Examining Predictions for Docetaxel in Detail

Keith A. Baggerly

September 24, 2009

Contents

1 Executive Summary

1.1 Introduction

In late 2006, Potti et al. [3] introduced a method for combining microarray profiles of cell lines with drug sensitivity data to derive "signatures" of sensitivity to specific drugs. These signatures could then be used to predict patient response. In theory, the approach is straightforward:

- Using drug sensitivity data for a panel of cell lines, select those that are most sensitive and most resistant to the drug of interest.
- Using array profiles of the identified cell lines, identify the most differentially expressed genes.
- Using the most differentially expressed genes, build a model that takes an array profile and returns a classification.

This report is part of a series in which we try to trace the specific steps involved in order to better understand the approach. In this report, we focus on response to docetaxel, which is the single drug described in most detail. It is empirically also one of the most confusing. In examining an independent test set supplied by Chang et al. [1], Potti et al. [3] reported correctly classifying 22/24 cases. Coombes et al. [2] attempted to reproduce this, but in the end reported results no better than chance. In reply, however, Potti and Nevins [4] claim that

> when Coombes et al. compared the results of models that create metagenes from training data alone to the more extensive model that creates metagenes with both training and test data, they obtained a very similar result to ours (Fig.8 in Supplementary Report 9). In short, they reproduce our result when they use our methods.

Potti and Nevins [4] also mention that more explicit details are now available on their website.

1.2 Methods

We acquired the Chang et al. [1] array data from GEO, and information giving sensitive/resistant status and selected gene expression values for each sample from the Chang et al. [1] supplementary table. We acquired two tables from the Potti et al. [3] web site, `http://data.genome.duke.edu/NatureMedicine.php`, at the time of the first correction in November 2007. The first, ChangClinicalData.doc, lists the Chang et al. [1] clinical data with some additions. The second, BreastData.txt, gives docetaxel quantification data for both cell lines and test samples together with sensitive and resistant labels for each. Potti et al. [5] posted a second correction in August of 2008. While this correction dealt primarily with doxorubicin, the posted clinical data for docetaxel was revised and the quantification data was removed. We acquired the revised clinical table, Chang_Docetaxel_Data1.doc.

We checked the clinical data from Chang et al. [1] against that provided in ChangClinicalData.doc to confirm we were working with the same data.

In buildRda.changAll, we used correlation between the selected expression values in the Chang et al. [1] supplementary table and the GEO array data to establish the mapping between the sample identifiers used

(a) in Chang et al. [1] and (b) at GEO. We then used correlations between the array data at GEO and the quantification data posted by Potti et al. [4] (suitably transformed) to identify the sample mapping used. We later checked the mapping used by Potti et al. [4] against that reported in Chang_Docetaxel_Data1.doc.

Using "the more extensive model that creates metagenes with both training and test data" and the parameter settings posted by Potti et al. [3], we predicted response status for the 24 samples from Chang et al. [1]. Since the fitting process involves a stochastic Markov chain Monte Carlo approach, we reran the fit 100 times to assess the variability associated with these predictions. We also ran 1000 "null simulations" in which sets of cell lines were picked at random from the data available to see how accurate our predictions could be expected to be in the absence of explicit structure. For each null simulation we recorded (a) the sets of cell lines used, and (b) the number of test samples accurately called. We also repeated this approach with the parameter settings used by Coombes et al. [2] in their SR9, which differ from those reported by Potti et al. [4], and using the Potti et al. [3] parameters after centering and scaling the training and test results separately.

1.3 Results

The clinical data posted by Potti et al. [3] is slightly different than that posted by Chang et al. [1]; the percent residual tumor for sample N14 is listed as 40 by Potti et al. [3], but 39 by Chang et al. [1]. This is relevant because Potti et al. [3] chose to use a cutoff of 40% for designating samples as sensitive or resistant; Chang et al. [1] used 25%. Potti et al. [3] change the Chang et al. [1] classifications for two cases (N12 and N13), but N14 should change as well.

In the array quantification data posted by Potti et al. [3], sensitive/resistant status is reversed for the cell lines, which should reverse the predictions obtained. The mapping Potti et al. [3] used for the test samples is shown in Figure 1. This mapping mislabels 6 resistant samples as sensitive and 3 sensitive samples as resistant before the fitting program is run. There is no entry for sample GSM4910; sample GSM4914 is listed twice, once as sensitive and once as resistant.

Our 100 data fits using the cell lines and parameter settings supplied by Potti et al. [3] predicted the same 11/24 cases correctly (5 sensitive, 6 resistant), as shown in Figure 1. in all 100 simulations. Using the null simulations as a baseline, the empirical p-value is 0.615. The 100 data fits using the parameter settings from Coombes et al. [2] are considerably more variable, and some sample calls change from one run to the next. The median accuracy in the 100 simulations was 12/24. Using the corresponding null simulations as a baseline, the p-value was 0.526. Using the Potti et al. [3] parameters after centering and scaling training and test data separately, the 100 data fits have an accuracy of 13/24.

1.4 Conclusions

Judging from the quantifications posted by Potti et al. [4], the training data labels were reversed and roughly half of the test data used was mislabeled before predictions were made. When the "more extensive model" is fit to correctly labeled data with the parameter values reported, an accuracy of 11/24 is obtained. Based on our null simulations, the empirical p-value associated with this result is 0.615, suggesting that these results are no better than might be expected due to chance alone. Our results do not agree with the assertion made by Potti et al. [4] that Coombes et al. [2] "get our results when they use our methods". Rather, they show that when the data are correctly labeled, using the Potti et al. [3] methods produces results no better than chance.

2 Options and Libraries

```
> options(width = 80)
```

3 Loading Data

Where necessary, we have converted the tables listed below to csv files for easier loading.

3.1 Earlier Rda Files: NCI60 Cell Line Data, Chang Clinical Data, Chang Array Data, Lists of Cell Lines Used

Here, we simply load RData objects assembled earlier for the U95Av2 NCI60 quantification data (novartisA), the Chang et al. [1] clinical data and supplementary information (changSuppAndTable), the Chang et al. [1] array quantifications from GEO, with names mapped to those used in the clinical information tables (changAll), and the lists of cell lines used (cellLinesUsed, built in enumeratingCellLines).

```
> rdaList <- c("novartisA", "changSuppAndTable", "changAll", "cellLinesUsed")
> for (rdaFile in rdaList) {
+     rdaFullFile <- file.path("RDataObjects", paste(rdaFile, "Rda",
+         sep = "."))
+     if (file.exists(rdaFullFile)) {
+         cat("loading ", rdaFullFile, " from cache\n")
+         load(rdaFullFile)
+     }
+     else {
+         cat("building ", rdaFullFile, " from raw data\n")
+         Stangle(file.path("RNowebSource", paste("buildRda", rdaFile,
+             "Rnw", sep = ".")))
+         source(paste("buildRda", rdaFile, "R", sep = "."))
+     }
+ }
```

```
loading  RDataObjects/novartisA.Rda  from cache
loading  RDataObjects/changSuppAndTable.Rda  from cache
loading  RDataObjects/changAll.Rda  from cache
loading  RDataObjects/cellLinesUsed.Rda  from cache
```

3.2 Docetaxel 2007 Clinical Data: ChangClinicalData.doc

The first table of clinical information for docetaxel supplied by Potti et al. [4] was ChangClinicalData.doc.

```
> changClinicalPotti07 <- read.table(file.path("RawData", "PottiNatMed",
+     "changClinicalDataPotti07.csv"), sep = ",", header = TRUE)
> dim(changClinicalPotti07)
```

```
[1] 24 12
```

```
> changClinicalPotti07[1:2, ]
```

```
  Patient Age..years. Menopausal.status Ethnic.origin
1       1          37     Premenopausal      Hispanic
2       2          55    Postmenopausal      Hispanic
  Bidimensional.tumour.size..cm. Clinical.axillary.nodes
1                         10x10                      No
```

```
2                          10x8                       Yes
  Oestrogen..receptor.status Progesterone..receptor.status HER.2 Tumour.type
1                          -                             -     -         IMC
2                          -                             -     +         IDC
  Percent.Residual.Tumor Sens.or.Res..S.is.less.than.40pct.
1                      1                                 S
2                      1                                 S
```

The data appears to be a combination of the clinical data supplied in Table 1 of Chang et al. [1] and the Supplementary Table of Chang et al. [1]

3.3 Docetaxel 2007 Quantifications: BreastData.txt

Of the files available on the Potti et al. [3] web site (`http://data.genome.duke.edu/NatureMedicine.php`) as of November 6, 2007, one, "BreastData.txt" involves the test samples for docetaxel. This file contains processed data for 38 samples: 14 training samples and 24 validation samples. A note to the side (in position AO3 when the table is read into Excel) states that "Data is standardized due to differences in mean signal intensities of training data and validations". We trimmed off this comment to get a more consistently formatted table for easier loading (the untrimmed table was saved as BreastDataOriginal.txt). There are two header lines. Row 1 indicates whether the column is training (Doce) or testing (Chang). Row 2 indicates whether the column is Sensitive or Resistant.

```
> tempDocetaxel07Header1 <- read.table(file.path("RawData", "PottiNatMed",
+     "BreastData.txt"), sep = "\t", nrows = 1, header = FALSE)
> tempDocetaxel07Header1 <- as.vector(t(tempDocetaxel07Header1))
> tempDocetaxel07Header2 <- read.table(file.path("RawData", "PottiNatMed",
+     "BreastData.txt"), sep = "\t", skip = 1, nrows = 1, header = FALSE)
> tempDocetaxel07Header2 <- as.vector(t(tempDocetaxel07Header2))
> tempDocetaxel07Header1

 [1] "Docetaxel 0" "0"           "0"           "0"           "0"
 [6] "0"           "0"           "1"           "1"           "1"
[11] "1"           "1"           "1"           "Docetaxel1"  "Chang2"
[16] "2"           "2"           "2"           "2"           "2"
[21] "2"           "2"           "2"           "2"           "2"
[26] "2"           "2"           "2"           "2"           "2"
[31] "2"           "2"           "2"           "2"           "2"
[36] "2"           "2"           "Chang2"

> tempDocetaxel07Header2

 [1] "Sensitive" "Sensitive" "Sensitive" "Sensitive" "Sensitive" "Sensitive"
 [7] "Sensitive" "Resistant" "Resistant" "Resistant" "Resistant" "Resistant"
[13] "Resistant" "Resistant" "Resistant" "Resistant" "Resistant" "Resistant"
[19] "Resistant" "Resistant" "Resistant" "Resistant" "Resistant" "Resistant"
[25] "Resistant" "Sensitive" "Sensitive" "Sensitive" "Sensitive" "Sensitive"
[31] "Sensitive" "Sensitive" "Sensitive" "Sensitive" "Sensitive" "Sensitive"
[37] "Sensitive" "Sensitive"

> table(tempDocetaxel07Header1[1:14], tempDocetaxel07Header2[1:14])
```

```
             Resistant Sensitive
  0                  0         6
  1                  6         0
  Docetaxel 0        0         1
  Docetaxel1         1         0
```

```
> table(tempDocetaxel07Header1[15:38], tempDocetaxel07Header2[15:38])
```

```
         Resistant Sensitive
  2             10        12
  Chang2         1         1
```

In the training set, 0 denotes sensitive and 1 denotes resistant. There are 7 lines labeled as resistant, and 7 lines labeled as sensitive. The test (Chang) data is all labeled 2, regardless of status. The test data contains 11 Resistant samples and 13 Sensitive samples, matching the numbers reported by Potti et al. [3]

We now name the samples and assemble the sample information.

```
> tempSampleNames <- c(paste("Training", c(1:14), sep = ""), paste("Test",
+     c(1:24), sep = ""))
> tempGroup <- c(rep("Training", 14), rep("Test", 24))
> tempStatus <- tempDocetaxel07Header2
> docetaxel07Info <- data.frame(sampleGroup = tempGroup, status = tempStatus,
+     row.names = tempSampleNames)
> docetaxel07Info[c(1:2, 14:16), ]
```

```
           sampleGroup    status
Training1     Training Sensitive
Training2     Training Sensitive
Training14    Training Resistant
Test1             Test Resistant
Test2             Test Resistant
```

```
> rm(list = ls(pattern = "^temp"))
```

Finally, we load the numerical quantifications.

```
> docetaxel07Numbers <- read.table(file.path("RawData", "PottiNatMed",
+     "BreastData.txt"), sep = "\t", skip = 2, header = FALSE)
> colnames(docetaxel07Numbers) <- rownames(docetaxel07Info)
> docetaxel07Numbers[1:4, c(1:2, 14:16)]
```

```
  Training1 Training2 Training14 Test1 Test2
1      0.54      2.21       3.92  1.00  1.43
2      0.29      4.95       0.63  0.43  0.67
3      5.39      1.83       1.15  1.48  1.45
4      1.92      0.31       5.33  3.52  1.23
```

3.4 Docetaxel 2008 Clinical Data: Chang_Docetaxel_Data1.doc

The clinical information for docetaxel supplied by Potti et al. [5] was Chang_Docetaxel_Data1.doc.

```
> changClinicalPotti08 <- read.table(file.path("RawData", "PottiNatMed",
+     "changClinicalDataPotti08.csv"), sep = ",", header = TRUE,
+     nrows = 24)
> dim(changClinicalPotti08)
```

```
[1] 24 13
```

```
> changClinicalPotti08[1:2, ]
```

```
  Patient Age..years. Menopausal.status Ethnic.origin
1       1          37     Premenopausal      Hispanic
2       2          55    Postmenopausal      Hispanic
  Bidimensional.tumour.size..cm. Clinical.axillary.nodes
1                          10x10                      No
2                           10x8                     Yes
  Oestrogen..receptor.status Progesterone..receptor.status HER.2 Tumour.type
1                          -                             -     -         IMC
2                          -                             -     +         IDC
  Percent.Residual.Tumor Sens.or.Res..S.is.less.than.40pct. GEO.Identifier
1                      1                                 S        GSM4903
2                      1                                 S        GSM4907
```

This is simply an expansion of their earlier table to include a column for GEO identifier.

4 Comparing the Clinical Information for Docetaxel

We want to check that the clinical data supplied Chang et al. [1] agrees with that supplied by Potti et al. [4] We begin with the first 10 columns of ChangClinicalData.doc, which seem to match Table 1 from Chang et al. [1]

```
> all(changTable1 == changClinicalPotti07[, 1:10])
```

```
[1] TRUE
```

This match is exact. Next we turn to the last 2 columns of ChangClinicalData.doc, which seem to match the clinical information from from the Chang et al. [1] supplementary table. We expect to see some disagreements in the Sensitive/Resistant labeling, since ChangClinicalData.doc notes that they changed the cutoff from the 25% residual disease used by Chang et al. [1] to 40%, shifting two samples from Resistant to Sensitive in the process.

```
> tempBadRows <- which(as.character(changClinicalPotti07[, "Sens.or.Res..S.is.less.than.40pct."]) !=
+     substr(as.character(changSuppClinical[, "Status"]), 1, 1))
> changClinicalPotti07[tempBadRows, "Sens.or.Res..S.is.less.than.40pct."]
```

```
[1] S* S*
Levels: R S S*
```

```
> changSuppClinical[tempBadRows, ]
```

```
    PercentResidualTumor    Status
N12                   36 Resistant
N13                   38 Resistant
```

We see the two changes we expected. By contrast, we expect all of the percent residual tumor values to agree across tables. Unexpectedly, we see a discrepancy.

```
> which(changClinicalPotti07[, "Percent.Residual.Tumor"] != changSuppClinical[,
+     "PercentResidualTumor"])

[1] 14

> changClinicalPotti07[13:15, c("Patient", "Percent.Residual.Tumor",
+     "Sens.or.Res..S.is.less.than.40pct.")]

   Patient Percent.Residual.Tumor Sens.or.Res..S.is.less.than.40pct.
13      13                    38                                 S*
14      14                    40                                  R
15      15                    44                                  R

> changSuppClinical[13:15, ]

    PercentResidualTumor    Status
N13                   38 Resistant
N14                   39 Resistant
N15                   44 Resistant
```

The percent residual tumor for patient 14 has changed from 39% (Chang) to 40% (Potti). This has an effect on classification, as Potti et al. [4] set 40% as the cutoff for the sensitive/resistant divide. Using a 40% cutoff with the numbers from Chang et al. [1] identifies 14 patients as sensitive, not 13.

5 Identifying the Cell Lines Used in BreastData.txt

5.1 Getting the NCI60 Quantifications

We begin by extracting the NCI60 array quantifications for the cell lines listed by Potti et al. [3] for docetaxel, using the orientation from the "cell lines in each chemo predictor" file posted on the Potti et al. [3] web site as of August 2008. First, we list the lines involved.

```
> cellLinesUsed[["docetaxel"]][["listPotti06CorrAug08"]]

$Sensitive
[1] "HL-60(TB)" "SF-539"    "HT29"      "HOP-62"    "SK-MEL-2"  "SK-MEL-5"
[7] "NCI-H522"

$Resistant
[1] "EKVX"    "IGROV1"  "OVCAR-4" "786-0"   "CAKI-1"  "SN12C"   "TK-10"
```

There are 7 lines in each group, so this gives no information about accuracy. We put the resistant group first.

```
> docetaxelNCI60Quants <-
+   novartisA[,c(cellLinesUsed[["docetaxel"]][[
+                              "listPotti06CorrAug08"]][["Resistant"]],
+                cellLinesUsed[["docetaxel"]][[
+                              "listPotti06CorrAug08"]][["Sensitive"]])]
> docetaxelNCI60Quants <-
+   docetaxelNCI60Quants[-grep("^AFFX",rownames(docetaxelNCI60Quants)),]
```

5.2 Transforming the BreastData.txt Data

The comment in cell AO3 about standardizing the data suggests treating the training and test data separately, with the goal of scaling the data to have mean 0 and variance 1.

```
> range(docetaxel07Numbers[, 1:14])
```

```
[1]  0.04 29.86
```

All of the reported values are positive, so they aren't standardized on the scale supplied. Given the positivity, we'll try log-transforming.

```
> mean(log(t(docetaxel07Numbers[1, 1:14])))
```

```
[1] -0.0004105719
```

```
> var(log(t(docetaxel07Numbers[1, 1:14])))
```

```
          1
1 0.9997468
```

The proximity to (0,1) suggests that the training data values were log-transformed, centered and scaled row by row, and then exponentiated to give the data reported. The data were truncated at some step, since values are only reported to two decimal places. We now try matching all the data.

```
> tempDoceStandardized <- exp(t(scale(t(log(docetaxelNCI60Quants)))))
> summary(apply(abs(tempDoceStandardized - docetaxel07Numbers[,
+     1:14]), 1, max))
```

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
0.002524 0.004529 0.004756 0.004666 0.004897 0.005000
```

This is a perfect match, given that the tabulated data has been truncated to two decimal places. The fact that the max absolute deviation is 0.005 suggests that truncation occurred was the last step.

5.3 Checking Direction

There is one difficulty with the training data. We put the resistant NCI60 lines first, and the sensitive lines last.

```
> cbind(docetaxel07Info[1:14, ], colnames(docetaxelNCI60Quants))
```

```
           sampleGroup    status colnames(docetaxelNCI60Quants)
Training1     Training Sensitive                          EKVX
Training2     Training Sensitive                        IGROV1
Training3     Training Sensitive                       OVCAR-4
Training4     Training Sensitive                         786-0
Training5     Training Sensitive                        CAKI-1
Training6     Training Sensitive                         SN12C
Training7     Training Sensitive                         TK-10
Training8     Training Resistant                     HL-60(TB)
Training9     Training Resistant                        SF-539
Training10    Training Resistant                          HT29
Training11    Training Resistant                        HOP-62
Training12    Training Resistant                       SK-MEL-2
Training13    Training Resistant                       SK-MEL-5
Training14    Training Resistant                      NCI-H522
```

The status labeling given in the BreastData.txt file is reversed.

6 Examining the BreastData.txt Test Data

We now want to match the test data in BreastData.txt with the Chang et al. [1] data posted at GEO. To do this, we first transform the GEO data for the relevant probesets in the manner outlined above for the training data, and then check the correlation between columns.

6.1 Standardizing the Data from GEO

We begin by standardizing the data from GEO.

```
> changGEOStandardized <- round(exp(t(scale(t(log(changAll[rownames(docetaxelNCI60Quants),
+     ])))))), 2)
> sum(is.na(changGEOStandardized))

[1] 240

> tempBadRows <- unique(which(is.na(changGEOStandardized), arr.ind = TRUE)[,
+     "row"])
> tempBadRows

 [1]  1116  1325  1691  1962  2464  3910  6691  7408 11482 12018
```

There is a problem in that 10 rows of data do not convert properly; they give NAs.

```
> summary(as.vector(changAll[rownames(changGEOStandardized)[tempBadRows],
+     ]))

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  5.898   5.898   5.898   5.898   5.898   5.898

> summary(unlist(docetaxel07Numbers[tempBadRows, 15:38]))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   2.66    2.66    2.66    2.66    2.66    2.66
```

Looking more closely at the raw numbers, we see that they all hit the lower bound of the values reported – dChip effectively called all of these values "absent". The code for standardizing is breaking when it attempts to estimate the variance of a set of tied values. Checking the corresponding values in the reported data, however, shows that these values are also all the same (2.66), so we can simply replace the NA values with 2.66 and proceed.

```
> changGEOStandardized[is.na(changGEOStandardized)] <- 2.66
```

6.2 Checking Correlations

We now compute pairwise correlations.

```
> pottiGEOCorrs <- cor(docetaxel07Numbers[, 15:38], changGEOStandardized)
> pottiGEOCorrs[1:3, 1:3]
```

```
               N1          N2          N3
Test1 -0.06790501 -0.08994123 -0.05396590
Test2 -0.05988653 -0.02261227  0.04369349
Test3 -0.04238540 -0.03694776  0.02446569
```

```
> summary(pottiGEOCorrs[, 1])
```

```
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-0.105000 -0.067620 -0.027080  0.016800  0.005872  1.000000
```

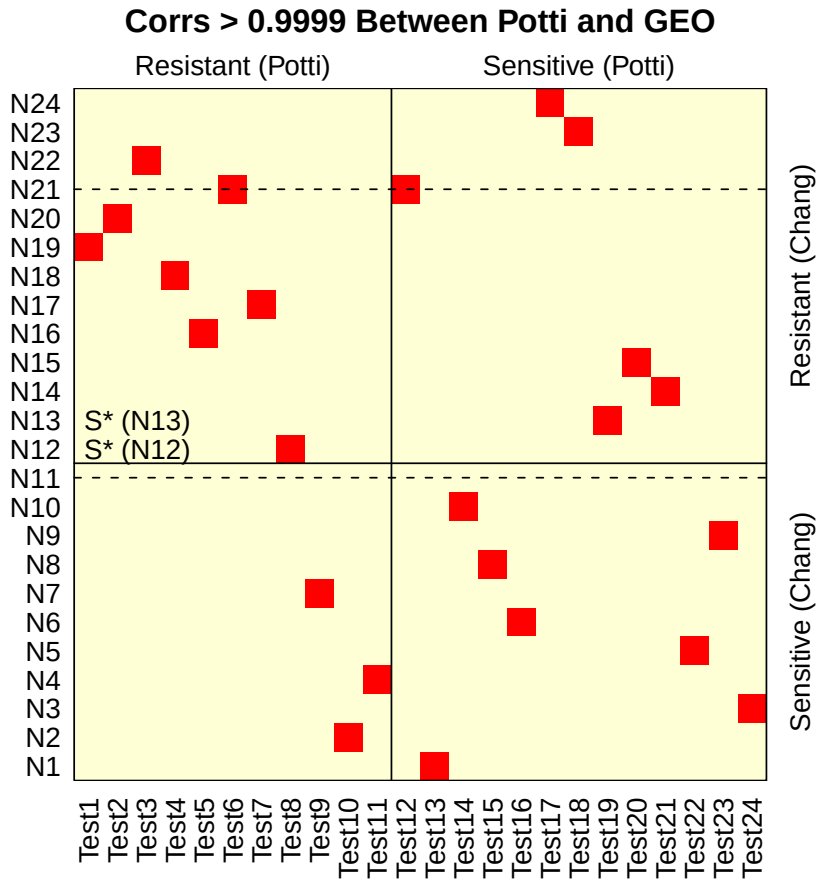
```
> sum(pottiGEOCorrs > 0.9999)
```

```
[1] 24
```

The number of essentially perfect "hits" above looks promising. Now let's take a look at them spatially.

```
> image(1:24, 1:24, pottiGEOCorrs < 0.9999, axes = FALSE, xlab = "",
+     ylab = "", asp = 1, main = "Corrs > 0.9999 Between Potti and GEO")
> lines(c(0.5, 24.5), c(11.5, 11.5))
> lines(c(11.5, 11.5), c(0.5, 24.5))
> rect(0.5, 0.5, 24.5, 24.5)
> axis(1, at = 1:24, labels = rownames(pottiGEOCorrs), las = 2,
+     line = -0.5, tick = 0)
> axis(2, at = 1:24, labels = colnames(pottiGEOCorrs), las = 2,
+     line = -2.1, tick = 0)
> lines(c(0.5, 24.5), c(11, 11), lty = "dashed")
> lines(c(0.5, 24.5), c(21, 21), lty = "dashed")
> axis(3, at = c(6, 18), labels = c("Resistant (Potti)", "Sensitive (Potti)"),
+     tick = 0, line = -0.8)
> axis(4, at = c(6, 18), labels = c("Sensitive (Chang)", "Resistant (Chang)"),
+     tick = 0, line = -2)
> axis(2, at = c(12, 13), labels = c("S* (N12)", "S* (N13)"), las = 2,
+     line = -6, tick = 0)
> changAllInfo[c("N21", "N11"), 1:4]
```

```
      geoID geoTitle    status pctResidTumor
N21 GSM4910      358 Resistant            70
N11 GSM4914      413 Sensitive            25
```



There are quite a few problems. One Chang sample (N21, GSM4910) is included twice, and labeled both sensitive and resistant. Another (N11, GSM4914) is omitted entirely. The two samples that Potti et al. [4] explicitly mention relabeling as sensitive (N12 and N13) appear with one labeled sensitive and the other resistant. In all, we count 10 samples mislabeled before fitting begins – the points in the lower left (3 samples), upper right (6 samples, including the one present twice), and the sample omitted.

7 Checking the 2008 Mapping

As of the second correction in August 2008, the BreastData.txt quantification file was removed, and the clinical data table was extended to include a mapping to GEO. Here, we check the agreement between the mapping we have inferred and the mapping now reported.

```
> cbind(changClinicalPotti08[, c("Percent.Residual.Tumor", "GEO.Identifier")],
+     changAllInfo[, "geoID"])
```

```
    Percent.Residual.Tumor GEO.Identifier changAllInfo[, "geoID"]
N1                       1        GSM4903                 GSM4917
N2                       1        GSM4907                 GSM4919
N3                       6        GSM4908                 GSM4908
N4                       6        GSM4912                 GSM4915
N5                      13        GSM4913                 GSM4903
N6                      14        GSM4914                 GSM4923
N7                      16        GSM4915                 GSM4913
N8                      17        GSM4917                 GSM4921
N9                      18        GSM4918                 GSM4907
N10                     22        GSM4919                 GSM4920
N11                     25        GSM4920                 GSM4914
N12                     36        GSM4921                 GSM4912
N13                     38        GSM4923                 GSM4918
N14                     40        GSM4901                 GSM4924
N15                     44        GSM4902                 GSM4922
N16                     45        GSM4904                 GSM4909
N17                     47        GSM4905                 GSM4911
N18                     60        GSM4906                 GSM4906
N19                     64        GSM4909                 GSM4901
N20                     65        GSM4910                 GSM4902
N21                     70        GSM4911                 GSM4910
N22                    100        GSM4916                 GSM4904
N23                    100        GSM4922                 GSM4916
N24                    131        GSM4924                 GSM4905
```

```
> sum(as.character(changClinicalPotti08$GEO.Identifier) == as.character(changAllInfo[,
+     "geoID"]))
```

```
[1] 2
```

```
> sum(sort(as.character(changClinicalPotti08$GEO.Identifier)[1:13]) ==
+     sort(as.character(changAllInfo[1:13, "geoID"])))
```

```
[1] 13
```

```
> sum(sort(as.character(changClinicalPotti08$GEO.Identifier)[14:24]) ==
+     sort(as.character(changAllInfo[14:24, "geoID"])))
```

```
[1] 11
```

In the revised mapping, all of the GEO samples are represented; there are no ties and there are no samples used more than once. Further, the first 13 identifiers in the two lists agree in terms of the samples involved – all of the samples that Potti et al. [4] as sensitive do indeed correspond to the 13 samples with the smallest percent residual tumor values. However, the precise one-to-one mapping is deeply flawed – only 2 of the 24 sample mappings are correct.

8 Running Binreg

Potti et al. [3] reported correctly classifying 22 of the 24 Chang et al. [1] cases. Coombes et al. [2] attempted to reproduce this, but in the end reported results no better than chance. Much of the Coombes et al. [2] effort focused on results obtained when a model constructed on the training set is used to make predictions on a wholly independent test set. The binreg software used by Potti et al. [3], however, constructs metagenes using a singular value decompostion of the training and test set together, so that the model used is not independent of the test set. This may be important, as Potti and Nevins [4] claim that

> when Coombes et al. compared the results of models that create metagenes from training data alone to the more extensive model that creates metagenes with both training and test data, they obtained a very similar result to ours (Fig.8 in Supplementary Report 9). In short, they reproduce our result when they use our methods.

In this section, we check this assertion by running their software using their "more extensive model" under four different sets of conditions. First, we run the model for docetaxel using their chosen cell lines and the Chang et al. [1] test data, though using a correct labeling throughout: we orient the cell lines so that EKVX is resistant, and we use the sensitive/resistant labels supplied by Chang et al. [1] Since there is some stochasticity in the modeling, we run these fits 100 times. Second, in order to get a better feel for the distribution of results using the extensive model in the absence of structure, we kept the Chang et al. [1] test data and model parameter settings as above, but chose 7 sensitive and 7 resistant lines at random and wihtout replacement from the set of 59 NCI60 array profiles available. We repeated this 1000 times, recording the number of test samples "correctly" predicted each time. Third and fourth, we repeated the first two cases described above but used the parameter settings used by Coombes et al. [2] in their supplementary report 9. The Matlab scripts for the above simulations are in MatlabFiles folder under DocetaxelSimulations as DocetaxelNCI60.m (the first two cases) and DocetaxelNCI60_SR9.m (the last two cases). These scripts make use of the nci60_numbers.csv files produced in matchingHsuHeatmaps, and of the DoceTrainAndTest m-files produced in Supplementary Report SR9 of Coombes et al. [2] The simulation results are saved in MatlabFiles under DocetaxelSimulations.

First, we read in the sample information matching the data arrangement used.

```
> tempInfoPart1 <- read.table(file.path("MatlabFiles", "DocetaxelSimulations",
+     "DoceTrainAndTestSampleInfo.csv"), sep = ",", header = TRUE,
+     nrows = 14)
> tempInfoPart2 <- read.table(file.path("MatlabFiles", "DocetaxelSimulations",
+     "DoceTrainAndTestSampleInfo.csv"), sep = ",", header = TRUE,
+     skip = 15)
> tempInfoPart1[1:3, ]

    index drugName responseStatus  Source NovartisName
108   108     Doce      Resistant    EKVX       A.EKVX
109   109     Doce      Resistant  IGROV1     A.IGROV1
110   110     Doce      Resistant OVCAR-4    A.OVCAR-4

> tempInfoPart2[1:3, ]

   GEO.ID Response
1 GSM4903     Resp
2 GSM4907     Resp
3 GSM4908     Resp
```

```
> doceSimulationStatus <- c(as.character(tempInfoPart1$responseStatus),
+     as.character(tempInfoPart2$Response))
> names(doceSimulationStatus) <- c(as.character(tempInfoPart1$Source),
+     as.character(tempInfoPart2$GEO.ID))
```

8.1 Basic Case, 100 Runs, Potti et al. [4] Parameters

Now we read in the indices of the genes chosen and the predicted probabilities of response for each of the 100 "base case" simulations where we use the same cell lines, test samples, and parameters that Potti et al. [4] report.

```
> baseProbesetIndices <- read.table(file.path("MatlabFiles", "DocetaxelSimulations",
+     "baseProbesetIndices.csv"), header = FALSE, sep = ",")
> baseProbesetIndices <- as.matrix(baseProbesetIndices)
> baseProbesetIndices <- t(apply(baseProbesetIndices, 1, sort))
> rownames(baseProbesetIndices) <- paste("simulation", c(1:100),
+     sep = "")
> colnames(baseProbesetIndices) <- paste("geneIndex", c(1:50),
+     sep = "")
> baseProbesetIndices[1:3, 1:3]
```

```
            geneIndex1 geneIndex2 geneIndex3
simulation1         70        184        903
simulation2         70        184        903
simulation3         70        184        903
```

```
> all(baseProbesetIndices == matrix(rep(baseProbesetIndices["simulation1",
+     ], 100), ncol = 50, byrow = TRUE))
```

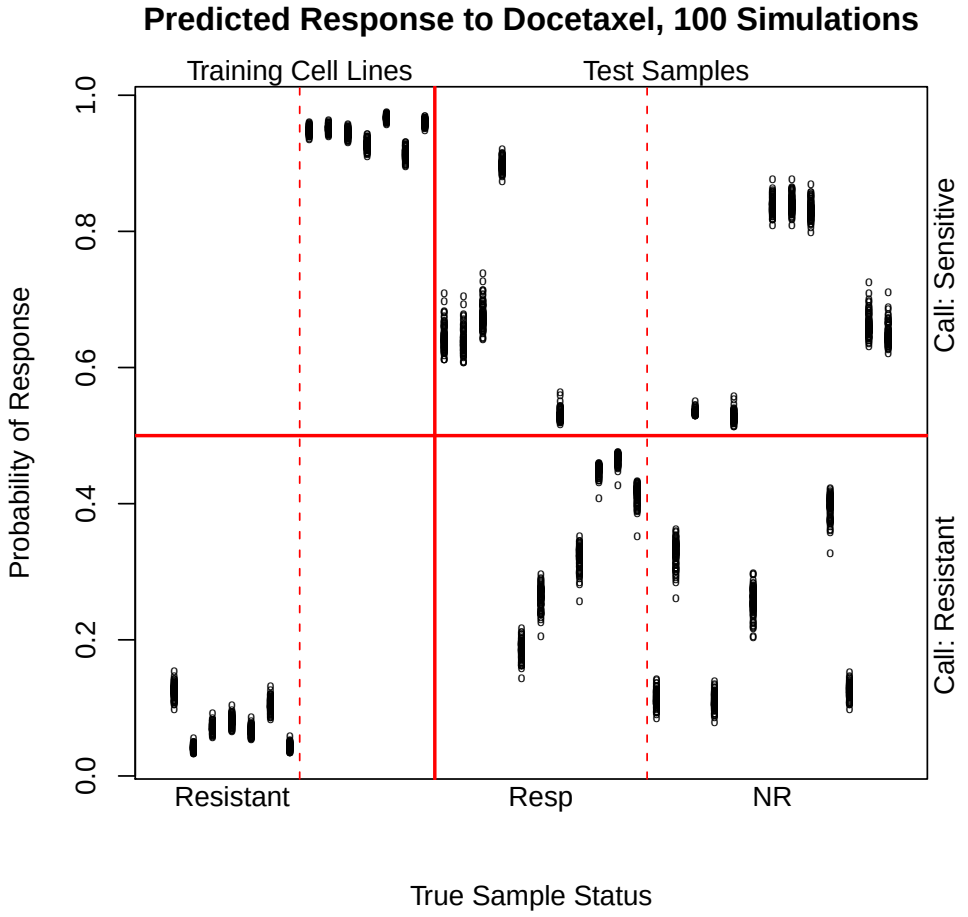
```
[1] TRUE
```

As expected, the same genes (gene indices) are chosen in every simulation. Now we look at the predictions.

```
> basePfitVectors <- read.table(file.path("MatlabFiles", "DocetaxelSimulations",
+     "basePfitVectors.csv"), header = FALSE, sep = ",")
> colnames(basePfitVectors) <- names(doceSimulationStatus)
> rownames(basePfitVectors) <- paste("simulation", c(1:100), sep = "")
```

Now we plot the probabilities of response to assess consistency and accuracy.

```
> matplot(t(basePfitVectors), pch = "o", col = "black", cex = 0.5,
+     xlim = c(0.5, 38.5), xaxt = "n", xlab = "True Sample Status",
+     ylab = "Probability of Response", main = "Predicted Response to Docetaxel, 100 Simulations")
> abline(h = 0.5, v = c(14.5), col = "red", lwd = 2)
> abline(v = c(7.5, 25.5), col = "red", lty = "dashed")
> axis(side = 1, at = c(4, 11, 20, 32), labels = c("Resistant",
+     "Sensitive", "Resp", "NR"), tick = 0, line = -1)
> axis(side = 3, at = c(7.5, 26.5), labels = c("Training Cell Lines",
+     "Test Samples"), tick = 0, line = -1)
> axis(side = 4, at = c(0.25, 0.75), labels = c("Call: Resistant",
+     "Call: Sensitive"), tick = FALSE, line = -1)
```



Classification of the training cell lines is perfect, as might be expected. Classification of the test samples, however, is more problematic.

```
> apply(basePfitVectors, 2, function(x) {
+     sum(x > 0.5)
+ })
```

```
     EKVX    IGROV1   OVCAR-4     786-0    CAKI-1     SN12C     TK-10 HL-60(TB)
        0         0         0         0         0         0         0       100
  SF-539      HT29    HOP-62 SK-MEL-2 SK-MEL-5 NCI-H522 GSM4903 GSM4907
      100       100       100       100       100       100       100       100
GSM4908 GSM4914 GSM4915 GSM4917 GSM4919 GSM4920 GSM4921 GSM4923
      100       100         0         0       100         0         0         0
GSM4913 GSM4901 GSM4902 GSM4904 GSM4905 GSM4906 GSM4909 GSM4910
        0         0         0       100         0       100         0       100
GSM4911 GSM4912 GSM4916 GSM4918 GSM4922 GSM4924
      100       100         0         0       100       100
```

```
> table(doceSimulationStatus, basePfitVectors[1, ] > 0.5)

doceSimulationStatus FALSE TRUE
           NR            6    7
           Resistant     7    0
           Resp          6    5
           Sensitive     0    7
```

While the classsifications are consistent across all 100 simulations, only 5 of the 11 responders and 6 of the 13 nonresponders are correctly predicted, for an accuracy of 11/24.

8.2 Null Simulations, 1000 Runs, Potti et al. [4] Parameters

In order to get a better feel for the accuracy we should expect in the absence of underlying structure, we ran 1000 "null" simulations in which the cell lines were chosen at random, and the number of "correct" classifications were counted. Since each cell line was equally likely to be labeled "sensitive" or "resistant", the distribution of accuracy values is clearly symmetric and centered at 12/24. To enable reproducibility, we have recorded the indices of the cell lines chosen in each simulation.

```
> nullCellLineIndices <- read.table(file.path("MatlabFiles", "DocetaxelSimulations",
+     "nullCellLineIndices.csv"), header = FALSE, sep = ",")
> nullCellLineIndices <- as.matrix(nullCellLineIndices)
> rownames(nullCellLineIndices) <- paste("nullSim", c(1:1000),
+     sep = "")
> colnames(nullCellLineIndices) <- c(paste("cellLineResIndex",
+     c(1:7), sep = ""), paste("cellLineSenIndex", c(1:7), sep = ""))
> nullCellLineIndices[1:3, 1:3]

         cellLineResIndex1 cellLineResIndex2 cellLineResIndex3
nullSim1                23                47                40
nullSim2                39                 6                48
nullSim3                39                31                44
```

Now we load the prediction values and compute the distribution of accuracy values seen.

```
> nullPfitVectors <- read.table(file.path("MatlabFiles", "DocetaxelSimulations",
+     "nullTrainAndTestPfitVectors.csv"), header = FALSE, sep = ",")
> colnames(nullPfitVectors) <- c(paste("nullRes", c(1:7), sep = ""),
+     paste("nullSim", c(1:7), sep = ""), paste("testResp", c(1:11),
+         sep = ""), paste("testNR", c(1:13), sep = ""))
> rownames(nullPfitVectors) <- paste("nullSim", c(1:1000), sep = "")
> nullNRespCorrect <- apply(nullPfitVectors[, 15:25], 1, function(x) {
+     sum(x > 0.5)
+ })
> nullNNRCorrect <- apply(nullPfitVectors[, 26:38], 1, function(x) {
+     sum(x < 0.5)
+ })
> nullNCorrect <- nullNRespCorrect + nullNNRCorrect
> table(nullNCorrect)
```

```
nullNCorrect
  1   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21
  1   6  16  32  45  61  56  68 100  69  80  85  83  89  68  52  42  27  13   5
 22
  2
```

```
> sum(nullNCorrect >= 11)
```

```
[1] 615
```

Overall, the p-value associated with the accuracy actually seen is about 0.615.

8.3 Basic Case, 100 Runs, Coombes et al. [2] Parameters

Now we read in the indices of the genes chosen and the predicted probabilities of response for each of the 100 "SR9 case" simulations where we use the same cell lines, and test samples that Potti et al. [4] report, but use the parameter values used by Coombes et al. [2]

```
> sr9ProbesetIndices <- read.table(file.path("MatlabFiles", "DocetaxelSimulations",
+     "sr9ProbesetIndices.csv"), header = FALSE, sep = ",")
> sr9ProbesetIndices <- as.matrix(sr9ProbesetIndices)
> sr9ProbesetIndices <- t(apply(sr9ProbesetIndices, 1, sort))
> rownames(sr9ProbesetIndices) <- paste("simulation", c(1:100),
+     sep = "")
> colnames(sr9ProbesetIndices) <- paste("geneIndex", c(1:50), sep = "")
> sr9ProbesetIndices[1:3, 1:3]
```

```
            geneIndex1 geneIndex2 geneIndex3
simulation1         70        184        903
simulation2         70        184        903
simulation3         70        184        903
```

```
> all(sr9ProbesetIndices == matrix(rep(sr9ProbesetIndices["simulation1",
+     ], 100), ncol = 50, byrow = TRUE))
```

```
[1] TRUE
```

```
> all(sr9ProbesetIndices == baseProbesetIndices)
```

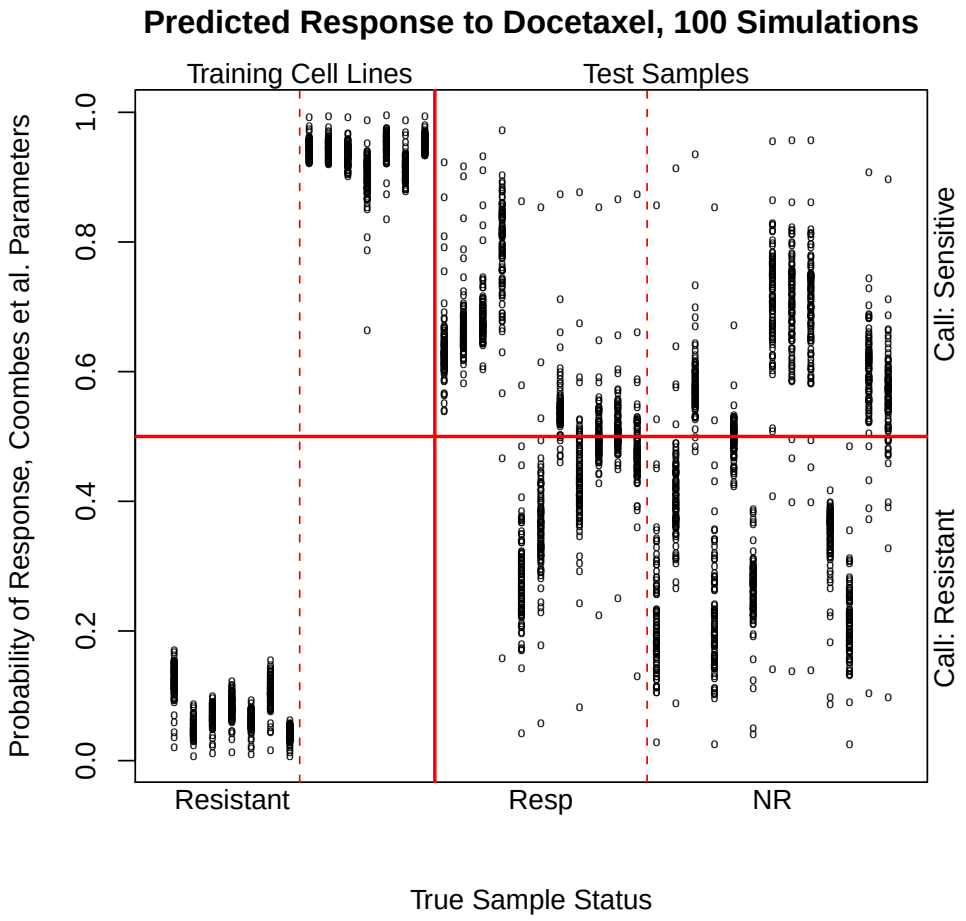
```
[1] TRUE
```

As expected, the same genes (gene indices) are chosen in every simulation, and these are the same ones we saw using the Potti et al. [4] parameter values. Now we look at the predictions.

```
> sr9PfitVectors <- read.table(file.path("MatlabFiles", "DocetaxelSimulations",
+     "sr9PfitVectors.csv"), header = FALSE, sep = ",")
> colnames(sr9PfitVectors) <- names(doceSimulationStatus)
> rownames(sr9PfitVectors) <- paste("simulation", c(1:100), sep = "")
```

Now we plot the probabilities of response to assess consistency and accuracy.

```
> matplot(t(sr9PfitVectors), pch = "o", col = "black", cex = 0.5,
+     xlim = c(0.5, 38.5), xaxt = "n", xlab = "True Sample Status",
+     ylab = "Probability of Response, Coombes et al. Parameters",
+     main = "Predicted Response to Docetaxel, 100 Simulations")
> abline(h = 0.5, v = c(14.5), col = "red", lwd = 2)
> abline(v = c(7.5, 25.5), col = "red", lty = "dashed")
> axis(side = 1, at = c(4, 11, 20, 32), labels = c("Resistant",
+     "Sensitive", "Resp", "NR"), tick = 0, line = -1)
> axis(side = 3, at = c(7.5, 26.5), labels = c("Training Cell Lines",
+     "Test Samples"), tick = 0, line = -1)
> axis(side = 4, at = c(0.25, 0.75), labels = c("Call: Resistant",
+     "Call: Sensitive"), tick = FALSE, line = -1)
```



Classification of the training cell lines is perfect, as might be expected. However, the results are considerably more noisy. This is largely because the number of burnin iterations has been reduced from 5000 to 1000, and the number of following iterations has been reduced from 1000 to 100. This doesn't really affect

the training data results. Classification of the test samples, however, is more problematic.

```
> sr9NRespCorrect <- apply(sr9PfitVectors[, 15:25], 1, function(x) {
+     sum(x > 0.5)
+ })
> sr9NNRCorrect <- apply(sr9PfitVectors[, 26:38], 1, function(x) {
+     sum(x < 0.5)
+ })
> sr9NCorrect <- sr9NRespCorrect + sr9NNRCorrect
> table(sr9NCorrect)

sr9NCorrect
11 12 13 14 15 16 17 19 20
24 26 14 13  9  3  7  3  1

> table(sr9NRespCorrect)

sr9NRespCorrect
 5  6  7  8  9 10
35 27 15 11  9  3

> table(sr9NNRCorrect)

sr9NNRCorrect
 6  7  8  9 10
45 38 13  2  2
```

The classsifications are not consistent across all 100 simulations. The median accuracy seen is 12/24, but the distribution is skewed with a longer upper tail.

8.4 Null Simulations, 1000 Runs, Coombes et al. [2] Parameters

Again, in order to get a better feel for the accuracy we should expect in the absence of underlying structure, we ran 1000 "null" simulations using the parameters from Coombes et al. [2] in which the cell lines were chosen at random, and the number of "correct" classifications were counted. Since each cell line was equally likely to be labeled "sensitive" or "resistant", the distribution of accuracy values is clearly symmetric and centered at 12/24. To enable reproducibility, we have recorded the indices of the cell lines chosen in each simulation.

```
> nullSR9CellLineIndices <- read.table(file.path("MatlabFiles",
+     "DocetaxelSimulations", "nullSR9CellLineIndices.csv"), header = FALSE,
+     sep = ",")
> nullSR9CellLineIndices <- as.matrix(nullSR9CellLineIndices)
> rownames(nullSR9CellLineIndices) <- paste("nullSim", c(1:1000),
+     sep = "")
> colnames(nullSR9CellLineIndices) <- c(paste("cellLineResIndex",
+     c(1:7), sep = ""), paste("cellLineSenIndex", c(1:7), sep = ""))
> nullSR9CellLineIndices[1:3, 1:3]
```

```
         cellLineResIndex1 cellLineResIndex2 cellLineResIndex3
nullSim1                37                 8                46
nullSim2                37                 9                44
nullSim3                59                 8                42
```

Now we load the prediction values and compute the distribution of accuracy values seen.

```
> nullSR9PfitVectors <- read.table(file.path("MatlabFiles", "DocetaxelSimulations",
+     "nullSR9TrainAndTestPfitVectors.csv"), header = FALSE, sep = ",")
> colnames(nullSR9PfitVectors) <- c(paste("nullRes", c(1:7), sep = ""),
+     paste("nullSim", c(1:7), sep = ""), paste("testResp", c(1:11),
+         sep = ""), paste("testNR", c(1:13), sep = ""))
> rownames(nullSR9PfitVectors) <- paste("nullSim", c(1:1000), sep = "")
> nullSR9NRespCorrect <- apply(nullSR9PfitVectors[, 15:25], 1,
+     function(x) {
+         sum(x > 0.5)
+     })
> nullSR9NNRCorrect <- apply(nullSR9PfitVectors[, 26:38], 1, function(x) {
+     sum(x < 0.5)
+ })
> nullSR9NCorrect <- nullSR9NRespCorrect + nullSR9NNRCorrect
> table(nullSR9NCorrect)

nullSR9NCorrect
 2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
 4  8 20 29 62 64 62 63 90 72 66 78 71 75 68 57 55 25 19 12

> sum(nullSR9NCorrect >= 12)

[1] 526
```

Overall, the p-value associated with the median accuracy actually seen is about 0.526.

8.5 Centering, 100 Simulations, Potti et al. [4] Parameters

As one last check, we considered the case where the training and test data sets were standardized separately before binreg was run. For this approach, we used the parameter values supplied by Potti et al. [4] We first export the processed data from our analysis steps above.

```
> write.table(rbind(rep(c(0, 1, 2), times = c(7, 7, 24)), cbind(tempDoceStandardized,
+     changGEOStandardized)), file = file.path("MatlabFiles", "DocetaxelSimulations",
+     "docetaxelCentered.csv"), row.names = FALSE, col.names = FALSE,
+     sep = ",")
```

We then invoke a separate Matlab script, DocetaxelNCI60Centered.m, which runs the 100 simulations. Here, we load and examine the results.

```
> centeredProbesetIndices <- read.table(file.path("MatlabFiles",
+     "DocetaxelSimulations", "centeredProbesetIndices.csv"), header = FALSE,
+     sep = ",")
```

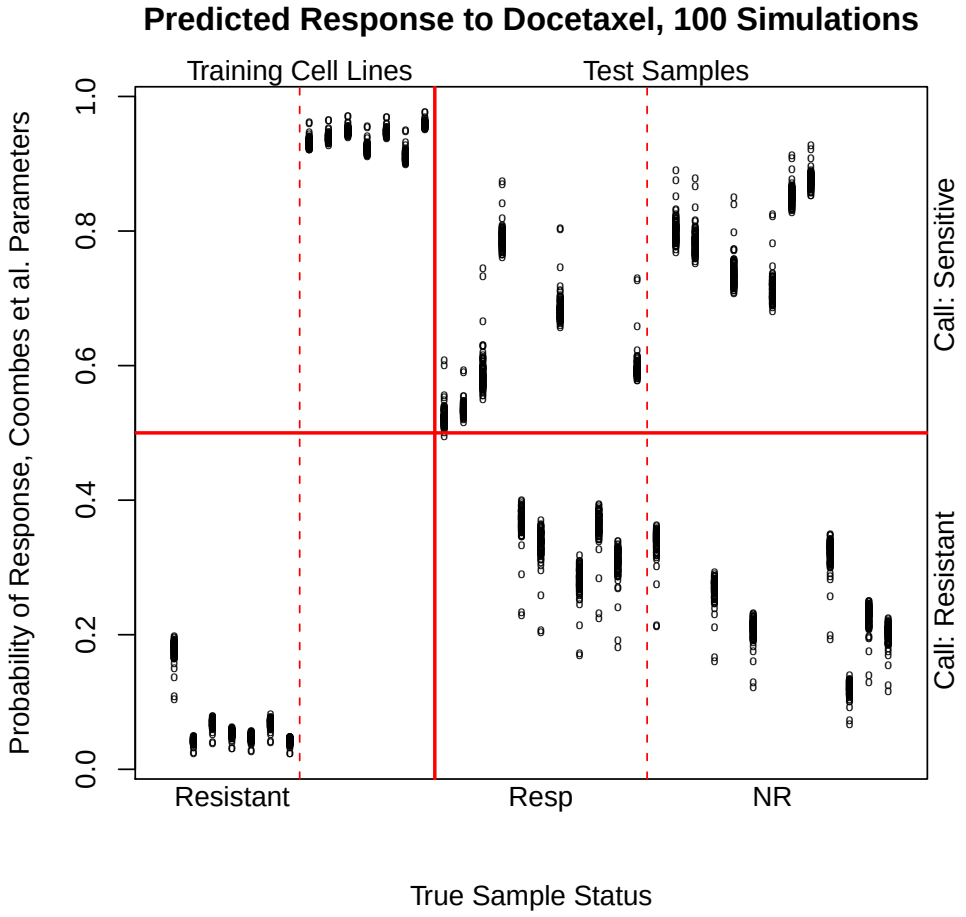
```
> centeredProbesetIndices <- as.matrix(centeredProbesetIndices)
> centeredProbesetIndices <- t(apply(centeredProbesetIndices, 1,
+     sort))
> rownames(centeredProbesetIndices) <- paste("simulation", c(1:100),
+     sep = "")
> colnames(centeredProbesetIndices) <- paste("geneIndex", c(1:50),
+     sep = "")
> centeredProbesetIndices[1:3, 1:3]

            geneIndex1 geneIndex2 geneIndex3
simulation1         70        184        659
simulation2         70        184        659
simulation3         70        184        659

> all(centeredProbesetIndices == matrix(rep(centeredProbesetIndices["simulation1",
+     ], 100), ncol = 50, byrow = TRUE))

[1] TRUE

> all(centeredProbesetIndices == baseProbesetIndices)

[1] FALSE

> length(intersect(centeredProbesetIndices[1, ], baseProbesetIndices[1,
+     ]))

[1] 24
```

As expected, the same genes (gene indices) are chosen in every simulation. However, these are not the same ones we saw before. Now we look at the predictions.

```
> centeredPfitVectors <- read.table(file.path("MatlabFiles", "DocetaxelSimulations",
+     "centeredPfitVectors.csv"), header = FALSE, sep = ",")
> colnames(centeredPfitVectors) <- names(doceSimulationStatus)
> rownames(centeredPfitVectors) <- paste("simulation", c(1:100),
+     sep = "")
```

Now we plot the probabilities of response to assess consistency and accuracy.

```
> matplot(t(centeredPfitVectors), pch = "o", col = "black", cex = 0.5,
+     xlim = c(0.5, 38.5), xaxt = "n", xlab = "True Sample Status",
+     ylab = "Probability of Response, Coombes et al. Parameters",
+     main = "Predicted Response to Docetaxel, 100 Simulations")
> abline(h = 0.5, v = c(14.5), col = "red", lwd = 2)
> abline(v = c(7.5, 25.5), col = "red", lty = "dashed")
> axis(side = 1, at = c(4, 11, 20, 32), labels = c("Resistant",
+     "Sensitive", "Resp", "NR"), tick = 0, line = -1)
> axis(side = 3, at = c(7.5, 26.5), labels = c("Training Cell Lines",
+     "Test Samples"), tick = 0, line = -1)
> axis(side = 4, at = c(0.25, 0.75), labels = c("Call: Resistant",
+     "Call: Sensitive"), tick = FALSE, line = -1)
```



Classification of the training cell lines is perfect, as might be expected. The results for the test samples are almost all consistent, with only one Resp sample changing classifications in a small number of simulations. Classification accuracy for the test samples, however, is still poor.

```
> centeredNRespCorrect <- apply(centeredPfitVectors[, 15:25], 1,
+     function(x) {
+         sum(x > 0.5)
+     })
> centeredNNRCorrect <- apply(centeredPfitVectors[, 26:38], 1,
+     function(x) {
+         sum(x < 0.5)
+     })
> centeredNCorrect <- centeredNRespCorrect + centeredNNRCorrect
> table(centeredNCorrect)

centeredNCorrect
12 13
```

```
 1 99

> table(centeredNRespCorrect)

centeredNRespCorrect
 5  6
 1 99

> table(centeredNNRCorrect)

centeredNNRCorrect
  7
100
```

The accuracy is 13/24 in all but one case, which is once again not significantly better than might be expected by chance alone (we did not run a separate 1000 simulations here).

9 Plotting the Results

Here, we simply condense some figures assembled above.

```
> ## This figure requires two panels, both identifying
> ## the presence of ties in the data.
>
> tempDoceHeatmapPanel <- c(0.10, 0.45, 0.1, 0.88)
> tempDocePredictionsPanel <- c(0.57, 0.97, 0.1, 0.88)
> par(plt=tempDoceHeatmapPanel)
> image(1:24, 1:24, pottiGEOCorrs < 0.9999, axes=FALSE, xlab="", ylab="",
+       asp=1,main="Mapping Potti Test Samples to Chang GEO")
> lines(c(0.5,24.5),c(11.5,11.5))
> lines(c(11.5,11.5),c(0.5,24.5))
> rect(0.5,0.5,24.5,24.5)
> axis(1, at=1:24, labels=rownames(pottiGEOCorrs), las=2, line=-1.5, tick=0);
> axis(2, at=1:24, labels=colnames(pottiGEOCorrs), las=2, tick=0,line=-0.5);
> lines(c(0.5,24.5),c(11,11),lty="dashed")
> lines(c(0.5,24.5),c(21,21),lty="dashed")
> axis(3,at=c(6,18),labels=c("Resistant (Potti)","Sensitive (Potti)"),
+      tick=0,line=-2)
> axis(4,at=c(6,18),labels=c("Sensitive (Chang)","Resistant (Chang)"),
+      tick=0,line=-1)
> axis(2,at=c(12,13),labels=c("S* (N12)","S* (N13)"),las=2,line=-4.5,tick=0)
> par(plt=tempDocePredictionsPanel, new=TRUE)
> matplot(t(basePfitVectors),pch="o",col="black",cex=0.5,xlim=c(0.5,38.5),
+         xaxt="n",xlab="",ylab="",
+         main="Predicted Response to Docetaxel, 100 Simulations")
> mtext("True Sample Status", side=1, line=1)
> mtext("Probability of Response", side=2, line=2.2)
> abline(h=0.5,v=c(14.5),col="red",lwd=2)
> abline(v=c(7.5,25.5),col="red",lty="dashed")
```

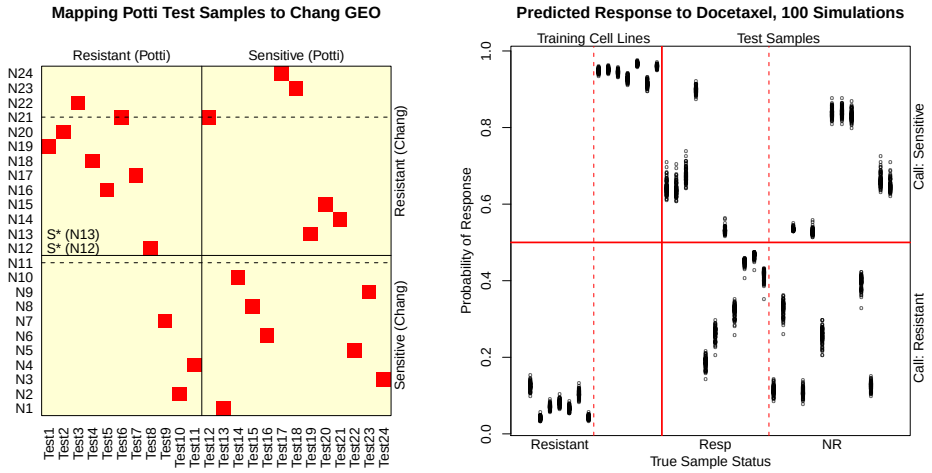


Figure 1: **A.** Mapping of test data quantifications from Potti et al. [4] to GEO samples posted by Chang et al. [1] GEO samples have been reordered to match the ordering given in the clinical data, which tracks with the percent residual tumor. Lines mark the sensitive/resistant boundaries from GEO and from the Potti et al. [4] quantifications. Potti et al. [4] mention using a different cutoff than Chang et al. [1] to determine sensitivity (40% instead of 25%), causing them to relabel 2 resistant samples (N12 and N13, shown) as sensitive. Ideally, all red squares should be in the upper left or lower right sections. One GEO sample (N21, GSM4910) is present twice in the Potti et al. [4] data, and labeled both ways; another (N11, GSM4914) is omitted entirely. Counting, 10/24 samples (6 upper right, 3 lower left, 1 omitted) are mislabeled before modeling begins. **B.** Predictions of docetaxel sensitivity using the Potti et al. [3] cell lines, software, and parameter settings, with the test data labeled according to Chang et al. [1]. When the data are correctly labeled, prediction accuracy is 11/24.

```
> axis(side=1,at=c(4,11,20,32),
+      labels=c("Resistant","Sensitive","Resp","NR"),tick=0,line=-1)
> axis(side=3,at=c(7.5,26.5),
+      labels=c("Training Cell Lines", "Test Samples"),tick=0,line=-1)
> axis(side=4,at=c(0.25,0.75),labels=c("Call: Resistant","Call: Sensitive"),
+      tick=FALSE,line=-1)

quartz
     2
```

10 Appendix

10.1 File Location

```
> getwd()

[1] "/Users/kabagg/ReproRsch/AnnAppStat"
```

10.2 Saves

10.3 SessionInfo

```
> sessionInfo()

R version 2.9.1 (2009-06-26)
i386-apple-darwin8.11.1

locale:
en_US.UTF-8/en_US.UTF-8/C/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] XML_2.6-0
```

References

[1] Chang JC, Wooten EC, Tsimelzon A, et al.: Gene expression profiling for the prediction of therapeutic response to docetaxel in patients with breast cancer. *Lancet*, **362**:362-369 (2003).

[2] Coombes KR, Wang J, Baggerly KA: Microarrays: retracing steps. *Nat Med*, **13**:1276-7, 2007.

[3] Potti A, Dressman HK, Bild A, et al: Genomic signatures to guide the use of chemotherapeutics. *Nat Med*, **12**:1294-1300, 2006

[4] Potti A, Nevins JR: Reply to Microarrays: retracing steps. *Nat Med*, **13**:1277-8, 2007.

[5] Potti A, Dressman HK, Bild A, et al: Corrigendum to Genomic signatures to guide the use of chemotherapeutics. *Nat Med*, **14**:889, 2008.