ABRF 2017 Satellite workshop - Hands-on 3 : Statistical hypothesis test

Meena Choi and Ting Huang

3/23/2017

Summary

- Statistical hypothesis testing by t-test (iPRG spike-in intensity).
  - iPRG study, where 6 proteins are spiked in
  - Label-free quantification based on precursor signal intensity
  - 4 conditions
- Comparison of two proportions (TCGA CRC dataset).
  - TCGA colorectal cohort
  - 95 patients with colorectal cancer
- Saving your work

---

1. Statistical hypothesis test in R

Two sample t-test for one protein with one feature

Now, we'll perform a t-test whether protein `sp|P44015|VAC2_YEAST` has a change in abundance between Condition 1 and Condition 2.

Hypothesis :

H_0 : no change in abundance, mean(Condition1) - mean(Condition2) = 0

H_a : change in abundance, mean(Condition1) - mean(Condition 2) $\neq 0$

observed $t = \frac{\textbf{difference of group means}}{\textbf{estimate of variation}} = \frac{(mean_1 - mean_2)}{SE} \sim t_{\alpha/2,df}$

Standard error, $SE = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$

n_1 : Number of replicates

$s_1^2 = \frac{1}{n_1-1}\sum(Y_{1i} - \bar{Y_{1\cdot}})^2$: Sample variance

R code

```
#load data from Section 2
load("Section2.RData")

# Let's start with one protein, named "sp|P44015|VAC2_YEAST"
oneproteindata <- iprg[iprg$Protein == "sp|P44015|VAC2_YEAST", ]

# Then, get two conditions only, because t.test only works for two groups (conditions).
oneproteindata.condition12 <- oneproteindata[oneproteindata$Condition %in%
```

```
                                                    c('Condition1', 'Condition2'), ]
unique(oneproteindata.condition12$Condition)
```

```
## [1] Condition1 Condition2
## Levels: Condition1 Condition2 Condition3 Condition4
```

```
unique(oneproteindata$Condition)
```

```
## [1] Condition1 Condition2 Condition3 Condition4
## Levels: Condition1 Condition2 Condition3 Condition4
```

```
# t test for different abundance (log2Int) between Groups (Condition)
result <- t.test(oneproteindata.condition12$Log2Intensity ~ oneproteindata.condition12$Condition,
                 var.equal=FALSE)
# show the summary of t-test including confidence level with 0.95
result
```

```
##
##  Welch Two Sample t-test
##
## data:  oneproteindata.condition12$Log2Intensity by oneproteindata.condition12$Condition
## t = 2.0608, df = 3.4001, p-value = 0.1206
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1025408  0.5619598
## sample estimates:
## mean in group Condition1 mean in group Condition2
##                 26.23632                 26.00661
```

We can redo the t-test and change the confidence level for the log2 fold change.

```
result.ci90 <- t.test(oneproteindata.condition12$Log2Intensity ~ oneproteindata.condition12$Condition,
                      var.equal=FALSE,
                      conf.level=0.9)
result.ci90
```

```
##
##  Welch Two Sample t-test
##
## data:  oneproteindata.condition12$Log2Intensity by oneproteindata.condition12$Condition
## t = 2.0608, df = 3.4001, p-value = 0.1206
## alternative hypothesis: true difference in means is not equal to 0
## 90 percent confidence interval:
##  -0.02049268  0.47991165
## sample estimates:
## mean in group Condition1 mean in group Condition2
##                 26.23632                 26.00661
```

Let's have a more detailed look at what information we can learn from the results our t-test.

```
# name of output
names(result)
```

```
## [1] "statistic"   "parameter"   "p.value"     "conf.int"    "estimate"
## [6] "null.value"  "alternative" "method"      "data.name"
```

```
# mean for each group
result$estimate
```

```
## mean in group Condition1 mean in group Condition2
##                 26.23632                 26.00661
```

```
# log2 transformed fold change between groups : Disease-Healthy
result$estimate[1]-result$estimate[2]
```

```
## mean in group Condition1
##                0.2297095
```

```
# test statistic value, T value
result$statistic
```

```
##        t
## 2.060799
```

```
# standard error
(result$estimate[1]-result$estimate[2])/result$statistic
```

```
## mean in group Condition1
##                0.1114662
```

```
# degree of freedom
result$parameter
```

```
##       df
## 3.400112
```

```
# p value for two-sides testing
result$p.value
```

```
## [1] 0.1206139
```

```
# 95% confidence interval for log2 fold change
result$conf.int
```

```
## [1] -0.1025408  0.5619598
## attr(,"conf.level")
## [1] 0.95
```

```
# p value calculation for one side
1-pt(result$statistic, result$parameter)
```

```
##          t
## 0.06030697
```

```
# p value for two sides, which is the same as pvalue from t test (result$p.value)
2*(1-pt(result$statistic, result$parameter))
```

```
##         t
## 0.1206139
```

We can also manually compute our t-test statistic using the formulas we descibed above and compare it with the `summaryresult`.

Recall the `summaryresult` we generated last section

```
summaryresult
```

```
##        Group     mean         sd         se length ciw.lower.95
## 1 Condition1 26.23632 0.10396539 0.06002444      3     26.04529
## 2 Condition2 26.00661 0.16268179 0.09392438      3     25.70770
## 3 Condition3 23.25609 0.09467798 0.05466236      3     23.08213
## 4 Condition4 20.97056 0.73140174 0.42227499      3     19.62669
##   ciw.upper.95 ciw.lower.99 ciw.upper.99
## 1     26.42734     25.88572     26.58691
## 2     26.30552     25.45800     26.55521
## 3     23.43005     22.93681     23.57537
## 4     22.31443     18.50409     23.43703
```

```
summaryresult12 <- summaryresult[1:2, ]

# test statistic, It is the same as 'result$statistic' above.
diff(summaryresult12$mean) # same as result$estimate[1]-result$estimate[2]
```

```
## [1] -0.2297095
```

```
sqrt(sum(summaryresult12$sd^2/summaryresult12$length)) # same as stand error
```

```
## [1] 0.1114662
```

```
diff(summaryresult12$mean)/sqrt(sum(summaryresult12$sd^2/summaryresult12$length))
```

```
## [1] -2.060799
```

Sample size estimation

R code

To calculate the required sample size, you'll need to know four things:

- α: confidence level
- *power*: 1 - σ, where σ is probability of a true positive discovery
- Δ: anticipated fold change
- σ: anticipated variance

Assuming equal varaince and number of samples across groups, the following formula is used for sample size estimation:

$$\frac{2\sigma^2}{n} \leq (\frac{\Delta}{z_{1-\beta} + z_{1-\alpha/2}})^2$$

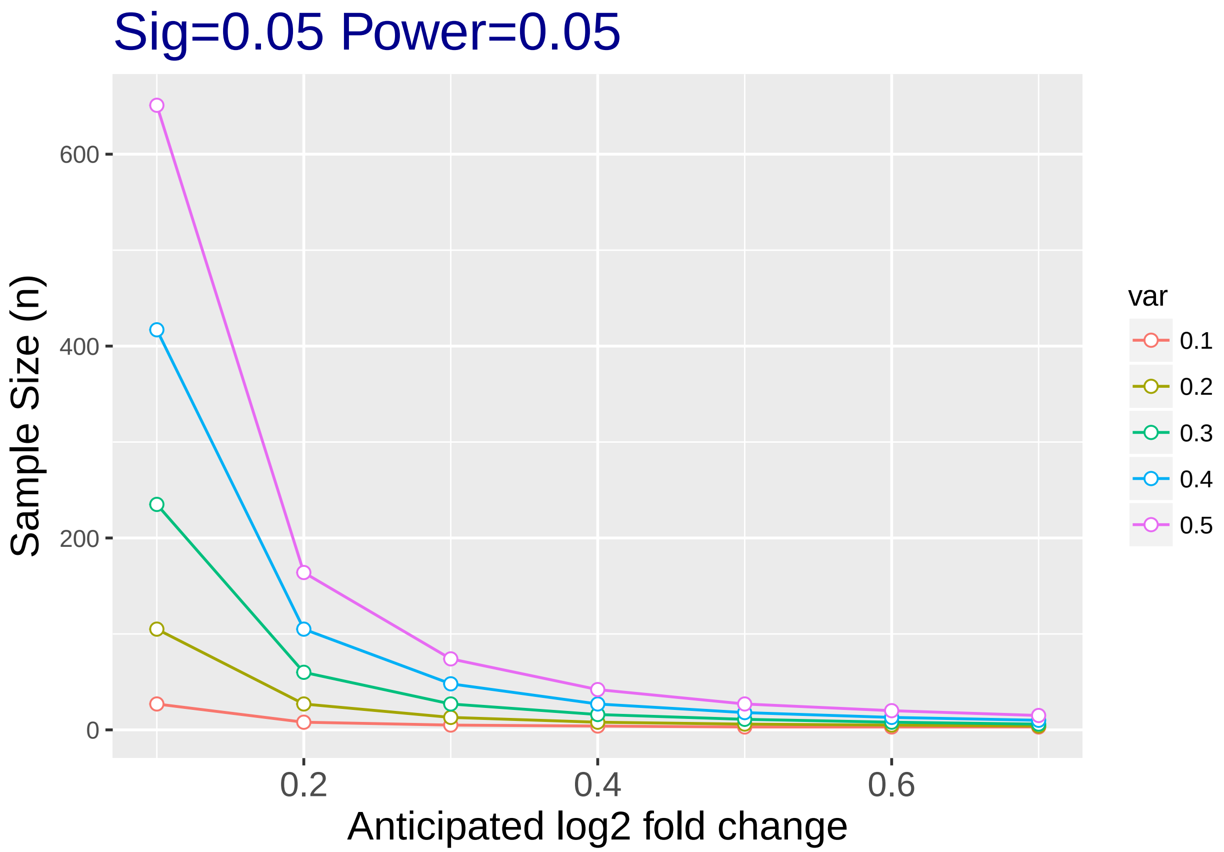
```
#install.packages("pwr")
library(pwr)

?pwr.t.test

# Significance level alpha
alpha <- 0.05

# Power = 1 - beta
power <- 0.95

# anticipated log2 fold change
delta <- 1

# anticipated variability
sigma <- 1.5

# Effect size
```

```r
d <- delta/sigma

#Sample size estimation
pwr.t.test(d = d, sig.level = alpha, power = power, type = 'two.sample')
```

```
##
##      Two-sample t test power calculation
##
##               n = 59.45416
##               d = 0.6666667
##       sig.level = 0.05
##           power = 0.95
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

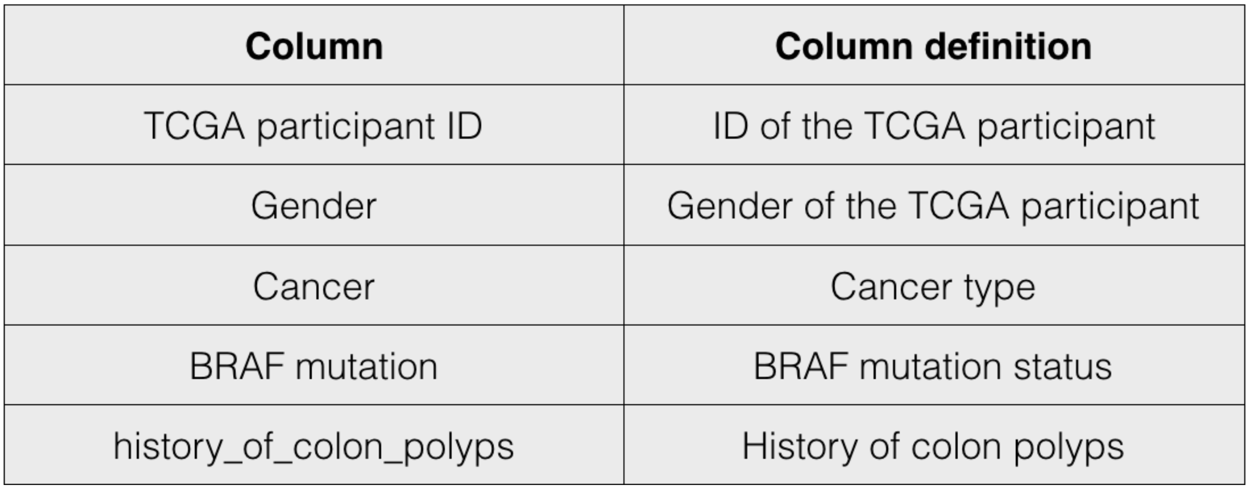
Then, we investigate the effect of required fold change and variance on the sample size estimation.

```r
# anticipated log2 fold change
delta <- seq(0.1, 0.7, .1)
nd <- length(delta)

# anticipated variability
sigma <- seq(0.1,0.5,.1)
ns <- length(sigma)

# obtain sample sizes
samsize <- matrix(0, nrow=ns*nd, ncol = 3)
counter <- 0
for (i in 1:nd){
  for (j in 1:ns){
    result <- pwr.t.test(d = delta[i]/sigma[j],
                         sig.level = alpha, power = power,
                         type = "two.sample")
    counter <- counter + 1
    samsize[counter,1] <- delta[i]
    samsize[counter,2] <- sigma[j]
    samsize[counter,3] <- ceiling(result$n)
  }
}
colnames(samsize) <- c("fd","var","value")


library(ggplot2)
samsize <- as.data.frame(samsize)
samsize$var <- as.factor(samsize$var)
ggplot(data=samsize, aes(x=fd, y=value, group = var, colour = var)) +
  geom_line() +
  geom_point(size=2, shape=21, fill="white") +
  labs(title="Sig=0.05 Power=0.05", x="Anticipated log2 fold change", y='Sample Size (n)') +
  theme(plot.title = element_text(size=20, colour="darkblue"),
        axis.title.x = element_text(size=15),
        axis.title.y = element_text(size=15),
        axis.text.x = element_text(size=13))
```



2. Comparison of two proportions in R

For part 2, we are using a new dataset, which contains the patient information from TCGA colorectal cohort. Rows in the data array are patients and columns are patient information. The column definition is shown as following:



Column	Column definition
TCGA participant ID	ID of the TCGA participant
Gender	Gender of the TCGA participant
Cancer	Cancer type
BRAF mutation	BRAF mutation status
history_of_colon_polyps	History of colon polyps

Figure 1:

2.1 Generate 2-way contingency tables

We first need to calculate 2-way contingency tables for the following tests.

```
#Dataset is from nature paper: Proteogenomic characterization of human colon and rectal cancer (Zhang e
#Load in the TCGA colorectal cancer sample informtaion
TCGA.CRC <- read.csv("TCGA_sample_information.csv")
head(TCGA.CRC)
```

```
##   TCGA.participant.ID Gender Cancer BRAF.mutation history_of_colon_polyps
## 1        TCGA-A6-3807 Female  Colon             0                      NO
## 2        TCGA-A6-3808   Male  Colon             0                     YES
## 3        TCGA-A6-3810   Male  Colon             0                     YES
## 4        TCGA-AA-3518 Female  Colon             0                      NO
## 5        TCGA-AA-3525   Male  Colon             1                      NO
## 6        TCGA-AA-3526   Male  Colon             0                     YES
```

```
#`colnames` is short for column names.
colnames(TCGA.CRC)
```

```
## [1] "TCGA.participant.ID"     "Gender"
## [3] "Cancer"                  "BRAF.mutation"
## [5] "history_of_colon_polyps"
```

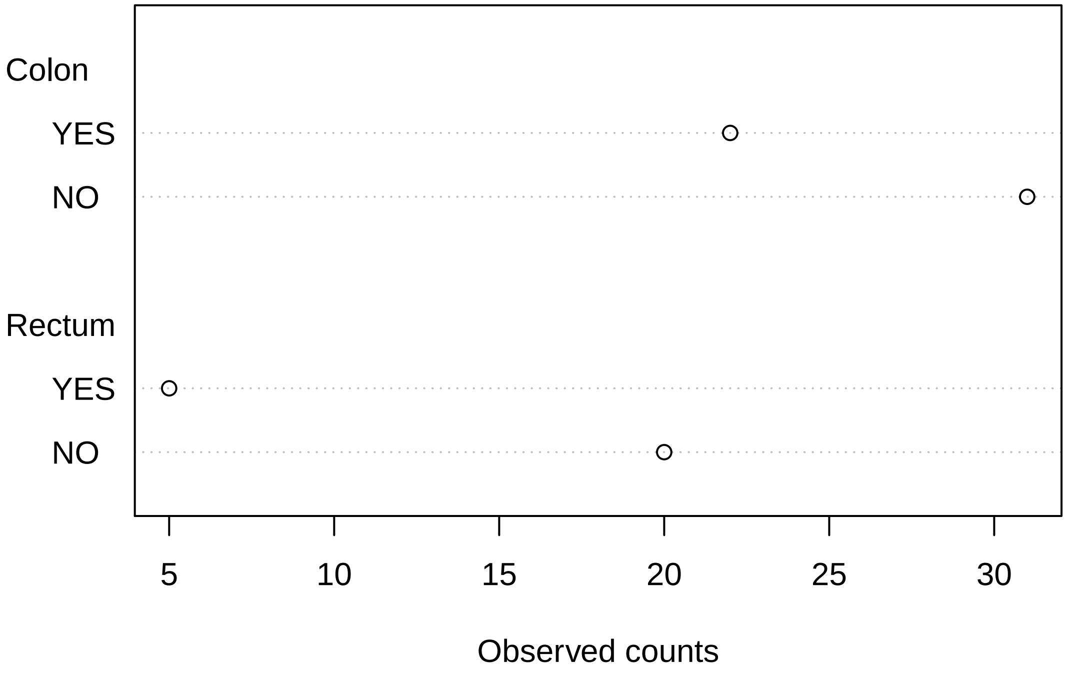
```
# Select columns from TCGA dataset:
# We are interested in the cancer type and history of colon polyps
TCGA.CRC.gc <- TCGA.CRC[, c('Cancer', 'history_of_colon_polyps')]
nrow(TCGA.CRC.gc)
```

```
## [1] 78
```

```
#Generate 2-way contingency tables
ov <- table(TCGA.CRC.gc)
ov
```

```
##         history_of_colon_polyps
## Cancer   NO YES
##   Colon  31  22
##   Rectum 20   5
```

```
#dotchart
dotchart(t(ov), xlab="Observed counts")
```



2.2 Chi-square test

Hypothesis :

H_0 : each population has the same proportion of observations, $\pi_{j|1} = ... = \pi_{j|I}$ for all j

H_a : different population has different proportion of observations$

$$\chi^2 = \sum_{i=1}^{2}\sum_{j=1}^{2} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \sim \chi^2_{(2-1)(2-1)}$$

O_{ij} : n_{ij}, which is the count within the cells

E_{ij} : $n_{i+}n_{+j}/n$, where n_{i+} is the row count sum, n_{+j} is the column count sum and n is the total count.

```
#Hypothesis: whether the proportion of patients who have history of colon polyps in the patients with c
#chi-square test
pt <- prop.test(ov)
pt
```

```
##
##  2-sample test for equality of proportions with continuity
##  correction
##
## data:  ov
## X-squared = 2.5871, df = 1, p-value = 0.1077
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.44991310  0.01972442
## sample estimates:
##    prop 1    prop 2
## 0.5849057 0.8000000
```

```
# name of output
names(pt)
```

```
## [1] "statistic"   "parameter"   "p.value"     "estimate"    "null.value" 
## [6] "conf.int"    "alternative" "method"      "data.name"
```

```
# proportion in each group
pt$estimate
```

```
##    prop 1    prop 2 
## 0.5849057 0.8000000
```

```
# test statistic value
pt$statistic
```

```
## X-squared 
##  2.587111
```

```
# degree of freedom
pt$parameter
```

```
## df 
##  1
```

2.3 Fisher's exact test

The Fisher's exact test can be used with small sample sizes. It compares distributions of counts within the 4 cells.

```
#Fisher's Exact Test
ft <- fisher.test(ov)
ft
```

```
## 
##  Fisher's Exact Test for Count Data
## 
## data:  ov
## p-value = 0.07734
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.09057002 1.18269896
## sample estimates:
## odds ratio 
##  0.3567853
```

```
# odds ratio
ft$estimate
```

```
## odds ratio 
##  0.3567853
```