Pitfalls in the use of DNA Microarray Data for Diagnostic and Prognostic Classification, R. Simon et al 2003

MC Li

Contents

1 One Data Set

Goal: Repeat the result from the paper published in JNCI 2003.

Settings:

- n=20=10+10
- p=6000
- Prediction method: compound covariate prediction
- Gene selection method: 10 genes based on two-sample t-test

Compare

1. Resubstitution
2. LOOCV removal of the left-out specimen after selection of differentially expressed genes (wrong)
3. LOOCV removal of the left-out specimen before selection of differentially expressed genes (right way to do)

```r
ccp.train <- function(x, tt) {
  cc <- apply(x, 2, function(y) sum(tt * y))
  cc
}

ccp.predict <- function(c1, c2, tt, xnew) {
  cnew <- apply(xnew, 2, function(y) sum(tt * y))
  cmean <- (c1+c2)/2.0
  if (c1 <= c2) {
    pred <- ifelse(cnew <= cmean, 1, 2)
  } else {
    pred <- ifelse(cnew > cmean, 1, 2)
  }
  pred
}

n <- 20
p <- 6000
pg <- 10  # select 10 genes
set.seed(1234)
x <- matrix(rnorm(n*p), nr=p)
# assume the first 10 samples are in class 1, the rest samples are in class 2

# Resubstitution
out <- t.testv(x, 10, 10)
indgene <- order(out$pval)[1:pg]
ccp.tr <- ccp.train(x[indgene, ], out$t[indgene])
ccp.pr <- ccp.predict(mean(ccp.tr[1:10]), mean(ccp.tr[11:20]),
                      out$t[indgene], x[indgene, ])
err <- sum(abs(ccp.pr - c(rep(1, 10), rep(2, 10))))
err
# 0

# LOOCV after gene selection
out <- t.testv(x, 10, 10)
indgene <- order(out$pval)[1:pg]
ccp.pr <- rep(NA, n)
for(j in 1:n) {
  ccp.tr <- ccp.train(x[indgene, -j], out$t[indgene])
  if (j <= 10) {
    ccp.pr[j] <- ccp.predict(mean(ccp.tr[1:9]), mean(ccp.tr[10:19]),
                        out$t[indgene], x[indgene, j, drop = F])
  } else {
    ccp.pr[j] <- ccp.predict(mean(ccp.tr[1:10]), mean(ccp.tr[11:19]),
                        out$t[indgene], x[indgene, j, drop = F])
  }
}
err <- sum(abs(ccp.pr - c(rep(1, 10), rep(2, 10))))
err
# 0

# LOOCV before gene selection
ccp.pr <- rep(NA, n)
```

```
for(j in 1:n) {
  if (j <= 10) {
    n1 <- 9; n2 <- 10
  } else {
    n1 <- 10; n2 <- 9
  }
  out <- t.testv(x[, -j], n1, n2)
  indgene <- order(out$pval)[1:pg]
  ccp.tr <- ccp.train(x[indgene, -j], out$t[indgene])
  if (j <= 10) {
    ccp.pr[j] <- ccp.predict(mean(ccp.tr[1:9]), mean(ccp.tr[10:19]),
                        out$t[indgene], x[indgene, j, drop = F])
  } else {
    ccp.pr[j] <- ccp.predict(mean(ccp.tr[1:10]), mean(ccp.tr[11:19]),
                        out$t[indgene], x[indgene, j, drop = F])
  }
}
err <- sum(abs(ccp.pr - c(rep(1, 10), rep(2, 10))))
err
# 8
```

2 Compound Covariate Predictor

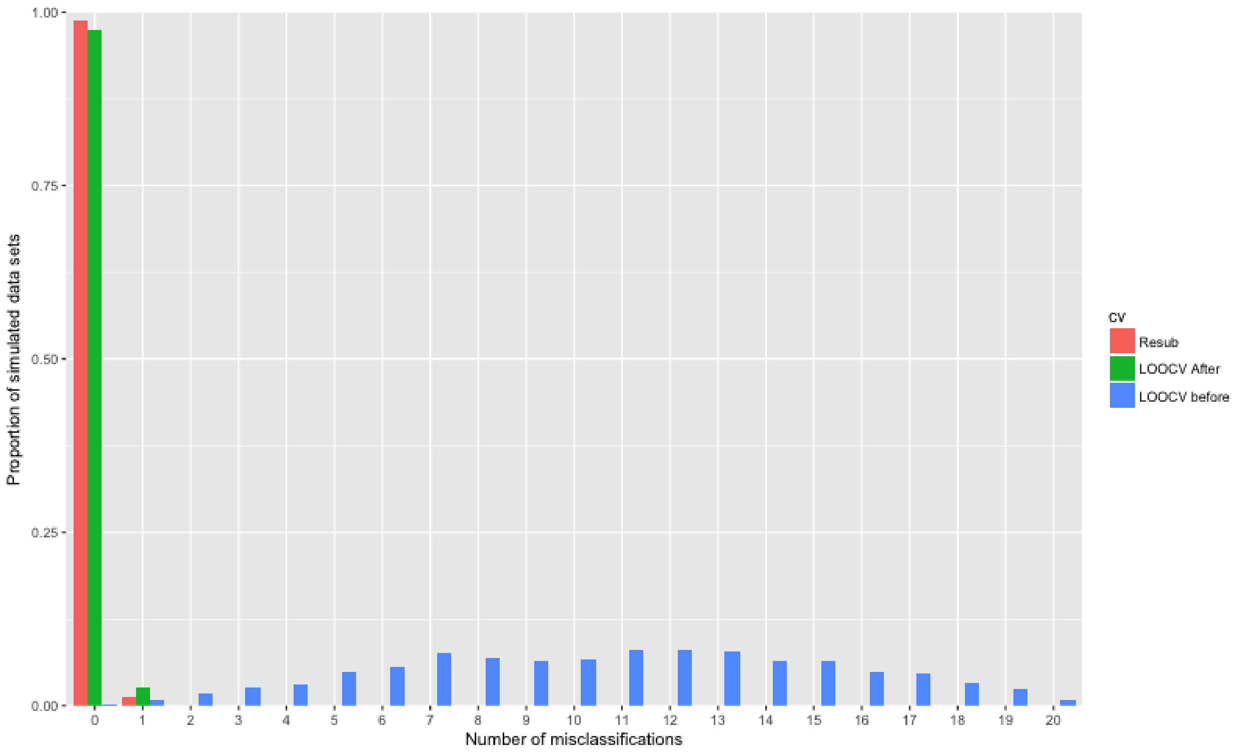
2.1 High Dimensional Case p = 6000, pg = 10

We can wrap the above scripts into 3 functions: rsbst(), loocv1() and loocv2(). rsbst() represents resubstitution method, loocv1() denotes LOOCV after gene selection and loocv2() denotes LOOCV before gene selection.

We draw a bar plot with X-axis = number of misclassifications, Y-axis = proportion of simulated data sets.

```
source("pitfalls.R")
nsim <- 2000
p <- 6000
pg <- 10  # select 10 genes

set.seed(1234)
out1 <- replicate(nsim, rsbst(p, pg))

set.seed(1234)
out2 <- replicate(nsim, loocv1(p, pg))

set.seed(1234)
out3 <- replicate(nsim, loocv2(p, pg))
save(out1, out2, out3, file = "out.rda")
load("out.rda")

# combine the result together
outall <- rbind(table(factor(out1, levels = as.character(0:20))),
                table(factor(out2, levels = as.character(0:20))),
                table(factor(out3, levels = as.character(0:20))))
outall
```

```
#        0  1  2  3  4   5   6   7   8   9  10  11  12  13  14  15 16 17 18 19 20
#[1,] 1973 27  0  0  0   0   0   0   0   0   0   0   0   0   0   0  0  0  0  0  0
#[2,] 1946 53  1  0  0   0   0   0   0   0   0   0   0   0   0   0  0  0  0  0  0
#[3,]    5 15 37 51 64 100 110 154 138 129 136 162 160 157 131 128 98 95 66 49 15

# base R plot version
png("outall.png", width=800, height=480)
barplot(outall/nsim, beside=TRUE,
        col=c("aquamarine3", "coral", "blue"),
        names.arg=as.character(0:20),
        xlab = "Number of misclassifications",
        ylab = "Proportion of simulated data sets")


legend("top", c("Resubstition", "LOOCV after", "LOOCV before"),
       col=c("aquamarine3", "coral", "blue"), pch=15)
grid(NA, 10, lwd = 2)
dev.off()

# ggplot2 version
library(ggplot2)
dat1 <- data.frame(
    cv = factor(c(rep("Resub", 21), rep("LOOCV After", 21), rep("LOOCV before", 21)), levels = c("Resub
    mis = factor(rep(0:20, 3)),
    total = c(outall[1, ]/nsim, outall[2, ]/nsim, outall[3, ]/nsim)
)
dat1
png("outall_gg.png", width=800, height=480)
ggplot(data=dat1, aes(x=mis , y=total, fill=cv)) +
    geom_bar(stat="identity", position=position_dodge()) +
    xlab("Number of misclassifications") +
    ylab("Proportion of simulated data sets") +
    scale_y_continuous(expand = c(0,0), limits = c(0,1))
dev.off()
```



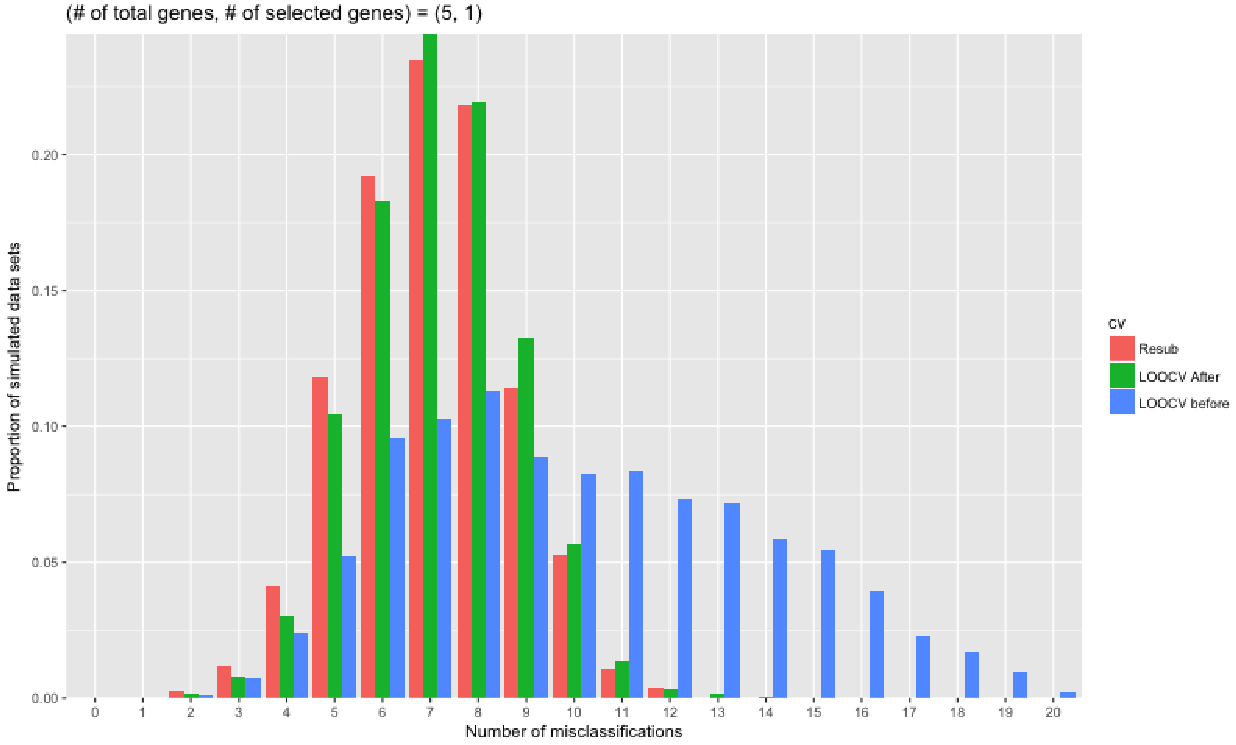
Observations:

Note that under the null hypothesis, the estimated error rates for simulated datasets should center around 0.5 (i.e. 10 misclassifications of 20).

- Resubstitution method is biased for small datasets. About 98% (=1973/2000) of the simulated datasets resulting in zero misclassifications
- LOOCV after gene selection does little to correct the bias, with 97% (=1946/2000) of simulated datasets still resulting in zero misclassifications.

2.2 Low Dimension Case p=5, pg=1

```
source("pitfalls.R")
nsim <- 2000
p <- 5
pg <- 1  # select 1 gene

set.seed(1234)
out4 <- replicate(nsim, rsbst(p, pg))

set.seed(1234)
out5 <- replicate(nsim, loocv1(p, pg))

set.seed(1234)
out6 <- replicate(nsim, loocv2(p, pg))
save(out4, out5, out6, file = "outlowd1.rda")
load("outlowd1.rda")
outlowd <- rbind(table(factor(out4, levels = as.character(0:20))),
                 table(factor(out5, levels = as.character(0:20))),
                 table(factor(out6, levels = as.character(0:20))))
outlowd
#     0 1 2  3  4   5   6   7   8   9  10  11  12  13  14  15  16 17 18 19 20
#[1,] 0 0 5 24 83 236 384 469 436 228 105  22   8   0   0   0   0  0  0  0  0
#[2,] 0 0 3 16 61 209 366 489 439 265 114  27   7   3   1   0   0  0  0  0  0
#[3,] 0 0 2 15 48 104 192 205 226 178 165 167 147 143 117 109 79 46 34 19  4

# ggplot2 version
dat1 <- data.frame(
    cv = factor(c(rep("Resub", 21), rep("LOOCV After", 21), rep("LOOCV before", 21)), levels = c("Resub
    mis = factor(rep(0:20, 3)),
    total = c(outlowd[1, ]/nsim, outlowd[2, ]/nsim, outlowd[3, ]/nsim)
)
png("lowdim1.png", width=800, height=480)
ggplot(data=dat1, aes(x=mis , y=total, fill=cv)) +
    geom_bar(stat="identity", position=position_dodge()) +
    xlab("Number of misclassifications") +
    ylab("Proportion of simulated data sets") +
    scale_y_continuous(expand = c(0,0)) +
    ggtitle(sprintf("(# of total genes, # of selected genes) = (%d, %d)", p, pg))
dev.off()
```

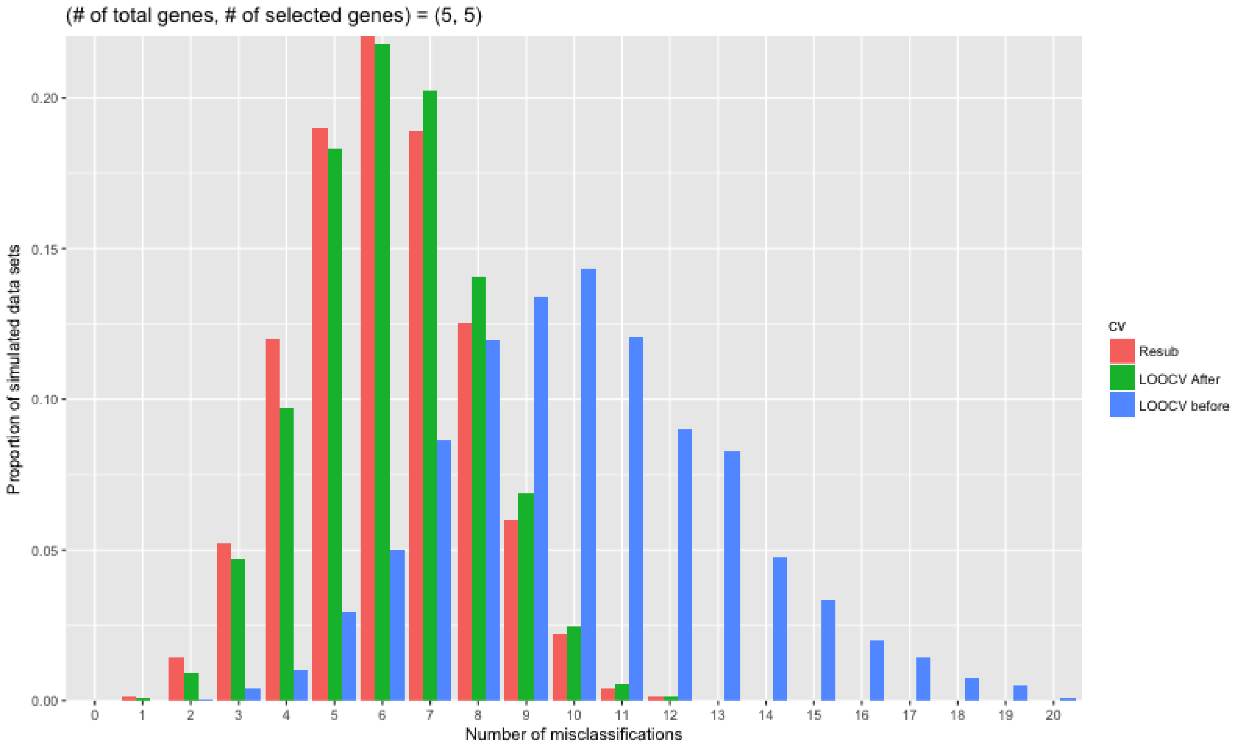


It is strange the LOOCV before gene selection method is also biased.

2.3 Low Dimension Case p=5, pg=5

Let's see what happened if the number of total genes equals to the number of selected genes.

```r
source("pitfalls.R")
nsim <- 2000
p <- 5
pg <- p

set.seed(1234)
out4 <- replicate(nsim, rsbst(p, pg))

set.seed(1234)
out5 <- replicate(nsim, loocv1(p, pg))

set.seed(1234)
out6 <- replicate(nsim, loocv2(p, pg))
save(out4, out5, out6, file = "outlowd5.rda")
load("outlowd5.rda")
outlowd <- rbind(table(factor(out4, levels = as.character(0:20))),
                 table(factor(out5, levels = as.character(0:20))),
                 table(factor(out6, levels = as.character(0:20))))
outlowd
#     0 1  2   3   4   5   6   7   8   9  10  11  12  13 14 15 16 17 18 19 20
#[1,] 0 3 29 104 240 380 441 378 250 120  44   8   3   0  0  0  0  0  0  0  0
#[2,] 0 2 18  94 195 366 436 405 282 138  50  11   3   0  0  0  0  0  0  0  0
#[3,] 0 0  1   8  20  59 100 173 239 268 287 241 180 166 95 67 40 29 15 10  2

# ggplot2 version
dat1 <- data.frame(
    cv = factor(c(rep("Resub", 21), rep("LOOCV After", 21), rep("LOOCV before", 21)), levels = c("Resub
    mis = factor(rep(0:20, 3)),
    total = c(outlowd[1, ]/nsim, outlowd[2, ]/nsim, outlowd[3, ]/nsim)
)
png("lowdim5.png", width=800, height=480)
ggplot(data=dat1, aes(x=mis , y=total, fill=cv)) +
    geom_bar(stat="identity", position=position_dodge()) +
    xlab("Number of misclassifications") +
    ylab("Proportion of simulated data sets") +
    scale_y_continuous(expand = c(0,0)) +
    ggtitle(sprintf("(# of total genes, # of selected genes) = (%d, %d)", p, pg))
dev.off()
```

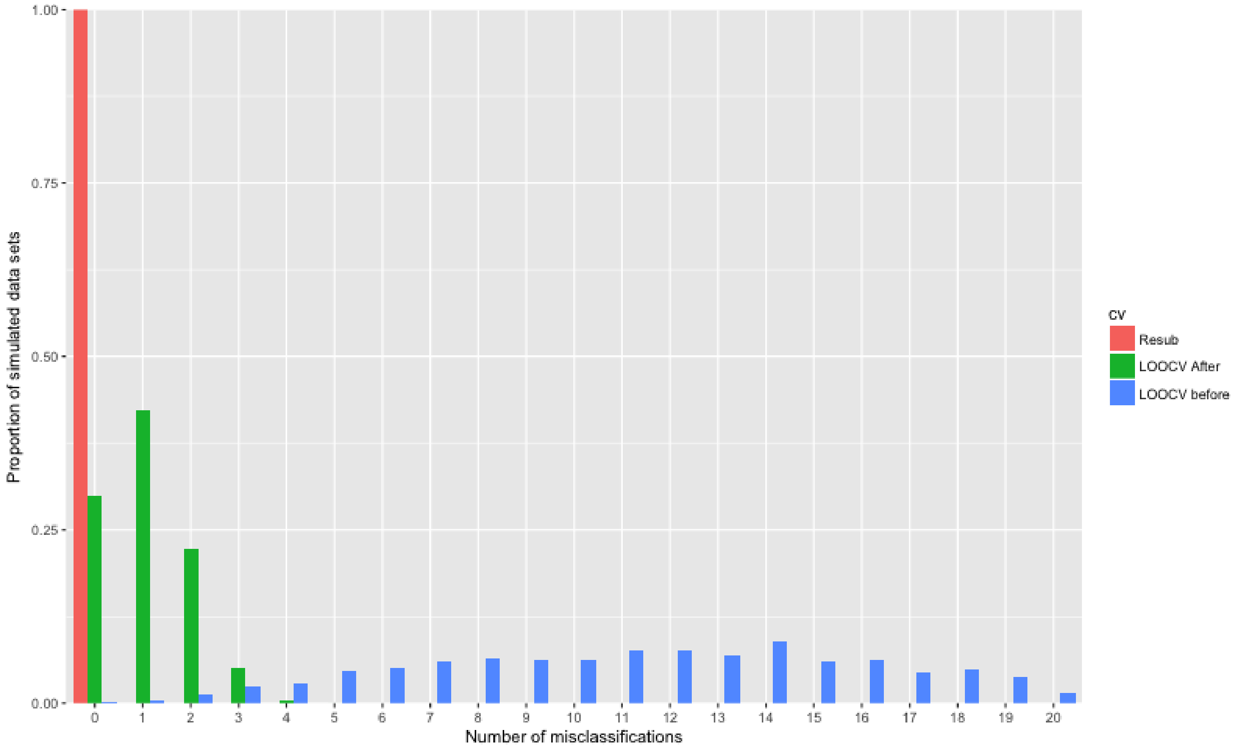


3 Random Forest Predictor

3.1 High Dimensional Case p = 6000, pg = 10

```
source("pitfalls.R")
nsim <- 2000
p <- 6000
pg <- 10  # select 10 genes

set.seed(1234)
out1 <- replicate(nsim, rsbst(p, pg, "randomForest"))

set.seed(1234)
out2 <- replicate(nsim, loocv1(p, pg, "randomForest"))

set.seed(1234)
out3 <- replicate(nsim, loocv2(p, pg, "randomForest"))
save(out1, out2, out3, file = "out_rf.rda")
load("out_rf.rda")

# combine the result together
outall <- rbind(table(factor(out1, levels = as.character(0:20))),
                table(factor(out2, levels = as.character(0:20))),
                table(factor(out3, levels = as.character(0:20))))
outall
#        0    1    2    3  4  5    6    7    8    9   10   11   12   13   14   15   16 17 18 19 20
#[1,] 2000    0    0    0  0  0    0    0    0    0    0    0    0    0    0    0    0  0  0  0  0
```

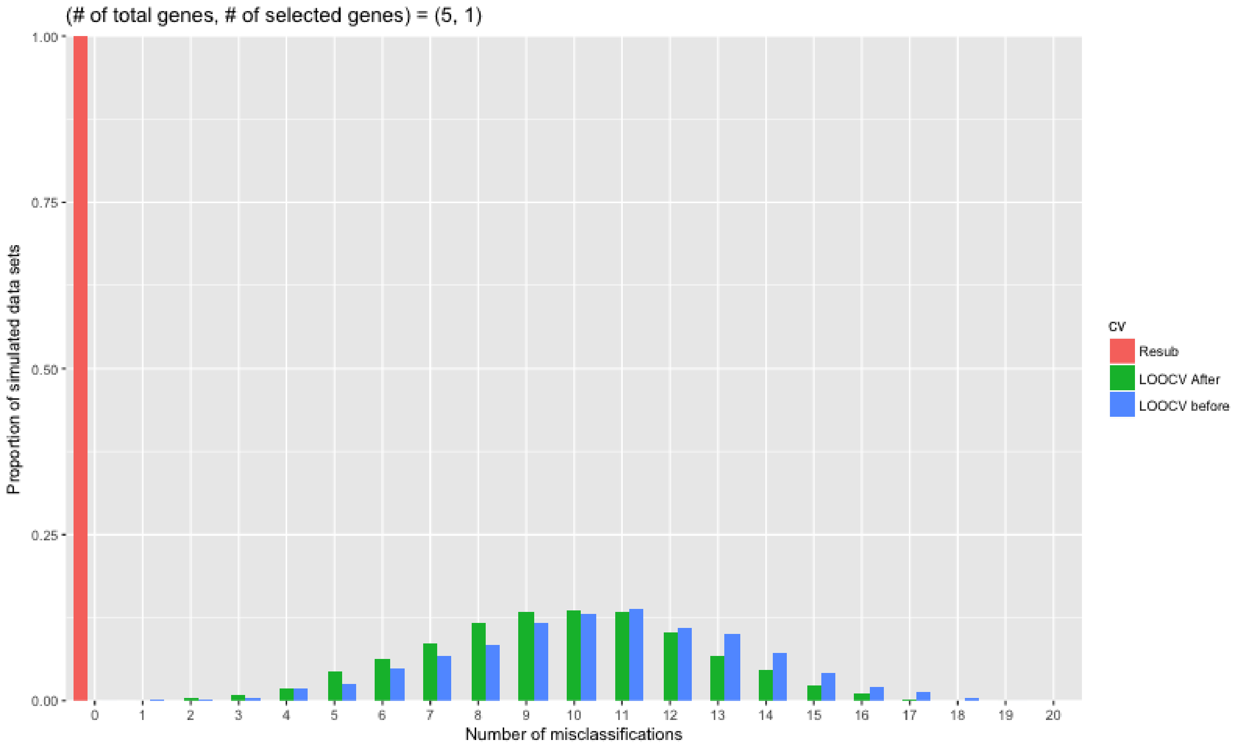
```
#[2,]  600 846 444 103   6   1   0   0   0   0   0   0   0   0   0   0   0  0  0  0  0
#[3,]    3  10  26  49 57 95 103 120 129 127 123 151 154 140 179 120 123 87 98 74 32

# ggplot2 version
library(ggplot2)
dat1 <- data.frame(
    cv = factor(c(rep("Resub", 21), rep("LOOCV After", 21), rep("LOOCV before", 21)), levels = c("Resub
    mis = factor(rep(0:20, 3)),
    total = c(outall[1, ]/nsim, outall[2, ]/nsim, outall[3, ]/nsim)
)
dat1
png("outall_rf.png", width=800, height=480)
ggplot(data=dat1, aes(x=mis , y=total, fill=cv)) +
    geom_bar(stat="identity", position=position_dodge()) +
    xlab("Number of misclassifications") +
    ylab("Proportion of simulated data sets") +
    scale_y_continuous(expand = c(0,0), limits = c(0,1))
dev.off()
```



3.2 Low Dimension Case p=5, pg=1

```
source("pitfalls.R")
nsim <- 2000
p <- 5
pg <- 1  # select 1 gene

set.seed(1234)
```

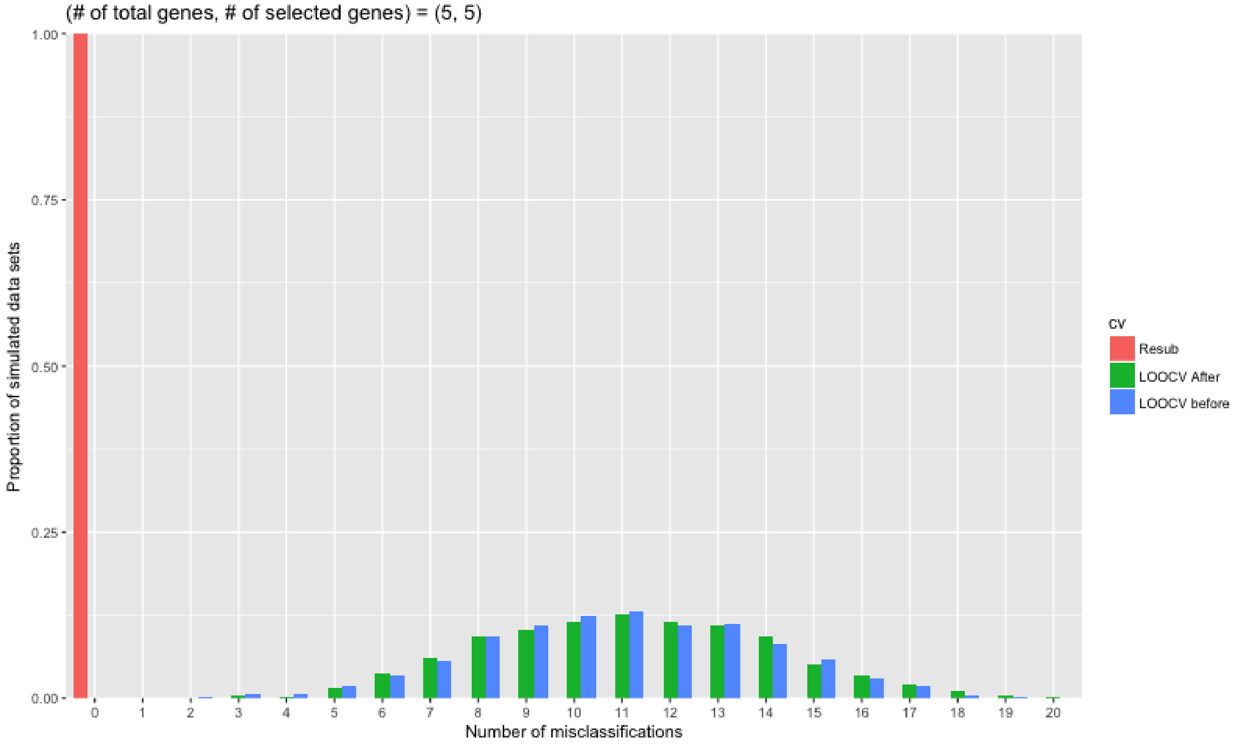
```r
out4 <- replicate(nsim, rsbst(p, pg, "randomForest"))

set.seed(1234)
out5 <- replicate(nsim, loocv1(p, pg, "randomForest"))

set.seed(1234)
out6 <- replicate(nsim, loocv2(p, pg, "randomForest"))
save(out4, out5, out6, file = "outlowd1_rf.rda")
load("outlowd1_rf.rda")
outlowd <- rbind(table(factor(out4, levels = as.character(0:20))),
                 table(factor(out5, levels = as.character(0:20))),
                 table(factor(out6, levels = as.character(0:20))))
outlowd
#        0 1 2  3  4  5   6   7   8   9  10  11  12  13  14 15 16 17 18 19 20
#[1,] 2000 0 0  0  0  0   0   0   0   0   0   0   0   0   0  0  0  0  0  0  0
#[2,]    0 1 7 20 38 86 127 173 235 265 272 265 206 135  95 44 24  6  0  1  0
#[3,]    1 3 3  9 35 51  97 137 170 234 260 275 219 202 147 82 42 25  7  1  0

# ggplot2 version
dat1 <- data.frame(
    cv = factor(c(rep("Resub", 21), rep("LOOCV After", 21), rep("LOOCV before", 21)), levels = c("Resub
    mis = factor(rep(0:20, 3)),
    total = c(outlowd[1, ]/nsim, outlowd[2, ]/nsim, outlowd[3, ]/nsim)
)
png("lowdim1_rf.png", width=800, height=480)
ggplot(data=dat1, aes(x=mis , y=total, fill=cv)) +
    geom_bar(stat="identity", position=position_dodge()) +
    xlab("Number of misclassifications") +
    ylab("Proportion of simulated data sets") +
    scale_y_continuous(expand = c(0,0)) +
    ggtitle(sprintf("(# of total genes, # of selected genes) = (%d, %d)", p, pg))
dev.off()
```



3.3 Low Dimension Case p=5, pg=5

```
source("pitfalls.R")
nsim <- 2000
p <- 5
pg <- p

set.seed(1234)
out4 <- replicate(nsim, rsbst(p, pg, "randomForest"))

set.seed(1234)
out5 <- replicate(nsim, loocv1(p, pg, "randomForest"))

set.seed(1234)
out6 <- replicate(nsim, loocv2(p, pg, "randomForest"))
save(out4, out5, out6, file = "outlowd5_rf.rda")
load("outlowd5_rf.rda")
outlowd <- rbind(table(factor(out4, levels = as.character(0:20))),
                 table(factor(out5, levels = as.character(0:20))),
                 table(factor(out6, levels = as.character(0:20))))
outlowd
#       0 1 2  3  4  5  6   7   8   9  10  11  12  13  14  15 16 17 18 19 20
#[1,] 2000 0 0  0  0  0  0   0   0   0   0   0   0   0   0   0  0  0  0  0  0
#[2,]    0 0 1  7  6 34 75 122 186 205 229 254 231 220 186 101 70 41 21  9  2
#[3,]    0 0 2 12 14 35 68 112 188 220 249 264 222 226 165 115 58 36  9  5  0

# ggplot2 version
```

```
dat1 <- data.frame(
    cv = factor(c(rep("Resub", 21), rep("LOOCV After", 21), rep("LOOCV before", 21)), levels = c("Resub
    mis = factor(rep(0:20, 3)),
    total = c(outlowd[1, ]/nsim, outlowd[2, ]/nsim, outlowd[3, ]/nsim)
)
png("lowdim5_rf.png", width=800, height=480)
ggplot(data=dat1, aes(x=mis , y=total, fill=cv)) +
    geom_bar(stat="identity", position=position_dodge()) +
    xlab("Number of misclassifications") +
    ylab("Proportion of simulated data sets") +
    scale_y_continuous(expand = c(0,0)) +
    ggtitle(sprintf("(# of total genes, # of selected genes) = (%d, %d)", p, pg))
dev.off()
```


(# of total genes, # of selected genes) = (5, 5)
Proportion of simulated data sets
Number of misclassifications
cv
Resub
LOOCV After
LOOCV before