Affymetrix Quality Assessment and Analysis Tool

Xiwei Wu and Xuejun Arthur Li

October 22, 2008

1 Introduction

Affymetrix GeneChip is a commonly used tool to study gene expression profiles. The purpose of this package is to provide a comprehensive and easy-to-use tool for quality assessment and to identify differentially expressed genes in the Affymetrix gene expression data. Initial data quality assssment is achieved by a series of QC plots in an HTML report for easy visualization. More importantly, functions are provided for biologists who have little statistical background to generate design and contrast matrices for simple, as well as complicated, designed experiments, such as one factor with multiple levels, multiple factors with interactions, or one or more factors with covariates. Users can select either an ordinary linear regression model, LIMMA [1], or permutation test for differentially expressed gene identification. Differentially expressed genes are reported in tabular format with annotations hyperlinked to online biological databases. Wrapper functions are also designed to make analysis even more simplified. This guide will use an example dataset to demonstrate how to perform analysis of experiments with commonly used designs by using this package.

2 Data

We will use the dataset `estrogen` (8 Affymetrix genechips) downloaded from the Bioconductor that includes 12,625 genes to demonstrate how to use this vignette. This dataset has also been used as example data in the `factDesign` and `LIMMA` vignette. The investigators are interested in the effect of estrogen on the genes in ER+ breast cancer cells and how they differ across different time periods. In this example, there are two time periods, 10 hours and 48 hours.

```
> library(AffyExpress)
> library(estrogen)
> datadir <- system.file("extdata", package = "estrogen")
> phenoD <- read.AnnotatedDataFrame("phenoData.txt", path = datadir,
+     sep = "", header = TRUE, row.names = "filename")
> raw <- ReadAffy(filenames = rownames(pData(phenoD)), phenoData = phenoD,
+     celfile.path = datadir)
> pData(raw)
```

```
             estrogen time.h
low10-1.cel    absent     10
low10-2.cel    absent     10
high10-1.cel  present     10
high10-2.cel  present     10
low48-1.cel    absent     48
low48-2.cel    absent     48
high48-1.cel  present     48
high48-2.cel  present     48
```

It is very important to have the correct phenotype before doing further analysis.

3 Quality Assessment

To run the quality assessment, run the following function

```
> AffyQA(parameters = c("estrogen", "time.h"), raw = raw)

[1] "Retrieving grouping parameters"

 *** Output redirected to directory:  /tmp/RtmpOleu6C
 *** Use HTMLStop() to end redirection.[1] "Generating RNA digestion plot..."
Background correcting
Normalizing
Calculating Expression
[1] "Generating sample hierarchical clustering plot..."
[1] "Generating pseudo-chip images..."
[1] "/tmp/RtmpOleu6C/AffyQA.html_main.html"
```

The `AffyQA` function will create a quality assessment report in AffyQA.html that contains a set of assessment plots, including Affymetrix recommended quality assessment, RNA quality assessment, sample quality assessment, quality diagnostic using affyPLM (pseudo-chip images and NUSE and RLE plots) in your current working directory.

4 Preprocessing and Filtering

We can use the `pre.process` function to convert the `AffyBatch` data set into an `ExpressionSet` using either `RMA` or `GCRMA` methods. Suppose that we are using the `RMA` method.

```
> normaldata <- pre.process(method = "rma", raw = raw)

Background correcting
Normalizing
Calculating Expression

> dims(normaldata)
```

```
         exprs
Features 12625
Samples      8
```

The next step, we will filter the normalized data by using the `Filter` function. Suppose that we would like to filter the data based on at least 2 of the chips whose expression value is greater than 6.

```
> filtered <- Filter(object = normaldata, numChip = 2, bg = 6)
```

```
[1] "After Filtering, N = 9038"
```

Now, we have 9038 genes left. The examples below will be based on these 9038 genes.

5 Identifying Differentially Expressed Genes

Identifying differentially expressed genes depends on a researcher's interest and applying correct statistical models during the analysis process. We will illustrate a few basic statistical models on this data set.

5.1 Single Factor

Suppose we would like to identify how differentially expressed genes respond to estrogen regardless of time period. Analysis on a single categorical variable is the same as One Way ANOVA. Since we only have two levels, `present` and `absent`, for the `estrogen` variable, this type of analysis is also equivalent to a two-sample t test.

5.1.1 Design Matrix and Contrast Matrix

To run the analysis, we need to create a design and a contrast matrix. One of the major strengths of this package that we can use the built-in function to create the design matrix and the contrast matrix using standard statistics approaches which is different from the design matrix from the `LIMMA` package. To create a design matrix, we will use the `make.design` function.

```
> design <- make.design(target = pData(filtered), cov = "estrogen")
> design
```

```
  (Intercept) estrogen/present
1           1                0
2           1                0
3           1                1
4           1                1
5           1                0
6           1                0
7           1                1
```

```
8           1                1
attr(,"assign")
[1] 0 1
attr(,"contrasts")
attr(,"contrasts")$estrogen
[1] "contr.treatment"
```

Notice that the name of the second column of the design matrix is `estrogen/present`, where `estrogen` is the name of the variable and `present` tells us that `present` corresponds to 1. Thus, the design matrix above is equvalent to the equation below:

$$y = \alpha + \beta_E x_E + \epsilon \tag{1}$$

where

$$x_E = \begin{cases} 1 & \text{if estrogen = "present"} \\ 0 & \text{if estrogen = "absent"} \end{cases}$$

Next we need to create a contrast matrix. Since we are comparing `present` versus `absent`, we will create the following contrast:

```
> contrast <- make.contrast(design.matrix = design, compare1 = "present",
+     compare2 = "absent")
> contrast

     [,1]
[1,]    0
[2,]    1
```

5.1.2 Analysis

Once the design matrix and contrast matrix are created, we can run the analysis by using the `regress` function. There are three types of regression methods that are being supported: `LIMMA` (computing moderated t-statistics and log-odds of differential expressions by empirical Bayes shrinkage of the standard errors towards a common value), permutation test (resampling the phenotype), and ordinary linear regression. Also, we can apply multiple comparison corrections by using the `adj` option, such as `fdr`. The default value for the `adj` is `none`

```
> result <- regress(object = filtered, design = design, contrast = contrast,
+     method = "L", adj = "fdr")
```

Here are the first three genes of the result

```
> result[1:3, ]

                ID Log2Ratio.1         F   P.Value adj.P.Val
1000_at     1000_at -0.23625810 2.9566239 0.1195380 0.4969571
1001_at     1001_at  0.12495314 1.0542776 0.3312475 0.7979252
1003_s_at 1003_s_at  0.06103375 0.2581607 0.6235681 0.9536450
```

For the next step, we can select differentaly expressed genes based on p value and/or fold change. Suppose that we would like to select genes with a p value <0.05 and a fold change value greater than 1.5.

```
> select <- select.sig.gene(top.table = result, p.value = 0.05,
+     m.value = log2(1.5))

[1] "There are 381 differentially expressed genes"
[1] "based on your selection criteria."
```

The `select.sig.gene` function adds an additional column, `significant`, which gives a value of either TRUE or FALSE indicating whether the gene is differentially expressed based on your selection criteria. In this example, there are 381 differentially expressed genes being selected.

5.1.3 Output Your Result

To output the differentially expressed genes along with annotations to an HTML file in your current working directory, we can use the `result2html` function.

```
> result2html(cdf.name = annotation(filtered), result = select,
+     filename = "singleFactor")
```

5.1.4 A Wrapper Function

There is a wrapper function, `AffyRegress`, that can acomplish all of the above steps together including: create a design and contrast matrix, run regression, select differentaly expressed genes, and output the differentally expressed gene to an html file.

```
> result.wrapper <- AffyRegress(normal.data = filtered, cov = "estrogen",
+     compare1 = "present", compare2 = "absent", method = "L",
+     adj = "fdr", p.value = 0.05, m.value = log2(1.5))

[1] "There are 381 differentially expressed genes"
[1] "based on your selection criteria."
```

5.2 Single Factor Adjusting for Covariates

We can analyze the single factor effect (`estrogen` in our example) and adjust for other covariates (`time.h`). This is not the researcher's interest. However, we will present this example only for illustration purposes. This statistical approach is the same as Randomized block design which is used to iosolate the variation due to the nuisance variable.

5.2.1 Design Matrix and Contrast Matrix

We can create the following design and contrast matrix.

```
> design.rb <- make.design(target = pData(filtered), cov = c("estrogen",
+     "time.h"))
> design.rb

  (Intercept) estrogen/present time.h/48
1           1                0         0
2           1                0         0
3           1                1         0
4           1                1         0
5           1                0         1
6           1                0         1
7           1                1         1
8           1                1         1
attr(,"assign")
[1] 0 1 2
attr(,"contrasts")
attr(,"contrasts")$estrogen
[1] "contr.treatment"

attr(,"contrasts")$time.h
[1] "contr.treatment"
```

The design matrix above is equvalent to the equation below:

$$y = \alpha + \beta_E x_E + \beta_T x_T + \epsilon \tag{2}$$

where

$$x_T = \begin{cases} 1 & \text{if time} = 48 \\ 0 & \text{if time} = 10 \end{cases}$$

```
> contrast.rb <- make.contrast(design.matrix = design.rb, compare1 = "present",
+     compare2 = "absent")
> contrast.rb

     [,1]
[1,]    0
[2,]    1
[3,]    0
```

Once we obtained the design and contrast matrix, we can use similar steps documented in section 5.1.2 to select differentially expressed genes. We can also use the wrapper function `AffyRegress` to complete all the steps at once.

5.3 Multifactor Analysis I

One possible interest is to examine the estrogen effect at either the 10 hour period or 48 hour period. One way of conducting the analysis is to subset the data set into two groups, with one containing data at the 10 hour period and the other containing data at the 48 hour period. Then we can use single-factor analysis

for each group. Instead of spliting the data into two groups, we can use a more complex model like the one below:

$$y = \alpha + \beta_E x_E + \beta_T x_T + \beta_{E:T} x_E x_T + \epsilon \tag{3}$$

The interaction term between estrogen and time allows us to analyze the estrogen effect at different time periods.

5.3.1 Design Matrix and Contrast Matrix

We will first create the following design matrix, which is equivalent to the model above.

```
> design.int <- make.design(pData(filtered), cov = c("estrogen",
+     "time.h"), int = c(1, 2))
> design.int
```

```
  (Intercept) estrogen/present time.h/48 estrogen/present:time.h/48
1           1                0         0                           0
2           1                0         0                           0
3           1                1         0                           0
4           1                1         0                           0
5           1                0         1                           0
6           1                0         1                           0
7           1                1         1                           1
8           1                1         1                           1
attr(,"assign")
[1] 0 1 2 3
attr(,"contrasts")
attr(,"contrasts")$estrogen
[1] "contr.treatment"

attr(,"contrasts")$time.h
[1] "contr.treatment"
```

Suppose we are interested in the estrogen effect at the 10 hour period. We will create the follwoing contrast:

```
> contrast.10 <- make.contrast(design.matrix = design.int, compare1 = "present",
+     compare2 = "absent", level = "10")
> contrast.10
```

```
     [,1]
[1,]    0
[2,]    1
[3,]    0
[4,]    0
```

Similarly to creating a contrast matrix at the 48 hour period, we will do the following:

```
> contrast.48 <- make.contrast(design.matrix = design.int, compare1 = "present",
+     compare2 = "absent", level = "48")
> contrast.48
```

```
     [,1]
[1,]    0
[2,]    1
[3,]    0
[4,]    1
```

5.3.2 Analysis and Output Your Results

Next we can use the same `regress`, `select.sig.gene`, and `result2html` function to complete the rest of the analysis for each time period. However, the wrapper function `AffyRegress` will not work for this situation.

5.4 Multifactor Analysis II

A commmon interest of biologists is to identify how genes respond to estrogen treatments differently at different time points. In statistical jargon, this is a test for interaction. Interaction is a statistical term refering to a situation when the relationship between the outcome and the variable of the main interest differs at different levels of the extraneous variable.

5.4.1 Design Matrix and Contrast Matrix

Like before, we need to create the design and contrast matrix to detect the interaction effect. The design matrix is the same as the one we just created `design.int`. However, we will create a new contrast in order to test the interaction effect.

```
> contrast.int <- make.contrast(design.int, interaction = TRUE)
> contrast.int
```

```
     [,1]
[1,]    0
[2,]    0
[3,]    0
[4,]    1
```

5.4.2 Identify Genes with Interaction Effect

We will use the same `regress` function to detect which genes have the interaction effect.

```
> result.int <- regress(object = filtered, design = design.int,
+     contrast = contrast.int, method = "L")
```

Suppose we would like to select genes having the interaction effect based on a p value <0.05, and fold change >1.5. Note the fold change here means the difference of the fold change of estrogen duringthe 48 hour period and the fold change of estrogen during the 10 hour period.

```
> select.int <- select.sig.gene(result.int, m.value = log2(1.5),
+     p.value = 0.05)

[1] "There are 224 differentially expressed genes"
[1] "based on your selection criteria."
```

There are 224 genes that are significant. That means among these 224 genes, the fold change from `absent` and `present` differ in the two time periods.

```
> sig.ID <- select.int$ID[select.int$significant == TRUE]
> sig.index <- match(sig.ID, rownames(exprs(filtered)))
```

The variable `sig.index` contains the column index of significant genes in the `ExpressionSet`.

5.4.3 Analysis on Genes with the Interaction Effect

For this group of genes, we can use the `post.interaction` function to analyze how we can further explore how genes are expressed differently at different time points. Since `time.h` has two factors, the data type returned from this function is `list` with length equaling two. Each component of the list has the same formatted table returned from the `gene.select` function.

```
> result2 <- post.interaction(strata.var = "time.h", compare1 = "present",
+     compare2 = "absent", design.int = design.int, object = filtered[sig.index,
+         ], method = "L", adj = "fdr", p.value = 0.05, m.value = log2(1.5))

[1] "At Level-10:"
[1] "There are 89 differentially expressed genes"
[1] "based on your selection criteria."
[1] "At Level-48:"
[1] "There are 156 differentially expressed genes"
[1] "based on your selection criteria."
```

Among these 224 genes, 89 are differently expressed at the 10 hour period and 156 are differently expressed at the 48 hour period.

Next, we can output the differentially expressed genes to an HTML file by using the `interaction.result2html` function. This HTML file is similar to the one created by `result2html`. It contains the `Log2 ratio` and the `P value` for each time period. In the last column of this file, it contains the `P value` for the interaction effect.

```
> interaction.result2html(cdf.name = annotation(filtered), result = result2,
+     inter.result = result.int)
```

5.4.4 Analysis on Genes without the Interaction Effect

For genes without the interaction effect, which means that they respond to estrogen treatment similarly between the two time points, we can use the same design and contrast matrix from section 5.1.1 to detect estrogen effect.

```
> result1 <- regress(object = filtered[-sig.index, ], design = design,
+     contrast = contrast, method = "L", adj = "fdr")
> select <- select.sig.gene(top.table = result1, p.value = 0.05,
+     m.value = log2(1.5))

[1] "There are 309 differentially expressed genes"
[1] "based on your selection criteria."
```

5.4.5 A Wrapper Function

The entire process above can also be acomplished by a wrapper function, `AffyInteraction`. This wrapper function will create a design matrix and contrast matrix for the interaction test. Then it tests for an interaction effect for each gene and identifies genes whose interaction test is significant. For genes with interaction effect, they'll fit a linear model for each gene in each time period. For genes that don't have interaction effect, they'll fit a linear model for each gene without splitting the data by time period. It will output signficant results in the end.

```
> result3 <- AffyInteraction(object = filtered, method = "L", main.var = "estrogen",
+     strata.var = "time.h", compare1 = "present", compare2 = "absent",
+     p.int = 0.05, m.int = log2(1.5), adj.int = "none", p.value = 0.05,
+     m.value = log2(1.5), adj = "fdr")

[1] "For interaction test:"
[1] "There are 224 differentially expressed genes"
[1] "based on your selection criteria."
[1] "-----------------------------------------------------"
[1] "For genes without interaction effect:"
[1] "There are 309 differentially expressed genes"
[1] "based on your selection criteria."
[1] "-----------------------------------------------------"
[1] "For genes with interaction effect:"
[1] "At Level-10:"
[1] "There are 89 differentially expressed genes"
[1] "based on your selection criteria."
[1] "At Level-48:"
[1] "There are 156 differentially expressed genes"
[1] "based on your selection criteria."
```

References

[1] Smyth, G. K. (2005). Limma: linear models for microarray data. *Bioinfomatics and Computational biology Solutions using R and Bioconductor*, R. Gentleman, V. Carey, S. Dudoit, R. Irizarry, W. Huber (eds.), Springer, New York