Checking The New Posted Gene Lists

Keith A. Baggerly and Kevin R. Coombes

20 August 2007

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

1 The New Gene Lists

After receiving our initial reports, Potti et al posted new gene lists to the Nature Medicine supplementary information site; these lists are found in the revised supplementary Table 1. We do not know precisely when this revision took place. The revised table notes that

> In the version of the supplementary information initially published online to accompany this article, a software error in the program that generates gene lists from expression signatures resulted in an incorrect listing of genes that make up the signatures for chemotherapy sensitivity (Supplementary Table 1). This error does not alter the results of the study with respect to the ability of the signatures to predict clinical response. The correct list of genes is contained in this file.

In this report, we check the agreement between these new reported gene lists and the lists we obtained from running the software ourselves.

2 Loading the Data

The supplementary info table is supplied as a single pdf file. Our first processing step involved lots of simple cutting and pasting to produce one text file for each of the drug lists, producing *six* files. The gene lists and required R Data objects are located in

```
> geneListPath <- file.path("..", "Original-info", "SuppInfoAug07")
> rDataPath <- file.path("..", "ReproducibilityMS", "Manuscript",
+     "RDataObjects")
```

The initially posted supplementary table 1 contained eight lists: one for each of the seven drugs discussed, and an additional one listing the genes they found to comprise the PI3K pathway signature. The revised table has no gene list for cytoxan (which we couldn't match before, in any event), and no gene list for the PI3K pathway signature.

Let's load the probeset lists for the other six drugs.

5-fluorouracil:

```
> temp5fu <- read.table(file.path(geneListPath, "5fu.csv"), sep = ",")
> temp5fu[1:3, ]
```

```
[1] 5- FU predictor
[2] Probe Set ID Gene Title Chromosomal Location
[3] 152_f_at histone 2, H4a /// histone H4/o /// similar to germinal histone H4 gene 1q21
51 Levels: 1). This error does not alter the results of the study with respect to the ability c
```

```
> temp5fu <- temp5fu[grep(pattern = "_at", temp5fu[, 1]), ]
> temp5fu <- as.character(temp5fu)
> temp5fu <- strsplit(temp5fu, split = "_at")
> temp5fu <- unlist(lapply(temp5fu, function(x) {
+     paste(x[1], "_at", sep = "")
+ }))
> length(temp5fu)
```

```
[1] 45
```

Etoposide:

```
> tempEtopo <- read.table(file.path(geneListPath, "etoposide.csv"),
+     sep = ",")
> tempEtopo[1:3, ]
```

```
[1] Etoposide Predictor
[2] Probe Set ID Gene Title Chromosomal Location
[3] 1015_s_at LIM domain kinase 1 7q11.23
52 Levels: 1015_s_at LIM domain kinase 1 7q11.23 ...
```

```
> tempEtopo <- tempEtopo[grep(pattern = "_at", tempEtopo[, 1]),
+     ]
> tempEtopo <- as.character(tempEtopo)
> tempEtopo <- strsplit(tempEtopo, split = "_at")
> tempEtopo <- unlist(lapply(tempEtopo, function(x) {
+     paste(x[1], "_at", sep = "")
+ }))
> length(tempEtopo)
```

```
[1] 50
```

Topotecan (note the addition of the "quote" argument to deal with a description involving the 3-prime UTR):

```
> tempTopo <- read.table(file.path(geneListPath, "topotecan.csv"),
+     sep = ",", quote = "\"")
> tempTopo[1:3, ]
```

```
[1] Topotecan Predictor
[2] Probe ID Gene Title Chromosomal Location
[3] 1005_at dual specificity phosphatase 1 5q34
152 Levels: 1005_at dual specificity phosphatase 1 5q34 ...
```

```
> tempTopo <- tempTopo[grep(pattern = "_at", tempTopo[, 1]), ]
> tempTopo <- as.character(tempTopo)
> tempTopo <- strsplit(tempTopo, split = "_at")
> tempTopo <- unlist(lapply(tempTopo, function(x) {
+     paste(x[1], "_at", sep = "")
+ }))
> length(tempTopo)
```

```
[1] 150
```

Adriamycin:

```
> tempAdria <- read.table(file.path(geneListPath, "adriamycin.csv"),
+     sep = ",")
> tempAdria[1:3, ]
```

```
[1] e 1 2
86 Levels: 1 ...
```

```
> tempAdria <- tempAdria[grep(pattern = "_at", tempAdria[, 1]),
+     ]
> tempAdria <- as.character(tempAdria)
> tempAdria <- strsplit(tempAdria, split = "_at")
> tempAdria <- unlist(lapply(tempAdria, function(x) {
+     paste(x[1], "_at", sep = "")
+ }))
> length(tempAdria)
```

```
[1] 80
```

Paclitaxel:

```
> tempPac <- read.table(file.path(geneListPath, "paclitaxel.csv"),
+     sep = ",")
> tempPac[1:3, ]
```

```
[1] Paclitaxel Predictor
[2] Probe Description Chromosomal location
[3] 1228_s_at CTAGE family, member 5 14q13.3
39 Levels: 1228_s_at CTAGE family, member 5 14q13.3 ...
```

```
> tempPac <- tempPac[grep(pattern = "_at", tempPac[, 1]), ]
> tempPac <- as.character(tempPac)
> tempPac <- strsplit(tempPac, split = "_at")
> tempPac <- unlist(lapply(tempPac, function(x) {
+     paste(x[1], "_at", sep = "")
+ }))
> length(tempPac)

[1] 36
```

Docetaxel:

```
> tempDoce <- read.table(file.path(geneListPath, "docetaxel.csv"),
+     sep = ",")
> tempDoce[1:3, ]

[1] Docetaxel Predictor
[2] Probe Set ID Gene Title Chromosomal Location
[3] 1259_at zinc finger protein 19 /// zinc finger protein 23 (KOX 16) 16q22
52 Levels: 1259_at zinc finger protein 19 /// zinc finger protein 23 (KOX 16) 16q22 ...

> tempDoce <- tempDoce[grep(pattern = "_at", tempDoce[, 1]), ]
> tempDoce <- as.character(tempDoce)
> tempDoce <- strsplit(tempDoce, split = "_at")
> tempDoce <- unlist(lapply(tempDoce, function(x) {
+     paste(x[1], "_at", sep = "")
+ }))
> length(tempDoce)

[1] 50
```

3 Checking Agreement

First, we load the old probeset lists and the full set of 12588 probesets used.

```
> load(file.path(rDataPath, "features.Rda"))
> load(file.path(rDataPath, "chemoPredictors.Rda"))
```

The old probeset lists involved an off-by-one indexing error, where the offset is relative to the full list of 12588 probeset ids supplied to the quantification software. The new lists should ideally correct this problem and match what the software reports.

3.1 5-FU

The offset was the only problem with the list initially reported for 5-FU; after we corrected for the indexing we got agreement for all 45 of the reported probesets. Let's see what happens here.

```
> rows5fu <- match(reportedFeatures$"5-FU", rownames(predictors))
> old5fuOffOne <- rownames(predictors)[rows5fu + 1]
> length(intersect(temp5fu, old5fuOffOne))

[1] 43
```

The problem has indeed been mostly fixed; 43 of the probesets now reported are indeed returned by the software. However, there are still two outliers.

```
> setdiff(temp5fu, old5fuOffOne)

[1] "152_f_at"  "1712_s_at"

> setdiff(old5fuOffOne, temp5fu)

[1] "151_s_at"  "1713_s_at"

> reportedFeatures$"5-FU"[match(setdiff(old5fuOffOne, temp5fu),
+     old5fuOffOne)]

[1] "1519_at" "1711_at"
```

The two outliers suggest that there may have been a problem with the fix employed to obtain the new lists. Let's look at each of these in turn.

The first problematic probeset is the one initially reported as 1519_at. If we go one index down in the full list initially fed to the software, we get 151_s_at, but the probeset now reported is 152_f_at. This latter probeset is what would be obtained if (a) the full list had all of the probesets in perfect lexicographical order, and (b) the lexical ordering used places underscores before digits. With respect to the list initially supplied to the software, neither of these two conditions holds. Let's take a look at the full list in the vicinity of this first problematic probeset.

```
> bad5FURow1 <- match(reportedFeatures$"5-FU"[match(setdiff(old5fuOffOne,
+     temp5fu), old5fuOffOne)], rownames(predictors))[1]
> rownames(predictors)[bad5FURow1:(bad5FURow1 + 15)]

 [1] "1519_at"   "151_s_at"  "1520_s_at" "1521_at"   "1522_at"   "1523_g_at"
 [7] "1524_at"   "1525_s_at" "1526_i_at" "1527_s_at" "1528_at"   "1529_at"
[13] "152_f_at"  "1530_g_at" "1531_at"   "1532_g_at"
```

Here, looking at the probesets beginning with 152, we see that they do appear to be in lexical order, but with the underscore coming after the digits so that 1529_at precedes 152_f_at.

The relative positioning of underscores and digits does not explain the problem encountered with the second outlying probeset, as the newly reported 1712_s_at should precede the correct 1713_s_at either way. Let's check the indices of these probesets in the full list.

```
> match("1711_at", rownames(predictors))

[1] 6837

> match("1712_s_at", rownames(predictors))

[1] 6038

> match("1713_s_at", rownames(predictors))

[1] 6838
```

The problem here is simply that 1712_s_at is far out of order; the list initially used was not fully sorted.

Thus, the new list for 5FU is mostly correct, but still has a few outliers that do not match the probesets reported by the software.

3.2 Etoposide

As with 5-FU, the offset was the only problem we noted with the list for etoposide. Let's check the agreement for the 50 probesets now.

```
> rowsEtopo <- match(reportedFeatures$ETOPO, rownames(predictors))
> oldEtopoOffOne <- rownames(predictors)[rowsEtopo + 1]
> length(intersect(tempEtopo, oldEtopoOffOne))

[1] 48
```

Again, it's much closer, with 48 out of 50 matching. However, there are outliers again.

```
> setdiff(tempEtopo, oldEtopoOffOne)

[1] "1590_s_at" "1670_at"

> setdiff(oldEtopoOffOne, tempEtopo)

[1] "160020_at" "1680_at"

> reportedFeatures$ETOPO[match(setdiff(oldEtopoOffOne, tempEtopo),
+     oldEtopoOffOne)]
```

```
[1] "159_at" "167_at"
```

Both of the outliers here follow from a problem noted for 5-FU: the relative positioning of underscores and digits in the lexical ordering.

As above, the new list for etoposide is mostly correct, but still has a few outliers that do not match the probesets reported by the software.

3.3 Topotecan

As with 5-FU and etoposide, the offset was the only problem we noted with the list for topotecan. Let's check the agreement for the 150 probesets now.

```
> rowsTopo <- match(reportedFeatures$TOPO, rownames(predictors))
> oldTopoOffOne <- rownames(predictors)[rowsTopo + 1]
> length(intersect(tempTopo, oldTopoOffOne))
```

```
[1] 142
```

Again, it's much closer to what it should be, with 142 out of 150 matching. However, there are outliers again.

```
> setdiff(tempTopo, oldTopoOffOne)
```

```
[1] "116_at"   "160_at"   "33800_at" "34218_at" "36057_at" "378_s_at" "39000_at"
[8] "41097_at"
```

```
> setdiff(oldTopoOffOne, tempTopo)
```

```
[1] "115_at"   "159_at"   "33900_at" "34318_at" "35751_at" "376_at"   "39100_at"
[8] "41197_at"
```

```
> reportedFeatures$TOPO[match(setdiff(oldTopoOffOne, tempTopo),
+     oldTopoOffOne)]
```

```
[1] "1159_at"    "1599_at"    "338_at"     "34317_g_at" "35750_at"
[6] "37699_at"   "390_at"     "41196_at"
```

Here, we see a new problem. For most of the outliers, the problem is still the underscore/digit positioning:

- (entry 1 of 8) 1159_at becomes 116_at when it should be 115_at
- (entry 2 of 8) 1599_at becomes 160_at when it should be 159_at
- (entry 3 of 8) 338_at becomes 33800_at when it should be 33900_at

- (entry 7 of 8) 390_at becomes 39000_at when it should be 39100_at

For some of the others, however, there appear to be typos, as the ones now reported do not follow in lexical order from the initial ones even when the underscore issue is taken into account. Two of these involve single digit mistypes:

- (entry 4 of 8) 34317_g_at becomes 34218_at when it should be 34318_at
- (entry 8 of 8) 41196_at becomes 41097_at when it should be 41197_at

The other two involve more extensive reorderings:

- (entry 6 of 8) 37699_at becomes 378_s_at when it should be 376_at
- (entry 5 of 8) 35750_at becomes 36057_at when it should be 35751_at

For the first of these, we suspect that 377_g_at was being aimed for. For the second, we do not understand how this came about.

The presence of typos is problematic, as it implies a human transcription step instead of automated processing.

Thus, as above, the new list for topotecan is mostly correct, but still has a few outliers that do not match the probesets reported by the software.

3.4 Adriamycin

The list for adriamycin is a bit more difficult, as we found that offsetting by one only allowed us to capture 75 of the 80 software-generated probesets before: there were 5 outliers that had been introduced in some other fashion. Let's see what happens here.

```
> rowsAdria <- match(reportedFeatures$ADRIA, rownames(predictors))
> oldAdriaOffOne <- rownames(predictors)[rowsAdria + 1]
> length(intersect(tempAdria, oldAdriaOffOne))

[1] 77
```

Here, we get agreement for 77 of the 80 probesets when we offset the initial list by one. This suggests that at least some of the five initial outliers were offset as well.

```
> setdiff(tempAdria, oldAdriaOffOne)

[1] "101_at"   "191_at"   "36826_at"

> setdiff(oldAdriaOffOne, tempAdria)

[1] "110_at"   "190_at"   "36828_at"
```

```
> reportedFeatures$ADRIA[match(setdiff(oldAdriaOffOne, tempAdria),
+     oldAdriaOffOne)]

[1] "1109_s_at" "1909_at"   "36827_at"
```

Of the three mismatches shown here, the first two follow from the now-familiar underscore/digit issue, and the last appears to be a typo following from counting up (from 36827_at to 36826_at) instead of down (from 36827_at to 36828_at).

Now, as it happens, the five probesets that we had flagged as non-offset outliers in the initial list are 1258_s_at, 1847_s_at, 1909_at, 1910_s_at, and 2034_s_at. A one-step offset was attempted for all of these (even though a more extensive correction is required), and the offset was successful for four (the middle one encountered the underscore problem).

These items remain outliers, in that we still see no way in which they could have been generated by the software. In this case, offsetting exacerbates the problem, as we do not see the rationale for most of the five "new outliers" that result, whereas the five "initial outliers" had some obvious biological interpretations.

3.5 Paclitaxel

For paclitaxel, the initial list involved 35 probesets, of which we found seven to be outliers. The revised table reports 36 probesets for paclitaxel, so there is obviously something that has changed.

```
> rowsPac <- match(reportedFeatures$PAC, rownames(predictors))
> oldPacOffOne <- rownames(predictors)[rowsPac + 1]
> length(intersect(tempPac, oldPacOffOne))

[1] 0
```

Here, the problem is far more severe than we anticipated; *none* of the probesets match. Let's take a look at the first 10 from both the initial and the revised lists.

```
> tempPac[1:10]

 [1] "1228_s_at"  "1356_at"    "243_g_at"   "250_at"     "31463_s_at"
 [6] "31511_at"   "31546_at"   "31867_at"   "32226_at"   "32378_at"

> reportedFeatures$PAC[1:10]

 [1] "1217_g_at" "1258_s_at" "1586_at"   "1802_s_at" "1823_g_at" "1870_at"
 [7] "1878_g_at" "1881_at"   "1902_at"   "2000_at"

> reportedFeatures$CYT[1:10]

 [1] "1002_f_at" "1190_at"   "1198_at"   "1891_at"   "1983_at"   "200_at"
 [7] "2037_s_at" "31430_at"  "31431_at"  "31719_at"
```

We don't see a pattern. We tried the old signature for cytoxan as well, on the off chance that this one was meant, but that didn't work either.

At this point, we have no idea how the list for paclitaxel was generated. It does not match what we are able to get from the software at all.

3.6 Docetaxel

Docetaxel is the one we expected to be most problematic *a priori*, as we found 19 of the 50 initially reported probesets to be outliers that could not be explained by simple offsetting. Let's see what type of agreement we get with the new list.

```
> rowsDoce <- match(reportedFeatures$DOCE, rownames(predictors))
> oldDoceOffOne <- rownames(predictors)[rowsDoce + 1]
> length(intersect(tempDoce, oldDoceOffOne))

[1] 43
```

Here, we get agreement for 43 of the 50 probesets when we offset the initial list by one. This suggests that most of the 19 initial outliers were offset as well.

```
> setdiff(tempDoce, oldDoceOffOne)

[1] "1410_at"  "1804_at"  "321_at"   "32828_at" "33434_at" "36642_at" "36663_at"

> setdiff(oldDoceOffOne, tempDoce)

[1] "1420_s_at" "1803_at"   "320_at"    "32332_at"  "33444_at"  "36641_at"
[7] "38663_at"

> reportedFeatures$DOCE[match(setdiff(oldDoceOffOne, tempDoce),
+     oldDoceOffOne)]

[1] "141_s_at"  "1802_s_at" "32099_at"  "32331_at"  "33443_at"  "36640_at"
[7] "38662_at"
```

Looking at the seven outliers, only two of them seem to be underscore/digit driven: 141_s_at and 32099_at. For the other five, typos of some type seem to be involved.

- 1802_s_at should be 1803_t, but 1804_at is now reported (an extra skip).
- 32331_at should be 32332_at, but 32828_at is now reported (we don't undertstand this one).
- 33443_at should be 33444_at, but 33434_at is now reported (a simple typo).
- 36640_at should be 36641_at, but 36642_at is now reported (and extra skip).
- 38662_at should be 38663_at, but 36663_at is now reported (a simple typo).

As was the case with adriamycin, the outliers have also been offset by one, but they are not output by the software in either form.

4 Summary

The new probeset lists are now indeed mostly correct for five of the six drugs given; for paclitaxel the new list appears simply wrong. However, even though the lists for the other five are "better" than they were, they are still not correct, in that they are still not the probeset lists that would be reported by the software when the models are fit. The new lists have apparently been generated by starting with the old lists and attempting an offset close to (but not the same as) the one encountered. This attempt was apparently made by hand (hence typos), and does not address the issue of the outliers we found in the signatures for adriamycin, paclitaxel and docetaxel.