Getting the Test Data Quantifications

Keith A. Baggerly

May 20, 2008

1 Introduction

In order to check the predictions made for the Bonnefoi et al. samples, we need the array quantifications. Quantifications and CEL files are available from GEO (GSE6861). In this report, we load and parse this data for later checking.

2 Options and Libraries

> options(width = 80)
> library(affy)
> library(XML)

   We also like to use “jet” colors for heatmap displays on occasion.

> source(file.path("OtherData", "jet.colors.R"))

3 Loading XML Info from GEO

We begin by grabbing data from the MINiML files available from GEO. We check the XML descriptions first to see what’s there. (We find it useful to skim this file in a web browser concurrently to figure out which fields we want to extract.)

> gse6861xml <- xmlTreeParse(file.path("PublicData", "MINiML", 
  + "GSE6861_family.xml"))
> root <- xmlRoot(gse6861xml)
> xmlName(root)
[1] "MINiML"

> xmlSize(root)
[1] 168

> table(xmlSApply(root, xmlName))

<table>
<thead>
<tr>
<th>Contributor</th>
<th>Database</th>
<th>Platform</th>
<th>Sample</th>
<th>Series</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>160</td>
<td>1</td>
</tr>
</tbody>
</table>
We have entries for 5 contributors, a database (referencing GEO), a platform (describing the X3P array), 160 samples, and the GSE series, respectively. Almost all of the information we’ll want is sample specific, so we’ll focus on those entries. We begin by looking at the first sample entry (entry 8 in the tree) in its entirety.

```r
> xmlSApply(root, xmlName)[c(1:8, 167:168)]

Contributor Contributor Contributor Contributor Contributor
Database Platform Sample Sample Series

> idxSample <- which(xmlSApply(root, xmlName) == "Sample")
```

We have entries for 5 contributors, a database (referencing GEO), a platform (describing the X3P array), 160 samples, and the GSE series, respectively. Almost all of the information we’ll want is sample specific, so we’ll focus on those entries. We begin by looking at the first sample entry (entry 8 in the tree) in its entirety.

```r
> root[[8]]

<Sample iid="GSM158188">
  <Status database="GEO">
    <Submission-Date>2007-01-25</Submission-Date>
    <Release-Date>2007-11-23</Release-Date>
    <Last-Update-Date>2007-11-23</Last-Update-Date>
  </Status>
  <Title>HB01bis</Title>
  <Accession database="GEO">GSM158188</Accession>
  <Type>RNA</Type>
  <Channel-Count>1</Channel-Count>
  <Channel position="1">
    <Source>Human Breast Tumor</Source>
    <Organism>Homo sapiens</Organism>
    <Characteristics>npCR</Characteristics>
    <Molecule>total RNA</Molecule>
    <Extract-Protocol>Qiagen RNeasy Mini Kit</Extract-Protocol>
    <Label>biotin</Label>
    <Label-Protocol>Standard Affymetrix procedure</Label-Protocol>
  </Channel>
  <Hybridization-Protocol>Standard Affymetrix procedure</Hybridization-Protocol>
  <Scan-Protocol>Standard Affymetrix procedure</Scan-Protocol>
  <Description>Series of 161 tumours</Description>
  <Data-Processing>RMA</Data-Processing>
  <Platform-Ref ref="GPL1352"/>
  <Contact-Ref ref="contri1"/>
  <Supplementary-Data type="CEL">ftp://ftp.ncbi.nih.gov/pub/geo/DATA/supplementary/samples/GSM158188/GSM158188.CEL.gz</Supplementary-Data>
  <Data-Table>
    <Column position="1">
      <Name>ID_REF</Name>
    </Column>
    <Column position="2">
      <Name>VALUE</Name>
      <Description>RMA-calculated signal intensity</Description>
    </Column>
  </Data-Table>
</Sample>
```
Looking at the entries, we clearly want Title (the array ID assigned by the contributors), Accession (the corresponding GSM ID), and Characteristics within Channel 1 (PCR or nPCR status). We grab the array IDs first.

```R
> arrayID <- sapply(idxSample, function(x) {
+   xmlValue(root[[x]][['Title']])
+ })
> as.character(arrayID)
```

[1] "HB01bis"  "HB02bis"  "HB03bis"  "HB04bis"  "HB05bis"  "HB06bis"
[7] "HB07bis"  "HB08bis"  "HB09bis"  "HB10bis"  "HB11bis"  "HB12bis"
[13] "HB13bis"  "HB14bis"  "HB15bis"  "HB16bis"  "HB17bis"  "HB18bis"
[19] "HB19bis"  "HB20bis"  "HB21bis"  "HB22bis"  "HB23bis"  "HB24bis"
[25] "HB25bis"  "HB26bis"  "HB27bis"  "HB28bis"  "HB29bis"  "HB30bis"
[31] "HB31bis"  "HB32bis"  "HB33bis"  "HB34bis"  "HB35bis"  "HB36bis"
[37] "HB37bis"  "HB38bis"  "HB39bis"  "HB40bis"  "HB41bis"  "HB42bis"
[43] "HB43bis"  "HB44bis"  "HB45bis"  "HB46bis"  "HB47bis"  "HB48bis"
[49] "HB49bis"  "HB50bis"  "HB51bis"  "HB52bis"  "HB53bis"  "HB54bis"
[55] "HB55bis"  "HB56bis"  "HB57bis"  "HB58bis"  "HB59bis"  "HB60bis"
[61] "HB61bis"  "HB62bis"  "HB63bis"  "HB64bis"  "HB65bis"  "HB66bis"
[67] "HB67bis"  "HB68bis"  "HB69bis"  "HB70bis"  "HB71bis"  "HB72bis"
[73] "HB73bis"  "HB74bis"  "HB75bis"  "HB76bis"  "HB77bis"  "HB78bis"
[79] "HB79bis"  "HB80bis"  "HB81bis"  "HB82bis"  "HB83bis"  "HB84bis"
[85] "HB85bis"  "HB86bis"  "HB87bis"  "HB88bis"  "HB89bis"  "HB90bis"
[91] "HB91bis"  "HB92bis"  "HB93bis"  "HB94bis"  "HB95bis"  "HB96bis"
[97] "HB97bis"  "HB98bis"  "HB99bis"  "HB100bis"  "HB101bis"  "HB102bis"
[103] "HB103"  "HB104"  "HB105"  "HB106"  "HB107"  "HB108"
[109] "HB109"  "HB110"  "HB111"  "HB112"  "HB113"  "HB114"
[115] "HB115"  "HB116"  "HB117"  "HB118"  "HB119"  "HB120"
[121] "HB121"  "HB122"  "HB123"  "HB124"  "HB125"  "HB126"
[127] "HB127"  "HB128"  "HB129"  "HB132"  "HB133"  "HB134"
[133] "HB135"  "HB136"  "HB137"  "HB138"  "HB139"  "HB140"
[139] "HB141"  "HB142"  "HB143"  "HB144"  "HB145"  "HB146"
[145] "HB147"  "HB148"  "HB149"  "HB150"  "HB151"  "HB152"
[151] "HB153"  "HB154"  "HB155"  "HB156"  "HB157"  "HB158"
[157] "HB159"  "HB160"  "HB161"  "HB162"
The array ids use a fairly consistent format (HB01bis-HB102bis, followed by HB103-HB162); we will discount the bis suffix for now. The numerical indices are monotonically increasing and are almost sequential (130 and 131 are missing).

Next, we grab the corresponding GSM ids.

```r
> gsmID <- sapply(idxSample, function(x) {
+   xmlValue(root[[x]]['Accession'])
+ })
> as.character(gsmID)
```

The GSM ids are sequential: GSM158188-GSM158347 (there are no gaps here).

Finally, we grab the corresponding patient response information.

```r
> patientResponse <- sapply(idxSample, function(x) {
+   xmlValue(root[[x]]['Channel']['Characteristics'])
+ })
> as.character(patientResponse)
```
The patient response values have clear interpretations, and do not appear to be grouped in any grossly visible pattern.

We also check a few other fields just to confirm that they are the same for all entries.

These answers are indeed consistent. All 160 samples are labeled as coming from a series of 161 tumors (suggesting that one, probably 130 or 131, was omitted). All of the samples were processed using RMA.

Let’s store the relevant info for later.

> geoInfo <- data.frame(arrayID = I(as.character(arrayID)), gsmID = I(as.character(gsmID)),
+ response = as.character(patientResponse))
> geoInfo[1:2, ]
arrayID   gsmID response
1  HB01bis GSM158188   npCR
2  HB02bis GSM158189   npCR
4 Loading the Quantification Data

The MINiML files also include one table of quantifications for each array. We check a few of the first entries here.

```r
> temp <- read.table(file.path("PublicData", "MINiML", "GSM158347-tbl-1.txt"),
+ row.names = 1)
> dim(temp)
[1] 61359 1
> temp[1:5, ]
> rownames(temp)[1:5]
[1] "1053_3p_at" "117_3p_at" "1494_3p_f_at" "1552275_3p_s_at"
[5] "1552281_3p_at"
> tempDir <- dir(file.path("PublicData", "MINiML"))
> tempDir <- tempDir[grep("^GSM", tempDir)]
> length(tempDir)
[1] 160
```

Everything seems to be in order. We load these quantifications into a matrix for easier manipulation later. We adjust the column names to use the array ids instead of names such as “GSM158347-tbl-1.txt”.

```r
> rda <- "lancetRMAFromGEO"
> rdaFile <- paste("RDataObjects", paste(rda, "Rda", sep = "."),
+ sep = .Platform$file.sep)
> if (file.exists(rdaFile)) {
+ cat(paste("Loading", rda, "from cache\n"))
+ load(rdaFile)
+ } else {
+ lancetRMAFromGEO <- matrix(0, dim(temp)[1], length(tempDir))
+ rownames(lancetRMAFromGEO) <- rownames(temp)
+ colnames(lancetRMAFromGEO) <- tempDir
+ for (i1 in 1:length(tempDir)) {
+   temp <- read.table(file.path("PublicData", "MINiML",
+                           tempDir[i1]), row.names = 1)
+   if (all(rownames(lancetRMAFromGEO) == rownames(temp))) {
+     lancetRMAFromGEO[, tempDir[i1]] <- temp[, 1]
+   } else {
+     lancetRMAFromGEO[, tempDir[i1]] <- temp[rownames(lancetRMAFromGEO),
+                                               1]
+   }
+ }
```
tempIDs <- substr(colnames(lancetRMAFromGEO, 1, 9))
colnames(lancetRMAFromGEO) <- geoInfo$arrayID[match(geoInfo$gsmID, tempIDs)]
save(lancetRMAFromGEO, file = rdaFile)
}

Loading lancetRMAFromGEO from cache

Next, we compute RMA values directly from the CEL files using justRMA. (Note: this takes a long time; about 20 minutes on my MacBook Pro laptop.) As with the GEO quantifications, we relabel columns to use the array ids instead of values like “GSM158188.CEL.gz”.

> rda <- "lancetRMAFromCEL"
> rdaFile <- paste("RDataObjects", paste(rda, "Rda", sep = "."), 
+ sep = .Platform$file.sep)
> if (file.exists(rdaFile)) {
+ cat(paste("Loading", rda, "from cache\n"))
+ load(rdaFile)
+ } else {
+ lancetRMAFromCEL <- justRMA(celfile.path = file.path("PublicData", 
+ "CEL"), compress = TRUE)
+ lancetRMAFromCEL <- exprs(lancetRMAFromCEL)
+ tempIDs <- substr(colnames(lancetRMAFromCEL, 1, 9))
+ colnames(lancetRMAFromCEL) <- geoInfo$arrayID[match(geoInfo$gsmID, tempIDs)]
+ save(lancetRMAFromCEL, file = rdaFile)
+ }

Loading lancetRMAFromCEL from cache

5 Check Agreement of Quantifications

Given the quantifications, we want to do some quick checks to see how well they line up. Given that RMA was used in both instances, we expect some very high correlations (they need not be perfect, especially if the “missing” 161st tumor was used to compute the RMA quantifications posted).

First, we check that the data matrices are lined up the same way.

> dim(lancetRMAFromCEL)
[1] 61359 160

> dim(lancetRMAFromGEO)
[1] 61359 160

> sum(rownames(lancetRMAFromGEO) == rownames(lancetRMAFromCEL))
[1] 61359

> sum(colnames(lancetRMAFromGEO) == colnames(lancetRMAFromCEL))
All of the rows and columns line up. Next, we check the pairwise sample correlations within and between groups. In all cases, we expect to see very high values only on the main diagonal.

```r
> corGEO <- cor(lancetRMAFromGEO)
> corCEL <- cor(lancetRMAFromCEL)
> corGEOwCEL <- cor(lancetRMAFromGEO, lancetRMAFromCEL)
```

We also compute correlations after centering and scaling by row.

```r
> corCenGEO <- cor(t(scale(t(lancetRMAFromGEO))))
> corCenCEL <- cor(t(scale(t(lancetRMAFromCEL))))
> corCenGEOwCEL <- cor(t(scale(t(lancetRMAFromGEO))), t(scale(t(lancetRMAFromCEL))))
```

### 5.1 GEO Correlations

We check the GEO quantifications first.

```r
> sum(corGEO > 0.999)
[1] 162
```

```r
> temp <- which(corGEO > 0.999, arr.ind = TRUE)
> temp[temp[, 1] != temp[, 2], ]
```

<table>
<thead>
<tr>
<th>row</th>
<th>col</th>
</tr>
</thead>
<tbody>
<tr>
<td>HB41bis</td>
<td>41 40</td>
</tr>
<tr>
<td>HB40bis</td>
<td>40 41</td>
</tr>
</tbody>
</table>

```r
> corGEO[39:42, 39:42]
```

<table>
<thead>
<tr>
<th></th>
<th>HB39bis</th>
<th>HB40bis</th>
<th>HB41bis</th>
<th>HB42bis</th>
</tr>
</thead>
<tbody>
<tr>
<td>HB39bis</td>
<td>1.00000000</td>
<td>0.8676024</td>
<td>0.8664591</td>
<td>0.8969153</td>
</tr>
<tr>
<td>HB40bis</td>
<td>0.8676024</td>
<td>1.0000000</td>
<td>0.9998865</td>
<td>0.8928609</td>
</tr>
<tr>
<td>HB41bis</td>
<td>0.8664591</td>
<td>0.9998865</td>
<td>1.0000000</td>
<td>0.8916970</td>
</tr>
<tr>
<td>HB42bis</td>
<td>0.8969153</td>
<td>0.8928609</td>
<td>0.8916970</td>
<td>1.0000000</td>
</tr>
</tbody>
</table>

Something is slightly off; the entries for HB40 and HB41 are much more tightly correlated than we would have expected with different arrays. This suggests that these may actually be requantifications of the same array.

Let’s do a quick check for broader patterns as well.

```r
> image(1:160, 1:160, corGEO < 0.91, xlab = "Array Index", ylab = "Array Index", +    main = "Pairwise GEOCors > 0.91")
> abline(v = 102.5, h = 102.5)
```
We see three visible “blocks” on the diagonal using a correlation cutoff of 0.91 (other values were tried with similar results; this specific value was chosen subjectively to give the best visual contrast). One block corresponds to the division between those samples with a “bis” suffix (the first 102 samples) and the rest; this division is shown on the plot with lines. The second block is in the upper right corner of the sub-block for the first 102 samples.

Now, we’re also interested in the correlations that result after we’ve centered and scaled the values for each row (probeset), as these focus our attention more on concerted changes. Let’s take a quick look at the distribution of the correlations after centering by row.

```r
> hist(corCenGEO, breaks = 50, xlab = "Correlation", main = "Pairwise GEO Sample Correlations, Rows Centered")
```
Pairwise GEO Sample Correlations, Rows Centered

These are for the most part centered near zero. We can again check for visual patterns.

```r
> image(1:160, 1:160, corCenGEO < 0.15, xlab = "Array Index", ylab = "Array Index", +     main = "Pairwise Centered GEO Cors > 0.15")
> abline(v = c(72.5, 102.5), h = c(72.5, 102.5))
```
The grouping noted before is even more stark now, and clear enough that we’re willing to specify the second border as coming after the first 72 samples.

### 5.2 CEL Correlations

Next, we perform the same checks for the sample correlations between pairs of justRMA quantifications.

```r
> sum(corCEL > 0.99)
[1] 160
> corCEL[39:42, 39:42]
```

```
   HB39bis HB40bis HB41bis HB42bis
HB39bis 1.0000000 0.9521609 0.9385439 0.9575648
HB40bis 0.9521609 1.0000000 0.9434628 0.9675374
HB41bis 0.9385439 0.9434628 1.0000000 0.9531455
HB42bis 0.9575648 0.9675374 0.9531455 1.0000000
```
All of the extremely high correlations are along the main diagonal here. Looking at samples 40 and 41, they don’t appear especially correlated, again suggesting that the agreement seen with the GEO quantifications is an artifact.

Next, we check for broader patterns.

```r
gtbox
gtbox
> image(1:160, 1:160, corCEL < 0.965, xlab = "Array Index", ylab = "Array Index", + main = "Pairwise CEL Cors > 0.965")
```

![Pairwise CEL Cors > 0.965](image)

The divisions are far less apparent than with the GEO quantifications.

Next, we check the results after row centering.

```r
gtbox
gtbox
> hist(corCenCEL, breaks = 50, xlab = "Correlation", main = "Pairwise CEL Sample Correlations, Rows Centered")
```
Again, things are centered around zero, though somewhat more dispersed.

```r
> image(1:160, 1:160, corCenCEL < 0.25, xlab = "Array Index", ylab = "Array Index",
+ main = "Pairwise Centered CEL Cors > 0.25")
```
After centering, the same divisions noted in the GEO correlations are seen here, though not so starkly.

5.3 GEO/CEL Cross-Correlations
Looking across the GEO/CEL divide, we begin by looking at just how big the correlations are.

> hist(corGEOwCEL, breaks = 50, xlab = "Correlation", main = "Pairwise GEO/CEL Sample Correlations")
What is surprising here is that there are no values in excess of 0.9. The CEL and GEO quantifications differ, and not just by a scaling factor. Let’s look for broader patterns.

```r
> image(1:160, 1:160, corGEOwCEL < 0.84, xlab = "Array Index", +     ylab = "Array Index", main = "Pairwise GEO/CEL Cors > 0.84")
```
Well, there is a clear story here – something is different between the first 102 samples and the last 58. Now let’s look at things after centering the data matrices.

```r
> hist(corCenGEOwCEL, breaks = 50, xlab = "Correlation", main = "Pairwise GEO/CEL Sample Correlations")
```
Again, centered about zero. What do the broad patterns look like?

```r
> image(1:160, 1:160, corCenGEOwCEL < 0.12, xlab = "Array Index",
+       ylab = "Array Index", main = "Pairwise GEO/CEL Cors > 0.12, Rows Centered")
```
We’re seeing the same divisions as before, which is reassuring. Where are the biggest correlations now?

```r
> image(1:160, 1:160, corCenGEOwCEL < 0.45, xlab = "Array Index",
+       ylab = "Array Index", main = "Pairwise GEO/CEL Cors > 0.45, Rows Centered")
> sum(corCenGEOwCEL > 0.45)
[1] 160
```
After centering the rows, the biggest correlations are between nominally identical samples. This is reassuring. There is only one big entry on the off-diagonal:

```r
> corCenGEOwCEL[39:42, 39:42]
```

HB39bis  HB40bis  HB41bis  HB42bis
HB39bis  0.68409841 0.04164098 0.0002666984 0.05935946
HB40bis  -0.04977208 0.02318002 0.6894704715 0.11096424
HB41bis   0.01975627 0.01822018 0.7021548121 0.06761917
HB42bis   0.03986600 0.12226289 0.0723827372 0.63343039

There is no GEO profile that has a high correlation with that from the CEL file for HB40, but there are two for HB41. HB40 is the one that is incorrectly recorded at GEO.

We’d also like to get a better feel for the correlations after centering (suppressing the highest ones on the main diagonal solely to enhance the visual contrast.

```r
> temp <- corCenGEOwCEL
> diag(temp) <- 0.3
```
> temp[40, 41] <- 0.3
> image(1:160, 1:160, temp, xlab = "Array Index", ylab = "Array Index",
+     main = "Pairwise GEO/CEL Cors", col = jet.colors(64))

Again, we see the three previously identified blocks showing up on the main diagonal.

6 Checking Differences

We had expected the GEO and CEL quantifications to be about the same. This was not the case. Let’s take a closer look at the GEO data to see if we can get some idea of what’s going on.

> geoMins <- apply(lancetRMAFromGEO, 2, min)
> geoMaxs <- apply(lancetRMAFromGEO, 2, max)
> table(geoMins, geoMaxs)

geoMaxs
geoMins       14.50555962 14.79550696
This gives a clear idea of part of what’s going on. The GEO values have been sorted to have the same quantiles for almost all samples (sample 40 is again the outlier). It is not clear how the distribution used was chosen, however – we don’t know how negative values could result.

Further, quantile matching was not the only processing step involved. If we check the first few rows from the first two arrays,

```r
> cbind(lancetRMAFromGEO[1:3, 1:2], lancetRMAFromCEL[1:3, 1:2])
```

```r
HB01bis HB02bis HB01bis HB02bis
1053_3p_at 8.431282 9.814409 8.575480 10.434413
117_3p_at  7.355402 8.052445 7.929677 8.371615
1494_3p_f_at 8.003195 7.412932 7.467080 7.651403
```

we see that rank-ordering is not maintained within a sample, nor do the differences between pairs of samples have the same sign. We do not yet see how to get from one set of “RMA” values to the other.

### 7 Summary

We parsed the XML data from GEO, loaded the quantifications from GEO, and computed quantifications from CEL files.

- There is a clear division in the names of the GEO array ids: the first 102 have a “bis” suffix, and the remaining 58 do not.
- The GEO quantification for sample 40 appears to actually be a requantification of sample 41.
- The GEO and CEL quantifications do not agree. The GEO quantification vectors appear to have been quantile normalized, with the exception of sample 40.
- There are three obvious “blocks” visible in the data when correlations are shown in array index order. The biggest division occurs after sample 102 at the boundary noted above. The next biggest occurs after sample 72 in the “bis” block.
- After centering and scaling the rows, the GEO and CEL quantifications seem to be reasonably well correlated.

### 8 Appendix

#### 8.1 Saves

```r
> save(geoInfo, file = file.path("RDataObjects", "geoInfo.Rda"))
> save(corGEO, corCEL, corGEOwCEL, corCenGEO, corCenCEL, corCenGEOwCEL,
+      file = file.path("RDataObjects", "pairwiseCors.Rda"))
```
8.2 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 States.1252;

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

other attached packages:
XML  affy  affyio  Biobase  ROC
"1.9-0"  "1.14.2"  "1.4.1"  "1.14.1"  "1.10.0"

References