Appendix MRes 2021 Assignment (Jan-Philipp Cieslik)

Setup - Bash (Terminal)

Download data sets from Xena Browser (TCGA BRCA)
https://xenabrowser.net/datapages/?cohort=TCGA%20Breast%20Cancer%20(BRCA)

```
wget -O survival.tsv \
https://tcga-xena-hub.s3.us-east-1.amazonaws.com/download/survival%2FBRCA_survival.txt
wget -O clinical_matrix.tsv \
https://tcga-xena-hub.s3.us-east-1.amazonaws.com/download/TCGA.BRCA.sampleMap%2FBRCA_clinicalMatrix
wget -O HiSeqV2.tsv.gz \
https://tcga-xena-hub.s3.us-east-1.amazonaws.com/download/TCGA.BRCA.sampleMap%2FHiSeqV2.gz
wget -O Methylation450k.tsv.gz \
https://tcga-xena-hub.s3.us-east-1.amazonaws.com/download/TCGA.BRCA.sampleMap%2FHumanMethylation450.gz
wget -O Methylation450k_probemap.tsv \
https://tcga-xena-hub.s3.us-east-1.amazonaws.com/download/probeMap%2FilluminaMethyl450_hg19_GPL16304_TCGAlegacy
wget -O CNV_thresholded.tsv.gz \
https://tcga-xena-hub.s3.us-east-1.amazonaws.com/download/TCGA.BRCA.sampleMap%2FGistic2_CopyNumber_Gistic2_all_thre
```

Unzip the downloaded files

```
gunzip HiSeqV2.tsv.gz
gunzip CNV_thresholded.tsv.gz
gunzip Methylation450k.tsv.gz
```

Setup - R

Load libraries

```
library(survival)
#with the given option the fread function from data.table behaves like read.table
#but is much quicker and memory efficient
library(data.table)
options(datatable.fread.datatable=FALSE)
```

Data Loading - RNA/Survival/Clinical

Load data and adjust row names

```
#set the gene of interest
gene <- "SPIB"
#using check.names=FALSE to prevent the change from hyphens to dots
rna.data <- fread("HiSeqV2.tsv", sep="\t", head=TRUE,
                  stringsAsFactors =FALSE, check.names=FALSE)

rownames(rna.data) <- make.unique(rna.data[, 1])
rna.data <- rna.data[,-1]
rna.data <- as.matrix(rna.data)
```

```
surv.data <- read.table("survival.tsv", sep="\t", header=T, row.names=1)
clin.data <- read.table("clinical_matrix.tsv", sep="\t", header=T, row.names=1, quote = "")
```

Generate survival data

```
os.time <- surv.data[colnames(rna.data),"OS.time"]
os.event <- as.numeric(surv.data[colnames(rna.data),"OS"])
brca.os <- Surv(os.time,os.event)
#Delete local variables (optional, just to keep the environment clean)
rm("os.time", "os.event")
```

Univariate Regression Analysis (RNA/Survival)

Create empty data frame for results

```
rna.survival.univariate<-array(NA, c(nrow(rna.data),4))
colnames(rna.survival.univariate)<-c("HR","LCI","UCI","PVAL")
rownames(rna.survival.univariate)<-rownames(rna.data)
rna.survival.univariate<-as.data.frame(rna.survival.univariate)
```

Iterate through all genes and generate Cox model

```
for(i in 1:nrow(rna.data))
{
  #Check if less than 2 samples are available for correlation
  if(sum(!is.na(rna.data[i,])) < 2){
    next
  }
 coxphmodel <- coxph(brca.os ~ as.numeric(rna.data[i,]))
 summary <-summary(coxphmodel)
 rna.survival.univariate$HR[i] <- summary$coef[1,2]
 rna.survival.univariate$LCI[i] <- summary$conf.int[1,3]
 rna.survival.univariate$UCI[i] <- summary$conf.int[1,4]
 rna.survival.univariate$PVAL[i] <- summary$coef[1,5]
}
rna.survival.univariate <- as.data.frame(rna.survival.univariate)
rna.survival.univariate$FDR <- p.adjust(rna.survival.univariate$PVAL,method="fdr")
rna.survival.univariate <-
  rna.survival.univariate[order(rna.survival.univariate$FDR, decreasing=F),]

#Remove local variables
rm("summary", "i")
```

Print results of univariate analysis

```
kable(rna.survival.univariate[1:5,])
```

	HR	LCI	UCI	PVAL	FDR
LOC729467	1.3728688	1.2272374	1.5357818	0	0.0001751
EPHA5	1.3973600	1.2402194	1.5744108	0	0.0001751
PSME2	0.5940589	0.4930553	0.7157534	0	0.0001751
LOC148145	1.6993532	1.4157505	2.0397671	0	0.0001751
ANO6	1.5970418	1.3539352	1.8837995	0	0.0001751

Multivariate Regression Analysis (RNA/Survival/Clinical)

Clinical data preparation

```
#subset clinical data to patients that also have RNA data
clin.data<-clin.data[colnames(rna.data),]
#create age variable
age<-as.numeric(clin.data$age_at_initial_pathologic_diagnosis)
#create stage high/low variable
x3<-grep("III",clin.data$Converted_Stage_nature2012)
x4<-grep("IV",clin.data$Converted_Stage_nature2012)
stage.high<-rep(0,nrow(clin.data))
stage.high[c(x3,x4)]<-1

#Remove local variable
rm("x3", "x4")
```

Create empty data frame for results

```
rna.survival.multivariate<-array(NA, c(nrow(rna.data),4))
colnames(rna.survival.multivariate)<-c("HR","LCI","UCI","PVAL")
rownames(rna.survival.multivariate)<-rownames(rna.data)
rna.survival.multivariate<-as.data.frame(rna.survival.multivariate)
```

Iterate through all genes to generate multivariate regression model

```
for(i in 1:nrow(rna.data))
{
  #Check if less than 2 samples are available for correlation
  if(sum(!is.na(rna.data[i,])) < 2){
    next
  }
 coxphmodel <- coxph(brca.os ~ rna.data[i,]+age+stage.high)
 summary <- summary(coxphmodel)
 rna.survival.multivariate$HR[i] <- summary$coef[1,2]
 rna.survival.multivariate$LCI[i] <- summary$conf.int[1,3]
 rna.survival.multivariate$UCI[i] <- summary$conf.int[1,4]
 rna.survival.multivariate$PVAL[i] <- summary$coef[1,5]
}
rna.survival.multivariate <- as.data.frame(rna.survival.multivariate)
rna.survival.multivariate$FDR <- p.adjust(rna.survival.multivariate$PVAL,method="fdr")
rna.survival.multivariate <-
  rna.survival.multivariate[order(rna.survival.multivariate$FDR, decreasing=F),]
#Remove local variables
rm("summary", "i")
```

Print results of multivariate analysis

```
kable(rna.survival.multivariate[1:5,])
```

	HR	LCI	UCI	PVAL	FDR
SIAH2	0.6628204	0.5720274	0.7680241	0e+00	0.0006595
LOC148145	1.6765143	1.3900125	2.0220682	1e-07	0.0006595
MRO	1.2952128	1.1721721	1.4311689	4e-07	0.0010961
PSME2	0.6155775	0.5105837	0.7421618	4e-07	0.0010961
PCDHGA3	1.2974274	1.1741901	1.4335992	3e-07	0.0010961

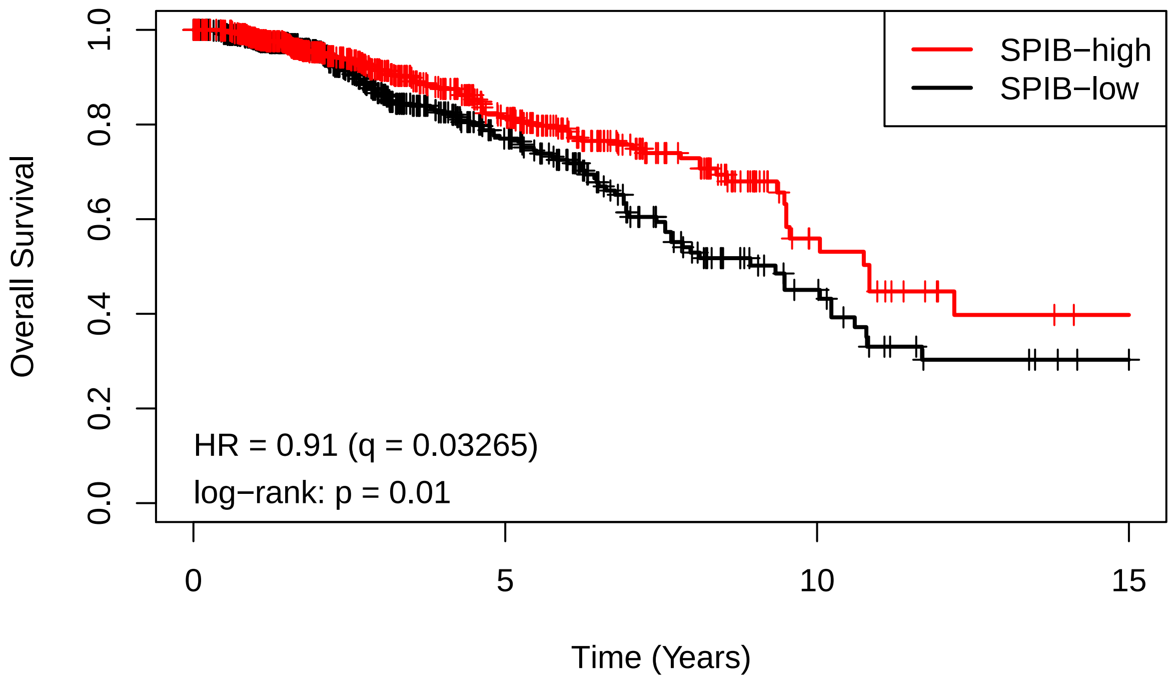
```
gene.info <- rna.survival.multivariate[gene,]
gene.high <- as.numeric(rna.data[gene,]>median(rna.data[gene,]))

#calculate log rank test
gene.high.logrank <- survdiff(brca.os ~ gene.high)
gene.high.logrank.p <- 1 - pchisq(gene.high.logrank$chisq, length(gene.high.logrank$n) - 1)
print(survfit(brca.os ~ gene.high))
```

```
## Call: survfit(formula = brca.os ~ gene.high)
##
##    4 observations deleted due to missingness
##                n events median 0.95LCL 0.95UCL
## gene.high=0 609    111   3409    2763    3873
## gene.high=1 605     87   3959    3472      NA
```

```
#create survival plot
plot.text <- paste0("HR = ",round(gene.info["HR"], digits = 2),
                    " (q = ",round(gene.info["FDR"], digits = 5),")")
plot.text2 <- paste0("log-rank: p = ",round(gene.high.logrank.p, digits = 2))
plot.legend <- c(paste0(gene,"-high"),paste0(gene,"-low"))


plot(survfit(brca.os ~ gene.high), col=c("black","red"), lwd=2,
     mark.time=TRUE, xlab="Time (Years)", ylab="Overall Survival",
     xscale = 365.25, xmax = 15*365.25)
legend("topright",legend=plot.legend,col=c("red","black"),lwd=2)
text(0,0.1,plot.text, adj = c(0,0))
text(0,0,plot.text2, adj = c(0,0))
```



```
#delete local variables
rm("plot.text", "plot.text2", "plot.legend")
```

Methylation data

Load methylation data

```
meth.annotation <- read.table("Methylation450k_probemap.tsv", sep="\t",
                              header=T, comment.char="")
meth.data <- fread("Methylation450k.tsv", sep="\t", header=TRUE)

row.names(meth.data) <- meth.data[,1]
row.names(meth.annotation) <- meth.annotation[,1]
meth.data <- meth.data[, -1]
meth.annotation <- meth.annotation[, -1]
meth.probes<- rownames(meth.annotation[grep(gene, meth.annotation$gene),])
```

Display methylation data

```
kable(meth.data[1:3,1:3])
```

	TCGA-OL-A66H-01	TCGA-3C-AALK-01	TCGA-AC-A5EH-01
cg13332474	0.0192	0.2032	0.3003
cg00651829	0.0179	0.2890	0.0892
cg17027195	0.0367	0.0750	0.0333

```
kable(meth.annotation[1:3,])
```

	gene	chrom	chromStart	chromEnd	strand
cg13332474	.	chr7	25935146	25935148	.
cg00651829	RSPH14,GNAZ	chr22	23413065	23413067	.
cg17027195	AUTS2	chr7	69064092	69064094	.

subset data set to samples that have rna, methylation and survival data.

```
samples <- intersect(colnames(meth.data),colnames(rna.data))
meth.intersect.rna <- meth.data[,samples]
rna.intersect.meth <- rna.data[,samples]
surv.intersect.rna.meth <-
  surv.data[intersect(rownames(surv.data),colnames(rna.intersect.meth)),]

meth.intersect.rna <- as.matrix(meth.intersect.rna)
rna.intersect.meth <- as.matrix(rna.intersect.meth)
surv.intersect.rna.meth <-
  as.data.frame(surv.intersect.rna.meth[colnames(meth.intersect.rna),])

meth.intersect.rna <-  meth.intersect.rna[meth.probes,,drop=FALSE]

rm("samples")
```

```
#exclude methylation sites that are not determined in more than 0.5 of samples
na.count <- apply(meth.intersect.rna,1,function(x) sum(as.numeric(is.na(x))))
exclude <- as.numeric(na.count>0.5*ncol(meth.intersect.rna))
meth.intersect.rna <- meth.intersect.rna[which(exclude==0),, drop=FALSE]

#generate empty array for results
results.meth<-array(NA,c(nrow(meth.intersect.rna),5))
rownames(results.meth)<-rownames(meth.intersect.rna)
colnames(results.meth)<-c("Cor","pval","qval","Mean.high","Mean.low")
gene.high.meth.rna <- as.numeric(
  as.numeric(rna.intersect.meth[gene,]) > median( as.numeric(rna.intersect.meth[gene,]) )
  )

#remove local variables
rm("na.count", "exclude")
```

Iterate through every methylation site of the selected gene and perform correlation.

```
for (i in 1:nrow(meth.intersect.rna))
{
  results.meth[i,1] <-
    cor.test(as.numeric(rna.intersect.meth[gene,]),as.numeric(meth.intersect.rna[i,]),
             use="c", method = "spearman",exact=FALSE)$est
  results.meth[i,2] <-
    cor.test(as.numeric(rna.intersect.meth[gene,]),as.numeric(meth.intersect.rna[i,]),
             use="c", method = "spearman",exact=FALSE)$p.value
}
results.meth[,4] <- apply(meth.intersect.rna[,which(gene.high.meth.rna==1), drop=FALSE],
                          1,mean,na.rm=T)
results.meth[,5] <- apply(meth.intersect.rna[,which(gene.high.meth.rna==0), drop=FALSE],
                          1,mean,na.rm=T)
results.meth[,3] <- p.adjust(results.meth[,2],method="fdr")
results.meth<-results.meth[order(results.meth[,3], decreasing=F),,drop=FALSE]

#remove local variables
rm("i")
```

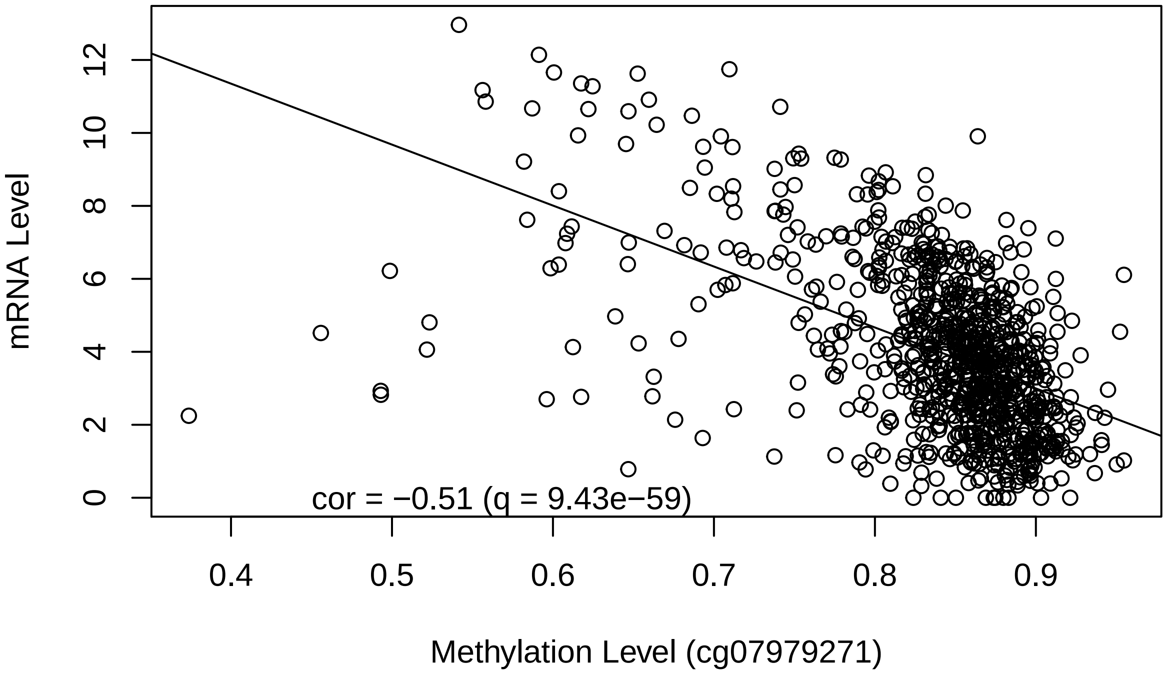
Display results

```
kable(results.meth)
```

	Cor	pval	qval	Mean.high	Mean.low
cg07979271	-5.14e-01	0.00e+00	0.00e+00	0.816	0.864
cg13403724	2.88e-01	0.00e+00	0.00e+00	0.248	0.206
cg17774764	-2.81e-01	0.00e+00	0.00e+00	0.737	0.775
cg13918544	-2.77e-01	0.00e+00	0.00e+00	0.832	0.859
cg06512885	2.72e-01	0.00e+00	0.00e+00	0.191	0.168
cg15690347	2.66e-01	0.00e+00	0.00e+00	0.383	0.301
cg24092179	2.65e-01	0.00e+00	0.00e+00	0.260	0.226
cg15007959	2.63e-01	0.00e+00	0.00e+00	0.253	0.203
cg18254819	2.46e-01	0.00e+00	0.00e+00	0.235	0.212
cg03763616	2.28e-01	0.00e+00	0.00e+00	0.271	0.244
cg22268231	-1.47e-01	1.22e-05	1.88e-05	0.450	0.487
cg21152077	-1.33e-01	7.89e-05	1.12e-04	0.875	0.881
cg26522743	-9.68e-02	4.20e-03	5.49e-03	0.394	0.424
cg04508467	9.21e-02	6.53e-03	7.92e-03	0.464	0.448
cg08201854	8.45e-02	1.25e-02	1.41e-02	0.132	0.143
cg22745102	7.05e-02	3.72e-02	3.96e-02	0.469	0.462
cg19387862	2.02e-05	1.00e+00	1.00e+00	0.696	0.686

```
#set the methylation site of interest
meth.site = "cg07979271"
plot.title <- paste0(gene, " in BRCA")
plot.text <- paste0("cor = ",round(results.meth[meth.site, "Cor"], digits = 2),
                    " (q = ",format(results.meth[meth.site, "qval"], scientific = TRUE),")")
plot.legend <- c(paste0(gene,"-high"),paste0(gene,"-low"))


plot(as.numeric(meth.intersect.rna[meth.site,]), as.numeric(rna.intersect.meth[gene,]),
     xlab=paste0("Methylation Level (", meth.site,")"), ylab="mRNA Level")
text(0.45,0,plot.text, adj = c(0,0.5))
abline(lm(rna.intersect.meth[gene,] ~ meth.intersect.rna[meth.site,]))
```



CNV data

Load CNV data and display table

```
cnv.data <- fread("CNV_thresholded.tsv", sep="\t", header=T)
rownames(cnv.data) <- make.unique(cnv.data[,1])
cnv.data <- cnv.data[,-1]
cnv.data <- as.matrix(cnv.data)
#subset data set to entries that are also available in the RNA data set
cols.intersect <- intersect(colnames(cnv.data), colnames(rna.data))
row.intersect <- intersect(rownames(cnv.data), rownames(rna.data))
cnv.intersect.rna <- cnv.data[row.intersect, cols.intersect]
rna.intersect.cnv <- rna.data[row.intersect, cols.intersect]
kable(cnv.data[1:5,1:3])
```

	TCGA-3C-AAAU-01	TCGA-3C-AALI-01	TCGA-3C-AALJ-01
ACAP3	0	-1	-1
ACTRT2	0	-1	-1
AGRN	0	-1	-1
ANKRD65	0	-1	-1
ATAD3A	0	-1	-1

```
rm("cols.intersect","row.intersect")
```

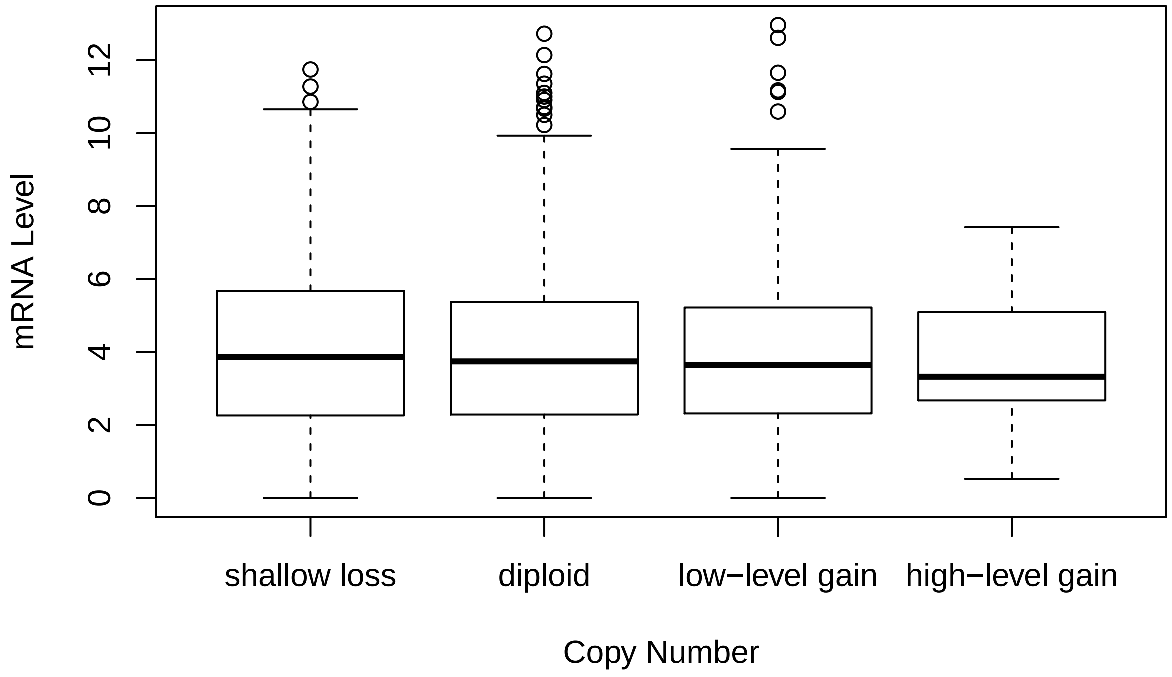
Calculate CNV changes in selected gene

```r
cnv.rna.df <- data.frame(CNV = cnv.intersect.rna[gene,],
                         RNA = rna.intersect.cnv[gene,],
                         stringsAsFactors=FALSE)
cnv.rna.df[which(cnv.rna.df$CNV == -2), "CNV"] <- "deep loss"
cnv.rna.df[which(cnv.rna.df$CNV == -1), "CNV"] <- "shallow loss"
cnv.rna.df[which(cnv.rna.df$CNV == 0), "CNV"] <- "diploid"
cnv.rna.df[which(cnv.rna.df$CNV == 1), "CNV"] <- "low-level gain"
cnv.rna.df[which(cnv.rna.df$CNV == 2), "CNV"] <- "high-level gain"
#"deep loss" is missing on purpose,
#as no entry falls into this category for our candidate gene
cnv.rna.df$CNV <- factor(cnv.rna.df$CNV,
                         levels=c("shallow loss", "diploid",
                                  "low-level gain", "high-level gain"),
                         ordered=TRUE)

aov(RNA ~ CNV, data = cnv.rna.df)
```

```
## Call:
##    aov(formula = RNA ~ CNV, data = cnv.rna.df)
##
## Terms:
##                       CNV Residuals
## Sum of Squares     2.716  5920.177
## Deg. of Freedom        3      1072
##
## Residual standard error: 2.350011
## Estimated effects may be unbalanced
## 2 observations deleted due to missingness
```

```r
boxplot(RNA~CNV, data=cnv.rna.df,
   xlab="Copy Number", ylab="mRNA Level")
```



mRNA correlation

Create empty data frame for results

```
rna.cor <-array(NA,c(nrow(rna.data),5))
rownames(rna.cor)<-rownames(rna.data)
colnames(rna.cor)<-c("Cor","pval","qval","Mean.high","Mean.low")

gene.high.meth.rna <- as.numeric(
  as.numeric(rna.data[gene,]) > median( as.numeric(rna.data[gene,]) )
  )
```

Loop through every gene and correlate it with the candidate gene

```
for (i in 1:nrow(rna.data)){
  #Check if less than 2 samples are available for correlation
  if(sum(!is.na(rna.data[i,]) & !is.na(rna.data[gene,])) < 2){
    next
  }
  #Check if all values are zero
  if(sum(rna.data[i,]) == 0){
    next
  }
  if(sum(!is.na(rna.data[i,])))
  result.temp <-
    cor.test(as.numeric(rna.data[gene,]), as.numeric(rna.data[i,]),
             use="c", method = "spearman", exact=FALSE)
  rna.cor[i, 1:2] <- c(result.temp$est, result.temp$p.value)
}
rna.cor[,4] <- apply(rna.data[,which(gene.high.meth.rna==1), drop=FALSE],1,mean,na.rm=T)
rna.cor[,5] <- apply(rna.data[,which(gene.high.meth.rna==0), drop=FALSE],1,mean,na.rm=T)
rna.cor[,3] <- p.adjust(rna.cor[,2],method="fdr")
rna.cor <- rna.cor[order(rna.cor[,3], decreasing=F),,drop=FALSE]
```

Display results

```
kable(rna.cor[1:5,])
```

	Cor	pval	qval	Mean.high	Mean.low
SPIB	1.0000000	0	0	5.687704	2.108523
MS4A1	0.8217339	0	0	6.883461	3.159181
TCL1A	0.8196190	0	0	4.234020	1.084256
CXCR5	0.8194083	0	0	5.152683	2.590970
LCK	0.8161056	0	0	7.855356	5.622874

Save workspace

```
save.image("assignment.RData")
```