Matching the Cisplatin Heatmap

Keith A. Baggerly

September 24, 2009

Contents

List of Figures

List of Tables

1 Executive Summary

1.1 Introduction

Hsu et al. [3] construct a signature for cisplatin using data from cell lines provided by Györffy et al. [2]. In explaining the biological plausibility of this signature, they note that

> "The cisplatin sensitivity predictor includes DNA repair genes such as ERCC1 and ERCC4, among others, that had altered expression in the list of cisplatin sensitivity predictor genes. Interestingly, one previously described mechanism of resistance to cisplatin therapy results from the increased capacity of cancer cells to repair DNA damage incurred, by activation of DNA repair genes."

In this report, we outline our reconstruction of the heatmap provided, and our inferences about the specific genes and cell lines involved.

1.2 Methods

We loaded two previously constructed Rda files: gyorffyAll and hsuReportedGeneLists. We extracted quantifications for the reported probesets, and examined heatmaps to see if there was clear separation of sensitive and resistant cell lines. We repeated this procedure after "offsetting" the probesets by one row to account for possible problems with binreg. We then constructed pairwise sample correlation matrices to suggest the specific cell lines used in each group. Starting from this initial guess, we then examined all other sets of cell lines in a local "neighborhood" and assigned each set a "score" of the number of reported probesets (offset or not) that were matched. We shifted to the set with the highest score in the neighborhood and repeated the process until a local maximum was reached. This scoring makes use of Matlab scripts which call the binreg software; the primary file is buildCisplatinHeatmap.m.

1.3 Results

Using probesets "off-by-one" from those reported, we were able to perfectly reconstruct the heatmap reported, thus identifying the specific genes and cell lines involved. The cell lines are
Resistant (9): 257p, A375, C8161, ES-2, ME-43, MeWo, SKMel19, SNU423, and SW13, and
Sensitive (6): BT20, DV-90, FU-OV-1, OAW42, OVCAR3, and R103.

Though the heatmap matches perfectly, we match only 41 of the 45 genes reported. The four genes we don't match are 203719_at (ERCC1), 210158_at (ERCC4), 228131_at (ERCC1), and 231971_at (FANCM, associated with DNA Repair). The last two of these could not be matched because they are not physically present on the U133A arrays used by Györffy et al. [2]; they're on the U133B.

The various intermediate files and gene lists for cisplatin are saved in RDataObjects as cisplatinAll.Rda.

1.4 Conclusions

The reported signature for cisplatin is incorrect due to an off-by-one indexing error, and should not include the specific genes named by Hsu et al. [3] as evidence of plausibility.

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: gyorffyAll and hsuReportedGenelists.

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

4 Matching the Reported Genes

In the context of using the NCI60 cell lines, Hsu et al. [3] note that "if a drug screening experiment did not result in widely variable GI50/IC50 and/or LC50 data, the generation of a predictor is not possible using our methods, as in the case of cisplatin." Thus, they use data on 30 cell lines published by Györffy et al. [2] Hsu et al. [3] selected 15 cell lines, 9 resistant and 6 sensitive. From these, the binreg software was used to select the 45 genes having the most extreme two-sample t-test values separating resistant from sensitive.

We loaded the list of reported genes above. We now check that all of these genes reported are present in the Györffy data matrix.

```
> unmatchedRows <- which(is.na(match(cisplatinReportedProbesets[,
+     "probesetID"], rownames(gyorffyAll))))
> unmatchedRows

[1] 44 45

> cisplatinReportedProbesets[unmatchedRows, ]

          probesetID geneSymbol
228131_at  228131_at      ERCC1
231971_at  231971_at      FANCM

> matchedCisplatinProbesets <- rownames(cisplatinReportedProbesets)[-unmatchedRows]
> cisplatinTable[cisplatinTable[, "Gene.Title"] == "231971_at",
+     ]

    Gene.Title                                Gene.Symbol
175  231971_at Fanconi anemia, complementation group M
    GO.Biological.Process.Description GO.Molecular.Function.Description
175                             FANCM                          DNA repair
    GO.Cellular.Component.Description
175                nucleotide binding
```

The last two reported probesets, 228131_at (ERCC1), and 231971_at (FANCM, a gene associated with DNA repair), don't match. Checking the GeneCard entries for ERCC1 (`http://www.genecards.org/cgi-bin/carddisp?gene=ERCC1`) and FANCM (`http://www.genecards.org/cgi-bin/carddisp?gene=FANCM`) shows the reason: these probesets are on the U133B chip platform, not the U133A. Györffy et al. [2] supply no data on these probesets because they didn't measure them. For now, we proceed with the remaining 43 genes.

5 Drawing Heatmaps of the Reported (and Offset) Genes

5.1 A Heatmap of the Reported Genes

We now draw a heatmap showing the expression levels of the reported genes across all of the samples. Since these genes were chosen specifically to separate one group of cell lines from another, we expect to see some clear differential structure. We center and scale the results for each row (probeset) before display (the heatmap function in R does this scaling by default). We also use an analog of the Matlab jet colormap. The heatmap is shown in Figure 1. There is very limited structure visible, and what there is appears to be driven largely by two probesets: 209771_x_at and 208650_s_at.

5.2 A Heatmap of the Reported Genes, Offset

Coombes et al. [1] noted that all of the gene lists initially reported by Potti et al. [4] had been "offset by one" due to an indexing error. We try introducing the same type of offset (e.g., replacing 200075_s_at with 200076_s_at) and redrawing the heatmap. This heatmap is shown in Figure 2. There is now clearly visible structure separating groups of cell lines. This suggests that the probesets shown were part of the "true" signature, and that the reported list is incorrect due to the same indexing error that affected the initial gene lists reported by Potti et al. [4].

6 Using Correlation to Guess Starting Cell Lines

We know of no simple analytic way of determining which cell lines were involved, and the names are not given in Hsu et al. [3] or in their supplementary material. We rely instead on a combination of informed guessing and brute force. We begin by looking at correlations between samples using the expression values for the offset genes identified above. A heatmap is shown in Figure 3. Looking at the correlation heatmap suggests some clear candidates for the larger (resistant) group: the 10 samples in the cluster at the top right. In alphabetical order, these are 257p, A375, C8161, ES-2, ME-43, MeWo, SKMel13, SKMel19, SNU423, and SW13. Likewise, the 7 samples in the cluster at the bottom left are good candidates for the smaller (sensitive) group. In alphabetical order, these are BT20, DV-90, FU-OV-1, OAW42, OVCAR3, R103, and SKOV-3. We will use these as starting guesses.

```
> startingResistantLines <- c("257p", "A375", "C8161", "ES-2",
+     "ME-43", "MeWo", "SKMel13", "SKMel19", "SNU423", "SW13")
> startingSensitiveLines <- c("BT20", "DV-90", "FU-OV-1", "OAW42",
+     "OVCAR3", "R103", "SKOV-3")
> gyorffyAllInfo[startingResistantLines, "Cisplatin"]

 [1] R R R R R R M R R R
Levels: M R S

> gyorffyAllInfo[startingSensitiveLines, "Cisplatin"]

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

The starting lists involve 10 and 7 cell lines, respectively, whereas Hsu et al. [3] use 9 and 6. Checking the sensitive/resistant labels assigned to these cell lines (a) confirms that our overall sensitive/resistant orientation is correct and (b) suggests which cell lines might be outliers: SKMel13 in the Resistant list, and SKOV-3 in the Sensitive list.

```
> tempMat <- gyorffyAll[matchedCisplatinProbesets, ]
> source(file.path("Scripts", "jet.colors.R"))
> heatmap(as.matrix(tempMat), col = jet.colors(64), margins = c(6,
+     6), cexRow = 0.75, cexCol = 0.9)
```

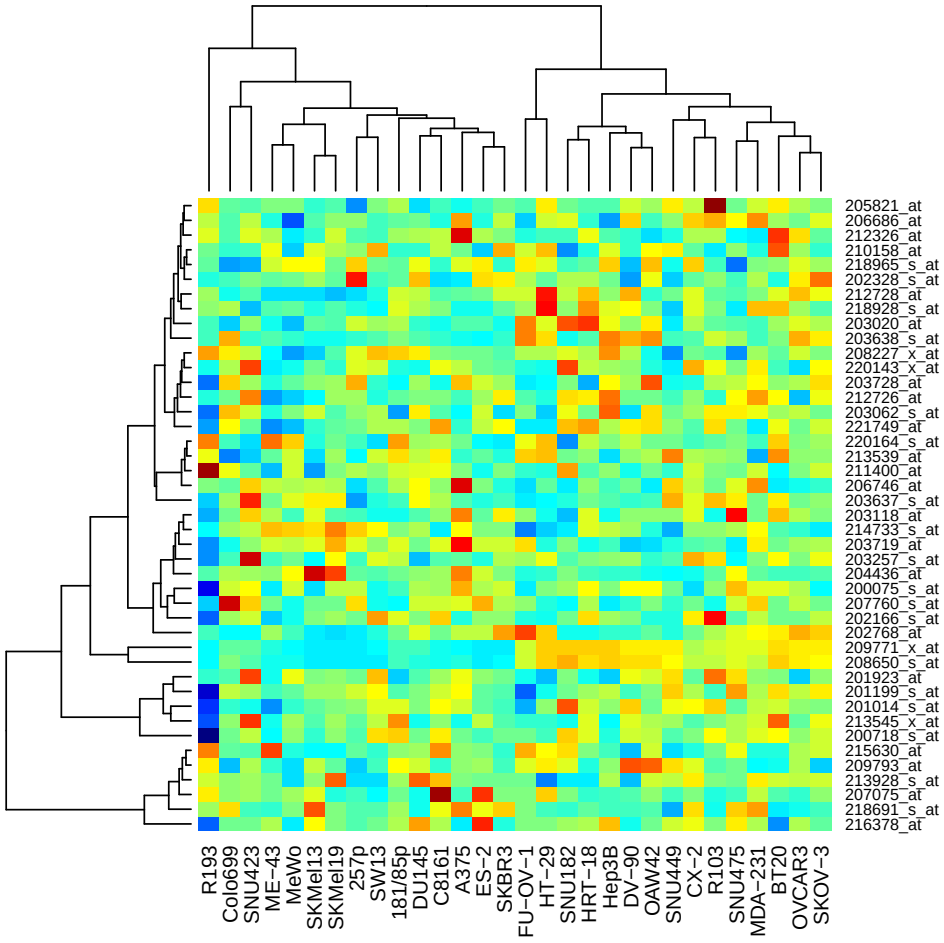

Figure 1: Heatmap of centered expression values from the Győrffy et al. [2] expression data for the matchable probesets reported by Hsu et al. [3] As these genes were chosen to separate sensitive from resistant lines, we expect to see fairly pervasive structure separating two subgroups. We do not see this structure.

```
> offsetCisplatinProbesets <- rownames(gyorffyAll)[match(matchedCisplatinProbesets,
+     rownames(gyorffyAll)) + 1]
> tempMat <- gyorffyAll[offsetCisplatinProbesets, ]
> heatmap(as.matrix(tempMat), col = jet.colors(64), margins = c(6,
+     6), cexRow = 0.75, cexCol = 0.9)
```

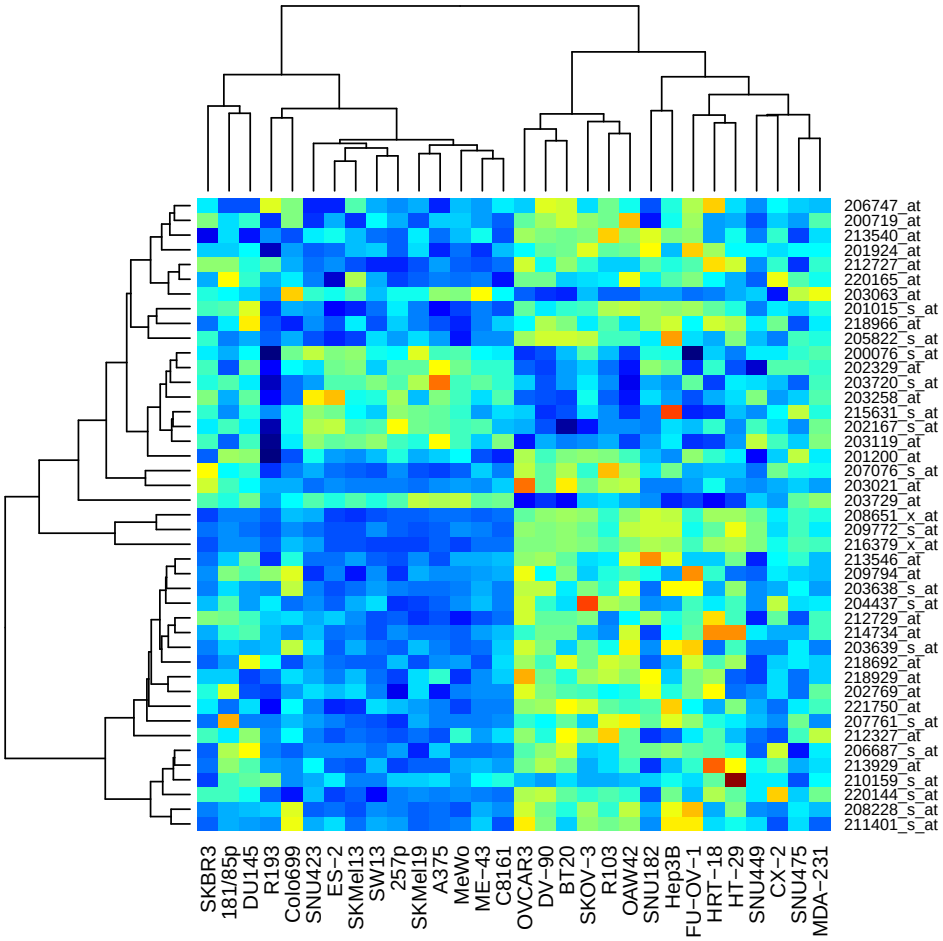

Figure 2: Heatmap of centered expression values from the Györffy et al. [2] expression data for the matchable probesets reported by Hsu et al. [3] after "offsetting" by one row. As genes were chosen to separate sensitive from resistant lines, we expect to see fairly pervasive structure separating two subgroups. We see this structure here, suggesting that the reported list is incorrect due to an indexing error.

```
> cisplatinCor <- cor(t(scale(t(tempMat))))
> heatmap(cisplatinCor, scale = "none", margins = c(6, 6), cexCol = 0.9,
+     cexRow = 0.9, col = jet.colors(64))
```

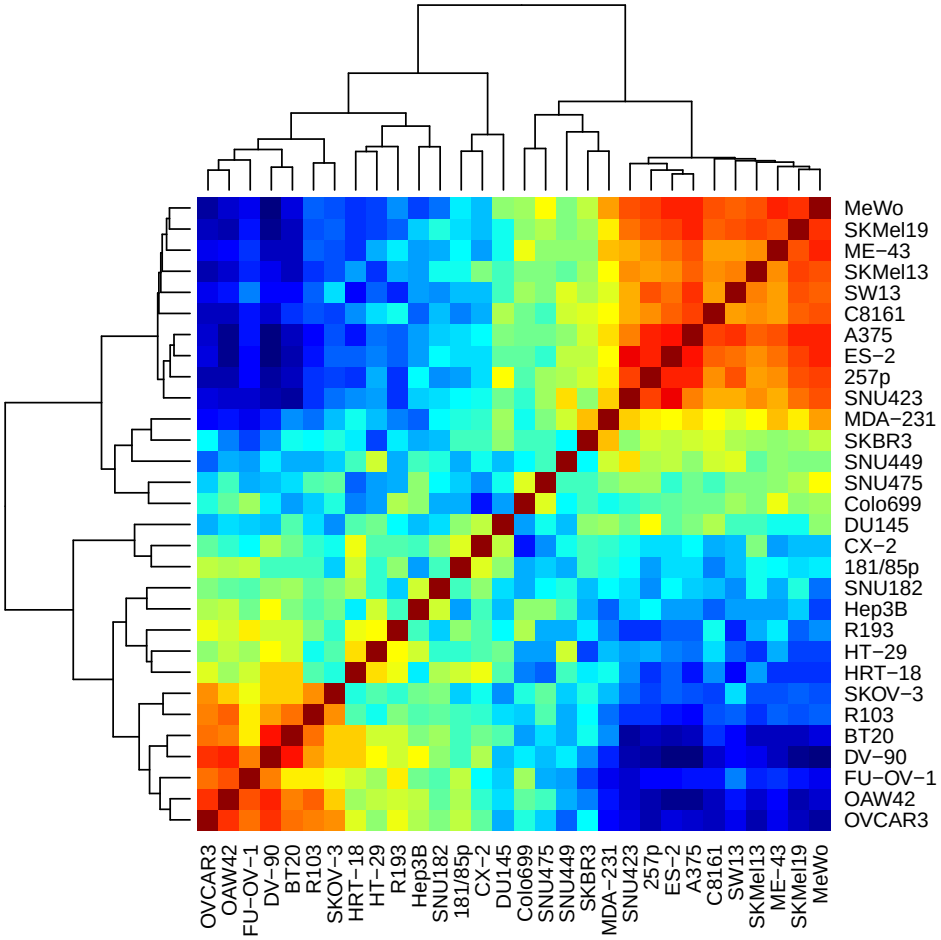

Figure 3: Heatmap of correlations between samples using the offset probesets from Figure 2. The clearest groups of cell lines are those in the upper right and lower left.

7 Steepest Ascent with Binreg

In order to improve our guess, we use a process of steepest ascent with the binreg software used by Potti et al. [4] Specifically, we give each set of cell lines a score corresponding to the number of target probesets we can match. From our starting set, we score all sets that can be reached by changing the status of at most one cell line (e.g., from sensitive to resistant, from sensitive to excluded, from excluded to resistant). We then shift our central location to the set in the neighborhood with the highest score and repeat until a local maximum is reached.

7.1 Exporting the Györffy Data

To conduct our search, we first export gyorffyAll in a format usable by binreg.

```
> write.table(gyorffyAll, file = file.path("MatlabFiles", "GyorffyData",
+     "gyorffyNumbers.csv"), sep = ",", row.names = FALSE, col.names = FALSE)
> write.table(rownames(gyorffyAll), file = file.path("MatlabFiles",
+     "GyorffyData", "gyorffyProbesets.csv"), sep = ",", row.names = FALSE,
+     col.names = FALSE, quote = FALSE)
> write.table(colnames(gyorffyAll), file = file.path("MatlabFiles",
+     "GyorffyData", "gyorffySamples.csv"), sep = ",", row.names = FALSE,
+     col.names = FALSE, quote = FALSE)
```

7.2 Exporting the Starting Guess

We likewise export our starting guesses for the resistant and sensitive cell lines, and the target probeset ids that we want to match.

```
> write.table(startingResistantLines, file = file.path("MatlabFiles",
+     "Cisplatin", "startingResistantLines.csv"), sep = ",", row.names = FALSE,
+     col.names = FALSE, quote = FALSE)
> write.table(startingSensitiveLines, file = file.path("MatlabFiles",
+     "Cisplatin", "startingSensitiveLines.csv"), sep = ",", row.names = FALSE,
+     col.names = FALSE, quote = FALSE)
> write.table(offsetCisplatinProbesets, file = file.path("MatlabFiles",
+     "Cisplatin", "targetProbesets.csv"), sep = ",", row.names = FALSE,
+     col.names = FALSE, quote = FALSE)
```

7.3 Invoking Matlab

The main Matlab script is buildCisplatinHeatmap.m. This script iteratively explores the neighborhood to increase the score until a maximum is found. At each iteration, it produces a figure indicating how the current guess should be changed. For cisplatin, this search and change process takes three steps, as illustrated in Figures 4-6. The first shift producing the maximum score (moving SKOV-3 from Sensitive to Unused) increases the score from 30 to 36. The second shift producing the maximum score (moving SKMel-13 from Resistant to Unused) increases the score from 36 to 41. This is a local maximum; any shift from the current guess decreases the score. We note in passing that the two cell lines dropped are those that might have been guessed based on the cell line status information shown at the end of the previous section.

After reaching a local maximum, we produce a heatmap using the final set of cell lines selected. This heatmap, shown in Figure 7, perfectly matches the one shown in Figure 1A of Hsu et al. [3], even though only 41 of the 43 matchable (offset) probesets are matched.

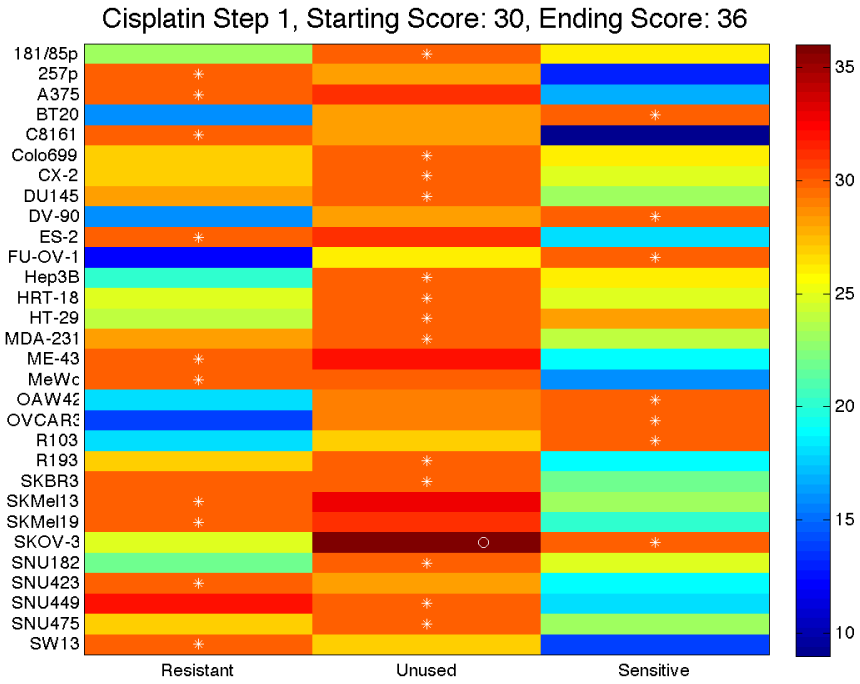

Figure 4: First iteration to find an overall maximum for cisplatin. White asterisks indicate the status of each cell line using the starting guess, and colors indicate how the score increases or decreses as a given change is made. The shift producing the maximum score (moving SKOV-3 from Sensitive to Unused) is indicated with an offset circle. This shift increases the score from 30 to 36.

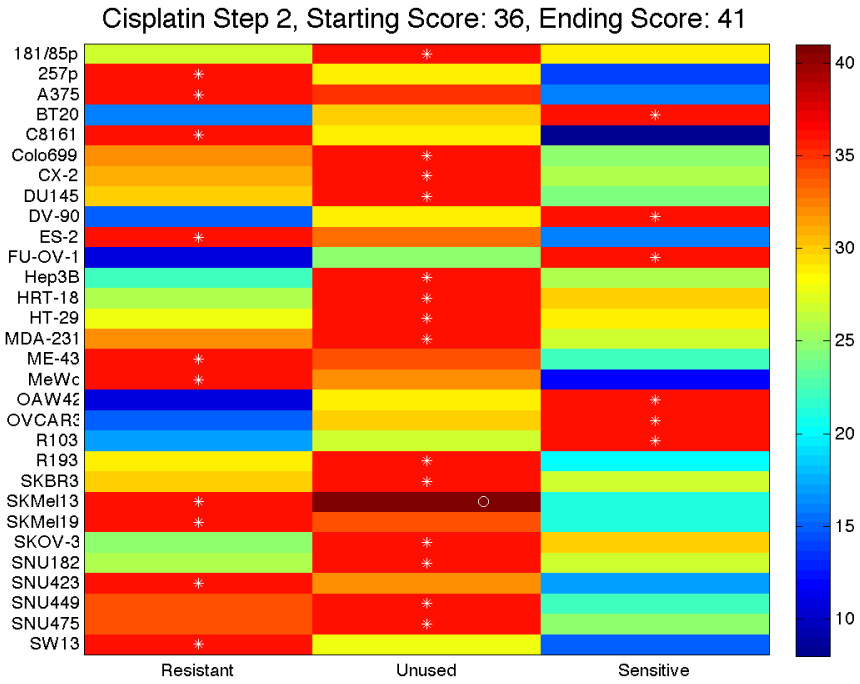

Figure 5: Second iteration to find an overall maximum for cisplatin. White asterisks indicate the status of each cell line using the starting guess, and colors indicate how the score increases or decreses as a given change is made. The shift producing the maximum score (moving SKMel-13 from Resistant to Unused) is indicated with an offset circle. This shift increases the score from 36 to 41.

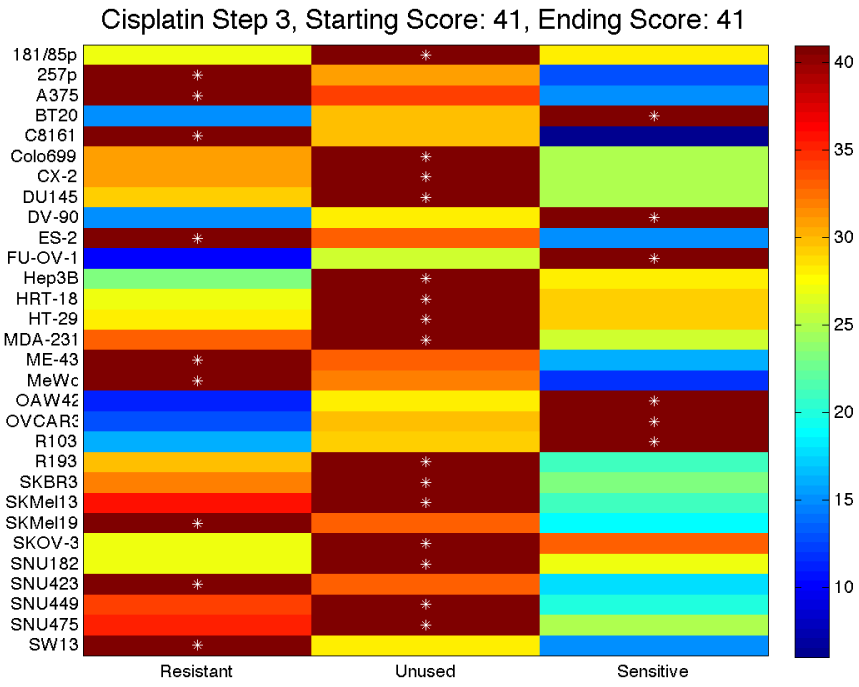

Figure 6: Third iteration to find an overall maximum for cisplatin. White asterisks indicate the status of each cell line using the starting guess, and colors indicate how the score increases or decreses as a given change is made. Any shift from the current guess decreases the score. Our best score is 41.

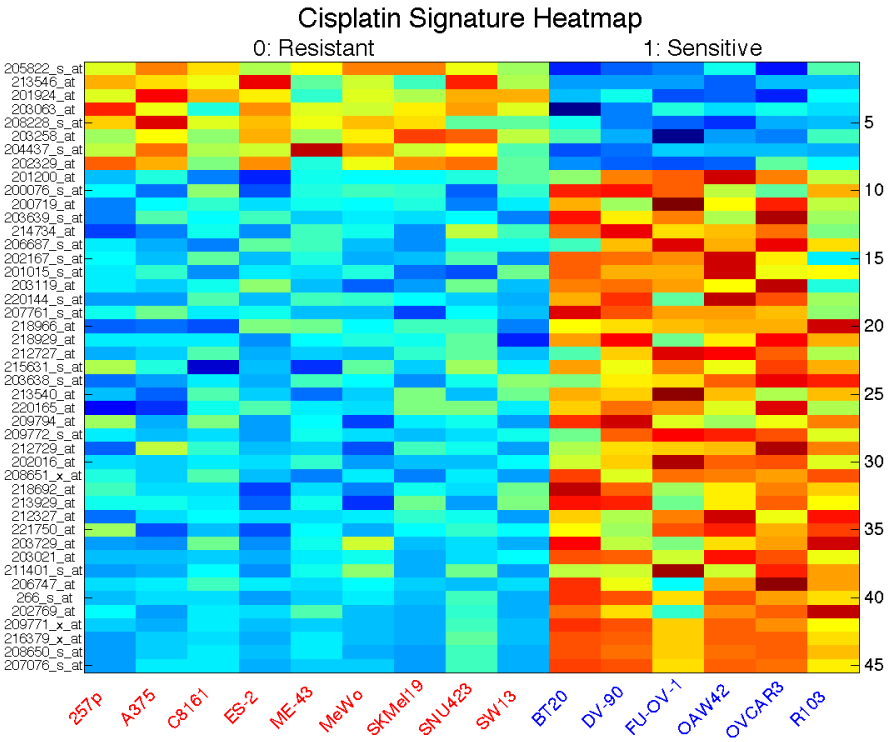

Figure 7: Heatmap for the cisplatin signature using the final set of cell lines obtained through our search procedure. This heatmap is a perfect match for that in Figure 1A of Hsu et al. [3], indicating that these are the cell lines and genes involved.

8 Loading Matlab Output and Comparing Gene Lists

8.1 Loading Scores from Each Iteration

We begin by loading the numerical scores and starting vectors for each iteration of the fitting procedure.

```
> cisplatinIteration1 <- read.table(file.path("MatlabFiles", "Cisplatin",
+     "cisplatinIteration1.csv"), header = TRUE, sep = ",", row.names = 1)
> cisplatinIteration2 <- read.table(file.path("MatlabFiles", "Cisplatin",
+     "cisplatinIteration2.csv"), header = TRUE, sep = ",", row.names = 1)
> cisplatinIteration3 <- read.table(file.path("MatlabFiles", "Cisplatin",
+     "cisplatinIteration3.csv"), header = TRUE, sep = ",", row.names = 1)
> cisplatinIteration3[1:5, ]
```

```
        startStatus Resistant Unused Sensitive
181/85p      Unused        27     41        28
257p      Resistant        41     31        13
A375      Resistant        41     34        15
BT20      Sensitive        15     30        41
C8161     Resistant        41     30         6
```

8.2 Extracting the Cell Lines Used

Next, we extract the cell lines used to obtain the perfectly matching heatmap.

```
> cisplatinResistantLines <- rownames(cisplatinIteration3)[cisplatinIteration3[,
+     "startStatus"] == "Resistant"]
> cisplatinSensitiveLines <- rownames(cisplatinIteration3)[cisplatinIteration3[,
+     "startStatus"] == "Sensitive"]
> cisplatinResistantLines
```

```
[1] "257p"    "A375"    "C8161"   "ES-2"    "ME-43"   "MeWo"    "SKMel19"
[8] "SNU423"  "SW13"
```

```
> cisplatinSensitiveLines
```

```
[1] "BT20"    "DV-90"   "FU-OV-1" "OAW42"   "OVCAR3"  "R103"
```

8.3 Loading the Heatmap Genes

Next, we load the probesets used in the final heatmap.

```
> softwareCisplatinProbesets <- read.table(file.path("MatlabFiles",
+     "Cisplatin", "topCisplatinGenesInHeatmapOrder.txt"), header = FALSE,
+     sep = ",", strip.white = TRUE)
> softwareCisplatinProbesets <- as.character(softwareCisplatinProbesets[,
+     1])
> sort(softwareCisplatinProbesets)
```

```
 [1] "200076_s_at" "200719_at"   "201015_s_at" "201200_at"   "201924_at"
 [6] "202016_at"   "202167_s_at" "202329_at"   "202769_at"   "203021_at"
[11] "203063_at"   "203119_at"   "203258_at"   "203638_s_at" "203639_s_at"
[16] "203729_at"   "204437_s_at" "205822_s_at" "206687_s_at" "206747_at"
[21] "207076_s_at" "207761_s_at" "208228_s_at" "208650_s_at" "208651_x_at"
[26] "209771_x_at" "209772_s_at" "209794_at"   "211401_s_at" "212327_at"
[31] "212727_at"   "212729_at"   "213540_at"   "213546_at"   "213929_at"
[36] "214734_at"   "215631_s_at" "216379_x_at" "218692_at"   "218929_at"
[41] "218966_at"   "220144_s_at" "220165_at"   "221750_at"   "266_s_at"
```

8.4 Comparing Gene Lists

Now we see which genes we can and cannot match.

```
> sum(!is.na(match(offsetCisplatinProbesets, softwareCisplatinProbesets)))

[1] 41

> setdiff(softwareCisplatinProbesets, offsetCisplatinProbesets)

[1] "202016_at"   "266_s_at"    "209771_x_at" "208650_s_at"
```

As noted above, the software output only matches 41 of the genes reported. For 2 of the 4 genes dropped, 208650_s_at and 209771_x_at, the next probeset in sequence is also in the list produced by the software, and hence these two are reintroduced into the reported list after the offset. These two probesets were identified above as the source of what visible structure there was using the probesets reported.

We now look at which genes were introduced, after adjusting for offsetting.

```
> softwareCisplatinMinus1Probesets <- rownames(gyorffyAll)[match(softwareCisplatinProbesets,
+     rownames(gyorffyAll)) - 1]
> setdiff(rownames(cisplatinReportedProbesets), softwareCisplatinMinus1Probesets)

[1] "203719_at" "210158_at" "228131_at" "231971_at"

> cisplatinReportedProbesets[setdiff(rownames(cisplatinReportedProbesets),
+     softwareCisplatinMinus1Probesets), ]

          probesetID geneSymbol
203719_at  203719_at      ERCC1
210158_at  210158_at      ERCC4
228131_at  228131_at      ERCC1
231971_at  231971_at      FANCM
```

The four genes introduced (using the names from the initially reported list before the offset) are 203719_at (ERCC1), 210158_at (ERCC4), 228131_at (ERCC1, U133B chip), and 231971_at (FANCM, U133B chip). These 4 genes are the ones specifically identified by Hsu et al. [3] as being of special interest, but their software does not produce them.

9 Save Rda File

Finally, we save the data associated with our examination of the cisplatin signature.

```
> save(cisplatinCor, cisplatinIteration1, cisplatinIteration2,
+     cisplatinIteration3, cisplatinReportedProbesets, cisplatinResistantLines,
+     cisplatinSensitiveLines, cisplatinTable, matchedCisplatinProbesets,
+     offsetCisplatinProbesets, softwareCisplatinMinus1Probesets,
+     softwareCisplatinProbesets, file = file.path("RDataObjects",
+         "cisplatinAll.Rda"))
```

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] geneplotter_1.22.0  lattice_0.17-25     annotate_1.22.0
[4] AnnotationDbi_1.6.1 Biobase_2.4.1       XML_2.6-0

loaded via a namespace (and not attached):
[1] DBI_0.2-4          grid_2.9.1         KernSmooth_2.23-2  RColorBrewer_1.0-2
[5] RSQLite_0.7-1      xtable_1.5-5
```

References

[1] Coombes KR, Wang J, Baggerly KA: Microarrays: retracing steps. *Nat Med*, **13**:1276-7, 2007. Author reply, 1277-8.

[2] Györffy B, Surowiak P, Kiesslich O, et al.: Gene expression profiling of 30 cancer cell lines predicts resistance towards 11 anticancer drugs at clinically achieved concentrations. *Int J Cancer*, **118**:1699-712, 2006.

[3] Hsu DS, Balakumaran BS, Acharya CR, et al.: Pharmacogenomic strategies provide a rational approach to the treatment of cisplatin-resistant patients with advanced cancer. *J Clin Oncol*, **25**:4350-4357, 2007

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