Statistical data analysis

March 2, 2017

```
opts_chunk$set(echo=TRUE, fig.path='figures/', cache=FALSE, dev='pdf')##postscript')
```

1 Load all libraries, functions, and data

Load everything, check structure of data objects:

```
library(ggplot2)

## Warning:  package 'ggplot2' was built under R version 3.3.2

library(reshape)

## Warning:  package 'reshape' was built under R version 3.3.2

library(genefilter)
library(RColorBrewer)
library(grid)
library(corrplot)

## Warning:  package 'corrplot' was built under R version 3.3.2

library(ROCR)

## Loading required package:  gplots
##
## Attaching package:  'gplots'
## The following object is masked from 'package:stats':
##
##    lowess

source("functions.R")

load("MasterFrame.RData")
```

2 Compare DMD cases and healthy controls considering the effects of age and study site

2.1 Fit regression models

Fit a linear model with transformed metabolite intensities as outcomes, regressing on DMD status, age, their interaction, and study site. Compare it to a model which has just age and study center, testing whether the coefficients of DMD and of DMD x age are 0. Thus, the test is to see whether there is any effect of DMD on the intensities in the presence of age.

```
##get p-values for testing whether there is any effect of DMD
##(so compare this model to just a model for age and study center)
pValsAnyDMD <- vapply(MasterFrame[,mets],
                      function(metInt, status, age, site){
                        ##interaction model
                        lm.metInt <- lm(metInt ~ status*age + as.factor(site));
                        ##age-only model
                        lm.metAge <- lm(metInt ~ age + as.factor(site));
                        anova(lm.metInt, lm.metAge)["Pr(>F)"][2,1]
                      },
                      FUN.VALUE = 0.1,
                      MasterFrame$Status, MasterFrame$Age, MasterFrame$Site)
```

2.2 Find top peaks

Now look at how many of the peaks have q-values less than or equal to 0.05 and 0.01 (use just the 0.01 moving forward), so that the false discovery rate is controlled at 0.05 and 0.01, as well as how many peaks are significant at a Bonferroni-corrected threshold of 0.05:

```
qValsAnyDMD <- p.adjust(pValsAnyDMD, method="BH")
length(qValsAnyDMD)
```

```
## [1] 2203
```

```
sum(qValsAnyDMD <= 0.05)
```

```
## [1] 37
```

```
sum(qValsAnyDMD <= 0.01)
```

```
## [1] 14
```

```
topFDRpeaks <- names(which(qValsAnyDMD <= 0.01))
##sort the top peaks by the p-values
topFDRpeaks <- topFDRpeaks[order(pValsAnyDMD[topFDRpeaks])]
topFDRpeaks
```

```
##  [1] "M357T30p"  "M369T366n" "M114T37p"  "M367T347n" "M312T36p"
##  [6] "M132T37p"  "M451T372p" "M270T359p" "M397T369n" "M174T37p"
## [11] "M449T366n" "M357T41n"  "M209T321p" "M432T331p"

topFWERpeaks <- names(which(pValsAnyDMD <= 0.05/length(pValsAnyDMD)))
##sort them by p-value, from lowest to highest
topFWERpeaks <- topFWERpeaks[order(pValsAnyDMD[topFWERpeaks])]
topFWERpeaks

## [1] "M357T30p"  "M369T366n" "M114T37p"  "M367T347n" "M312T36p"  "M132T37p"
## [7] "M451T372p" "M270T359p"
```

Add in the most likely annotations for these peaks:

```
##get their most likely annotations
AnnTopFWERpeaks <- topFWERpeaks
names(AnnTopFWERpeaks) <- topFWERpeaks
AnnTopFWERpeaks["M357T30p"] <- "m/z=357.25"
AnnTopFWERpeaks["M114T37p"] <- "Creatinine"
AnnTopFWERpeaks["M369T366n"] <- "5a-DHT"
AnnTopFWERpeaks["M367T347n"] <- "Testost. sulf."
AnnTopFWERpeaks["M312T36p"] <- "m/z=312.01"
AnnTopFWERpeaks["M132T37p"] <- "Creatine"
AnnTopFWERpeaks["M451T372p"] <- "m/z=451.17"
AnnTopFWERpeaks["M270T359p"] <- "m/z=270.32"
AnnTopFWERpeaks["M397T369n"] <- "m/z=397.21"
AnnTopFWERpeaks["M174T37p"] <- "Arginine"
AnnTopFWERpeaks["M449T366n"] <- "m/z=449.25"
AnnTopFWERpeaks["M357T41n"] <- "m/z=357.03"
AnnTopFWERpeaks["M209T321p"] <- "m/z=209.12"
AnnTopFWERpeaks["M432T331p"] <- "m/z=432.24"

##add in annotations for the FDR-significant peaks that have annotations
AnnTopFDRpeaks <- topFDRpeaks
names(AnnTopFDRpeaks) <- topFDRpeaks
AnnTopFDRpeaks[names(AnnTopFWERpeaks)] <- AnnTopFWERpeaks
```

2.3 Plots for top peaks

Make some nice plots for them:

```
##save all the top plots in list
ggTop <- list()
##also save all boxplots in list
ggTopBox <- list()
```

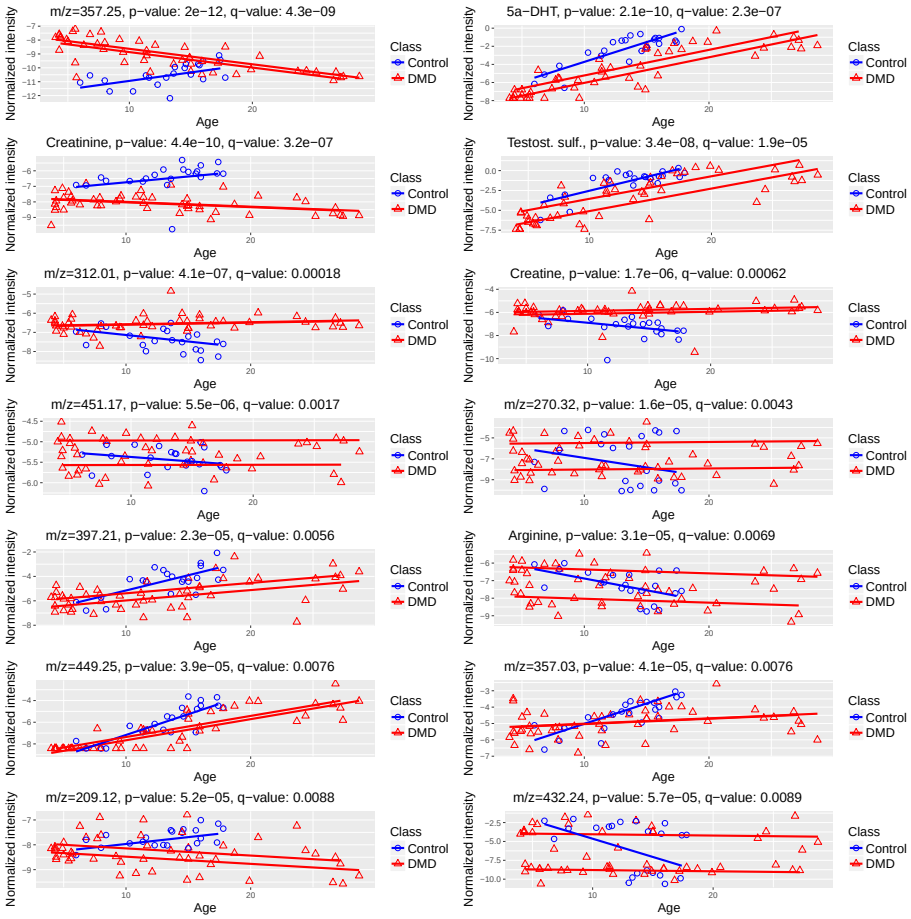
```r
##get all sites
sites <- levels(MasterFrame$Site)
nrSites <- length(sites)
for(met in topFDRpeaks)
{
  lmMet <- lm(MasterFrame[,met] ~
                MasterFrame[,"Status"]*MasterFrame[,"Age"]+as.factor(MasterFrame[,"Site"]),
              data=MasterFrame)
  lmMet

  pred <- predict(lmMet)

  ##get the linear model predictions for Calgary controls and DMD
  whichCalgContr <- which(MasterFrame$Site == "Calgary" &
                            MasterFrame$Status == "Control")
  whichCalgDMD <- which(MasterFrame$Site == "Calgary" &
                          MasterFrame$Status == "DMD")
  predCalgContr <- cbind(MasterFrame$Age[whichCalgContr],
                         pred[whichCalgContr])
  predCalgDMD <- cbind(MasterFrame$Age[whichCalgDMD],
                       pred[whichCalgDMD])

  ##get the linear model predictions for the Davis group (only DMD)
  whichDavis <- which(MasterFrame$Site == "UC Davis")
  predDavis <- cbind(MasterFrame$Age[whichDavis],
                     pred[whichDavis])

  ##get the values for min and max age for these groups (to plot the segments)
  CalgDMDseg <- matrix(c(min(predCalgDMD[,1]),
                         predCalgDMD[which.min(predCalgDMD[,1]),2],
                         max(predCalgDMD[,1]),
                         predCalgDMD[which.max(predCalgDMD[,1]),2]),
                       nrow=2, byrow=TRUE)
  CalgContrSeg <- matrix(c(min(predCalgContr[,1]),
                           predCalgContr[which.min(predCalgContr[,1]),2],
                           max(predCalgContr[,1]),
                           predCalgContr[which.max(predCalgContr[,1]),2]),
                         nrow=2, byrow=TRUE)
  DavisDMDseg <- matrix(c(min(predDavis[,1]),
                          predDavis[which.min(predDavis[,1]),2],
                          max(predDavis[,1]),
                          predDavis[which.max(predDavis[,1]),2]),
                        nrow=2, byrow=TRUE)
  segs <- data.frame(Site = c("Calgary", "Calgary", "UC Davis"),
                     Status = c("DMD", "Control", "DMD"),
```

```
                    x = c(CalgDMDseg[1,1], CalgContrSeg[1,1], DavisDMDseg[1,1]),
                    y = c(CalgDMDseg[1,2], CalgContrSeg[1,2], DavisDMDseg[1,2]),
                    xend = c(CalgDMDseg[2,1], CalgContrSeg[2,1], DavisDMDseg[2,1]),
                    yend = c(CalgDMDseg[2,2], CalgContrSeg[2,2], DavisDMDseg[2,2]))

  ggTop[[met]] <- ggplot(MasterFrame, aes_string(x="Age", y=met, shape="Status", color="Stat
    geom_point(size=2.5) +
    geom_segment(data=segs, aes(x=x, y=y,
                                xend=xend, yend=yend),
                 size=1.1) +
    scale_color_manual(name = "Class",
                       breaks = c("Control", "DMD"),
                       labels = c("Control", "DMD"),
                       values = c(4,2)) +
    scale_shape_manual(name = "Class",
                       breaks = c("Control", "DMD"),
                       labels = c("Control", "DMD"),
                       values = c(1,2)) +
    scale_y_continuous(name="Normalized intensity") +
    labs(title=(paste(AnnTopFDRpeaks[met], ", ",
                      "p-value: ", signif(pValsAnyDMD[met],2), ", ",
                      "q-value: ", signif(qValsAnyDMD[met],2),
                      sep=""))) +
    theme(plot.title = element_text(size = 15, hjust = 0.2, vjust=1.5),
          legend.title = element_text(size = 14),
          legend.text = element_text(size=14),
          axis.title = element_text(size=14))

  ggTopBox[[met]] <- ggplot(MasterFrame, aes_string(x="Status", y=met))+
    geom_boxplot() +
    geom_point(size=3.0, aes(color=Category)) +
    scale_y_continuous(name="Normalized intensity") +
    labs(title=(paste(AnnTopFDRpeaks[met]))) +
    scale_color_discrete(name = "Age category") +
    theme(plot.title = element_text(size = 15, hjust = 0.2, vjust=1.5),
          legend.title = element_text(size = 14),
          legend.text = element_text(size=14),
          axis.title = element_text(size=14))
}
```

2.3.1 Plots versus age for top peaks

Plots of the intensities versus age, including some of the fitted regression lines, coded by case/control status:

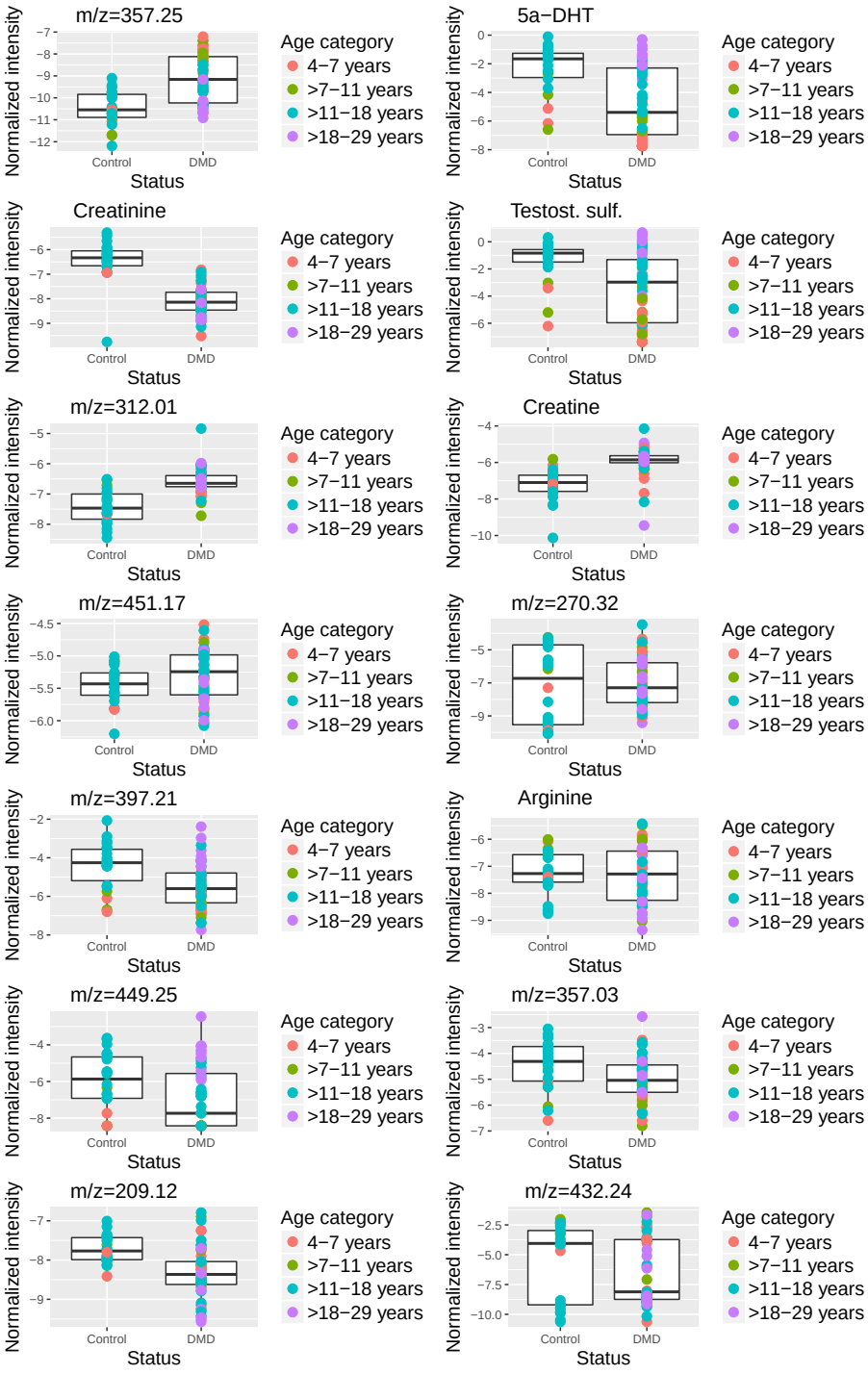
```
multiplot(plotlist=ggTop, layout=matrix(1:length(ggTop), ncol=2, byrow=TRUE))
```



2.3.2 Boxplots for case/control status for top peaks

Boxplots of the intensities versus case control status, color-coded by age category:

```
multiplot(plotlist=ggTopBox, layout=matrix(1:length(ggTopBox), ncol=2, byrow=TRUE))
```


m/z=357.25
5a−DHT
Creatinine
Testost. sulf.
m/z=312.01
Creatine
m/z=451.17
m/z=270.32
m/z=397.21
Arginine
m/z=449.25
m/z=357.03
m/z=209.12
m/z=432.24
Normalized intensity
Status
Control
DMD
Age category
4−7 years
>7−11 years
>11−18 years
>18−29 years

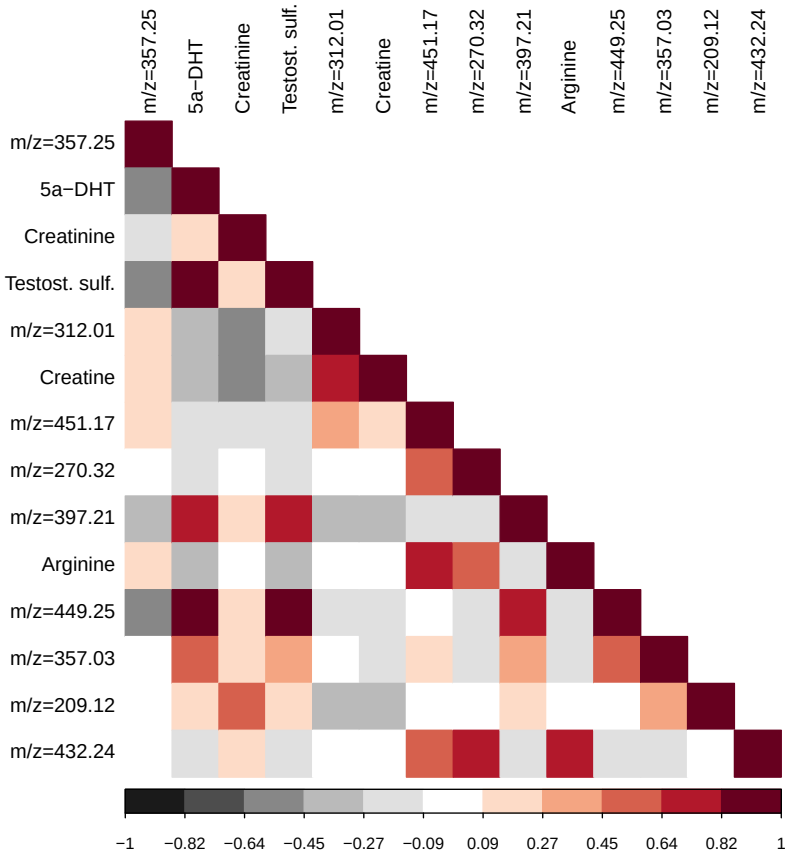
2.3.3 Correlation plot for top peaks

Also calculate and plot the correlations between the top peaks:

```
##make it in matrix version as well:
cor.matrix <- matrix(NA, length(topFDRpeaks), length(topFDRpeaks))

for(i1 in 1:length(topFDRpeaks))
{
  for(i2 in 1:length(topFDRpeaks))
  {
    Met1 <- topFDRpeaks[i1]
    Met2 <- topFDRpeaks[i2]

    cor.matrix[i1, i2] <-
      cor(MasterFrame[, Met1], MasterFrame[, Met2])
  }
}
rownames(cor.matrix) <- colnames(cor.matrix) <- AnnTopFDRpeaks[topFDRpeaks]

round(cor.matrix, 2)
```

```
##                m/z=357.25 5a-DHT Creatinine Testost. sulf. m/z=312.01
## m/z=357.25           1.00  -0.60      -0.24          -0.57       0.18
## 5a-DHT              -0.60   1.00       0.24           0.92      -0.32
## Creatinine          -0.24   0.24       1.00           0.17      -0.52
## Testost. sulf.      -0.57   0.92       0.17           1.00      -0.27
## m/z=312.01           0.18  -0.32      -0.52          -0.27       1.00
## Creatine             0.24  -0.33      -0.48          -0.30       0.74
## m/z=451.17           0.17  -0.14      -0.12          -0.21       0.28
## m/z=270.32           0.06  -0.18       0.05          -0.19      -0.01
## m/z=397.21          -0.37   0.72       0.20           0.76      -0.37
## Arginine             0.10  -0.30       0.07          -0.33       0.02
## m/z=449.25          -0.48   0.88       0.13           0.83      -0.23
## m/z=357.03          -0.06   0.51       0.25           0.41      -0.01
## m/z=209.12           0.03   0.19       0.49           0.13      -0.36
## m/z=432.24           0.01  -0.19       0.18          -0.25      -0.06
##                Creatine m/z=451.17 m/z=270.32 m/z=397.21 Arginine
## m/z=357.25         0.24       0.17       0.06      -0.37     0.10
## 5a-DHT            -0.33      -0.14      -0.18       0.72    -0.30
## Creatinine        -0.48      -0.12       0.05       0.20     0.07
## Testost. sulf.    -0.30      -0.21      -0.19       0.76    -0.33
## m/z=312.01         0.74       0.28      -0.01      -0.37     0.02
## Creatine           1.00       0.19      -0.06      -0.42     0.05
## m/z=451.17         0.19       1.00       0.48      -0.13     0.74
## m/z=270.32        -0.06       0.48       1.00      -0.18     0.57
```

```
## m/z=397.21        -0.42      -0.13      -0.18       1.00    -0.23
## Arginine           0.05       0.74       0.57      -0.23     1.00
## m/z=449.25        -0.23      -0.08      -0.12       0.74    -0.23
## m/z=357.03        -0.12       0.13      -0.16       0.34    -0.11
## m/z=209.12        -0.40       0.00      -0.08       0.10    -0.01
## m/z=432.24         0.00       0.57       0.69      -0.24     0.77
##                m/z=449.25 m/z=357.03 m/z=209.12 m/z=432.24
## m/z=357.25          -0.48      -0.06       0.03       0.01
## 5a-DHT               0.88       0.51       0.19      -0.19
## Creatinine           0.13       0.25       0.49       0.18
## Testost. sulf.       0.83       0.41       0.13      -0.25
## m/z=312.01          -0.23      -0.01      -0.36      -0.06
## Creatine            -0.23      -0.12      -0.40       0.00
## m/z=451.17          -0.08       0.13       0.00       0.57
## m/z=270.32          -0.12      -0.16      -0.08       0.69
## m/z=397.21           0.74       0.34       0.10      -0.24
## Arginine            -0.23      -0.11      -0.01       0.77
## m/z=449.25           1.00       0.52       0.07      -0.17
## m/z=357.03           0.52       1.00       0.32      -0.11
## m/z=209.12           0.07       0.32       1.00       0.06
## m/z=432.24          -0.17      -0.11       0.06       1.00

par(mar=c(17,16,0.5,0.5)+0.1, oma=c(1,1,1,1),
    cex = 0.95)
corrplot(round(cor.matrix,2), col=rev(brewer.pal(11, "RdGy")),##[c(1,3,5,6,7,9,11)]),
         method="color",
         tl.pos="lt",
         type="lower", tl.col="black")
```



3 Compare DMD cases and healthy controls in the subset of study participants aged 4-18

3.1 Fit regression models

Given that the age distributions are different for DMD cases versus healthy controls (more old and more young study participants among the cases), we also consider an analysis which has the age range of 4-18 years.

```
MasterFrame4.18 <- MasterFrame[MasterFrame$Age <= 18,]
dim(MasterFrame4.18)
```

```
## [1]   62 2208
```

```
##check that the minimum age is actually 4
min(MasterFrame$Age)
```

```
## [1] 4

##see how many participants are left now
table(MasterFrame4.18$Status)

##
## Control     DMD
##      22      40

table(MasterFrame4.18$Site)

##
##       Australia         Calgary            CNMC U of Pittsburgh
##               1              32               4               4
##        UC Davis
##              21

table(MasterFrame4.18$Status, MasterFrame4.18$Site)

##
##           Australia Calgary CNMC U of Pittsburgh UC Davis
##   Control         0      16    4               2        0
##   DMD             1      16    0               2       21

##get p-values for testing whether there is any effect of DMD
##(so compare this model to just a model for age)
pValsAnyDMD4.18 <- vapply(MasterFrame4.18[,mets],
                          function(metInt, status, age, site){
                            ##interaction model
                            lm.metInt <- lm(metInt ~ status*age + as.factor(site));
                            ##age-only model
                            lm.metAge <- lm(metInt ~ age + as.factor(site));
                            anova(lm.metInt, lm.metAge)["Pr(>F)"][2,1]
                          },
                          FUN.VALUE = 0.1,
                          MasterFrame4.18$Status, MasterFrame4.18$Age, MasterFrame4.18$Site)
```

3.2 Find top peaks

Now look at how many of the peaks have q-values less than or equal to 0.05 and 0.01 (use just the 0.01 moving forward), so that the false discovery rate is controlled at 0.05 and 0.01, as well as how many peaks are significant at a Bonferroni-corrected threshold of 0.05:

```r
qValsAnyDMD4.18 <- p.adjust(pValsAnyDMD4.18, method="BH")
sum(qValsAnyDMD4.18 <= 0.05)
```

```
## [1] 10
```

```r
sum(qValsAnyDMD4.18 <= 0.01)
```

```
## [1] 6
```

```r
sum(pValsAnyDMD4.18 <= 0.05/length(pValsAnyDMD4.18))
```

```
## [1] 6
```

```r
which(pValsAnyDMD4.18 <= 0.05/length(pValsAnyDMD4.18))
```

```
##  M132T37p  M312T36p  M357T30p  M114T37p M369T366n M449T372n
##         1         2         3         4      1896      2006
```

```r
topFDRpeaks4.18 <- names(which(qValsAnyDMD4.18 <= 0.01))
topFDRpeaks4.18 <- topFDRpeaks4.18[order(pValsAnyDMD4.18[topFDRpeaks4.18])]

topFWERpeaks4.18 <- names(which(pValsAnyDMD4.18 <= 0.05/length(pValsAnyDMD4.18)))
topFWERpeaks4.18 <- topFWERpeaks4.18[order(pValsAnyDMD4.18[topFWERpeaks4.18])]

sort(pValsAnyDMD4.18[topFWERpeaks4.18])
```

```
##     M357T30p    M449T372n    M369T366n     M114T37p     M132T37p
## 2.782674e-10 2.678872e-07 2.346169e-06 3.532358e-06 6.691164e-06
##     M312T36p
## 1.594811e-05
```

```r
sort(qValsAnyDMD4.18[topFWERpeaks4.18])
```

```
##     M357T30p    M449T372n    M369T366n     M114T37p     M132T37p
## 6.130232e-07 2.950777e-04 1.722870e-03 1.945446e-03 2.948127e-03
##     M312T36p
## 5.855614e-03
```

3.3 Comparison between full data analysis and data analysis for ages 4-18

Now do some comparisons with the full data analysis:

```r
##what is the intersection with the top FWER peaks when using the whole dataset?
intersect(topFWERpeaks, topFWERpeaks4.18)
```

```
## [1] "M357T30p"  "M369T366n" "M114T37p"  "M312T36p"  "M132T37p"
```

```r
##what is the intersection with the top FDR (at 0.01) peaks from the whole dataset?
intersect(topFWERpeaks4.18, topFDRpeaks)
```

```
## [1] "M357T30p"  "M369T366n" "M114T37p"  "M132T37p"  "M312T36p"
```

```r
##what is the set difference between the top peaks for ages 4-18 and the top FDR peaks from
setdiff(topFWERpeaks4.18, topFDRpeaks)
```

```
## [1] "M449T372n"
```

```r
##what are the q-values of the top FDR peaks in the whole dataset in the dataset for ages 4
qValsAnyDMD4.18[topFDRpeaks]
```

```
##     M357T30p    M369T366n     M114T37p    M367T347n     M312T36p
## 6.130232e-07 1.722870e-03 1.945446e-03 3.098468e-02 5.855614e-03
##     M132T37p    M451T372p    M270T359p    M397T369n     M174T37p
## 2.948127e-03 2.624842e-02 6.433008e-02 1.847101e-02 8.675175e-02
##    M449T366n     M357T41n    M209T321p    M432T331p
## 9.992524e-02 1.578959e-01 5.270305e-02 8.675175e-02
```

```r
max(qValsAnyDMD4.18[topFDRpeaks])
```

```
## [1] 0.1578959
```

```r
##what are the ranks of the top peaks for ages 4-18 in the full data analysis?
sort(rank(pValsAnyDMD)[topFWERpeaks4.18])
```

```
##  M357T30p M369T366n  M114T37p  M312T36p  M132T37p M449T372n
##         1         2         3         5         6        16
```

```r
sort(qValsAnyDMD[topFWERpeaks4.18])
```

```
##     M357T30p    M369T366n     M114T37p     M312T36p     M132T37p
## 4.326175e-09 2.284963e-07 3.233945e-07 1.799352e-04 6.243629e-04
##    M449T372n
## 1.337954e-02
```

```r
sort(rank(pValsAnyDMD)[topFDRpeaks4.18])
```

```
##  M357T30p M369T366n  M114T37p  M312T36p  M132T37p M449T372n
##         1         2         3         5         6        16
```

```r
sort(qValsAnyDMD[topFDRpeaks4.18])
```

```
##     M357T30p    M369T366n     M114T37p     M312T36p     M132T37p
## 4.326175e-09 2.284963e-07 3.233945e-07 1.799352e-04 6.243629e-04
##    M449T372n
## 1.337954e-02
```

4 Compare DMD cases and healthy controls considering the effects of age in Calgary subgroup

4.1 Fit regression models

As a protection against heterogeneity due to site, we repeat this analysis (i.e. linear model considering DMD status, age, and their interaction) in the Calgary subgroup only:

```
MasterFrameCalg <- MasterFrame[as.character(MasterFrame$Site) == "Calgary",]
dim(MasterFrameCalg)

## [1]   35 2208

##get p-values for testing whether there is any effect of DMD
##(so compare this model to just a model for age)
pValsAnyDMDCalg <- vapply(MasterFrameCalg[,mets],
                          function(metInt, status, age){
                            ##interaction model
                            lm.metInt <- lm(metInt ~ status*age);
                            ##age-only model
                            lm.metAge <- lm(metInt ~ age);
                            anova(lm.metInt, lm.metAge)["Pr(>F)"][2,1]
                          },
                          FUN.VALUE = 0.1,
                          MasterFrameCalg$Status, MasterFrameCalg$Age)
```

4.2 Find top peaks

Now look at how many of the peaks have q-values less than or equal to 0.05 and 0.01 (use just the 0.01 moving forward), so that the false discovery rate is controlled at 0.05 and 0.01, as well as how many peaks are significant at a Bonferroni-corrected threshold of 0.05:

```
qValsAnyDMDCalg <- p.adjust(pValsAnyDMDCalg, method="BH")
sum(qValsAnyDMDCalg <= 0.05)

## [1] 29

sum(qValsAnyDMDCalg <= 0.01)

## [1] 8

sum(pValsAnyDMDCalg <= 0.05/length(pValsAnyDMDCalg))

## [1] 8
```

```r
which(pValsAnyDMDCalg <= 0.05/length(pValsAnyDMDCalg))
```

```
##  M357T30p  M114T37p M209T321p  M175T33p M451T372p M223T321n M369T366n
##         3         4        31       116      1326      1892      1896
## M367T347n
##      1903
```

```r
topFDRpeaksCalg <- names(which(qValsAnyDMDCalg <= 0.01))
topFDRpeaksCalg <- topFDRpeaksCalg[order(pValsAnyDMDCalg[topFDRpeaksCalg])]

topFWERpeaksCalg <- names(which(pValsAnyDMDCalg <= 0.05/length(pValsAnyDMDCalg)))
topFWERpeaksCalg <- topFWERpeaksCalg[order(pValsAnyDMDCalg[topFWERpeaksCalg])]
```

4.3 Comparison between full data analysis and Calgary-only data analysis

Now do some comparisons with the full data analysis:

```r
##what is the intersection with the top FWER peaks when using the whole dataset?
intersect(topFWERpeaks, topFWERpeaksCalg)
```

```
## [1] "M357T30p"  "M369T366n" "M114T37p"  "M367T347n" "M451T372p"
```

```r
##what is the intersection with the top FDR (at 0.01) peaks from the whole dataset?
intersect(topFWERpeaksCalg, topFDRpeaks)
```

```
## [1] "M357T30p"  "M369T366n" "M209T321p" "M451T372p" "M114T37p"  "M367T347n"
```

```r
##what is the set difference between the top Calgary peaks and the top FDR peaks from the wh
setdiff(topFWERpeaksCalg, topFDRpeaks)
```

```
## [1] "M223T321n" "M175T33p"
```

```r
##what are the q-values of the top FDR peaks in the whole dataset in the Calgary dataset?
qValsAnyDMDCalg[topFDRpeaks]
```

```
##     M357T30p    M369T366n     M114T37p    M367T347n     M312T36p
## 3.300679e-07 2.732030e-04 4.296618e-03 4.296618e-03 1.699789e-02
##     M132T37p    M451T372p    M270T359p    M397T369n     M174T37p
## 1.636677e-02 4.296618e-03 8.657422e-02 1.796232e-02 1.796232e-02
##    M449T366n     M357T41n    M209T321p    M432T331p
## 1.920430e-02 1.676303e-02 2.677080e-03 3.553113e-02
```

```r
max(qValsAnyDMDCalg[topFDRpeaks])
```

```
## [1] 0.08657422
```

```
##what are the ranks of the top FDR peaks in the whole dataset in the Calgary dataset?
sort(rank(pValsAnyDMDCalg)[topFDRpeaks])
```

```
##  M357T30p M369T366n M209T321p M451T372p  M114T37p M367T347n  M132T37p
##         1         3         4         6         7         8         9
##  M357T41n  M312T36p M397T369n  M174T37p M449T366n M432T331p M270T359p
##        10        12        14        15        17        22        42
```

```
##what are the ranks of the top Calgary peaks in the full data analysis?
sort(rank(pValsAnyDMD)[topFWERpeaksCalg])
```

```
##  M357T30p M369T366n  M114T37p M367T347n M451T372p M209T321p M223T321n
##         1         2         3         4         7        13        15
##  M175T33p
##       612
```

```
sort(qValsAnyDMD[topFWERpeaksCalg])
```

```
##     M357T30p     M369T366n      M114T37p     M367T347n     M451T372p
## 4.326175e-09 2.284963e-07 3.233945e-07 1.859220e-05 1.733772e-03
##     M209T321p     M223T321n      M175T33p
## 8.772293e-03 1.011651e-02 9.962584e-01
```

```
sort(rank(pValsAnyDMD)[topFDRpeaksCalg])
```

```
##  M357T30p M369T366n  M114T37p M367T347n M451T372p M209T321p M223T321n
##         1         2         3         4         7        13        15
##  M175T33p
##       612
```

```
sort(qValsAnyDMD[topFDRpeaksCalg])
```

```
##     M357T30p     M369T366n      M114T37p     M367T347n     M451T372p
## 4.326175e-09 2.284963e-07 3.233945e-07 1.859220e-05 1.733772e-03
##     M209T321p     M223T321n      M175T33p
## 8.772293e-03 1.011651e-02 9.962584e-01
```

5 Fit model to difference between creatine and creatinine

Note that all the metabolites have been log transformed, so the difference is the same as the log of the ratio. First create that variable:

```
diffCreat <- MasterFrame[,"M132T37p"]-MasterFrame[,"M114T37p"]
```

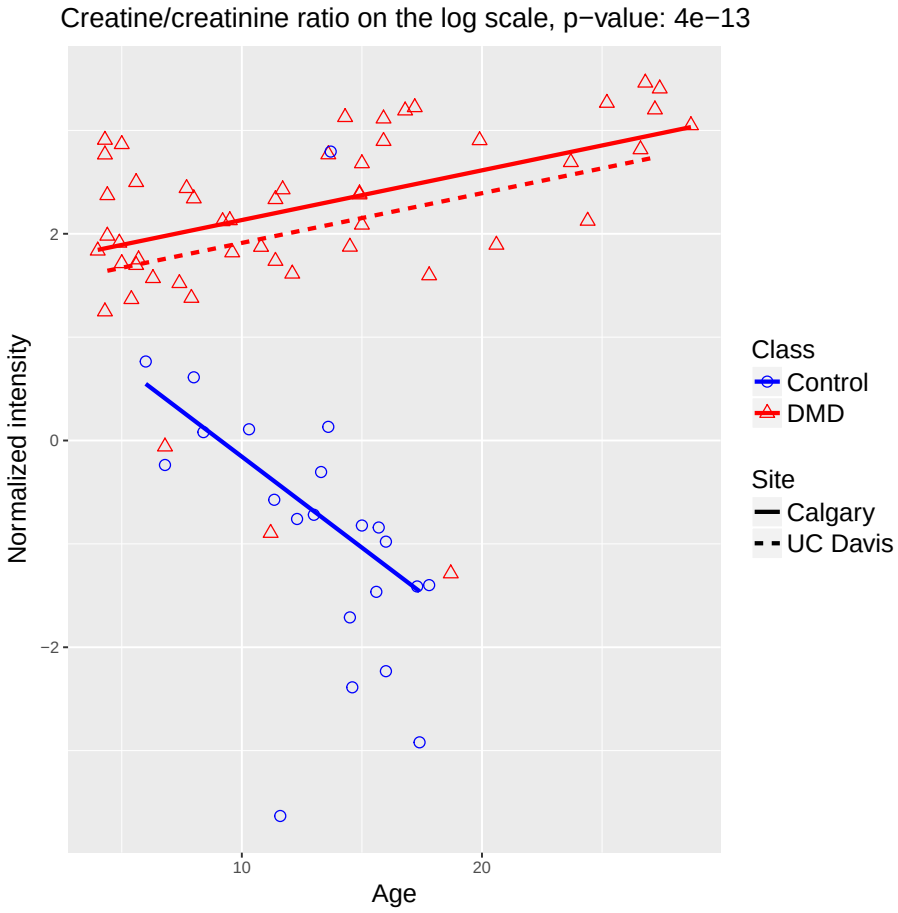
Now run the models and compare:

```
##interaction model
lm.metInt <- lm(diffCreat ~ MasterFrame$Status*MasterFrame$Age + as.factor(MasterFrame$Site)
##age-only model
lm.metAge <- lm(diffCreat ~ MasterFrame$Age + as.factor(MasterFrame$Site));
anova(lm.metInt, lm.metAge)["Pr(>F)"][2,1]

## [1] 4.017637e-13
```

Plot versus age:

```
lmMet <- lm(diffCreat ~
              MasterFrame[,"Status"]*MasterFrame[,"Age"]+as.factor(MasterFrame[,"Site"]),
            data=MasterFrame)

pred <- predict(lmMet)

##get the linear model predictions for Calgary controls and DMD
whichCalgContr <- which(MasterFrame$Site == "Calgary" &
                          MasterFrame$Status == "Control")
whichCalgDMD <- which(MasterFrame$Site == "Calgary" &
                        MasterFrame$Status == "DMD")
predCalgContr <- cbind(MasterFrame$Age[whichCalgContr],
                       pred[whichCalgContr])
predCalgDMD <- cbind(MasterFrame$Age[whichCalgDMD],
                     pred[whichCalgDMD])

##get the linear model predictions for the Davis group (only DMD)
whichDavis <- which(MasterFrame$Site == "UC Davis")
predDavis <- cbind(MasterFrame$Age[whichDavis],
                   pred[whichDavis])

##get the values for min and max age for these groups (to plot the segments)
CalgDMDseg <- matrix(c(min(predCalgDMD[,1]),
                       predCalgDMD[which.min(predCalgDMD[,1]),2],
                       max(predCalgDMD[,1]),
                       predCalgDMD[which.max(predCalgDMD[,1]),2]),
                     nrow=2, byrow=TRUE)
CalgContrSeg <- matrix(c(min(predCalgContr[,1]),
                         predCalgContr[which.min(predCalgContr[,1]),2],
                         max(predCalgContr[,1]),
                         predCalgContr[which.max(predCalgContr[,1]),2]),
                       nrow=2, byrow=TRUE)
DavisDMDseg <- matrix(c(min(predDavis[,1]),
                        predDavis[which.min(predDavis[,1]),2],
```

```
                        max(predDavis[,1]),
                        predDavis[which.max(predDavis[,1]),2]),
                      nrow=2, byrow=TRUE)
segs <- data.frame(Site = c("Calgary", "Calgary", "UC Davis"),
                   Status = c("DMD", "Control", "DMD"),
                   x = c(CalgDMDseg[1,1], CalgContrSeg[1,1], DavisDMDseg[1,1]),
                   y = c(CalgDMDseg[1,2], CalgContrSeg[1,2], DavisDMDseg[1,2]),
                   xend = c(CalgDMDseg[2,1], CalgContrSeg[2,1], DavisDMDseg[2,1]),
                   yend = c(CalgDMDseg[2,2], CalgContrSeg[2,2], DavisDMDseg[2,2]))

##add the difference in the log values to the data frame
MasterFrame$diffCreat <- diffCreat

ggplot(MasterFrame, aes_string(x="Age", y="diffCreat", shape="Status", color="Status")) +
  geom_point(size=2.5) +
  geom_segment(data=segs, aes(x=x, y=y,
                              xend=xend, yend=yend,
                              linetype=Site),
               size=1.1) +
  scale_color_manual(name = "Class",
                     breaks = c("Control", "DMD"),
                     labels = c("Control", "DMD"),
                     values = c(4,2)) +
  scale_shape_manual(name = "Class",
                     breaks = c("Control", "DMD"),
                     labels = c("Control", "DMD"),
                     values = c(1,2)) +
  scale_y_continuous(name="Normalized intensity") +
  labs(title=(paste("Creatine/creatinine ratio on the log scale", ", ",
                    "p-value: ", signif(anova(lm.metInt, lm.metAge)["Pr(>F)"][2,1], 2),
                    sep=""))) +
  theme(plot.title = element_text(size = 15, hjust = 0.2, vjust=1.5),
        legend.title = element_text(size = 14),
        legend.text = element_text(size=14),
        axis.title = element_text(size=14))
```

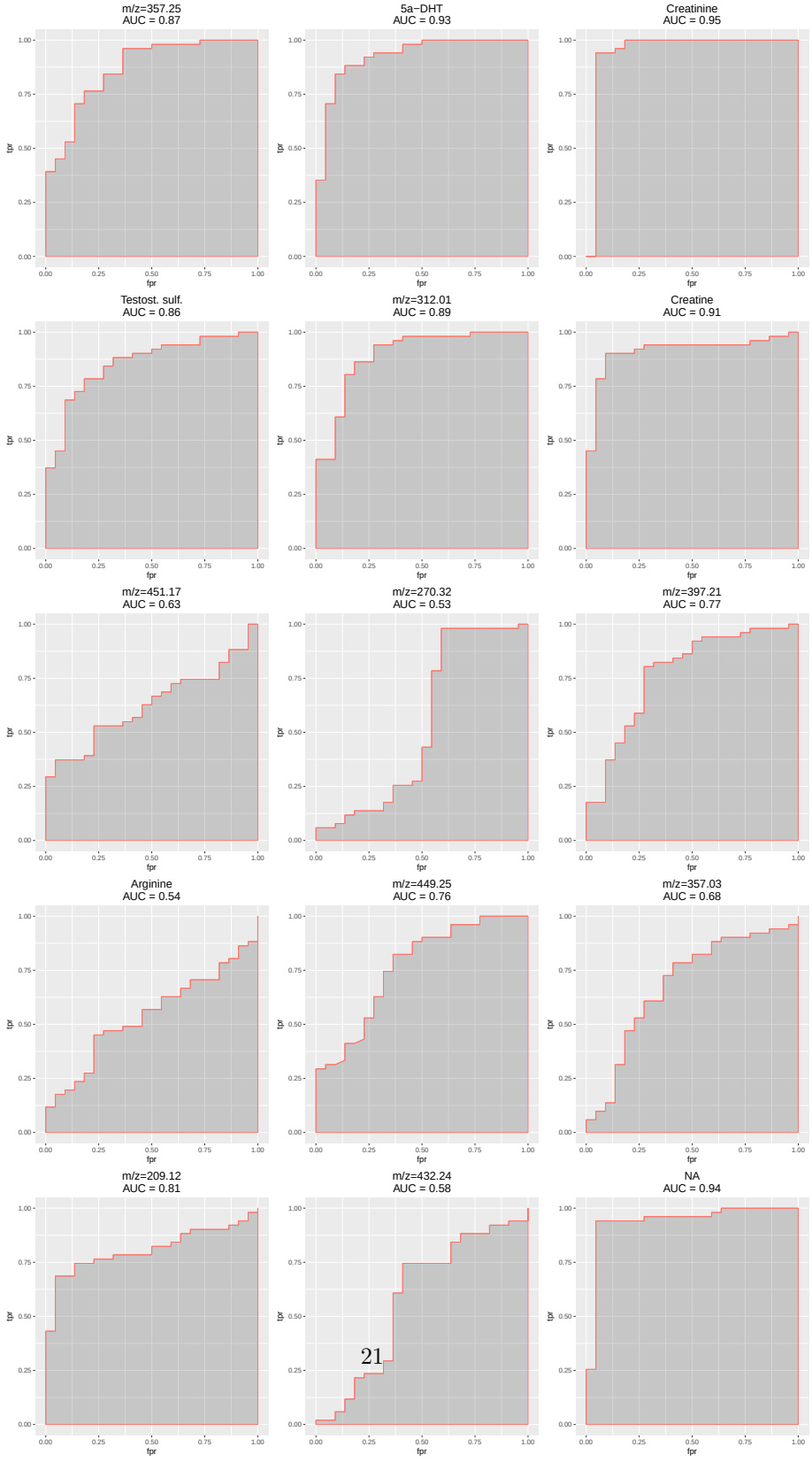


6 Make ROC plots for top metabolites + creatine/creatinine ratio

```
ggROC <- list()
for(met in c(topFDRpeaks, "diffCreat"))
{
  roc.data <- data.frame()

  fit <- glm(as.formula(paste("Status ~", met, "+ Age")),
             data=MasterFrame,
             family=binomial)

  prob <- predict(fit, newdata=MasterFrame, type="response")
  pred <- prediction(prob, MasterFrame$Status)
```

```
  perf <- performance(pred, measure = "tpr", x.measure = "fpr")

  aucTemp <- performance(pred, measure = "auc")
  auc <- aucTemp@y.values[[1]]

  roc.data <- rbind(roc.data,
                    data.frame(fpr=unlist(perf@x.values),
                               tpr=unlist(perf@y.values),
                               col="1"))

  ggROC[[met]] <- ggplot(roc.data, aes(x=fpr, ymin=0, ymax=tpr, col=col)) +
    geom_ribbon(alpha=0.2) +
    geom_line(aes(y=tpr)) +
    theme(legend.position = "none",
          plot.title = element_text(size = 15, hjust = 0.5, vjust=1.5))+
    ggtitle(paste0(AnnTopFDRpeaks[met],"\n",
                   "AUC = ",
                   paste(sort(round(auc,2)))))
}
```

```
multiplot(plotlist=ggROC, layout=matrix(1:length(ggROC), ncol=3, byrow=TRUE))
```


m/z=357.25
AUC = 0.87
5a−DHT
AUC = 0.93
Creatinine
AUC = 0.95
Testost. sulf.
AUC = 0.86
m/z=312.01
AUC = 0.89
Creatine
AUC = 0.91
m/z=451.17
AUC = 0.63
m/z=270.32
AUC = 0.53
m/z=397.21
AUC = 0.77
Arginine
AUC = 0.54
m/z=449.25
AUC = 0.76
m/z=357.03
AUC = 0.68
m/z=209.12
AUC = 0.81
m/z=432.24
AUC = 0.58
NA
AUC = 0.94
tpr
fpr
0.00
0.25
0.50
0.75
1.00

7 Comparison with CKM for DMD cases

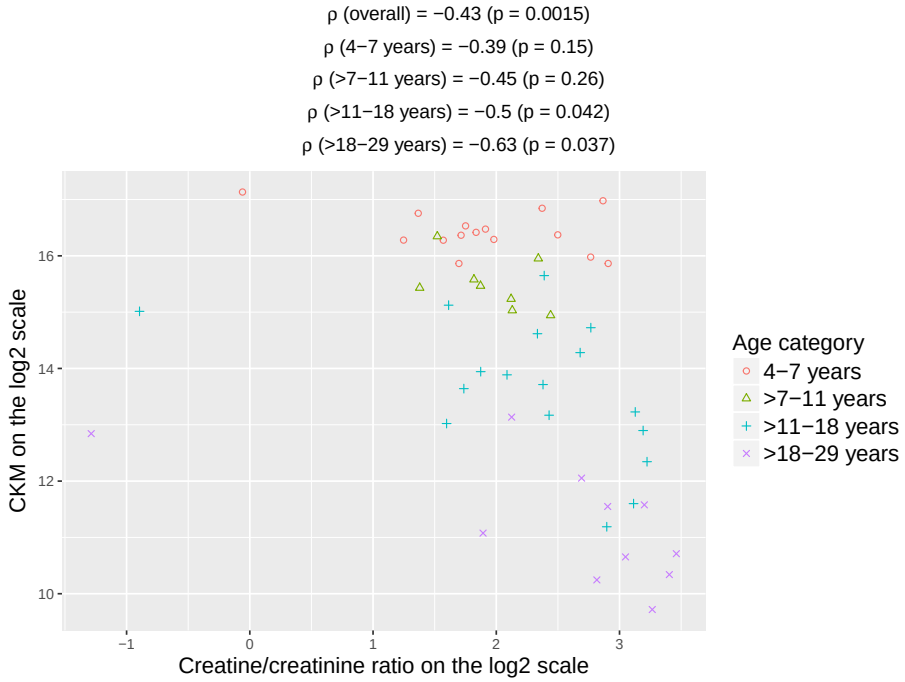
Only consider cases:

```
MasterFrame <- MasterFrame[MasterFrame$Status == "DMD",]
```

7.1 S2 Figure

Plot creatine/kinase ratio against CKM:

```
plotCKratioComp(MasterFrame, isoform="CKM")
```



8 Get session info

Session info:

```
sessionInfo()

## R version 3.3.1 (2016-06-21)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 14393)
##
```

```
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] ROCR_1.0-7         gplots_3.0.1       corrplot_0.77
## [4] RColorBrewer_1.1-2 genefilter_1.56.0  reshape_0.8.6
## [7] ggplot2_2.2.1      knitr_1.15.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.9          plyr_1.8.4           highr_0.6
##  [4] bitops_1.0-6         tools_3.3.1          digest_0.6.11
##  [7] annotate_1.52.1      evaluate_0.10        RSQLite_1.1-2
## [10] memoise_1.0.0        tibble_1.2           gtable_0.2.0
## [13] lattice_0.20-33      Matrix_1.2-6         DBI_0.5-1
## [16] parallel_3.3.1       stringr_1.1.0        caTools_1.17.1
## [19] gtools_3.5.0         S4Vectors_0.12.1     IRanges_2.8.1
## [22] stats4_3.3.1         Biobase_2.34.0       AnnotationDbi_1.36.1
## [25] XML_3.98-1.5         survival_2.40-1      gdata_2.17.0
## [28] magrittr_1.5         scales_0.4.1         BiocGenerics_0.20.0
## [31] splines_3.3.1        assertthat_0.1       xtable_1.8-2
## [34] colorspace_1.3-2     labeling_0.3         KernSmooth_2.23-15
## [37] stringi_1.1.2        RCurl_1.95-4.8       lazyeval_0.2.0
## [40] munsell_0.4.3
```