

Evaluate testing data (multi-class)

Andrew Chang

2018-11-07

Contents

0. Load Data	1
1. Scores	1
1. Scores	2
1.1 uncalibrated average	2
3. Important Features	4
4. Hyper-parameters	8

Labels: 0: Basal 1: Luma

0. Load Data

```
library(tidyverse)
library(ggrepel)
library(caret)
library(pROC)
source("~/ml-pipeline/ML_performanceCheck/PerformanceUtils.R")

home <- "~/ml-pipeline/tests"
project_name <- "test_binaryclass_1"
training_data <- "test_binaryclass_tcga_brca.csv"
label_name <- "pam50_RNAseq"

## output path for analyzed data
outfile <- paste0(home, "/results/", project_name)
```

199 of samples were used

100 of full features

10 runs, each run contains 3 CVs.

Labels:

Var1	Freq
0	50
1	149

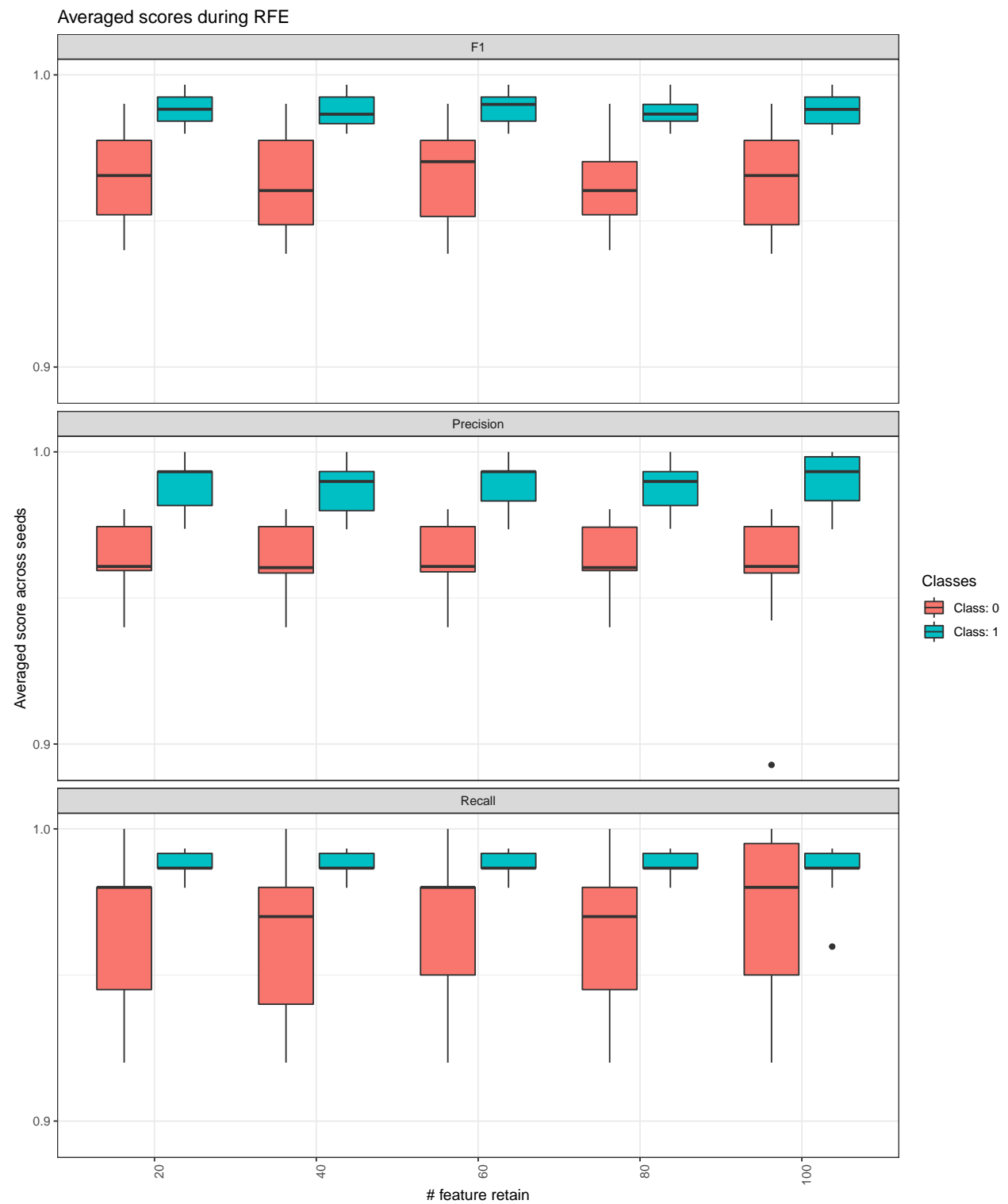
1. Scores

average = T: scores are based on pre-validation (combine predictions from all CVs per job and then calculate a single score per job).

average = F: report all the scores from each CV across entire jobs.

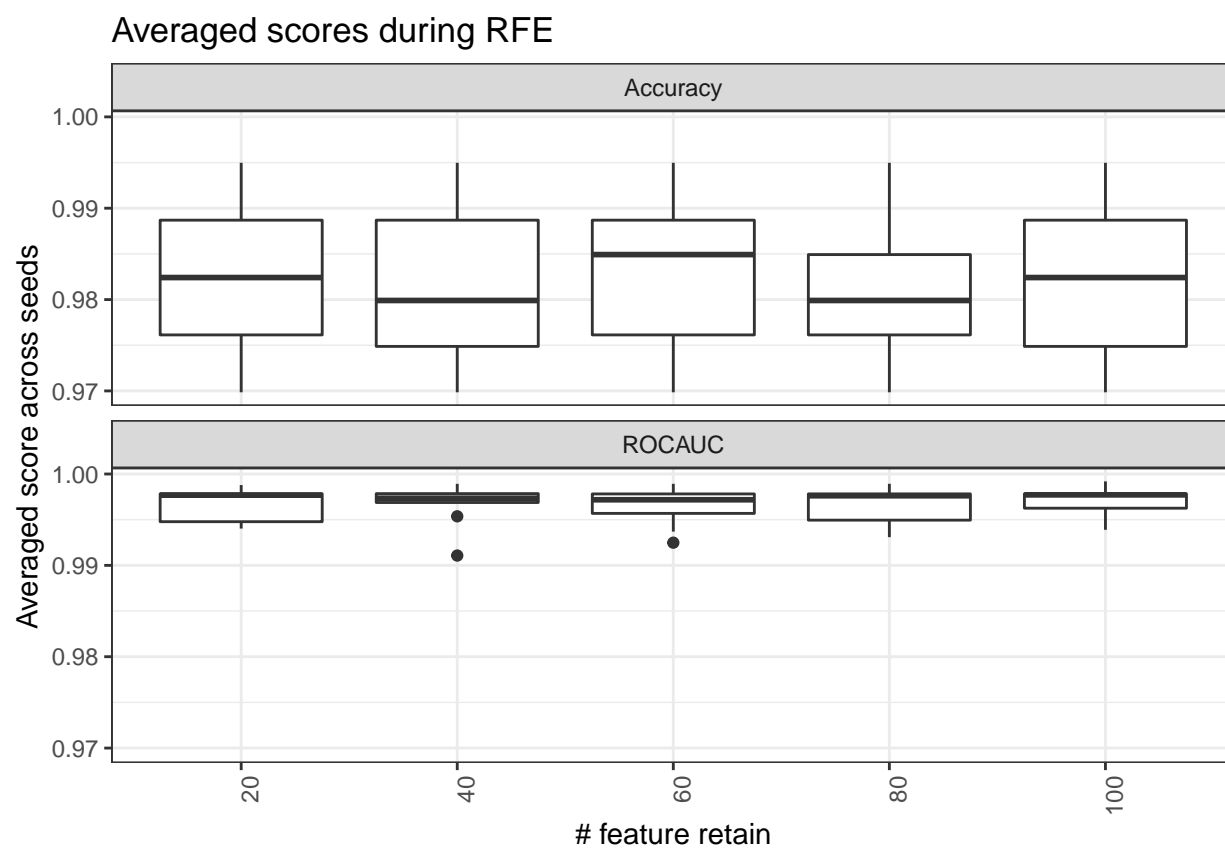
1. Scores

1.1 uncalibrated average



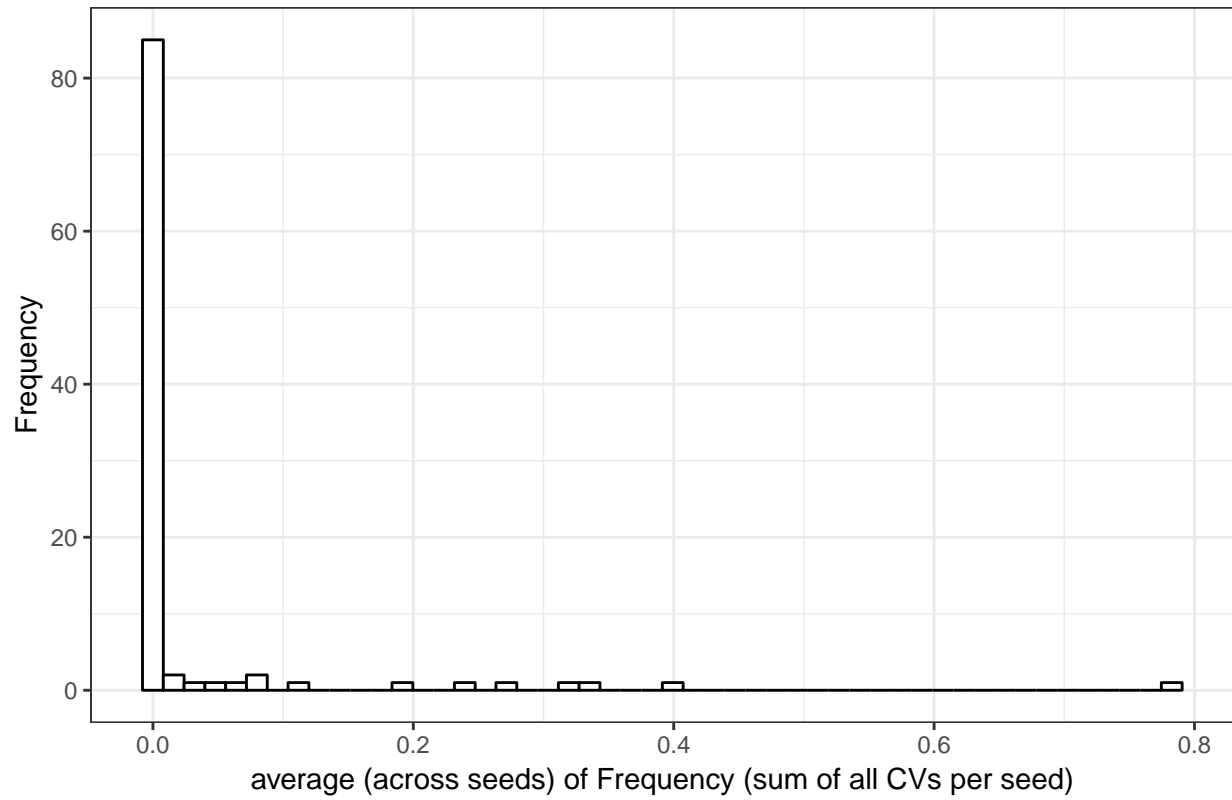
```
plt$overall +  
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
```

```
scale_y_continuous(breaks = seq(0, 1, 0.01))
```

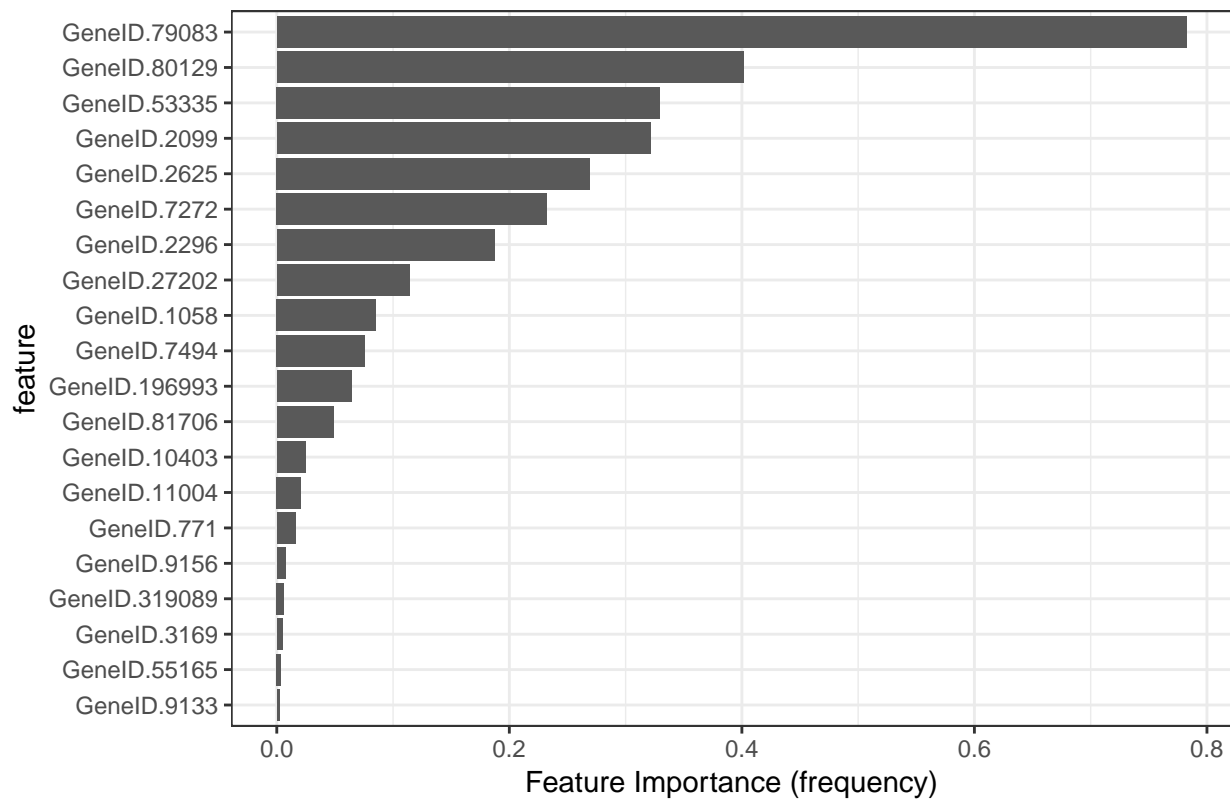


3. Important Features

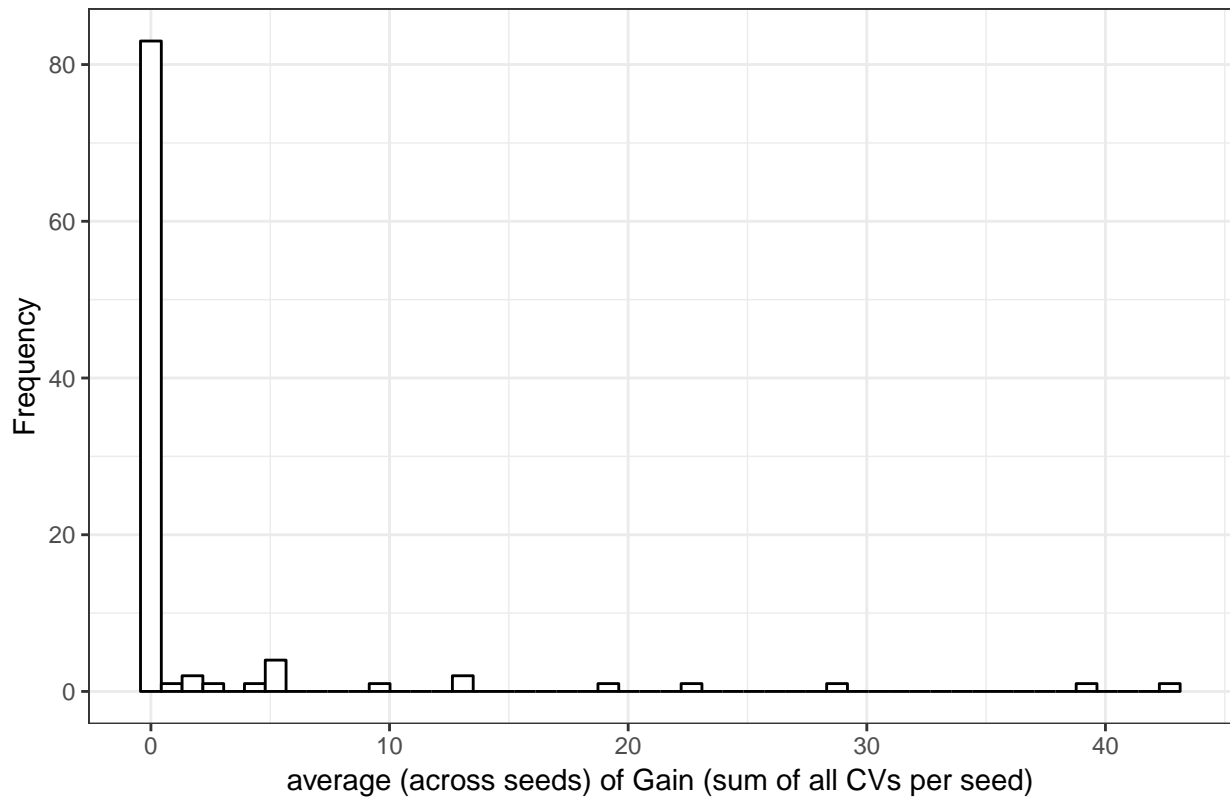
with 100 features based on Frequency



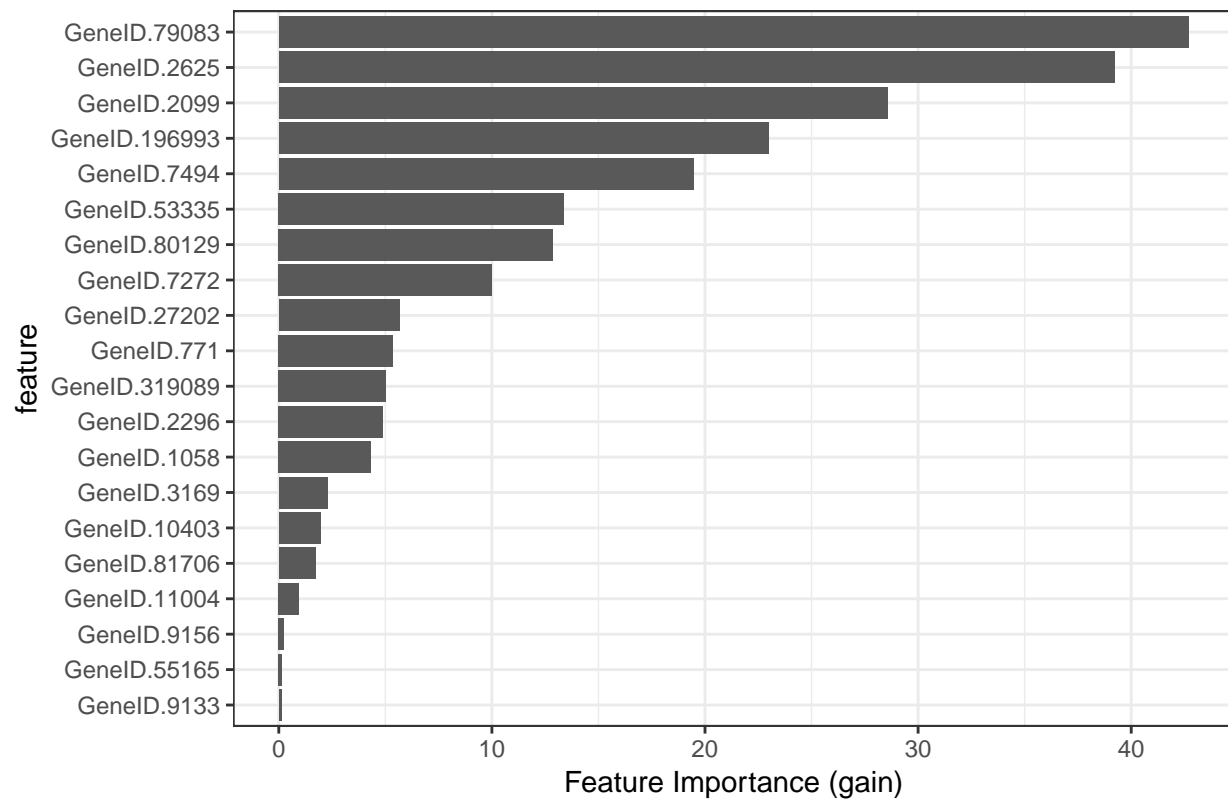
Top 20 features at 100 feature set based on Frequency



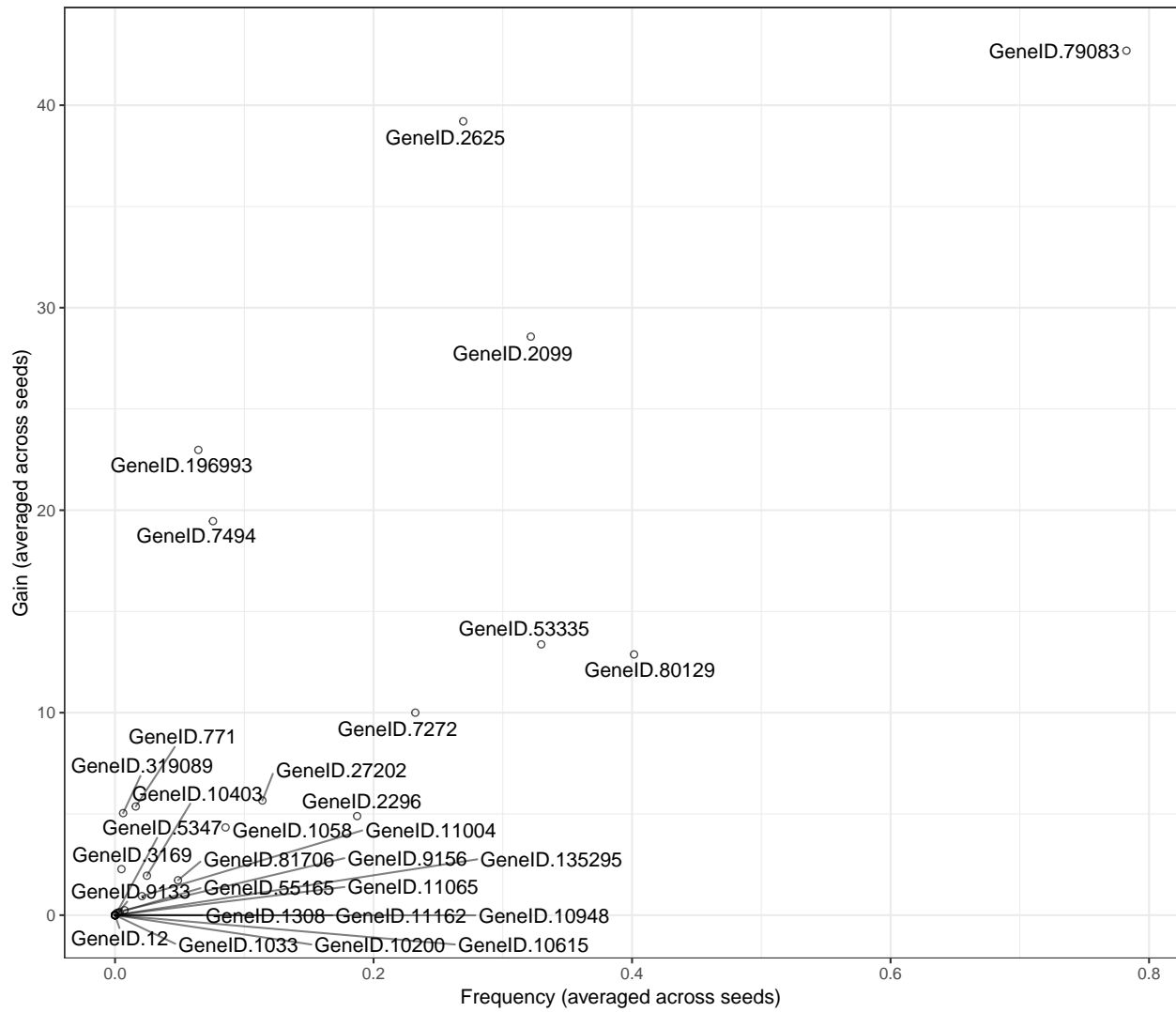
with 100 features based on Gain



Top 20 features at 100 feature set based on Gain



Top 30 features at 100 feature set



4. Hyper-parameters

