random data - figures and code

Peter Hettegger

27 9 2020

For reproducibility, the `sessionInfo()` is provided at the end of the document.

```r
rm(list = ls())

set.seed(0)

library(randRotation)
library(GGally)
## Loading required package: ggplot2
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
library(ggplot2)

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

features <- 2000

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

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

##### Raw data (no batch effect) - "p0" values

fit0 <- limma::lmFit(edata, mod1)
fit0 <- limma::eBayes(fit0)
tt0 <- limma::topTable(fit0, 2, Inf, sort.by = "none", adjust.method = "none")
p.0 <- tt0$P.Value
```

```r
################### add batch effect ####################

## additive and multiplicative batch effect
edata <- t( t(edata)*rep(c(0.8,1,1.2), c(10,10,10)) + rep(c(-2,0,2), c(10,10,10)) )

#########################################################
```

```r
##### ComBat - "p ComBat" values
edata.combat <- sva::ComBat(edata, pdata$batch, mod1)
## Found3batches
## Adjusting for1covariate(s) or covariate level(s)
```

```r
## Standardizing Data across genes
## Fitting L/S model and finding priors
## Finding parametric adjustments
## Adjusting the Data
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


##### Limma (batch as covariate) - "p Limma (+batch)" values

mod2 <- model.matrix(~group+batch, pdata)
fit2 <- limma::lmFit(edata, mod2)
fit2 <- limma::eBayes(fit2)
tt2 <- limma::topTable(fit2, 2, Inf, sort.by = "none", adjust.method = "none")
p.lim <- tt2$P.Value


###### ComBat with random rotation - "p ComBat - 100 rot." values

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])
}
```

```r
rs1 <- rotateStat(rr1, 100, statistic, mod1, pdata$batch)
```

```r
p.rot <- pFdr(rs1, pooled = TRUE)[,1]

ps <- cbind(p.0, p.com, p.lim, p.rot)
colnames(ps) <- c("p0", "p ComBat", "p Limma (+batch)", "p ComBat - 100 rot.")
```

```r
#### p-vals scatterplot

df1 <- data.frame(ps)
colnames(df1) <- colnames(ps)

F1 <- function(...){
    ggally_points(..., size = 0.7)+
        scale_y_log10()+
        scale_x_log10()+
        geom_abline(slope = 1, intercept = 0, lty = 2, lwd = 0.5, col = "red")
```

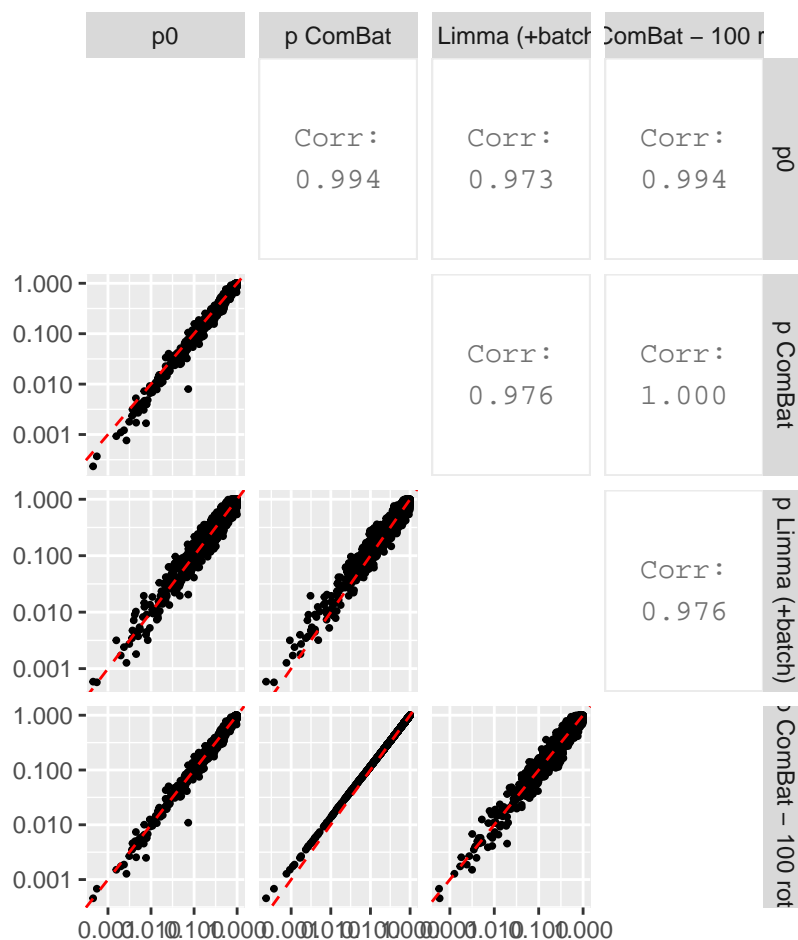**random data - figures and code**

```
}
lower.pan <- list(continuous = F1, combo = "facethist", discrete = "facetbar",
                  na = "na")

my.cor <- function(...)
    ggally_statistic(text_fn = function(x,y)formatC(cor(log(x),log(y)), digits = 3, format = "f"), title = "C
upper.pan <- list(continuous = my.cor, combo = "box_no_facet",
                         discrete = "count", na = "na")

ggpairs(df1, lower = lower.pan, upper = upper.pan,
        diag = NULL, columnLabels = colnames(ps))
```
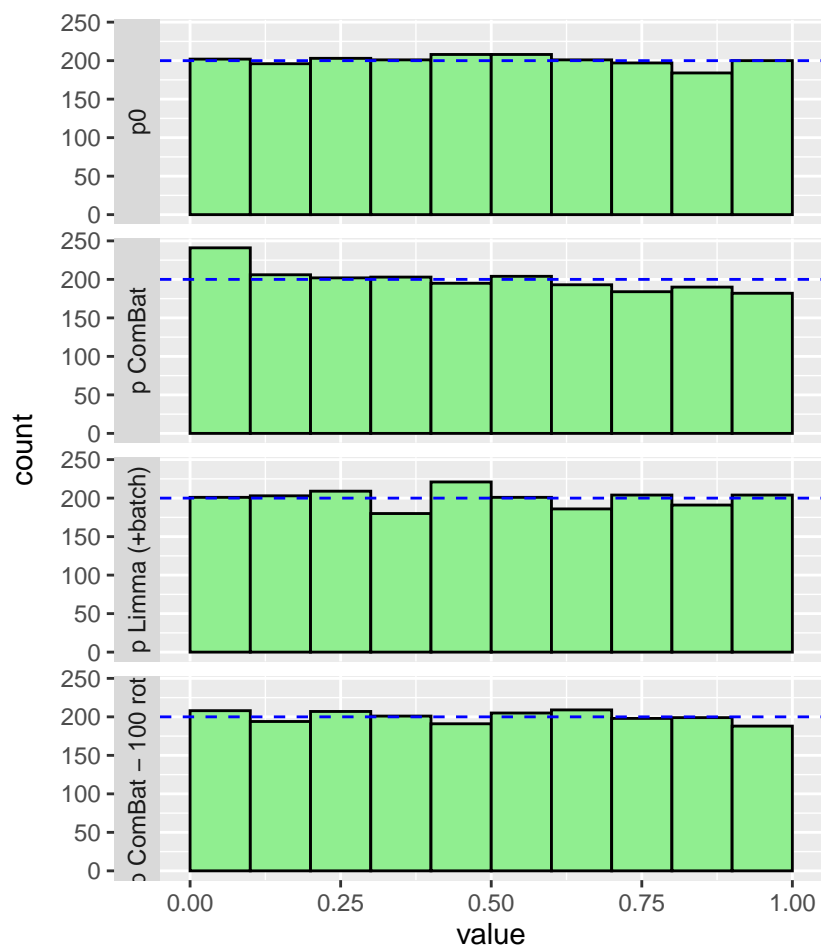


```
#### Histograms


df2 <- reshape2::melt(ps)

ggplot(df2, aes(x=value))+
    geom_histogram(colour="black", fill="lightgreen", binwidth = 0.1, boundary = 0)+
    facet_grid(Var2 ~ ., switch = "y")+
    geom_abline(slope = 0, intercept = 200, lty = 2, col = "blue")+
```
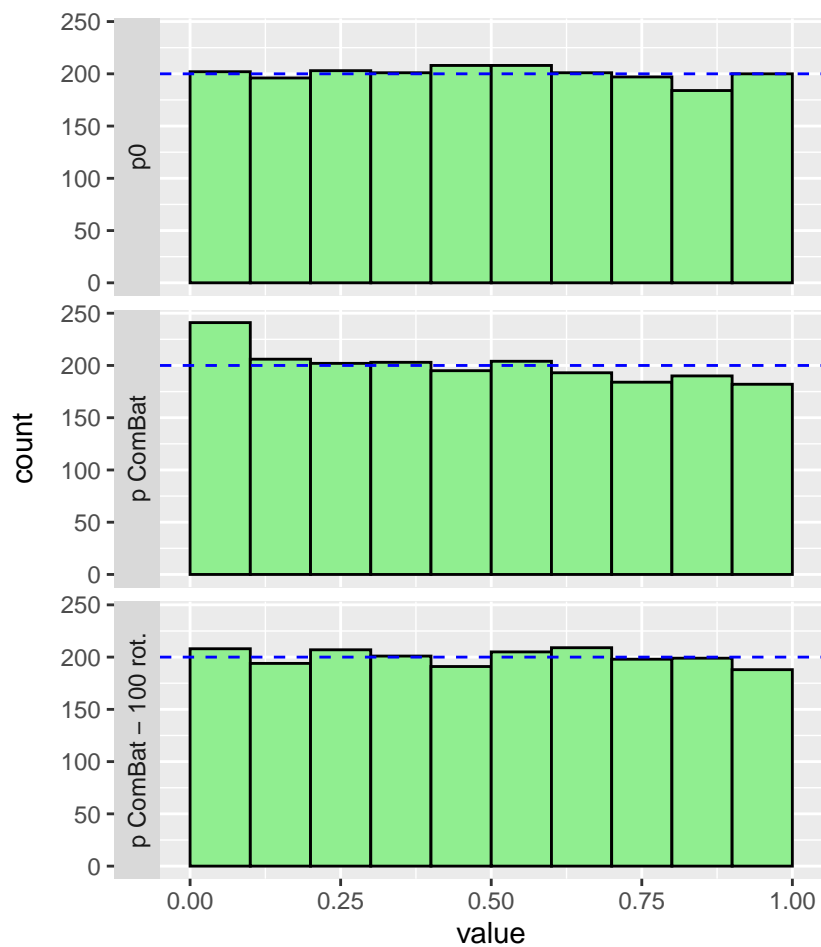
```
        theme(axis.title.y = element_text(vjust=+3.3))
```



```
df2 <- reshape2::melt(ps[,-3])

ggplot(df2, aes(x=value))+
    geom_histogram(colour="black", fill="lightgreen", binwidth = 0.1, boundary = 0)+
    facet_grid(Var2 ~ ., switch = "y")+
    geom_abline(slope = 0, intercept = 200, lty = 2, col = "blue")+
    theme(axis.title.y = element_text(vjust=+3.3))
```
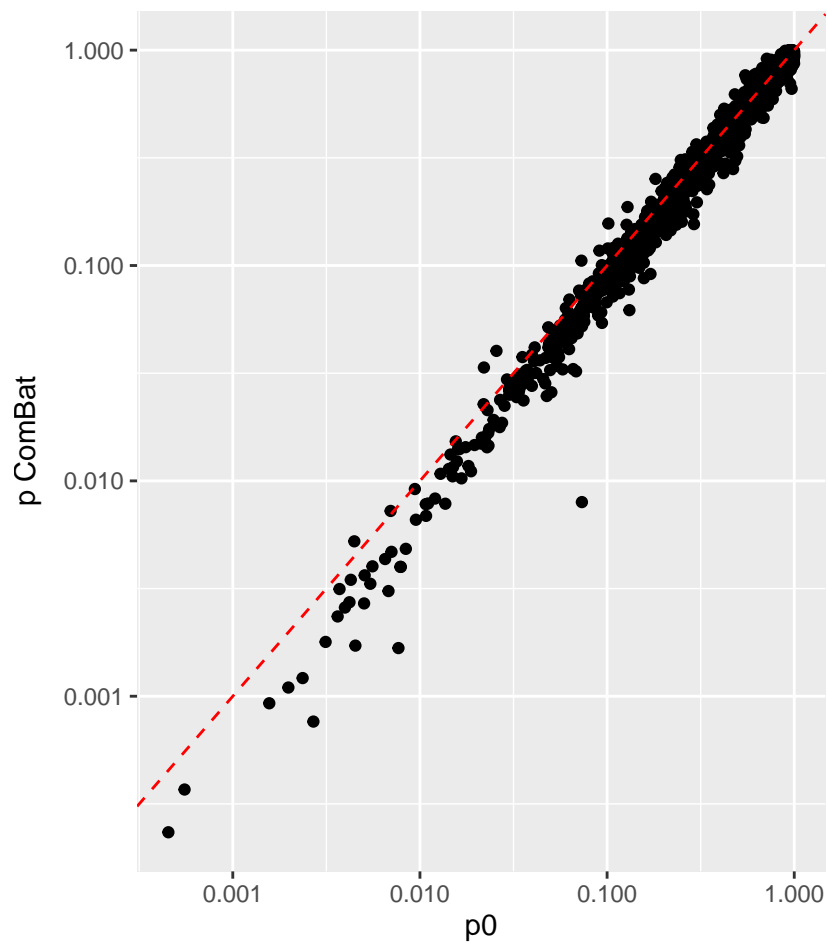
random data - figures and code


```
##### Scatterplots


ggplot(data = df1, aes(x = `p0`, y = `p ComBat`)) +
    geom_point() +
    scale_y_log10()+
    scale_x_log10()+
    theme(axis.title.y = element_text(vjust=+3.3))+
    geom_abline(slope = 1, intercept = 0, lty = 2, lwd = 0.5, col = "red")
```

random data - figures and code


```
ggplot(df1, aes(x = `p0`, y = `p ComBat - 100 rot.`)) +
    geom_point() +
    scale_y_log10()+
    scale_x_log10()+
    theme(axis.title.y = element_text(vjust=+3.3))+
    geom_abline(slope = 1, intercept = 0, lty = 2, lwd = 0.5, col = "red")
```

random data - figures and code
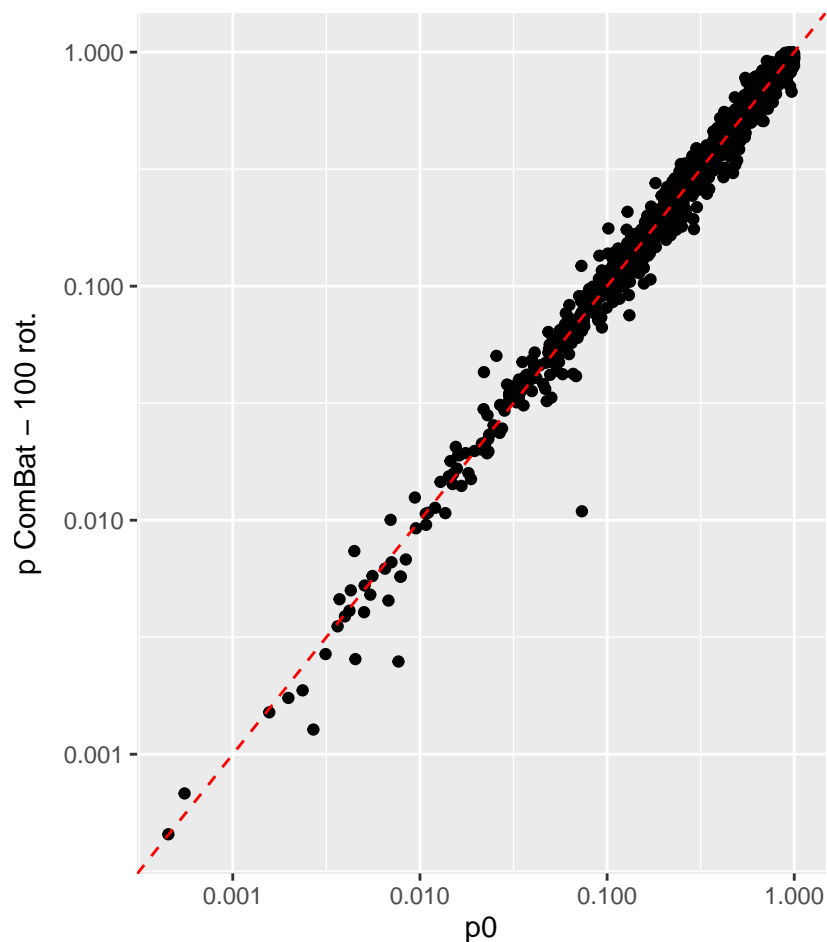
1 Session Info

```
sessionInfo()
## R version 4.0.2 (2020-06-22)
## 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     randRotation_1.1.4 BiocStyle_2.17.1
```

random data - figures and code

```
## 
## loaded via a namespace (and not attached):
##  [1] locfit_1.5-9.4       Rcpp_1.0.5          lattice_0.20-41
##  [4] digest_0.6.25        R6_2.4.1            plyr_1.8.6
##  [7] stats4_4.0.2         RSQLite_2.2.0       evaluate_0.14
## [10] sva_3.37.0           pillar_1.4.6        Rdpack_1.0.0
## [13] rlang_0.4.7          annotate_1.67.1     blob_1.2.1
## [16] S4Vectors_0.27.12    Matrix_1.2-18       rmarkdown_2.3
## [19] labeling_0.3         splines_4.0.2       BiocParallel_1.23.2
## [22] stringr_1.4.0        RCurl_1.98-1.2      bit_4.0.4
## [25] munsell_0.5.0        compiler_4.0.2      xfun_0.16
## [28] pkgconfig_2.0.3      BiocGenerics_0.35.4 mgcv_1.8-31
## [31] htmltools_0.5.0      tidyselect_1.1.0    tibble_3.0.3
## [34] edgeR_3.31.4         bookdown_0.20       IRanges_2.23.10
## [37] matrixStats_0.56.0   XML_3.99-0.5        reshape_0.8.8
## [40] crayon_1.3.4         dplyr_1.0.2         withr_2.3.0
## [43] bitops_1.0-6         grid_4.0.2          nlme_3.1-148
## [46] xtable_1.8-4         gtable_0.3.0        lifecycle_0.2.0
## [49] DBI_1.1.0            magrittr_1.5        scales_1.1.1
## [52] bibtex_0.4.2.3       stringi_1.4.6       reshape2_1.4.4
## [55] farver_2.0.3         genefilter_1.71.0   limma_3.45.13
## [58] ellipsis_0.3.1       generics_0.0.2      vctrs_0.3.4
## [61] RColorBrewer_1.1-2   tools_4.0.2         bit64_4.0.5
## [64] Biobase_2.49.1       glue_1.4.2          purrr_0.3.4
## [67] parallel_4.0.2       survival_3.1-12     yaml_2.2.1
## [70] AnnotationDbi_1.51.3 colorspace_1.4-1    BiocManager_1.30.10
## [73] gbRd_0.4-11          memoise_1.1.0       knitr_1.30
```