Evaluation of Hubs, DCGs and DCLs on predicting AD with XGBoost

Gilderlanio Santana de Araújo

2023-03-09

```
library("xgboost")
library("reshape2")
library("ggpubr")
```

```
## Loading required package: ggplot2
```

Initial considerations.

This script deals with pre-processing of gene expression data to investigate the predictive power in a classification study (Case x Control). XGBoost was implemented to verify the prediction regarding the following attibutes: differentially expressed genes, hubs identified in co-expression analyses, DCGs and DCLs identified in differential co-expression analyses.

To evaluate XGBoost, an AUC plot is generated with the average AUC obtained by cross-validation.

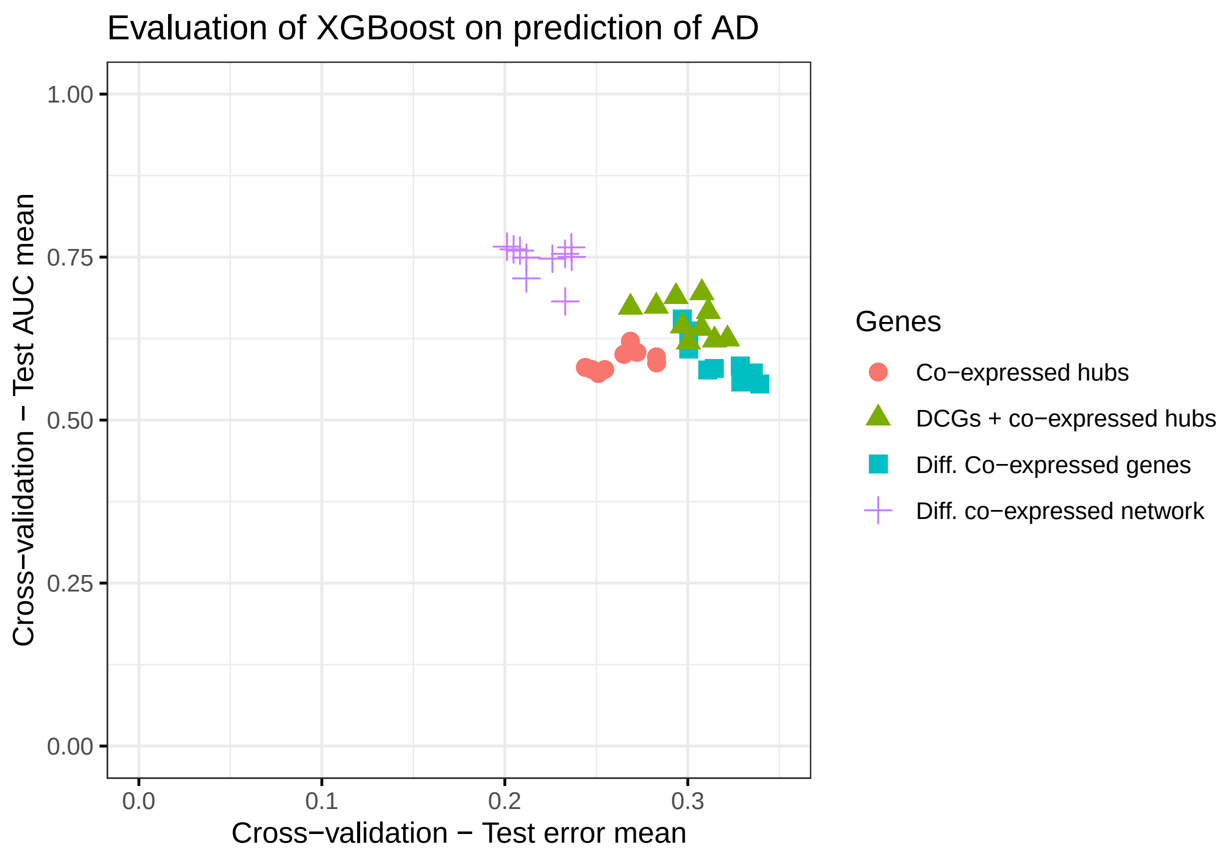
Pre-processing gene expression data

To add classes for each sample. Gene expression data must be randomized once the groups are in order. From 1 to 214 are AD cases and from 215 to 283 are control samples.

```
# Annotation samples with classes (0 - Control, 1 - AD)

expression <- as.data.frame(t(read.table("counts/GSE125583_DiffCoexpInput.txt",
                                          header = T, sep = "\t")))
class <- rep(c(1, 0), c(214, 69))
expression$class <- class

# Shuffling samples (3x)

expression = expression[sample(1:nrow(expression)), ]
expression = expression[sample(1:nrow(expression)), ]
expression = expression[sample(1:nrow(expression)), ]

# getting classes after shuffling
class <- expression$class
expression$class <- NULL
```

```r
# To extract gene expression for dcgs, hubs, and diffcoexp network

dcg.expression <- expression[,c("FAM153B", "CYP2C8", "CKMT1B")]
hub.expression <- expression[,c("PKN2", "FNDC3A", "NRIP1", "TMTC2")]
dcg.hub.expression <- expression[,c("FAM153B", "CYP2C8", "CKMT1B", "PKN2",
                                    "FNDC3A", "NRIP1", "TMTC2")]
dcls <- read.csv("results/DCL_GSE12.txt", sep = "\t")
net.expression <- expression[,union(dcls$Gene.1, dcls$Gene.2)]

# To perform XBoost cross-validation for dcgs, hubs, and diffcoexp network

cv.dcg <- xgb.cv(data = as.matrix(dcg.expression),
                 label = class, nrounds = 10,
                 nthread = 2, nfold = 5, metrics = list("rmse","auc", "error"),
                 max_depth = 2, eta = 1, objective = "binary:logistic", verbose = F)

cv.hub <- xgb.cv(data = as.matrix(hub.expression),
                 label = class, nrounds = 10,
                 nthread = 2, nfold = 5, metrics = list("rmse","auc", "error"),
                 max_depth = 2, eta = 1, objective = "binary:logistic", verbose = F)

cv.dcg.hub <- xgb.cv(data = as.matrix(dcg.hub.expression),
                     label = class, nrounds = 10,
                     nthread = 2, nfold = 5, metrics = list("rmse","auc", "error"),
                     max_depth = 2, eta = 1, objective = "binary:logistic", verbose = F)

cv.net <- xgb.cv(data = as.matrix(net.expression),
                 label = class, nrounds = 10,
                 nthread = 2, nfold = 5, metrics = list("rmse","auc", "error"),
                 max_depth = 2, eta = 1, objective = "binary:logistic", verbose = F)

# To get raw evaluation results

cv.dcg.df <- as.data.frame(cv.dcg$evaluation_log)
cv.hub.df <- as.data.frame(cv.hub$evaluation_log)
cv.dcg.hub.df <- as.data.frame(cv.dcg.hub$evaluation_log)
cv.net.df <- as.data.frame(cv.net$evaluation_log)

# To annotate raw evaluation results

cv.dcg.df$Genes <- "Diff. Co-expressed genes"
cv.hub.df$Genes <- "Co-expressed hubs"
cv.dcg.hub.df$Genes <- "DCGs + co-expressed hubs"
cv.net.df$Genes <- "Diff. co-expressed network"

cv.all <- Reduce(rbind, list(cv.dcg.df, cv.hub.df, cv.dcg.hub.df, cv.net.df))

# To extract error mean and AUC mean

cv.all.target <- cv.all[, c("test_auc_mean", "test_error_mean", "Genes", "iter")]
cv.all.target <- melt(cv.all.target, id.vars = c("Genes", "iter"))
```

```
# To plot AUC and error mean

p <- ggscatter(cv.all, x = "test_error_mean",
          y = "test_auc_mean", color = "Genes",
          ggtheme = theme_bw(), shape = "Genes", size = 3,
          ylab = "Cross-validation - Test AUC mean",
          title = "Evaluation of XGBoost on prediction of AD",
          xlab = "Cross-validation - Test error mean")
ggpar(p, xlim =c(0.0, 0.35), ylim=c(0, 1))
```


Evaluation of XGBoost on prediction of AD
Cross−validation − Test AUC mean
Cross−validation − Test error mean
Genes
Co−expressed hubs
DCGs + co−expressed hubs
Diff. Co−expressed genes
Diff. co−expressed network