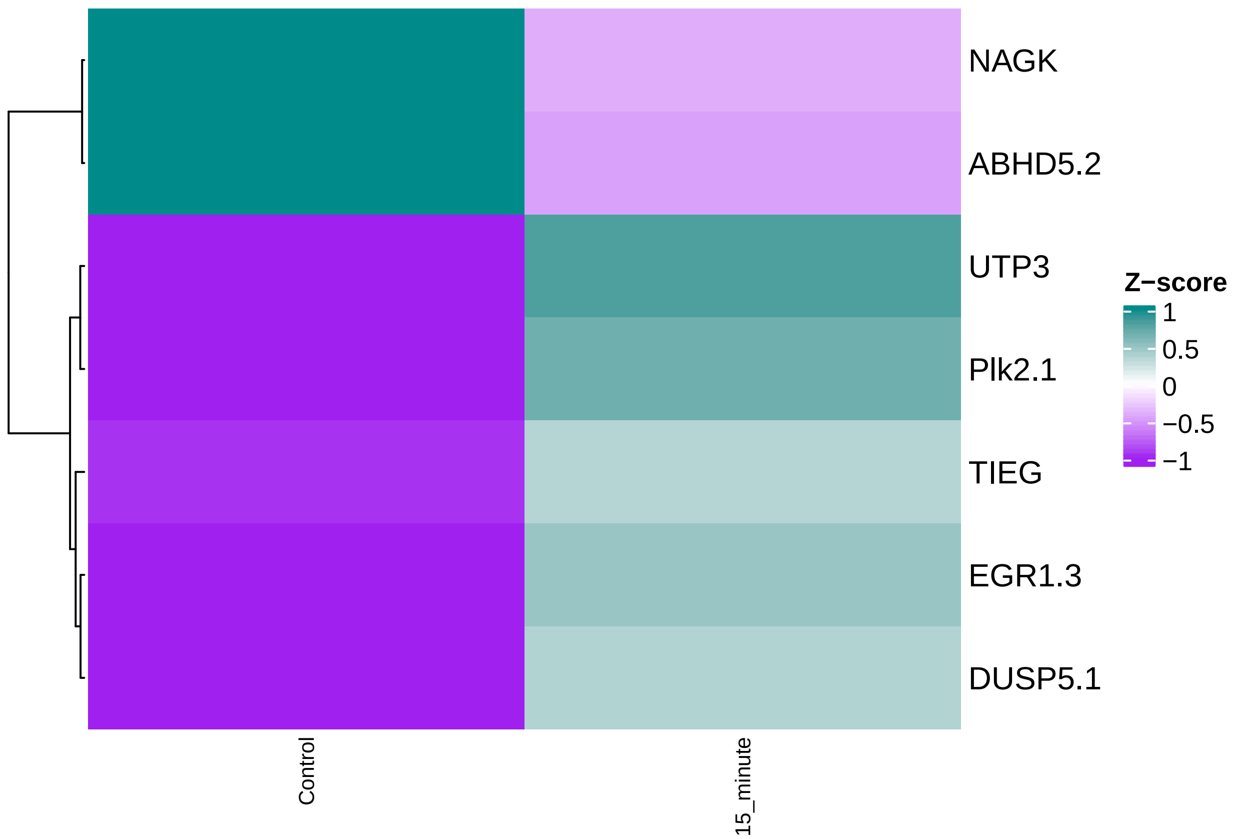
```r
Heatmap(HR60UPDOWN_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



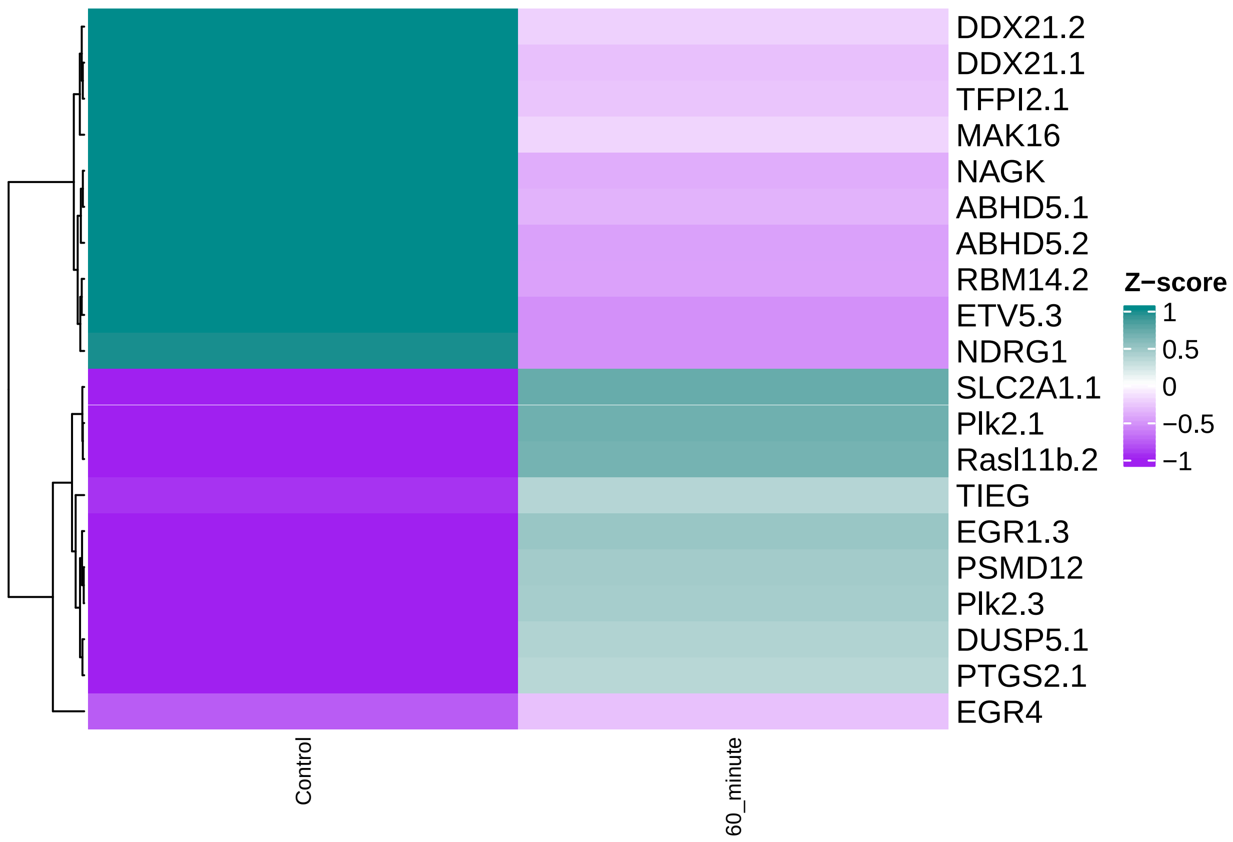
```
Heatmap(ieg15_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



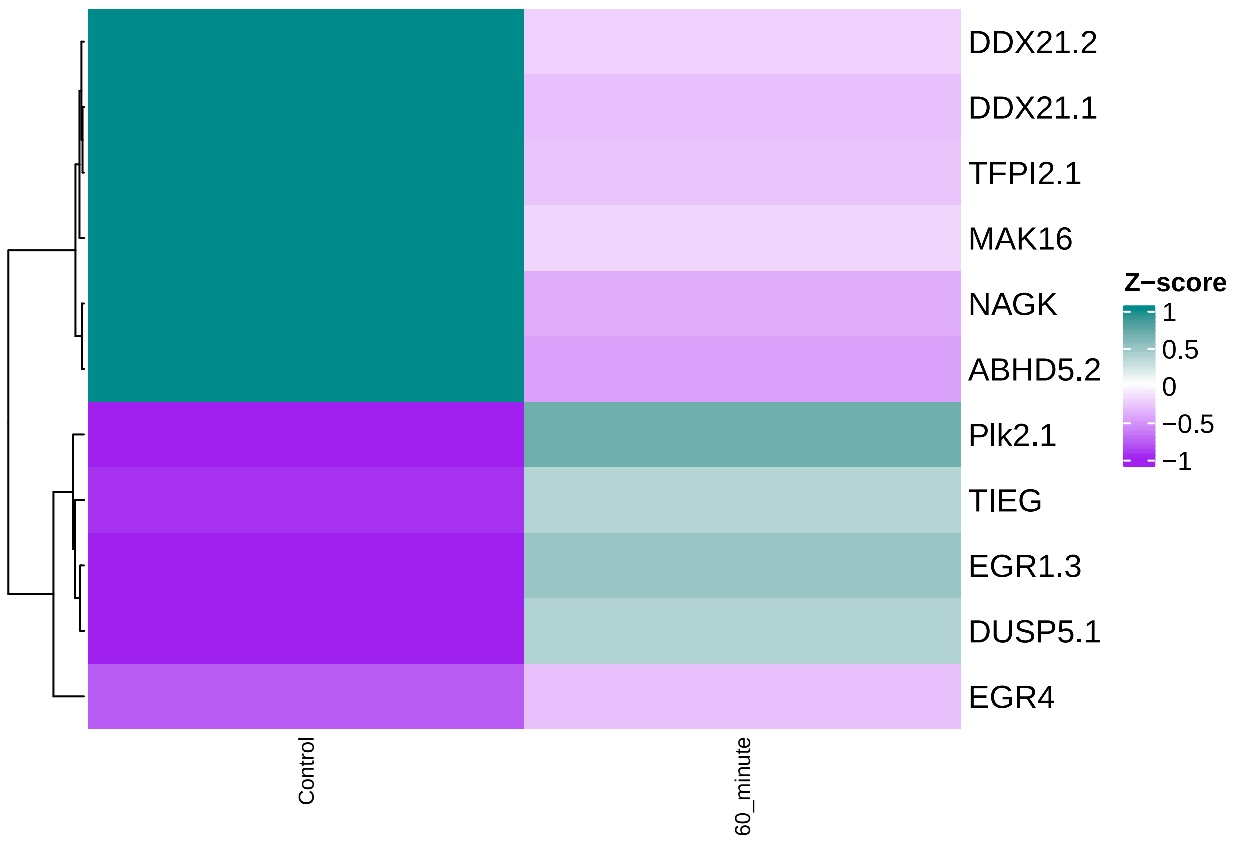
```
Heatmap(ieg15UPDOWN_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



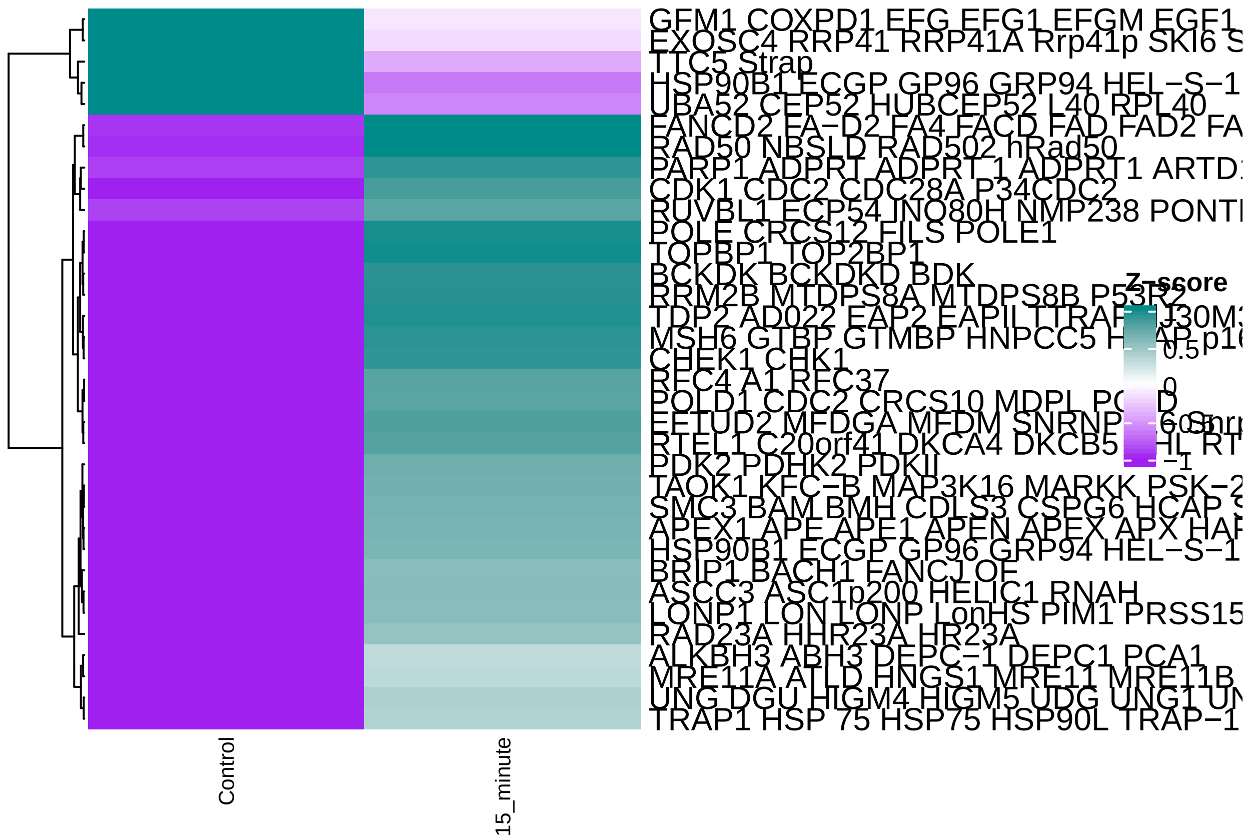
```
Heatmap(ieg60_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



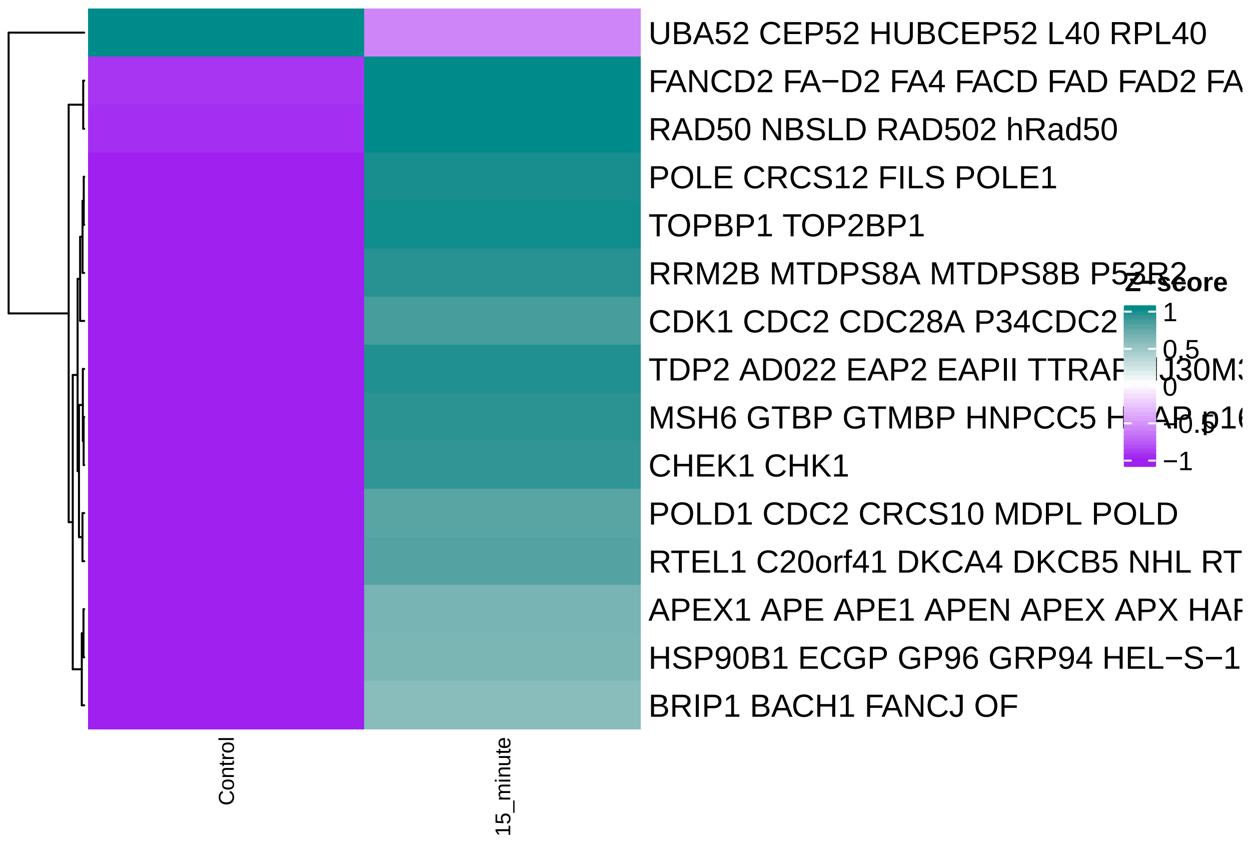
```
Heatmap(ieg60UPDOWN_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



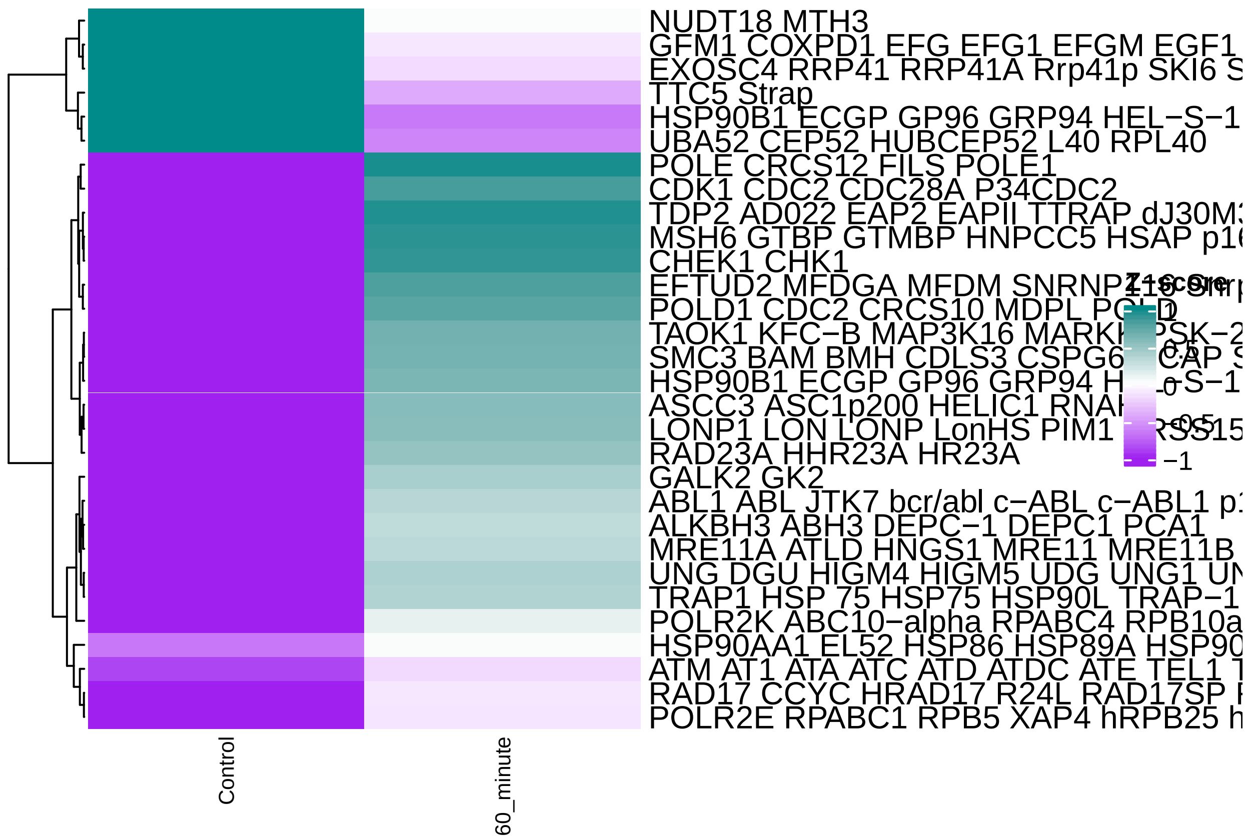
```
Heatmap(MMR15_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



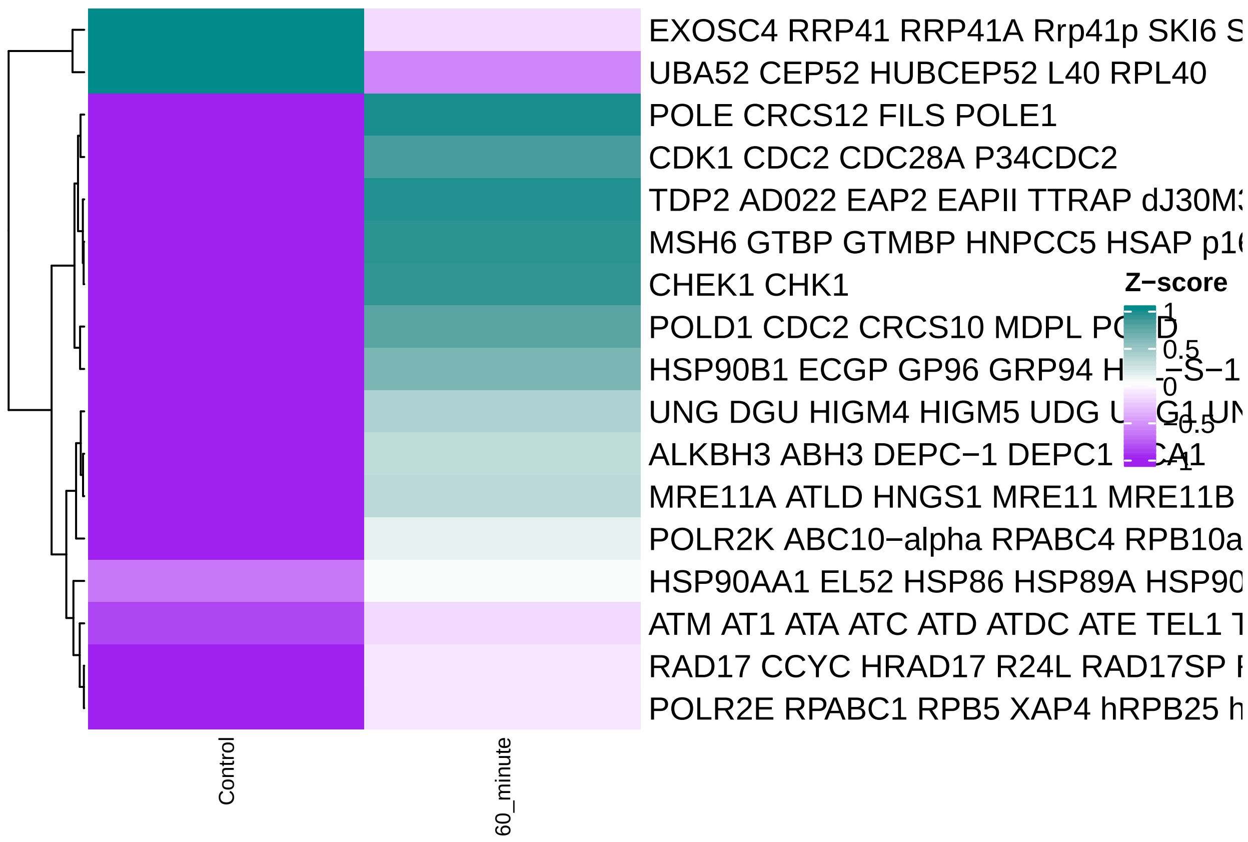
```
Heatmap(MMR15UPDOWN_hm, name = "Z-score", col = myco1z, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



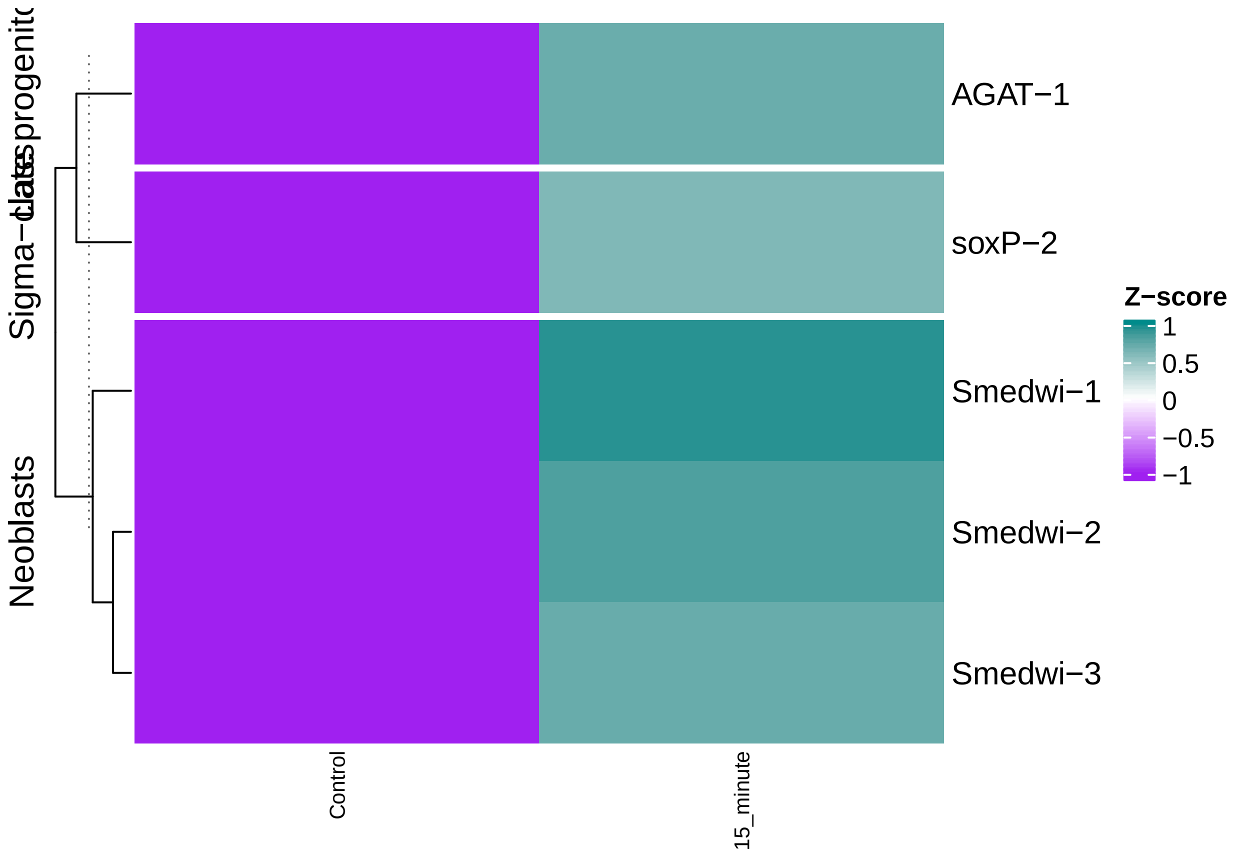
```
Heatmap(MMR60_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



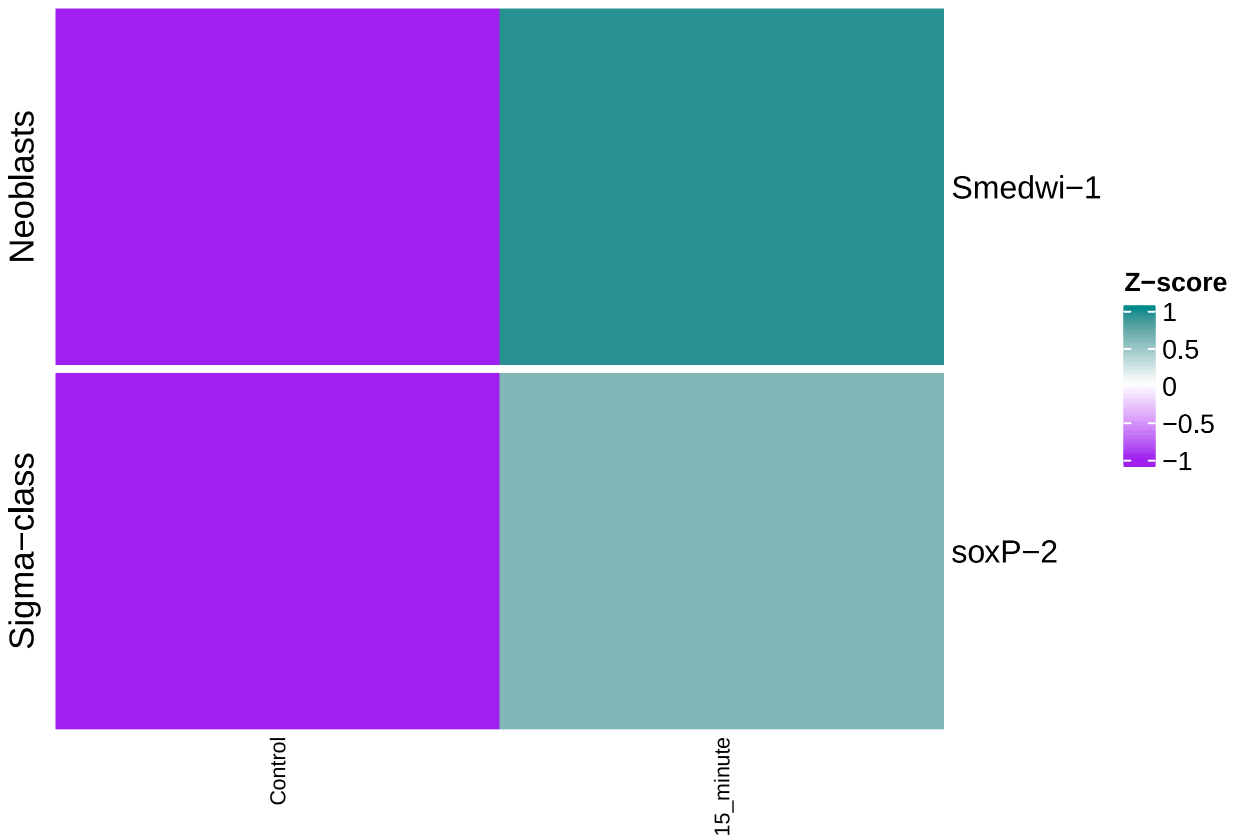
```
Heatmap(MMR60UPDOWN_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



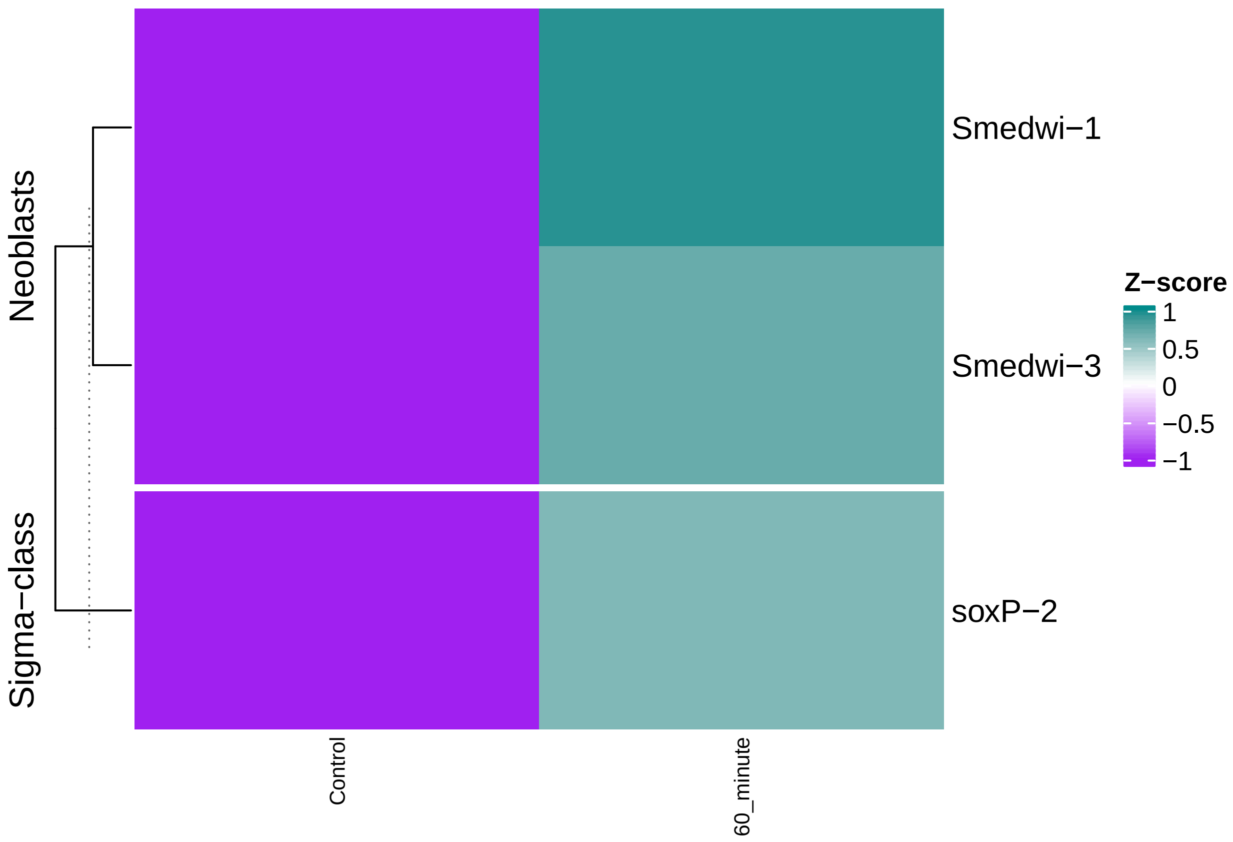
```
Heatmap(NBFIG15_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE, split = NBFIG15$Cell.type)
```



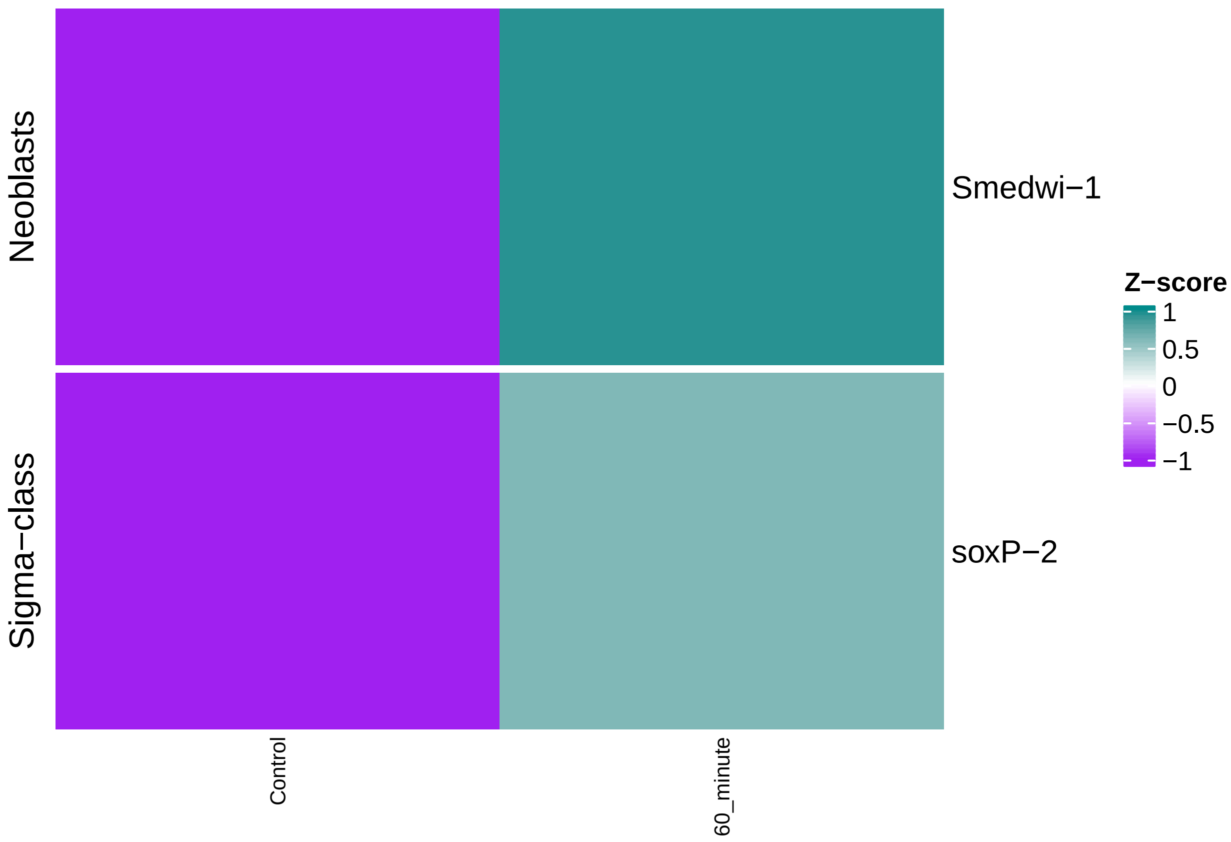
```
Heatmap(NBFIG15UPDOWN_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE, split = NBFIG15UPDOWN$Cell.type)
```



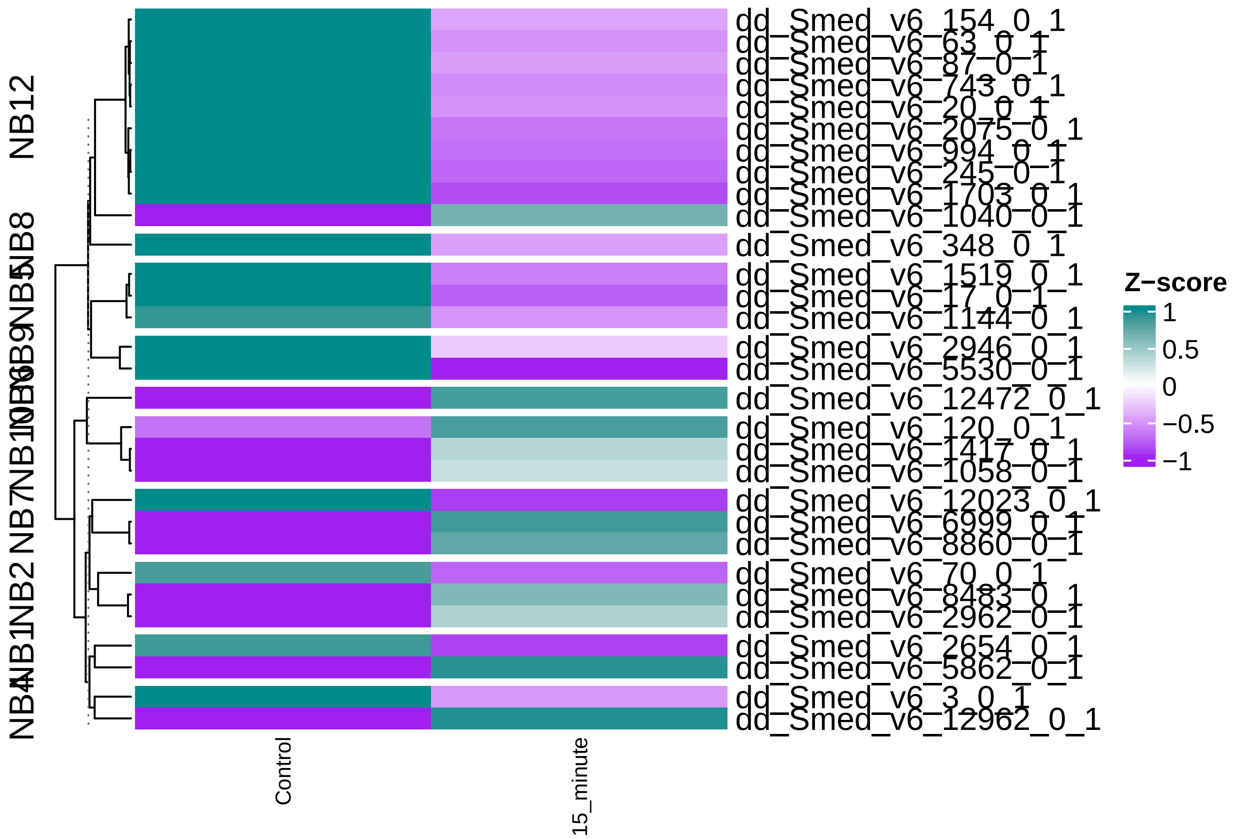
```
Heatmap(NBFIG60_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE, split = NBFIG60$Cell.type)
```



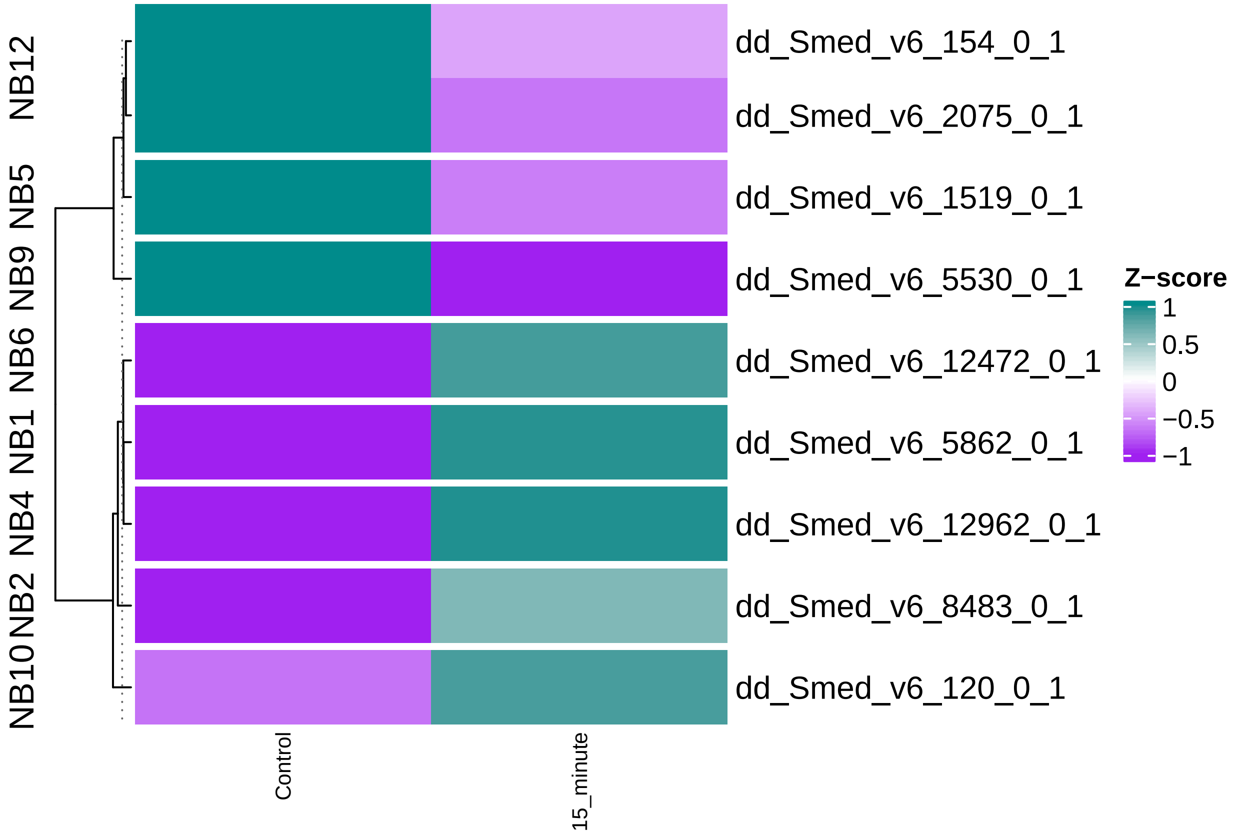
```
Heatmap(NBFIG60UPDOWN_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE, split = NBFIG60UPDOWN$Cell.type)
```



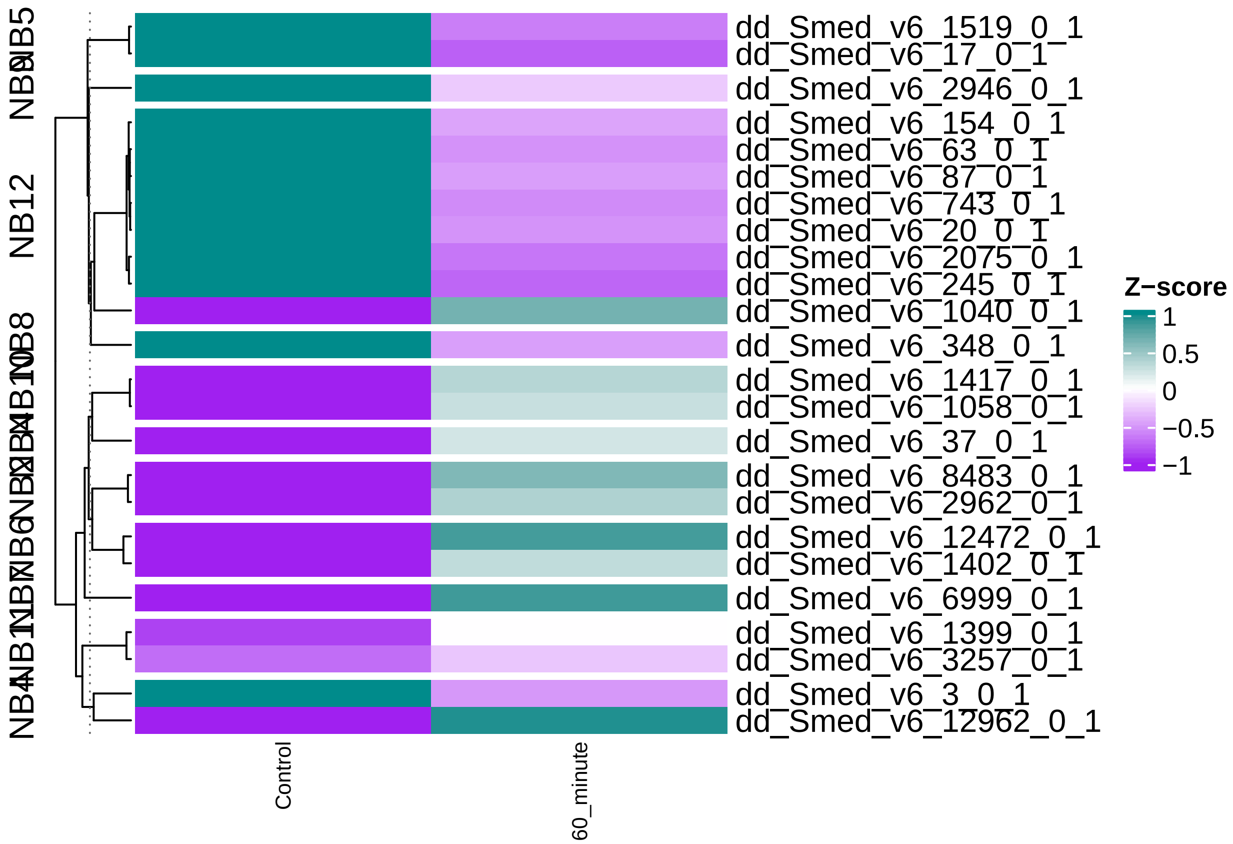
```
Heatmap(nbsc15_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE, split = nbsc15$NB.subclass)
```



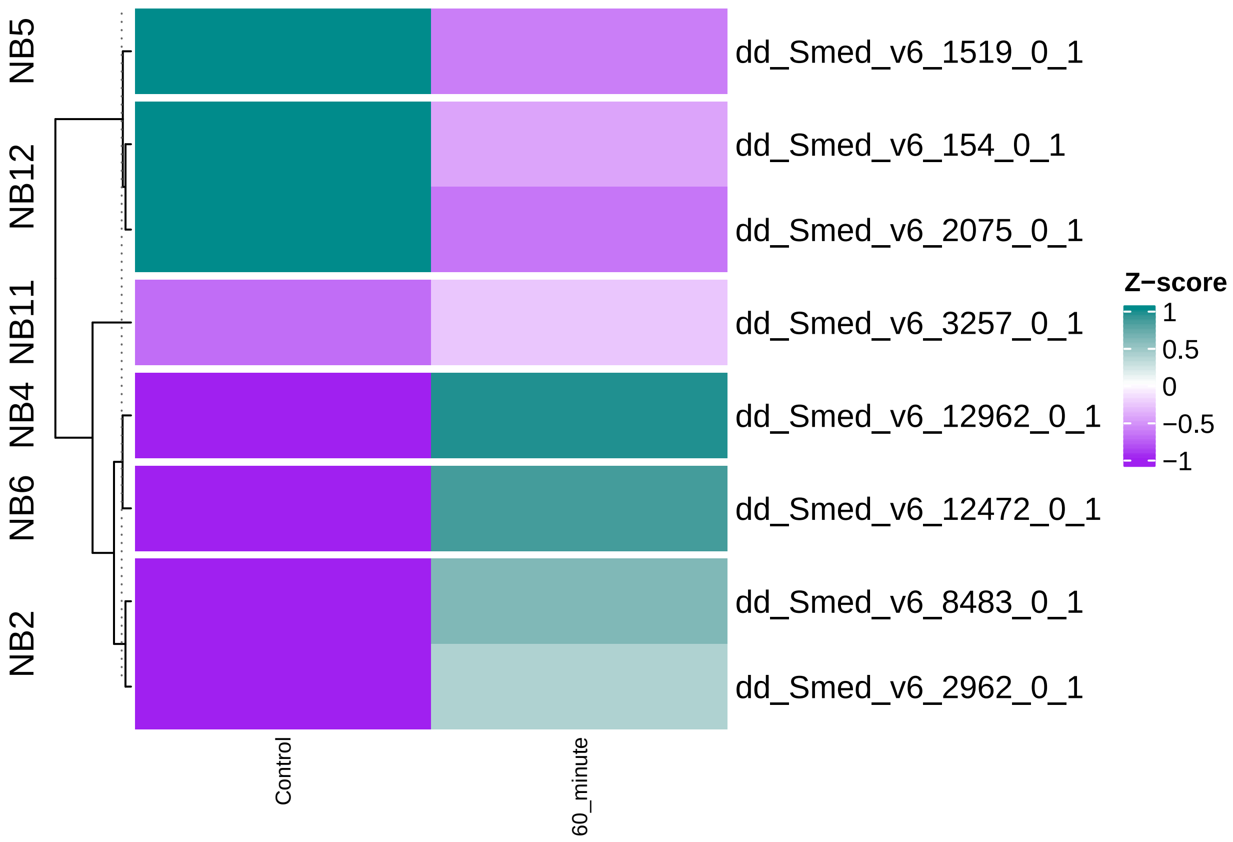
```
Heatmap(nbsc15UPDOWN_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE, split = nbsc15UPDOWN$NB.subclass)
```



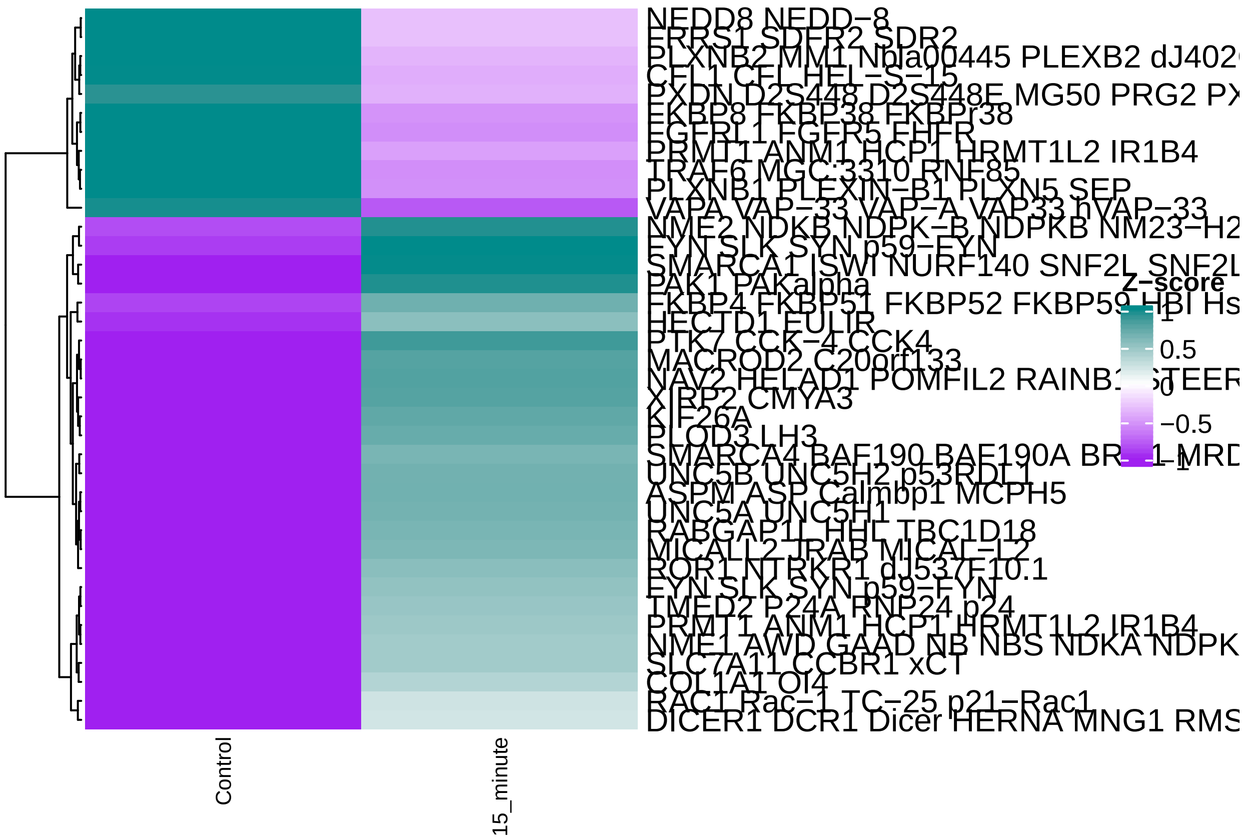
```
Heatmap(nbsc60_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE, split = nbsc60$NB.subclass)
```



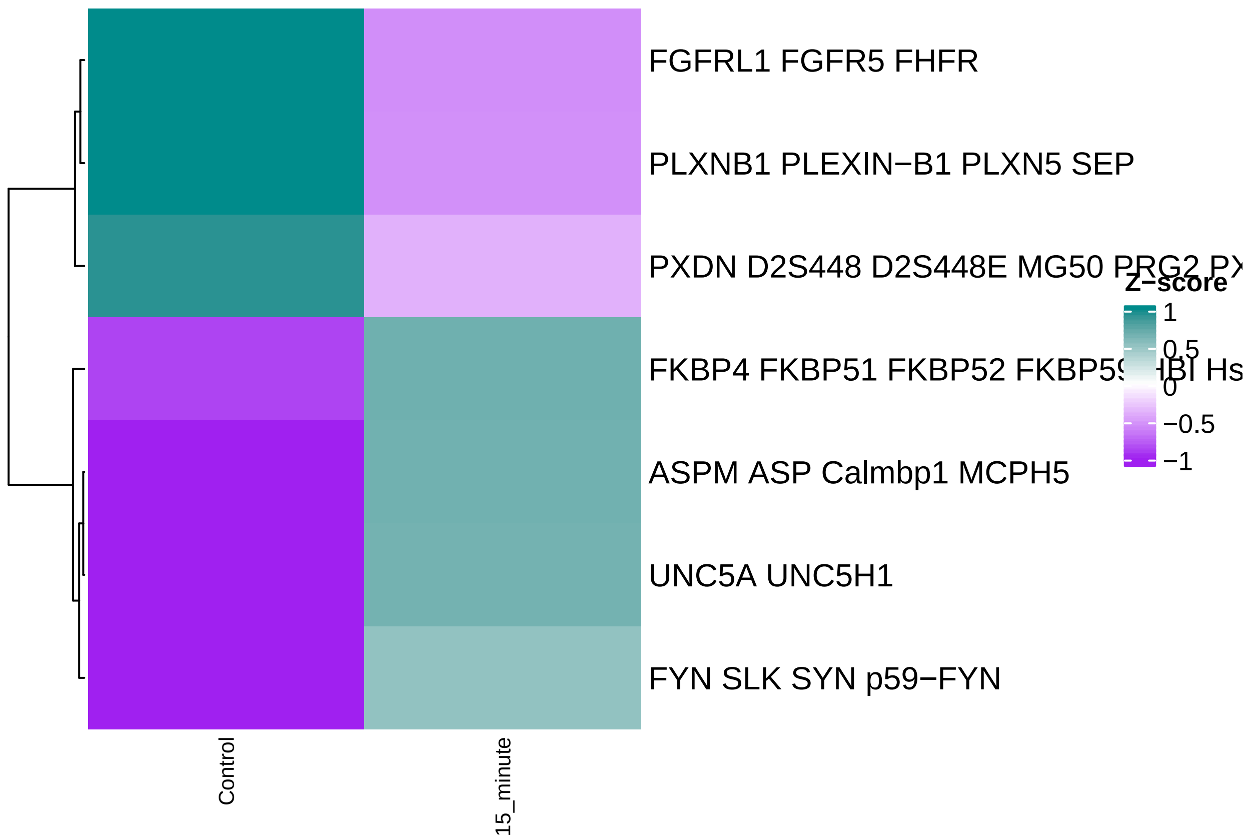
```
Heatmap(nbsc60UPDOWN_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE, split = nbsc60UPDOWN$NB.subclass)
```



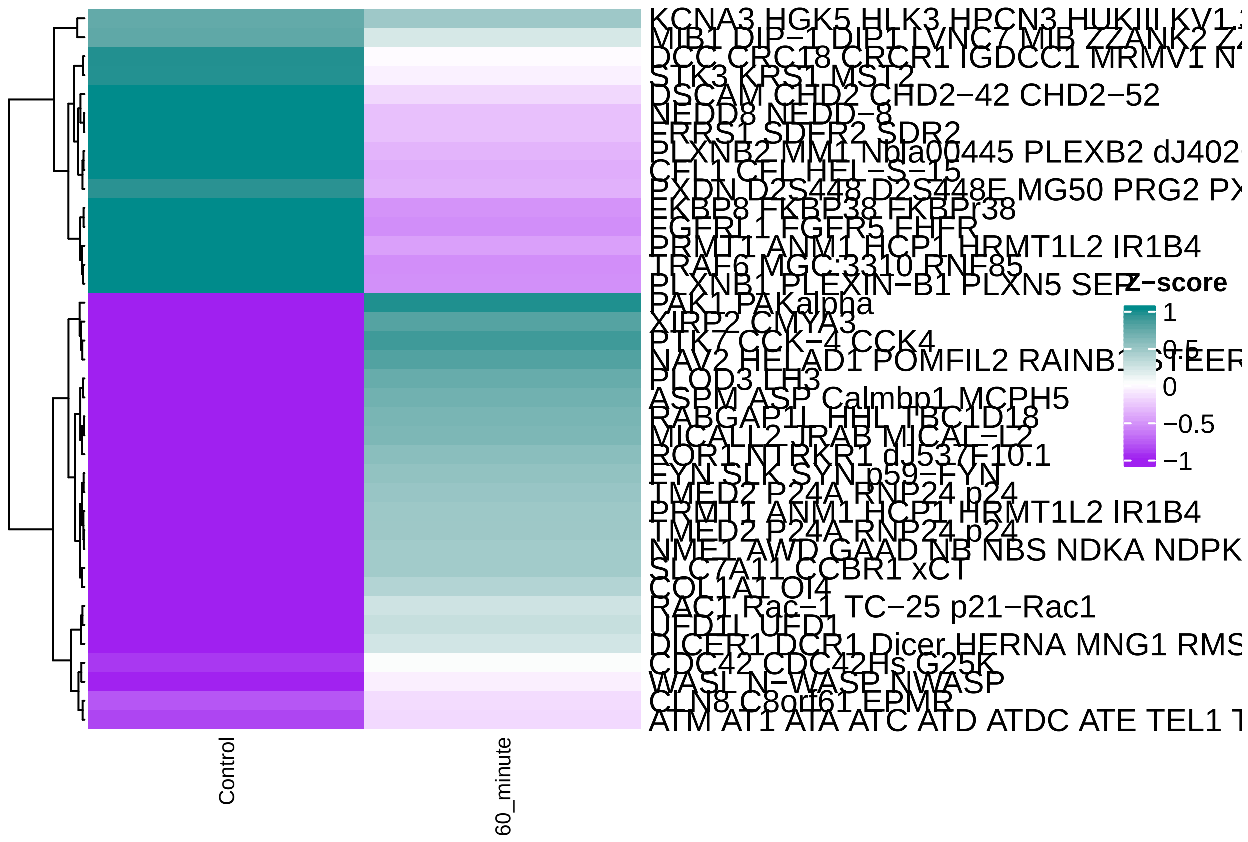
```
Heatmap(neural15_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



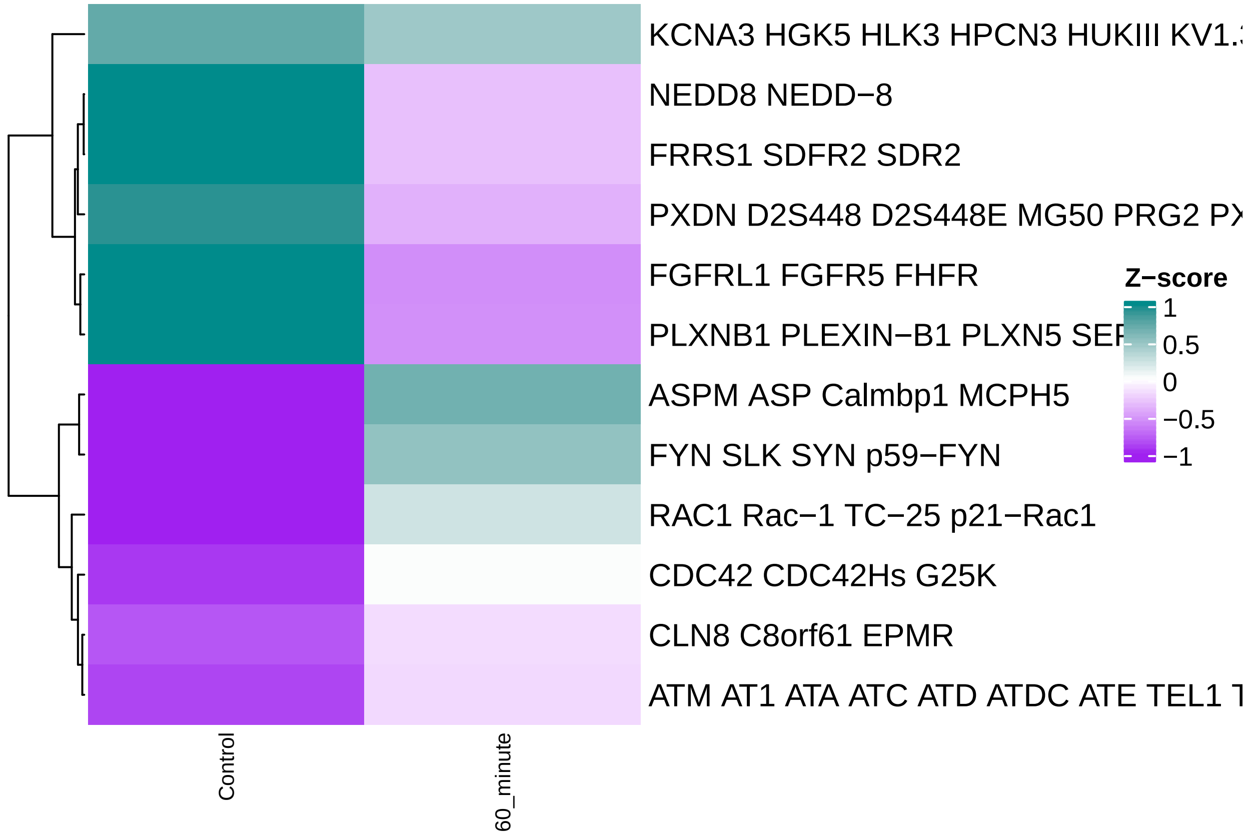
```r
Heatmap(neural15UPDOWN_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



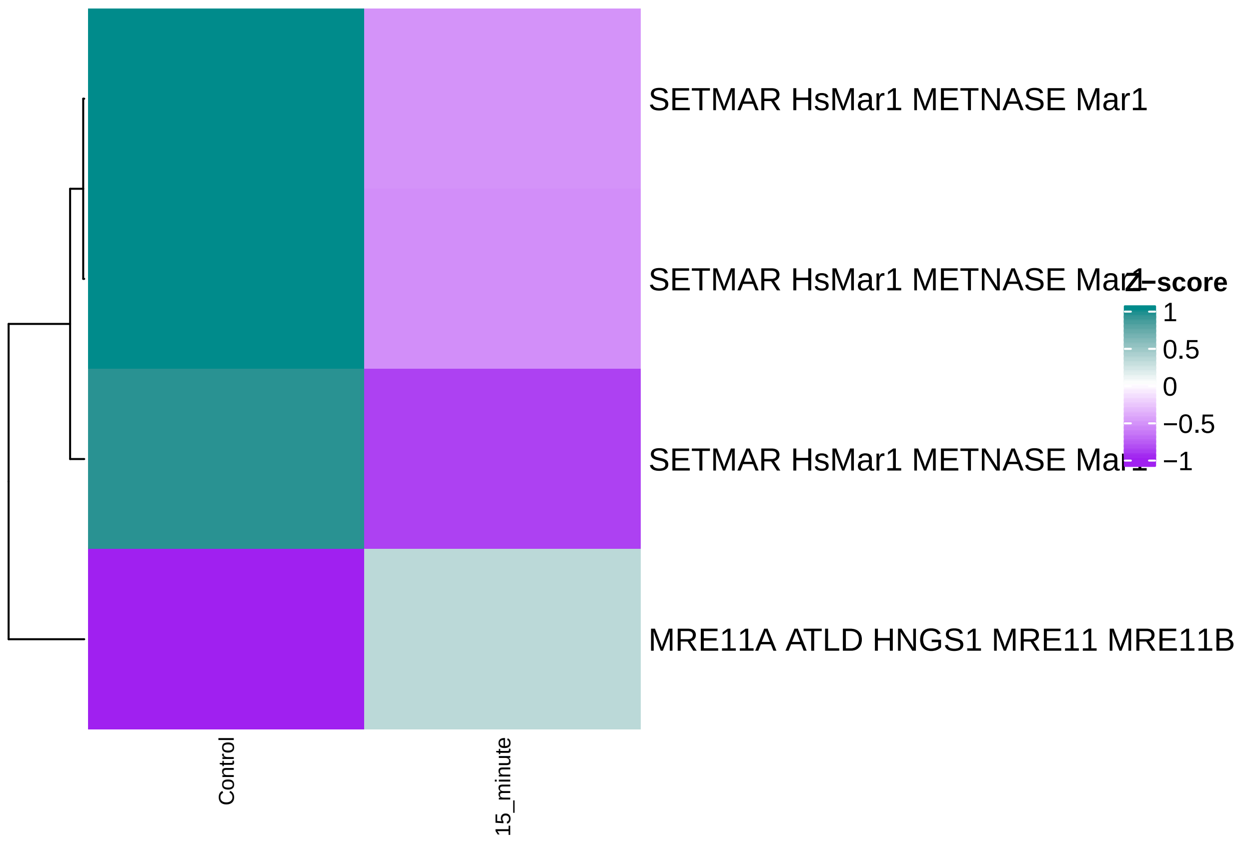
```
Heatmap(neural60_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



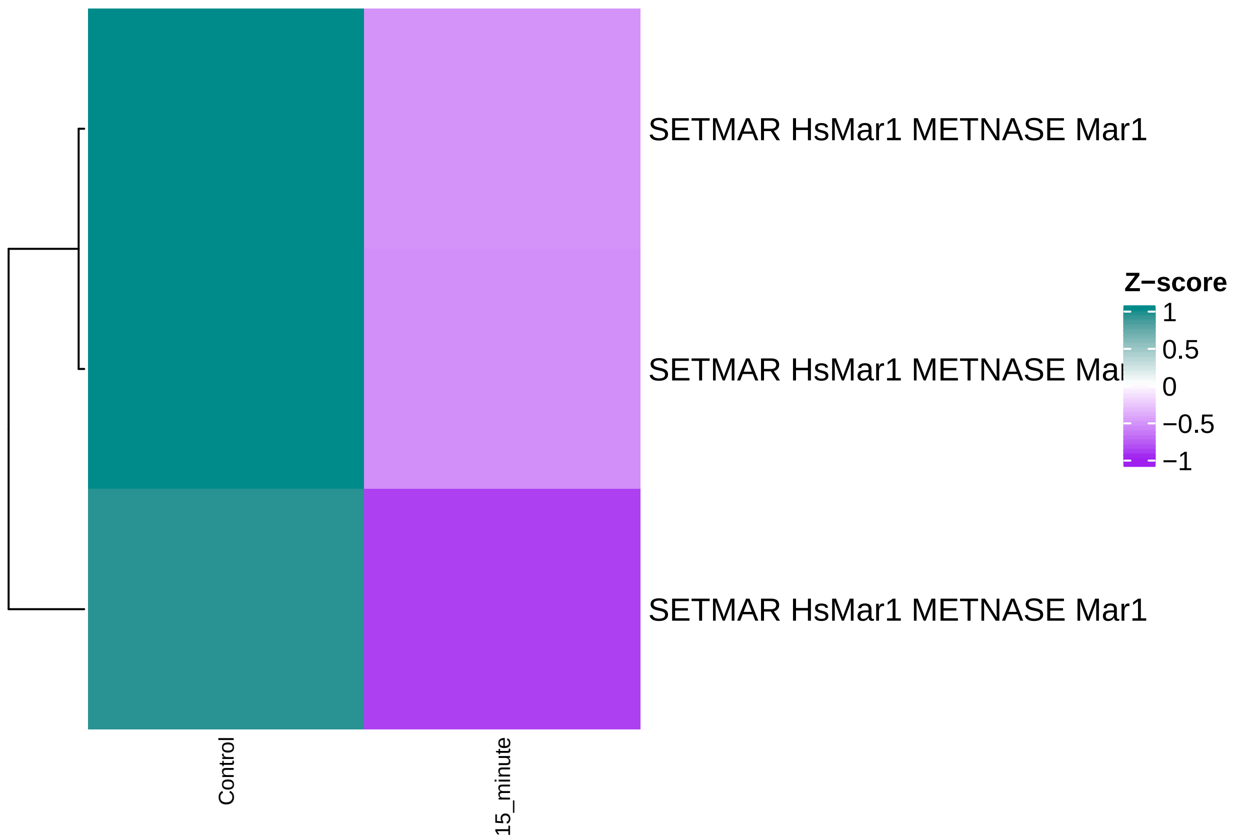
```
Heatmap(neural60UPDOWN_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



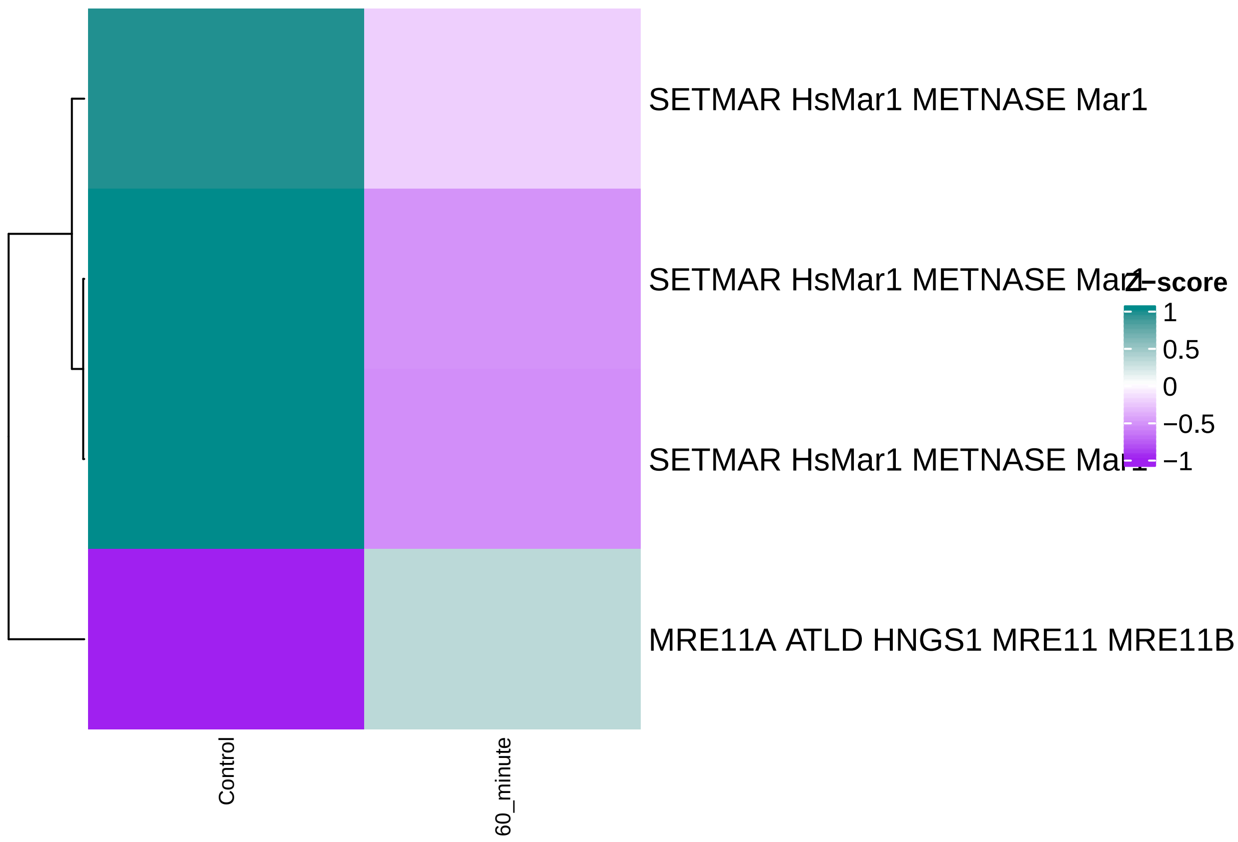
```
Heatmap(NHEJ15_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



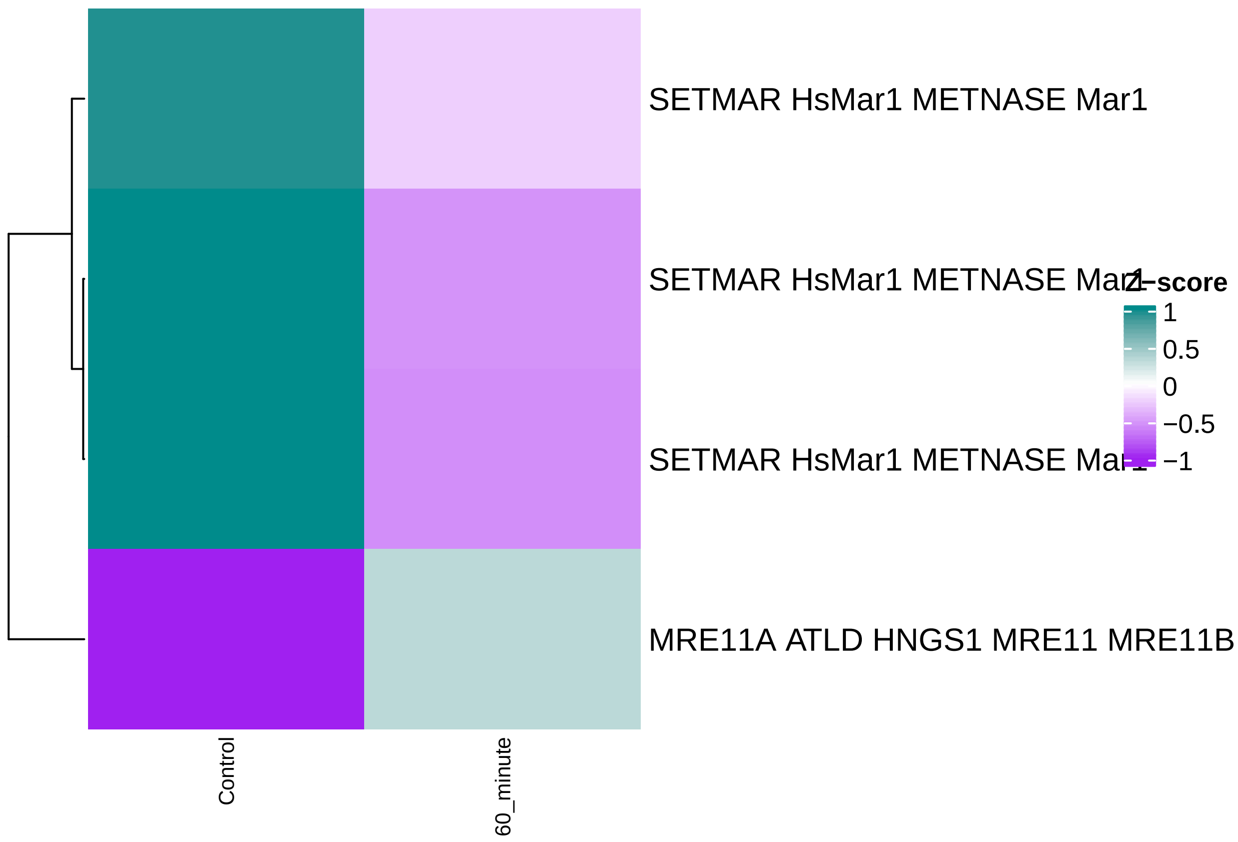
```
Heatmap(NHEJ15UPDOWN_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



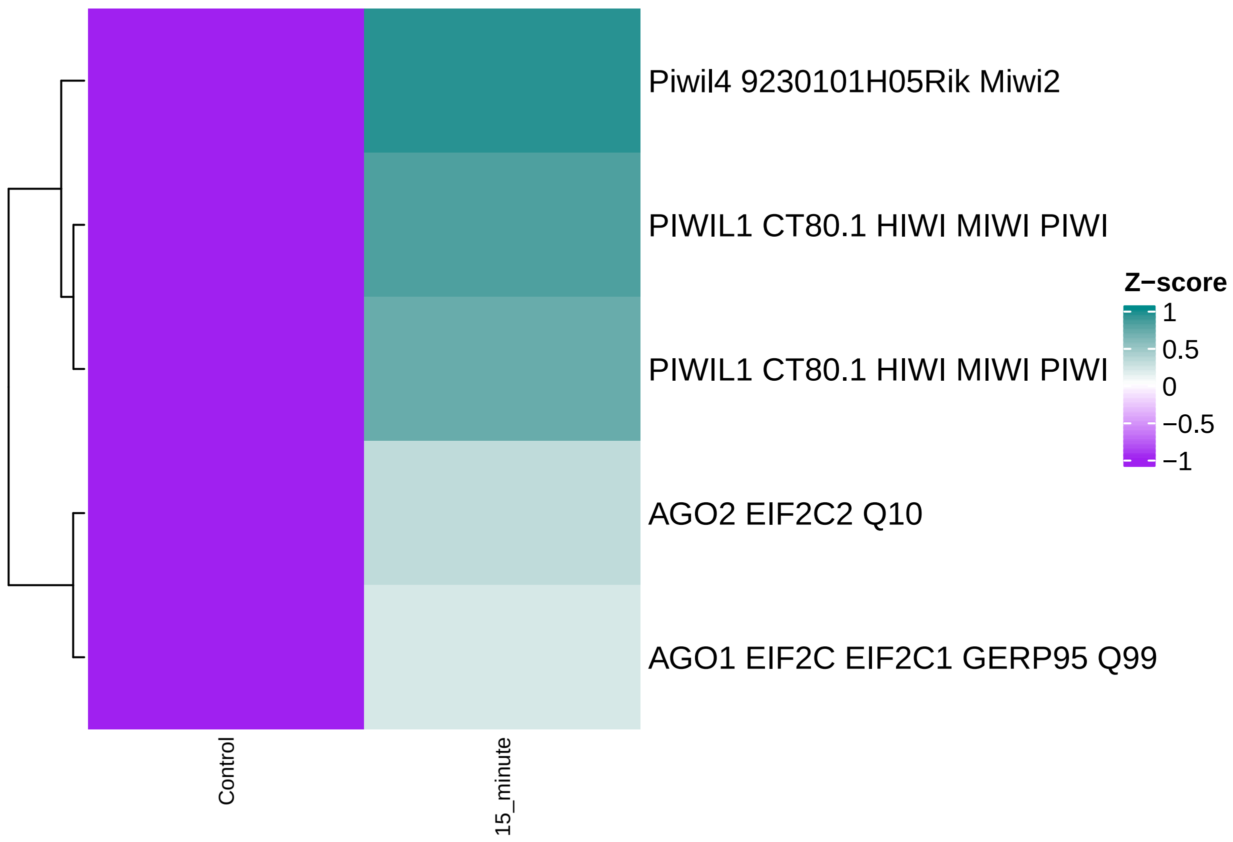
```
Heatmap(NHEJ60_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



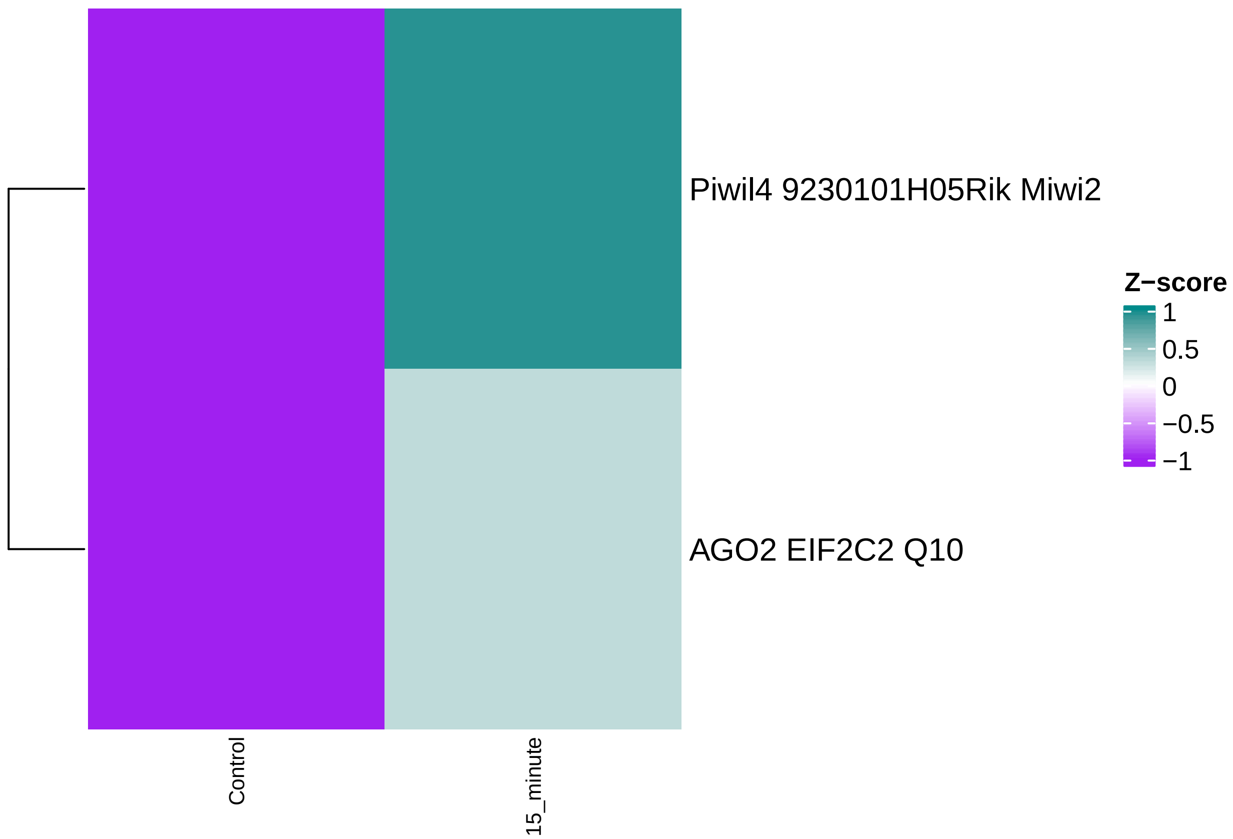
```
Heatmap(NHEJ60UPDOWN_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



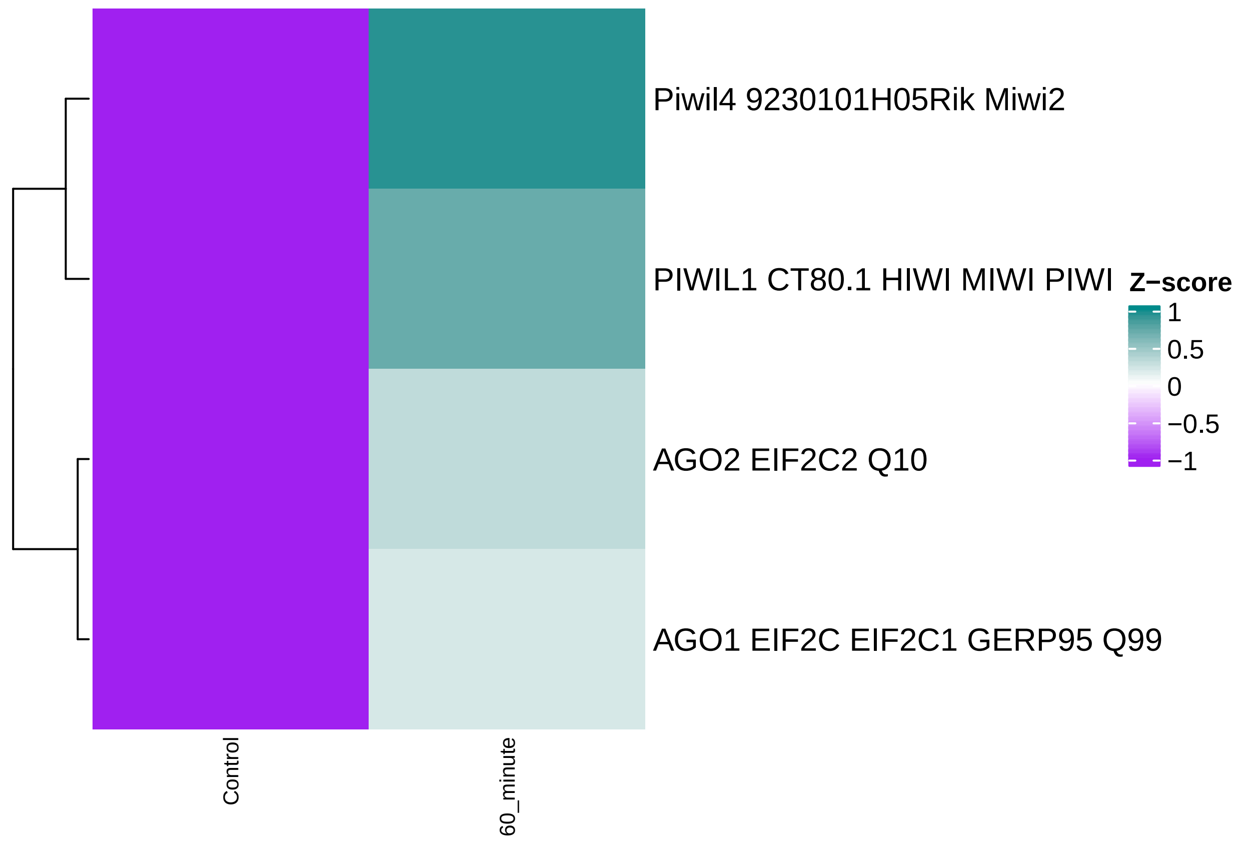
```
Heatmap(piwi15_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



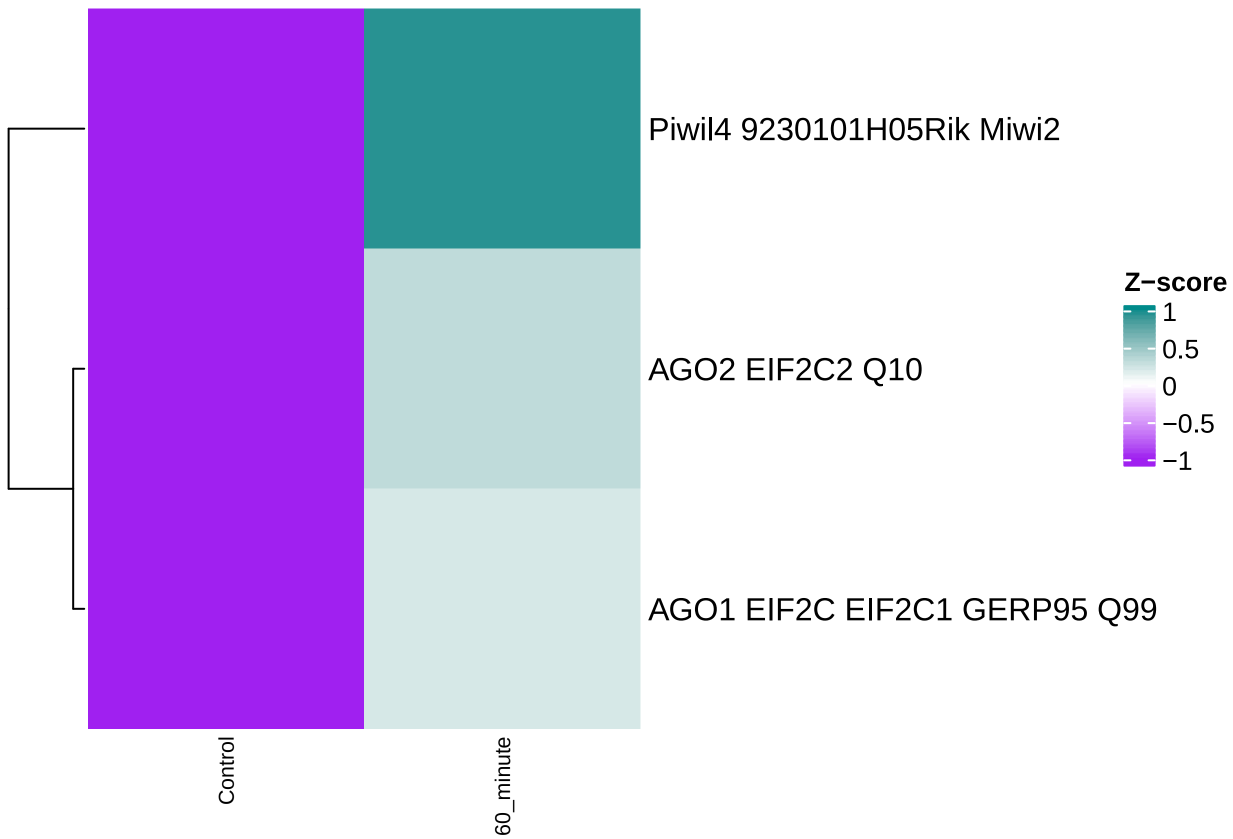
```
Heatmap(piwi15UPDOWN_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



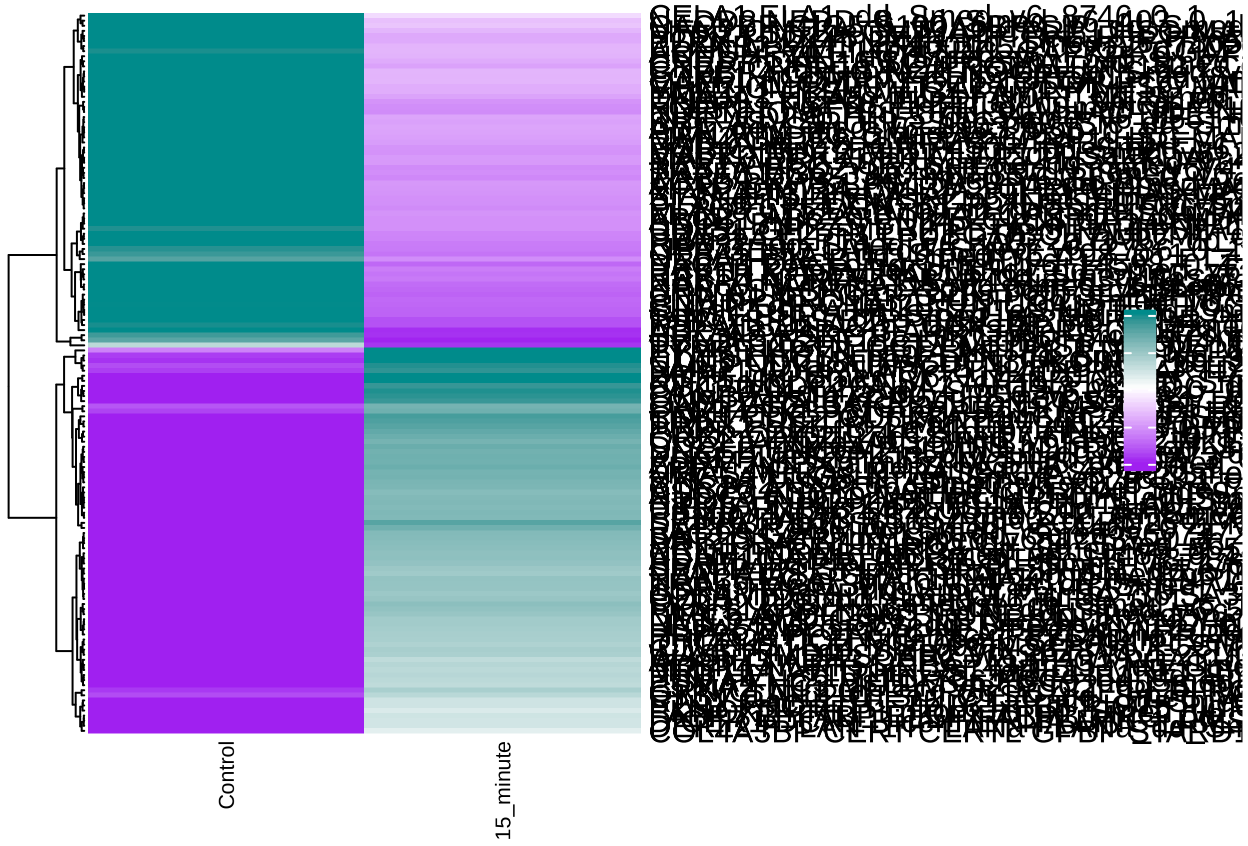
```
Heatmap(piwi60_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



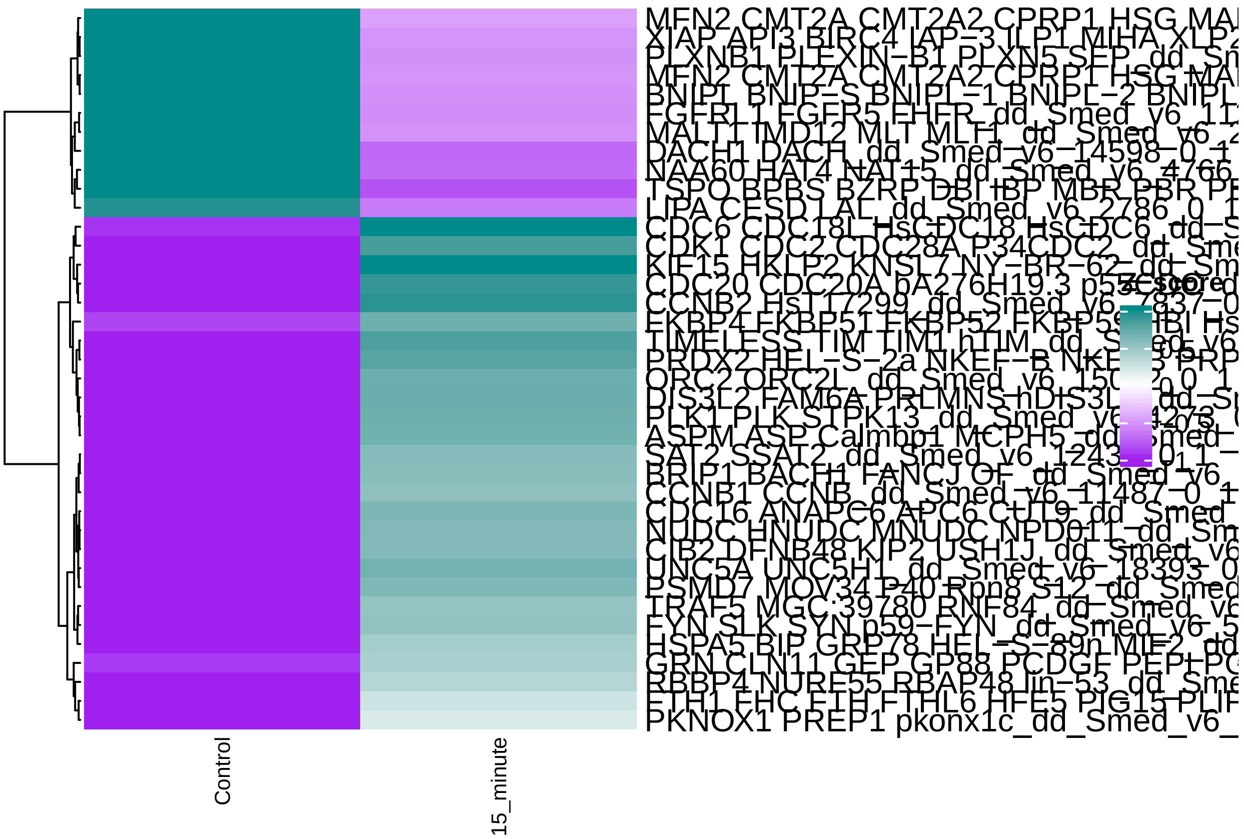
```
Heatmap(piwi60UPDOWN_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



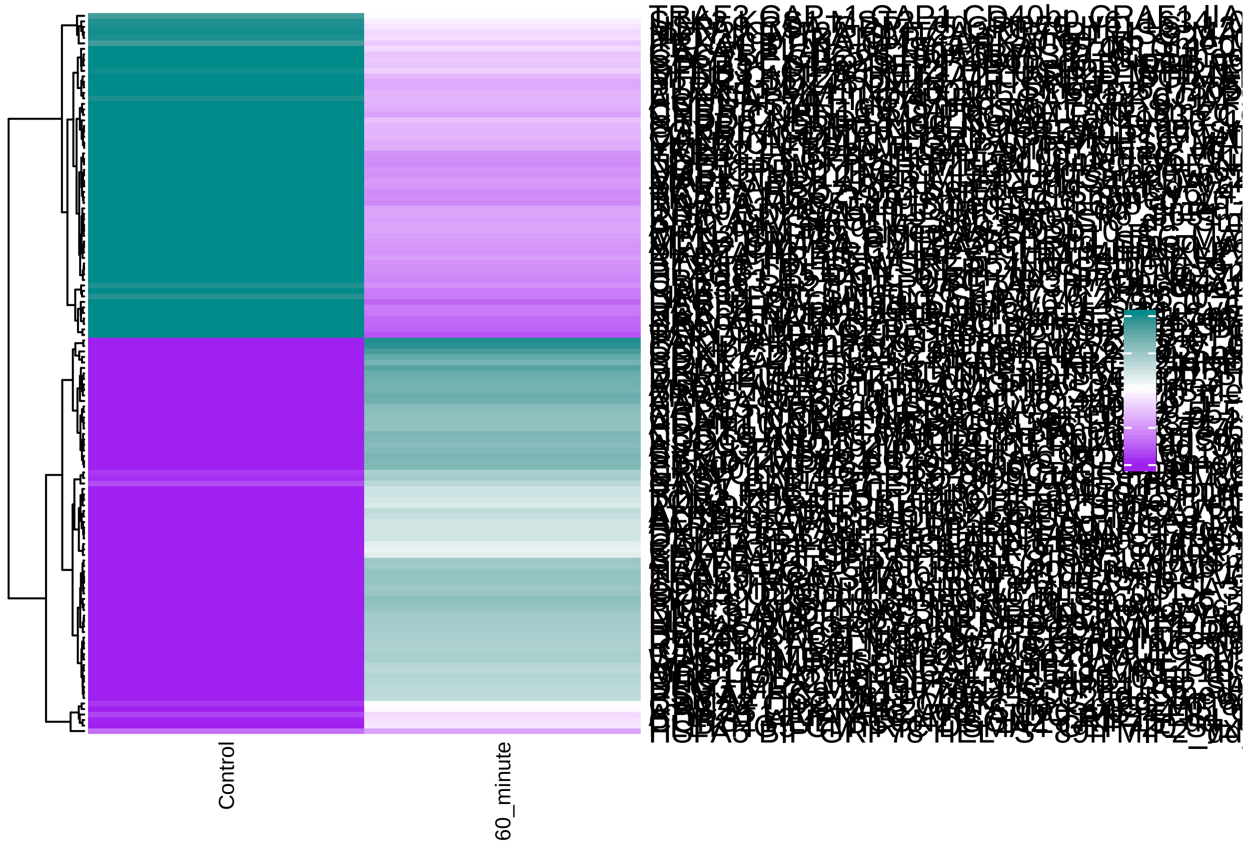
```
Heatmap(proliferation15_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



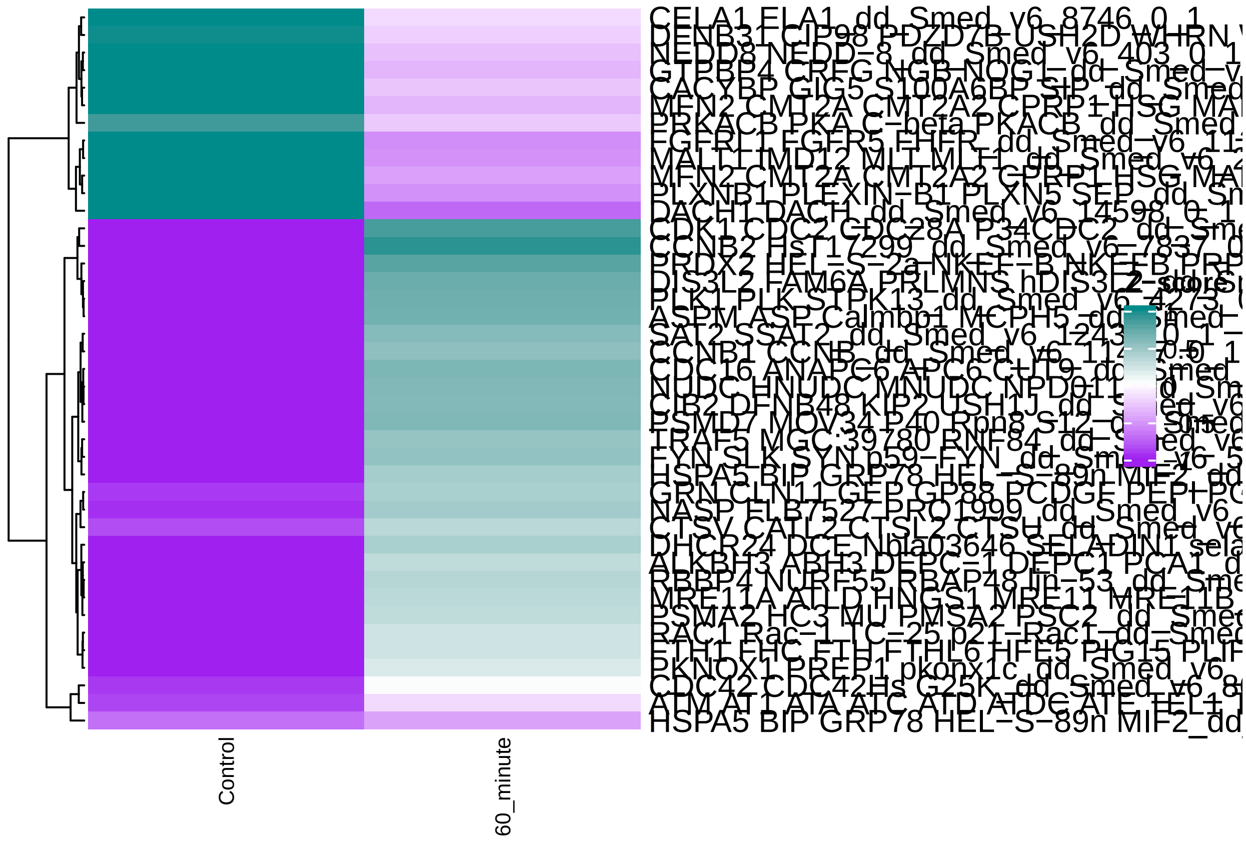
```
Heatmap(proliferation15UPDOWN_hm, name = "Z-score", col = mycolz,
    column_names_gp = gpar(fontsize = 8), cluster_columns = FALSE,
    cluster_rows = TRUE)
```



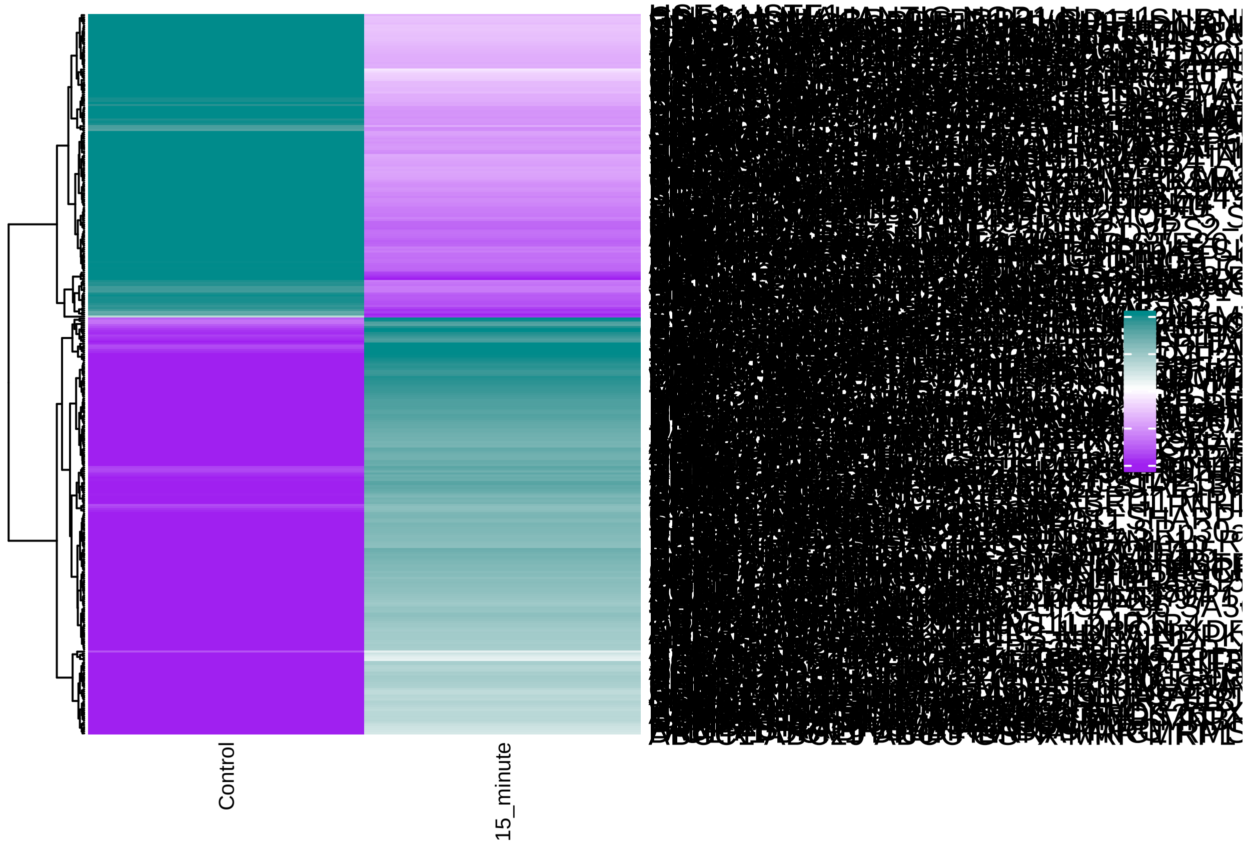
```r
Heatmap(proliferation60_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



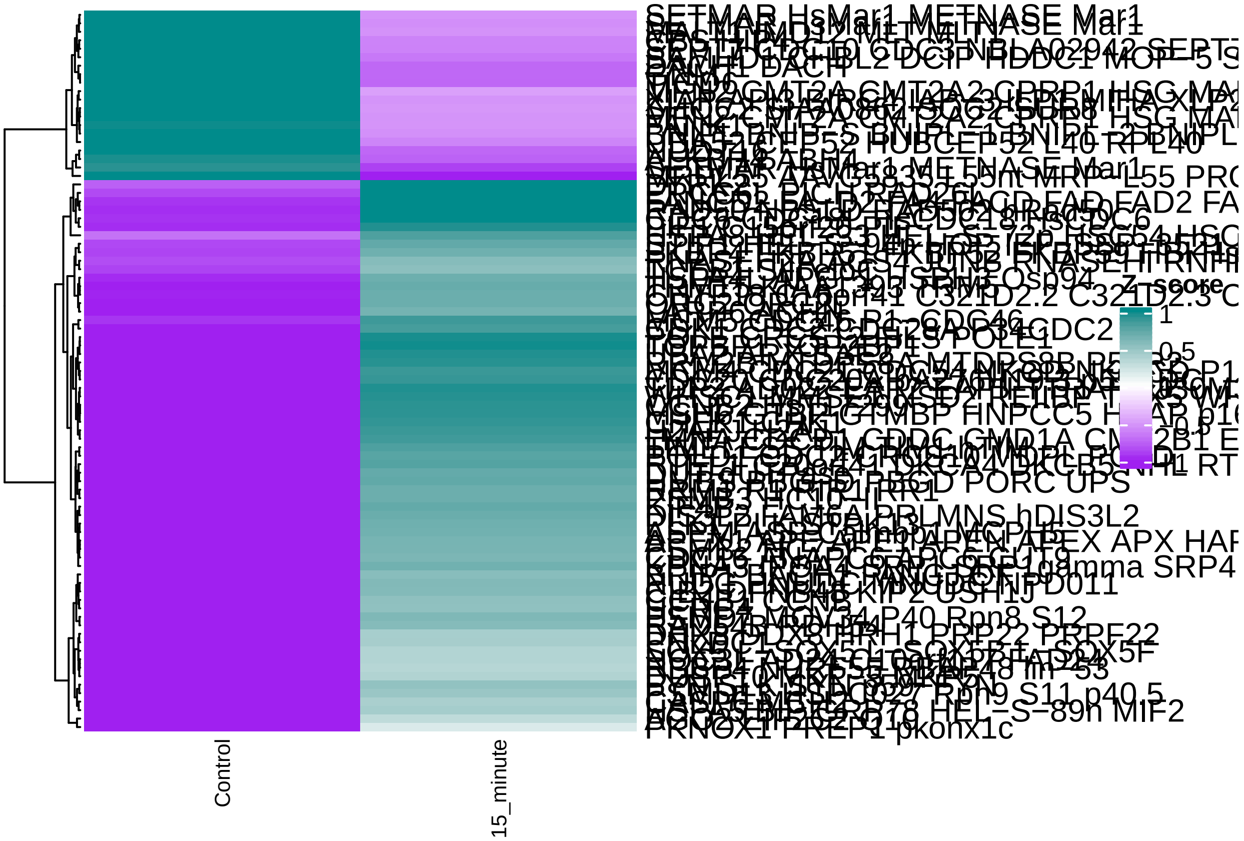
```
Heatmap(proliferation60UPDOWN_hm, name = "Z-score", col = mycolz,
    column_names_gp = gpar(fontsize = 8), cluster_columns = FALSE,
    cluster_rows = TRUE)
```



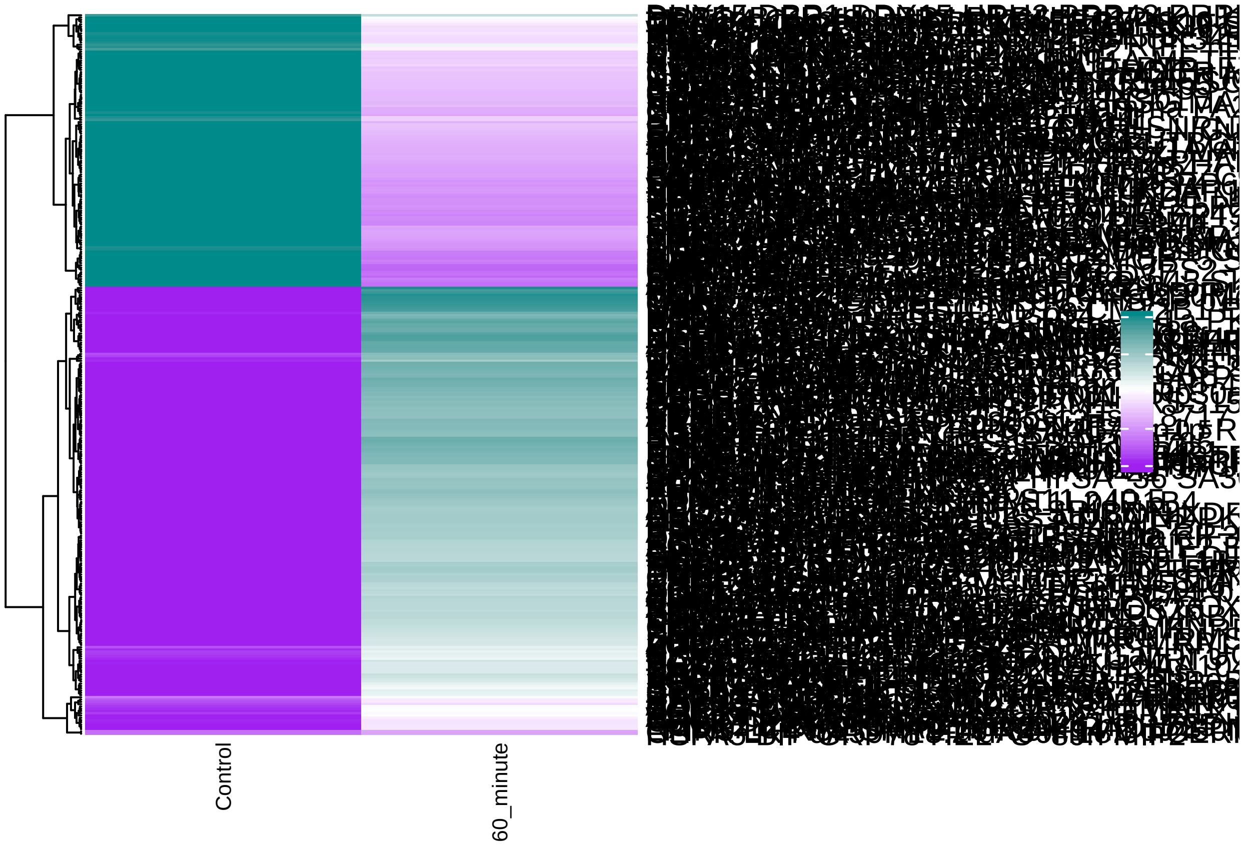
```r
Heatmap(replication15_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



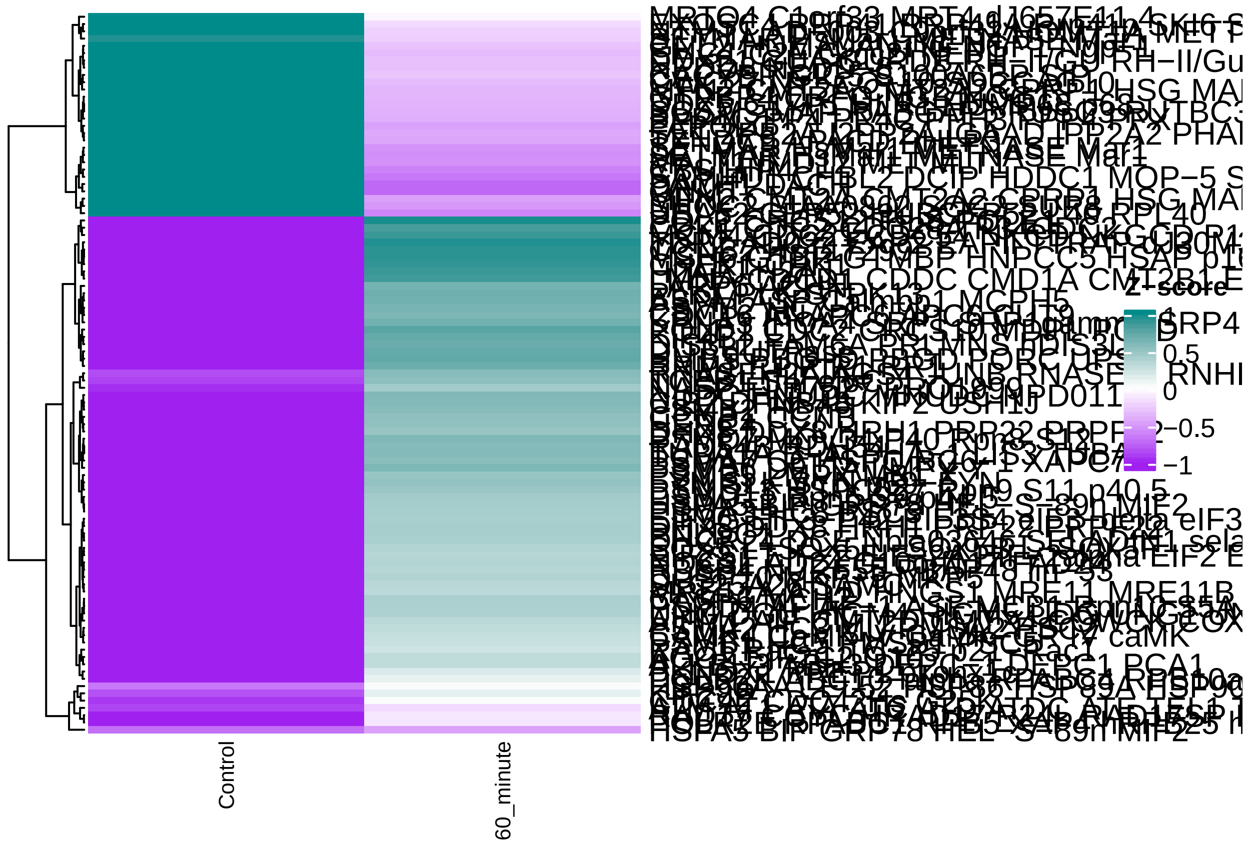
```r
Heatmap(replication15UPDOWN_hm, name = "Z-score", col = mycolz,
    column_names_gp = gpar(fontsize = 8), cluster_columns = FALSE,
    cluster_rows = TRUE)
```



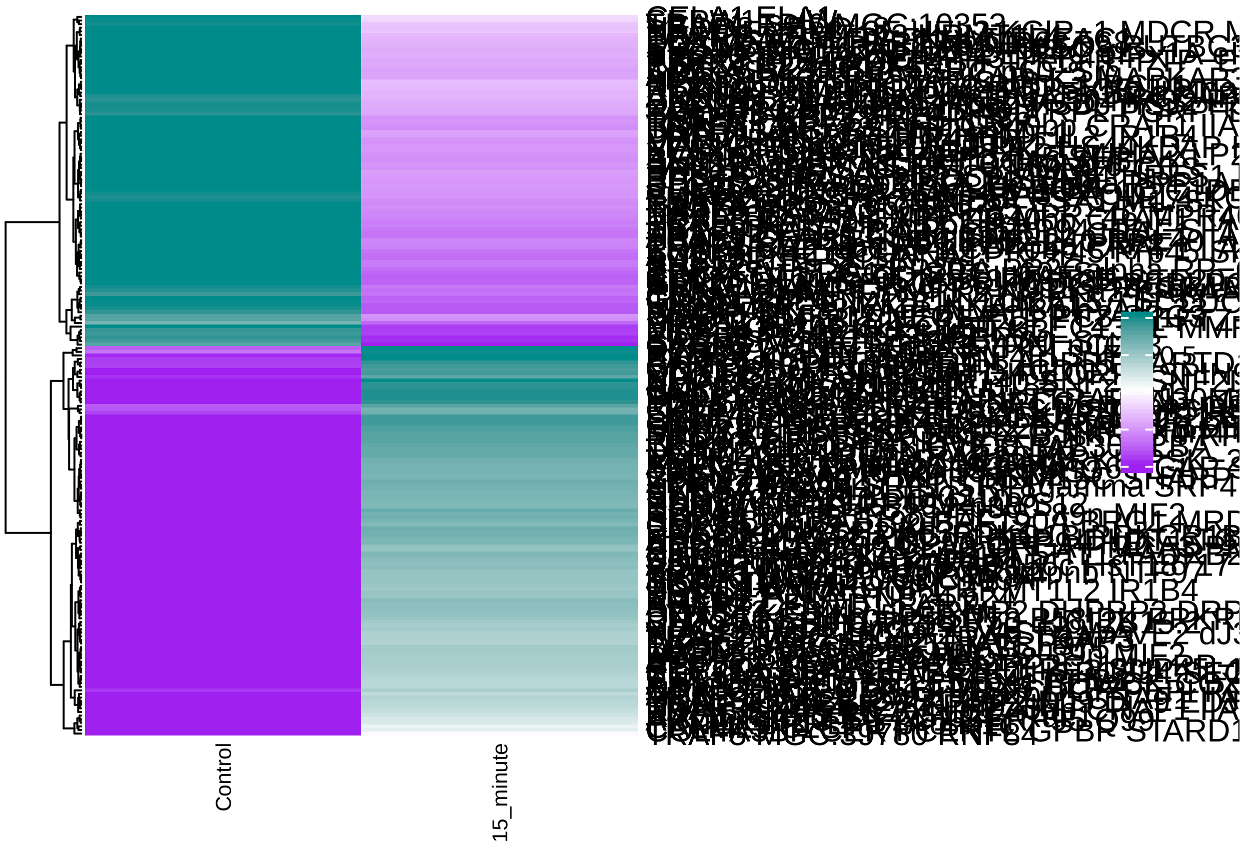
```r
Heatmap(replication60_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



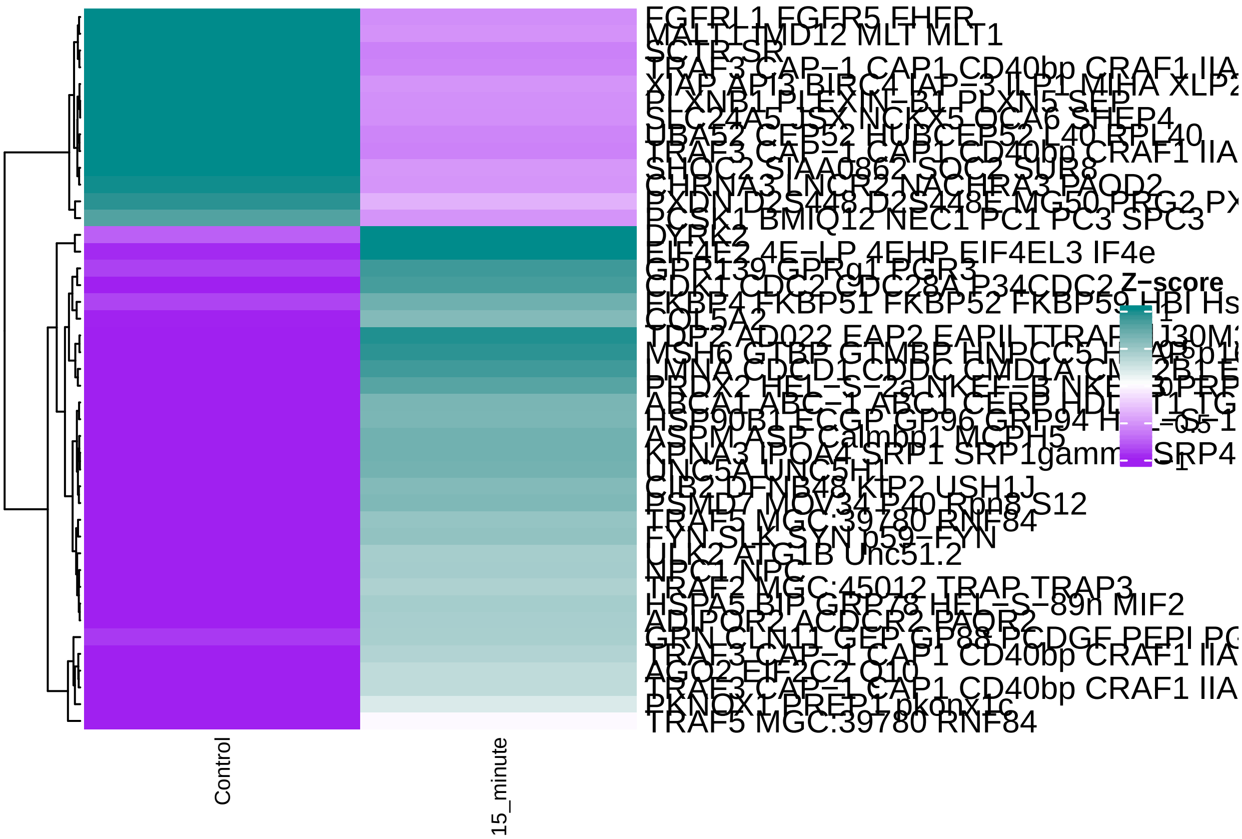
```
Heatmap(replication60UPDOWN_hm, name = "Z-score", col = mycolz,
    column_names_gp = gpar(fontsize = 8), cluster_columns = FALSE,
    cluster_rows = TRUE)
```



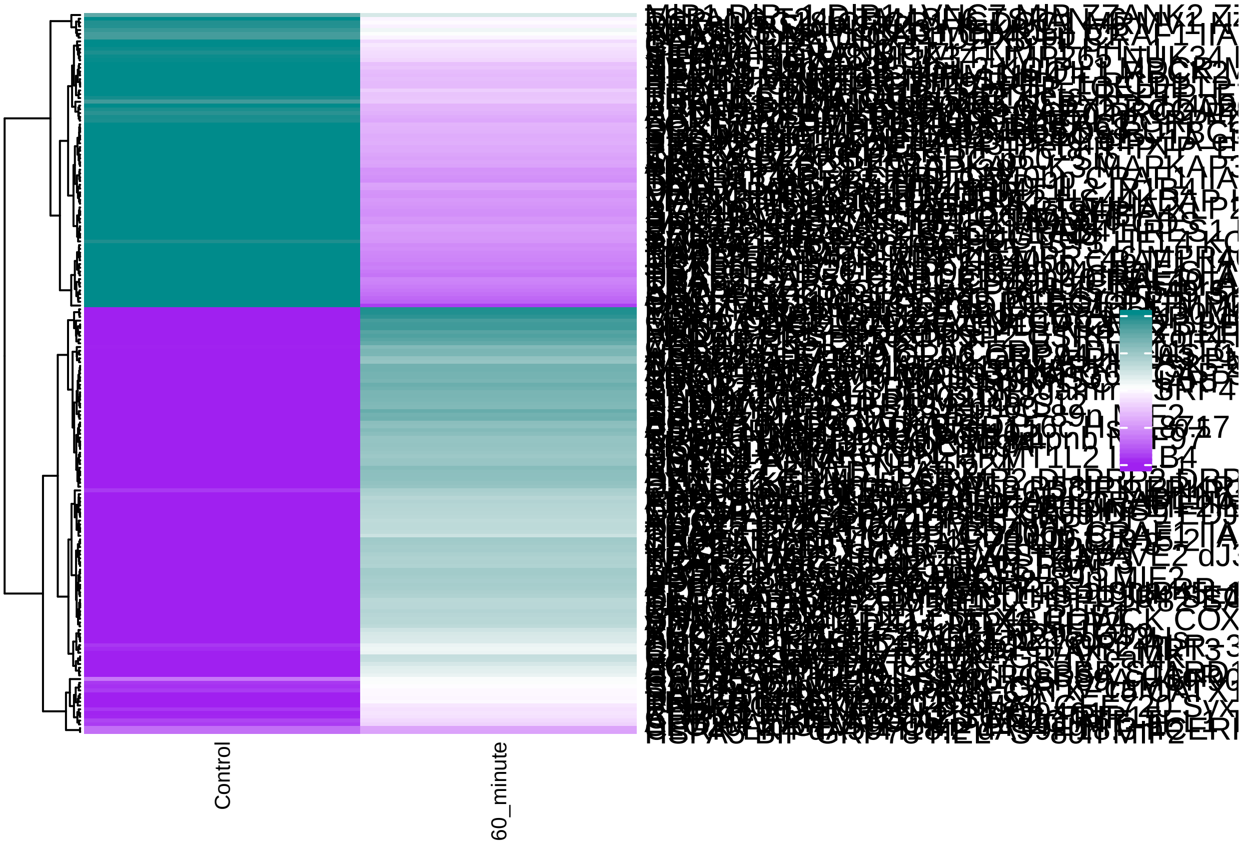
```r
Heatmap(signaling15_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



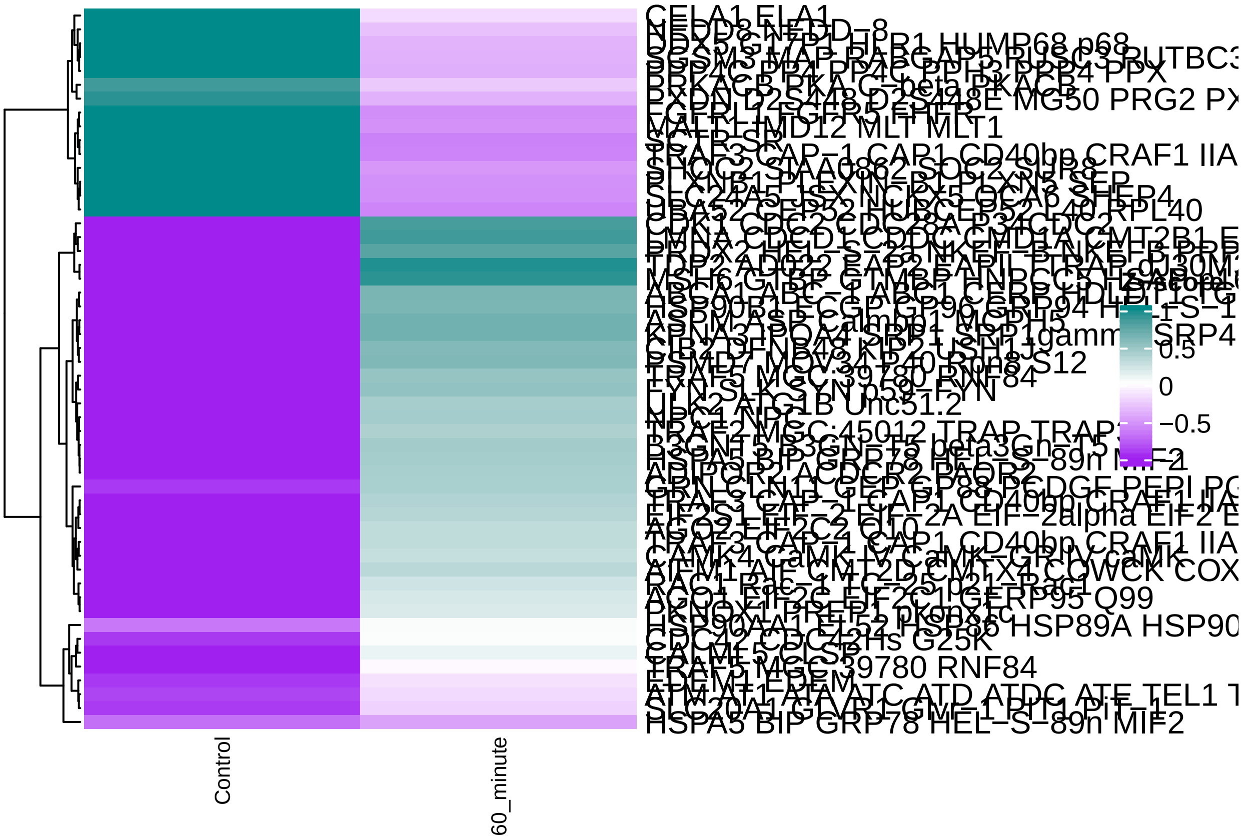
```r
Heatmap(signaling15UPDOWN_hm, name = "Z-score", col = mycolz,
    column_names_gp = gpar(fontsize = 8), cluster_columns = FALSE,
    cluster_rows = TRUE)
```



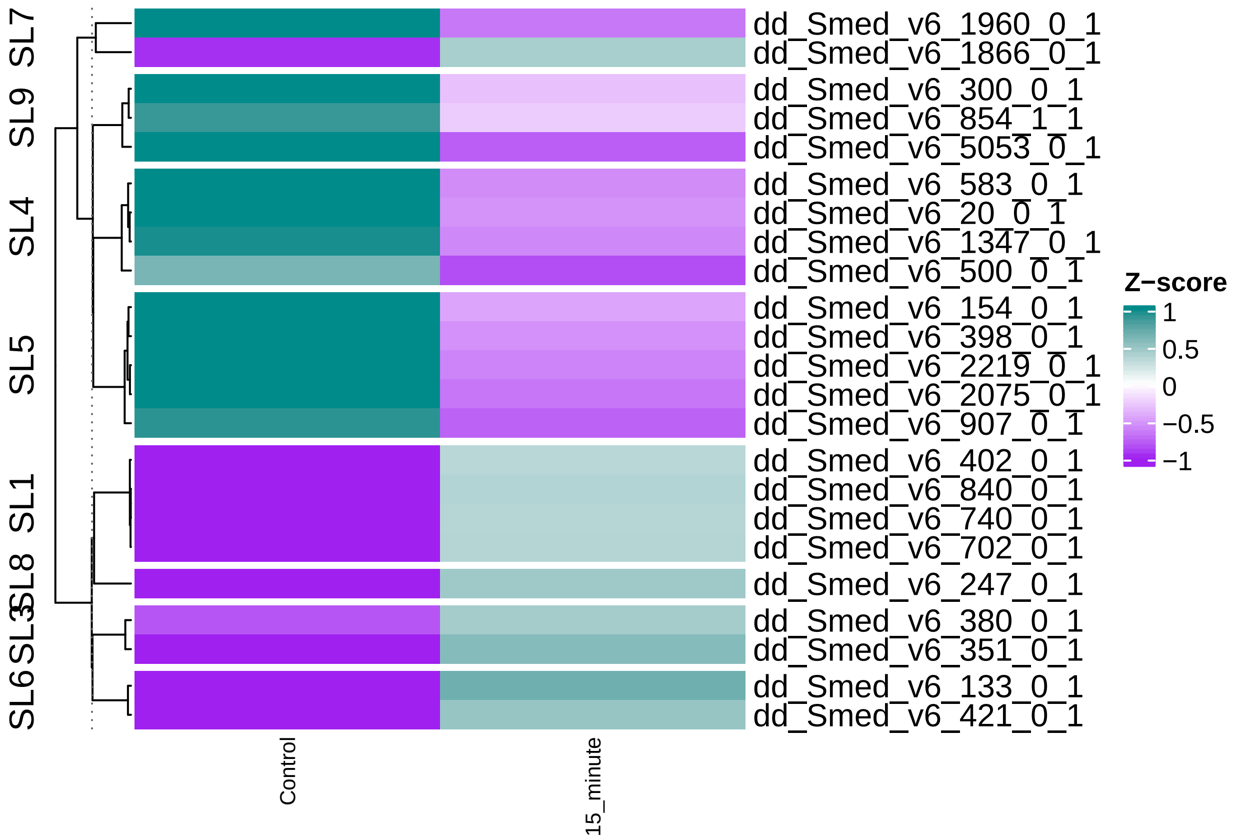
```r
Heatmap(signaling60_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE)
```



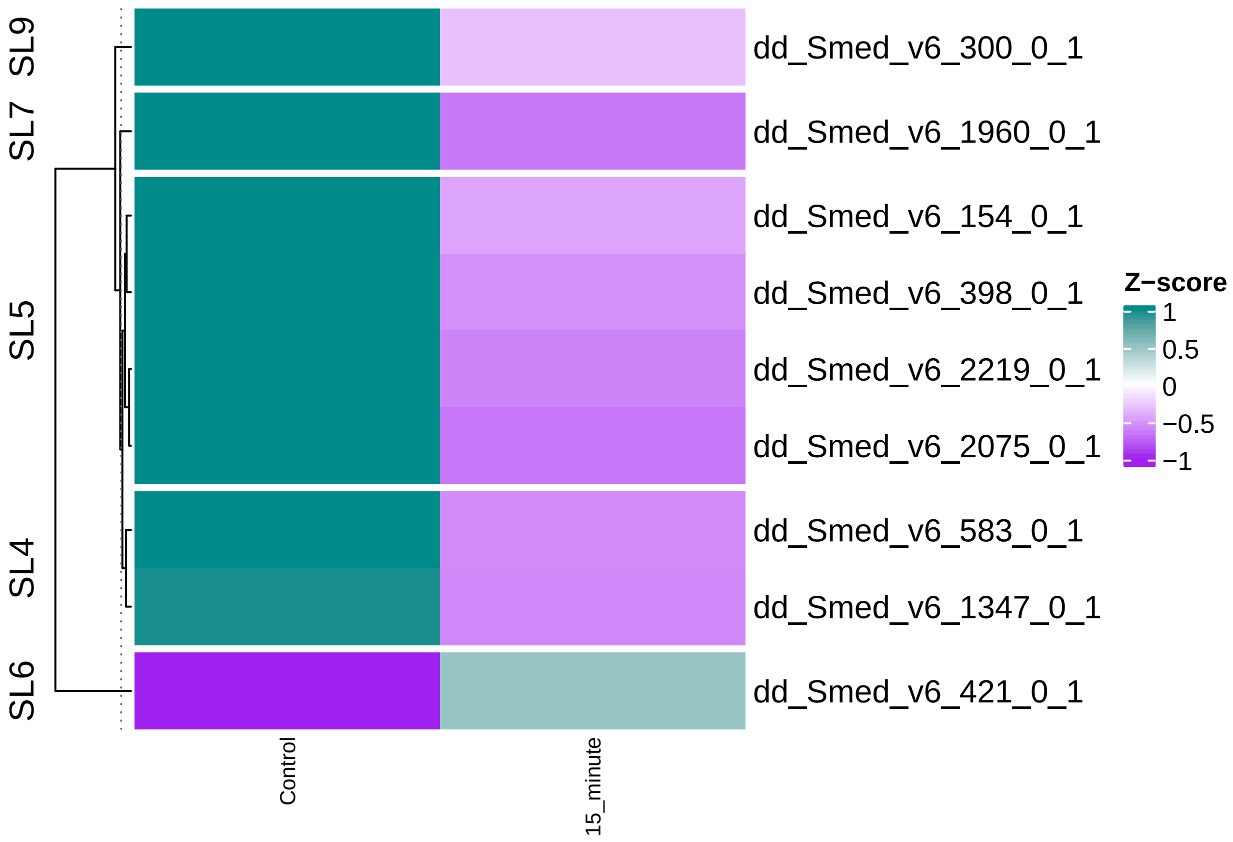
```
Heatmap(signaling60UPDOWN_hm, name = "Z-score", col = mycolz,
    column_names_gp = gpar(fontsize = 8), cluster_columns = FALSE,
    cluster_rows = TRUE)
```



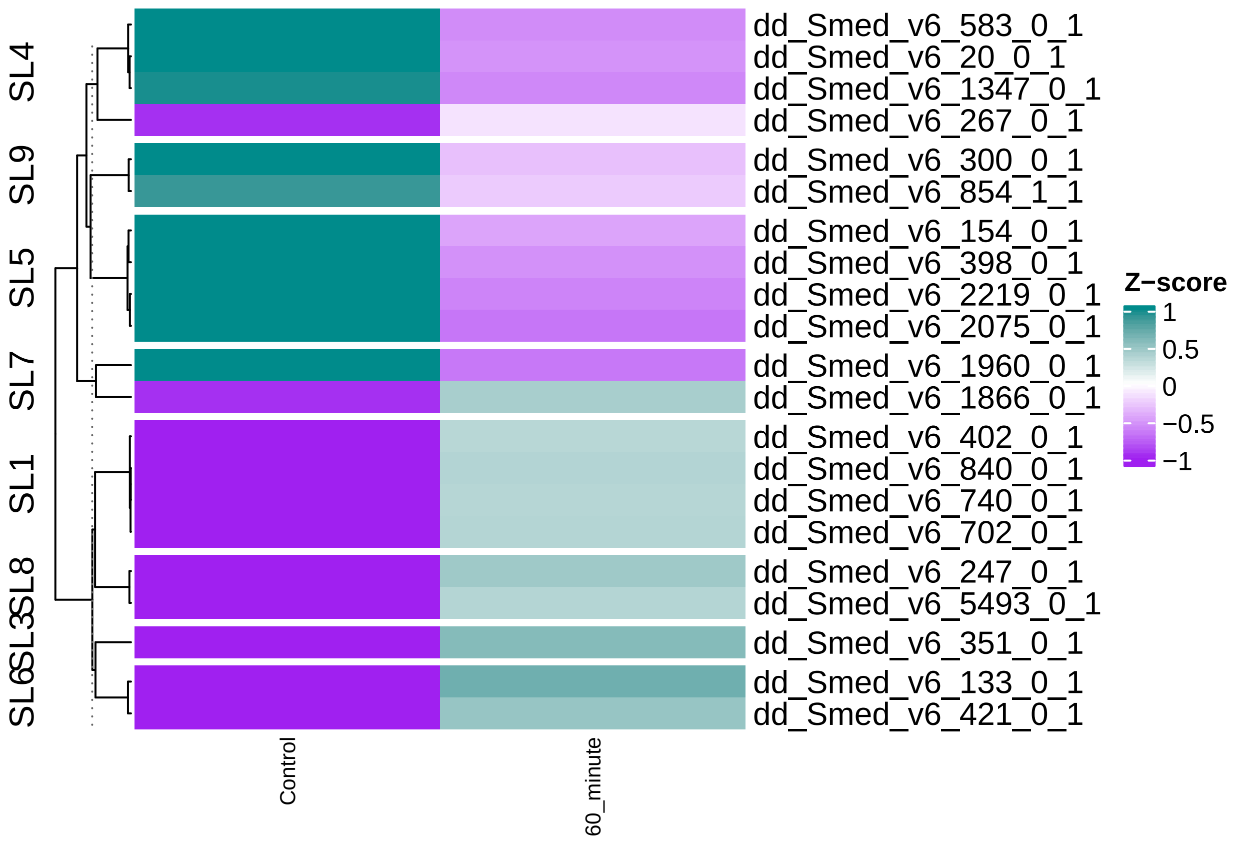
```
Heatmap(sublethal15_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE, split = sublethal15$Sub.lethal..SL...cell.cluster)
```



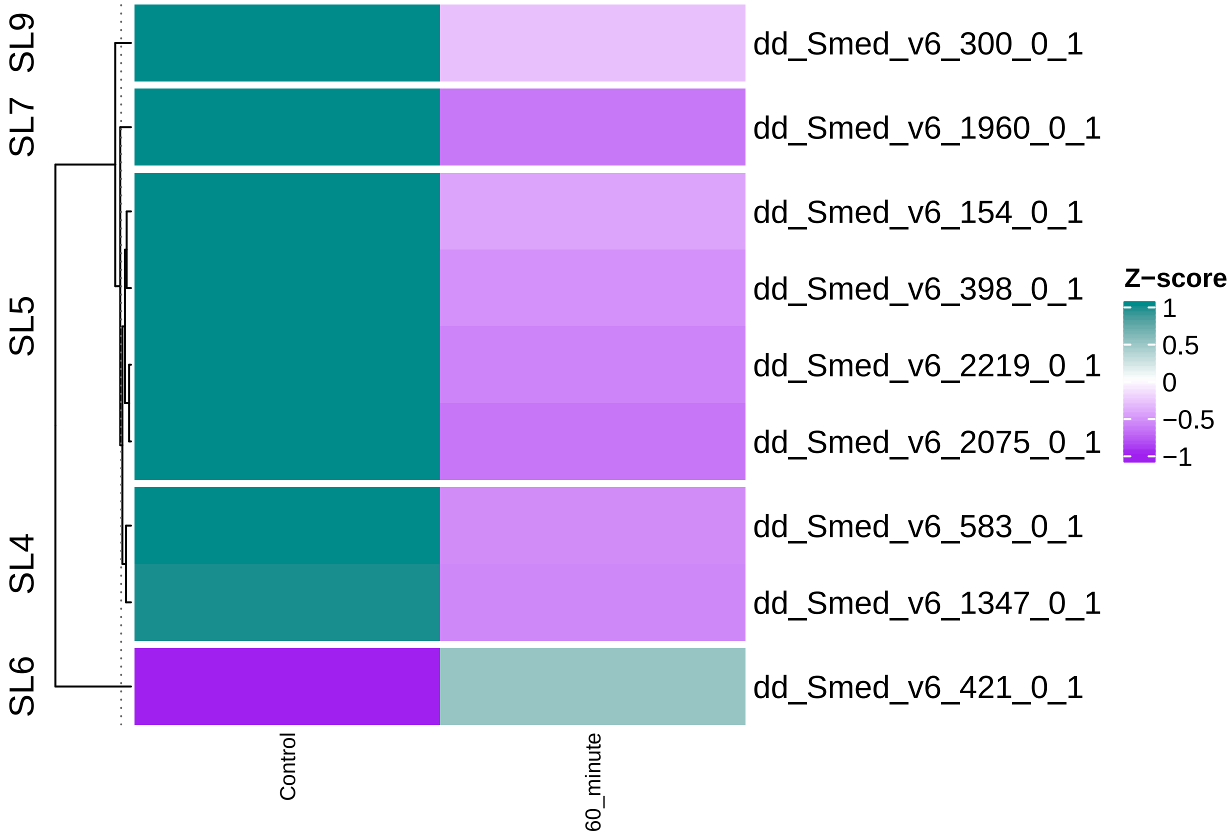
```r
Heatmap(sublethal15UPDOWN_hm, name = "Z-score", col = mycolz,
    column_names_gp = gpar(fontsize = 8), cluster_columns = FALSE,
    cluster_rows = TRUE, split = sublethal15UPDOWN$Sub.lethal..SL...cell.cluster)
```



```
Heatmap(sublethal60_hm, name = "Z-score", col = mycolz, column_names_gp = gpar(fontsize = 8),
    cluster_columns = FALSE, cluster_rows = TRUE, split = sublethal60$Sub.lethal..SL...cell.cluster)
```



```r
Heatmap(sublethal60UPDOWN_hm, name = "Z-score", col = mycolz,
    column_names_gp = gpar(fontsize = 8), cluster_columns = FALSE,
    cluster_rows = TRUE, split = sublethal60UPDOWN$Sub.lethal..SL...cell.cluster)
```



## Gene set enrichment analysis for 15 minute and 60 minute timepoints Next a gene enrichment analysis was performed using the topGO platform downloaded from Bioconductor here: :https://bioconductor.org/packages/release/bioc/html/topGO.html. The documentation for this platform can be found here: https://bioconductor.org/packages/release/bioc/vignettes/topGO/inst/doc/topGO.pdf. I used gene ontology terms annotated to the dd_Smed_v6 transcriptome from the planmine database found here: http://planmine.mpi-cbg.de/planmine/begin.do. Additionally I used the gene rankings from my analysis above to determine pathway enrichment. The Kolmogorv-Smirnov test is used to determine pathway significance.

```
library("topGO")
```

```
Loading required package: BiocGenerics

Loading required package: parallel


Attaching package: 'BiocGenerics'

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

    clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
    clusterExport, clusterMap, parApply, parCapply, parLapply,
    parLapplyLB, parRapply, parSapply, parSapplyLB

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

    combine, intersect, setdiff, union
```

```
The following object is masked from ’package:limma’:

    plotMA


The following objects are masked from ’package:stats’:

    IQR, mad, sd, var, xtabs


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

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


Loading required package: graph



Attaching package: ’graph’


The following object is masked from ’package:circlize’:

    degree


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: GO.db


Loading required package: AnnotationDbi


Loading required package: stats4


Loading required package: IRanges


Loading required package: S4Vectors



Attaching package: ’S4Vectors’


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

    first, rename
```

```
The following object is masked from ’package:gplots’:

    space

The following object is masked from ’package:base’:

    expand.grid


Attaching package: ’IRanges’

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

    collapse, desc, slice


Attaching package: ’AnnotationDbi’

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

    select



Loading required package: SparseM


Attaching package: ’SparseM’

The following object is masked from ’package:base’:

    backsolve


groupGOTerms:   GOBPTerm, GOMFTerm, GOCCTerm environments built.


Attaching package: ’topGO’

The following object is masked from ’package:IRanges’:

    members

The following object is masked from ’package:grid’:

    depth
```

```r
library("genefilter")
```

```

Attaching package: ’genefilter’
```

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

    dist2
```

```
genes <- read.delim("2018.5.7_geneID2GO.txt")
annot <- genes[, 1:2]
names(annot) <- c("gene", "GO_ID")
write.table(annot, file = "2018.5.20_gene2GO.map", sep = "\t",
    quote = F, row.names = F, col.names = F)

geneID2GO <- readMappings(file = "2018.5.20_gene2GO.map")
head(geneID2GO)
```

```
$dd_Smed_v6_10003_0_1
[1] "GO:0045454"

$dd_Smed_v6_100044_0_1
[1] "GO:0003777"

$dd_Smed_v6_100044_0_1
[1] "GO:0005524"

$dd_Smed_v6_100044_0_1
[1] "GO:0007018"

$dd_Smed_v6_100044_0_1
[1] "GO:0008017"

$dd_Smed_v6_10006_0_1
[1] "GO:0001522"
```

```
ctrlv15 <- as.matrix(toptable_15vctrlred)
ctrlv60 <- as.matrix(toptable_60vctrlred)

ctrlv15FDR <- ctrlv15[, 5]
ctrlv60FDR <- ctrlv60[, 5]

topDiffGenes <- function(allScore) {
    +return(allScore < 0.05)
}
##### 15 min v ctrl ############################# Create topGO
##### data object for Biological Process
GOdata <- new("topGOdata", description = "Gene Enrichment with 15 mins pDCS",
    ontology = "BP", allGenes = ctrlv15FDR, geneSelectionFun = topDiffGenes,
    annot = annFUN.gene2GO, gene2GO = geneID2GO)
```

```
Building most specific GOs .....

    ( 176 GO terms found. )


Build GO DAG topology .........
```

```
    ( 736 GO terms and 1472 relations. )


Annotating nodes ...............

    ( 871 genes annotated to the GO terms. )
```

```r
# obtain list of genes in GO object
golist <- genes(GOdata)
numGenes(GOdata)
```

```
[1] 871
```

```r
head(golist)
```

```
[1] "dd_Smed_v6_663_0_1"  "dd_Smed_v6_1595_0_1" "dd_Smed_v6_1769_0_1"
[4] "dd_Smed_v6_219_0_1"  "dd_Smed_v6_8024_0_1" "dd_Smed_v6_257_0_1"
```

```r
graph(GOdata)
```

```
A graphNEL graph with directed edges
Number of Nodes = 736
Number of Edges = 1472
```

```r
# Creates list of significant gene IDs but not with
# associated GO terms
sg <- sigGenes(GOdata)
sg <- sigGenes(GOdata)
numSigGenes(GOdata)
```

```
[1] 117
```

```r
write.table(sg, file = "2021.2.23_15vctrl_BP_GOgenelist.txt",
    sep = "\t", quote = F, row.names = F, col.names = F)

# Kolmogorov-Smirnov testing
resultKS <- runTest(GOdata, algorithm = "elim", statistic = "ks")
```

```


             -- Elim Algorithm --

         the algorithm is scoring 736 nontrivial nodes
         parameters:
             test statistic: ks
             cutOff: 0.01
             score order: increasing


     Level 14:  3 nodes to be scored    (0 eliminated genes)
```

```
    Level 13:  7 nodes to be scored    (0 eliminated genes)

    Level 12:  16 nodes to be scored   (0 eliminated genes)

    Level 11:  30 nodes to be scored   (0 eliminated genes)

    Level 10:  44 nodes to be scored   (0 eliminated genes)

    Level 9:   81 nodes to be scored   (0 eliminated genes)

    Level 8:   98 nodes to be scored   (0 eliminated genes)

    Level 7:   102 nodes to be scored  (4 eliminated genes)

    Level 6:   115 nodes to be scored  (4 eliminated genes)

    Level 5:   120 nodes to be scored  (4 eliminated genes)

    Level 4:   63 nodes to be scored   (4 eliminated genes)

    Level 3:   43 nodes to be scored   (4 eliminated genes)

    Level 2:   13 nodes to be scored   (4 eliminated genes)

    Level 1:   1 nodes to be scored    (503 eliminated genes)
```

```
tab <- GenTable(GOdata, KS = resultKS, topNodes = length(resultKS@score))
write.table(tab, file = "2021.2.23_15vctrl_BP_topnodes.txt",
    sep = "\t", quote = F, row.names = F, col.names = F)

################################################## Create topGO data object for Cellular Component
GOdata <- new("topGOdata", description = "Gene Enrichment with 15 mins pDCS",
    ontology = "CC", allGenes = ctrlv15FDR, geneSelectionFun = topDiffGenes,
    annot = annFUN.gene2GO, gene2GO = geneID2GO)
```

```
Building most specific GOs .....
```

```
    ( 94 GO terms found. )


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

    ( 248 GO terms and 424 relations. )


Annotating nodes ...............

    ( 944 genes annotated to the GO terms. )
```

```r
# obtain list of genes in GO object
golist <- genes(GOdata)
numGenes(GOdata)
```

```
[1] 944
```

```r
head(golist)
```

```
[1] "dd_Smed_v6_4005_0_1" "dd_Smed_v6_2696_0_1" "dd_Smed_v6_1089_0_1"
[4] "dd_Smed_v6_7837_0_1" "dd_Smed_v6_1731_0_1" "dd_Smed_v6_2119_0_1"
```

```r
graph(GOdata)
```

```
A graphNEL graph with directed edges
Number of Nodes = 248
Number of Edges = 424
```

```r
# Creates list of significant gene IDs but not with
# associated GO terms
sg <- sigGenes(GOdata)
sg <- sigGenes(GOdata)
numSigGenes(GOdata)
```

```
[1] 125
```

```r
write.table(sg, file = "2021.2.23_15vctrl_CC_GOgenelist.txt",
    sep = "\t", quote = F, row.names = F, col.names = F)

# Kolmogorov-Smirnov testing
resultKS <- runTest(GOdata, algorithm = "elim", statistic = "ks")
```

```


             -- Elim Algorithm --

         the algorithm is scoring 248 nontrivial nodes
         parameters:
             test statistic: ks
             cutOff: 0.01
             score order: increasing
```

```
     Level 13:  1 nodes to be scored    (0 eliminated genes)

     Level 12:  5 nodes to be scored    (0 eliminated genes)

     Level 11:  8 nodes to be scored    (0 eliminated genes)

     Level 10:  16 nodes to be scored   (0 eliminated genes)

     Level 9:   21 nodes to be scored   (0 eliminated genes)

     Level 8:   34 nodes to be scored   (0 eliminated genes)

     Level 7:   41 nodes to be scored   (5 eliminated genes)

     Level 6:   32 nodes to be scored   (5 eliminated genes)

     Level 5:   30 nodes to be scored   (5 eliminated genes)

     Level 4:   32 nodes to be scored   (5 eliminated genes)

     Level 3:   23 nodes to be scored   (16 eliminated genes)

     Level 2:   4 nodes to be scored    (16 eliminated genes)

     Level 1:   1 nodes to be scored    (16 eliminated genes)
```

```r
tab <- GenTable(GOdata, KS = resultKS, topNodes = length(resultKS@score))
write.table(tab, file = "2021.2.23_15vctrl_CC_topnodes.txt",
    sep = "\t", quote = F, row.names = F, col.names = F)

################################################## Create topGO data object for Molecular Function
GOdata <- new("topGOdata", description = "Gene Enrichment with 15 mins pDCS",
    ontology = "MF", allGenes = ctrlv15FDR, geneSelectionFun = topDiffGenes,
    annot = annFUN.gene2GO, gene2GO = geneID2GO)
```

```
Building most specific GOs .....
```

```
    ( 363 GO terms found. )


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

    ( 692 GO terms and 903 relations. )


Annotating nodes ...............

    ( 7047 genes annotated to the GO terms. )
```

```r
# obtain list of genes in GO object
golist <- genes(GOdata)
numGenes(GOdata)
```

```
[1] 7047
```

```r
head(golist)
```

```
[1] "dd_Smed_v6_659_0_1"   "dd_Smed_v6_3281_0_1"  "dd_Smed_v6_11100_0_1"
[4] "dd_Smed_v6_5635_0_1"  "dd_Smed_v6_12472_0_1" "dd_Smed_v6_2075_0_1"
```

```r
graph(GOdata)
```

```
A graphNEL graph with directed edges
Number of Nodes = 692
Number of Edges = 903
```

```r
# Creates list of significant gene IDs but not with
# associated GO terms
sg <- sigGenes(GOdata)
sg <- sigGenes(GOdata)
numSigGenes(GOdata)
```

```
[1] 914
```

```r
write.table(sg, file = "2021.2.23_15vctrl_MF_GOgenelist.txt",
    sep = "\t", quote = F, row.names = F, col.names = F)

# Kolmogorov-Smirnov testing
resultKS <- runTest(GOdata, algorithm = "elim", statistic = "ks")
```

```


             -- Elim Algorithm --

         the algorithm is scoring 692 nontrivial nodes
         parameters:
             test statistic: ks
             cutOff: 0.01
             score order: increasing
```

```
     Level 12:  1 nodes to be scored    (0 eliminated genes)

     Level 11:  3 nodes to be scored    (0 eliminated genes)

     Level 10:  14 nodes to be scored   (0 eliminated genes)

     Level 9:   34 nodes to be scored   (0 eliminated genes)

     Level 8:   62 nodes to be scored   (0 eliminated genes)

     Level 7:   103 nodes to be scored  (0 eliminated genes)

     Level 6:   190 nodes to be scored  (0 eliminated genes)

     Level 5:   142 nodes to be scored  (36 eliminated genes)

     Level 4:   94 nodes to be scored   (167 eliminated genes)

     Level 3:   37 nodes to be scored   (167 eliminated genes)

     Level 2:   11 nodes to be scored   (1504 eliminated genes)

     Level 1:   1 nodes to be scored    (2041 eliminated genes)
```

```
tab <- GenTable(GOdata, KS = resultKS, topNodes = length(resultKS@score))
write.table(tab, file = "2021.2.23_15vctrl_MF_topnodes.txt",
    sep = "\t", quote = F, row.names = F, col.names = F)

##### 60 min v ctrl ############################## Create topGO
##### data object for Biological Process
GOdata <- new("topGOdata", description = "Gene Enrichment with 60 mins pDCS",
    ontology = "BP", allGenes = ctrlv60FDR, geneSelectionFun = topDiffGenes,
    annot = annFUN.gene2GO, gene2GO = geneID2GO)
```

```
Building most specific GOs .....

    ( 176 GO terms found. )
```

```
Build GO DAG topology ..........

    ( 736 GO terms and 1472 relations. )


Annotating nodes ...............

    ( 871 genes annotated to the GO terms. )
```

```r
# obtain list of genes in GO object
golist <- genes(GOdata)
numGenes(GOdata)
```

```
[1] 871
```

```r
head(golist)
```

```
[1] "dd_Smed_v6_3230_0_1" "dd_Smed_v6_1231_0_1" "dd_Smed_v6_1595_0_1"
[4] "dd_Smed_v6_147_0_1"  "dd_Smed_v6_8163_0_1" "dd_Smed_v6_219_0_1"
```

```r
graph(GOdata)
```

```
A graphNEL graph with directed edges
Number of Nodes = 736
Number of Edges = 1472
```

```r
# Creates list of significant gene IDs but not with
# associated GO terms
sg <- sigGenes(GOdata)
sg <- sigGenes(GOdata)
numSigGenes(GOdata)
```

```
[1] 121
```

```r
write.table(sg, file = "2021.2.23_60vctrl_BP_GOgenelist.txt",
    sep = "\t", quote = F, row.names = F, col.names = F)

# Kolmogorov-Smirnov testing
resultKS <- runTest(GOdata, algorithm = "elim", statistic = "ks")
```

```

             -- Elim Algorithm --

         the algorithm is scoring 736 nontrivial nodes
         parameters:
             test statistic: ks
             cutOff: 0.01
             score order: increasing
```

```
Level 14:  3 nodes to be scored    (0 eliminated genes)

Level 13:  7 nodes to be scored    (0 eliminated genes)

Level 12:  16 nodes to be scored   (0 eliminated genes)

Level 11:  30 nodes to be scored   (0 eliminated genes)

Level 10:  44 nodes to be scored   (0 eliminated genes)

Level 9:   81 nodes to be scored   (0 eliminated genes)

Level 8:   98 nodes to be scored   (0 eliminated genes)

Level 7:   102 nodes to be scored  (4 eliminated genes)

Level 6:   115 nodes to be scored  (4 eliminated genes)

Level 5:   120 nodes to be scored  (4 eliminated genes)

Level 4:   63 nodes to be scored   (4 eliminated genes)

Level 3:   43 nodes to be scored   (8 eliminated genes)

Level 2:   13 nodes to be scored   (9 eliminated genes)

Level 1:   1 nodes to be scored    (9 eliminated genes)
```

```
tab <- GenTable(GOdata, KS = resultKS, topNodes = length(resultKS@score))
write.table(tab, file = "2021.2.23_60vctrl_BP_topnodes.txt",
    sep = "\t", quote = F, row.names = F, col.names = F)

################################################## Create topGO data object for Cellular Component
GOdata <- new("topGOdata", description = "Gene Enrichment with 60 mins pDCS",
    ontology = "CC", allGenes = ctrlv60FDR, geneSelectionFun = topDiffGenes,
    annot = annFUN.gene2GO, gene2GO = geneID2GO)
```

```
Building most specific GOs .....

    ( 94 GO terms found. )


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

    ( 248 GO terms and 424 relations. )


Annotating nodes ...............

    ( 944 genes annotated to the GO terms. )
```

```r
# obtain list of genes in GO object
golist <- genes(GOdata)
numGenes(GOdata)
```

```
[1] 944
```

```r
head(golist)
```

```
[1] "dd_Smed_v6_2696_0_1" "dd_Smed_v6_1089_0_1" "dd_Smed_v6_4005_0_1"
[4] "dd_Smed_v6_2119_0_1" "dd_Smed_v6_6150_0_1" "dd_Smed_v6_2051_0_1"
```

```r
graph(GOdata)
```

```
A graphNEL graph with directed edges
Number of Nodes = 248
Number of Edges = 424
```

```r
# Creates list of significant gene IDs but not with
# associated GO terms
sg <- sigGenes(GOdata)
sg <- sigGenes(GOdata)
numSigGenes(GOdata)
```

```
[1] 119
```

```r
write.table(sg, file = "2021.2.23_60vctrl_CC_GOgenelist.txt",
    sep = "\t", quote = F, row.names = F, col.names = F)

# Kolmogorov-Smirnov testing
resultKS <- runTest(GOdata, algorithm = "elim", statistic = "ks")
```

```
        -- Elim Algorithm --

    the algorithm is scoring 248 nontrivial nodes
    parameters:
        test statistic: ks
        cutOff: 0.01
        score order: increasing


Level 13:  1 nodes to be scored    (0 eliminated genes)


Level 12:  5 nodes to be scored    (0 eliminated genes)


Level 11:  8 nodes to be scored    (0 eliminated genes)


Level 10:  16 nodes to be scored   (0 eliminated genes)


Level 9:   21 nodes to be scored   (0 eliminated genes)


Level 8:   34 nodes to be scored   (0 eliminated genes)


Level 7:   41 nodes to be scored   (0 eliminated genes)


Level 6:   32 nodes to be scored   (0 eliminated genes)


Level 5:   30 nodes to be scored   (103 eliminated genes)


Level 4:   32 nodes to be scored   (103 eliminated genes)


Level 3:   23 nodes to be scored   (103 eliminated genes)


Level 2:   4 nodes to be scored    (103 eliminated genes)


Level 1:   1 nodes to be scored    (103 eliminated genes)
```

```r
tab <- GenTable(GOdata, KS = resultKS, topNodes = length(resultKS@score))
write.table(tab, file = "2021.2.23_60vctrl_CC_topnodes.txt",
    sep = "\t", quote = F, row.names = F, col.names = F)

################################################# Create topGO data object for Molecular Function
GOdata <- new("topGOdata", description = "Gene Enrichment with 60 mins pDCS",
    ontology = "MF", allGenes = ctrlv60FDR, geneSelectionFun = topDiffGenes,
    annot = annFUN.gene2GO, gene2GO = geneID2GO)
```

```
Building most specific GOs .....

    ( 363 GO terms found. )


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

    ( 692 GO terms and 903 relations. )


Annotating nodes ...............

    ( 7047 genes annotated to the GO terms. )
```

```r
# obtain list of genes in GO object
golist <- genes(GOdata)
numGenes(GOdata)
```

```
[1] 7047
```

```r
head(golist)
```

```
[1] "dd_Smed_v6_4392_0_1"  "dd_Smed_v6_7295_0_1"  "dd_Smed_v6_7144_0_1"
[4] "dd_Smed_v6_11100_0_1" "dd_Smed_v6_3281_0_1"  "dd_Smed_v6_1913_0_1"
```

```r
graph(GOdata)
```

```
A graphNEL graph with directed edges
Number of Nodes = 692
Number of Edges = 903
```

```r
# Creates list of significant gene IDs but not with
# associated GO terms
sg <- sigGenes(GOdata)
sg <- sigGenes(GOdata)
numSigGenes(GOdata)
```

```
[1] 893
```

```r
write.table(sg, file = "2021.2.23_60vctrl_MF_GOgenelist.txt",
    sep = "\t", quote = F, row.names = F, col.names = F)

# Kolmogorov-Smirnov testing
resultKS <- runTest(GOdata, algorithm = "elim", statistic = "ks")
```

```
             -- Elim Algorithm --

         the algorithm is scoring 692 nontrivial nodes
         parameters:
             test statistic: ks
             cutOff: 0.01
             score order: increasing


     Level 12:  1 nodes to be scored    (0 eliminated genes)


     Level 11:  3 nodes to be scored    (0 eliminated genes)


     Level 10:  14 nodes to be scored   (0 eliminated genes)


     Level 9:   34 nodes to be scored   (0 eliminated genes)


     Level 8:   62 nodes to be scored   (0 eliminated genes)


     Level 7:   103 nodes to be scored  (0 eliminated genes)


     Level 6:   190 nodes to be scored  (25 eliminated genes)


     Level 5:   142 nodes to be scored  (296 eliminated genes)


     Level 4:   94 nodes to be scored   (751 eliminated genes)


     Level 3:   37 nodes to be scored   (751 eliminated genes)


     Level 2:   11 nodes to be scored   (874 eliminated genes)


     Level 1:   1 nodes to be scored    (2645 eliminated genes)
```

```
tab <- GenTable(GOdata, KS = resultKS, topNodes = length(resultKS@score))
write.table(tab, file = "2021.2.23_60vctrl_MF_topnodes.txt",
    sep = "\t", quote = F, row.names = F, col.names = F)


# looking at termination of G-protein coupled recepto... GO
# term goID <- tab2[1, 'GO.ID']
# print(showGroupDensity(GOdata, goID, ranks = TRUE))
```