RWC23_ELT2_ChIP_Boeck_Time_Resolved_RNA

Note: Ensure BioConductor is version 3.10 or above

Install libraries

```
# fill this in
```

Note: you must load `biomaRt` before loading `tidyverse`

Load libraries

```
library(biomaRt)
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.2
```

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

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.2     v dplyr   1.0.0
## v tidyr   1.1.1     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

```
## Warning: package 'tidyr' was built under R version 4.0.2
```

```
## Warning: package 'readr' was built under R version 4.0.2
```

```
## Warning: package 'forcats' was built under R version 4.0.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks biomaRt::select()
```

```
library(ComplexHeatmap)
```

```
## Warning: package 'ComplexHeatmap' was built under R version 4.0.2
```

```
## Loading required package: grid
```

```
## ========================================
## ComplexHeatmap version 2.4.3
## 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))
## ========================================
```

Load custom functions

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

Pseudocode: - Bring in Boeck Data - Translate to WBGeneID - Filter for ELT-2 ChIP bound genes, make heatmap - Filter for intestine expressed genes (spencer data), make heatmap, add row annotation for binding cluster

Import Time-resolved RNA

```
time_resolved_rna <-
  read.delim(
    "../02_Public_Intesine_RNA/01_input/9_Boeck_et_al_2016_time-resolved_transcriptome/Unified_dcpm_per
    quote = "",
    stringsAsFactors = FALSE
  )

paramart <-
  useMart("parasite_mart",
          dataset = "wbps_gene",
          host = "https://parasite.wormbase.org",
          port = 443)

time_resolved_rna <- getBM(
  mart = paramart,
  filter = c("wormbase_gseqname"),
  value = time_resolved_rna$WormbaseName,
  attributes = c("wormbase_gseq", "wbps_gene_id", "wikigene_name")
) %>% right_join(time_resolved_rna, by = c("wormbase_gseq" = "WormbaseName"))


time_resolved_rna <- time_resolved_rna %>% drop_na(wbps_gene_id)
```

```
intestine_gene_list <-
  read_csv("../02_Public_Intesine_RNA/02_output/RWC23_Public_Intestine_RNA_Data.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   WBGeneID = col_character()
## )
```

Import wTF3.0 worm transcription factor database

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

Filter time-resolved RNA-seq based on intestine expression

```
time_resolved_rna_intestine_df <- time_resolved_rna %>%
  remove_rownames() %>%
  arrange(wbps_gene_id) %>%
  filter(wbps_gene_id %in% intestine_gene_list$WBGeneID) %>%
  select(-(emb_4cell:emb_471min),-DE,-D,-DX,-Soma,-Male,-AdultSPE9,-gonad,-LENGTH)
head(time_resolved_rna_intestine_df)
```

```
##   wormbase_gseq   wbps_gene_id wikigene_name emb_510min emb_548min emb_587min
## 1     T13A10.10 WBGene00000005         aat-4     0.1841     0.1632     0.1776
## 2       T11F9.4 WBGene00000007         aat-6     0.1513     0.1482     0.1586
## 3       ZK455.1 WBGene00000040         aco-1     2.3243     1.9498     1.8170
## 4       T25C8.2 WBGene00000067         act-5    15.2874    16.6729    16.8900
## 5       F57F5.4 WBGene00000073         add-2     0.7871     0.7277     0.6445
## 6      D2030.10 WBGene00000084         aex-1     0.1429     0.1878     0.1805
##   emb_626min emb_665min         L1          L2        L3         L4        YA
## 1     0.1677     0.1630  0.0436931  0.2184170  0.265660  0.3224440  0.408817
## 2     0.1554     0.1584  0.1681510  0.2751570  0.349014  0.3264440  0.271406
## 3     1.8299     1.9978  5.0249900  5.9824800  8.917410  2.3600200  4.554760
## 4    16.6729    18.0843 29.1548000 49.1039000 71.569300 29.3725000 34.417200
## 5     0.4962     0.3919  0.5606450  0.3947570  0.335628  0.1979400  0.387552
## 6     0.1832     0.1716  0.1049800  0.0941584  0.122310  0.0752852  0.155656
```

```
time_resolved_rna_intestine_matrix <-
  time_resolved_rna_intestine_df %>%
  select(-wormbase_gseq, -wikigene_name) %>%
  remove_rownames() %>%
  arrange(wbps_gene_id) %>%
  column_to_rownames(var = "wbps_gene_id") %>%
  as.matrix()
head(time_resolved_rna_intestine_matrix)
```

```
##                emb_510min emb_548min emb_587min emb_626min emb_665min
## WBGene00000005     0.1841     0.1632     0.1776     0.1677     0.1630
## WBGene00000007     0.1513     0.1482     0.1586     0.1554     0.1584
## WBGene00000040     2.3243     1.9498     1.8170     1.8299     1.9978
## WBGene00000067    15.2874    16.6729    16.8900    16.6729    18.0843
## WBGene00000073     0.7871     0.7277     0.6445     0.4962     0.3919
## WBGene00000084     0.1429     0.1878     0.1805     0.1832     0.1716
##                         L1         L2        L3         L4        YA
## WBGene00000005  0.0436931  0.2184170  0.265660  0.3224440  0.408817
## WBGene00000007  0.1681510  0.2751570  0.349014  0.3264440  0.271406
## WBGene00000040  5.0249900  5.9824800  8.917410  2.3600200  4.554760
## WBGene00000067 29.1548000 49.1039000 71.569300 29.3725000 34.417200
## WBGene00000073  0.5606450  0.3947570  0.335628  0.1979400  0.387552
## WBGene00000084  0.1049800  0.0941584  0.122310  0.0752852  0.155656
```

Perform row normalization

```
time_resolved_rna_intestine_matrix_scaled <-
  t(apply(unlist(time_resolved_rna_intestine_matrix), 1, scale))
colnames(time_resolved_rna_intestine_matrix_scaled) <-
  colnames(time_resolved_rna_intestine_matrix)
```

Store index of relavent genes for row annotations. Use custom function

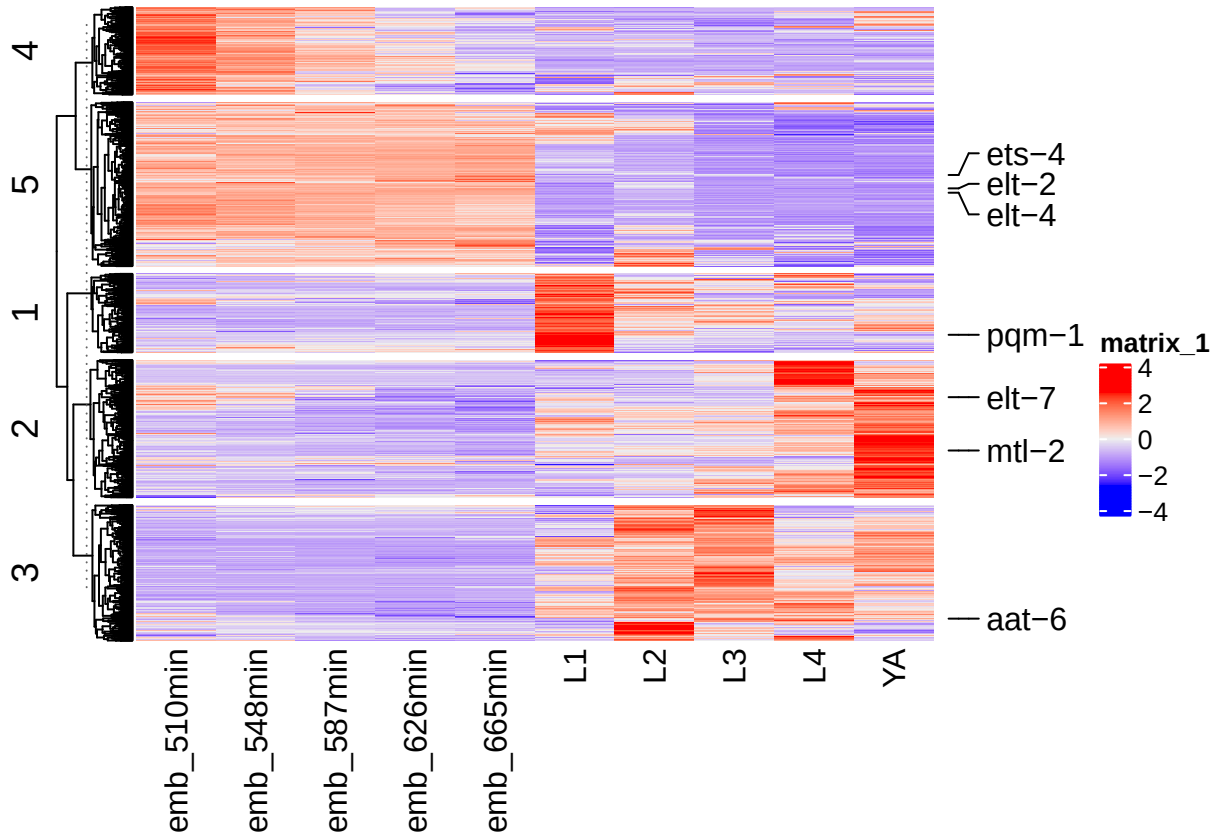
```
gene_names <-
  c("elt-2", "elt-7", "elt-4", "pqm-1", "mtl-2", "ets-4", "aat-6")
GOI_df <-
  GOI_annotate_heatmap(gene_names, time_resolved_rna_intestine_df$wikigene_name)
GOI_df
```

```
##    name index
## 1 elt-2   159
## 2 elt-7  2115
## 3 elt-4   161
## 4 pqm-1   457
## 5 mtl-2   350
## 6 ets-4  2459
## 7 aat-6     2
```

```
time_resolved_rna_intestine_df %>% filter(wikigene_name %in% GOI_df$name)
```

```
##   wormbase_gseq   wbps_gene_id wikigene_name emb_510min emb_548min emb_587min
## 1       T11F9.4 WBGene00000007         aat-6     0.1513     0.1482     0.1586
## 2       C33D3.1 WBGene00001250         elt-2     1.0771     0.7690     0.7890
## 3      C39B10.6 WBGene00001252         elt-4     0.4558     0.5370     0.5394
## 4      T08G5.10 WBGene00003474         mtl-2     0.0855     0.0869     0.0960
## 5       F40F8.7 WBGene00004096         pqm-1     1.1130     0.9787     0.9506
## 6       C18G1.2 WBGene00015981         elt-7     0.3221     0.3533     0.3047
## 7       F22A3.1 WBGene00017687         ets-4     1.6876     1.9241     2.0529
##   emb_626min emb_665min        L1       L2        L3        L4         YA
## 1     0.1554     0.1584 0.1681510 0.275157 0.3490140 0.3264440  0.2714060
## 2     0.8991     0.8958 0.2715040 0.531793 0.5176460 0.3497100  0.3836130
## 3     0.5519     0.5908 0.0731321 0.093782 0.0656069 0.0929745  0.0279745
## 4     0.1147     0.1162 2.6928700 3.669790 6.0893500 6.8993700 11.9476000
## 5     0.9540     1.0173 2.4094400 1.813160 1.0956700 1.0476000  0.8723370
## 6     0.1257     0.0675 0.3254970 0.393028 0.2046660 0.4542380  0.2526850
## 7     2.0319     2.0283 0.4815530 1.165750 1.3026800 0.4953370  0.5412020
```

```
Boeck_intestine_RNA <-
  Heatmap(
    time_resolved_rna_intestine_matrix_scaled,
    cluster_columns = FALSE,
    show_row_names = FALSE,
    row_km = 5
  ) +
  rowAnnotation(foo = anno_mark(GOI_df$index, labels = GOI_df$name))
Boeck_intestine_RNA
```




```
# pdf(file = "./03_plots/200915_Boeck_RNA_Intestine.pdf", width = 7, height = 7)
# Boeck_intestine_RNA
# dev.off()
```

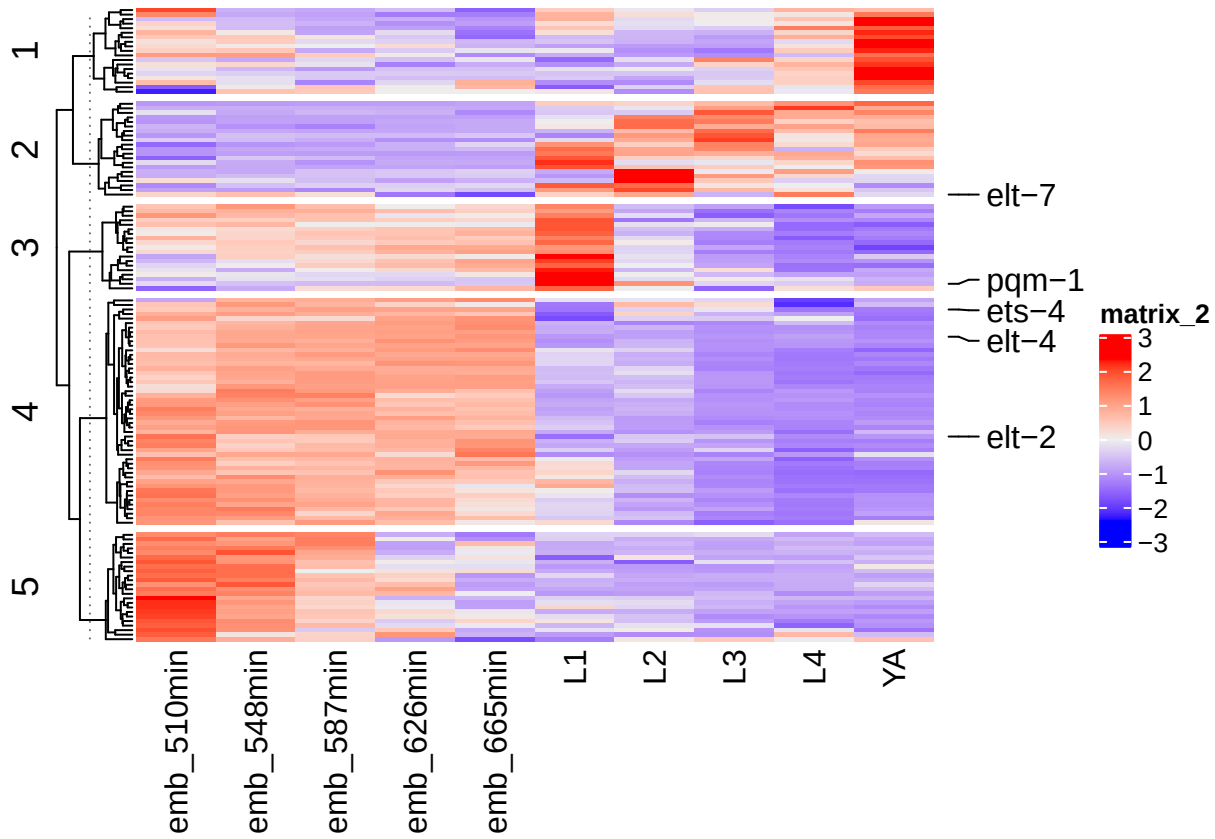
Filter heatmap for only transcription factors. This is very ugly, fix later.

```
time_resolved_rna_intestine_matrix_scaled_TFONLY <-
  matrix_select(time_resolved_rna_intestine_matrix_scaled, wTF3.0$WBGeneID)

tf_GOI_df <-
  GOI_df %>%
  left_join(time_resolved_rna_intestine_df, by = c("name" = "wikigene_name")) %>%
  select(name:wbps_gene_id, -index) %>% filter(wbps_gene_id %in% wTF3.0$WBGeneID)
tf_GOI_df
```

```
##    name wormbase_gseq   wbps_gene_id
## 1 elt-2       C33D3.1 WBGene00001250
## 2 elt-7       C18G1.2 WBGene00015981
## 3 elt-4      C39B10.6 WBGene00001252
## 4 pqm-1       F40F8.7 WBGene00004096
## 5 ets-4       F22A3.1 WBGene00017687
```

```
tf_GOI_df <-
  GOI_annotate_heatmap(
    tf_GOI_df$wbps_gene_id,
    rownames(time_resolved_rna_intestine_matrix_scaled_TFONLY)
  ) %>% full_join(tf_GOI_df, by = c("name" = "wbps_gene_id"))
```

```
Heatmap(
  time_resolved_rna_intestine_matrix_scaled_TFONLY,
  cluster_columns = FALSE,
  show_row_names = FALSE,
  row_km = 5
) +
  rowAnnotation(foo = anno_mark(at = tf_GOI_df$index,
                                labels = tf_GOI_df$name.y))
```



Import ELT-2 ChIP-seq binding data

```
chip_df <-
  read_csv(file = "../01_ChIPseq_RNAseq_Integration/01_input/200719_annotatedPeaks.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   name = col_character(),
##   cluster.description = col_character(),
##   peak = col_character(),
##   WBGeneID = col_character(),
##   feature_strand = col_character(),
##   insideFeature = col_character(),
##   fromOverlappingOrNearest = col_character()
```

```
## )
## i Use `spec()` for the full column specifications.
```

```
head(chip_df)
```

```
## # A tibble: 6 x 32
##    LE_1  LE_2  L1_1  L1_2  L3_1  L3_2 LE_IDR L1_IDR L3_IDR summit_agreement
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl>  <dbl>            <dbl>
## 1  1.93  1.60  4.25  3.77  4.88  5.01      0      1      1             27.4
## 2  2.11  1.94  4.05  4.46  4.95  5.94      0      1      1             12.4
## 3  1.22  1.53  2.61  2.85  2.45  2.86      0      0      1            137
## 4  1.81  1.42  2.74  3.28  4.18  4.49      0      0      1              2.5
## 5  2.22  2.17  2.24  2.13  4.02  4.10      1      1      1             10
## 6  1.89  2.10  3.43  2.85  3.42  3.53      0      0      1            124.
## # ... with 22 more variables: k4cluster <dbl>, k11cluster <dbl>,
## #   k4weights <dbl>, k11weights <dbl>, LE_nonNormed <dbl>, L1_nonNormed <dbl>,
## #   L3_nonNormed <dbl>, LE_std <dbl>, L1_std <dbl>, L3_std <dbl>, name <chr>,
## #   cluster.description <chr>, variance <dbl>, peak <chr>, WBGeneID <chr>,
## #   start_position <dbl>, end_position <dbl>, feature_strand <chr>,
## #   insideFeature <chr>, distancetoFeature <dbl>, shortestDistance <dbl>,
## #   fromOverlappingOrNearest <chr>
```

Subset ELT-2 ChIP with literature Intestine Expression

Do this earlier in the code to have k4labels stored in the time_resolved_rna dataframe and subsequent subsetting

```
chip_rna_df <- chip_df %>%
  select(name, cluster.description, WBGeneID)  %>%
  right_join(time_resolved_rna_intestine_df,
             by = c("WBGeneID" = "wbps_gene_id")) %>%
  replace_na(list("cluster.description" = "Not_Bound", "name" = "Not_Bound"))

chip_rna_df$cluster.description <-
  factor(
    chip_rna_df$cluster.description,
    levels = c(
      "Embryo_Specific",
      "Larval",
      "Increasing",
      "L3_High",
      "Not_Changing",
      "Not_Bound"
    )
  )
```

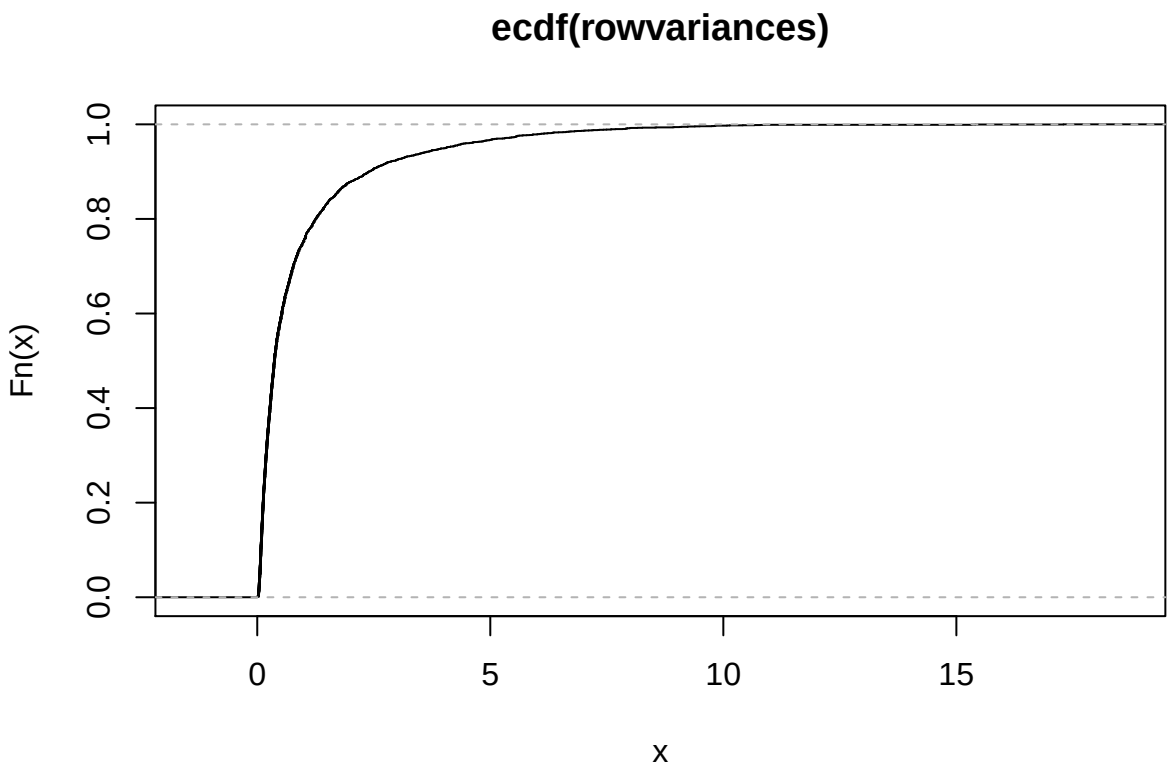
Subset heatmap based on ELT-2 binding pattern

```
#### Handle duplicate rows created by 1:many gene:peak mapping

# match will return the first index of each non-redundant gene
nr_gene_name_ixs = match(unique(chip_rna_df$wikigene_name),chip_rna_df$wikigene_name)
#length(nr_gene_name_ixs)
```

```
#[1] 3286

chip_rna_df = chip_rna_df[nr_gene_name_ixs,]


chip_rna_matrix <-
  chip_rna_df %>% select(emb_510min:YA) %>% as.matrix()
  #chip_rna_df %>% select(emb_548min,emb_626min,L1,L2,L3,L4) %>% as.matrix()


#### Handle 0's and take the log
# 1. Just replace 0's as NAs so we can apply log(). Alternatively, we could do log(x + .01), but there
chip_rna_matrix_na = chip_rna_matrix;
chip_rna_matrix_na[0 == chip_rna_matrix_na] <- NA
# 2. Apply log()
chip_rna_matrix_log = log( chip_rna_matrix_na )
# 3. Do variances row-wise, make sure to set na.rm=T
rowvariances = apply(chip_rna_matrix_log, 1, var, na.rm=T)

range(rowvariances) # no NaNs
```

```
## [1]  0.001551605 17.297836019
```

```
# 4. Plot distribution of row variances of the log data...
# You can do hist with 10,100,1000 breaks, and there will always be
# a dominant spike all the way on the lowest value. This is because the data elicit no natural lowest b
plot(ecdf(rowvariances)) # no "steps" anywhere, just a smooth curve
```




```
# therefore, we will choose to exclude the lowest 5% of the rows by their variance

changing = rowvariances > 1  # quantile(rowvariances,.1) # or .05
chip_rna_matrix = chip_rna_matrix_log
rownames(chip_rna_matrix) <- chip_rna_df$wikigene_name
chip_rna_matrix_scaled <- row_scale(chip_rna_matrix) # calls base::scale() via RWC23_Functions.R
```

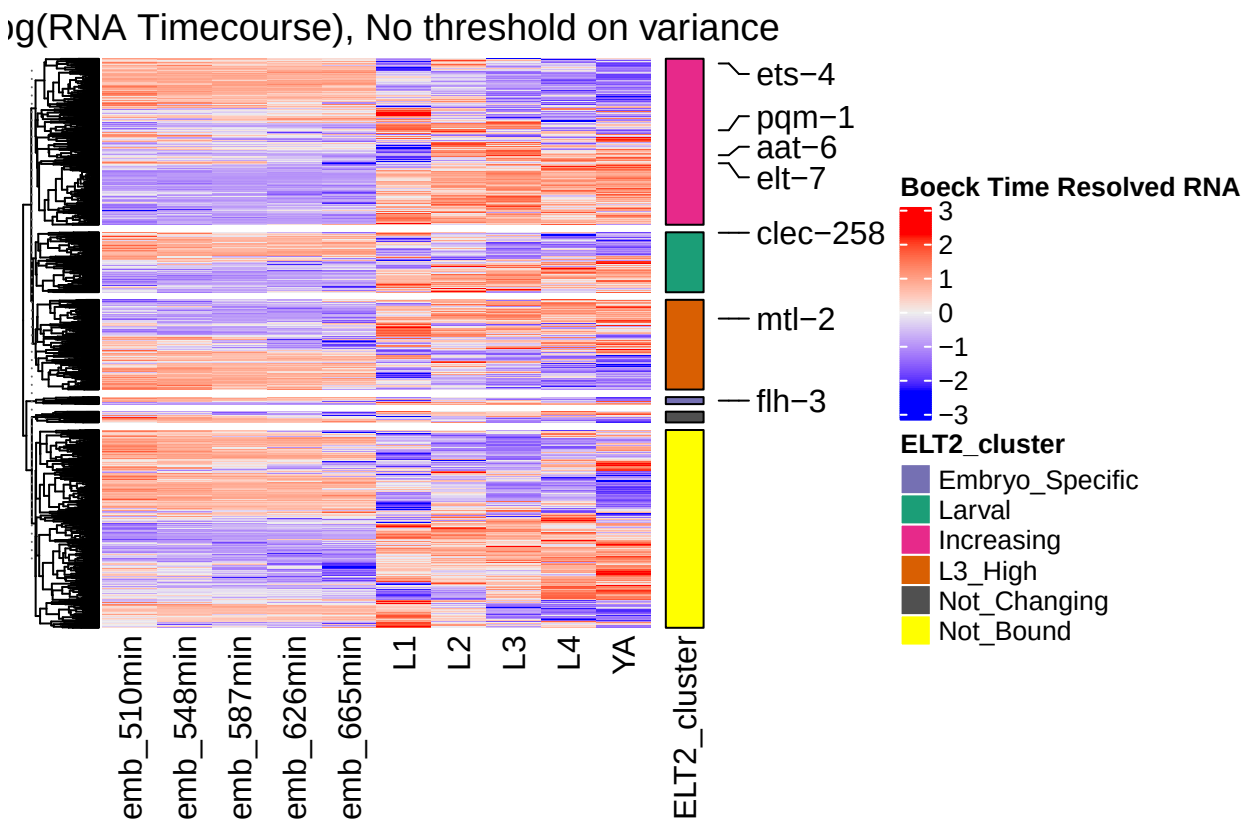
```
for (name in gene_names) {
  index <- which(rownames(chip_rna_matrix_scaled) == name)
  for (i in 1:length(index)) {
    print(c(name, index[i]))
  }
}
```

```
## [1] "elt-2" "2013"
## [1] "elt-7" "1085"
## [1] "elt-4" "2014"
## [1] "pqm-1" "414"
## [1] "mtl-2" "1320"
## [1] "ets-4" "1523"
## [1] "aat-6" "1239"
```

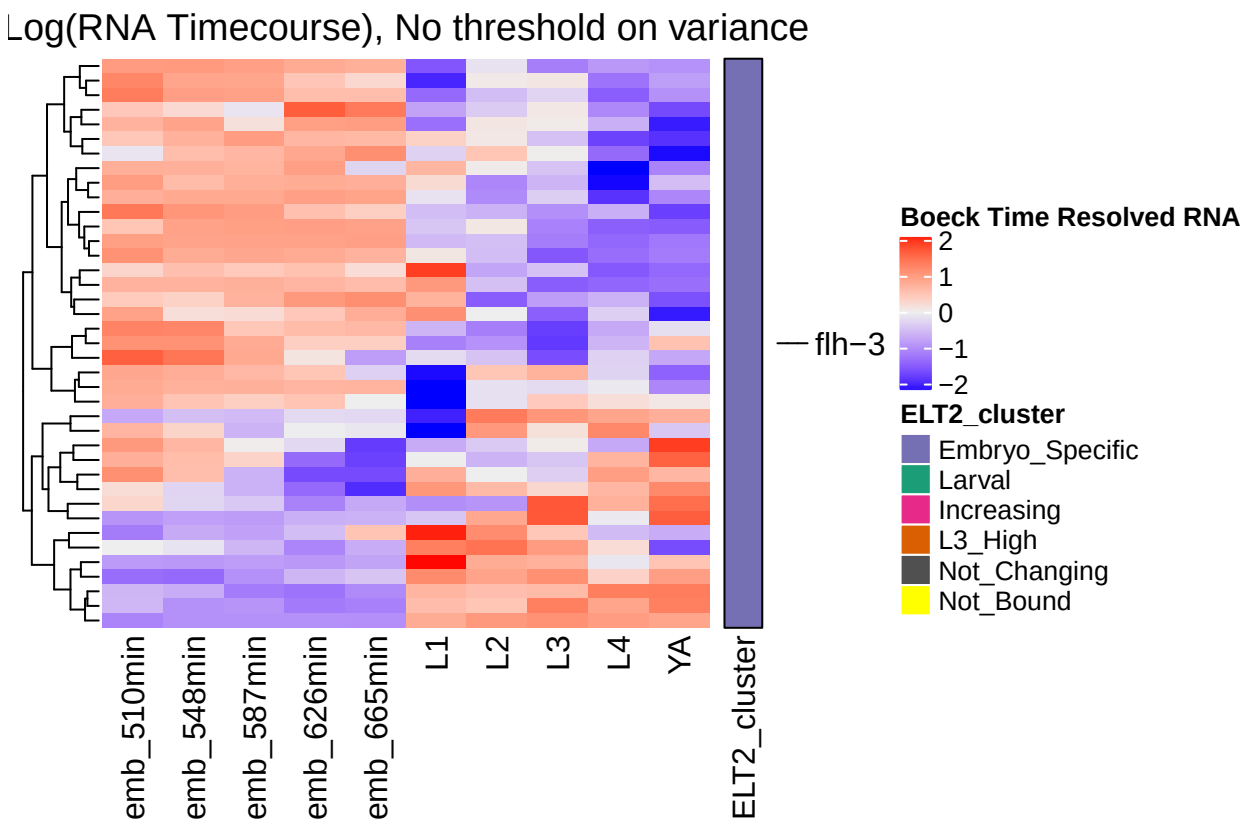
```
BoeckRNA_ELT2_chip_Heatmap <- function(subsetrows, label) {

  ix=which(rownames(chip_rna_matrix_scaled)[subsetrows] %in% gene_names)
  chip_GOI_df = data.frame(name=rownames(chip_rna_matrix_scaled)[subsetrows][ix], index =ix )
```

```
  BoeckRNA_ELT2_chip <- Heatmap(
  chip_rna_matrix_scaled[subsetrows,],
  name = "Boeck Time Resolved RNA",
  row_split = chip_rna_df$cluster.description[subsetrows],
  column_title = label,
  row_title = NULL,
  cluster_columns = FALSE
) +
  rowAnnotation(
    ELT2_cluster = chip_rna_df$cluster.description[subsetrows],
    col = list(
      ELT2_cluster = c(
        "Embryo_Specific" = "#7570B3",
        "Larval" = "#1B9E77",
        "Increasing" = "#E7298A",
        "L3_High" = "#D95F02",
        "Not_Changing" = "#505050",
        "Not_Bound" = "yellow"
      )
    ),
    border = TRUE
  ) +  rowAnnotation(foo = anno_mark(at = chip_GOI_df$index,
                                     labels = chip_GOI_df$name))
BoeckRNA_ELT2_chip
}
```

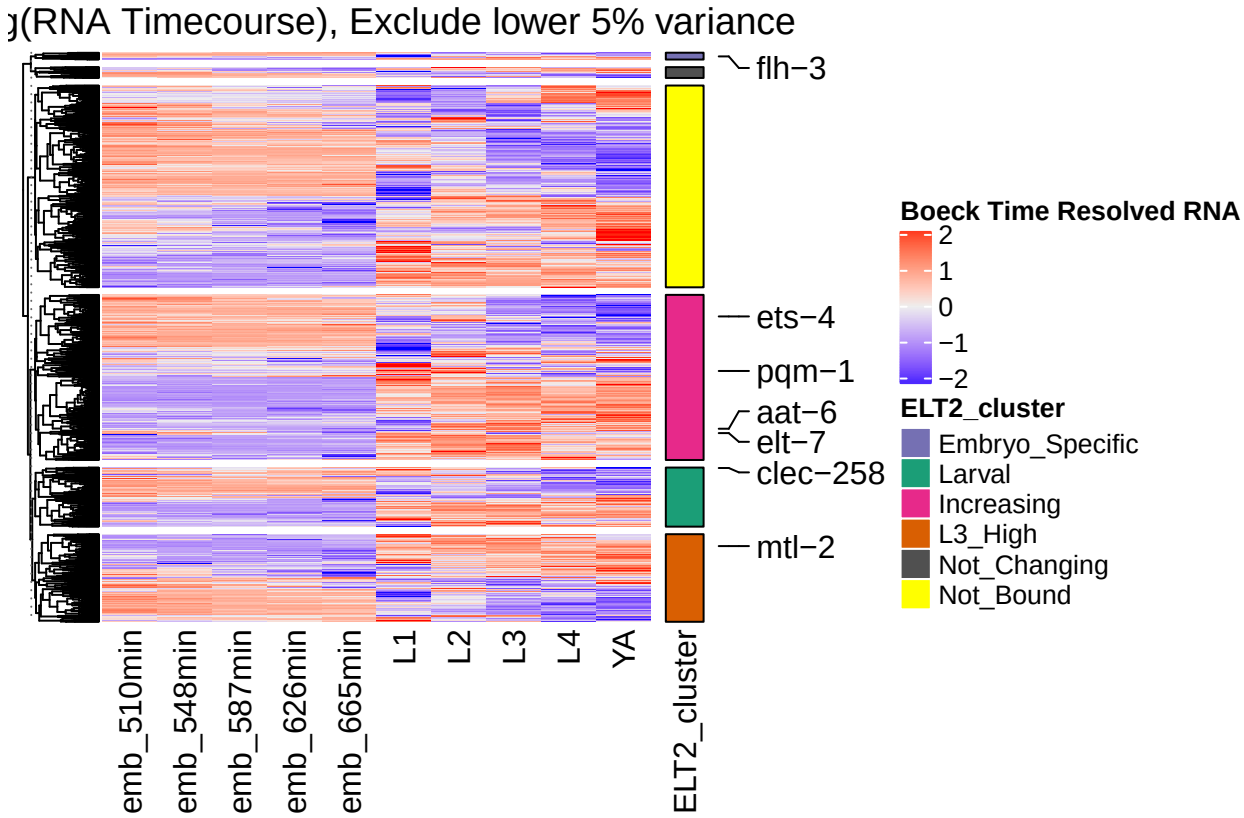
```
gene_names <-
  c("flh-3", "elt-7", "clec-258", "pqm-1", "mtl-2", "ets-4", "aat-6")

real = apply(chip_rna_matrix_scaled, 1, function(x) { ! any(is.na(x)) }) # NA's introduced by log trans
embryo_specific = chip_rna_df$cluster.description == "Embryo_Specific"
larval = chip_rna_df$cluster.description == "Larval"

#changing = rowvariances > 0.1355294 # .05 thresh from chipseq

### ALL ###
changing = rowvariances > 0
BoeckRNA_ELT2_chip_Heatmap(real & changing, "Log(RNA Timecourse), No threshold on variance")
```



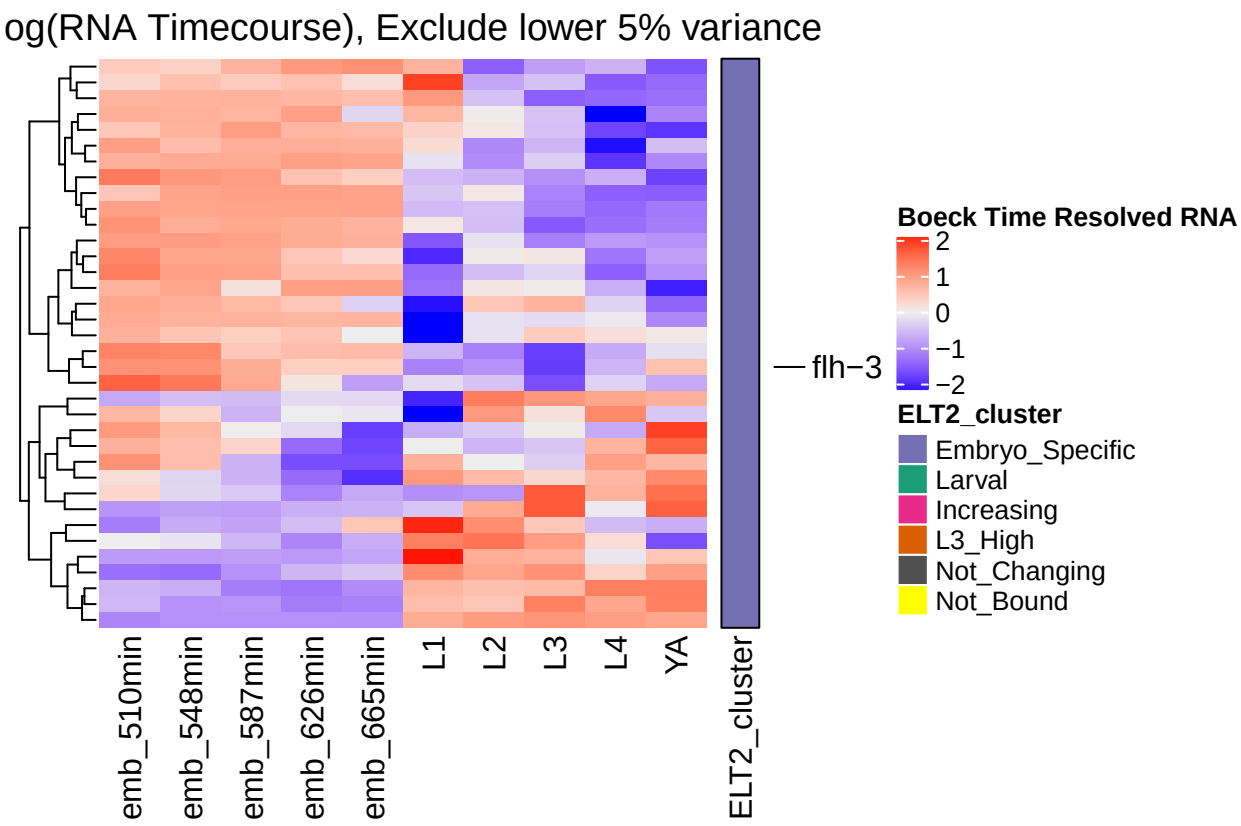

```
BoeckRNA_ELT2_chip_Heatmap(real & changing & embryo_specific, "Log(RNA Timecourse), No threshold on vari
```



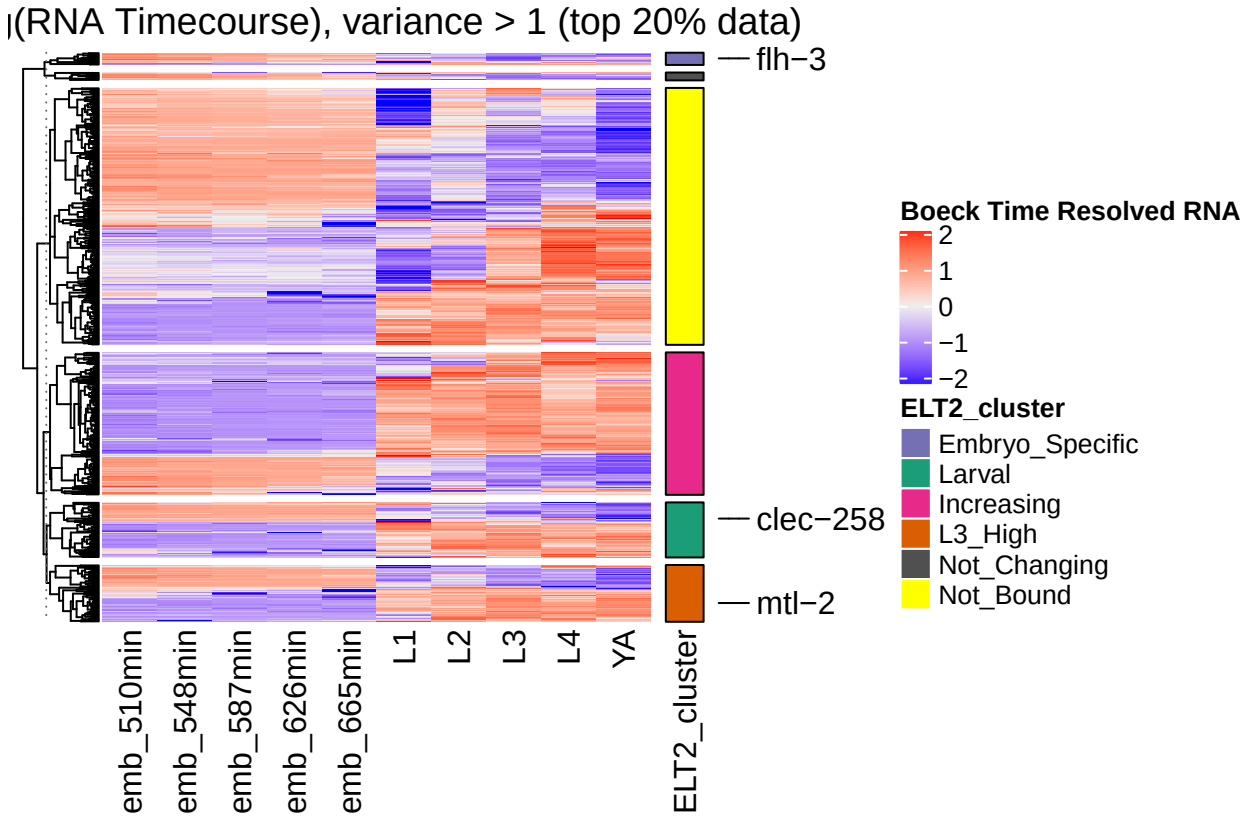

```
### Exclude the least changing, lower 5% of variance ###
changing = rowvariances > quantile(rowvariances,.05)
BoeckRNA_ELT2_chip_Heatmap(real & changing, "Log(RNA Timecourse), Exclude lower 5% variance")
```



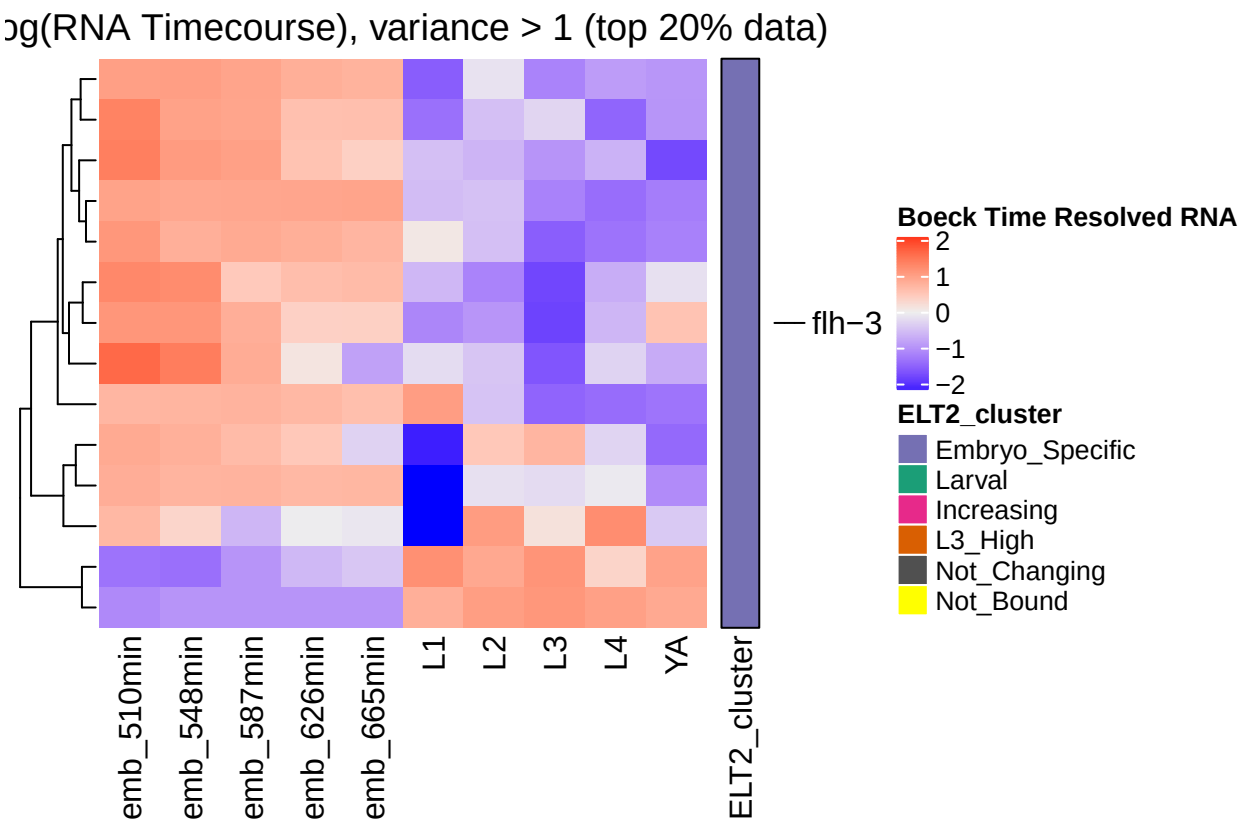

```
BoeckRNA_ELT2_chip_Heatmap(real & changing & embryo_specific, "Log(RNA Timecourse), Exclude lower 5% va
```



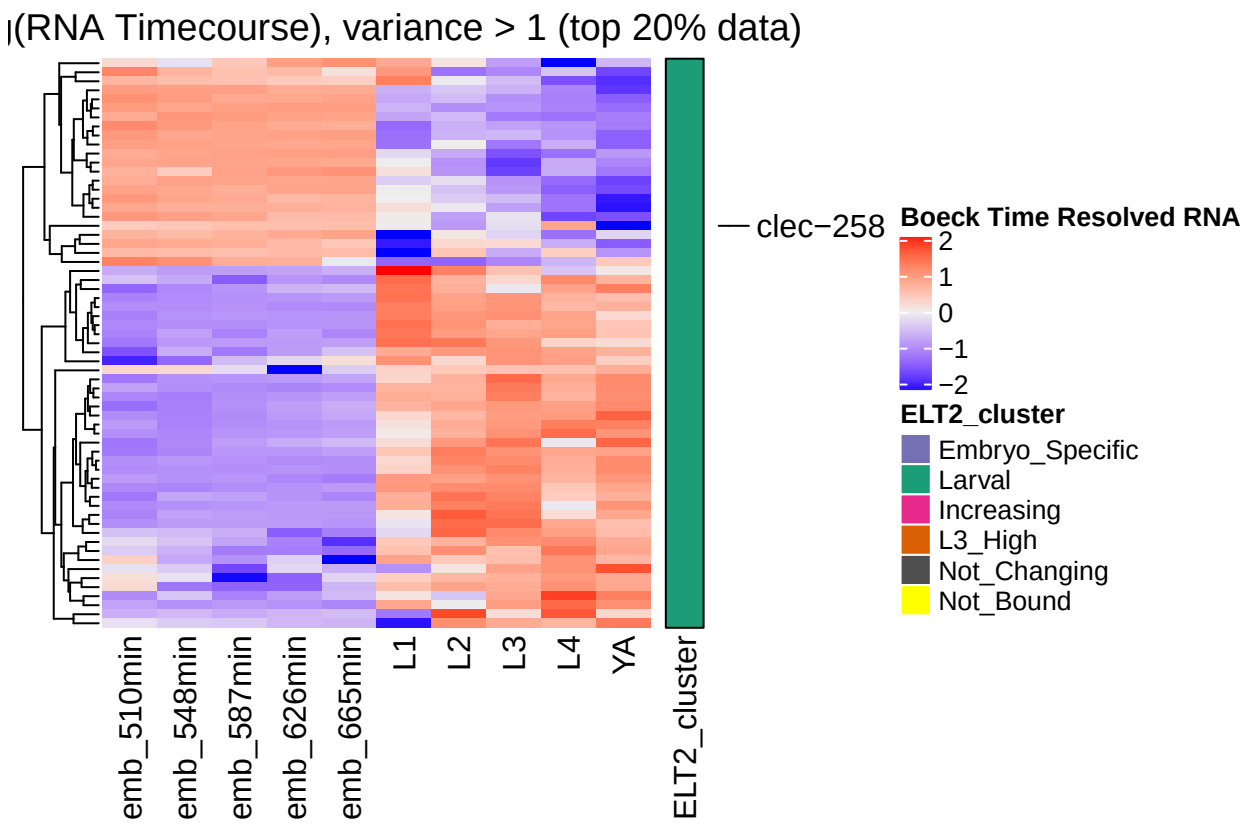

```
### Include only rows with a variance greater than 1 ###
changing = rowvariances > 1
BoeckRNA_ELT2_chip_Heatmap(real & changing, "Log(RNA Timecourse), variance > 1 (top 20% data)")
```




```
BoeckRNA_ELT2_chip_Heatmap(real & changing & embryo_specific, "Log(RNA Timecourse), variance > 1 (top 2
```




```
BoeckRNA_ELT2_chip_Heatmap(real & changing & larval, "Log(RNA Timecourse), variance > 1 (top 20% data)"
```




```
# pdf(file = "./03_plots/201008_Boeck_RNA_ELT2ChIP_Patterns_Subset.pdf",
#     width = 7, height = 7)
# BoeckRNA_ELT2_chip
# dev.off()
```