```
#Pre processamento
install.packages("dplyr")
x_train<-read.csv(file="C:/Users/maria.s.matias/Desktop/MestradoEachUsp/SIN5007_Ver2_Turma1_2017_ET/Trabalho/x_train.csv",header=T,sep=",")
library(dplyr)
newdata<-
group_by(x_train,GeneId)%>%summarise(H3K4me3=sum(H3K4me3),H3K4me1=sum(H3K4me1),H3K36me3=sum(H3K36me3),H3K9me3=sum(H3K9me3),H3K27me3=sum(H3K27me3))
write.csv(newdata, "C:/Users/maria.s.matias/Desktop/MestradoEachUsp/SIN5007_Ver2_Turma1_2017_ET/Trabalho/dadosTratados.csv", row.names = FALSE)
train<-read.csv(file="C:/Users/maria.s.matias/Desktop/MestradoEachUsp/SIN5007_Ver2_Turma1_2017_ET/Trabalho/dadosTratados.csv",header=T,sep=",")
View(train)

#Covariância
cov_train <- train
cov_train <- cov_train[,-c(1)]
result_cov <- cov (cov_train)
View(result_cov)
write.csv(result_cov, "C:/Users/maria.s.matias/Desktop/MestradoEachUsp/SIN5007_Ver2_Turma1_2017_ET/Trabalho/covariancia.csv", row.names = FALSE)

#Correlação
cor_train <- train
cor_train <- cor_train[,-c(1)]
result_cor <- cor (cor_train)
View(result_cor)
write.csv(result_cor, "C:/Users/maria.s.matias/Desktop/MestradoEachUsp/SIN5007_Ver2_Turma1_2017_ET/Trabalho/correlacao.csv", row.names = FALSE)

#PCA usando princomp com matriz de correlação
pca.cor<-princomp(train[,-c(1)], cor=TRUE, scores = TRUE)
biplot (pca.cor)
screeplot(pca.cor)
View(pca.cor)
write.csv(pca.cor$scores, "C:/Users/maria.s.matias/Desktop/MestradoEachUsp/SIN5007_Ver2_Turma1_2017_ET/Trabalho/pca_correlacao.csv", row.names = TRUE)
new_pca_cor <- read.csv(file="C:/Users/maria.s.matias/Desktop/MestradoEachUsp/SIN5007_Ver2_Turma1_2017_ET/Trabalho/pca_correlacao.csv",header=T,sep=",")
new_pca_cor <- new_pca_cor[,-c(5,6)]
names(new_pca_cor)[1] <- c("GeneID")
View(new_pca_cor)
write.csv(new_pca_cor, "C:/Users/maria.s.matias/Desktop/MestradoEachUsp/SIN5007_Ver2_Turma1_2017_ET/Trabalho/pcaCorTratado.csv", row.names = TRUE)

#PCA usando princomp com matriz de covariância
pca.cov<-princomp(train[,-c(1)], cor=FALSE, scores = TRUE)
biplot (pca.cov)
screeplot(pca.cov)
View(pca.cov)
write.csv(pca.cov$scores, "C:/Users/maria.s.matias/Desktop/MestradoEachUsp/SIN5007_Ver2_Turma1_2017_ET/Trabalho/pca_covariancia.csv", row.names = TRUE)
new_pca_cov <- read.csv(file="C:/Users/maria.s.matias/Desktop/MestradoEachUsp/SIN5007_Ver2_Turma1_2017_ET/Trabalho/pca_covariancia.csv",header=T,sep=",")
new_pca_cov <- new_pca_cov[,-c(3,4,5,6)]
names(new_pca_cov)[1] <- c("GeneID")
View(new_pca_cov)
write.csv(new_pca_cov, "C:/Users/maria.s.matias/Desktop/MestradoEachUsp/SIN5007_Ver2_Turma1_2017_ET/Trabalho/pcaCovTratado.csv", row.names = TRUE)

#Seleção de características
#aplicação Relief (descrição, dados, vizinhos à encontrar por instância, nº exemplos amostra )
library(FSelector)
classes<-read.csv("C:/Users/maria.s.matias/Desktop/MestradoEachUsp/SIN5007_Ver2_Turma1_2017_ET/Trabalho/y_train.csv")
dados_relief <- merge(train,classes)
View(dados_relief)
write.csv(dados_relief, "C:/Users/maria.s.matias/Desktop/MestradoEachUsp/SIN5007_Ver2_Turma1_2017_ET/Trabalho/dadosRelief.csv", row.names = FALSE)
dados_relief <- dados_relief[,-c(1)]

pesos<-relief(Prediction~., dados_relief, neighbours.count = 10, sample.size = 20)
View(pesos)
write.csv(pesos, "C:/Users/maria.s.matias/Desktop/MestradoEachUsp/SIN5007_Ver2_Turma1_2017_ET/Trabalho/pesosRelief.csv", row.names = FALSE)
subset <- cutoff.k(pesos,3)
View(subset)
write.csv(subset, "C:/Users/maria.s.matias/Desktop/MestradoEachUsp/SIN5007_Ver2_Turma1_2017_ET/Trabalho/subsetRelief.csv", row.names = FALSE)
result <- as.simple.formula(subset, "Prediction") #converte o nome dos atributos classificados
View(result)
new_relief <- train [,-c(3,5)]
View (new_relief)
write.csv(new_relief, "C:/Users/maria.s.matias/Desktop/MestradoEachUsp/SIN5007_Ver2_Turma1_2017_ET/Trabalho/resultRelief.csv", row.names = FALSE)

pesos<-relief(Prediction~., dados_relief, neighbours.count = 20, sample.size = 60)
View(pesos)
write.csv(pesos, "C:/Users/maria.s.matias/Desktop/MestradoEachUsp/SIN5007_Ver2_Turma1_2017_ET/Trabalho/pesos2Relief.csv", row.names = FALSE)
subset <- cutoff.k(pesos,3)
View(subset)
write.csv(subset, "C:/Users/maria.s.matias/Desktop/MestradoEachUsp/SIN5007_Ver2_Turma1_2017_ET/Trabalho/subset2Relief.csv", row.names = FALSE)
result <- as.simple.formula(subset, "Prediction") #converte o nome dos atributos classificados
View(result)
new_relief <- train [,-c(3,5)]
View (new_relief)
write.csv(new_relief, "C:/Users/maria.s.matias/Desktop/MestradoEachUsp/SIN5007_Ver2_Turma1_2017_ET/Trabalho/result2Relief.csv", row.names = FALSE)


#Estimativa erro - k-fold cross validation
#kfold = function(data,folds){
#  if (nrow(data) %% folds == 0) {
#    cnt <- 1
#    while (cnt <= folds) {
#
#
#      cnt = cnt + 1
#    }
#
#  }
#  else{
#
#  }
#
#
#}

#ordena pela classe. ou separa em dois conjuntos distintos um com uma classe e outro com outra.
#dai sorteia x de uma classe, x de outra e monta o k fold
#k-fold cross validation:
#o conjunto é dividido em k conjuntos iguais
#k validações cruzadas são feitas. cada uma com k-1 conjuntos de treinamento e 1 de teste

#faz os k folds e salva em arquivos porque isso precisa ser persistente

#seta o work directory
#path<-"C:/Users/lsantos/Desktop/reconhecimento"
path<-"C:/Users/Lara/Desktop/reconhecimento"
```

```
setwd(path)

#le o arquivo
dados_sem_classe<-read.csv("dadosTratados.csv")
rotulos<-read.csv("y_train.csv")

#unifica os arquivos
dadosRotulados<-merge(dados_sem_classe,rotulos)



#dados da classe 1
arrclasse1<-dadosRotulados[ which(dadosRotulados$Prediction==1) , ]
#dados classe 2
arrclasse2<-dadosRotulados[ which(dadosRotulados$Prediction==0) , ]

qtdClasse1<-dim(arrclasse1)[1]
qtdClasse2<-dim(arrclasse2)[1]
print(paste("quantidade de dados da classe 1 (cjto todo): ",qtdClasse1," quantidade de dados da classe 2 (cjto todo):",qtdClasse2 ))
print("#####")

################################################################################
#k fold
################################################################################
#divids em k partes iguais

k<-5
nrAmostrasClasse1<-qtdClasse1%/% k
restoClasse1<-qtdClasse1%%k

nrAmostrasClasse2<-qtdClasse2%/%k
restoClasse2<-qtdClasse2%%k

print(nrAmostrasClasse1)
print(nrAmostrasClasse2)

#cada classe tem x amostras da classe 1, x da classe 2
# e os dois restos são distribuidos entre as classes
print(restoClasse1)
print(restoClasse2)
inicio1<-1
inicio2<-1
fim1<-0
fim2<-0

#para cada fold
for(i in 1:k){
  #sorteia nrAmostrasClasse1 e nrAmostrasClasse2

  if(i!=1)inicio1<-fim1+1

  if(i!=1)inicio2<-fim2+1



  fim1<-inicio1+nrAmostrasClasse1-1
  fim2<-inicio2+nrAmostrasClasse2-1

  #if(i<restoClasse1){
  #fim1<-fim1+1
  #}


  #if(i<restoClasse2) fim2<-fim2+1

  itensC1<-arrclasse1[inicio1: fim1,]
  itensC2<-arrclasse2[inicio2: fim2,]
  print(paste("inicio1",inicio1,"ini2", inicio2))
  print(paste("fim1",(fim1),"fim2",(fim2)))

  combinado<-rbind(itensC1,itensC2)
  combinado<-combinado[sample(nrow(combinado)),]

  write.csv(combinado, paste("fold",i,".csv"), row.names = FALSE)
  # print(dim(arrclasse1[inicio1: inicio1+nrAmostrasClasse1-1+fim1,]))
  #se i <restoClasse1 sorteia mais uma da cla
  #sse1
  #se i< restoclasse2 sorteia mais um da classe2

  #print(paste("itensC1:",dim(itensC1)[1]," itensc2:",dim(itensC2)[1]))
}



#SVM
#svm para todos os dados, mas com k fold

#descomentar caso não tenha o package
#install.packages("e1071")
library(caret)
library("e1071")


path<-"C:/Users/Lara/Desktop/reconhecimento"
setwd(path)

k<-5
#le a fold 1
data1<-read.csv(paste("fold",1,".csv"))
#le a fold 2
data2<-read.csv(paste("fold",2,".csv"))
#le a fold 3
data3<-read.csv(paste("fold",3,".csv"))
#le a fold 4
data4<-read.csv(paste("fold",4,".csv"))
#le a fold 5
data5<-read.csv(paste("fold",5,".csv"))

################################################################################

#utilizando o fold 1 como teste
```

```
################################################################################

x <- subset(data1, select=-data1$Prediction)
y <- data1$Prediction

#unifica os arquivos
dadostreinamento<-rbind(data4,data2)
dadostreinamento<-rbind(dadostreinamento,data3)
dadostreinamentor<-rbind(dadostreinamento,data5)

#valido para todos com a configuracao 1
#svm(formula = Prediction ~ ., data = dadostreinamento, type = "C-classification")
#Parameters:
#   SVM-Type:  C-classification
#   SVM-Kernel:  radial
#   cost:  1
#   gamma:  0.1666667

################################################
###configuracao 1 - kernel radial, custo 1              #
################################################
#treinamento
svm_model <- svm(Prediction~.,data=dadostreinamento, type="C-classification")
summary(svm_model)
#teste, o data1 é a amostra de teste
pred <- predict(svm_model,data1)
#verificacao
confusionMatrix(y, pred)
#Confusion Matrix and Statistics
#
#          Reference
#Prediction    0    1
#         0  376 1169
#         1    6 1545
#treinamento

################################################
###configuracao 2 - kernel linear, custo 1              #
################################################
svm_model <- svm(Prediction~.,data=dadostreinamento, type="C-classification",kernel="linear")
summary(svm_model)
#teste, o data1 é a amostra de teste
pred <- predict(svm_model,data1)
#verificacao
confusionMatrix(y, pred)

#Confusion Matrix and Statistics
#
#          Reference
#Prediction    0    1
#         0 1014  531
#         1   64 1487


################################################
###configuracao 3 - kernel polinomial, custo 1          #
################################################
svm_model <- svm(Prediction~.,data=dadostreinamento, type="C-classification",kernel="polynomial")
summary(svm_model)
#teste, o data1 é a amostra de teste
pred <- predict(svm_model,data1)
#verificacao
confusionMatrix(y, pred)

#Confusion Matrix and Statistics
#
#          Reference
#Prediction    0    1
#         0  542 1003
#         1   18 1533

################################################
###configuracao 4 - kernel sigmoide, custo 1            #
################################################
svm_model <- svm(Prediction~.,data=dadostreinamento, type="C-classification",kernel="sigmoid")
summary(svm_model)
#teste, o data1 é a amostra de teste
pred <- predict(svm_model,data1)
#verificacao

#confusionMatrix(y, pred)
#
#       Reference
#Prediction    0    1
#         0  885  660
#         1  153 1398


#Redes Neurais - dados originais -- 1 camada oculta com 2 neurônios, retornando hessiana, número max de pesos = 10
library(nnet)
install.packages("confusionMatrix")
library(caret)
fold <-
read.csv(file="C:/Users/maria.s.matias/Desktop/MestradoEachUsp/SIN5007_Ver2_Turma1_2017_ET/Trabalho/Parciais/folds/foldstotal/fold_1.csv",header=T,sep=",")

View(fold)
result_nnet1 <- nnet(Prediction~., fold, rang=0.1,size=2, maxit=500, subset=NULL, Hess=TRUE, na.action=NULL, MaxNWts=1000,contrasts=NULL, linout = FALSE,
decay = seq(from = 0.1, to = 0.5, by = 0.1))
View(result_nnet1)
#result_nnet1 <- nnet(Prediction~.,fold,weights=NULL, size=2, Wts=100,rang = 1, decay = 0, maxit = 100, Hess = TRUE, trace = TRUE, MaxNWts = 1000, abstol
= 1.0e-4, reltol = 1.0e-8)
results1 <- predict(result_nnet1, newdata=fold)
conf1 <- confusionMatrix(results1, fold$Prediction)
table(factor(pred, levels=min(test):max(test)), factor(test, levels=min(test):max(test)))
table(fold$Prediction, predict(result_nnet1, fold, type = "class"))


#############################################################################Naive Bayes
#naivebayes em R

#install.packages("naivebayes")
```

```
library(naivebayes)

path<-"C:/Users/Lara/Desktop/reconhecimento"
setwd(path)

k<-5
#le a fold 1
data1<-read.csv(paste("fold",1,".csv"))
#le a fold 2
data2<-read.csv(paste("fold",2,".csv"))
#le a fold 3
data3<-read.csv(paste("fold",3,".csv"))
#le a fold 4
data4<-read.csv(paste("fold",4,".csv"))
#le a fold 5
data5<-read.csv(paste("fold",5,".csv"))



mysummary <- function(fold){


  dadosTeste<-read.csv (paste(fold,".csv"))

  dadosTreinamento <- data.frame(lapply(dadostreinamento, as.character), stringsAsFactors=FALSE)

  model<- naive_bayes(Prediction ~ ., data = dadosTreinamento)
  class(model)
  summary(model)

  preds <- predict(model, newdata = dadosTeste)

  conf_matrix <- table(preds, dadosTeste$Prediction)
  write.csv(conf_matrix,paste(fold,"confusao.csv"))
}

################################################################################################

#utilizando o fold 1 como teste
################################################################################################

x <- subset(data1, select=-data1$Prediction)
y <- data1$Prediction

#unifica os arquivos
dadostreinamento<-rbind(data4,data2)
dadostreinamento<-rbind(dadostreinamento,data3)
dadostreinamentor<-rbind(dadostreinamento,data5)

mysummary("fold 1")

################################################################################################

#utilizando o fold 2 como teste
################################################################################################

x <- subset(data2, select=-data2$Prediction)
y <- data2$Prediction

#unifica os arquivos
dadostreinamento<-rbind(data4,data1)
dadostreinamento<-rbind(dadostreinamento,data3)
dadostreinamentor<-rbind(dadostreinamento,data5)

mysummary("fold 2")
################################################################################################

#utilizando o fold 3 como teste
################################################################################################

x <- subset(data3, select=-data3$Prediction)
y <- data3$Prediction

#unifica os arquivos
dadostreinamento<-rbind(data4,data1)
dadostreinamento<-rbind(dadostreinamento,data2)
dadostreinamentor<-rbind(dadostreinamento,data5)

mysummary("fold 3")


################################################################################################

#utilizando o fold 4 como teste
################################################################################################

x <- subset(data4, select=-data4$Prediction)
y <- data4$Prediction

#unifica os arquivos
dadostreinamento<-rbind(data2,data1)
dadostreinamento<-rbind(dadostreinamento,data3)
dadostreinamentor<-rbind(dadostreinamento,data5)

mysummary("fold 4")

################################################################################################

#utilizando o fold 5 como teste
################################################################################################

x <- subset(data5, select=-data5$Prediction)
y <- data5$Prediction

#unifica os arquivos
dadostreinamento<-rbind(data4,data1)
dadostreinamento<-rbind(dadostreinamento,data3)
dadostreinamentor<-rbind(dadostreinamento,data2)

mysummary("fold 5")
```

```
#############################################################################
#Cálculo das médias naive bayes
fold,"confusao.csv"

#calcula a media

#para cada custo, 1, 10, 100
#le as folds de 1 a 5
#calcula as precisao, erro e sensibilidade e faz a media
#printa a media em um arquivo
path<-"C:/Users/Lara/Desktop/reconhecimento"
setwd(path)


calculamedia <- function() {

  ErroTotal<-0
  SensibilidadeTotal<-0
  PrecisaoTotal<-0
  for (linha in 1:5){

    print((paste("fold",linha,"confusao.csv")))
    result<-read.csv(paste("fold",linha,"confusao.csv"))
    TP<-result[1,"X0"]
    FP<-result[2,"X0"]
    FN<-result[1,"X1"]
    TN<-result[2,"X1"]
    #print(result)
    Precisao<-TP/(TP+FP)
    Sensibilidade<-TP/(TP+FN)
    Erro<-(FP+FN)/(TP+TN+FP+FN)

    ErroTotal<-ErroTotal+Erro
    SensibilidadeTotal<-SensibilidadeTotal+Sensibilidade
    PrecisaoTotal<-PrecisaoTotal+Precisao


  }
  log_con <- file(paste("media",".txt"))


  cat(paste(
    "Precisão md: ",PrecisaoTotal/5,"\n","Erro md: ",ErroTotal/5,"\n","Sensibilidade md: ",SensibilidadeTotal/5),file=log_con)




}
calculamedia()

#############################################################################
```