Analyze Protein Data

September 10, 2014

1 Cleaning the data

```
## Protein of interest for tech paper 01/27/2014
library("lme4")
library("plyr")
library("ggplot2")
library("MASS")
# read data
dat <- read.csv("239 tech paper_08_19_2014.csv")

dim(dat)

## [1] 241  66

x <- colnames(dat) # column name of data
x

##  [1] "BVAExport"                          "Image"
##  [3] "Gel_1_Run_1_Pig_1807_45034.CY3.gel" "X"
##  [5] "Gel_1_Run_1_Pig_1807_45034.CY5.gel" "X.1"
##  [7] "Gel_1_Run_2_Pig_1807_45046.CY3.gel" "X.2"
##  [9] "Gel_1_Run_2_Pig_1807_45046.CY5.gel" "X.3"
## [11] "Gel_2_Run_1_4810_45035.CY3.gel"     "X.4"
## [13] "Gel_2_Run_1_4810_45035.CY5.gel"     "X.5"
## [15] "Gel_2_Run_2_Pig_4810_45047.CY3.gel" "X.6"
## [17] "Gel_2_Run_2_Pig_4810_45047.CY5.gel" "X.7"
## [19] "Gel_3_Run_1_Pig_1209_45036.CY3.gel" "X.8"
## [21] "Gel_3_Run_1_Pig_1209_45036.CY5.gel" "X.9"
## [23] "Gel_3_Run_2_Pig_1209_45048.CY3.gel" "X.10"
## [25] "Gel_3_Run_2_Pig_1209_45048.CY5.gel" "X.11"
## [27] "Gel_4_Run_1_Pig_1906_45037.CY3.gel" "X.12"
## [29] "Gel_4_Run_1_Pig_1906_45037.CY5.gel" "X.13"
## [31] "Gel_4_Run_2_Pig_1906_45049.CY3.gel" "X.14"
## [33] "Gel_4_Run_2_Pig_1906_45049.CY5.gel" "X.15"
## [35] "Gel_5_Run_1_Pig_3908_45038.CY3.gel" "X.16"
## [37] "Gel_5_Run_1_Pig_3908_45038.CY5.gel" "X.17"
## [39] "Gel_5_Run_2_Pig_3908_45050.CY3.gel" "X.18"
## [41] "Gel_5_Run_2_Pig_3908_45050.CY5.gel" "X.19"
## [43] "Gel_6_Run_1_Pig_3106_45039.CY3.gel" "X.20"
## [45] "Gel_6_Run_1_Pig_3106_45039.CY5.gel" "X.21"
## [47] "Gel_6_Run_2_Pig_3106_45051.CY3.gel" "X.22"
## [49] "Gel_6_Run_2_Pig_3106_45051.CY5.gel" "X.23"
```

```
## [51] "Gel_7_Run_1_Pig_2107_45040.CY3.gel" "X.24"
## [53] "Gel_7_Run_1_Pig_2107_45040.CY5.gel" "X.25"
## [55] "Gel_7_Run_2_Pig_2107_45052.CY3.gel" "X.26"
## [57] "Gel_7_Run_2_Pig_2107_45052.CY5.gel" "X.27"
## [59] "Gel_8_Run_1_Pig_2712_45041.CY3.gel" "X.28"
## [61] "Gel_8_Run_1_Pig_2712_45041.CY5.gel" "X.29"
## [63] "Gel_8_Run_2_Pig_2712_44495.CY3.gel" "X.30"
## [65] "Gel_8_Run_2_Pig_2712_44495.CY5.gel" "X.31"
```

```
y <- c( "Image","X", paste("X.", 1:31, sep = "")) # Column name unsued

# Column name used in analysis including the id of protein and 16 gels
# of 8 animals, each gel runs twice

use_col <- setdiff(x,y)
use_col

##  [1] "BVAExport"                          "Gel_1_Run_1_Pig_1807_45034.CY3.gel"
##  [3] "Gel_1_Run_1_Pig_1807_45034.CY5.gel" "Gel_1_Run_2_Pig_1807_45046.CY3.gel"
##  [5] "Gel_1_Run_2_Pig_1807_45046.CY5.gel" "Gel_2_Run_1_4810_45035.CY3.gel"
##  [7] "Gel_2_Run_1_4810_45035.CY5.gel"     "Gel_2_Run_2_Pig_4810_45047.CY3.gel"
##  [9] "Gel_2_Run_2_Pig_4810_45047.CY5.gel" "Gel_3_Run_1_Pig_1209_45036.CY3.gel"
## [11] "Gel_3_Run_1_Pig_1209_45036.CY5.gel" "Gel_3_Run_2_Pig_1209_45048.CY3.gel"
## [13] "Gel_3_Run_2_Pig_1209_45048.CY5.gel" "Gel_4_Run_1_Pig_1906_45037.CY3.gel"
## [15] "Gel_4_Run_1_Pig_1906_45037.CY5.gel" "Gel_4_Run_2_Pig_1906_45049.CY3.gel"
## [17] "Gel_4_Run_2_Pig_1906_45049.CY5.gel" "Gel_5_Run_1_Pig_3908_45038.CY3.gel"
## [19] "Gel_5_Run_1_Pig_3908_45038.CY5.gel" "Gel_5_Run_2_Pig_3908_45050.CY3.gel"
## [21] "Gel_5_Run_2_Pig_3908_45050.CY5.gel" "Gel_6_Run_1_Pig_3106_45039.CY3.gel"
## [23] "Gel_6_Run_1_Pig_3106_45039.CY5.gel" "Gel_6_Run_2_Pig_3106_45051.CY3.gel"
## [25] "Gel_6_Run_2_Pig_3106_45051.CY5.gel" "Gel_7_Run_1_Pig_2107_45040.CY3.gel"
## [27] "Gel_7_Run_1_Pig_2107_45040.CY5.gel" "Gel_7_Run_2_Pig_2107_45052.CY3.gel"
## [29] "Gel_7_Run_2_Pig_2107_45052.CY5.gel" "Gel_8_Run_1_Pig_2712_45041.CY3.gel"
## [31] "Gel_8_Run_1_Pig_2712_45041.CY5.gel" "Gel_8_Run_2_Pig_2712_44495.CY3.gel"
## [33] "Gel_8_Run_2_Pig_2712_44495.CY5.gel"

dat_used <- as.matrix(dat[, use_col])
dim(dat_used)

## [1] 241  33
```

```
# The first row is sample  classification, the second row is the name of Std. Abund,
# therefore the data actually in use is the dat_used except the first 2 rows
dat_final <- matrix(as.numeric(dat_used[-c(1:2),]),
                    nrow = nrow(dat_used)-2,
                    ncol = ncol(dat_used),
                    byrow = F)
dim(dat_final)

## [1] 239  33
```

```
# sample type of each sample

group <- dat_used[1,-1]
group
```

```
## Gel_1_Run_1_Pig_1807_45034.CY3.gel Gel_1_Run_1_Pig_1807_45034.CY5.gel
##                            "Whole"                           "Depleted"
## Gel_1_Run_2_Pig_1807_45046.CY3.gel Gel_1_Run_2_Pig_1807_45046.CY5.gel
##                            "Whole"                           "Depleted"
##     Gel_2_Run_1_4810_45035.CY3.gel     Gel_2_Run_1_4810_45035.CY5.gel
##                         "Depleted"                            "Whole"
## Gel_2_Run_2_Pig_4810_45047.CY3.gel Gel_2_Run_2_Pig_4810_45047.CY5.gel
##                         "Depleted"                            "Whole"
## Gel_3_Run_1_Pig_1209_45036.CY3.gel Gel_3_Run_1_Pig_1209_45036.CY5.gel
##                            "Whole"                           "Depleted"
## Gel_3_Run_2_Pig_1209_45048.CY3.gel Gel_3_Run_2_Pig_1209_45048.CY5.gel
##                            "Whole"                           "Depleted"
## Gel_4_Run_1_Pig_1906_45037.CY3.gel Gel_4_Run_1_Pig_1906_45037.CY5.gel
##                         "Depleted"                            "Whole"
## Gel_4_Run_2_Pig_1906_45049.CY3.gel Gel_4_Run_2_Pig_1906_45049.CY5.gel
##                         "Depleted"                            "Whole"
## Gel_5_Run_1_Pig_3908_45038.CY3.gel Gel_5_Run_1_Pig_3908_45038.CY5.gel
##                            "Whole"                           "Depleted"
## Gel_5_Run_2_Pig_3908_45050.CY3.gel Gel_5_Run_2_Pig_3908_45050.CY5.gel
##                            "Whole"                           "Depleted"
## Gel_6_Run_1_Pig_3106_45039.CY3.gel Gel_6_Run_1_Pig_3106_45039.CY5.gel
##                         "Depleted"                            "Whole"
## Gel_6_Run_2_Pig_3106_45051.CY3.gel Gel_6_Run_2_Pig_3106_45051.CY5.gel
##                         "Depleted"                            "Whole"
## Gel_7_Run_1_Pig_2107_45040.CY3.gel Gel_7_Run_1_Pig_2107_45040.CY5.gel
##                            "Whole"                           "Depleted"
## Gel_7_Run_2_Pig_2107_45052.CY3.gel Gel_7_Run_2_Pig_2107_45052.CY5.gel
##                            "Whole"                           "Depleted"
## Gel_8_Run_1_Pig_2712_45041.CY3.gel Gel_8_Run_1_Pig_2712_45041.CY5.gel
##                         "Depleted"                            "Whole"
## Gel_8_Run_2_Pig_2712_44495.CY3.gel Gel_8_Run_2_Pig_2712_44495.CY5.gel
##                         "Depleted"                            "Whole"
```

```
# Obtain data for each sample type: depleted and whole, the first column of dat_final
# contains name of protein
deplete <- dat_final[,-1][,group=="Depleted"]
dim(deplete)
```

```
## [1] 239  16
```

```
whole <- dat_final[,-1][,group=="Whole"]
dim(whole)
```

```
## [1] 239  16
```

```
# Find out which Cy is used for each run

group[group == "Depleted"]  # Cy for depleted sample : rep(c(5,5,3,3),4)
```

```
## Gel_1_Run_1_Pig_1807_45034.CY5.gel Gel_1_Run_2_Pig_1807_45046.CY5.gel
```

```
##                           "Depleted"                           "Depleted"
##     Gel_2_Run_1_4810_45035.CY3.gel Gel_2_Run_2_Pig_4810_45047.CY3.gel
##                           "Depleted"                           "Depleted"
## Gel_3_Run_1_Pig_1209_45036.CY5.gel Gel_3_Run_2_Pig_1209_45048.CY5.gel
##                           "Depleted"                           "Depleted"
## Gel_4_Run_1_Pig_1906_45037.CY3.gel Gel_4_Run_2_Pig_1906_45049.CY3.gel
##                           "Depleted"                           "Depleted"
## Gel_5_Run_1_Pig_3908_45038.CY5.gel Gel_5_Run_2_Pig_3908_45050.CY5.gel
##                           "Depleted"                           "Depleted"
## Gel_6_Run_1_Pig_3106_45039.CY3.gel Gel_6_Run_2_Pig_3106_45051.CY3.gel
##                           "Depleted"                           "Depleted"
## Gel_7_Run_1_Pig_2107_45040.CY5.gel Gel_7_Run_2_Pig_2107_45052.CY5.gel
##                           "Depleted"                           "Depleted"
## Gel_8_Run_1_Pig_2712_45041.CY3.gel Gel_8_Run_2_Pig_2712_44495.CY3.gel
##                           "Depleted"                           "Depleted"
```

```
group[group == "Whole"]  # Cy for whole sample : rep(c(3,3,5,5),4)
```

```
## Gel_1_Run_1_Pig_1807_45034.CY3.gel Gel_1_Run_2_Pig_1807_45046.CY3.gel
##                           "Whole"                           "Whole"
##     Gel_2_Run_1_4810_45035.CY5.gel Gel_2_Run_2_Pig_4810_45047.CY5.gel
##                           "Whole"                           "Whole"
## Gel_3_Run_1_Pig_1209_45036.CY3.gel Gel_3_Run_2_Pig_1209_45048.CY3.gel
##                           "Whole"                           "Whole"
## Gel_4_Run_1_Pig_1906_45037.CY5.gel Gel_4_Run_2_Pig_1906_45049.CY5.gel
##                           "Whole"                           "Whole"
## Gel_5_Run_1_Pig_3908_45038.CY3.gel Gel_5_Run_2_Pig_3908_45050.CY3.gel
##                           "Whole"                           "Whole"
## Gel_6_Run_1_Pig_3106_45039.CY5.gel Gel_6_Run_2_Pig_3106_45051.CY5.gel
##                           "Whole"                           "Whole"
## Gel_7_Run_1_Pig_2107_45040.CY3.gel Gel_7_Run_2_Pig_2107_45052.CY3.gel
##                           "Whole"                           "Whole"
## Gel_8_Run_1_Pig_2712_45041.CY5.gel Gel_8_Run_2_Pig_2712_44495.CY5.gel
##                           "Whole"                           "Whole"
```

2 Function to fit a Linear Mixed Effect Model for each spot in each sample

Function to fit a linear mixed model for each spot, with fixed effects are Cy, RFI Line and the random effect is animal.

```
out_model <- function(x, depleted){ # x is the row of data (i.e., data of each protein spot)
  if (depleted == "TRUE"){
    cy <- as.factor(rep(c(5,5,3,3),4))
  } else {
    cy <- as.factor(rep(c(3,3,5,5),4))
  }
  animal <- as.factor(rep(1:8, each = 2))
  line <- as.factor(rep(c(1,2), each = 8))
  # check if all obsetvations for one Cy are missing or not

  if ((sum(is.na(x[cy==3]))==8|sum(is.na(x[cy==5]))==8) & # if all cy is missing
```

```
      (sum(is.na(x[line==1]))==8|sum(is.na(x[line==2]))==8)){# if all Line is missing
    model <- lmer(x~ (1|animal), na.action="na.omit")
    s_model <- summary(model)
    mean_est <- s_model$coeff[,1]
    se_est <- as.vector(sqrt(s_model$vcov))
    #str(s_model)
  }
  if ((sum(is.na(x[cy==3]))==8|sum(is.na(x[cy==5]))==8) & # if cy is missing
        ((sum(is.na(x[line==1]))!=8)&(sum(is.na(x[line==2]))!=8))){ # if line is not missing
    model <- lmer(x~ line + (1|animal), na.action="na.omit")
    s_model <- summary(model)
    mean_est <- s_model$coeff[1,1] + s_model$coeff[2,1]/2
    se_est <- as.vector(sqrt(t(c(1,1/2)) %*%s_model$vcov%*%c(1,1/2)))
  }

  if ((sum(is.na(x[cy==3]))!=8&sum(is.na(x[cy==5]))!=8) &  # if cy is not missing
        ((sum(is.na(x[line==1]))==8)|(sum(is.na(x[line==2]))==8))){ # if line is missing
    model <- lmer(x~ cy + (1|animal), na.action="na.omit")
    s_model <- summary(model)
    mean_est <- s_model$coeff[1,1] + s_model$coeff[2,1]/2
    se_est <- as.vector(sqrt(t(c(1,1/2)) %*%s_model$vcov%*%c(1,1/2)))
  }

  if ((sum(is.na(x[cy==3]))!=8& sum(is.na(x[cy==5]))!=8) &
        ((sum(is.na(x[line==1]))!=8)&(sum(is.na(x[line==2]))!=8))){
    model <- lmer(x~ cy + line + (1|animal), na.action="na.omit")
    s_model <- summary(model)
    #str(s_model)
   mean_est <- s_model$coeff[1,1] +
     s_model$coeff[2,1]/2 +
     s_model$coeff[3,1]/2

    se_est <- as.vector(sqrt(t(c(1, 1/2, 1/2))%*% s_model$vcov %*%c(1,1/2, 1/2)))  }


  return(c(se_est, mean_est))
}
se_depleted <- laply(1:dim(dat_final)[1], function(i)out_model(deplete[i,], depleted = "TRUE")[1])

se_whole <- laply(1:dim(dat_final)[1], function(i)out_model(whole[i,], depleted = "FALSE")[1])

lsmean_depleted <- laply(1:dim(dat_final)[1], function(i)out_model(deplete[i,], depleted = "TRUE")[2])

lsmean_whole <- laply(1:dim(dat_final)[1], function(i)out_model(whole[i,], depleted = "FALSE")[2])
```

3 Results of Comparison of Standard Errors between Two Sample Types

3.1 Proportion of protein spots whose standard error in the depleted samples larger than that one in the whole samples

Proportion of protein spots whose standard error in the depleted samples larger than that one in the whole samples.

```r
mean(se_depleted > se_whole)

## [1] 0.5941
```

Figure 1 show the log of standard error of all protein spots in 2 sample types.

```r
log_se <- data.frame(
  logse = log(c(se_whole, se_depleted)),
  sample = rep(c("whole", "depleted"), each = length(se_whole)))
# write.table(log_se, file = "log_se.txt")

p <- ggplot(log_se, aes(sample, logse))
p + geom_boxplot(aes(fill = sample)) +
  ggtitle("Standard Error for Each Sample Type") +
  scale_fill_discrete(name= "Sample Type") +
  xlab("Sample Type") +
  ylab("log(Standard Error)") +
  theme(text = element_text(size=11))
```

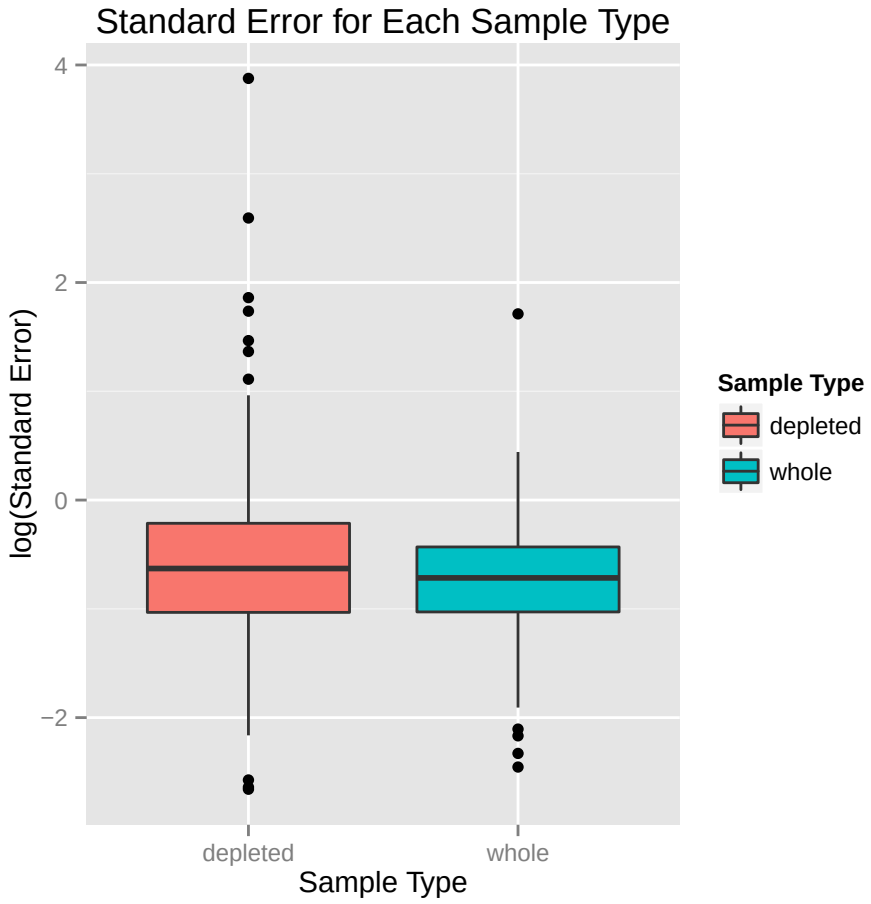


Figure 1: Log(Standard Error) for Each Sample Type

Figure 2 is the scatter plot of logse of Depleted Sample vs. Whole Sample.

```r
#reg <- lm(log(se_depleted)~log(se_whole))
par(cex=.8)
plot(log(se_whole), log(se_depleted), main ="Scatter plot of Log(Standard Error)" ,
     xlim = c(-3, 4), ylim = c(-3, 4))
abline(a =0, b = 1)
```

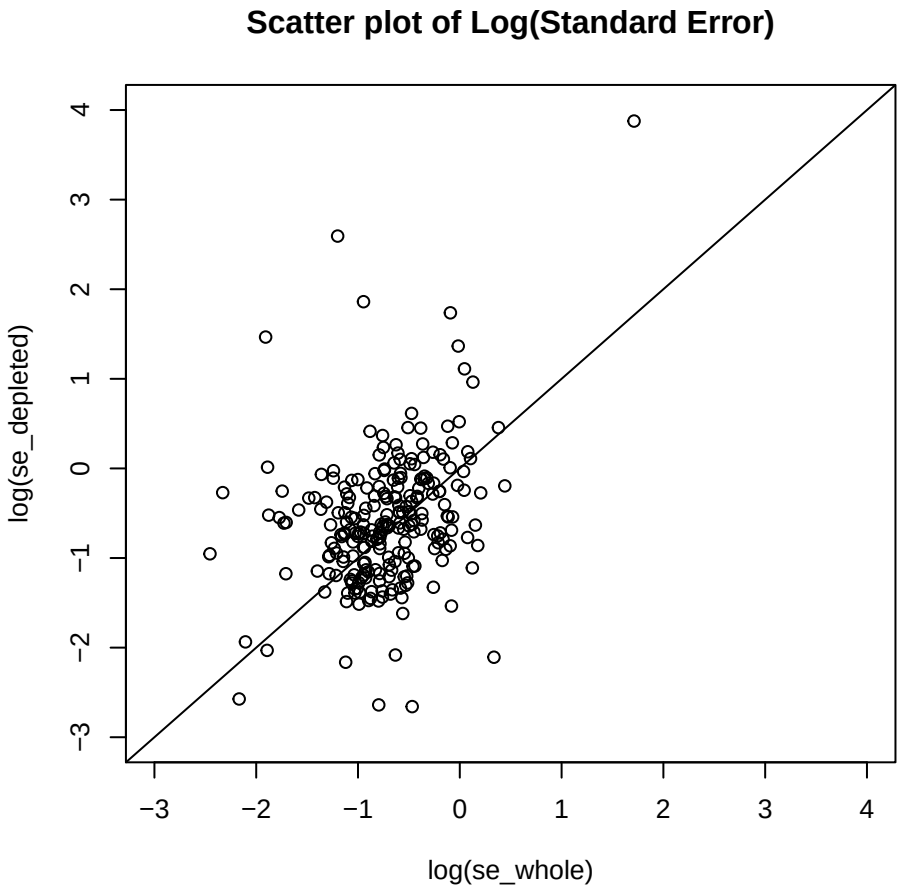


Figure 2: Scatter plot of Log(Standard Error)

Figure 3 is the scatter plot of difference of logse vs. the average of logse.

```r
diff_logse <- log(se_depleted) - log(se_whole)
aver_logse <- (log(se_depleted) + log(se_whole))/2

par(cex=.8)
plot(aver_logse, diff_logse, main ="Scatter plot of diff_logse vs. aver_logse" ,
     xlim = c(-3, 4), ylim = c(-3, 4))
abline(h = 0)
```

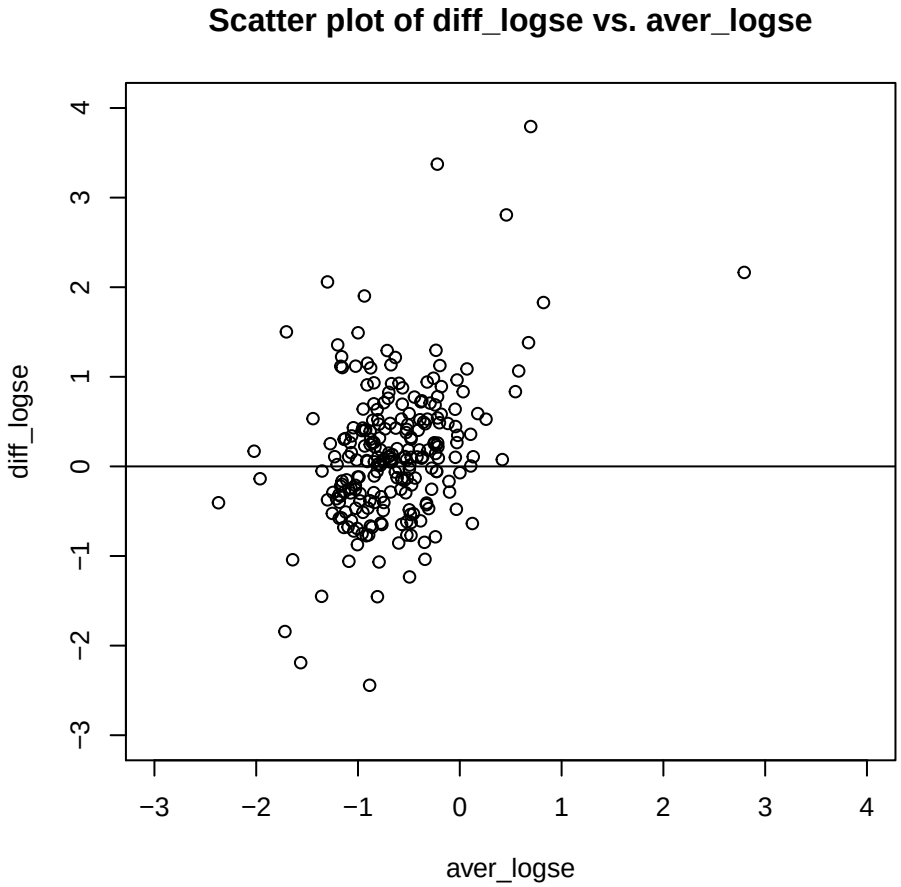


Figure 3: Scatter plot of Log(Standard Error) II

3.2 One-sided Wilcoxon test for log Standard Error

Consider a Wilcoxon signed-rank test H0: The distribution of the standard errors across protein is the same for depleted samples and whole samples, H1: the standard errors tend to be larger for depleted samples than for whole samples. The test using log-transformed standard errors has p-value = 0.004275.

```
# One-sided Test for the log-transformed standard errors
wilcox.test(log(se_depleted/se_whole), alternative = "greater")

##
##  Wilcoxon signed rank test with continuity correction
##
## data:  log(se_depleted/se_whole)
## V = 17154, p-value = 0.004275
## alternative hypothesis: true location is greater than 0
```

4 Results of Comparison of lsmean between Two Sample Types

4.1 Proportion of protein spots whose lsmean in the depleted samples larger than that one in the whole samples

Proportion of protein spots whose lsmean in the depleted samples larger than that one in the whole samples.

```
mean(lsmean_depleted > lsmean_whole)

## [1] 0.569
```

Figure 4 shows the lsmean of all protein spots in 2 sample types.

```
ls_mean <- data.frame(
  lsmean = (c(lsmean_whole, lsmean_depleted)),
  sample = rep(c("whole", "depleted"), each = length(lsmean_whole)))
# write.table(log_se, file = "log_se.txt")

p <- ggplot(ls_mean, aes(sample, lsmean))
p + geom_boxplot(aes(fill = sample)) +
  ggtitle("All data") +
  scale_fill_discrete(name= "Sample Type") +
  xlab("Sample Type") +
  ylab("lsmean") +
  theme(text = element_text(size=11))
```

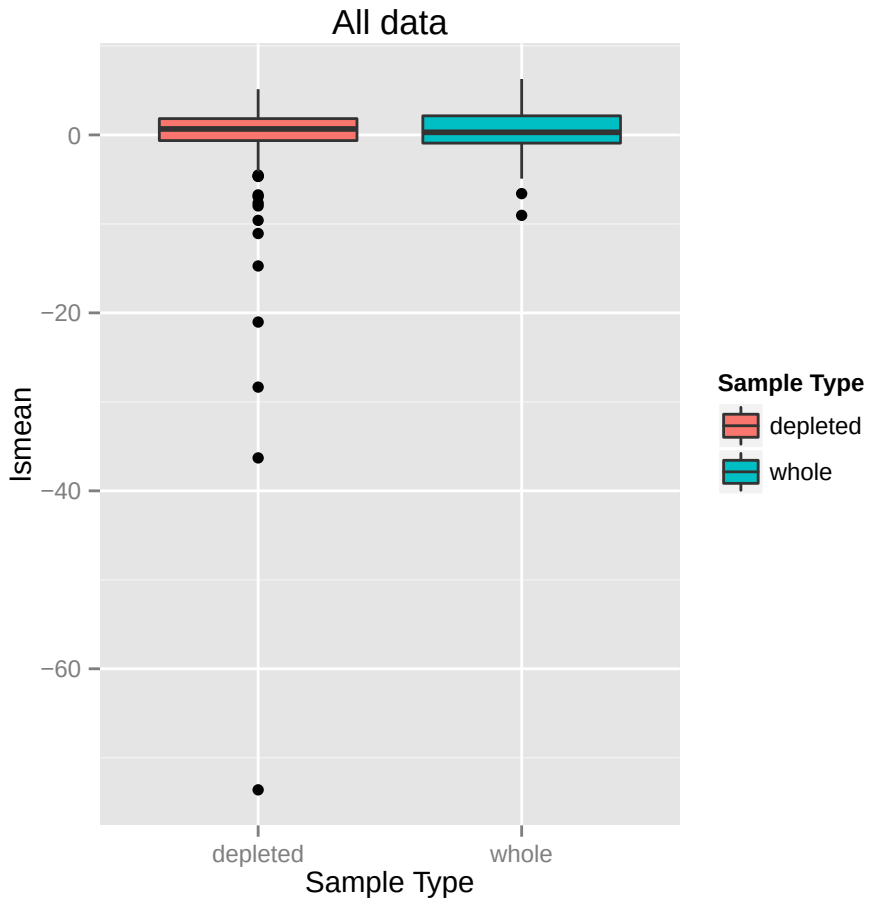


Figure 4: lsmean for Each Sample Type

Figure 5 shows the lsmean of all protein spots in 2 sample types, excluding the spots whose lsmean of depleted sample less than or equal -10.

```
ls_mean <- data.frame(
  lsmean = (c(lsmean_whole[lsmean_depleted >-10], lsmean_depleted[lsmean_depleted >-10])),
  sample = rep(c("whole", "depleted"), each = length(lsmean_whole[lsmean_depleted >-10])))
# write.table(log_se, file = "log_se.txt")

p <- ggplot(ls_mean, aes(sample, lsmean))
p + geom_boxplot(aes(fill = sample)) +
  ggtitle("Data excluding Values less than -10") +
  scale_fill_discrete(name= "Sample Type") +
  xlab("Sample Type") +
  ylab("lsmean") +
  theme(text = element_text(size=11))
```

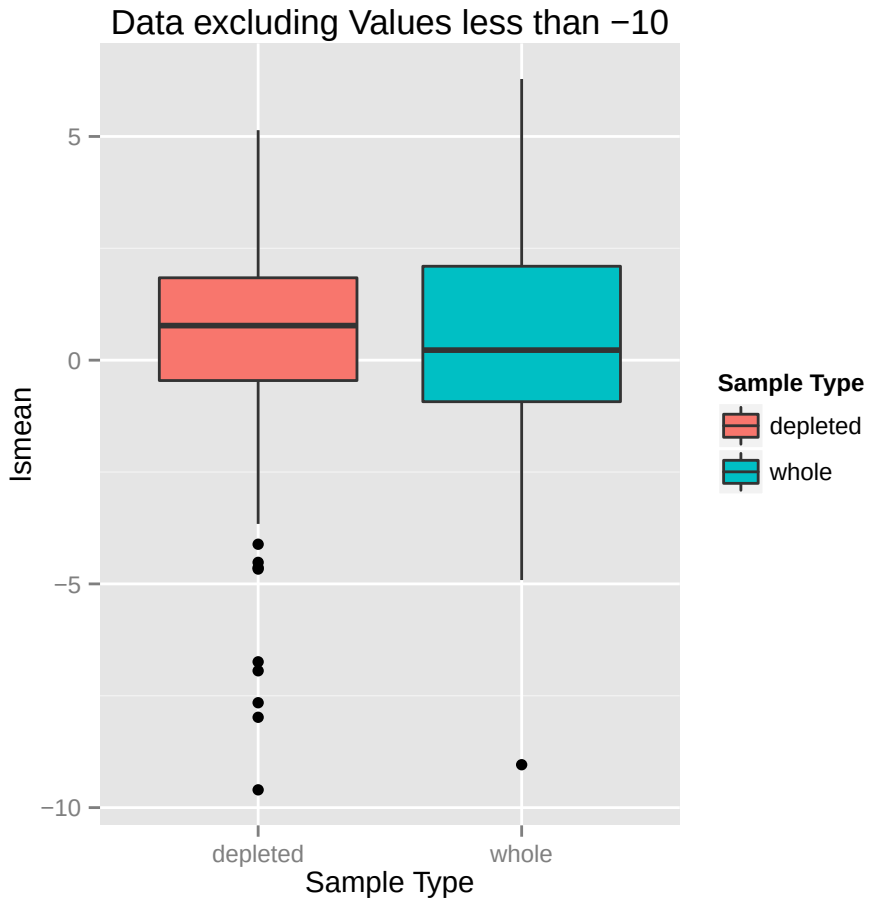


Figure 5: lsmean for Each Sample Type, deleted too small values

4.2 One-sided Wilcoxon test for the lsmean

Consider a Wilcoxon signed-rank test H0: The distribution of the standardized abundance level across protein is the same for depleted samples and whole samples, H1: the standardized abundance level tend to be larger for depleted samples than for whole samples. P-value is 0.169 for the subset of data excluding those with estimated abundance less than -10 as in Figure 5, and is 0.3977 for all data.

```
# One-sided Test for the log-transformed standard errors
wilcox.test(lsmean_depleted[lsmean_depleted >-10],
            lsmean_whole[lsmean_depleted >-10],
            alternative ="greater", paired = TRUE)

##
##  Wilcoxon signed rank test with continuity correction
##
## data:  lsmean_depleted[lsmean_depleted > -10] and lsmean_whole[lsmean_depleted > -10]
## V = 14618, p-value = 0.169
## alternative hypothesis: true location shift is greater than 0

wilcox.test(lsmean_depleted,
            lsmean_whole,
```

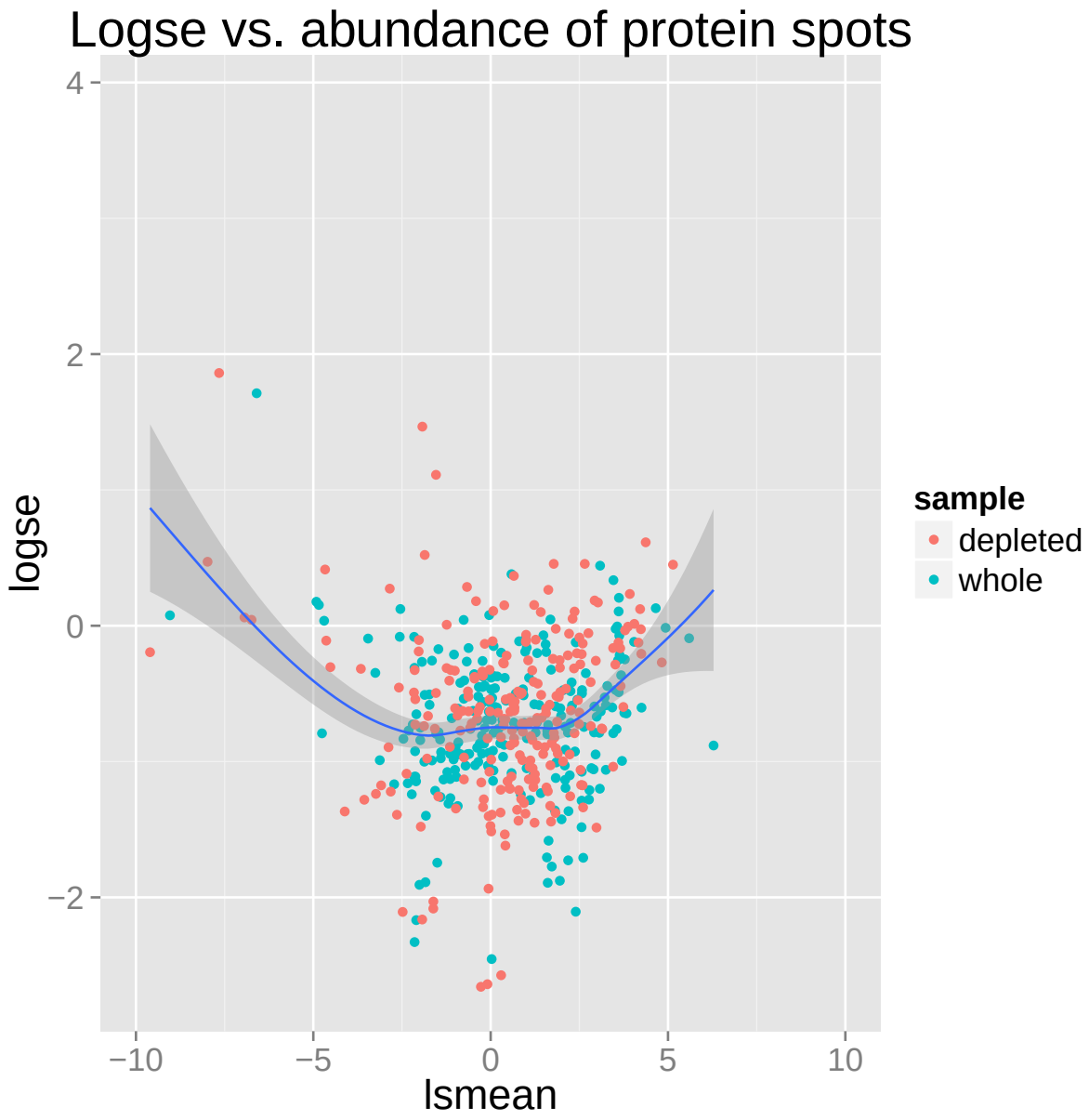
```
            alternative ="greater", paired = TRUE)

##
##  Wilcoxon signed rank test with continuity correction
##
## data:  lsmean_depleted and lsmean_whole
## V = 14618, p-value = 0.3977
## alternative hypothesis: true location shift is greater than 0
```

5 Plot logse of LSmean vs LSmean for the data excluding LSmean less than -10

```
res_out <- data.frame(
  logse = log(c(se_whole, se_depleted)),
  lsmean = c(lsmean_whole,lsmean_depleted),
  sample = rep(c("whole", "depleted"), each = length(se_whole)))


p1 <- ggplot(res_out, aes(lsmean, logse), colour = sample) +
  geom_point(aes(colour = sample)) +
  xlim(-10, 10)+
  theme(text = element_text(size=18))+
  ggtitle("Logse vs. abundance of protein spots") +
  geom_smooth()
p1
```


Logse vs. abundance of protein spots
logse
lsmean
4
2
0
−2
−10
−5
0
5
10
sample
depleted
whole