

Tandom Power Calculations

Initialization

```
library(dplyr)
library(reshape2)
library(ggplot2)

library(pwr)
```

```
wd <- "/Users/pascaltimshel/Dropbox/011_Work/work-PersLab/2017-01 - TP Tandem Grant power calculation/"
setwd(wd)
```

We start by reading in the data.

```
### Load data
file.data <- "ADIGEN_metabolomics.csv"
df.raw <- read.table(file.data, header=T, sep=",")
# head(df.raw)
```

We also need to process the data.

Process data

```
### Process data
df.data <- df.raw %>% mutate(metaID = paste(df.raw$ID, df.raw$Metabolite.name, sep="|"))
# df.data <- df.data %>% filter(!grepl("Unknown", Metabolite.name)) # if you only want 166 known metabo
df.data <- df.data %>% select(metaID, starts_with("X"))

### Melt data
df.data.melt <- melt(df.data, id.vars="metaID")
str(df.data.melt)

## 'data.frame': 313892 obs. of 3 variables:
## $ metaID : chr "16|3-Hydroxybutyric acid, 2TMS" "24|Alanine, 2TMS" "7|Arachidonic acid, TMS" "18|
## $ variable: Factor w/ 388 levels "X4334","X4340",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ value : num 52 305.4 42.9 76.4 2400.1 ...
```

Count

```
### count number of samples
length(unique(df.data.melt$variable)) # --> 388 samples

## [1] 388

### count number of metabolites
length(unique(df.data.melt$metaID)) # --> 809 metabolites

## [1] 809
```

Log-transform data

$\log_2(x+1)$ transform the data to make the normality assumptions hold.

```
df.data.melt$value_log2 <- log2(df.data.melt$value+1)
```

Summary stat calculation

Calculate mean and sd

```
df.summary <- df.data.melt %>%  
  group_by(metaID) %>%  
  summarise(mean=mean(value_log2, na.rm=T), sd=sd(value_log2, na.rm=T)) %>%  
  arrange(desc(mean))
```

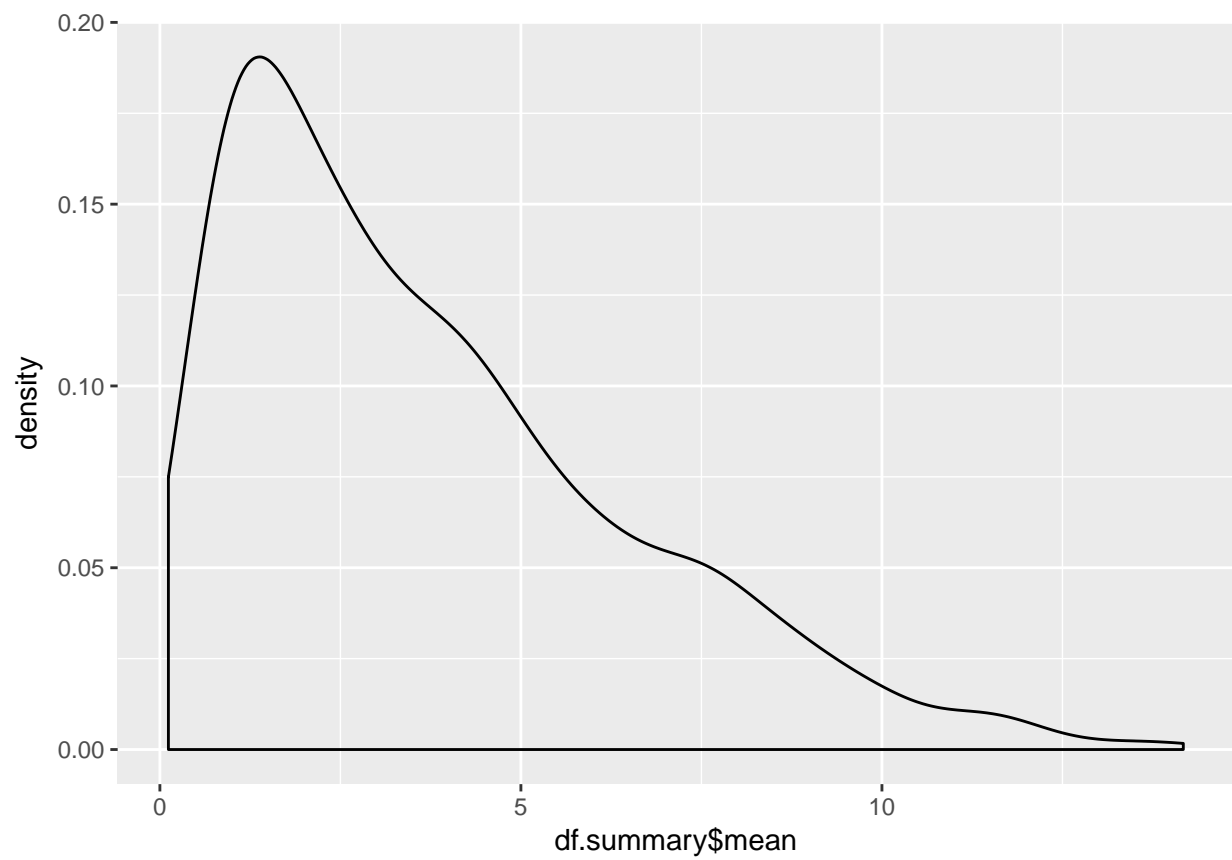
show distributive stats (median, mean, ...)

```
summary(df.summary)
```

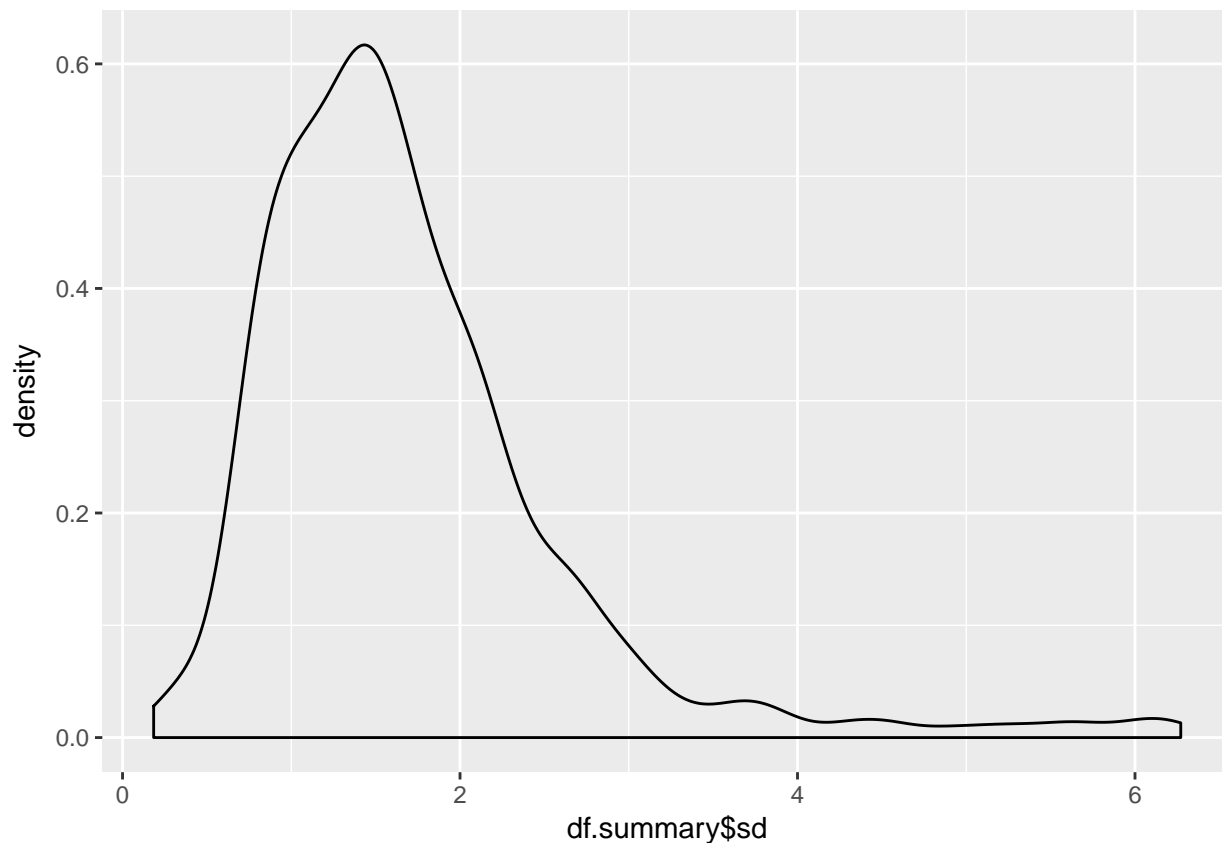
##	metaID	mean	sd
##	Length:809	Min. : 0.1177	Min. :0.1845
##	Class :character	1st Qu.: 1.4857	1st Qu.:1.1029
##	Mode :character	Median : 3.0216	Median :1.5149
##		Mean : 3.7302	Mean :1.7123
##		3rd Qu.: 5.2202	3rd Qu.:2.0452
##		Max. :14.1739	Max. :6.2712

plot distributions of mean and standard deviation

```
print(qplot(df.summary$mean, geom="density"))
```



```
print(qplot(df.summary$sd, geom="density"))
```



Difference in distribution for known and unknown metabolites

We observe little difference from using only known metabolites to estimate the median variance.

```
#### STATS for known and unknown metabolites
#      metaID          mean          sd
# Length:809      Min.   : 0.1177   Min.   :0.1845
# Class :character 1st Qu.: 1.4857   1st Qu.:1.1029
# Mode  :character Median : 3.0216   Median :1.5149 <---
#                      Mean  : 3.7302   Mean   :1.7123
#                      3rd Qu.: 5.2202   3rd Qu.:2.0452
#                      Max.   :14.1739   Max.   :6.2712
#
#### STATS for known metabolites only
#      metaID          mean          sd
# Length:166      Min.   : 0.4873   Min.   :0.1845
# Class :character 1st Qu.: 3.0785   1st Qu.:0.8529
# Mode  :character Median : 5.7052   Median :1.3347 <---
#                      Mean  : 5.7022   Mean   :1.7671
#                      3rd Qu.: 7.7978   3rd Qu.:2.1486
#                      Max.   :14.1739   Max.   :6.2712
```

Power Analysis

Power analysis in R: <http://www.statmethods.net/stats/power.html>

Our best estimates of effect size

Cohen, J. (1988) suggests that d values of 0.2, 0.5, and 0.8 represent small, medium, and large effect sizes respectively.

Pooled variance estimate: We use the median of the standard deviation for metabolites in the ADIGEN dataset. Since we assume that both groups (time points) have equal variance, the “pooling” does not change the estimation. Hence we set `sigma_pooled = 1.51` on a log2 scale.

Difference in groups (mu1-mu2): We would like to be able to detect to *two-fold increases* in metabolite levels. Let μ_{tx} denote the metabolite level at time t_x . We have $\mu_{t1}=X$ and $\mu_{t2}=2X$. Remember, we work with log-transformed data. We can now write: $\log_2(\mu_{t2}) - \log_2(\mu_{t1}) = \log_2(\mu_{t2}/\mu_{t1}) = \log_2(2X/X) = \log_2(2) = 1$ Hence we set $\mu_1 - \mu_2 = 1$ on a log2-scale.

[Side note | You could make the same “numeric example”: We could assume that the metabolite level for the first time point, T1, is the median of the mean metabolite level: $M(T1) = 3.02$ and at T2 we would then have $M(T2)=6.02$.]

```
p.mudiff <- 1 # log2 scale. this corresponds to a two-fold increase
p.sd <- 1.5149 # log2 scale

p.d <- p.mudiff/p.sd
print(sprintf("Effect size estimate for a two-fold increase in metabolite levels: %s", p.d)) # 0.66010957818998

## [1] "Effect size estimate for a two-fold increase in metabolite levels: 0.66010957818998"
```

We control the family-wise error using the Bonferroni procedure.

```
n_metabolites <- 300
alpha_FWER_corrected <- 0.05/n_metabolites
alpha_FWER_corrected

## [1] 0.0001666667
```

Sample size calculations using Cohen’s d suggestions

We see that using the best estimate for the effect size, we need 55.7 individuals at a 80% power level.

```
# our best estimate for the effect size
pwr.t.test(n=NULL, d=0.66, sig.level=alpha_FWER_corrected, power=0.8, type="paired", alternative="two.sided")

##
##      Paired t test power calculation
##
##              n = 55.70991
##              d = 0.66
##      sig.level = 0.0001666667
##              power = 0.8
##      alternative = two.sided
##
## NOTE: n is number of *pairs*

# large effect
pwr.t.test(n=NULL, d=0.8, sig.level=alpha_FWER_corrected, power=0.8, type="paired", alternative="two.sided")

##
##      Paired t test power calculation
##
##              n = 40.10861
```

```
##           d = 0.8
##       sig.level = 0.0001666667
##           power = 0.8
##       alternative = two.sided
##
## NOTE: n is number of *pairs*

# medium effect
pwr.t.test(n=NULL , d=0.5 , sig.level=alpha_FWER_corrected, power=0.8, type="paired", alternative="two.sided")

##
##       Paired t test power calculation
##
##           n = 91.91355
##           d = 0.5
##       sig.level = 0.0001666667
##           power = 0.8
##       alternative = two.sided
##
## NOTE: n is number of *pairs*

# small effect
pwr.t.test(n=NULL , d=0.2 , sig.level=alpha_FWER_corrected, power=0.8, type="paired", alternative="two.sided")

##
##       Paired t test power calculation
##
##           n = 537.5624
##           d = 0.2
##       sig.level = 0.0001666667
##           power = 0.8
##       alternative = two.sided
##
## NOTE: n is number of *pairs*
```

Plot power curve

Reference: <http://www.statmethods.net/stats/power.html>

Here we define a function a function to make the power plots for paired t-tests.

```
plot_power_curve <- function(d) {
  # Plot sample size curves for detecting various effect sizes.

  nd <- length(d)

  # power values
  p <- seq(.4,.9,.1)
  np <- length(p)

  # obtain sample sizes
  samsize <- array(numeric(nd*np), dim=c(nd,np))
  for (i in 1:np){
    for (j in 1:nd){
      result <- pwr.t.test(n=NULL , d=d[j] , sig.level=alpha_FWER_corrected, power=p[i], type="paired",
        samsize[j,i] <- ceiling(result$n)
```

```

    }
  }

  # set up graph
  xrange <- range(d)
  yrange <- round(range(samsize))
  colors <- rainbow(length(p))
  p.plot <- plot(xrange, yrange, type="n",
    xlab="Cohen's Effect Size (d)",
    ylab="Sample Size (n)" )

  # add power curves
  for (i in 1:np){
    lines(d, samsize[,i], type="l", lwd=2, col=colors[i])
  }

  # add annotation (grid lines, title, legend)
  abline(v=0, h=seq(0,yrange[2],50), lty=2, col="grey89")
  abline(h=0, v=seq(xrange[1],xrange[2],.02), lty=2,
    col="grey89")
  title(sprintf("Sample Size Estimation for Paired T-test \n
    Sig=%.2g (Two-tailed)", alpha_FWER_corrected))
  legend("topright", title="Power", as.character(p),
    fill=colors)

  return(p.plot)
}

```

Now we plot the power curves

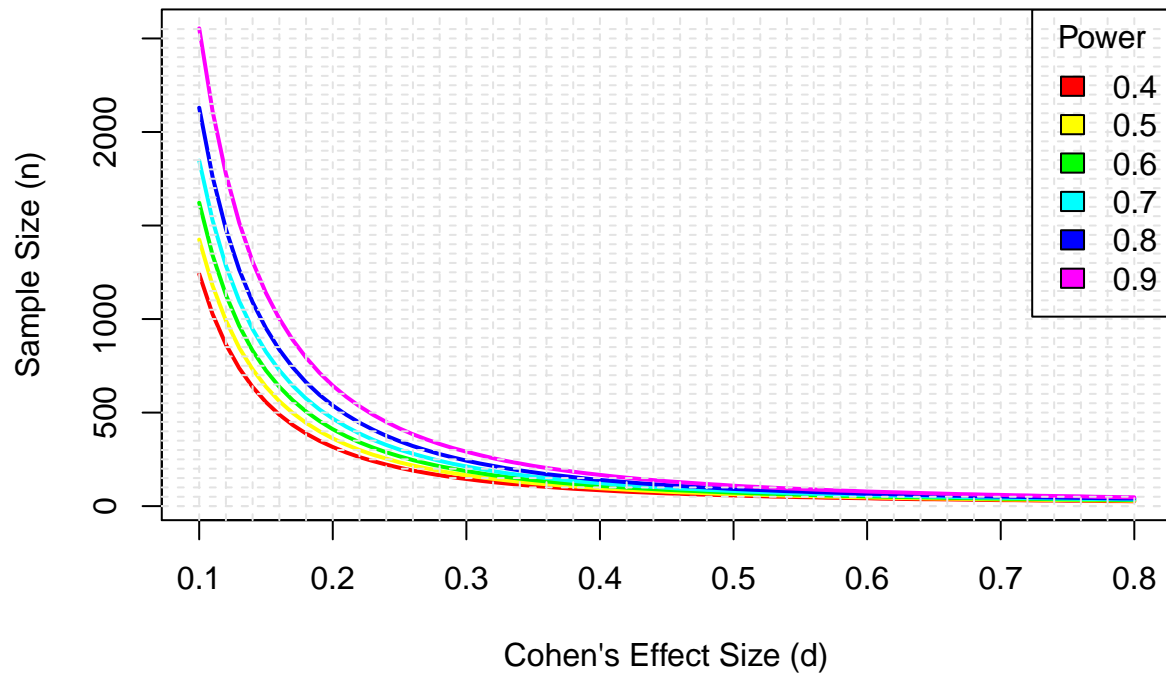
```

# range of effect size
# d <- seq(.1,.8,.01)
plot_power_curve(seq(.1,.8,.01))

```

Sample Size Estimation for Paired T-test

Sig=0.00017 (Two-tailed)

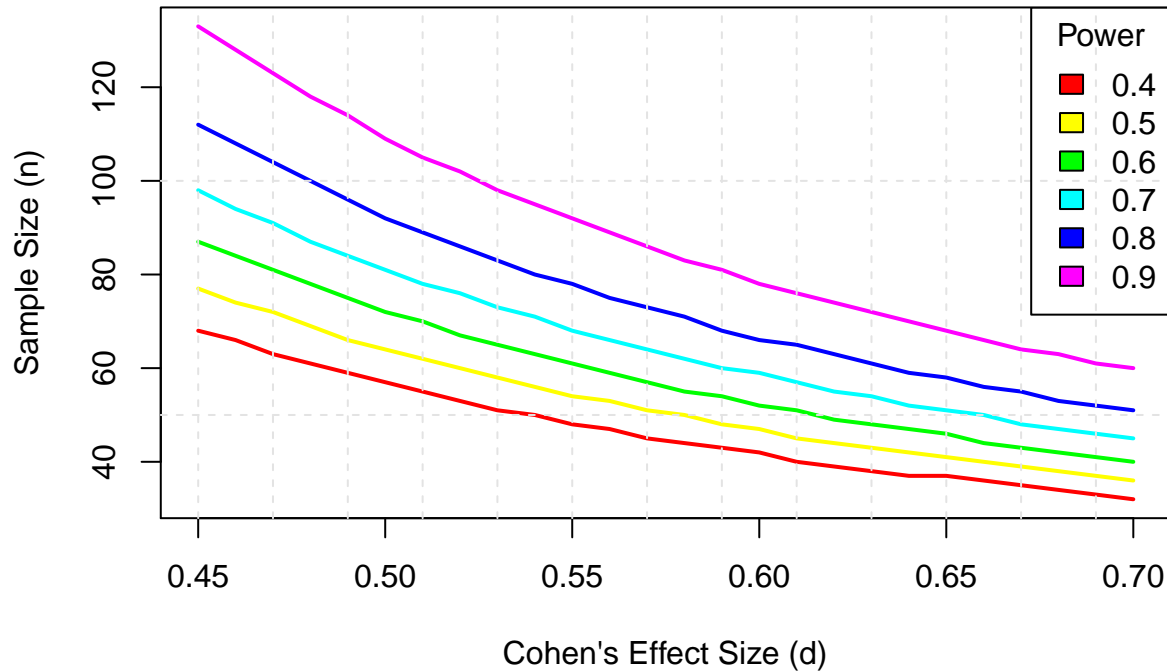


```
## NULL
```

```
plot_power_curve(seq(.45,.7,.01))
```


Sample Size Estimation for Paired T-test

Sig=0.00017 (Two-tailed)



```
## NULL
```

```
# plot_power_curve(seq(.1,.2,.01))  
# plot_power_curve(seq(.2,.4,.01))
```

Appendix

Effect size

- Two-sample t-test: Cohen's $d = (\mu_1 - \mu_2) / \sigma_{\text{pooled}}$
 - $\sigma_{\text{pooled}} = \dots$ standard pooling procedure ...
- Two-sample paired t-test
 - use d_z (see below)

How can I calculate the effect-size for a repeated measures t-test? Available from: https://www.researchgate.net/post/How_can_I_calculate_the_effect-size_for_a_repeated_measures_t-test [accessed Jan 6, 2017].

1. create a new variable of the differences between both groups ($z = \text{Time1} - \text{Time2}$),
2. obtain the mean (M_z) and standard deviation (SD_z) for this new variable,
3. divide M_z by SD_z , which will give you the effect size for dependent groups ($d_z = M_z / SD_z$)

References

- Microarray Power Analysis: <http://sph.umd.edu/departement/epib/sample-size-and-power-calculations-microarray-studies>
 - uses multiple hypothesis correction, but a fixed number of “genes”

- Online calculator: <http://www.sample-size.net/sample-size-study-paired-t-test/>
- Online calculator #2: <http://biomath.info/power/>