

## Multiple Comparisons Exercise

In the world of high throughput molecular biology, we are frequently performing many statistical tests on a single data set. Read in a large set of Affymetrix data from a prostate cancer study.


Best CJM, Gillespie JW, Yi Y, Chandramouli GVR, Perlmutter MA, Gathright Y, Erickson HS, Georgevich L, Tangrea MA, Duray PH, Gonzalez S, Velasco A, Linehan WM, Matusik RJ, Price DK, Figg WD, Emmert-Buck MR and Chuaqui RF (2005) Molecular alterations in primary prostate cancer after androgen ablation therapy. *Clinical Cancer Research* **11**:6823–6834.

1. To load the data, add the folder containing the data to your Matlab path and use the command:

```
load prostatecancerarraydata
```

2. Use the variable editor to look at the data. What is its structure? The two variables in the MAT-file, `dependentData` and `independentData`, are two matrices of gene expression values from two experimental conditions (e.g. absence vs. presence of a drug), where each row corresponds to a gene and each column corresponds to a replicate. How many repeat measurements were performed? What is the third variable and what might you use it for?

3. Our goal is to look for differences in gene expression produced by the treatment. What is  $H_0$ ? We have now used two different "Monte Carlo" techniques to solve statistical problems. Spend some time in small groups and try to come up with a similar approach. Recall the general method of "breaking the association" between the treatment. Don't write any MATLAB code at this time—just outline a method that you think might work and create some "pseuo-code" in the form of comment lines.

4. *Homework exercise*: Take a look at the two MATLAB functions: `ttest_perm_uncommented` and `ttest_norm_uncommented`. See if you can figure out what it is they're doing. Add your own comments at each line that begins with a '%'. 

5. Of course, MATLAB has a built-in function that will do this much more efficiently: `ttest2`. Write a function that will perform this test for each gene and return a variable containing all of the p-values. Each p-value will reflect the likelihood that expression of the corresponding gene is affected by the parameter varied between the two experimental conditions. Look at the distribution of p-values, by generating a histogram. *How many genes show differential expression?*

See Step 2 of `MultipleComparisonsExercise.m` for help.

6. How many “significant” p-values would you expect if the data was randomly generated? Generate two arrays of random “data” with the same mean and standard deviation as the real data contained in the array `independentData`. (For help with this, see `DataRand`) Repeat the 2-sample t-test with these arrays, and compare the resulting distribution of p-values to the one from part 2.

7. If the probability density function is uniform, what does the cumulative density function look like? Plot the cdf of p-values. HINT: Use `'hist'` to do the heavy lifting, followed by `'cumsum'`. What's really nice

is that the slope of this line gives us an estimate of the *true* number of null hypotheses. *What should it be for the p-values that we generated? What do you get from your plot?*

See Step 3 of `MultipleComparisonsExercise.m` for help.

8. Create a p-value plot. (See T. Schweder, E. Spjøtvoll, *Biometrika* 69 (1982) 493-502, pdf included in class materials.) Schweder & Spjøtvoll used the above facts to devise a procedure for plotting p-values from a large number of simultaneous comparisons. For each P-value,  $p$ , obtained, one calculates the number of P-values in the set that are greater than that P-value,  $N_p$ , and then plots  $N_p$  vs.  $1 - p$ . From step 4 above you'll recognize that  $N_p$  is just the "area beyond" in the cumulative density of the P-value distribution. Using this convention, and plotting  $N_p$  versus  $1 - p$  (instead of  $p$ ), simply puts the interesting, extreme values (low  $p$ ) to the far right of the plot. *Write a MATLAB function to make this P-value plot.*

The p-value plot provides a way to estimate the proportion of genes tested that responded to the experimental manipulation.

See Step 4 of `MultipleComparisonsExercise.m` for help.

9. Perform the above analyses on some different microarray data:

```
load prostatecancerexpdata
```

See Step 5 of `MultipleComparisonsExercise.m` for help.

10. Use the function provided (`PvalPlot`) to estimate the true number of null hypotheses ( $T0 + 2 * T0sd$ ). Make a plot of the mean values of each gene expression level for all of the data in black, with *independentData* on the x-axis and *dependentData* on the y-axis. Then find the corresponding values with truly significant differences and plot their means on top in red. Label the value of the largest difference in the entire data set using the *probesetIDs* variable. *Why does the 'a' look funny? Why is the labeled point not red?*

See Step 6 of `MultipleComparisonsExercise.m` for help.

### Other tests: false discovery rates and "q"

The beautiful thing about the P-value plot is that it lets us estimate the rate of truly null features, based on the common sense notion that values to the left of the P-value plot (i.e. high P-values) consist almost exclusively of true H0s and the fact that P is distributed uniformly under H0. Once we have this estimate, we can calculate the "false discovery rate," or "q". While there are various methods to do this, they all rely on the fundamental insights contained in the P-value plot. If you are interested in pursuing this, below are the references behind the FDR tests performed in MATLAB's Bioinformatics Toolbox.

```
figure;  
[fdr, q] = mafdr(pvalues, 'showplot', true);
```

## References:

Storey JD (2002) A direct approach to false discovery rates. *Journal of the Royal Statistical Society* **64**:479–498.

Storey JD and Tibshirani R (2003) Statistical significance for genomewide studies. *Proc Nat Acad Sci USA* **100**:9440–9445.

Storey JD, Taylor JE, and Siegmund D (2004) Strong control conservative point estimation and simultaneous conservative consistency of false discovery rates: A unified approach. *Journal of the Royal Statistical Society* **66**:187–205.

Benjamini Y and Hochberg Y (1995) Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society* **57**:289–300.

## Revisions:

***RTB wrote it, 27 May 2012***

***MG (Miriam Ginzberg) adapted for exercise, 30 May 2012***

***RTB updated and added 'ttest\_perm' and 'ttest\_norm' exercises, 19 August 2012***

## Appendix. False Discovery Rate and the q-value:

	Called significant (H0 rejected)	Called not significant (H0 accepted)	<b>Total</b>
H0 True	<b>F</b> False Positive "False Alarm" Type I error	<b>m<sub>0</sub> - F</b> True Negative "Correct Rejection"	<b>m<sub>0</sub></b>
HA True (H0 False)	<b>T</b> True Positive "Hit"	<b>m<sub>A</sub> - T</b> False Negative "Miss" Type II error	<b>m<sub>A</sub></b>
<b>Total</b>	<b>S</b>	<b>m - S</b>	<b>m</b>

The table above helps us think about hypothesis testing in general. It gives all four possible outcomes of such a test, with the rows corresponding to the "ground truth" and the columns corresponding to how our test classifies the comparison. [**Note:** This same formalism is used in "Signal Detection Theory," which was developed by physicists and is now heavily used in the psychology of perception. I've included the corresponding terms in quotation marks.]

Most folks are comfortable with P-values, which give the probability of wrongly rejecting H0, also known as the *false positive rate (FPR)*. A related, but critically different, measure is called the *false discovery rate (FDR)*, which gives the proportion of tests labeled as "significant" when, in fact, H0 was true. In both cases, the numerator is the number of false positives, **F**, but, for the former, the denominator is the upper row total (**m<sub>0</sub>**), and, for the latter, it is the left column total (**S**).

$$\mathbf{FPR} = F / (F + m_0 - F) = F / m_0 \quad \mathbf{FDR} = F / (F + T) = F/S$$

Related values that are also used to discuss this issue:

$$sensitivity = T / m_A \quad specificity = (m_0 - F) / m_0$$

The Q-value is to the FDR what the P-value is to the FPR. It simply allows one to estimate what fraction of the comparisons that one has called "significantly different" is likely to be wrong. The derivation is relatively straightforward and can be found in Storey (2002) or Storey & Tibshirani (2003). It is worthwhile for all molecular biologists to become familiar with the basis of these statistics, since they are in wide use. As you saw in Step 10, it is easy to apply the various formulas (e.g. [mafdr](#)). However, doing so without understanding the underlying assumptions and their applicability to data obtained with a particular experimental method can lead to disastrous results.