7659HW1

Guannan Shen

September 17, 2018

Contents

```r
library(readr)
library(pwr)
library(ssize)
```

```
## Loading required package: gdata
```

```
## gdata: read.xls support for 'XLS' (Excel 97-2004) files ENABLED.
```

```
##
```

```
## gdata: read.xls support for 'XLSX' (Excel 2007+) files ENABLED.
```

```
##
## Attaching package: 'gdata'
```

```
## The following object is masked from 'package:stats':
##
##     nobs
```

```
## The following object is masked from 'package:utils':
##
##     object.size
```

```
## The following object is masked from 'package:base':
##
##     startsWith
```

```
## Loading required package: xtable
```

```r
library(samr)
```

```
## Loading required package: impute
```

```
## Loading required package: matrixStats
```

```r
library(tidyverse)
```

```
## -- Attaching packages -------------------------------------------------------------------------- tid
```

```
## √ ggplot2 3.0.0     √ purrr   0.2.5
## √ tibble  1.4.2     √ dplyr   0.7.6
## √ tidyr   0.8.1     √ stringr 1.3.1
## √ ggplot2 3.0.0     √ forcats 0.3.0
```

```
## -- Conflicts ----------------------------------------------------------------------------- tidyverse_
## x dplyr::combine() masks gdata::combine()
## x dplyr::count()   masks matrixStats::count()
## x dplyr::filter()  masks stats::filter()
## x dplyr::first()   masks gdata::first()
## x purrr::keep()    masks gdata::keep()
## x dplyr::lag()     masks stats::lag()
## x dplyr::last()    masks gdata::last()
```

```
## sample size calculation Include the assumptions, null and
## alternative hypothesis, statistical test and expected
## number of false positives.  alpha = 0.001, beta = 0.05,
## 0.10, 0.15, 0.20, diff = 1 and sigma = 0.50 based on log2
## scale 20,000 probe sets Design and Analysis of DNA
## Microatray Investigations, equation 3.2
beta <- c(0.2, 0.15, 0.1, 0.05)
m_ssize <- sapply(beta, function(b) {
    n = 4 * (qnorm(0.001/2, 0, 1) + qnorm(b/2, 0, 1))^2/(1/0.5)^2
    return(ceiling(n/2))
})
fal_po <- 0.001 * 20000
m_ssize <- c(m_ssize, fal_po)
Power_Levels <- c("0.8", "0.85", "0.9", "0.95", "False Positive (No.)")
m_ssize <- data.frame(`Power Levels` = Power_Levels, `Sample Size` = m_ssize)
row.names(m_ssize) <- c("alpha = 0.001", "delta = 1 (log2)",
    "sigma = 0.5 (log2)", "", " ")
kable(m_ssize, caption = "Two-group Microarray Sample Size Summary",
    row.names = TRUE, col.names = c("Power Levels", "Sample Size per Group"),
    align = "c")
```

Table 1: Two-group Microarray Sample Size Summary

	Power Levels	Sample Size per Group
alpha = 0.001	0.8	11
delta = 1 (log2)	0.85	12
sigma = 0.5 (log2)	0.9	13
	0.95	14
	False Positive (No.)	20

```
# sample size comparison pwr
pwr_ss <- c(pwr.t.test(d = 1/0.5, sig.level = 0.001, power = 0.8)$n,
    pwr.t.test(d = 1/0.5, sig.level = 0.001, power = 0.95)$n)
ceiling(pwr_ss)
```

```
## [1] 12 15
```

```
pwr.t.test(d = 1/0.5, sig.level = 0.001, power = 0.8)
```

```
##
##      Two-sample t test power calculation
##
##               n = 11.3292
##               d = 2
##       sig.level = 0.001
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

```
paste("there is no pi0 in pwr.t.test")
```

```
## [1] "there is no pi0 in pwr.t.test"
```

```r
# ssize
power.t.test.FDR(sd = 0.5, delta = 1, FDR.level = 0.05, power = 0.8,
    pi0 = 0.8, type = "two.sample", alternative = "two.sided")
```

```
##
##      Two-sample t test power calculation
##
##               n = 7.578653
##           delta = 1
##              sd = 0.5
##       FDR.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

```r
power.t.test.FDR(sd = 0.5, delta = 1, FDR.level = 0.05, power = 0.95,
    pi0 = 0.9, type = "two.sample", alternative = "two.sided")
```

```
##
##      Two-sample t test power calculation
##
##               n = 11.81863
##           delta = 1
##              sd = 0.5
##       FDR.level = 0.05
##           power = 0.95
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

```r
power.t.test.FDR(sd = 0.5, delta = 1, FDR.level = 0.05, power = 0.95,
    pi0 = 0.8, type = "two.sample", alternative = "two.sided")
```

```
##
##      Two-sample t test power calculation
##
##               n = 10.28735
##           delta = 1
##              sd = 0.5
##       FDR.level = 0.05
##           power = 0.95
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

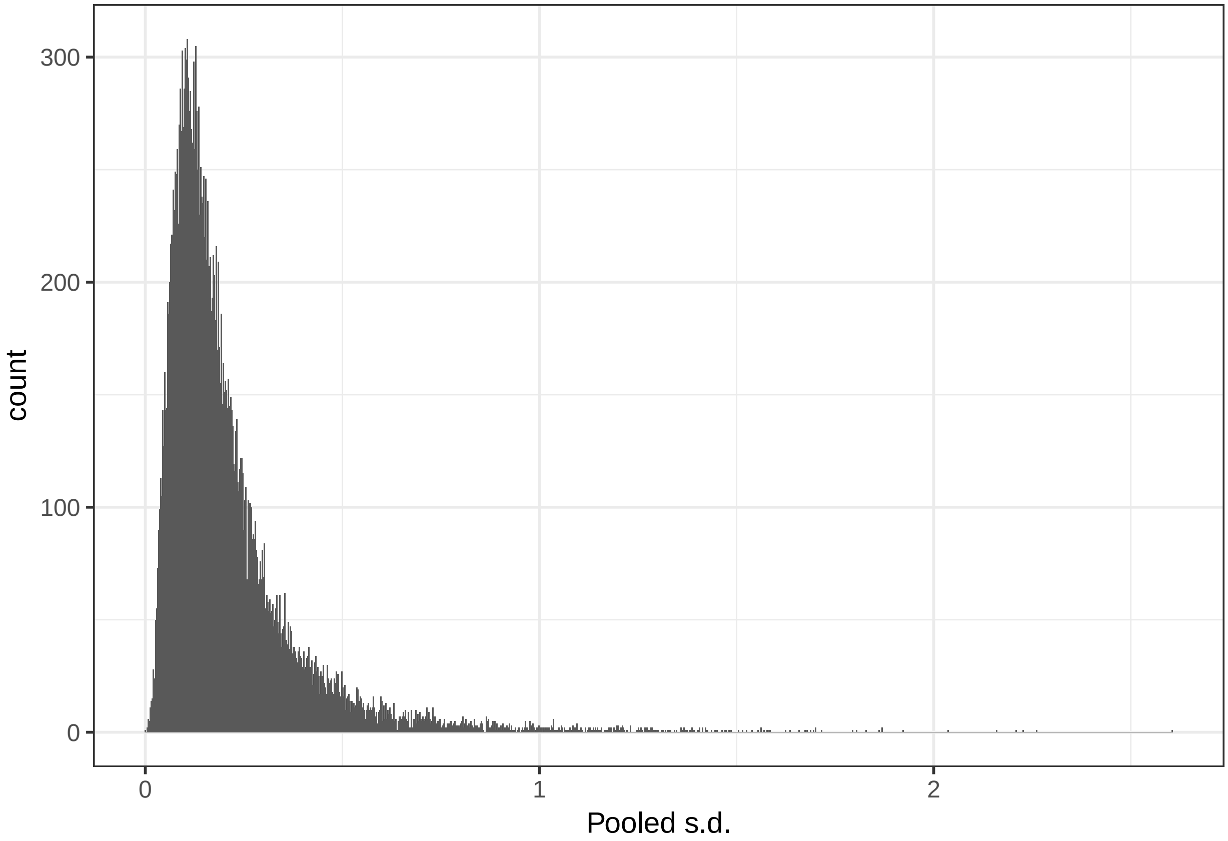
```r
power.t.test.FDR(sd = 0.5, delta = 1, FDR.level = 0.05, power = 0.8,
    pi0 = 0.9, type = "two.sample", alternative = "two.sided")
```

```
##
##      Two-sample t test power calculation
##
##               n = 8.872857
##           delta = 1
##              sd = 0.5
##       FDR.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

```
paste("pi0 proportion of non-differentially expressed genes")
```

```
## [1] "pi0 proportion of non-differentially expressed genes"
```

```
# pooled standard deviations for question 3
sdvalues <- read.table("~/Documents/Stats/CIDA_OMICs/7659Stats_Genetics/HW1/sdvalues.txt",
    quote = "\"", comment.char = "")
ggplot(sdvalues, aes(sdvalues[, 2])) + geom_histogram(bins = 1000) +
    theme_bw() + labs(x = "Pooled s.d.")
```



```
ggsave("hist_sd.png", dpi = 300)
```

```
## Saving 6.5 x 4.5 in image
```

```
sdsize <- ssize(sd = sdvalues[, 2], delta = 1, sig.level = 0.001,
    power = 0.8, alpha.correct = "None")
```

```
## ....................................................................................................
```

```
png(filename = "sdsize.png")
ssize.plot(sdsize, marks = c(4, 10, 20))
dev.off()
```

```
## pdf
##   2
```

```
# e samr
x <- as.matrix(read.table("~/Documents/Stats/CIDA_OMICs/7659Stats_Genetics/HW1/arraydata.txt",
    row.names = 1, header = TRUE))
data <- list(x = x, y = c(rep(1, 4), rep(2, 4)), geneid = row.names(x),
    genenames = row.names(x), logged2 = TRUE)
samr.obj <- samr(data, resp.type = "Two class unpaired", nperms = 100)
```

```
## perm= 1
## perm= 2
## perm= 3
## perm= 4
## perm= 5
## perm= 6
## perm= 7
## perm= 8
## perm= 9
## perm= 10
## perm= 11
## perm= 12
## perm= 13
## perm= 14
## perm= 15
## perm= 16
## perm= 17
## perm= 18
## perm= 19
## perm= 20
## perm= 21
## perm= 22
## perm= 23
## perm= 24
## perm= 25
## perm= 26
## perm= 27
## perm= 28
## perm= 29
## perm= 30
## perm= 31
## perm= 32
## perm= 33
## perm= 34
## perm= 35
## perm= 36
## perm= 37
## perm= 38
## perm= 39
## perm= 40
## perm= 41
## perm= 42
## perm= 43
## perm= 44
## perm= 45
## perm= 46
## perm= 47
## perm= 48
## perm= 49
## perm= 50
## perm= 51
## perm= 52
## perm= 53
## perm= 54
## perm= 55
## perm= 56
## perm= 57
## perm= 58
## perm= 59
## perm= 60
## perm= 61
## perm= 62
## perm= 63
## perm= 64
## perm= 65
## perm= 66
## perm= 67
## perm= 68
## perm= 69
## perm= 70
## perm= 71
## perm= 72
## perm= 73
## perm= 74
## perm= 75
## perm= 76
## perm= 77
## perm= 78
## perm= 79
## perm= 80
## perm= 81
## perm= 82
## perm= 83
## perm= 84
## perm= 85
## perm= 86
## perm= 87
## perm= 88
## perm= 89
## perm= 90
## perm= 91
## perm= 92
## perm= 93
## perm= 94
## perm= 95
## perm= 96
## perm= 97
## perm= 98
## perm= 99
## perm= 100
```

```
png("samr.png")
samr.assess.samplesize.plot(samr.assess.samplesize(samr.obj,
    data, dif = 1, samplesize.factors = c(2, 3, 4, 5)))
```

```
## NULL
```

```
dev.off()
```

```
## pdf
##   2
```