Class Comparison with OOMPA

Kevin R. Coombes

January 4, 2007

Contents

1 Introduction

OOMPA is a suite of object-oriented tools for processing and analyzing large biological data sets, such as those arising from mRNA expression microarrays or mass spectrometry proteomics. The *ClassComparison* package in OOMPA provides tools to work on the "class comparison" problem. Class comparison is one of the three primary types of applictions of microarrays described by Richard Simon and colleagues. The point os these problems is to identify genes that behave differently in known classes; in other words, a typical class comnparison problem is to find the genes that are differentially expressed between two types of samples.

2 Getting Started

No one will be surprised to learn that we start by loading the package into the current R session:

```
> library(ClassComparison)
```

For the examples in this vignette, we will use simulated data that represents different groups of samples:

```
> nGenes <- 5000
> nSamp <- 15
> nDif <- 150
> delta <- 1
> fake.class <- factor(rep(c("A", "B"), each = nSamp))
> fake.data <- matrix(rnorm(nGenes * nSamp * 2), nrow = nGenes,
+     ncol = 2 * nSamp)
> fake.data[1:nDif, 1:nSamp] <- fake.data[1:nDif, 1:nSamp] + delta
> fake.data[(nDif + 1):(2 * nDif), 1:nSamp] <- fake.data[(nDif +
+     1):(2 * nDif), 1:nSamp] - delta
```

3 Gene-by-gene t-tests

The simplest weay to find differentially expressed genes is to perform a two-sample t-test on each gene. The `MultiTtest` class handles this operation, with a summary that carefully ensures that you know which class is associated with a positive t-statistic.

```
> mtt <- MultiTtest(fake.data, fake.class)
> summary(mtt)

Row-by-row two-sample t-tests with 5000 rows
Positive sign indicates an increase in class: A

Call: MultiTtest(data = fake.data, classes = fake.class)

T-statistics:
      Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
-7.1270000 -0.7330000 -0.0007713 -0.0143000  0.7303000  5.4240000

P-values:
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
9.362e-08 2.022e-01 4.700e-01 4.730e-01 7.384e-01 1.000e+00
```

4 Beta-uniform mixture models for multiple testing

As everyone now knows, an inherent difficulty with performing a separate test for each gene is that the p-values must be adjusted to account for multiple testing. A simple approach models the set of p-values using a beta-uniform mixture (BUM). We can perform this analysis quite simply:

```
> bum <- Bum(mtt@p.values)
> summary(bum)
```

```
> hist(mtt, breaks = 101)
> plot(mtt)
> plot(mtt, mtt@p.values)
```

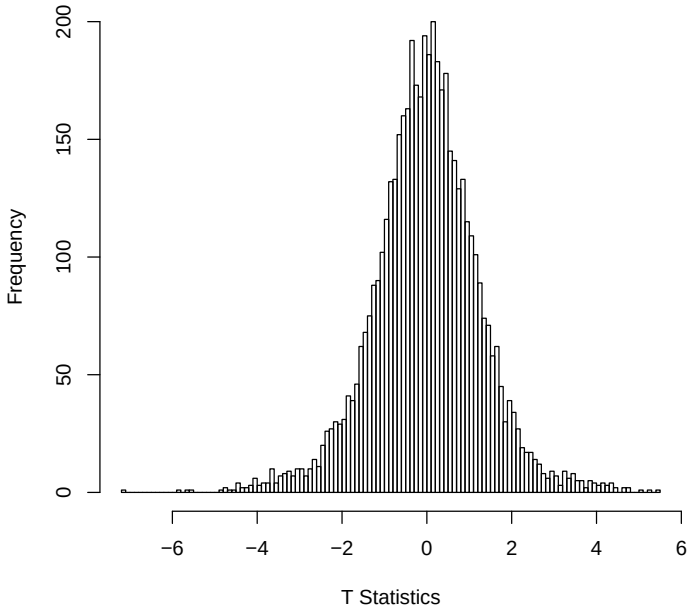

Figure 1: Histogram of the gene-by-gene two-sample t-statistics

```
Beta-Uniform Mixture Model

MLE Estimates: ahat = 0.36619 , lhat = 0.84058
Upper Bound on Fraction Unchanged: pihat = 0.89896

   tau          TP          FN           FP        TN
1 0.01 0.02893862 0.07210223 0.008989592 0.8899696
```

The default value of the `summary` command is not very enlightening, but we can get a graphical overview of the distribution. The region below the horizontal blue line in Figure 3 represents the uniform component of the mixture (i.e., genes that are not differentially expressed); the region between the blue line and the green curve represents the beta component (i.e., genes that are differentially expressed). If we set a threshold for significance using some cutoff on the p-value (such as the one indicated by the vertical purple line in Figure 3), then we can divide the area into four regions represnting true positives, false positives, true negatives, and false negatives. These areas can then be used to estimate the false discovery rate (FDR) as a function of the threshold (Figure **??**).

The usual application of this idea is to choose a threshold that achieves a desired level of FDR. For example, selecting genes with a p-value less than

```
> cutoffSignificant(bum, alpha = 0.1, by = "FDR")

[1] 0.002015007
```

should keep the FDR less than 10%. The number of such genes is easily obtained with the command:

```
> countSignificant(bum, alpha = 0.1, by = "FDR")

[1] 108
```

You can also get a logical vector that selects the significant genes:

```
> selected <- selectSignificant(bum, alpha = 0.1, by = "FDR")
```

In our example, the truly significant genes are iamong the first 300 genes. We can use this information to find out how close we are to the truth; the achieved FDR in this simulated example is pretty close to the target value of 10%.

```
> truth <- rep(FALSE, nGenes)
> truth[1:(2 * nDif)] <- TRUE
> sum(selected & truth)

[1] 99

> mean(!truth[selected])

[1] 0.08333333
```

```
> hist(bum)
> abline(v = 0.05, col = "purple", lwd = 2)
```

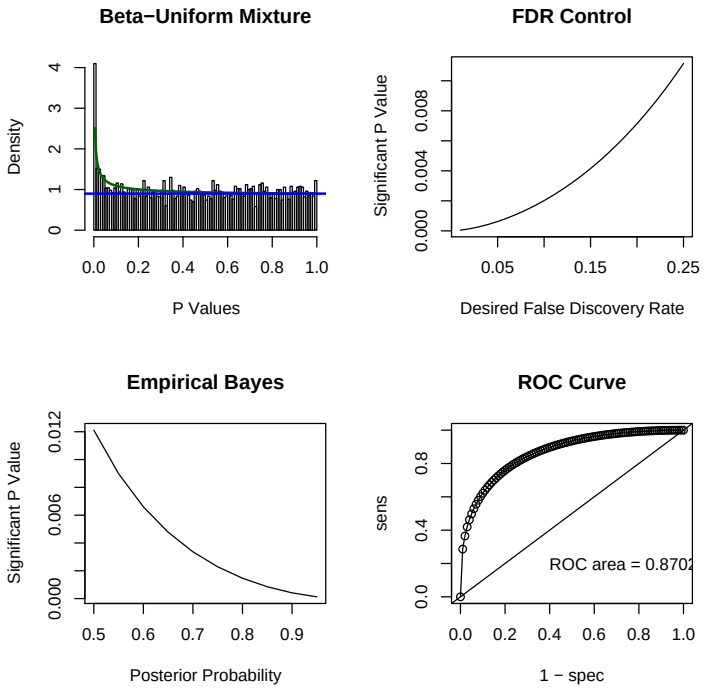

Figure 2: Results of the BUM analysis of the p-values.

```
> image(bum)
```

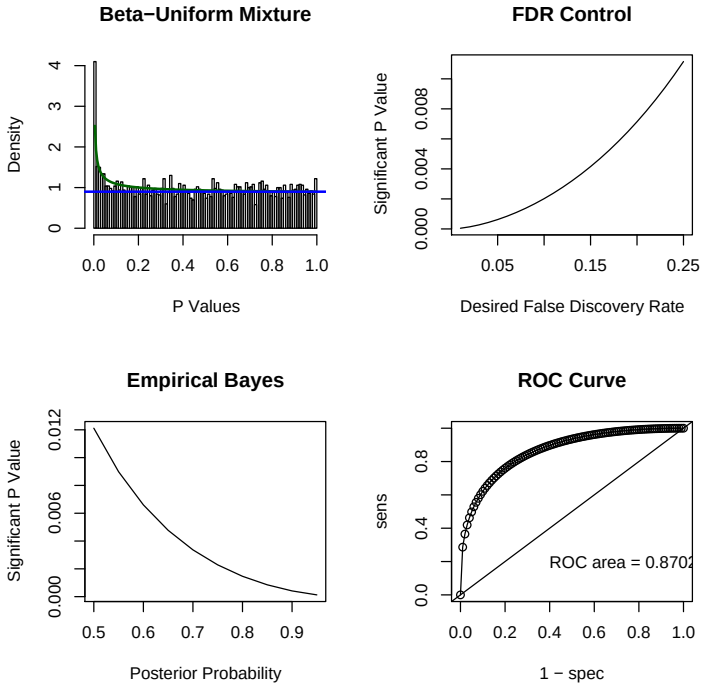

Figure 3: Results of the BUM analysis of the p-values.

5 Wilcoxon rank sum tests and empirical Bayes

In many applications of microarrays, it is unclear how trhe data should be transofrmed to achieve the approximate normality needed to justify a t-test. It may just be simpler to ignore the transformation problem and use nonparametric methods, like the Wilcoxon rank-sum test, that only use the ranks of the samples for the expression of each gene.

```
> mw <- MultiWilcoxonTest(fake.data, fake.class)
> summary(mw)

Call: MultiWilcoxonTest(data = fake.data, classes = fake.class)
Row-by-row Wilcoxon rank-sum tests with 5000 rows

Rank-sum statistics:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  126.0   215.0   232.0   232.3   251.0   330.0

Large values indicate an increase in class: A

With prior = 1 and alpha = 0.9
        the upper tail contains 22 values above 312
        the lower tail contains 28 values below 154
```

A histogram (Figure 4) of the Wilcoxon statistics indicates that the observed values have larger tails than expected by chance, suggesting that we ought to be able to pick out some genes that are significantly different. To do this, we use an empirical Bayes method originally suggested by Efron and Tibshirani. The idea is that we can decompose the WIlcoxon statistics as a mixture of those that arise from the null distribution (which is Wilcoxon with parameters based on the number of samples in each group) and some othe component represnting the differentially edxpressed genes. In that case, we can write the observed distribution $f(x)$ in the form:

$$f(x) = \pi f_0(x) + (1 - \pi) f_1(x)$$

where $f_0(x)$ is the known Wilcoxon distribution and $f_1(x)$ is unknown. Since we can estimate $f(x)$ from the opbserved data, we can simply solve for the unknown distribution $f_1(x)$ provided we know the mixing parameter π, which represents the prior probability that a gene is not differentially expressed. The "empirical" part of this empirical Bayes method comes down to selecting the prior π after looking at the data. For, if we start with $\pi = 1$, the posterior probability of being differntially expressed as a function of the observed statistic ends up taking on negative values (Figure 6), which is rather unpleasant.

By trial and error, we can find a value for π that ensures that the posterior probabilities are always positive (Figure **??**). In this case, something close to 0.94 works okay. We can then use a threshold on the posterior probabilities to set a significance cutoff on the Wilcoxon statistics.

```
> hist(mw)
```

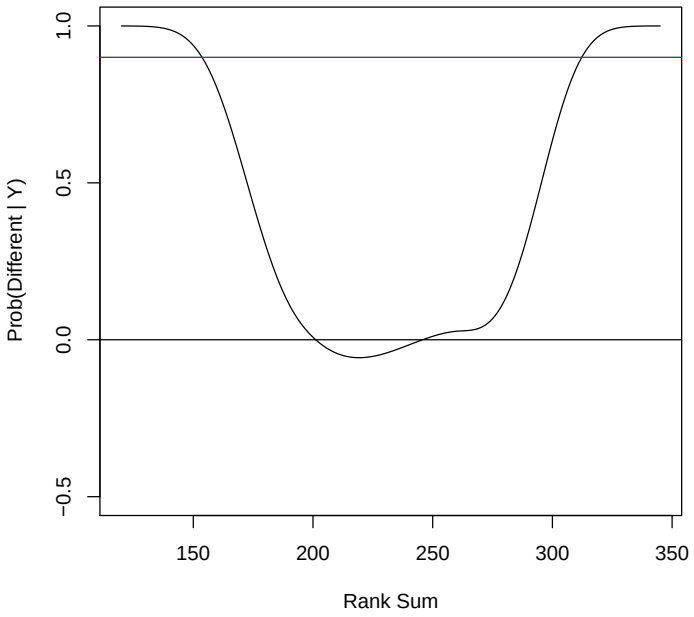

Figure 4: Histogram of the observed gene-by-gene Wilcoxon statistics.

```
> plot(mw)
> abline(h = 0)
```

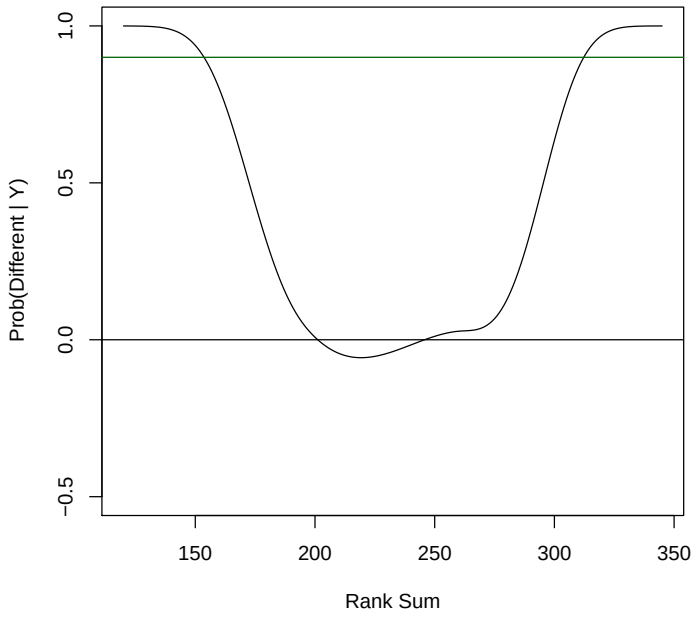

Figure 5: Plot of the posterior probability of being differentially expressed, assuming *a priori* that no gensd are different.

```
> plot(mw, prior = 0.94, signif = 0.9)
> abline(h = 0)
```

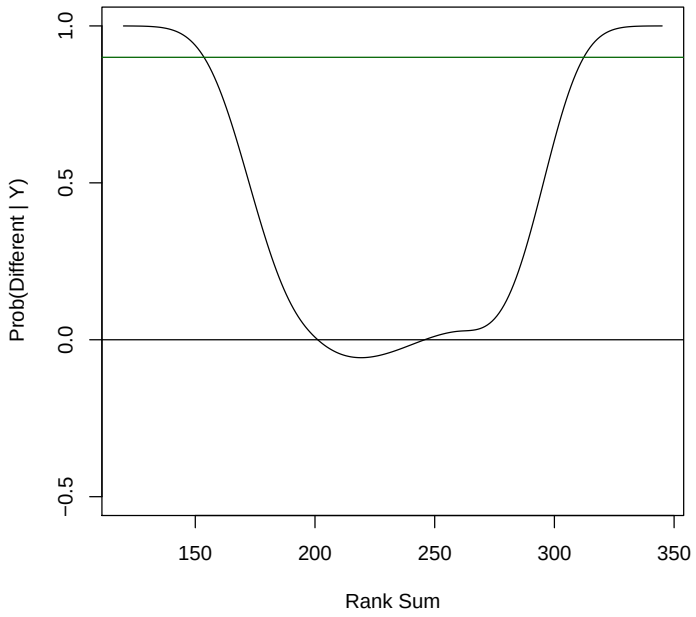

Figure 6: Plot of the posterior probability of being differentially expressed, assuming *a priori* that 94% of the genes are not different.

```
> cutoffSignificant(mw, prior = 0.94, signif = 0.8)

$low
[1] 161

$high
[1] 306

> countSignificant(mw, prior = 0.94, signif = 0.8)

[1] 92

> wilsel <- selectSignificant(mw, prior = 0.94, signif = 0.8)
> sum(selected & wilsel)

[1] 84

> sum(truth & wilsel)

[1] 84
```