

# Looking at Cisplatin and Pemetrexed Signatures in Lung Cancer

Keith A. Baggerly and Kevin R. Coombes

October 14, 2007

## 1 Introduction

A major goal of personalized medicine is to predict, before administering a treatment, whether the patient will respond to it. Recently, Hsu et al. [3] used genomic signatures to make such predictions in the context of lung cancer, focusing on cisplatin and pemetrexed. Starting with expression profiles of cell lines with known sensitivity or resistance to the target agents, they select the most and least sensitive to contrast. The genes showing the biggest differences are the elements of the genomic signature for that drug, with weights ideally ascribed based on the consistency of separation. Given this weighted signature, both the cell line data and profiles of patient samples are then scored, with higher scores indicating a greater probability of response.

This approach closely tracks that of Potti et al. [4], where profiles and sensitivity information for the NCI60 cell lines were used to assemble signatures for 7 drugs. Unfortunately, we encountered problems with the Potti et al. report. These problems, documented in Coombes et al. [1], are severe enough that we believe the conclusions to be invalid. In this context, then, we are curious about the effectiveness of the approach in lung cancer.

## 2 Options and Libraries

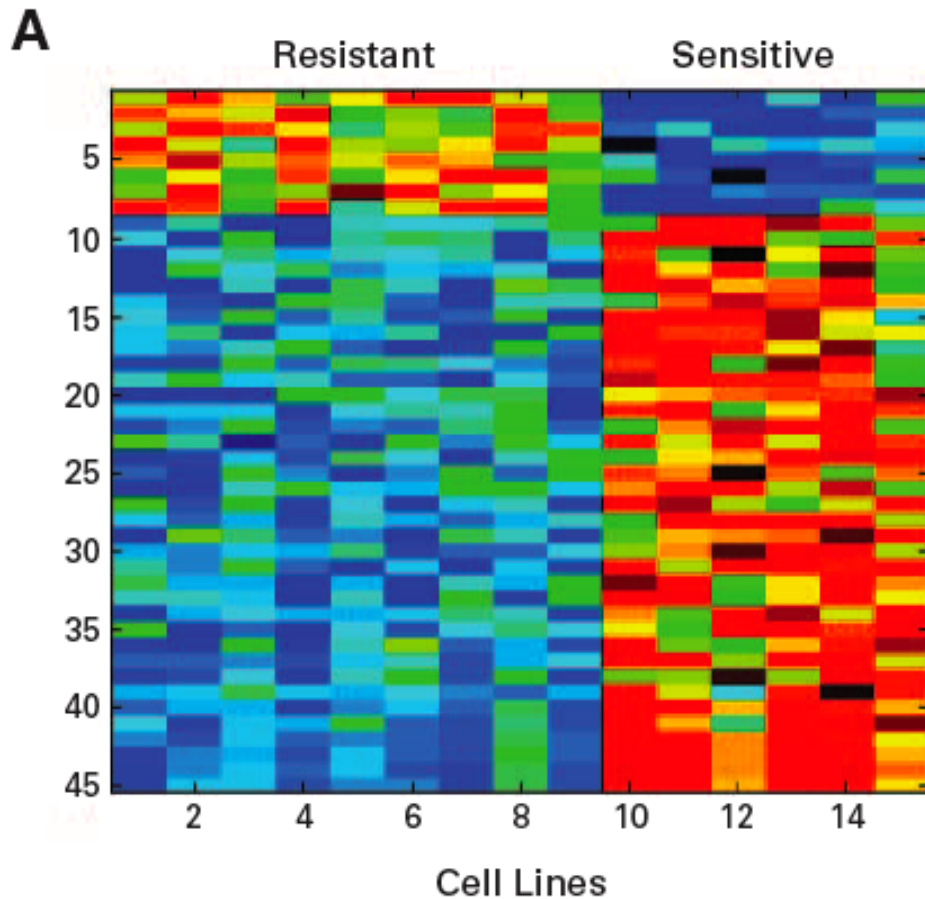
```
> options(width = 80)
> library(ClassComparison)
> library(ClassDiscovery)
```

The ClassComparison and ClassDiscovery packages are available from our website, <http://bioinformatics.mdanderson.org>.

## 3 Cisplatin

We begin with cisplatin. In the context of using the NCI60 profiles here, Hsu et al. 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 turn to data on 30 cell lines published by Gyorffy et al. [2].

Fifteen of the Gyorffy cell lines were selected, 9 resistant and 6 sensitive. A panel of 45 genes was selected, producing a separating heatmap. These 45 genes are chosen as the set having the most extreme two-sample t-test values separating resistant from sensitive. The reported heatmap is shown below.



Hsu et al.'s gene list for cisplatin is given in a supplementary table (table 1) at the JCO website, <http://jco.ascopubs.org/cgi/content/full/25/28/4350/DC1>. We load the probeset ids (but not the annotation) below.

```
> reportedCisplatinGenes <- c("200075_s_at", "200718_s_at", "201014_s_at",
+ "201199_s_at", "201923_at", "202166_s_at", "202328_s_at",
+ "202768_at", "203020_at", "203062_s_at", "203118_at", "203257_s_at",
+ "203637_s_at", "203638_s_at", "203719_at", "203728_at", "204436_at",
+ "205821_at", "206686_at", "206746_at", "207075_at", "207760_s_at",
+ "208227_x_at", "208650_s_at", "209771_x_at", "209793_at",
+ "210158_at", "211400_at", "212326_at", "212726_at", "212728_at",
+ "213539_at", "213545_x_at", "213928_s_at", "214733_s_at",
+ "215630_at", "216378_at", "218691_s_at", "218928_s_at", "218965_s_at",
+ "220143_x_at", "220164_s_at", "221749_at", "228131_at", "231971_at")
```

Hsu et al. 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.

### 3.1 The Gyorffy et al. Data

The raw data from Gyorffy et al. is available from the *International Journal of Cancer* website, <http://www.interscience.wiley.com/jpages/0020-7136/suppmat/ijc.21570.html>. Supporting table 1, in particular, gives the probeset by sample table of expression values. The 30 cell lines were profiled using Affymetrix U133A arrays. The text of Gyorffy et al. states that several quantification methods (including MAS5.0 and RMA) were used to check robustness, but only one set of numbers is reported. While we do not know precisely which method generated these numbers, we do know that they are on a log scale, as the maximum value seen is just under 15. We acquired the table, saved it in csv format for easier loading, and import it here.

```
> gyorffy <- read.table(file.path("OtherData", "Gyorffy", "gyorffy_data.csv"),
+   header = TRUE, row.names = 1, sep = ",", check.names = FALSE)
> dim(gyorffy)

[1] 22283    30

> gyorffy[1:3, 1:3]

           181P    257P    A375
1007_s_at 9.977162 8.830360 9.232587
1053_at   8.509234 7.975630 8.601956
117_at    7.320390 7.674556 7.571382
```

### 3.2 Surveying the Reported Genes, and Offset

In order to see the separating gene expression structure, we assemble a heatmap of all of the reported genes across all samples. We begin by finding the matching rows.

```
> match(reportedCisplatinGenes, rownames(gyorffy))

[1] 96 246 542 727 1451 1694 1856 2296 2549 2591 2647 2784
[13] 3164 3165 3246 3255 3963 5348 6212 6272 6601 7281 7732 8147
[25] 9259 9281 9642 10815 11711 12111 12113 12920 12926 13307 14109 15003
[37] 15749 18055 18292 18329 19507 19528 21109 NA NA

> reportedCisplatinGenes[is.na(match(reportedCisplatinGenes, rownames(gyorffy)))]

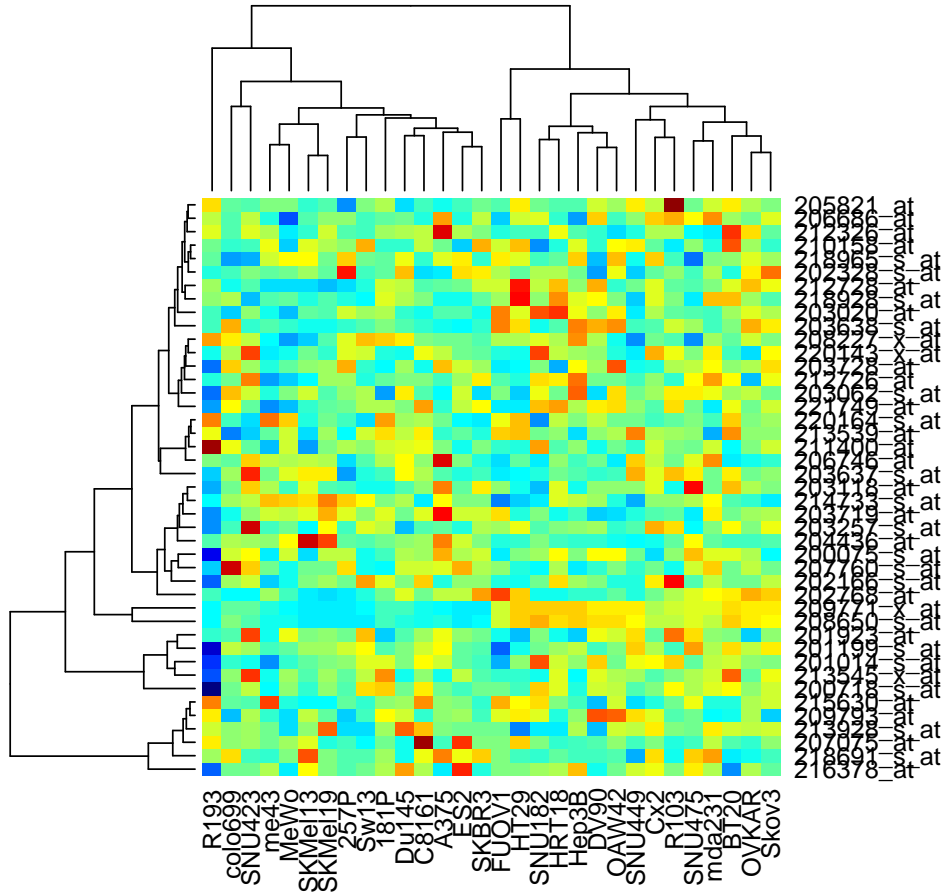
[1] "228131_at" "231971_at"
```

At this point we encounter our first snag. We can't find the data for two of the reported probesets: (1) 228131\_at, ERCC1, and (2) 231971\_at, FANCM, another gene associated with DNA repair. Checking over at GeneCards 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 for the difficulty: these probesets are on the U133B chip, not the U133A. Given that Gyorffy et al. supply no data on these probesets because they didn't measure them, we're not sure how they were chosen. For now, we will proceed with the remaining 43 genes. In order to better track the images reported in Hsu et al., we center and scale the results for each row (probeset) before display if necessary (the heatmap function does such scaling by default). We also use an analog of the Matlab jet colormap.

```

> tempMat <- gyorffy[reportedCisplatinGenes[1:43], ]
> source(file.path("OtherData", "jet.colors.R"))
> heatmap(as.matrix(tempMat), col = jet.colors(64), margins = c(8,
+ 8), cexRow = 1.1, cexCol = 1.1)

```



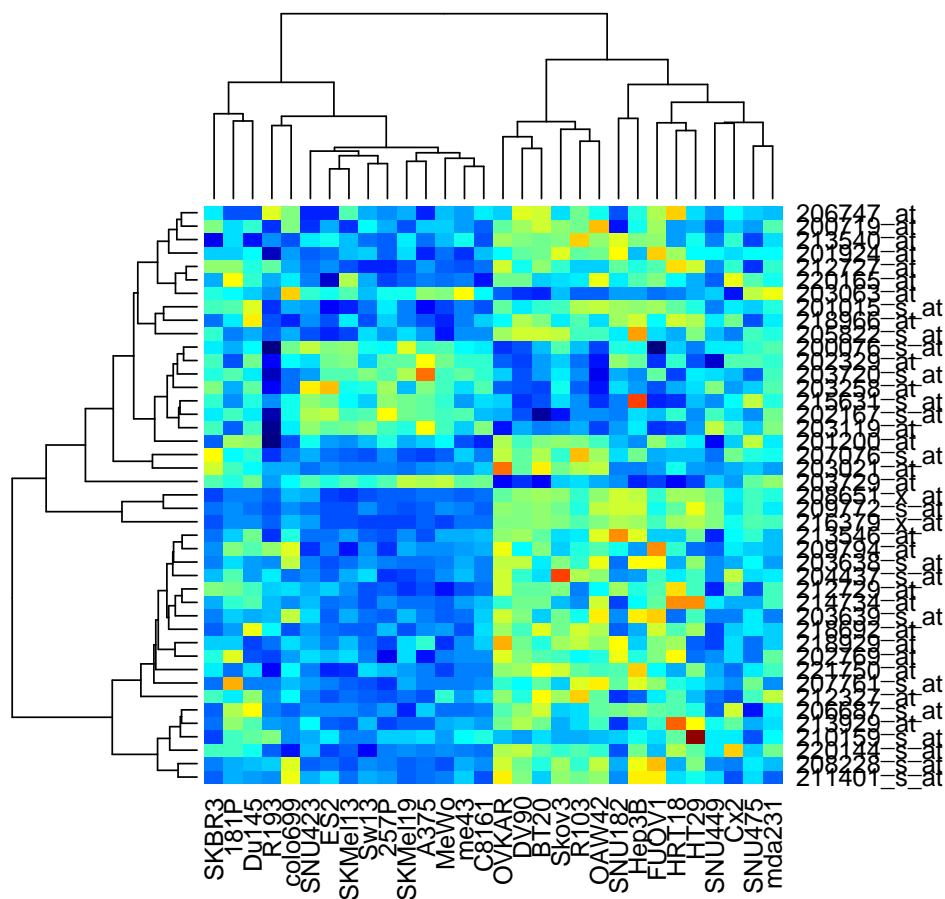
Actually, this is not what we wanted to see. There is no particularly visible structure in the heatmap.

This problem, having heatmaps with no visible structure, was also noted by Coombes et al. in looking at the data from Potti et al. There, the problem was that all of the probeset ids had been offset by one due to an indexing bug. We try introducing the same type of offset here.

```

> offsetCisplatinGenes <- rownames(gyorffy)[match(reportedCisplatinGenes[1:43],
+ rownames(gyorffy)) + 1]
> tempMat <- gyorffy[offsetCisplatinGenes[1:43], ]
> heatmap(as.matrix(tempMat), col = jet.colors(64), margins = c(8,
+ 8), cexRow = 1.1, cexCol = 1.1)

```

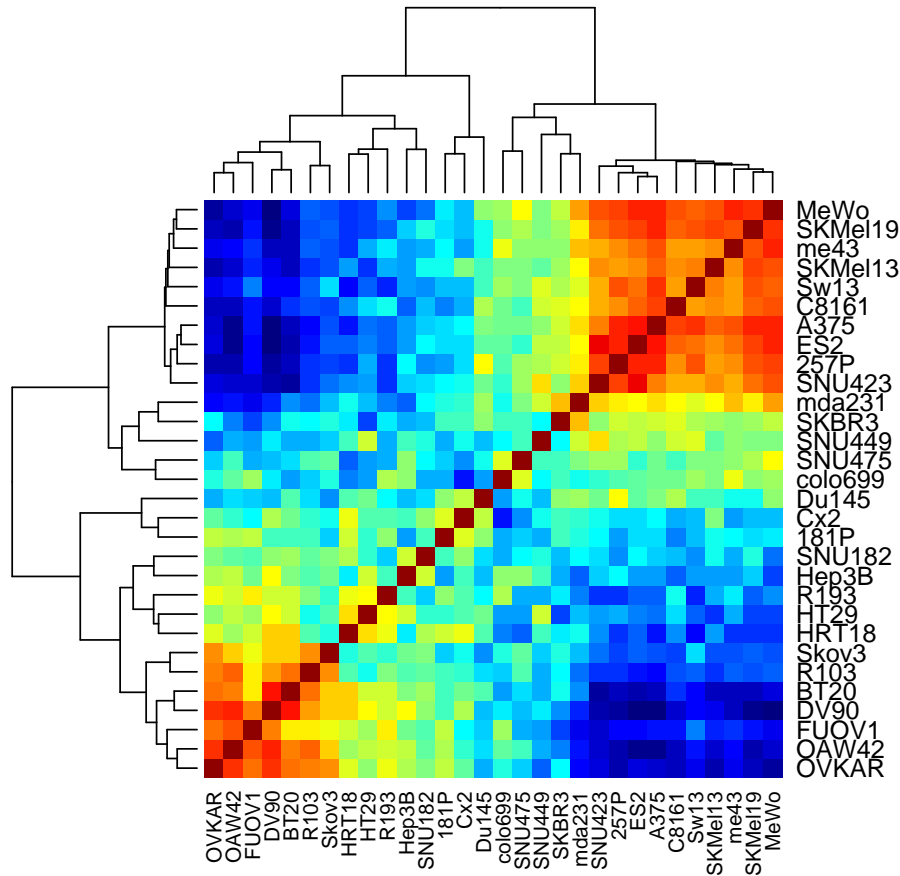


This is much better. There are clear subgroups of genes with locally antithetical behavior (high/low, and low/high).

### 3.3 Identifying the Samples Involved

We know of no simple analytic way of determining precisely which cell lines were involved, and the names are not given in the paper or the supplementary materials. We rely on a combination of informed guessing at brute force. We begin by looking at the correlations between samples using the expression values for the genes chosen above.

```
> cisplatinCor <- cor(t(scale(t(tempMat))))
> heatmap(cisplatinCor, scale = "none", margins = c(8, 8), cexRow = 1.1,
+   col = jet.colors(64))
```



Looking at the correlation heatmap, there are some clear candidates for the group of resistant samples: 257P, A375, C8161, ES2, me43, MeWo, SKMel13, SKMel19, SNU423, Sw13 (from the cluster at top right, in alphabetical order). Likewise, there are some good candidates for the sensitive lines: BT20, DV90, FUOV1, HRT18, OAW42, OVKAR, R103, Skov3 (from the cluster at bottom left, in alphabetical order). We assigned the “sensitive” and “resistant” labels to the two clusters on the based on the number of genes in the expression heatmap showing higher levels in that group, and matching this with the expression heatmap from the paper.

The numbers of “guessed” resistant and sensitive samples (10 and 8) are close enough to the final goals of 9 and 6 that we can simply produce heatmaps for all possible combinations produced by subsetting, and visually compare these heatmaps with Figure 1A from Hsu et al. The number of genes higher in the resistant group varies widely with the combination of samples, so filtering pictures isn’t too difficult. Following this comparison, we believe the resistant and sensitive lines used to produce the Hsu et al. cisplatin heatmap are

- **Resistant:** 257P, A375, C8161, ES2, me43, MeWo, SKMel19, SNU423, and Sw13,
- **Sensitive:** BT20, DV90, FUOV1, OAW42, OVKAR, and R103,

in that order.

```

> cisplatinResistantLines <- c("257P", "A375", "C8161", "ES2",
+   "me43", "MeWo", "SKMe119", "SNU423", "Sw13")
> cisplatinSensitiveLines <- c("BT20", "DV90", "FUOV1", "OAW42",
+   "OVKAR", "R103")

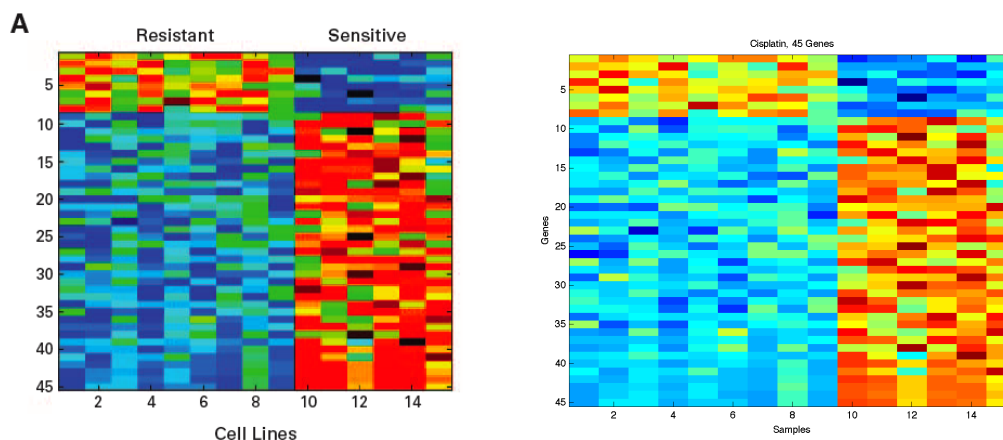
```

As a quick check, we look at the status of these samples according to Gyorffy et al. (shown in Figure 2 of that paper). All of the lines used are indeed listed by Gyorffy et al. as resistant or sensitive, respectively, so that is consistent. (The cell line labeled OVCAR3 in Gyorffy et al.'s Figure 2 is labeled OVKAR in the table of quantifications.) However, other cell lines in the set of 30 examined were also labeled as sensitive or resistant; only 3 of the 30 were labeled as indeterminate. We have no idea how or why this subset of just 15 samples was selected.

### 3.4 Duke Software Output

We obtained the Duke quantification software previously as part of our analysis for Coombes et al. Our script invoking this software to produce our cisplatin heatmap is stored in MatlabFiles as CisplatinGyorffy.m, with supporting files of CisplatinResponsePredictorProbesets.csv, gyorffy\_numbers.csv, and scripts from the Duke software in Pittman\_zip. Because the Gyorffy numbers are already on the log scale, we do not invoke the normalization option in the Duke software.

Figure 1A from Hsu et al. and the cisplatin heatmap we produced are juxtaposed below.



This is an exact match modulo shading, so we are working with the data used to produce the initial figure.

The Duke software also lists the probesets used in the heatmap. Here, these are stored in MatlabFiles in both CisplatinTrainingImageGeneList.txt and CisplatinTrainingTopGeneCoeffs.txt. We grab the gene list from the latter file, and check it against the list reported by Hsu et al.

```

> temp <- read.table(file.path("MatlabFiles", "CisplatinTrainingTopGeneCoeffs.txt"))
> temp[1:3, ]

```

```

      V1      V2
1 3.459336 208650_s_at
2 3.414082 216379_x_at
3 2.954229 209771_x_at

```

```

> softwareCisplatinGenes <- as.character(temp[, 2])
> softwareCisplatinGenes <- sort(softwareCisplatinGenes)
> softwareCisplatinGenes[1:4]
[1] "200076_s_at" "200719_at" "201015_s_at" "201200_at"
> sum(!is.na(match(offsetCisplatinGenes, softwareCisplatinGenes)))
[1] 41
> setdiff(softwareCisplatinGenes, offsetCisplatinGenes)
[1] "202016_at" "208650_s_at" "209771_x_at" "266_s_at"
> setdiff(offsetCisplatinGenes, softwareCisplatinGenes)
[1] "203720_s_at" "210159_s_at"
> softwareCisplatinMinus1Genes <- rownames(gyorffy)[match(softwareCisplatinGenes,
+ rownames(gyorffy)) - 1]
> setdiff(softwareCisplatinGenes, softwareCisplatinMinus1Genes)
[1] "203719_at" "210158_at" "228131_at" "231971_at"

```

After accounting for the off-by-one error, we match the reported list for 41 of the 45 genes. The four dropped from our list are 202016\_at, 208650\_s\_at, 209771\_x\_at, and 266\_s\_at. The four new outliers (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). All 4 of these are associated with “DNA repair”.

We now check the two-sample t-statistics for the two outlier probesets on the U133A chip.

```

> tempMat <- gyorffy[, c(cisplatinResistantLines, cisplatinSensitiveLines)]
> cisT <- MultiTtest(tempMat, classes = as.factor(rep(c("Resistant",
+ "Sensitive"), times = c(9, 6))))
> cisT@p.values[rownames(gyorffy) == "203719_at"]
[1] 0.03828598
> cisT@p.values[rownames(gyorffy) == "210158_at"]
[1] 0.5072549
> range(sort(cisT@p.values[match(softwareCisplatinGenes, rownames(gyorffy))]))
[1] 7.760450e-10 2.903084e-04
> range(sort(cisT@p.values[match(offsetCisplatinGenes, rownames(gyorffy))]))
[1] 7.760450e-10 6.249453e-01
> range(sort(cisT@p.values[match(offsetCisplatinGenes, rownames(gyorffy))])[1:41])
[1] 7.760450e-10 2.903084e-04
> range(sort(cisT@p.values[match(softwareCisplatinGenes, rownames(gyorffy))]))
[1] 5.650401e-09 9.937137e-01

```

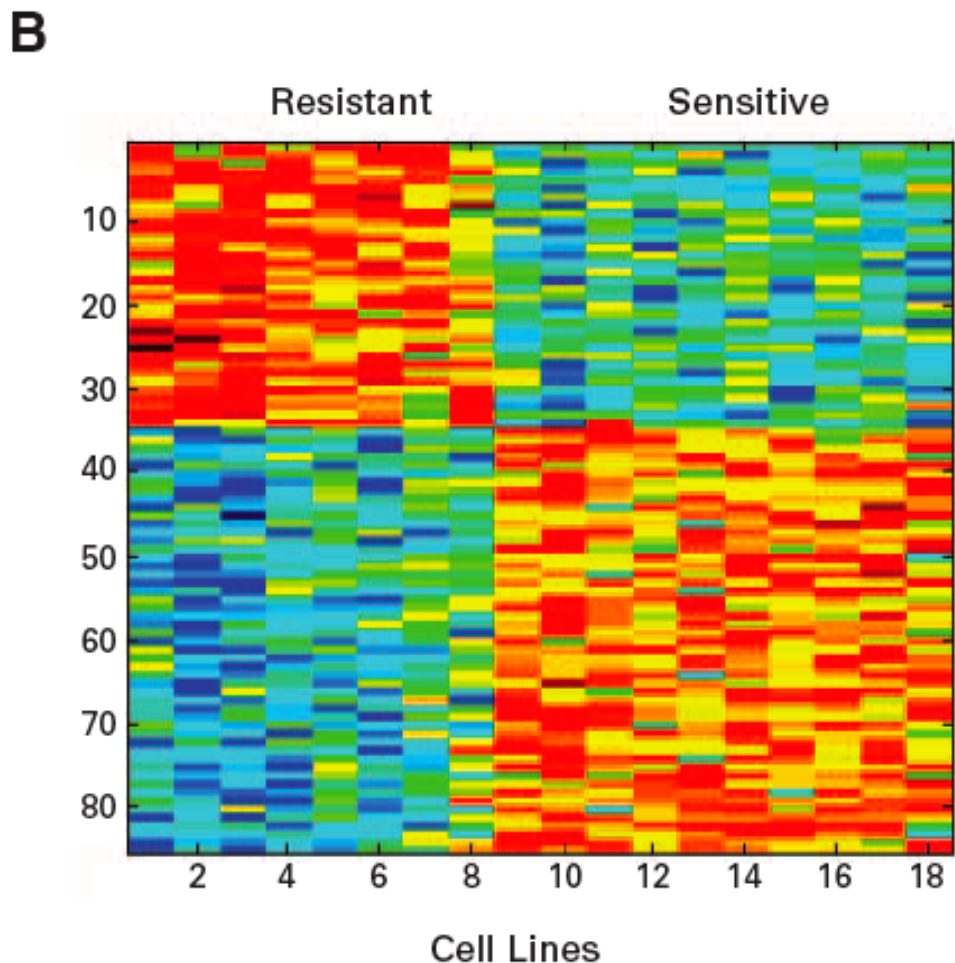
The two new p-values are not small relative to the others, particularly the one for 210158\_at, ERCC4. We see no way for these outliers to be selected on the basis of contrasting the resistant and sensitive cell lines.



## 4 Pemetrexed

We now turn to pemetrexed. Here, Hsu et al. were able to make use of the NCI60 cell line data to construct a signature. We collected this data as part of the analysis performed in Coombes et al.; here we make use of some of the RDataObjects created for that report. Details are given at <http://bioinformatics.mdanderson.org/Supplements/ReproRsch-Chemo>.

Eighteen of the NCI60 cell lines were selected, 8 resistant and 10 sensitive. A panel of 85 genes was assembled, producing a separating heatmap. This heatmap, Figure 1B from Hsu et al., is shown below.



Hsu et al.'s gene list for pemetrexed is also given in a supplementary table (table 2) at the JCO website, <http://jco.ascopubs.org/cgi/content/full/25/28/4350/DC1>. We load the probeset ids (but not the annotation) below.

```
> reportedPemetrexedGenes <- c("1100_at", "1227_g_at", "1318_at",
+   "1355_g_at", "241_g_at", "242_at", "31462_f_at", "31510_s_at",
+   "31537_at", "31545_at", "32225_at", "32251_at", "32259_at",
+   "32317_s_at", "32433_at", "32573_at", "32748_at", "32835_at",
```

```
+ "32892_at", "33144_at", "33361_at", "33377_at", "33451_s_at",
+ "33613_at", "33854_at", "33918_s_at", "34245_at", "34318_at",
+ "34858_at", "34859_at", "35351_at", "35434_at", "35699_at",
+ "35747_at", "35762_at", "36118_at", "36191_at", "36535_at",
+ "36584_at", "36988_at", "37344_at", "37374_at", "37484_at",
+ "37744_r_at", "37746_r_at", "38119_at", "38287_at", "38404_at",
+ "38478_at", "38545_at", "38908_s_at", "39018_at", "39149_at",
+ "39169_at", "39247_at", "39328_at", "39329_at", "39350_at",
+ "39543_at", "39749_at", "39797_at", "39799_at", "40212_at",
+ "40327_at", "40393_at", "40492_at", "40683_at", "40821_at",
+ "40854_at", "40864_at", "40952_at", "41127_at", "41234_at",
+ "41402_at", "41435_at", "41442_at", "41448_at", "41459_at",
+ "41643_at", "41738_at", "41757_at", "41833_at", "41853_at",
+ "590_at", "797_at")
```

#### 4.1 The NCI60/Novartis Data

There are actually several different sets of array profiles of the NCI60 cell lines. Most can be accessed from <http://discover.nci.nih.gov>. As noted in Coombes et al., the particular data used by Potti et al. (and, we conjecture, by Hsu et al.) came from U95Av2 arrays run by Novartis, with 59 samples (MDA-N is missing) run in triplicate. The arrays are indexed with and A, B, or C to designate the replicate set. Only the numbers from the A set were used. Further, probesets corresponding to Affymetrix control probes were excluded, leaving 12558 probesets as opposed to 12625. More details on the Novartis samples and the R data objects we use here are available at <http://bioinformatics.mdanderson.org/Supplements/ReproRsch-Chemo>.

```
> load(file.path("RDataObjects", "novartis.Rda"))
> load(file.path("RDataObjects", "chemoPredictors.Rda"))
> dim(novartis)
```

```
[1] 12625  180
```

```
> dim(predictors)
```

```
[1] 12558  121
```

```
> novartis[1:3, 1:3]
```

```
      A.786-0  A.A498 A.A549/ATCC
36460_at  93.56745  77.9148   51.38758
36461_at 148.71699 109.6346   63.00203
36462_at 249.93599 275.4420  340.17966
```

```
> predictors[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
```

Here, we want to extract just the A set arrays, and restrict our attention to the 12558 key probesets.

```
> novA <- novartis[rownames(predictors), grep("^A", colnames(novartis))]
> dim(novA)
```

```
[1] 12558    59
```

```
> colnames(novA)
```

```
[1] "A.786-0"           "A.A498"           "A.A549/ATCC"
[4] "A.ACHN"           "A.BT-549"         "A.CAKI-1"
[7] "A.CCRF-CEM"       "A.COLO 205"       "A.DU-145"
[10] "A.EKVX"           "A.HCC-2998"       "A.HCT-116"
[13] "A.HCT-15"         "A.HL-60(TB)"      "A.HOP-62"
[16] "A.HOP-92"         "A.HS 578T"        "A.HT29"
[19] "A.IGROV1"         "A.K-562"          "A.KM12"
[22] "A.LOX IMVI"       "A.M14"            "A.MALME-3M"
[25] "A.MCF7"           "A.MDA-MB-231/ATCC" "A.MDA-MB-435"
[28] "A.MOLT-4"         "A.NCI-H226"       "A.NCI-H23"
[31] "A.NCI-H322M"      "A.NCI-H460"       "A.NCI-H522"
[34] "A.NCI/ADR-RES"    "A.OVCAR-3"        "A.OVCAR-4"
[37] "A.OVCAR-5"        "A.OVCAR-8"        "A.PC-3"
[40] "A.RPMI-8226"      "A.RXF 393"        "A.SF-268"
[43] "A.SF-295"         "A.SF-539"         "A.SK-MEL-2"
[46] "A.SK-MEL-28"     "A.SK-MEL-5"       "A.SK-OV-3"
[49] "A.SN12C"          "A.SNB-19"         "A.SNB-75"
[52] "A.SR"             "A.SW-620"         "A.T-47D"
[55] "A.TK-10"          "A.U251"           "A.UACC-257"
[58] "A.UACC-62"        "A.UO-31"
```

```
> novAInfo <- novartisInfo[grep("^A", rownames(novartisInfo)),
+ ]
> dim(novAInfo)
```

```
[1] 59    5
```

```
> tempNames <- unlist(lapply(strsplit(colnames(novA), "^A\\.\\."),
+   function(x) {
+     x[2]
+   }))
> tempNames[1:3]
```

```
[1] "786-0"    "A498"    "A549/ATCC"
```

```
> colnames(novA) <- tempNames
```

The ordering of sample names given here is simply alphabetical. This differs from the ordering used by Potti et al. in their other predictor lists, where they are first grouped by panel (e.g., breast, lung, colon, etc.). Here, we reorganize the columns to match their ordering.

First, we extract the individual panels.

```
> levels(novAInfo$pname)
```

```
[1] "Breast" "CNS" "Colon" "Leukemia" "Melanoma" "NSCLC" "Ovarian"
[8] "Prostate" "Renal"

> for (panel in levels(novAInfo$pname)) {
+   assign(paste("panel", panel, sep = "."), as.character(novAInfo$cellname[novAInfo$pname ==
+   panel]))
+ }
```

Within the panels, there are a few cases where the samples are not listed in alphabetical order in the Potti et al. predictors. To match the Potti et al. names, we adjust the ordering for the Breast and Leukemia panels.

```
> predictorsInfo[!is.na(match(predictorsInfo$Source, panel.Breast)),
+ ]
```

	index	drugName	responseStatus	Source	NovartisName
	3	Adria	Resistant	MDA-MB-435	A.MDA-MB-435
	11	Adria	Sensitive	NCI/ADR-RES	A.NCI/ADR-RES
	24	Etopo	Resistant	BT-549	A.BT-549
	25	Etopo	Resistant	MDA-MB-231/ATCC	A.MDA-MB-231/ATCC
	26	Etopo	Resistant	NCI/ADR-RES	A.NCI/ADR-RES
	32	Etopo	Sensitive	MCF7	A.MCF7
	40	5-FU	Resistant	MCF7	A.MCF7
	48	5-FU	Sensitive	NCI/ADR-RES	A.NCI/ADR-RES
	49	5-FU	Sensitive	MDA-MB-435	A.MDA-MB-435
	58	Cyttox	Resistant	MCF7	A.MCF7
	61	Cyttox	Sensitive	HS 578T	A.HS 578T
	62	Cyttox	Sensitive	MDA-MB-231/ATCC	A.MDA-MB-231/ATCC
	63	Cyttox	Sensitive	MDA-MB-435	A.MDA-MB-435
	71	Topo	Resistant	HS 578T	A.HS 578T
	83	Topo	Sensitive	MDA-MB-435	A.MDA-MB-435
	93	Taxol	Resistant	HS 578T	A.HS 578T
	94	Taxol	Resistant	MDA-MB-435	A.MDA-MB-435

```
> predictorsInfo[!is.na(match(predictorsInfo$Source, panel.Leukemia)),
+ ]
```

	index	drugName	responseStatus	Source	NovartisName
	55	Cyttox	Resistant	K-562	A.K-562
	56	Cyttox	Resistant	MOLT-4	A.MOLT-4
	57	Cyttox	Resistant	HL-60(TB)	A.HL-60(TB)
	81	Topo	Sensitive	K-562	A.K-562
	82	Topo	Sensitive	RPMI-8226	A.RPMI-8226
	100	Taxol	Sensitive	CCRF-CEM	A.CCRF-CEM
	115	Doce	Sensitive	HL-60(TB)	A.HL-60(TB)

```
> panel.Breast
```

```
[1] "BT-549" "HS 578T" "MCF7" "MDA-MB-231/ATCC"
[5] "MDA-MB-435" "NCI/ADR-RES" "T-47D"
```

```

> panel.Breast <- panel.Breast[c(1:4, 6, 5, 7)]
> panel.Leukemia

[1] "CCRF-CEM" "HL-60(TB)" "K-562"      "MOLT-4"      "RPMI-8226" "SR"

> panel.Leukemia <- panel.Leukemia[c(1, 3:4, 2, 5:6)]

```

For Breast, NCI/ADR-RES comes before MDA-MB-435, based on the ordering of the Sensitive samples for 5FU. For Leukemia, HL-60(TB) comes after K-562 and MOLT-4, based on the ordering of the Resistant samples for Cytoxan. For the other panels, the information we have is either consistent with alphabetical ordering (CNS, NSCLC, Ovarian, Prostate, Renal) or has some ambiguities (Colon, Melanoma). The details are not shown here (the checks are exactly the same as those performed above), and alphabetical ordering is retained for these panels. Finally, the panels themselves are arranged in the order shown below.

```

> novA <- novA[, c(panel.Leukemia, panel.CNS, panel.Breast, panel.Colon,
+   panel.NSCLC, panel.Melanoma, panel.Prostate, panel.Ovarian,
+   panel.Renal)]

```

## 4.2 Surveying the Reported Genes, and Offset

We now assemble a heatmap of the genes reported for pemetrexed across all samples. As with cisplatin, the first step is to assure that all of the reported probesets can be matched.

```

> match(reportedPemetrexedGenes, rownames(novA))

[1] 6154 6288 6383 6423 7284 7285 7512 7561 7587 7595 8282 8308
[13] 8316 8375 8492 8633 8810 8898 8955 9210 9429 9445 9520 9684
[25] 9927 9992 10322 10396 10941 10942 11439 11523 11790 11839 11854 12214
[37] 12287 77 126 532 892 922 1033 1296 1298 1674 1843 1961
[49] 2035 2103 2470 2581 2713 2733 2812 2894 2895 2916 3111 3319
[61] 3367 3369 3786 3902 3968 4068 4261 4401 4434 4444 4533 4710
[73] 4818 4987 5020 5027 5033 5044 5230 5326 5345 5422 5442 5630
[85] 5834

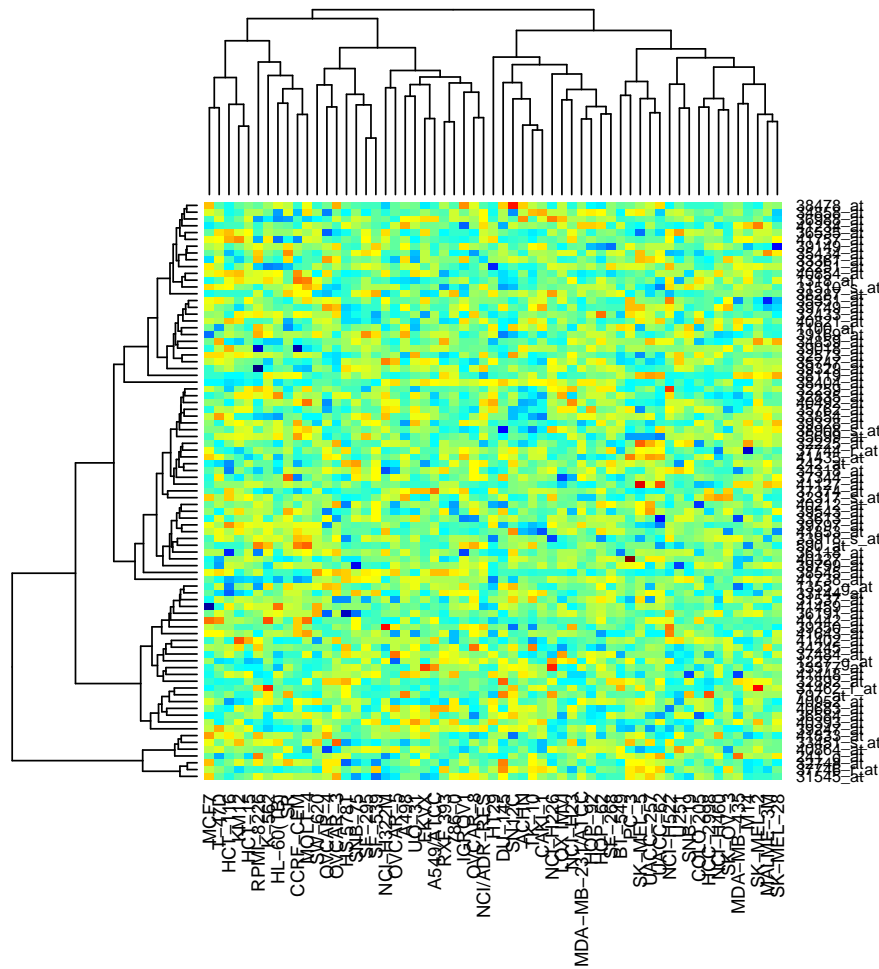
```

There are no NAs this time, which is good. Now we assemble the heatmap after log2-transforming the expression values (which are MAS5.0) to match the scale used by the Duke software.

```

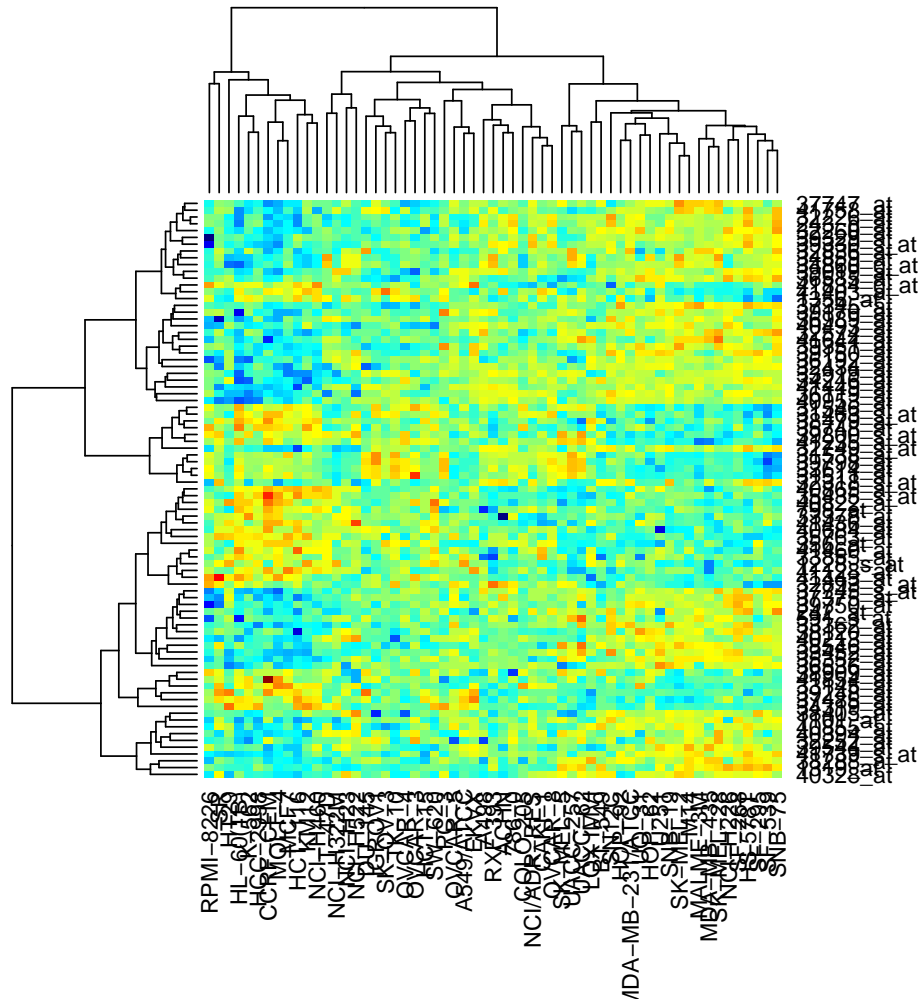
> temp1 <- log2(novA[reportedPemetrexedGenes, ])
> heatmap(as.matrix(temp1), col = jet.colors(64), margins = c(8,
+   8))

```



As with cisplatin, there is no immediately visible structure here. Next, we offset the probset ids by one.

```
> offsetPemetrexedGenes <- rownames(novA)[match(reportedPemetrexedGenes,
+       rownames(novA)) + 1]
> temp1 <- log2(novA[offsetPemetrexedGenes, ])
> heatmap(as.matrix(temp1), col = jet.colors(64), margins = c(8,
+       8), cexRow = 1, cexCol = 1)
```



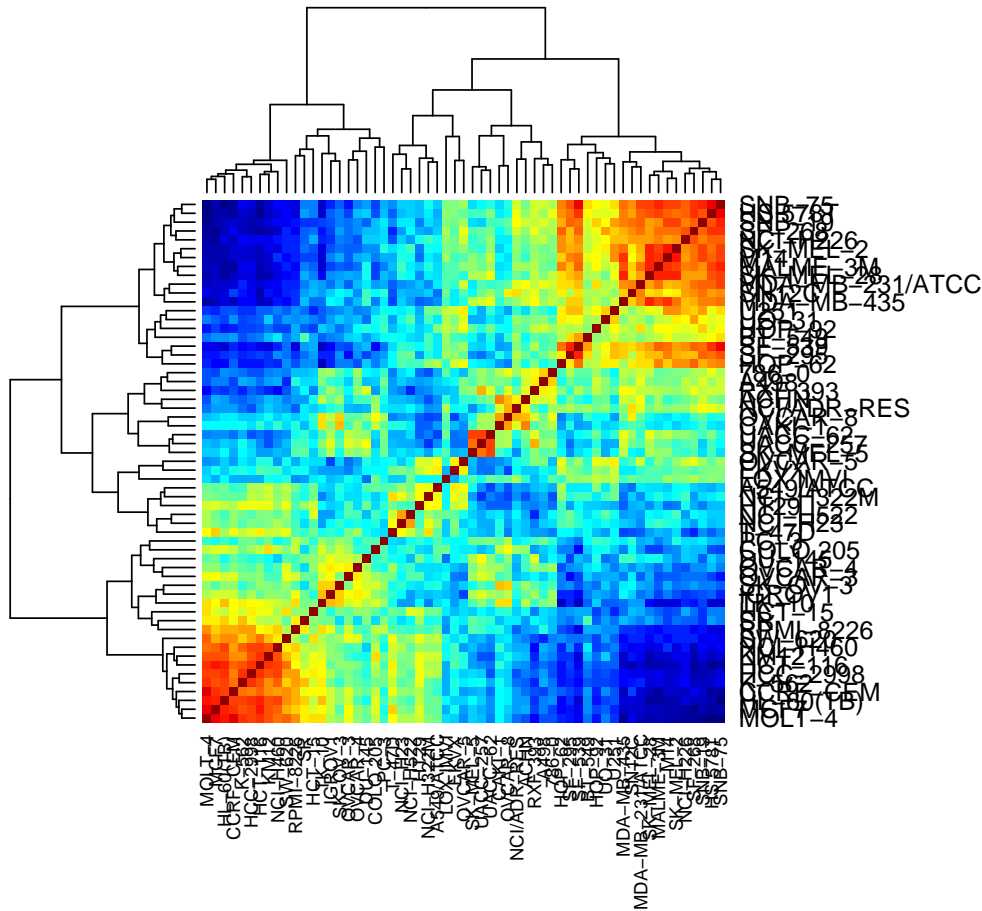
There is structure here. The reported gene list is offset by one.

### 4.3 Identifying the Samples Involved

There are more samples to examine for Pemetrexed than there were for Cisplatin (59 vs 30). Thus, we take a few steps to winnow down the number of samples examined.

Again, we begin by looking at correlations.

```
> pemetrexedCor <- cor(t(scale(t(temp1))))
> heatmap(pemetrexedCor, scale = "none", col = jet.colors(64),
+   margins = c(10, 10), cexRow = 1.1)
```



This plot suggests some obvious lines to try first; the topmost and bottommost clusters. The decision of which cluster to call “sensitive” and which “resistant” is based on the number of strongly expressed genes in each of the groups, and how these numbers compare with the results shown in Figure 1B from Hsu et al. Given the lists of names, we reorder the names to match the Potti et al. arrangement defined above.

```
> pemetrexedResistantLines <- c("SW-620", "NCI-H460", "KM12", "HCT-116",
+   "HCC-2998", "K-562", "CCRF-CEM", "HL-60(TB)", "MCF7", "MOLT-4")
> pemetrexedSensitiveLines <- c("SNB-75", "HS 578T", "SNB-19",
+   "SF-268", "NCI-H226", "SK-MEL-2", "M14", "MALME-3M", "SK-MEL-28",
+   "MDA-MB-231/ATCC", "SN12C", "MDA-MB-435")
> pemetrexedResistantLines <- colnames(novA)[!is.na(match(colnames(novA),
+   pemetrexedResistantLines))]
> pemetrexedSensitiveLines <- colnames(novA)[!is.na(match(colnames(novA),
+   pemetrexedSensitiveLines))]
> pemetrexedResistantLines

[1] "CCRF-CEM" "K-562" "MOLT-4" "HL-60(TB)" "MCF7" "HCC-2998"
```



```

[7] "HCT-116"    "KM12"       "SW-620"     "NCI-H460"
> pemetrexedSensitiveLines
[1] "SF-268"      "SNB-19"     "SNB-75"     "HS 578T"
[5] "MDA-MB-231/ATCC" "MDA-MB-435" "NCI-H226"   "M14"
[9] "MALME-3M"    "SK-MEL-2"   "SK-MEL-28"  "SN12C"

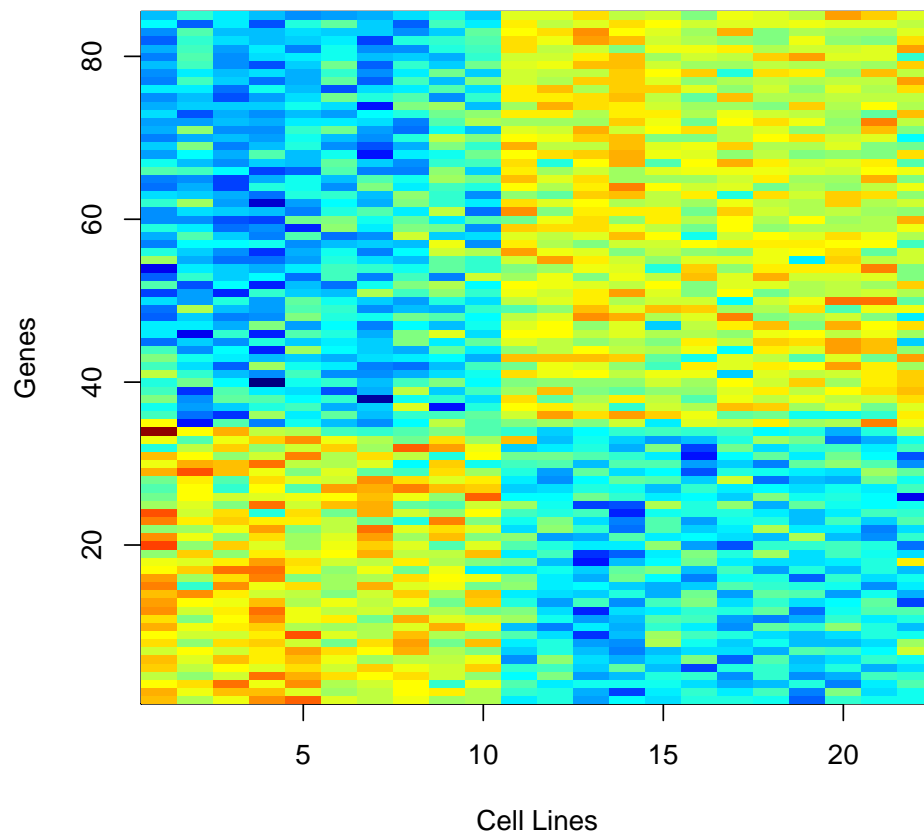
```

At this point, we have 10 resistant and 12 sensitive lines; we want 8 and 10, respectively. To prune these lists, we look at several heatmaps, beginning with one assembled using the 22 lines currently under consideration.

```

> temp2 <- temp1[, c(pemetrexedResistantLines, pemetrexedSensitiveLines)]
> temp2svd <- svd(t(scale(t(temp2))))
> image(1:dim(temp2)[2], 1:dim(temp2)[1], scale(t(temp2))[, order(temp2svd$u[,
+   1])], col = jet.colors(64), xlab = "Cell Lines", ylab = "Genes")

```

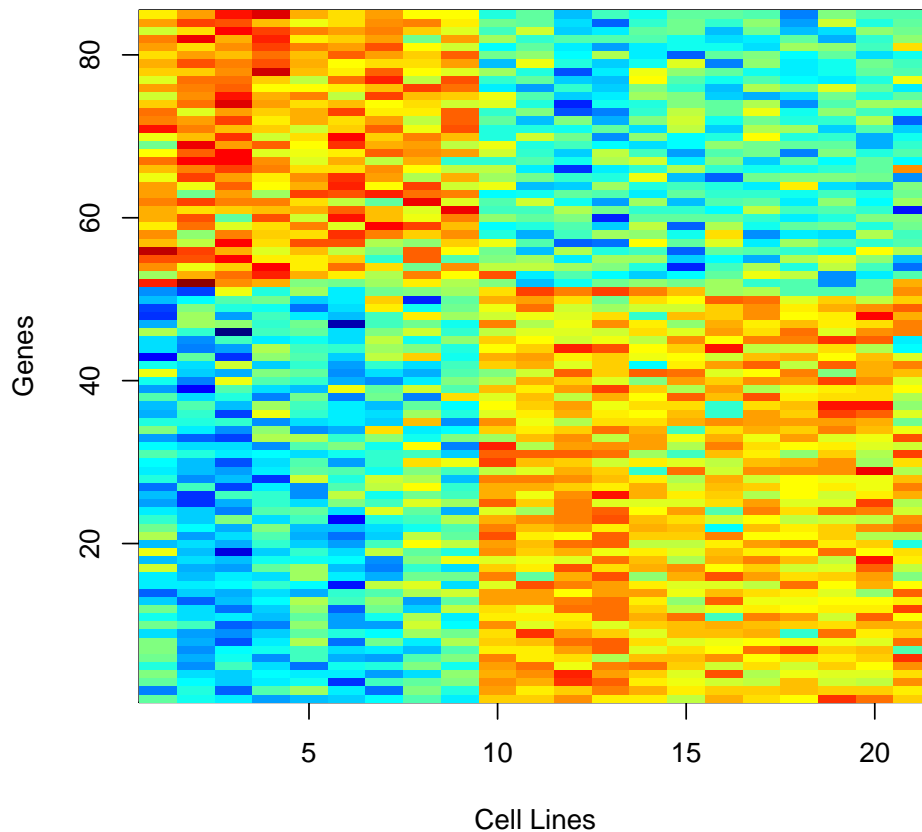


Here, the first resistant line, CCRF-CEM, has a very high outlier, distorting the dynamic range. We drop that line, and assemble a new heatmap.

```

> temp2 <- temp1[, c(pemetrexedResistantLines[2:10], pemetrexedSensitiveLines)]
> temp2svd <- svd(t(scale(t(temp2))))
> image(1:dim(temp2)[2], 1:dim(temp2)[1], scale(t(temp2))[, order(temp2svd$u[,
+   1])]), col = jet.colors(64), xlab = "Cell Lines", ylab = "Genes")

```

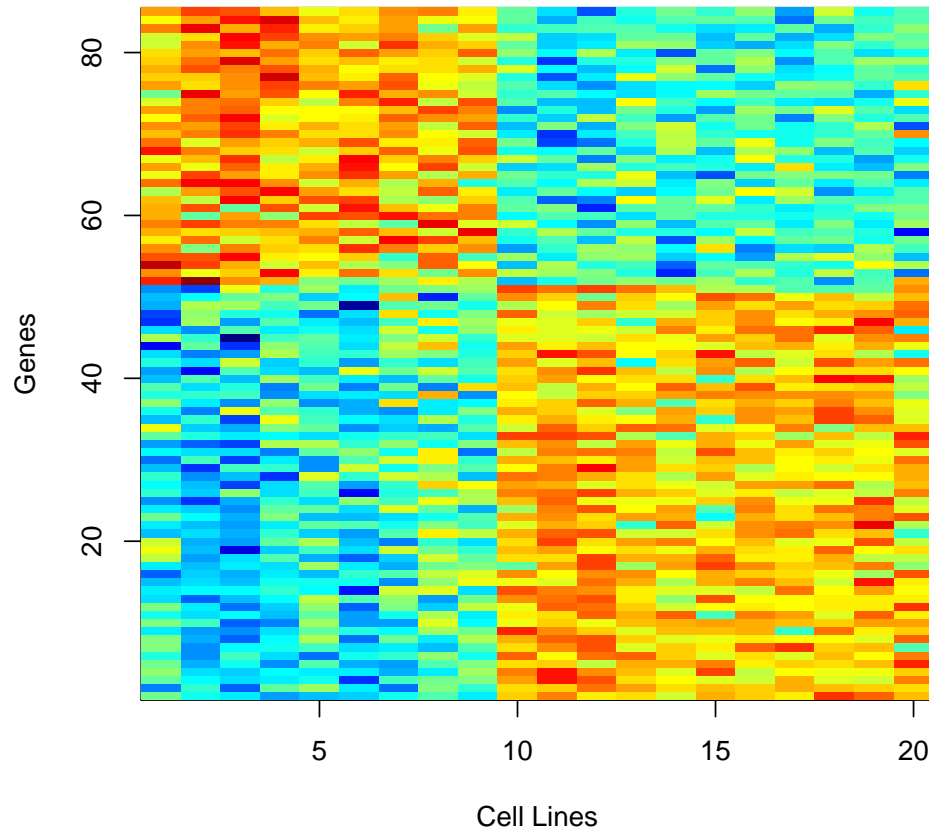


The first sensitive line, SF-268, has a high (red) reading in the block of genes that are high in the resistant samples. This feature is not present in the first few sensitive lines in Hsu et al.'s Figure 1B. We drop this line, and assemble a new heatmap.

```

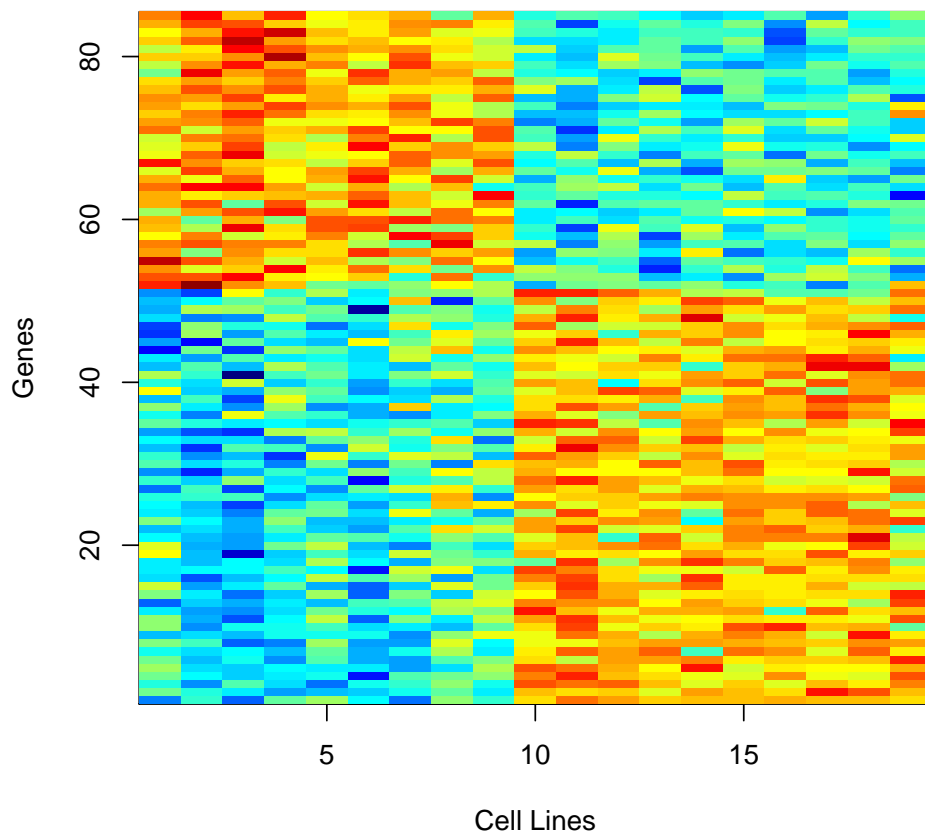
> temp2 <- temp1[, c(pemetrexedResistantLines[2:10], pemetrexedSensitiveLines[2:12])]
> temp2svd <- svd(t(scale(t(temp2))))
> image(1:dim(temp2)[2], 1:dim(temp2)[1], scale(t(temp2))[, order(temp2svd$u[,
+   1])]), col = jet.colors(64), xlab = "Cell Lines", ylab = "Genes")

```



Looking at the remaining sensitive lines, the second and third shown have lots of blue at top, as in the second sensitive line shown in the Hsu et al.'s Figure 1B. Of the two shown here, the second line, SNB-75, seems a worse fit than the third, HS 578T, as there are "too many greens" in the region where its expression is high. So, we drop SNB-75 and assemble another heatmap.

```
> temp2 <- temp1[, c(pemetrexedResistantLines[2:10], pemetrexedSensitiveLines[c(2,
+ 4:12)])]
> temp2svd <- svd(t(scale(t(temp2))))
> image(1:dim(temp2)[2], 1:dim(temp2)[1], scale(t(temp2))[, order(temp2svd$u[,
+ 1])], col = jet.colors(64), xlab = "Cell Lines", ylab = "Genes")
```



At this point, the sensitive lines look plausible, and the resistant lines aren't bad. However, none of the resistant lines seem to match the last one shown in Hsu et al.'s Figure 1B, though that last line is still positively correlated with the with the other resistant lines. Looking at the cluster closest to the "resistant cluster" on the correlation heatmap, there are 11 samples that we would view as candidates for the missing line.

```
> pemetrexedMiddleLines <- c("PC-3", "COLO 205", "DU-145", "OVCAR-4",
+   "OVCAR-3", "SK-OV-3", "IGROV1", "TK-10", "HCT-15", "SR",
+   "RPMI-8226")
> pemetrexedMiddleLines <- colnames(novA)[!is.na(match(colnames(novA),
+   pemetrexedMiddleLines))]
> pemetrexedMiddleLines

 [1] "RPMI-8226" "SR"          "COLO 205"  "HCT-15"   "DU-145"   "PC-3"
 [7] "IGROV1"    "OVCAR-3"   "OVCAR-4"  "SK-OV-3"  "TK-10"
```

If we allow for (a) dropping 2 of the 9 resistant samples and (b) including 1 of the 11 potential ones, that gives  $36 \cdot 11 = 396$  possibilities, which is small enough to try by brute force.

We export the `novA` table for analysis using the Duke software, splitting apart components corresponding to the numbers, probeset ids, and sample IDs. (We didn't include this explicit splitting with the Gyorffy quantifications as we were already working with an extant table.)

```
> write.table(novA, file = file.path("MatlabFiles", "nci60_numbers.csv"),
+   sep = ",", row.names = FALSE, col.names = FALSE)
> write.table(rownames(novA), file = file.path("MatlabFiles", "nci60_probesets.csv"),
+   sep = ",", row.names = FALSE, col.names = FALSE, quote = FALSE)
> write.table(colnames(novA), file = file.path("MatlabFiles", "nci60_samples.csv"),
+   sep = ",", row.names = FALSE, col.names = FALSE, quote = FALSE)
```

The indices for the resistant, middle, and sensitive groups, used in the Matlab file, are easily extracted.

```
> match(pemetrexedResistantLines, colnames(novA))
```

```
[1] 1 2 3 4 15 21 22 25 26 34
```

```
> match(pemetrexedMiddleLines, colnames(novA))
```

```
[1] 5 6 20 23 44 45 46 47 48 51 58
```

```
> match(pemetrexedSensitiveLines, colnames(novA))
```

```
[1] 7 10 11 14 16 18 31 37 38 39 40 57
```

The script involved, `PemetrexedNCI60.m`, is in `MatlabFiles`. Here, we do invoke the normalization option for the Duke software, as the values are not initially on the  $\log_2$  scale.

After examining the results, we believe the resistant and sensitive lines used to produce Hsu et al.'s Fig 1B are

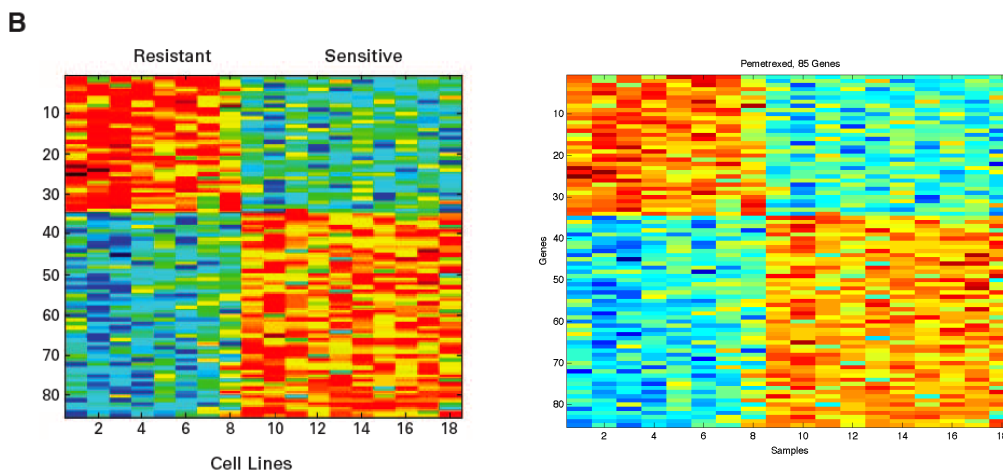
- **Resistant:** K-562, MOLT-4, HL-60(TB), MCF7, HCC-2998, HCT-116, NCI-H460, and TK-10,
- **Sensitive:** SNB-19, HS 578T, MDA-MB-231/ATCC, MDA-MB-435, NCI-H226, M14, MALME-3M, SK-MEL-2, SK-MEL-28, and SN12C,

in that order.

```
> pemetrexedResistantLines <- c("K-562", "MOLT-4", "HL-60(TB)",
+   "MCF7", "HCC-2998", "HCT-116", "NCI-H460", "TK-10")
> pemetrexedSensitiveLines <- c("SNB-19", "HS 578T", "MDA-MB-231/ATCC",
+   "MDA-MB-435", "NCI-H226", "M14", "MALME-3M", "SK-MEL-2",
+   "SK-MEL-28", "SN12C")
```

#### 4.4 Duke Software Output

Figure 1B from Hsu et al. is juxtaposed with the our pemetrexed heatmap below.



As with cisplatin, this is an exact match modulo shading, so we are working with the data used to produce the initial figure.

The Duke software also lists the probesets used in the heatmap. Here, these are stored in MatlabFiles in both PemetrexedTrainingImageGeneList.txt and PemetrexedTrainingTopGeneCoeffs.txt. We grab the gene list from the latter and check it against the list reported by Hsu et al.

```
> temp <- read.table(file.path("MatlabFiles", "PemetrexedTrainingTopGeneCoeffs.txt"))
> temp[1:3, ]
```

```
      V1      V2
1 0.630099 40953_at
2 0.774173 36119_at
3 1.097585 41739_s_at
```

```
> softwarePemetrexedGenes <- as.character(temp[, 2])
> softwarePemetrexedGenes <- sort(softwarePemetrexedGenes)
> softwarePemetrexedGenes[1:4]
```

```
[1] "1101_at" "1228_s_at" "1319_at" "1356_at"
```

```
> sum(!is.na(match(offsetPemetrexedGenes, softwarePemetrexedGenes)))
```

```
[1] 85
```

Checking the gene list produced by the Duke software, we match all 85 of the reported genes after accounting for the off-by-one error. There are no outliers here.

## 4.5 Sensitive and Resistant

Checking the sensitivity of the various cell lines used with respect to pemetrexed requires the drug assay data from the NCI, and an NSC number for pemetrexed to serve as a key.

We found the NSC number by first going to the NCI Drug Dictionary <http://www.cancer.gov/drugdictionary>, which in turn led us to the Drug Thesaurus entry at [http://nciterns.nci.nih.gov/NCIBrowser/ConceptReport.jsp?dictionary=NCI\\_Thesaurus&code=C1533](http://nciterns.nci.nih.gov/NCIBrowser/ConceptReport.jsp?dictionary=NCI_Thesaurus&code=C1533). The NSC number for pemetrexed is 698037.

We then obtained the 50% growth inhibition (GI50), total growth inhibition (TGI), and 50% lethal concentration (LC50) summary tables for pemetrexed from the Developmental Therapeutics Program query page, at [http://dtp.nci.nih.gov/docs/dtp\\_search.html](http://dtp.nci.nih.gov/docs/dtp_search.html). These values come from the July 2007 release.

```
> pemetrexedGI50 <- read.table(file.path("OtherData", "NCI", "nsc698037_pemetrexed_gi50.csv"),
+   sep = ",", header = TRUE)
> pemetrexedTGI <- read.table(file.path("OtherData", "NCI", "nsc698037_pemetrexed_tgi.csv"),
+   sep = ",", header = TRUE)
> pemetrexedLC50 <- read.table(file.path("OtherData", "NCI", "nsc698037_pemetrexed_lc50.csv"),
+   sep = ",", header = TRUE)
> dim(pemetrexedGI50)
```

```
[1] 58 10
```

```
> dim(pemetrexedTGI)
```

```
[1] 60 10
```

```
> dim(pemetrexedLC50)
```

```
[1] 60 10
```

```
> pemetrexedLC50[1:3, ]
```

	NSC	ConcentrationUnit	LogHighConc	CellPanelName	CellLineName	CellPanelNbr
1	698037	log10(M)	-4	Leukemia	CCRF-CEM	7
2	698037	log10(M)	-4	Leukemia	HL-60(TB)	7
3	698037	log10(M)	-4	Leukemia	K-562	7
	CellLineNbr	logValue	NumTestsNSC	CellLine	NumofExp	
1	3	-4		1	2	
2	8	-4		1	2	
3	5	-4		2	2	

Note that GI50 values are missing for 2 of the NCI60 cell lines.

We now do some quick summary checks, starting with LC50.

```
> summary(pemetrexedLC50)
```

	NSC	ConcentrationUnit	LogHighConc	CellPanelName
Min.	:698037	log10(M):60	Min. :-4	Non-Small Cell Lung : 9
1st Qu.:	698037		1st Qu. :-4	Breast : 8
Median :	698037		Median :-4	Melanoma : 8
Mean :	698037		Mean :-4	Renal : 8
3rd Qu.:	698037		3rd Qu. :-4	Colon : 7
Max.	:698037		Max. :-4	Central Nervous System: 6 (Other) :14
	CellLineName	CellPanelNbr	CellLineNbr	logValue
786-0	: 1	Min. : 1.000	Min. : 1.000	Min. :-4
A498	: 1	1st Qu.: 4.000	1st Qu.: 3.000	1st Qu. :-4
A549/ATCC:	1	Median : 6.500	Median : 8.500	Median :-4
ACHN	: 1	Mean : 6.683	Mean : 9.833	Mean :-4

```

BT-549 : 1 3rd Qu.:10.000 3rd Qu.:15.000 3rd Qu.: -4
CAKI-1 : 1 Max. :12.000 Max. :29.000 Max. : -4
(Other) :54
NumTestsNSCCellLine NumofExp
Min. :1.000 Min. :2
1st Qu.:1.000 1st Qu.:2
Median :2.000 Median :2
Mean :1.633 Mean :2
3rd Qu.:2.000 3rd Qu.:2
Max. :2.000 Max. :2

```

Pemetrexed hasn't been assayed that often. There have only been two assays with each cell line, and for about a third of those, only one that yielded usable test data. Further, the reported mean LC50 values are useless for discriminating between cell lines; all of the "logValue" values are -4. Since -4 was the maximum log concentration tested, pemetrexed failed to kill off 50% or more of the cells in any of the cell line/concentration combinations tested.

Next, we look at TGI.

```
> summary(pemetrexedTGI)
```

```

      NSC      ConcentrationUnit LogHighConc      CellPanelName
Min. :698037 log10(M):60      Min. :-4 Non-Small Cell Lung : 9
1st Qu.:698037      1st Qu.: -4 Breast : 8
Median :698037      Median :-4 Melanoma : 8
Mean :698037      Mean :-4 Renal : 8
3rd Qu.:698037      3rd Qu.: -4 Colon : 7
Max. :698037      Max. :-4 Central Nervous System: 6
      (Other) :14

```

```

      CellLineName CellPanelNbr      CellLineNbr      logValue
786-0 : 1 Min. : 1.000 Min. : 1.000 Min. :-7.015
A498 : 1 1st Qu.: 4.000 1st Qu.: 3.000 1st Qu.: -4.000
A549/ATCC: 1 Median : 6.500 Median : 8.500 Median :-4.000
ACHN : 1 Mean : 6.683 Mean : 9.833 Mean :-4.050
BT-549 : 1 3rd Qu.:10.000 3rd Qu.:15.000 3rd Qu.: -4.000
CAKI-1 : 1 Max. :12.000 Max. :29.000 Max. :-4.000
(Other) :54

```

```

NumTestsNSCCellLine NumofExp
Min. :1.000 Min. :2
1st Qu.:1.000 1st Qu.:2
Median :2.000 Median :2
Mean :1.633 Mean :2
3rd Qu.:2.000 3rd Qu.:2
Max. :2.000 Max. :2

```

```
> sort(pemetrexedTGI$logValue)[1:4]
```

```
[1] -7.015 -4.000 -4.000 -4.000
```

```
> pemetrexedTGI[which(pemetrexedTGI$logValue < -4), ]
```



```

      NSC ConcentrationUnit LogHighConc      CellPanelName CellLineName
25 698037      log10(M)          -4 Central Nervous System      SF-539
      CellPanelNbr CellLineNbr logValue NumTestsNSCCellLine NumofExp
25      12          16      -7.015          1          2

```

The TGI values are almost useless for differentiating the cell lines; only one cell line (SF-539) has a logValue other than -4. Given that a log concentration of -7.015 is sufficient to achieve TGI for this line, SF-539 would seem to be unusually sensitive.

Finally, we look at GI50.

```
> summary(pemetrexedGI50)
```

```

      NSC      ConcentrationUnit LogHighConc      CellPanelName
Min.   :698037  log10(M):58      Min.   :-4  Non-Small Cell Lung   : 9
1st Qu.:698037      1st Qu.:-4  Breast      : 8
Median :698037      Median :-4  Melanoma    : 8
Mean   :698037      Mean   :-4  Colon       : 7
3rd Qu.:698037      3rd Qu.:-4  Renal       : 7
Max.   :698037      Max.   :-4  Central Nervous System: 6
      (Other)      :13

      CellLineName CellPanelNbr      CellLineNbr      logValue
786-0   : 1      Min.   : 1.000      Min.   : 1.00      Min.   :-8.000
A498    : 1      1st Qu.: 4.000      1st Qu.: 3.25     1st Qu.:-6.248
A549/ATCC: 1      Median : 6.000      Median : 9.00     Median :-4.000
ACHN    : 1      Mean   : 6.638      Mean   :10.05     Mean   :-4.967
BT-549  : 1      3rd Qu.:10.000     3rd Qu.:15.00     3rd Qu.:-4.000
CAKI-1  : 1      Max.   :12.000     Max.   :29.00     Max.   :-4.000
(Other) :52

NumTestsNSCCellLine      NumofExp
Min.   :1.000      Min.   :2
1st Qu.:1.000      1st Qu.:2
Median :2.000      Median :2
Mean   :1.655      Mean   :2
3rd Qu.:2.000      3rd Qu.:2
Max.   :2.000      Max.   :2

```

```
> sum(pemetrexedGI50$logValue < -4)
```

```
[1] 20
```

```
> pemetrexedGI50[pemetrexedGI50$CellLineName == "SF-539", ]
```

```

      NSC ConcentrationUnit LogHighConc      CellPanelName CellLineName
24 698037      log10(M)          -4 Central Nervous System      SF-539
      CellPanelNbr CellLineNbr logValue NumTestsNSCCellLine NumofExp
24      12          16          -8          1          2

```

Here, we have more information; fully a third of the samples have a GI50 value less than -4. This does point out a potential difficulty, though — if 40 of the cell lines assayed failed to hit GI50 at the maximum dose, how are we to choose a subset of these 40 to treat as the “most” resistant? At the other extreme, we see

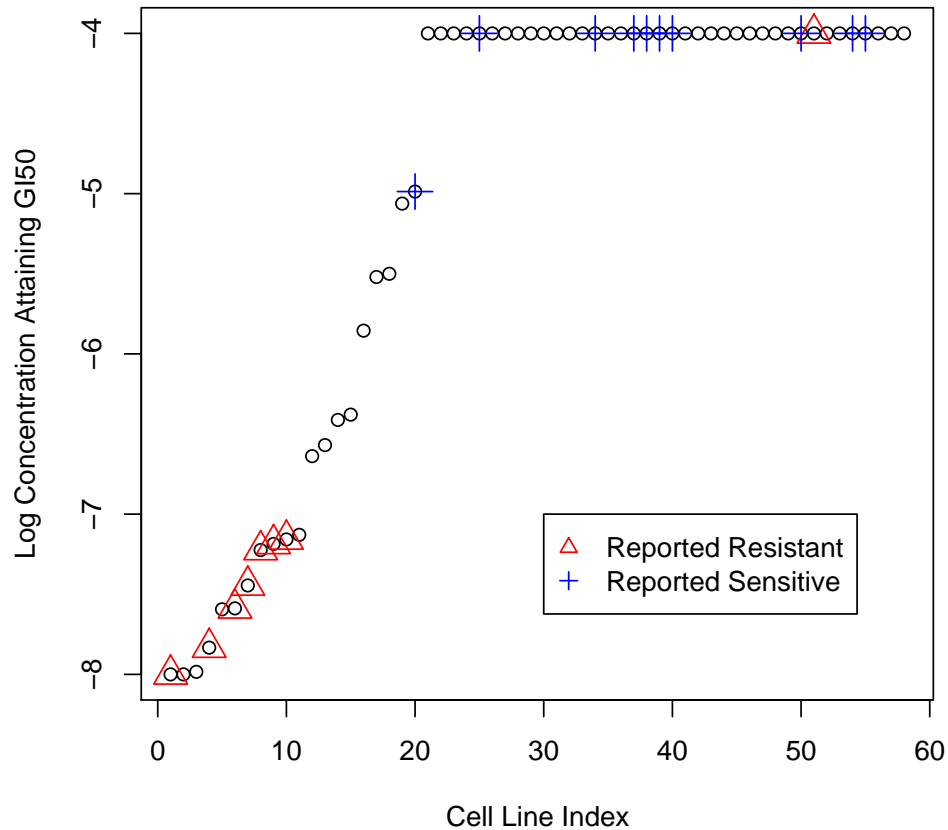
that SF-539, the one sample sensitive enough to show TGI at a non-boundary dose, is sensitive here as well. Indeed, as -8 is the weakest concentration tested (the assays typically involve 5 unit steps on a log10 scale), GI50 is attained at the weakest concentration tested, so SF-539 is again one of the most sensitive lines.

Given the lack of differentiation in the LC50 and TGI scores, We will only work with the GI50 data for the rest of our analysis.

Now, we plot the sorted GI50 values, marking the resistant and sensitive lines inferred from the pemetrexed heatmap.

```
> pemetrexedGI50Sorted <- pemetrexedGI50[order(pemetrexedGI50$logValue),
+ ]
> resistantIndices <- match(pemetrexedResistantLines, as.character(pemetrexedGI50Sorted$CellLineName))
> sensitiveIndices <- match(pemetrexedSensitiveLines, as.character(pemetrexedGI50Sorted$CellLineName))
> plot(pemetrexedGI50Sorted$logValue, xlab = "Cell Line Index",
+      ylab = "Log Concentration Attaining GI50", main = "Sorted Pemetrexed GI50s, More Negative = More S",
+      points(resistantIndices, pemetrexedGI50Sorted$logValue[resistantIndices],
+            col = "red", pch = 2, cex = 2)
+      points(sensitiveIndices, pemetrexedGI50Sorted$logValue[sensitiveIndices],
+            col = "blue", pch = 3, cex = 2)
> legend(30, -7, legend = c("Reported Resistant", "Reported Sensitive"),
+      col = c("red", "blue"), pch = c(2, 3), cex = 1)
```

### Sorted Pemetrexed GI50s, More Negative = More Sensitive



This figure is rather disturbing. The biggest problem is that most of the cell lines labeled resistant in Figure 1B of Hsu et al. are actually *sensitive*, and most reported as sensitive are in fact resistant. The figure's labeling is the reverse of what it should be.

Even allowing for the reversal, the GI50 values for the sensitive and resistant lines overlap, pointing to an inconsistent rule. Finally, and again accounting for the reversal, even with almost 40 cell lines showing maximal resistance values of -4, one of the 10 lines chosen for the group of lines at this extreme does not attain -4.

Let's add the identities of some of the interesting points.

```
> <<plotPemetrexedGI50s>>
> pemetrexedResistantLines[which(pemetrexedGI50Sorted$logValue[resistantIndices] == -4)]
[1] "TK-10"
> arrows(51,-5, 51, -4.1)
> text(51,-5, labels="TK-10", pos=1)
> pemetrexedSensitiveLines[which(pemetrexedGI50Sorted$logValue[sensitiveIndices] < -4)]
```

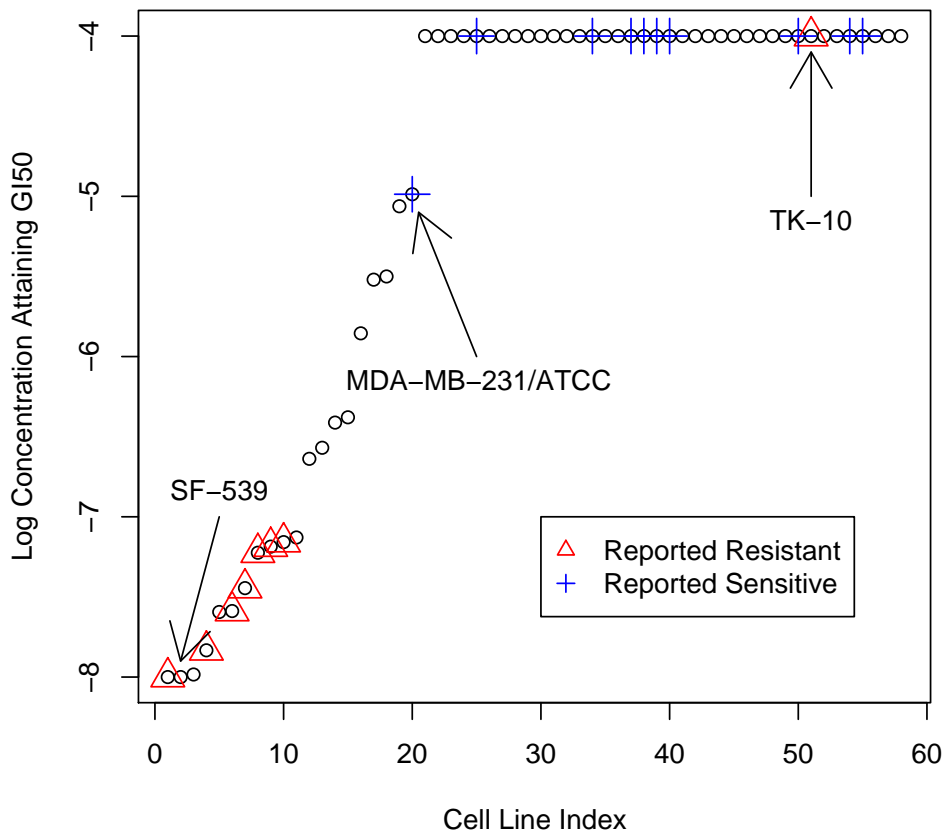
```
[1] "MDA-MB-231/ATCC"

> arrows(25,-6, 20.5, -5.1)
> text(25,-6, labels="MDA-MB-231/ATCC", adj=1, pos=1)
> which(pemetrexedGI50Sorted$CellLineName == "SF-539")

[1] 2

> arrows(5,-7, 2, -7.9)
> text(5,-7, labels="SF-539", pos=3)
```

### Sorted Pemetrexed GI50s, More Negative = More Sensitive



The renal line TK-10 is the “outlier” of the samples reported to be resistant (it’s the only one that actually is). The breast line MDA-MB-231/ATCC is the outlier of the samples reported to be sensitive, as its GI50 is actually attained in the range of concentrations tested. Finally, we note that SF-539, the only sample with a non-saturated TGI value, is not present in either of the lists, though we would have characterized it as the most sensitive line present.

## 5 Summary

Most of the problems noted in the Coombes et al. analysis of the Potti et al. results are also present here. Specifically,

1. We cannot reproduce their selection of cell lines. The sets of “sensitive” and “resistant” GI50 values for pemetrexed overlap.
2. The lists of genes reported are wrong, due to an off-by-one indexing error.
3. Using their software, we can perfectly reproduce the published heatmaps for both cisplatin and pemetrexed. However, after correcting for the off-by-one error, we can only match the gene list exactly for pemetrexed. The list for cisplatin contains outliers.
4. For cisplatin, their software produces only 41 of the 45 reported probesets. The remaining 4 correspond to ERCC1 (U133A), ERCC4 (U133A), ERCC1 (U133B), and FANCM (U133B). These probesets cannot be obtained from the training data. Indeed, *the last two probesets are not present on the arrays used in the training set*. The only genes named in the paper are ERCC1 and ERCC4. The paper also notes enrichment for DNA repair genes. Their reported list contains 5 genes with this function, 4 of which are the outliers mentioned above.
5. In the case of pemetrexed, the sensitive/resistant labeling has been reversed. Using the model with this labeling suggests administering the drug just to the patients it would not benefit.

Modulo wording, points 1-4 are qualitatively identical to points 1-4 of Coombes et al. Point 5 here is roughly analogous to point 6 in Coombes et al., where label reversal appeared to affect the results for adriamycin.

We have not checked the results of using the models for prediction, as we do not know precisely which samples were used. For ovarian cancer, GSE3149 (<http://www.ncbi.nlm.nih.gov/projects/geo/query/acc.cgi?acc=GSE3149>) contains quantifications for 153 arrays, but the paper reports using 59. For lung cancer, GSE3141 (<http://www.ncbi.nlm.nih.gov/projects/geo/query/acc.cgi?acc=GSE3141>) contains quantifications for 111 arrays, but the paper reports using 91.

Given that we think the results of Potti et al. are wrong, and the fact that the same problems are present with the data from this paper, we are not persuaded by the results reported by Hsu et al.

## 6 Appendix

### 6.1 Saves

```
> save.image(file = file.path("RDataObjects", "wholeAnalysis.Rda"))
```

### 6.2 SessionInfo

```
> sessionInfo()
```

```
R version 2.5.1 (2007-06-27)
```

```
i386-apple-darwin8.9.1
```

```
locale:
```

```
en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
```

attached base packages:

```
[1] "splines"  "tools"    "stats"    "graphics" "grDevices" "utils"
[7] "datasets" "methods"  "base"
```

other attached packages:

```
ClassDiscovery      cluster ClassComparison      PreProcess      oompaBase
      "2.5.0"        "1.11.7"        "2.5.0"        "2.5.0"        "2.5.0"
      affy          affyio          Biobase
      "1.14.2"      "1.4.1"        "1.14.0"
```

## References

- [1] Coombes KR, Wang J, Baggerly KA: Genomic signatures based on the NCI60 cell lines do not predict patient response to chemotherapy. *Nat Med*, to appear.
- [2] Györfy 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**:1699-1712, 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