MDS Plots for Identifying Low Quality Samples

Annika Jorgensen

2023-06-07

R Markdown

Title: "MDS Plots for Identifying Low Quality Samples"
Author: Annika Jorgensen
Date: 06/07/2023

Purpose This is the second version of MDS_Plots.Rmd file with the goal of finding the Outlier Points in the combined male/female/tumor/tumor adjacent MDS plots created with the top 25 genes and top 50 genes.

Methods An argument was added to the plotMDS function where the points were labeled based off of the column names in the vCorrectLibtype matrix.

Multiple MDS plots were then created to identify where the points were colored based off of the sample ID to see if their paired sample was also mislabled (e.g. if a tumor sample RK-XXX was identified as an outlier then and MDS plot was created to see if the adjacent sample was also an outlier).

Results ### **Libraries** The first chunk of code is dedicated to installing the libraries. These libraries are to help execute the differential analysis and helps visualize the data. The code was not included for concision.

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]
```

Read in data

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.

```r
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=TR
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 RK023 due to low quality
metadata <- metadata[!(metadata$ID == "RK023") , ]

#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))
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
```

```r
# 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
##       8      34       9      36      33      59      40      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] 12554     7
```

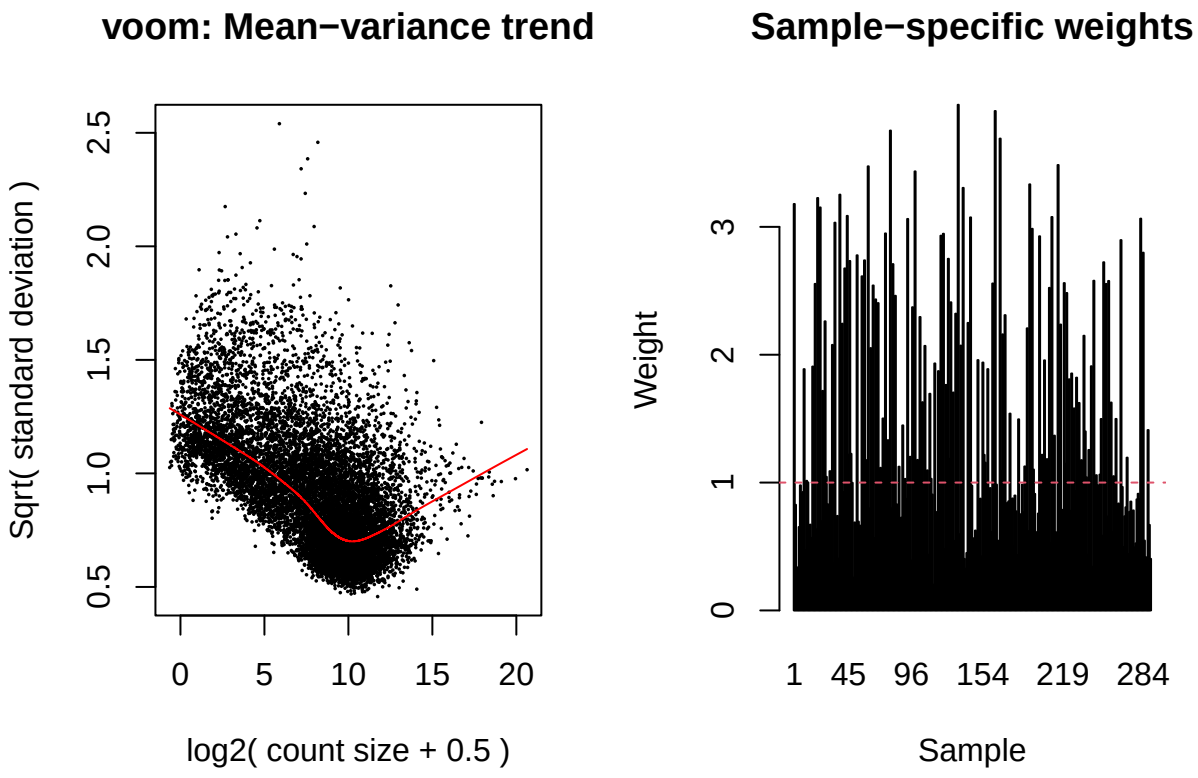
```r
# ==================================== ======================
# ============================================================
# Analysis of all tumor vs. tumor-adjacent regardless of sex
# ============================================================
# ============================================================

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

```
colnames(design) <- gsub("dge\\$samples\\$Ta2", "Ta2", colnames(design))
colnames(design) <- gsub("dge\\$samples\\$Ta3", "Ta3", colnames(design))
colnames(design) <- gsub("dge\\$samples\\$Ta4", "Ta4", colnames(design))
head(design)
```

```
##   tumor0 tumor1 library_type Ta2 Ta3 Ta4
## 1      1      0            1   0   0   1
## 2      0      1            1   0   0   1
## 3      1      0            1   1   0   0
## 4      0      1            1   1   0   0
## 5      1      0            1   0   1   0
## 6      0      1            1   0   1   0
```

```
# 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/Identifying_low_quality_samples/figures/MDS_P
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("bottomleft", 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
```

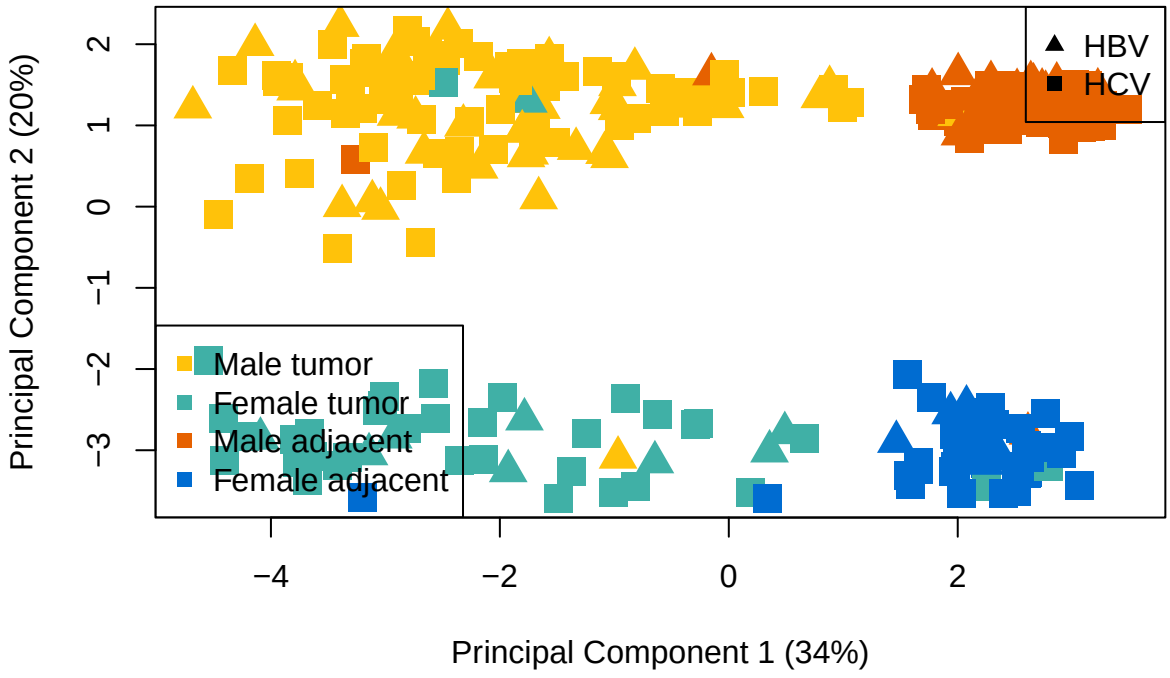
```
## Warning in axis(side = side, at = at, labels = labels, ...): "ndim" is not a
## graphical parameter

## 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("bottomleft", 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"))
```



This MDS plot writes the same name for each dot on the MDS plot seen above. The intent of this is to see the sample name of the samples that are potential outliers.

```
##extracting all of the pairwise squared distances
#matrixdistance <- mds$distance.matrix.squared

#changing it to dataframes
#matrixdistance <-as.data.frame(matrixdistance)
```

```
##extracting the female tumor samples from the distance matrix
#femaletumordistance <- matrixdistance[, grep("tumor-XX" ,colnames(matrixdistance))] + #matrixdistance[

pdf("~/Desktop/ResearchProjects/SexChromosomeGithubUpload/Identifying_low_quality_samples/figures/Sampl
mds <- plotMDS(vCorrectLibtype, top = 25, labels= colnames(vCorrectLibtype), ndim = 10, dim.plot = c(1,
               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

## Warning in text.default(x$x, x$y, labels = labels, cex = cex, ...): "ndim" is
## not a graphical parameter
```

```
legend("bottomleft", 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
```

```
#vCorrectLibtype <- as.data.frame(vCorrectLibtype)
```

This code chunk colors all samples with the ID RK106 yellow and the rest of the samples are colored black. The MDS plots have kept the labels that are the column names of the variable vCorrectLibtype.

```
pdf("~/Desktop/ResearchProjects/SexChromosomeGithubUpload/Identifying_low_quality_samples/figures/RK106_
mds <- plotMDS(vCorrectLibtype, top = 25, labels= colnames(vCorrectLibtype), ndim = 10, dim.plot = c(1,2
               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$targets$ID=="RK106",
                          "#40B0A6", "black"),
               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

## Warning in text.default(x$x, x$y, labels = labels, cex = cex, ...): "ndim" is
## not a graphical parameter
```

```
legend("bottomleft", 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
```

This code chunk colors all samples with the ID RK113 yellow and the rest of the samples are colored black. The MDS plots have kept the labels that are the column names of the variable vCorrectLibtype.

The goal of the coloring is to see where the samples are located on the MDS plot.

```
pdf("~/Desktop/ResearchProjects/SexChromosomeGithubUpload/Identifying_low_quality_samples/figures/RK113_
mds <- plotMDS(vCorrectLibtype, top = 25, labels= colnames(vCorrectLibtype), ndim = 10, dim.plot = c(1,2
               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$targets$ID=="RK113",
                          "#FFC20A", "black"),
               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

## Warning in text.default(x$x, x$y, labels = labels, cex = cex, ...): "ndim" is
## not a graphical parameter
```

```
legend("bottomleft", 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
```

This code chunk colors all samples by patient ID and the rest of the samples are colored black. The MDS plots have kept the labels that are the column names of the variable vCorrectLibtype.

The goal of the coloring is to see where the samples are located on the MDS plot.

```
pdf("~/Desktop/ResearchProjects/SexChromosomeGithubUpload/Identifying_low_quality_samples/figures/RK169_
mds <- plotMDS(vCorrectLibtype, top = 25, labels= colnames(vCorrectLibtype), ndim = 10, dim.plot = c(1,2
               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$targets$ID=="RK169",
                          "#FFC20A", "black"),
               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

## Warning in text.default(x$x, x$y, labels = labels, cex = cex, ...): "ndim" is
## not a graphical parameter
```

```
legend("bottomleft", 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
```

```
pdf("~/Desktop/ResearchProjects/SexChromosomeGithubUpload/Identifying_low_quality_samples/figures/RK135_
mds <- plotMDS(vCorrectLibtype, top = 25, labels= colnames(vCorrectLibtype), ndim = 10, dim.plot = c(1,2
               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$targets$ID=="RK135",
                          "#FFC20A", "black"),
               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

## Warning in text.default(x$x, x$y, labels = labels, cex = cex, ...): "ndim" is
## not a graphical parameter
```

```
legend("bottomleft", 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
```

```
pdf("~/Desktop/ResearchProjects/SexChromosomeGithubUpload/Identifying_low_quality_samples/figures/RK105_
mds <- plotMDS(vCorrectLibtype, top = 25, labels= colnames(vCorrectLibtype), ndim = 10, dim.plot = c(1,1
               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$targets$ID=="RK105",
                          "#FFC20A", "black"),
               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

## Warning in text.default(x$x, x$y, labels = labels, cex = cex, ...): "ndim" is
## not a graphical parameter
```

```
legend("bottomleft", 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
```

```
pdf("~/Desktop/ResearchProjects/SexChromosomeGithubUpload/Identifying_low_quality_samples/figures/RK065_
mds <- plotMDS(vCorrectLibtype, top = 25, labels= colnames(vCorrectLibtype), ndim = 10, dim.plot = c(1,1
               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$targets$ID=="RK065",
                          "#40B0A6", "black"),
               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

## Warning in text.default(x$x, x$y, labels = labels, cex = cex, ...): "ndim" is
## not a graphical parameter
```

```
legend("bottomleft", 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
```

```
pdf("~/Desktop/ResearchProjects/SexChromosomeGithubUpload/Identifying_low_quality_samples/figures/RK116_
mds <- plotMDS(vCorrectLibtype, top = 25, labels= colnames(vCorrectLibtype), ndim = 10, dim.plot = c(1,1
               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$targets$ID=="RK116",
                          "#40B0A6", "black"),
               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

## Warning in text.default(x$x, x$y, labels = labels, cex = cex, ...): "ndim" is
## not a graphical parameter
```

```
legend("bottomleft", 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
```

```
pdf("~/Desktop/ResearchProjects/SexChromosomeGithubUpload/Identifying_low_quality_samples/figures/RK066_
mds <- plotMDS(vCorrectLibtype, top = 25, labels= colnames(vCorrectLibtype), ndim = 10, dim.plot = c(1,2
               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$targets$ID=="RK066",
                          "#40B0A6", "black"),
               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

## Warning in text.default(x$x, x$y, labels = labels, cex = cex, ...): "ndim" is
## not a graphical parameter
```

```
legend("bottomleft", 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
```

```
pdf("~/Desktop/ResearchProjects/SexChromosomeGithubUpload/Identifying_low_quality_samples/figures/RK096_
mds <- plotMDS(vCorrectLibtype, top = 25, labels= colnames(vCorrectLibtype), ndim = 10, dim.plot = c(1,2
               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$targets$ID=="RK096",
                          "#FFC20A", "black"),
               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

## Warning in text.default(x$x, x$y, labels = labels, cex = cex, ...): "ndim" is
## not a graphical parameter
```

```
legend("bottomleft", 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
```

Conclusion Nine samples and their paired samples were investigated in this file. Sample RK-135-tumor-XX is a female tumor sample that was located in the male tumor sample cluster was found to have no paired adjacent sample based on looking through the "metadata" variable. This sample may end up being removed from the sample data based on this information.

Sample RK106-tumor-XX is a female tumor sample also found in the male tumor cluster. Its paired sample, RK106-adjacent-XX, was found to be located in the male adjacent cluster. Based on this information, these samples possibly have mislabeled sex, however, further investigation is needed.

Sample RK113-adjacent-XY is a male adjacent sample located in the male tumor cluster. Its paired sample is a male tumor sample that is located in the proper male tumor cluster. Based on this information it is thought that some tumor sample contaminated the normal sample. This sample may be removed based on this information.

Sample RK169-adjacent-XY is a male adjacent sample located in the male tumor cluster. Its paired sample is a male tumor sample located in the male adjacent cluster. Based on this information, these samples possibly have mislabeled lesion type, however, further investigation is needed.

Sample RK105-adjacent-XY is a male adjacent sample located in the female tumor cluster. Its paired sample is a male tumor sample located in the female adjacent cluster. Based on this information, these samples possibly have mislabeled sex , however, further investigation is needed.

Sample RK065-tumor-XX is a female tumor sample found in the female adjacent cluster. Its paired sample is a female adjacent sample that is located in the female tumor cluster. Based on this information, these samples possibly have mislabeled lesion type, however, further investigation is needed.

Sample RK116-tumor-XX is a female tumor sample that is found in the female adjacent cluster. Its paired sample is a female adjacent that is located in the proper female adjacent cluster. Based on this information, it is possible that the female tumor sample was contaminated by the normal sample. However, more investigation is needed.

Sample RK066-adjacent-XX is a female adjacent sample that is found in the female tumor cluster. Its paired sample is a female tumor sample that is located in the proper female tumor cluster. Based on this information, it is possible that the female normal sample was contaminated by the tumor sample. However, more investigation is needed.

Sample RK096-tumor-XY is a male tumor sample found in the male adjacent cluster. The sample has no paired normal tissue sample based on looking through the “metadata” variable.This sample may end up being removed from the sample data based on this information.