Removed Samples Tumor MDS Plots Take 2

Annika Jorgensen

2023-12-02

Defining Colors

This chunk defines color palette variables that are going to be used in plots later on the script. These variables are defined by converting BrewerCode palettes into palettes that can be used in R.

```
viralPalette <- brewer.pal(8, "Set1")
hbvColor <- viralPalette[1]
hcvColor <- viralPalette[2]
bothColor <- viralPalette[3]
neitherColor <- viralPalette[4]

sexTissuePalette <- brewer.pal(12, "Paired")
maleTumorColor <- sexTissuePalette[4]
maleAdjacentColor <- sexTissuePalette[3]
femaleTumorColor <- sexTissuePalette[6]
femaleAdjacentColor <- sexTissuePalette[5]
```

This code is where you read in all the data files that are going to be used in the script. The data is also converted into a variety of variables that makes the data easier to handle. The data is also cleaned up to make sure the analysis done later is accurate and precise.

```
metadata <- read.table("~/Desktop/ResearchProjects/LiverCancer/Metadata/metadata_for_de.csv", row.names=
tumorAdjacentExp <- read.table("~/Desktop/ResearchProjects/LiverCancer/Metadata/japan_all_samples_salmo
colnames(tumorAdjacentExp) <- gsub("\\.", "-", colnames(tumorAdjacentExp)) #changing the column names

# Importing gene annotations
#genes <- read.table("gencode.v25.chr_patch_hapl_scaff.annotation.bed", header=FALSE, sep="\t")
genes <- read.table("~/Desktop/ResearchProjects/LiverCancer/Metadata/gencodeTranscripts.txt", header=TRU
genes <- data.frame(genes)
tumorAdjacentExp <- tumorAdjacentExp[rownames(tumorAdjacentExp) %in% genes$GENEID ,]
genes <- genes[match(rownames(tumorAdjacentExp), genes$GENEID),]
# Calculating gene length, this is needed for calculating the FPKM values
genes$length <- with(genes, end - start)

# Removing Samples due to low quality
metadata <- metadata[!(metadata$ID == "RK023") , ]
metadata <- metadata[!(metadata$ID == "RK106") , ]
metadata <- metadata[!(metadata$ID == "RK113") , ]
metadata <- metadata[!(metadata$ID == "RK135") , ]
metadata <- metadata[!(metadata$ID == "RK105") , ]
metadata <- metadata[!(metadata$ID == "RK116") , ]
metadata <- metadata[!(metadata$ID == "RK066") , ]
```

```r
metadata <- metadata[!(metadata$ID == "RK096") , ]

#Removing both and NBNC samples
metadata <- metadata[!(metadata$Virus_infection == "NBNC"), ]
metadata <- metadata[!(metadata$Virus_infection == "both"), ]

# Subsetting and ordering metadata to match the count matrix
tumorAdjacentExpSubset <- tumorAdjacentExp[,colnames(tumorAdjacentExp) %in% metadata$sampleid]
metadataSubset <- metadata[metadata$sampleid %in% colnames(tumorAdjacentExpSubset),]
metadataSubset <- metadataSubset[match(colnames(tumorAdjacentExpSubset), metadataSubset$sampleid),]
rownames(metadataSubset) <- metadataSubset$sampleid

# Adding tissue type, converting categorical variables to factors
metadataSubset$tumor <- as.numeric(grepl('tumor', metadataSubset$sampleid, ignore.case=T))

#Swapping lesion type for sample RK169
metadataSubset["RK169-tumor-XY","tumor"] <- 0
metadataSubset["RK169-adjacent-XY","tumor"] <- 1

#Changing rownames to match swapped lesion type
rownames(metadataSubset)[rownames(metadataSubset)=="RK169-tumor-XY"] <- "RK169_adjacent-XY"
rownames(metadataSubset)[rownames(metadataSubset)=="RK169-adjacent-XY"] <- "RK169_tumor-XY"
rownames(metadataSubset)[rownames(metadataSubset)=="RK169_adjacent-XY"] <- "RK169-adjacent-XY"

rownames(tumorAdjacentExpSubset)[rownames(tumorAdjacentExpSubset)=="RK169-tumor-XY"] <- "RK169_adjacent
rownames(tumorAdjacentExpSubset)[rownames(tumorAdjacentExpSubset)=="RK169-adjacent-XY"] <- "RK169_tumor
rownames(tumorAdjacentExpSubset)[rownames(tumorAdjacentExpSubset)=="RK169_adjacent-XY"] <- "RK169-adjac

#Swapping lesion type for sample RK065
metadataSubset["RK065-tumor-XX","tumor"] <- 0
metadataSubset["RK065-adjacent-XX","tumor"] <- 1

#Changing rownames in metadata to match swapped lesion type
rownames(metadataSubset)[rownames(metadataSubset)=="RK065-tumor-XY"] <- "RK065_adjacent-XY"
rownames(metadataSubset)[rownames(metadataSubset)=="RK065-adjacent-XY"] <- "RK065_tumor-XY"
rownames(metadataSubset)[rownames(metadataSubset)=="RK065_adjacent-XY"] <- "RK065-adjacent-XY"

rownames(tumorAdjacentExpSubset)[rownames(tumorAdjacentExpSubset)=="RK065-tumor-XY"] <- "RK065_adjacent
rownames(tumorAdjacentExpSubset)[rownames(tumorAdjacentExpSubset)=="RK065-adjacent-XY"] <- "RK065_tumor
rownames(tumorAdjacentExpSubset)[rownames(tumorAdjacentExpSubset)=="RK065_adjacent-XY"] <- "RK065-adjac

metadataSubset$gender_tissue <- paste(metadataSubset$Gender, metadataSubset$tumor, sep="_")
metadataSubset$gender_tissue_viral <- paste(metadataSubset$gender_tissue, metadataSubset$Virus_infectio
metadataSubset$library_type <- metadataSubset$strandedness
metadataSubset$library_type <- factor(metadataSubset$library_type)
metadataSubset$tumor <- factor(metadataSubset$tumor)
metadataSubset$Ta <- factor(metadataSubset$Ta)
metadataSubset$Portal_vein_invasion <- factor(metadataSubset$Portal_vein_invasion)
metadataSubset$Hepatic_vein_invasion <- factor(metadataSubset$Hepatic_vein_invasion)
metadataSubset$Bile_duct_invasion <- factor(metadataSubset$Bile_duct_invasion)
metadataSubset$Liver_fibrosisc <- factor(metadataSubset$Liver_fibrosisc)
metadataSubset$Prognosis <- factor(metadataSubset$Prognosis)
```

```r
# Creating the DGEList object
dge <- DGEList(counts=tumorAdjacentExpSubset, genes=genes)
colnames(dge) <- colnames(tumorAdjacentExpSubset)
dge$samples$sex <- metadataSubset$Gender
dge$samples$viral <- factor(metadataSubset$Virus_infection)
dge$samples$ID <- metadataSubset$ID
dge$samples$tumor <- metadataSubset$tumor
dge$samples$gender_tissue <- metadataSubset$gender_tissue
dge$samples$gender_tissue_viral <- metadataSubset$gender_tissue_viral
dge$samples$library_type <- metadataSubset$library_type
dge$samples$edmonson_grade <- metadataSubset$Edmondson_grade
dge$samples$Ta <- metadataSubset$Ta
dge$samples$survival <- metadataSubset$Overall_survival_month

# Inspecting the N of samples in each group
table(dge$samples$gender_tissue_viral)
```

```
##
## F_0_HBV F_0_HCV F_1_HBV F_1_HCV M_0_HBV M_0_HCV M_1_HBV M_1_HCV
##       7      32       8      33      31      59      37      71
```

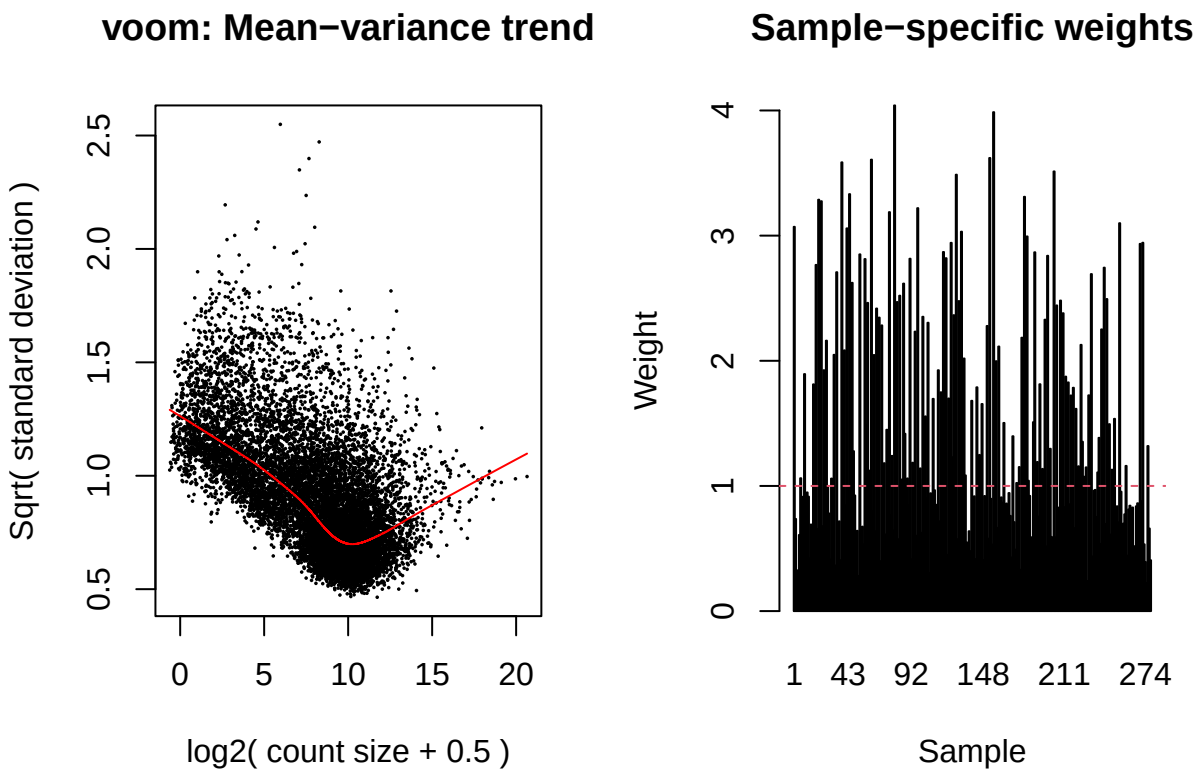
```r
# ====================================
# Filtering expression data
# ====================================

# Keeping genes that have a mean FPKM of at least 0.5 in at least one of the
# groups under investigation and at least 6 reads in at least 10 samples
fpkm <- rpkm(dge, gene.length=dge$genes$length)

M_1_HBV_mean_fpkm <- apply(as.data.frame(fpkm)[(dge$samples$gender_tissue_viral=="M_1_HBV")],1,mean,na.
M_0_HBV_mean_fpkm <- apply(as.data.frame(fpkm)[(dge$samples$gender_tissue_viral=="M_0_HBV")],1,mean,na.
M_1_HCV_mean_fpkm <- apply(as.data.frame(fpkm)[(dge$samples$gender_tissue_viral=="M_1_HCV")],1,mean,na.
M_0_HCV_mean_fpkm <- apply(as.data.frame(fpkm)[(dge$samples$gender_tissue_viral=="M_0_HCV")],1,mean,na.

F_1_HBV_mean_fpkm <- apply(as.data.frame(fpkm)[(dge$samples$gender_tissue_viral=="F_1_HBV")],1,mean,na.
F_0_HBV_mean_fpkm <- apply(as.data.frame(fpkm)[(dge$samples$gender_tissue_viral=="F_0_HBV")],1,mean,na.
F_1_HCV_mean_fpkm <- apply(as.data.frame(fpkm)[(dge$samples$gender_tissue_viral=="F_1_HCV")],1,mean,na.
F_0_HCV_mean_fpkm <- apply(as.data.frame(fpkm)[(dge$samples$gender_tissue_viral=="F_0_HCV")],1,mean,na.

keep <- (M_1_HBV_mean_fpkm > 0.5 | M_0_HBV_mean_fpkm > 0.5 |
           M_1_HCV_mean_fpkm > 0.5 | M_0_HCV_mean_fpkm > 0.5 |
           F_1_HBV_mean_fpkm > 0.5 | F_0_HBV_mean_fpkm > 0.5 |
           F_1_HCV_mean_fpkm > 0.5 | F_0_HCV_mean_fpkm > 0.5 )

dge <- dge[keep,,keep.lib.sizes=FALSE]
dge <- calcNormFactors(dge, method="TMM")
keep <- rowSums(dge$counts > 6) >= 10
dge <- dge[keep,,keep.lib.size=FALSE]
dge <- calcNormFactors(dge, method="TMM")

# N of genes retained after filtering
dim(dge$genes)
```

```
## [1] 12465     7
```

```
# ================================== ====================
# ============================================================
# Analysis of all tumor vs. tumor-adjacent regardless of sex
# ============================================================
# ============================================================

# Creating a new design model matrix with the variable of interest and the
# library type
design <- model.matrix(~0 + dge$sample$tumor + dge$samples$library_type)
colnames(design) <- gsub("dge\\$samples\\$tumor", "tumor", colnames(design))
colnames(design) <- gsub("dge\\$samples\\$library_typeunstranded", "library_type", colnames(design))
head(design)
```

```
##   dge$sample$tumor0 dge$sample$tumor1 library_type
## 1                 1                 0            1
## 2                 0                 1            1
## 3                 1                 0            1
## 4                 0                 1            1
## 5                 1                 0            1
## 6                 0                 1            1
```

```
# Running voom again with the new design matrix.
v <- voomWithQualityWeights(dge, design, plot=TRUE)
```



Top 25 gene MDS Plot This creates outputs an MDS Plot that we've previously see showing the dissimilarity between Male and Female Tumor-Tumor adjacent

```
# Removing batch effects
vCorrectLibtype <- removeBatchEffect(v, batch=v$targets$library_type)
```

```
pdf("~/Desktop/ResearchProjects/SexChromosomeGithubUpload/Removed_samples_MDS_plot/figures/Library_type_
mds <- plotMDS(v, top = 25, dim.plot = c(1,2), plot=TRUE, cex=2,
               pch=ifelse(v$targets$viral %in% c("HBV"), 17,
                          ifelse(v$targets$viral %in% c("HCV"), 15, 3)),
               col=ifelse(v$targets$library_type=="stranded", "#1A85FF",
                          ifelse(v$targets$library_type=="unstranded", "#D41159", "black")),
               gene.selection = "common")
legend("top", pch=c(15),
       col=c("#D41159", "#1A85FF") ,
       legend=c("Unstranded", "Stranded"))
legend("center", pch=c(17, 15),
       col=c("black"),
       legend=c("HBV", "HCV"))

dev.off()
```

```
## pdf
##   2
```

Recreating the MDS plot we have seen previously with the removed samples to see if all low quality samples have been removed.

```
# Removing batch effects
vCorrectLibtype <- removeBatchEffect(v, batch=v$targets$library_type)

#creating MDS plot top 25 genes PC 1 and 2 coloring by lesion type and sex
pdf("~/Desktop/ResearchProjects/SexChromosomeGithubUpload/Removed_samples_MDS_plot/figures/removed_sampl
mds <- plotMDS(vCorrectLibtype, top = 25, ndim = 10, dim.plot = c(1,2), plot=TRUE, cex=2,
               pch=ifelse(v$targets$viral %in% c("HBV"), 17,
                          ifelse(v$targets$viral %in% c("HCV"), 15,
                                 ifelse(v$targets$viral %in% c("both"), 16,  3))),
               col=ifelse(v$targets$gender_tissue=="M_1",
                          "#FFC20A",
                          ifelse(v$targets$gender_tissue=="M_0",
                                 "#E66100",
                                 ifelse(v$targets$gender_tissue=="F_1",
                                        "#40B0A6", "#006CD1"))),
               gene.selection = "common")
```

```
## Warning in plot.window(...): "ndim" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "ndim" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "ndim" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "ndim" is not a
## graphical parameter
```

```
## Warning in box(...): "ndim" is not a graphical parameter

## Warning in title(...): "ndim" is not a graphical parameter
```

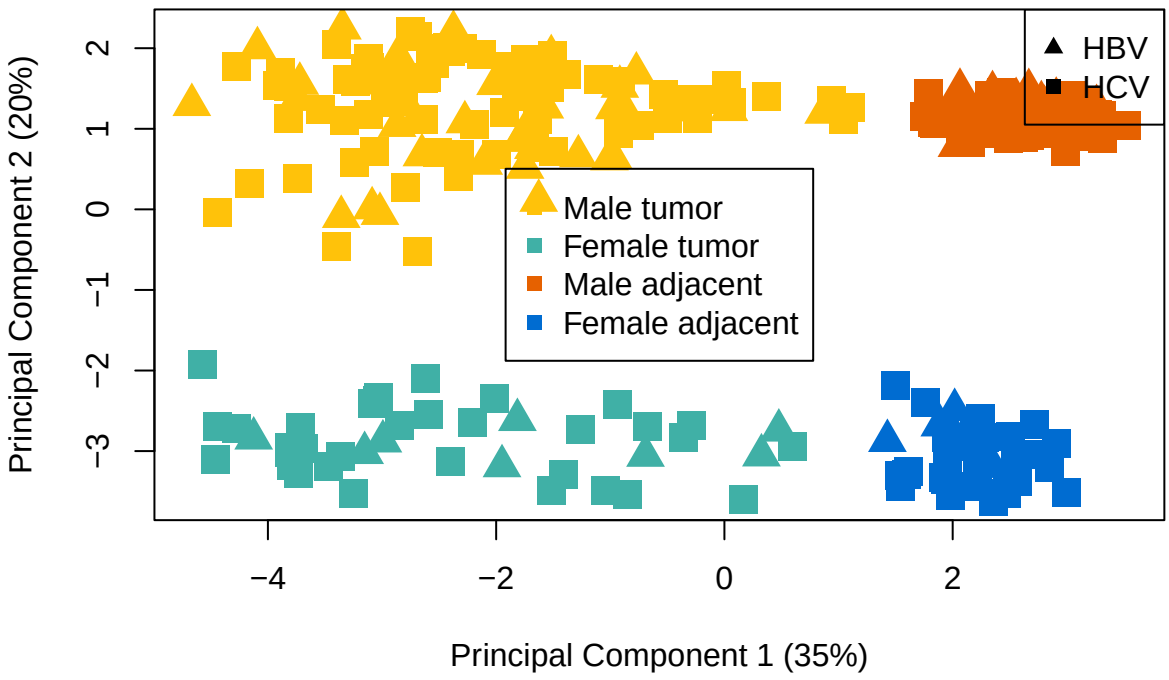
```
legend("center", pch=c(15),
       col=c( "#FFC20A", "#40B0A6","#E66100", "#006CD1"),
      legend=c("Male tumor","Female tumor", "Male adjacent","Female adjacent"))
legend("topright", pch=c(17, 15, 16, 3),
       col=c("black"),
       legend=c("HBV", "HCV"))
dev.off()
```

```
## pdf
##   2
```

```
mds <- plotMDS(vCorrectLibtype, top = 25, ndim = 10, dim.plot = c(1,2), plot=TRUE, cex=2,
               pch=ifelse(v$targets$viral %in% c("HBV"), 17,
                          ifelse(v$targets$viral %in% c("HCV"), 15,
                                 ifelse(v$targets$viral %in% c("both"), 16,  3))),
               col=ifelse(v$targets$gender_tissue=="M_1",
                          "#FFC20A",
                         ifelse(v$targets$gender_tissue=="M_0",
                                "#E66100",
                                 ifelse(v$targets$gender_tissue=="F_1",
                                        "#40B0A6", "#006CD1"))),
               gene.selection = "common")
```

```
## Warning in plot.window(...): "ndim" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "ndim" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "ndim" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "ndim" is not a
## graphical parameter

## Warning in box(...): "ndim" is not a graphical parameter

## Warning in title(...): "ndim" is not a graphical parameter
```

```
legend("center", pch=c(15),
       col=c( "#FFC20A", "#40B0A6","#E66100", "#006CD1"),
      legend=c("Male tumor","Female tumor", "Male adjacent","Female adjacent"))
legend("topright", pch=c(17, 15, 16, 3),
 #      col=c("black"),
       legend=c("HBV", "HCV"))
```


HBV
HCV
Male tumor
Female tumor
Male adjacent
Female adjacent
Principal Component 2 (20%)
Principal Component 1 (35%)
−4
−2
0
2
−3
−2
−1
0
1
2