a. Perform ANOVA test on the discriminant analysis scores of nuclear localization signals of

both nuclear and non-nuclear proteins by class variables (Target).

b. Which class is significantly different from others?

#1. Title: Protein Localization Sites

#2. Creator and Maintainer:

# Kenta Nakai

##Osaka, University

##nakai@imcb.osaka-u.ac.jp

# http://www.imcb.osaka-u.ac.jp/nakai/psort.html

#Donor: Paul Horton (paulh@cs.berkeley.edu)

#Date: September, 1996

#See also: ecoli database

#3. Past Usage.

#Reference: "A Probablistic Classification System for Predicting the Cellular

# Localization Sites of Proteins", Paul Horton & Kenta Nakai,

# Intelligent Systems in Molecular Biology, 109-115.

# St. Louis, USA 1996.

#Results: 55% for Yeast data with an ad hoc structured

# probability model. Also similar accuracy for Binary Decision Tree and

# Bayesian Classifier methods applied by the same authors in

# unpublished results.

#Predicted Attribute: Localization site of protein. ( non-numeric ).

#4. The references below describe a predecessor to this dataset and its

#development. They also give results (not cross-validated) for classification

#by a rule-based expert system with that version of the dataset.

#Reference: "Expert Sytem for Predicting Protein Localization Sites in

# Gram-Negative Bacteria", Kenta Nakai & Minoru Kanehisa,

# PROTEINS: Structure, Function, and Genetics 11:95-110, 1991.

#Reference: "A Knowledge Base for Predicting Protein Localization Sites in

# Eukaryotic Cells", Kenta Nakai & Minoru Kanehisa,

# Genomics 14:897-911, 1992.

#5. Number of Instances: 1484 for the Yeast dataset.

#6. Number of Attributes.

# for Yeast dataset: 9 ( 8 predictive, 1 name )

#7. Attribute Information.

# 1. Sequence Name: Accession number for the SWISS-PROT database

# 2. mcg: McGeoch's method for signal sequence recognition.

# 3. gvh: von Heijne's method for signal sequence recognition.

# 4. alm: Score of the ALOM membrane spanning region prediction program.

# 5. mit: Score of discriminant analysis of the amino acid content of

# the N-terminal region (20 residues long) of mitochondrial and

# non-mitochondrial proteins.

# 6. erl: Presence of "HDEL" substring (thought to act as a signal for

# retention in the endoplasmic reticulum lumen). Binary attribute.

# 7. pox: Peroxisomal targeting signal in the C-terminus.

# 8. vac: Score of discriminant analysis of the amino acid content of

# vacuolar and extracellular proteins.

# 9. nuc: Score of discriminant analysis of nuclear localization signals

# of nuclear and non-nuclear proteins.

#8. Missing Attribute Values: None.

#9. Class Distribution. The class is the localization site. Please see Nakai

&

# Kanehisa referenced above for more details.

# CYT (cytosolic or cytoskeletal) 463

# NUC (nuclear) 429

# MIT (mitochondrial) 244

# ME3 (membrane protein, no N-terminal signal) 163

# ME2 (membrane protein, uncleaved signal) 51

# ME1 (membrane protein, cleaved signal) 44

#EXC (extracellular) 37

# VAC (vacuolar) 30

#POX (peroxisomal) 20

#ERL (endoplasmic reticulum lumen) 5

yeast <- read.table("C:/Users/Seshan/Desktop/sv R related/google search R/ass

ignment12/yeast.txt", quote="\"", comment.char="")

View(yeast)

summary(yeast)

## V1 V2 V3 V4

## EF1A\_YEAST: 2 Min. :0.1100 Min. :0.1300 Min. :0.21

## H3\_YEAST : 2 1st Qu.:0.4100 1st Qu.:0.4200 1st Qu.:0.46

## H4\_YEAST : 2 Median :0.4900 Median :0.4900 Median :0.51

## IF4A\_YEAST: 2 Mean :0.5001 Mean :0.4999 Mean :0.50

## MAT2\_YEAST: 2 3rd Qu.:0.5800 3rd Qu.:0.5700 3rd Qu.:0.55

## MTC\_YEAST : 2 Max. :1.0000 Max. :1.0000 Max. :1.00

## (Other) :1472

## V5 V6 V7 V8

## Min. :0.0000 Min. :0.5000 Min. :0.0000 Min. :0.0000

## 1st Qu.:0.1700 1st Qu.:0.5000 1st Qu.:0.0000 1st Qu.:0.4800

## Median :0.2200 Median :0.5000 Median :0.0000 Median :0.5100

## Mean :0.2612 Mean :0.5047 Mean :0.0075 Mean :0.4999

## 3rd Qu.:0.3200 3rd Qu.:0.5000 3rd Qu.:0.0000 3rd Qu.:0.5300

## Max. :1.0000 Max. :1.0000 Max. :0.8300 Max. :0.7300

##

## V9 V10

## Min. :0.0000 CYT :463

## 1st Qu.:0.2200 NUC :429

## Median :0.2200 MIT :244

## Mean :0.2762 ME3 :163

## 3rd Qu.:0.3000 ME2 : 51

## Max. :1.0000 ME1 : 44

## (Other): 90

dim(yeast)

## [1] 1484 10

str(yeast)

## 'data.frame': 1484 obs. of 10 variables:

## $ V1 : Factor w/ 1462 levels "6P2K\_YEAST","6PGD\_YEAST",..: 33 34 35 3 5 4

6 101 7 8 ...

## $ V2 : num 0.58 0.43 0.64 0.58 0.42 0.51 0.5 0.48 0.55 0.4 ...

## $ V3 : num 0.61 0.67 0.62 0.44 0.44 0.4 0.54 0.45 0.5 0.39 ...

## $ V4 : num 0.47 0.48 0.49 0.57 0.48 0.56 0.48 0.59 0.66 0.6 ...

## $ V5 : num 0.13 0.27 0.15 0.13 0.54 0.17 0.65 0.2 0.36 0.15 ...

## $ V6 : num 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...

## $ V7 : num 0 0 0 0 0 0.5 0 0 0 0 ...

## $ V8 : num 0.48 0.53 0.53 0.54 0.48 0.49 0.53 0.58 0.49 0.58 ...

## $ V9 : num 0.22 0.22 0.22 0.22 0.22 0.22 0.22 0.34 0.22 0.3 ...

## $ V10: Factor w/ 10 levels "CYT","ERL","EXC",..: 7 7 7 8 7 1 7 8 7 1 ...

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.

2.1 --

## v ggplot2 3.0.0 v purrr 0.2.5

## v tibble 1.4.2 v dplyr 0.7.6

## v tidyr 0.8.1 v stringr 1.3.1

## v readr 1.1.1 v forcats 0.3.0

## -- Conflicts ------------------------------------------ tidyverse\_conflict

s() --

## x dplyr::filter() masks stats::filter()

## x dplyr::lag() masks stats::lag()

yeast <- read.table('https://archive.ics.uci.edu/ml/machine-learning-database

s/yeast/yeast.data', stringsAsFactors = FALSE)

l <- readLines('https://archive.ics.uci.edu/ml/machine-learning-databases/yea

st/yeast.names')

l <- l[(grep('^7', l) + 1):(grep('^8', l) - 1)]

l <- l[grep('\\d\\..\*:', l)]

names(yeast) <- make.names(c(sub('.\*\\d\\.\\s+(.\*):.\*', '\\1', l), 'class'))

str(yeast)

## 'data.frame': 1484 obs. of 10 variables:

## $ Sequence.Name: chr "ADT1\_YEAST" "ADT2\_YEAST" "ADT3\_YEAST" "AAR2\_YEAST"

...

## $ mcg : num 0.58 0.43 0.64 0.58 0.42 0.51 0.5 0.48 0.55 0.4 ...

## $ gvh : num 0.61 0.67 0.62 0.44 0.44 0.4 0.54 0.45 0.5 0.39 ...

## $ alm : num 0.47 0.48 0.49 0.57 0.48 0.56 0.48 0.59 0.66 0.6 ..

.

## $ mit : num 0.13 0.27 0.15 0.13 0.54 0.17 0.65 0.2 0.36 0.15 ..

.

## $ erl : num 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...

## $ pox : num 0 0 0 0 0 0.5 0 0 0 0 ...

## $ vac : num 0.48 0.53 0.53 0.54 0.48 0.49 0.53 0.58 0.49 0.58 .

..

## $ nuc : num 0.22 0.22 0.22 0.22 0.22 0.22 0.22 0.34 0.22 0.3 ..

.

## $ class : chr "MIT" "MIT" "MIT" "NUC" ...

pca <- princomp(yeast[, 2:9], cor=T) # principal components analysis using co

rrelation matrix

pc.comp <- pca$scores

PrincipalComponent1 <- -1\*pc.comp[,1] # principal component 1 scores (negated

for convenience)

PrincipalComponent2 <- -1\*pc.comp[,2] # principal component 2 scores (negated

for convenience)

clustering.data <- cbind(PrincipalComponent1, PrincipalComponent2)

# K-Mean Clustering

set.seed(100)

km <- kmeans(clustering.data, 8, iter.max = 30, nstart=30)

km

## K-means clustering with 8 clusters of sizes 3, 110, 192, 191, 399, 199, 26

0, 130

##

## Cluster means:

## PrincipalComponent1 PrincipalComponent2

## 1 3.6562743 -8.49636811

## 2 0.8085402 -1.96932237

## 3 1.0686111 1.39771690

## 4 -0.9430200 -1.09237604

## 5 0.2882285 0.01334731

## 6 1.6051406 -0.17329450

## 7 -0.7214210 0.76632446

## 8 -2.8601651 0.09471085

##

## Clustering vector:

## [1] 7 7 7 5 2 5 4 3 2 3 6 6 5 4 4 5 3 5 6 4 5 5 8 5 7 4 7 5 4 8 7 4 3 5

## [35] 8 4 8 3 4 4 8 8 8 7 6 6 4 5 7 3 5 3 5 5 4 4 7 5 5 4 5 6 5 3 7 7 3 5

## [69] 5 8 5 7 4 4 2 2 4 4 2 4 4 4 4 2 8 7 3 8 5 5 6 5 6 3 7 5 4 8 8 4 5 3

## [103] 2 6 6 7 3 5 7 5 5 7 6 5 3 2 4 2 2 6 8 4 4 4 7 4 6 4 5 6 5 5 2 4 5 3

## [137] 3 5 6 6 6 4 4 5 5 3 5 5 3 5 5 3 5 5 7 3 6 4 5 8 6 6 5 3 7 3 5 5 3 3

## [171] 6 7 5 6 4 5 6 5 6 2 6 5 5 4 7 5 5 3 7 3 8 8 5 4 4 4 4 8 8 5 7 6 5 4

## [205] 7 8 2 4 2 2 4 4 4 5 7 5 8 5 8 8 7 7 5 2 2 7 2 6 8 2 5 6 3 8 8 6 6 6

## [239] 5 5 5 5 5 6 3 5 7 6 7 3 5 7 6 5 7 2 6 6 6 5 6 6 7 7 6 4 6 5 7 7 6 5

## [273] 7 5 6 5 8 5 5 6 7 8 3 5 5 3 7 8 7 4 7 7 8 7 7 7 5 6 3 7 7 7 7 3 8 8

## [307] 3 5 7 7 6 7 7 7 3 7 3 3 8 7 7 6 8 3 7 8 8 5 5 2 3 4 7 8 5 6 7 3 7 2

## [341] 7 5 5 4 7 5 8 6 5 3 3 6 4 3 3 5 5 6 5 5 3 5 5 8 6 5 7 3 3 3 5 5 6 5

## [375] 5 5 3 7 7 6 5 6 5 3 3 5 5 7 5 5 4 6 7 2 6 7 3 5 3 4 6 6 2 7 5 3 6 6

## [409] 2 2 7 3 7 5 7 8 6 5 6 5 5 5 3 3 8 2 7 6 4 4 5 7 4 2 7 5 5 6 6 4 6 5

## [443] 7 7 3 7 5 7 7 3 5 4 4 5 4 4 5 4 4 4 5 4 5 2 5 5 6 5 7 3 8 4 7 7 7 2

## [477] 3 7 5 5 3 5 8 8 7 7 6 6 8 6 7 8 8 4 4 8 8 3 6 7 4 2 6 4 5 8 8 7 4 5

## [511] 5 8 8 8 8 6 7 2 3 6 4 4 3 3 7 3 5 5 4 5 5 3 5 3 5 4 4 3 7 5 6 3 6 6

## [545] 3 6 5 4 4 4 3 8 5 2 6 5 7 7 7 4 4 7 5 6 5 4 4 7 8 7 5 6 7 5 5 7 7 6

## [579] 2 4 8 2 4 5 5 3 6 5 7 4 7 5 5 8 8 8 4 7 4 7 3 4 2 2 5 2 5 2 2 4 6 4

## [613] 5 2 5 2 2 4 4 4 2 5 2 4 4 2 4 4 5 4 4 4 7 7 8 5 6 4 3 7 7 4 4 6 3 5

## [647] 7 5 4 4 5 4 2 5 5 5 6 6 5 7 5 2 7 4 8 7 5 4 6 5 4 6 2 6 5 4 8 7 7 4

## [681] 4 3 3 6 6 7 7 5 3 5 6 3 3 6 6 4 3 6 8 3 5 4 3 4 4 7 6 8 8 3 8 7 4 5

## [715] 2 5 3 3 3 8 6 6 5 7 5 5 5 4 5 5 7 4 4 5 5 5 5 7 5 8 3 4 3 6 4 8 5 5

## [749] 8 7 6 5 5 5 4 4 5 6 5 5 4 5 5 5 7 2 6 5 3 5 3 2 8 8 5 8 6 8 7 7 3 6

## [783] 3 2 4 3 4 4 4 4 5 8 7 3 3 7 8 8 3 7 3 7 5 5 6 7 3 6 7 5 7 2 5 7 4 5

## [817] 6 7 7 8 4 4 4 7 5 5 2 6 2 5 8 5 6 5 6 6 5 5 3 6 6 3 5 3 5 5 3 5 5 4

## [851] 4 5 8 5 6 3 2 6 5 4 5 5 7 5 5 4 7 5 5 7 6 4 3 5 6 6 5 3 5 5 5 3 3 5

## [885] 5 6 5 7 5 5 3 3 7 4 5 5 5 4 5 5 6 4 6 6 7 7 3 6 6 5 5 5 5 7 6 4 4 3

## [919] 2 5 4 5 7 4 5 5 2 2 6 6 5 2 2 2 2 2 2 6 7 6 5 5 5 5 3 5 6 6 6 5 5 2

## [953] 2 5 6 3 6 6 3 3 3 6 2 2 2 2 7 7 5 5 5 5 5 2 2 2 2 4 4 4 6 6 5 6 2 2

## [987] 2 2 1 1 1 5 5 2 6 5 7 7 7 7 8 5 5 7 4 6 2 2 6 6 5 6 6 6 4 4 5 5 5 5

## [1021] 5 4 2 3 4 4 6 6 7 2 6 6 2 7 5 6 8 5 3 8 5 7 4 5 4 4 4 4 5 7 8 7 6 3

## [1055] 7 5 3 7 7 7 7 5 5 5 7 5 8 7 7 3 7 3 5 7 6 5 8 7 6 3 5 3 5 4 8 8 7 4

## [1089] 7 7 2 5 5 5 2 5 3 3 3 5 3 5 7 2 3 3 5 8 7 6 5 5 3 7 6 7 3 5 4 3 3 5

## [1123] 3 3 5 2 2 6 7 4 7 7 2 6 5 6 2 2 5 7 5 5 4 4 5 7 3 5 6 3 5 7 5 4 3 3

## [1157] 7 6 5 7 7 2 5 6 5 6 5 3 5 5 3 7 5 5 5 5 5 5 6 4 5 3 2 5 3 8 5 5 8 5

## [1191] 3 7 2 7 3 8 6 7 7 2 6 7 5 5 3 7 7 8 8 4 5 6 5 3 5 5 6 6 6 4 6 5 3 3

## [1225] 5 8 3 7 3 7 5 7 7 5 5 5 5 4 5 4 5 6 7 7 5 5 8 5 7 7 5 3 3 5 4 6 6 6

## [1259] 6 3 6 6 3 7 7 8 5 8 8 7 7 5 6 5 3 3 5 5 5 6 2 6 2 6 4 5 3 3 7 5 5 7

## [1293] 6 5 7 5 7 4 5 8 5 5 5 5 4 2 6 7 7 5 7 7 7 5 2 7 6 3 3 3 7 6 5 5 5 7

## [1327] 5 2 2 7 5 7 7 5 5 8 6 8 5 7 5 4 2 7 3 5 7 5 6 4 4 7 5 5 8 7 8 6 8 3

## [1361] 7 7 7 8 8 7 5 5 5 8 3 5 5 4 6 3 3 7 6 4 7 3 3 3 3 7 8 6 5 7 7 7 8 8

## [1395] 7 4 2 4 7 8 7 3 8 7 5 7 5 5 8 5 4 3 7 4 5 3 7 3 5 7 8 7 6 8 8 3 7 5

## [1429] 7 5 3 5 8 7 8 3 8 8 7 8 8 3 8 8 7 7 2 3 3 7 8 6 3 6 4 3 8 5 7 3 4 8

## [1463] 4 3 3 7 3 2 5 5 4 5 7 4 5 4 5 2 2 8 6 7 3 7

##

## Within cluster sum of squares by cluster:

## [1] 3.998783 113.647111 145.595268 144.310502 126.152899 114.078257

## [7] 127.815144 149.922267

## (between\_SS / total\_SS = 79.8 %)

##

## Available components:

##

## [1] "cluster" "centers" "totss" "withinss"

## [5] "tot.withinss" "betweenss" "size" "iter"

## [9] "ifault"

km$cluster

## [1] 7 7 7 5 2 5 4 3 2 3 6 6 5 4 4 5 3 5 6 4 5 5 8 5 7 4 7 5 4 8 7 4 3 5

## [35] 8 4 8 3 4 4 8 8 8 7 6 6 4 5 7 3 5 3 5 5 4 4 7 5 5 4 5 6 5 3 7 7 3 5

## [69] 5 8 5 7 4 4 2 2 4 4 2 4 4 4 4 2 8 7 3 8 5 5 6 5 6 3 7 5 4 8 8 4 5 3

## [103] 2 6 6 7 3 5 7 5 5 7 6 5 3 2 4 2 2 6 8 4 4 4 7 4 6 4 5 6 5 5 2 4 5 3

## [137] 3 5 6 6 6 4 4 5 5 3 5 5 3 5 5 3 5 5 7 3 6 4 5 8 6 6 5 3 7 3 5 5 3 3

## [171] 6 7 5 6 4 5 6 5 6 2 6 5 5 4 7 5 5 3 7 3 8 8 5 4 4 4 4 8 8 5 7 6 5 4

## [205] 7 8 2 4 2 2 4 4 4 5 7 5 8 5 8 8 7 7 5 2 2 7 2 6 8 2 5 6 3 8 8 6 6 6

## [239] 5 5 5 5 5 6 3 5 7 6 7 3 5 7 6 5 7 2 6 6 6 5 6 6 7 7 6 4 6 5 7 7 6 5

## [273] 7 5 6 5 8 5 5 6 7 8 3 5 5 3 7 8 7 4 7 7 8 7 7 7 5 6 3 7 7 7 7 3 8 8

## [307] 3 5 7 7 6 7 7 7 3 7 3 3 8 7 7 6 8 3 7 8 8 5 5 2 3 4 7 8 5 6 7 3 7 2

## [341] 7 5 5 4 7 5 8 6 5 3 3 6 4 3 3 5 5 6 5 5 3 5 5 8 6 5 7 3 3 3 5 5 6 5

## [375] 5 5 3 7 7 6 5 6 5 3 3 5 5 7 5 5 4 6 7 2 6 7 3 5 3 4 6 6 2 7 5 3 6 6

## [409] 2 2 7 3 7 5 7 8 6 5 6 5 5 5 3 3 8 2 7 6 4 4 5 7 4 2 7 5 5 6 6 4 6 5

## [443] 7 7 3 7 5 7 7 3 5 4 4 5 4 4 5 4 4 4 5 4 5 2 5 5 6 5 7 3 8 4 7 7 7 2

## [477] 3 7 5 5 3 5 8 8 7 7 6 6 8 6 7 8 8 4 4 8 8 3 6 7 4 2 6 4 5 8 8 7 4 5

## [511] 5 8 8 8 8 6 7 2 3 6 4 4 3 3 7 3 5 5 4 5 5 3 5 3 5 4 4 3 7 5 6 3 6 6

## [545] 3 6 5 4 4 4 3 8 5 2 6 5 7 7 7 4 4 7 5 6 5 4 4 7 8 7 5 6 7 5 5 7 7 6

## [579] 2 4 8 2 4 5 5 3 6 5 7 4 7 5 5 8 8 8 4 7 4 7 3 4 2 2 5 2 5 2 2 4 6 4

## [613] 5 2 5 2 2 4 4 4 2 5 2 4 4 2 4 4 5 4 4 4 7 7 8 5 6 4 3 7 7 4 4 6 3 5

## [647] 7 5 4 4 5 4 2 5 5 5 6 6 5 7 5 2 7 4 8 7 5 4 6 5 4 6 2 6 5 4 8 7 7 4

## [681] 4 3 3 6 6 7 7 5 3 5 6 3 3 6 6 4 3 6 8 3 5 4 3 4 4 7 6 8 8 3 8 7 4 5

## [715] 2 5 3 3 3 8 6 6 5 7 5 5 5 4 5 5 7 4 4 5 5 5 5 7 5 8 3 4 3 6 4 8 5 5

## [749] 8 7 6 5 5 5 4 4 5 6 5 5 4 5 5 5 7 2 6 5 3 5 3 2 8 8 5 8 6 8 7 7 3 6

## [783] 3 2 4 3 4 4 4 4 5 8 7 3 3 7 8 8 3 7 3 7 5 5 6 7 3 6 7 5 7 2 5 7 4 5

## [817] 6 7 7 8 4 4 4 7 5 5 2 6 2 5 8 5 6 5 6 6 5 5 3 6 6 3 5 3 5 5 3 5 5 4

## [851] 4 5 8 5 6 3 2 6 5 4 5 5 7 5 5 4 7 5 5 7 6 4 3 5 6 6 5 3 5 5 5 3 3 5

## [885] 5 6 5 7 5 5 3 3 7 4 5 5 5 4 5 5 6 4 6 6 7 7 3 6 6 5 5 5 5 7 6 4 4 3

## [919] 2 5 4 5 7 4 5 5 2 2 6 6 5 2 2 2 2 2 2 6 7 6 5 5 5 5 3 5 6 6 6 5 5 2

## [953] 2 5 6 3 6 6 3 3 3 6 2 2 2 2 7 7 5 5 5 5 5 2 2 2 2 4 4 4 6 6 5 6 2 2

## [987] 2 2 1 1 1 5 5 2 6 5 7 7 7 7 8 5 5 7 4 6 2 2 6 6 5 6 6 6 4 4 5 5 5 5

## [1021] 5 4 2 3 4 4 6 6 7 2 6 6 2 7 5 6 8 5 3 8 5 7 4 5 4 4 4 4 5 7 8 7 6 3

## [1055] 7 5 3 7 7 7 7 5 5 5 7 5 8 7 7 3 7 3 5 7 6 5 8 7 6 3 5 3 5 4 8 8 7 4

## [1089] 7 7 2 5 5 5 2 5 3 3 3 5 3 5 7 2 3 3 5 8 7 6 5 5 3 7 6 7 3 5 4 3 3 5

## [1123] 3 3 5 2 2 6 7 4 7 7 2 6 5 6 2 2 5 7 5 5 4 4 5 7 3 5 6 3 5 7 5 4 3 3

## [1157] 7 6 5 7 7 2 5 6 5 6 5 3 5 5 3 7 5 5 5 5 5 5 6 4 5 3 2 5 3 8 5 5 8 5

## [1191] 3 7 2 7 3 8 6 7 7 2 6 7 5 5 3 7 7 8 8 4 5 6 5 3 5 5 6 6 6 4 6 5 3 3

## [1225] 5 8 3 7 3 7 5 7 7 5 5 5 5 4 5 4 5 6 7 7 5 5 8 5 7 7 5 3 3 5 4 6 6 6

## [1259] 6 3 6 6 3 7 7 8 5 8 8 7 7 5 6 5 3 3 5 5 5 6 2 6 2 6 4 5 3 3 7 5 5 7

## [1293] 6 5 7 5 7 4 5 8 5 5 5 5 4 2 6 7 7 5 7 7 7 5 2 7 6 3 3 3 7 6 5 5 5 7

## [1327] 5 2 2 7 5 7 7 5 5 8 6 8 5 7 5 4 2 7 3 5 7 5 6 4 4 7 5 5 8 7 8 6 8 3

## [1361] 7 7 7 8 8 7 5 5 5 8 3 5 5 4 6 3 3 7 6 4 7 3 3 3 3 7 8 6 5 7 7 7 8 8

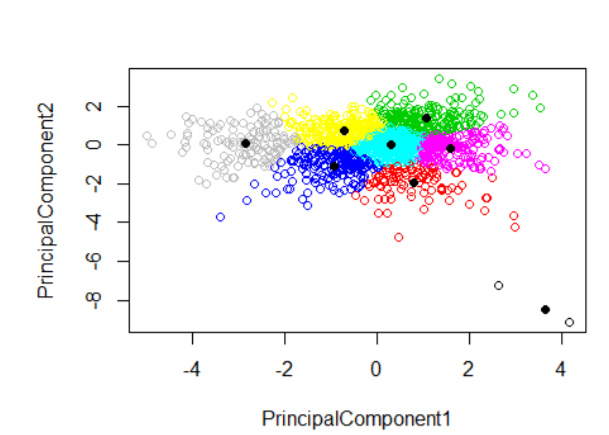
## [1395] 7 4 2 4 7 8 7 3 8 7 5 7 5 5 8 5 4 3 7 4 5 3 7 3 5 7 8 7 6 8 8 3 7 5

## [1429] 7 5 3 5 8 7 8 3 8 8 7 8 8 3 8 8 7 7 2 3 3 7 8 6 3 6 4 3 8 5 7 3 4 8

## [1463] 4 3 3 7 3 2 5 5 4 5 7 4 5 4 5 2 2 8 6 7 3 7

plot(PrincipalComponent1, PrincipalComponent2, col=km$cluster)

points(km$centers, pch=16)

 names(yeast)<- c("SequenceName", "mcg", "gvh", "alm", "mit", "erl", "pox", "v

ac", "nuc", "LocalizationSite")

aggregate(yeast[, 2:9],by=list(km$cluster),mean)

## Group.1 mcg gvh alm mit erl pox

## 1 1 0.3766667 0.2133333 0.9300000 0.7966667 0.5000000 0.000000000

## 2 2 0.4693636 0.4452727 0.5797273 0.3632727 0.5000000 0.004545455

## 3 3 0.3833333 0.4115104 0.4686458 0.1800000 0.5052083 0.000000000

## 4 4 0.5817277 0.5768063 0.5130366 0.4321466 0.5026178 0.004345550

## 5 5 0.4792231 0.4787719 0.5196992 0.2337343 0.5000000 0.012080201

## 6 6 0.3757286 0.3686935 0.5618593 0.2151759 0.5000000 0.004170854

## 7 7 0.5357692 0.5591154 0.4424231 0.2018462 0.5096154 0.012769231

## 8 8 0.7648462 0.7179231 0.4101538 0.3045385 0.5230769 0.006384615

## vac nuc

## 1 0.1600000 0.006666667

## 2 0.4034545 0.215727273

## 3 0.5272396 0.408750000

## 4 0.4853927 0.240471204

## 5 0.5055138 0.259548872

## 6 0.4818090 0.276532663

## 7 0.5303462 0.273076923

## 8 0.5196923 0.247153846

table(km$cluster, yeast$LocalizationSite)

##

## CYT ERL EXC ME1 ME2 ME3 MIT NUC POX VAC

## 1 3 0 0 0 0 0 0 0 0 0

## 2 48 0 2 0 0 0 32 26 1 1

## 3 45 0 0 0 4 49 3 88 0 3

## 4 36 0 12 1 3 2 113 21 1 2

## 5 179 0 1 0 0 25 48 130 10 6

## 6 76 0 0 0 0 3 11 105 1 3

## 7 73 0 0 0 14 78 23 57 5 10

## 8 3 5 20 43 30 6 14 2 2 5

#Spectral Clustering

library(kknn)

## Warning: package 'kknn' was built under R version 3.5.1

cl <- specClust(clustering.data, centers=8, nn=50, iter.max=100)

cl

## K-means clustering with 8 clusters of sizes 186, 219, 195, 156, 172, 161,

235, 160

##

## Cluster means:

## [,1] [,2] [,3] [,4] [,5] [,6]

## 1 -0.3808008 -0.010307326 -0.335345170 -0.34971429 -0.12569821 0.12122808

## 2 -0.3859207 -0.356402209 0.001120503 -0.18897403 0.15872031 0.42663259

## 3 -0.3490415 0.263465580 -0.365421550 0.18599020 -0.39027381 -0.23528531

## 4 -0.3706620 0.009057016 0.499097988 -0.08118804 0.25164079 -0.49508121

## 5 -0.3261405 0.473393062 0.201756081 0.43978983 0.28173634 0.29245070

## 6 -0.3113686 -0.308406853 -0.445304695 0.27223211 0.38520253 -0.27321926

## 7 -0.3971609 0.303283449 0.099096132 -0.38195357 -0.03938998 -0.02858538

## 8 -0.3253228 -0.481735595 0.380420920 0.30376846 -0.43922822 0.00971019

## [,7] [,8]

## 1 0.2889526 0.51087588

## 2 0.1959341 -0.34792688

## 3 0.1942592 -0.29581023

## 4 0.2549182 -0.02314869

## 5 0.1178271 0.14336792

## 6 -0.3357884 -0.03074654

## 7 -0.5215953 -0.04244720

## 8 -0.1457893 0.17068547

##

## Clustering vector:

## [1] 2 2 4 7 6 1 2 7 6 5 7 3 1 2 2 7 5 7 3 2 7 7 8 7 4 2 4 1 2 8 4 6 5 1

## [35] 8 8 8 7 2 2 8 8 8 4 3 3 6 1 8 5 7 5 1 1 2 2 2 2 1 6 1 3 7 7 4 4 5 7

## [69] 7 8 7 4 6 6 6 6 2 2 6 6 1 2 2 6 8 5 5 8 1 1 3 7 3 5 2 1 8 8 8 6 2 7

## [103] 6 3 1 4 7 1 4 7 1 7 5 1 5 6 6 6 6 3 8 2 2 2 4 2 3 2 1 3 2 1 6 2 7 7

## [137] 5 7 3 3 3 2 2 1 7 5 1 7 5 1 7 5 7 7 4 3 3 8 1 8 3 3 7 5 4 5 7 7 7 5

## [171] 3 2 2 3 2 1 3 1 3 6 3 1 1 2 2 7 7 5 2 5 8 8 7 2 2 6 6 8 8 7 2 3 2 6

## [205] 4 8 6 6 6 6 6 2 8 2 4 7 8 1 8 8 4 4 2 6 6 4 6 3 8 6 7 1 5 8 8 1 3 3

## [239] 1 1 7 1 1 3 5 1 2 3 4 5 1 4 3 1 4 6 3 3 3 7 3 3 2 4 3 2 3 7 4 4 3 1

## [273] 5 2 3 1 8 7 2 3 5 8 5 2 2 5 8 8 4 2 4 4 8 4 8 4 7 3 7 4 4 2 5 5 8 8

## [307] 5 7 2 4 3 2 2 2 5 2 5 5 8 2 4 3 8 5 4 8 8 7 1 6 5 8 4 8 2 3 2 5 4 6

## [341] 4 7 1 2 4 1 8 3 1 5 7 3 2 5 5 7 7 3 1 1 5 1 1 8 3 7 4 5 5 7 7 7 3 7

## [375] 7 7 5 4 4 3 2 3 1 5 5 1 1 7 7 1 2 3 4 6 3 4 5 7 5 2 3 3 6 4 7 5 3 3

## [409] 6 6 4 5 4 7 4 8 3 1 3 1 7 7 5 5 8 6 4 3 2 2 7 4 2 6 4 1 1 3 3 2 3 1

## [443] 7 7 5 4 7 4 4 5 1 2 6 1 2 2 7 2 6 6 1 2 7 6 7 1 3 1 4 5 8 2 4 4 4 6

## [477] 5 4 1 1 5 7 8 8 8 4 3 3 8 3 4 8 8 6 2 8 8 7 3 7 2 6 3 6 3 8 8 4 6 7

## [511] 1 8 8 8 8 3 7 6 5 3 6 2 3 7 4 5 2 7 6 7 2 5 7 5 7 6 2 5 4 1 3 5 3 3

## [545] 5 3 1 6 2 8 7 8 7 6 3 7 2 4 2 6 2 8 7 3 2 2 2 4 8 4 2 1 4 1 7 2 2 3

## [579] 6 8 8 6 2 1 7 5 3 7 4 2 4 1 1 8 8 8 2 7 6 2 5 2 6 6 1 6 1 6 6 2 3 6

## [613] 1 6 1 6 6 2 2 2 6 1 6 6 6 6 2 6 1 2 2 2 2 2 8 2 3 2 5 2 2 6 2 3 7 7

## [647] 2 1 6 2 1 2 6 1 7 7 3 3 7 4 7 6 4 2 8 4 1 6 3 1 2 7 1 3 7 2 8 4 2 2

## [681] 2 5 5 3 3 5 4 7 5 7 3 5 5 3 3 8 5 3 8 5 7 8 7 2 6 2 3 8 8 5 8 4 6 1

## [715] 6 7 5 5 5 8 3 3 1 2 2 1 1 6 1 7 4 2 6 7 1 7 1 4 7 8 5 2 5 3 2 8 7 7

## [749] 8 7 3 2 1 2 8 6 1 3 1 2 6 1 7 2 5 6 3 1 4 7 5 6 8 8 7 8 3 8 4 4 5 3

## [783] 5 6 2 5 6 6 6 6 2 8 4 5 5 7 8 8 5 5 7 4 7 7 3 7 7 3 4 1 4 6 1 8 2 7

## [817] 3 2 8 8 2 2 2 2 1 1 6 3 6 7 8 7 3 1 3 3 7 1 7 3 3 5 1 5 1 7 7 1 2 2

## [851] 2 1 8 1 3 5 6 3 7 2 2 7 4 1 7 2 2 1 2 2 3 2 5 1 3 3 7 5 1 1 1 5 5 1

## [885] 2 3 1 2 1 1 5 5 5 2 7 7 1 2 1 1 3 2 3 3 7 4 5 3 3 1 7 1 1 4 3 2 2 5

## [919] 6 1 2 1 5 2 1 1 6 3 3 3 1 6 6 6 6 6 6 1 4 3 7 7 1 7 7 2 3 3 3 1 1 6

## [953] 6 7 3 5 3 3 7 5 5 3 6 6 6 6 4 4 1 7 7 1 1 6 6 6 6 6 2 2 3 3 2 1 6 6

## [987] 6 6 6 6 6 1 1 6 3 1 4 4 4 8 8 7 2 4 2 3 6 6 3 3 1 3 3 3 1 1 1 1 1 1

## [1021] 1 2 6 5 2 2 3 3 7 6 3 3 6 4 1 3 8 2 5 8 7 5 2 7 6 6 8 2 7 4 8 4 3 5

## [1055] 7 7 7 4 4 4 4 7 1 7 4 7 8 4 8 7 4 5 2 4 3 7 8 8 3 5 1 5 7 2 8 8 4 2

## [1089] 7 8 6 1 1 2 6 1 5 5 5 7 7 7 4 6 5 5 7 8 4 3 7 7 5 4 3 4 5 2 2 5 5 7

## [1123] 5 7 1 6 6 3 4 2 4 4 6 3 1 3 6 6 7 4 1 7 2 6 7 7 5 7 3 5 1 4 2 2 7 5

## [1157] 7 7 7 4 2 6 7 3 7 3 1 5 1 7 5 4 7 7 1 7 7 7 3 2 7 5 6 7 5 8 7 7 8 1

## [1191] 5 4 6 4 5 8 3 4 2 6 3 4 7 7 3 4 8 8 8 8 1 3 2 5 7 7 3 3 3 2 3 7 5 5

## [1225] 7 8 7 2 5 7 1 7 7 7 2 7 1 2 1 2 7 3 7 7 7 7 8 1 4 4 1 5 5 1 2 3 3 3

## [1259] 3 7 3 3 5 4 4 8 1 8 8 2 2 7 3 1 5 5 1 2 1 3 6 3 3 3 2 1 5 5 2 7 7 2

## [1293] 3 7 4 1 2 2 2 8 1 1 1 7 8 6 3 2 4 2 4 4 4 7 6 4 3 5 5 5 4 3 7 1 7 7

## [1327] 7 6 6 5 1 4 4 7 7 8 3 8 7 2 1 2 6 4 5 7 8 7 3 2 8 7 1 7 8 4 8 3 8 5

## [1361] 4 2 4 8 8 7 7 2 7 8 5 7 1 6 3 5 5 4 3 2 4 5 5 5 5 4 8 3 7 4 4 8 8 8

## [1395] 4 6 6 6 4 8 7 5 8 4 1 4 7 7 8 1 8 5 7 2 7 5 2 5 7 4 8 4 3 8 8 5 4 7

## [1429] 2 1 5 2 8 4 8 5 8 8 4 8 8 5 8 8 2 4 6 5 5 2 8 3 5 3 6 5 8 7 2 5 2 8

## [1463] 6 5 5 4 5 6 7 1 2 7 4 2 1 2 2 6 6 8 3 8 5 2

##

## Within cluster sum of squares by cluster:

## [1] 45.59679 70.44780 60.00491 36.33080 40.60411 32.00669 74.81030 29.0714

8

## (between\_SS / total\_SS = 69.9 %)

##

## Available components:

##

## [1] "cluster" "centers" "totss" "withinss"

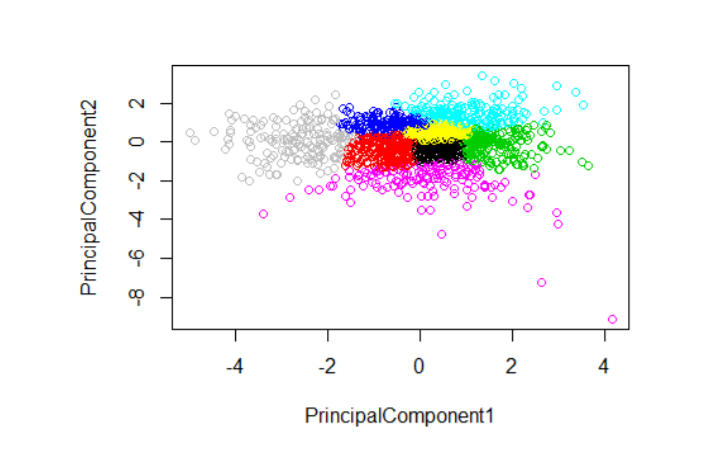
## [5] "tot.withinss" "betweenss" "size" "iter"

## [9] "ifault" "eigenvalue" "eigenvector" "data"

## [13] "indAll" "indUnique" "L" "archetype"

## [17] "call"

plot(PrincipalComponent1, PrincipalComponent2, col=cl$cluster)



table(cl$cluster, yeast$LocalizationSite)

##

## CYT ERL EXC ME1 ME2 ME3 MIT NUC POX VAC

## 1 71 0 1 0 0 3 31 70 9 1

## 2 72 0 2 0 2 11 93 33 2 4

## 3 74 0 0 0 1 3 11 102 1 3

## 4 42 0 0 0 10 55 8 34 2 5

## 5 35 0 0 0 2 50 2 80 0 3

## 6 52 0 7 1 2 0 67 29 2 1

## 7 110 0 0 0 1 29 12 75 2 6

## 8 7 5 25 43 33 12 20 6 2 7

aggregate(yeast[, 2:9],by=list(cl$cluster),mean)

## Group.1 mcg gvh alm mit erl pox

## 1 1 0.4755914 0.4774731 0.5415054 0.2611290 0.5000000 0.018763441

## 2 2 0.5599087 0.5620548 0.5038813 0.3309132 0.5022831 0.013652968

## 3 3 0.3738462 0.3676923 0.5625641 0.2140000 0.5000000 0.004256410

## 4 4 0.5244231 0.5530769 0.4301282 0.1937821 0.5096154 0.005320513

## 5 5 0.3816860 0.4130233 0.4589535 0.1778488 0.5087209 0.000000000

## 6 6 0.5042236 0.4885714 0.5668323 0.4272671 0.5000000 0.003105590

## 7 7 0.4680426 0.4667660 0.4985957 0.2000000 0.5000000 0.007063830

## 8 8 0.7473125 0.7039375 0.4175625 0.3013750 0.5218750 0.005187500

## vac nuc

## 1 0.4927957 0.2497312

## 2 0.5059817 0.2422831

## 3 0.4819487 0.2754359

## 4 0.5362179 0.2767949

## 5 0.5279651 0.4241860

## 6 0.4114907 0.2188199

## 7 0.5174043 0.2804681

## 8 0.5192500 0.2461250

#Hierarchical Clustering

d\_yeast<- dist(clustering.data)

hclusters <- hclust(d\_yeast, method = "average")

clusterCut <- cutree(hclusters, 8)

clusterCut

## [1] 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 2 3 2 1 1

## [35] 2 2 2 1 1 1 2 2 2 1 1 1 4 1 2 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

## [69] 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 3 1 2 1 1 1 1 1 3 1 1 2 2 2 1 1 1

## [103] 1 1 1 1 1 1 2 1 1 1 1 1 3 1 2 1 5 1 2 1 2 1 2 1 1 2 1 1 1 1 1 1 1 1

## [137] 1 1 1 1 1 1 1 1 1 3 1 1 3 1 1 3 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 3

## [171] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 2 1 1 2 2 1 1 1 1 1

## [205] 2 2 1 1 1 1 1 1 2 1 1 1 2 1 2 2 2 1 1 1 1 1 1 1 2 1 1 1 3 2 2 1 1 1

## [239] 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

## [273] 3 1 1 1 2 1 1 1 1 2 1 1 1 1 2 2 1 1 1 1 2 1 2 1 1 1 1 1 2 1 1 1 2 2

## [307] 3 1 1 2 1 1 1 1 1 1 1 3 2 1 1 1 2 3 1 2 2 1 1 1 1 2 1 2 1 1 1 1 1 1

## [341] 1 1 1 1 1 1 2 1 1 1 1 1 1 1 3 1 1 1 1 1 3 1 1 2 1 1 1 1 1 1 1 1 1 1

## [375] 1 1 3 1 1 1 1 1 1 3 1 1 1 1 1 1 2 1 1 5 1 2 3 1 1 2 1 1 1 1 1 1 1 1

## [409] 5 5 1 1 1 1 2 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1

## [443] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 5

## [477] 1 2 1 1 1 1 2 2 2 1 1 1 2 1 1 2 2 1 1 2 2 1 1 1 1 1 1 4 1 2 2 1 4 1

## [511] 1 2 2 2 2 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 3 1 3 1 1 1 1 1 1 1 1 1 1

## [545] 1 1 1 1 1 2 1 2 1 5 1 1 1 2 1 1 1 2 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1

## [579] 1 4 2 1 1 1 1 3 1 1 2 2 1 1 1 2 2 2 1 1 1 1 3 2 1 1 1 1 1 1 1 1 1 1

## [613] 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1

## [647] 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1

## [681] 1 3 1 1 1 1 1 1 3 1 1 3 3 1 1 2 1 1 2 1 1 2 1 2 4 1 1 2 2 1 2 2 2 1

## [715] 1 1 3 3 3 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 3 1 1 1 1 2 1 1

## [749] 2 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 3 1 1 1 1 1 3 1 2 2 1 2 1 2 2 1 1 1

## [783] 3 1 2 1 1 4 4 4 1 2 1 3 1 1 2 2 3 1 1 1 1 1 1 1 1 1 1 1 1 5 1 2 1 1

## [817] 1 1 2 2 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 3 1 3 1 1 1 1 1 1

## [851] 1 1 2 1 1 3 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 3 3 1

## [885] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3

## [919] 1 1 1 1 1 2 1 1 5 1 1 1 1 1 1 1 1 5 5 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

## [953] 1 1 1 1 1 1 1 3 3 1 5 5 1 1 1 1 1 1 1 1 1 5 1 1 1 1 2 2 1 1 1 1 1 1

## [987] 5 5 6 6 6 1 1 1 1 1 1 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

## [1021] 1 1 5 3 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 2 1 2 1 1 1 2 1 1 1

## [1055] 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 2 2 1 3 1 3 1 1 2 2 1 1

## [1089] 1 2 5 1 1 1 1 1 3 1 3 1 1 1 1 1 1 3 1 2 2 7 1 1 1 1 1 1 3 1 1 1 1 1

## [1123] 3 1 1 8 1 1 1 1 1 1 1 1 1 1 5 5 1 1 1 1 2 4 1 1 3 1 1 3 1 1 1 1 1 3

## [1157] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 3 5 1 1 2 1 1 2 1

## [1191] 1 1 5 1 3 2 1 1 1 1 7 1 1 1 1 1 2 2 2 2 1 1 1 3 1 1 1 1 1 1 1 1 3 3

## [1225] 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 3 1 1 1 1 1

## [1259] 1 1 1 1 1 1 1 2 1 2 2 1 1 1 1 1 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

## [1293] 1 1 2 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 5 1 1 3 1 1 1 1 1 1 1 1

## [1327] 1 5 5 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 2 1 2 1 2 1

## [1361] 1 1 1 2 2 1 1 1 1 2 1 1 1 1 1 3 3 1 1 1 1 1 3 1 1 1 2 1 1 1 1 2 2 2

## [1395] 1 1 1 1 1 2 1 1 2 2 1 1 1 1 2 1 2 1 1 1 1 3 1 1 1 1 2 2 1 2 2 3 1 1

## [1429] 1 1 3 1 2 1 2 3 2 2 1 2 2 1 2 2 1 1 1 3 3 1 2 1 3 1 4 3 2 1 1 3 1 2

## [1463] 1 1 3 1 3 1 1 1 2 1 1 1 1 1 1 5 5 2 1 2 1 1

table(clusterCut, yeast$LocalizationSite)

##

## clusterCut CYT ERL EXC ME1 ME2 ME3 MIT NUC POX VAC

## 1 411 0 4 0 14 130 194 358 17 21

## 2 16 5 25 43 36 19 46 10 3 9

## 3 16 0 0 0 0 14 2 49 0 0

## 4 0 0 6 1 1 0 0 2 0 0

## 5 17 0 0 0 0 0 1 8 0 0

## 6 3 0 0 0 0 0 0 0 0 0

## 7 0 0 0 0 0 0 1 1 0 0

## 8 0 0 0 0 0 0 0 1 0 0

aggregate(yeast[, 2:9],by=list(clusterCut),mean)

## Group.1 mcg gvh alm mit erl pox

## 1 1 0.4715405 0.4762228 0.5104178 0.2522715 0.5013055 0.008964317

## 2 2 0.7120283 0.6790566 0.4296698 0.3211792 0.5188679 0.003915094

## 3 3 0.3545679 0.3871605 0.4807407 0.1697531 0.5185185 0.000000000

## 4 4 0.7750000 0.7390000 0.5210000 0.4280000 0.5000000 0.000000000

## 5 5 0.4115385 0.4076923 0.5992308 0.3080769 0.5000000 0.000000000

## 6 6 0.3766667 0.2133333 0.9300000 0.7966667 0.5000000 0.000000000

## 7 7 0.2350000 0.1700000 0.7000000 0.3100000 0.5000000 0.000000000

## 8 8 0.6600000 0.4300000 0.5700000 0.6000000 0.5000000 0.000000000

## vac nuc

## 1 0.5007659 0.268398607

## 2 0.5205189 0.248867925

## 3 0.5237037 0.497530864

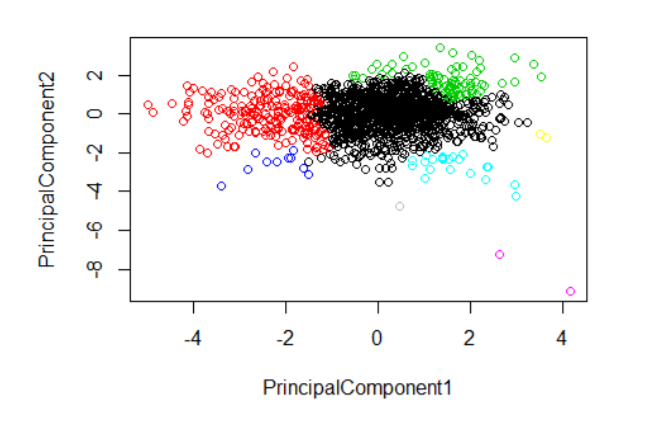
## 4 0.3660000 0.241000000

## 5 0.3219231 0.200384615

## 6 0.1600000 0.006666667

## 7 0.4900000 0.230000000

## 8 0.1900000 0.330000000

plot(PrincipalComponent1, PrincipalComponent2, col=clusterCut)  # Show a random sample

set.seed(1234)

dplyr::sample\_n(yeast, 10)

## SequenceName mcg gvh alm mit erl pox vac nuc LocalizationSite

## 169 CHS2\_YEAST 0.39 0.42 0.38 0.40 0.5 0 0.49 0.47 ME3

## 923 RNA1\_YEAST 0.45 0.52 0.50 0.12 0.5 0 0.60 0.22 CYT

## 903 R104\_YEAST 0.44 0.33 0.55 0.16 0.5 0 0.49 0.22 NUC

## 924 RN12\_YEAST 0.56 0.51 0.32 0.49 0.5 0 0.48 0.22 NUC

## 1275 TOP1\_YEAST 0.41 0.42 0.53 0.17 0.5 0 0.48 0.58 NUC

## 948 RPB5\_YEAST 0.40 0.30 0.57 0.13 0.5 0 0.46 0.22 NUC

## 15 ACR1\_YEAST 0.66 0.55 0.45 0.19 0.5 0 0.46 0.22 MIT

## 344 GAL8\_YEAST 0.60 0.60 0.49 0.30 0.5 0 0.53 0.22 NUC

## 984 RL34\_YEAST 0.38 0.43 0.53 0.22 0.5 0 0.48 0.11 CYT

## 759 PT91\_YEAST 0.59 0.45 0.58 0.21 0.5 0 0.49 0.22 MIT

# Show the levels

levels(yeast$group)

## NULL

library(dplyr)

group\_by(yeast, SequenceName) %>%

summarise(

count = n(),

mean = mean(nuc, na.rm = TRUE),

sd = sd(nuc, na.rm = TRUE)

)

## # A tibble: 1,462 x 4

## SequenceName count mean sd

## <chr> <int> <dbl> <dbl>

## 1 6P2K\_YEAST 1 0.3 NaN

## 2 6PGD\_YEAST 1 0.31 NaN

## 3 AAR2\_YEAST 1 0.22 NaN

## 4 AATC\_YEAST 1 0.22 NaN

## 5 AATM\_YEAST 1 0.22 NaN

## 6 ABC1\_YEAST 1 0.22 NaN

## 7 ABF2\_YEAST 1 0.22 NaN

## 8 ABP1\_YEAST 1 0.3 NaN

## 9 ACE1\_YEAST 1 0.27 NaN

## 10 ACE2\_YEAST 1 0.290 NaN

## # ... with 1,452 more rows

c. Perform ANOVA test on the discriminant analysis scores of

nuclear localization signals of both nuclear and non-nuclear

proteins by class variables (Target).

d. Which class is significantly different from others?

# Compute the analysis of variance

res.aov<-aov(nuc~ LocalizationSite,data=yeast)

summary(res.aov)

## Df Sum Sq Mean Sq F value Pr(>F)

## LocalizationSite 9 1.993 0.22141 22.01 <2e-16 \*\*\*

## Residuals 1474 14.825 0.01006

## ---

## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

res.aov <- aov(nuc ~ vac, data = yeast)

# Summary of the analysis

summary(res.aov)

## Df Sum Sq Mean Sq F value Pr(>F)

## vac 1 0.135 0.13529 12.02 0.000542 \*\*\*

## Residuals 1482 16.682 0.01126

## ---

## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

res.aov <- aov(nuc ~ pox, data = yeast)

summary(res.aov)

## Df Sum Sq Mean Sq F value Pr(>F)

## pox 1 0.021 0.02138 1.887 0.17

## Residuals 1482 16.796 0.01133

res.aov <- aov(nuc ~ erl, data = yeast)

summary(res.aov)

## Df Sum Sq Mean Sq F value Pr(>F)

## erl 1 0.00 0.000135 0.012 0.913

## Residuals 1482 16.82 0.011348

res.aov <- aov(nuc ~mit , data = yeast)

summary(res.aov)

## Df Sum Sq Mean Sq F value Pr(>F)

## mit 1 0.05 0.05050 4.463 0.0348 \*

## Residuals 1482 16.77 0.01131

## ---

## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

res.aov <- aov(nuc ~alm , data = yeast)

summary(res.aov)

## Df Sum Sq Mean Sq F value Pr(>F)

## alm 1 0.008 0.008171 0.72 0.396

## Residuals 1482 16.809 0.011342

res.aov <- aov(nuc ~gvh , data = yeast)

summary(res.aov)

## Df Sum Sq Mean Sq F value Pr(>F)

## gvh 1 0.178 0.17836 15.89 7.05e-05 \*\*\*

## Residuals 1482 16.639 0.01123

## ---

## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

res.aov <- aov(nuc ~ mcg, data = yeast)

summary(res.aov)

## Df Sum Sq Mean Sq F value Pr(>F)

## mcg 1 0.261 0.26085 23.35 1.49e-06 \*\*\*

## Residuals 1482 16.557 0.01117

## ---

## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1