```
---
title: "201-107 Statistical Analysis"
output:
  html_document: default
  pdf_document: default
---

```{r setup, echo=TRUE, include=FALSE}
knitr::opts_chunk$set("echo" = FALSE)
knitr::opts_chunk$set("tidy" = TRUE)
knitr::opts_chunk$set("eval" = TRUE)
knitr::opts_chunk$set("warning" = FALSE)
knitr::opts_chunk$set("message" = FALSE)
knitr::opts_chunk$set("cache" = FALSE) #settings for pdf/doc/html generation

deps = c("AUCRF","pROC", "vegan", "dplyr", 'knitr', "gridExtra", "grid", "xtable", "devtools", "pgirmess", "scales", "tidyr",
"ggplot2", "Hmisc", "cowplot", "rmarkdown", "knitcitations", "ggpubr", "car", "ggsignif", "formatR", "tinytex");
for (dep in deps){
        if (dep %in% installed.packages()[,"Package"] == FALSE){
                install.packages(as.character(dep), quiet=TRUE);
        }
        library(dep, verbose=FALSE, character.only=TRUE)
} #loads packages used for analysis and rendering

```

```{r data, echo=TRUE}
set.seed(123) #sets seed start for consistant results
longformdata <- read.table("201-107_DATAANALYSIS_longform_Rreadable.txt", header = T, sep = "\t") #calls data
data <- subset(longformdata, select = c("TxSamp", "Tx", "Matrix", "Time_hr", "analyte", "Conc_nM")) #removes unneeded data fields
for analysis
data$analyte <- ordered(data$analyte, levels = c("Prodrug", "CDV", "PP")) #orders levlels of analyte Pro>CDV>CDVPP for figure
generation
levels(data$analyte) <-  c("Prodrug", "CDV", "CDV-PP") #renames the levels
data$Tx <- ordered(data$Tx, levels = c("NPP669", "NPP666", "NPP663", "CDV", "CMX001", "USC505")) #orders the test articles for
figure generation
data$SampleType <- gsub(".*_(T\\d+_\\w+)\\d+_\\w+","\\1", data$TxSamp) #pulls info from TxSamp column to indicate time collected
and if sample is dose stability (ds), from medium (med), or from lysate (lys) and creates a new column "SampleType"

lysate_mL <- 0.333 #volume of lysate in mL
media_mL <- 10 #volume of media in mL

#the following funtion converts nanomolar to nanomoles
Nanomol <- function(x){
  nmol <- ifelse(data$Matrix=="media", (x/1000*media_mL), (x/1000*lysate_mL))
return(nmol)
}

data$Nanomol <- Nanomol(data$Conc_nM) #utilizes above function to convert concentration to raw nanomoles for each sample in a new
column

data$Picomol <- data$Nanomol*1000 #converts nanomoles to picomoles for each sample in a new column

summary(data) #shows structure of data and provides breakdowns by column
data

lysate <- data[data$Matrix=="lysate",] #selects all data with lysate as the matrix to determine analyte per cells


cellcountsdf <- read.table("201-107_rawcellcounts.txt", header = T, sep="\t") #calls in the cell count data

#the following function coverts the raw hempocytometer counts to cells per flask
cells_per_flask <- function(rawcells){
  cellsperflask <- rawcells/4*2*10000*5
  return(cellsperflask)
}

cellcountsdf$cellsperflask <- cells_per_flask(cellcountsdf$rawcells) #uses above function
avecellcounts <- as.data.frame(cellcountsdf%>%group_by(Tx)%>%summarise(meancells = mean(cellsperflask))) #calculates the mean cell
count by treatment group as a new dataframe

summary(cellcountsdf) #summary of cell count table


avecellcounts

lysate <- merge(lysate, avecellcounts, by="Tx") #combines the lysate data with average cell count data by group
lysate$pmolPERmillCell <- lysate$Picomol/(lysate$meancells/1000000) #calculates picomole per 10^6 cells for each sample

summary(lysate) #summary of lysate table
head(lysate) #shows beginning of dataframe

avepmolpcell <- lysate %>% group_by(Tx, analyte)%>%summarise(meanper = mean(pmolPERmillCell), sd=(sd(pmolPERmillCell))) #calculates
average picomoles per 10^6 cells by treatment group

avepmolpcell
```

```{r anova, echo=TRUE}
lys_pro <- lysate[lysate$analyte=="Prodrug",] #isolates prodrug samples in lysate for statistical analysis
lys_cdv <- lysate[lysate$analyte=="CDV",] #isolates CDV samples in lysate for statistical analysis
lys_pp <- lysate[lysate$analyte=="CDV-PP",] #isolates CDV-PP samples in lysate for statistical analysis

set.seed(123)
```
```

```
norm_pp <- leveneTest(pmolPERmillCell~Tx, lys_pp) #test data for normal distribution
aov_pp <- aov(pmolPERmillCell~Tx, lys_pp) #perform anova comparing picomoles per 10^6 cells by treatment group
summary(aov_pp) #anova results
mct_pp <- TukeyHSD(aov_pp) #performs Tukey post-hoc test for multiple comparisions and adjusts p-values accordingly, same for the
below anovas
mct_pp

set.seed(123)
norm_cdv <- leveneTest(pmolPERmillCell~Tx, lys_cdv)
aov_cdv <- aov(pmolPERmillCell~Tx, lys_cdv)
summary(aov_cdv)
mct_cdv <- TukeyHSD(aov_cdv)
mct_cdv

set.seed(123)
norm_pro <- leveneTest(pmolPERmillCell~Tx, lys_pro)
aov_pro <- aov(pmolPERmillCell~Tx, lys_pro)
summary(aov_pro)
mct_pro <- TukeyHSD(aov_pro)

mct_pro

#figure building
stats_labs <- data.frame(
  Tx = factor(rep(c("NPP669","NPP666","NPP663","CDV","CMX001","USC505"), 3)),
  analyte = factor(c(rep("Prodrug", 6), rep("CDV",6), rep("CDV-PP", 6))),
  labs =
factor(c("bcf","ace","abe","sdgsna","bcf","ae","bcdef","acdef","ab","abef","abd","abd","bcdef","acdef","abd","abcef","abdf","abde"))

) # labels columns for statistical comparisons from Tukey tests

ann_text <- data.frame(
  Tx = factor(c("NPP669","NPP666","NPP663","CDV","CMX001","USC505")),
  labs = factor(c("NPP-669\n(a)","NPP-666\n(b)","NPP-663\n(c)","CDV\n(d)","CMX-001\n(e)","USC-505\n(f)"))
) #annotates the column labels with letter for statistical comparisons

noCDVpro <- data.frame(
  Tx = "CDV",
  analyte = "Prodrug",
  y=20,
  lab = "italic(na)") #labels CDV-Prodrug as "na" since there isnt a prodrug

labels <- merge(stats_labs, avepmolpcell, by=c("Tx","analyte"), all=T) #merges label datafram with avepmolcell dataframe for
calling figure labels
```

```{r picopermillcell_fig}
set.seed(123)

library(RColorBrewer) #calls color pallette package for figure
myColors <- brewer.pal(3,"Pastel1") #selects 3 complementary colors from "Pastel1" color pallette

names(myColors) <- levels(lysate$analyte) #assigns a color to an analyte

colScale <- scale_fill_manual(name = "analyte",values = myColors) #instructs ggplot how to color columns

p <- ggplot(lysate, aes(Tx, pmolPERmillCell, fill=analyte))+stat_summary(geom = "bar", fun.y = mean, position = "dodge",
color="black")+stat_summary(geom = "errorbar", fun.ymin = function(x) mean(x) - sd(x), fun.ymax = function(x) mean(x) + sd(x),
position = "dodge", width=0.5)+scale_x_discrete(breaks=ann_text$Tx, labels=ann_text$labs)+geom_text(labels, mapping = aes(Tx,y=
(meanper+sd+20),label=labs))+scale_y_continuous(limits= c(0,425),expand = c(0,0))+facet_grid(analyte~.,
switch="both")+geom_text(noCDVpro, mapping = aes(x=Tx,y=y, label=lab),parse=T)+scale_fill_brewer(name= "Analyte", breaks=
c("Prodrug", "CDV", "CDV-PP"), labels=c("Prodrug", "CDV", "CDV-
PP"))+colScale+theme_bw()+theme(legend.position="none")+theme(strip.text = element_text(face="bold"))+labs(x="Treatment",
y=bquote(paste("Picomoles Analyte per 10"^"6"," HFF-1 cells"))) #figure building: ggplot calls data and variables to display,
stat_summary selects bar chart display and errorbars using mean + SD, scale_x and geom_text add labels to x axis, scale_y does same
for y axis, facet_grid breaks up figure by analyte and places the facet label on the left, geom_text labels CDV-Prodrug as na,
scale_fill_brewer and colScale colors the columns and legend, theme_bw() instructs how figure background looks, remaining theme()
calls remove the legend bolds the facet labels, titles the x axis and y axis

p #renders figure

```

```{r pdf, include=FALSE}
#creates pdf of figure
pdf("picopercellWITHstats.pdf", width = 8, height = 6)
p
dev.off()
```