```r
# set random seed start for consistant results
set.seed(123)
# call data file
longformdata <- read.table("201-107_DATAANALYSIS_longform_Rreadable.txt",
    header = T, sep = "\t")
# remove unneeded data fields for analysis
data <- subset(longformdata, select = c("TxSamp", "Tx", "Matrix",
    "Time_hr", "analyte", "Conc_nM"))
# order levlels of analyte Pro>CDV>CDVPP for figure
# generation
data$analyte <- ordered(data$analyte, levels = c("Prodrug", "CDV",
    "PP"))
# rename the levels
levels(data$analyte) <- c("Prodrug", "CDV", "CDV-PP")
# order the test articles for figure generation
data$Tx <- ordered(data$Tx, levels = c("NPP669", "NPP666", "NPP663",
    "CDV", "CMX001", "USC505"))
# pull info from TxSamp column to indicate time collected and
# if sample is dose stability (ds), from medium (med), or
# from lysate (lys) and creates a new column 'SampleType'
data$SampleType <- gsub(".*_(T\\d+_\\w+)\\d+_\\w+", "\\1", data$TxSamp)
# volume of lysate in mL
lysate_mL <- 0.333
# volume of media in mL
media_mL <- 10

# the following funtion converts nanomolar to nanomoles
Nanomol <- function(x) {
    nmol <- ifelse(data$Matrix == "media", (x/1000 * media_mL),
        (x/1000 * lysate_mL))
    return(nmol)
}
# use above function to convert concentration to raw
# nanomoles for each sample in a new column
data$Nanomol <- Nanomol(data$Conc_nM)
# convert nanomoles to picomoles for each sample in a new
# column
data$Picomol <- data$Nanomol * 1000

# show structure of data and breakdowns by column
summary(data)
```

```
##              TxSamp         Tx        Matrix      Time_hr
##  CDV_T0_ds1_CDV :  1   NPP669:27   lysate:51   Min.   : 0.00
##  CDV_T0_ds2_CDV :  1   NPP666:27   media :90   1st Qu.:72.00
##  CDV_T0_ds3_CDV :  1   NPP663:27               Median :72.00
##  CDV_T72_ds1_CDV:  1   CDV   :15               Mean   :56.68
##  CDV_T72_ds2_CDV:  1   CMX001:27               3rd Qu.:72.00
##  CDV_T72_ds3_CDV:  1   USC505:18               Max.   :72.00
##  (Other)        :135
##     analyte      Conc_nM        SampleType           Nanomol
##  Prodrug:51   Min.   :   0.0   Length:141         Min.   : 0.000
##  CDV    :72   1st Qu.:   0.0   Class :character   1st Qu.: 0.000
##  CDV-PP :18   Median : 387.6   Mode  :character   Median : 0.152
```

```
##                Mean   : 558.6                     Mean   : 3.104
##                3rd Qu.:1068.8                     3rd Qu.: 5.802
##                Max.   :1784.5                     Max.   :13.463
##
##     Picomol
##  Min.   :    0
##  1st Qu.:    0
##  Median :  152
##  Mean   : 3104
##  3rd Qu.: 5802
##  Max.   :13463
##
```

```
# select all data with lysate as the matrix to determine
# analyte per cells
lysate <- data[data$Matrix == "lysate", ]
# call in the cell count data
cellcountsdf <- read.table("201-107_rawcellcounts.txt", header = T,
    sep = "\t")

# 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)
}
# use above function to add cells per flask column to cell
# count dataframe
cellcountsdf$cellsperflask <- cells_per_flask(cellcountsdf$rawcells)
# calculate the mean cell count by treatment group as a new
# dataframe
avecellcounts <- as.data.frame(cellcountsdf %>% group_by(Tx) %>%
    summarise(meancells = mean(cellsperflask)))
# summary of cell count table
summary(cellcountsdf)
```

```
##       Tx       rawcells      cellsperflask
##  CDV   :3   Min.   : 50.00   Min.   :1250000
##  CMX001:3   1st Qu.: 69.00   1st Qu.:1725000
##  DMSO  :3   Median : 75.00   Median :1875000
##  NPP663:3   Mean   : 81.24   Mean   :2030952
##  NPP666:3   3rd Qu.: 95.00   3rd Qu.:2375000
##  NPP669:3   Max.   :120.00   Max.   :3000000
##  USC505:3
```

```
# average counts by group
avecellcounts
```

```
##       Tx meancells
## 1    CDV   2583333
## 2 CMX001   1575000
## 3   DMSO   2375000
## 4 NPP663   2075000
## 5 NPP666   2250000
## 6 NPP669   1566667
## 7 USC505   1791667
```

```r
# combine the lysate data with average cell count data by
# group
lysate <- merge(lysate, avecellcounts, by = "Tx")
# calculate picomole per 10^6 cells for each sample
lysate$pmolPERmillCell <- lysate$Picomol/(lysate$meancells/1e+06)

# summary of lysate table
summary(lysate)
```

```
##       Tx                 TxSamp      Matrix      Time_hr      analyte
##  NPP669:9   CDV_T72_lys1_CDV: 1   lysate:51   Min.   :72   Prodrug:15
##  NPP666:9   CDV_T72_lys1_PP : 1   media : 0   1st Qu.:72   CDV    :18
##  NPP663:9   CDV_T72_lys2_CDV: 1               Median :72   CDV-PP :18
##  CDV   :6   CDV_T72_lys2_PP : 1               Mean   :72
##  CMX001:9   CDV_T72_lys3_CDV: 1               3rd Qu.:72
##  USC505:9   CDV_T72_lys3_PP : 1               Max.   :72
##             (Other)         :45
##     Conc_nM        SampleType           Nanomol           Picomol
##  Min.   :   0.0   Length:51          Min.   :0.00000   Min.   :  0.00
##  1st Qu.: 282.0   Class :character   1st Qu.:0.09391   1st Qu.: 93.91
##  Median : 449.5   Mode  :character   Median :0.14968   Median :149.68
##  Mean   : 709.8                      Mean   :0.23636   Mean   :236.36
##  3rd Qu.:1319.8                      3rd Qu.:0.43949   3rd Qu.:439.49
##  Max.   :1784.5                      Max.   :0.59424   Max.   :594.24
##
##    meancells       pmolPERmillCell
##  Min.   :1566667   Min.   :  0.00
##  1st Qu.:1575000   1st Qu.: 53.35
##  Median :1791667   Median : 84.83
##  Mean   :1937745   Mean   :133.69
##  3rd Qu.:2250000   3rd Qu.:204.68
##  Max.   :2583333   Max.   :379.30
##
```

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

```
## # A tibble: 17 x 4
## # Groups:   Tx [?]
##    Tx     analyte meanper     sd
##    <ord>  <ord>     <dbl>  <dbl>
##  1 NPP669 Prodrug 314.    57.2
##  2 NPP669 CDV     325.    35.9
##  3 NPP669 CDV-PP  288.    13.9
##  4 NPP666 Prodrug 132.     6.66
##  5 NPP666 CDV     129.    22.3
##  6 NPP666 CDV-PP  203.     3.13
##  7 NPP663 Prodrug  47.9    6.25
##  8 NPP663 CDV      37.1    5.84
##  9 NPP663 CDV-PP   65.4    3.36
```

```
## 10 CDV    CDV       0.434  0.752
## 11 CDV    CDV-PP    0      0
## 12 CMX001 Prodrug 336.     4.09
## 13 CMX001 CDV      69.5   18.0
## 14 CMX001 CDV-PP   61.4    2.06
## 15 USC505 Prodrug 113.    15.0
## 16 USC505 CDV      69.3   16.8
## 17 USC505 CDV-PP   80.9    2.75
```

```
# isolate prodrug samples in lysate for statistical analysis
lys_pro <- lysate[lysate$analyte == "Prodrug", ]
# isolate CDV samples in lysate for statistical analysis
lys_cdv <- lysate[lysate$analyte == "CDV", ]
# isolate CDV-PP samples in lysate for statistical analysis
lys_pp <- lysate[lysate$analyte == "CDV-PP", ]

set.seed(123)
# verify normal distribution
norm_pp <- leveneTest(pmolPERmillCell ~ Tx, lys_pp)
# perform anova comparing picomoles per 10^6 cells by
# treatment group
aov_pp <- aov(pmolPERmillCell ~ Tx, lys_pp)
# anova results table
summary(aov_pp)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## Tx           5 171988   34398   917.1 4.62e-15 ***
## Residuals   12    450      38
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# performs Tukey post-hoc test for multiple comparisions and
# adjusts p-values accordingly, same for the below anovas
mct_pp <- TukeyHSD(aov_pp)
# display multiple comparison results
mct_pp
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = pmolPERmillCell ~ Tx, data = lys_pp)
##
## $Tx
##                       diff         lwr        upr     p adj
## NPP666-NPP669  -84.752410 -101.548804  -67.95602 0.0000000
## NPP663-NPP669 -222.411192 -239.207586 -205.61480 0.0000000
## CDV-NPP669    -287.818277 -304.614670 -271.02188 0.0000000
## CMX001-NPP669 -226.433515 -243.229908 -209.63712 0.0000000
## USC505-NPP669 -206.888435 -223.684829 -190.09204 0.0000000
## NPP663-NPP666 -137.658782 -154.455176 -120.86239 0.0000000
## CDV-NPP666    -203.065867 -219.862260 -186.26947 0.0000000
## CMX001-NPP666 -141.681105 -158.477499 -124.88471 0.0000000
## USC505-NPP666 -122.136025 -138.932419 -105.33963 0.0000000
## CDV-NPP663     -65.407084  -82.203478  -48.61069 0.0000002
## CMX001-NPP663   -4.022322  -20.818716   12.77407 0.9612412
```

```
## USC505-NPP663   15.522758   -1.273636   32.31915 0.0763444
## CMX001-CDV      61.384762   44.588368   78.18116 0.0000004
## USC505-CDV      80.929842   64.133448   97.72624 0.0000000
## USC505-CMX001   19.545080    2.748686   36.34147 0.0197853
```

```
# anova steps same as above but for CDV and Prodrug
set.seed(123)
norm_cdv <- leveneTest(pmolPERmillCell ~ Tx, lys_cdv)
aov_cdv <- aov(pmolPERmillCell ~ Tx, lys_cdv)
summary(aov_cdv)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## Tx           5 201185   40237    99.5 2.43e-09 ***
## Residuals   12   4853     404
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
mct_cdv <- TukeyHSD(aov_cdv)
mct_cdv
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = pmolPERmillCell ~ Tx, data = lys_cdv)
##
## $Tx
##                         diff        lwr         upr     p adj
## NPP666-NPP669 -195.9618667 -251.11258 -140.811156 0.0000006
## NPP663-NPP669 -287.9259759 -343.07669 -232.775265 0.0000000
## CDV-NPP669    -324.5740258 -379.72474 -269.423315 0.0000000
## CMX001-NPP669 -255.4620952 -310.61281 -200.311385 0.0000000
## USC505-NPP669 -255.7316093 -310.88232 -200.580899 0.0000000
## NPP663-NPP666  -91.9641092 -147.11482  -36.813399 0.0012511
## CDV-NPP666    -128.6121591 -183.76287  -73.461448 0.0000535
## CMX001-NPP666  -59.5002286 -114.65094   -4.349518 0.0320161
## USC505-NPP666  -59.7697426 -114.92045   -4.619032 0.0311408
## CDV-NPP663     -36.6480499  -91.79876   18.502661 0.2917495
## CMX001-NPP663   32.4638807  -22.68683   87.614591 0.4067818
## USC505-NPP663   32.1943666  -22.95634   87.345077 0.4150204
## CMX001-CDV      69.1119306   13.96122  124.262641 0.0119223
## USC505-CDV      68.8424165   13.69171  123.993127 0.0122555
## USC505-CMX001   -0.2695141  -55.42022   54.881197 1.0000000
```

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

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## Tx           4 198439   49610   68.88 3.07e-07 ***
## Residuals   10   7202     720
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
mct_pro <- TukeyHSD(aov_pro)
mct_pro
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = pmolPERmillCell ~ Tx, data = lys_pro)
##
## $Tx
##                       diff         lwr        upr     p adj
## NPP666-NPP669 -181.66717 -253.783338 -109.55099 0.0000656
## NPP663-NPP669 -266.20031 -338.316480 -194.08414 0.0000020
## CMX001-NPP669   22.04638  -50.069795   94.16255 0.8469444
## USC505-NPP669 -200.83179 -272.947961 -128.71562 0.0000270
## NPP663-NPP666  -84.53314 -156.649314  -12.41697 0.0207850
## CMX001-NPP666  203.71354  131.597371  275.82971 0.0000238
## USC505-NPP666  -19.16462  -91.280795   52.95155 0.8998891
## CMX001-NPP663  288.24669  216.130513  360.36286 0.0000010
## USC505-NPP663   65.36852   -6.747653  137.48469 0.0804850
## USC505-CMX001 -222.87817 -294.994338 -150.76199 0.0000105
```

```
# figure building label columns for statistical comparisons
# from Tukey tests
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")))

# annotate the column labels with letter for statistical
# comparisons
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)")))

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

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

```
set.seed(123)
# call color pallette package for figure
library(RColorBrewer)
# select 3 complementary colors for 3 analytes from 'Pastel1'
# color pallette
myColors <- brewer.pal(3, "Pastel1")
# assign a color to an analyte
names(myColors) <- levels(lysate$analyte)
# instruct ggplot how to color columns
colScale <- scale_fill_manual(name = "analyte", values = myColors)

# figure building: ggplot calls data and variables to
```

```r
# 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, 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 <- 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) + 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")))
```