ELT-2 Regulated Genes

Rtpw

2/12/2020

```
knitr::opts_chunk$set(echo = TRUE)
```

Next steps

HIGH - perform GO on up and down regulated genes LOW - elt-2 chip or promoter motifs of up and down regulated genes

Done steps

- Do Z score of row normalization, divide by the standard deviation

Improvements

Align RNA seq data to ce11 genome with more recent annotation.

Libraries

```
library(biomaRt)
library(DESeq2)
```

```
## Loading required package: S4Vectors
## Loading required package: stats4
## Loading required package: BiocGenerics
## Loading required package: parallel
##
## Attaching package: 'BiocGenerics'
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
```

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

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

##
## Attaching package: 'S4Vectors'

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

## Loading required package: IRanges

## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb

## Loading required package: SummarizedExperiment

## 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: DelayedArray

## Loading required package: matrixStats

##
## Attaching package: 'matrixStats'

## The following objects are masked from 'package:Biobase':
##
##     anyMissing, rowMedians

## Loading required package: BiocParallel

##
## Attaching package: 'DelayedArray'

## The following objects are masked from 'package:matrixStats':
##
##     colMaxs, colMins, colRanges, rowMaxs, rowMins, rowRanges

## The following objects are masked from 'package:base':
##
##     aperm, apply, rowsum
library(tidyverse)

## -- Attaching packages --------------------------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.0     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.5
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------------------------- tidyverse_conflicts() --
## x dplyr::collapse()   masks IRanges::collapse()
## x dplyr::combine()    masks Biobase::combine(), BiocGenerics::combine()
## x dplyr::count()      masks matrixStats::count()
## x dplyr::desc()       masks IRanges::desc()
## x tidyr::expand()     masks S4Vectors::expand()
## x dplyr::filter()     masks stats::filter()
## x dplyr::first()      masks S4Vectors::first()
## x dplyr::lag()        masks stats::lag()
## x ggplot2::Position() masks BiocGenerics::Position(), base::Position()
## x purrr::reduce()     masks GenomicRanges::reduce(), IRanges::reduce()
## x dplyr::rename()     masks S4Vectors::rename()
## x dplyr::select()     masks biomaRt::select()
## x purrr::simplify()   masks DelayedArray::simplify()
## x dplyr::slice()      masks IRanges::slice()
```

```
library(pheatmap)
library(readxl)
library(matrixStats)
```

Source required functions.

```
source("./RWC23_Functions.R")
```

Differentail Expression

Load data

```
RNAcounts <- read.csv("./01_input/Table_S1_Raw_Read_Counts.csv", header=TRUE, row.names = 1)
```

This count file contains more samples than what I want to analyze. Subset the columns to just have `wt_sorted_*` and `elt2D_sorted_*`. Also select columns that correspond to ce11 genome assembly, since this is the genome used for the ChIP-seq analysis.

```
cts <- RNAcounts %>% select(wt_sorted_1, wt_sorted_2, wt_sorted_3, wt_sorted_4, elt2D_sorted_1, elt2D_s
head(cts)
```

```
##                wt_sorted_1 wt_sorted_2 wt_sorted_3 wt_sorted_4 elt2D_sorted_1
## WBGene00000001         532         462         458         525            546
## WBGene00000002         192         165         185         195            169
## WBGene00000003         577         425         649         694            371
## WBGene00000004        2111        1794        2131        1999           1158
## WBGene00000005          11           8          13           6              9
## WBGene00000007          71          82          69          92             19
##                elt2D_sorted_2 elt2D_sorted_3 elt2D_sorted_4 elt2Delt7D_sorted_1
## WBGene00000001            919            575            661                799
## WBGene00000002            226            157            147                291
## WBGene00000003            557            405            429                510
## WBGene00000004           1832           1233           1288               1481
## WBGene00000005             11              8             10                  3
```

```
## WBGene00000007             36             15             18             22
##                 elt2Delt7D_sorted_2 elt2Delt7D_sorted_3
## WBGene00000001                 675                 482
## WBGene00000002                 271                 194
## WBGene00000003                 489                 425
## WBGene00000004                1304                1347
## WBGene00000005                   7                   1
## WBGene00000007                  22                  13
```

make coldata

```
coldata <- data.frame(condition = c("wt", "wt", "wt", "wt", "elt2D", "elt2D", "elt2D", "elt2D", "elt2Del
coldata
```

```
##                       condition
## wt_sorted_1                  wt
## wt_sorted_2                  wt
## wt_sorted_3                  wt
## wt_sorted_4                  wt
## elt2D_sorted_1            elt2D
## elt2D_sorted_2            elt2D
## elt2D_sorted_3            elt2D
## elt2D_sorted_4            elt2D
## elt2Delt7D_sorted_1 elt2Delt7D
## elt2Delt7D_sorted_2 elt2Delt7D
## elt2Delt7D_sorted_3 elt2Delt7D
```

Check that column matrix and coldata match

```
all(rownames(coldata) == colnames(cts))
```

```
## [1] TRUE
```

Generate DESeqDataSet

```
dds <- DESeqDataSetFromMatrix(countData = cts, colData = coldata, design = ~ condition)

# add gene names
moreFeatures <- data.frame(gene_name = RNAcounts$gene_id_val, sequence_id = RNAcounts$sequence_id_list)
mcols(dds) <- DataFrame(mcols(dds), moreFeatures)
mcols(dds)
```

```
## DataFrame with 16708 rows and 2 columns
##                 gene_name sequence_id
##                  <factor>    <factor>
## WBGene00000001     aap-1  Y110A7A.10
## WBGene00000002     aat-1     F27C8.1
## WBGene00000003     aat-2     F07C3.7
## WBGene00000004     aat-3     F52H2.2
## WBGene00000005     aat-4   T13A10.10
## ...                  ...         ...
## WBGene00043705        NA          NA
## WBGene00015013        NA          NA
## WBGene00008743        NA          NA
## WBGene00235114        NA          NA
## WBGene00077643        NA          NA
```

Tell DESeq which samples are "control" and which are "control" vs "treatment". This sets up the fold change

comparison manually rather than letting the alphabetical determination of factor levels.

with this step: logfoldchange(elt2D/wt)

```
dds$condition <- factor(dds$condition, levels = c("wt", "elt2D", "elt2Delt7D"))
```

Perform differential expression analysis

```
dds <- DESeq(dds)
```

```
## estimating size factors
## estimating dispersions
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates
## fitting model and testing
```

```
res <- results(dds)


# Convert res to dataframe
res.df <- as.data.frame(res)

# Export the results table
#write.csv(res.df, file = "./02_DESeq2/200218_L1_wt_vs_elt2D_results.csv")

# Print results table information
head(res)
```

```
## log2 fold change (MLE): condition elt2Delt7D vs wt
## Wald test p-value: condition elt2Delt7D vs wt
## DataFrame with 6 rows and 6 columns
##                         baseMean     log2FoldChange             lfcSE
##                        <numeric>          <numeric>         <numeric>
## WBGene00000001 591.515264995531  0.374766904987348 0.100832250987597
## WBGene00000002 196.941946891564  0.431645564160724 0.122403401331081
## WBGene00000003 499.031409070873 -0.309270596061639 0.142658833176119
## WBGene00000004 1597.07886131967 -0.542369069007949 0.105519396720516
## WBGene00000005 7.82653928628189  -1.41145497345249 0.629860204886353
## WBGene00000007 41.2854757245755   -2.0673345543798 0.274722012968266
##                             stat               pvalue                padj
##                        <numeric>            <numeric>           <numeric>
## WBGene00000001  3.71673647386337 0.000201812748989876 0.000660611216589531
## WBGene00000002  3.52641805265847 0.000421221497079934  0.00129548701082756
## WBGene00000003 -2.16790358631232   0.0301660229459714   0.0605533480446432
## WBGene00000004 -5.13999402824958 2.74747200542808e-07  1.4149353669423e-06
## WBGene00000005 -2.24090196920309   0.0250324256740502   0.0515377102412202
## WBGene00000007 -7.52518712295034 5.26448735432883e-14 5.66900376974855e-13
```

```
summary(res)
```

```
##
## out of 16707 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 4609, 28%
```

```
## LFC < 0 (down)     : 4404, 26%
## outliers [1]       : 16, 0.096%
## low counts [2]     : 0, 0%
## (mean count < 0)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

Perform vst and rlog transformation of read counts

```
vsd <- vst(dds)
rld <- rlog(dds)
```

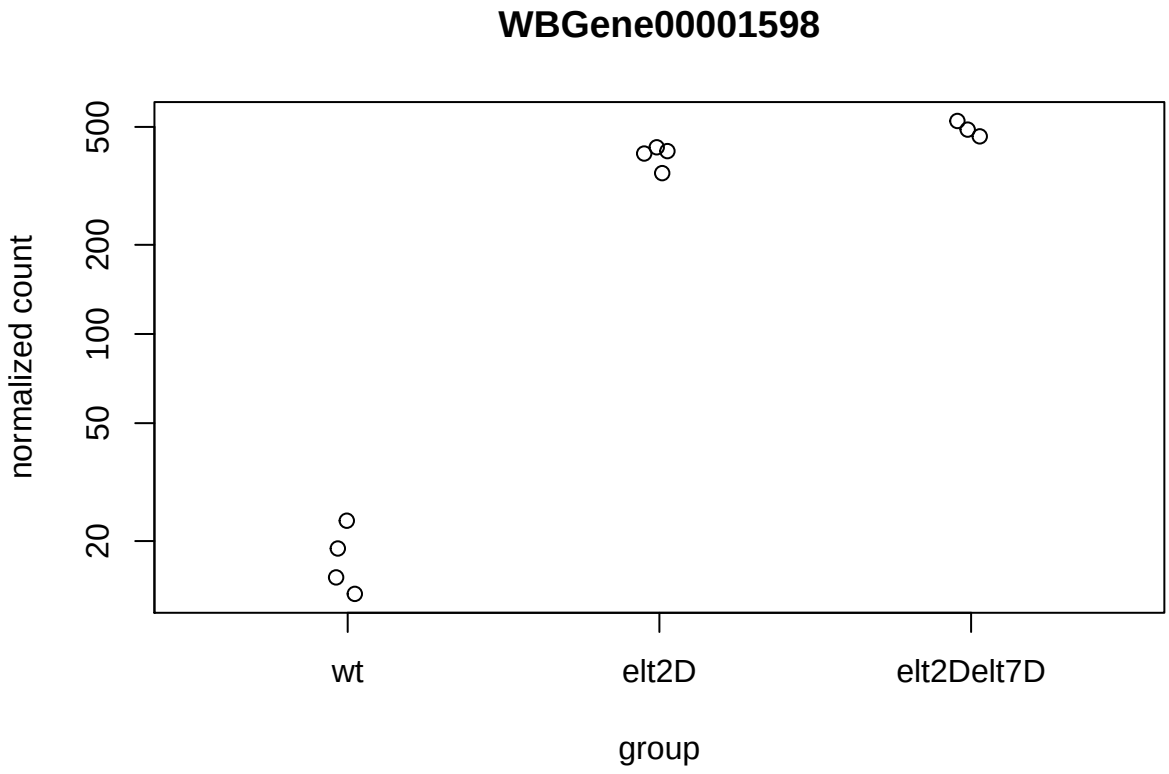
Explore Differential Expression

Determine if glh-1 is upregulated in elt-2(-)
glh-1 = WBGene00001598
glh-1 is a germline specific gene
It is upregulated

```
res.df[rownames(res.df) == "WBGene00001598",]
```

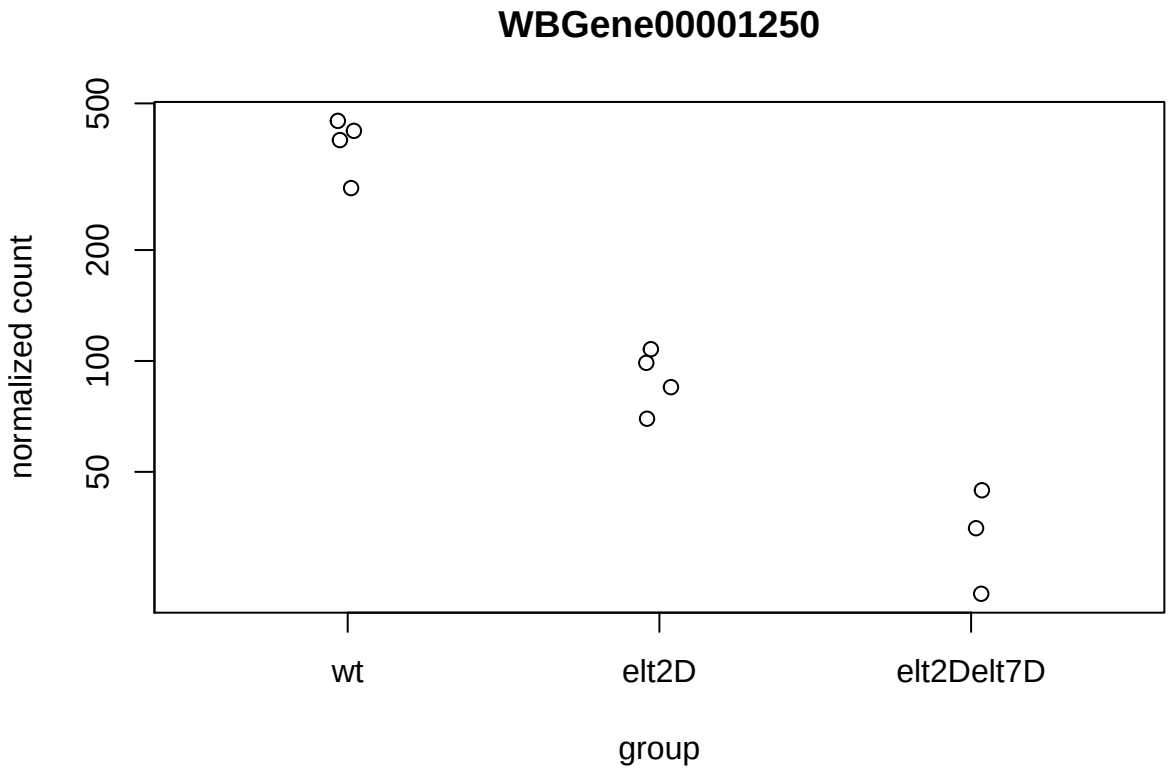
```
##                baseMean log2FoldChange     lfcSE     stat        pvalue
## WBGene00001598 285.4151       4.843145 0.2083577 23.24438 1.621405e-119
##                         padj
## WBGene00001598 5.106203e-117
```

```
plotCounts(dds, gene = "WBGene00001598")
```



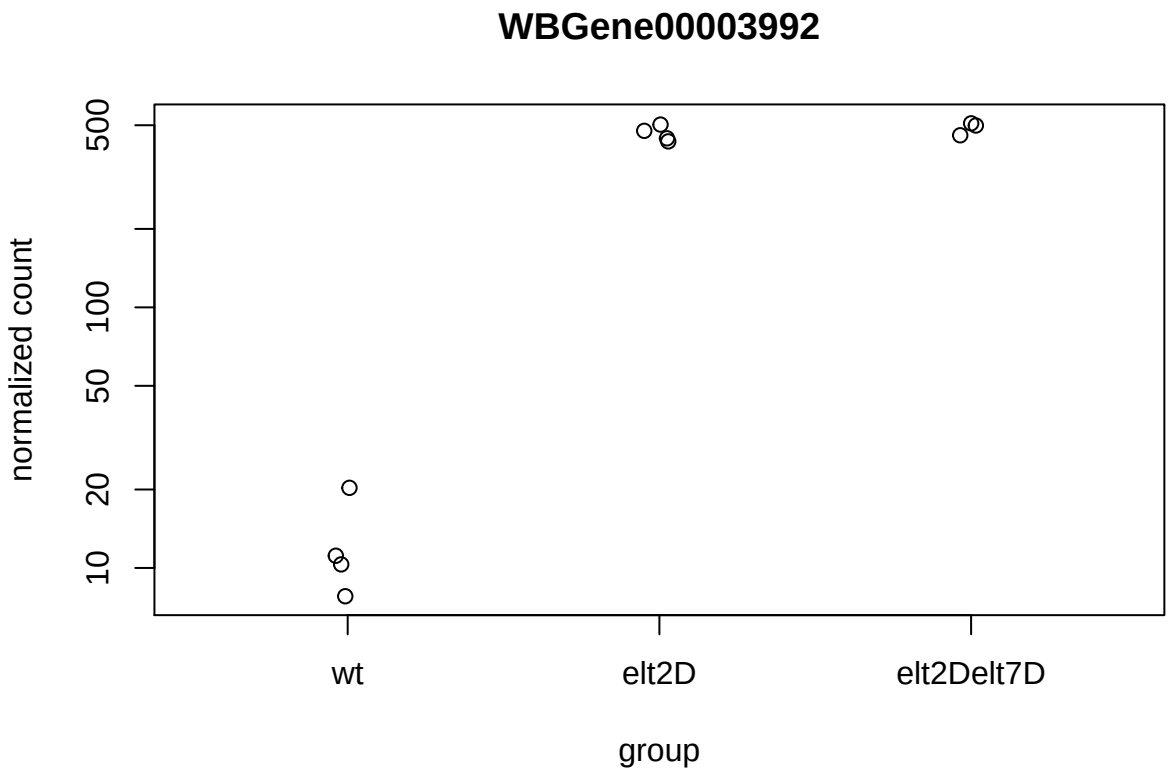
See if elt-2 is depleted
It is depleted

```
plotCounts(dds, gene = "WBGene00001250")
```



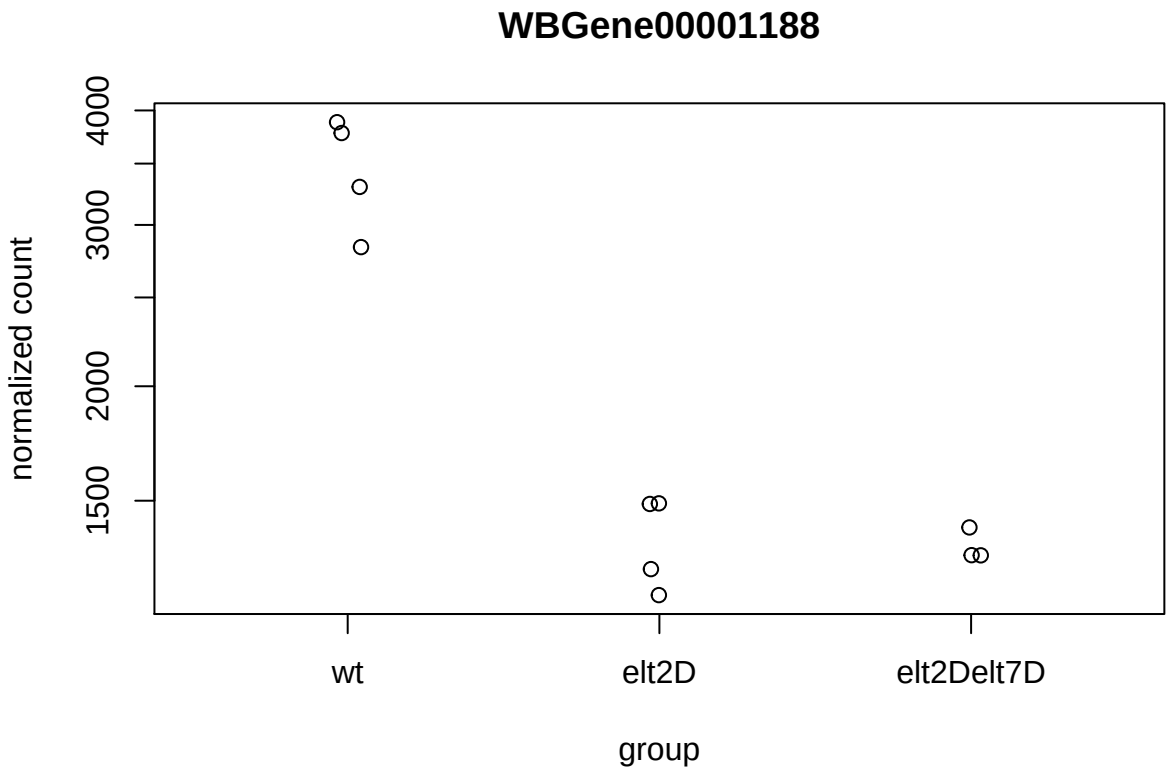
See if pgl-1 is enriched
pgl-1 = WBGene00003992
Looks like it is enriched.

```
plotCounts(dds, gene = "WBGene00003992")
```



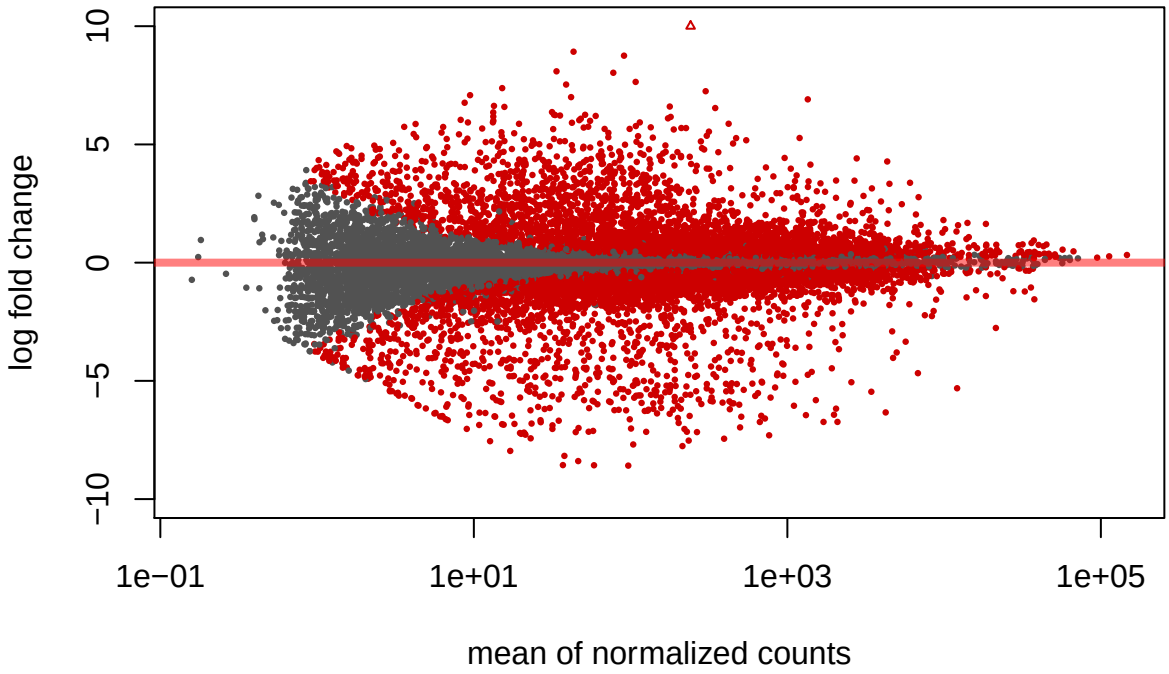
see if egl-20 (ligand of Wnt pathway) is expressed
Also is depleted

```
plotCounts(dds, gene = "WBGene00001188")
```



Make an MA plot for all the data.

```
plotMA(res, ylim = c(-10, 10))
```



ELT-2 Bound and Reuglated Genes

This section will integrate the L1 stage ELT-2 ChIP data analyzed by David.

Load in data.

```
elt2_peaks <- read_excel("./01_input/200406_peaksForBigBed.xlsx")
head(elt2_peaks)
```

```
## # A tibble: 6 x 13
##   chrom start   end `peak name` WBID  mapping cluster `cluster descri~ kweight
##   <chr> <dbl> <dbl> <chr>       <chr> <chr>     <dbl> <chr>              <dbl>
## 1 chrI   3691  4222 ELT2peak00~ WBGe~ overla~       0 Not-changing or~   0
## 2 chrI  11044 11533 ELT2peak00~ WBGe~ overla~       0 Not-changing or~   0
## 3 chrI  13560 14890 ELT2peak00~ WBGe~ inside        0 Not-changing or~   0
## 4 chrI  15179 15647 ELT2peak00~ WBGe~ inside        0 Not-changing or~   0
## 5 chrI  16706 17483 ELT2peak00~ WBGe~ overla~       3 L3-high            0.997
## 6 chrI  26789 27576 ELT2peak00~ WBGe~ downst~       0 Not-changing or~   0
## # ... with 4 more variables: LE <dbl>, L1 <dbl>, L3 <dbl>, `peak summit
## #   agreement` <dbl>
```

```
names(elt2_peaks)[names(elt2_peaks)=="WBID"] <- "WBGeneID"

names(elt2_peaks)
```

```
##  [1] "chrom"                "start"                "end"
##  [4] "peak name"            "WBGeneID"             "mapping"
##  [7] "cluster"              "cluster description"  "kweight"
## [10] "LE"                   "L1"                   "L3"
## [13] "peak summit agreement"
```

```
# Subset for genes bound in the L1 stage
elt2_L1_peaks <- elt2_peaks %>%
  select(WBGeneID, L1) %>%
  filter(L1 == 1) %>%
  select(WBGeneID) %>% unique()

wTF3.0 <- read.csv("./01_input/TF3-0_namesonly.txt", sep = "\t", header = TRUE) %>% select(WBGeneID)
```

WT and elt-2 (-) Analysis

First focus the analysis only on genes changing between wildtype and elt-2 (-) samples only.

Use the functions in `RWC23_Functions.R` to subset and row normalize the matrix.

```
wt_elt2_counts <- subset(assay(rld), select = wt_sorted_1:elt2D_sorted_4)

elt2_bound_matrix <- matrix_select(wt_elt2_counts, elt2_L1_peaks$WBGeneID)

elt2bound_rownormMatrix <- row_normalize_matrix_cutoff(
  count_matrix = elt2_bound_matrix,
  variance_cutoff =  0.5
  )

head(elt2bound_rownormMatrix)
```

```
##                wt_sorted_1 wt_sorted_2 wt_sorted_3 wt_sorted_4 elt2D_sorted_1
## WBGene00000067  -0.6349486  -0.7162717  -0.5976849  -0.7991346      0.7163916
## WBGene00000136   0.8830630   0.7788424   0.9443123   0.8462292     -0.8363313
## WBGene00000214   1.0059107   1.2721320   1.2921968   0.6844732     -1.2613747
## WBGene00000215   1.5264065   1.7582623   1.5177720   1.4515993     -1.8277849
## WBGene00000218   0.7968876   1.1356586   0.9211414   0.4544767     -0.9668025
## WBGene00000219  -0.7905671  -0.2483230  -0.7344153  -1.0518433      0.5740744
##                elt2D_sorted_2 elt2D_sorted_3 elt2D_sorted_4
## WBGene00000067      0.6291500      0.7166546      0.6858437
## WBGene00000136     -0.9800856     -0.7232705     -0.9127594
## WBGene00000214     -0.9080412     -1.2016349     -0.8836618
## WBGene00000215     -1.6160352     -1.3386745     -1.4715456
## WBGene00000218     -0.7074971     -0.8325836     -0.8012810
## WBGene00000219      0.7993100      0.6411026      0.8106617
```

Replace the WBGeneIDs in the row name with gene names.

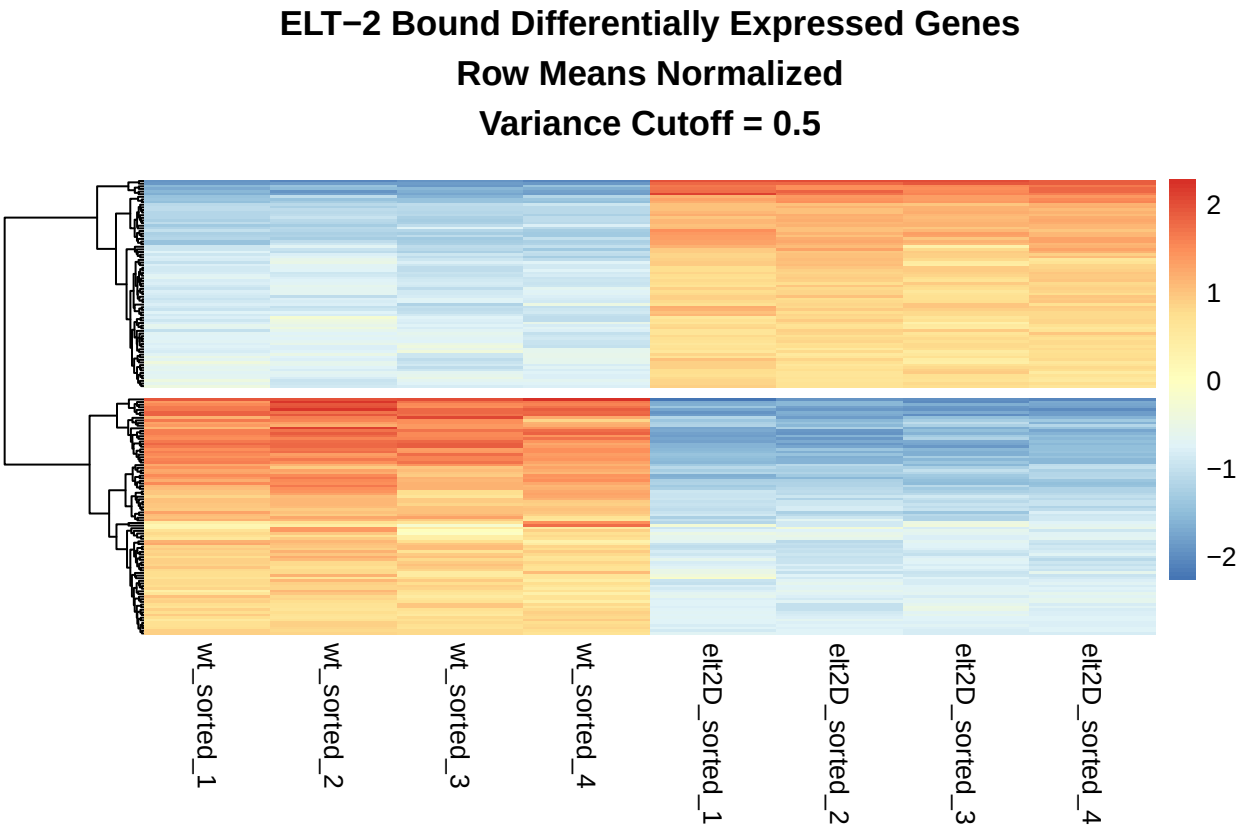
```
elt2bound_rowNormGeneNameMatrix <- id2name(elt2bound_rownormMatrix)

head(elt2bound_rowNormGeneNameMatrix)
```

```
##       wt_sorted_1 wt_sorted_2 wt_sorted_3 wt_sorted_4 elt2D_sorted_1
## act-5  -0.6349486  -0.7162717  -0.5976849  -0.7991346      0.7163916
## amt-4   0.8830630   0.7788424   0.9443123   0.8462292     -0.8363313
## asp-1   1.0059107   1.2721320   1.2921968   0.6844732     -1.2613747
## asp-2   1.5264065   1.7582623   1.5177720   1.4515993     -1.8277849
## asp-5   0.7968876   1.1356586   0.9211414   0.4544767     -0.9668025
## asp-6  -0.7905671  -0.2483230  -0.7344153  -1.0518433      0.5740744
##       elt2D_sorted_2 elt2D_sorted_3 elt2D_sorted_4
```

```
## act-5      0.6291500      0.7166546      0.6858437
## amt-4     -0.9800856     -0.7232705     -0.9127594
## asp-1     -0.9080412     -1.2016349     -0.8836618
## asp-2     -1.6160352     -1.3386745     -1.4715456
## asp-5     -0.7074971     -0.8325836     -0.8012810
## asp-6      0.7993100      0.6411026      0.8106617
```

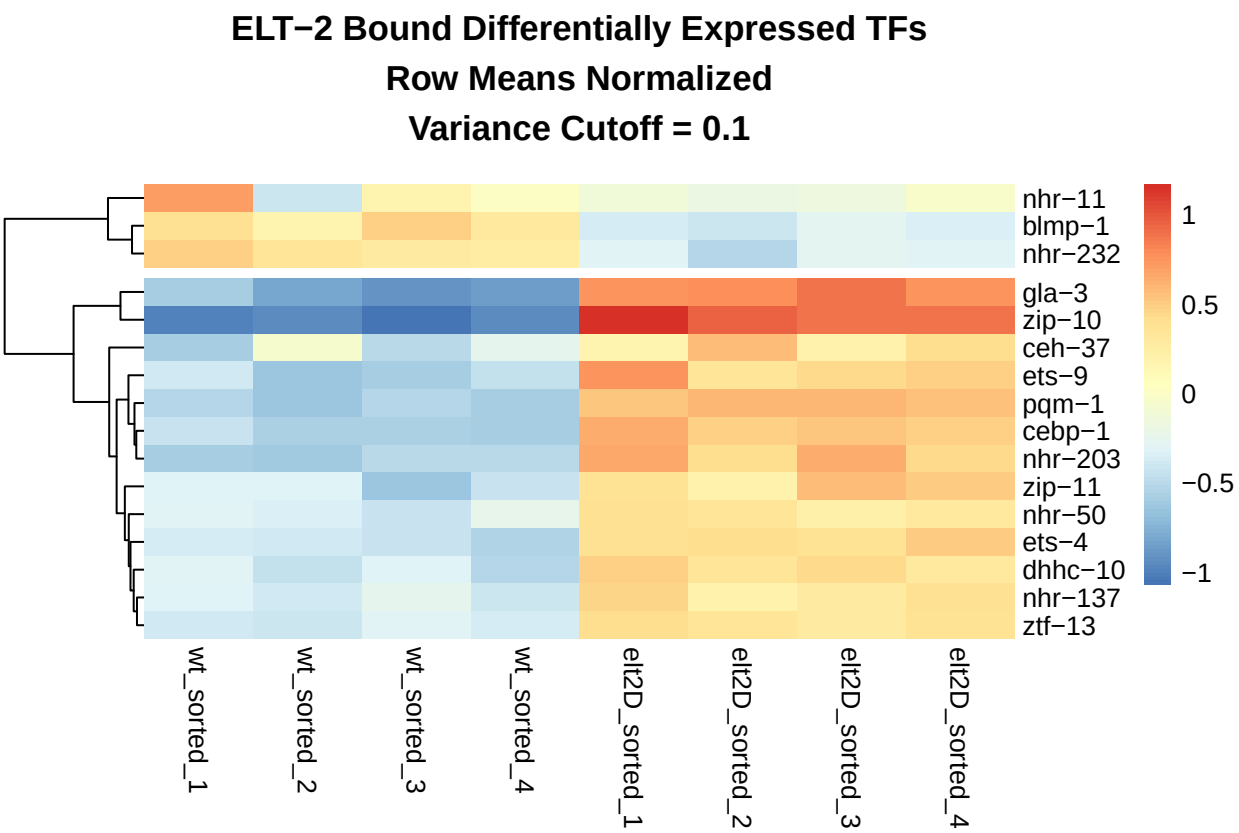
Now plot a heatmap of ELT-2 regulated genes.

```
myPheatmap(elt2bound_rowNormGeneNameMatrix, "ELT-2 Bound Differentially Expressed Genes\nRow Means Norma
```



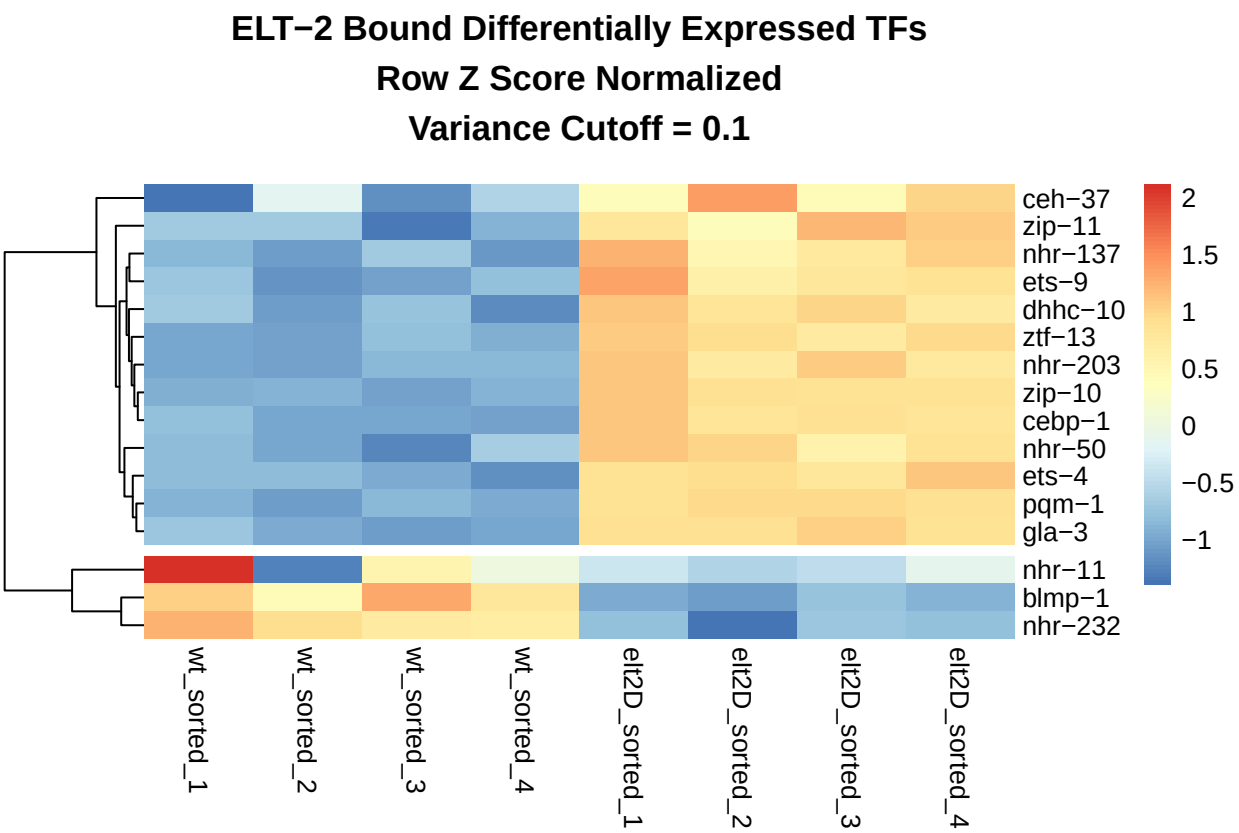
Do a similar analysis for TFs only.

```
elt2_bound_TF_matrix <- matrix_select(count_matrix = elt2_bound_matrix, gene_subset_vector = wTF3.0$WBG

elt2_bound_TF_rowNorm_matrix <- row_normalize_matrix_cutoff(
  count_matrix = elt2_bound_TF_matrix,
  variance_cutoff =  0.1
  )


elt2_bound_TF_rowNorm_matrix <- id2name(elt2_bound_TF_rowNorm_matrix)

mysmallPheatmap(elt2_bound_TF_rowNorm_matrix, "ELT-2 Bound Differentially Expressed TFs\nRow Means Norma
```



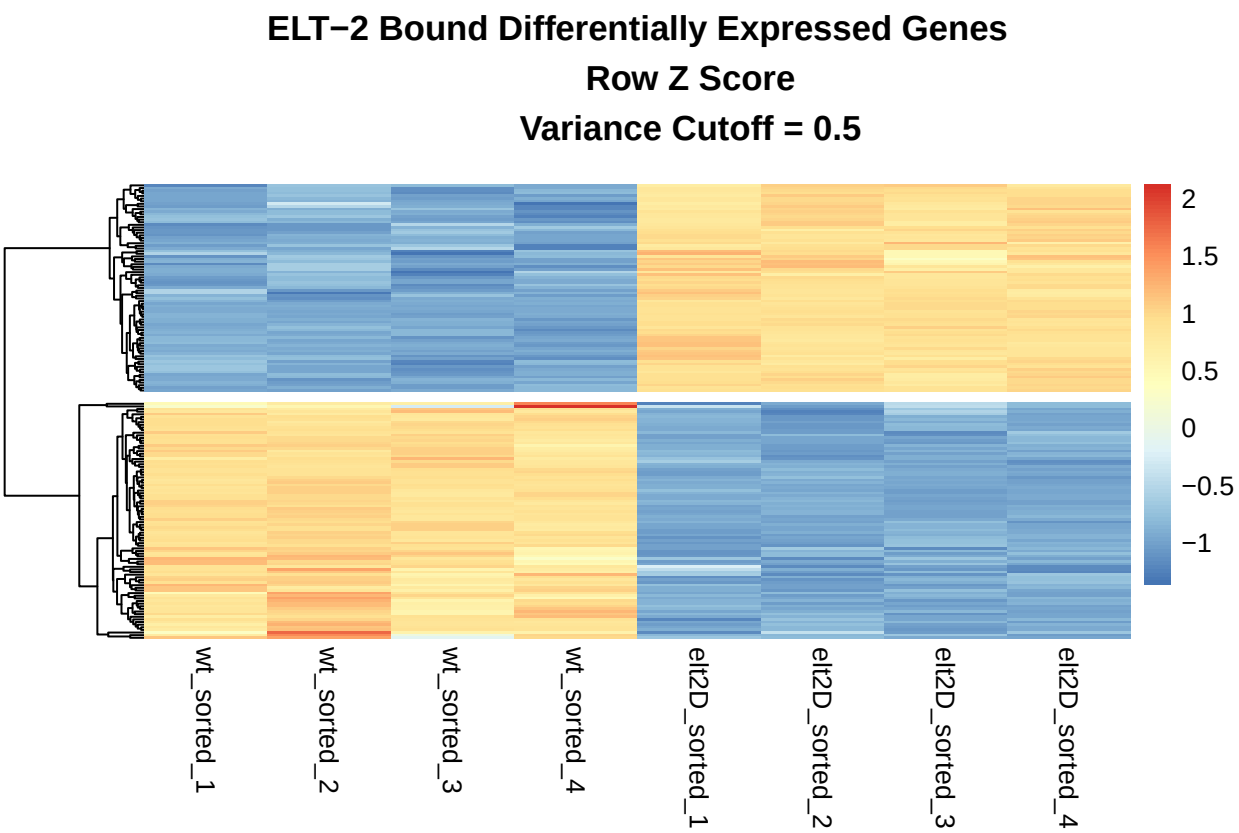
Do it for transcription factors.

```
elt2_bound_TF_zscore_matrix <- row_zscore_matrix_cutoff(
  count_matrix = elt2_bound_TF_matrix,
  variance_cutoff =  0.1
  )

elt2_bound_TF_zscore_matrix <- id2name(elt2_bound_TF_zscore_matrix)

mysmallPheatmap(elt2_bound_TF_zscore_matrix,
           title = "ELT-2 Bound Differentially Expressed TFs\nRow Z Score Normalized\nVariance Cutoff =
           rowspace = 2)
```



Do Z Score normalization for all genes.

```
elt2bound_zscore_matrix <- row_zscore_matrix_cutoff(
  count_matrix = elt2_bound_matrix,
  variance_cutoff =  0.5
  )

elt2bound_zscore_matrix <- id2name(elt2bound_zscore_matrix)

myPheatmap(elt2bound_zscore_matrix,
           title = "ELT-2 Bound Differentially Expressed Genes
           Row Z Score
           Variance Cutoff = 0.5",
           rowspace = 2)
```



Use pairwise differential expression as regulated gene filter

Load in data.

```
up_in_wt_v_elt2 <- read_excel("01_input/Table_S4_Pairwise_Diff_Expression.xlsx",
    sheet = "1_up_in_wt_v_elt2", col_names = FALSE)
```

```
## New names:
## * `` -> ...1
```

```
down_in_wt_v_elt2 <- read_excel("01_input/Table_S4_Pairwise_Diff_Expression.xlsx",
    sheet = "2_down_in_wt_v_elt2", col_names = FALSE)
```

```
## New names:
## * `` -> ...1
```

```
up_in_wt_v_elt7 <- read_excel("01_input/Table_S4_Pairwise_Diff_Expression.xlsx",
    sheet = "3_up_in_wt_v_elt7", col_names = FALSE)
```

```
## New names:
## * `` -> ...1
```

```
up_in_wt_v_elt7elt2 <- read_excel("01_input/Table_S4_Pairwise_Diff_Expression.xlsx",
    sheet = "5_up_in_wt_v_elt7elt2", col_names = FALSE)
```

```
## New names:
## * `` -> ...1
```

```
down_in_wt_v_elt7elt2 <- read_excel("01_input/Table_S4_Pairwise_Diff_Expression.xlsx",
    sheet = "6_down_in_wt_v_elt7elt2", col_names = FALSE)
```

```
## New names:
## * `` -> ...1
```

```
up_in_elt2_v_elt7elt2 <- read_excel("01_input/Table_S4_Pairwise_Diff_Expression.xlsx",
    sheet = "7_up_in_elt2_v_elt7elt2", col_names = FALSE)
```

```
## New names:
## * `` -> ...1
```

```
down_in_elt2_v_elt7elt2 <- read_excel("01_input/Table_S4_Pairwise_Diff_Expression.xlsx",
    sheet = "8_down_in_elt2_v_elt7elt2", col_names = FALSE)
```

```
## New names:
## * `` -> ...1
```

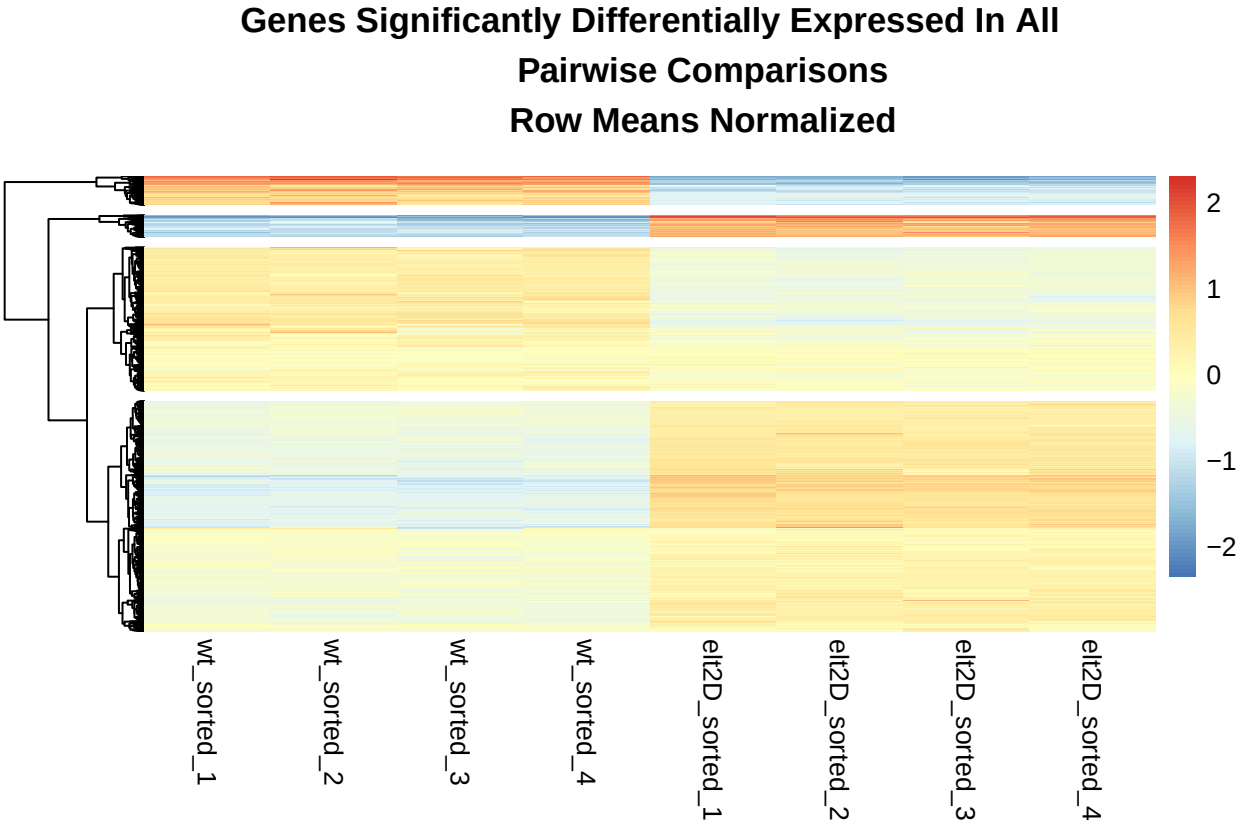
```
colnames(up_in_wt_v_elt2) <- c("WBGeneID")
colnames(down_in_wt_v_elt2) <- c("WBGeneID")
colnames(up_in_wt_v_elt7) <- c("WBGeneID")
colnames(up_in_wt_v_elt7elt2) <- c("WBGeneID")
colnames(down_in_wt_v_elt7elt2) <- c("WBGeneID")
colnames(up_in_elt2_v_elt7elt2) <- c("WBGeneID")
colnames(down_in_elt2_v_elt7elt2) <- c("WBGeneID")
```

Make a union of these lists with unique WBGeneIDs.

```
union_elt2elt7_DE <- data.frame(WBGeneID = c(up_in_wt_v_elt2$WBGeneID,
                        down_in_wt_v_elt2$WBGeneID,
                        up_in_wt_v_elt7$WBGeneID,
                        up_in_wt_v_elt7elt2$WBGeneID,
                        down_in_wt_v_elt7elt2$WBGeneID,
                        up_in_elt2_v_elt7elt2$WBGeneID,
                        down_in_elt2_v_elt7elt2$WBGeneID
                        )) %>% unique()
```

Subset count matrix for presence in union of all pairwise comparisons.

```
all_pairwise_subset <- matrix_select(wt_elt2_counts, union_elt2elt7_DE$WBGeneID)

row_normalize_matrix <- function(count_matrix){
  namevarRowNormalized <- count_matrix - rowMeans(count_matrix)
  return(namevarRowNormalized)
}

all_pairwise_subset_rownorm <- row_normalize_matrix(all_pairwise_subset)

myPheatmap(all_pairwise_subset_rownorm,
           title = "Genes Significantly Differentially Expressed In All
           Pairwise Comparisons
           Row Means Normalized",
           rowspace = 4)
```



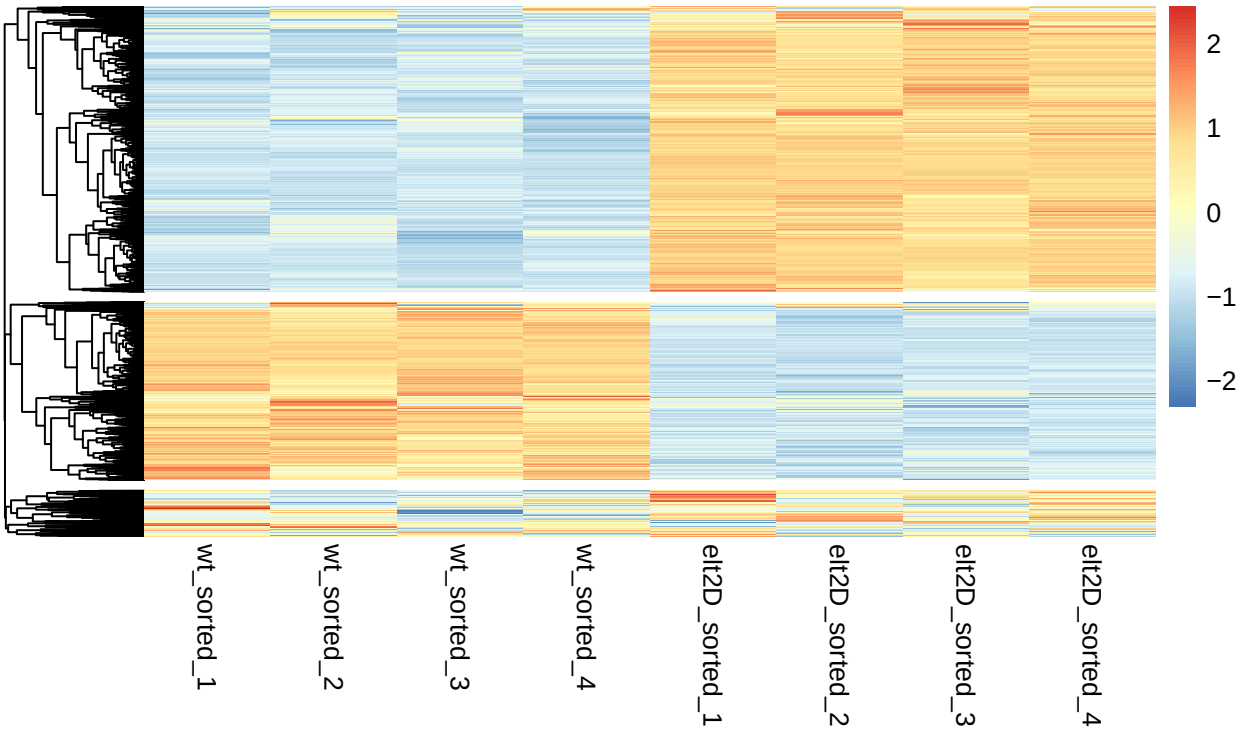
Hard to see anything useful with this.

Do the same thing but use Z score. Maybe there will be more detail.

```
all_pairwise_subset_Zscore <- row_zscore_matrix(all_pairwise_subset)

# remove columns with NA
all_pairwise_subset_Zscore <- all_pairwise_subset_Zscore[complete.cases(all_pairwise_subset_Zscore), ]

unique(is.na(all_pairwise_subset_Zscore))
```

```
##                wt_sorted_1 wt_sorted_2 wt_sorted_3 wt_sorted_4 elt2D_sorted_1
## WBGene00000007       FALSE       FALSE       FALSE       FALSE          FALSE
##                elt2D_sorted_2 elt2D_sorted_3 elt2D_sorted_4
## WBGene00000007          FALSE          FALSE          FALSE
```

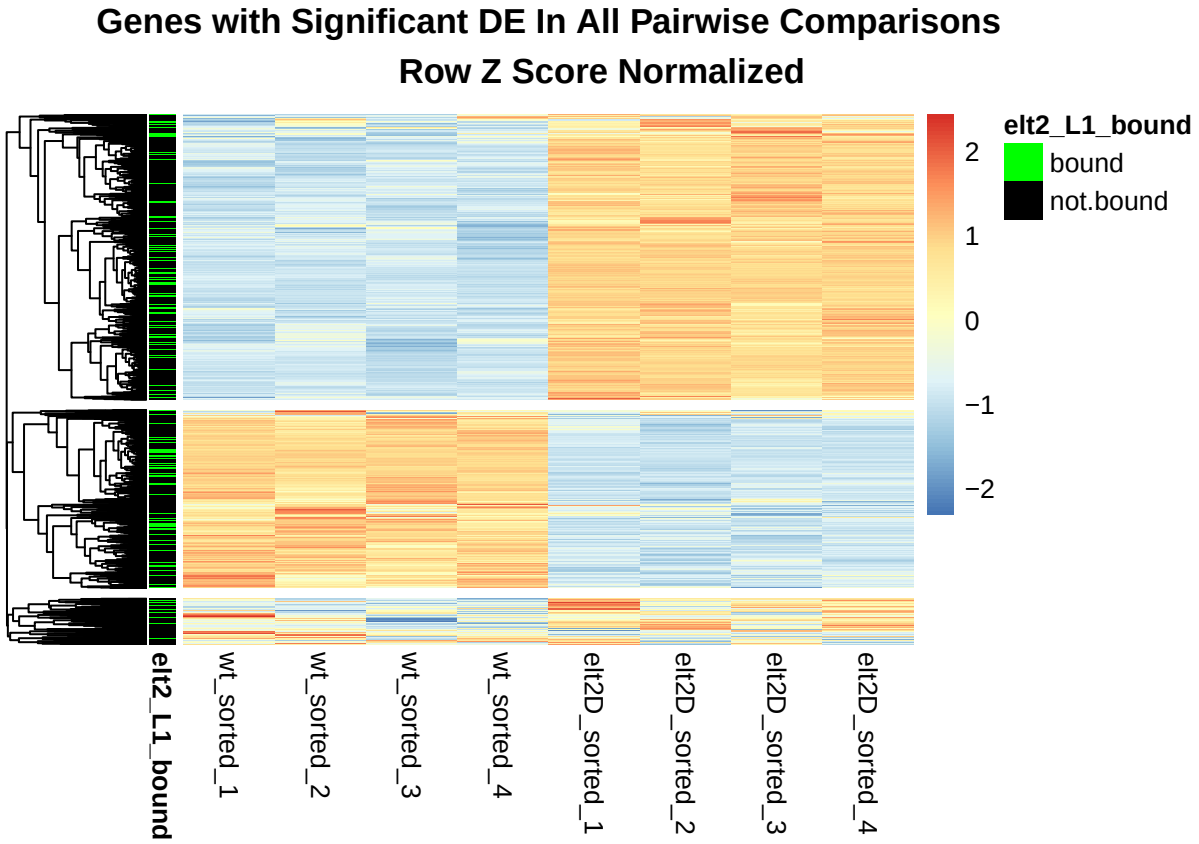
```
myPheatmap(all_pairwise_subset_Zscore,
           title = "Genes with Significant DE In All Pairwise Comparisons
           Row Z Score Normalized",
           rowspace = 3)
```



Clusters are a little more obvious.

Add annotation to side of heatmap that indicates binding of ELT-2 in L1 stage.

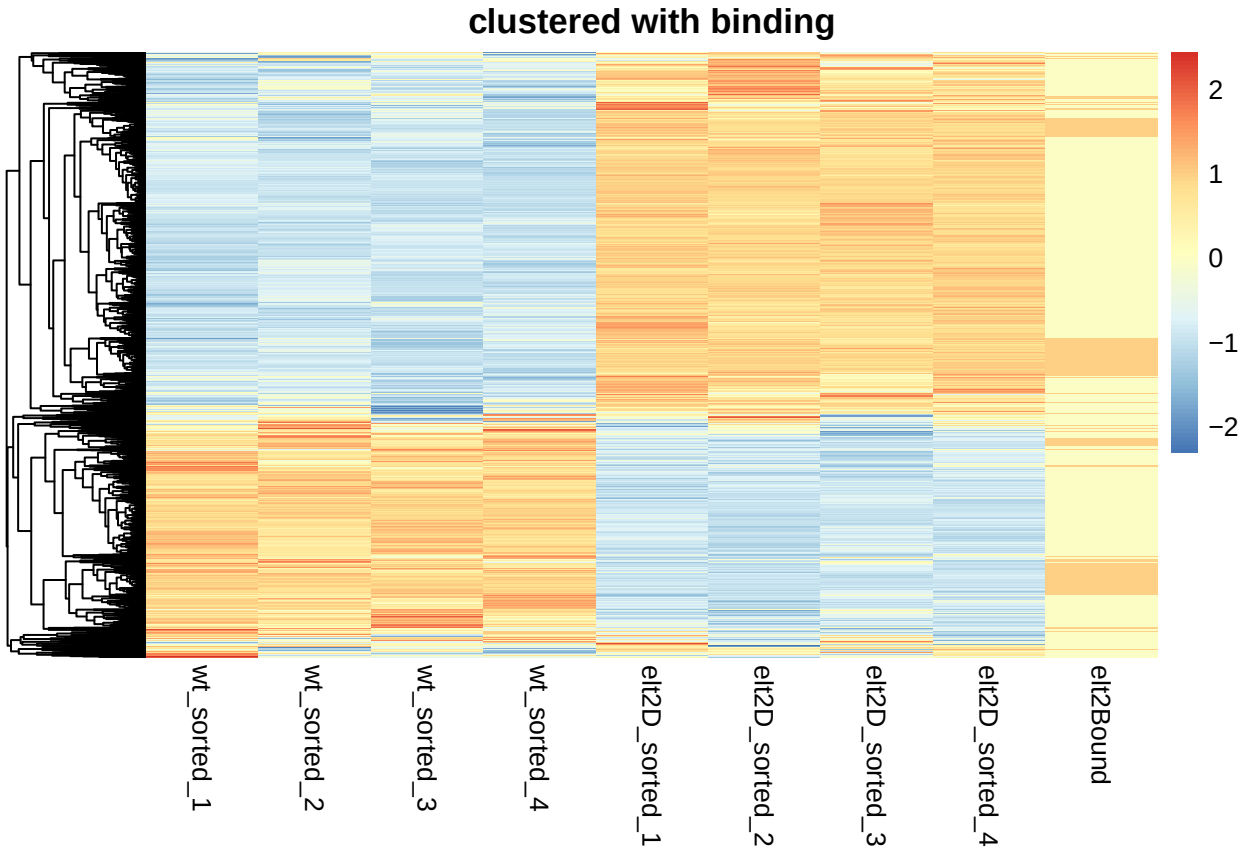
```
my_row_anno <- data.frame(elt2_L1_bound = ifelse(test = rownames(all_pairwise_subset_Zscore) %in% elt2_


rownames(my_row_anno) <- rownames(all_pairwise_subset_Zscore)

ann_colors = list(
  elt2_L1_bound = c(bound = "green", not.bound = "black")
)

pheatmap(all_pairwise_subset_Zscore,
         annotation_row = my_row_anno,
         annotation_colors = ann_colors,
           cluster_cols = FALSE,
           cluster_rows = TRUE,
           show_rownames = FALSE,
           border_color = NA,
           cutree_rows = 3,
         main = "Genes with Significant DE In All Pairwise Comparisons
           Row Z Score Normalized",
         width = 6,
         height = 6)#,
```




```
        #file = "./03_plots/200406_All_DE_Genes_Elt2_Elt7_rowZscore_Bound_Annotation.pdf")
```

Add 0 or 1 binding value to matrix and use to separate bound and unbound in clustered heatmap.

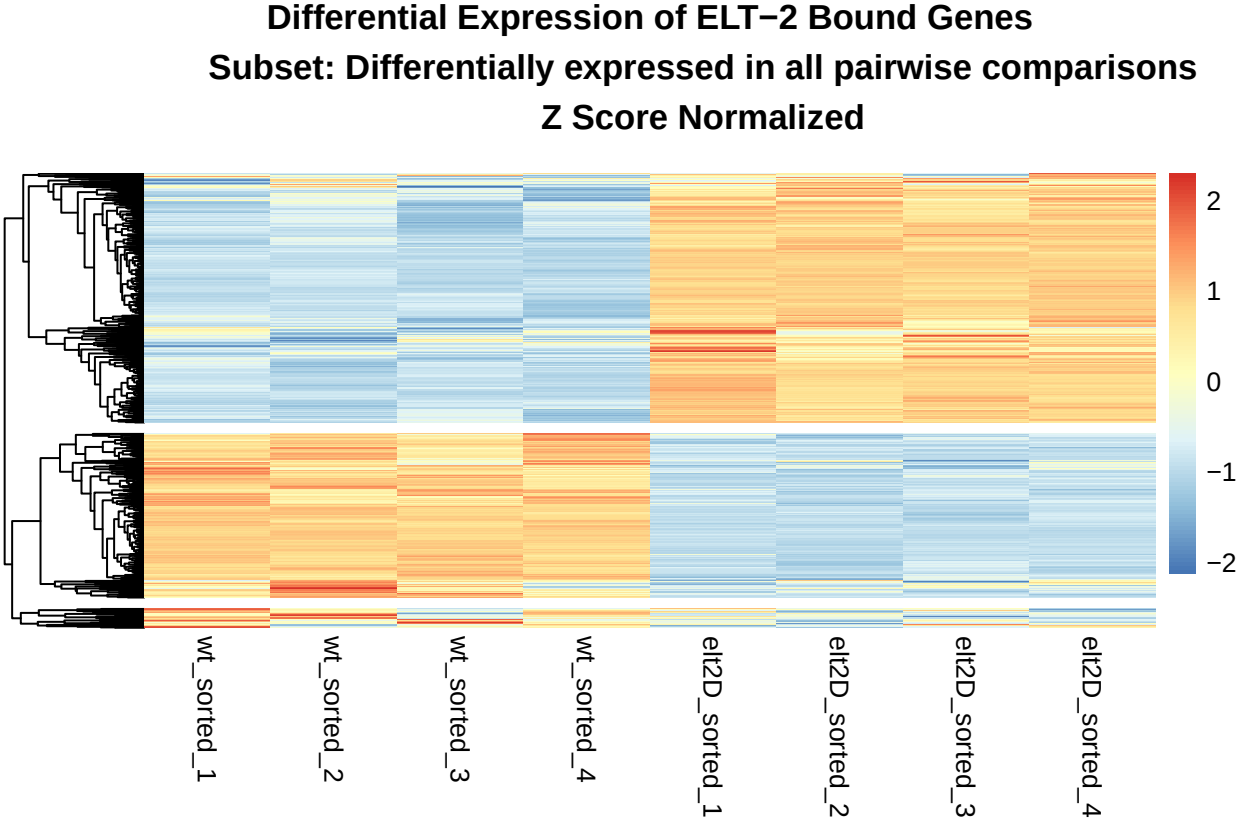
```
all_pairwise_subset_Zscore_bindStatus <- cbind(all_pairwise_subset_Zscore, elt2Bound = ifelse(test = ro

myPheatmap(all_pairwise_subset_Zscore_bindStatus,
           title = "clustered with binding",
           rowspace = 1)
```



Doesn't seem to change clustering.

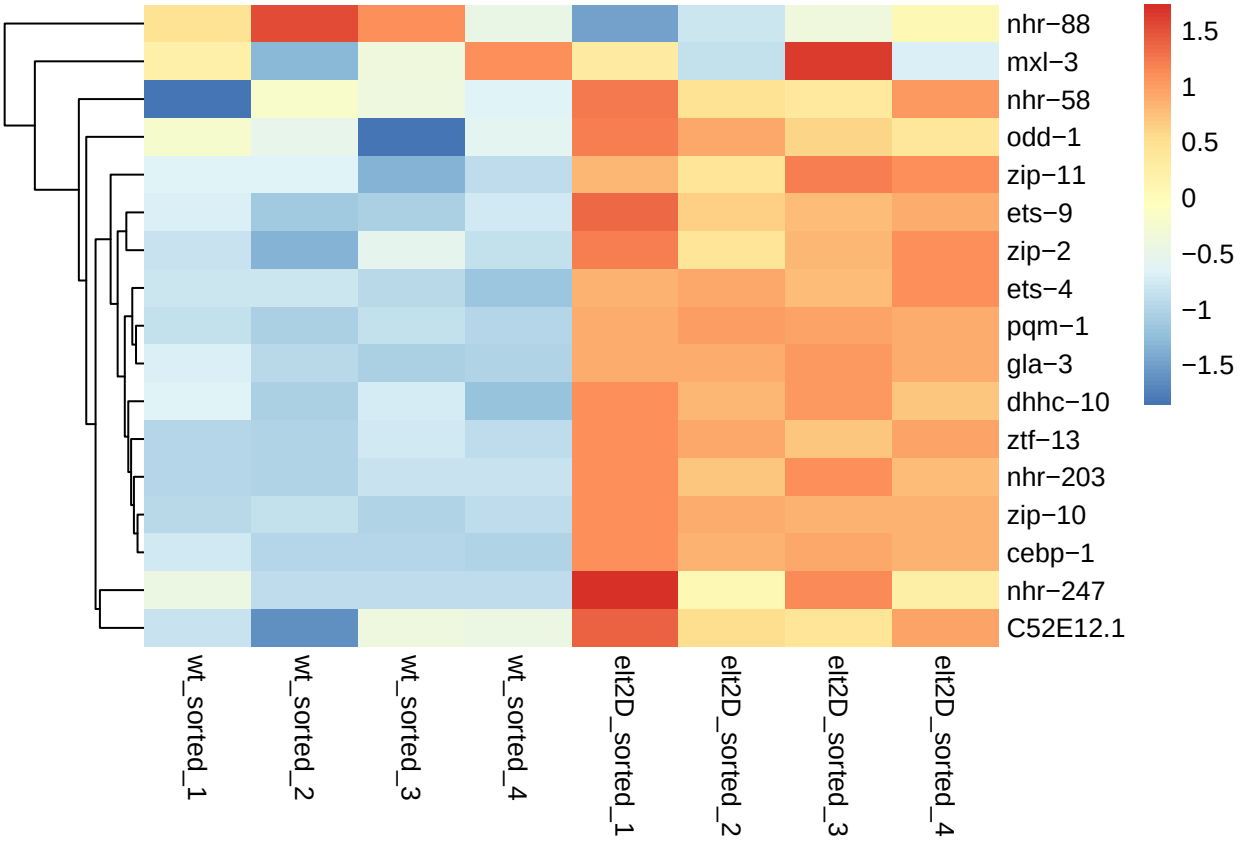
Now subset the plot for ELT-2 binding in the L1 stage.

```
elt2bound_all_pairwise_subset_Zscore <- matrix_select(all_pairwise_subset_Zscore, elt2_L1_peaks$WBGeneI

myPheatmap(elt2bound_all_pairwise_subset_Zscore,
           title = "Differential Expression of ELT-2 Bound Genes
           Subset: Differentially expressed in all pairwise comparisons
           Z Score Normalized",
           rowspace = 3)
```



Now subset for genes that are transcription factors.

```
elt2bound_all_pairwise_subset_Zscore_TF <- matrix_select(elt2bound_all_pairwise_subset_Zscore, wTF3.0$WI

elt2bound_all_pairwise_subset_Zscore_TF <- id2name(elt2bound_all_pairwise_subset_Zscore_TF)

pheatmap(elt2bound_all_pairwise_subset_Zscore_TF,
         cluster_rows = TRUE,
         cluster_cols = FALSE,
         show_rownames = TRUE,
         border_color = NA
)
```



WT, elt-2 (-) and elt-2(-);elt-7(-) Analysis

```
wt_elt2_elt7double_counts <-assay(rld)

elt2_bound_matrix <- matrix_select(wt_elt2_elt7double_counts, elt2_L1_peaks$WBGeneID)

elt2bound_rownormMatrix <- row_normalize_matrix_cutoff(
  count_matrix = elt2_bound_matrix,
  variance_cutoff =  0.5
  )

head(elt2bound_rownormMatrix)
```

```
##                wt_sorted_1 wt_sorted_2 wt_sorted_3 wt_sorted_4 elt2D_sorted_1
## WBGene00000136   1.5086891   1.4044684   1.5699383   1.4718552     -0.2107053
## WBGene00000172   0.5423177   0.6035933   0.2775824   0.4495869      0.7846391
## WBGene00000214   1.6529634   1.9191847   1.9392496   1.3315259     -0.6143219
## WBGene00000215   2.0334889   2.2653447   2.0248544   1.9586817     -1.3207025
## WBGene00000216   0.5644287   0.8648009   0.6114388   0.3241784      0.8255890
## WBGene00000218   1.4878912   1.8266621   1.6121449   1.1454803     -0.2757989
##                elt2D_sorted_2 elt2D_sorted_3 elt2D_sorted_4 elt2Delt7D_sorted_1
## WBGene00000136    -0.35445954    -0.09764439     -0.2871333           -1.715373
## WBGene00000172     0.94729559     0.66884185      0.9323174           -1.670421
## WBGene00000214    -0.26098841    -0.55458218     -0.2366091           -2.040986
## WBGene00000215    -1.10895281    -0.83159209     -0.9644632           -1.457021
```

```
## WBGene00000216     1.00299393     1.00647003      0.9358731           -2.182086
## WBGene00000218    -0.01649353    -0.14157996     -0.1102774           -2.054410
##                elt2Delt7D_sorted_2 elt2Delt7D_sorted_3
## WBGene00000136           -1.643827           -1.645808
## WBGene00000172           -1.818126           -1.717628
## WBGene00000214           -1.818103           -1.317332
## WBGene00000215           -1.417654           -1.181984
## WBGene00000216           -1.992389           -1.961298
## WBGene00000218           -1.845645           -1.627973
```

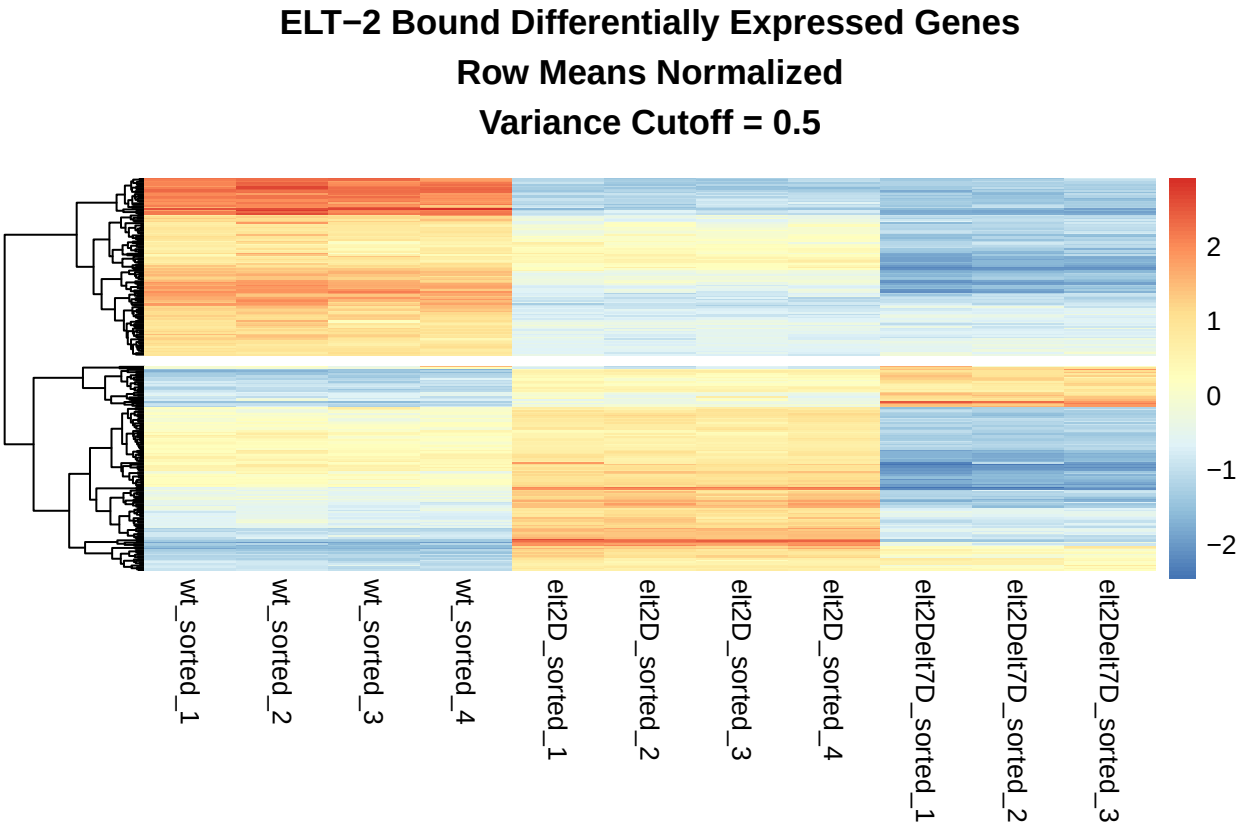
Replace the WBGeneIDs in the row name with gene names.

```
elt2bound_rowNormGeneNameMatrix <- id2name(elt2bound_rownormMatrix)

head(elt2bound_rowNormGeneNameMatrix)
```

```
##       wt_sorted_1 wt_sorted_2 wt_sorted_3 wt_sorted_4 elt2D_sorted_1
## amt-4   1.5086891   1.4044684   1.5699383   1.4718552     -0.2107053
## aqp-4   0.5423177   0.6035933   0.2775824   0.4495869      0.7846391
## asp-1   1.6529634   1.9191847   1.9392496   1.3315259     -0.6143219
## asp-2   2.0334889   2.2653447   2.0248544   1.9586817     -1.3207025
## asp-3   0.5644287   0.8648009   0.6114388   0.3241784      0.8255890
## asp-5   1.4878912   1.8266621   1.6121449   1.1454803     -0.2757989
##       elt2D_sorted_2 elt2D_sorted_3 elt2D_sorted_4 elt2Delt7D_sorted_1
## amt-4    -0.35445954    -0.09764439     -0.2871333           -1.715373
## aqp-4     0.94729559     0.66884185      0.9323174           -1.670421
## asp-1    -0.26098841    -0.55458218     -0.2366091           -2.040986
## asp-2    -1.10895281    -0.83159209     -0.9644632           -1.457021
## asp-3     1.00299393     1.00647003      0.9358731           -2.182086
## asp-5    -0.01649353    -0.14157996     -0.1102774           -2.054410
##       elt2Delt7D_sorted_2 elt2Delt7D_sorted_3
## amt-4           -1.643827           -1.645808
## aqp-4           -1.818126           -1.717628
## asp-1           -1.818103           -1.317332
## asp-2           -1.417654           -1.181984
## asp-3           -1.992389           -1.961298
## asp-5           -1.845645           -1.627973
```

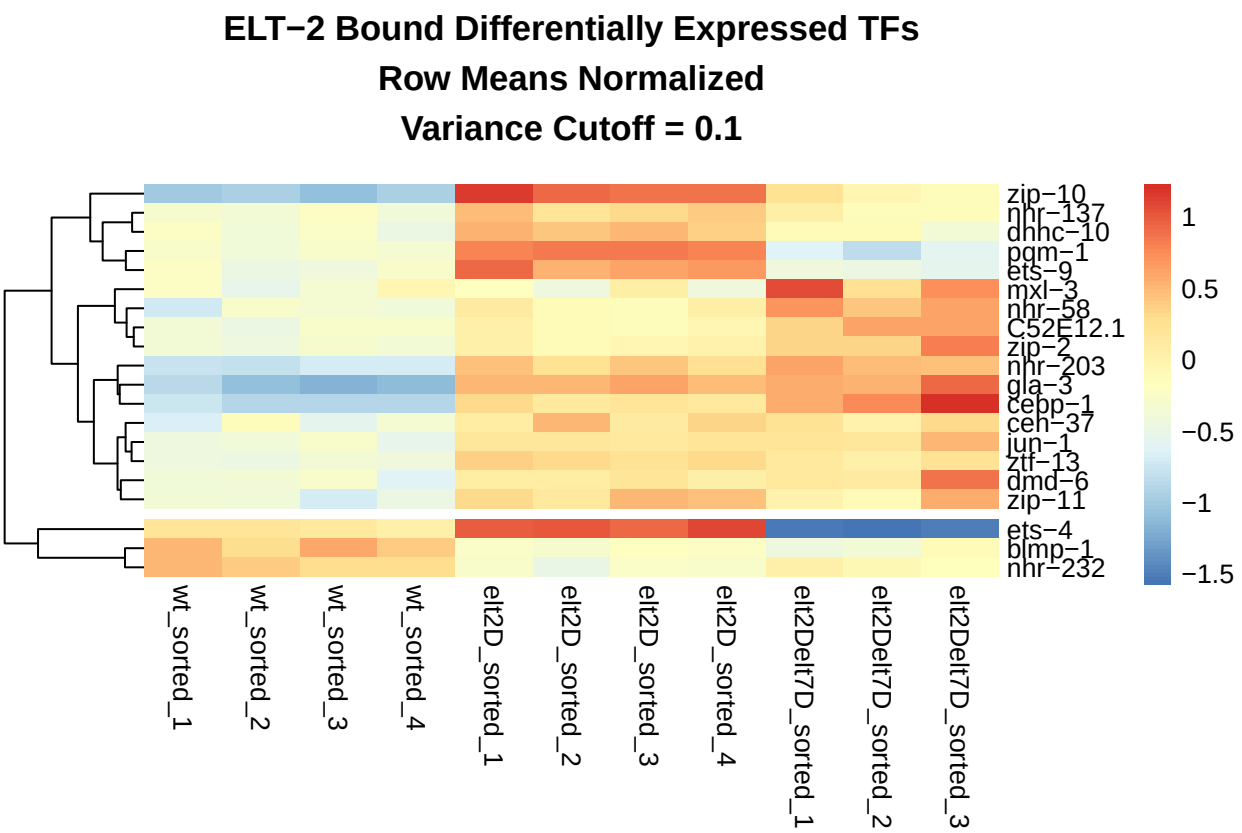
Now plot a heatmap of ELT-2 regulated genes.

```
myPheatmap(elt2bound_rowNormGeneNameMatrix, "ELT-2 Bound Differentially Expressed Genes\nRow Means Norma
```



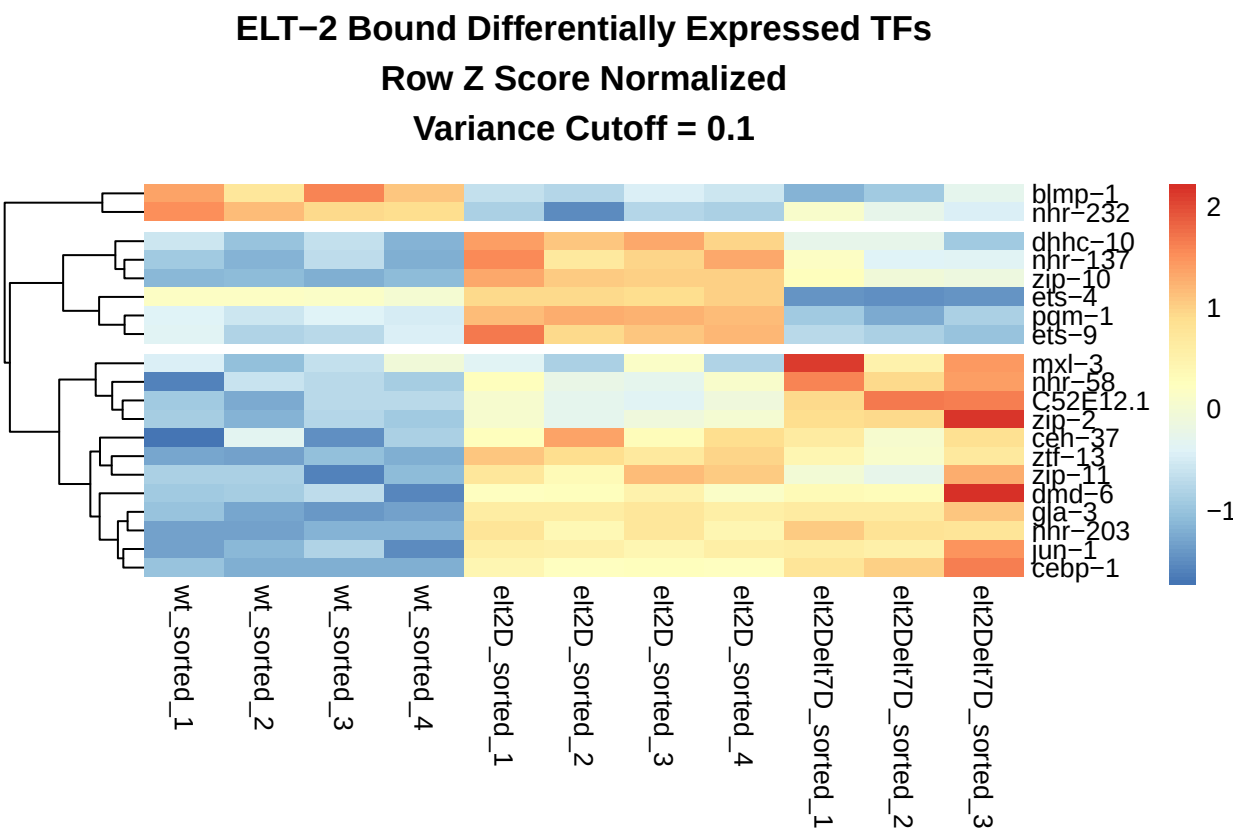
Do a similar analysis for TFs only.

```
elt2_bound_TF_matrix <- matrix_select(count_matrix = elt2_bound_matrix, gene_subset_vector = wTF3.0$WBG

elt2_bound_TF_rowNorm_matrix <- row_normalize_matrix_cutoff(
  count_matrix = elt2_bound_TF_matrix,
  variance_cutoff =  0.1
  )


elt2_bound_TF_rowNorm_matrix <- id2name(elt2_bound_TF_rowNorm_matrix)

mysmallPheatmap(elt2_bound_TF_rowNorm_matrix, "ELT-2 Bound Differentially Expressed TFs\nRow Means Norma
```



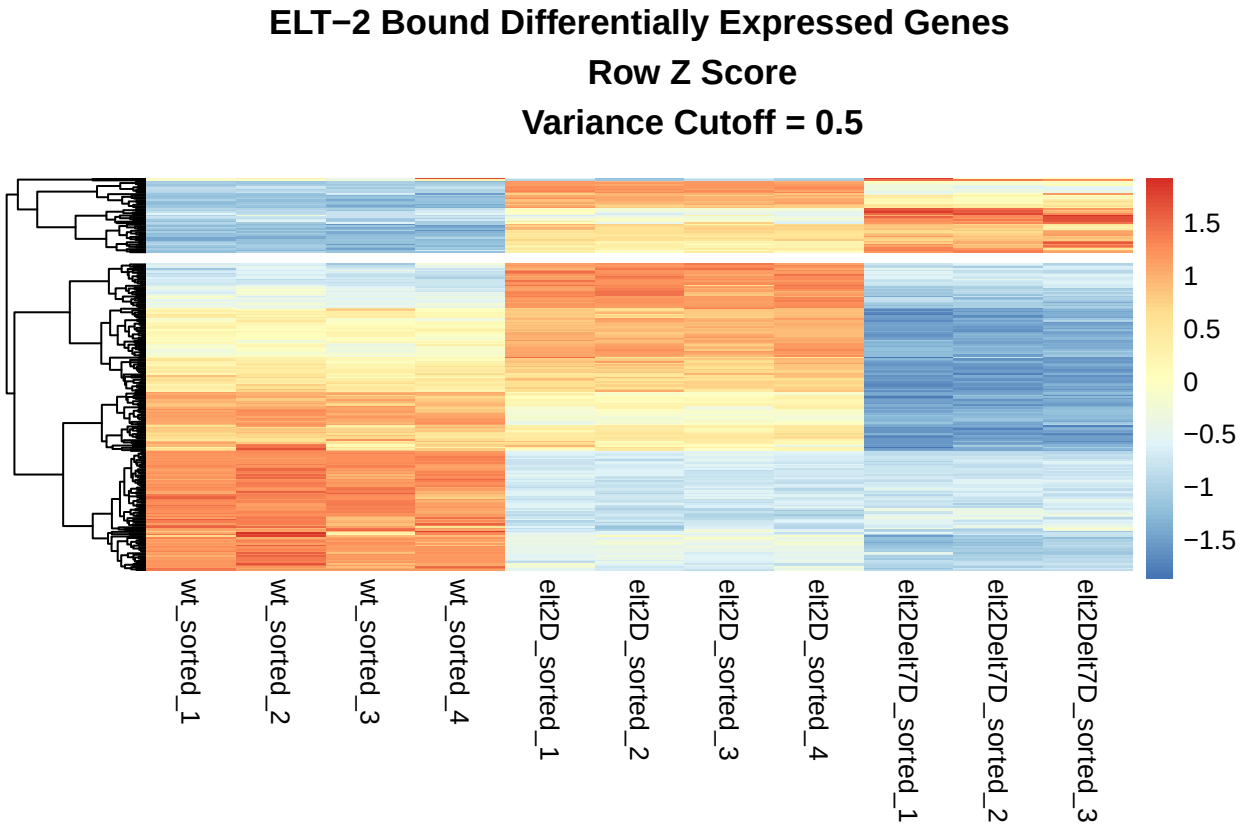
Do it for transcription factors.

```
elt2_bound_TF_zscore_matrix <- row_zscore_matrix_cutoff(
  count_matrix = elt2_bound_TF_matrix,
  variance_cutoff =  0.1
  )

elt2_bound_TF_zscore_matrix <- id2name(elt2_bound_TF_zscore_matrix)

mysmallPheatmap(elt2_bound_TF_zscore_matrix,
           title = "ELT-2 Bound Differentially Expressed TFs\nRow Z Score Normalized\nVariance Cutoff =
           rowspace = 3)
```



Do Z Score normalization for all genes.

```
elt2bound_zscore_matrix <- row_zscore_matrix_cutoff(
  count_matrix = elt2_bound_matrix,
  variance_cutoff =  0.5
  )

elt2bound_zscore_matrix <- id2name(elt2bound_zscore_matrix)

myPheatmap(elt2bound_zscore_matrix,
           title = "ELT-2 Bound Differentially Expressed Genes
           Row Z Score
           Variance Cutoff = 0.5",
           rowspace = 2)
```

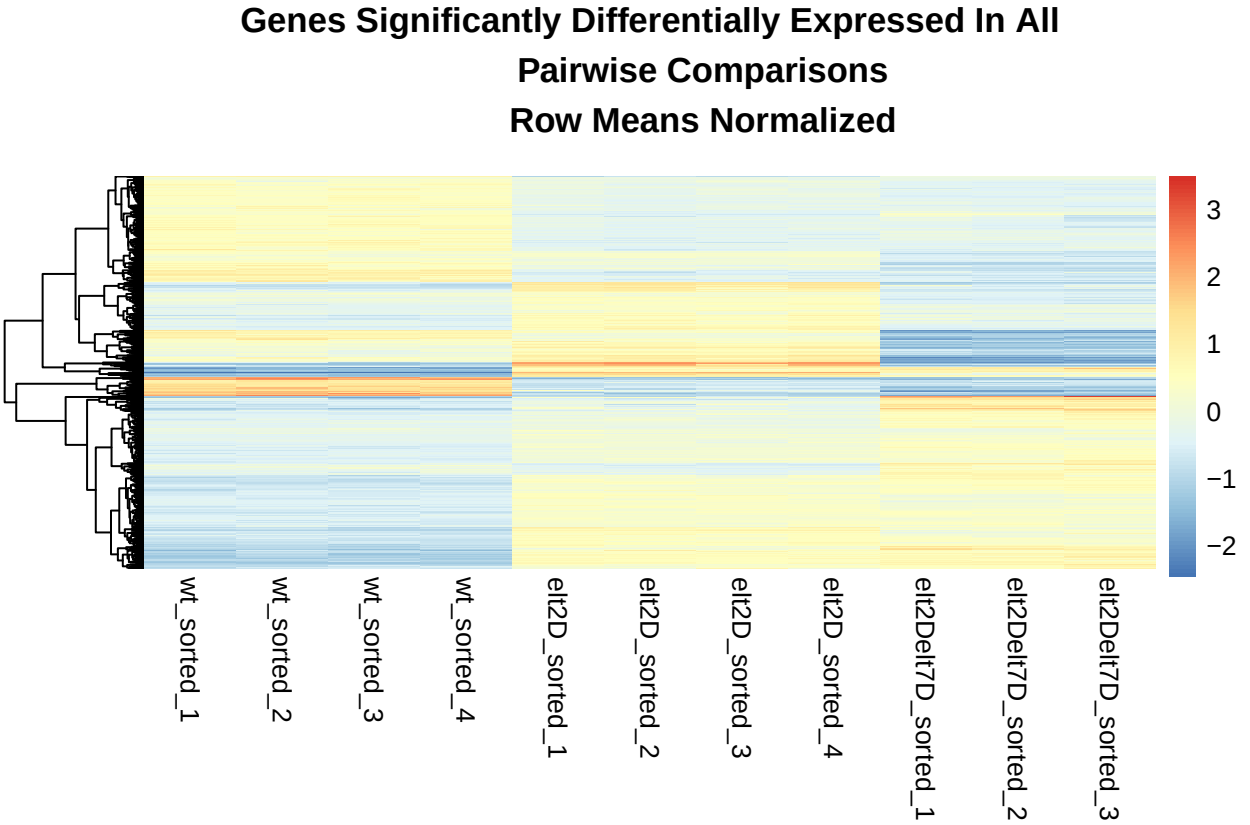


Use pairwise differential expression as regulated gene filter

Data was loaded in the elt-2 (-) section above.

Subset count matrix for presence in union of all pairwise comparisons.

```
all_pairwise_subset <- matrix_select(wt_elt2_elt7double_counts, union_elt2elt7_DE$WBGeneID)

row_normalize_matrix <- function(count_matrix){
  namevarRowNormalized <- count_matrix - rowMeans(count_matrix)
  return(namevarRowNormalized)
}

all_pairwise_subset_rownorm <- row_normalize_matrix(all_pairwise_subset)

myPheatmap(all_pairwise_subset_rownorm,
           title = "Genes Significantly Differentially Expressed In All
           Pairwise Comparisons
           Row Means Normalized",
           rowspace = 1)
```



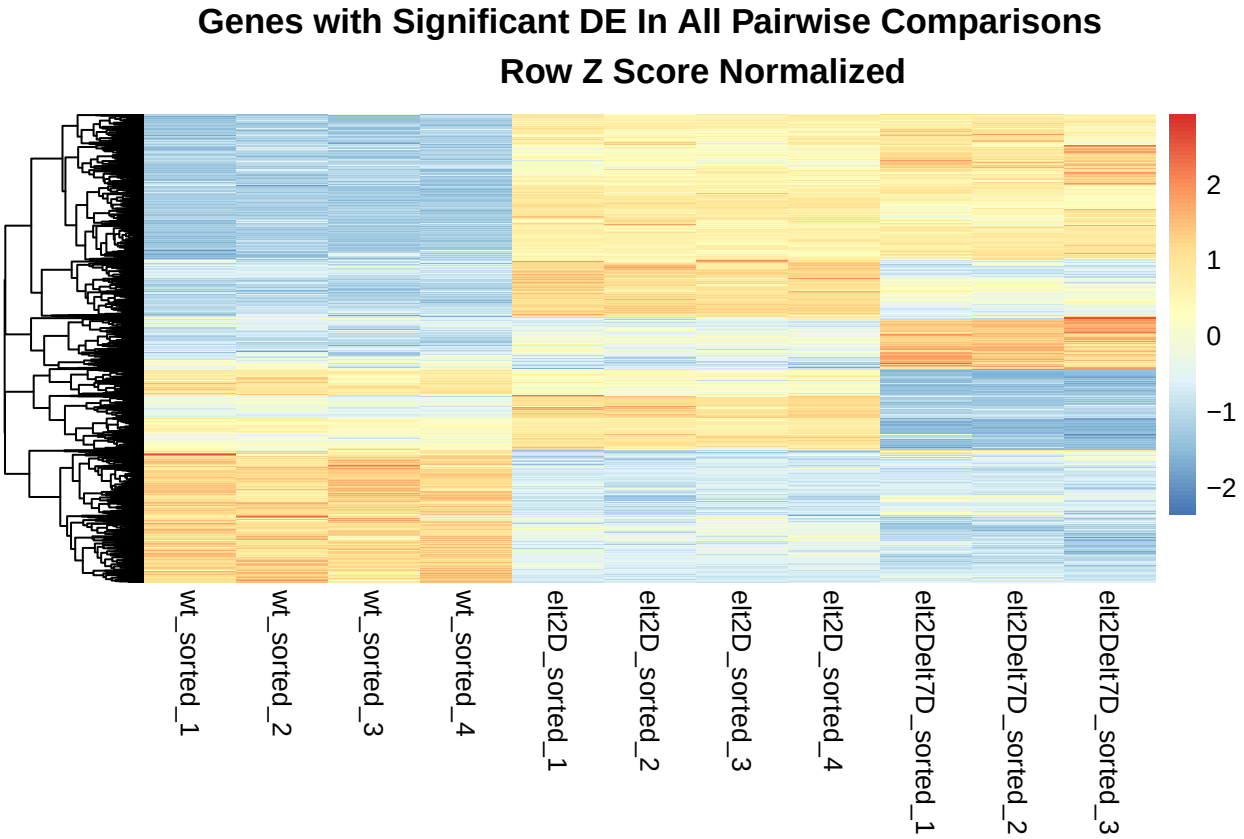
Hard to see anything useful with this.

Do the same thing but use Z score. Maybe there will be more detail.

```
all_pairwise_subset_Zscore <- row_zscore_matrix(all_pairwise_subset)

# remove columns with NA
all_pairwise_subset_Zscore <- all_pairwise_subset_Zscore[complete.cases(all_pairwise_subset_Zscore), ]

unique(is.na(all_pairwise_subset_Zscore))
```

```
##                wt_sorted_1 wt_sorted_2 wt_sorted_3 wt_sorted_4 elt2D_sorted_1
## WBGene00000007       FALSE       FALSE       FALSE       FALSE          FALSE
##                elt2D_sorted_2 elt2D_sorted_3 elt2D_sorted_4 elt2Delt7D_sorted_1
## WBGene00000007          FALSE          FALSE          FALSE               FALSE
##                elt2Delt7D_sorted_2 elt2Delt7D_sorted_3
## WBGene00000007               FALSE               FALSE
```

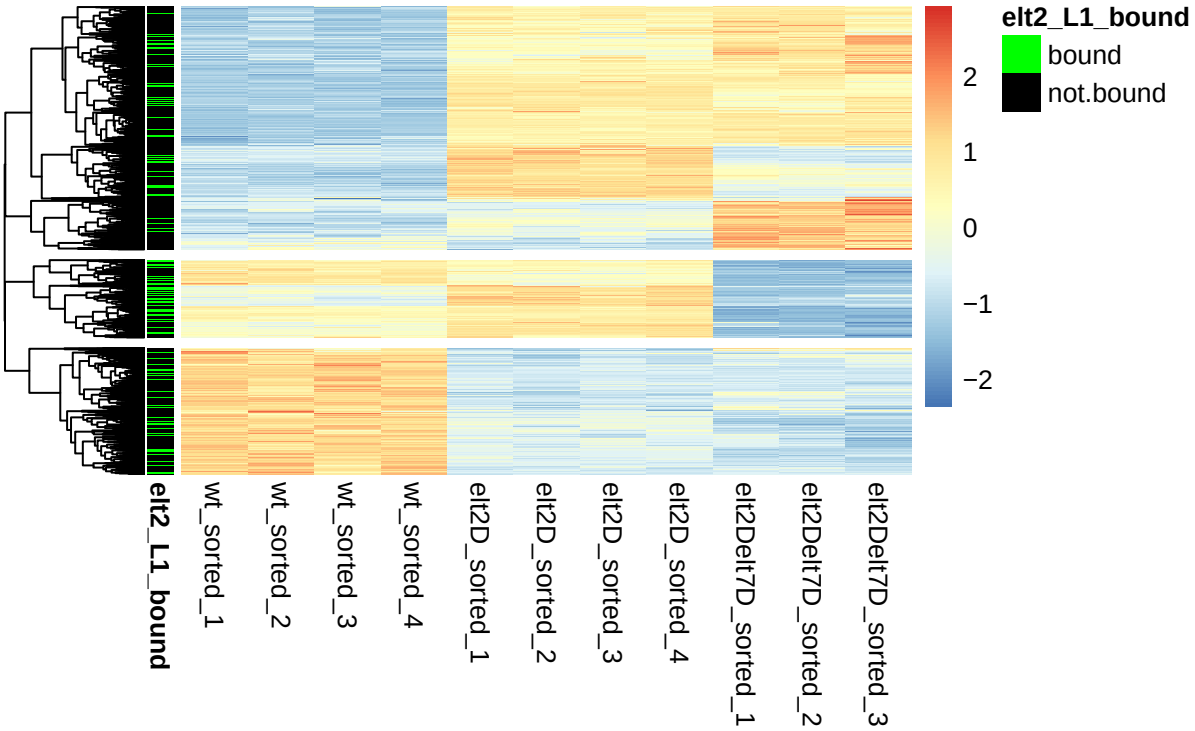
```
myPheatmap(all_pairwise_subset_Zscore,
           title = "Genes with Significant DE In All Pairwise Comparisons
           Row Z Score Normalized",
           rowspace = 1)
```



Clusters are a little more obvious.

Add annotation to side of heatmap that indicates binding of ELT-2 in L1 stage.

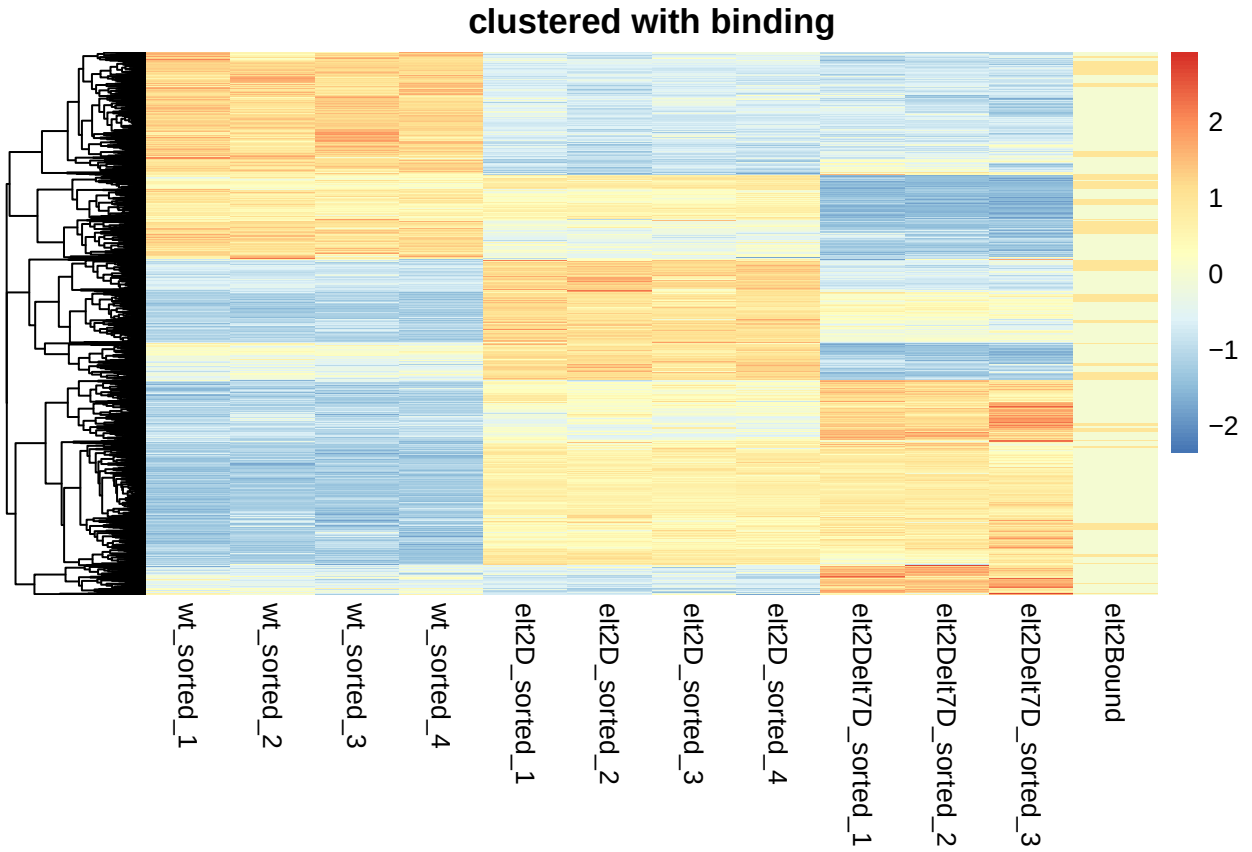
```
my_row_anno <- data.frame(elt2_L1_bound = ifelse(test = rownames(all_pairwise_subset_Zscore) %in% elt2_


rownames(my_row_anno) <- rownames(all_pairwise_subset_Zscore)

ann_colors = list(
  elt2_L1_bound = c(bound = "green", not.bound = "black")
)

pheatmap(all_pairwise_subset_Zscore,
         annotation_row = my_row_anno,
         annotation_colors = ann_colors,
           cluster_cols = FALSE,
           cluster_rows = TRUE,
           show_rownames = FALSE,
           border_color = NA,
           cutree_rows = 3,
         main = "Genes with Significant DE In All Pairwise Comparisons
           Row Z Score Normalized",
         width = 6,
         height = 6)#,
```




```
        #file = "./03_plots/200406_All_DE_Genes_Elt2_Elt7_rowZscore_Bound_Annotation.pdf")
```

Add 0 or 1 binding value to matrix and use to separate bound and unbound in clustered heatmap.

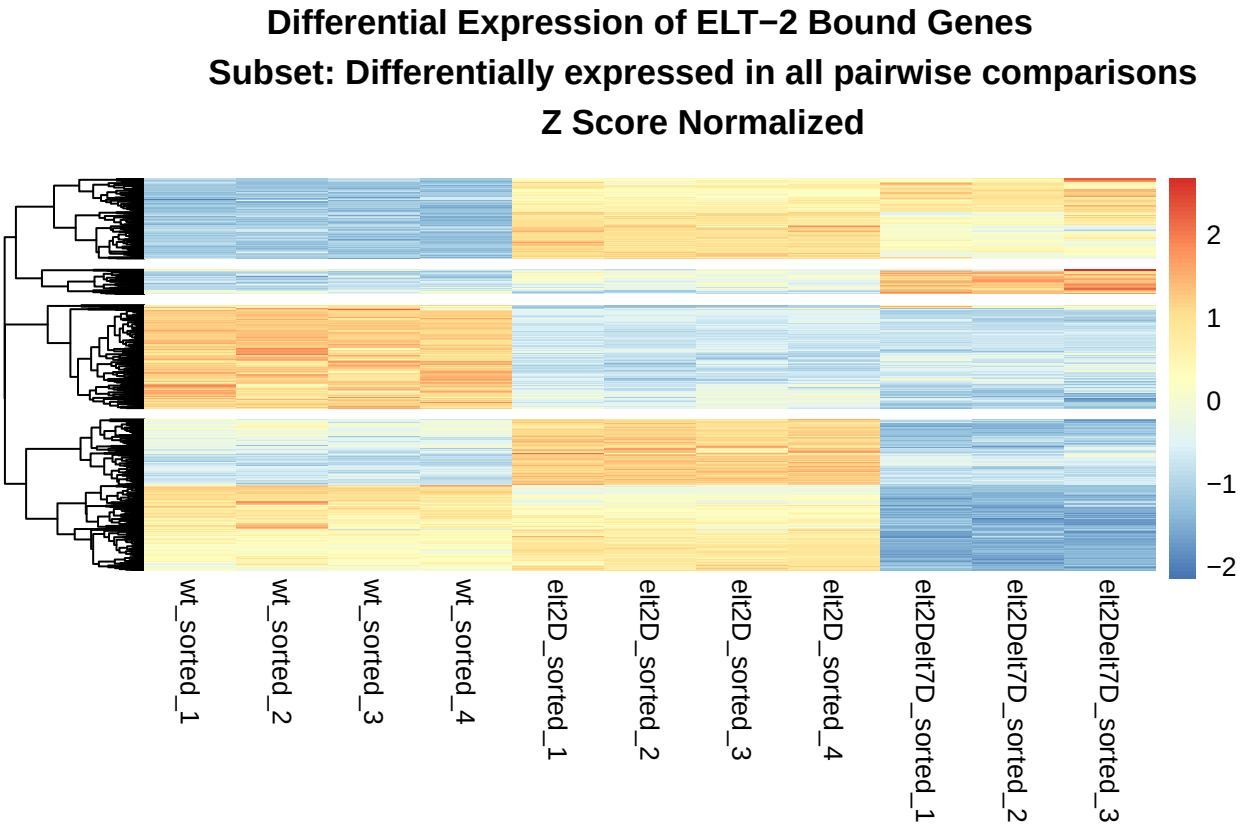
```
all_pairwise_subset_Zscore_bindStatus <- cbind(all_pairwise_subset_Zscore, elt2Bound = ifelse(test = ro

myPheatmap(all_pairwise_subset_Zscore_bindStatus,
           title = "clustered with binding",
           rowspace = 1)
```



Doesn't seem to change clustering.

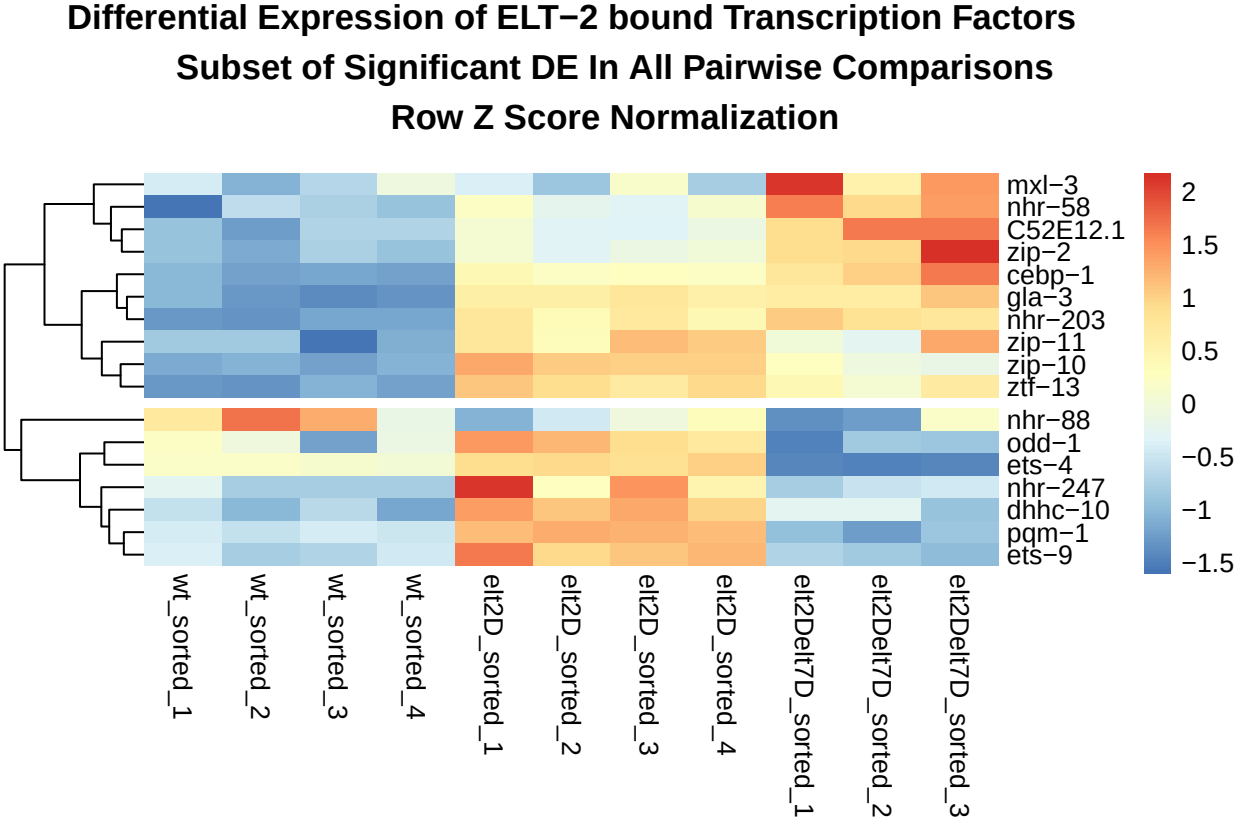
Now subset the plot for ELT-2 binding in the L1 stage.

```
elt2bound_all_pairwise_subset_Zscore <- matrix_select(all_pairwise_subset_Zscore, elt2_L1_peaks$WBGeneI

myPheatmap(elt2bound_all_pairwise_subset_Zscore,
           title = "Differential Expression of ELT-2 Bound Genes
           Subset: Differentially expressed in all pairwise comparisons
           Z Score Normalized",
           rowspace = 4)
```



Now subset for genes that are transcription factors.

```
elt2bound_all_pairwise_subset_Zscore_TF <- matrix_select(elt2bound_all_pairwise_subset_Zscore, wTF3.0$WI

elt2bound_all_pairwise_subset_Zscore_TF <- id2name(elt2bound_all_pairwise_subset_Zscore_TF)

pheatmap(elt2bound_all_pairwise_subset_Zscore_TF,
         cluster_rows = TRUE,
         cluster_cols = FALSE,
         show_rownames = TRUE,
         border_color = NA,
         cutree_rows = 2,
         main = "Differential Expression of ELT-2 bound Transcription Factors
         Subset of Significant DE In All Pairwise Comparisons
         Row Z Score Normalization"
)
```



Add intestine expressed annotation to rows. From the project RWC19 aka TF_TEAM.

Load in data.

```
spencerLEgenes <- read.table("/Users/rtpw/Documents/12_GITHUB_REPO/TF_Team/02_Data/6_Spencer_et_al_2010_
colnames(spencerLEgenes) <- str_c("spencer_LE_", colnames(spencerLEgenes))
spencer_LE_subset <- spencerLEgenes %>% select(spencer_LE_ID, spencer_LE_AveExpr, spencer_LE_adj_P_Val,

spencerL2genes <- read.table("/Users/rtpw/Documents/12_GITHUB_REPO/TF_Team/02_Data/6_Spencer_et_al_2010_
colnames(spencerL2genes) <- str_c("spencer_L2_", colnames(spencerL2genes))
spencer_L2_subset <- spencerL2genes %>% select(spencer_L2_ID, spencer_L2_AveExpr, spencer_L2_adj_P_Val,
```

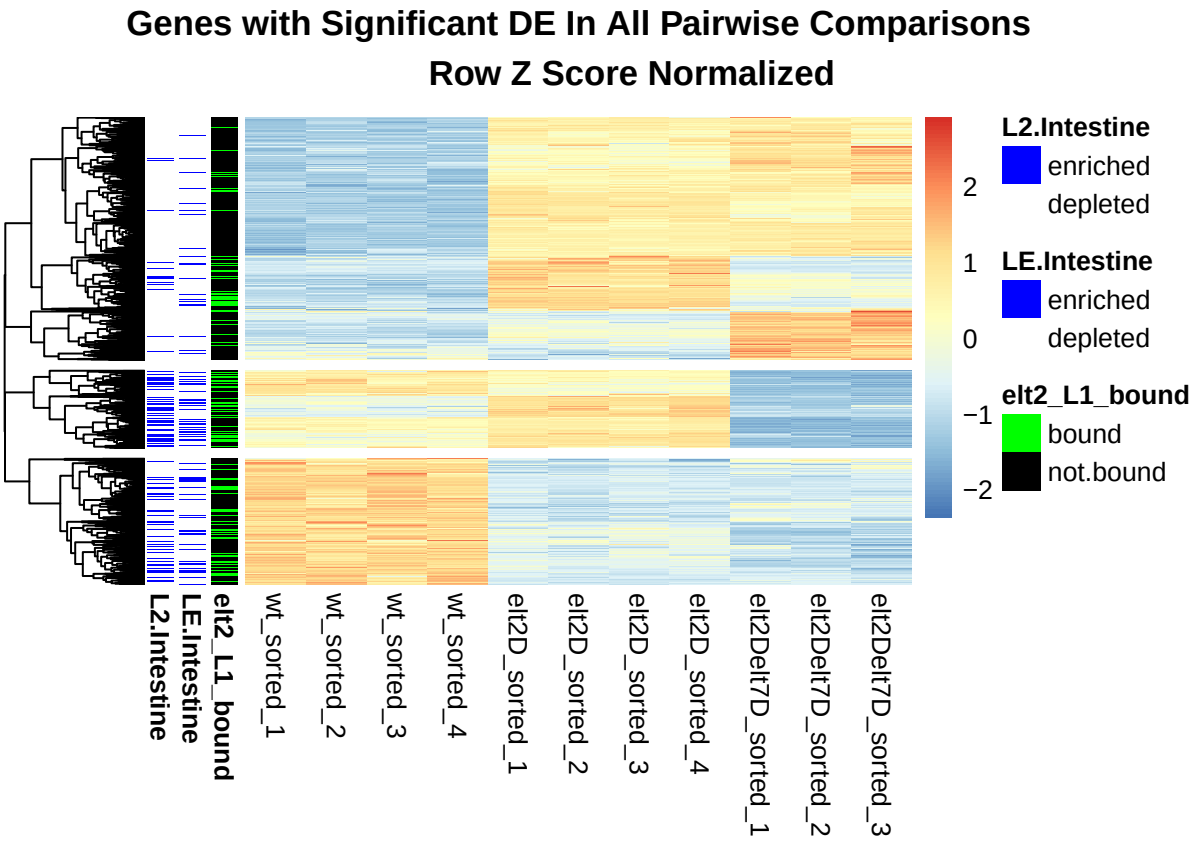
Add an annotation column for late embryo and larval stage 2 intestine expression.

```
bound_expressed_annotation <- cbind(my_row_anno,
                                    LE.Intestine = ifelse(test = rownames(all_pairwise_subset_Zscore) %
                                    L2.Intestine = ifelse(test = rownames(all_pairwise_subset_Zscore) %

bound_expressed_annotation %>% head()
```

```
##                elt2_L1_bound LE.Intestine L2.Intestine
## WBGene00000007     not.bound     depleted     enriched
## WBGene00000008         bound     depleted     depleted
## WBGene00000009     not.bound     depleted     depleted
## WBGene00000013     not.bound     depleted     depleted
## WBGene00000016     not.bound     depleted     depleted
## WBGene00000017     not.bound     depleted     depleted
```

```
bound_expressed_ann_colors <- list(
  elt2_L1_bound = c(bound = "green", not.bound = "black"),
```

```
  LE.Intestine = c(enriched = "blue", depleted = "white"),
  L2.Intestine = c(enriched = "blue", depleted = "white")
  )


pheatmap(all_pairwise_subset_Zscore,
         annotation_row = bound_expressed_annotation,
         annotation_colors = bound_expressed_ann_colors,
           cluster_cols = FALSE,
           cluster_rows = TRUE,
           show_rownames = FALSE,
           border_color = NA,
         cutree_rows = 3,
         main = "Genes with Significant DE In All Pairwise Comparisons
           Row Z Score Normalized",
         width = 6,
         height = 6)#,
```




```
         #file = "./03_plots/200409_All_DE_Genes_Elt2_Elt7_rowZscore_Bound_Expressed_Annotation.pdf")
```

Session info

Document session info.

```
sessionInfo()
```

```
## R version 3.6.3 (2020-02-29)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Sierra 10.12.5
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] readxl_1.3.1                pheatmap_1.0.12
##  [3] forcats_0.5.0               stringr_1.4.0
##  [5] dplyr_0.8.5                 purrr_0.3.3
##  [7] readr_1.3.1                 tidyr_1.0.2
##  [9] tibble_2.1.3                ggplot2_3.3.0
## [11] tidyverse_1.3.0             DESeq2_1.26.0
## [13] SummarizedExperiment_1.16.1 DelayedArray_0.12.2
## [15] BiocParallel_1.20.1         matrixStats_0.56.0
## [17] Biobase_2.46.0              GenomicRanges_1.38.0
## [19] GenomeInfoDb_1.22.0         IRanges_2.20.2
## [21] S4Vectors_0.24.3            BiocGenerics_0.32.0
## [23] biomaRt_2.42.0
##
## loaded via a namespace (and not attached):
##  [1] colorspace_1.4-1       htmlTable_1.13.3       XVector_0.26.0
##  [4] base64enc_0.1-3        fs_1.3.2               rstudioapi_0.11
##  [7] farver_2.0.3           bit64_0.9-7            fansi_0.4.1
## [10] AnnotationDbi_1.48.0   lubridate_1.7.4        xml2_1.2.5
## [13] splines_3.6.3          geneplotter_1.64.0     knitr_1.28
## [16] Formula_1.2-3          jsonlite_1.6.1         broom_0.5.5
## [19] annotate_1.64.0        cluster_2.1.0          dbplyr_1.4.2
## [22] png_0.1-7              compiler_3.6.3         httr_1.4.1
## [25] backports_1.1.5        assertthat_0.2.1       Matrix_1.2-18
## [28] cli_2.0.2              acepack_1.4.1          htmltools_0.4.0
## [31] prettyunits_1.1.1      tools_3.6.3            gtable_0.3.0
## [34] glue_1.3.2             GenomeInfoDbData_1.2.2 rappdirs_0.3.1
## [37] Rcpp_1.0.4             cellranger_1.1.0       vctrs_0.2.4
## [40] nlme_3.1-145           xfun_0.12              rvest_0.3.5
## [43] lifecycle_0.2.0        XML_3.99-0.3           zlibbioc_1.32.0
## [46] scales_1.1.0           hms_0.5.3              RColorBrewer_1.1-2
## [49] yaml_2.2.1             curl_4.3               memoise_1.1.0
## [52] gridExtra_2.3          rpart_4.1-15           latticeExtra_0.6-29
## [55] stringi_1.4.6          RSQLite_2.2.0          genefilter_1.68.0
## [58] checkmate_2.0.0        rlang_0.4.5            pkgconfig_2.0.3
## [61] bitops_1.0-6           evaluate_0.14          lattice_0.20-40
## [64] htmlwidgets_1.5.1      bit_1.1-15.2           tidyselect_1.0.0
## [67] magrittr_1.5           R6_2.4.1               generics_0.0.2
## [70] Hmisc_4.3-1            DBI_1.1.0              withr_2.1.2
```

```
## [73] pillar_1.4.3           haven_2.2.0           foreign_0.8-76
## [76] survival_3.1-11        RCurl_1.98-1.1        nnet_7.3-13
## [79] modelr_0.1.6           crayon_1.3.4          utf8_1.1.4
## [82] BiocFileCache_1.10.2   rmarkdown_2.1         jpeg_0.1-8.1
## [85] progress_1.2.2         locfit_1.5-9.1        grid_3.6.3
## [88] data.table_1.12.8      blob_1.2.1            reprex_0.3.0
## [91] digest_0.6.25          xtable_1.8-4          openssl_1.4.1
## [94] munsell_0.5.0          askpass_1.1
```