```
#Adapted from 'mcr' package: https://CRAN.R-project.org/package=mcr (Last accessed Oct 13, 2016)
calcDiff <- function(X,Y,EPS=1E-12) {
  dRes <- X-Y
  dRes[abs(dRes)<EPS*(abs(X)+abs(Y))/2] <- 0
  return(dRes)
}
#Adapted from 'mcr' package: https://CRAN.R-project.org/package=mcr (Last accessed Oct 13, 2016)
deming<-function(X, Y, error.ratio)
{
  # Check validity of parameters

  stopifnot(!is.na(X))
  stopifnot(!is.na(Y))
  stopifnot(is.numeric(X))
  stopifnot(is.numeric(Y))
  stopifnot(length(X)==length(Y))
  stopifnot(length(X) > 0)
  stopifnot(!is.na(error.ratio))
  stopifnot(is.numeric(error.ratio))
  stopifnot(error.ratio>0)
  stopifnot(length(error.ratio) > 0)

  #--
  n <- length(X)
  mX <- mean(X)
```

```
 mY <- mean(Y)
 u <- sum((X-mX)^2)
 q <- sum((Y-mY)^2)
 p <- sum((X-mX)*(Y-mY))
 r <- p/sqrt(u*q)

 ## Estimated points
 # [ Ref. K.Linnet. Estimation of the linear relationship between
 #       the measurements of two methods with  Proportional errors.
 #       STATISTICS IN MEDICINE, VOL. 9, 1463-1473 (1990)].
 #

 b1 <- ((error.ratio*q-u)+sqrt((u-error.ratio*q)^2+4*error.ratio*p^2))/(2*error.ratio*p)
 b0 <- mean(Y)-b1*mean(X)

 ## Standard error
 # [Ref. Strike, P. W. (1991) Statistical Methods in Laboratory Medicine.
 #      Butterworth-Heinemann, Oxford ].

 se.b1 <- sqrt(b1^2*(calcDiff(1,r^2)/r^2)/(n-2))
 se.b0 <- sqrt(se.b1^2*mean(X^2))

 return(list(b0=b0, b1=b1, se.b0=se.b0, se.b1=se.b1, xw=mX,  weight=rep(1,length(X))))
}
```

```
# Adapted from:
# Bland-Altman plot R function.
# Author: jmmateos@mce.hggm.es

###BA plot function - assay comparison###
baplot <- function(m1, m2, l1, l2, title,...) {
  # Assign values to "m1" and "m2" to compare
  means <- (m1 + m2) / 2
  diffs <- m1 - m2
  mdiff <- mean(diffs)
  sddiff <- sd(diffs)
  #Compute the figure limits
  ylimh <- mdiff + 3 * sddiff
  yliml <- mdiff - 3 * sddiff
  #Plot data
  plot(diffs ~ means, xlab = "Average signal", pch=16, col="grey",
      #main=paste0("Bland-Altman analysis of ",l1," and ",l2),
      ylab = paste0("Signal differences (",l1," - ",l2,")"),
      ylim = c(yliml, ylimh), cex.main=1, cex.lab=1
      ,main=title
      #,xaxt='n',
      #yaxt='n'
  )
  #diplay 0 bias horizontal line
```

```
  abline(h=0)

  #display calculated bias line

  abline(h = mdiff, col="blue", lwd=2) # Center line

  abline(lm(diffs~means), col="red", lwd=3, lty=3)

  # Plot standard deviations lines

  abline(h = mdiff + 1.96 * sddiff, lty = 2, lwd=2)

  abline(h = mdiff - 1.96 * sddiff, lty = 2, lwd=2)

  #display degree of agreement

  text(0.25*max(means), ylimh*0.85, label=paste0("Degree of agreement:\n",abs(signif(sddiff*1.98, 3))," units"), cex=1, font=2)

  # plotting bias p value

  pval <- t.test(diffs, mu=0)$p.value

  # strip to 3 significant digits

  pval <- signif(pval, 3)

  # add asterisk if <0.05

  pval <- ifelse(pval<0.001, "< 0.001",

              ifelse(pval<0.01, "< 0.01",

                   ifelse(pval<0.05, "< 0.05","> 0.05")))

  # annotate it near lower left corner

  text(0.55*max(means), ylimh*0.85,paste0("Constant error:\n",abs(signif(mdiff,3))," units \n(blue line)\np ", pval), cex=1, font=2)

  # systematic error

  systematic<-coef(summary.lm(lm(diffs~means)))[2,1]

  pval<- coef(summary.lm(lm(diffs~means)))[2,4]

  pval <- ifelse(pval<0.001, "< 0.001",

              ifelse(pval<0.01, "< 0.01",
```

```
                    ifelse(pval<0.05, "< 0.05","> 0.05")))
  pval <- paste0("Proportional error:\n",abs(signif(systematic, 3))," units\n(red line)\np ", pval)
  text(0.85*max(means), ylimh*0.85,label=pval, cex=1, font=2)
}

##########################################################################################
#Assay comparison script using Deming reg.

assay.metrics <- function(assay1, assay2, name1, name2, title, ...) {

  lm1<-deming(assay1,assay2,error.ratio=1)
  se.b1<-lm1$se.b1
  slope<-lm1$b1
  b1<-1-lm1$b1
  b0<-lm1$b0


  plot(assay1, assay2, pch=16, col="grey", xlab=name1, ylab=name2, main=title, cex.main=1, cex.lab=1,
      xlim=c(min(assay1,assay2),max(assay1,assay2)),
      ylim=c(min(assay1,assay2),max(assay1,assay2)))
  #add assay1 line
  abline(0,1,col="black", lwd=2)
  #add assay2 line
  abline(a=b0,b=slope, col="red", lwd=3, lty=3)
  #Dem. reg.
```

```
 #display p value for the accuracy metric
 pval<-t.test(assay2, assay1, paired=TRUE)$p.value
 text(max(assay1)*0.35,ifelse(max(assay2)>max(assay1),max(assay2), max(assay1))*0.9,
     paste0("Constant error: ", abs(signif(mean(assay2)-mean(assay1),3)), " units (p ",
           ifelse(pval<0.001, "< 0.001",
                 ifelse(pval<0.01, "< 0.01",
                       ifelse(pval<0.05, "< 0.05","> 0.05"))), ")"), cex=1, font=2)
 #systematic error (possibly not needed)
 tstats <- (1-b1/se.b1)
 # Calculates two tailed probability
 pval<- 2 * pt(abs(tstats), df = length(x)-2, lower.tail = FALSE)

 text(max(assay1)*0.35,ifelse(max(assay2)>max(assay1),max(assay2), max(assay1))*0.8,
     paste0("Proportional error: ", signif(b1, 3), " units (p ",
           ifelse(pval<0.001, "< 0.001",
                 ifelse(pval<0.01, "< 0.01",
                       ifelse(pval<0.05, "< 0.05","> 0.05"))), ")"), cex=1, font=2)

 text(max(assay1)*0.35,ifelse(max(assay2)>max(assay1),max(assay2), max(assay1))*0.7,
     paste0("Deming: y = ", signif(1-b1, 3), "x ",ifelse(b0<0, "- ",
                                         ifelse(b0>0, "+ ")), signif(abs(b0), 3)), cex=1, font=2)

 }

#######PRECISION AND ACCURACY FOR SPIKE-AND-RECOVERY
EXPERIMENTS######################
```

```
# Adapted from:
# Bland-Altman plot R function.
# Author: jmmateos@mce.hggm.es

baplot <- function(m1, m2, l1, l2, title, ...) {
 # Assign values to "m1" and "m2" to compare
 means <- (m1 + m2) / 2
 diffs <- m1 - m2
 mdiff <- mean(diffs)
 sddiff <- sd(diffs)
 #Compute the figure limits
 ylimh <- mdiff + 3 * sddiff
 yliml <- mdiff - 3 * sddiff
 #Plot data
 plot(diffs ~ means, xlab = "Average signal", pch=16, col="grey", cex.main=1, cex.lab=1,
     #main=paste0("Bland-Altman analysis of ",l1," and ",l2),
     ylab = paste0("Signal differences (",l1," - ",l2,")"),
     ylim = c(yliml, ylimh), cex.main=1, cex.lab=1, main=title
     #,xaxt='n',
     #yaxt='n'
 )
 #diplay 0 bias horizontal line
 abline(h=0)
 #display calculated bias line
```

```
  abline(h = mdiff, col="blue", lwd=2) # Center line

  abline(lm(diffs~means), col="red", lwd=3, lty=3)

  # Plot standard deviations lines

  abline(h = mdiff + 1.96 * sddiff, lty = 2, lwd=2)

  abline(h = mdiff - 1.96 * sddiff, lty = 2, lwd=2)

  #display degree of agreement

  text(0.25*max(means), ylimh*0.85, label=paste0("Precision:\n",abs(signif(sddiff*1.98, 3))," units"),
cex=1, font=2)

  # plotting bias p value

  pval <- t.test(diffs, mu=0)$p.value

  # strip to 3 significant digits

  pval <- signif(pval, 3)

  # add asterisk if <0.05

  pval <- ifelse(pval<0.001, "< 0.001",

              ifelse(pval<0.01, "< 0.01",

                  ifelse(pval<0.05, "< 0.05","> 0.05")))

  # annotate it near lower left corner

  text(0.55*max(means), ylimh*0.85,paste0("Constant error:\n",abs(signif(mdiff,3))," units \n(blue
line)\np ", pval), cex=1, font=2)

  # systematic error

  systematic<-coef(summary.lm(lm(diffs~means)))[2,1]

  pval<- coef(summary.lm(lm(diffs~means)))[2,4]

  pval <- ifelse(pval<0.001, "< 0.001",

              ifelse(pval<0.01, "< 0.01",

                  ifelse(pval<0.05, "< 0.05","> 0.05")))

  pval <- paste0("Proportional error:\n",abs(signif(systematic, 3))," units\n(red line)\np ", pval)
```

```
  text(0.85*max(means), ylimh*0.85,label=pval, cex=1, font=2)
}

##############################################################################################

#Linear reg function; display accuracy and precision


assay.metrics <- function(expected, observed, title, ...) {

  lm1<-lm(observed ~ expected)
  lm2<-lm(observed ~ expected, offset=1*expected)
  #calculate prediction intervals
  newx <- seq(ifelse(min(observed)>min(expected),min(observed), min(expected)),
              ifelse(max(observed)>max(expected),max(observed), max(expected)), 0.1)
  b <- predict(lm1, newdata=data.frame(expected=newx), interval="prediction")
  #plot the regression
  plot(expected, observed, pch=16, col="grey", xlab="Expected", ylab="Observed", main=title,
cex.main=1, cex.lab=1,
       xlim=c(min(expected,observed),max(expected,observed)),
       ylim=c(min(expected,observed),max(expected,observed)))
  #add expected line
  abline(0,1,col="black", lwd=2)
  #add observed line
  abline(lm1, col="red", lwd=3, lty=3)
  #add upper and lower prediction interval lines
```

```
lines(newx,b[,2], lty=3, lwd=3, col="black")
lines(newx,b[,3], lty=3, lwd=3, col="black")
#calculate smallest and largest prediction intervals within reportable range:
small.int<-(min(b[,3]-b[,2]))/2
large.int<-(max(b[,3]-b[,2]))/2
#display precision as a range between smallest and largest prediction interval values

text(max(expected)*0.35,
    ifelse(max(observed)>max(expected),max(observed), max(expected)),
    paste0("Precision: ", signif(small.int, 3)," - ", signif(large.int, 3)," units"),
    cex=1, font=2)
#display p value for the accuracy metric
pval<-t.test(observed, expected, paired=TRUE)$p.value
text(max(expected)*0.35,ifelse(max(observed)>max(expected),max(observed), max(expected))*0.9,
    paste0("Constant error: ", abs(signif(mean(observed)-mean(expected),3)), " units (p ",
        ifelse(pval<0.001, "< 0.001",
            ifelse(pval<0.01, "< 0.01",
                ifelse(pval<0.05, "< 0.05","> 0.05"))), ")"), cex=1, font=2)
#systematic error (possibly not needed)
pval<-coef(summary.lm(lm2))[2,4]
systematic<-coef(summary.lm(lm1))[2,1]
y.int<-coef(summary.lm(lm1))[1,1]
text(max(expected)*0.35,ifelse(max(observed)>max(expected),max(observed), max(expected))*0.8,
    paste0("Proportional error: ", abs(signif(coef(summary.lm(lm2))[2,1], 3)), " units (p ",
        ifelse(pval<0.001, "< 0.001",
```

```
                ifelse(pval<0.01, "< 0.01",
                    ifelse(pval<0.05, "< 0.05","> 0.05"))), ")"), cex=1, font=2)
  #display equation of the line
  text(max(assay1)*0.35,ifelse(max(assay2)>max(assay1),max(assay2), max(assay1))*0.7,
      paste0("SLR: y = ", signif(systematic, 3), "x ",ifelse(y.int<0, "- ",
                                          ifelse(y.int>0, "+ ")), signif(abs(y.int), 3)), cex=1, font=2)
}
```