Clustering of modENCODE/Reinke ChIP-seq peaks

DC King - Onish lab

Spring 2019

Script version

```
## origin   git@github.com:meekrob/onish_ChIP_R_Analysis.git (fetch)
## origin   git@github.com:meekrob/onish_ChIP_R_Analysis.git (push)
## commit 3489af64d1d2b90038179988895d51575a687b120
## Author: David King <davidcking.inbox@gmail.com>
## Date:   Mon May 6 17:28:40 2019 -0600
##
##     ignore .bedlike files
##  M cluster_heatmaps.Rmd
##  M cluster_heatmaps.pdf
## ?? Cao_et_al_2017_vignette.RData
## ?? intestine_TF_database.csv
```

R version, libraries

```
R.version
```

```
##                _
## platform       x86_64-apple-darwin15.6.0
## arch           x86_64
## os             darwin15.6.0
## system         x86_64, darwin15.6.0
## status
## major          3
## minor          5.1
## year           2018
## month          07
## day            02
## svn rev        74947
## language       R
## version.string R version 3.5.1 (2018-07-02)
## nickname       Feather Spray
```

```
library(pheatmap)
```

```
## Warning: package 'pheatmap' was built under R version 3.5.2
```

```
library(stringr)
```

```
## Warning: package 'stringr' was built under R version 3.5.2
```

```
library(reshape2)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

```
library(ggExtra)
```

Process data

```
getwd()
```

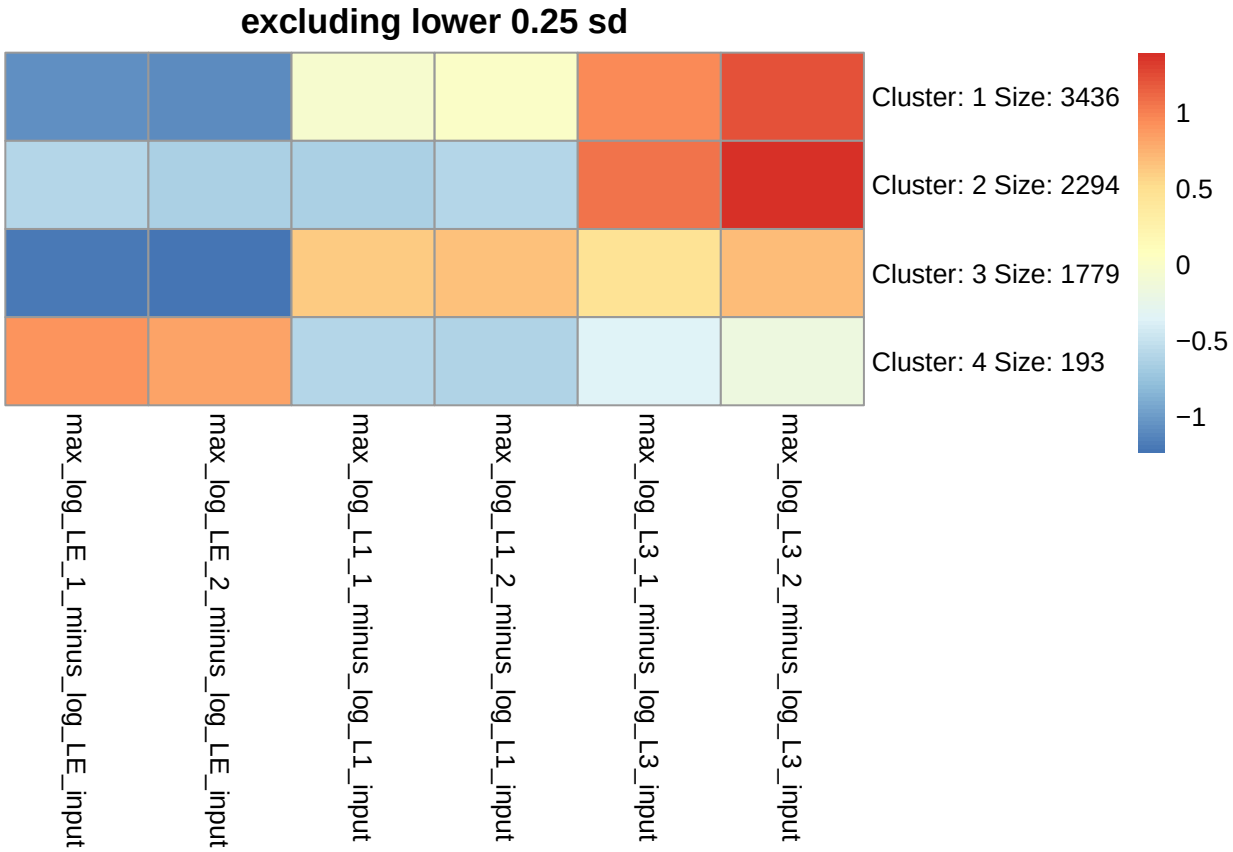
```
## [1] "/Users/david/work/onish_ChIP_R_Analysis"
```

```
pth="allStagesUNION.IDR_0.05.sorted.bed_s.df"
df = read.table(pth, header=T, sep="\t")
all_max_col_ix = which(! is.na(str_match(colnames(df),"max")))
df_max = df[,all_max_col_ix]
all_nan_rows = apply(df_max, 1, function(x) { all(!is.finite(x))})
df_max = df_max[!all_nan_rows,]
df_max[is.na(df_max)] <- 0
colnames(df_max)<-c("max_log_L1_1_minus_log_L1_input", "max_log_L1_2_minus_log_L1_input", "max_log_L3_1
colnames_in_stage_order = colnames(df_max)[c(5,6,1:4)]
position_columns = c("chrom","chromStart","chromEnd","ID")
# make numeric, with
data = as.matrix(df_max[,colnames_in_stage_order])

# perform normalization
peaks_sd = apply(data, 1, sd)
peaks_mu = apply(data, 1, mean)
# normalize by row
row_scaled_x = (data - peaks_mu) /  peaks_sd

# changing versus not-changing
q.peaks_zeroed_nans_sd = function(q) { quantile(peaks_sd, q)}

THRESHOLDS = c(.25,.33,.50,.66)
#THRESHOLDS=seq(.08,.96,length.out=48)

for (THRESHOLD in THRESHOLDS) {
  #THRESHOLD = .66 # exclude this fraction of dataset
  threshold_q = q.peaks_zeroed_nans_sd(THRESHOLD)
  threshold_ix = peaks_sd > threshold_q
  threshold_x = row_scaled_x[ threshold_ix,]

  # df colnames differ from manual ones on line 44
  excluded_set = df[! threshold_ix,c(position_columns, str_c(colnames_in_stage_order, "_z.bw"))]
  excluded_filename = paste("allStagesUNION_max_sd_", THRESHOLD, "_0.bedlike", sep="")

  # add the comment char to the first column
  colnames(excluded_set) <- c("#chrom", colnames(excluded_set)[2:ncol(excluded_set)])
  # write data file of excluded regions
  write.table(excluded_set, excluded_filename, quote=F, sep="\t", row.names=F)

  print(paste("clustering with threshold", THRESHOLD, sep=" "))

  nclust=4
  set.seed(31415)
  pobj = pheatmap(threshold_x, kmeans=nclust, cluster_rows=F, cluster_cols = F, main=paste("excluding l
```

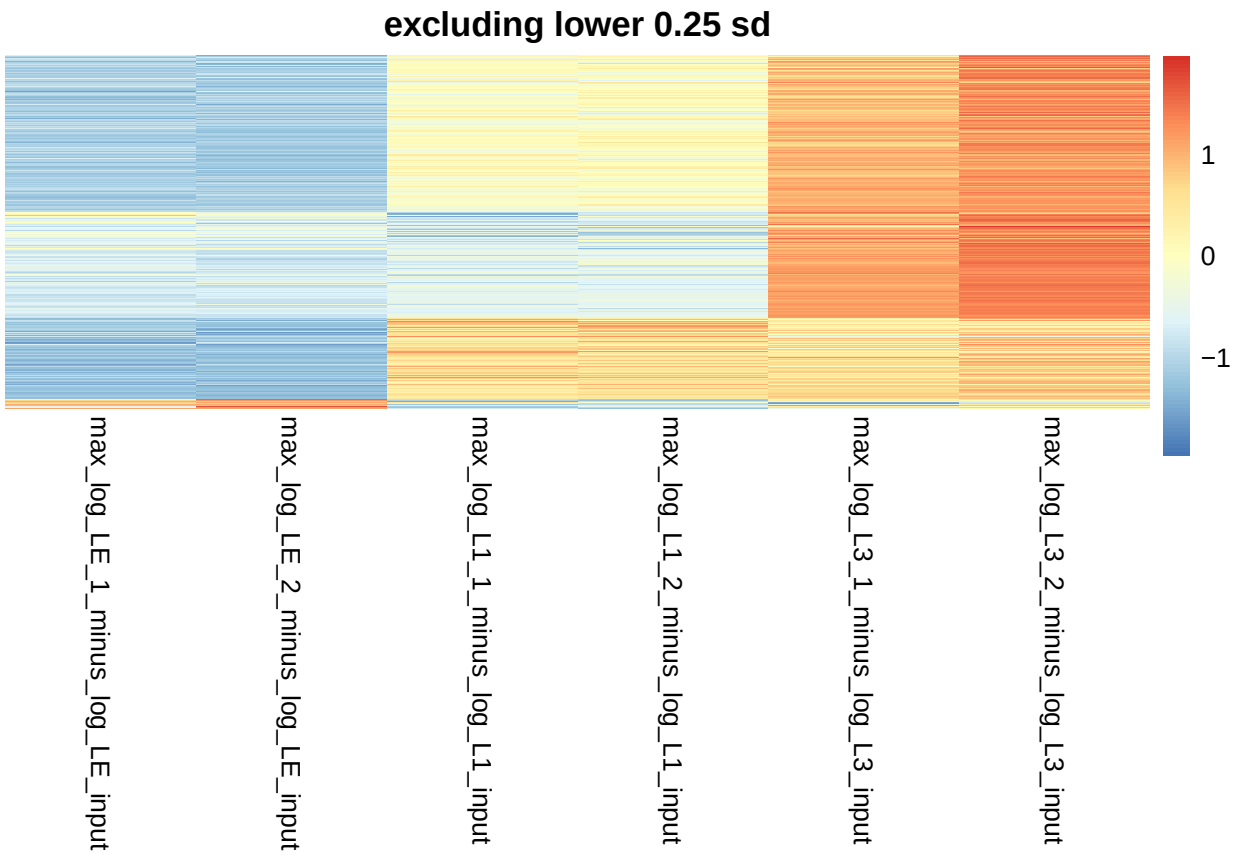
```
kclusters = pobj$kmeans$cluster


# prepare non-normalized output
# df colnames differ from manual ones on line 44
included_set = df[threshold_ix,c(position_columns, str_c(colnames_in_stage_order, "_z.bw"))]
split_non_normalized=split(included_set, kclusters)
for (k_i in names(split_non_normalized)) {
  k_subset = split_non_normalized[[k_i]]
  kcluster_filename = paste("allStagesUNION_max_sd_", THRESHOLD, "_", k_i, ".bedlike", sep="")
  print(kcluster_filename)
  write.table(k_subset, kcluster_filename, quote=F, sep="\t", row.names=F)
}

korder = order(kclusters, peaks_sd[threshold_ix])

pheatmap(threshold_x[korder,], cluster_rows=F, cluster_cols=F, show_rownames=F, main=paste("excluding

patterns = rbind((pobj$kmeans$centers[,1] + pobj$kmeans$centers[,2])/2, (pobj$kmeans$centers[,3] + pob
rownames(patterns) <- c("LE","L1","L3")

LEs = apply(pobj$kmeans$centers[,1:2],1, mean)
L1s = apply(pobj$kmeans$centers[,3:4],1, mean)
L3s = apply(pobj$kmeans$centers[,5:6],1, mean)

trends = cbind(LE=LEs,L1=L1s,L3=L3s)
trends_ordered = trends[order(LEs,L1s,L3s,decreasing=TRUE),]
rownames(trends_ordered) <- c("LE specific", "L3", "increasing",  "other k")


trends_ordered_long = melt(trends_ordered)
colnames(trends_ordered_long) <- c("description", "stage", "center")
ggplot(trends_ordered_long, aes(x=stage,y=center, group=description,colour=description )) + geom_line
print(trends_ordered_long)

if (FALSE) {
  plot(c(1,3), c(min(patterns),2), type='n', axes=F, main=paste("Cluster centers, excluding lower",TH

  legend_text=c(); for (clustn in 1:nclust) { legend_text=c(legend_text,str_c("clust",clustn, sep=" "
  color_thing = RColorBrewer::brewer.pal(name="Dark2",n=nclust)
  legend(1,2, legend_text,lwd=2, col=color_thing,horiz=T,bty='n')
  axis(1, at=1:3, c('LE','L1','L3'))
  axis(2, at=axTicks(2, c(-1,1.5,5), c(-1,1)))
  abline(h=0, lwd=.5,lty=2)
  for (wd in 1:nclust) {
    lines(patterns[,wd],col=color_thing[wd], lwd=2)
  }
  rect(par('usr')[1], par('usr')[3], par('usr')[2], 1.51)
}
## recoup the original file, the normalized data, and the cluster assignment
# Going to add 7 columns: the normalized value for each stage (times 2 reps) plus the
# cluster assignment
```

```
  # normalized data
  norm_LE_1 = rep(NA, nrow(df)); norm_LE_1[!all_nan_rows] = row_scaled_x[,1]
  norm_LE_2 = rep(NA, nrow(df)); norm_LE_2[!all_nan_rows] = row_scaled_x[,2]
  norm_L1_1 = rep(NA, nrow(df)); norm_L1_1[!all_nan_rows] = row_scaled_x[,3]
  norm_L1_2 = rep(NA, nrow(df)); norm_L1_2[!all_nan_rows] = row_scaled_x[,4]
  norm_L3_1 = rep(NA, nrow(df)); norm_L3_1[!all_nan_rows] = row_scaled_x[,5]
  norm_L3_2 = rep(NA, nrow(df)); norm_L3_2[!all_nan_rows] = row_scaled_x[,6]

  # kmeans cluster assignment
  vec1 = rep(NA, nrow(row_scaled_x))
  vec1[threshold_ix] = kclusters
  kclust_mapping = rep(NA, nrow(df))
  kclust_mapping[! all_nan_rows] = vec1

  # sd and mean
  sd_mapping = rep(NA, nrow(df))
  sd_mapping[!all_nan_rows] = peaks_sd
  mean_mapping = rep(NA, nrow(df))
  mean_mapping[!all_nan_rows] = peaks_mu
  ecdf_sd = ecdf(peaks_sd)

  # peaks_quantile: this row is changing more (via sd) than peaks_quantile fraction of the dataset
  peaks_quantile = rep(NA, nrow(df))
  peaks_quantile[!all_nan_rows] = ecdf_sd(peaks_sd)
  # combine new columns
  new_df = data.frame(norm_LE_1,norm_LE_2,norm_L1_1,norm_L1_2,norm_L3_1,norm_L3_2,mean_mapping, sd_mapp

  # paste new columns onto original file
  big_df = cbind(df, new_df)
  # add the comment char to the first column
  colnames(big_df) <- c("#chrom", colnames(big_df)[2:38])

  ## add gene ids, processed as in the repository
  gene_ids = read.table("only.gene_ids")
  new_colnames = c(colnames(big_df), "WBID")
  big_df = cbind(big_df, gene_ids)
  colnames(big_df) <- new_colnames

  # write data file
  output_filename = paste("./valerie_peaks_processed_", THRESHOLD, ".bedlike",sep="")
  write.table(big_df, output_filename, quote=F, sep="\t", row.names=F)

}
```

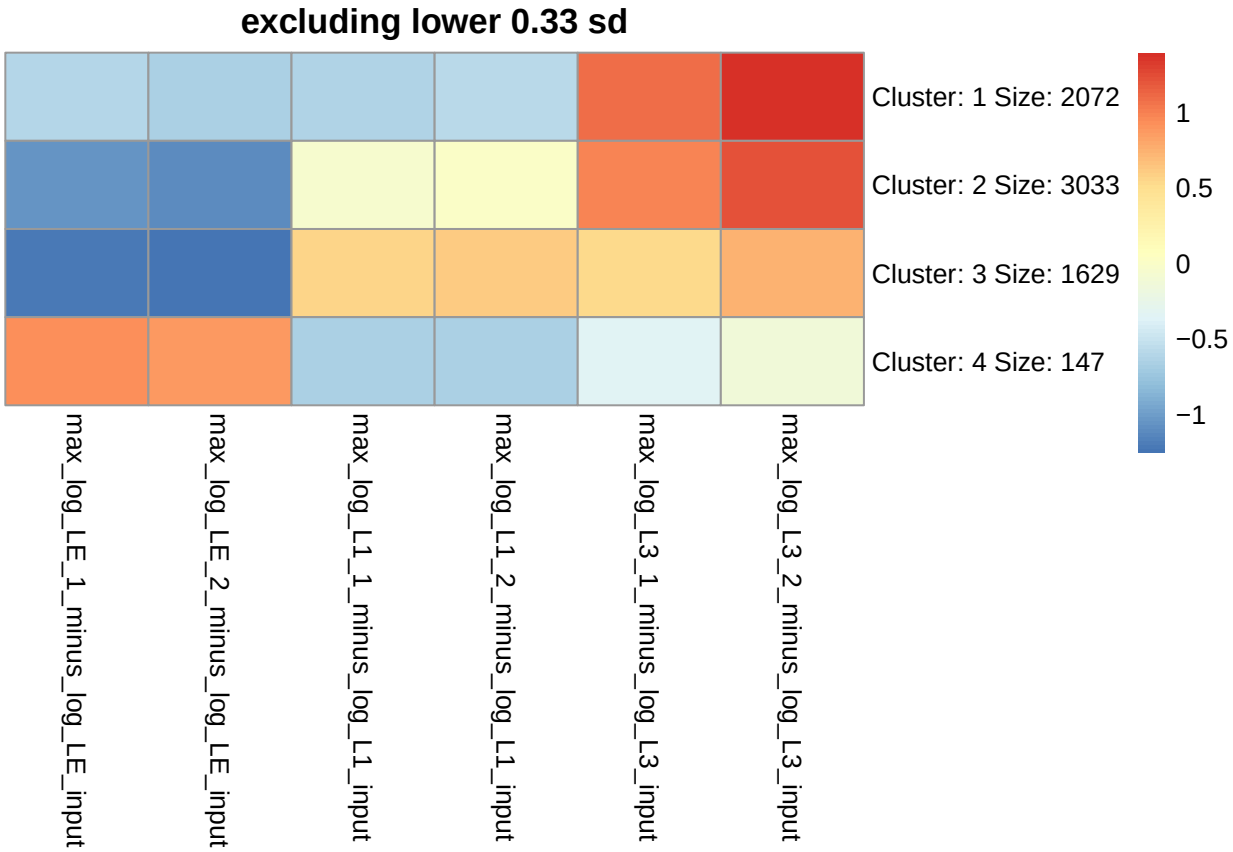
```
## [1] "clustering with threshold 0.25"
```



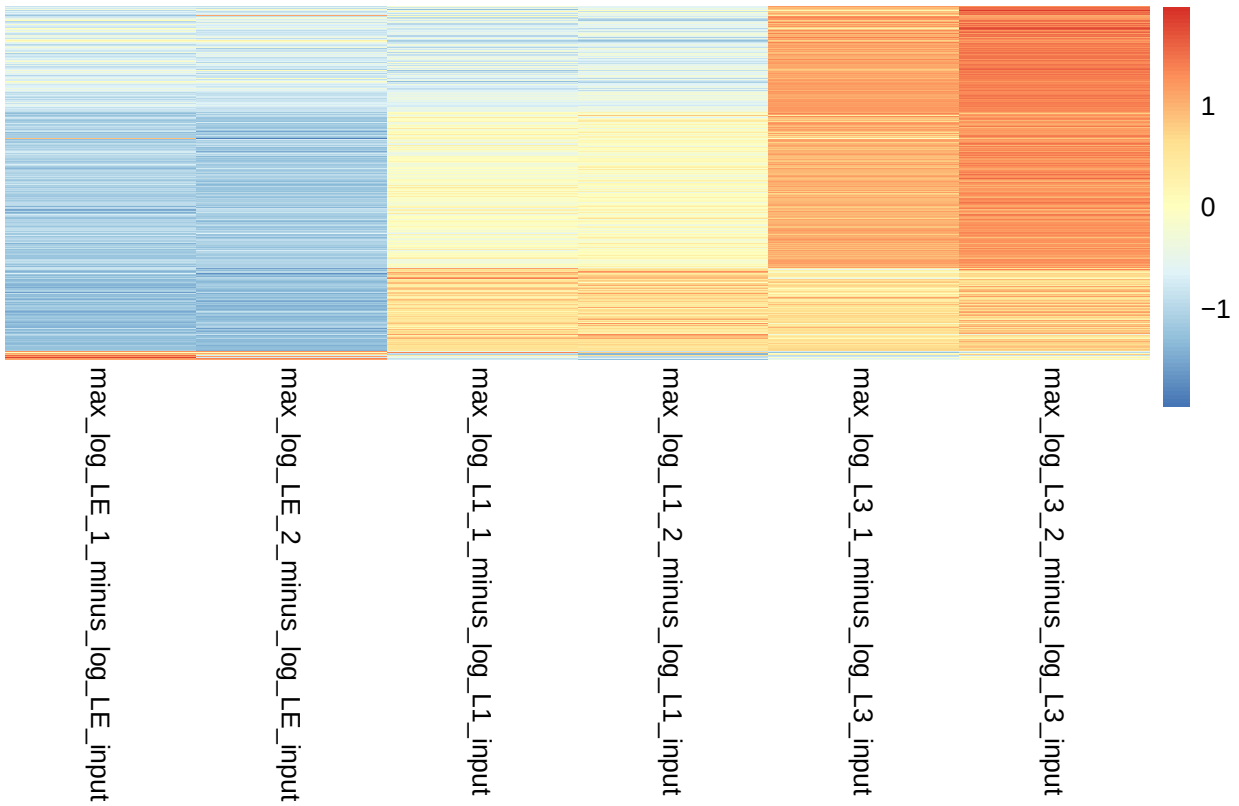

```
## [1] "allStagesUNION_max_sd_0.25_1.bedlike"
## [1] "allStagesUNION_max_sd_0.25_2.bedlike"
## [1] "allStagesUNION_max_sd_0.25_3.bedlike"
## [1] "allStagesUNION_max_sd_0.25_4.bedlike"
```



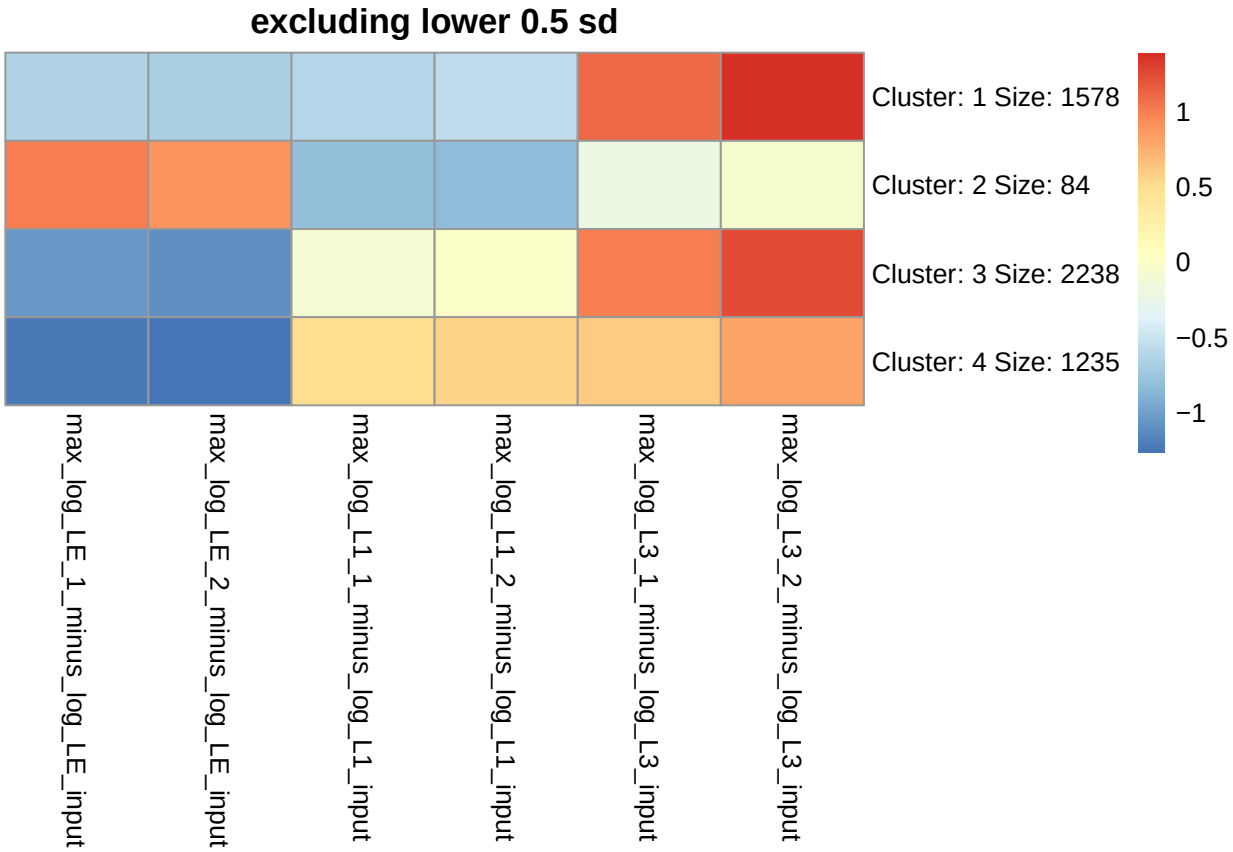

```
##    description stage      center
## 1  LE specific    LE  0.87303224
## 2           L3    LE -0.61716357
## 3   increasing    LE -1.08242319
## 4       other k    LE -1.21716454
## 5  LE specific    L1 -0.61521457
## 6           L3    L1 -0.61875682
## 7   increasing    L1 -0.00632231
## 8       other k    L1  0.64024991
## 9  LE specific    L3 -0.25781767
## 10          L3    L3  1.23592039
## 11  increasing    L3  1.08874550
## 12      other k    L3  0.57691463
## [1] "clustering with threshold 0.33"
```



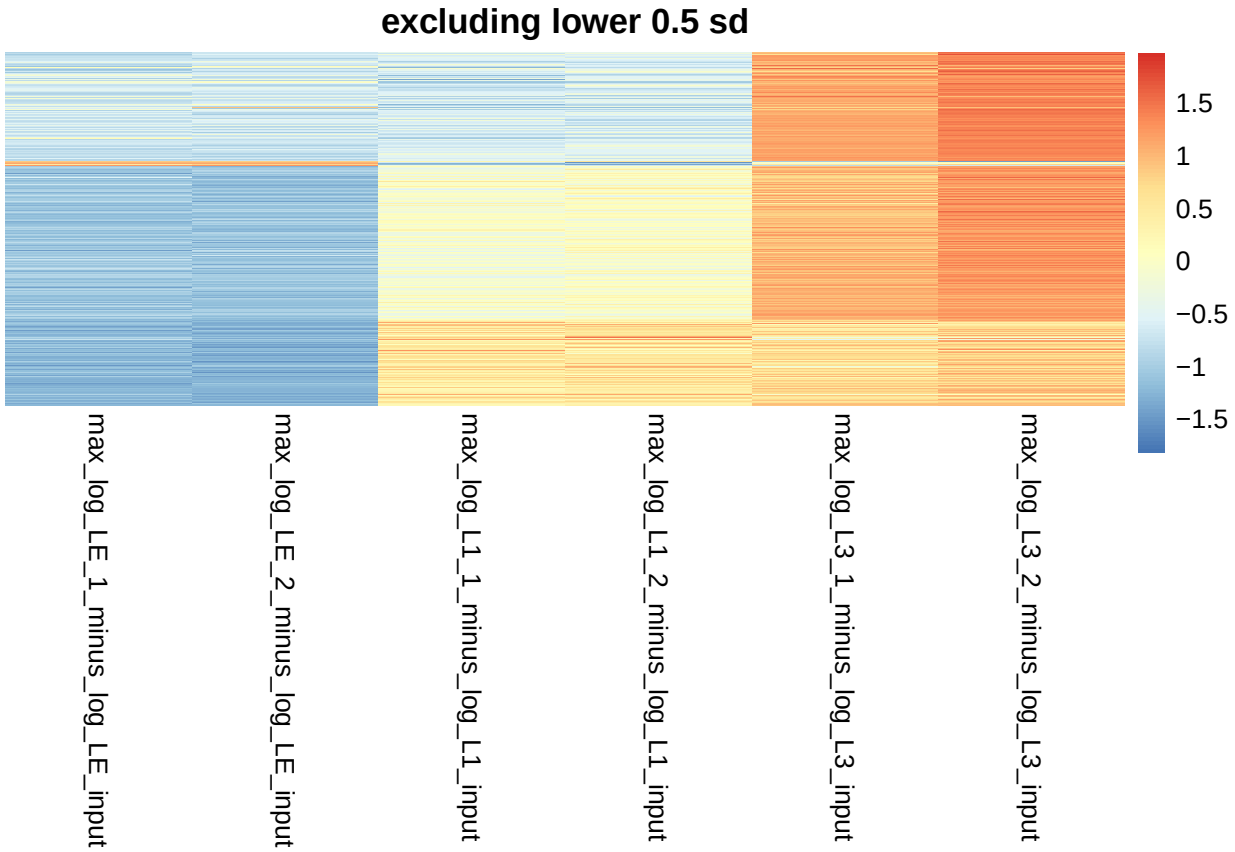

```
## [1] "allStagesUNION_max_sd_0.33_1.bedlike"
## [1] "allStagesUNION_max_sd_0.33_2.bedlike"
## [1] "allStagesUNION_max_sd_0.33_3.bedlike"
## [1] "allStagesUNION_max_sd_0.33_4.bedlike"
```



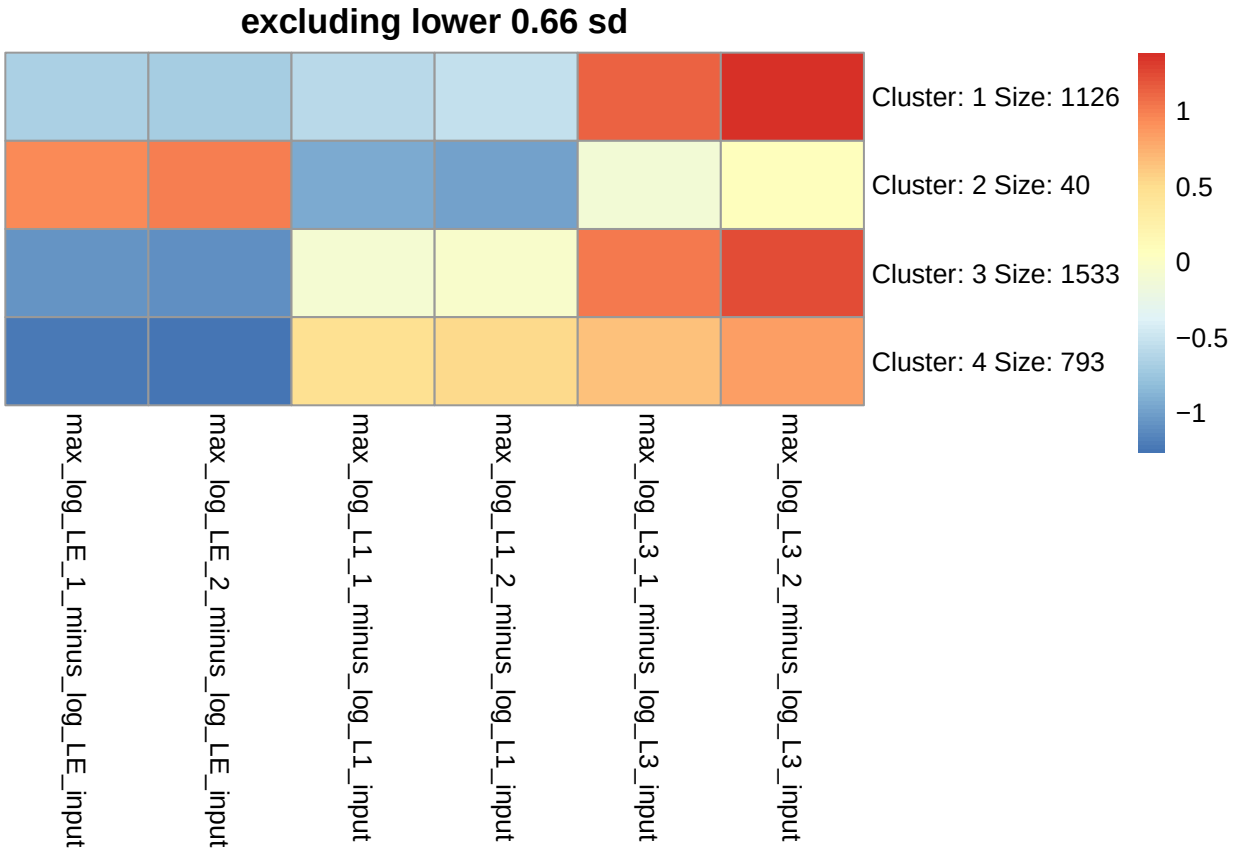

```
##    description stage      center
## 1  LE specific    LE  0.90093250
## 2           L3    LE -0.63293917
## 3   increasing    LE -1.07779963
## 4      other k    LE -1.23248674
## 5  LE specific    L1 -0.65378017
## 6           L3    L1 -0.60914421
## 7   increasing    L1 -0.02276464
## 8      other k    L1  0.59565470
## 9  LE specific    L3 -0.24715233
## 10          L3    L3  1.24208338
## 11  increasing    L3  1.10056427
## 12     other k    L3  0.63683204
## [1] "clustering with threshold 0.5"
```



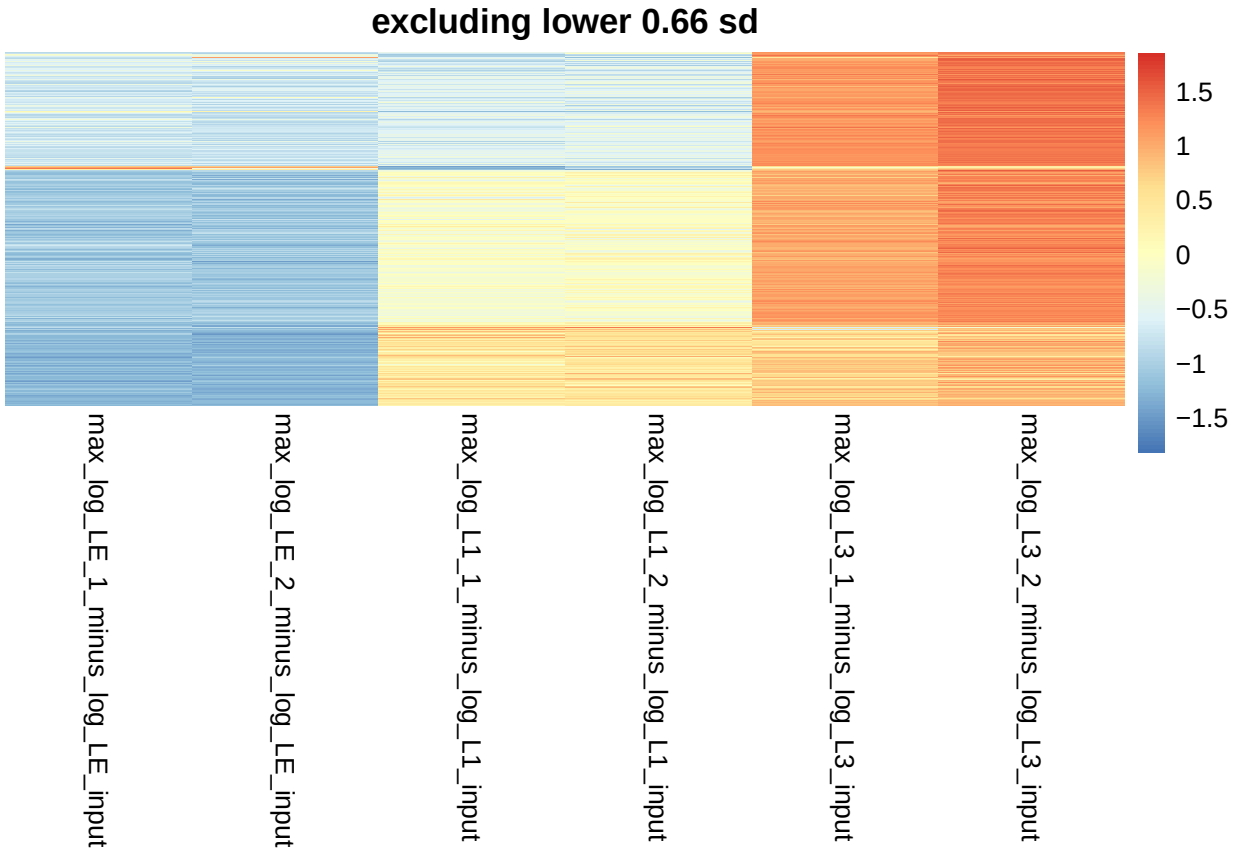

```
## [1] "allStagesUNION_max_sd_0.5_1.bedlike"
## [1] "allStagesUNION_max_sd_0.5_2.bedlike"
## [1] "allStagesUNION_max_sd_0.5_3.bedlike"
## [1] "allStagesUNION_max_sd_0.5_4.bedlike"
```




```
##    description stage      center
## 1  LE specific    LE  0.94971720
## 2           L3    LE -0.65718239
## 3   increasing    LE -1.07142505
## 4      other k    LE -1.24291118
## 5  LE specific    L1 -0.80286936
## 6           L3    L1 -0.59305469
## 7   increasing    L1 -0.04727058
## 8      other k    L1  0.52871332
## 9  LE specific    L3 -0.14684784
## 10          L3    L3  1.25023708
## 11  increasing    L3  1.11869563
## 12     other k    L3  0.71419785
## [1] "clustering with threshold 0.66"
```




```
## [1] "allStagesUNION_max_sd_0.66_1.bedlike"
## [1] "allStagesUNION_max_sd_0.66_2.bedlike"
## [1] "allStagesUNION_max_sd_0.66_3.bedlike"
## [1] "allStagesUNION_max_sd_0.66_4.bedlike"
```




```
##    description stage      center
## 1  LE specific    LE  0.97898674
## 2           L3    LE -0.68450130
## 3   increasing    LE -1.06976537
## 4      other k    LE -1.24438063
## 5  LE specific    L1 -0.96363200
## 6           L3    L1 -0.57298286
## 7   increasing    L1 -0.05793865
## 8      other k    L1  0.48705674
## 9  LE specific    L3 -0.01535474
## 10          L3    L3  1.25748416
## 11  increasing    L3  1.12770402
## 12     other k    L3  0.75732390
```