Supplementary Materials
Xiaohua Douglas Zhang and Zhaozhi Zhang

1. How to load the package displayHTS

Our package has been built and checked successfully by using “Rcmd INSTALL --build”, “Rcmd build” and “Rcmd check”. The package can be obtained through CRAN. Here we provide the instruction for obtaining the package in Rgui for Windows users as follows.

- Open a Rgui window as follows

- Click “Package” on the top line of the Rgui window and choose “set CRAN mirror…” from the pull-down menu
- From the pop-out window, choose and double-click the mirror site that you want to download a package
- Click “Package” in Rgui again and choose “Install package(s)” from the pull-down menu
- From the pop-out window named “Packages”, scroll down, find and double-click “displayHTS”. The package “displayHTS” will be installed to your computer
Type `library(displayHTS)` in Rgui to load the package in R

```r
> library(displayHTS)
```

**2. Codes for generating Figure 1 in the manuscript**

Run the R codes below to see how to generate Figure 1 in the manuscript through using the functions in the package. Note, the functions in our package can be applied to any input data format as long as the format can allow the data to be read in R as a data frame containing the required columns and information for the functions.

```r
# fig1.A: plate-well series plot
data(HTSdataSort)
wells = as.character(unique(HTSdataSort[, "WELL_USAGE"]))
colors = c("black", "pink", "grey", "blue", "skyblue", "green", "red")
```
```r
orders=c(1, 3, 2, 4, 5, 7, 6)
par( mfrow=c(1,1) )
plateWellSeries.fn(data.df = HTSdataSort[1:(384*2),], intensityName="log2Intensity",
  plateName="BARCODE", wellName="WELL_USAGE",
  rowName="XPOS", colName="YPOS", show.wellTypes=wells,
  order.wellTypes=orders, color.wells=colors, pch.wells=rep(1, 7), ppf=6, byRow=TRUE,
  yRange=NULL, cex.point=0.75,cex.legend=0.75,
  main="A: Plate-well series plot")

# fig1.B: hit and control image
data(HTSresults)
condtSample = HTSresults[, "WELL_USAGE"] == "Sample"
condtUp = HTSresults[, "ssmd"] >= 1 & HTSresults[, "mean"] >= log2(1.2)
condtDown = HTSresults[, "ssmd"] <= -1 & HTSresults[, "mean"] <= -log2(1.2)
sum(condtSample & (condtUp | condtDown)) / sum(condtSample)

hit.vec = as.character(HTSresults[, "WELL_USAGE"])
hit.vec[condtSample & condtUp] = "up-hit"
hit.vec[condtSample & condtDown] = "down-hit"
result.df = cbind(HTSresults, "hitResult"=hit.vec)
wells = as.character(unique(result.df[, "hitResult"])); wells
colors = c("black", "green", "white", "red", "grey", "purple1", "purple2", "pink", "purple3")
par( mfrow=c(1,1) )
imageDesign.fn(result.df[1:384,], wellName="hitResult", rowName="XPOS",
  colName="YPOS", wells=wells, colors=colors,
  title="B: Image of hits and controls")

## fig1.C: image of intensity in a plate
imageIntensity.fn(HTSdataSort[1:384,], intensityName="log2Intensity",
  plateName="BARCODE", wellName="WELL_USAGE",
  rowName="XPOS", colName="YPOS", sampleName="Sample",
  sourcePlateName="SOBARCODE")
#title("C: Image of intensity in a plate")

## fig1.D: dual-flashlight plot
par( mfrow=c(1, 1) )
dualFlashlight.fn(HTSresults, wellName="WELL_USAGE", x.name="mean",
  y.name="ssmd", sampleName="Sample", sampleColor="black",
  controls = c("negCTRL", "posCTRL1", "mock1"),
  controlColors = c("green", "red", "lightblue"),
  xlab="Average Fold Change", ylab="SSMD",
  main="C: Dual-Flashlight Plot", x.legend=0.1, y.legend=-12,
  cex.point=1, cex.legend = 0.8, xat=log2( c(1/4, 1/2, 1/1.2, 1, 1.2, 2, 4) ),
  xMark=c("1/4", "1/2", "1/1.2", "1", "1.2", "2", "4"),
  xLines=log2( c(1/4, 1/2, 1/1.2, 1, 1.2, 2, 4) ),
  yLines=c(-5, -3, -2, -1, 0, 1, 2, 3, 5 )
)

## fig1.E: volcano plot
```
result.df = cbind(HTSresults, "neg.log10.pval" = -log10(HTSresults[, "p.value" ]))
dualFlashlight.fn(result.df, wellName="WELL_USAGE", x.name="mean",
    y.name="neg.log10.pval", sampleName="Sample", sampleColor="black",
    controls=c("negCTRL", "posCTRL1", "mock1"),
    controlColors=c("green", "red", "lightblue"),
    xlab="Average Fold Change", ylab="p-value in -log10 scale",
    main="D: Volcano Plot", x.legend=NA, y.legend=-log10(0.006),
    cex.point=1, cex.legend=0.8, xat=log2( c(1/4, 1/2, 1/1.2, 1, 1.2, 2, 4 ) ),
    xMark=c("1/4", "1/2", "1/1.2", "1", "1.2", "2", "4"),
    xLines=log2( c(1/4, 1/2, 1/1.2, 1, 1.2, 2, 4 ) ),
    yLines=c(-5, -3, -2, -1, 0, 1, 2, 3, 5 ) )