Bioinformatics HW 6

Ryan Gallagher

2023-12-08

## 1) Use the dataset “120217\_cellCycleData.txt” and perform KNN imputation. Use only the alpha-factor synchronized data (the 6th to 23rd columns) in the dataset.

dat = read.table('~/Desktop/MedicalCollegeofWisconsin/Bioinformatics/HW6/120217\_W7\_cellCycleData.txt', sep='\t', header=TRUE)  
labs = dat %>% select(1, length(dat)) %>% rename('gene\_name' = row.names)  
dat = dat %>% select(c(6:23))  
  
dat.knn = impute.knn(as.matrix(dat))

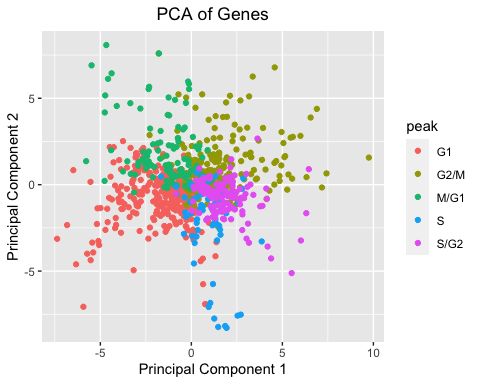
## Warning in knnimp(x, k, maxmiss = rowmax, maxp = maxp): 8 rows with more than 50 % entries missing;  
## mean imputation used for these rows

dat.df = as.data.frame(dat.knn$data)

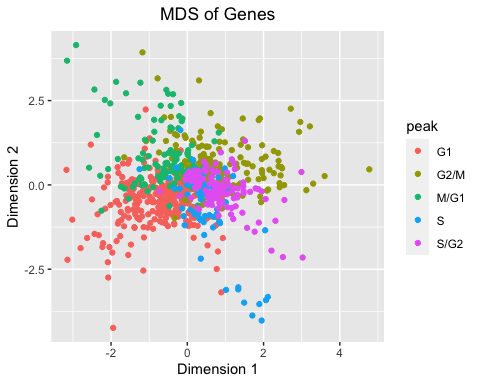
### a) Project the 800 genes onto two-dimensional space using PCA or MDS respectively. Label the genes of different functional annotation (“G1”, “S”, “S/G2”, “G2/M”, “M/G1”) with different colors (col=1~5) and text symbols (1~5)

#PCA results  
dat.pca = prcomp(dat.df, scale. = TRUE)  
dat.pca = as.data.frame(dat.pca$x[,1:2])  
dat.pca = cbind(labs, dat.pca)  
  
#MDS results  
dat.dist = dist(dat.knn$data, method = 'euclidean')  
dat.mds = cmdscale(dat.dist, k=2)  
dat.mds = as.data.frame(dat.mds)  
dat.mds = cbind(labs, dat.mds)  
names(dat.mds) = c("gene\_name", "peak", 'Dim1', 'Dim2')

## PCA Plots  
ggplot(dat.pca, aes(x = PC1, y = PC2, color=peak)) +  
 geom\_point() +  
 xlab("Principal Component 1") +  
 ylab("Principal Component 2") +  
 ggtitle("PCA of Genes") +   
 theme(plot.title = element\_text(hjust=0.5))



### MDS Plots  
  
ggplot(dat.mds, aes(x = Dim1, y = Dim2, color=peak)) +  
 geom\_point() +  
 xlab("Dimension 1") +  
 ylab("Dimension 2") +  
 ggtitle("MDS of Genes") +  
 theme(plot.title = element\_text(hjust=0.5))



### b) Perform hierarchical clustering (K=5) and K-means (K=5). Compare clustering result from each algorithm to the truth given in the last column of the data (“G1”, “S1”, …). Calculate adjusted Rand indexes (“adjustedRandIndex” in “mclust” package) for the comparisons. Which is better? Summarize what you observe.

I’m a little confused at how the indexes are related to the labeling from our orginial dataset. I would assume that theyre alphabetical: G1=1, G2/M=2, M/G1=3, S=4, S/G2=5 - since this is how theyre orgnaized in this provided dataset. However, I’m getting a RandIndex near 0 for both methods. I can’t tell if it’s a result of the algorithms, or I’m misunderstanding the index to label relationship.

set.seed(218)  
hc = hclust(dat.dist, method = 'complete')  
clusters = cutree(hc, k=5)  
cluster\_names = setNames(unique(labs$peak), 1:5)  
named\_clusters = cluster\_names[as.character(clusters)]  
  
kmeans\_result = kmeans(dat.df, centers = 5, nstart = 25)  
named\_clisters2 = cluster\_names[as.character(kmeans\_result$cluster)]  
  
dat.df2 = cbind(labs, dat.df)  
dat.df2$cluster\_pred = named\_clusters  
dat.df2$kmeans\_pred = named\_clisters2  
  
library(mclust)

## Package 'mclust' version 6.0.1  
## Type 'citation("mclust")' for citing this R package in publications.

##   
## Attaching package: 'mclust'

## The following object is masked from 'package:purrr':  
##   
## map

#clustering method  
adjustedRandIndex(dat.df2$peak, dat.df2$cluster\_pred)

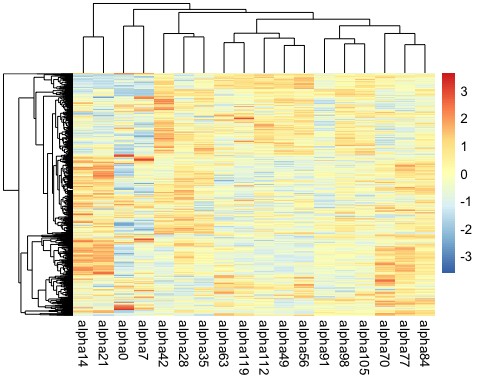
## [1] 0.01394729

#kmeans method  
adjustedRandIndex(dat.df2$peak, dat.df2$kmeans\_pred)

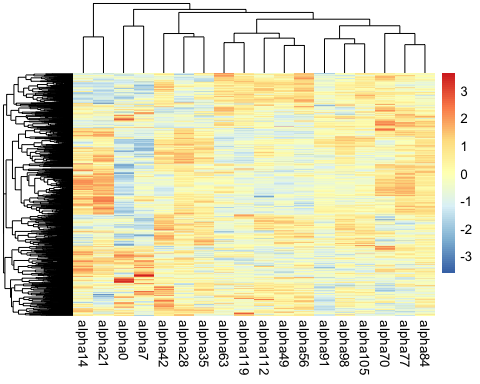
## [1] 0.1328062

### c) Plot the heatmap of cluster patterns from each method

library(pheatmap)  
#heatmap(as.matrix(dat.knn$data), Rowv = as.dendrogram(hc), Colv = NA, scale='row')  
pheatmap(dat.knn$data, cluster\_rows = hc, scale='row')



ordered\_km = dat.knn$data[order(kmeans\_result$cluster), ]  
pheatmap(ordered\_km, scale='row')



## 2) In this problem we are going to practice pathway analysis. Let’s re-use the DE analysis results from HW4 1e. For pathway database, load in the GO gene set from file ‘c5.all.v2.5.symbols.gmt’. In order to convert the identifier in Golub\_Merge to gene symbol, you need to download the package ‘hu6800’ from Bioconductor. Use function ‘hu6800SYMBOL’ to achieve this goal.

### a) How many gene sets are available from ‘c5.all.v2.5.symbols.gmt’? Overlap genes available in this file and Golub\_Merge which is used in HW4. How many genes are overlapped? Update the gene sets by dropping genes that do not appear in Golub\_Merge.

library(golubEsets)

## Loading required package: Biobase

## Loading required package: BiocGenerics

##   
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:lubridate':  
##   
## intersect, setdiff, union

## The following objects are masked from 'package:dplyr':  
##   
## combine, intersect, setdiff, union

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

## The following objects are masked from 'package:base':  
##   
## anyDuplicated, aperm, 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

## Welcome to Bioconductor  
##   
## Vignettes contain introductory material; view with  
## 'browseVignettes()'. To cite Bioconductor, see  
## 'citation("Biobase")', and for packages 'citation("pkgname")'.

library(GSA)  
library(hu6800.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:lubridate':  
##   
## second, second<-

## The following objects are masked from 'package:dplyr':  
##   
## first, rename

## The following object is masked from 'package:tidyr':  
##   
## expand

## The following object is masked from 'package:utils':  
##   
## findMatches

## The following objects are masked from 'package:base':  
##   
## expand.grid, I, unname

##   
## Attaching package: 'IRanges'

## The following object is masked from 'package:lubridate':  
##   
## %within%

## The following objects are masked from 'package:dplyr':  
##   
## collapse, desc, slice

## The following object is masked from 'package:purrr':  
##   
## reduce

##   
## Attaching package: 'AnnotationDbi'

## The following object is masked from 'package:dplyr':  
##   
## select

## Loading required package: org.Hs.eg.db

##

##

data(Golub\_Merge)  
golub = Golub\_Merge  
pheno\_data = pData(Golub\_Merge)  
expn = exprs(golub)  
  
gmt\_file = GSA.read.gmt('/Users/ryangallagher/Desktop/MedicalCollegeofWisconsin/Bioinformatics/HW6/c5\_all\_v2\_5\_symbols.gmt')  
  
## Get names  
feature\_id = featureNames(golub)  
gene\_symbols = mget(feature\_id, hu6800SYMBOL, ifnotfound=NA)  
gene\_symbols\_df = data.frame(ProbeID = names(gene\_symbols),   
 GeneSymbol = unlist(gene\_symbols),   
 stringsAsFactors = FALSE)  
  
## Get only genes in .gmt that are in golub  
gmt\_names = as.data.frame(unlist(gmt\_file$genesets))  
colnames(gmt\_names) = c('names')  
  
gmt\_filtered = gmt\_names %>%  
 filter(names %in% gene\_symbols\_df$GeneSymbol)  
  
## relist them  
filtered\_genesets = lapply(gmt\_file$genesets, function(geneset) {  
 intersect(geneset, gene\_symbols\_df$GeneSymbol)  
})  
  
new\_gmt = gmt\_file   
new\_gmt$genesets = filtered\_genesets

## Output  
print(paste("There are", length(gmt\_file$genesets), "gene sets in the .gmt file"))

## [1] "There are 1454 gene sets in the .gmt file"

print(paste("There are", nrow(unique(gmt\_filtered)), "overlap genes"))

## [1] "There are 4109 overlap genes"

### b) Filter out any gene sets that contain less than 5 genes or more than 200 genes. How many gene sets are left?

# Filter out gene sets with less than 5 or more than 200 genes  
new\_gmt$genesets = Filter(function(geneset) {  
 len = length(geneset)  
 len >= 5 && len <= 200  
}, new\_gmt$genesets)  
  
print(paste(length(new\_gmt$genesets), 'left after filtering.'))

## [1] "1254 left after filtering."

### c) Perform Fisher’s exact test for each gene set on the DE genes derived in HW4 1(e). Pay attention to the background gene that you use. How many pathways are significant? List the top 10 pathways. Comment on the results

## Find DE genes like in Hw4  
library(golubEsets)  
library(limma)

##   
## Attaching package: 'limma'

## The following object is masked from 'package:BiocGenerics':  
##   
## plotMA

exprs\_data = exprs(Golub\_Merge)  
class\_labels = pData(Golub\_Merge)$ALL.AML  
  
design = model.matrix(~class\_labels)  
fit = lmFit(exprs\_data, design)  
fit = eBayes(fit)  
tab = topTable(fit, coef=2, number=Inf, p.value=0.01)  
  
## Identify DE gene names  
tab$name = row.names(tab)  
tab$gene\_names = unlist(mget(tab$name, hu6800SYMBOL, ifnotfound=NA))

tab = tab %>% mutate(significant = ifelse(P.Value < 0.5,1,0)) %>% filter(gene\_names %in% gene\_symbols\_df$GeneSymbol)  
  
results = lapply(new\_gmt$genesets, function(gene\_set) {  
 #build contigency table elements  
 significant\_in\_set = sum(gene\_set %in% tab$name[tab$significant])  
 significant\_not\_in\_set = sum(!gene\_set %in% tab$name[tab$significant])  
 not\_significant\_in\_set = sum(gene\_set %in% tab$name[!tab$significant])  
 not\_significant\_not\_in\_set = sum(!gene\_set %in% tab$name[!tab$significant])  
  
 matrix = matrix(c(significant\_in\_set, significant\_not\_in\_set,   
 not\_significant\_in\_set, not\_significant\_not\_in\_set),   
 nrow = 2, byrow = TRUE)  
  
 fisher.test(matrix)$p.value  
})  
  
p\_values = unlist(results)  
significant\_gene\_sets = which(p\_values < 0.05)

A little confused on the values for this table. I imagine we’d extract a p\_value based on the FDR from the golub data, then compare it to the gene set in someway via a 2x2 table.