Chromosomal Instability Confers Intrinsic Multidrug Resistance
Supplementary Document 1 - Data Analysis

Lee et al.

December 21, 2010

This document describes the data analysis.
All analysis was performed in the R statistical environment.

1 Getting started

1.1 Required Packages

```
> library(limma)
> library(corpcor)
> library(grofit)
> library(gdata)
> library(gregmisc)
> library(multtest)
```

1.2 A few functions

```
> fun.hochberg = function(pvalue) {
+     help <- cbind(pvalue, 1:length(pvalue))
+     help <- help[order(help[, 1]), ]
+     help <- cbind(help, 1:nrow(help))
+     pk <- (help[, 3]/nrow(help)) * 0.05
+     help <- cbind(help, help[, 1] <= pk)
+     help <- help[order(help[, 2]), ]
+     return(help)
+ }
> function.segment <- function(x, chrom, pos) {
+     end <- c()
+     start <- c()
+     chrom0 <- c()
+     num.prob <- c()
+     val <- c()
+     chrom <- as.numeric(chrom)
+     for (i in 1:22) {
+         w.chr <- which(chrom == i)
+         sub.pos <- pos[w.chr]
+         sub.x <- x[w.chr]
+         sub.chr <- chrom[w.chr]
+         unique.x <- unique(sub.x)
+         for (k in unique.x) {
+             wu <- which(sub.x == k)
+             if (length(wu) == 1) {
+                 start <- c(start, sub.pos[wu])
+                 end <- c(end, sub.pos[wu])
+                 num.prob <- c(num.prob, 1)
+                 chrom0 <- c(chrom0, i)
+                 val <- c(val, sub.x[wu])
+             }
+             else {
+                 start <- c(start, sub.pos[wu[1]])
+                 val <- c(val, sub.x[wu[1]])
+                 chrom0 <- c(chrom0, sub.chr[wu[1]])
+                 wu2 <- wu - c(wu[1] - 1, wu[1:(length(wu) - 1)])
+                 find.start <- wu[wu2 != 1]
+                 find.end <- wu[which(wu2 != 1) - 1]
+                 num.prob <- c(num.prob, c(find.end, wu[length(wu)]) -
+                   c(wu[1], find.start) + 1)
+                 start <- c(start, sub.pos[find.start])
```

```
+                 end <- c(end, sub.pos[find.end])
+                 val <- c(val, sub.x[find.start])
+                 chrom0 <- c(chrom0, sub.chr[find.start])
+                 end <- c(end, sub.pos[wu[length(wu)]])
+             }
+         }
+     }
+     ret <- cbind(chrom0, start, end, num.prob, val)
+     ret <- ret[order(ret[, 1], partial = ret[, 2]), ]
+     return(ret)
+ }
> gii.fun = function(seg.mat) {
+     gii <- c()
+     bp <- c()
+     for (i in unique(seg.mat[, 1])) {
+         s <- subset(seg.mat, seg.mat[, 1] == i)
+         med <- weighted.median(as.numeric(s[, 6]), w = as.numeric(s[,
+             4]) - as.numeric(s[, 3]))
+         b <- sum(as.numeric(s[, 4]) - as.numeric(s[, 3]))
+         a <- s[as.numeric(s[, 6]) > med | as.numeric(s[, 6] <
+             med), ]
+         gii <- c(gii, sum(as.numeric(a[, 4]) - as.numeric(a[,
+             3]))/b)
+         bp <- c(bp, nrow(a))
+     }
+     gii2 <- cbind(gii, bp)
+     rownames(gii2) <- unique(seg.mat[, 1])
+     return(gii2)
+ }
> no.tel <- function(seg.out, chrominfo, min.size = 1e+06, min.probes = 1,
+     max.size = 1e+09) {
+     if (class(seg.out) == "DNAcopy") {
+         seg.out <- seg.out$output
+     }
+     tmp.segs <- seg.out
+     tmp.segs <- tmp.segs[!as.numeric(tmp.segs[, 5]) < min.probes,
+         ]
+     tmp.segs[, 2] <- as.character(tmp.segs[, 2])
+     tmp.segs[as.numeric(tmp.segs[, 6]) == 1, 6] <- 2
+     for (j in 1:length(unique(tmp.segs[, 1]))) {
+         tmp.sample <- tmp.segs[tmp.segs[, 1] == unique(tmp.segs[,
+             1])[j], ]
+         for (i in 1:22) {
+             tmp1 <- tmp.sample[as.numeric(tmp.sample[, 2]) ==
+                 i, , drop = F]
+             if (as.numeric(tmp1[1, 6]) == 2 & nrow(tmp1) != 1 &
+                 as.numeric(tmp1[1, 4]) < (chrominfo[i, 3] * 1000)) {
+                 tmp.sample[as.numeric(tmp.sample[, 2]) == i,
+                   6][1] <- 1
+             }
+             if (as.numeric(tmp1[nrow(tmp1), 6]) == 2 & nrow(tmp1) !=
+                 1 & as.numeric(tmp1[nrow(tmp1), 3]) > (chrominfo[i,
+                 3] * 1000)) {
+                 tmp.sample[as.numeric(tmp.sample[, 2]) == i,
```

```
+                   6][nrow(tmp.sample[as.numeric(tmp.sample[,
+                   2]) == i, ])] <- 1
+             }
+             if (is.null(dim(tmp.sample[as.numeric(tmp.sample[,
+                 2]) == i, ])) & as.numeric(tmp.sample[as.numeric(tmp.sample[,
+                 2]) == i, ][6]) != 0) {
+                 tmp.sample[as.numeric(tmp.sample[, 2]) == i,
+                   6][1] <- 3
+             }
+         }
+         tmp.segs[tmp.segs[, 1] == unique(tmp.segs[, 1])[j], ] <- tmp.sample
+     }
+     no.events <- matrix(0, nrow = length(unique(tmp.segs[, 1])),
+         ncol = 28)
+     rownames(no.events) <- unique(tmp.segs[, 1])
+     colnames(no.events) <- c("Telomeric AI", "Mean size", "Interstitial AI",
+         "Mean Size", "Wholo chr AI", 1:23)
+     for (i in unique(tmp.segs[, 1])) {
+         tmp <- tmp.segs[tmp.segs[, 1] == i, ]
+         tmp <- tmp[(as.numeric(tmp[, 4]) - as.numeric(tmp[, 3])) >
+             min.size, ]
+         tmp <- tmp[(as.numeric(tmp[, 4]) - as.numeric(tmp[, 3])) <
+             max.size, ]
+         if (!is.null(dim(tmp[as.numeric(tmp[, 6]) == 1, ])))
+             no.events[i, 1] <- nrow(tmp[as.numeric(tmp[, 6]) ==
+                 1, ])
+         else no.events[i, 1] <- 1
+         no.events[i, 2] <- mean(as.numeric(tmp[as.numeric(tmp[,
+             6]) == 1, 4]) - as.numeric(tmp[as.numeric(tmp[, 6]) ==
+             1, 3]))
+         if (!is.null(dim(tmp[as.numeric(tmp[, 6]) == 2, ])))
+             no.events[i, 3] <- nrow(tmp[as.numeric(tmp[, 6]) ==
+                 2, ])
+         else no.events[i, 3] <- 1
+         no.events[i, 4] <- mean(as.numeric(tmp[as.numeric(tmp[,
+             6]) == 2, 4]) - as.numeric(tmp[as.numeric(tmp[, 6]) ==
+             2, 3]))
+         if (!is.null(dim(tmp[as.numeric(tmp[, 6]) == 3, ])))
+             no.events[i, 5] <- nrow(tmp[as.numeric(tmp[, 6]) ==
+                 3, ])
+         else no.events[i, 5] <- 1
+         no.events[i, tmp[as.numeric(tmp[, 6]) == 3, 2]] <- 1
+     }
+     return(no.events)
+ }
> removedrugs <- function(x, thresh = 0.8) {
+     if (length(which(x > thresh)) == length(which(!is.na(x))))
+         return(TRUE)
+     else return(FALSE)
+ }
> fun.probesMinimalDistance <- function(time5.MUT, time5.MUT2,
+     time5.WT, time5.WT2, t = 0.8) {
+     min.mat = matrix(NA, nr = nrow(time5.MUT), nc = ncol(time5.MUT))
+     help.mat1 = matrix(NA, nr = nrow(time5.MUT), nc = ncol(time5.MUT))
```

```
+     for (i in 1:ncol(time5.MUT)) {
+         for (k in 1:nrow(time5.MUT)) {
+             vect = c(abs(time5.MUT[k, i] - time5.WT[k, i]), abs(time5.MUT[k,
+                 i] - time5.WT2[k, i]), abs(time5.MUT2[k, i] -
+                 time5.WT[k, i]), abs(time5.MUT2[k, i] - time5.WT2[k,
+                 i]))
+             wi = which(vect == min(vect[!is.na(vect)]))
+             if (length(wi) > 1)
+                 wi = wi[1]
+             if (wi == 1) {
+                 min.mat[k, i] = time5.MUT[k, i] - time5.WT[k,
+                   i]
+                 if (time5.MUT[k, i] > t & time5.WT[k, i] > t)
+                   min.mat[k, i] = NA
+             }
+             if (wi == 2) {
+                 min.mat[k, i] = time5.MUT[k, i] - time5.WT2[k,
+                   i]
+                 if (time5.MUT[k, i] > t & time5.WT2[k, i] > t)
+                   min.mat[k, i] = NA
+             }
+             if (wi == 3) {
+                 min.mat[k, i] = time5.MUT2[k, i] - time5.WT[k,
+                   i]
+                 if (time5.MUT2[k, i] > t & time5.WT[k, i] > t)
+                   min.mat[k, i] = NA
+             }
+             if (wi == 4) {
+                 min.mat[k, i] = time5.MUT2[k, i] - time5.WT2[k,
+                   i]
+                 if (time5.MUT2[k, i] > t & time5.WT2[k, i] >
+                   t)
+                   min.mat[k, i] = NA
+             }
+         }
+     }
+     return(min.mat)
+ }
> interaction.test.SW <- function(depend, fact1, fact2, nreps = 1000) {
+     model1 <- lm(depend ~ fact1 * fact2)
+     ANOVA <- summary(aov(model1))
+     Finteract <- ANOVA[[1]]$"F value"[3]
+     model2 <- lm(depend ~ fact1 + fact2)
+     resid <- model2$residuals
+     SHint <- numeric(nreps)
+     SHint[1] <- Finteract
+     for (i in 2:nreps) {
+         newcomb <- sample(resid, length(depend), replace = FALSE)
+         model3 <- summary(aov(newcomb ~ fact1 * fact2))
+         SHint[i] <- model3[[1]]$"F value"[3]
+     }
+     probInt <- length(SHint[SHint >= Finteract])/nreps
+     return(probInt)
+ }
```

1.3 Ploidy status and combined aberration score

```
> load("J:/SensitivityData/dataend2.Rdata")
> load("J:/SensitivityData/LOHdata.RData")
> load("J:/SensitivityData/chrominfo.RData")
> dataend <- dataend[-which(dataend[, 1] == 23 | dataend[, 1] ==
+     24), ]
> tmp.seg <- apply(dataend[, 3:ncol(dataend)], 2, function.segment,
+     chrom = dataend[, 1], pos = dataend[, 2])
> seg.mat <- cbind(rep(names(tmp.seg)[1], nrow(tmp.seg[[1]])),
+     tmp.seg[[1]])
> for (i in 2:length(tmp.seg)) seg.mat <- rbind(seg.mat, cbind(rep(names(tmp.seg)[i],
+     nrow(tmp.seg[[i]])), tmp.seg[[i]]))
> seg.mat[, 1] <- substr(seg.mat[, 1], 7, nchar(seg.mat[, 1]) -
+     4)
> gii.med <- gii.fun(seg.mat)
> LOH.seg <- apply(LOH[, 3:ncol(LOH)], 2, function.segment, chrom = LOH[,
+     1], pos = LOH[, 2])
> seg.mat.LOH <- cbind(rep(names(LOH.seg)[1], nrow(LOH.seg[[1]])),
+     LOH.seg[[1]])
> for (i in 2:length(LOH.seg)) seg.mat.LOH <- rbind(seg.mat.LOH,
+     cbind(rep(names(LOH.seg)[i], nrow(LOH.seg[[i]])), LOH.seg[[i]]))
> seg.mat.LOH[, 1] <- substr(seg.mat.LOH[, 1], 7, nchar(seg.mat.LOH[,
+     1]) - 4)
> tel <- no.tel(seg.mat.LOH, chrominfo)
> vecthelp <- c()
> for (i in unique(seg.mat[, 1])) {
+     sub <- subset(seg.mat, seg.mat[, 1] == i)
+     vecthelp <- c(vecthelp, weighted.mean(as.numeric(sub[, "val"]),
+         w = as.numeric(sub[, "end"]) - as.numeric(sub[, "start"])))
+ }
> names(vecthelp) <- unique(seg.mat[, 1])
> comb.ab <- gii.med[, 1]/max(gii.med[, 1]) + gii.med[, 2]/max(gii.med[,
+     2]) + (tel[, 1] + tel[, 3])/max(tel[, 1] + tel[, 3])
> comb.ab <- comb.ab/max(comb.ab)
> MIN <- c(4:7, 13, 18, 26)
> modal <- c(78, NA, 54, 46, NA, 45, 46, NA, 72, NA, 77, NA, 45,
+     NA, 102, NA, 66, 46, 75, NA, NA, 63, 70, NA, 68, 47, 50,
+     67, 56)
> vecthelp2 <- vecthelp[c(1, 3, 4, 6, 7, 9, 11, 13, 15, 17, 18,
+     19, 22, 23, 24, 25, 26, 27, 28, 29)]
> print(cor(vecthelp, modal/23, use = "complete.obs"))

[1] 0.9403694

> print(cor.test(vecthelp, modal/23, use = "complete.obs"))

        Pearson's product-moment correlation

data:  vecthelp and modal/23
t = 11.3985, df = 17, p-value = 2.201e-09
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.8486314 0.9771948
sample estimates:
```

```
      cor
0.9403694
```

Figure 1 A

```
> ba <- barplot(sort(vecthelp2)/2, las = 3, ylab = "Weighted Mean PICNIC Copy Numbers",
+     cex.axis = 0.7, cex.lab = 0.7, cex.main = 0.7)
> abline(h = 1.1, col = "red")
> axis(1, ba[1:6], names(sort(vecthelp2))[1:6], las = 3, col.axis = "red",
+     tick = F)
```

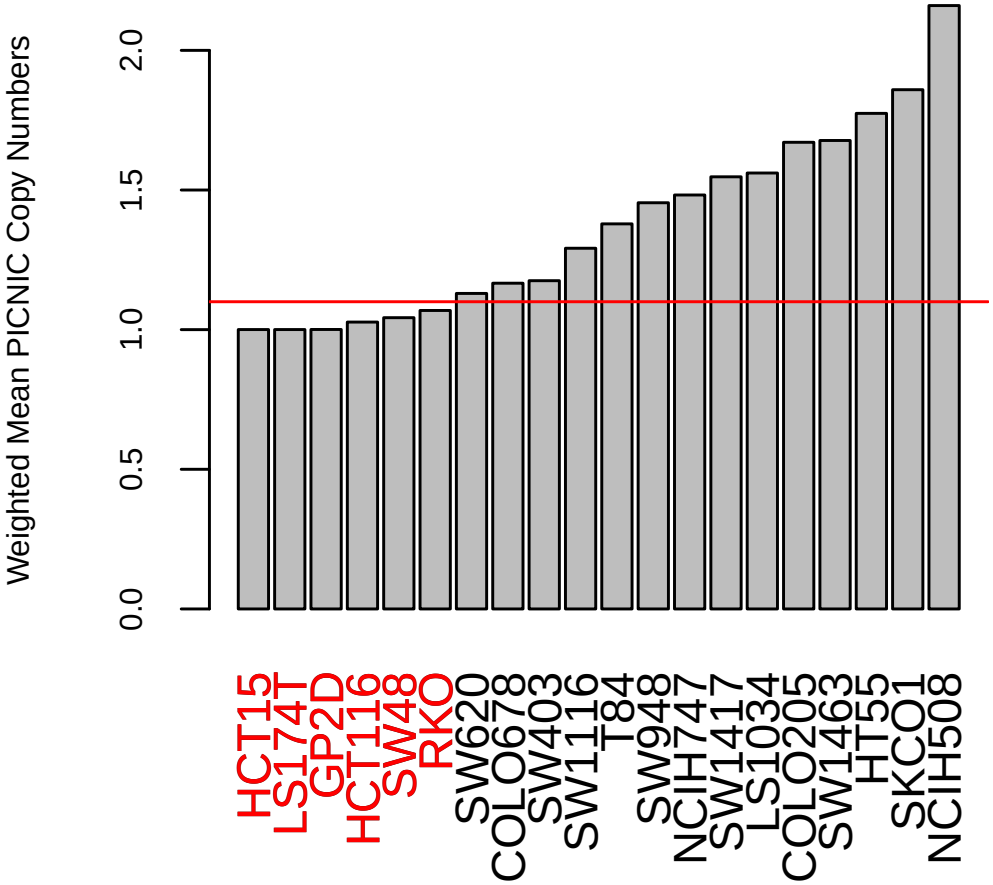
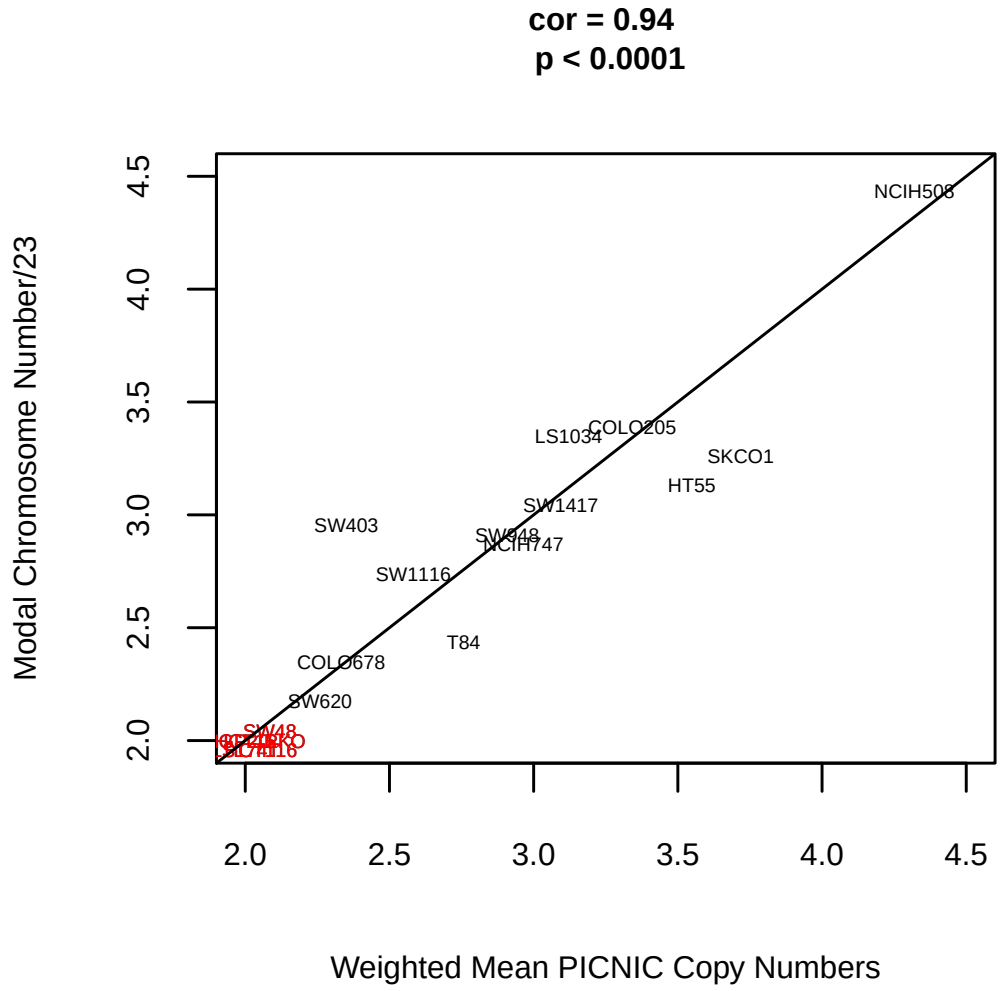


Figure 1 B

```
> plot(vecthelp, modal/23, xlim = c(2, 4.5), ylim = c(2, 4.5),
+     ylab = "Modal Chromosome Number/23", xlab = "Weighted Mean PICNIC Copy Numbers",
+     type = "n", main = "cor = 0.94 \n p < 0.0001", cex.axis = 0.7,
+     cex.lab = 0.7, cex.main = 0.7)
> text(vecthelp, modal/23, names(comb.ab), ylim = c(2, 4.5), cex = 0.4)
> text(vecthelp[MIN], (modal/23)[MIN], names(vecthelp)[c(4:7, 13,
```

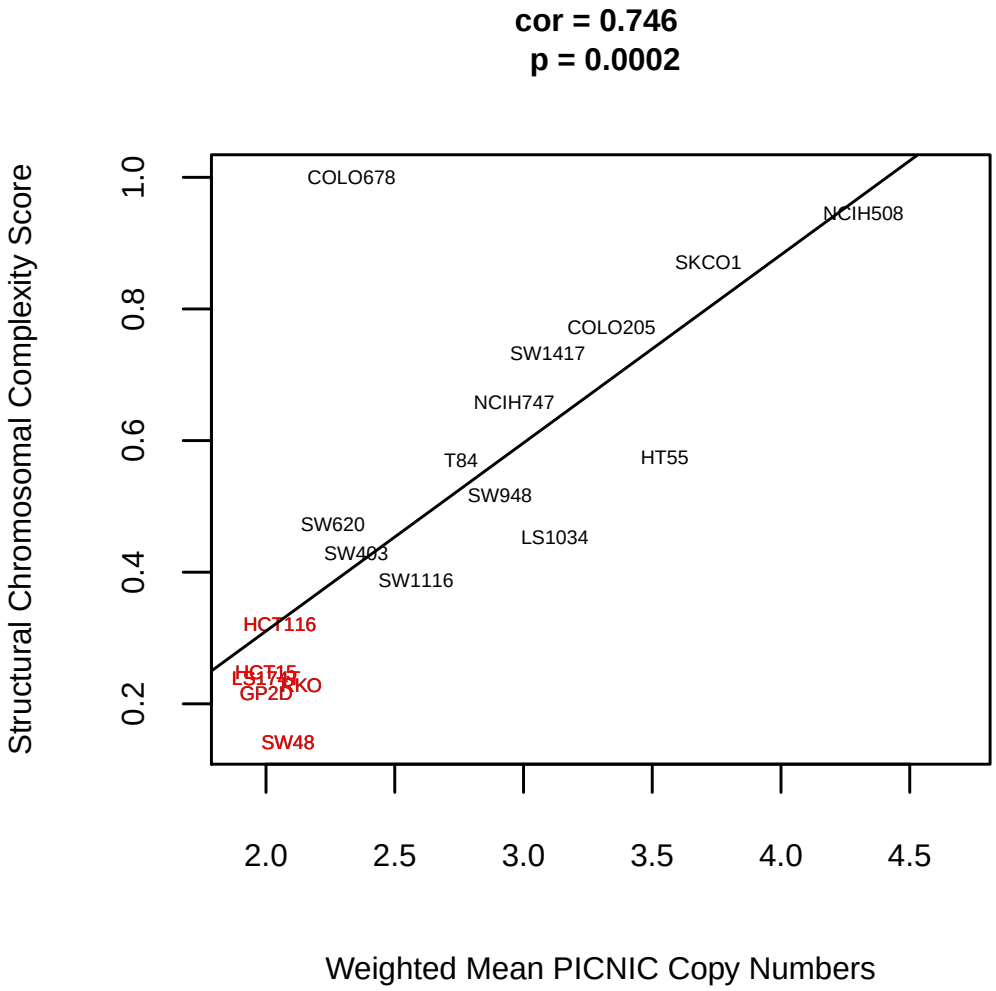
```
+     18, 26)], ylim = c(2, 4.5), cex = 0.4, col = "red")
> abline(c(0, 1))
```




```
> print(cor(vecthelp[!is.na(modal)], comb.ab[!is.na(modal)]))

[1] 0.74702

> print(cor.test(vecthelp[!is.na(modal)], comb.ab[!is.na(modal)]))

        Pearson's product-moment correlation

data:  vecthelp[!is.na(modal)] and comb.ab[!is.na(modal)]
t = 4.633, df = 17, p-value = 0.0002377
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.4431849 0.8969060
sample estimates:
    cor
0.74702
```

Figure 1 C

```
> plot(vecthelp[!is.na(modal)], comb.ab[!is.na(modal)], xlab = "Weighted Mean PICNIC Copy Numbers
+     ylab = "Structural Chromosomal Complexity Score", type = "n",
+     xlim = c(1.9, 4.7), main = "cor = 0.746 \n p = 0.0002", cex.axis = 0.7,
+     cex.lab = 0.7, cex.main = 0.7)
> text(vecthelp[!is.na(modal)], comb.ab[!is.na(modal)], names(vecthelp[!is.na(modal)]),
+     cex = 0.4, xlim = c(1.9, 4.7))
> text(vecthelp[!is.na(modal)][c(3:5, 8, 11, 16)], comb.ab[!is.na(modal)][c(3:5,
+     8, 11, 16)], names(vecthelp[!is.na(modal)])[c(3:5, 8, 11,
+     16)], cex = 0.4, xlim = c(1.9, 4.7), col = "red")
> abline(lm(comb.ab[!is.na(modal)] ~ vecthelp[!is.na(modal)])$coefficients)
```



1.4 Analysis of Calbiochem Library results for 18 CIN+ and 9 CIN- cell lines

```
> load("J:/SensitivityData/RObjects/CalbiochemCINvsMIN.RData")
> drugs[drugs > 1.4] <- NA
```

```
> drug.mat = matrix(0, nc = ncol(drugs), nr = 0)
> for (i in seq(3, nrow(drugs), 3)) drug.mat <- rbind(drug.mat,
+     apply(drugs[(i - 2):i, ], 2, mean, na.rm = T))
> rownames(drug.mat) <- unique(rownames(drugs))
> drug.mat10uM <- drug.mat
> stables.sens <- c(4:6, 8:9, 14, 18, 25, 30)
> unstables.sens <- (1:30)[-stables.sens]
> unstables.sens <- unstables.sens[-6]
> unstables.sens <- unstables.sens[-8]
> unstables.sens <- unstables.sens[-4]
> wil.fun <- function(x) return(wilcox.test(x[unstables.sens],
+     x[stables.sens], alternative = "greater", na.rm = T)$p.value)
> fun.findresdrugs <- function(x) return(length(which(x > 0.8))/length(x[!is.na(x)]))
> remove <- apply(drug.mat[c(stables.sens, unstables.sens), ],
+     2, fun.findresdrugs)
> drug.mat <- drug.mat[, -which(remove > 0.75)]
> pvalue <- apply(drug.mat, 2, wil.fun)
> drug.tmp <- drug.mat[c(unstables.sens, stables.sens), ]
> multip <- fun.hochberg(pvalue)

> allunstabs <- as.vector(drug.mat[unstables.sens, ])
> allstabs <- as.vector(drug.mat[stables.sens, ])
> pval <- ks.test(allunstabs, allstabs, alternative = "less")$p.value
> seq.step <- seq(0, max(c(allunstabs, allstabs), na.rm = T) +
+     0.1, 0.01)
> sum.unstab <- 0
> sum.stab <- 0
> vect.unstab <- c()
> vect.stab <- c()
> for (k in 1:length(seq.step)) {
+     sum.unstab <- sum.unstab + length(which(allunstabs >= seq.step[k -
+         1] & allunstabs < seq.step[k]))
+     sum.stab <- sum.stab + length(which(allstabs >= seq.step[k -
+         1] & allstabs < seq.step[k]))
+     vect.unstab <- c(vect.unstab, sum.unstab)
+     vect.stab <- c(vect.stab, sum.stab)
+ }
```

Figure 2 A

```
> mar.old <- par()$mar
> par(mar = c(5.1, 5.1, 4.1, 2.1))
> plot(seq.step, vect.unstab/length(which(!is.na(allunstabs))),
+     col = "blue", type = "l", xlab = "Relative Number of Cells",
+     ylab = "Cumulative Distribution of Relative Cell \n Number after Inhibitor Treatment",
+     main = paste("p < 0.0001"), cex.axis = 0.7, cex.lab = 0.7,
+     cex.main = 0.7)
> points(seq.step, vect.stab/length(which(!is.na(allstabs))), col = "red",
+     type = "l")
> legend("topleft", c("CIN+", "CIN-"), col = c("blue", "red"),
+     lty = c(1, 1))
> par(mar = mar.old)
```

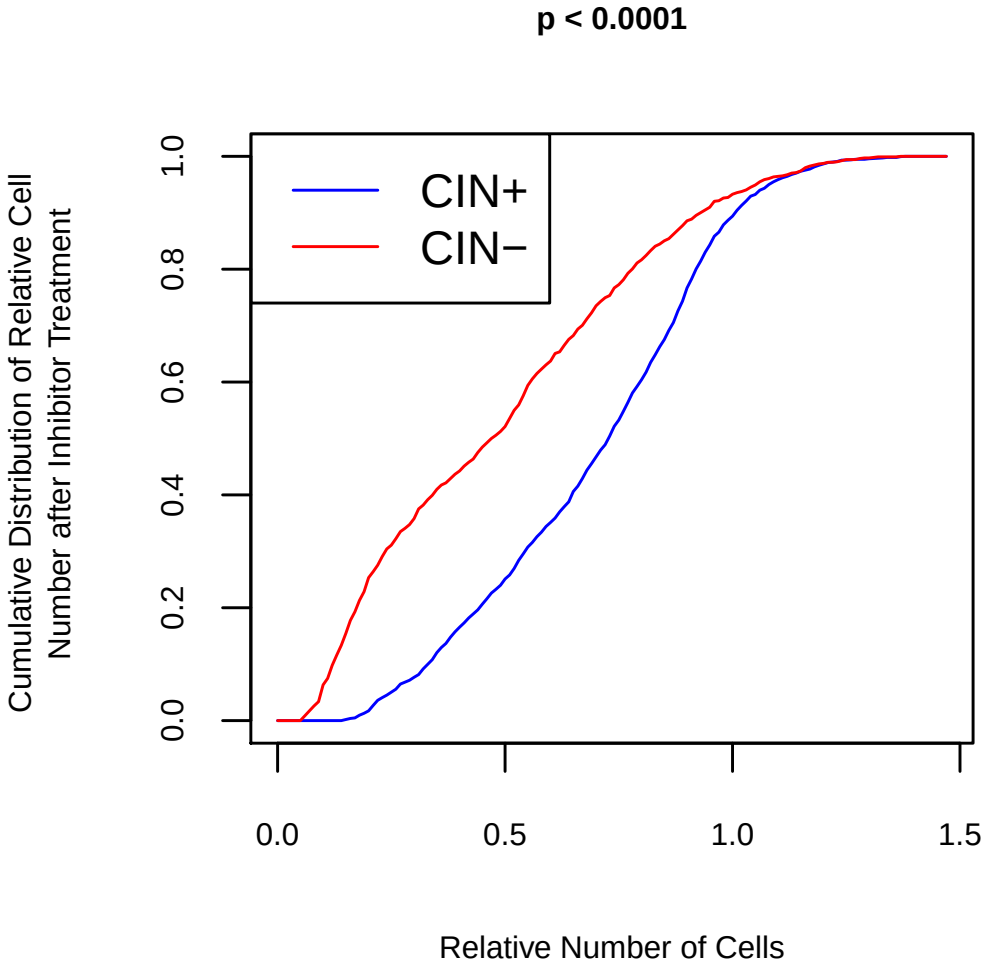
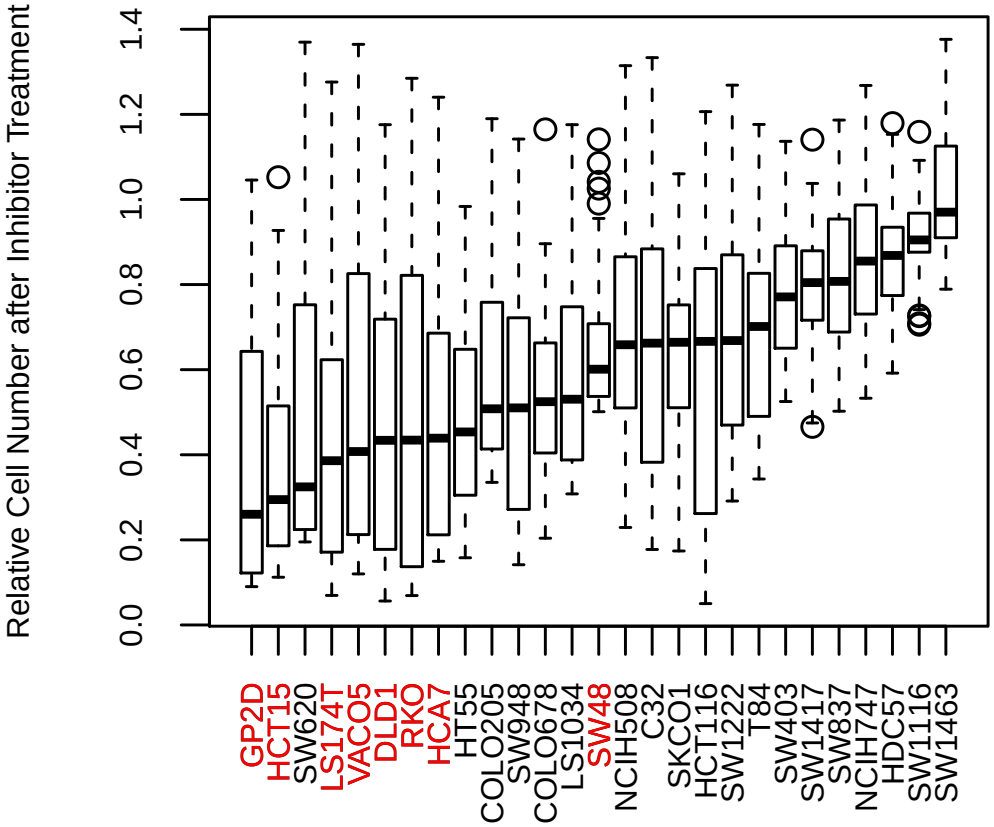


Figure 2 B

```
> MIN.tmp <- c(1, 2, 4, 5, 6, 7, 8, 14)
> names <- rownames(drug.tmp)
> names[c(10, 16, 19, 23, 25)] <- c("COLO205", "C32", "SW1222",
+     "SW837", "HDC57")
> drug.tmp <- drug.tmp[order(apply(drug.tmp, 1, median, na.rm = T)),
+     ]
> boxplot(t(drug.tmp), las = 3, ylab = "Relative Cell Number after Inhibitor Treatment",
+     xaxt = "n", cex.axis = 0.7, cex.lab = 0.7, cex.main = 0.7)
> axis(1, 1:length(names), names, las = 3, cex.axis = 0.7)
> axis(1, MIN.tmp, names[MIN.tmp], las = 3, col.axis = "red", tick = F,
+     cex.axis = 0.7)
```

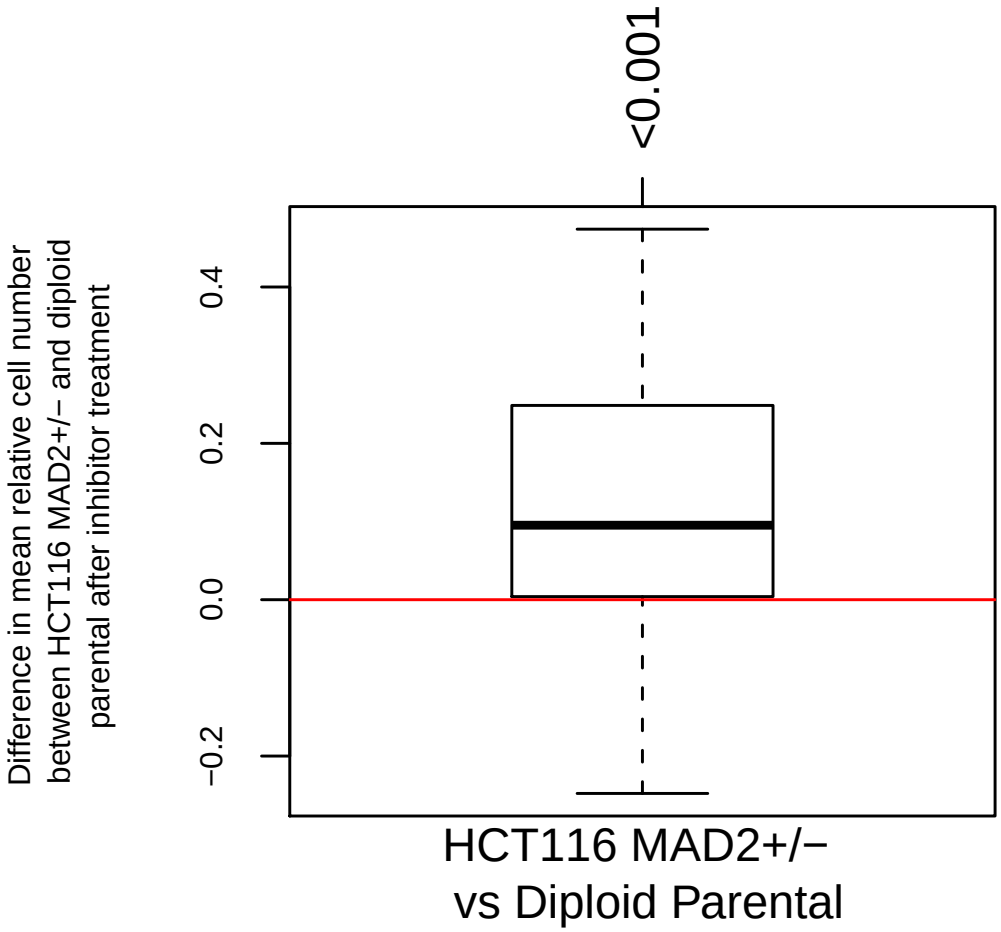


1.5 MAD2 Calbiochem Library

```
> load("J:/SensitivityData/RObjects/CalbiochemMAD2.RData")
> drugs.MAD2.Calbiochem[drugs.MAD2.Calbiochem > 1.4] <- NA
> drug.mat.MAD2.Calbio <- matrix(0, nc = ncol(drugs.MAD2.Calbiochem),
+     nr = 0)
> for (i in seq(3, nrow(drugs.MAD2.Calbiochem), 3)) drug.mat.MAD2.Calbio <- rbind(drug.mat.MAD2.C
+     apply(drugs.MAD2.Calbiochem[(i - 2):i, ], 2, mean, na.rm = T))
> rownames(drug.mat.MAD2.Calbio) <- c("minus", "plus")
> remove <- apply(drug.mat.MAD2.Calbio, 2, removedrugs, thresh = 0.8)
> tmp.test <- drug.mat.MAD2.Calbio[, !remove]
> print(wilcox.test(tmp.test[1, ], tmp.test[2, ], paired = T, alternative = "greater"))

        Wilcoxon signed rank test

data:  tmp.test[1, ] and tmp.test[2, ]
V = 602, p-value = 3.268e-05
alternative hypothesis: true location shift is greater than 0
```

Figure 3 A

```
> ylab <- "Difference in mean relative cell number \n between HCT116 MAD2+/- and diploid \n paren
> mar.old <- par()$mar
> par(mar = c(5.1, 5.4, 4.1, 2.1))
> boxplot(tmp.test[1, ] - tmp.test[2, ], ylab = ylab, cex.axis = 0.7,
+     cex.lab = 0.7, cex.main = 0.7)
> abline(h = 0, col = "red")
> axis(3, 1, "<0.001", las = 3)
> axis(1, 1, "HCT116 MAD2+/- \n vs Diploid Parental", tick = F)
> par(mar = mar.old)
```



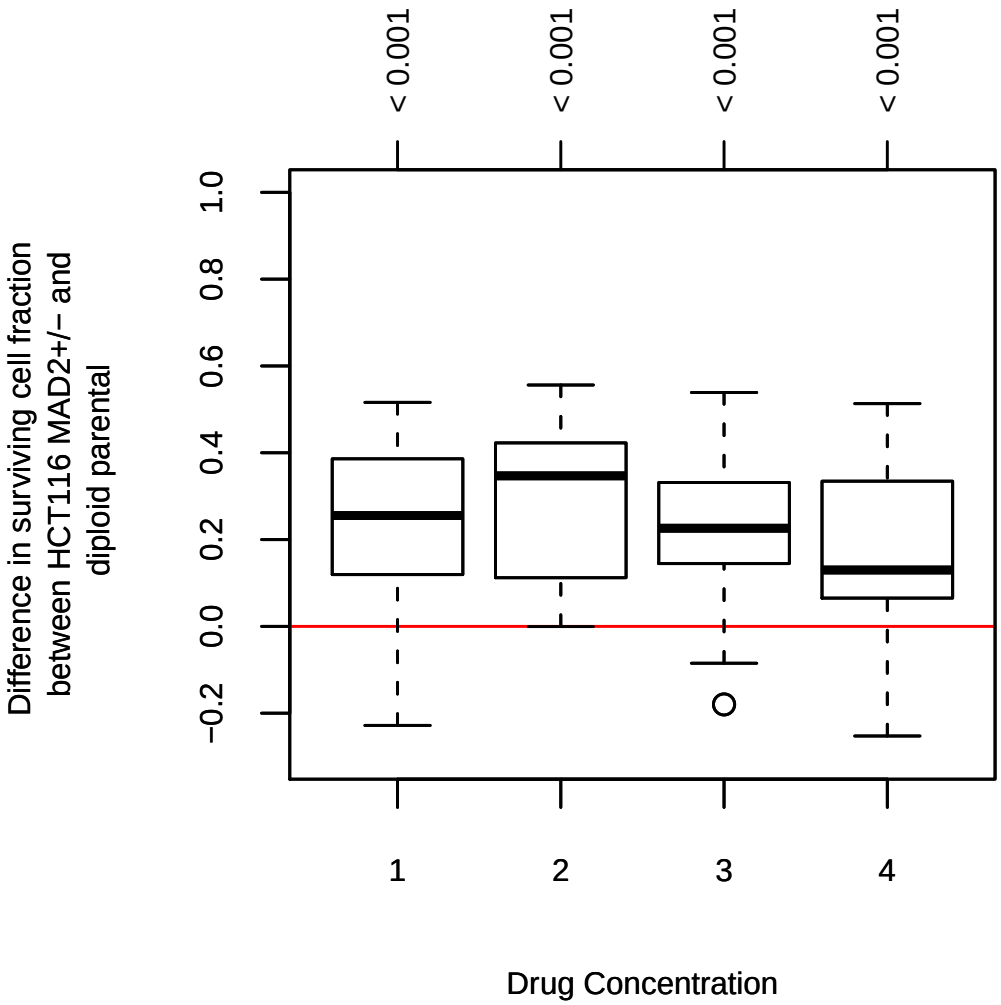
1.6 MAD2 Biolog Library

```
> load("J:/SensitivityData/RObjects/firstDuplicateMUT_MAD2.Rdata")
> load("J:/SensitivityData/RObjects/secondDuplicateMUT_MAD2.Rdata")
> load("J:/SensitivityData/RObjects/firstDuplicateWT_MAD2.Rdata")
> load("J:/SensitivityData/RObjects/secondDuplicateWT_MAD2.Rdata")
```

```
> time5.MUT.MAD2[time5.MUT.MAD2 > 1.4] <- NA
> time5.MUT2.MAD2[time5.MUT2.MAD2 > 1.4] <- NA
> time5.WT.MAD2[time5.WT.MAD2 > 1.4] <- NA
> time5.WT2.MAD2[time5.WT2.MAD2 > 1.4] <- NA
> min.mat.MAD2 <- fun.probesMinimalDistance(time5.MUT = time5.MUT.MAD2,
+     time5.MUT2 = time5.MUT2.MAD2, time5.WT = time5.WT.MAD2, time5.WT2 = time5.WT2.MAD2,
+     t = 0.8)
> wil.fun <- function(x) return(wilcox.test(x, alternative = "greater")$p.value)
> wilp <- apply(min.mat.MAD2, 1, wil.fun)
```

Figure 3 C1

```
> ylab <- "Difference in surviving cell fraction \n between HCT116 MAD2+/- and \n diploid parenta
> mar.old <- par()$mar
> par(mar = c(5.1, 5.4, 4.1, 2.1))
> boxplot(as.vector(t(min.mat.MAD2)) ~ c(rep(1, ncol(min.mat.MAD2)),
+     rep(2, ncol(min.mat.MAD2)), rep(3, ncol(min.mat.MAD2)), rep(4,
+         ncol(min.mat.MAD2))), xlab = "Drug Concentration", ylab = ylab,
+     ylim = c(-0.3, 1), cex.axis = 0.7, cex.lab = 0.7, cex.main = 0.7)
> axis(3, at = 1:4, c("< 0.001", "< 0.001", "< 0.001", "< 0.001"),
+     las = 3, cex.axis = 0.7)
> abline(h = 0, col = "red")
> boxplot(as.vector(t(min.mat.MAD2)) ~ c(rep(1, ncol(min.mat.MAD2)),
+     rep(2, ncol(min.mat.MAD2)), rep(3, ncol(min.mat.MAD2)), rep(4,
+         ncol(min.mat.MAD2))), xlab = "Drug Concentration", ylab = ylab,
+     ylim = c(-0.3, 1), add = T, cex.axis = 0.7, cex.lab = 0.7,
+     cex.main = 0.7)
> par(mar = mar.old)
```

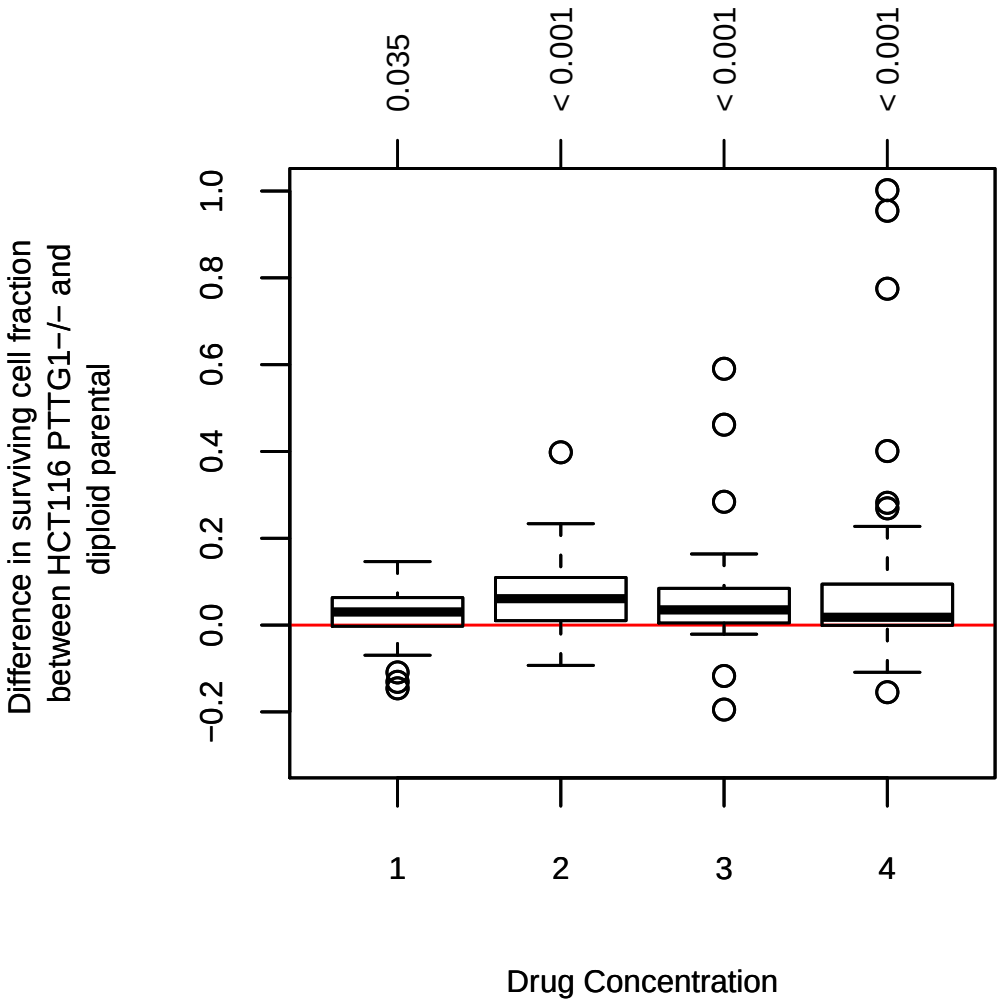


1.7 Securin Biolog Library

```
> load("J:/SensitivityData/RObjects/firstDuplicateMUT_Securin.Rdata")
> load("J:/SensitivityData/RObjects/secondDuplicateMUT_Securin.Rdata")
> load("J:/SensitivityData/RObjects/firstDuplicateWT_Securin.Rdata")
> load("J:/SensitivityData/RObjects/secondDuplicateWT_Securin.Rdata")
> time5.MUT.Securin[time5.MUT.Securin > 1.4] = NA
> time5.MUT2.Securin[time5.MUT2.Securin > 1.4] = NA
> time5.WT.Securin[time5.WT.Securin > 1.4] = NA
> time5.WT2.Securin[time5.WT2.Securin > 1.4] = NA
> min.mat.Securin <- fun.probesMinimalDistance(time5.MUT = time5.MUT.Securin,
+     time5.MUT2 = time5.MUT2.Securin, time5.WT = time5.WT.Securin,
+     time5.WT2 = time5.WT2.Securin, t = 0.8)
> wil.fun <- function(x) return(wilcox.test(x, alternative = "greater")$p.value)
> wilp <- apply(min.mat.Securin, 1, wil.fun)
```

Figure 3 C2

```
> mar.old <- par()$mar
> par(mar = c(5.1, 5.4, 4.1, 2.1))
> ylab <- "Difference in surviving cell fraction \n between HCT116 PTTG1-/- and \n diploid parent
> boxplot(as.vector(t(min.mat.Securin)) ~ c(rep(1, ncol(min.mat.Securin)),
+     rep(2, ncol(min.mat.Securin)), rep(3, ncol(min.mat.Securin)),
+     rep(4, ncol(min.mat.Securin))), xlab = "Drug Concentration",
+     ylab = ylab, ylim = c(-0.3, 1), cex.axis = 0.7, cex.lab = 0.7,
+     cex.main = 0.7)
> axis(3, at = 1:4, c((round(wilp * 1000)/1000)[1], "< 0.001",
+     "< 0.001", "< 0.001"), las = 3, cex.axis = 0.7)
> abline(h = 0, col = "red")
> boxplot(as.vector(t(min.mat.Securin)) ~ c(rep(1, ncol(min.mat.Securin)),
+     rep(2, ncol(min.mat.Securin)), rep(3, ncol(min.mat.Securin)),
+     rep(4, ncol(min.mat.Securin))), xlab = "Drug Concentration",
+     ylab = ylab, ylim = c(-0.3, 1), add = T, cex.axis = 0.7,
+     cex.lab = 0.7, cex.main = 0.7)
> par(mar = mar.old)
```

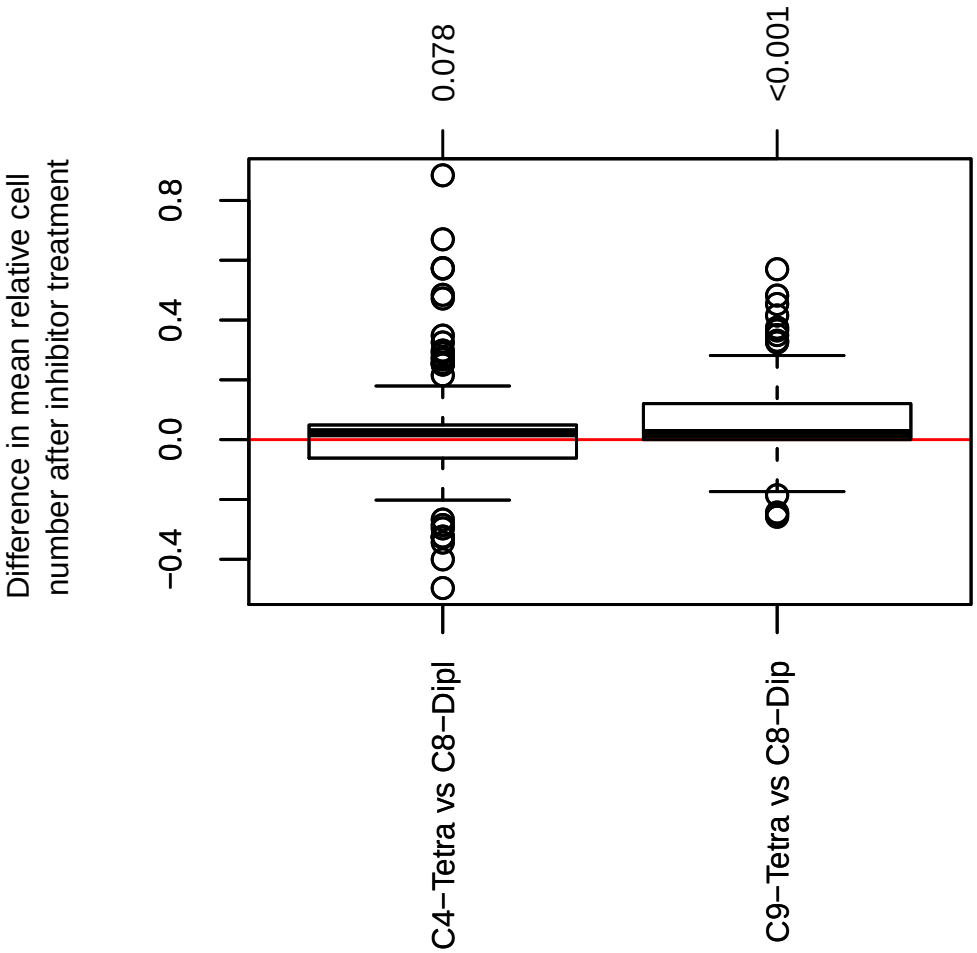


1.8 Parental, Diploid, Tetraploid Calbiochem

```
> load("J:/SensitivityData/RObjects/ParentalTetraDiplCalbiochem.Rdata")
> drug.ParDiplTetra[drug.ParDiplTetra > 1.4] <- NA
> drug.mat.ParDiplTetra <- matrix(0, nc = ncol(drug.ParDiplTetra),
+     nr = 0)
> for (i in seq(3, nrow(drug.ParDiplTetra), 3)) drug.mat.ParDiplTetra <- rbind(drug.mat.ParDiplTe
+     apply(drug.ParDiplTetra[(i - 2):i, ], 2, mean, na.rm = T))
> rownames(drug.mat.ParDiplTetra) <- c("C4 Tetraploid", "C8 Diploid",
+     "C9 Tetraploid", "Parental")
> remove <- apply(drug.mat.ParDiplTetra, 2, removedrugs, thresh = 0.8)
> drug.mat.ParDiplTetra <- drug.mat.ParDiplTetra[, !remove]
> vects <- c(drug.mat.ParDiplTetra[1, ] - drug.mat.ParDiplTetra[2,
+     ], drug.mat.ParDiplTetra[3, ] - drug.mat.ParDiplTetra[2,
+     ])
> fact <- c(rep("C4-Tetra vs C8-Dipl", ncol(drug.mat.ParDiplTetra)),
+     rep("C9-Tetra vs C8-Dip", ncol(drug.mat.ParDiplTetra)))
> p1 <- wilcox.test(drug.mat.ParDiplTetra[1, ], drug.mat.ParDiplTetra[2,
+     ], alternative = "greater", paired = T)$p.value
> p2 <- wilcox.test(drug.mat.ParDiplTetra[3, ], drug.mat.ParDiplTetra[2,
+     ], alternative = "greater", paired = T)$p.value
> p <- c(p1, p2)
> p <- round(p * 1000)/1000
> print(p)

[1] 0.078 0.000
```

Figure 4 A

```
> par(mar = c(8, 5.1, 4.1, 2.1))
> ylab <- "Difference in mean relative cell \n number after inhibitor treatment"
> boxplot(vects ~ fact, las = 3, cex.axis = 0.7, cex.lab = 0.7,
+     cex.main = 0.7, ylab = ylab)
> abline(h = 0, col = "red")
> boxplot(vects ~ fact, las = 3, add = T, cex.axis = 0.7, cex.lab = 0.7,
+     cex.main = 0.7)
> axis(3, at = 1:2, labels = c(p[1], "<0.001"), las = 3, cex.axis = 0.7)
```

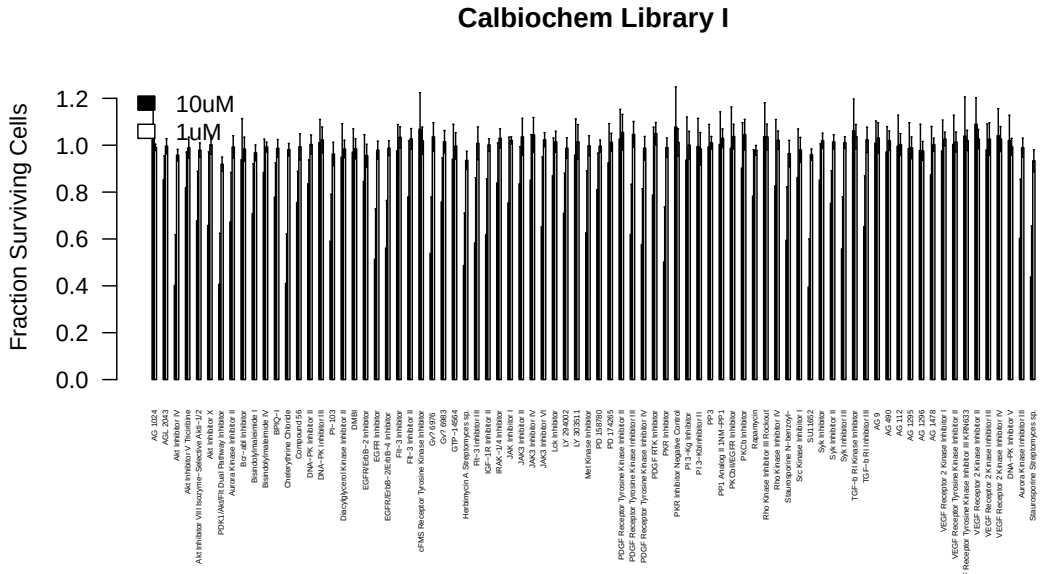


1.9 Comparison 1uM and 10uM Calbiochem Data

```
> load("J://SensitivityData/mat1uM.RData")
> mat10uM <- drug.mat10uM
> stables.sens1uM = c(3, 4)
> unstables.sens1uM = (1:12)[-stables.sens1uM]
> stables.sens = c(4:6, 8:9, 14, 18, 25, 30)
> unstables.sens = (1:30)[-stables.sens]
> unstables.sens <- unstables.sens[-6]
> unstables.sens <- unstables.sens[-8]
> unstables.sens <- unstables.sens[-4]
> mean10uM <- apply(mat10uM[unstables.sens, ], 2, mean, na.rm = T)
> sd10uM <- apply(mat10uM[unstables.sens, ], 2, sd, na.rm = T)
> mean1uM <- apply(mat1uM[unstables.sens1uM, ], 2, mean, na.rm = T)
> sd1uM <- apply(mat1uM[unstables.sens1uM, ], 2, sd, na.rm = T)
> mat.mean <- rbind(mean10uM, mean1uM)
> mat.sd <- rbind(sd10uM, sd1uM)
```

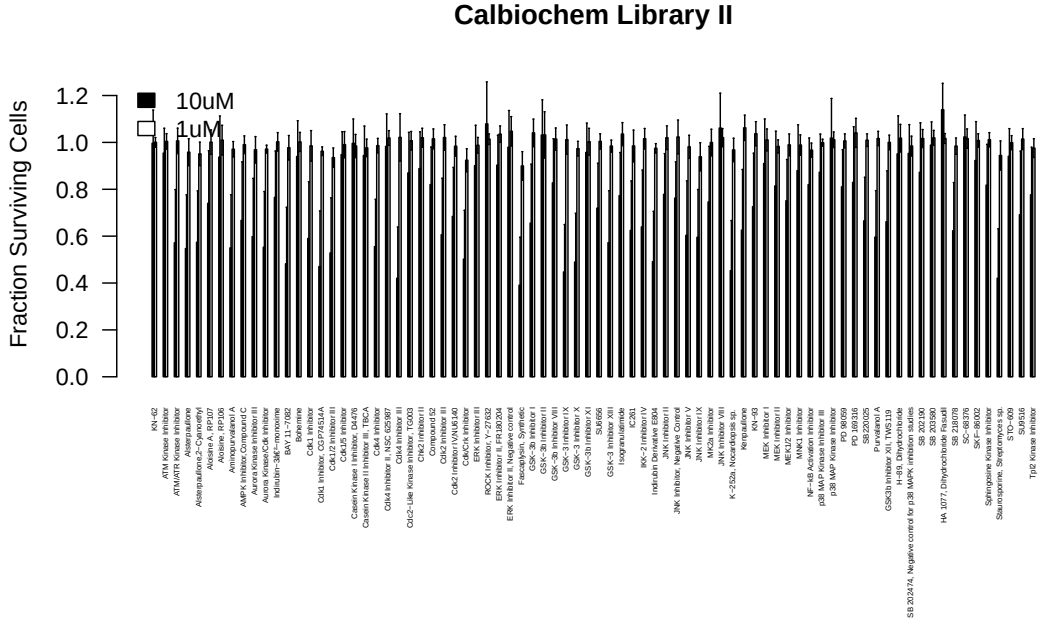
Supplementary Figure 1 (Calbiochem Library I)

```
> par(mar = c(10, 4, 4, 2) + 0.1)
> barplot2(mat.mean[, 1:80], beside = T, space = c(0, 3), border = "black",
+     col = c("black", "white"), cex.names = 0.3, las = 2, ylim = c(0,
+         1.3), ylab = "Fraction Surviving Cells", main = "Calbiochem Library I",
+     ci.l = mat.mean[, 1:80] - mat.sd[, 1:80], ci.u = mat.mean[,
+         1:80] + mat.sd[, 1:80], plot.ci = T)
> legend("topleft", c("10uM", "1uM"), fill = c("black", "white"),
+     bty = "n")
```



textbfSupplementary Figure 1 (Calbiochem Library II)

```
> par(mar = c(10, 4, 4, 2) + 0.1)
> barplot2(mat.mean[, 81:160], beside = T, space = c(0, 3), border = "black",
+     col = c("black", "white"), cex.names = 0.3, las = 2, ylim = c(0,
+         1.3), ylab = "Fraction Surviving Cells", main = "Calbiochem Library II",
+     ci.l = mat.mean[, 81:160] - mat.sd[, 81:160], ci.u = mat.mean[,
+         81:160] + mat.sd[, 81:160], plot.ci = T)
> legend("topleft", c("10uM", "1uM"), fill = c("black", "white"),
+     bty = "n")
```



1.10 Proliferation Analysis

```
> data2 <- read.table("J:/SensitivityData/plate2grouped.txt", fill = F,
+     header = T)
> data2 <- as.matrix(data2)
> data2 <- data2[, c(2, 5:12)]
> data2 = data2[-c(1:4), ]
> timepoints <- data2[, 1]
> time <- t(matrix(rep(as.numeric(timepoints), ncol(data2) - 1),
+     c(31, ncol(data2) - 1)))
> prohelp <- apply(data2[, 2:ncol(data2)], 2, as.numeric)
> data <- cbind(1:ncol(prohelp), 1:ncol(prohelp), rep(1, ncol(prohelp)),
+     t(prohelp))
> MyOpt1 <- grofit.control(smooth.gc = 0.7, interactive = F)
> gg <- grofit(time, data, control = MyOpt1)

= 1. growth curve ================================
-------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 2. growth curve ================================
-------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 3. growth curve ================================
-------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to
```

```
= 4. growth curve ================================
------------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 5. growth curve ================================
------------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 6. growth curve ================================
------------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 7. growth curve ================================
------------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 8. growth curve ================================
------------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to
```

```
> mu <- c()
> for (i in 1:length(gg$gcFit$gcFittedSplines)) mu <- c(mu, gg$gcFit$gcFittedSplines[[i]]$paramet
> names(mu) <- colnames(prohelp)
> prolif2 <- mu
> prolif <- read.csv("J:/SensitivityData/Data2.csv")
> prolif <- prolif[, -2]
> prolif <- as.matrix(prolif)
> timepoints <- prolif[, 1]
> time <- t(matrix(rep(timepoints, ncol(prolif) - 1), c(47, ncol(prolif) -
+     1)))
> prohelp <- prolif[, 2:ncol(prolif)]
> data <- cbind(1:ncol(prohelp), 1:ncol(prohelp), rep(1, ncol(prohelp)),
+     t(prohelp))
> data[10, c(10, 32, 45)] = NA
> data[11, c(6, 7, 9, 10, 32, 45)] = NA
> data[12, c(24, 27, 32, 45)] = NA
> data[13, c(10, 32, 45)] = NA
> data[14, c(10, 24, 27, 32, 45)] = NA
> data[15, c(10, 24, 27, 32)] = NA
> MyOpt1 <- grofit.control(smooth.gc = 0.7, interactive = F)
> gg <- grofit(time, data, control = MyOpt1)
```

```
= 1. growth curve ================================
------------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 2. growth curve ================================
------------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to
```

```
= 3. growth curve ================================
--------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 4. growth curve ================================
--------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 5. growth curve ================================
--------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 6. growth curve ================================
--------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 7. growth curve ================================
--------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 8. growth curve ================================
--------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 9. growth curve ================================
--------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 10. growth curve ===============================
--------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 11. growth curve ===============================
--------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 12. growth curve ===============================
--------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 13. growth curve ===============================
--------------------------------------------------
```

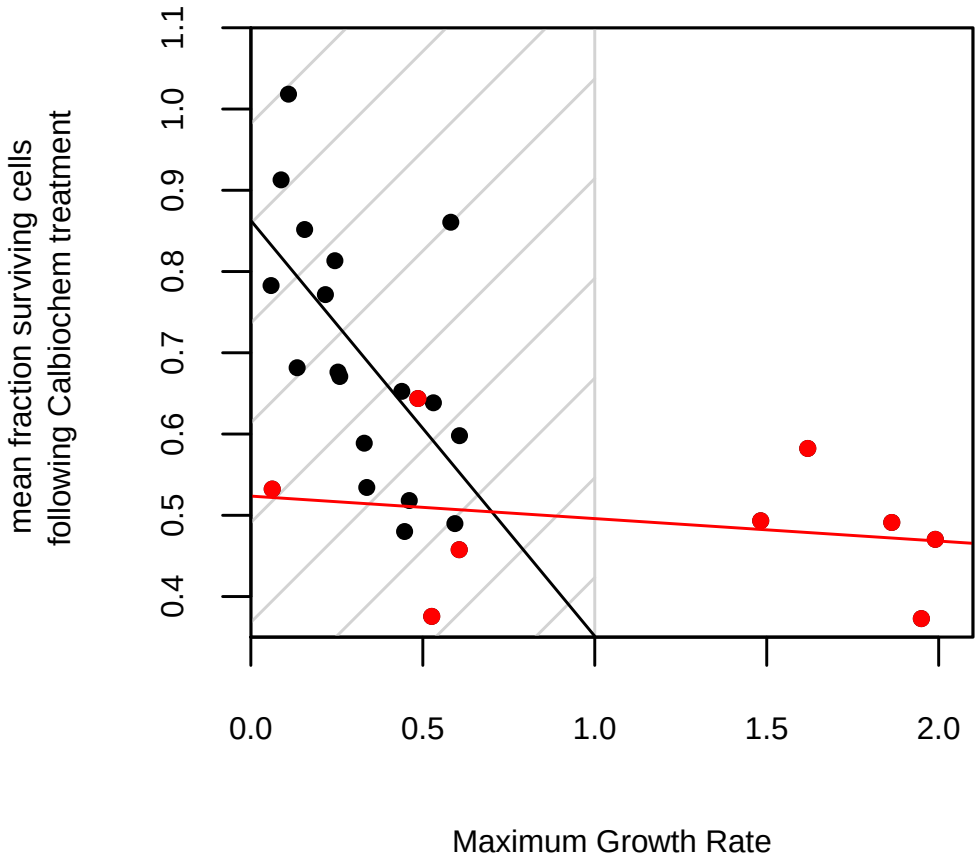
```
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 14. growth curve ================================
----------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 15. growth curve ================================
----------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 16. growth curve ================================
----------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 17. growth curve ================================
----------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 18. growth curve ================================
----------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 19. growth curve ================================
----------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 20. growth curve ================================
----------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 21. growth curve ================================
----------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 22. growth curve ================================
----------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 23. growth curve ================================
----------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 24. growth curve ================================
```

```
-----------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 25. growth curve ================================
-----------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 26. growth curve ================================
-----------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to


= 27. growth curve ================================
-----------------------------------------------------
--> Try to fit model logistic--> Try to fit model richards--> Try to fit model gompertz--> Try to

> mu <- c()
> for (i in 1:length(gg$gcFit$gcFittedSplines)) mu <- c(mu, gg$gcFit$gcFittedSplines[[i]]$paramet
> names(mu) <- colnames(prohelp)
> prolif <- mu
> prolif <- prolif[-c(20:24, 26)]
> prolif <- c(prolif, prolif2)
> prolif <- prolif[-5]
> matendnew <- drug.mat[-c(11, 15), ]
> matend.forprolif <- matendnew[c(3, 5, 7, 12, 14, 16:18, 21:24,
+     25, 28, 4, 6, 8, 13, 11, 26, 1, 2, 9, 10, 15, 20, 19, 27),
+     ]
> comb = rowMeans(matend.forprolif, na.rm = T)
> find.stab <- rep("CIN", length(prolif))
> find.stab[c(2, 6, 11, 14, 15, 16, 17, 18, 23)] <- "MIN"
> prolif <- prolif[-3]
> find.stab <- find.stab[-3]
> comb <- comb[-3]
> print(summary(lm(comb ~ find.stab * prolif)))

Call:
lm(formula = comb ~ find.stab * prolif)

Residuals:
     Min       1Q   Median       3Q      Max
-0.15608 -0.08314  0.01041  0.05807  0.29528

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)         0.86280    0.05713  15.102 1.99e-13 ***
find.stabMIN       -0.33921    0.09469  -3.582  0.00158 **
prolif             -0.51158    0.15424  -3.317  0.00301 **
find.stabMIN:prolif  0.48391    0.16377   2.955  0.00710 **
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

Residual standard error: 0.1166 on 23 degrees of freedom
```

```
Multiple R-squared: 0.5653,        Adjusted R-squared: 0.5086
F-statistic: 9.968 on 3 and 23 DF,  p-value: 0.0002114
```

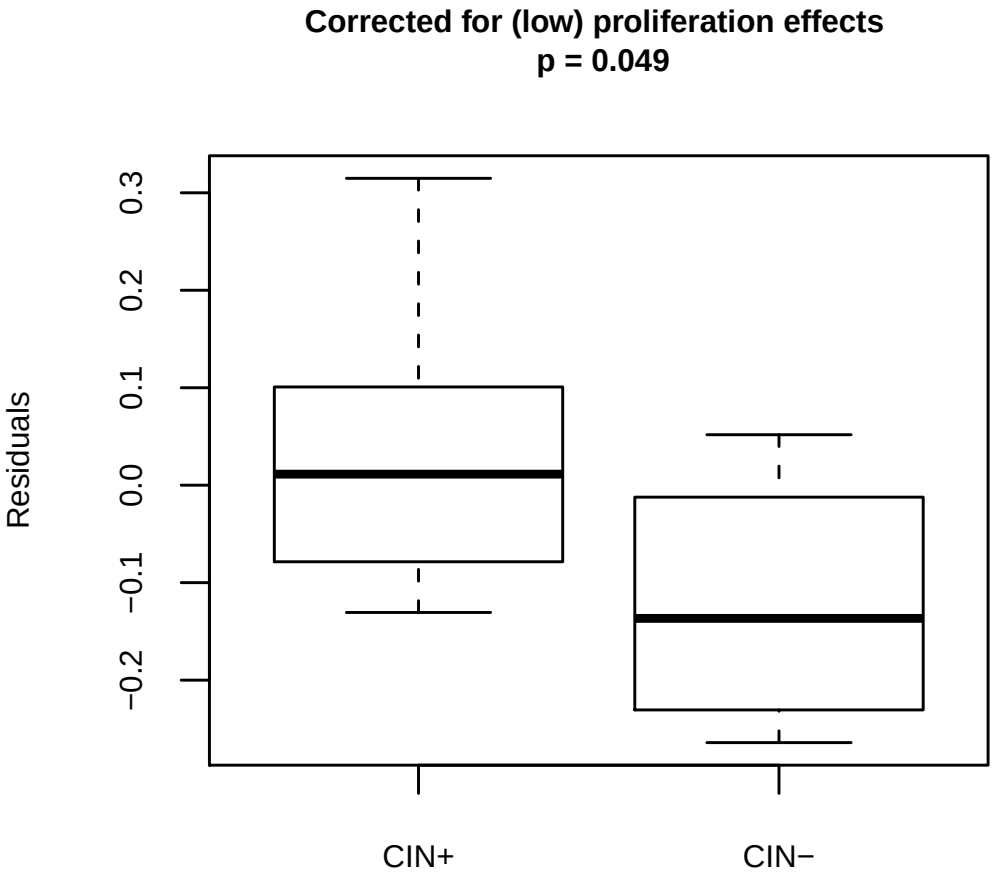
Supplementary Figure 2 A

```
> mar.old <- par()$mar
> par(mar = c(5.1, 5.1, 4.1, 2.1))
> plot(prolif, comb, xlab = "Maximum Growth Rate", ylab = "mean fraction surviving cells \n follo
+     pch = 20, xlim = c(0, 2.1), ylim = c(0.35, 1.1), xaxs = "i",
+     yaxs = "i", type = "n", cex.axis = 0.7, cex.lab = 0.7, cex.main = 0.7)
> abline(v = 1, col = "lightgray")
> polygon(c(0, 0, 1, 1), c(0.35, 1.1, 1.1, 0.35), col = "lightgray",
+     border = "NA", angle = 45, density = 4)
> points(prolif, comb, pch = 20)
> points(prolif[find.stab == "MIN"], comb[find.stab == "MIN"],
+     col = "red", pch = 20)
> abline(lm(comb[find.stab == "CIN"] ~ prolif[find.stab == "CIN"])$coefficients)
> abline(lm(comb[find.stab == "MIN"] ~ prolif[find.stab == "MIN"])$coefficients,
+     col = "red")
> abline(h = c(0.35, 1.1))
> abline(v = c(0, 2.1))
> par(mar = mar.old)
```



Supplementary Figure 2 B

```
> find.lowpro <- find.stab[prolif < 1]
> lm.lowpro <- lm(comb[prolif < 1] ~ prolif[prolif < 1])
> print(wilcox.test(lm.lowpro$residuals ~ find.lowpro, alternative = "greater"))

        Wilcoxon rank sum test

data:  lm.lowpro$residuals by find.lowpro
W = 56, p-value = 0.04908
alternative hypothesis: true location shift is greater than 0

> boxplot(lm.lowpro$residuals ~ find.lowpro, main = "Corrected for (low) proliferation effects \n
+     ylab = "Residuals", cex.axis = 0.7, cex.lab = 0.7, cex.main = 0.7,
+     xaxt = "n")
> axis(1, at = 1:2, labels = c("CIN+", "CIN-"), cex.axis = 0.7)
```



1.11 Mutations for grouped inhibitors

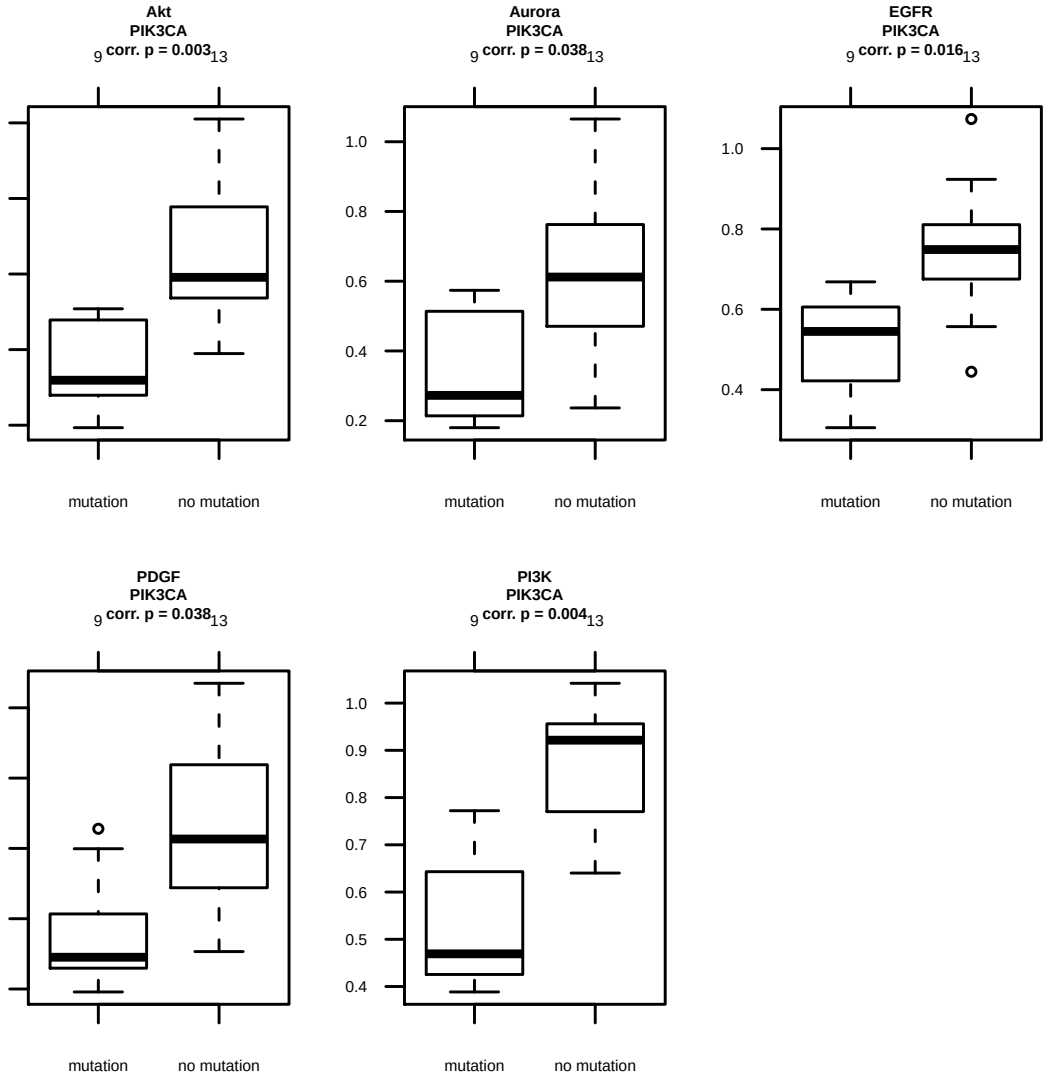
```
> mut.update <- read.xls("J:/SensitivityData/Tablemutnew16122010.xls")
> mut.update <- as.matrix(mut.update)
> mut.update[mut.update == " 0"] <- "no mutation"
> mut.update[mut.update == " 1"] <- "mutation"
> mut.update <- mut.update[, -c(25, 26, 27)]
> mut <- mut.update[, -c(1:4)]
> groups <- c("Akt", "Aurora", "Cdk", "EGFR", "Flt-3", "GSK-3",
+     "JAK3", "JNK", "MEK", "PDGF", "Syk", "PI3K")
> p1.list <- list()
> p2.list <- list()
> count <- 1
> for (ii in groups) {
+     if (ii == "PI3K")
+         gr <- which(colnames(drug.mat) == "LY 294002 " | colnames(drug.mat) ==
+             "PI 3-Kg Inhibitor  ")
```

```
+     else gr <- grep(ii, colnames(drug.mat))
+     act.group <- drug.mat[, gr]
+     combined.simple <- apply(act.group, 1, mean, na.rm = T)
+     cc <- c(4, 5, 6, 8, 9, 14, 18, 25, 30, 1, 2, 3, 7, 10, 12,
+         13, 16, 17, 19, 20, 21, 22, 23, 24, 26:29)
+     combined.simple <- combined.simple[cc]
+     combined.simple <- combined.simple[-13]
+     p1 <- c()
+     count2 <- 1
+     for (i in 1:ncol(mut)) {
+         fact <- mut[, i]
+         if (length(which(fact == "no mutation")) <= 1 | length(which(fact ==
+             "mutation")) <= 1)
+             next()
+         p1 <- c(p1, wilcox.test(combined.simple ~ fact)$p.value)
+         count2 <- count2 + 1
+     }
+     p1.list[[count]] <- p1
+     count <- count + 1
+ }
> corr.p1 <- list()
> for (iii in 1:length(p1.list)) {
+     pvalue <- p1.list[[iii]]
+     help <- cbind(pvalue, 1:length(pvalue))
+     help <- help[order(help[, 1]), ]
+     help <- cbind(help, 1:nrow(help))
+     pk <- help[, 1] * (nrow(help)/help[, 3])
+     help <- cbind(help, pk)
+     help <- help[order(help[, 2]), ]
+     corr.p1[[iii]] <- help[, 4]
+     corr.p1[[iii]][corr.p1[[iii]] > 1] <- 1
+ }
```

Supplementary Figure 3A

```
> count <- 1
> par(mfrow = c(2, 3))
> for (ii in groups) {
+     if (ii == "PI3K")
+         gr <- which(colnames(drug.mat) == "LY 294002 " | colnames(drug.mat) ==
+             "PI 3-Kg Inhibitor  ")
+     else gr <- grep(ii, colnames(drug.mat))
+     act.group <- drug.mat[, gr]
+     combined.simple <- apply(act.group, 1, mean, na.rm = T)
+     cc <- c(4, 5, 6, 8, 9, 14, 18, 25, 30, 1, 2, 3, 7, 10, 12,
+         13, 16, 17, 19, 20, 21, 22, 23, 24, 26:29)
+     combined.simple <- combined.simple[cc]
+     combined.simple <- combined.simple[-13]
+     count2 <- 1
+     for (i in 1:ncol(mut)) {
+         fact <- mut[, i]
+         if (length(which(fact == "no mutation")) <= 1 | length(which(fact ==
+             "mutation")) <= 1)
+             next()
+         if (corr.p1[[count]][count2] < 0.05) {
```

```
+             par(mar = c(2.1, 1, 4.1, 2.1))
+             boxplot(combined.simple ~ fact, main = paste(ii,
+                 "\n", colnames(mut)[i], "\ncorr. p = ", round(1000 *
+                   corr.p1[[count]][count2])/1000, sep = ""),
+                 las = 1, cex.main = 0.5, cex.axis = 0.5, cex.lab = 0.5)
+             axis(3, at = 1:2, labels = c(length(which(fact ==
+                 "mutation")), length(fact) - length(which(fact ==
+                 "mutation")) - length(which(is.na(fact)))), cex.axis = 0.5)
+         }
+         count2 <- count2 + 1
+     }
+     count <- count + 1
+ }
```



1.12 Mutations for all inhibitors (combined)

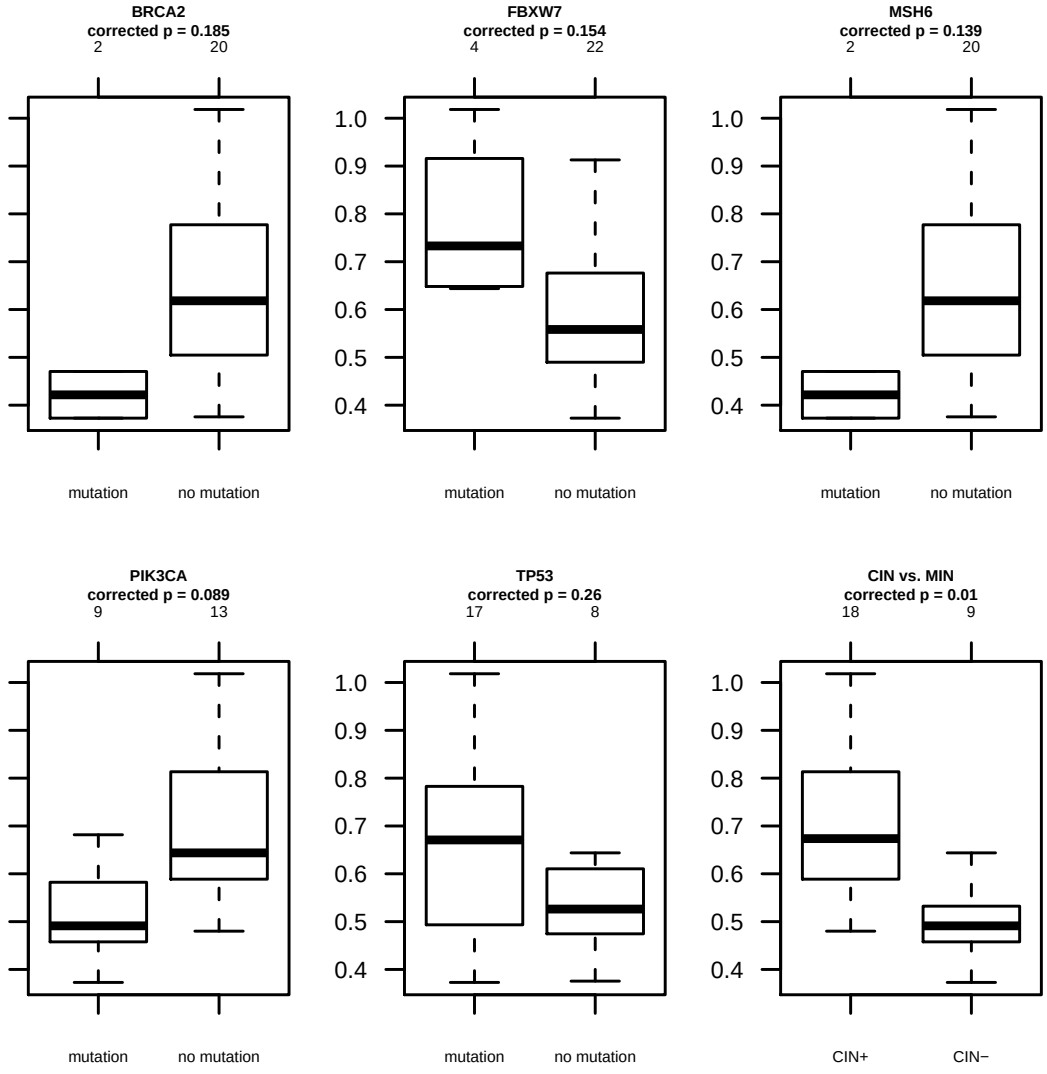
```
> mut <- mut.update[, -c(1:4)]
> combined.simple <- apply(drug.mat, 1, mean, na.rm = T)
```

```
> combined.simple <- combined.simple[cc]
> combined.simple <- combined.simple[-13]
> p3 <- c()
> p4 <- c()
> count <- 1
> for (i in 1:ncol(mut)) {
+     fact <- mut[, i]
+     if (length(which(fact == "no mutation")) <= 1 | length(which(fact ==
+         "mutation")) <= 1)
+         next()
+     p3 <- c(p3, wilcox.test(combined.simple ~ fact)$p.value)
+     count <- count + 1
+ }
> corr.p3 <- c()
> pvalue <- p3
> help <- cbind(pvalue, 1:length(pvalue))
> help <- help[order(help[, 1]), ]
> help <- cbind(help, 1:nrow(help))
> pk <- help[, 1] * (nrow(help)/help[, 3])
> help <- cbind(help, pk)
> help <- help[order(help[, 2]), ]
> corr.p3 <- help[, 4]
> corr.p3[corr.p3 > 1] <- 1
> combined.simple2 <- apply(drug.mat[c(unstables.sens, stables.sens),
+     ], 1, mean, na.rm = T)
> p6 <- c(p3, wilcox.test(combined.simple2 ~ c(rep(1, 18), rep(2,
+     9)))$p.value)
> corr.p6 <- c()
> pvalue <- p6
> help <- cbind(pvalue, 1:length(pvalue))
> help <- help[order(help[, 1]), ]
> help <- cbind(help, 1:nrow(help))
> pk <- help[, 1] * (nrow(help)/help[, 3])
> help <- cbind(help, pk)
> help <- help[order(help[, 2]), ]
> corr.p6 <- help[, 4]
> corr.p6[corr.p6 > 1] <- 1
```

Supplementary Figure 3 B

```
> count <- 1
> par(mfrow = c(2, 3))
> for (i in 1:ncol(mut)) {
+     fact <- mut[, i]
+     if (length(which(fact == "no mutation")) <= 1 | length(which(fact ==
+         "mutation")) <= 1)
+         next()
+     if (corr.p6[count] < 0.6) {
+         par(mar = c(2.1, 1, 4.1, 2.1))
+         boxplot(combined.simple ~ fact, main = paste(colnames(mut)[i],
+             "\ncorrected p = ", round(1000 * corr.p6[count])/1000,
+             sep = ""), las = 1, cex.axis = 0.7, cex.lab = 0.7,
+             cex.main = 0.5, xaxt = "n", ylab = "Mean Relative Cell Number")
+         axis(3, at = 1:2, labels = c(length(which(fact == "mutation")),
+             length(fact) - length(which(fact == "mutation")) -
```

```
+                 length(which(is.na(fact)))), cex.axis = 0.5)
+         axis(1, at = 1:2, c("mutation", "no mutation"), cex.axis = 0.5)
+     }
+     count <- count + 1
+ }
> boxplot(combined.simple2 ~ c(rep("CIN", 18), rep("MIN", 9)),
+     main = paste("CIN vs. MIN", "\ncorrected p = ", round(1000 *
+         corr.p6[length(corr.p6)])/1000, sep = ""), las = 1, cex.axis = 0.7,
+     cex.lab = 0.7, cex.main = 0.5, xaxt = "n")
> axis(3, at = 1:2, labels = c(18, 9), cex.axis = 0.5)
> axis(1, at = 1:2, labels = c("CIN+", "CIN-"), cex.axis = 0.5)
```



1.13 Thymidylate synthase Inhibitors MAD2

```
> load("J://SensitivityData/RObjects/firstDuplicateMUT_MAD2.Rdata")
> load("J://SensitivityData/RObjects/secondDuplicateMUT_MAD2.Rdata")
> load("J://SensitivityData/RObjects/firstDuplicateWT_MAD2.Rdata")
> load("J://SensitivityData/RObjects/secondDuplicateWT_MAD2.Rdata")
```

```
> pval <- c()
> conc <- rep(1:4, 4)
> mut <- c(rep("mut", 8), rep("par", 8))
> data <- cbind(conc, mut, rbind(time5.MUT.MAD2, time5.MUT2.MAD2,
+     time5.WT.MAD2, time5.WT2.MAD2))
> summary(lm(as.numeric(data[, "Fluorouracil "]) ~ as.numeric(data[,
+     "conc"])))

Call:
lm(formula = as.numeric(data[, "Fluorouracil "]) ~ as.numeric(data[,
    "conc"]))

Residuals:
      Min        1Q    Median        3Q       Max
-0.191515 -0.088772  0.001555  0.057528  0.238528

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                1.22233    0.07529  16.236 1.78e-10 ***
as.numeric(data[, "conc"]) -0.19998    0.02749  -7.274 4.07e-06 ***
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

Residual standard error: 0.1229 on 14 degrees of freedom
Multiple R-squared: 0.7908,        Adjusted R-squared: 0.7758
F-statistic: 52.92 on 1 and 14 DF,  p-value: 4.068e-06

> res.camp <- lm(as.numeric(data[, "Fluorouracil "]) ~ as.numeric(data[,
+     "conc"]))$residuals
> print(wilcox.test(res.camp ~ as.factor(data[, "mut"])))

        Wilcoxon rank sum test

data:  res.camp by as.factor(data[, "mut"])
W = 52, p-value = 0.03792
alternative hypothesis: true location shift is not equal to 0

> summary(lm(as.numeric(data[, "Floxuridine"]) ~ as.numeric(data[,
+     "conc"])))

Call:
lm(formula = as.numeric(data[, "Floxuridine"]) ~ as.numeric(data[,
    "conc"]))

Residuals:
      Min        1Q    Median        3Q       Max
-0.196420 -0.108560 -0.004063  0.107194  0.184844

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                0.83412    0.07952  10.489 5.15e-08 ***
as.numeric(data[, "conc"]) -0.17092    0.02904  -5.886 3.96e-05 ***
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1
```

```
Residual standard error: 0.1299 on 14 degrees of freedom
Multiple R-squared: 0.7122,        Adjusted R-squared: 0.6916
F-statistic: 34.65 on 1 and 14 DF,  p-value: 3.963e-05

> res.camp <- lm(as.numeric(data[, "Floxuridine"]) ~ as.numeric(data[,
+     "conc"]))$residuals
> print(wilcox.test(res.camp ~ as.factor(data[, "mut"])))

        Wilcoxon rank sum test

data:  res.camp by as.factor(data[, "mut"])
W = 64, p-value = 0.0001554
alternative hypothesis: true location shift is not equal to 0

> summary(lm(as.numeric(data[, "5-Fluoro-5'-"]) ~ as.numeric(data[,
+     "conc"])))

Call:
lm(formula = as.numeric(data[, "5-Fluoro-5'-"]) ~ as.numeric(data[,
    "conc"]))

Residuals:
     Min       1Q   Median       3Q      Max
-0.57059 -0.05915  0.07898  0.15186  0.22486

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                0.84749    0.14107   6.008 3.21e-05 ***
as.numeric(data[, "conc"]) -0.04206    0.05151  -0.817    0.428
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

Residual standard error: 0.2304 on 14 degrees of freedom
Multiple R-squared: 0.04546,        Adjusted R-squared: -0.02272
F-statistic: 0.6668 on 1 and 14 DF,  p-value: 0.4279

> res.camp <- lm(as.numeric(data[, "5-Fluoro-5'-"]) ~ as.numeric(data[,
+     "conc"]))$residuals
> print(wilcox.test(res.camp ~ as.factor(data[, "mut"])))

        Wilcoxon rank sum test

data:  res.camp by as.factor(data[, "mut"])
W = 44, p-value = 0.2345
alternative hypothesis: true location shift is not equal to 0

> summary(lm(as.numeric(data[, "Carmofur"]) ~ as.numeric(data[,
+     "conc"])))

Call:
lm(formula = as.numeric(data[, "Carmofur"]) ~ as.numeric(data[,
    "conc"]))

Residuals:
     Min       1Q   Median       3Q      Max
```

```
-0.13079 -0.09402 -0.02205  0.06660  0.21021

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                1.23397    0.06717  18.372 3.39e-11 ***
as.numeric(data[, "conc"]) -0.20320    0.02453  -8.285 9.09e-07 ***
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

Residual standard error: 0.1097 on 14 degrees of freedom
Multiple R-squared: 0.8306,        Adjusted R-squared: 0.8185
F-statistic: 68.64 on 1 and 14 DF,  p-value: 9.089e-07
```

```
> res.camp <- lm(as.numeric(data[, "Carmofur"]) ~ as.numeric(data[,
+     "conc"]))$residuals
> print(wilcox.test(res.camp ~ as.factor(data[, "mut"])))

        Wilcoxon rank sum test

data:  res.camp by as.factor(data[, "mut"])
W = 59, p-value = 0.002953
alternative hypothesis: true location shift is not equal to 0
```

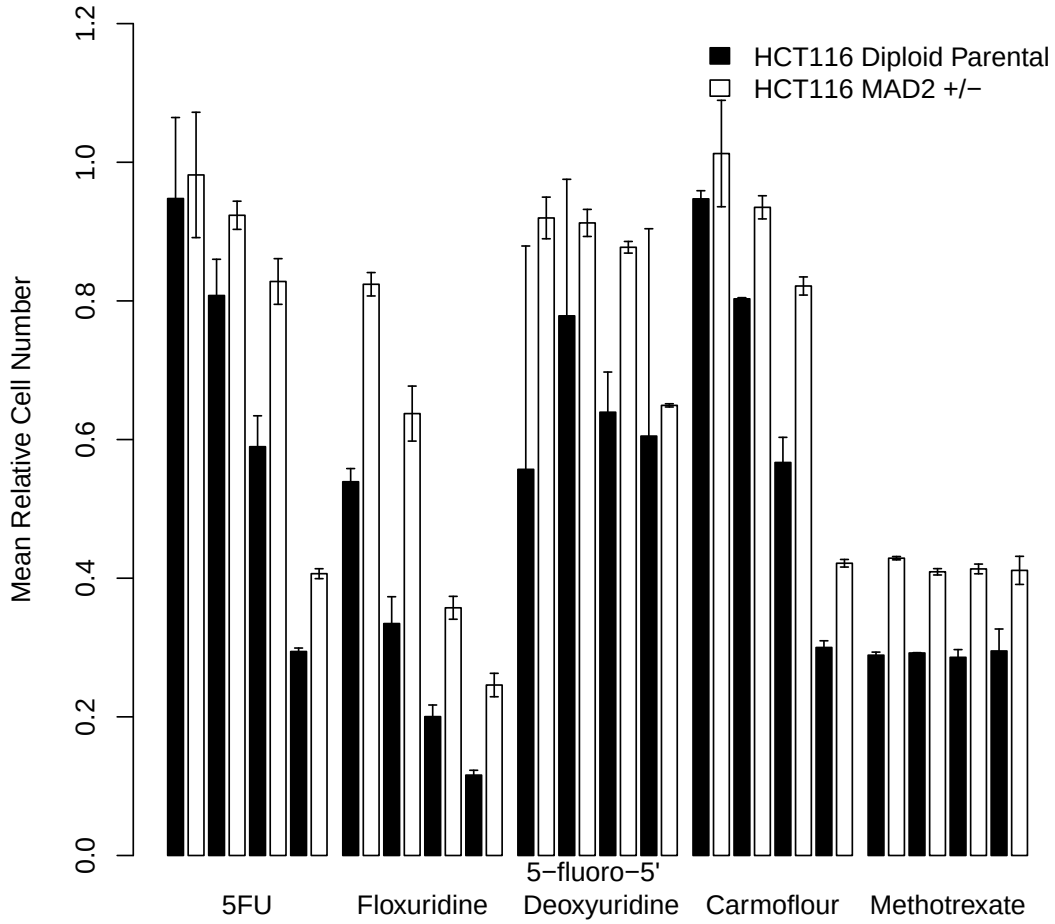
```
> summary(lm(as.numeric(data[, "Methotrexate"]) ~ as.factor(data[,
+     "conc"])))

Call:
lm(formula = as.numeric(data[, "Methotrexate"]) ~ as.factor(data[,
    "conc"]))

Residuals:
     Min       1Q   Median       3Q      Max
-0.08984 -0.06062  0.00572  0.06424  0.07840

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                0.358925   0.037178   9.654 5.23e-07 ***
as.factor(data[, "conc"])2 -0.008244   0.052578  -0.157    0.878
as.factor(data[, "conc"])3 -0.009318   0.052578  -0.177    0.862
as.factor(data[, "conc"])4 -0.005776   0.052578  -0.110    0.914
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

Residual standard error: 0.07436 on 12 degrees of freedom
Multiple R-squared: 0.003124,        Adjusted R-squared: -0.2461
F-statistic: 0.01254 on 3 and 12 DF,  p-value: 0.998
```

```
> res.camp <- lm(as.numeric(data[, "Methotrexate"]) ~ as.factor(data[,
+     "conc"]))$residuals
> print(wilcox.test(res.camp ~ as.factor(data[, "mut"])))

        Wilcoxon rank sum test

data:  res.camp by as.factor(data[, "mut"])
W = 64, p-value = 0.0001554
alternative hypothesis: true location shift is not equal to 0
```

```
> dataforplot <- data[, c("Fluorouracil ", "Floxuridine", "5-Fluoro-5'-",
+     "Carmofur", "Methotrexate")]
> mean.mat <- matrix(NA, nr = 8, nc = 5)
> sd.mat <- matrix(NA, nr = 8, nc = 5)
> dataforplot <- dataforplot[c(1, 9, 2, 10, 3, 11, 4, 12, 5, 13,
+     6, 14, 7, 15, 8, 16), ]
> dataforplot <- apply(dataforplot, 2, as.numeric)
> for (i in 1:8) {
+     mean.mat[i, ] <- (dataforplot[i, ] + dataforplot[i + 8, ])/2
+     sd.mat[i, ] <- dataforplot[i, ] - (dataforplot[i, ] + dataforplot[i +
+         8, ])/2
+ }
```

Supplementary Figure 6 A (MAD2)

```
> reorder <- c(2, 1, 4, 3, 6, 5, 8, 7)
> barplot2(mean.mat[reorder, ], plot.ci = T, ci.l = mean.mat[reorder,
+     ] - sd.mat[reorder, ], ci.u = mean.mat[reorder, ] + sd.mat[reorder,
+     ], beside = T, col = rep(c("black", "white"), 20), space = c(rep(0.3,
+     8), 1, rep(0.3, 7), 1, rep(0.3, 7), 1, rep(0.3, 7), 1, rep(0.3,
+     7)), names = c("5FU", "Floxuridine", "5-fluoro-5' \n Deoxyuridine",
+     "Carmoflour", "Methotrexate"), ylab = "Mean Relative Cell Number",
+     ylim = c(0, 1.2))
> legend("topright", c("HCT116 Diploid Parental", "HCT116 MAD2 +/-"),
+     fill = c("black", "white"), bty = "n")
```



1.14 Thymidylate synthase Inhibitors PTTG1

```
> load("J:/SensitivityData/RObjects/firstDuplicateMUT_Securin.Rdata")
> load("J:/SensitivityData/RObjects/secondDuplicateMUT_Securin.Rdata")
> load("J:/SensitivityData/RObjects/firstDuplicateWT_Securin.Rdata")
> load("J:/SensitivityData/RObjects/secondDuplicateWT_Securin.Rdata")
> time5.MUT.Securin[time5.MUT.Securin > 1.4] = NA
> time5.MUT2.Securin[time5.MUT2.Securin > 1.4] = NA
> time5.WT.Securin[time5.WT.Securin > 1.4] = NA
> time5.WT2.Securin[time5.WT2.Securin > 1.4] = NA
> load("J:/SensitivityData/namesBiolog.RData")
> colnames(time5.MUT.Securin) <- nam
> colnames(time5.MUT2.Securin) <- nam
> colnames(time5.WT.Securin) <- nam
> colnames(time5.WT2.Securin) <- nam
> pval <- c()
> conc <- rep(1:4, 4)
```

```
> mut <- c(rep("mut", 8), rep("par", 8))
> data <- cbind(conc, mut, rbind(time5.MUT.Securin, time5.MUT2.Securin,
+     time5.WT.Securin, time5.WT2.Securin))
> summary(lm(as.numeric(data[, "Fluorouracil "]) ~ as.numeric(data[,
+     "conc"])))

Call:
lm(formula = as.numeric(data[, "Fluorouracil "]) ~ as.numeric(data[,
    "conc"]))

Residuals:
     Min       1Q   Median       3Q      Max
-0.14853 -0.05153 -0.00911  0.03801  0.14610

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                 1.25289    0.05039   24.86 5.52e-13 ***
as.numeric(data[, "conc"]) -0.24881    0.01840  -13.52 1.99e-09 ***
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

Residual standard error: 0.08228 on 14 degrees of freedom
Multiple R-squared: 0.9289,        Adjusted R-squared: 0.9238
F-statistic: 182.9 on 1 and 14 DF,  p-value: 1.990e-09

> res.camp <- lm(as.numeric(data[, "Fluorouracil "]) ~ as.numeric(data[,
+     "conc"]))$residuals
> print(wilcox.test(res.camp ~ as.factor(data[, "mut"])))

        Wilcoxon rank sum test

data:  res.camp by as.factor(data[, "mut"])
W = 51, p-value = 0.04988
alternative hypothesis: true location shift is not equal to 0

> summary(lm(as.numeric(data[, "Floxuridine"]) ~ as.numeric(data[,
+     "conc"])))

Call:
lm(formula = as.numeric(data[, "Floxuridine"]) ~ as.numeric(data[,
    "conc"]))

Residuals:
      Min        1Q    Median        3Q       Max
-0.051114 -0.026583  0.001734  0.020412  0.060745

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                0.397001   0.022561   17.60 6.05e-11 ***
as.numeric(data[, "conc"]) -0.082887   0.008238  -10.06 8.66e-08 ***
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

Residual standard error: 0.03684 on 14 degrees of freedom
Multiple R-squared: 0.8785,        Adjusted R-squared: 0.8698
F-statistic: 101.2 on 1 and 14 DF,  p-value: 8.657e-08
```

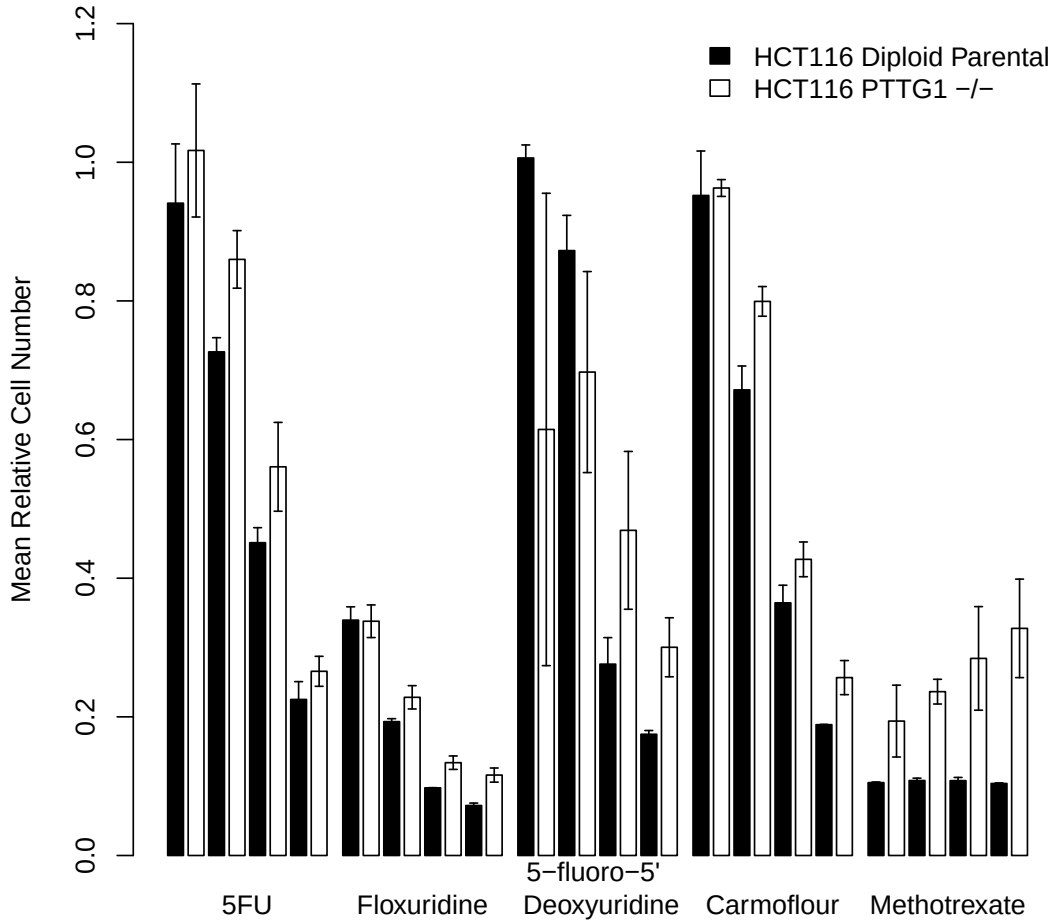
```
> res.camp <- lm(as.numeric(data[, "Floxuridine"]) ~ as.numeric(data[,
+     "conc"]))$residuals
> print(wilcox.test(res.camp ~ as.factor(data[, "mut"])))

        Wilcoxon rank sum test

data:  res.camp by as.factor(data[, "mut"])
W = 46, p-value = 0.1605
alternative hypothesis: true location shift is not equal to 0

> summary(lm(as.numeric(data[, "5-Fluoro-5'-"]) ~ as.numeric(data[,
+     "conc"])))

Call:
lm(formula = as.numeric(data[, "5-Fluoro-5'-"]) ~ as.numeric(data[,
    "conc"]))

Residuals:
     Min       1Q   Median       3Q      Max
-0.59710 -0.09360  0.05516  0.14202  0.26547

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                1.08405    0.13194   8.216 1.00e-06 ***
as.numeric(data[, "conc"]) -0.21307    0.04818  -4.422 0.000579 ***
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

Residual standard error: 0.2155 on 14 degrees of freedom
Multiple R-squared: 0.5828,        Adjusted R-squared: 0.553
F-statistic: 19.56 on 1 and 14 DF,  p-value: 0.0005792

> res.camp <- lm(as.numeric(data[, "5-Fluoro-5'-"]) ~ as.numeric(data[,
+     "conc"]))$residuals
> print(wilcox.test(res.camp ~ as.factor(data[, "mut"])))

        Wilcoxon rank sum test

data:  res.camp by as.factor(data[, "mut"])
W = 28, p-value = 0.7209
alternative hypothesis: true location shift is not equal to 0

> summary(lm(as.numeric(data[, "Carmofur"]) ~ as.numeric(data[,
+     "conc"])))

Call:
lm(formula = as.numeric(data[, "Carmofur"]) ~ as.numeric(data[,
    "conc"]))

Residuals:
     Min       1Q   Median       3Q      Max
-0.11164 -0.05158 -0.00304  0.04101  0.11565

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)                 1.21389    0.03977   30.52 3.29e-14 ***
as.numeric(data[, "conc"]) -0.25441    0.01452  -17.52 6.43e-11 ***
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

Residual standard error: 0.06495 on 14 degrees of freedom
Multiple R-squared: 0.9564,        Adjusted R-squared: 0.9533
F-statistic: 306.9 on 1 and 14 DF,  p-value: 6.43e-11

> res.camp <- lm(as.numeric(data[, "Carmofur"]) ~ as.numeric(data[,
+     "conc"]))$residuals
> print(wilcox.test(res.camp ~ as.factor(data[, "mut"])))

        Wilcoxon rank sum test

data:  res.camp by as.factor(data[, "mut"])
W = 53, p-value = 0.02813
alternative hypothesis: true location shift is not equal to 0

> summary(lm(as.numeric(data[, "Methotrexate"]) ~ as.factor(data[,
+     "conc"])))

Call:
lm(formula = as.numeric(data[, "Methotrexate"]) ~ as.factor(data[,
    "conc"]))

Residuals:
     Min       1Q   Median       3Q      Max
-0.11270 -0.07154 -0.02549  0.05519  0.18292

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                0.14951    0.05260   2.843   0.0148 *
as.factor(data[, "conc"])2 0.02272    0.07438   0.305   0.7652
as.factor(data[, "conc"])3 0.04664    0.07438   0.627   0.5424
as.factor(data[, "conc"])4 0.06624    0.07438   0.890   0.3907
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

Residual standard error: 0.1052 on 12 degrees of freedom
Multiple R-squared: 0.06956,        Adjusted R-squared: -0.163
F-statistic: 0.2991 on 3 and 12 DF,  p-value: 0.8255

> res.camp <- lm(as.numeric(data[, "Methotrexate"]) ~ as.factor(data[,
+     "conc"]))$residuals
> print(wilcox.test(res.camp ~ as.factor(data[, "mut"])))

        Wilcoxon rank sum test

data:  res.camp by as.factor(data[, "mut"])
W = 64, p-value = 0.0001554
alternative hypothesis: true location shift is not equal to 0

> dataforplot <- data[, c("Fluorouracil ", "Floxuridine", "5-Fluoro-5'-",
+     "Carmofur", "Methotrexate")]
```

```
> mean.mat <- matrix(NA, nr = 8, nc = 5)
> sd.mat <- matrix(NA, nr = 8, nc = 5)
> dataforplot <- dataforplot[c(1, 9, 2, 10, 3, 11, 4, 12, 5, 13,
+     6, 14, 7, 15, 8, 16), ]
> dataforplot <- apply(dataforplot, 2, as.numeric)
> for (i in 1:8) {
+     mean.mat[i, ] <- (dataforplot[i, ] + dataforplot[i + 8, ])/2
+     sd.mat[i, ] <- dataforplot[i, ] - (dataforplot[i, ] + dataforplot[i +
+         8, ])/2
+ }
```

Supplementary Figure 6 A (PTTG1)

```
> reorder <- c(2, 1, 4, 3, 6, 5, 8, 7)
> barplot2(mean.mat[reorder, ], plot.ci = T, ci.l = mean.mat[reorder,
+     ] - sd.mat[reorder, ], ci.u = mean.mat[reorder, ] + sd.mat[reorder,
+     ], beside = T, col = rep(c("black", "white"), 20), space = c(rep(0.3,
+     8), 1, rep(0.3, 7), 1, rep(0.3, 7), 1, rep(0.3, 7), 1, rep(0.3,
+     7)), names = c("5FU", "Floxuridine", "5-fluoro-5' \n Deoxyuridine",
+     "Carmoflour", "Methotrexate"), ylab = "Mean Relative Cell Number",
+     ylim = c(0, 1.2))
> legend("topright", c("HCT116 Diploid Parental", "HCT116 PTTG1 -/-"),
+     fill = c("black", "white"), bty = "n")
```



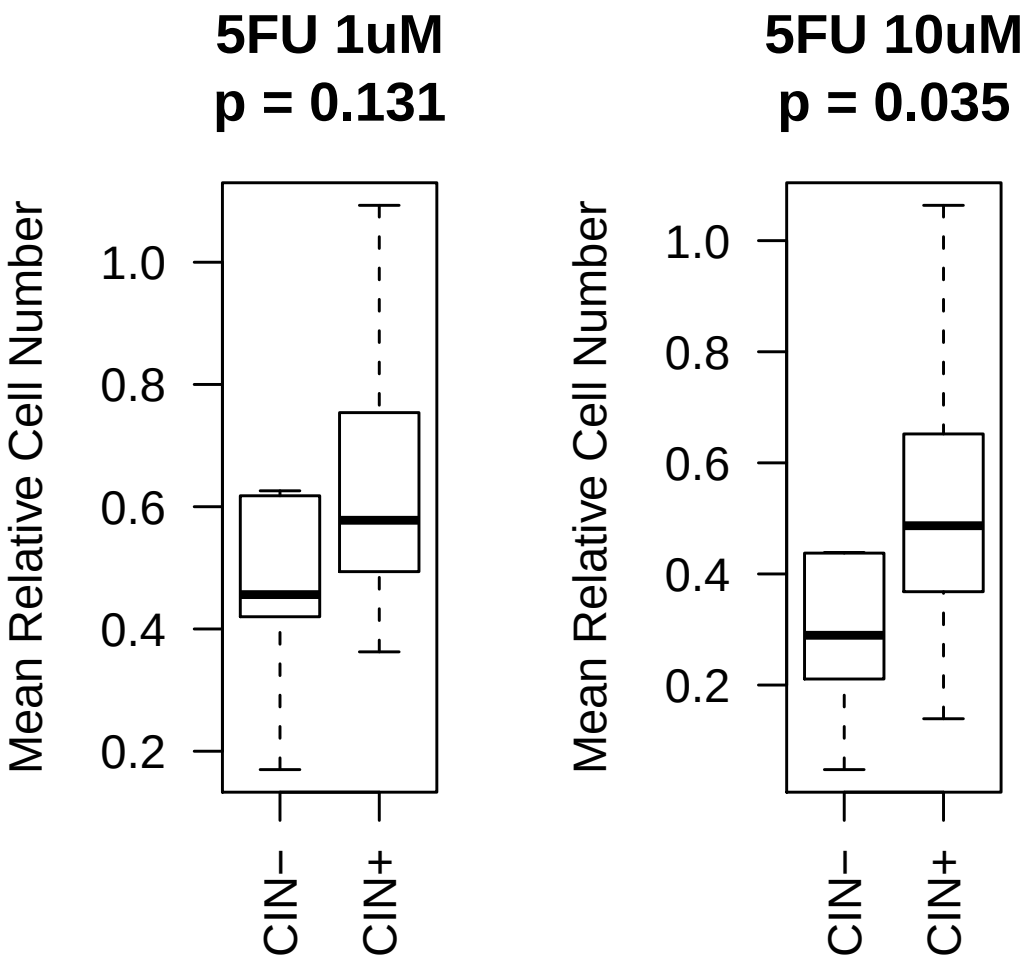
1.15 5FU drugs Colon cell lines

```
> files <- paste("J:/SensitivityData/FU5SN38/FU5drugs/", list.files("J:/SensitivityData/FU5SN38/F
+     sep = "")
> tmp <- substr(files, 1, nchar(files) - 5)
> data.list <- list()
> for (i in 1:length(files)) {
+     data.list[[i]] <- read.csv(files[i], header = F)
+     data.list[[i]] <- as.matrix(data.list[[i]])[, 3:10]
+     data.list[[i]] <- sweep(data.list[[i]][1:4, ], MARGIN = 2,
+         FUN = "/", apply(data.list[[i]][5:8, ], 2, mean, na.rm = T))
+ }
> names(data.list) <- files
> data.list.new <- list()
> count <- 1
> store <- 0
> mean.mat <- matrix(NA, nr = 27, nc = 8)
```

```
> sd.mat <- matrix(NA, nr = 27, nc = 8)
> nam <- c()
> for (i in 1:length(tmp)) {
+     if (i %in% store)
+         next()
+     if (length(which(tmp == tmp[i])) == 2) {
+         store <- c(store, which(tmp == tmp[i]))
+         data.list.new[[count]] <- rbind(data.list[[which(tmp ==
+             tmp[i])[1]]], data.list[[which(tmp == tmp[i])[2]]])
+     }
+     else data.list.new[[count]] <- data.list[[which(tmp == tmp[i])]]
+     nam <- c(nam, tmp[i])
+     mean.mat[count, ] <- apply(data.list.new[[count]], 2, mean,
+         na.rm = T)
+     sd.mat[count, ] <- apply(data.list.new[[count]], 2, sd, na.rm = T)
+     count <- count + 1
+ }
> names(data.list.new) <- nam
> rownames(mean.mat) <- nam
> rownames(sd.mat) <- nam
> colnames(mean.mat) <- c("5FU 10uM", "5FU 1uM", "5FU 100nM", "5FU 10nM",
+     "SN38 1nM", "SN38 10nM", "SN38 100nM", "SN38 1uM")
> colnames(sd.mat) <- c("5FU 10uM", "5FU 1uM", "5FU 100nM", "5FU 10nM",
+     "SN38 1nM", "SN38 10nM", "SN38 100nM", "SN38 1uM")
> MIN <- c(4, 5, 6, 7, 8, 12, 15, 22, 27)
> CIN <- (1:nrow(mean.mat))[-MIN]
> fact <- rep(NA, nrow(mean.mat))
> fact[MIN] <- "CIN-"
> fact[CIN] <- "CIN+"
```

Supplementary Figure 6 B

```
> par(mfrow = c(1, 2))
> for (i in 2:1) boxplot(mean.mat[, i] ~ fact, main = paste(colnames(mean.mat)[i],
+     "\np = ", round(1000 * wilcox.test(mean.mat[, i] ~ fact)$p.value)/1000,
+     sep = ""), ylab = "Mean Relative Cell Number", las = 2)
```


5FU 1uM
p = 0.131
Mean Relative Cell Number
1.0
0.8
0.6
0.4
0.2
CIN−
CIN+
5FU 10uM
p = 0.035
Mean Relative Cell Number
1.0
0.8
0.6
0.4
0.2
CIN−
CIN+