Understanding Adria

Keith A. Baggerly and Kevin R. Coombes

November 13, 2007

1 Introduction

Here, we're trying to better understand the structure in the processed Adriamycin data. Only exploratory tests (checking correlations) are involved.

2 Options and Libraries

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

3 Loading The Duke Data

```
> dukeHeader1 <- read.table(file.path("DukeWebSite", "Adria_ALL(n = 122).txt"),
+     sep = "\t", nrows = 1, header = FALSE)
> dukeHeader1 <- as.vector(t(dukeHeader1))
> dukeHeader2 <- read.table(file.path("DukeWebSite", "Adria_ALL(n = 122).txt"),
+     sep = "\t", skip = 1, nrows = 1, header = FALSE)
> dukeHeader2 <- as.vector(t(dukeHeader2))
> dukeAdria <- read.table(file.path("DukeWebSite", "Adria_ALL(n = 122).txt"),
+     sep = "\t", skip = 2, header = FALSE)
> table(dukeHeader1)

dukeHeader1
          0           1           2      Adria0      Adria1 Validation2
          9          11         120           1           1           2

> table(dukeHeader2)

dukeHeader2
       NR Resistant      Resp      Sens
       99        10        23        12

> dim(dukeAdria)

[1] 8958  144

> dukeAdria[1:3, 1:10]
```

```
    V1   V2   V3   V4   V5   V6   V7   V8   V9  V10
1 1.18 1.12 3.46 0.65 3.07 1.57 0.13 1.05 2.38 1.53
2 1.75 4.02 0.43 0.31 0.76 0.37 0.21 0.69 0.15 1.65
3 0.13 0.35 1.13 1.14 0.84 0.27 0.63 0.89 2.40 2.33
```

4 Checking Correlations

Having loaded the data, let's compute correlations.

```
> corAdriaSelf <- cor(dukeAdria)
> sum(corAdriaSelf > 0.9999)
```

```
[1] 256
```

```
> sum(diag(corAdriaSelf) > 0.9999)
```

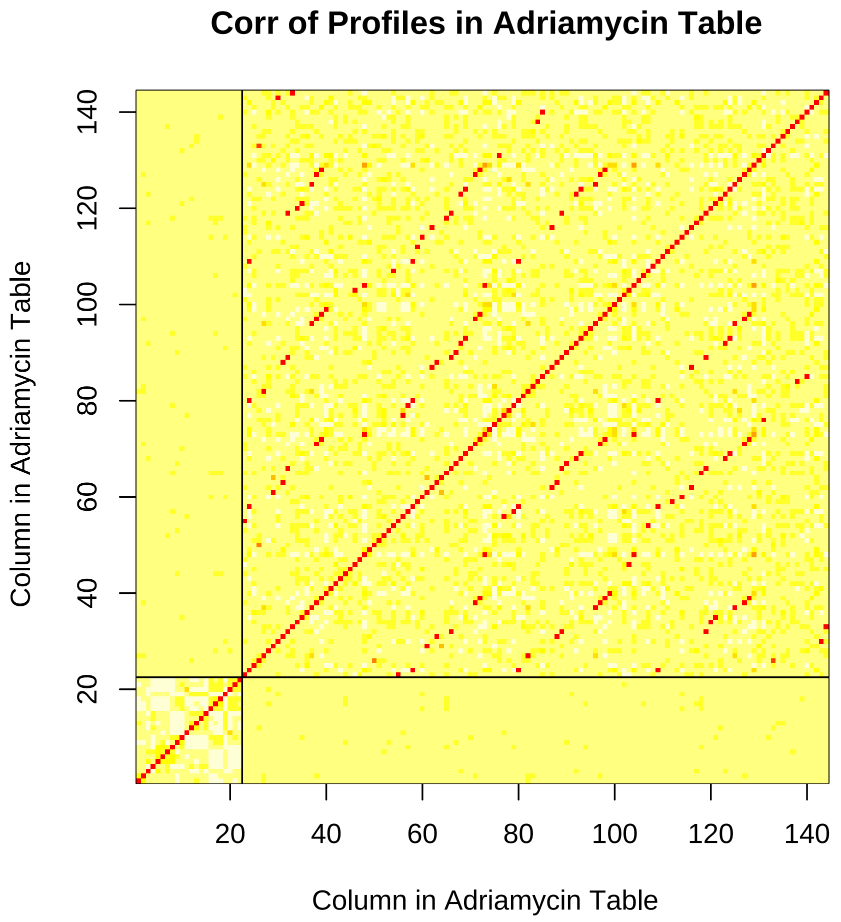
```
[1] 144
```

Some columns appear to be redundant.

5 Checking Adriamycin

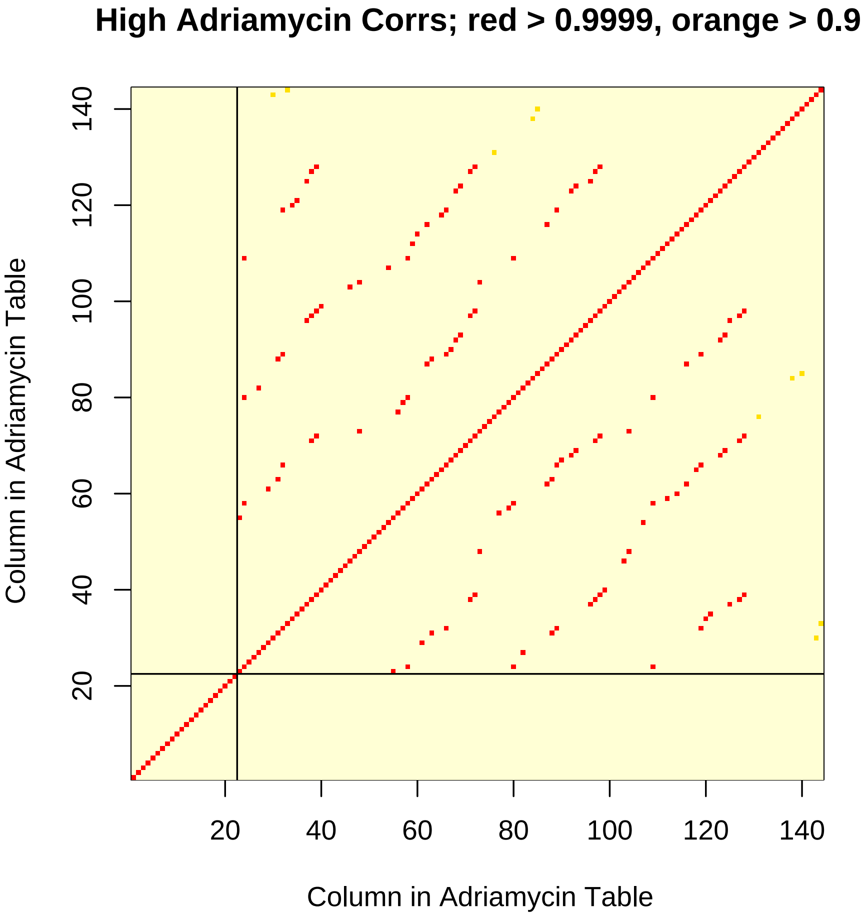
There's a problem with the adriamycin data, in that the correlations suggest that not all of the samples are distinct. This may be more obvious if we check the data graphically.

```
> oldPin <- par()$pin
> par(pin = c(min(oldPin), min(oldPin)))
> image(1:144, 1:144, 1 - corAdriaSelf, xlab = "Column in Adriamycin Table",
+     ylab = "Column in Adriamycin Table", main = "Corr of Profiles in Adriamycin Table")
> abline(h = 22.5, v = 22.5)
> par(pin = oldPin)
```



A very regular structure appears in the correlation heatmap, almost band-diagonal in nature. Let's cast this in starker relief by focusing on the really high values.

```
> oldPin <- par()$pin
> par(pin = c(min(oldPin), min(oldPin)))
> image(1:144, 1:144, (corAdriaSelf < 0.9) + 2 * (corAdriaSelf <
+     0.9999), xlab = "Column in Adriamycin Table", ylab = "Column in Adriamycin Table",
+     main = "High Adriamycin Corrs; red > 0.9999, orange > 0.9")
> abline(h = 22.5, v = 22.5)
> par(pin = oldPin)
```



The problem is clearly confined to the test data, so we will ignore the training data for the rest of this report. The last 16 samples appear not to have any exact ties, but there are a small number of pairs with correlations that are nonetheless pretty high. Let's count how many independent samples we're actually dealing with.

```
> nCopies <- apply(corAdriaSelf[23:144, 23:144], 1, function(x) {
+     sum(x > 0.99)
+ })
> table(nCopies)

nCopies
 1  2  3  4
60 28 18 16

> sum(table(nCopies)/(1:4))

[1] 84
```

All told, we're dealing with 84 distinct samples, not 122. That's a pretty big drop. How are these split between the responders and nonresponders? Are there cases where the same sample appears multiple times with different classifications?

```
> adriaStatus <- dukeHeader2
> adriaStatus <- unlist(adriaStatus)
> table(adriaStatus)

adriaStatus
       NR Resistant      Resp      Sens
       99        10        23        12

> adriaTestStatus <- as.factor(as.character(adriaStatus[23:144]))
> table(adriaTestStatus, nCopies)

               nCopies
adriaTestStatus  1  2  3  4
           NR   42 25 17 15
           Resp 18  3  1  1
```

The numbers of Resp and NR columns do match the numbers reported in the Nature Medicine paper. The cross tabulation with status, however, shows a problem. There are 4 samples that are replicated 4 times each, accounting for 16 of the columns. Of these 16, only 15 are classed as NR. This means that one of these 4 samples is classified as NR 3 times, and Resp 1 time. This will make fitting a classifier somewhat difficult. Similar mismatches are apparent for the samples replicated 2 and 3 times each; we know that there are problems, but we don't know which samples these problems affect. Let's see if we can arrange things to make the structure more clear.

First the pairs.

```
> pairStatus <- matrix(adriaStatus[22 + which(nCopies == 2)][order(t(dukeAdria[1,
+     22 + which(nCopies == 2)]))], 14, 2, byrow = TRUE)
> pairNames <- matrix(colnames(dukeAdria)[22 + which(nCopies ==
+     2)][order(t(dukeAdria[1, 22 + which(nCopies == 2)]))], 14,
+     2, byrow = TRUE)
> pairFirstValues <- matrix(dukeAdria[1, 22 + which(nCopies ==
+     2)][order(t(dukeAdria[1, 22 + which(nCopies == 2)]))], 14,
+     2, byrow = TRUE)
> pairInfo <- data.frame(pairFirstValues, pairNames, pairStatus)
> pairInfo[order(pairInfo$X1.1), ]

     X1   X2 X1.1 X2.1 X1.2 X2.2
8   0.6  0.6  V23  V55   NR Resp
11 1.23 1.23  V27  V82   NR   NR
12 2.25 2.25  V29  V61 Resp   NR
5  0.37 0.37  V34 V120   NR   NR
1  0.19 0.19  V35 V121 Resp   NR
9  0.94 0.94  V40  V99   NR   NR
2   0.2  0.2  V46 V103   NR   NR
13 2.35 2.35  V54 V107   NR   NR
3   0.3  0.3  V56  V77   NR   NR
```

```
6   0.4  0.4  V57  V79   NR   NR
7  0.49 0.49  V59 V112   NR   NR
10  1.2  1.2  V60 V114   NR   NR
4  0.36 0.36  V65 V118   NR   NR
14  3.5  3.5  V67  V90   NR   NR
```

Then the triples.

```
> tripleStatus <- matrix(adriaStatus[22 + which(nCopies == 3)][order(t(dukeAdria[1,
+     22 + which(nCopies == 3)]))], 6, 3, byrow = TRUE)
> tripleNames <- matrix(colnames(dukeAdria)[22 + which(nCopies ==
+     3)][order(t(dukeAdria[1, 22 + which(nCopies == 3)]))], 6,
+     3, byrow = TRUE)
> tripleFirstValues <- matrix(dukeAdria[1, 22 + which(nCopies ==
+     3)][order(t(dukeAdria[1, 22 + which(nCopies == 3)]))], 6,
+     3, byrow = TRUE)
> tripleInfo <- data.frame(tripleFirstValues, tripleNames, tripleStatus)
> tripleInfo[order(tripleInfo$X1.1), ]
```

```
    X1   X2   X3 X1.1 X2.1 X3.1 X1.2 X2.2 X3.2
1 0.35 0.35 0.35  V31  V63  V88   NR   NR   NR
4 0.81 0.81 0.81  V37  V96 V125   NR Resp   NR
6  2.3  2.3  2.3  V48  V73 V104   NR   NR   NR
5 0.89 0.89 0.89  V62  V87 V116   NR   NR   NR
2 0.67 0.67 0.67  V68  V92 V123   NR   NR   NR
3 0.79 0.79 0.79  V69  V93 V124   NR   NR   NR
```

Then the quartets.

```
> quartetStatus <- matrix(adriaStatus[22 + which(nCopies == 4)][order(t(dukeAdria[1,
+     22 + which(nCopies == 4)]))], 4, 4, byrow = TRUE)
> quartetNames <- matrix(colnames(dukeAdria)[22 + which(nCopies ==
+     4)][order(t(dukeAdria[1, 22 + which(nCopies == 4)]))], 4,
+     4, byrow = TRUE)
> quartetFirstValues <- matrix(dukeAdria[1, 22 + which(nCopies ==
+     4)][order(t(dukeAdria[1, 22 + which(nCopies == 4)]))], 4,
+     4, byrow = TRUE)
> quartetInfo <- data.frame(quartetFirstValues, quartetNames, quartetStatus)
> quartetInfo[order(quartetInfo$X1.1), ]
```

```
    X1   X2   X3   X4 X1.1 X2.1 X3.1 X4.1 X1.2 X2.2 X3.2 X4.2
4 3.53 3.53 3.53 3.53  V24  V58  V80 V109   NR   NR   NR   NR
1 0.74 0.74 0.74 0.74  V32  V66  V89 V119 Resp   NR   NR   NR
3 2.87 2.87 2.87 2.87  V38  V71  V97 V127   NR   NR   NR   NR
2 1.73 1.73 1.73 1.73  V39  V72  V98 V128   NR   NR   NR   NR
```

The labeling with respect to status seems rather scrambled with respect to sample identity.

6 Conclusions

There are only 84 independent samples, not 122. Some replicates have different status assignments, so the same sample is listed as both responsive and nonresponsive. These ties can substantially distort the results.

The posted data is wrong.

7 Appendix

7.1 Saves

7.2 SessionInfo

```
> sessionInfo()

R version 2.5.1 (2007-06-27)
i386-pc-mingw32

locale:
LC_COLLATE=English_United States.1252;LC_CTYPE=English_United States.1252;LC_MONETARY=English_United Sta

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

other attached packages:
R.matlab     R.oo
 "1.1.3"  "1.3.0"
```