BISC 577a

Kalinga Pavan Thushara Silva

R scripts used

(1).

(b).

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

# 04.23.2017

# Multiple Linear Regression (MLR) example

# BISC 577

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

## Install packages

# Bioconductor

source("https://bioconductor.org/biocLite.R")

biocLite()

biocLite("BiocUpgrade")

# DNAshapeR

biocLite("DNAshapeR")

# Caret

install.packages("caret")

## Initialization

biocLite("GenomeInfoDb")

library(DNAshapeR)

library(caret)

(4)

Script used:

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

# 04.23.2017

# Multiple Linear Regression (MLR) example

# BISC 577

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

## Install packages

# Bioconductor

source("https://bioconductor.org/biocLite.R")

biocLite()

# DNAshapeR

biocLite("DNAshapeR")

# Caret

install.packages("caret")

## Initialization

library(DNAshapeR)

library(caret)

## Predict DNA shapes

fn\_fasta <- "/Users/kptsi/Downloads/BISC577/gcPBM/Max.txt.fa"

pred <- getShape(fn\_fasta)

## Encode feature vectors

featureType <- c("1-mer", "1-shape")

featureVector <- encodeSeqShape(fn\_fasta, pred, featureType)

head(featureVector)

## Build MLR model by using Caret

# Data preparation

fn\_exp <- "/kptsi/Downloads/BISC577/gcPBM/Mad.txt"

exp\_data <- read.table(fn\_exp)

df <- data.frame(affinity=exp\_data$V2, featureVector)

# Arguments setting for Caret

trainControl <- trainControl(method = "cv", number = 10, savePredictions = TRUE)

# Prediction without L2-regularized

model <- train (affinity~ ., data = df, trControl=trainControl,

method = "lm", preProcess=NULL)

summary(model)

# Prediction with L2-regularized

model2 <- train(affinity~., data = df, trControl=trainControl,

method = "glmnet", tuneGrid = data.frame(alpha = 0, lambda = c(2^c(-15:15))))

model2

(5).

R script:

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

# 04.23.2017

# Plotting results example

# BISC 577

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

## Install and initialize packages

install.packages("ggplot2")

install.packages("grid")

library(ggplot2)

library(grid)

## Theme

my.theme <- theme(plot.margin = unit(c(0.1, 0.5, 0.1, 0.1), "cm"),axis.text = element\_text(colour="black", size=12),axis.title.x = element\_text(colour="black", size=12),axis.title.y = element\_text(colour="black", size=12),panel.grid.major = element\_blank(),panel.grid.minor = element\_blank(),panel.background = element\_blank(),axis.line = element\_line(colour = "black"), axis.text = element\_text(colour ="black"),axis.ticks = element\_line(colour = "black"))

## Data preparation

data1 <- c(0.6, 0.65)

data2 <- c(0.8, 0.85)

## Ploting

ggplot() +

geom\_point(aes(x = data1, y = data2), color = "red", size=1) +

geom\_abline(slope=1) + geom\_vline(xintercept=0) + geom\_hline(yintercept=0) +

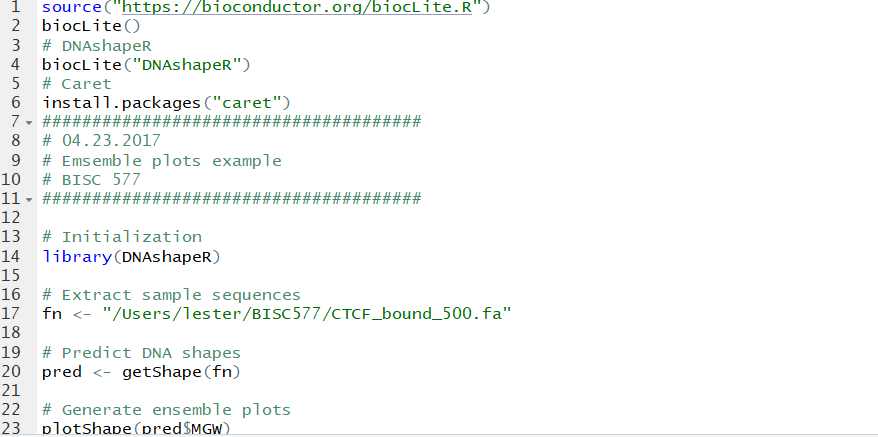
coord\_fixed(ratio = 1, xlim = c(0,1), ylim = c(0,1)) +

scale\_x\_continuous(expand = c(0, 0)) + scale\_y\_continuous(expand = c(0, 0)) +

my.theme)

(6).

(b).



(8).

Script:

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

# 04.23.2017

# Logistic regression on ChIP-seq data

# BISC 577

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

## Install packages

install.packages("caret")

install.packages("e1071")

install.packages("ROCR")

biocLite("Biostrings")

## Initialization

library(DNAshapeR)

library(caret)

library(ROCR)

library(Biostrings)

workingPath <- "/Users/kptsi/Downloads/BISC577/CTCF/"

## Generate data for the classifcation (assign Y to bound and N to non-bound)

# bound

boundFasta <- readDNAStringSet(paste0(workingPath, "bound\_30.fa"))

sequences <- paste(boundFasta)

boundTxt <- data.frame(seq=sequences, isBound="Y")

# non-bound

nonboundFasta <- readDNAStringSet(paste0(workingPath, "unbound\_30.fa"))

sequences <- paste(nonboundFasta)

nonboundTxt <- data.frame(seq=sequences, isBound="N")

# merge two datasets

writeXStringSet( c(boundFasta, nonboundFasta), paste0(workingPath, "ctcf.fa"))

exp\_data <- rbind(boundTxt, nonboundTxt)

## DNAshapeR prediction

pred <- getShape(paste0(workingPath, "ctcf.fa"))

## Encode feature vectors

featureType <- c("1-mer", "1-shape")

featureVector <- encodeSeqShape(paste0(workingPath, "ctcf.fa"), pred, featureType)

df <- data.frame(isBound = exp\_data$isBound, featureVector)

## Logistic regression

# Set parameters for Caret

trainControl <- trainControl(method = "cv", number = 10,

savePredictions = TRUE, classProbs = TRUE)

# Perform prediction

model <- train(isBound~ ., data = df, trControl = trainControl,

method = "glm", family = binomial, metric ="ROC")

summary(model)

## Plot AUROC

prediction <- prediction( model$pred$Y, model$pred$obs )

performance <- performance( prediction, "tpr", "fpr" )

plot(performance)

## Caluculate AUROC

auc <- performance(prediction, "auc")

auc <- unlist(slot(auc, "y.values"))

auc