The GI50 Values of Their Sensitive and Resistant Cell Lines Overlap

Kevin R. Coombes, Jing Wang, and Keith A. Baggerly

13 March 2007

1 Description of the problem

Recently, Potti and colleagues [?] published an article in *Nature Medicine* that claims to find genomic signatures, based on the NCI60 cell lines response to drug treatment, that can predict patient response to chemotherapy. In this note, we choose cell lines that we believe should be sensitive or resistant, and we investigate whether the cell lines they use for docetaxel are actually sensitive or resistant based on their own criteria.

2 Load the prepared data

The first code chunk loads the Novartis U95A data for 59 of the NCI60 cell lines. It also loads the dose response data to ten drugs for those cell lines. Basically, this reduces to loading the file `novartis.Rda` if it has already been created; otherwise, it creates it.

```
> prep <- file.path("Tangled", "prepareData.R")
> Stangle(file.path("RNowebSource", "prepareData.Rnw"),
+     output = prep)

Writing to file Tangled/prepareData.R

> source(prep)
> rm(prep)
```

The second code chunk loads the information we need in order to figure out which cell lines were called sensitive and resistant by Potti and colleagues.

```
> pred <- file.path("Tangled", "predCellLines.R")
> chem <- file.path("RDataObjects", "chemoPredictors.Rda")
> if (file.exists(chem)) {
+     load(chem)
+ } else {
```

```
+     Stangle(file.path("RNowebSource", "predCellLines.Rnw"),
+         output = pred)
+     source(pred)
+ }
> rm(pred, chem)
```

3 Which cell lines do Potti et al. say are sensitive or resistant to docetaxel?

The predictorsInfo data frame contains the drug names and response status, as described by Potti and colleagues.

```
> pi <- predictorsInfo
> pds <- which(pi[, "drugName"] == "Doce" & pi[, "responseStatus"] ==
+     "Sensitive")
> pdr <- which(pi[, "drugName"] == "Doce" & pi[, "responseStatus"] ==
+     "Resistant")
> pottiCellnameDoceSens <- as.character(pi$Source)[pds]
> pottiCellnameDoceResi <- as.character(pi$Source)[pdr]
> rm(pi, pds, pdr)
```

Here are the lines that say are sensitive to docetaxel:

```
HL-60(TB), SF-539, HT29, HOP-62, SK-MEL-2, SK-MEL-5, NCI-H522
```

Here are the lines that say are resistant to docetaxel:

```
EKVX, IGROV1, OVCAR-4, 786-0, CAKI-1, SN12C, TK-10
```

4 Getting the data describing response to docetaxel

Now we get the GI50, LC50, and TGI data for the NCI60 cell line response to treatment with docetaxel. This requires us to convert the drug name to an NSC number:

```
> doceNSC <- as.character(nsc["Doce"])
> doceNSC

[1] "628503"
```

Now all three of the objects (GI50, LC50, and TGI) contain the same column headings:

```
> sum(colnames(LC50) != colnames(TGI))

[1] 0
```

```
> sum(colnames(LC50) != colnames(GI50))

[1] 0
```

So, we can, in principle, extract the same columns from each of the three sources. We start, however, by determining which of three concentrations actually gives useful information. For LC50 (Figure 1), there is essentially no useful data. For the experiment with the lowest starting concentration (10^{-7} M), everything was truncated at the same value. For the experiment with the intermediate starting concentration (10^{-6} M), all but three of the values are truncated. Although there are some differences in the experiment with starting concentration 10^{-4} M, with the exception of `COLO 205`, these differences are extremely small.

For TGI (Figure 2), the two lower starting concentrations do not give useful information, but the highest starting concentration is meaningful. For GI50 (Figure 3), the two lower starting concentrations give useful (and correlated) information.

So, in the end, we use the first experiment (using the highest starting dose concentration) to define LC50 and TGI values, and take an unweighted average of the last two experiments (using lower starting concentrations) to define the GI50 values.

```
> doceGI50 <- as.vector((apply(GI50[, isDocetaxel[2:3]],
+     1, mean, na.rm = TRUE)))
> doceLC50 <- as.vector((LC50[, isDocetaxel[1]]))
> doceTGI <- as.vector((TGI[, isDocetaxel[1]]))
> names(doceGI50) <- rownames(GI50)
> names(doceLC50) <- rownames(LC50)
> names(doceTGI) <- rownames(TGI)
> rm(isDocetaxel)
```

Note that three cell lines are missing data for either the GI50 values or the TGI values:

```
> missing <- names(doceGI50)[is.na(doceGI50) | is.na(doceTGI)]
> missing

[1] "DU-145"          "MDA-MB-231/ATCC" "T-47D"
```

5 Which cell lines do we think are sensitive or resistant?

Because of our earlier observation that the differences in LC50 values were too small to be meaningful, we did not use the LC50 values to select cell lines. Our plan is to remove the central third of the data based on the GI50 scores an on the TGI scores, and only keep cell lines that are in the same extreme third for both measures. Here "central third" is defined robustly, in terms of quantiles.

```
> quantCut <- 0.335
> cellLineNames <- function(v) names(doceGI50[v])
```

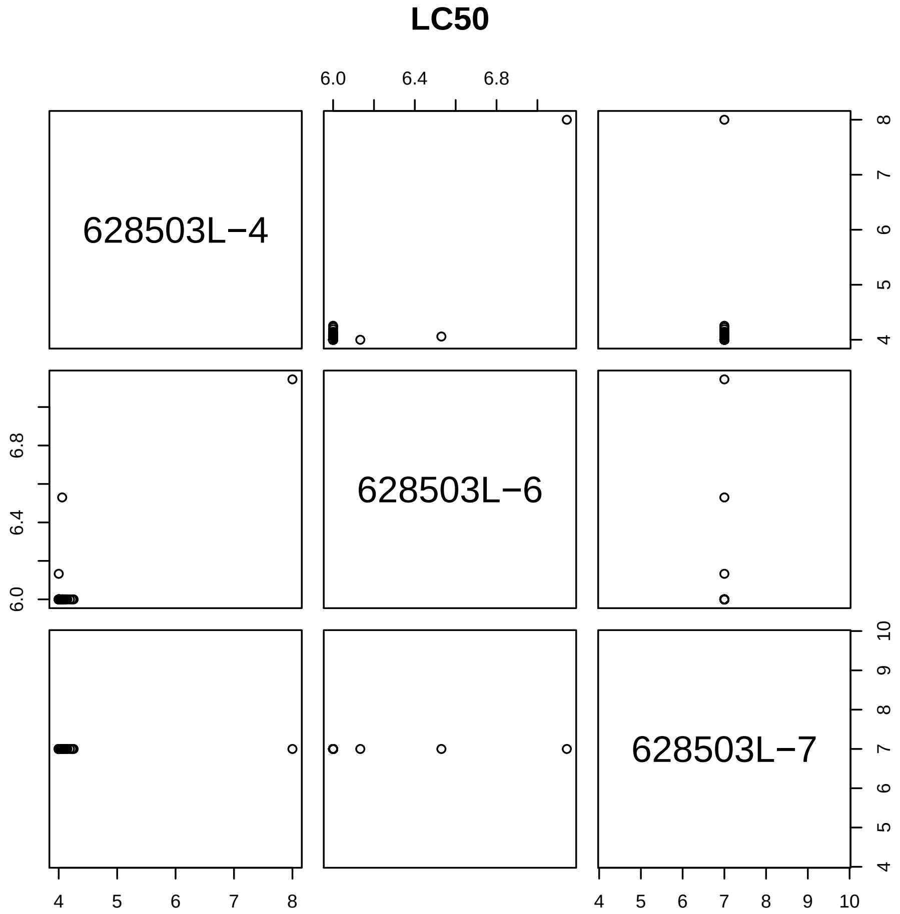

Figure 1: Pairs plot of the three experiments attempting to measure LC50 values for response to docetaxel.

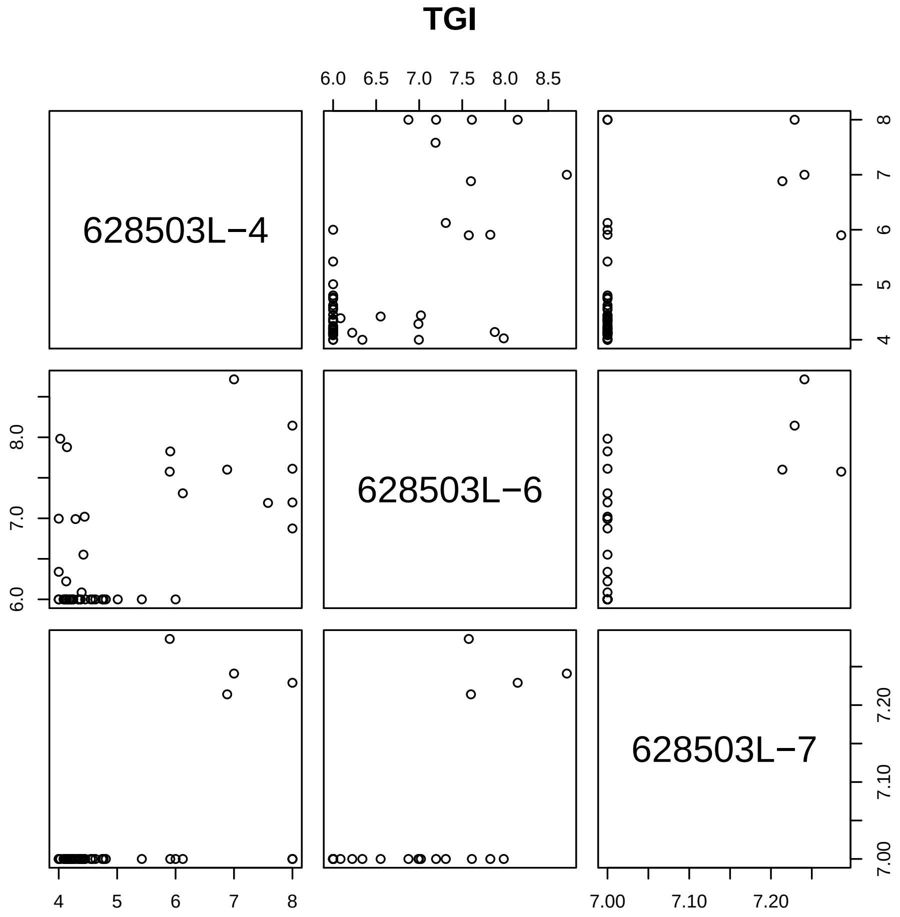

Figure 2: Pairs plot of the three experiments attempting to measure TGI values for response to docetaxel.

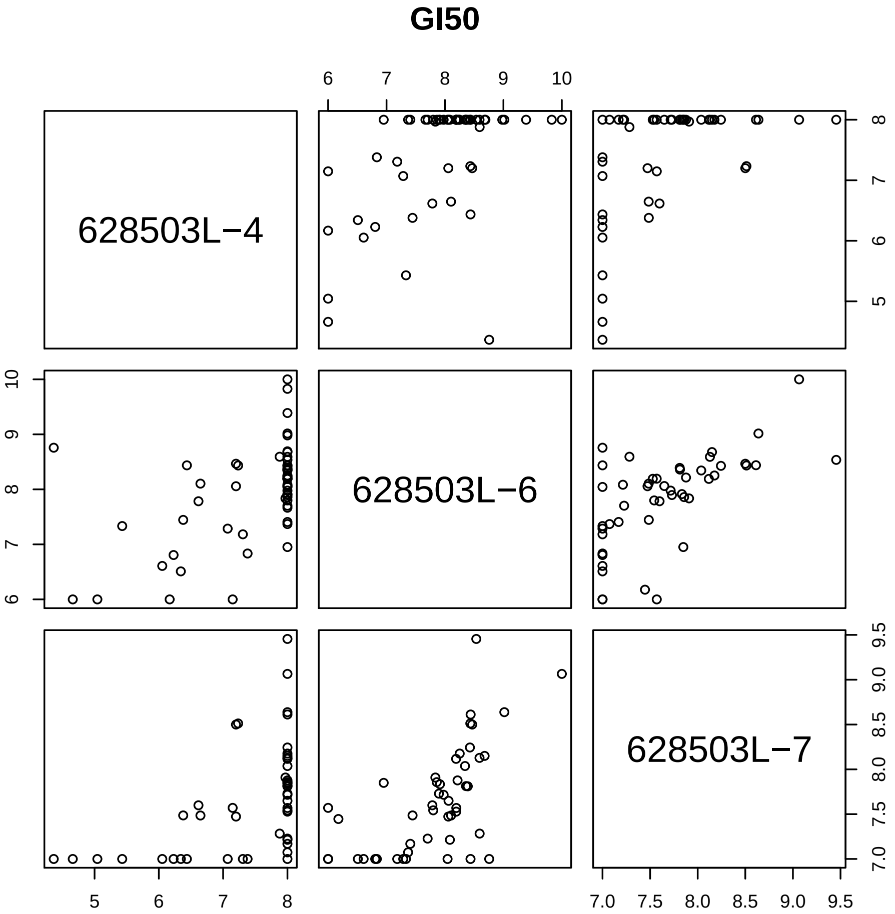

Figure 3: Pairs plot of the three experiments attempting to measure GI50 values for response to docetaxel.

```
> temp <- doceGI50 > quantile(doceGI50, 1 - quantCut, na.rm = TRUE) &
+     doceTGI > quantile(doceTGI, 1 - quantCut, na.rm = TRUE) &
+     !is.na(doceGI50) & !is.na(doceTGI)
> ourCellnameDoceSens <- cellLineNames(temp)
> temp <- doceGI50 < quantile(doceGI50, quantCut, na.rm = TRUE) &
+     doceTGI < quantile(doceTGI, quantCut, na.rm = TRUE) &
+     !is.na(doceGI50) & !is.na(doceTGI)
> ourCellnameDoceResi <- cellLineNames(temp)
> rm(missing, temp)
```

Here are the lines we think are sensitive:

```
COLO 205, HCC-2998, HL-60(TB), HT29, MDA-MB-435, NCI-H522, RPMI-8226, SF-539
```

Here are the lines we think are resistant:

```
786-0, ACHN, CAKI-1, EKVX, IGROV1, OVCAR-4, SF-268
```

6 How well do these lists agree?

We address this question graphically in Figure 4, which plots the GI50 values against the TGI values, coloring or circling the chosen cell lines. Although they identify 7 sensitive cell lines and we identify 8, only 4 cell lines are on both lists. Three of the cell lines that are most sensitive, based on the TGI scores, are on our list but not on their list. Significantly, two of the cell lines that they call sensitive have GI50 values that are in the middle of the range of the GI50 values for the ines they call resistant.

6.1 Did they use the LC50 values?

In Figure 5, we extend our plot of GI50 and TGI values by plotting both of them against the LC50 values. As you can see, only one cell line (COLO 205) really appears sensitive based on the LC50 values. Although this cell line also has extreme values for both TGI and GI50, it is not included on their list (but is on our list). It does look as though two of the other cell lines that we included as sensitive might have been eliminated from their list because the LC50 values (like almost every other cell line) were truncated at the maximum starting concentration of 10^{-4}.

6.2 Did they really use the GI50 values?

According to their supplementary methods, they only included cell lines that were more than one standard deviation away from the mean GI50 value. We now look to see how many cell lines meet that criterion.

```
> m <- mean(doceGI50, na.rm = TRUE)
> s <- sd(doceGI50, na.rm = TRUE)
```

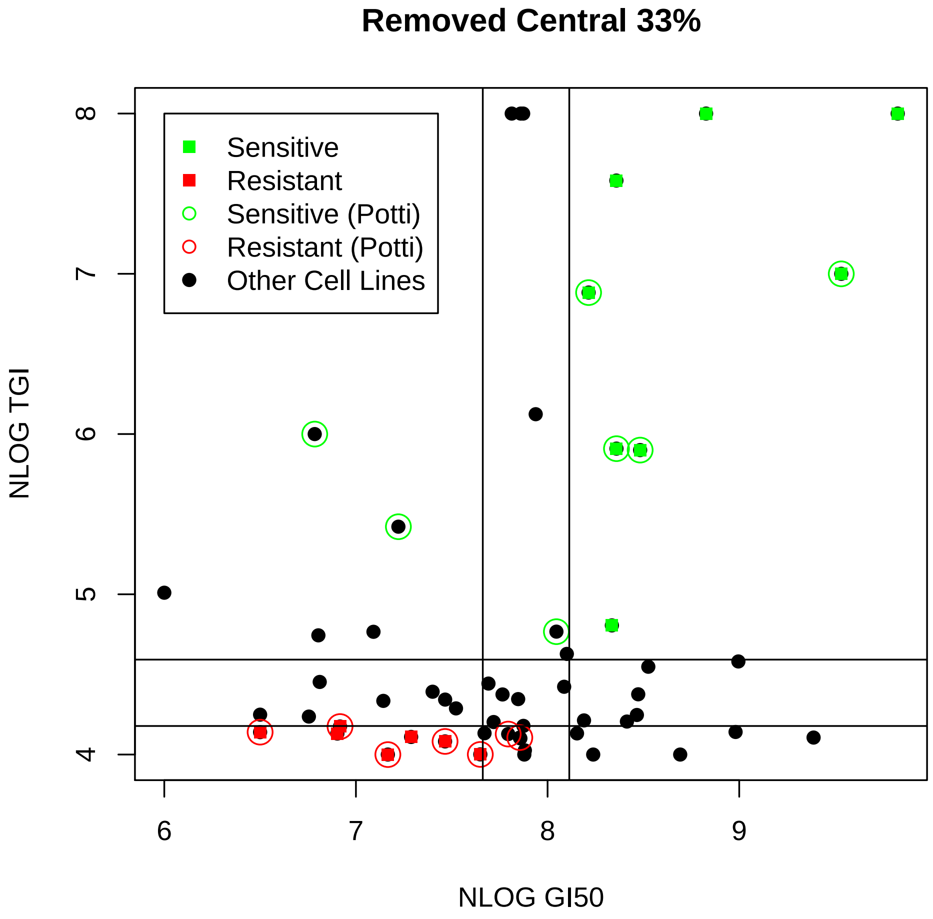

Figure 4: Scatter plot of the negative base-ten logarithm of the GI50 and TGI values for 59 cell lines. The observed values were separated into thirds. Genes selected as sensitive (resp., resistant) were in the top (resp., bottom) third on both measures.

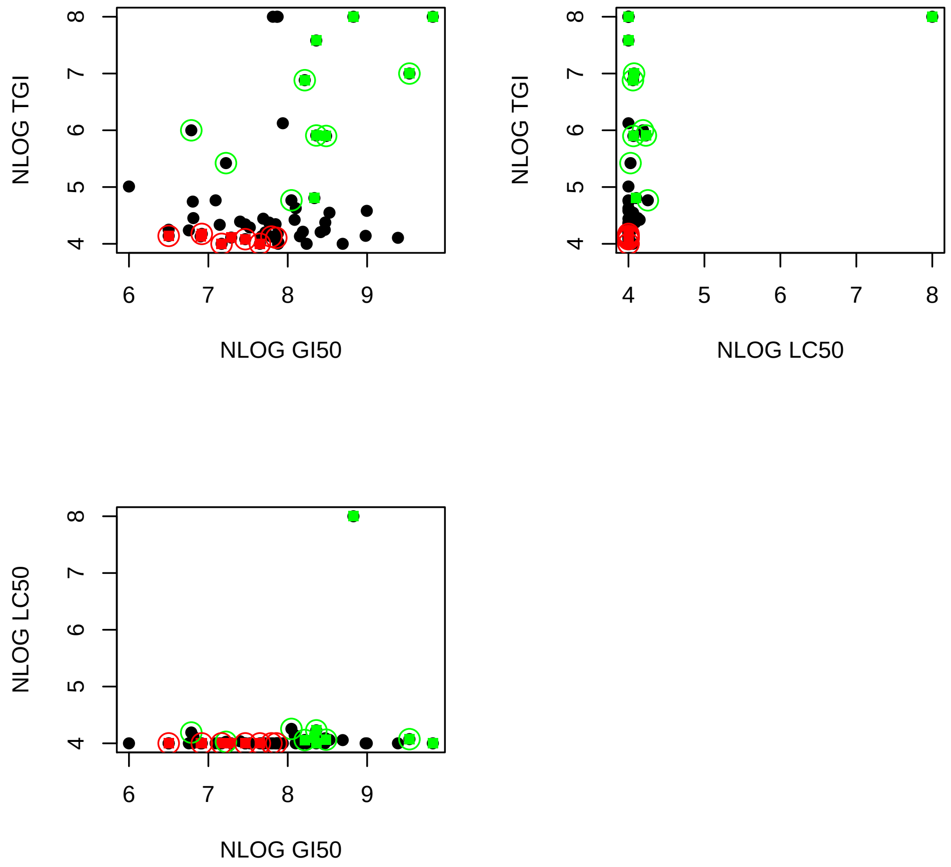

Figure 5: Pairwise plots of negative log GI50, TGI, and LC50 values.

```
> gsen <- cellLineNames(!is.na(doceGI50) & doceGI50 > m +
+     s)
> gres <- cellLineNames(!is.na(doceGI50) & doceGI50 < m -
+     s)
> gres
```

```
[1] "ACHN"        "CAKI-1"      "HOP-92"      "MALME-3M"
[5] "NCI/ADR-RES" "OVCAR-4"     "SK-MEL-2"    "UACC-257"
[9] "UO-31"
```

```
> gsen
```

```
[1] "COLO 205"   "HCT-116"    "HS 578T"    "MCF7"       "MDA-MB-435"
[6] "NCI-H460"   "NCI-H522"
```

Well, that is a substantial number of cell lines, but they do not look familiar. How many of the lines on their list are also here?

```
> pottiCellnameDoceResi[which(pottiCellnameDoceResi %in%
+     gres)]
```

```
[1] "OVCAR-4" "CAKI-1"
```

```
> pottiCellnameDoceSens[which(pottiCellnameDoceSens %in%
+     gsen)]
```

```
[1] "NCI-H522"
```

Not very many.

6.3 Potti Rule

In fact, Potti and colleagues say that the cell lines also have to satisfy similar criteria for LC50 and TGI. The next function combines the “mean ± one SD” rule for GI50 with a simpler above/below the median rule for LC50 and TGI.

```
> pottiRule <- function(i, strict = FALSE) {
+     if (!i %in% c(4, 6, 7))
+         stop("The only valid starting concentrations are 4, 6, and 7")
+     expt <- paste(doceNSC, "L-", i, sep = "")
+     gi50 <- GI50[, expt]
+     gm <- mean(gi50, na.rm = TRUE)
+     gs <- sd(gi50, na.rm = TRUE)
+     tgi <- TGI[, expt]
+     tm <- median(tgi, na.rm = TRUE)
+     lc50 <- LC50[, expt]
+     lm <- median(lc50, na.rm = TRUE)
+     valid <- !is.na(gi50) & !is.na(tgi) & !is.na(lc50)
+     if (strict) {
+         sens <- which(valid & (gi50 > gm + gs) & (tgi >
+             tm) & (lc50 > lm))
+         resi <- which(valid & (gi50 < gm - gs) & (tgi <
+             tm) & (lc50 < lm))
+     }
+     else {
+         sens <- which(valid & (gi50 > gm + gs) & (tgi >=
+             tm) & (lc50 >= lm))
+         resi <- which(valid & (gi50 < gm - gs) & (tgi <=
+             tm) & (lc50 <= lm))
+     }
+     list(sens = sens, resi = resi)
+ }
```

Using the "-4" starting concentration, we get no cell lines that are sensitive:

```
> pottiRule(4)

$sens
integer(0)

$resi
     ACHN      EKVX    HOP-92  NCI-H226    OVCAR-4 SK-MEL-28    SNB-19
        4        10        16        29         36        46        50
 UACC-257
       57

> pottiRule(4, strict = TRUE)

$sens
integer(0)

$resi
integer(0)
```

Using the "-6" starting concentration, we get more lines, but most of them have values equal to the median (which is a truncation level), as we see by changing to a strict inequality:

```
> pottiRule(6)
```

```
$sens
COLO 205  HS 578T     MCF7 NCI-H522
       8       17       25       33

$resi
       ACHN      CAKI-1      HOP-92    MALME-3M NCI/ADR-RES
          4           6          16          24          34
    OVCAR-4    SK-MEL-2    UACC-257       UO-31
         36          45          57          59

> pottiRule(6, strict = TRUE)

$sens
COLO 205
       8

$resi
integer(0)
```

The same thing happens with the "-7" starting concentration:

```
> pottiRule(7)

$sens
 COLO 205 HL-60(TB)  LOX IMVI  NCI-H460  NCI-H522      U251
        8        14        22        32        33        56

$resi
    A498     ACHN   CAKI-1     EKVX   HCT-15   HOP-92 NCI-H226
       2        4        6       10       13       16       29
 OVCAR-4  OVCAR-5   SNB-19 UACC-257    UO-31
      36       37       50       57       59

> pottiRule(7, strict = TRUE)

$sens
integer(0)

$resi
integer(0)
```

Clearly, they did not actually do what they said in the methods section.

7 Wrapup

Finally, we save the items from this analysis that we might want to use elsewhere.

```
> rm(m, s, gsen, gres)
```

```
> save(cellLineNames, doceNSC, doceGI50, doceLC50, doceTGI,
+     quantCut, ourCellnameDoceSens, ourCellnameDoceResi,
+     pottiCellnameDoceSens, pottiCellnameDoceResi, file = file.path("RDataObjects",
+         "doceGI50.Rda"))
```