Matching Chemo Predictor Columns

Keith A. Baggerly

July 25, 2009

Contents

List of Figures

List of Tables

1 Executive Summary

1.1 Introduction

In response to queries about the identities of the cell lines used to assemble signatures of drug response, we were sent two tables of quantifications. The first table covered all seven drugs mentioned in Potti et al. [2], whereas the second focused solely on docetaxel. Neither table identified the cell lines by name. Rather, the columns of quantifications pertaining to each drug were identified, and headers of "0" or "1" were supplied.

In this report, we match the quantifications reported to numbers from various datasets in order to identify the specific cell lines involved.

1.2 Methods

We loaded two previously constructed Rda files: novartisAll and changAll. Next, we loaded the two quantification tables: "Chemo predictors (U-95) All - FINAL.txt" and "Docetaxel_Predictor__U-95_.txt". We identified matching samples first by comparing values for specific probesets, and then by checking the degree of overall correlation.

1.3 Results

We were able to match and identify all of the cell lines reported. For five of the drugs in the first quantification table, all of the matching cell lines come from the A series of replicates from the Novartis U95A array profiles of the NCI-60. For the the other two, docetaxel and cyclophosphamide, the columns matched quantifications from Chang et al. [1]. For the columns matching numbers from Chang et al. [1], all of the probeset names were scrambled. Further, the lines reported for docetaxel and cyclophosphamide were the same, but with group labels of 0 and 1 reversed. In the second table, focusing on docetaxel alone, the columns also match cell lines coming from the Novartis A set.

We produced a 2x7 (group by drug) matrix of lists of the cell lines used, and saved this matrix, "cellLinesFromPredictors" in RDataObjects as "cellLinesFromPredictors.Rda."

1.4 Conclusions

Only the A set of Novartis U95A replicates was used in the initial construction of drug sensitivity signatures. Some errors in bookkeeping allowed samples from Chang et al. [1] to be confused with those from the NCI-60. The cases of docetaxel and cyclophosphamide show that at some stages numerical values were detached from probeset identifiers, sample identifiers, and sensitive/resistant labels, and that these identifiers and labels were occasionally reattached incorrectly.

2 Options and Libraries

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

3 Loading and Parsing Data

3.1 Earlier Rda Files

We begin by loading two Rda files assembled earlier: novartisAll and changAll.

```
> rdaList <- c("novartisAll", "changAll")
> 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/novartisAll.Rda  from cache
loading  RDataObjects/changAll.Rda  from cache
```

3.2 Chemo Predictors (U-95) All - FINAL.txt

Next, we load in the first table of quantifications provided.

```
> chemoPredictors <- read.table(file.path("RawData", "PottiNatMed",
+     "Chemo predictors (U-95) All - FINAL.txt"), header = TRUE,
+     sep = "\t", row.names = 1)
> dim(chemoPredictors)

[1] 12558   134

> chemoPredictors[1:3, 1:3]

            Adria0        X0     X0.1
36460_at  41.67195  21.82034 125.7948
36461_at 171.39024 122.09759 218.3590
36462_at 147.49791 203.84113 211.1068
```

The predictors table doesn't include all of the probesets available on the U95A array. We suspect the difference involves the Affymetrix control probesets.

```
> length(grep("^AFFX", rownames(novartisAll)))

[1] 67

> length(grep("^AFFX", rownames(chemoPredictors)))
```

```
[1] 0
```

```
> length(intersect(rownames(novartisAll), rownames(chemoPredictors)))
```

```
[1] 12558
```

```
> length(setdiff(rownames(novartisAll), rownames(chemoPredictors)))
```

```
[1] 67
```

It is indeed the control probesets that are missing.

We'd also like to assign some slightly more usable column names to the matrix.

```
> colnames(chemoPredictors)
```

```
  [1] "Adria0" "X0"     "X0.1"   "X0.2"   "X0.3"   "X0.4"   "X0.5"   "X0.6"
  [9] "X0.7"   "X0.8"   "X1"     "X1.1"   "X1.2"   "X1.3"   "X1.4"   "X1.5"
 [17] "X1.6"   "X1.7"   "X1.8"   "X1.9"   "X1.10"  "Adria1" "Doce0"  "X0.9"
 [25] "X0.10"  "X0.11"  "X0.12"  "X0.13"  "X0.14"  "X0.15"  "X0.16"  "X0.17"
 [33] "X1.11"  "X1.12"  "X1.13"  "X1.14"  "X1.15"  "X1.16"  "X1.17"  "X1.18"
 [41] "X1.19"  "Doce1"  "Etopo0" "X0.18"  "X0.19"  "X0.20"  "X0.21"  "X0.22"
 [49] "X0.23"  "X0.24"  "X0.25"  "X1.20"  "X1.21"  "X1.22"  "X1.23"  "X1.24"
 [57] "X1.25"  "X1.26"  "Etopo1" "X5.FU0" "X0.26"  "X0.27"  "X0.28"  "X0.29"
 [65] "X0.30"  "X0.31"  "X0.32"  "X1.27"  "X1.28"  "X1.29"  "X1.30"  "X1.31"
 [73] "X1.32"  "X5.FU1" "Cytox0" "X0.33"  "X0.34"  "X0.35"  "X0.36"  "X0.37"
 [81] "X0.38"  "X0.39"  "X0.40"  "X0.41"  "X1.33"  "X1.34"  "X1.35"  "X1.36"
 [89] "X1.37"  "X1.38"  "X1.39"  "X1.40"  "X1.41"  "Cytox1" "Topo0"  "X0.42"
 [97] "X0.43"  "X0.44"  "X0.45"  "X0.46"  "X0.47"  "X0.48"  "X0.49"  "X0.50"
[105] "X0.51"  "X0.52"  "X0.53"  "X1.42"  "X1.43"  "X1.44"  "X1.45"  "X1.46"
[113] "X1.47"  "X1.48"  "X1.49"  "X1.50"  "Topo1"  "Taxol0" "X0.54"  "X0.55"
[121] "X0.56"  "X0.57"  "X0.58"  "X0.59"  "X0.60"  "X0.61"  "X1.51"  "X1.52"
[129] "X1.53"  "X1.54"  "X1.55"  "X1.56"  "X1.57"  "Taxol1"
```

```
> drugBoundaries <- c(1:dim(chemoPredictors)[2])[-grep("^X[01]",
+     colnames(chemoPredictors))]
> drugBoundaries
```

```
 [1]   1  22  23  42  43  59  60  74  75  94  95 117 118 134
```

```
> drugStarts <- drugBoundaries[seq(1, length(drugBoundaries), 2)]
> drugStops <- drugBoundaries[seq(2, length(drugBoundaries), 2)]
> tempNames <- colnames(chemoPredictors)
> for (tempDrugIndex in 1:length(drugStarts)) {
+     drugPrefix <- substr(tempNames[drugStarts[tempDrugIndex]],
+         1, nchar(tempNames[drugStarts[tempDrugIndex]]) - 1)
+     nZeros <- length(grep("^X0", tempNames[drugStarts[tempDrugIndex]:drugStops[tempDrugIndex]])) +
+         1
+     nOnes <- length(grep("^X1", tempNames[drugStarts[tempDrugIndex]:drugStops[tempDrugIndex]])) +
+         1
+     tempNames[drugStarts[tempDrugIndex]:drugStops[tempDrugIndex]] <- paste(drugPrefix,
```

```
+         rep(c(0, 1), times = c(nZeros, nOnes)), c(1:nZeros, 1:nOnes),
+         sep = ".")
+ }
> tempNames[1:3]

[1] "Adria.0.1" "Adria.0.2" "Adria.0.3"

> colnames(chemoPredictors) <- tempNames
```

3.3 Docetaxel_Predictor__U-95_.txt

Now we load in the second table of quantifications provided.

```
> docetaxelPredictor <- read.table(file.path("RawData", "PottiNatMed",
+     "Docetaxel_Predictor__U-95_.txt"), header = TRUE, sep = "\t",
+     row.names = 1)
> dim(docetaxelPredictor)

[1] 12558    14

> docetaxelPredictor[1:3, 1:3]

                X0     X0.1      X0.2
36460_at  63.30723 120.0093  65.80951
36461_at  71.10506 186.4581  98.86178
36462_at 340.30246 259.1901 146.64468

> colnames(docetaxelPredictor)

 [1] "X0"   "X0.1" "X0.2" "X0.3" "X0.4" "X0.5" "X0.6" "X1"   "X1.1" "X1.2"
[11] "X1.3" "X1.4" "X1.5" "X1.6"

> all(rownames(docetaxelPredictor) == rownames(chemoPredictors))

[1] TRUE
```

Since the probesets are the same as those from the previous table, the control probesets are again missing. As above, we revise the column names to give something more informative.

```
> colnames(docetaxelPredictor) <- paste("Doce", rep(c(0, 1), times = c(7,
+     7)), c(1:7, 1:7), sep = ".")
```

4 Matching Columns

4.1 Pass 1: Matching Most of the Chemo Predictors Columns

We start by assuming the quantifications supplied match some columns in the Novartis data. To test this, we look at the very first probeset.

```
> dim(novartisAll)

[1] 12625   180
```

```
> length(unique(novartisAll[1, ]))

[1] 180

> match(chemoPredictors[1, ], novartisAll[1, ])

  [1] 21 23 30 45 49 50 51 53 54 56 28 35 36 40 43 57 58 59 60 61 62 68 NA NA NA
 [26] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA 21 25 29 28 41 44 53 55
 [51] 65 27 33 35 38 43 64 59 60 27 32 34 46 48 54 66 72 28 30 38 40 49 61 70 NA
 [76] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA 21 23 24 26 41 44
[101] 45 48 61 66 67 68 72 13 16 30 54 33 34 35 43 53 32 20 21 26 30 32 33 36 58
[126] 63 15 38 39 40 50 53 61 65

> table(is.na(match(chemoPredictors[1, ], novartisAll[1, ])), substr(colnames(chemoPredictors),
+     1, 4))

       Adri Cyto Doce Etop Taxo Topo X5.F
  FALSE   22    0    0   17   17   23   15
  TRUE     0   20   20    0    0    0    0
```

We appear to have perfect hits for all columns relating to 5 of the 7 drugs examined; we don't match the columns for cyclophosphamide (cytoxan) or docetaxel. Next, we look at what the mappings actually are, and confirm that the matches extend beyond the first probeset.

```
> chemoMapping <- matrix("", nrow = dim(chemoPredictors)[2], ncol = 2)
> colnames(chemoMapping) <- c("TableColumn", "MatchedSample")
> chemoMapping[, "TableColumn"] <- colnames(chemoPredictors)
> chemoMapping[, "MatchedSample"] <- colnames(novartisAll)[match(chemoPredictors[1,
+     ], novartisAll[1, ])]
> allMatched <- rep(NA, dim(chemoMapping)[1])
> for (tempIndex in 1:dim(chemoMapping)[1]) {
+     if (!is.na(chemoMapping[tempIndex, "MatchedSample"])) {
+         allMatched[tempIndex] <- all(chemoPredictors[, chemoMapping[tempIndex,
+             "TableColumn"]] == novartisAll[rownames(chemoPredictors),
+             chemoMapping[tempIndex, "MatchedSample"]])
+     }
+ }
> cbind(chemoMapping, allMatched)[!is.na(allMatched), ]

      TableColumn  MatchedSample       allMatched
 [1,] "Adria.0.1"  "SF-539.A"          "TRUE"
 [2,] "Adria.0.2"  "SNB-75.A"          "TRUE"
 [3,] "Adria.0.3"  "MDA-MB-435.A"      "TRUE"
 [4,] "Adria.0.4"  "NCI-H23.A"         "TRUE"
 [5,] "Adria.0.5"  "M14.A"             "TRUE"
 [6,] "Adria.0.6"  "MALME-3M.A"        "TRUE"
 [7,] "Adria.0.7"  "SK-MEL-2.A"        "TRUE"
 [8,] "Adria.0.8"  "SK-MEL-28.A"       "TRUE"
 [9,] "Adria.0.9"  "SK-MEL-5.A"        "TRUE"
[10,] "Adria.0.10" "UACC-62.A"         "TRUE"
```

```
[11,] "Adria.1.1"  "NCI/ADR-RES.A"     "TRUE"
[12,] "Adria.1.2"  "HCT-15.A"          "TRUE"
[13,] "Adria.1.3"  "HT29.A"            "TRUE"
[14,] "Adria.1.4"  "EKVX.A"            "TRUE"
[15,] "Adria.1.5"  "NCI-H322M.A"       "TRUE"
[16,] "Adria.1.6"  "IGROV1.A"          "TRUE"
[17,] "Adria.1.7"  "OVCAR-3.A"         "TRUE"
[18,] "Adria.1.8"  "OVCAR-4.A"         "TRUE"
[19,] "Adria.1.9"  "OVCAR-5.A"         "TRUE"
[20,] "Adria.1.10" "OVCAR-8.A"         "TRUE"
[21,] "Adria.1.11" "SK-OV-3.A"         "TRUE"
[22,] "Adria.1.12" "CAKI-1.A"          "TRUE"
[23,] "Etopo.0.1"  "SF-539.A"          "TRUE"
[24,] "Etopo.0.2"  "BT-549.A"          "TRUE"
[25,] "Etopo.0.3"  "MDA-MB-231/ATCC.A" "TRUE"
[26,] "Etopo.0.4"  "NCI/ADR-RES.A"     "TRUE"
[27,] "Etopo.0.5"  "HOP-62.A"          "TRUE"
[28,] "Etopo.0.6"  "NCI-H226.A"        "TRUE"
[29,] "Etopo.0.7"  "SK-MEL-28.A"       "TRUE"
[30,] "Etopo.0.8"  "UACC-257.A"        "TRUE"
[31,] "Etopo.0.9"  "786-0.A"           "TRUE"
[32,] "Etopo.1.1"  "MCF7.A"            "TRUE"
[33,] "Etopo.1.2"  "HCC-2998.A"        "TRUE"
[34,] "Etopo.1.3"  "HCT-15.A"          "TRUE"
[35,] "Etopo.1.4"  "SW-620.A"          "TRUE"
[36,] "Etopo.1.5"  "NCI-H322M.A"       "TRUE"
[37,] "Etopo.1.6"  "PC-3.A"            "TRUE"
[38,] "Etopo.1.7"  "OVCAR-4.A"         "TRUE"
[39,] "Etopo.1.8"  "OVCAR-5.A"         "TRUE"
[40,] "X5.FU.0.1"  "MCF7.A"            "TRUE"
[41,] "X5.FU.0.2"  "COLO 205.A"        "TRUE"
[42,] "X5.FU.0.3"  "HCT-116.A"         "TRUE"
[43,] "X5.FU.0.4"  "NCI-H460.A"        "TRUE"
[44,] "X5.FU.0.5"  "LOX IMVI.A"        "TRUE"
[45,] "X5.FU.0.6"  "SK-MEL-5.A"        "TRUE"
[46,] "X5.FU.0.7"  "A498.A"            "TRUE"
[47,] "X5.FU.0.8"  "UO-31.A"           "TRUE"
[48,] "X5.FU.1.1"  "NCI/ADR-RES.A"     "TRUE"
[49,] "X5.FU.1.2"  "MDA-MB-435.A"      "TRUE"
[50,] "X5.FU.1.3"  "SW-620.A"          "TRUE"
[51,] "X5.FU.1.4"  "EKVX.A"            "TRUE"
[52,] "X5.FU.1.5"  "M14.A"             "TRUE"
[53,] "X5.FU.1.6"  "OVCAR-8.A"         "TRUE"
[54,] "X5.FU.1.7"  "SN12C.A"           "TRUE"
[55,] "Topo.0.1"   "SF-539.A"          "TRUE"
[56,] "Topo.0.2"   "SNB-75.A"          "TRUE"
[57,] "Topo.0.3"   "U251.A"            "TRUE"
[58,] "Topo.0.4"   "HS 578T.A"         "TRUE"
```

```
[59,] "Topo.0.5"   "HOP-62.A"          "TRUE"
[60,] "Topo.0.6"   "NCI-H226.A"        "TRUE"
[61,] "Topo.0.7"   "NCI-H23.A"         "TRUE"
[62,] "Topo.0.8"   "LOX IMVI.A"        "TRUE"
[63,] "Topo.0.9"   "OVCAR-8.A"         "TRUE"
[64,] "Topo.0.10"  "A498.A"            "TRUE"
[65,] "Topo.0.11"  "ACHN.A"            "TRUE"
[66,] "Topo.0.12"  "CAKI-1.A"          "TRUE"
[67,] "Topo.0.13"  "UO-31.A"           "TRUE"
[68,] "Topo.1.1"   "K-562.A"           "TRUE"
[69,] "Topo.1.2"   "RPMI-8226.A"       "TRUE"
[70,] "Topo.1.3"   "MDA-MB-435.A"      "TRUE"
[71,] "Topo.1.4"   "SK-MEL-5.A"        "TRUE"
[72,] "Topo.1.5"   "HCC-2998.A"        "TRUE"
[73,] "Topo.1.6"   "HCT-116.A"         "TRUE"
[74,] "Topo.1.7"   "HCT-15.A"          "TRUE"
[75,] "Topo.1.8"   "NCI-H322M.A"       "TRUE"
[76,] "Topo.1.9"   "SK-MEL-28.A"       "TRUE"
[77,] "Topo.1.10"  "COLO 205.A"        "TRUE"
[78,] "Taxol.0.1"  "SF-295.A"          "TRUE"
[79,] "Taxol.0.2"  "SF-539.A"          "TRUE"
[80,] "Taxol.0.3"  "HS 578T.A"         "TRUE"
[81,] "Taxol.0.4"  "MDA-MB-435.A"      "TRUE"
[82,] "Taxol.0.5"  "COLO 205.A"        "TRUE"
[83,] "Taxol.0.6"  "HCC-2998.A"        "TRUE"
[84,] "Taxol.0.7"  "HT29.A"            "TRUE"
[85,] "Taxol.0.8"  "OVCAR-3.A"         "TRUE"
[86,] "Taxol.0.9"  "DU-145.A"          "TRUE"
[87,] "Taxol.1.1"  "CCRF-CEM.A"        "TRUE"
[88,] "Taxol.1.2"  "SW-620.A"          "TRUE"
[89,] "Taxol.1.3"  "A549/ATCC.A"       "TRUE"
[90,] "Taxol.1.4"  "EKVX.A"            "TRUE"
[91,] "Taxol.1.5"  "MALME-3M.A"        "TRUE"
[92,] "Taxol.1.6"  "SK-MEL-28.A"       "TRUE"
[93,] "Taxol.1.7"  "OVCAR-8.A"         "TRUE"
[94,] "Taxol.1.8"  "786-0.A"           "TRUE"
```

The mappings are perfect throughout, and only involve samples from the "A" set of replicates.

4.2 Pass 2: Matching the Docetaxel Predictor Columns

While we didn't match the docetaxel columns in the chemo predictors table, that may be fixed in the new docetaxel table. Again, we start by matching the first probeset, and then check the rest of the rows.

```
> match(docetaxelPredictor[1, ], novartisAll[1, ])

 [1] 40 57 59 65 68 70 71 17 21 36 41 51 54 95

> docetaxelMapping <- matrix("", nrow = dim(docetaxelPredictor)[2],
+     ncol = 2)
```

```
> colnames(docetaxelMapping) <- c("TableColumn", "MatchedSample")
> docetaxelMapping[, "TableColumn"] <- colnames(docetaxelPredictor)
> docetaxelMapping[, "MatchedSample"] <- colnames(novartisAll)[match(docetaxelPredictor[1,
+     ], novartisAll[1, ])]
> allDocetaxelMatched <- rep(NA, dim(docetaxelMapping)[1])
> for (tempIndex in 1:dim(docetaxelMapping)[1]) {
+     allDocetaxelMatched[tempIndex] <- all(docetaxelPredictor[,
+         docetaxelMapping[tempIndex, "TableColumn"]] == novartisAll[rownames(docetaxelPredictor),
+         docetaxelMapping[tempIndex, "MatchedSample"]])
+ }
> cbind(docetaxelMapping, allDocetaxelMatched)
```

```
      TableColumn MatchedSample allDocetaxelMatched
 [1,] "Doce.0.1"  "EKVX.A"      "TRUE"
 [2,] "Doce.0.2"  "IGROV1.A"    "TRUE"
 [3,] "Doce.0.3"  "OVCAR-4.A"   "TRUE"
 [4,] "Doce.0.4"  "786-0.A"     "TRUE"
 [5,] "Doce.0.5"  "CAKI-1.A"    "TRUE"
 [6,] "Doce.0.6"  "SN12C.A"     "TRUE"
 [7,] "Doce.0.7"  "TK-10.A"     "TRUE"
 [8,] "Doce.1.1"  "HL-60(TB).A" "TRUE"
 [9,] "Doce.1.2"  "SF-539.A"    "TRUE"
[10,] "Doce.1.3"  "HT29.A"      "TRUE"
[11,] "Doce.1.4"  "HOP-62.A"    "TRUE"
[12,] "Doce.1.5"  "SK-MEL-2.A"  "TRUE"
[13,] "Doce.1.6"  "SK-MEL-5.A"  "TRUE"
[14,] "Doce.1.7"  "NCI-H522.A"  "TRUE"
```

We match all of these samples perfectly using profiles from the Novartis A set of replicates.

4.3 Pass 3: Matching the Difficult Chemo Predictor Columns

The initial columns for docetaxel and cyclophosphamide didn't immediately line up with the Novartis quantifications. We next tried matching them to the quantifications from the other set of U95A data examined in the drug signature paper: the quantifications from Chang et al. [1]

```
> which(changAll == chemoPredictors[1, "Doce.0.1"], arr.ind = TRUE)
```

```
         row col
31307_at  68   5
```

```
> match(chemoPredictors[1, ], changAll["31307_at", ])
```

```
  [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA  5  9  3
 [26]  7  4  1  2 10  8  6 19 20 22 24 18 16 21 17 12 11 NA NA NA NA NA NA NA NA
 [51] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA 19
 [76] 20 22 24 18 16 21 17 12 11  5  9  3  7  4  1  2 10  8  6 NA NA NA NA NA NA
[101] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
[126] NA NA NA NA NA NA NA NA NA
```

```
> length(grep("^AFFX", rownames(changAll)[1:67]))
```

```
[1] 67
```

All of the first row entries for docetaxel and cyclophosphamide match entries in the 68th row of the changAll table. Now, 68 is an interesting number. The first 67 rows are Affymetrix control probesets, so this is the first "real" data row. We now check the extent of the mapping, focusing just on the first docetaxel sample.

```
> sum(chemoPredictors[, "Doce.0.1"] == changAll[68:12625, "N5"])
```

```
[1] 12535
```

```
> which(chemoPredictors[, "Doce.0.1"] != changAll[68:12625, "N5"])
```

```
160034_s_at   160020_at 160021_r_at   160036_at   160037_at   160035_at
      12534       12535       12536       12537       12538       12539
  160039_at   160031_at   160024_at   160041_at 160044_g_at 160033_s_at
      12540       12541       12542       12543       12544       12545
160038_s_at 160028_s_at   160022_at   160030_at   160029_at   160023_at
      12547       12548       12549       12550       12551       12552
  160040_at   160025_at 160027_s_at   160026_at 160042_s_at
      12553       12554       12555       12556       12558
```

```
> bottomOrder <- order(rownames(changAll)[c(12534:12558) + 67])
> all(chemoPredictors[12534:12558, "Doce.0.1"] == changAll[c((12534:12558) +
+     67)[bottomOrder], "N5"])
```

```
[1] TRUE
```

```
> sum(rownames(chemoPredictors[, "Doce.0.1"]) == rownames(changAll[c(68:12600,
+     c(12601:12625)[bottomOrder]), "N5"]))
```

```
[1] 0
```

The values match exactly for all but the last 25 or so probesets. Looking at these last probesets more closely shows that they are not in alphabetical order; when these last probesets are so ordered the mapping of numbers now matches. The mapping of probeset names, however, does not match. None of the names are correct.

Next, we check whether the ordering identified above holds for the other samples matched as well.

```
> chemoMappingChang <- matrix("", nrow = dim(chemoPredictors)[2],
+     ncol = 2)
> colnames(chemoMappingChang) <- c("TableColumn", "MatchedSample")
> chemoMappingChang[, "TableColumn"] <- colnames(chemoPredictors)
> chemoMappingChang[, "MatchedSample"] <- colnames(changAll)[match(chemoPredictors[1,
+     ], changAll[68, ])]
> allMatchedChang <- rep(NA, dim(chemoMappingChang)[1])
> for (tempIndex in 1:dim(chemoMappingChang)[1]) {
+     if (!is.na(chemoMappingChang[tempIndex, "MatchedSample"])) {
+         allMatchedChang[tempIndex] <- all(chemoPredictors[, chemoMappingChang[tempIndex,
+             "TableColumn"]] == changAll[c(68:12600, c(12601:12625)[bottomOrder]),
```

```
+             chemoMappingChang[tempIndex, "MatchedSample"]])
+     }
+ }
> cbind(chemoMappingChang, allMatchedChang)[!is.na(allMatchedChang),
+     ]

      TableColumn  MatchedSample allMatchedChang
 [1,] "Doce.0.1"   "N5"          "TRUE"
 [2,] "Doce.0.2"   "N9"          "TRUE"
 [3,] "Doce.0.3"   "N3"          "TRUE"
 [4,] "Doce.0.4"   "N7"          "TRUE"
 [5,] "Doce.0.5"   "N4"          "TRUE"
 [6,] "Doce.0.6"   "N1"          "TRUE"
 [7,] "Doce.0.7"   "N2"          "TRUE"
 [8,] "Doce.0.8"   "N10"         "TRUE"
 [9,] "Doce.0.9"   "N8"          "TRUE"
[10,] "Doce.0.10"  "N6"          "TRUE"
[11,] "Doce.1.1"   "N19"         "TRUE"
[12,] "Doce.1.2"   "N20"         "TRUE"
[13,] "Doce.1.3"   "N22"         "TRUE"
[14,] "Doce.1.4"   "N24"         "TRUE"
[15,] "Doce.1.5"   "N18"         "TRUE"
[16,] "Doce.1.6"   "N16"         "TRUE"
[17,] "Doce.1.7"   "N21"         "TRUE"
[18,] "Doce.1.8"   "N17"         "TRUE"
[19,] "Doce.1.9"   "N12"         "TRUE"
[20,] "Doce.1.10"  "N11"         "TRUE"
[21,] "Cytox.0.1"  "N19"         "TRUE"
[22,] "Cytox.0.2"  "N20"         "TRUE"
[23,] "Cytox.0.3"  "N22"         "TRUE"
[24,] "Cytox.0.4"  "N24"         "TRUE"
[25,] "Cytox.0.5"  "N18"         "TRUE"
[26,] "Cytox.0.6"  "N16"         "TRUE"
[27,] "Cytox.0.7"  "N21"         "TRUE"
[28,] "Cytox.0.8"  "N17"         "TRUE"
[29,] "Cytox.0.9"  "N12"         "TRUE"
[30,] "Cytox.0.10" "N11"         "TRUE"
[31,] "Cytox.1.1"  "N5"          "TRUE"
[32,] "Cytox.1.2"  "N9"          "TRUE"
[33,] "Cytox.1.3"  "N3"          "TRUE"
[34,] "Cytox.1.4"  "N7"          "TRUE"
[35,] "Cytox.1.5"  "N4"          "TRUE"
[36,] "Cytox.1.6"  "N1"          "TRUE"
[37,] "Cytox.1.7"  "N2"          "TRUE"
[38,] "Cytox.1.8"  "N10"         "TRUE"
[39,] "Cytox.1.9"  "N8"          "TRUE"
[40,] "Cytox.1.10" "N6"          "TRUE"
```

All of the columns match perfectly; they are drawn from the Chang et al. [1] data. Interestingly, the same samples are used for both docetaxel and cytoxan, but with the 0/1 labeling reversed. Of the 24 Chang et al. [1], the four that are missing are N13, N14, N15, and N23.

Next, We check how the Chang et al. [1] samples are ordered using the annotation from GEO.

```
> changAllInfo[chemoMappingChang[grep("^Doce", chemoMappingChang[,
+     "TableColumn"]), "MatchedSample"], 1:3]
```

```
      geoID geoTitle    status
N5  GSM4903       71 Sensitive
N9  GSM4907      142 Sensitive
N3  GSM4908      273 Sensitive
N7  GSM4913      377 Sensitive
N4  GSM4915      425 Sensitive
N1  GSM4917      437 Sensitive
N2  GSM4919      447 Sensitive
N10 GSM4920      458 Sensitive
N8  GSM4921      492 Sensitive
N6  GSM4923      558 Sensitive
N19 GSM4901       44 Resistant
N20 GSM4902       51 Resistant
N22 GSM4904      113 Resistant
N24 GSM4905      118 Resistant
N18 GSM4906      136 Resistant
N16 GSM4909      356 Resistant
N21 GSM4910      358 Resistant
N17 GSM4911      359 Resistant
N12 GSM4912      370 Resistant
N11 GSM4914      413 Sensitive
```

```
> changAllInfo[c("N23", "N13", "N15", "N14"), 1:3]
```

```
      geoID geoTitle    status
N23 GSM4916      432 Resistant
N13 GSM4918      438 Resistant
N15 GSM4922      555 Resistant
N14 GSM4924      562 Resistant
```

The GEO ids of the 0 and 1 groups form two ascending runs, and this is parallelled in the numerical values assigned as the GEO titles. Looking at the four samples that were omitted, they fall at the bottom end of the second run, both in terms of GEO id and title.

5 The Final Mappings

Here, we clean up and organize the lists of cell lines identified for easier processing.

First, we allocate a data structure.

```
> cellLinesFromPredictors <- matrix(vector("list", 2 * 7), nrow = 2,
+     ncol = 7)
```

```
> rownames(cellLinesFromPredictors) <- c("Group0", "Group1")
> colnames(cellLinesFromPredictors) <- c("Docetaxel", "Doxorubicin",
+     "Paclitaxel", "Fluorouracil", "Cyclophosphamide", "Topotecan",
+     "Etoposide")
```

Next, we fill in the components one drug at a time.

5.1 Docetaxel

Docetaxel (from the Docetaxel Predictor):

```
> temp <- docetaxelMapping[grep("^Doce\\.0", docetaxelMapping[,
+     "TableColumn"]), "MatchedSample"]
> cellLinesFromPredictors[["Group0", "Docetaxel"]] <- substr(temp,
+     1, nchar(temp) - 2)
> temp <- docetaxelMapping[grep("^Doce\\.1", docetaxelMapping[,
+     "TableColumn"]), "MatchedSample"]
> cellLinesFromPredictors[["Group1", "Docetaxel"]] <- substr(temp,
+     1, nchar(temp) - 2)
> cellLinesFromPredictors[, "Docetaxel"]

$Group0
[1] "EKVX"    "IGROV1"  "OVCAR-4" "786-0"   "CAKI-1"  "SN12C"   "TK-10"

$Group1
[1] "HL-60(TB)" "SF-539"    "HT29"      "HOP-62"    "SK-MEL-2"  "SK-MEL-5"
[7] "NCI-H522"
```

5.2 Doxorubicin

Doxorubicin (Adriamycin):

```
> temp <- chemoMapping[grep("^Adria\\.0", chemoMapping[, "TableColumn"]),
+     "MatchedSample"]
> cellLinesFromPredictors[["Group0", "Doxorubicin"]] <- substr(temp,
+     1, nchar(temp) - 2)
> temp <- chemoMapping[grep("^Adria\\.1", chemoMapping[, "TableColumn"]),
+     "MatchedSample"]
> cellLinesFromPredictors[["Group1", "Doxorubicin"]] <- substr(temp,
+     1, nchar(temp) - 2)
> cellLinesFromPredictors[, "Doxorubicin"]

$Group0
 [1] "SF-539"     "SNB-75"     "MDA-MB-435" "NCI-H23"    "M14"
 [6] "MALME-3M"   "SK-MEL-2"   "SK-MEL-28"  "SK-MEL-5"   "UACC-62"

$Group1
 [1] "NCI/ADR-RES" "HCT-15"      "HT29"        "EKVX"        "NCI-H322M"
 [6] "IGROV1"      "OVCAR-3"     "OVCAR-4"     "OVCAR-5"     "OVCAR-8"
[11] "SK-OV-3"     "CAKI-1"
```

5.3 Paclitaxel

Paclitaxel (Taxol):

```
> temp <- chemoMapping[grep("^Taxol\\.0", chemoMapping[, "TableColumn"]),
+     "MatchedSample"]
> cellLinesFromPredictors[["Group0", "Paclitaxel"]] <- substr(temp,
+     1, nchar(temp) - 2)
> temp <- chemoMapping[grep("^Taxol\\.1", chemoMapping[, "TableColumn"]),
+     "MatchedSample"]
> cellLinesFromPredictors[["Group1", "Paclitaxel"]] <- substr(temp,
+     1, nchar(temp) - 2)
> cellLinesFromPredictors[, "Paclitaxel"]

$Group0
[1] "SF-295"     "SF-539"     "HS 578T"    "MDA-MB-435" "COLO 205"
[6] "HCC-2998"   "HT29"       "OVCAR-3"    "DU-145"

$Group1
[1] "CCRF-CEM"  "SW-620"    "A549/ATCC" "EKVX"      "MALME-3M"  "SK-MEL-28"
[7] "OVCAR-8"   "786-0"
```

5.4 Fluorouracil

Fluorouracil (5-FU):

```
> temp <- chemoMapping[grep("^X5\\.FU\\.0", chemoMapping[, "TableColumn"]),
+     "MatchedSample"]
> cellLinesFromPredictors[["Group0", "Fluorouracil"]] <- substr(temp,
+     1, nchar(temp) - 2)
> temp <- chemoMapping[grep("^X5\\.FU\\.1", chemoMapping[, "TableColumn"]),
+     "MatchedSample"]
> cellLinesFromPredictors[["Group1", "Fluorouracil"]] <- substr(temp,
+     1, nchar(temp) - 2)
> cellLinesFromPredictors[, "Fluorouracil"]

$Group0
[1] "MCF7"     "COLO 205" "HCT-116"  "NCI-H460" "LOX IMVI" "SK-MEL-5" "A498"
[8] "UO-31"

$Group1
[1] "NCI/ADR-RES" "MDA-MB-435"  "SW-620"      "EKVX"        "M14"
[6] "OVCAR-8"     "SN12C"
```

5.5 Cyclophosphamide

Cyclophosphamide (Cytoxan), from Chang data:

```
> temp <- chemoMappingChang[grep("^Cytox\\.0", chemoMappingChang[,
+     "TableColumn"]), "MatchedSample"]
```

```
> cellLinesFromPredictors[["Group0", "Cyclophosphamide"]] <- temp
> temp <- chemoMappingChang[grep("^Cytox\\.1", chemoMappingChang[,
+     "TableColumn"]), "MatchedSample"]
> cellLinesFromPredictors[["Group1", "Cyclophosphamide"]] <- temp
> cellLinesFromPredictors[, "Cyclophosphamide"]

$Group0
 [1] "N19" "N20" "N22" "N24" "N18" "N16" "N21" "N17" "N12" "N11"

$Group1
 [1] "N5"  "N9"  "N3"  "N7"  "N4"  "N1"  "N2"  "N10" "N8"  "N6"
```

5.6 Topotecan

Topotecan:

```
> temp <- chemoMapping[grep("^Topo\\.0", chemoMapping[, "TableColumn"]),
+     "MatchedSample"]
> cellLinesFromPredictors[["Group0", "Topotecan"]] <- substr(temp,
+     1, nchar(temp) - 2)
> temp <- chemoMapping[grep("^Topo\\.1", chemoMapping[, "TableColumn"]),
+     "MatchedSample"]
> cellLinesFromPredictors[["Group1", "Topotecan"]] <- substr(temp,
+     1, nchar(temp) - 2)
> cellLinesFromPredictors[, "Topotecan"]

$Group0
 [1] "SF-539"   "SNB-75"   "U251"     "HS 578T"  "HOP-62"   "NCI-H226"
 [7] "NCI-H23"  "LOX IMVI" "OVCAR-8"  "A498"     "ACHN"     "CAKI-1"
[13] "UO-31"

$Group1
 [1] "K-562"      "RPMI-8226"  "MDA-MB-435" "SK-MEL-5"   "HCC-2998"
 [6] "HCT-116"    "HCT-15"     "NCI-H322M"  "SK-MEL-28"  "COLO 205"
```

5.7 Etoposide

Etoposide:

```
> temp <- chemoMapping[grep("^Etopo\\.0", chemoMapping[, "TableColumn"]),
+     "MatchedSample"]
> cellLinesFromPredictors[["Group0", "Etoposide"]] <- substr(temp,
+     1, nchar(temp) - 2)
> temp <- chemoMapping[grep("^Etopo\\.1", chemoMapping[, "TableColumn"]),
+     "MatchedSample"]
> cellLinesFromPredictors[["Group1", "Etoposide"]] <- substr(temp,
+     1, nchar(temp) - 2)
> cellLinesFromPredictors[, "Etoposide"]
```

```
$Group0
[1] "SF-539"          "BT-549"          "MDA-MB-231/ATCC" "NCI/ADR-RES"
[5] "HOP-62"          "NCI-H226"        "SK-MEL-28"       "UACC-257"
[9] "786-0"

$Group1
[1] "MCF7"      "HCC-2998"  "HCT-15"    "SW-620"    "NCI-H322M" "PC-3"
[7] "OVCAR-4"   "OVCAR-5"
```

5.8 Checking the Numbers of Lines

Finally, we take a look at the numbers of cell lines in each drug/group combination.

```
> cellLinesFromPredictors
```

```
       Docetaxel   Doxorubicin  Paclitaxel  Fluorouracil Cyclophosphamide
Group0 Character,7 Character,10 Character,9 Character,8  Character,10
Group1 Character,7 Character,12 Character,8 Character,7  Character,10
       Topotecan    Etoposide
Group0 Character,13 Character,9
Group1 Character,10 Character,8
```

6 Save Rda File

Finally, we save the lists we assembled.

```
> save(cellLinesFromPredictors, file = file.path("RDataObjects",
+     "cellLinesFromPredictors.Rda"))
```

7 Appendix

7.1 File Location

```
> getwd()
```

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

7.2 Saves

7.3 SessionInfo

```
> sessionInfo()
```

```
R version 2.8.1 (2008-12-22)
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
```

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