Random data ROC

Peter Hettegger

2020-11-30

This document provides a simulation example for evaluating ROC AUC values for:

1. batch effect correction with `ComBat`.
2. batch effect correction with `ComBat` and correction of p-values with 1000 random rotations.

We assume a sample design of 3 `batches` containing samples of two `groups` (control/case), 30 samples in total. We assume to have 5000 `features` or genes, where the first 1000 features are differentially expressed.

```
library(randRotation)
library(pROC)
library(GGally)
set.seed(0)
```

```
pdata <- data.frame(batch = as.factor(rep(c(1:3), c(10,10,10))),
                    group = as.factor(rep(c("Control", "Case"), c(5,5))))
with(pdata, table(batch, group))
```

```
##      group
## batch Case Control
##     1    5       5
##     2    5       5
##     3    5       5
```

We generate random normal data for 5000 features

```
features <- 5000

# Matrix with random gene expression data
edata <- matrix(rnorm(features * nrow(pdata)), features)
rownames(edata) <- paste("feature", 1:nrow(edata))
```

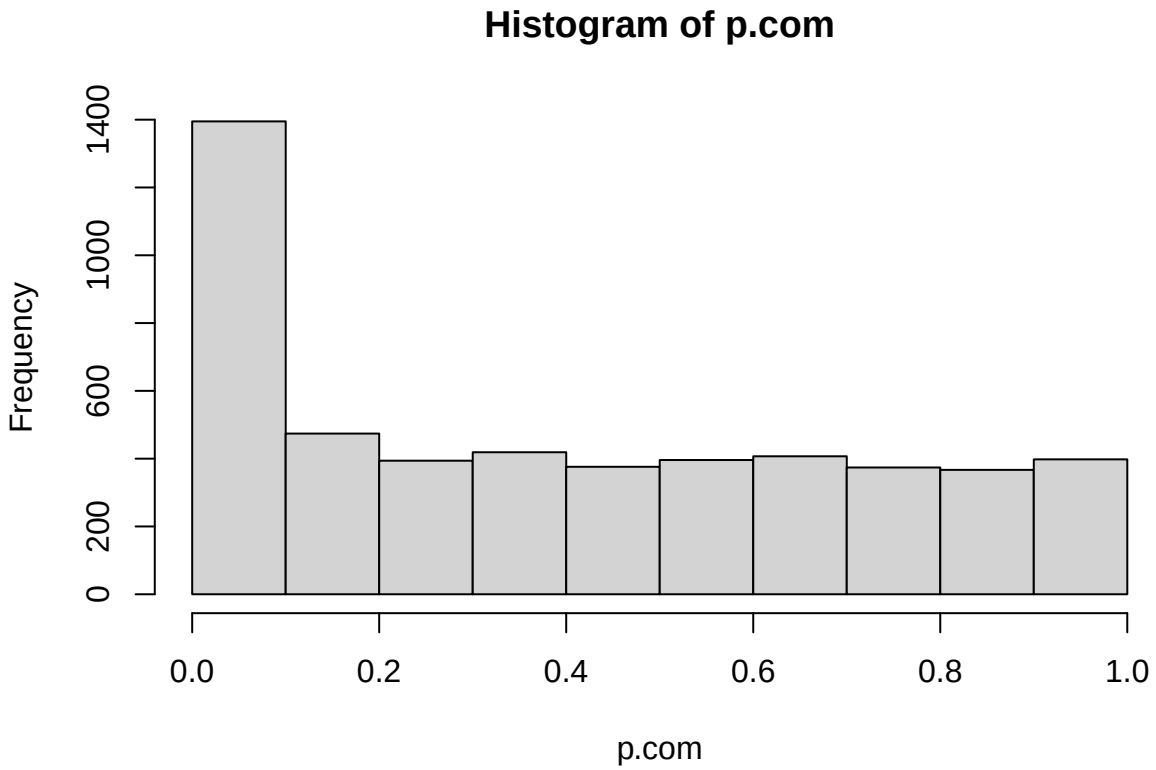
and add a small artificial covariate effect to the first 1000 features:

```
######## add covariate effect for features 1:1000 ###########
feat.sign <- 1:1000
edata[feat.sign,] <- t(t(edata[feat.sign,]) + as.numeric(pdata$group) - 1.5)
#############################################################
```

We now perform analysis (1) with batch effect correction by `ComBat`, but without random rotation:

```
mod1 <- model.matrix(~group, pdata)

##### ComBat
edata.combat <- sva::ComBat(edata, pdata$batch, mod1)
fit1 <- limma::lmFit(edata.combat, mod1)
fit1 <- limma::eBayes(fit1)
tt1  <- limma::topTable(fit1, 2, Inf, sort.by = "none", adjust.method = "none")
```

```
p.com <- tt1$P.Value

hist(p.com)
```

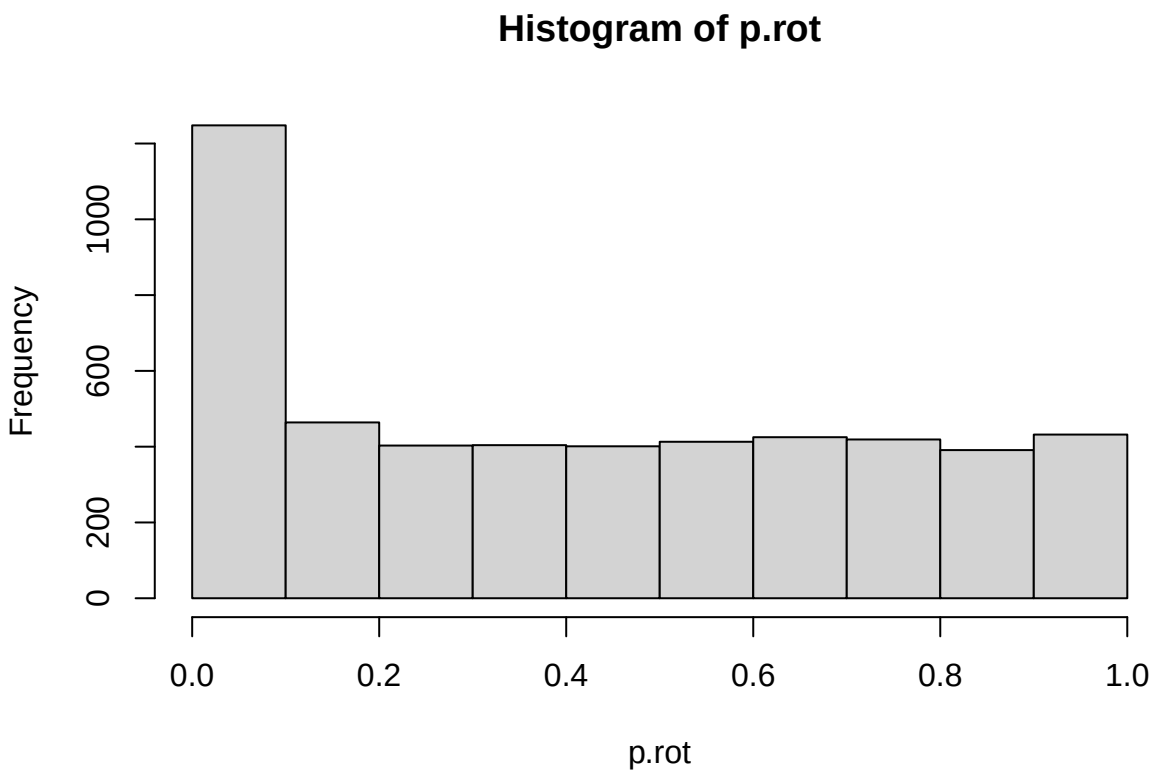


Now we perform the same analysis, but with p-value correction by random rotation (analysis (2)):

```
###### with random rotation

rr1 <- initBatchRandrot(edata, mod1, 2, pdata$batch)
```

```
## Initialising batch "1"
## Initialising batch "2"
## Initialising batch "3"
```

```
statistic <- function(Y, mod, batch){
    Y <- sva::ComBat(Y, batch, mod)
    fit1 <- limma::lmFit(Y, mod)
    fit1 <- limma::eBayes(fit1)
    abs(fit1$t[,2])
}

rs1 <- rotateStat(rr1, 1000, statistic, mod1, pdata$batch, parallel = TRUE)
p.rot <- pFdr(rs1, pooled = TRUE)[,1]

hist(p.rot)
```



The following code generates the ROC curves for both analyses:

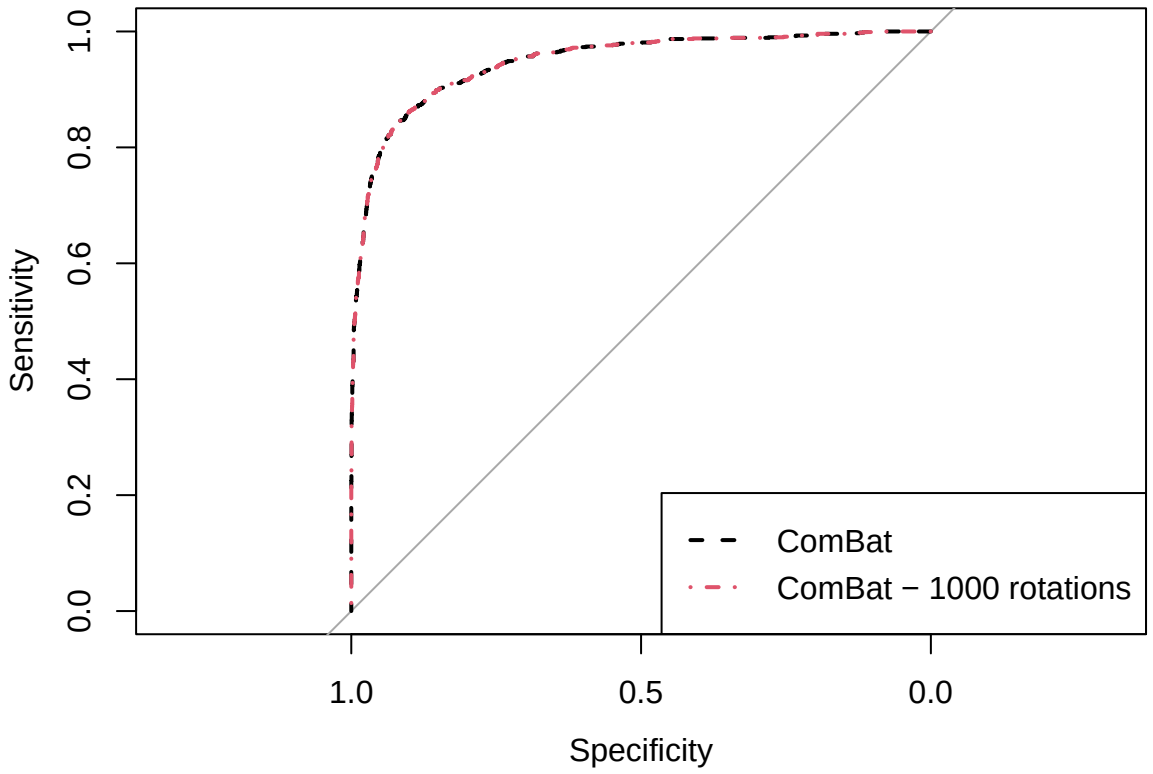
```
r.com <- roc(cases = p.com[feat.sign], controls = p.com[-feat.sign])
r.com
```

```
##
## Call:
## roc.default(controls = p.com[-feat.sign], cases = p.com[feat.sign])
##
## Data: 4000 controls > 1000 cases.
## Area under the curve: 0.948
```

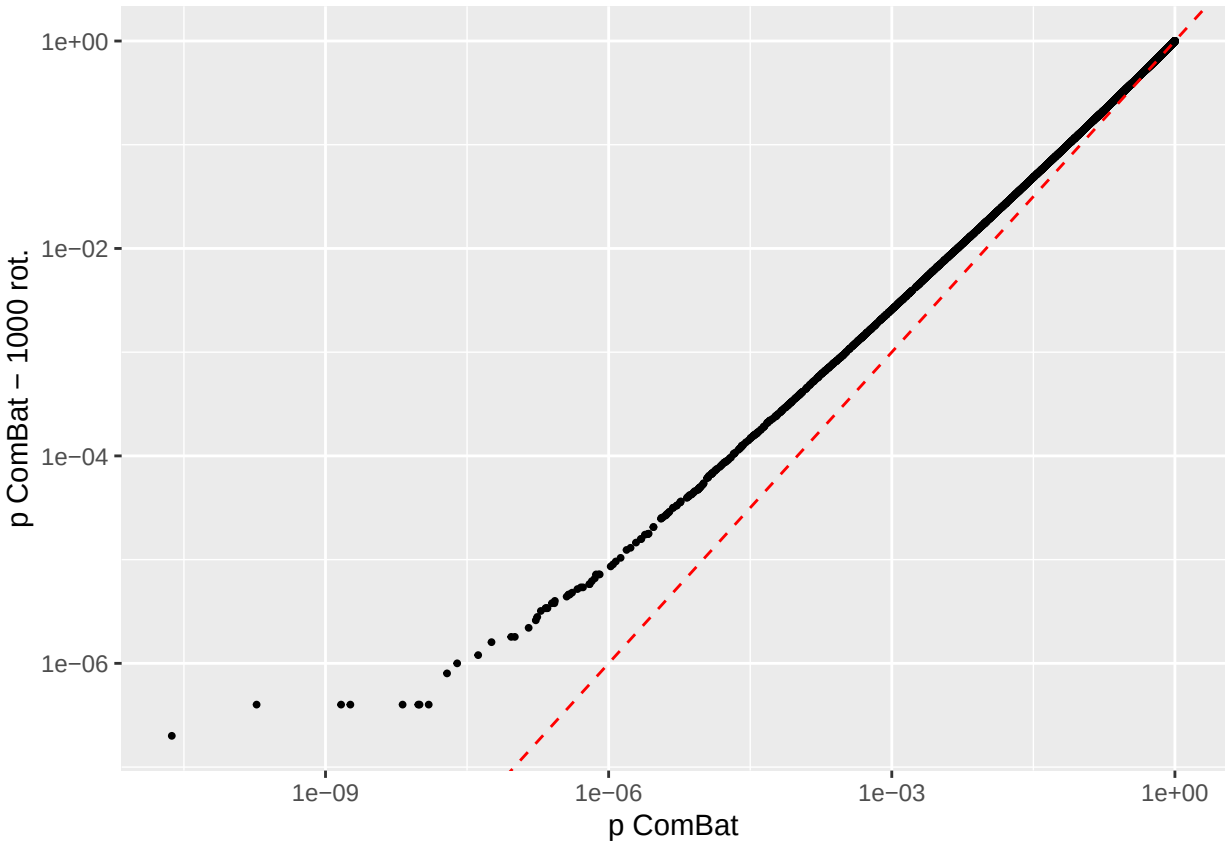
```
r.rot <- roc(cases = p.rot[feat.sign], controls = p.rot[-feat.sign])
r.rot
```

```
##
## Call:
## roc.default(controls = p.rot[-feat.sign], cases = p.rot[feat.sign])
##
## Data: 4000 controls > 1000 cases.
## Area under the curve: 0.948
```

```
plot(r.com, lty = 2, lwd = 2)
lines(r.rot, col = 2, lty = 4, lwd = 2)

legend("bottomright", legend = c("ComBat", "ComBat - 1000 rotations"),
       lty = c(2,4), col = 1:2, lwd = 2)
```




```
df1 <- data.frame(p.com, p.rot)
colnames(df1) <- c("p ComBat", "p ComBat - 1000 rot.")

ggally_points(df1, aes(x = `p ComBat`, y = `p ComBat - 1000 rot.`), size = 0.7)+
    scale_y_log10()+
    scale_x_log10()+
    geom_abline(slope = 1, intercept = 0, lty = 2, lwd = 0.5, col = "red")
```



Session info

```
sessionInfo()
```

```
## R Under development (unstable) (2020-11-14 r79432)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19041)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=German_Austria.1252  LC_CTYPE=German_Austria.1252
## [3] LC_MONETARY=German_Austria.1252 LC_NUMERIC=C
## [5] LC_TIME=German_Austria.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] GGally_2.0.0       ggplot2_3.3.2      pROC_1.16.2        randRotation_1.3.4
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.5           locfit_1.5-9.4       lattice_0.20-41
##  [4] snow_0.4-3           digest_0.6.27        R6_2.5.0
##  [7] plyr_1.8.6           stats4_4.1.0         RSQLite_2.2.1
## [10] evaluate_0.14        sva_3.39.0           httr_1.4.2
```

```
## [13] pillar_1.4.6         Rdpack_2.1           rlang_0.4.8
## [16] annotate_1.69.0      blob_1.2.1           S4Vectors_0.29.3
## [19] Matrix_1.2-18        rmarkdown_2.5        splines_4.1.0
## [22] BiocParallel_1.25.1  stringr_1.4.0        bit_4.0.4
## [25] munsell_0.5.0        compiler_4.1.0       xfun_0.19
## [28] pkgconfig_2.0.3      BiocGenerics_0.37.0  mgcv_1.8-33
## [31] htmltools_0.5.0      tibble_3.0.4         edgeR_3.33.0
## [34] IRanges_2.25.2       matrixStats_0.57.0   XML_3.99-0.5
## [37] reshape_0.8.8        crayon_1.3.4         withr_2.3.0
## [40] rbibutils_1.4        grid_4.1.0           nlme_3.1-150
## [43] xtable_1.8-4         gtable_0.3.0         lifecycle_0.2.0
## [46] DBI_1.1.0            magrittr_1.5         scales_1.1.1
## [49] stringi_1.5.3        farver_2.0.3         genefilter_1.73.0
## [52] limma_3.47.0         xml2_1.3.2           ellipsis_0.3.1
## [55] vctrs_0.3.4          RColorBrewer_1.1-2   tools_4.1.0
## [58] bit64_4.0.5          Biobase_2.51.0       glue_1.4.2
## [61] parallel_4.1.0       survival_3.2-7       yaml_2.2.1
## [64] AnnotationDbi_1.53.0 colorspace_2.0-0     gbRd_0.4-11
## [67] memoise_1.1.0        knitr_1.30
```