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 bb847f8828bef5a058d994016ebad498748c8195
## Author: David King <davidcking.inbox@gmail.com>
## Date:   Fri May 3 15:14:10 2019 -0600
##
##     improved labels
##  M cluster_heatmaps.Rmd
## ?? cluster_heatmaps.html
## ?? cluster_heatmaps.pdf
## ?? valerie_peaks_processed.bedlike
```

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(ggplot2)
```

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

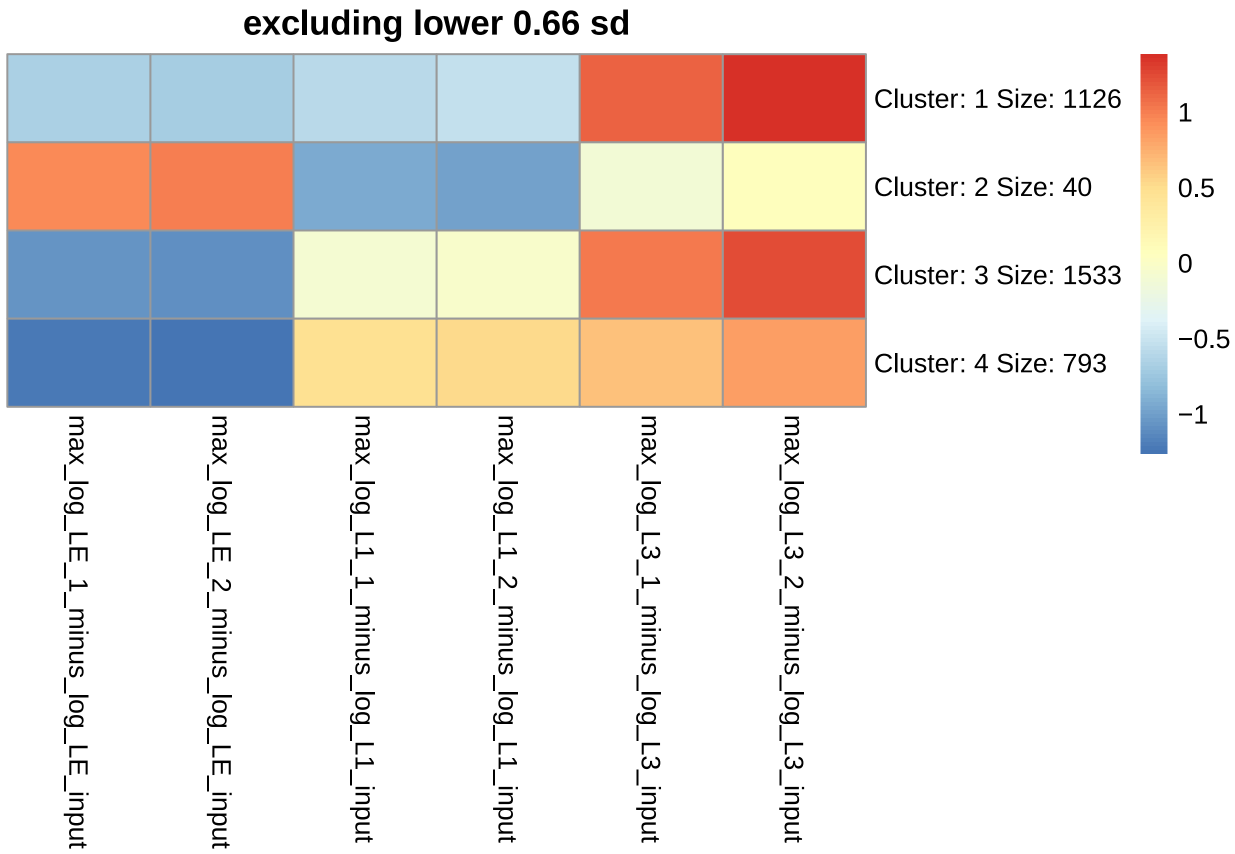
```
library(ggExtra)
```

Process data

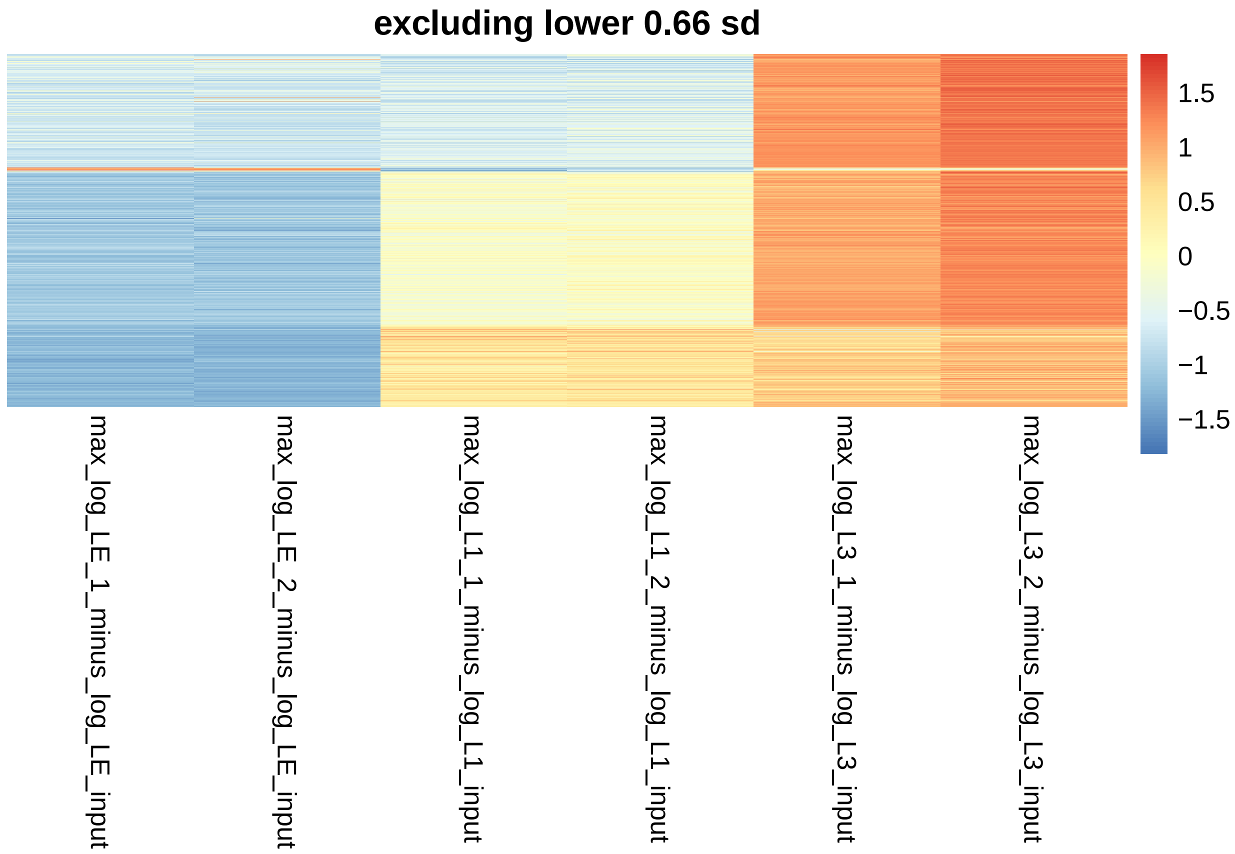
```r
getwd()
```

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

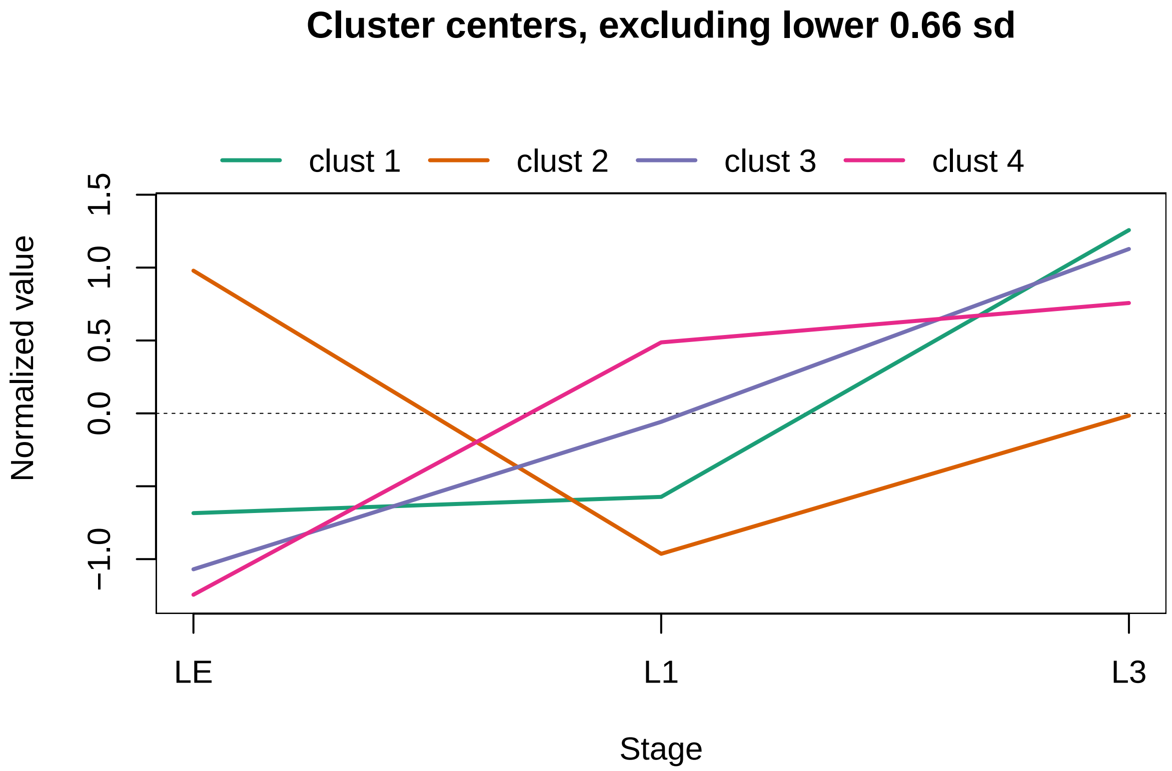
```r
pth="allStagesUNION.IDR_0.05.sorted.bed_s.df"
df = read.table(pth, header=T, sep="\t")
all_max_col_ix = which(! is.na(str_match(colnames(df),"max")))
df_max = df[,all_max_col_ix]
all_nan_rows = apply(df_max, 1, function(x) { all(!is.finite(x))})
df_max = df_max[!all_nan_rows,]
df_max[is.na(df_max)]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

# 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)}
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,]

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



```
kclusters = pobj$kmeans$cluster



korder = order(kclusters, peaks_sd[threshold_ix])

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



```
patterns = rbind((pobj$kmeans$centers[,1] + pobj$kmeans$centers[,2])/2, (pobj$kmeans$centers[,3] + pobj$
rownames(patterns) <- c("LE","L1","L3")
plot(c(1,3), c(min(patterns),2), type='n', axes=F, main=paste("Cluster centers, excluding lower",THRESH

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_mapping

# 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
write.table(big_df, "./valerie_peaks_processed.bedlike", quote=F, sep="\t", row.names=F)
```