Building novartisAll.Rda

Keith A. Baggerly

July 18, 2009

Contents

1 Executive Summary **1**
  1.1 Introduction . . . 1
  1.2 Methods . . . 2
  1.3 Results . . . 2

2 Options and Libraries **2**

3 Loading and Parsing Data **2**
  3.1 Loading WEB DATA . . . 2
  3.2 Checking Order Entry Consistency . . . 3
  3.3 Checking WEB HOOKS . . . 4
  3.4 Reformatting Cell Line Names . . . 4
  3.5 Arranging Quantifications in Matrix Form . . . 5
  3.6 Extracting the Annotation Information . . . 5

4 Save Rda File **5**

5 Appendix **6**
  5.1 File Location . . . 6
  5.2 Saves . . . 6
  5.3 SessionInfo . . . 6

List of Figures

List of Tables

1 Executive Summary

1.1 Introduction

In this report, we assemble a matrix of the U95A triplicate quantifications of the NCI-60 cell lines available from Novartis.

1.2 Methods

We acquired one zip file of quantifications, “WEB_DATA_NOVARTIS_ALL.ZIP,” from the NCI DTPs download page for molecular target data, `http://www.dtp.nci.nih.gov/mtargets/download.html`. We acquired another, older, zip file of quantifications, “WEB_HOOKS_NOV_GC_ALL.ZIP,” directly from the NCI DTP’s FTP server `http://dtpws4.ncifcrf.gov/FTP`. We parsed the contents (a) to arrange the quantifications in a probeset by sample matrix, and (b) to extract annotation information related to the samples. We altered the formatting of the cell line names to match that provided in the drug sensitivity data. We compared numbers to confirm that the quantification values were the same.

1.3 Results

Quantifications from both files are the same. We created a “novartisAll” matrix of array quantifications and a “novartisAllInfo” data frame of sample information. We stored these in RDataObjects as “novartisAll.Rda.”

2 Options and Libraries

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

3 Loading and Parsing Data

In order to assemble signatures of drug sensitivity from cell line data, we need microarray quantifications of the cell lines involved. Several years ago, Novartis used Affymetrix U95A arrays to profile the NCI-60 cell line panel in triplicate, producing 180 arrays worth of data in all. These quantifications are freely available from the NCI DTP.

We acquired one zip file of quantifications, “WEB_DATA_NOVARTIS_ALL.ZIP,” from the NCI DTPs download page for molecular target data, `http://www.dtp.nci.nih.gov/mtargets/download.html`. We acquired another, older, zip file of quantifications, “WEB_HOOKS_NOV_GC_ALL.ZIP,” directly from the NCI DTP’s FTP server `http://dtpws4.ncifcrf.gov/FTP`. The older set was used by Coombes et al. [1]. We believe the more recent data contains almost all the same entries modulo column rearrangement, but the more recent file uses gene assignments from Unigene build U214 (Jun 08).

3.1 Loading WEB DATA

First, we load in the most recent data.

```
> ## The columns of WEB_DATA_NOVARTIS_ALL.TXT are:
> ## [1] Probe Set Name,ID,Gene,panelnbr,cellnbr,pname,
> ## [7] cellname,Signal,Detection,P Value
>
> d1 <- date()
> tempTable1 <-
+   read.table(unz(file.path("RawData","NCI60",
+                             "WEB_DATA_NOVARTIS_ALL.ZIP"),
+                  "WEB_DATA_NOVARTIS_ALL.TXT"),
+              header=TRUE, sep=",", quote="",
+              colClasses=c("character","character","NULL","NULL",
+                "NULL","NULL","character","numeric","NULL","NULL"))
```

```
> d2 <- date()
> c(d1,d2)
```

```
[1] "Sat Jul 18 13:18:15 2009" "Sat Jul 18 13:18:46 2009"
```

```
> dim(tempTable1)
```

```
[1] 2272680       4
```

```
> nProbes  <- 12626
> nSamples <-   180
> nProbes * nSamples
```

```
[1] 2272680
```

```
> tempTable1[1:3,]
```

```
  Probe.Set.Name        ID cellname    Signal
1       36460_at GC26855_B    K-562 120.47874
2       36460_at GC26855_B   MOLT-4 113.54223
3       36460_at GC26855_B CCRF-CEM  67.42072
```

This takes about 30 seconds on my laptop, and the entries appear to have a regular structure.

3.2 Checking Order Entry Consistency

Next, we explicitly check that values for each probeset are grouped together in the same order each time.

```
> probesetNames <- tempTable1[seq(from = 1, to = nProbes * nSamples,
+     by = nSamples), "Probe.Set.Name"]
> all(tempTable1[, "Probe.Set.Name"] == rep(probesetNames, each = nSamples))
```

```
[1] TRUE
```

```
> replicateSet <- tempTable1[1:nSamples, "ID"]
> replicateSet <- substr(replicateSet, nchar(replicateSet), nchar(replicateSet))
> cellLine <- tempTable1[1:nSamples, "cellname"]
> all(substr(tempTable1[, "ID"], nchar(tempTable1[, "ID"]), nchar(tempTable1[,
+     "ID"])) == rep(replicateSet, times = nProbes))
```

```
[1] TRUE
```

```
> all(tempTable1[, "cellname"] == rep(cellLine, times = nProbes))
```

```
[1] TRUE
```

The sample order is the same within each probeset.

3.3 Checking WEB HOOKS

Next, we load the older data and confirm that the relevant table entries are the same.

```
> d3 <- date()
> tempTable2 <- read.table(unz(file.path("RawData", "NCI60", "WEB_HOOKS_NOV_GC_ALL.ZIP"),
+     "WEB_HOOKS_NOV_GC_ALL.TXT"), header = TRUE, sep = ",", quote = "",
+     colClasses = c("character", "character", "NULL", "character",
+         "NULL", "NULL", "NULL", "numeric", "NULL", "NULL"))
> d4 <- date()
> c(d3, d4)
```

```
[1] "Sat Jul 18 13:18:48 2009" "Sat Jul 18 13:19:18 2009"
```

```
> dim(tempTable2)
```

```
[1] 2272680       4
```

```
> all(tempTable1 == tempTable2)
```

```
[1] TRUE
```

Again, this takes about 30 seconds on my laptop. All of the relevant entries are the same, so we can work with just the first set.

3.4 Reformatting Cell Line Names

Next, we check whether the formatting of the cell line names is the same as that used in the drug sensitivity data. To do this, we load in a table of information extracted from the GI50 table.

```
> nci60Info <- read.table(file.path("RawData", "NCI60", "nci60Info.csv"),
+     sep = ",", header = TRUE, colClasses = c("character", "character",
+         "numeric", "numeric"))
> nci60Info[1:3, ]
```

```
       CELL                PANEL PANELNBR CELLNBR
1   NCI-H23 Non-Small Cell Lung        1       1
2  NCI-H522 Non-Small Cell Lung        1       3
3 A549/ATCC Non-Small Cell Lung        1       4
```

```
> which(sort(nci60Info[, "CELL"]) != sort(unique(cellLine)))
```

```
[1] 41
```

```
> sort(nci60Info[, "CELL"])[41]
```

```
[1] "RXF 393"
```

```
> sort(unique(cellLine))[41]
```

```
[1] "RXF-393"
```

```
> cellLine[cellLine == "RXF-393"] <- "RXF 393"
```

All but one of the names have the same format, and the one that was different was easily fixed.

3.5 Arranging Quantifications in Matrix Form

Next, we extract the quantification data and arrange it as a matrix.

```
> tempMatrix <- matrix(tempTable1[, "Signal"], nrow = nProbes,
+     ncol = nSamples, byrow = TRUE)
> which(duplicated(probesetNames))
```

```
[1] 6123
```

```
> probesetNames[6122:6123]
```

```
[1] "100_g_at" "100_g_at"
```

```
> all(tempMatrix[6122, ] == tempMatrix[6123, ])
```

```
[1] TRUE
```

```
> probesetNames <- probesetNames[-6123]
> tempMatrix <- tempMatrix[-6123, ]
> rownames(tempMatrix) <- probesetNames
> colnames(tempMatrix) <- paste(cellLine, replicateSet, sep = ".")
> novartisAll <- tempMatrix
> dim(novartisAll)
```

```
[1] 12625   180
```

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

```
           K-562.B  MOLT-4.B CCRF-CEM.B
36460_at 120.47874 113.54223   67.42072
36461_at  98.52838  80.76026   72.12962
36462_at 219.98921 185.12505  255.20783
```

3.6 Extracting the Annotation Information

Finally, we bundle the relevant annotation information in a data frame.

```
> novartisAllInfo <- data.frame(cellLine = I(cellLine), replicateSet = I(replicateSet))
> rownames(novartisAllInfo) <- colnames(novartisAll)
> novartisAllInfo[1:3, ]
```

```
           cellLine replicateSet
K-562.B       K-562            B
MOLT-4.B     MOLT-4            B
CCRF-CEM.B CCRF-CEM            B
```

4 Save Rda File

Finally, we save the quantification matrix and the annotation information.

```
> save(novartisAll, novartisAllInfo, file = file.path("RDataObjects",
+     "novartisAll.Rda"))
```

5 Appendix

5.1 File Location

```
> getwd()
```

```
[1] "/Users/kabagg/ReproRsch/WebSite"
```

5.2 Saves

5.3 SessionInfo

```
> sessionInfo()
```

```
R version 2.8.1 (2008-12-22)
i386-apple-darwin8.11.1

locale:
en_US.UTF-8/en_US.UTF-8/C/C/en_US.UTF-8/en_US.UTF-8

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

other attached packages:
[1] XML_2.3-0
```

References

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